European Commission
Information Society and Media

SEVENTH FRAMEWORK PROGRAMME

Large Scale Integrating Project

EXALTED

Expanding LTE for Devices

FP7 Contract Number: 258512

WP4 – End-to-End (E2E) M2M System

D 4.2
IP Networking System for M2M communications for EXALTED use cases

Contractual Date of Delivery:	30 June 2012
Actual Date of Delivery:	October 31st, 2012
Responsible Beneficiary:	CEA
Contributing Beneficiaries:	TST, CEA, EYU, TKS, VGSL
Estimated Person-Months:	46
Security:	Public
Nature	Deliverable
Version:	2.0

Document Information

Document ID: EXALTED_WP4_D4.2
Version Date: 31 October 2012
Total Number of Pages: 96

Authors

Name	Organisation	Email
Javier Valiño	TST	jvalino@tst-sistemas.es
Juan Rico	TST	jrico@tst-sistemas.es
David Garcés	TST	dgarces@tst-sistemas.es
Sofiane Imadali	CEA	sofiane.imadali@cea.fr
Dejan Drajic	EYU	dejan.drajic@ericsson.com
Aleksandar Obradović	TKS	aleksandarob@telekom.rs
Gorica Nikolić	TKS	gorican@telekom.rs
Arnaud Kaiser	CEA	arnaud.kaiser@cea.fr
Mohamed Reza Akhavan	VGSL	reza.akhavan@vodafone.com
Alexandru Petrescu	CEA	alexandru.petrescu@cea.fr

Document History

Version	Date	Modification
1.0	30 June 2012	First official version of document, as delivered to the European Commission.

2.0	12 October 2012	• 25 new pages.
		• Editorial changes:
		  ○ introduced a coherent figure numbering scheme, unique font family throughout running text, indentation, etc.
		  ○ added "List of Figures" at the beginning of document.
		  ○ homogeneous unique list of References (instead of 2).
		  ○ corrected various syntax and improved expression at places.
		• Substantial content addition and modifications:
		  ○ created new section "VIN - IP Address Translation Schemes" in the section "State of the Art for Address Translation Schemes".
		  ○ created new section "Address Translation Mechanism for Vehicular Communications" in the section "Mechanisms for M2M Communications for Use-Cases of EXALTED".
		  ○ created new sections "Requirements from communications in vehicular environments" and "Requirements for M2M IP Networking" in the section "Vehicular Network Use-Cases".
		  ○ created new section "VIN-Base Numeral System Specification" as an Appendix.
		  ○ significantly expanded the "Executive Summary" section to better describe the contents and motivate the relationship to project goals.
		  ○ significantly expanded the "Introduction" section for a clearer explanation, by e.g. creating new figures and answering various critiques from reviewers.

Executive Summary

The document describes technical work related to M2M IP Networking. The aim is the construction of an IP networking system being used in a setting where most devices are of machine class.

The document is divided in two parts: first, we present an analysis of existing mechanisms pertaining to machine-to-machine communications at the scale of the Internet: addressing, routing, and fundamental principles like “end-to-end”. Since the existing Internet architecture is not well adapted for machine-type communications, a number of additional aspects are described to play important roles: Future Internet architectures supporting mobility from the inception (clean slate designs), address translation mechanisms, low-power networks. As an application, we consider one particular use-case relevant to project EXALTED, which is represented by the vehicular communications. These aspects are described in detail in the first part of the document.

An IP networking system is needed for M2M for several reasons. The fundamental principles of the initial Internet design led to a widespread deployment – not only a commercial success but a cornerstone in the evolution of society. Naturally, it is expected that extending the use of these principles to the landscape of M2M communications can only be advantageous. For example, when an end-to-end connection from one machine-class device all the way to another machine-class device situated at a distance of planetary scale new interaction paradigms enriching existing applications are enabled; it is known that an “end-to-end” connection can be guaranteed solely by the use of TCP/IP technologies. Additionally, the widespread availability of TCP/IP implementations for platforms ranging from lower-end to the high end devices, as well as the plethora of communicating applications enabled with TCP/IP capabilities form the basis for building a comprehensive IP networking system for M2M. The M2M environments, if they are to evolve from today’s limited deployments, need to use IP technologies in order to replicate the initial success of applications running today on Internet.

The key challenges in designing an IP networking system for M2M communications can be divided in two categories: structural challenges (e.g. addressing and routing mechanisms) and functional challenges (protocol mechanisms). The addressing mechanisms used in the current and near-future Internet can be considered to be of the family of IPv6 (Internet Protocol version 6, an evolution of the older IPv4 whose space has been depleted in year 2011). On another hand, addressing mechanisms used at the smaller scale of M2M communications are specific particular applications (e.g. vehicular addressing using VIN, ZigBee addressing, etc.) It is a challenge to design a homogeneous addressing system which accommodates both addressing systems.

The functional challenges of designing an IP networking system for M2M communications are generated by the discrepancy between non-M2M protocols whose implementations run on generous hardware – the state machine of TCP, the large http requests, are all examples of implementations which can only work on systems with large memory available. Additionally, the characteristics of ultra-fast wired links (bandwidth in the order of several gigabytes per second common on PCs) and computers with clock speeds in the Giga-Hertz range slowly led to protocols exchanging large amounts of data in a single exchange. To quantify this aspect, one should compare the MTU (Maximum Transmission Unit) of tens of kilobytes of fast optical links to the hundreds of bytes of slow wireless links used in M2M. For these links and protocols and M2M systems, not only the message encoding should be more efficient but the inherent logic of protocol should allow for maximum functionality for the least amount of data exchanged.

The proposed solutions in this deliverable try to address these structural and functional challenges by offering address translation mechanisms as well as protocols for capillary-to-capillary-to-infrastructure IP communications, relying on the use of TCP/IP.

The main outcome is reflected in the second part of the document. We describe a number of mechanisms that we consider novel, pushing the limits of existing designs of protocols for machine-to-machine communications in the Internet: address formation and translation for 6lowpan and ZigBee, as well as a protocol for capillary-to-capillary-to-infrastructure communications as applied in a V2V2I scenario.

Table of Contents

1. Introduction

The features of an IP networking system for machine-to-machine vehicular communications in the Future Internet are described in this document.

Such a system should take into account, as much as possible, the fundamental principles of Internet design; the end-to-end argument, network of interconnected networks, packet-based data, layering, are all simple principles of existing Internet design. Or, the core functionalities of Internet are not well suited for use by machine-to-machine communications. For example, because placing the functionality in the network is reduced to a minimum (keep the network 'dumb'), each node in current Internet should have an address of its own; and, when these devices are very numerous, it means that addresses are very long. This leads to a paradoxical situation where end nodes communicating by sending packets may need to send much more description data (the address of source, of destination, etc.) than the data itself. A typical illustration of this effect is an IPv6 sensor sending presence data: the presence bit (the useful data) is much less than the packet containing it (the mandatory IPv6 base header is of length 320 bits). Hence, address translation mechanisms are necessary.

The system architecture of project EXALTED is specified to contain an LTE/LTE-M Access Network, a set of several capillary networks and an IP Core Network hosting the services. Within this context, the M2M IP Networking system is concerned mainly with interactions within and between capillary networks and between capillary networks and the LTE/LTE-M Access Network. This is illustrated in the figure below:

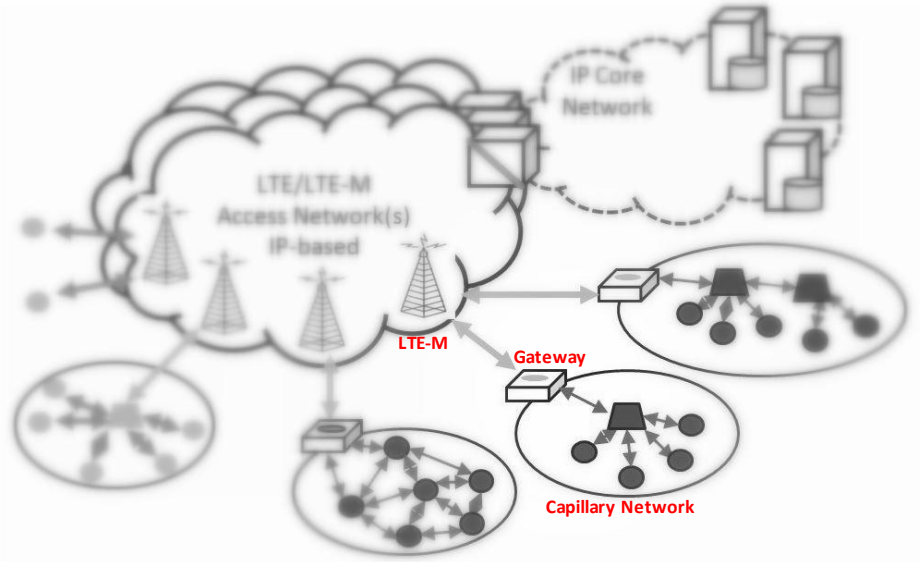


Figure 1: Components of M2M IP Networking System in the System Architecture of EXALTED

The protocols and mechanisms of an M2M IP Networking System are grouped in two main classes: the class of IP addressing, routing and autoconfiguration and the class of address translation and mapping. This is illustrated in the figure below:

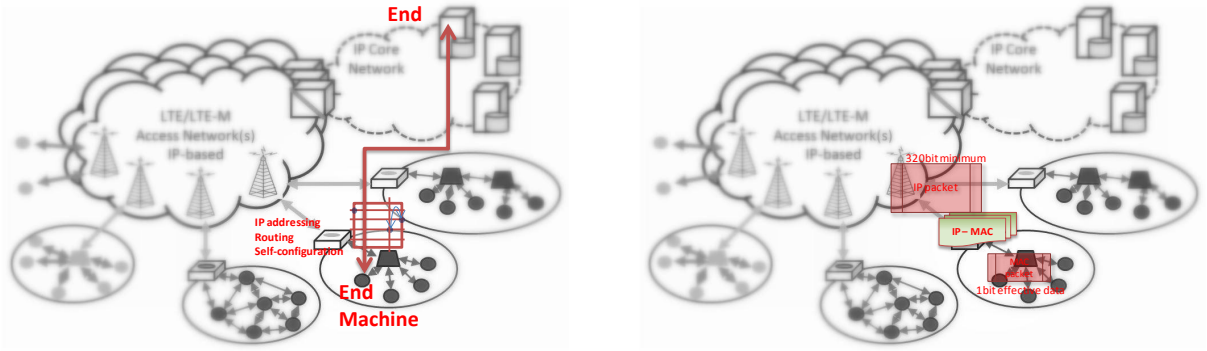


Figure 2: IP addressing/routing and translation schemes

In the figure above, the left hand diagram depicts E2E communications between an End Machine (new) and an End server (existing). These communications must be supported by novel addressing, routing and self-configuration schemes implemented by the M2M Gateway. On another hand, the right hand diagram depicts address translation mechanisms also implemented on the M2M Gateway; the 'MAC' addressing is an example; it helps to illustrate the need that exists to translate an IPv6 packet whose minimal length is 320bit into a packet whose useful data may be 1 single bit – the shorter the better for machine-class devices.

An additional obstacle to the use of existing IP technologies for machine-to-machine communications is represented by the relatively static nature of routes in the current Internet. One address being topologically correct at only one place in the Internet means that mobile devices may not be accommodated naturally (tunneling protocols like Mobile IP are used). And, since machine devices are often highly mobile (as in the case of vehicular communications), the mobility of capillary networks should be taken into account. Hence there may be a necessity to build a new Internet that is able to cope with the movements of not only large number of devices but also groupings of these devices (vehicles). A number of efforts have been undertaken to specify new protocols for Future Internet; they have been sorted in two main classes: clean-slate approaches (start from scratch) and evolutionary approaches (add new protocols on top of existing ones, or patch). The Host-Identity Protocol, SHIM6, MAST, LIN6, LISP, GSE and more are described in this document as existing protocols for Future Internet.

At the short range, what is typically called a wired Local Area Network of desktop computers is obviously not well adapted for machines communicating by a wireless lossy low-power network. A Machine-area Network would be a better name for such a network and the protocols that are more pertinent to this are situated in the family named '6LoWPAN'. This is a set of protocols recently standardized at IETF by the 6lowpan Working Group, sometimes relying on the IEEE 802.15.4 MAC layer. Within such a local network (which can be called 'capillary' network for conformance with EXALTED) mobility of devices is possible, and accommodated by protocols such as Mobile IPv6, Fast MIPv6 and Hierarchical MIPv6. Mobility of entire networks (NEMO) is possible also within 6lowpan and this is described. Further, MANET-based, RoLL-based and Neighbor Discovery protocols are presented as candidate solutions for mobility within capillary networks using 6lowpan.

Vehicular communications environments present challenges particularly difficult with respect to machine-to-machine communications. On one hand, novel machine-type devices (conforming to e.g. IEEE, 3GPP standards) are constructed outside the vehicular manufacturing industries whose experience is standardized at different standards development organizations. Addressing a vehicle is completely different than addressing an end node in the Internet. To fill this gap, there may exist a need to convert between

vehicular and Internet addressing systems. Additionally, the highly dynamic nature of vehicles movements are very hardly tackled by classical auto-configuration protocols such as Neighbor Discovery and DHCPv6. We present a novel mechanism to offer vehicle-to-vehicle-to-infrastructure communications which expands on the use of ND and DHCP protocols (extensions for both are presented).

As presented earlier, address translation mechanisms are necessary in order to deal with the following problems exhibited by machine-to-machine communications in the Future Internet:

- Communicating directly between machines within one capillary network unnecessarily consumes bandwidth when more descriptive data is sent rather than actual data.
- Low end devices often have very little storage memory and IPv6 stack implementation may overcome it.
- Some link layer protocols pertinent to IP communications explicitly remove the IP headers (header compression).

To cover this space of addressing problems we proceed by proposing two address translation mechanisms:

1) A ZigBee address translation mechanism allowing a server in the infrastructure to address a device within a ZigBee capillary network, without this latter to have an IP address.
2) A 6LoWPAN address translation mechanism that allows conversion between LTE-M access and a 6LoWPAN device.

Each of these three mechanisms is implemented by a Machine-to-Machine Gateway (which sits within and moves together with) the capillary network. In some cases, a peer device is needed in the infrastructure to perform the reverse translation.

Although not a specific focus of an IP networking system, the MAC protocols below IP may make huge differences when characteristics like low-power and lossy nature are taken into account. When a MAC protocol is present which offers capabilities to deal with these characteristics, it should be taken into account by specific IP protocols like RPL

2. Baseline for M2M IP Networking in M2M Future Internet, and Address Translation

The technologies developed in the landscape of M2M communications over LTE networks are necessarily using IP networking connectivity. On another hand, the TCP/IP protocols were designed at a time when the smallest computer was the size of an armchair and hence it followed different constraints than machine-class devices.

The fundamental characteristics that make M2M communications on LTE different are:

- Autoconfiguration mechanisms are of very high importance: small devices need not be attended and even less configured by human intervention.
- Small devices are by nature highly dynamic: their mobility is of paramount importance in the case of vehicles, for example. In the case of event surveillance networks the mobility is less important, but the dynamic nature of their connection follows naturally occurring events (and not human decisions) and hence their reliability is of high importance as well.
- Implementations should be able to be fit in small memory footprints running on computers cycled at very low (sometimes intermittent) speeds.
- The protocol exchanges should happen on links of very limited capacity (bandwidth, etc.)
- There should be a hierarchy of chained connections covering heterogeneous links with characteristics ranging from short-range (the first step out of Machine) to long-range (LTE connection).

2.1 State-of-the-Art for Addressing and Routing in Future Internet

After successfully connecting machines (in the 90s) and people later, the new era of Internet is about connecting things. Due to the increasing demands in terms of addresses, mobility, scalability, security and other new unattended challenges, the evolution of current Internet architecture is subject to major debate worldwide [2]. This section presents an overview of the main trends.

2.1.1 Fundamental building blocks

The current Internet architecture expansion has been driven by fundamental technical and non-technical principles. These requirements originated from different design objectives such as survivability, distribution of management, resource sharing and supporting different types of services. In the remainder of this section, we will describe these design principles and explain how they influenced the expansion of the Internet, and why these principles are made obsolete by the new challenges.

In EXALTED, it is considered that LTE connections are IP enabled. Thus, the entire set of fundamental building blocks which make up Internet need to be reused in EXALTED.

2.1.1.1 End-to-End Argument

In order to present the “end-to-end” argument, it is necessary to first acknowledge that the use of this acronym in different contexts means different things. In some contexts, “end-to-end” means simply to picture a layered protocol stack and state that each layer interacts with the corresponding layer on another computer (optionally “bridged” by intermediary less high stacks). Here, an “end-to-end” connection is made through a sequence of interfaces (APIs) from the highest layers (Application) to intermediary layers and all the way down to the physical layer (all within one stack).

In other contexts, “end-to-end” means that a packet is sent by a source, through a number of relays, to finally reach a destination (as opposed to communications not involving

intermediaries). There are as many interpretations of the acronym “end-to-end” as there are perspectives of what networking means. But, there is only one *argument.*

The End to End (E2E) argument [27][59] is one of the most cited of the Internet design principles. It states that mechanism should not be placed in the network if it can be placed at the end node, and that the core of the network should provide a general service, not one that is tailored to a specific application [27] [59]. One of the consequences of this approach is the design of a ‘dumb’ (sometimes called ‘stupid’) network and ‘smart’ (or ‘intelligent’) endpoints [31] [5]. The rise of the Internet, in the 90’s, has benefited from the widespread use of the PC and the cheap deployment of this dumb network was due to the smart endpoints that would use it eventually. This migration of intelligence toward the edges led to the concentration of administration and maintenance in the edges also [5].

The main advantage of the end-to-end approach is innovation. The deployment of various applications is due to the simplicity of the Internet and its very general purpose design and objective (carry a set of bits). Another advantage that arose from the E2E principle is the reliability of applications as long as the network stays simple [27].

The E2E is not an absolute rule but rather a guideline for application and protocol design analysis [59]. Mail delivery system, where users send their mails to mail servers (SMTP) rather than endpoints, is one example where the E2E principle does not apply.

In today’s Internet, other mechanisms are clouding up our vision of the entire system and challenging the E2E concept. If encryption was the E2E principle designers answer to the security concerns [59], deploying firewalls at the network boundaries is much more common these days. Firewalls break the E2E model, and change the nature of the Internet which is less transparent and no longer trusted [27]. Network Address Translation (NAT) mechanism is another technical concept breaking the E2E design of the early Internet. NAT mechanism is the answer to the IPv4 addressing space shortage, privacy concerns and private address space management [112] [40]. Due to NAT Boxes, the non-mutability characteristic of the IP address, that is the source and destination addresses sent in a packet are those received by the destination, is no longer valid. The same goes for the omniscience of an IP address, that is each host knows what address a peer host could use to send packets to it [88] [79].

Revisiting the E2E concept and redefining it is an ongoing tussle [27] between those who want to enhance their applications with more functionality and reliability and those who want to preserve the simplicity and transparency that made the success of the Internet.

2.1.1.2 Network of interconnected networks

In the early days of the Internet, the major concern of the DARPA Internet Architecture was the development of effective techniques to interconnect and use already interconnected networks [30]. The interconnection of the packet radio network [91] with the already existing ARPANET in the late 70’s was a major achievement in this context. The goal was to access services offered by the ARPANET servers (measurements and analysis).
fig
The Internet’s original components are networks, and one main design objective is to interconnect them in order to provide a larger service. In this, we can say that the Internet was built from down to top. The alternative top-down design would have been a unified large system incorporating the needed technologies and modular enough to allow extensions for unattended applications; an impossible task.

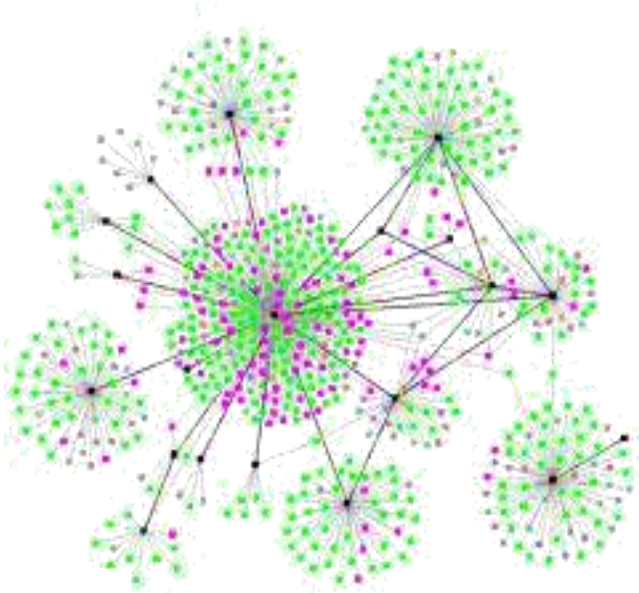


Figure 3: Network of networks. The overall picture represents a decentralized network (Source: Scale-free networks)

The universality of the Internet is also due to the universality of the IP layer [88] that runs on top of (almost) any technology and allows interactions between heterogeneous technologies like Ethernet, X.25, FDDI, Cellular, modem and other communication technology standards. The wide use of IP is clearly one of the reasons of the IPv4 addressing space shortage. The advent of the IPv6 with its huge addressing space (296 times bigger) will certainly encourage other technologies to consider merging with the Internet, using Address Translation Gateways, speaking IPv6 on their egress interface and some other technology (802.15.4, for example) on the ingress interface. Note that the initial meaning of E2E principle is changing, as the gateways are responsible of managing translation tables between nodes IDs in the non IP technology part of the network and IPv6 addresses for these same nodes. This mapping is essential for maintaining E2E communication sessions (as in 6LowPAN). This is a broad scope problem faced by the Internet of things [62].

A high-level overview of the Internet shows that it can be broken down into a set of Autonomous Systems (ASes) each composed of multiple routers organized into collaborating networks. The routing decisions are taken based on a routing table at each router calculated in a distributed manner: within an AS, interior gateway protocols (IS-IS and OSPF) are used and exterior gateway protocols (BGP) between two (or more) ASes [2]. This distributed design which continues to provide communications service, even when networks and gateways are failing (survivability) is a military context legacy [30].

Along with the survivability objective, the down to top design of the Internet allowed to achieve distributed management of its resources and to support multiple types of services. These design goals have strongly shaped the Internet as we know it [2] [30].

2.1.1.3 Packets as the basic unit of data exchange

The datagram is a self-describing packet containing (mainly, but not only) an invariant source and destination IP addresses (non-mutability characteristic of IP addresses), a source and destination port numbers and a data payload.

A shortest path between the source and the destination addresses is selected in a distributed manner (no coordination between routers) in order to carry the packet to the destination host. The destination port number is used within the host to deliver the payload to the right

application. Delivering packets is then a two-phased dispatch operation: First, between nodes on an IP-layer decision basis, second, within the node on a port-ID decision basis [29].

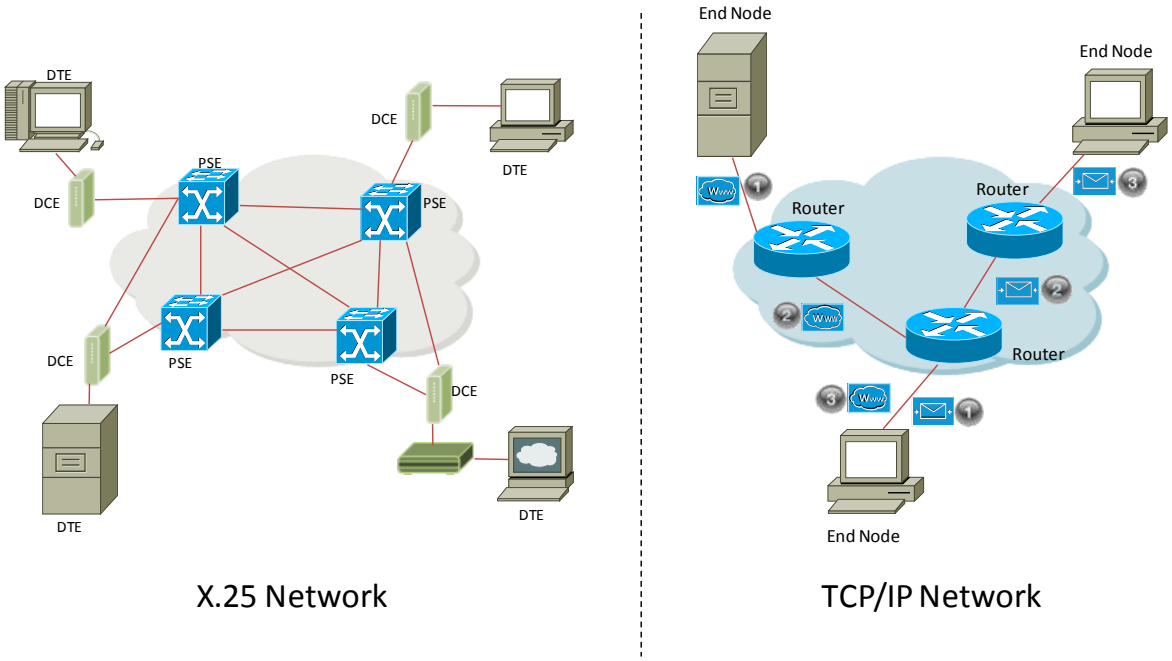


Figure 4: A comparison between virtual circuit-switched networks and TCP/IP networks.

Experience has proven that the datagrams for universal fine-grained statistical multiplexing is the right data exchange model to apply in large heterogeneous networks, especially for bursty and intermittent traffic [28]. The best effort model and QoS mechanisms are direct consequences of this design choice.

A global stateless routing system is another important design objective achieved by packet switching. There is no connection state saved within the intermediate switching nodes (routers) and thus, after a failure, these nodes can recover without concerns about state. Only endpoints will save the current state information of communication sessions (namely, TCP); when failing (the session), this is highly likely due to host failure, what is often referred to as “fate-sharing” [30].

2.1.1.4 Layering

The network layering model (or vertical integration) has various advantages as reduction of complexity, isolation of functionality and a unified model for designing network protocols. These layers, during a communication session between two (or more) hosts, show a bilateral agreement (logical communication) between the endpoints. The network layer, IP, is the only layer requiring universal agreement [108]. The Internet, in the TCP/IP layered model, has five layers, from top to down, application, transport (known as upper layers), network (IP), link and physical (bottom layers). The upper and bottom layers experience frequent and rapid innovations, whereas the network layer is difficult to evolve as it implies a universal change. This state is sometimes referred to as ossification [17]. The IP layer, for its simplicity and capacity to run on top of (almost) any technology, is the main reason for the Internet’s success.

Another view of the Internet protocol architecture [18] shows the protocol stack as an hourglass where the IP is the common waist between all IP-capable nodes regardless of the communication technology used in lower layers ad applications above. This is what enabled the integration of heterogeneous network technologies into the global Internet [2].

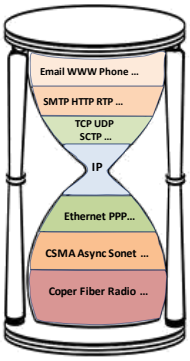


Figure 5: The hourglass model of the TCP/IP stack

The IP layer comes with a universal IP numbering space that allows identifying every IP-capable unambiguously. The IP address is carried on each packet sent and according to the original Internet design (E2E principle) the address is unchanged across the network towards the destination [71]. The advent of private addressing space ([112]) as an answer to the IPv4 address space shortage and the use of NAT Boxes have changed the classical network-layer addressing characteristics. IP addresses are now ephemeral, non-unique and a same node (its interface) can be assigned a different address each time it connects to the network, even if it does so from the same location.

Another issue related to the layering mis-specification, is the semantic overload of the IP address. The TCP/IP had been a single protocol in the architecture, originally, but the desire to provide another type of transport service (namely, UDP) caused the separation of the two protocols into a network-layer and a transport layer [30]. The 2006 IAB Workshop on Routing & Addressing [23] has clearly pointed to the overloading of IP address semantics as one of the major causes of the scalability problems experienced in the Default Free Zone (DFZ). The addressing has a “who” significance (endpoint ID at the transport layer) and “where” semantics (locators for the routing system). This is what recent studies qualify as the “locator/identifier overload” of the IP address. Different approaches aim at splitting (separating) both functionalities. Note that some solutions answer to the problematic with the addition of new layer between transport and networks [88] [34] while others try to specify two separate spaces: one for identification and one for location [19] [22], and some have tried to redefine the IP numbering space [70].

According to [53], the only addressing problem that interested the TCP/IP protocol designers was the width of the IP numbering space. Despite seminal works on naming and addressing [58] [56], it seems that the same problematic is facing the todays Internet protocol design, especially with the new challenges of mobility and multihoming.

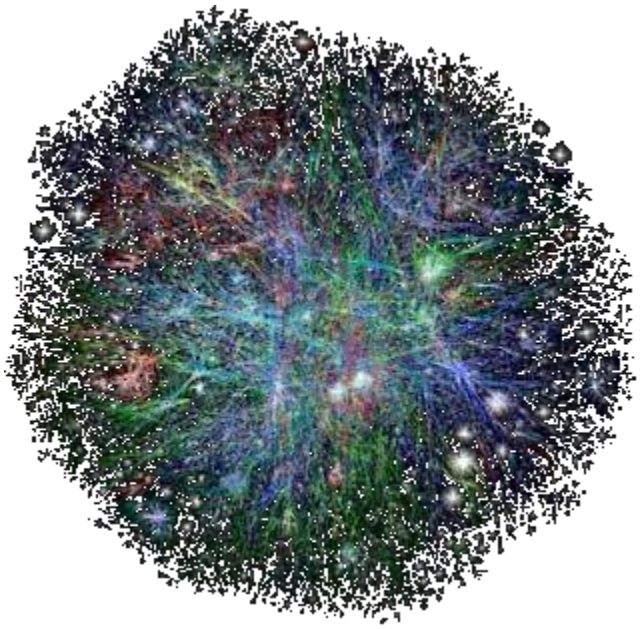


Figure 6: Internet map as of 16th January 2009 (Source: Cisco IBSG, 2006-2011, Guo-Qing Zhang New Journal of Physics, Guardian, UK)

2.1.1.5 A permanently evolving continuously engineered system

The Internet is also a field of trials for engineers. Principles like “rough consensus and running code”, which prevails in the decision making process at the Internet Engineering Task Force (IETF) meetings, also strongly shapes the overall Internet architecture. “Running code” means that any new proposal made to improve the Internet is backed by running code, which is a necessary ingredient in gaining acceptance of a particular proposal (often proposals come competing and sometimes the implemented one has the edge).

At the Internet Service Provider (ISP) level, and more generally at the Autonomous System level, Traffic Engineering is very important. Traffic engineering is about optimizing the performance of networks and is becoming more and more popular due to the popularity of the services provided by the Internet. Problems like TCP congestion, TCP unfairness and flow management are tempered and avoided by means of extensions to the current standards.

Other non-technical factors also promote changes to the current architecture. For example, we can consider the ISP traffic shaping due to external political pressures as one these phenomena. The loss of trust is also one of the most critical. Indeed, the simple early Internet model when a known number of mutually trusting parties attached to a transparent network and exchanged files is gone forever. This growing concern about trust promotes new security architectures and other solutions that break the end-to-end principle which limits the innovation.

2.1.2 Evolutionary Approaches

In project EXALTED, the current orientation is to design an IP networking system for M2M communications by using evolutionary approaches. Since there are high chances of demonstrating the technologies of EXALTED on a down-to-Earth setting, it is necessary to use an approach where the existing mechanisms (addressing, protocol) are enhanced in a step-by-step manner.

The Internet is a heavy complex engineering system: any significant change to the IP layer, the waist of the hourglass that holds the system together, can lead to great instability in several domains, as more and more applications rely on the Internet as a middleware.

Recently, proposed enhancements like IPv6, Mobile IP, IPSec, QoS mechanisms and multihoming despite their intrinsic worth, cost too much in terms of deployment: triangular suboptimal routing, deployment of new entities breaking the E2E principle and more. Consequently, these enhancements remain as unresolved challenges, at least for the global Internet [55].

Usually, we know two main directions to follow in order to change a system:

1) Evolving the system incrementally, by deploying new mechanisms (hardware and software) having the new desired features and stay backwards compliant with previous versions of the system. We can refer to these mechanisms by patches. Some call this approach “engineering method”, as the costs of the overall solution appear amongst the first design goals.

2) Redesign the system from the scratch regardless of the already deployed system, following new core principles and having the desired features. This design method is the clean-slate approach, to which we can refer to as revolutionary method, opposed to the evolutionary one. It is often considered as a research task, where the costs of the overall solution are the last design goals.

This section will cover the evolutionary approach in the recent Internet enhancements proposals.

The IAB Workshop on Routing and Addressing [23] is the starting point of several proposals in the new IP Locator/Identifier split realm. The workshop participants pointed to the semantics overload of IP along with multihoming growing interest among ASes as the main reasons for the DFZ RIB growth causing overall scalability issues on the whole system [38].

We can classify proposed solutions based on the parts of the network that are affected by the patches. Indeed, some proposals (namely, Shim6 and HIP) applying the end-to-end principle, imply a change above the IP layer on all hosts and other solutions (LISP and GSE) imply an incremental deployment of routers with new capabilities in the core network.

2.1.2.1 Host-based solutions

There are two main solutions currently proposed at the IETF: Host Identity Protocol (HIP) [88] and shim6 protocol [34]. Both solutions change the network protocol stack to add a new layer in order to better handle the identities of hosts.

2.1.2.1.1 Host identity Protocol (HIP)

There are two major contributions in this proposal: a host identity namespace and a new protocol layer, the host identity protocol layer.

The Internet has two important namespaces widely in use: the IP addresses and the domain names. The Host Identity namespace, defined as a set of cryptographic host identifiers, is the answer to the IP semantic overload and supposed to add completeness to already deployed namespaces [88].

The HIP protocol implies changes at the host stack. Endpoints are identified by Host identities used above the IP layer (IPv4 or IPv6) in the transport layer (TCP/UDP and more).

Hosts will be able to authenticate their peers directly when knowing the Identity, with this cryptographic namespace [79].

This additional namespace enhances the original Internet architecture by implementing the desired Identity/Locator split and changes the transport layer session binding to the Host Identifier and no longer to the IP address, making a number of networking challenges such as mobility, multihoming and even security easier to deal with. More than an additional namespace, HIP aims at providing a new layer of indirection as it is believed that effective mobility support requires an additional level of indirection [79]. Thereby, mapping a transport session to the identity will ease handling mobility challenge. Multihoming is another hard networking problem tackled by HIP.

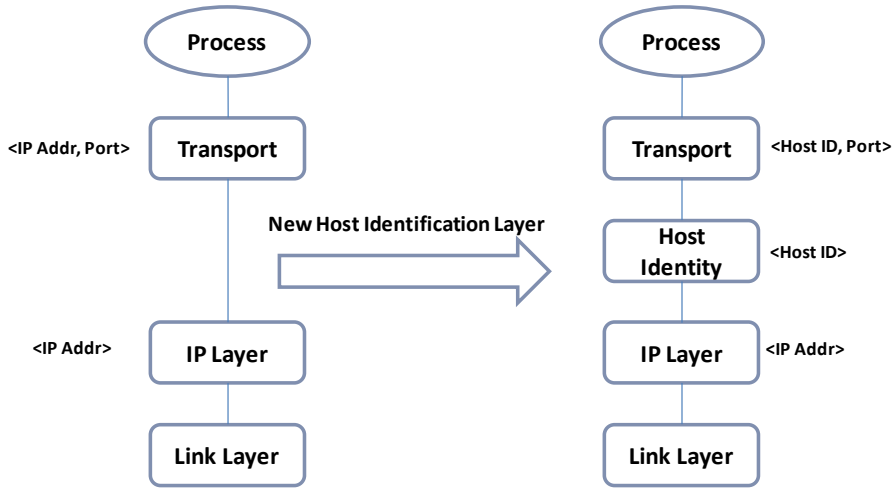


Figure 7: HIP layering model. The integration of a new Host Identity layer

As part of the Base Exchange, IP addresses are used as locators and can be updated during a communication session [89] [80]. Renumbering, which is an unavoidable administrative burden, is handled as a particular case scenario of mobility. After the four-way handshake between the two peers, which is based on a sigma-compliant Diffie-Hellman key exchange using public key identifiers as a way for mutual authentication [89], shorter Host Identities are used in the HIP header to exchange packets. The 128bit Host Identity Tag (HIT) and 32bit Local Scope Identifier (LSI) are such short identifiers. A HIT is built in an IPv6 format, where the 28bit prefix is 2001:0010::/28 and the remaining 100 bits are taken from the crypto hash of the host public key [79]. The HIT can be compared to the CGA address in the SEND context [102] where a 64bit Interface ID is generated through an algorithm where the host public key (among other parameters) is hashed to obtain the resulting IPv6 address.

When HITS are intended for global use as IPv6 addresses, LSIs are locally unique IPv4 addresses equivalents and cannot be reliably used to name hosts outside the network [88]. HITs are unstructured, not human friendly and not aggregatable. In order to retrieve a HIT (supposing HITs are stored in a distributed hierarchical database, such as DNS) a user must fetch the IP address, knowing a URN, along with its associated HIT. The opposite, i.e. starting with a HIT and fetching IP/URN from the DNS, is not possible currently. These issues, namely, a mapping/resolution system are discussed within the IETF HIP Working Group [79].

In order to provide mobility, a new entity is introduced: the RendezVous Server (RVS). The RVS solves the simultaneous movement of endpoints problem and provides location management. The RVS acts as a permanent HIP host reachable whenever a correspondent becomes unreachable (it is the case during mobility). The RVS is involved in the HIP readdress packets by forwarding the I1 message to the correspondent host. RVS is solicited

with HIP control packets only, once the locators are updated, hosts will communicate directly with no proxy server. The RVS is compared to the Home Agent of MIP protocol, but with more flexibility (HIP host knows more than one RVS, can change them dynamically and only solicited for control messages). In practice, stationary HIP hosts in the public Internet could provide a rendezvous service [79] [21] after a registration procedure [54]. DoS attacks are another topic addressed by HIP, for which it provides protection for transport protocols running on top of it [4].

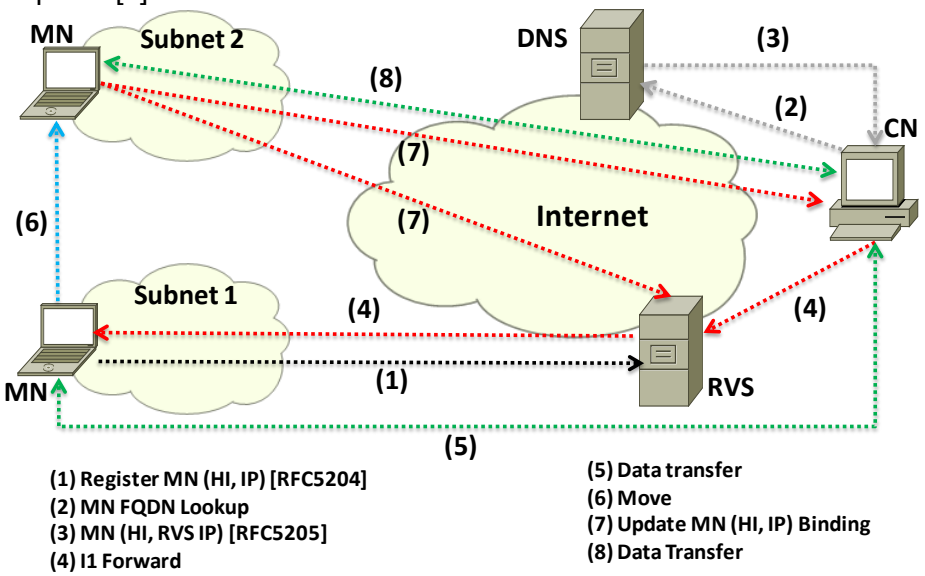


Figure 8: HIP mobility model

2.1.2.1.2 Shim6

The shim6 addresses the multihoming problem and provides a locator/identifier split by the addition of a shim between the network and transport layers. From a technical point of view, shim6 provides stability to the upper-layer protocols (TCP, SCTP) by presenting a stable source and destination identifier pair, called Upper-Layer IDentifier (ULID) while changing IP addresses depending on the prefix in use (locators). Shim6, also known as “level 3 multihoming Shim Protocol for IPv6” [34] is designed for a better scalability of the global routing system using provider allocated prefixes (PA) to facilitate provider-based prefix aggregation [23].

This host-based approach supports a new networking layer (a shim) between the current network layer and the transport layer. An additional protocol, REAchability Porotocol (REAP) [52], is responsible of detecting failures between Shim6 communicating nodes, and switch between locators to re-establish the communication session. REAP is an enhanced ICMP protocol for Shim6. In the protocol design, the shim layer is the one performing forwarding actions as selecting a suitable next hop for some destination, while IP contains end-to-end mechanisms, as IPSec [3]. One enhancement provided by the protocol, is the possibility of using different pair of locators (ULID) for different directions of the same communication session. Different communication sessions can use the same shim6 context. So the shim is shared between upper-layer sessions, i.e. different ULIDs may belong to the same session, and different sessions may have the same shim6 context.

In order to establish a shim6 communication between two hosts, a four-way handshake is specified. After this procedure, each host knows the different locators available for a given communication. The shim6 context creation (four-way handshake) does not have to occur at the beginning of the communication. Two messages update request and acknowledgement allow the hosts to change the set of available locators during a session. These messages can be used to support mobility or site renumbering. Once a communication context is

established (creation of ULIDs with a set of locators), the context can be discarded, recovered or forked [3].

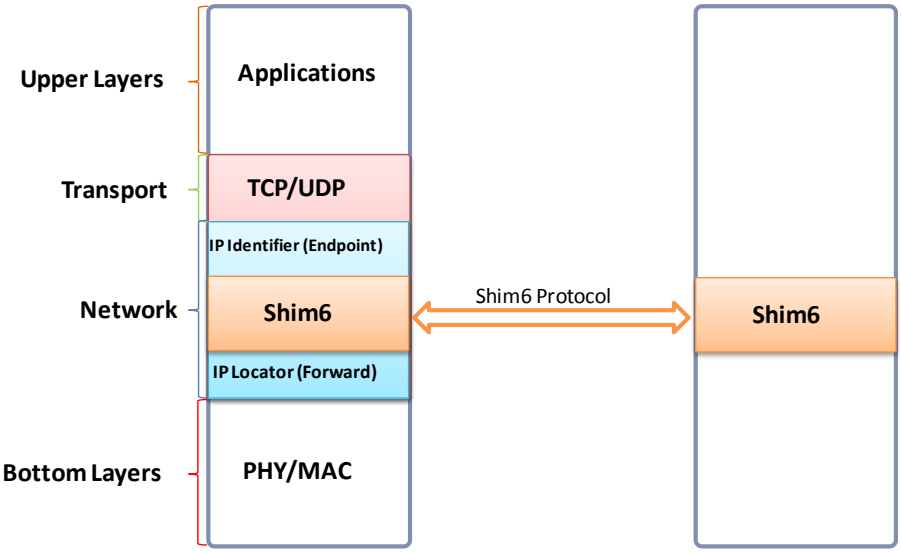


Figure 9: Overview of the Shim6 protocol

The REAP protocol completes the shim6 architecture by detecting and recovering from failures [52]. REAP is implemented at the host level and allows finding new pair of locators when unidirectional path failure occurs. A set of messages (Probe, Keepalive) and a timer (Send) are the protocol tools used to maintain the reachability of hosts and session continuity.

In order to prevent Hijacking and flooding attacks, Shim6 proposes to map a cryptographic hash of Host Identity into the IPv6 address, i.e. using CGA and HBA [102] [65] and to use REAP Probe as a mean to detect communication diversion to random victims (flooding) by a shim6 context malicious update.

The overall cost of the Shim6 solution must not be neglected. First, every host stack has to be upgraded to support the new shim. The REAP protocol at the host takes responsibility of maintaining communication sessions and switching to a working identities pair when the currently used one fails. Another implication of REAP and other ULIDs facilities is the maintenance of additional information state about current communications. Finally, as a host-based solution, it prevents ISPs from doing traffic engineering [23].

2.1.2.1.3 Other host-based approaches

Focusing on the mobility and multihoming enhancements, other host-based approaches exist. Multiple Address Service for Transport (MAST) proposal [18] [17] is another between-network-and-transport-layers approach. The author [17] suggests to use a control protocol between communicating endpoints, in order to map between a pool of locators (IP addresses) located in the bottom IP layer (IP-TR) to a unique endpoint identifier (EID) located in the upper IP layer (IP-EP). The first used IP address is the identifier presented to transport layer, while additional dynamic IP addresses as the host moves are considered as locators associated with the initial IP address (EID).

A set of basic control messages (INIT, SET, PROBE and SHUT) are exchanged between hosts to start, maintain and close a MAST association. In order to maintain a permanent dynamic presence service and allow session establishment during host movement, MAST defines a new DNS SRV record [17] to associate a domain name (public stable EID) with the

set of currently used IP addresses. Standard messaging operations to maintain the coherent state of this record are defined through XMPP [81].

LIN6 [21] [32] uses a different approach from previous proposals, to split the IPv6 into an identifier and locator parts. The proposal considers the address as composed of a node identifier and a node locator, and the mapping operation between the two is done with the network layer. This proposal is not based on a new identification layer in the stack. In technical terms, there are three different concepts. (1) The LIN6 prefix, which is constant, (2) the LIN6 ID, which globally unique and every LIN6 node has one, and (3) the current topologically correct IPv6 prefix. The combination between the LIN6 Prefix and ID is globally unique and remains unchanged within the host even if the node moves or is multihomed.

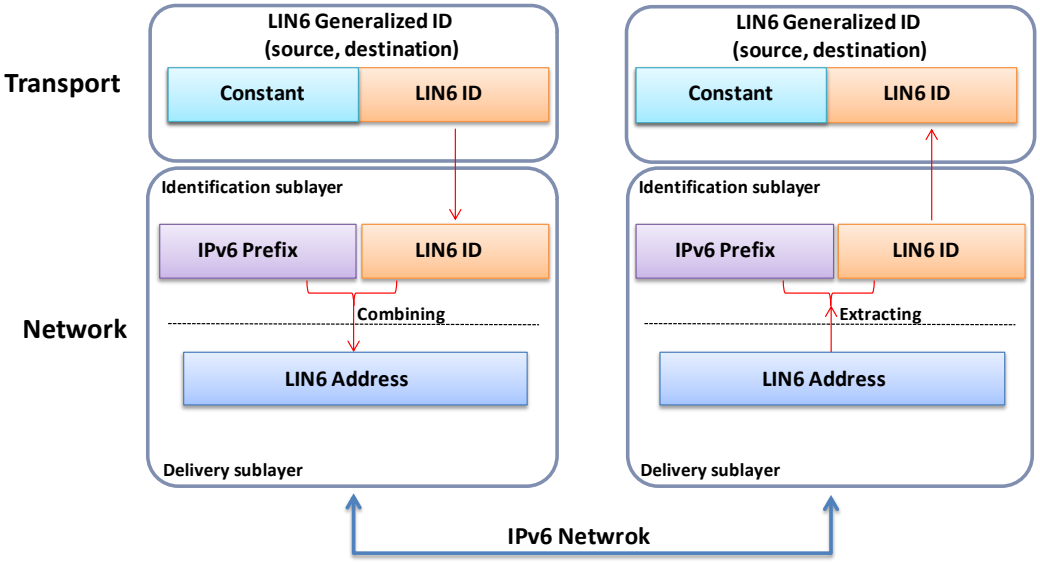


Figure 10: Transmission and Reception in LIN6

The LIN6 address is composed of a LIN6 ID and a topologically correct network prefix. The resulting IPv6 address is then globally routable. The last 64bit part of the address (Interface ID for IPv6, LIN6 ID for LIN6) remains stable during node movements. To send packets across LIN6 architecture, an additional functional element is specified: the Mapping Agent (MA). The MA manages the mapping between a LIN6 ID and the current network prefix. When a peer queries DNS to obtain a mapping to an FQDN, the DNS server returns a LIN6 ID. This peer has to query the MA to obtain the topologically correct network prefix for the given LIN6 ID, and then the peer can send packets to its correspondent by concatenating both information. The MA is updated whenever the registered node changes the network location and the CN mapping is refreshed by another control message form the mobile node. If the Refresh Request has no authentication header, the CN has to query the MA to obtain the new network location of its peer.

2.1.2.2 Network-based approaches:

The IAB Workshop on Routing and Addressing [23] considered the routing scalability of the global Internet as the number one problem that must be rapidly fixed. The main clue for this rapid and unscalable growth is the DFZ RIB and FIB size which evolves on an over linear growth [38]. Other issues related to scalability, like convergence time, cost and energy-consumption have been noticed. It is also believed that the advent of the IPv6 will worsen the problem with its huge addressing space, when IPv4 with its limited address space constrained the phenomenon.

Recent network-based approaches focus on the locator and identifier realms split. These proposals describe the Internet as two parts evolving at different speeds. (1) Edge network, where the clients reside and where IP prefixes de-aggregation happens and (2) Core network, where aggressive IP prefixes aggregation should happen. By differentiating the problems, recent proposals [19] [24] [22] aim at providing a stable Internet where prefix aggregation would help reducing the routing table sizes in the core of the system. Another early proposal [70] tried to rewrite the IPv6 address to change its semantics and provide a way to enhance prefixes aggregation at different levels.

2.1.2.2.1 Locator/ID Separation Protocol

LISP is a map-and-encap network-based protocol [24] [86]. The basic idea is to define two sets of elements: Routing locators (RLOCs) and Endpoint IDentifiers (EIDs) on a same numbering space, the IP, regardless of the version. EIDs will be used by hosts as identities, and RLOCs used by Ingress/Egress Tunnel Routers to route the packets in the core network. The expected advantages are similar to those of provider-allocated IP address space, where the aggregation is made simple, as opposed to provider-independent IP blocks used by some organizations to avoid the administrative burden of renumbering, even if it means additional non-aggregatable entries in core routers RIBs.

Mapping-and-encapsulating was first defined in ENCAPS protocol [86]. The specification describes a simple method based on a combination of mapping operation and packet encapsulation as a medium term solution to evolve the existing Internet. The proposition is a medium term transition protocol with low costs, allowing the deployment of a new long term solution.

The LISP proposal aims at evolving the Internet by differentiating between hosts that use EIDs as identifiers and border routers that use RLOCs to forward (through tunnels) hosts packets to destinations. The border router decision on forwarding is made after an EID-to-RLOC mapping. The packet is then encapsulated. The inner-header will carry source and destination EIDs and outer-header the source and destination RLOCs. EIDs are much likely site scoped, but RLOCs must be global scoped.

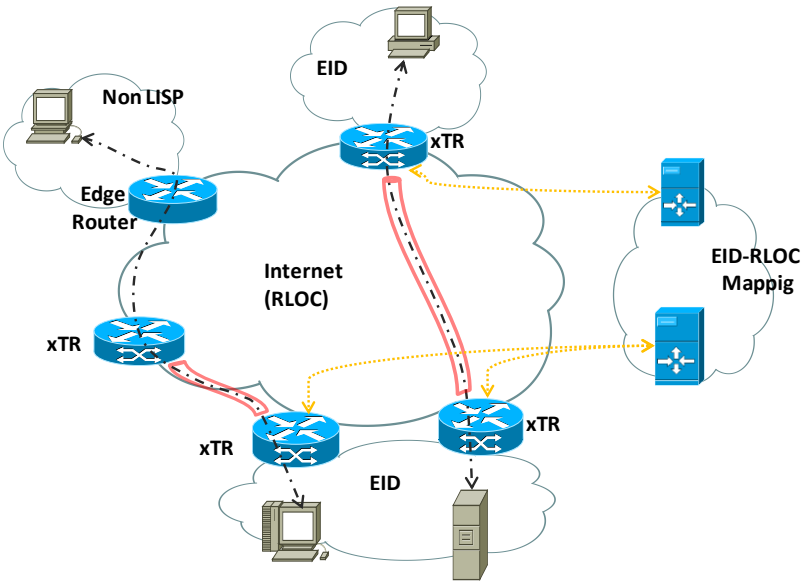


Figure 11: LISP Architecture

LISP approach separates the protocol into two modules: data plane (map-and-encap) and control plane (mapping system). Various proposals for the mapping system exist, for

example, based on distributed hash tables (DHT-LISP) [63]. LISP does not require host changes and does not change the core routing infrastructure. Two functional elements are needed to deploy the solution: Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR). ITRs do LISP-encapsulation and the mapping operations, while ETRs do LISP-decapsulation and deliver packets to destinations [19]. At a host level, nodes sending data will do a DNS lookup to get destination EID before sending the packets. This does not change current hosts practice. The packets delivery will be handled by ITR/ETR routers by tunneling, after EID-to-RLOC ITR mapping operation. LISP specification proposes the use of MIPv4 and MIPv6 to handle the fast host mobility use case [19].

LISP raises some performance considerations about encapsulation overhead and mapping lookup latency (control plane) [24].

2.1.2.2.2 Global, Site, and End-system address elements (GSE)

GSE is an indirection approach to provide scalable multihoming in the network. The proposal [70] aims at providing aggressive topological aggregation to control the routing tables growth in the core network. The IPv6 address has to be redefined in order to achieve this. The address will then bear new semantics: (1) locator part, called Routing Goop (RG), (2) a local site information, called Site Topology Partition (STP), and (3) an interface ID of the endpoint, which is the End System Designator (ESD) in GSE terminology.

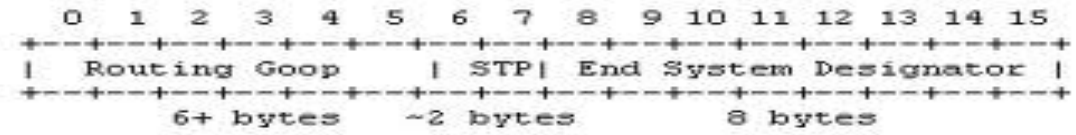


Figure 12: GSE IPv6 addressing format

The original proposition [69], called 8+8, of which GSE is the evolution, illustrates the IPv6 address format rewrite. The 16byte IPv6 address is split into two main parts. The first 8 bytes (left to right) are about site attachment and used to maintain compact routing tables with aggressive aggregation. The first part of the 8 bytes (about 6 bytes) is the RG. It specifies a path from the root to a point in the topology. If a terminal is attached to this point of attachment in the topology [56] then the hosting network is a site defined by this unique RG. The Internet topology is consequently partitioned hierarchically in a tree fashion. Although cut-throughs (shortcuts) can be defined through the hierarchy to illustrate the Directed Acyclic Graph (DAG) nature of the Internet [70], some network architecture experts [53] criticize this approach for this tree-like Internet shape that the IPv6 address will have to bear in the front part following the operating system model too closely.

The rest of the first 8 bytes (about 2 bytes) is the STP. This part is close to the meaning of the prefix. The author describes it as a partition of the site topology, or a segment. If a site administration wants to protect its network internals (as does the NAT Boxes), it can present a non-significant STP part to its peers. Otherwise, if the organization is presented as a structured site, inter-site topology will be disclosed as part of routing control messages, for example.

The second main part (last 8 bytes) of the IPv6 address is the ESD. This part is dedicated to the Endpoint (one interface on the system, to be accurate) and identifies it globally and unambiguously. The author proposes to create a new pool to generate such identifiers, especially for nodes not equipped with IEEE MAC address. Other nodes (majority) could use EUI-64 as an ESD.

The DNS mapping service will be augmented with a new association: to a name (FQDN) will be associated a (ESD, STP) pair and RG information in a “AAA” record. This will serve the

source end-system before sending a packet to a destination. If the RG information is not available a special unspecified value can be put in the first 8 bytes and the border routers will replace this part with the appropriate value if the ESD is not on the site. To access on site-resources, the site will provide a differentiated name service based on the source address: for internal requests, only ESD (and STP)
will be provided, but fully-general IPv6 addresses (actual RG information) will be returned to external queries.

The GSE proposal intends to ease renumbering burden associated with multihoming. Obviously, the RG part has to be redefined whenever a rehoming operation occurs. Different site types are assessed (provider, leaf) and rehoming courtesy and tunnels between former and new providers are presented as short-term solution reducing packet loss.

2.1.2.2.3 Other network-based approaches

Another network architecture design approach [22] suggests to separate the IP addressing space into globally routable addresses (GRA) and globally deliverable addresses (GRA). Claimed enhancements are improved routing scalability and ease of site-multihoming.

GRAs are the addresses used within the DFZ domain, and are only reachable from inside the DFZ, while GDAs are globally unique and used to be reachable everywhere and do not appear in the DFZ tables. The point is that, rather than focusing on splitting between locator and identifier realms of IP, it is more effective to separate customer networks (edge) from provider networks (core) on an addressing-basis. According to the authors, the GRA addressing space should provide a topologically aggregatable space that will help maintaining routing table size at an acceptable size. The GDA works with a mapping and tunneling system (map-and-encap) similar to the LISP approach. Border routers of source and destination (not located on the same site) hosts, encapsulate packets to traverse the DFZ, as it ignores the GDA addressing information state necessary to do the forwarding operation.

In the previous presented solutions, we see that the host-based approaches are based on the observation that the classical layering model lacks in an identity layer. The authors proposed to add such a layer and built different protocols upon different definitions of what an identification space could be. These approaches do not contradict the network-based solution, but rather complete them. The network-based approaches try to split the global Internet into two types of networks running at different speeds. (1) The core network, where the routing operations have to be simple and routing tables compact. (2) The edge network, where as few changes as possible should be made and where prefix aggregation and deaggregation should maintain the scalability objective of the system.

2.1.3 Clean-slate architecture core principles

In project EXALTED, the current orientation is to avoid designing mechanisms that may involve the change of fundamental core principles; the clean-slate design is to be avoided, for the time being (and use an evolutionary approach instead).

New engineering challenges such as multicast, mobility, QoS mechanisms, multihoming, security and more arose with the growing interest of different domains in the Internet. Different types of applications call for different types of service which pushed the E2E design principle to the limits. A flat general purpose network design coupled with rich, complex and intelligent end systems is far from being the answer to all these interrogations, at least from an efficiency point of view. Some engineering approaches treated the problem, but this is not the only way to solve these issues. Clean-slate network design is another view of what could be the future Internet. Researchers and engineers of this field claim that a number of hard

networking problems results from early Internet design legacy and therefore a design from the scratch could alleviate the burden and ease the integration of numerous enhancements.

By (temporarily) ignoring practical constraints and exploring a larger solution space, right solutions to current Internet technical issues should be provided and then adapted in an incrementally deployment scenario [55]. Different research initiatives tackling various problems have been described. The US Global Environment for Network Innovation (GENI) [41] initiative is a common infrastructure for future Internet proposals implementation. It is the experimental facility for the Future InterNet Design (FIND) from US National Science Foundation (NSF) research program. Future Internet is also one of the European Commission research targets as part of the Seventh Framework Program (FP7). The AKARI Japanese project is another instance of future Internet design initiatives [55] [2].

2.1.3.1 Content-Centric Networking (a.k.a. Networking Named Content)

Content-Centric Networking (CCN) [16] considers the content as being the building block and the original component of a new way to do networking. According to the authors [108] [26] the networking problem that originally guided the Internet design, namely resource sharing, is no longer a viable model to build the future Internet. The network users value the content and not the container. Instead of asking the question “where can I get this content?” (Basically, “classic” network design is about answering that question), users ask “what content can I get from the network?” (CCN makes the content as the priority and design the network according to that).

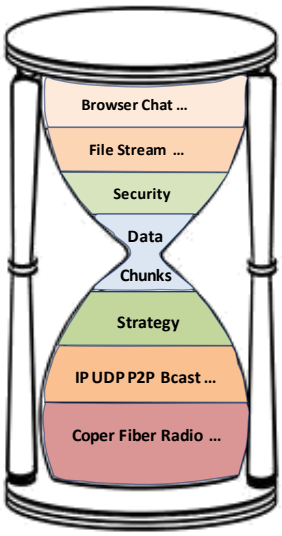


Figure 13: CCN's new hourglass

CCN is also concerned with security. While the first Internet design was built on trust assumptions, the CCN is designed with strong security objectives. The communication model change implies a security realm change too: when IPSec secures connections on which the packets travel, CCN secures the content itself.

The communication design follows data consumer model with only two packet types: Interest and Data. An Interest is broadcasted over available connectivity and a Data packet is sent by a node that hears the request. While the TCP congestion is handled by sliding windows, every Interest is consumed by the answering Data, so the flow control is maintained at each hop of the communication.

In order to perform basic CCN operations, three new data structures are defined. (1) The Forwarding Information Base (FIB), like IP FIB, it is used to forward Interest packets to

potential sources on different interfaces (called faces in the CCN terminology). (2) The Content Store (CS), a buffer memory with maximum utility policy replacement (LRU, LFU) that enhances sharing between hosts. When implemented in forwarding routers, CCN queries can be satisfied before arriving at the source. Data Integrity is of paramount importance in this context. (3) The pending Interest Table (PIT). The authors compare the Interest Packets journey through the network to “bread crumbs” left through the path in order, for the traversed nodes, to find a way back to the sender. The PIT is the data structure that keeps this trace. Whenever a Data packet answers an Interest, the pending PIT entry is removed. An additional data structure, an Index, is consulted (in longest match lookup on queried Content Name basis) to find a suitable outcome for an Interest packet: if available on the Content Store, the Interest is satisfied. Otherwise, PIT then FIB will be consulted, respectively.

The CCN transport provides delay tolerant networking whenever there is an opportunity to forward packets. CCN transport protocol is stateless and the application running above is responsible of resending an unsatisfied Interest request.
CCN names are hierarchical and humanly readable. In order to split data into chunks, unlike TCP sequence numbers, CCN uses versioning and segmentation notation along with a globally-routable name.

CCN enhances mobility by construction. While TCP sessions are bound to an IP address, which makes mobility a challenging concept, CCN does not need a binding at lowest layers, taking advantage of currently connected interfaces and choosing which one fits best its Interests. The strategy layer plays an active role to achieve this mission.

A CCN router can be placed in a routing domain among IP routers. For Intra-domain routing, CCN routers learn how to reach some content by some CCN router after hearing an announcement concerning this content. The router will install a FIB entry towards the announcing router, on a certain face for a given content. Same mechanisms apply for greater scope deployment (inter-domain) in a bottom-up driven deployment [108].

Security is also a central concern of the proposal [26]. Instead of trusting the original sender of the content and securing the path on which the data travels, CCN’s approach is to use public keys to authenticate the link between names and content.

The evaluation results show an interesting behavior of failover recovery during intermittent connectivity with no data loss. These benefits come with the price of changing the application development model.

2.1.3.2 Routing on Flat Labels (ROFL)

This proposal aims at routing on host identities and ignoring network locations. In recent Locator/ID split proposals, most designs introduced a mapping or resolution service at some point in the routing process. ROFL [66] proposes a location free network layer and route on the identifier information. Hosts are named on a flat namespace with no particular semantics given to the name. These names can be public keys hashes, and are not mandatorily unique. Non-uniqueness is used in ROFL to perform anycast and multicast. ROFL work can be linked to compact routing [97].

As in CHORD [48], a circular namespace is created and notions of predecessor/successor helps to perform a reliable routing. In ROFL terminology, a host attached to a router is said to be “resident” at this gateway router. The router is hosting that host ID.

Nodes are of three types: routers, stable and ephemeral hosts. The distinction between ephemeral and stable hosts is made by hosting router administrator. ROFL runs on top of

intra-domain routing protocols, that helps detecting link failures and assumes self-certifying identifiers to prove a node's identity (spoofing prevention). In order to achieve intra-domain routing, a newly attached host ID is considered as the predecessor ID of some (previously attached) node and the hosting router of this node is contacted, so it can install a source route to this newly attached node as well. This is the part of the CHORD join algorithm to establish source routes in the router cache. The routing is done from a node along its successor pointers: it is greedy. For inter-domain routing ROFL proposes a similar approach on an AS-level scale. To forward a packet, a router performs a host match function (known closest ID to destination) as opposed to longest-prefix match in hierarchically structured namespaces.

An interesting property of routing in ROFL is the isolation; that is, of packets are exchanged between in-AS hosts, no external pointers (path across different ASes) are used. For hosts of different ASes, ROFL ensures that packets will not traverse higher than least common ancestor in the DAG resulting from merging rings. The isolation property guarantees also that failures and instability are experienced within one site and do not bias neighboring ASes routing. The authors argue that despite non-ideal performance results, the research in this should continue and the idea of routing on flat, non-hierarchical, semantic-free labels in chord-like graphs cannot be dismissed.

2.1.3.3 NIRA: A New Inter-Domain Routing Architecture

NIRA proposal [111] is about giving Internet users the choice of providers for their packets traversal. The main objective is to encourage the ISP market competitiveness, enrich the offers, reduce costs [27] and improve the end-to-end experience by giving the users the power of choice between domain-level routes. The end-to-end model [59] is redefined to contain three parts: the sender, receiver and the core. Technically, NIRA is built on top of two protocols: Topology Information Propagation Protocol (TIPP) and Name-to-Route Lookup Service (NRLS). TIPP maintains the user view of the up-graph network part of the overall architecture with two modules. (1) Path-vector part that distributes a set of available provider-level routes to the user, (2) policy based link state part that informs the user of the network conditions and allows a failure free packet delivery.

Along technical concerns in the system design, some practical questions, such as payment modes, have been investigated to allow future concrete deployment. To achieve hierarchical route representation [70], NIRA chose a provider-rooted hierarchical address representation to encode the user-up-graph into the user's address.

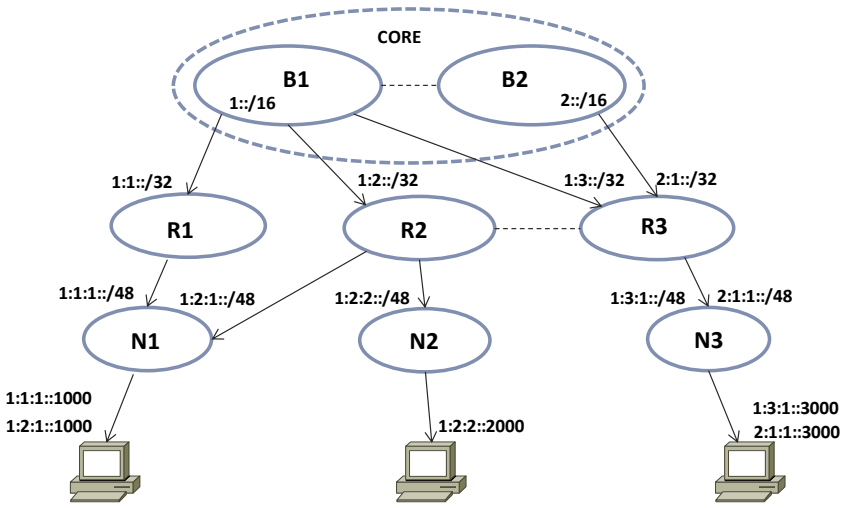


Figure 14: NIRA's provider rooted addresses

Therefore, source and destination addresses are (both) used for forwarding and spoofing is limited since the address represents a hierarchy. The NIRA address representation can have two forms: (1) a fixed-length address with large addressing space. IPv6 is one example used by the authors [111] and (2) a variable-length address, which has not been demonstrated. For packet forwarding, NIRA proposes three forwarding tables routing model. The routing information is grouped through TIPP and forwarding decisions is made according to downhill table (for destination address) and uphill table (source address). If no match is found and no Core link is used to route the packet, a special table entry for a router with peering link may indicate the bridge table for the forwarding decision.

The performance analysis of different parameters in NIRA (control overhead, convergence speed and setup latency) shows acceptable results for practical deployment. Some other issues, as temporary route oscillation and suboptimal route choice are left for future work.

2.1.3.4 More clean-slate design approaches

Internet architecture clean slate design is a new and widespread trend in network design. Different approaches tackling different angles are proposed [55]. The Japanese AKARI Project [7] aims at developing a deployable network architecture on short term. Different technologies have been considered for integration (radio, optical) and functionalities like guaranteed service, mobility, and security are considered early on the design. ID/Locator split is also one design goal for AKARI [109]. Hosts and border routers protocol stacks are augmented with an Identity Layer to achieve better mobility, multihoming and security.

TRIAD Project [106] [25] as well as IPNL [82] considered a large scale NAT architecture, where routing could be done on FQDN-basis, considering them as hosts identifiers. TRIAD takes a content distribution perspective while IPNL focuses on routing and IPv4 addresses depletion problems.

In FP7 projects, Trilogy project [107] considers the separation of naming and addressing in IP issues, in collaboration with the IETF. For example, Multipath TCP (MTCP) is a joined effort to allow the use of several IP addresses and interfaces on TCP. A Linux implementation exists [74]. 4WARD [1] is another FP7 project for future Internet. Solution space includes technical issues, as network virtualization and management functions, with non-technical problems, as finding innovative ways to generate value and employment opportunities.

The clean-slate design model can benefit to the current Internet in many ways as some of the proposed changes can fit in the current architecture or included progressively. The security and mobility enhancements are such examples.

2.2 Overview of protocols for 6LoWPAN

2.2.1 Mobility protocols in 6LoWPAN

A smart object is any physical device (mostly battery operated) capable to exchange information in real-time with any other device via a local network. An example of a smart object is a Machine device, and a node in the 6LoWPAN is a Machine. Since IP runs on top of most communications media (from high-speed Ethernet links to low-power 802.15.4 and 802.11 links) it may prove convenient to use IP to enable communication between smart objects and other embedded networked devices. In order to support the large number of applications for smart objects, the networking technology must be scalable, interoperable, stable, manageable and flexible. IP is an omnipresent protocol fulfilling all of these requirements.

Mobility of network devices became the common requirement in modern technologies. Edge router could be considered as M2M (Machine-to-machine) device. Due to the radio changes, devices can force nodes to foresee new route paths and even to change LoWPAN network without moving.

Two types of mobility are possible in 6LoWPAN networks itself: micro and macro mobility [114]. Micro mobility in 6LoWPAN refers to the mobility of a node within 6LoWPAN where the IPv6 prefix remains the same; likewise macro mobility means the mobility between two 6LoWPAN networks with different IPv6 prefixes. In the first case we have only handover, while in the second we have joint roaming and handover mobility in place. The same definition, regarding macro and micro mobility of nodes can be applied for the edge routers.

In the figure below the possible types of mobility are presented.

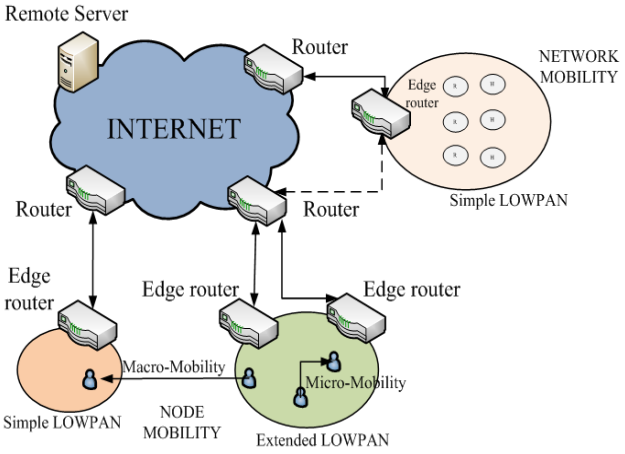


Figure 15: Mobility types inside 6LoWPAN

From the network perspective there are node and network mobility [TKS3]. Node mobility is covered in previously described cases of macro and micro mobility. Network mobility occurs when the edge router changes its point of attachment [34], while all nodes from 6LoWPAN network remain still the same. This is the type of macro mobility, because when the edge router changes its point of attachment, the IPv6 address is also changing which results in change of node's IP addressing.

When the node changes its point of attachment there are several things to be done in order to resume data flows:

- reestablish link by commissioning,
- assign IPV6 address by bootstrapping node,
- update of DNS settings with new IPv6,
- notification to application layers etc.

When micro mobility takes place, link layer is sufficient to cope with mobility without any notification to network layer. Today, 802.15.4 intends to leave mobility issues to the network layer, and all topology changes are node controlled. Dealing with mobility issues is especially hard from the perspective of an application. If a node is acting as a client, the best way that also fits to 6LoWPAN, is that when ever a node detects change in the IPv6 address the application restarts itself. However, this is not so practically for the case where a node is acting as a server due to the requirement that servers must be reliable 100% of time. In this scenario within a 6LoWPAN network, application is dealt on application level using SIP, URI or DNS.

2.2.1.1 Mobile IPv6

Mobile IPv6 provides one way dealing with this problem on network layer, so that edge routers can perform mobile IPv6 on behalf of LoWPAN nodes by using proxy home agent (HA) [13].

In cases when network mobility takes place and an edge router changes its point of attachment, dragging all along networks nodes within 6LoWPAN network, we can use mobile IPv6 for edge router and nodes inside 6LoWPAN network for normal IPv6 address auto-configuration. In the cases where a LoWPAN or an IP backbone network does not have the mechanisms to cope with mobility on network layer, the mobility can be done on application layer, thus forming better optimization for certain applications. The main task regarding mobility on application layer is to answer on question how the transport protocol reacts to an IP address change. 6LoWPAN network applications should use UDP as transport protocol due to independence between each datagram, and because it is possible to correlate the new IP address to the same end point. As TCP is based on congestion avoidance design and therefore harder to compress, it is not suited for lossy wireless mesh networks. UDP is mainly used within 6LoWPAN because it is simpler, easier to compress and fits to most applications protocol needs. 6LoWPAN network requires to deal with reliability if needed, out-of-order packets and datagrams rather than streams, so UDP is better to use compared to TCP. Therefore, TCP application and streaming RTSP could cause sessions to break. It is necessary that application server that is communicating with 6LoWPAN mobile nodes uses unique identifiers for each node within 6LoWPAN so that IPv6 changing address is possible. For these purposes a server can use either EUI64, URI, Universal URI, or domain name resolved using DNS, where we would have mapping between domain name and IPv6 address within 6LoWPAN.

SIP as a protocol has its own proprietary feature regarding mobility. It is a feature built in the protocol itself where we can use SIP URI for tracking 6LoWPAN nodes. The form of URI Identifier is nodex@home.6LoWPAN.com. Through RE-INVITE message SIP has the mechanisms to indicate that a session endpoint is changed during a session itself. SIP must be adapted to 6LoWPAN network requirements in order to work properly. Mobile IPv6, derived from IPv4 protocol, goal is to cope with mobility roaming scenarios, thus enabling communication with a node regardless of its location on the internet. Mobile IP is based on concept of Home Address associated with host's home network, and each time when node roams to visited network the new IP address is configured there.

MIP has a host agent, that forwards traffic from/to the node that roams and it pairs the Home Address of a node with his Care-of Address (CoA) when he is roaming, using binding messages [13] between node and Home Agent (HA) sent trough bidirectional IPv6 tunnel. So when ever packet is sent to home node address (node is roaming) packet is being intercepted by HA and then encapsulated in IPv6 with his new temporary roaming IP in the header. All derived above is presented in the following figure.

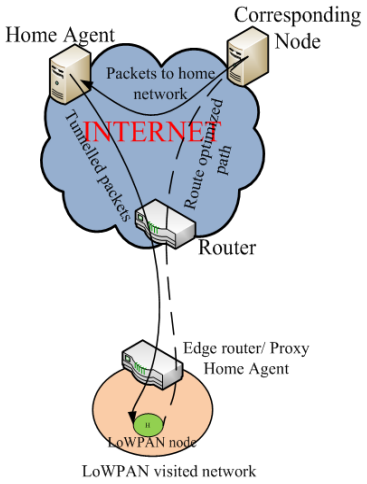


Figure 16: MIPv6 in 6LoWPAN

One of the scenarios for further discussion is when edge router or some other entity in the visited network could proxy MIPv6 on behalf of LoWPAN nodes. Also, it is possible that edge routers perform compression and decompression of messages to full IPv6.

2.2.1.1.1 HMIPv6

HMIPv6 (Hierarchical Mobile IPv6) [43] [37] is an extension of MIPv6 and its separates global mobility from local mobility. It is designed so that the nodes can move within a domain without having to update their HA or Correspondent Nodes (CNs) every time they move. Here, a new node is introduced, called Mobile Anchor Point (MAP) which acts as local HA for nodes in 6LoWPAN. When a node moves in a new domain there are two addresses: a Regional CoA (RCoA) on the MAPs link and On-Link CoA (LCoA). After the node has sent binding message to the HA and the registration process took place, a bidirectional tunnel between the node and the MAP is established. In case that node changes its position inside the network, a new LCoA will be assigned but the RCoA remains the same.

2.2.1.1.2 FMIPv6

FMIPv6 (Fast Handovers for Mobile IPv6) [37] [87] is another enhancement of MIPv6, that allows nodes to predict their mobility on IP layer, by discovering new router prefix before being disconnected from the current router.

2.2.1.2 Proxy Home Agent

Proxy Home Agent (Proxy HA) [114], shown on next figure, has triple function, it performs MIPv6 functions on behalf of mobile node, communicates with home agent of the node as well it handles the routing optimisation. PHA in a 6LoWPAN surrounding would be LoWPAN edge router, and it would react as a normal MIPv6 host but it will perform binding updates, host agent tunneling and route optimisation. Mobile host should just perform local binding update with credentials and to create a single tunnel to the PHA. The LoWPAN node will registrate by 6LoWPAN-nd node registration message, which includes home agent address or home prefix [13], the node Home Address and some credentials. In this case tunnel is local between LoWPAN edge router and LoWPAN node so it can be managed by a simple IPv6 extension header option with low overhead.

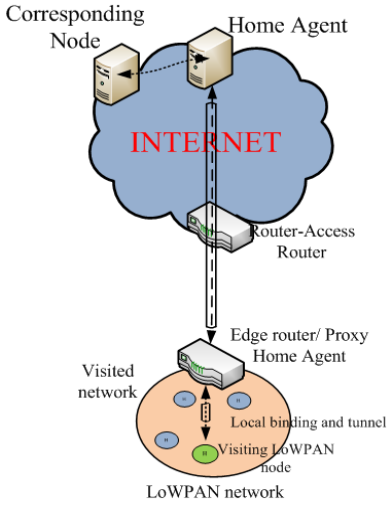


Figure 17: Proxy Home Agent

2.2.1.3 Proxy MIPv6

In M2M world it is quite often to change point of attachment in the same domain and IETF has devolved standardized Proxy MIPv6 [TKS8], that consist of a local hierarchical structure of routers which handle mobility on behalf of nodes. It is very suitable to be used for LoWPAN networks. It allows LoWPAN edge routers to proxy MIPv6 for attached LoWPAN nodes.

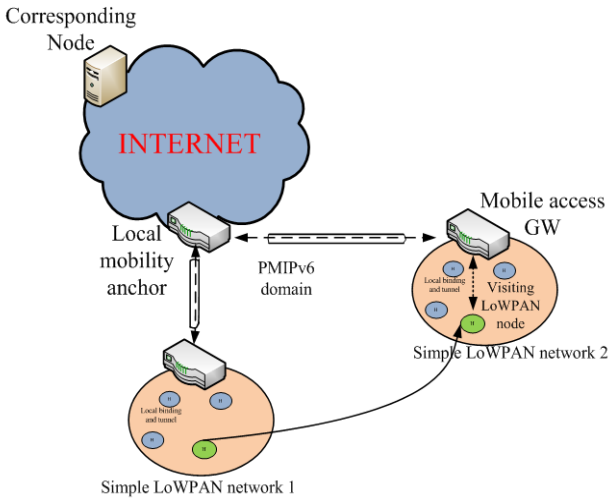


Figure 18: PMIP in 6LoWPAN

As it can be seen in the figure above, PMIPv6 domain is controlled by LMA (local mobility anchor) [114] [TKS8], and it can be integrated with home agent. LMA works with MAGs (mobile access GWs). MAGs are points of attachment regarding support for PMIPv6. MAGs are sending proxy binding updated towards LMA on behalf of mobile nodes attached to them. The mapping is then done in the LMA between this address and the temporary address of the visited MAG. In each moment there is a bidirectional channel between MAGs and LMA, enabling LMA to sent traffic towards mobile nodes static addresses (mobile node Home Address).

PMIPv6 uses RS/RA (Router Solicitation / Router Advertisement) [103] communication between mobile nodes and MAG in order to detect when one of mobile nodes has change its point of attachment. In order to apply MIPv6, each LoWPAN router must act as MAG, and also PMIPv6 means providing separate 64 bit prefix address for each mobile node. Network-based mobility idea is to expand mobile IP so that each node do not have the necessity to

run on mobile IP. The basic idea is that the edge routers in 6LoWPAN, which are full IPv6 capable (MIPv6 as well) are sufficient to cope with network mobility in general, it total for routers and nodes attached to them.

2.2.1.4 Network Mobilty (NEMO)

NEMO introduces the term mobile router, and mobile nodes within mobile network are called mobile networks nodes (MNN). If NEMO is applied in the 6LoWPAN network, even though each 6LoWPAN node is not running a mobility protocol, it can keep up session continuity for all the mobile network nodes, even when the mobile router dynamically changes its point of attachment to the Internet, through the 6LoWPAN mobile router.NEMO protocol [34], [114], enables the extension of the home agent so the agent becomes able to work with prefixes as with Home Addresses of mobile nodes.

The figure below illustrates the communication flows between nodes when using NEMO.

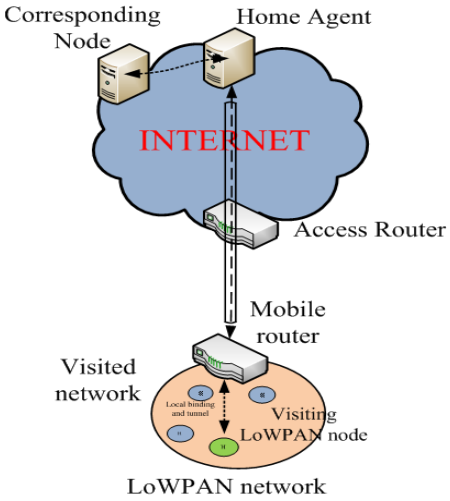


Figure 19: NEMO in 6LoWPAN

Mobile router in its communication with home agent negotiates the prefixes which are forwarded back to it. Home agent then forwards all packets that matches with bound prefix of MNNs towards mobile router. This can be a good solution for a network mobility in 6LoWPAN when mobile nodes and edge router all together are changing their point of attachment.In 6LoWPAN edge router becomes a mobile router that binds new address in the visited network with home LoWPAN prefix.

In practice this means there is no change visible inside LoWPAN network when network mobility occurs, because LoWPAN uses still the same prefix as in its home network. Home agent then transfers all the data destined to the same prefix using a tunnel between HA and edge router. The disadvantage of NEMO is that it cannot deal with individual node mobility on behalf LoWPAN nodes unless MIPv6 is installed on nodes itself or if we use home agent or PMIP. Prefix delegation can be done by DHCPv6 (the version of the Dynamic Host Configuration Protocol (DHCP) for Internet Protocol Version 6 (IPv6) networks).

2.2.1.5 Nested NEMO & MANEMO

In order to support 6LoWPAN mobility, the 6LoWPAN router is sending binding messages towards HA and vice versa. Since the 6LoWPAN packet format has no solution to compress or support a mobility header compression for exchanged binding messages, and in order to reduce the signaling overhead, a compressed mobility header, introduced by Lightweight NEMO protocol can be used. When Lightweight NEMO protocol [60] is introduced in 6LoWPAN network, a 6LoWPAN mobile router obtains the current network ID (16 bits) from the beacon message, and each time when mobile router moves to another 6LoWPAN

network, the router exchanges RS/RA messages directly with 6LoWPAN GW. The RA message includes the global IPv6 prefix of the current 6LoWPAN network and the 16-bit CoA option.

Nested Nemo [73] [90] is the case where mobile nodes can travel to other fixed or mobile networks.

MANEMO is required in 6LoWPAN networks. Ad hoc LoWPAN network can be created when mobile router and mobile nodes converge to the edge of the Internet, making possible that router and nodes can offer Internet access to each other. MANEMO is a concept of combining MANET (Mobile Ad-hoc Network) and NEMO protocols that is attracting a lot of attention in vehicular communication network and post disaster emergency network. With the application of MANEMO in 6LoWPAN networks we can avoid the nested NEMO since MANET is consisting of 6LoWPAN mobile routers that can directly communicate with each other, thus avoiding the problem with NEMO itself, where we had the nesting of 6LoWPAN routers which causes end-to-end packet delays. When using MANEMO in 6LoWPAN possible problems are networks loops, unoptimized paths and multiple access to the internet. MANEMO solution is still in its initial phase but it should be very important for 6LoWPAN.

2.2.1.6 Other mobility protocols

There are number of mobility protocols that can be introduced in 6LoWPAN network.. such as:

- SHIM6 (Site Multihoming by IPv6 Intermediation) [34],that enables small sites to be multihomed without requiring independent IPv6 address for the site,
- LISP (Locator/Identifiers Separation Protocol) [19] which introduces the separation of device identity (end-point identifier) from its location (routing identifier)
- home based protocols like HAWAII (Handoff-Aware Wireless Access Internet Infrastructure), transparent to MIP, where location management is distributed and the home agent is in the domain home router;
- Cellular IP, able to work with MIP; intended to provide local mobility and handover support, where location information is stored in distributed data bases.

2.2.2 Routing protocols in 6LoWPAN

It is very important to consider IP routing for 6LoWPAN in the sense of two different flavours of routing can happen: as routing inside a LoWPAN, i.e. intra-LoWPAN routing and as routing between a LoWPAN and another IP network, i.e. border routing. Both forwarding and routing can be executed at Layer 2 or at Layer 3. L2 forwarding, well-known as “Mesh-Under” places routing functions at the link layer occur when routing and forwarding happen at layer 2, they are performed based on layer-2 addresses, i.e., 64-bit EUI-64 or 16-bit short addresses. To forward the packet to its final Layer 2 destination, the node needs to know its address, the final destination address, the source Layer 2. L3 routing, well-known as “Route-Over” places all routing functions at IP layer. In contrast to layer-2 mesh forwarding, layer-3 Route-Over forwarding does not require any special support from the adaptation layer format.

6LoWPAN Routers in general execute forwarding on a single wireless interface and unlike standard forwarding on IP routers, forwarding is done by receiving and sending packets via the same interface. A 6LoWPAN is a set of 6LoWPAN Nodes which share a common IPv6 address prefix which means that IPv6 address remains the same independently of position of a node in a 6LoWPAN. The 6LoWPAN architecture introduces IPv6 stub networks meaning that there are no transit networks between two different subnets.

Routing typically involves one or more routing protocols created by MANET or ROLL Working Groups.

2.2.2.1 MANET based protocols

Routing protocols created by MANET Working Group are:

- AODV (The Ad hoc On-demand Distance Vector) [12],
- DYMO (Dynamic MANET On-demand) [44]
- OLSR (Optimized Link-State Routing) [101].

AODV protocol enables mobile ad hoc multihop networks by quickly establishing and maintaining routes between nodes in such a way that creates routes to destinations when needed for data communications, and only maintains actively used routes. After routes have been established they are simply utilized by IP for forwarding in AODV protocol.

DYMO protocol makes use of the same types of route discovery and maintenance messages as AODV. It has many improvements compared to AODV such as improved convergence in dynamic topologies, use of the common MANET packet format [RFC5444], support for a wide range of traffic flows, consideration for Internet interconnectivity and it takes hosts and routers into account.

OLSR algorithm is a proactive link-state routing protocol. OLSR routers regularly exchange topology information with other routers. The flooding of this information is controlled by the use of selected multipoint relay (MPR) nodes. These MPR nodes are used as intermediate routers, and thus enable a kind of clustering technique. OLSR is not very well served to 6LoWPAN Routers because of the large amount of signaling and routing state.

2.2.2.2 RoLL based protocols

The routing protocol created by RoLL (Routing over Low power and Lossy) Working Group is RPL (IPv6 Routing Protocol for Low power and Lossy Networks). RPL is a routing protocol designed for IP smart objects networks and embedded applications. RPL has been created to face routing issues in LLNs (Low-power and lossy networks).

The IETF workgroup ROLL recently proposed RPL as a standard routing protocol for low power and lossy networks [105]. RPL builds a Directed Acyclic Graph (DAG) in which paths are created from nodes towards the root. These are called Destination Oriented DAGs, or DODAGs. A DODAG offers redundant paths to increase reliability of the network. If topology permits, there is always more than one path from each leaf node to the DODAG root [57].

Following the design of RPL, we assume that each node obtains a rank in the network which indicates its distance to a common destination (sink). The nodes' ranks form a gradient, i.e., the further away from the destination, the larger the rank. Nodes having the same rank are called siblings. The interested reader is referred to [114] and [113] for a complete description of the RPL routing protocol.

2.2.2.3 Neighbor Discovery

Neighbor Discovery (ND) [103], also known as "One-hop routing protocol", was originally designed for interfaces in LAN environment and always-on equipment. Standard ND for IPv6 is not appropriate for 6LoWPAN because of assumption of a single link for an IPv6 subnet prefix and that nodes are always on.

6LoWPAN Neighbor Discovery provides an appropriate link and subnet model for low-power wireless and minimized node-initiated control traffic. Hosts play a special role in LoWPANs, and the ND bootstrapping process allows them to attach to a LoWPAN without the need to participate in routing. 6LRs (6LoWPAN Routers) [113] respond to Router Solicitation (RS) messages from 6LNs (6LoWPAN Nodes - other hosts or routers) [113] with Router

Advertisement (RA) messages. RAs contain the needed prefix and context information for a node to discover the LoWPAN and autoconfigure its addresses.

2.3 State-of-the-Art for Address Translation Schemes

In this section we present the existing methods and mechanisms of IP address translation for the following kinds of addresses (translate between IP and each of these kinds):

- Vehicular Identification Numbers (VIN)
- ZigBee
- 6LoWPAN

2.3.1 VIN – IP Address Translation Schemes

The Vehicle Identification Number (VIN) is a 17 characters alphanumeric code that uniquely identifies a vehicle worldwide. This code has been standardized in ISO-3779 in 1981 and other standardization bodies' implementation of this code (NHTSA, SAE) is compliant with the mentioned standard.

The VIN is used to uniquely identify a vehicle and therefore must appear on each vehicle. Some public information related to a vehicle can be obtained knowing its VIN code. This possibility is used in thefts prevention by assisting law enforcement authorities in tracing and recovering parts from stolen motor vehicles, or reporting vehicle history to sell/buy a used car.

In this section we review other uses of VIN numbers related to address generation and addressing schemes.

2.3.1.1 Vehicle Identification Number (VIN)

The VIN is an alphanumeric code used by the automotive industry to uniquely identify vehicles worldwide. A similar format was used by General Motors in the USA since 1951, and until 1981 there was no standard format globally defined for this information so each car manufacturer used its own [50].

The National Highway Traffic Safety Administration (NHTSA) proposed a standard format for the VIN number in 1981. Two ISO documents, ISO-3779 [50] and ISO-3780 [51], describe the format, allowed values and sections specifications.

The VIN is written in 17 alphanumeric characters with a combination of the 10 Arabic numbers and 33 capital letters of the Latin alphabet. Excluded letters are I(i), O(o) and Q (q). This is to avoid confusion between these characters and the numbers 0 and 1.

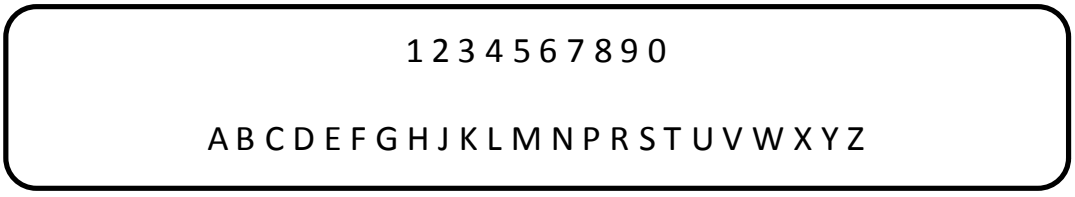


Figure 20: Alphabet for the generation of VIN codes

Although the definition of the code differs from USA, where the NHTSA definition is used [15], and Europe where the ISO [50] is used, both VIN codes are compliant and do not cause confusion. There are some other implementations of the same code, for example within the Society of Automotive Engineers (SAE) and Australian Design Rules (ADR) which are also compliant with the European ISO format of the VIN [100].

As depicted in the below figure, the VIN code contains three sections.

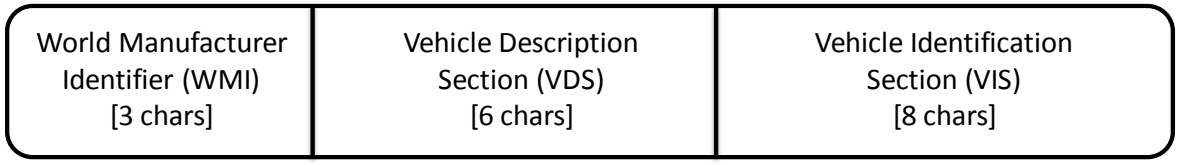


Figure 21: VIN Code Sections

2.3.1.1.1 World Manufacturer Identifier (WMI)

The WMI is the first part of the VIN code. It is 3 digits long (1 to 3 included) and uniquely designates the car manufacturer as well as the continent and the country of this manufacturer. The 3 digit codes are defined in ISO-3780 and the database is maintained by the SAE [51]. A WMI code can be revoked but cannot be used to designate another car manufacturer before 30 years since its revocation date.

The ISO-3780 standard specifies the use of each bit in the WMI code. From the left to the right, the first bit value designates the region of the car manufacturer. Values [A-C] are reserved for Africa, [1-5] for North America, [8-9] for South America, [J-R] for Asia, [S-Z] for Europe and [6] for Oceania. Multiple alphanumeric values can be assigned for the same region, depending on the needs. Some values are reserved for future use.

Second bit indicates a country in the region designated by the first bit. Multiple values are possible for the same country if needed. For example, if the first bit is V (in Europe) and the second bit is one the letters (F to R), then the designated country is France. Unique identity of a country is assured by the combination of these two values. The assigned codes by country are maintained by SAE and listed in ISO-3780.

Third character of this section designates a national unique value for the car manufacturer, maintained by national authorities. It is possible to assign more than one value to the manufacturer if needed. Thus, a unique identification of the car manufacturer can be obtained by the combination of the three values.

The ISO-3780 standard defines two types of car manufacturers depending on the number of cars built each year. A manufacturer that builds less than 500 cars per year uses the value 9 in the third position of the WMI. Therefore, in order to uniquely define all the manufacturers that build less than 500 cars a year in the same country, 3 extra bits ranging from position 12 to 14 (included) are used to create a unique manufacturer identifier. If the manufacturer builds more than 500 cars a year, it has one or more identification number(s) depending on the needs. For example, “1FA” and “3FA” identifies Ford in USA and Mexico, respectively. “VF3” is one of the WMI codes of French manufacturer Peugeot.

2.3.1.1.2 Vehicle Description Section (VDS)

The VDS is the second part of the VIN code. It is 6 characters long and gives more information about the vehicle. The description of the vehicle is not unique and each manufacturer has its own mapping table for this section; that is, a same character may have different meanings depending on the manufacturer, and sometimes differ upon the vehicle model.

The information given by this section may relate to the vehicle weight, the model, the engine type, the body style or the engine power, for example. Some real life examples for “Ford” cars are shown in [35]. It is also possible for the manufacturer to fill this section with “dummy” information if it does not want to use it, as spaces (blanks) are not allowed. Therefore, the manufacturer may not rely on this section to complete the unique identification of a vehicle.

Last position of this section (9th digit) is called the check digit. Like the TCP checksum, the check digit is the result of a standard algorithm where the values of other positions are used to generate this value. Possible values for the check digit are numbers 0 to 9 and the letter X [100]. The algorithm is described in further details in [76].

2.3.1.1.3 Vehicle Identification Section (VIS)

The VIS is the third section of the VIN code. It is 8 characters long and, combined with the VDS section, uniquely identifies a vehicle within a car manufacturer for 30 years [50]. The combination of the VIS and the WMI section allows to uniquely identify a vehicle worldwide [76] [15]. This section goes from the 10th digit to the 17th.

Digit number 10 designates the year model. It is the year during which the vehicle has been manufactured, or the vehicle model year depending on the manufacturer choice. For this digit, the allowed values are 1 to 9 (0 forbidden) and uppercase Latin alphabet letters except I, O, Q, U and Z.

Years from 1980 to 2000 are coded with letters from A to Y; years from 2001 to 2009 are represented with digits from 1 to 9; year 2010 and later are represented with letters from A to Y. This representation gives a cycle of 30 years during which a VIS code is guaranteed to be unique.

11th digit of the VIN designates the plant of the manufacturer where the vehicle has been assembled. For example, 'E' refers to 'Kentucky Truck' plant (Jefferson county, Kentucky) and 'T' refers to 'Otosan Kocaeli Assembly' plant (Kocaeli, Turkey), both plants belong to the manufacturer 'Ford'. The same value may represent different plants depending on the manufacturer.

Positions 12 to 17 represent the sequential identification part which is unique on the production line and assigned by the car manufacturer. Digits from 12 to 14 (if digit 3 of WMI is '9') represent the rest of the WMI code of the car manufacturer that produces less than 500 cars a year. Otherwise (general case), positions 12 to 17 are considered as a whole and generally considered as a sequential number. ISO-3779:2009 specifies that the last 4 positions must be numeric, which is applied in Europe. In North America (Canada and USA), the last 5 digits must be numeric for some kind of vehicles and only the last 4 digits for the rest.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ISO-3779	**WMI**			**VDS**						**VIS**							
More than 500 cars a year	**WMI**			**Vehicle Attributes**					**Check Digit**	**Model Year**	**Plant Code**	**Sequential Number**					
Less than 500 cars a year	**WMI**			**Vehicle attributes**					**Check Digit**	**Model Year**	**Plant Code**	**Manufacturer Identifier**			**Sequential Number**		

Table 1: Summary of the VIN code, sections and use

2.3.1.2 Setting an IPv6 address using a Vehicle Identification Number

Creating an IP address using Vehicle Identification Numbers has been described in the US Patent number 2005/0273505 A1 [10] of which Samsung Electronics Korea is the applicant.

This patent describes how to set an Internet Protocol address (IPv6) from a given VIN number. The method detects the special values that give inherent information about the vehicle (as described previously) and creates an IPv6 address accordingly. To be accurate, what is claimed is the generation of EUI-64 (Extended Unique Identifier) Interface ID used in Stateless Address Auto-configuration (SLAAC) to obtain an IPv6 address with an advertised IPv6 prefix. SLAAC is defined in RFC 4862 and part of Neighbor Discovery Protocol.

The below flowchart (extracted from the patent) summarizes the proposed method. In the first step, values representing the necessary information are extracted from the VIN. Some digits only will be used. The chosen information relates to the manufacturer nationality, the name of the manufacture, the vehicle classification, the kind of the vehicle, the particulars of the vehicle kind, the manufacturing date, and the serial number of manufacture. Some of these values have fixed positions in the VIN. The rest of the values must be located by the manufacturer itself, since their positions (in the VDS, for instance) are fixed locally.

The second step of the process is about converting the obtained values into a more suitable format. This operation is necessary in order for all these values to fit into a restricted 64 bits area. A mapping operation is done: “A” is mapped into 0, “B” into 1…and “9” into 35, respectively. These 36 possible values can be expressed by the use of 6 consecutive bits. The converted values are then combined to generate a number on 62 bits. The remaining 2 bits are for the U/L (Universal/Local) bits of the Interface ID part of the IPv6 address.

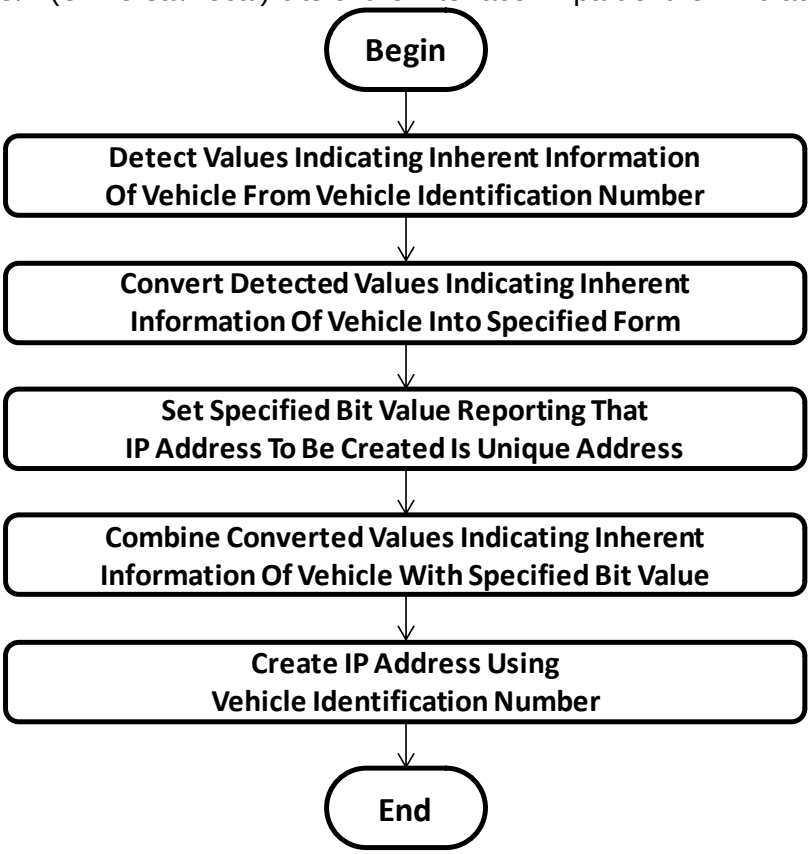


Figure 22: Flowchart of the process of setting an IPv6 address from a VIN

The figure below is one example of the procedure. These values are extracted from the patent and set as an example. First mapping step gives the following result: “k” is mapped

onto 10 and therefore as 001010 in binary; “M” is mapped into 12 and then 001100 in binary; “H” is mapped onto 7 and then 000111 in binary; “E” is mapped onto 4 and then 000100 in binary; “N” is mapped onto 13 and then 001101 in binary; “X” is mapped onto 23 and then 010111 in binary; “A” is mapped onto 0 and then 000000 in binary, “B” is mapped onto 1 and then 000001 in binary; and finally, the last four digits that are numeric, can be expressed on 14bits in binary; that is 01001000110100. The total amount of obtained bits is 62. After setting the U/L bits, the obtained EUI-64 in HEX is 28:30:71:0C:5C:00:52:34, and that is the final result.

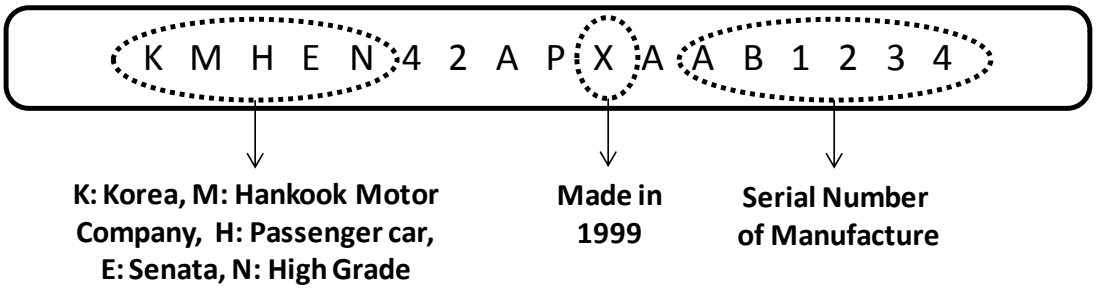


Figure 23: Extraction of information from the VIN

2.3.1.3 Setting an IPv6 address for a component of a Vehicle using a Vehicle Identification Number

A method to determine the IPv6 address of a component inside a vehicle is described in [67]. The Vehicle Identification is determined and stored in the first part (network prefix) of the IPv6 address and the identification of the component in the vehicle is stored in the Interface ID part of the address. The components identified by the generated addresses can be accessed from outside the vehicle. This work is related to the project “Security in Embedded IP-based Systems (SEIS)” in Germany.

This method proposes to decompose the IPv6 address in several parts as depicted in the below figure. The network field part is composed of 4 sections. The Global Identification section is 32 bits long and contains global identification information for the vehicle manufacturer. The Brand section is 3 bits long and identifies the brand of a vehicle (its model). The 37th bit is reserved for future use. The following section is the VIN-7. This section value is extracted from the VIN code by using the 7-digits vehicle identification section (two alphanumeric and 5 numeric characters). The host field starts with 48 reserved digits. The rest (16 bits) can be used to address control units in the vehicle.

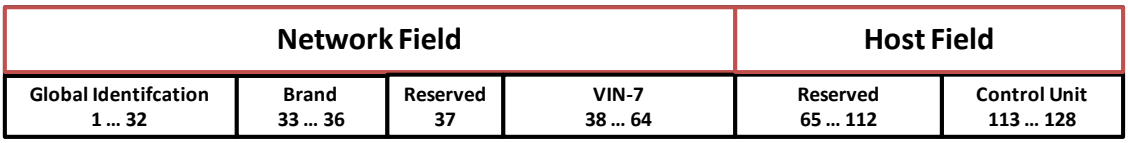


Figure 24: A possible addressing architecture based on the method

This method allows the identification of the control units inside the vehicle (in the host field) and builds a full IPv6 address for each control unit of each vehicle (different values for the Vin-7).

VIN numbers are proper to the automotive field. Similar work to set an IPv6 address from an IMEI (International Mobile Equipment Identity) number has been described and proposed within 3GPP. However, there is no standard method to do the same for vehicles.

Along with the previous two patents, other methods use the VIN numbers to uniquely identify a vehicle in large databases in a remote diagnostic scenario, for example.

2.3.1.4 Setting an IPv6 address with C2C Net ID

The GeoNet project [42] contributes to enhance vehicle communications by implementing an ETSI specification of a geographic addressing and routing protocol that supports IPv6 and uses it to deliver non-critical safety messages between cars, and between cars and the roadside infrastructure within a designated destination area (geographic area). Geographic addressing and routing allows information dissemination among nodes within a designated geographic area. C2C NET is in charge of information dissemination over multiple hops to ensure that vehicles receive the needed information within the destination area.

IPv6 over C2C NET [72] proposal intends to specify the use of IPv6 over C2C Net link. The C2C reference architecture is presented in the below picture. In order to use IPv6 addresses in this architecture two assumptions are made. The IPv6 address generated in such a network stack is formed by a combination of a topologically correct network prefix and a C2C NET ID. The latter is defined as a 64-bit unique code assigned to each C2C Interface, which is similar to the EUI-64 standardized by IEEE. The second assumption states that a given IPv6 prefix uniquely identifies a given C2C NET link. That is, a dedicated geographic area around an Access Router (also called roadside units).

C2C Net ID is used in stateless IPv6 unicast address auto-configuration mechanism (Interface Identifier part) combined with a valid advertised prefix. This information in used at the network layer level. For multicast (geo-broadcast) uses, the C2C area ID is allocated to the geographical areas. A packet for geo-broadcast is sent with the C2C area ID that matches to geographic area.

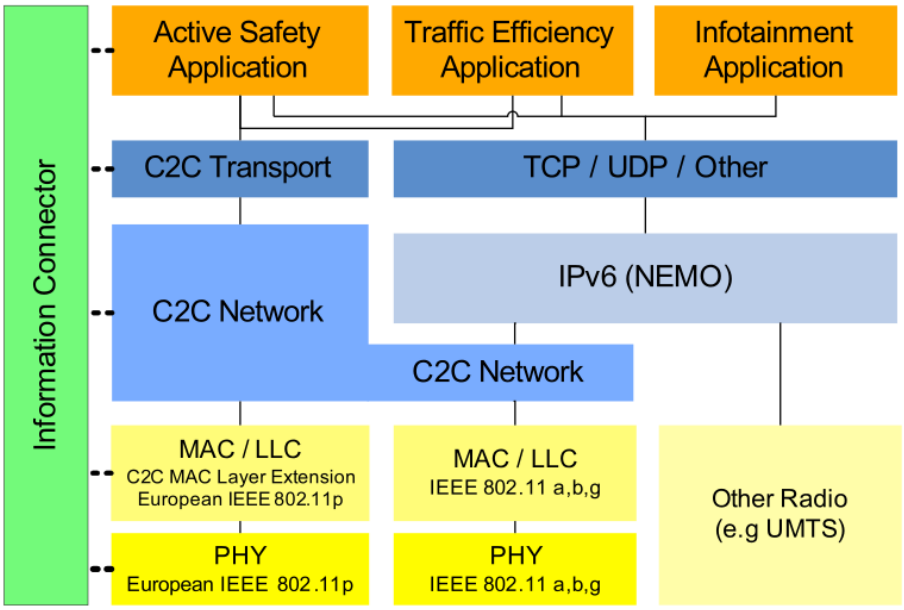


Figure 25: C2C Architecture [72]

We believe that such approach can greatly benefit from the use of the Vehicle Identification Numbers, in order to give practical values to the C2C Net ID at the network layer. The mechanisms presented in section 4.2 describe how to achieve that.

2.3.2 ZigBee to IP Address Translation Schemes

Due to the simplicity of devices considered in capillary networks, they operate using different addressing schemes depending on the vendor and application. Meanwhile LTE-M network will handle IP addresses, either IPv6 or IPv4. As one of the key points of the EXALTED project, it is necessary to provide continuous connectivity among nodes, outside and inside the capillary networks. That requires the definition of mechanisms that allow the following:

- Accessing nodes within a capillary network from external nodes.
- Accessing to public servers from the capillary network.

LTE/LTE-A does not support capillary networks in the way that LTE-M does, so the address translation procedure introduced by EXALTED will be a novelty in this kind of networks. Up to day, there is some research focused on how to provide E2E connectivity between nodes in the capillary networks and out of it, on one side IP and on the other one ZigBee or similar. The most common solution is the implementation of a web server that maps ZigBee addresses into ports in the IP interface of the gateway node [36]. Recently, ZigBee Alliance has proposed a methodology based on XML RPC calls aiming at providing continuous connectivity from the IP networks to capillary [115]. Due to the particularities of M2M communications analyzed in EXALTED both schemes are being investigated in order to select the one that better fits in the project framework.

The figure below depicts how this translation is done at the gateway node.

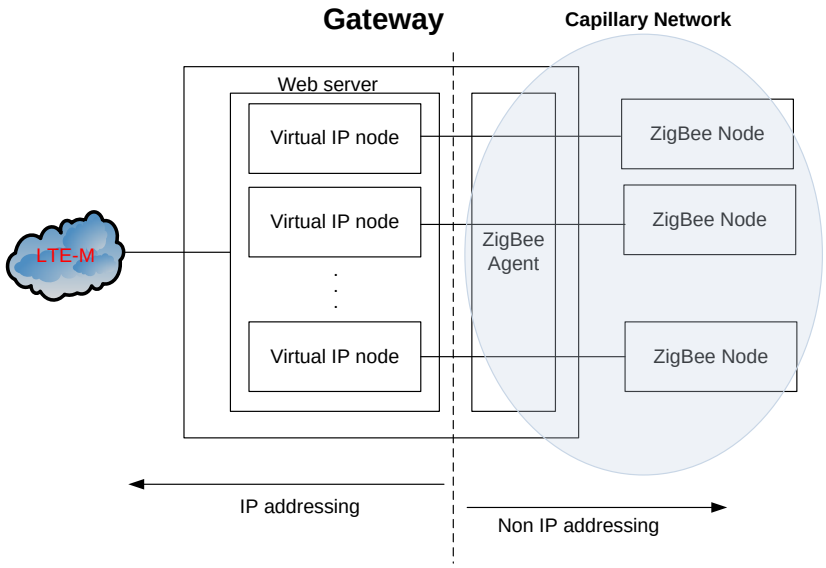


Figure 26: Address Translation between LTE-M and Capillary Networks

The resulting mechanism that is described on section 3.4 aims to provide the same functionalities identified on the previously mentioned literature, while reducing to the most the introduced overhead.

In order to be able to properly study which is the best option, it is important to describe firstly all elements and technologies involved on the way from Non-IP devices to elements placed on the core IP network.

Adapting the elements to fit into EXALTED architecture, and looking always for a solution suitable to be implemented and measured over real equipment available in the project, the following technologies have been identified as key for the definition of the algorithm:

- *Near Field Communication* (NFC). It is not directly related with the address translation mechanism, but it is used as user interface. The way to initiate and force communications from devices is done by NFC tags. This way, the amount of data transmitted from nodes is tightly related to this standard, so it is depicted on section 2.3.2.1.
- *802.15.4 Standard*, responsible for the communication between end nodes and the gateway in charge of the address translation. It is necessary to study the format of

frames related to this standard, so as to be able to parse them and then translate it into IP packets on the gateway, and vice versa. This study can be seen on section 2.3.2.2.

2.3.2.1 Near-Field Communications (NFC)

NFC comprises a number of standards related to mobile phones, intelligent cards and related devices. Its goal is to establish communication (via radio interface) between two elements by touching them together or bringing them into distances of no more than a few centimetres.

NFC standards define not only communication protocols, but also data exchange formats. Its physical behaviour is based on existing radio-frequency identification (RFID) standards such as ISO/IEC 14443 and FeliCa [77]. They also include ISO/IEC 18092 [49] and those defined by the NFC Forum.

Some already deployed application examples using this technology are data exchange, contactless payments, and a way to start/set up more complex communications (for instance WiFi). The communication also enables that one side of the transaction is not powered. These kinds of elements are called "tags".

There is a plethora of compliant tags to use within this standard, and, depending on the model selected, the way to store and retrieve the data, as well as the length of that data may vary. For that reason, it is important to clearly define which tag is going to be used in the study, so the mechanism is aware of what to look for and where.

For EXALTED purposes, as it is oriented to low cost devices, the most lightweight and cheapest one is selected: MIFARE Ultralight.

MIFARE is the trademark of widely used chips for smart and proximity cards, owned by NXP Semiconductors (spin off from Philips Electronics in 2006). It is placed in Eindhoven, the Netherlands, and main other business site.

MIFARE ultralight technology is especially suitable for low cost applications, regarding not only costs but also maintenance.

The memory embedded on each tags is of 512 bits as ISO 14443A standard says. It can be read up to 10 cm away from the active NFC reader, and it is not powered. The memory is structured in pages of 4 bytes each one. The first two pages contain the unique identifier of the tag. The third and fourth pages are reserved, and from the fifth page the memory is available for user data.

For testing the EXALTED mechanism, this fifth page is used, so the amount of data read and used from tags is 4 bytes.

2.3.2.2 IEEE 802.15.4

IEEE standard 802.15.4 defines the lower layers of a particular wireless personal area network (WPAN), characterized by its objective about low-speed and low-cost ubiquitous communication. More precisely, the stress is put in exploiting low power consumption related to the communication of nearby low cost devices, translated into extremely low manufacturing and operation costs associated to technological simplicity, but without sacrificing flexibility or generality.

The typical use case associated to it consists on 10-meter communications range with 250 kbit/s throughput. Nevertheless, configurations including lower data rates and/or higher coverage areas are also possible.

The standard defines the behaviour of the two lower layers from the OSI model. The physical layer (PHY) manages the physical RF transceiver, is in charge of channel selection and manages energy and signal. It is possible to operate in three different unlicensed frequency bands:

- 868.0 - 868.6 MHz (Europe) with one communication channel (2003, 2006)
- 902 - 928 MHz (USA) with ten channels (2003), extended to thirty (2006)
- 2400 - 2483.5 MHz (worldwide) with sixteen channels (2003, 2006)

The medium access control (MAC) layer is defined to enable frame transmission through the physical channel. It also manages the access to the physical channel and network beaconing. There other important functions as guaranteeing time slots, frame validation, and handling node associations.

On top of that two layers defined by the standard, a *routing mechanism* is needed in order to establish the mesh network (topology enabled by 802.15.4 standard) and secure the transactions.

For that purpose, several approaches are commercially available. Among them, *ZigBee* is the most known, due to the large amount of chips that integrate it. But it is not the most suitable one for EXALTED purposes, even though it is optimized for low power devices.

Other options as *DigiMesh*, a proprietary protocol from Digi manufacturer, have some advantages when compared to ZigBee. First of all, it defines just one type of devices, instead of three of them (coordinator, router and end device) as ZigBee does. It translates into easier integrations in terms of needed code and programming costs.

DigiMesh is a proprietary peer-to-peer networking topology for use in wireless end-point connectivity solutions. The nature of its peer-to-peer architecture allows DigiMesh to be both easy to use and equipped with advanced networking features, including support for sleeping routers and dense mesh networks. Overhead associated with the protocol and data payload is optimized for network performance and addressing is made simple so less time is spent defining the network, and more time on the application [33].

DigiMesh is an ideal solution for:

- Robust mesh networking, with dense network support.
- A power-optimized protocol with support for sleeping routers for power-sensitive or battery-dependent applications.
- An easy-to-use protocol that simplifies mesh networking (no need to define and organize coordinators, routers or end-nodes).
- The ability to deploy wireless solutions in both 900 MHz & 2.4 GHz.
- Deployable on low-cost/low-power and long-range products.
- Peer-to-peer structure is easy to develop around and deploy.

There are two ways of transmitting information through DigiMesh protocol: *transparent mode* and *API mode*. The one selected for this application is API mode.

When operating in transparent mode, the modules act as a serial line replacement, whereas API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a UART Data Frame.

The frame structure for API mode can be seen on the figure below. Note that the frame varies depending whether it is sent or received.

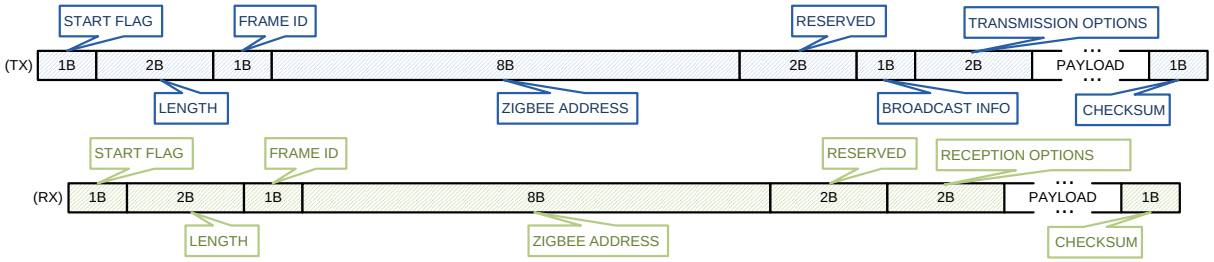


Figure 27: DigiMesh Frame structure for transmission and reception

With this definition, and the specific implementation of the payload (which is application dependent and is defined on section 3.4.1) it is sufficient to start the study of the address translation mechanism.

2.3.3 IP to Capillary Address Translation Schemes, IP to 6LoWPAN

General characteristics of constrained nodes used in capillary networks could be shortly summarized as:

- slower CPU,
- little memory,
- little storage.

Also main characteristics of constrained networks are:

- small bandwidth,
- very low TX power
- small packet size,
- lossy and instable links.

Due to these limitations of capillary networks and nodes, 6LoWPAN was developed to enable Wireless Embedded Internet by simplifying IPv6 functionality, defining very compact header formats from the one side, and from the other side taking into account the nature of wireless networks. 6LoWPAN standards actually enable the efficient use of IPv6 over low-power, low-rate wireless networks on simple embedded devices through an adaptation layer and the optimization of related protocols. The result of 6LoWPAN is the efficient extension of IPv6 into the wireless embedded domain, thus enabling *end-to-end* IP networking and features for a wide range of embedded applications. In this section, IP/Capillary address translation for the IP devices (6LoWPAN) that are behind the gateways in the capillary networks will be analyzed. More about 6LoWPAN protocols could be found in the section 3.3.

Since in LTE/LTE-A there is no support for communication with 6LoWPAN networks as it supposed to be in LTE-M network, address translation procedure should be defined for EXALTED.

Features of the IPv6 design such as a simple header structure, and its hierarchical addressing model, made it ideal for use in wireless embedded networks with 6LoWPAN. Additionally, by creating a dedicated group of standards for these networks, the minimum requirements for implementing a lightweight IPv6 stack with 6LoWPAN could be aligned with the most minimal devices. Originally 6LoWPAN was targeted IEEE 802.15.4 radio standards and assumed layer-2 mesh forwarding [39], later on it was generalized, with additional support for IP routing, for all similar link technologies [Zach].

Communication between 6LoWPAN Nodes and IP nodes in other networks happens in an end-to-end manner, just as between the ordinary IP nodes. 6LoWPAN Nodes are identified by a unique IPv6 address, and are capable of sending and receiving IPv6 packets. A simple IPv6 protocol stack with 6LoWPAN is almost identical to a normal IP stack with differences that 6LoWPAN only supports IPv6, for which a small adaptation layer (called the *LoWPAN adaptation layer*) has been defined to optimize IPv6 over IEEE 802.15.4 and similar link layers in [39]. IEEE 802.15.4 standard is the most common 2.4 GHz wireless technology for embedded networking applications, and has been used as a baseline for 6LoWPAN development. Other technologies used with 6LoWPAN include sub-GHz radios, long-range telemetry links and even power-line communications. The 6LoWPAN group has defined encapsulation and header compression mechanisms that allow IPv6 packets to be sent to and received from devices connected to IEEE 802.15.4 based networks which are shortly described in 3.5.2.2.. IPv6 nodes are assigned 128 bit IP addresses in a hierarchical manner, through an arbitrary length network prefix. IEEE 802.15.4 devices use 16 bit addresses that are unique within a PAN. The mapping between 16-bit short addresses and the IPv6 addresses is necessary in order to communicate with external IPv6 networks.

The gateway architecture for the interoperability between 6LoWPAN and external IPv6 networks is proposed in [61]. The gateway does the compression and decompression of IPv6 packets and performs the mapping between 16 bit short addresses and the IPv6 addresses for both the external IPv6 networks and 6LoWPAN, respectively.

In [61], [39] it is recommended for the gateway to maintain mapping table(s) for this translation. The gateway may have the internal and external device address mapping tables (internal and external):

- *Internal Device Address Mapping Table*
  This table consists of 64-bit interface identifier (IID) and 16-bit short address. This table MUST contain the mapping information of all devices in the 6LoWPAN. The maximum size of the mapping table is 2^16 entries
- *External Device Address Mapping Table*
  This table consists of 128-bit IPv6 address, 16-bit short address and ET (Expiration Time).

The edge router expands the compressed LoWPAN and IPv6 headers to a full IPv6 header along with the UDP header if compressed. Incoming packets are also processed at the edge router, compressing IPv6 and UDP headers as much as possible.

IPv4 interconnectivity: 6LoWPAN natively supports only IPv6, however often it will be necessary for 6LoWPAN Nodes to interact with IPv4 nodes or across IPv4 networks. There are several ways to deal with IPv4 interconnectivity, including IPv6-in-IPv4 tunneling and address translation. These mechanisms are typically collocated on 6LoWPAN Edge Routers, on a local gateway router, or on a node configured for that purpose on the Internet.

Generally, M2M systems include M2M modules integrated into embedded devices together with an Internet-based back-end system. The M2M module measures and controls the device, and communicates over IP with the back-end M2M service. More recently, M2M gateways to local embedded networked devices have become more common. Thanks to native IP, 6LoWPAN networks can be connected to M2M services through simple routers and thus 6LoWPAN can be considered to be a natural extension of M2M. In LTE-M networks a M2M gateway is responsible for IP address translation between LTE-M and capillary networks, in this case between LTE-M and 6LoWPAN. More about this could be found in section 4.4.

2.4 Vehicular Network Use-Cases

The Future Internet (FI) paradigms consider mobility as a first-class citizen – it is taken into account at the most fundamental principles of their design. Future Internet mobility is not simply an added patch on an existing design, as is with Mobile IP and Internet today. For example, as mentioned earlier, the existing protocol SHIM6 (a good candidate for FI designs) is using a unique IPv6 address at a new layer in the stack (the 'shim') which allows for dynamic mappings between it and temporary addresses acquired during movement.

Or, the vehicular settings represent the mobility application by excellence – entities are almost always in movement (as opposed to devices deployed in, e.g., offices). It is for this reason that vehicular networked communications are an excellent driver for the design of Future Internet protocols.

A number of applications of vehicular communications have been described in deliverables of project EXALTED. For example, in [116], the following vehicular scenarios are presented in detail:

- Remote Monitoring of Vehicle data: mileage, engine, temperature.
- In-Vehicle M2M diagnosis: in-vehicle wireless check (e.g. rear-light bulb).
- Parking time check.
- Vehicle collision management.
- Gateway vehicle for car-to-car communications.

One of these scenarios – the 'gateway vehicle for car-to-car communications' – is being addressed in more detail in this document: the capillary-to-capillary-to-infrastructure communication mechanisms are proposing novel techniques to realize address auto-configuration and route setup between vehicles and the infrastructure.

2.4.1 Requirements from communications in vehicular environments

The vehicular networks use case is the most demanding scenario, due to its inherent device and communication complexity. Every single node must be embedded on the vehicle, and must be capable of communicate to peers either on the same vehicle or remote ones. Moreover, each vehicle could be seen as a capillary network itself, made of several sensors distributed and centralised by a smart unit with gateway capabilities.

On a typical transportation application, each vehicle should be able to receive data from different sources such as GPS position; information from applications using LTE-M network or data received directly from other vehicles forming a mobile capillary network. In any case the equipment incorporated to the vehicle must be capable of handling all those data in a reasonably time.

Regarding all of these aspects, the requirements derived from this scenario can be described as follows:

Deployment architecture	Device-to-device communication	NEEDED
	Infrastructure-to-device communication and vice versa	NEEDED
	Hybrid architecture, devices-to-cluster head	POSSIBLE
	Cluster head -to-infrastructure	POSSIBLE
Transmission mode	Simplex	NEEDED
	Half-duplex	NEEDED
	Full-duplex	POSSIBLE
Communication type	Peer-to-peer	NEEDED
	Unicast	NEEDED

	Multicast	NEEDED
	End to end communication	NEEDED
	Broadcast	NEEDED
Network partitioning	Clustering	POSSIBLE
	Cluster-head selection	POSSIBLE
Devices' transmission range	Short Range (<1m)	NEEDED
	Medium Range (<100m)	NEEDED
	Large Range (>100m)	NEEDED
MAC cooperation	Homogenous MAC cooperation (one distinct MAC)	IRRELEVANT
	Heterogeneous MAC cooperation (from different systems)	IRRELEVANT
Device capabilities	Power consumption	HIGH
	Bandwidth usage	NARROWBAND
	Delay	MINIMUM
Gateway	Address translation (between LTE-M and capillary network)	NEEDED
	Maximum supported devices (Scalability)	VEHICLE: Fixed Number eNODE B: Limited Number
	Power consumption (Batteries or plugged)	HIGH (Plugged)
Mobility models	Nodes are static and sinks are mobile	NEEDED
	Nodes are mobile and sinks are static	NEEDED
	Both nodes and sinks are mobile	NEEDED

When referring to power consumptions both on devices and gateway, HIGH will mean that the equipment must be permanently connected to a power source (either the electrical net or the vehicle battery).

On the scalability requirement (Maximum supported devices), the nodes deployed on the vehicle will be installed by the manufacturer, so it will be a fixed number of them always connected, with no possibility to add or remove some of them. On the other hand, when a M2M network of vehicles is managed by a Gateway, the maximum supported vehicles will be given by the Gateway specifications, but, in any case, the number of nodes could vary and grow until hundreds of vehicles for instance.

2.4.2 Requirements for M2M IP Networking

M2M IP networking is an architecture and set of protocols which relies on the Internet Protocol suite; it is adapted for an M2M environment (machine-to-machine communications); as such, it inherits requirements from IP and at the same time it needs to satisfy requirements generated from the constrained environments typical for M2M.

Category	Description	Requirements	
Mobility Cardinality	This category of requirements refers to the requirements related to differences between mobility of a single entity (cardinal 1) and mobility of groups of entities (cardinal n).	Support single mobile entity	NEEDED
		Support a set of mobile entities, moving together as a group	NEEDED

- Mobility of entities should be considered from the standpoints of single mobile entity as well as groups of mobile devices;
- Network connection establishment between two M2M entities (at network layer) should be realised as fast as possible, in order to accommodate movements as fast as possible.

- The addressing architecture should allow for very numerous nodes (millions of vehicles, thousands of individually-addressable devices per vehicle);
- At the same time, the addressing architecture should allow for devices having very limited resources (a trade-off may be needed, because long addresses, although good for describing a large address space for the previous requirements, are difficult to implement in devices using a small memory footprint).
- The typical communication profile should be formed by messages between two end points. Group or multiparty communications should be considered as exceptions, although they may be supported.
- The ratio of effective payload data vs. headers of IP packets should be maximised whenever possible. This is a particularly challenging requirement, knowing that the devices are numerous (i.e. large addresses in the headers) and that the typical data exchange is formed of very simple status or trigger data (small payload).
- For each addressable device, there should be a possibility to exchange data directly with any other similar addressable device;
- The direction of the initiation of communication should be both ways: communication initiated from A to B should be possible as well as from B to A.
- Whenever there is choice of adding intelligence between end systems and intermediary systems the end systems should be preferred.
- A wide range of link layer topologies should be supported, from simple point-to-point, to point-to-multipoint, to bus-like shared links, to star-like topologies.
- A wide range of higher-layer applications should be supported.
- Protocol use: the existing communications protocols should be reused whenever possible; in case of need, extensions to existing protocols should be added, prior to considering designing entirely new protocols.
- When and if a new networking system is designed, it should be interoperable with the existing deployed IP Internet; if needed, gatewaying mechanisms should be considered; the number of entities in the existing IP Internet which would need to be modified to interoperate with an M2M E2E networking system should be minimised.
- The routing algorithms (build paths, find paths) together with the addressing architecture should scale well: it should be possible to gradually add new entities over time, and the growth in network size should not add too much strain on the existing system.
- The typical communication pattern should be considered as intermittent and bursty: very short datagrams, relatively periodically, with very varying periods: from seconds to weeks to years.
- The algorithms should not be compute intensive and place very little load on the computer.
- Packet-switched data communications should be considered first, and circuit-switched should be considered when providing certain benefits to M2M (battery consumption, computing cycles).

Self-configuration and reliability: the entities connecting to the M2M networking system should be self-configurable (in terms of networking parameters); reliability should be built in: when a communication path fails an alternative path should be proposed by an inner building block of the system, and not by an extension.

3. Mechanisms for M2M Communications for Use-Cases of EXALTED

In this section we present several novel mechanisms designed for the use of IP protocols in the context of machine-to-machine communications with an application to capillary-to-capillary-to-infrastructure communications (such as V2V and V2V2I vehicular communications).

We present a new address and route auto-configuration protocol for V2V2I which allows, in its most extensive form, the communications between machines in different vehicles as well as machine-to-infrastructure communications, and with maintenance of ongoing sessions and reachability at permanent addresses (full mobility management).

In building the capillary networks, one often needs specific address translation mechanisms. For capillary networks we describe mechanisms of address translation between IP, LTE-M, ZigBee and 6LoWPAN.

The mechanisms presented in this section consider the following:

- The IP addressing within a vehicle may be decoupled from the addressing outside the vehicle. Different scopes of addresses are used in these two zones. Within a vehicle, an IP ULA (Unique Local Address) addressing mechanism that is based on VIN (Vehicle Identification Number) is proposed.
- It is possible that within a vehicle the ZigBee and/or 6lowpan-compatible link layers are used. In these cases, there is a need of a translation scheme between IP and ZigBee and/or 6lowpan. We describe two different mechanisms for this problem.
- The dynamic establishment of addresses (auto-configuration) and routing paths between vehicles is achieved with the mechanism of capillary-to-capillary-to-infrastructure exchanges.
- A vehicular network (a network deployed within a vehicle) is considered to be conceptually the same thing as a capillary network. However, there may be a need of an analysis to show that these mechanisms remain consistent for other use cases and scenarios. This may constitute future work.

3.1 Requirements

Several requirements have been established by project EXALTED and documented. Among those requirements, we have selected the ones most pertinent for capillary-to-capillary-to-infrastructure communications and M2M addressing schemes in vehicular environments. These requirements are listed below. The full list of requirements can be found in [116].

ID and Title	FU.1 – Support of large number of devices
Priority	Mandatory
Dependencies	NT.10
Description	M2M services are expected to require that a large number of devices (larger than the capacity of cellular systems for human-to-human communication) are connected simultaneously to one base station, either directly or through a gateway. The following aspects have to be considered: • Identification of a large number of devices both individually and by group (e.g. connected via a gateway) • Differentiation between M2M and non-M2M devices • Tracking the state of each M2M device (e.g. active, inactive, sleeping, “dead”) • Provision of ubiquitous device management (e.g. when the devices move to different cells), mobility management (for both directly and

	indirectly connected devices) • Management of different device / service classes (e.g. different priority levels)
Rationale	Supporting a large number of devices is a mandatory crucial requirement for most M2M services. The incapability of current cellular networks to support a very large number of devices is one main reason that the envisioned M2M services and applications cannot be supported. Thus, technical solutions that fulfill this requirement are of particular interest.

ID and Title	**FU.6 – Unique identification of devices**
Priority	High
Dependencies	NT.16
Description	M2M Devices in EXALTED system should be able to uniquely identify and refer to each other. Inside capillary networks, devices that are not shared globally may still have non-global names (e.g. a NAT IP-address). The latter is required to support the existing technology that does not support global identifiers.
Rationale	This is needed to identify the devices (sensors, actuators, or gateways) which are part of the M2M services.

ID and Title	**SV.5 – Delegation and distribution of functionalities**
Priority	Mandatory
Dependencies	NT.4
Description	The capacity for the gateway to assume functions otherwise located in other nodes of the network. Delegation has to be tightly controlled by the entity that delegates.
Rationale	Delegation is a mechanism that allows a secured distribution of functions in order to prevent congestion at a single access point; the benefit is a better use of network resources, a better control over application deployment and device management. Delegation, while interesting to split network load over several nodes, requires tightly secure schemes. Moreover, delegation requires the support of multi-hop communication (see NT.4).

ID and Title	**NT.1 – Heterogeneous networks**
Priority	Mandatory
Dependencies	
Description	EXALTED should support heterogeneous networking between LTE network and all types of capillary networks such as Bluetooth network, IEEE802.15.4 network, IEEE802.11 network, etc. This should include both ad-hoc and infrastructure-based networking.
Rationale	Network access technologies for M2M devices widely vary, based on scenario, application, or location. EXALTED system should accommodate the promising network access technologies for capillary networks.

ID and Title	**NT.3 – Minimum number of modifications in the network infrastructure**
Priority	Mandatory
Dependencies	NT.2
Description	LTE network infrastructure, e.g. LTE base stations, is already deployed. The introduction of LTE-M will require some modifications. The number of modifications should be kept at minimum.

Rationale	Such an approach will minimize the upgrade costs from LTE to LTE-M. If possible, the RF processing should not be affected, but only the baseband processing. Preferably, the required modifications can be done by means of software updates.

ID and Title	**NT.4 – Support of multi-hop communication**
Priority	Medium
Dependencies	
Description	Retransmission of neighbor's data or transmission of their own messages to a pair, in case the neighbors or the node itself is not able to reach the sink/aggregator/gateway node.
Rationale	This is one of the EXALTED general assumptions when talking about capillary networks. For large sets of M2M devices, the multi-hop requirement appears as critical in order to achieve the auto-deployment and self-healing characteristics of this kind of networks. This requirement, combined with the appropriate routing protocols, enables scalability of capillary networks.

ID and Title	**NT.6 – End-to-end device-to-device communication**
Priority	Mandatory
Dependencies	NT.4
Description	M2M devices can establish and maintain direct E2E communication, with or without the support of the core LTE-M network. If using the core network, the communication will be between devices on different environments (i.e. different capillary networks or between a capillary network and devices connecting directly to the LTE-M network). The core network may not be used if the communication is between nodes within the same capillary network or between capillary networks operating in the same technology (if gateways support direct communication between pair within coverage).
Rationale	Developing smart M2M devices capable of establishing such communication will reduce the operational cost.

ID and Title	**NT.7 – Flexible addressing scheme**
Priority	Mandatory
Dependencies	NT.3
Description	Each device should be able to be reached using a unique address, both from inside its own network and from any other point connected to the LTE-M network. Moreover, LTE-M network should handle variations on the population of a cell in an efficient way, by assigning each device an address, reusing the ones previously freed.
Rationale	In order to be able to enable all the features intended to be demonstrated on the EXALTED project regarding routing protocols, mobility, device management, etc., the use of an addressing scheme is critical. An IP-based scheme would be the most desirable scheme in order to reach all the requirements, but, as some M2M devices may have little resources, at least other simple addressing schemes must be implemented. Then, the gateway should take care of routing in an appropriate way the data coming from and going to each node.

ID and Title	**NT.8 – Mobility management**
Priority	Mandatory
Dependencies	

Description	Mobility management is responsible for the handover of M2M devices and gateways. This requirement strongly depends on the characteristics of the use cases.
Rationale	Particularly for the use case of *Intelligent Transportation System* (ITS), we expect mobile M2M devices. It may happen that a big number of devices request a handover at the same time, or the gateway of one complete capillary network requests a handover. On the other hand, assuming stationary devices, mobility tracking can be switched off to reduce signaling. Therefore, mobility management has to be highly scalable and needs particular consideration in the design of the EXALTED system.

ID and Title	**NT.12 – Self-diagnostic and self-healing operation**
Priority	Medium
Dependencies	DV.1
Description	Detecting network status and dynamic reconfiguration when topology changes.
Rationale	The network itself should evaluate its status and detect changes in the topology regardless of events that cause those changes. LTE-M network should adapt to their capabilities and available resources for offering the best QoS possible to the connected devices.

ID and Title	**NT.15 – End-to-end session continuity**
Priority	Mandatory
Dependencies	NT.6, NT.8, NT.9
Description	Regardless node activity (mobility), session should be maintained although handover process will be triggered.
Rationale	An active session is maintained regardless of the network topology and the number of network nodes involved in communication between devices. Session continuity is provided on all LTE-M network segments. Legacy IP and 3gpp (wireless) protocols should be also supported.

ID and Title	**NF.1 – Scalability**
Priority	Mandatory
Dependencies	NF.6
Description	EXALTED system should be scalable with the number of devices, networks, as well as with the demand.
Rationale	Local capillary networks of M2M devices may have a very high density (hundreds, thousands, or even more) of sensors, and/or actuators. EXALTED system needs to support communication to and from these in a way that respects the other non-functional requirements as well as between capillary networks through the LTE network with potentially trillions of sensors and actuators, and billions of users world wide.

ID and Title	**NF.6 – Address space scalability**
Priority	High
Dependencies	
Description	Future-proof address concept needs to be developed.
Rationale	Due to the foreseen shortage of MSISDN addresses, number of devices that can be deployed may be limited and scalability of the system impaired.

ID and Title	DV.7 – Protocol translation at the gateway
Priority	Mandatory
Dependencies	NT.1
Description	The capacity to provide protocol translation between capillary networks and LTE/LTE-M protocols in order to interface two different network technologies.
Rationale	Typical Exalted scenarios involving gateways show groups of devices belonging to a capillary network exchanging information with a remote application server. In order to reach the remote server, the devices get WAN access through a LTE/LTE-M gateway. Such a gateway extracts the data received through its connection to the capillary network and forwards them to the distant server using its LTE/LTE-M connectivity realizing protocol translation between the capillary network protocol and the LTE/LTE-M protocol.

ID and Title	DV.8 – Information routing at the Gateway
Priority	Mandatory
Dependencies	NT.1
Description	The capacity for the gateway to select a path in the network for the traffic it forwards.
Rationale	A gateway may be connected to different PDN or other network equipments and shall be able to select the correct path in order to minimize the number of hops packets will suffer before reaching the end point of the communication.

3.2 Capillary-to-Capillary-to-Infrastructure Communications

Capillary-to-capillary-to-Infrastructure communications are exhibited in the scenarios of vehicular communications. Considering that one capillary network is actually a network deployed within a vehicle, it becomes immediately relevant that vehicle-to-vehicle-to-infrastructure communications are equivalent scenarios. One particular use case is “remote monitoring of vehicle data” – two vehicles are in range, but only one is covered by LTE-M; the second vehicle obtains connectivity to Internet from the first; a server in the infrastructure queries the data offered by a machine-class device deployed in the second vehicle.

A number of other scenarios which are pertinent to M2M (e.g. smart metering) are not currently described in this document. Further analysis is needed in order to identify whether and how capillary networks are applied in a smart metering environment (e.g. could a capillary network be deployed in a house network, etc.) This represents future work.

When a node connects to an IP network, it has to be configured in order to be able to communicate with other nodes present in the network. The configuration of the node relies on two steps that have to be successfully completed. The first step is the *addressing scheme*: the node needs a unique IP address[1] to be identified in the network. The second step is the *routing scheme*: the node needs to fill in its routing table in order to be able to send IP datagrams to specific destinations in the IP network. At least one entry has to be added in the routing table: the *default route*. Once both these parameters (the network prefix

[1] In IPv6 networks, the IP address of a node is split in two parts: the *routing prefix* part and the *interface identifier* part. The interface identifier can be generated by the node itself based on its MAC address or using some random processes. Thus, in order to complete its IPv6 address, the node only needs the routing prefix part of the IP address. This latter is provided by the routers during the configuration phase.

and the default route) are initialized at the node, this latter is able to communicate with any other node present in the network.

In the EXALTED context, an end-to-end machine-to-machine communication is possible only if the IP-enabled end devices in the capillary network are properly configured. To this end, in this section, an addressing and routing mechanism for end devices in a capillary network is presented. This mechanism enables end devices to configure themselves dynamically in a fully automatic way. More precisely, the presented scheme deals with capillary-to-capillary-to-infrastructure communications.

3.2.1 Existing addressing and routing protocols

In the earlier sections we have presented an analysis of the state of the art with respect to addressing mechanisms in vehicular networks, address translation mechanisms, 6lowpan, ZigBee and Future Internet. The goal of that analysis was to identify the main directions along which the capillary-to-capillary-to-infrastructure mechanisms could follow towards enriching the M2M and FI landscapes. However, whenever designing any new mechanism one realizes that an existing landscape of protocols already exist in place and offers (almost) the same functionality. This "down-to-Earth" evolutionary approach to designing mechanisms for Future Internet suggests that an analysis of the state-of-the-art of existing addressing and routing protocols should be performed as well.

Thus, in this section we briefly mention the relevant existing protocols for addressing and routing: Neighbor Discovery, DHCP version 6, Prefix Delegation and other mechanisms to assign the default routes.

In IPv6 networks, several protocols exist for the dynamic auto-configuration of nodes – namely, the Neighbor Discovery Protocol (NDP) [103], the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [83] and some dynamic routing protocols such as, for example, Open Shortest-Path First (OSPF) or Routing Information Protocol (RIP). A quick overview of the functionalities provided by NDP and DHCPv6 is described below. For more details about these protocols, please refer to section 3.2.1.6 "Particularly relevant IETF methods of IETF of assigning route and prefix" in reference [80].

DHCPv6 [83] main functionality is to assign a complete IP address (i.e. both the routing prefix and the interface identifier parts) to a node. Relying on DHCPv6 to autoconfigure the IP address of a node is known as the *stateful address auto configuration* mechanism (in contrast with the *stateless address auto configuration (SLAAC)* mechanism which relies on *Router Advertisement* (RA) described further). The DHCPv6 protocol is based on a Client/Server architecture. That is, each node that needs configuration information has to send a requesting message to a DHCPv6 Server. However, as the node is not configured yet, its only valid IP address is its link-local[2] one. As this IP address is not routable, the node can only reach nodes connected to the same link. Therefore, the stateful address auto configuration mechanism requires that a *DHCPv6 Server* is present on the same link as the node. If this is not the case, a *DHCPv6 Relay* node must be present on the link. Indeed, this latter can forward the node's requesting message to a DHCPv6 Server located away from the link. The DHCPv6 protocol can also provide other configuration parameters (e.g. informations about DNS servers).

[2] In IPv6 networks, a node connecting to a network automatically generates a link-local IP address using the reserved link-local routing prefix (fe80::/10) and generating an interface identifier based on its MAC address or some random processes. This IP address has a link-local scope and thus cannot be used to communicate with nodes that are away (in terms of network hops) from the link.

Prefix Delegation [78] is an extension to the DHCPv6 protocol. In addition to the typical functionality of DHCP to assign IP address, this extension allows the assignment of a *routing prefix* to a Client. This mechanism is particularly interesting for the assignment of addresses to the Gateway of a capillary network. As described above, the DHCPv6 protocol is specified to work with Relay and Server entities; when assigning a prefix the routing paths need to be updated on both Relay and Server entities. A recent reference [64] describes a mechanism of supporting Prefix Delegation when the Relay and the Server are used. This can be useful in the case an LTE-M PDN Gateway is specified as a DHCPv6 Relay. This classic prefix delegation with DHCPv6 to a capillary network can be used when the IP address of the Gateway does not change, and when no tunnels are necessary.

Prefix Delegation for Network Mobility [84] is a specification of behavior for the existing DHCPv6 Prefix Delegation such as to work in the context of network mobility. Network Mobility (NEMO) is an extension to the Mobile IP protocol to support groups of devices moving together; such a group can be understood as a capillary network (like, for instance, a vehicular network). This particular prefix delegation mechanism specifies the placement of the Requesting Router (on the Gateway) and of the Delegating Router (the Home Agent), as well as the placement of the DHCP Relay (on the Gateway). This mechanism can be well applied in a capillary network. This can be needed for the cases where the Gateway changes its Care-of Address, and relies on the workings of a Home Agent in the infrastructure to maintain the associations Home Address – Care-of Address.

Default route assignment mechanisms exist basically under two distinct forms. The first is RA-based (the use of stateless address auto-configuration), which is part of the NDP protocol, and the second is a dynamic routing protocol (OSPF may assigns default routes in addition to exchanging specific routes). Currently these two mechanisms are the only IETF mechanisms[3] to assign a default route to an end node. Basically, in both of these mechanisms, routers periodically send signaling messages on links in which they are connected to. Among other configuration information, these messages contain a default route that can be used by the nodes present on the link.

In most deployed IPv6 networks nowadays, the addressing and routing mechanism are provided by both NDP and DHCPv6 protocols: the first one is used for the stateless address auto configuration of the nodes whereas the second is mainly used to provide additional configuration parameters that complete the SLAAC or when the stateful address auto configuration is used. DHCPv6 is also used for its Prefix Delegation extension. However, recent work at the IETF proposes new extensions to DHCPv6 and NDP in order to be able to completely configure nodes in the network using only one of the two existing protocols. Therefore, a method to provide the default route using the DHCPv6 protocol is described in [68]. More details about this mechanism can also be found in section 3.2.1.6 of reference [80]. On the other hand, works about the Prefix Delegation option using the NDP protocol are also proposed in [75] [9] [104].

The proposed addressing and routing mechanism for capillary-to-capillary-to-infrastructure communication is presented in the following section. This mechanism mainly relies on the NDP and the DHCPv6 protocols and also provides some new extensions to these protocols in order to fit in the EXALTED uses cases.

[3] In the context of IPv6 protocols, only two mechanisms to assign such routes are defined: RA-based and dynamic routing protocols. On the other hand, in the context of IPv4 protocols a different set of mechanisms exists for assigning the default route; for example, DHCPv4 does assign default routes, whereas DHCPv6 does not assign default routes.

3.2.2 Topology and algorithm applied to V2V2I

The proposed addressing and routing mechanism is detailed in this section. The algorithm is illustrated using the context of vehicular-area network as, according to sections 2.2.2 “Capillary Networks” and 3.2 “M2M IP networking” of reference [80], a vehicular network can be considered as a particular case of a capillary network. Following this concept, capillary-to-capillary-to-infrastructure communications can be substituted by vehicular-to-vehicular-to-infrastructure (V2V2I) communications, as depicted in the figure below.

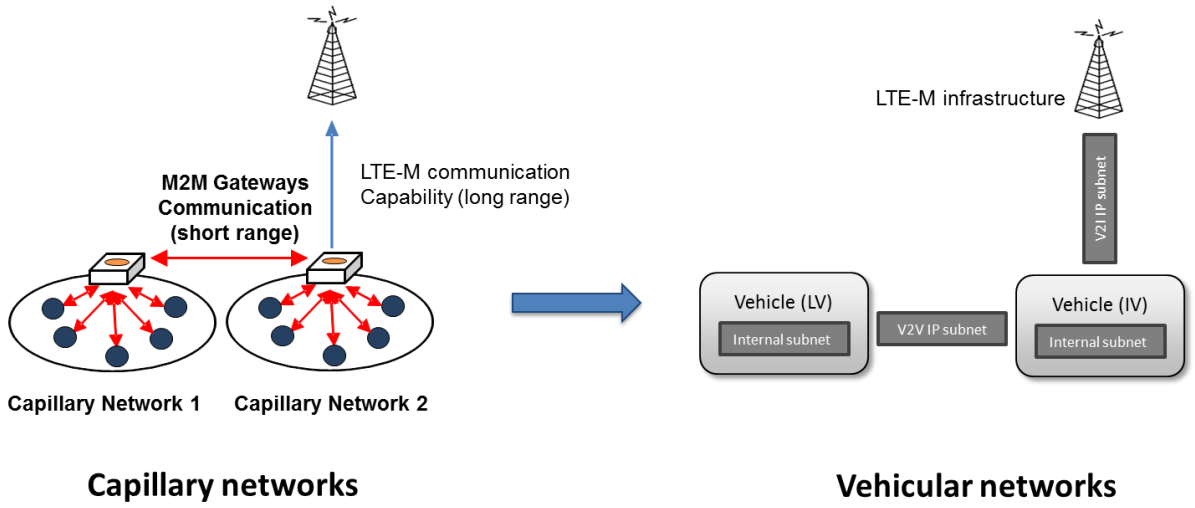


Figure 28 : Vehicular networks as capillary networks

The V2V2I communication scenario involves two kinds of vehicles: a Leaf Vehicle (LV) and an Internet Vehicle (IV). Basically, the difference between those two vehicle lies in the fact that the LV cannot access directly to the infrastructure for two reasons: 1) the M2M Gateway in the LV is not equipped with a long range egress interface (LTE/LTE-M) but only with a short range egress interface (typically Wifi) or 2) the LV is too far from the eNodeB and cannot use its LTE/LTE-M interface directly. In both situations, the LV is not able to connect to the infrastructure. Therefore, it uses its short range egress interface to connect to another vehicle (the IV) that provides access to the infrastructure. For more details about the types of vehicle that can be found in a vehicular network, please refer to section 3.2.1.3 “Types of Vehicles” in reference [80].

The proposed algorithm is divided into four steps that are presented in the next table. Each step is completed using different addressing and routing protocols functionalities that are detailed below.

Step	Active communication	Brief description	
		Vehicular context	EXALTED context
1	V	Communications between end devices and the Mobile Router, all within one vehicle, is possible.	Communications between end devices and the Gateway in a capillary network is possible.
2	V2V	Communications between end	Capillary-to-capillary

		devices in a LV and end devices in an IV is possible.	communications is possible.
3	V2V2I	Communications between end devices in a LV and an application server in the infrastructure is possible.	Capillary-to-capillary-to-infrastructure communications is possible
4	V2V2I with mobility management	Communications between end devices in a LV and an application server in the infrastructure are maintained in a mobile environment, i.e. the MRs change their egress address whereas end devices don't, thus maintaining ongoing communications.	Capillary-to-capillary-to-infrastructure communications are maintained in a mobile environment

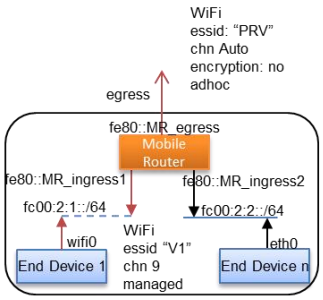
Table 2: From V to V2V2I communications

Step 1: V communications

The first step consists of enabling local communications between end devices and the Mobile Router (MR) inside a vehicle (i.e., in the generic context of EXALTED, inside a capillary network).

The *routing scheme* is completed by the NDP protocol: the MR informs the end devices via RA messages that it can be used as a default route. Thus, upon reception of the RA messages from the MR, the end nodes add a default route in their routing table with the MR as next-hop.

The next figure depicts the available communications at this point.



Leaf Vehicle

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

End Devices Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface

Mobile Router Routing Table

Figure 29 : Configuration details of the V communication

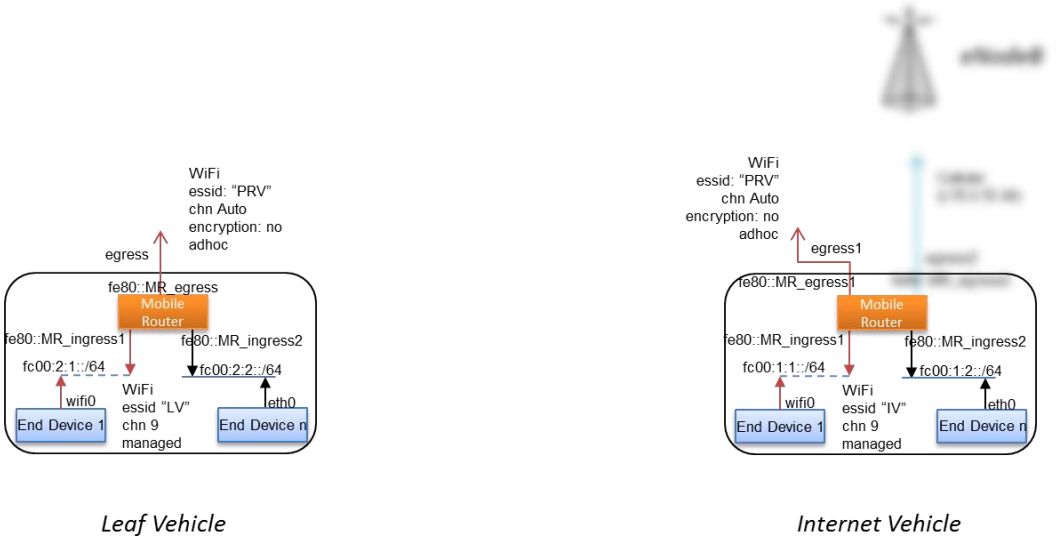
Step 2: V2V communications

The V2V step starts when a LV approaches an IV. Each vehicle has already set its own private network based on VULA addresses. Also, each end device has already set its default route. Therefore, in order to enable communications between the end devices of one vehicle and the other, the routing tables and the egress interfaces of both MRs have to be configured.

The *addressing scheme* of the MR-to-MR link is based on link-local addresses: the routing prefix part is the standardized link-local prefix (fe80::/10) and the interface identifier part is generated locally based on the MAC address of the respective egress interface of each Mobile Router.

Now that the MRs can communicate with each other locally, they have to share the prefix of their private network in order to enable end devices communication from one vehicle to the other. More precisely, the LV MR needs to add in its routing table a route to the capillary network (based on VULA) of the IV and vice versa. This *routing scheme* is completed using a new extension of the NDP protocol: a new option in the RA messages which enables prefix exchange between nodes. The details about this mechanism are not described here and can be found in [6].

The next figure depicts the available communications at this point.



Leaf Vehicle *Internet Vehicle*

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

End Devices Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:1::/48	fe80::MR_egress1	2	egress

Mobile Router Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:2::/48	fe80::MR_egress	2	egress1

Figure 30 : Configuration details of the V2V communication

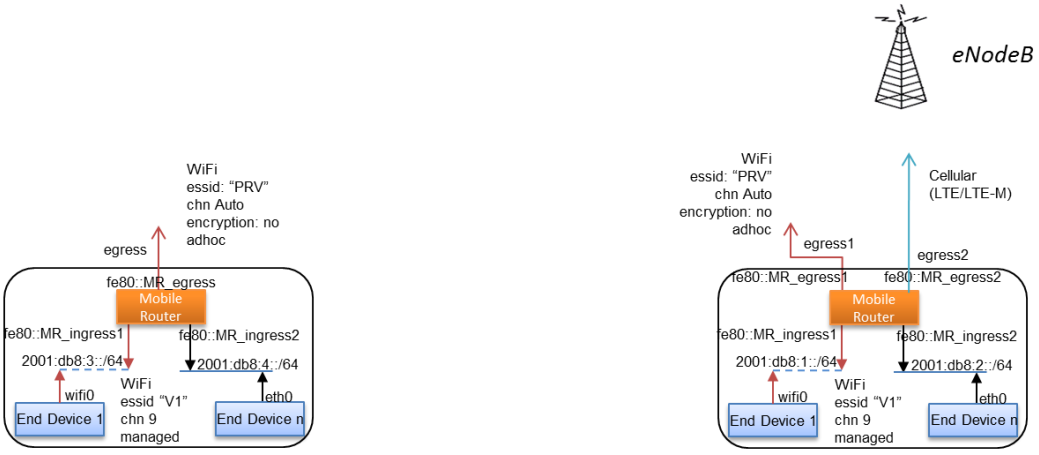
Step 3: V2V2I communications

The V2V2I step starts when the V2V communication is established between a LV and an IV. The IV can now share its connectivity to the infrastructure with the LV. Although the end devices of the LV are already configured with VULA-based addresses that enable them to communicate with the end devices in the IV, the site-scope limitation of VULA addresses does not allow the use these VULA-based addresses to communicate with the infrastructure. Therefore, the end nodes have to be configured using a global prefix.

As explained previously, the usual way to obtain a global prefix for a network is to use the DHCPv6 Prefix Delegation extension [78]. However, as the DHCPv6 protocol relies on the fact that there must be at least one DHCP Server (or DHCPv6 Relay) on the link, it may not be a reliable protocol in some cases. For example, in the V2V2I context, relying on the DHCPv6 protocol to provide a global prefix to the LV assumes that the IV provides a DHCPv6 service (i.e. the MR in the IV implements a DHCPv6 Relay or Server). In order to avoid this assumption, one solution would be to extend the NDP protocol such that it also provides the Prefix Delegation option. Indeed, the NDP protocol is a native protocol of the IPv6 stack: a node that does not implement the NDP protocol cannot configure its link-local address and therefore is unable to connect to an IPv6 network. Therefore, it seems more appropriate in our context to improve the NDP protocol with a Prefix Delegation extension rather than relying on a potential DHCPv6 implementation at the IV. Nevertheless, our algorithm relies on both protocols in order to provide a global prefix to the LV: NDP Prefix Delegation is first tried and if it fails for some reasons, then the DHCPv6 way is tried. Therefore, the *addressing scheme* in this step is provided by either NDP or DHCPv6.

The *routing scheme* part of this step consist of only one action: in order for the MR in the LV to forward IP datagrams coming from the end devices to the infrastructure, it has to add a new route entry in its routing table. This route entry is its default route and points to the MR in the IV. The MR in the LV adds this route in its routing table when it has received a global prefix from the IV (i.e. when the addressing scheme part is successfully done).

The available communications at this point are depicted in the figure below.



Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

End Devices Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:1::/48	fe80::MR_egress1	2	egress
::/0	fe80::MR_egress1	1	egress

Mobile Router Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:2::/48	fe80::MR_egress	2	egress1
::/0	eNodeB	1	egress2

Figure 31 : Configuration details of the V2V2I communication

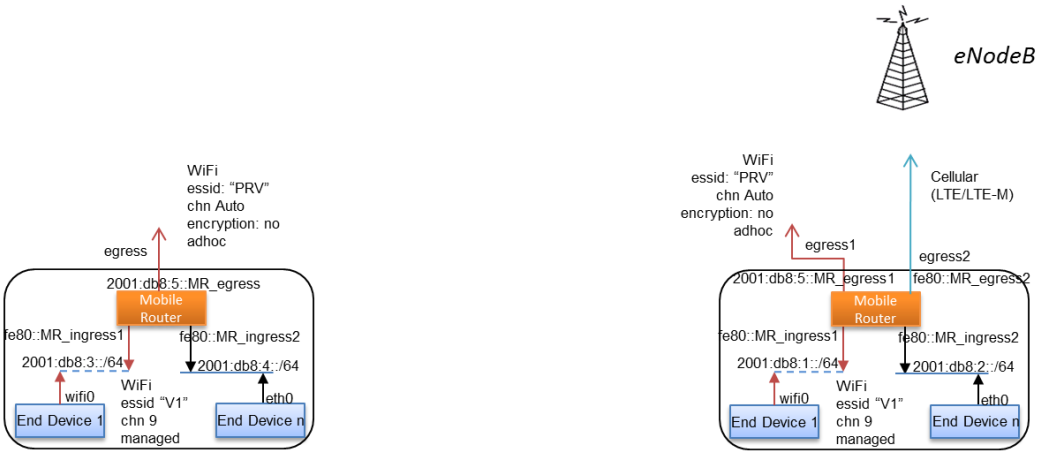
Step 4: V2V2I communications in a mobile environment.

The last step of the proposed algorithm deals with mobility management. If the LV goes out of the radio range of the IV and then uses another IV to re-establish a V2V2I communication, every current communication between the end devices in the LV and the infrastructure are reinitialized. This kind of situation can be avoided by using the Mobile IP Protocol (MIPv6) [20]. However, MIP relies on the concept of care-of addresses to work. A care-of address corresponds to the address that is used by the moving nodes to connect to the Internet. In our context, enabling MIPv6 in a LV in order to maintain the connectivity between the end nodes and the infrastructure even if the LV moves from one IV to another, means that the egress interface of the MR in the LV must have a global IP address. Indeed, up to now, the MR-to-MR communication is ensured by link-local addresses (see V2V communication step) that cannot be used as a care-of address (a CoA must be of global scope). In the current configuration, MIP cannot be used in a LV.

The *routing scheme* in this step consists in assigning a globally routable address to the egress interface of the LV (the Care-of Address, CoA). Together with this assignment, a routing table entry may be set up on the MR of IV to state that this CoA is reachable at the LV's egress interface.

In this way, the LV may use the Mobile IPv6 protocol (NEMOv6 extensions) to inform its own Home Agent about its current position. The nodes within the LV's capillary network be reachable at their addresses derived from the MNP (Mobile Network Prefix) and have session continuity. These two aspects (reachability at a permanent address and session continuity) are the two salient characteristics of mobility management using Mobile IP.

The next figure details the configuration modifications related to this last step in the V2V2I communication.



Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

End Devices Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
::/0	fe80::MR_ingressX	1	wifi0/eth0

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:1::/48	fe80::MR_egress1	2	egress
::/0	fe80::MR_egress1	1	egress

Mobile Router Routing Table

Network Prefix	Next Hop	Metric (hops)	Outgoing Interface
fc00:2::/48	fe80::MR_egress	2	egress1
::/0	eNodeB	1	egress2

Figure 32 : Configuration details of the V2V2I communication with mobility management

3.2.3 Message exchange diagrams for V2V2I

In the following we illustrate the V2V2I method by means of illustrative message exchange diagrams.

The topology used for this algorithm is the following:

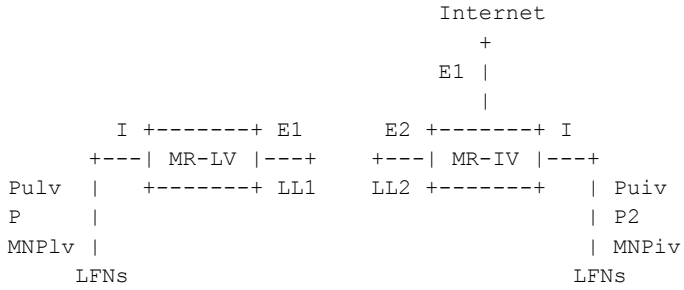


Figure 33: Topology for the V2V2I Algorithm

In this topology, we used the following notation:

- Pulv: prefix based on VULA, used in the capillary network of LV.
- P: globally scoped prefix, assigned by MR-IV to MR-LV.
- MNPlv: the Mobile Network Prefix, used for mobility management.
- MR-LV: the Mobile Router of LV.
- I: the ingress interface of LV.
- E1: the egress interface of MR-LV.
- LL1: the link-local address of E1.
- E2: the egress interface of MR-IV.
- LL2: the link-local address of E2.
- MR-IV: the Mobile Router of the IV.
- E1: the egress interface of the MR-IV.
- I: the ingress interface of MR-IV.
- Puiv: the prefix based on VULA, used in the capillary network of IV.
- P2: the globally-scoped prefix used in the capillary network of MR-IV, topologically correct at the fixed infrastructure.
- MNPiv: the Mobile Network Prefix valid in the MR-IV, used for mobility management.
- LFNs: Local Fixed Nodes, the end nodes in the capillary networks.

The algorithm is described in a condensed manner by the following figure:

```
IV->LV[Puiv]                                  [1]
LV[Puiv,LL2,E1]                               [1]
LV->IV[Pulv]                                  [1]
IV[Pulv,LL1,E2]                               [1]
V2V (Pvula)
If [P,DR] IV[ND]                              [2,5]
      V2V2I (Pglob)
      If [CoA] IV[DHCPv6]                     [7]
            V2V2I (MNP)
      End
Elseif [P] IV[ND]                             [2]
      IV[P,LL1,E2]                            [2]
      If [DR] IV[DHCP_DR]                     [4]
            V2V2I (Pglob)
      If [CoA] IV[DHCPv6]                     [7]
            V2V2I (MNP)
      End
      Else
            IV->LV[P2]                        [1]
            LV[P2,LL2,E1]                     [1]
            V2V (Pglob)
      End
```

```
Elseif [DR] IV[ND]                            [5]
      If [P] IV[DHCP_PD]                      [3]
            V2V2I (Pglob)
            If [CoA] IV[DHCP]                 [7]
                  V2V2I (MNP)
            End
      End
Else
      If [P,DR] IV[DHCP]                      [3,4]
            V2V2I (Pglob)
            If [CoA] IV[DHCPv6]               [7]
                  V2V2I (MNP)
            End
      Elseif [P] IV[DHCP_PD]                  [3]
            IV->LV[P2]                        [1]
            LV[P2,LL2,E1]                     [1]
            V2V (Pglob)
      End
End
```

Figure 34: Condensed Description of V2V2I Algorithm

This description of the algorithm is composed of two columns. The execution starts with the left column, read top-down, and then the right column in a similar manner.

The algorithm represents a view 'from above': it describes in a mixed manner the behaviors of both LV and IV. For implementation, a view 'from down to Earth' is necessary future work, where the behavior is described only from LV perspective, or only from IV.

The algorithm is organized as an alternation of preferences. Mainly, the LV acts depending on various information received from IV. For example, at line [2,5] on the left column, if the IV delivers prefix and default route by using ND then LV takes a number of actions. The same kind of behavior is repeated for all cases of using DHCPv6 or ND.

The left column concentrates on the case where IV uses ND to deliver various data to the IV (prefix, default route) and LV insists by using DHCPv6 when one such parameter is not delivered, whereas the right column is vice-versa.

First, IV informs LV about its Puiv; then LV inserts a routing table entry corresponding to Puiv, LL2 and E1. Similaryly, LV informs IV about Pulv and IV inserts a routing table entry. At this point, V2V is possible based on Pvula.

Then, on the line [2,5], if IV delivers a prefix P and a default route DR, using ND, then V2V2I communications are possible, using Pglob (or otherwise simply named P). Further, if a CoA is delivered by IV to LV by using DHCPv6, then V2V2I with mobility management is possible for LV, using its MNP.

Otherwise (line [2]), if only the prefix P is assigned by IV to LV with ND (and not the default route), then the IV inserts a routing table entry for P towards LL1 on E2. At this point V2V communications are possible.

In addition, if a default route is offered by IV by using DHCPv6, then V2V2I communications are possible, using Pglob. Moreover, of a CoA is offered by IV using DHCPv6 to LV, then V2V2I with mobility management is possible, based on the MNP.

Finally, the right column represents the behavior whereby the IV prefers the use of DHCPv6 and, optionally, ND.

The following message exchange diagram illustrates the behavior of MRs when ND is preferred initially for delivery of prefix and default route.

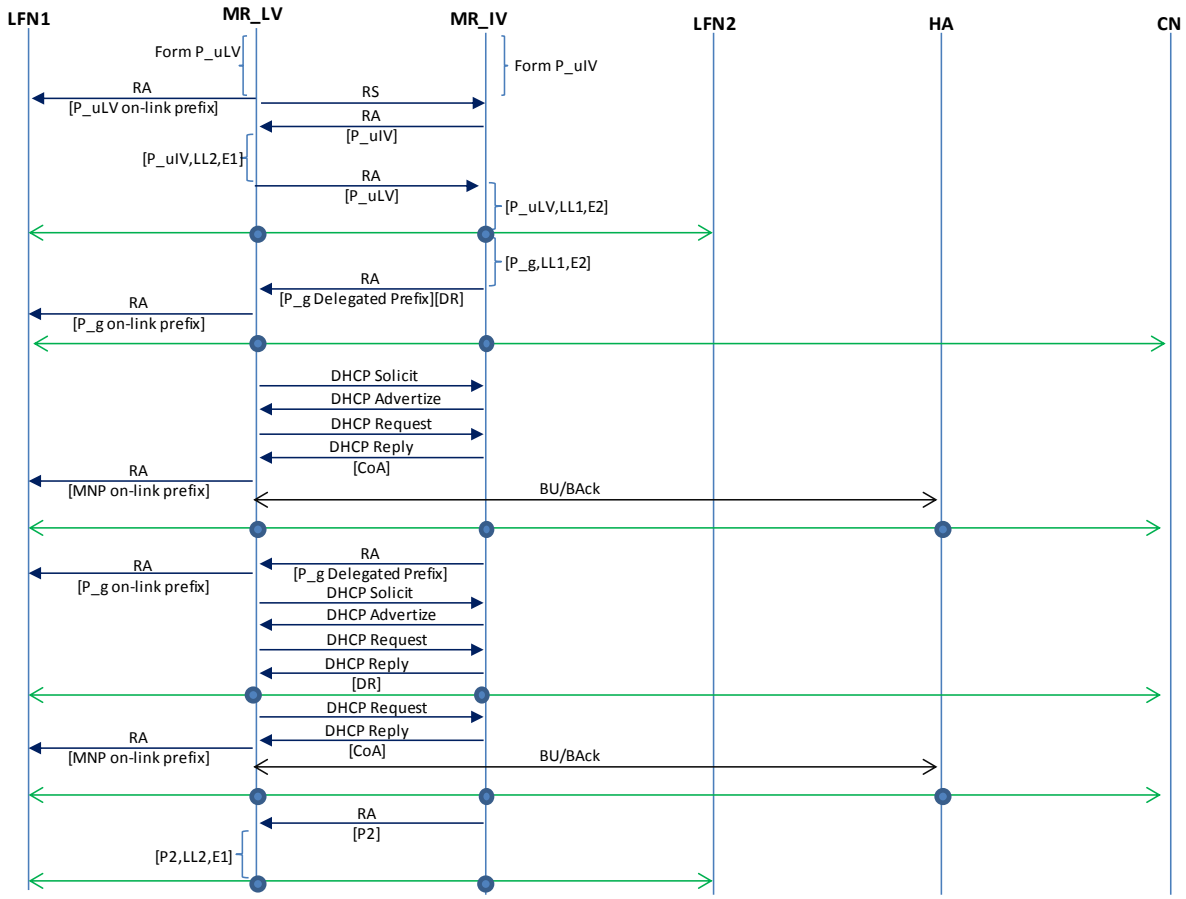


Figure 35: Message Exchange Diagram for ND Preference (left column)

The columns are representing the entities in the topology and are named as follows:

- LFN1: a Local Fixed Node, an end node, within the capillary network of LV.
- MR_LV: the Mobile Router of the LV.
- MR_IV: the Mobile Router of the IV.
- LFN2: a Local Fixed Node within the capillary network of IV.
- HA: the Home Agent of MR_LV.
- CN: a Correspondent Node of the LFN1.

Initially, MR_LV and MR_IV form prefixes based on their respective VULAs: P_uLV and P_uIV Once the P_uLV is formed, it is advertised by the MR_LV to the LFN1, by using a Router Advertisement (RA). This is advertised as a on-link prefix within the capillary network of the LV.

Subsequently, or maybe simultaneously, MR_IV and MR_LV exchange their respective prefixes using RAs on their egress interfaces. Upon reception of such an RA, a MR inserts a routing table entry corresponding to the received prefix. Upon completion of this message exchange, communication betweek LFN1 and LFN2 is possible (V2V communication).

Next, it may be possible that MR_IV has a globally-routable prefix (P_g) that it can allocate to MR_LV. Before performing this allocation, it must add an entry in its routing table stating that P_g is reachable at the egress interface of MR_LV. After this addition, it will send a RA containing that P_g and the indication of being a default router. Upon reception of P_g, the MR_LV will advertise P_g to the nodes in its capillary network. After this exchange, the LFNs in LV's network are able to communicate with arbitrary CNs in the infrastructure.

Further, if MR_LV is able to obtain a Care-of Address from MR_IV, by using DHCPv6, then MR_LV will be able to advertise the MNP (Mobile Network Prefix) to the nodes in its capillary network and, subsequently, send a NEMOv6 Binding Update to its Home Agent. At this point, the nodes in LV may use their globally-scoped permanently reachable addresses and session continuity – mobility management with V2V2I. The communication from LFN1 to CN will tunneled through MR_LV's Home Agent.

On another hand, if in the earlier step the RA only sent the P_g to MR_LV, and not the default route, the MR_LV will request the assignment of default route by using DHCPv6. After the 4 messages of the DHCPv6 exchange containing the default route, the LFN1 will be able to communicate with CN. Additionally, if the DHCP exchange provides a CoA to MR_LV, then MR_LV is again able to advertise MNP to its nodes and perform BU/Back (Binding Acknowledgement) with its Home Agent.

The last two messages show what happens in case neither the Delegated Prefix P_g nor the default route are provided neither by DHCP nor by ND. In this case it is possible for MR_IV to advertise its globally routable prefix (P2) using Router Advertisement, such that the two vehicles may still communicate in a V2V manner (as a safety case when MR_IV may not be able to generate its P_uIV).

For completeness, we indicate below the message exchange diagram for the right column of the concise representation of the V2V2I algorithm; this is the case where the initial preference is that IV uses DHCPv6 and further down the *if* branches, the ND protocol.

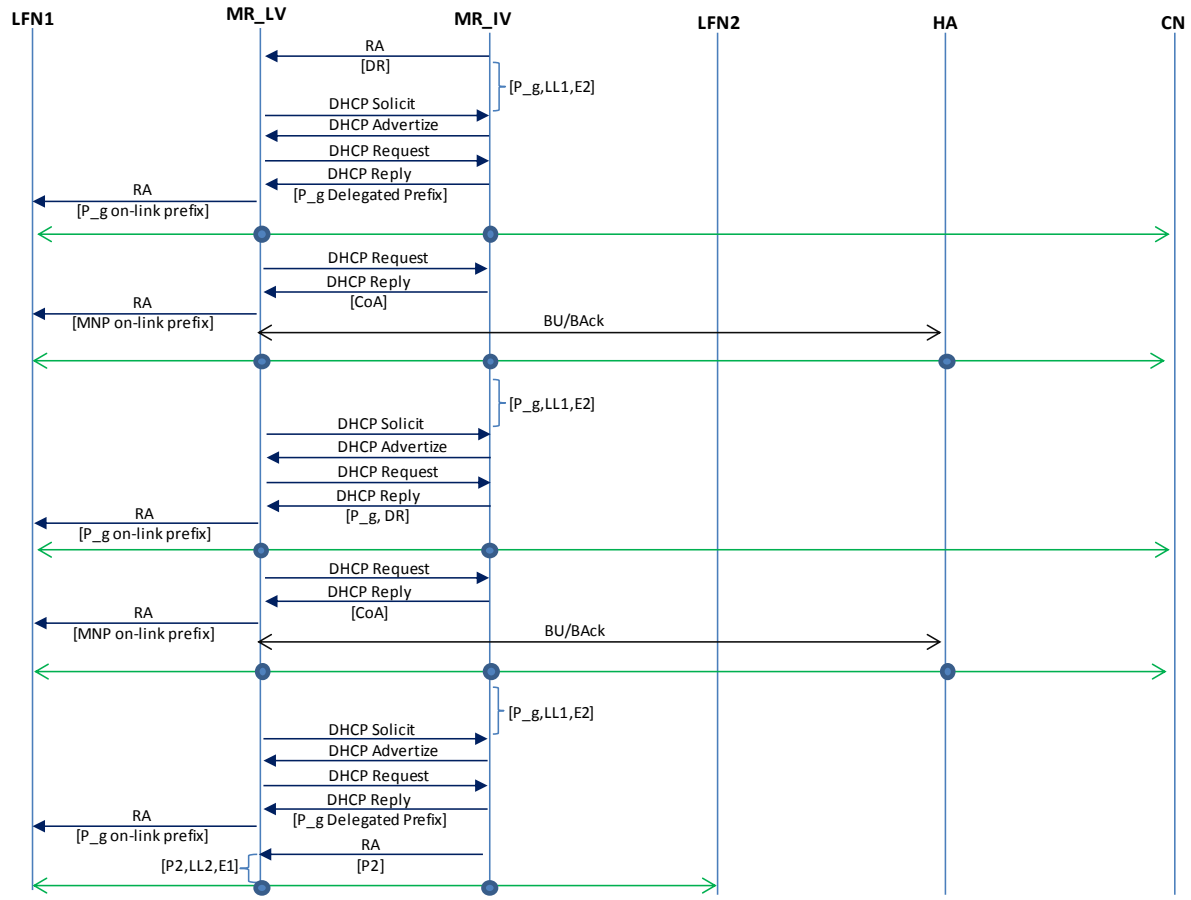


Figure 36: Message Exchange Diagram for DHCPv6 Preference (right column)

3.3 Address Translation Mechanism for Vehicular Communications

Section 3.5.1 presented the Vehicle Identification Number. The VIN code is an identifier specifically defined for the automotive industry. Two main standards define the structure and the use of it. ISO-3779 defines 3 sections for the VIN: World Manufacturer Identifier (WMI), Vehicle Description Section (VDS), and Vehicle Identification Section (VIS). The WMI uniquely identifies the manufacturer if it builds more than 500 cars a year. Otherwise, 3 additional digits from the VIS section are used to complete the identification. The VDS gives some general information about the vehicle: the engine power, body style or the overall weight. Finally, the VIS section completes the identification of the vehicle with the vehicle sequential number. The second standard, ISO-3780:2009, details the structure of the WMI and gives an updated (2009) database of the currently used codes. This database is maintained by the SAE.

Section 3.5.1 also presented some methods relating to the use of VIN codes in the networking area. Two examples of the mapping and the conversion of VIN onto IPv6 addresses are described. In the remainder of this section, a new proposition for the conversion of VIN codes to network prefix and network addresses are given. An efficient mapping method using VIN codes is described, and the use of this method to convert the VIN to an IPv6 address/prefix is also detailed.

3.3.1 Initial assumptions

As stated previously, in order to identify a vehicle two sections are important: WMI (digits 1 to 3) and VIS (digits 12 to 17). According to [63], this information may not be enough but it is certainly mandatory.

The WMI uniquely identifies the car manufacturer: region, country, and unique national code. Its validity is maintained by the SAE. The VIS is the sequential identifier of the vehicle and unique by definition. The combination of these two values is much likely [63] enough to uniquely identify a vehicle.

The VDS section can be inferred if the two other sections values are known. A multi-key query on a local database is necessary. Only the original manufacturer is able to achieve this.

3.3.2 Mapping method

The objective for the mapping method is to have the most efficient way to represent the maximum number of VIN digits in a minimum number of bits.

A straightforward and also intuitive manner is described in [70] (section 3.5.1.2). Two steps are necessary: mapping and conversion. For every VIN digit to be included in the IPv6 address, 36 values are possible. The mapping consists in representing a VIN digit in decimal and the conversion phase consists in converting this decimal to a binary number. For example, “A” is mapped onto 0 (decimal) and converted into 000000 (binary). The final binary result is 6 positions long in order to cover all the possibilities.

The proposed approach differs from the previous by relying on alphanumeric numeral systems. These numeral systems use both Arabic numbers (0 to 9) and Latin letters (A to Z) to generate numbers.

For instance, Base 36 [72] is one example that contains all alphanumeric characters and uses 36 as the radix. The 10 Arabic numerals (0 to 9) and the 26 Latin letters (A to Z) are used to represent numbers written in this numeral base. Numbers written in this numeral base can be converted to other systems (hexadecimal, decimal, octal or binary) with simple

arithmetic operations (multiplication/division). Numbers in this system are ordered as follows: 0 < 1 < 2 …< 9 < A < B …< Y < Z.

This numeral system is a good start for our mapping operation but three characters (I, O, Q) do not exist in the alphabet for the generation of VIN codes (section 3.5.1.1) and should be removed to avoid confusion. The new base that corresponds to the VIN alphabet is the VIN-Base, or the VIN numeral system. Further details about conversion from the VIN-Base to other numeral systems and the opposite, are given in the dedicated annex.

The basic idea behind the mapping proposal is the following: a VIN code can be considered as a whole and read as a number written in the VIN-Base. This assumption will help compressing the number of necessary bits to represent a certain value. The below table summarizes the benefits of using the VIN-Base rather than Base 36 and compares both methods with the initial mapping and converting methodology. For the comparison, we need to know how many bits are necessary to encode a value in a certain base. To have this number, it is necessary to know the logarithm base 2 of the maximum value to encode (the number of possible values).

Amount of digits to encode	Number of bits if the digits are Base-VIN	Number of bits if the digits are Base-36	Number of bits if the digits are independent
1	6	6	6
2	11	11	12
3	16	16	18
4	21	21	24
5	26	26	30
6	**31**	**32**	**36**
7	36	37	42
8	41	42	48
9	46	47	54
10	51	52	60
11	56	57	66
12	**61**	**63**	**72**
13	66	68	78
14	71	73	84
15	76	78	90
16	81	83	96
17	**86**	**88**	**102**

Table 3: Comparison between the amounts of bits necessary to encode a value depending on the representation

The above table and the below graphics demonstrate a clear gain in terms of encoding: the number of bits necessary to represent a certain amount of information (digits) is more important (hence less interesting) if we consider the intuitive method described in [9] rather than one of the proposed numeral bases. The gain of the VIN-Base over Base-36, due to the absence of unused values (I, O, Q), is visible starting from 6 digits to encode (first red line).

The intuitive method is a general method and usable with variable length codes. The VIN-Base and Base-36 methods take advantage of the limited number of digits in the VIN (17) to

gain a number of bits in the final conversion. These methods are designed for the VIN mapping and converting use case.

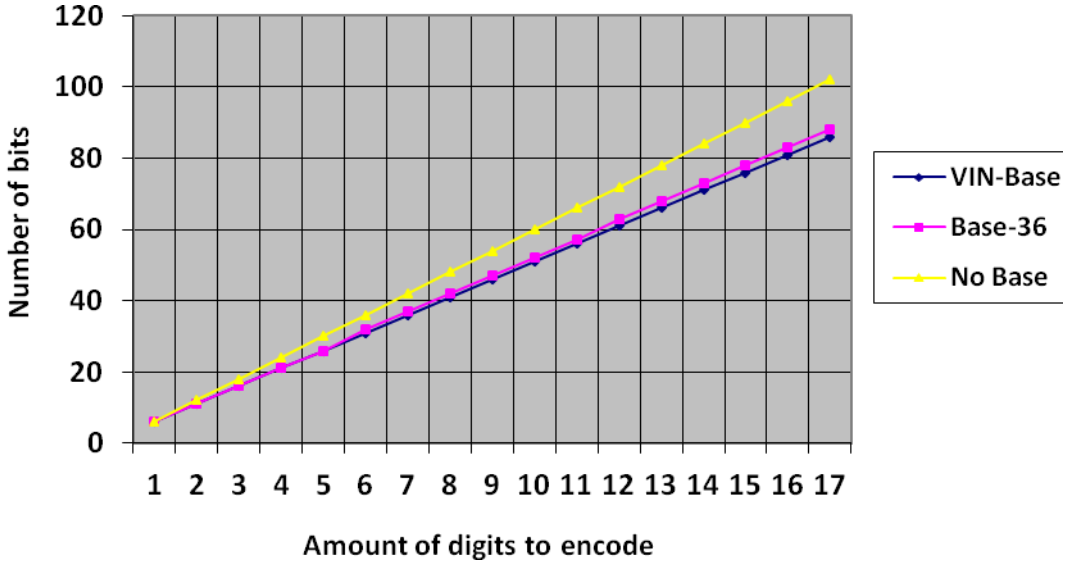


Figure 37 : Comparison between the amounts of bits necessary to encode a value depending on the representation

In the previous calculations, we do not consider the structure of the VIN code to deem the possible gains from the restricted set of allowed values. The below figure depicts the detailed sections of the VIN. With respect to the definitions in section 3.5.1.1, section A is the WMI, section B is VDS (except the check digit), section C is the check digit, section D is the year model, section E is alphanumeric part of the VIS, and section F is the numeric part of the VIS.

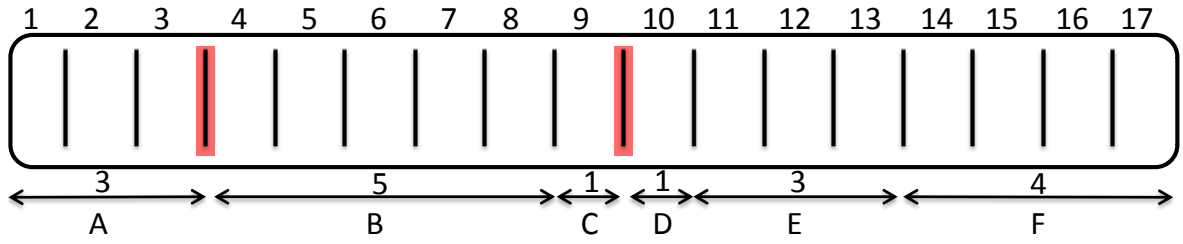


Figure 38: Detailed structure of VIN code

The below table summarizes the real costs of the conversion in terms of the resulting number of bits, taking into account the restricted set of allowed values for the different VIN sections.

Section	Number of possibilities	Amount of bits with Base-VIN representation
A	33^3	16
B	33^5	26
C	11	4
D	30	5
E	33^3	16
F	10^4	14

Table 4: Amount of bits necessary to encode a value of a VIN section in binary

Using this detailed description of the VIN, the total amount of bits necessary to encode the whole structured information drops from 86 (in the first table) to 81 bits only (sum of the last column values) if the sections are separated, and drops to 78 bits if the VIN is considered a whole (logarithm of base 2 of the product of the values in column 2).

3.3.3 Setting an IPv6 address with the new mapping method

As described in the assumptions section, in order to set an IPv6 address from a VIN code, we chose to convert two sections: WMI and VIS. Previous section detailed the structure of the VIN and described an efficient mapping method to obtain the maximum number of digits encoded in a minimum number of bits. With respect to the previous statements, sections A, D, E, and F from figure 4.2.2.2 are used in the conversion process.

Sections are converted separately and, as detailed in table 4.2.2.2, the total amount of used bits would be 51. Section A, which is the WMI, is converted to obtain 16 bits (Log2(333)); section D, which the year model, is converted to obtain 5 bits (Log2(30)); section E, which is the first alphanumeric part of the VIS, is converted to obtain 16 bits (Log2(333)); section F, which is the second numeric part of the VIS, is converted to obtain 14 bits(Log2(104)). The reverse conversion is also possible: from the converted values (binary) we can return to the initial coded sections (VIN numbers) thanks to the reverse operations, as explained above.

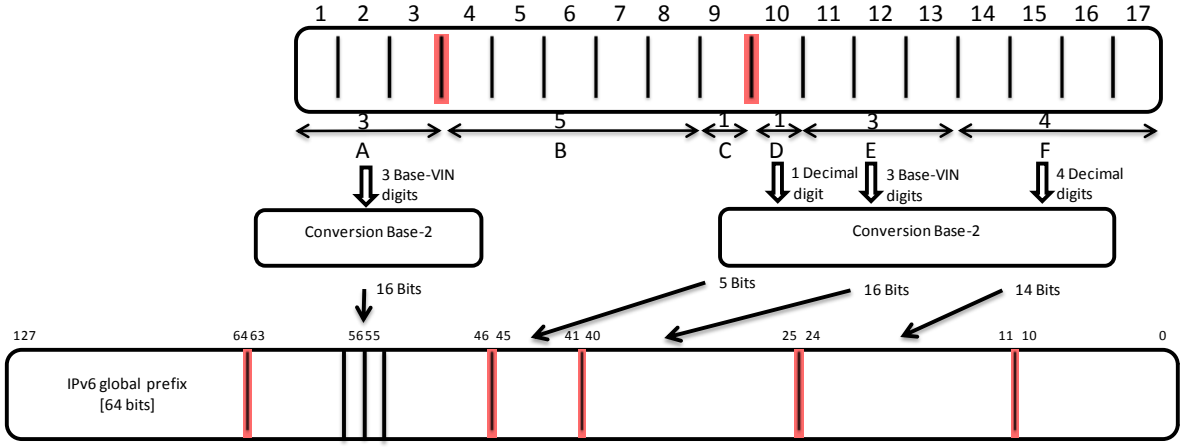


Figure 39: Final conversion method from the VIN number to an IPv6 address

The above figure describes the full conversion method starting from the VIN number to obtain an IPv6 address. The previously converted values are included into the 64bits of the Interface Identifier. With respect to RFC 4862, bits 55 and 56 that correspond to U/L bits are set accordingly. The final result is written on 53 bits, from position 63 to 11 (included) on the IPv6 address. 11 bits (10 to 0, included) are left blank.

We propose to use these 11 free bits to fix the possible IPv6 address collisions. A collision happens if at least two interfaces try to set the same IPv6 address in the same subnet. The 11 bits can then be used to fix the collision by pulling another adjacent address (2048 possibilities). With the information provided in ISO-3779, and following the proposed mapping/converting method, a collision is highly unlikely but theoretically possible; hence the proposal of the collision avoidance using the 11 bits is necessary.

3.3.4 Setting an IPv6 prefix with the new mapping method

The IPv6 addresses are valid on a certain scope: local, site or global. The new definition of the site-scoped IPv6 addresses (after the depreciation of the previous format in RFC 3879) is given in RFC 4193: Unique Local IPv6 Unicast Addresses (ULA). These prefixes can be generated with a pseudo-random algorithm, and the obtained prefixes are considered as global and used in the same way inside the site.

3.3.4.1 Unique local IPv6 addresses

RFC 4193 defines how to generate local IPv6 unicast addresses that are globally unique and intended for local communications. These addresses are meant to be used inside the scope of a limited area such as a site (limited by a border router). These addresses are routed in the same way as the globally scoped IPv6 addresses, but are not expected to be routable in the global Internet. Routing between a limited set of sites is also possible.

These characteristics and limitations are due to the pseudo-random generation method of these addresses. Depending on the generation method of the prefix, a collision is highly unlikely, but theoretically possible (with a very low probability). The main advantage of ULA prefixes is the generation method which is completely local the final site: in the absence of a link to an Internet Service Provider (ISP), responsible for the global prefix delegation (being the only authority that can guarantee the global uniqueness of a prefix) site communications cannot be made. Such communications may be considered as capillary to capillary communications over IPv6. For instance, if a ULA prefix is generated at the M2M Gateway level and advertised inside the capillary network, it allows such capillary to capillary communications (site to site).

The ULA prefix bitmap is represented in the below figure extracted from RFC 4193. The sections are as follows:

- Prefix: Fixed to a value of FC00::/7. It identifies the Local IPv6 Unicast Addresses.
- L: Set to 1 if the prefix is locally assigned. The value 0 is not yet defined.
- Global ID: 40-bit global identifier used to create a globally unique prefix.
- Subnet ID: 16-bit subnet identifier of a subnet within the site.
- Interface ID: 64-bit Interface ID as defined in RFC 4291.

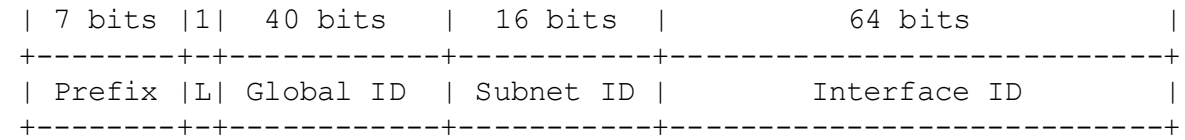


Figure 40: Local IPv6 Unicast Address format

3.3.4.2 First method for creating ULA prefix from VIN numbers

According to the described algorithm in section 3.2.2 of RFC 4193: “a suitably unique identifier, local to the node, should be used (e.g. system serial number)” to generate an IPv6 ULA prefix. In particular, this step of the algorithm is concerned with the creation of the Global ID part of the ULA; that is the pseudo random part of the prefix. We propose to use

the VIN as the unique system identifier in this step. Here is the aforementioned algorithm (RFC 4193) with the above update:

1. Obtain the current time of the day in 64-bit NTP format
2. Use the VIN number as a unique system serial number, and convert it to obtain a binary value using the previous mapping proposal.
3. Concatenate both information in order to create a key
4. Compute a SHA-1 digest on the obtained key. Resulting value is 160-bit long.
5. Use the least significant 40 bits as the Global ID.
6. Concatenate FC00::/7, the L bit set to 1, and the 40-bit Global ID to create a Local IPv6 Address prefix.

The obtained ULA prefix is then advertised inside the capillary network (ingress interface of the M2M gateway) to be used in stateless address auto-configuration process.

3.3.4.3 Second method for creating ULA prefix from VIN numbers

Another possible method to generate an IPv6 ULA prefix make use of the previous mapping and conversion method to fit a certain number of VIN extracted information into 56 bits (General ID and Subnet ID parts). It is a matter of future work to choose which sections are more pertinent to this operation.

One limitation of RFC 4193 to this proposal is the obligation that the General ID part must be pseudo-randomly generated. Even if the VIN code provides enough guarantees with respect to uniqueness, the generation following this method must be done in a separate Internet Draft proposal due to this constraint. The draft proposal is also a matter of future work.

3.3.5 Other possible use of the new mapping method

The mapping and conversion proposal in section 4.2.2 can be used to generate compact Mobile Node Identifiers (MNIDs) as defined in RFC 4283 (Mobile Node Identifier Option for Mobile IPv6 (MIPv6)). This RFC proposes a standard identification method at the Home Agent level of protocol Mobile IPv6, and its extension Proxy Mobile IPv6.

A new value for the “subtype” field of this option (below figure) can be defined within IANA to indicate that the payload (value of the option) designates a VIN code. The mapping method helps compressing the total VIN information into 88 bits (multiple of 8), which is very convenient for option alignment matters.

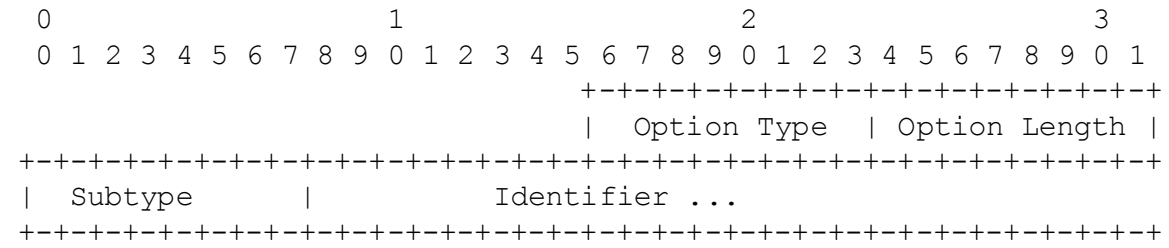


Figure 41: Mobile Node Identifier option format

The use of this option is possible, but not limited, to these services:

- authentication and authorization using an existing AAA (Authentication, Authorization, and Accounting) infrastructure or via an HLR/AuC (Home Location Register/Authentication Center)
- dynamic allocation of a mobility anchor point
- dynamic allocation of a home address.

3.4 Address Translation Mechanism for ZigBee and IP

Certain capillary networks use ZigBee and IP for addressing and forwarding packets between nodes within the capillary network and outside it. Since this section describes all the address translation mechanisms used throughout the project it makes sense to describe the ZigBee/IP translation scheme as well. Parts of textual description in this section may be found in other project deliverables.

As introduced on section 2.3.2, non-IP end nodes from capillary networks considered in EXALTED need to be addressed somehow in order to seamlessly identify them from the application server placed at the IP core network.

For that reason, an address translation mechanism is proposed and described in this section. The general objectives of this protocol, which are explained in detail in next subsections, can be summarized as follows:

- Be able to handle as much end nodes as possible with the same gateway, to assure large amount devices requirement.
- Introduce the lower overhead possible.
- Maintain the connection active since the node is initialized until it is de-registered.

For that purpose, the following architecture is proposed.

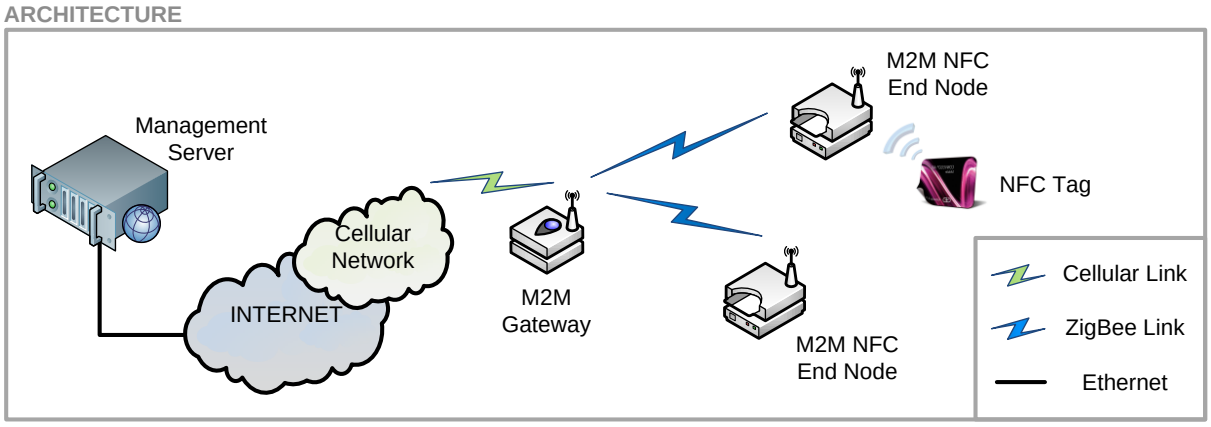


Figure 42: Address translation architecture

The figure above shows the address translation architecture envisaged to be implemented on real hardware and tested in terms of performance. It includes:

- Two end nodes behind the gateway, communicating using DigiMesh protocol. They implement NFC interfaces to let users interact with them. When approaching the INIT tag, the module is registered on the application server and the communication initiated. They can receive as well commands in order to act over LEDs or general purpose I/Os.
- A gateway in charge of implementing the address translation mechanism. Once a frame is received on the 802.15.4 interface it parses it, checks the sender's address and encapsulate the info on an IP datagram. In case it is the first message from a specific node, a new session is created to handle the communication.
- Finally a remote application server able to handle the devices. It will gather the information sent by end devices and will be able to send commands back.

All these features, as well as other issues related to the algorithm deployment are described on the following sections.

3.4.1 General aspects about the algorithm

There are two main types of communication regarding the address translation mechanisms: one related to device initialization and the other for the normal operation.

It is important to note that the way it is conceived, integrating NFC capabilities, pursues two big EXALTED goals:

- On the one hand, it proves heterogeneity on the technologies handled by the network.
- On the other side, it gives visibility to the operations performed. Both initialization and sending of the data could be programmed on nodes, but this way it can be shown on demand. This way anyone can test the correct behavior of the algorithm at a certain moment.

Registration is the first task needed to be performed on the devices. If it is not done, no other operations are enabled by the system.

Once the INIT tag is approached to one of the devices, it sends a 802.15.4 message to the gateway containing the ID of the tag. This info is translated into an IP datagram and transmitted to the application server through a new socket session started for this purpose (further considerations about the behavior of the gateway are treated on section 3.4.3). The application server then registers the node and sends back the confirmation, firstly to the gateway and, then, to the node, as shown in the figure below.

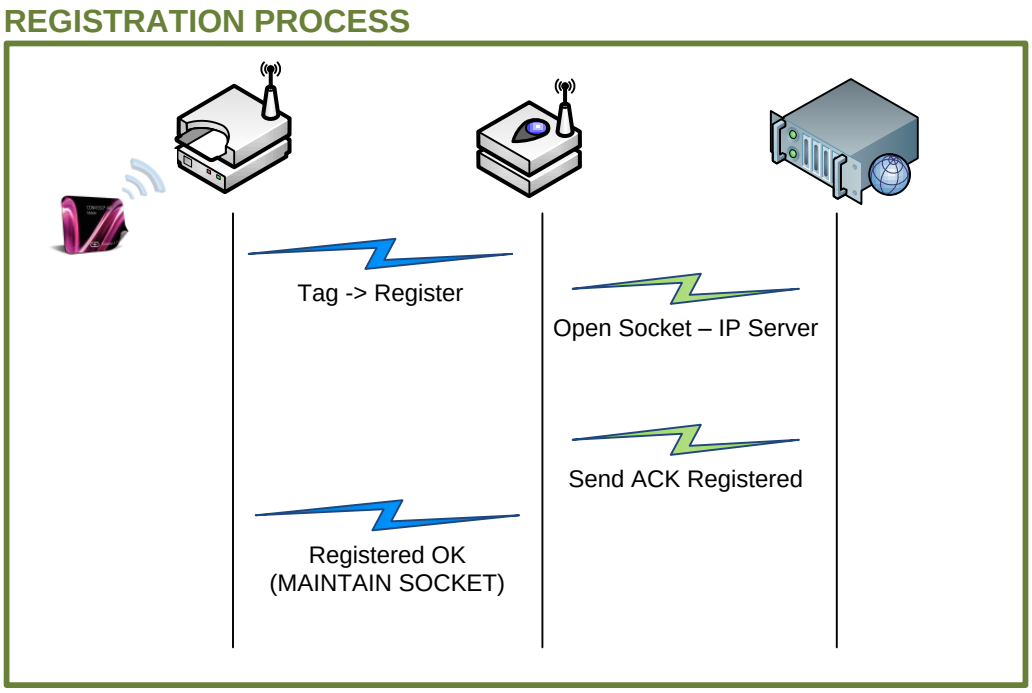


Figure 43: Registration process

Once the node is registered, any tag can be approached to the nodes, resulting on a different response from the server. The info is sent via 802.15.4 air interface to the gateway which capsules it into IP and sends it towards the application server using the previously created socket connection. The same way, the response is routed back to the node, as can be seen on figure below:

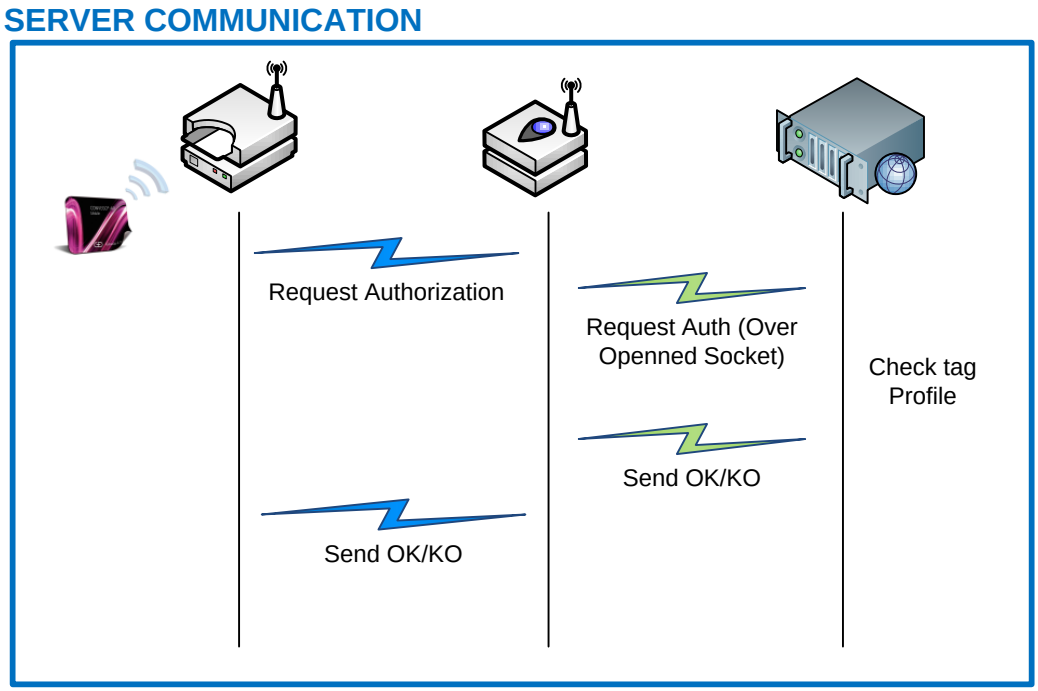


Figure 44: Server communication

The payload data included on DigiMesh fames, depending whether it comes from the nodes (TX) or from the application server (RX) is pictured on figure below.

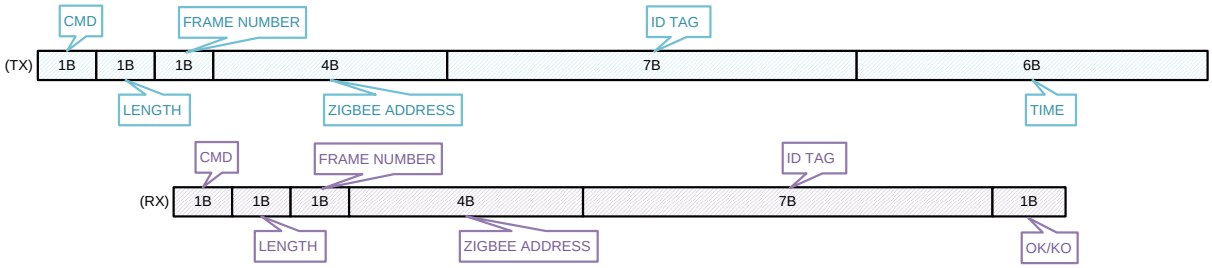


Figure 45: Payload structure

With this architecture in mind, the only thing left to define is the specific behaviour programmed on each element of the architecture.

3.4.2 Tasks performed by End Devices

End nodes are in charge of identifying tags and send data to the gateway. For the first time, it waits until the INIT tag is approached, and then sends the info to the gateway.

From that point, any info related to any tag approached is sent to the application server. This behaviour is shown in figure below.

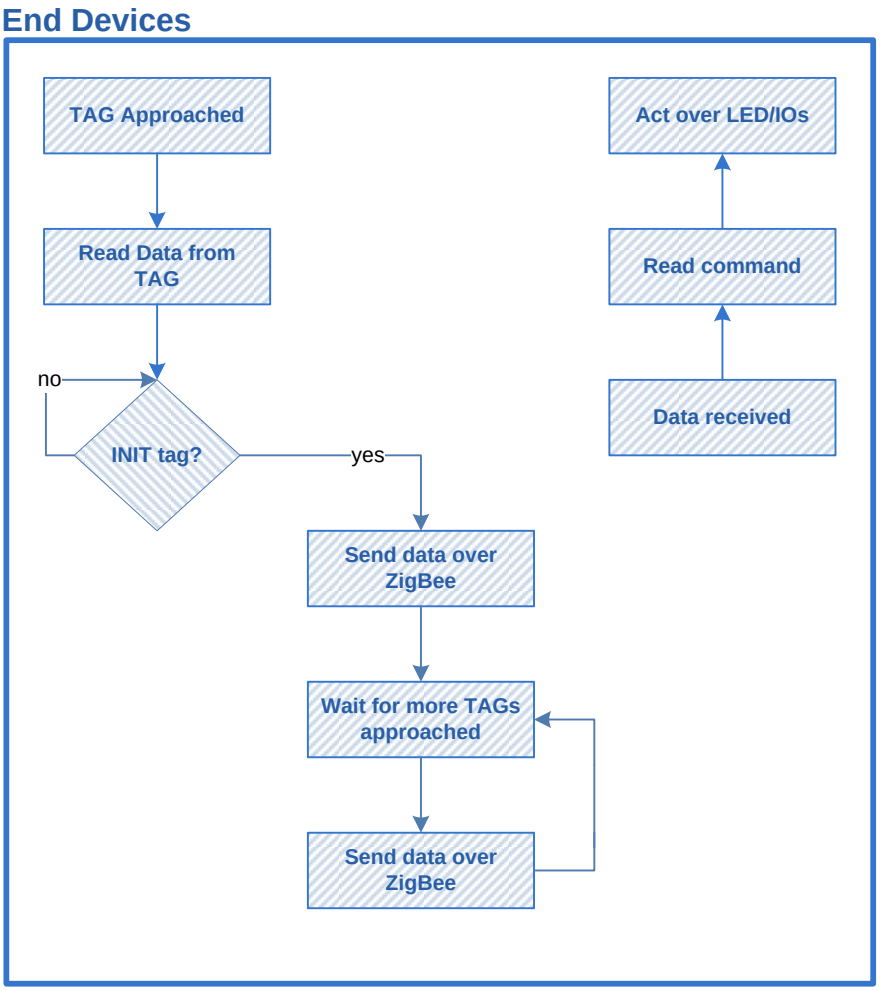


Figure 46: Tasks performed by end devices

3.4.3 Tasks performed by the Gateway

The Gateway implements the core of the address translation mechanism. It is responsible to handle the devices behind it on the capillary network.

The mapping between 802.15.4 addresses and IP addresses is made by the set up of a number of sockets equals to the number of nodes managed. Theoretically, that means that one single gateway is able to handle up to 65535 devices, which is the maximum port number (each socket connects to a different port number, so the maximum number of sockets is limited by the maximum ports enabled). But, in the real life, cellular modems have limitations in terms of maximum number of TCP connections.

The module selected for this address translation proof of concept is Sagem's HiLo [98]. Using this module, the maximum number of opened sessions is limited to 200, so this is our real maximum. Anyway, by selecting less restrictive cellular modems it is possible to enhance this capability.

Given this important information, the main tasks performed by the gateway can be summarized as follows:

- Every time a 802.15.4 packet is received from the capillary networks, it parses it, checks the sender address and:

- In case it is a new node, a new session (socket connection) is initialized, using a unique port. In addition, a new entry on the local database is added for the node mapping address of the node with the new session identifier.
- If it is an existing node, it looks up in the database which session is associated to the node, and, then routes the packet (capsule as IP datagram) through this connection.

- In case it receives a command from the application server, the proper receiver is identified by looking for the session identifier related to the socket from which the data is received. Once identified, the packet is parsed and sent using DigMesh format.

Previously mentioned steps are depicted on the flow chart from figure below:

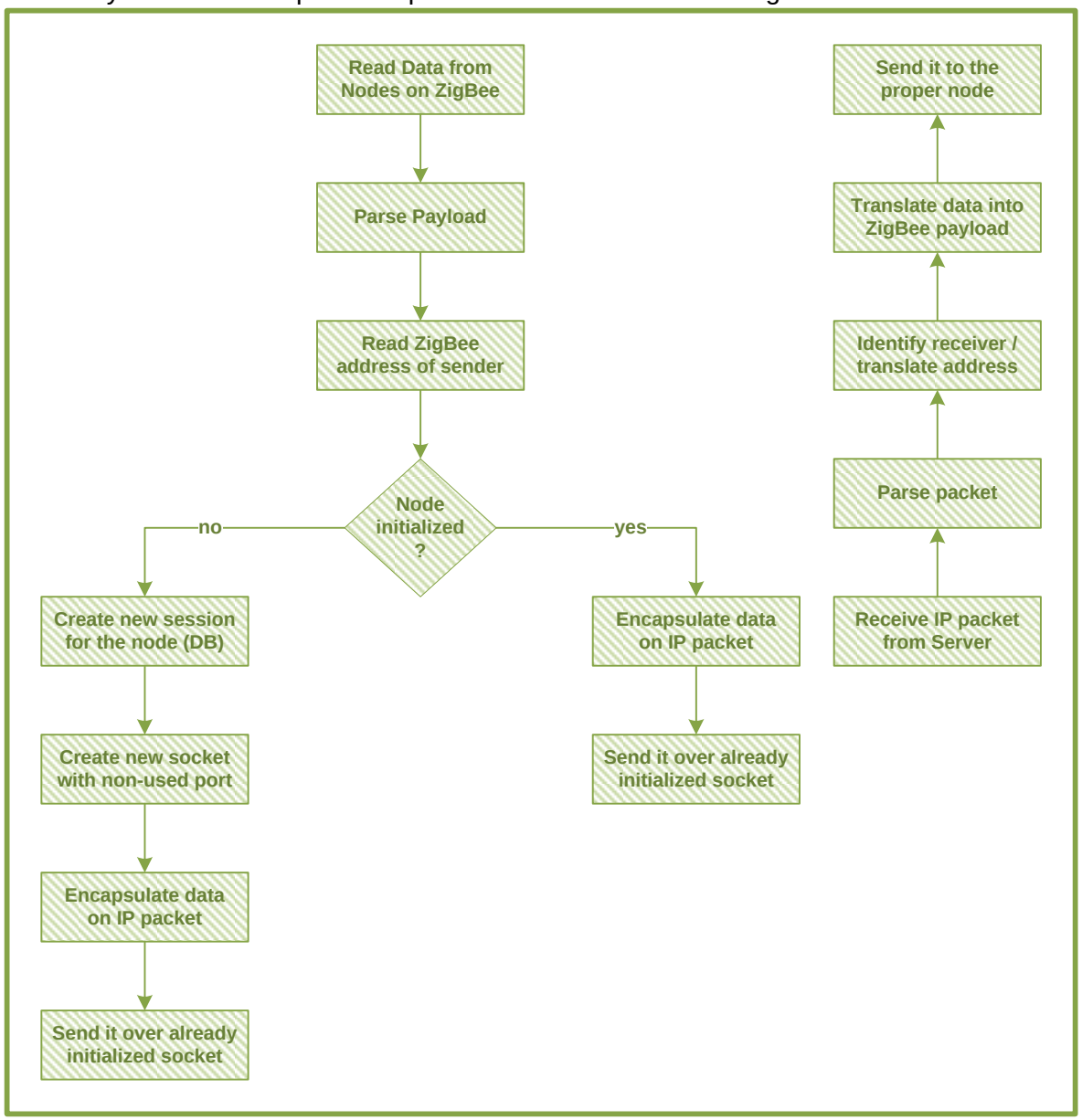


Figure 47: Tasks performed by the Gateway

3.4.4 Tasks performed by the application server

The application server is a piece of hardware placed remotely somewhere on the core IP networks. For the correct behavior of the algorithm, it is enough to know the current IP address of the server. Each node has it programmed in advance, and, at the time it is on the initialization process, the connection is set up using that address.

The functionality of the server is based, then, on a socket server listening on all ports. Every time a petition is received from one node, the server reads it and place the response using the same port as the received connection. This way, it enables the gateway to clearly identify the node to which the response is routed.

As well as in previous cases, once the data are parsed, it is important to check whether it is an initialization petition or just a normal operation of a tag being approached to the node.

In case it is the first option, a new entry must be created on the database of supported nodes, associating the new entry with rights for tags or whatever is needed, depending on the application. For instance, if it is an access control use case, the node must be associated with a list of tags with granted access. If the application is about logistics, the initialization must trigger the creation of a set of tables controlling when and which type of tags are approaching to the node.

In all cases, the server must respond with the confirmation of the process being completed successfully.

Finally, if it is just a petition from a regular tag, the server checks into the corresponding database tables associated to the node from which the connection is made in order to response accordingly. Following the same previous examples, in access control the server must respond with an OK for tag accepted or KO for permission denied. In the logistic case, an ACK message may be enough.

As in previous sections, a visual example of this flow is represented in the figure below.

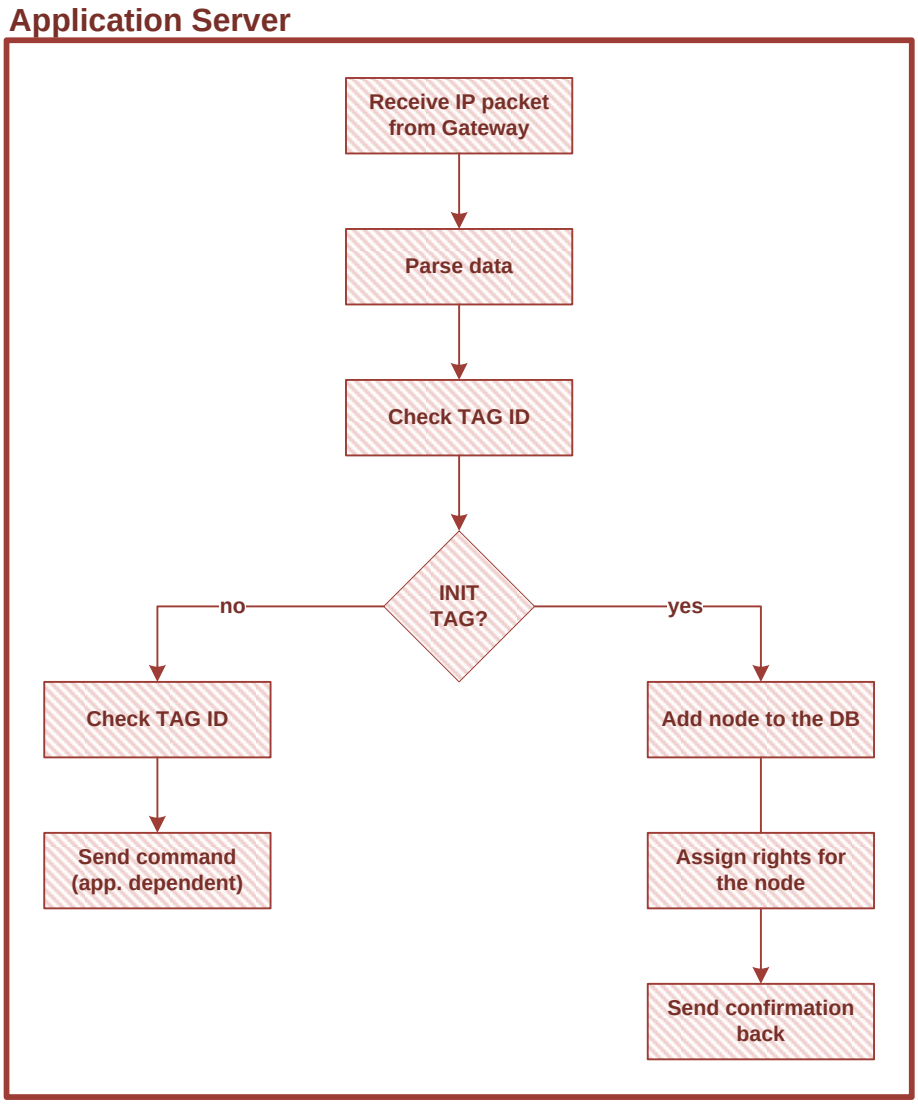


Figure 48: Tasks performed by the Application Server

3.4.5 Overhead introduced

The final step for clearly understand and fully define this study about address translation mechanism is looking into the disadvantages of the method, translated into the unavoidable overhead introduced by encapsulating the data in IP datagrams.

This study considers 20 Bytes of data info transmitted by nodes into the 802.15.4 payload. In case IPv4 is used, the resulting overhead derived from the operation is summarized on Table 5.

Table 5. Overhead due to IPv4 connections.

	OUT (B)	IN (B)
Handsake	96	48
Outbound Data	68	
ACK		48
Close	48	
Final ACK		48

Initialize/Close	96	96
Data per Connection	68	48

Efficiency (%)	17,2

Three way handshake is considered for initializing the connection, and close/ack for terminating it, just once for each node. The overhead introduced by IPv4 header is considered to be 48 bytes. With all these constraints, the resultant efficiency is slightly over 17%.

For IPv6 case, the same parameters are calculated on Table 6. On this case, the overhead introduced by the IPv6 header is 68 bytes.

Table 6. Overhead due to IPv6 connections.

	OUT (B)	IN (B)
Handsake	136	68
Outbound Data	88	
ACK		68
Close	68	
Final ACK		68

Initialize/Close	136	136
Data per Connection	88	68

Efficiency (%)	12,8

Considering a network made up of 100 nodes, each one transmitting data with 30 seconds periodicity, it is possible to calculate the amount of data transferred on the network for one day period. This is what shows the figure below, painting in red the data transferred using IPv6, in blue the same thing regarding IPv4 and in green the real data sent on both cases.

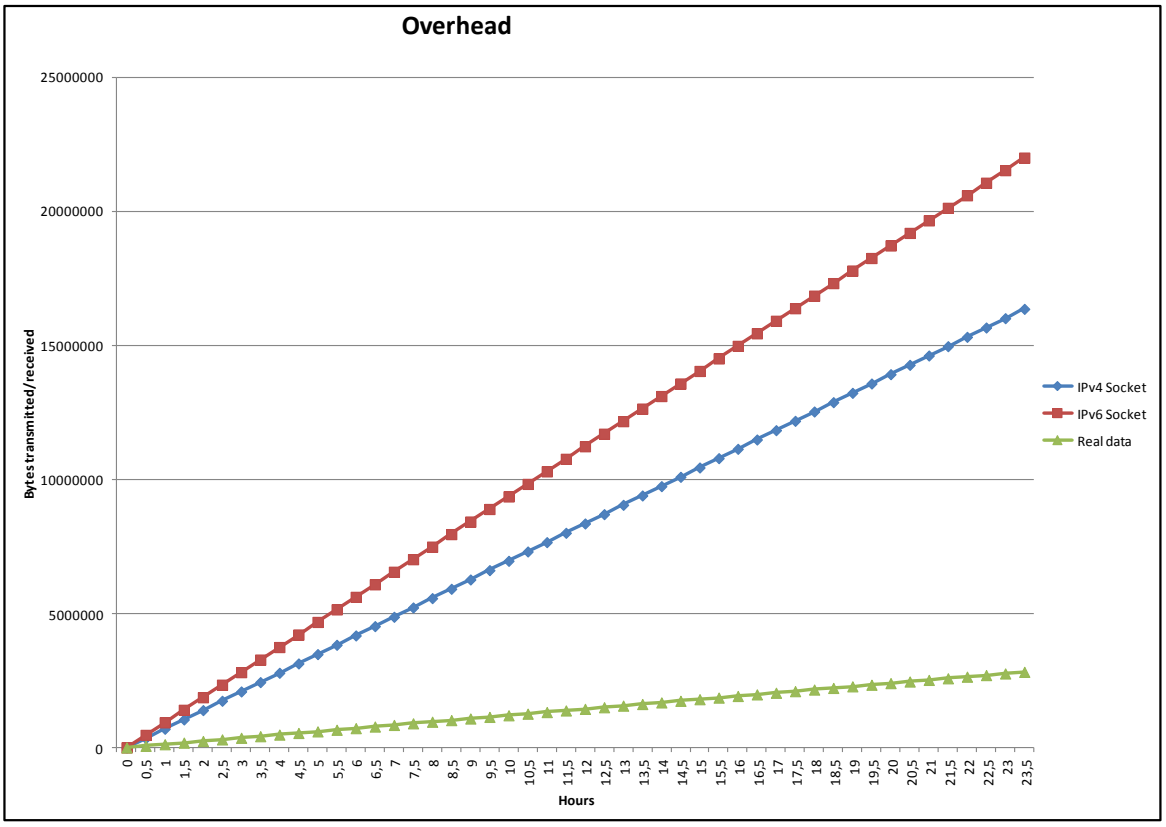


Figure 49: Overhead introduced by the mechanism

Finally, considering variable length in data payloads, it is possible to calculate the resultant efficiency of both IP stacks. In the figure below, this is presented varying the amount of data sent by sensors from 5 to 50 bytes.

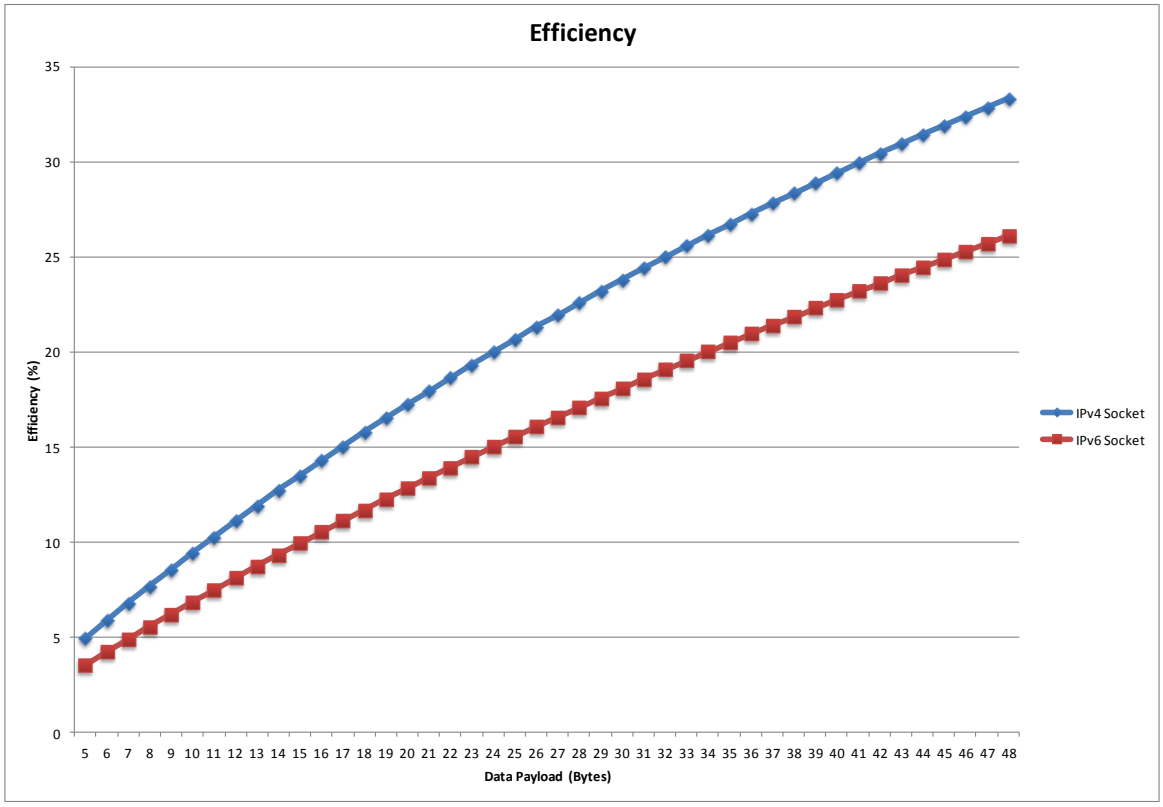


Figure 50: Efficiency of the mechanism

3.5 Address Translation Mechanism for Capillary (6LoWPAN) and IP

One of the most important tasks of the EXALTED project is design of establishment continuous connectivity among nodes, outside and inside the different capillary networks and the LTE-M network. All devices in EXALTED must be addressed and be reachable for the purpose of end-to-end communication and unique identification from the M2M application server point of view. In order to provide this, an address translation mechanism is proposed end elaborated for 6LowPAN capillary network address translation to IPv6 addresses provided by the LTE-M network. In section 2.3.3 general principles of 6LoWPAN and IPv6 address translations and routing in IP networks are presented, while in this section these address translations are considered and discussed for the LTE-M networks.

As explained in section 2.3.3 an LTE-M gateway is responsible for address translation between LTE-M and 6LoWPAN capillary networks. The implemented translation procedure is supposed to provide a possibility to handle address translation and connection with the rest of the network for as much as possible M2M end-devices (nodes) within the same gateway.

In order to perform appropriate 6LowPAN – IP address translation, the LTE-M gateway is supposed to perform the following functionalities:

- during the registration procedure to the LTE-M network, the LTE-M gateway will obtain an IP address from the LTE-M network;
- The LTE-M gateway performs maintaining of device mapping table, for the mapping of 16 bit short address for all devices in the 6LoWPAN capillary network to IP addresses recognizable by the LTE-M network (IP address + port);
- The LTE-M gateway maps 6LoWPAN addresses into ports in the IP interface of the LTE-M gateway node, to create socket for every device;
- when the LTE-M gateway received frame on the 802.15.4 interface it parses it, checks the sender's address and encapsulate the info on an IP datagram;

- for the case that new device sends data for the first time, the LTE-M gateway is supposed to create new socket connection, i.e. to map LTE-M gateway IP address to appropriate port, and to add and save this address into the address translation table;
- if device is already registered, i.e. previously mentioned initialization is done, then LTE-M gateway will perform table lookup for that node, and route the packet to appropriate socket;
- one single gateway theoretically is capable to handle up to 65535 devices (maximum number of sockets), which number highly depends on the gateway hardware and implementation.

End devices behind the LTE-M gateway, communicate using 6LowPAN protocol. End devices are performing adequate measurements and send these results towards the appropriate application servers, which will perform collection and analysis of the data. Also application server can ask end devices for their measurements. Application server is responsible for the collection and data processing obtained from the LTE-M devices through the LTE-M network. Each end device is supposed to be aware of the LTE-M gateway address towards which will send data.

Regarding the procedures of sending data from the LTE-M gateway towards the application server, there are the following options:

- One option is to continuously perform the maintenance of address translation mapping tables both on the LTE-M gateway and on the application server, which will require continuously provisioning of the data on the application server, i.e. maintenance of the table with devices sockets, i.e. when new sensor is added into the capillary network, the table on the application server should be updated.
- Other approach could be to use tags for identifications of the end devices, i.e. end device will send initiation tag towards application server, which will interpret tag, register device, and create communication with device. One possible way to initiate and force communications from devices is by using the NFC tags. In section 3.5.2.1 communication using the NFC tags is presented in details.

Proposed procedure for address translation mechanism for the capillary and the IP networks will obviously introduce some overhead, because of encapsulating data in the IP datagrams.

4. Conclusions

In this document we have tried to provide a fresh perspective in building an IP networking system for machine-to-machine communications in the Future Internet, with an application in the field of vehicular communications.

In the context of project EXALTED, an M2M IP networking system would be based on IPv6 protocols. In addition, for capillary-to-capillary-to-infrastructure communications (sometimes applied as Vehicle-to-Vehicle-to-Infrastructure communications), a protocol is proposed for address autoconfiguration and routing between these networks; since devices within capillary networks may use a different addressing scheme than IP, several address translation schemes are proposed for ZigBee and 6lowpan addressing schemes. Finally, for the particular use case of vehicular communications we proposed a mapping between the Vehicle Identification Number (VIN) and IPv6 addressing.

In order to better understand these concepts, we have first presented a state-of-the-art set of fundamental principles behind the workings of the Internet today, as well as the clean-slate and evolutionary approaches for the design of Future Internet – when applied in project EXALTED the evolutionary approach has been selected. For the application, we have identified the mobility protocols, 6lowpan and address translation mechanisms for capillary networks as vehicular moving networks.

We have presented a number of IP translation mechanisms (for ZigBee, for 6lowpan), a routing and address auto-configuration mechanism allowing for capillary-to-capillary-to-infrastructure (V2V2I) communications.

List of Acronyms

Acronym	Meaning
3GPP	3rd Generation Partnership Project
4WARD	Architecture and Design for the Future Internet
6LN	6LoWPAN Node
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
6LR	6LoWPAN Router
AAA	Authorization Authentication and Accounting (an IETF term)
ADR	Australian Design Rules
AKARI	Architecture Design Project for New Generation Network
AODV	Ad-hoc On-demand Distant Vector
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous System
BGP	Border Gateway Protocol
C2C	Car-to-Car
C2C NET	Car-to-Car Network
CCN	Content Centric Networking
CEA	Commissariat à l'énergie atomique et aux énergies alternatives (Atomic Energy Commission)
CGA	Cryptographically Generated Addresses
CHORD	Not an Acronym, name of a lookup service, see [48].
CN	Correspondent Node
CoA	Care-of Address
CPU	Central Processing Unit
CS	Content Store
DAG	Directed Acyclic Graph
DARPA	Defense Advanced Research Projects Agency
DFZ	Default Free Zone
DFZ FIB	Default Free Zone Forward Information Base
DFZ RIB	Default Free Zone Routing Information Base
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol for Internet Protocol Version 6
DHT-LISP	Distributed Hash Tables - Locator/ID Separation Protocol
DigiMesh	Not an Acronym, name of a Networking Protocol, see [33].
DNS	Domain Name Service
DNS SRV	Domain Name Service Service
DODAG	Destination Oriented DAG
DoS	Denial-of-Service (attack)
DYMO	Dynamic MANET On-demand
E2E	End-to-End
EID	Endpoint Identifier
ENCAPS	Encapsulation
ESD	End System Designator
ET	Expiration time
Ethernet	Not an Acronym, (origin: "ether" and "networking")
ETR	Egress Tunnel Router
ETX	Expected Transmission Count
EUI-64	64-bit Extended Unique Identifier
EYU	Ericsson Yugoslavia

FDDI	Fiber Distributed Data Interface
FeliCa	Felicity Card
FI	Future InterNet
FIB	Forwarding Information Base
FIND	Future InterNet Design
FMIPv6	Fast Handovers for Mobile IPv6
FP7	Framework Programme 7
FQDN	Fully Qualified Domain Name
GDA	Globally Deliverable Addresses
GENI	Global Environment for Network Innovation
GRA	Globally Routable Addresses
GSE	Global, Site, and End-system address elements
HA	Home Agent
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
HBA	Hash-Based Addresses
HiLo	Not an Acronym; name of a hardware device (GPRS M2M module) commercialized by company Sagem, as of 2012. Also name of a hardware programming technique.
HIP	Host Identity Protocol
HIT	Host Identity Tag
HLR/AuC	HLR: Home Location Register/AuC-unknown
HMIPv6	Hierarchical Mobile IPv6
I1	"Initiator" 1
IAB	Internet Architecture Board
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ID	Identifier
IETF	Internet Engineering Task Force
IID	Interface Identifier
IP	Internet Protocol
IP-EP	IP-Endpoint
IPNL	IP Next Layer
IPSec	Internet Protocol Security
IP-TR	IP-Transit
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
ITR	Ingress Tunnel Router
IV	Internet Vehicle
LCoA	Local CoA
LFN	Local Fixed Node
LFU	Least Frequently Used
LIN6	Location Independent Network Architectur for IPv6
LISP	Locator Identifier Separation Protocol
LLN	Low-power and Lossy Links
LMA	Local Mobility Anchor
LoWPAN	Low power Wireless Personal Area Networks
LRU	Least Recently Used
LSI	Local Scope Identifier
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
LTE-M	Long Term Evolution Network with M2M enhancements
LV	Leaf Vehicle

M2M	Machine to Machine
MAC	Media Access Control
MAG	Mobile Access Gateway
MANEMO	MANET NEMO
MANET	Mobile Ad-hoc Network
MAP	Mobile Anchor Point
MAST	Multiple Address Service for Transport
MIFARE	Not an Acronym, trademark of widely used chips for smart and proximity cards, owned by NXP Semiconductors.
MIP	Mobile IPv6
MIPv6	Mobile IPv6
MN	Mobile Node
MNID	Mobile Node Identifier
MNN	Mobile Networks Node
MNP	Mobile Network Prefix
MPR	Multipoint Relay
MR	Mobile Router
MTCP	Multipath TCP
NAT	Network Address Translation
ND	Neighbor Discovery
NDP	Neighbor Discovery Protocol
NEMO	Network Mobility
NFC	Near Field Communications
NHTSA	National Highway Traffic Safety Administration
NIRA	New Inter-Domain Routing Architecture
NRLS	Name-to-Route Lookup Service
NSF	National Science Foundation
NXP Semiconductors	Next Experience Semiconductors (brand)
OLSR	Optimized Link-State Routing
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PA	Provider Address
PAN	Personal Area Netowrk
PDN	Packet Data Network
PHA	Proxy Home Agent
PHY	Physical
PIT	Pending Interest Table
PMIP	Proxy Mobile IPv6
PMIPv6	Proxy Mobile IPv6
QoS	Quality of Service
RA	Router Advertisement
RCoA	Regional CoA
REAP	REAchability Protocol
RFID	Radio Frequecy Identification
RG	Routing Goop
RIB	Routing Information Base
RIP	Routing Information Protocol
RLOC	Routing LOCator
ROFL	Routing On Flat Labels
RoLL	Routing over Low power and Lossy
RPL	IPv6 Routing Protocol for Low power and Lossy Networks
RS	Router Solicitation
RS/RA	Router Solicitation / Router Advertisement

RVS	Rendez-Vous Server
RX	Reception
SAE	Society of Automotive Engineers
SCTP	Stream Control Transmission Protocol
SEIS	SECURITY IN EMBEDDED IP-BASED SYSTEMS
SEND	SEcure Neighbor Discovery Protocol
SHA-1	Secure Hash Algorithm, version 1.
SHIM6	Site Multihoming by IPv6 Intermediation
SIP	Session Initiation Protocol
SLAAC	StateLess Address AutoConfiguration
SMTP	Simple Mail Transfer Protocol
STP	Site Topology Partition
TCP	Transport Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
TIPP	Topology Information Propagation Protocol
TKS	Telekom Serbia
TRIAD	Translating Relaying Internet Architecture integrating Active Directories
TST	Tecnologías, Servicios Telemáticos y Sistemas S.A.
TX	Transmission
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
ULA	IPv6 Unique Local Addresses
ULID	Upper-Layer IDentifier
URN	Uniform Ressource Name
V2V	Vehicle-to-Vehicle
V2V2I	Vehicle-to-Vehicle-to-Infrastructure
VDS	Vehicle Description Section
VIS	Vehicle Identification Section
WMI	World Manufacturer Identifier
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
X.25	A packet-oriented communication protocol; “X” is first in a series – other protocols in same domain use “V”, e.g. V.24.
X-MAC	Short-preamble MAC protocol
XML RPC	eXtended Markup Language Remote Procedure Call
XMPP	Extensible Messaging and Presence Protocol
xTR	Ingress/Egress Tunnel Router
ZigBee	Bee flying in a zig zag manner, from flower to flower, hop to hop.

VIN-Base Numeral System Specification

1- Definitions

The VIN-Base is the numeral system where all VIN codes belong. A number in VIN-Base is convertible in other numeral systems (decimal, binary, hexadecimal, etc) by simple multiplication operations and vice-versa, with simple division operations.

The VIN-Base numeral system contains 33 different ordered digits. Their value in the decimal system goes from 0 to 32, and the symbols used are those defined in the VIN description: One of the letters in the set [ABCDEFGHJKLMNPRSTUVWXYZ] or a numeral in the set [0123456789].

The ordered set of VIN-Base numerals is defined as follow:

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VIN-Base	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G
Decimal	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
VIN-Base	H	J	K	L	M	N	P	R	S	T	U	V	W	X	Y	Z	

2- Conversion from decimal to VIN-Base

A decimal number can be written in the VIN-Base following simple rules using simple division operations. The algorithm is similar to conversion from decimal to other numeral systems (binary for example) and it is defined as an extension of it. Here are the necessary steps:
Suppose we have to convert X from decimal system to VIN system. The resulting number is Y
1- Divide X by 33 while the quotient is greater than 33
2- Write the remainder in the VIN-Base numeral system
3- Stop division when the quotient is less than 33 and write it in VIN-Base
4- Read Y backwards from last quotient to first remainder, putting the last quotient in the most significant bit (MSB) and the first remainder in the least significant bit (LSB). Y is now in the VIN-Base numeral system

Example
Let us convert 1200 from decimal system to VIN system.

1) 1200/33 = 36 + 12[33] ---> 12 is the first remainder. It is C in VIN-Base
2) 36 is greater than 33, so we shall continue
3) 36/33 = 1 + 3[33] ---> 3 is the second remainder. It is 3 in VIN-Base
4) 1 is less than 33, so this is the end of our algorithm. 1 is 1 in VIN-Base
5) The result of the conversion is '13C'

3- Conversion from VIN-Base to decimal

A VIN-Base number can be written in the decimal sytem following simple rules using simple multiplication and power operations. The algorithm is similar to conversion from other numeral systems to decimal (binary for example) and it defined as an extension of it. Here are the necessary steps:
Suppose we have to convert 'X' from VIN-Base numeral system to decimal. The resulting number is 'Y'. Let 'n' be the number of positions in X and X(i) the VIN-Base digit of X in ieth position starting from 1, reading the number from the right (LSB) and converted to decimal (from the table above). Then:

```
Y = 0
for (i = n-1; i == 0; i--) {
      Y+=X(i+1)*(33^i);
}
```

Example:
Let us reconvert the result from the previous example, from VIN-Base to decimal.
X = 13C, n = 3, LSB = C and MSB = 1.
Y = 1*(33^2) + 3*(33^1) + C*(33^0)
= 1*(1089) + 3* (33) + 12*(1)
= 1200

4- Number of positions

It is possible to know how many positions are necessary to write a number in the VIN-Base numeral system before doing the conversion from decimal. It is necessary for that to have the immediate power of 33 that is greater than or equal to the decimal number to convert. The number of necessary positions is his power of 33. For example, suppose that you have to convert 1200 from decimal numeral system to VIN-Base. 1200 is immediately less than 33^3 and greater than 33^2. So 1200 will be written in 3 VIN-Base positions after conversion (We saw above that 1200 is actually '13C' in VIN-Base).

References

[1] 4WARD, The FP7 4WARD project, Online on 20/03/2012 at: http://www.4ward-project.eu/index.php?s=overview.

[2] A. Feldmann. Internet Clean-Slate Design : What and Why ? ACM SIGCOMM, 2007.

[3] A. Garcia-Martinez, M. Bagnulo, and I. Beijnum. The Shim6 Architecture for IPv6 Multihoming.IEEE Communications Magazine, 2010.

[4] A. Gurtov, M. Komu, and R. Moskowitz. Host Identity Protocol : Identifier/Locator Split for Host Mobility and Multihoming. The Internet Protocol Journal, March 2009.

[5] A. Odlyzko. 'Smart' and 'Stupid' Networks : why the Internet is like Microsoft. Mixed Media, 1997.

[6] A. Petrescu, C. Janneteau, N. Demailly and S. Imadali, "Router Advertisements for Routing between Moving Networks," draft-petrescu-autoconf-ra-based-routing-02, 2012.

[7] AKARI, Architecture Design Project for New Generation Network, Online on 20/03/2012 at : http://akari-project.nict.go.jp/eng/index2.htm

[8] B. Carpenter. RFC1958 - Architectural Principles of the Internet. IETF, 1996.

[9] B. Haberman and J. Martin, "Automatic Prefix Delegation Protocol for Internet Protocol Version 6 (IPv6)," draft-haberman-ipngwg-auto-prefix-02, 2002.

[10] B.-W. Kim, "Method for setting an internet protocol address using a vehicle identification number," Samsung Electronics Co., Ltd US PATENT 7917603, http://patents.com/us-7917603.html, Nov. 29, 2004.

[11] Base-36. Wikipedia - the free encyclopedia. [Online]. http://en.wikipedia.org/wiki/Base36

[12] C. Perkins, E. Belding-Royer, S. Das, "RFC3561:Ad hoc On-Demand Distance Vector (AODV) Routing", July 2003

[13] C. Perkins, Ed., D. Johnson, J. Arkko, "RFC6275: Mobility Support in IPv6" , July 2011

[14] C. William, D. Huo "Mobility Considerations for 6LoWPAN" - draft-williams-6lowpan-mob-02.txt, March 8, 2010

[15] Code of Federal Regulations. (1996, Jun.) PART 565--VEHICLE IDENTIFICATION NUMBER REQUIREMENTS. [Online]. http://www.nhtsa.gov/DOT/NHTSA/Rulemaking/Rules/Associated%20Files/VIN_Final_Rule_April_08.pdf

[16] Craig Partridge. A conversation with van jacobson, ACM. Making the case for contentcentric networking.

[17] D. Cocker. Multiple Address Service For Transport (MAST) : An Extended Proposal (Internet Draft). IETF, 2003.

[18] D. Cocker. Multiple Address Service for Transport (MAST). Applications and the Internet, IEEE/IPSJ International Symposium on, 2004.

[19] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. Locator/ID Separation Protocol (LISP) (Internet Draft). IETF, 2009.

[20] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, 2004.

[21] D. Le, X. Fu, and D. Hogrefe. A Review of Mobility Support Paradigms for the Internet. IEEE Communications Surveys, 2006.

[22] D. Massey, L. Wang, B. Zhang, and L. Zhang. A Scalable Routing System Design for Future Internet. In IPv6'07. ACM, August 2007.

[23] D. Meyer, L. Zhang, and K. Fall. RFC4948 - Report from the IAB Workshop on Routing and Addressing. IETF, 2007.

[24] D. Meyer. The Locator Identifier Separation Protocol (LISP). The Internet Protocol Journal, March 2008.

[25] D. R. Cheriton and M. Gritter. TRIAD : A Scalable Deployable NAT-based Internet Architecture. Stanford Computer Science Technical Report, 2000.
[26] D. Smetters and V. Jacobson. Securing Network Content. 2009.
[27] D.D. Clark, K.R. Sollins, J. Wroclawski, and R. Braden. Tussle in Cyberspace : defining Tomorrow's Internet. In ACM SIGCOMM'02. ACM, August 2002
[28] D.D. Clark, K.R. Sollins, J. Wroclawski, and T. Faber. Addressing Reality : An Architectural Response to Real-World Demands on the Evolving Internet. In ACM SIGCOMM'03. ACM, August 2003.
[29] D.D. Clark. RFC814 - Name, Addresses, Ports, and Routes. IETF, 1982.
[30] D.D. Clark. The Design Philosophy of the DARPA Internet Protocols. In ACM SIGCOMM. ACM, August 1988.
[31] D.S. Isenberg. The Rise of the Stupid Network : Why the Intelligent Network Was a Good Idea Once But Isn't Anymore. Essay, 1997.
[32] Department of Information Teraoka-lab, Faculty of Science Computer Science, and KEIO Univ. Technology. Lin6. Online on 15/03/2012 at: http://www.tera.ics.keio.ac.jp/?page_id=653, 2010.
[33] Digi International Inc. "XBee/XBee-PRO®DigiMesh 2.4 RF Modules" January 2012
[34] E. Nordmark and M. Bagnulo. RFC5533 - Shim6 : Level 3 Multihoming Shim Protocol for IPv6. IETF, 2009.
[35] Ford's VIN decoder. VIN Guide for 2011 -- Your Guide to 2011 Vehicle Identification. [Online]. https://www.fleet.ford.com/maintenance/vin_tools/pdfs/VIN2011.pdf
[36] FP7 EXALTED, "D4.1 – M2M Packet Data Protocols between LTE-M and Capillary Networks", project report, July 2012.
[37] G. Bag, M. Taqi Raza, Ki-Hyung Kim, Seung-Wha Yoo " LoWMob: Intra-PAN Mobility Support Schemes for 6LoWPAN", June 23, 2009
[38] G. Huston. Growth of the BGP Table - 1994 to Present (http ://bgp.potaroo.net/), 2012.
[39] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks",Internet-draft: RFC 4944, Sep. 2007.
[40] G. Van de Velde, T. Hain, R. Droms, B. Carpenter, and E. Klein. IPv6 Network Architecture Protection. IETF, 2005.
[41] GENI, exploring networks of the future, Online on 19/03/2012 at : http://www.geni.net/.
[42] Geographic addressing and routing for vehicular communications (GeoNet), FP7 ICT, Online on 08/05/2012 at: http://www.geonet-project.eu/?page id=9
[43] H. Soliman, C. Castelluccia, K. ElMalki Athonet, L. Bellier " Hierarchical Mobile IPv6 (HMIPv6) Mobility Management", October 2008
[44] I. Chakeres, C. Perkins, "Dynamic MANET On-demand (DYMO) Routing" - draft-ietf-manet-dymo-21, July 26, 2010
[45] IANA. Mobile Node Identifier Option Subtypes. [Online]. http://www.iana.org/assignments/mobility-parameters/mobility-parameters.xml#mobility-parameters-5
[46] IEEE standards association. Guidelines for 64-bit Global Identifier (EUI-64™) Registration Authority. [Online]. http://standards.ieee.org/develop/regauth/tut/eui64.pdf
[47] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4-2003, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)", New York, IEEE Press. October 1, 2003.

[48] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. SIGCOMM '01, pages 149–160. ACM, 2001

[49] ISO, "ISO/IEC 18092:2004 Information technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1)",. Retrieved December 2011.

[50] ISO-3779. Véhicules routiers -- Numéro d'identification des véhicules (VIN) -- Contenu et structure. [Online]. http://www.iso.org/iso/fr/catalogue_detail?csnumber=9305

[51] ISO-3780. Véhicules routiers -- Code d'identification mondiale des constructeurs (WMI). [Online]. http://www.iso.org/iso/fr/catalogue_detail?csnumber=45844

[52] J. Arkko and I. van Beijnum. RFC5534 - Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming. IETF, 2009.

[53] J. Day. Patterns in Network Architecture, a return to fundamentals, chapter 5, "Backgroundon Naming and Addressing", pages 141–183. Prentice Hall, 2008.

[54] J. Laganier, T. Koponen, and L. Eggert. RFC5203 - Host Identity Protocol (HIP) RegistrationExtension. IETF, 2008.

[55] J. Roberts. The clean-slate approach to future internet design: a survey of research initiatives.Springer, 2009.

[56] J. Saltzer. RFC1498 - On the Naming and Binding of Network Destinations. IETF, 1993.

[57] J. Vasseur, A. Dunkels, Interconnection Smart Objects with IP, Morgan-Kaufmann Pulishers, 2010.

[58] J.F. Shoch. A note on Inter-Network Naming, Addressing, and Routing. Internet ExperimentNote, 1978.

[59] J.H. Saltzer, D.P. Reed, and D.D. Clark. End-to-End Arguments in System Design. In MIT Lab for computer science. ACM, November 1984.

[60] Jin Ho Kim, Choong Seon Hong, Taeshik Shon " A Lightweight NEMO Protocol to Support 6LoWPAN", October 5, 2008

[61] K. Kim, S. Yoo, H. Kim, S. Daniel Park and J. Lee, "Interoperability of 6LoWPAN", Internet draft: draft-daniel-6LoWPANinteroperability- 01.txt, July 2005.

[62] L. Heuser, Z. Nochta, and N-C. Trunk. ICT Shaping The World : A Scientific View , chapter 5, "Towards The Internet of Things". John Wiley & sons, 2008.

[63] L. Mathy and L. Iannone. LISP-DHT : Towards a DHT to map identifiers onto locators. In ReArch'08. ACM, December 2008

[64] L. Yeh, T. Tsou, M. Boucadair, and J. Schoenwaelder, "Prefix Pool Option for DHCPv6 Relay Agents on Provider Edge Routers," Internet Draft, IETF, draft-yeh-dhc-dhcpv6-prefix-pool-opt-05, 2011.

[65] M. Bagnulo. RFC5535 - Hash-Based Addresses (HBA). IETF, 2009.

[66] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker. ROFL : Routing On Flat Labels. In SIGCOMM'06. ACM, September 2006.

[67] M. Kicherer, T. Schlichter, L. Voelker (DE), "METHOD FOR DETERMINING AN ADDRESS OF A COMPONENT OF A VEHICLE," German US Patent Application Publication US2012/0054340A1, Mar. 1, 2012.

[68] M. Mouton, A. Petrescu, and C. Janneteau, "default Router List Option for DHCPv6 (DRLO)," draft-mouton-mif-dhcpv6-drlo-00, 2011.

[69] M. O'Dell. 8+8 - An Alternate Addressing Architecture for IPv6 (Inetrnet Draft). Online on 19/03/2012 at: http: // www. potaroo. net/ ietf/ all-ids/ draft-odell-8+ 8-00. txt , 1996.

[70] M. O'Dell. GSE - An Alternate Addressing Architecture for IPv6 (Internet Draft). IETF, 1997.
[71] M.S. Blumental and D.D. Clark. Rethinking the Design of the Internet : The End-to-End Arguments vs. the Brave New World. In ACM Transactions on Internet Technology. ACM, August 2001.
[72] Manabu Tsukada, Yacine Khaled and Thierry Ernst. "Basic and Advanced features of IPv6 Over C2C NETs", July 31, 2009, INRIA Report.
[73] M-K. Shin, T. Camilo, J. Silva, D. Kaspar, "Mobility Support in 6LoWPAN" - draft-shin-6lowpan-mobility-01 November 11, 2007.
[74] MPTCP, MultiPath TCP - Linux Kernel implementation, Online on 20/03/2012 at : http: //mptcp.info.ucl.ac.be/
[75] N. Lutchansky, "IPv6 Router Advertisement Prefix Delegation Option," draft-lutchann-ipv6-delegate-option-00, 2002.
[76] National Highway Traffic Safety Administration. Vehicle Identification Numbers (VINs). [Online]. http://www.nhtsa.gov/Vehicle+Safety/Vehicle-Related+Theft/Vehicle+Identification+Numbers+%28VINs%29
[77] NFC Forum."Technical Specifications". Retrieved December 2011.
[78] O. Troan and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6," RFC 3633, 2003.
[79] P. Nikander, A. Gurtov, and T.R. Henderson. Host Identity Protocol (HIP) : Connectivity, Mobility, Multi-Homing, Security, and Privacy over IPv4 and IPv6 Networks. IEEE Communications Surveys & Tutorials, 2010.
[80] P. Nikander, T. Henderson, C. Vogt, and J. Arkko. RFC5202 - End-Host Mobility and Multihoming with the Host Identity Protocol. IETF, 2008..
[81] P. Saint-Andre. RFC6120 - Extensible Messaging and Presence Protocol (XMPP) : Core.IETF, 2011.
[82] Paul Francis and Ramakrishna Gummadi. IPNL : A NAT-extended internet architecture. SIGCOMM '01, pages 69–80. ACM, 2001
[83] R. Droms, et al., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, 2003.
[84] R. Droms, P. Thubert, F. Dupont, W. Haddad, and C. Bernardos, "DHCPv6 Prefix delegation for NEMO," draft-ietf-mext-nemo-pd-07, 2010.
[85] R. Hinden and B. Haberman, "Unique Local IPv6 Unicast Addresses," RFC 4193, 2005.
[86] R. Hinden. RFC1955 - New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG. IETF, 1996.
[87] R. Koodli "Fast Handovers for Mobile IPv6", July 2005
[88] R. Moskowitz and P. Nikander. RFC4423 - Host Identity Protocol (HIP) Architecture. IETF, 2006.
[89] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. RFC5201 - Host Identity Protocol.IETF, 2008.
[90] R. Wakikawa, T. Clausen, B. McCarthy, A. Petrescu "MANEMO Topology and Addressing Architecture" - draft-wakikawa-manemoarch-00.txt, July 1, 2007
[91] R.E. Kahn, S.A. Gronemeyer, and R.C. Burchfiel, J.and Kunzelman. Advances in Packet Radio Technology. In Proceedings of the IEEE, vol. 66, no. 11. IEEE, November 1978.
[92] RFC 4283. IANA Considerations. [Online]. http://tools.ietf.org/html/rfc4283#page-5

[93] S. Akhshabi and C. Dovrolis. The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture. SIGCOMM'11, 2011.
[94] S. Chakrabarti, S. Daniel Park "LowPan Mobility Requirements and Goals" - draft-chakrabarti-mobopts-lowpan-req-01, March 1, 2007C. Perkins, Ed., D. Johnson, J. Arkko, "RFC6275: Mobility Support in IPv6" , July 2011
[95] S. Deering. Watching the Waist of the Protocol Hourglass. IETF 51 plenary, London, 2001.
[96] S. Gundavelli, Ed., K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, "RFC5213:Proxy Mobile IPv6", August 2008
[97] S.D. Strowes. Compact routing for the future internet. PhD thesis, University of Glasgow, 2012.
[98] Sagem, "AT COMMAND SET HILO/HILONC MODULES", May 2010
[99] Sample Vehicle History Report. [Online]. http://www.dmv.org/sample-autocheck-report.php
[100] Standards/Australian Design Rules for Vehicle. (2005) Vehicle Standard (Australian Design Rule 61/02 - Vehicle Marking) 2005. [Online]. http://www.comlaw.gov.au/Details/F2005L03994DMV.org.
[101] T. Clausen, Ed., P. Jacquet, Ed., "RFC3626: Optimized Link State Routing Protocol (OLSR)",October 2003
[102] T. Aura. RFC3972 - Cryptographically Generated Addresses (CGA). IETF, 2005
[103] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6," RFC4861, 2007.
[104] T. Savolainen and J. Korhonen, "Sateless IPv6 Prefix Delegation for IPv6 enabled networks," draft-savolainen-stateless-pd-01, 2010.
[105] T. Winter, P. Thubert, JP. Vasseur, et al. draft-ietf-roll-rpl, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks", work in progress. Available at tools.ietf.org.
[106] TRIAD, Translating Relaying Internet Architecture integrating Active Directories, Online on 20/03/2012 at : http://www-dsg.stanford.edu/triad/.
[107] Trilogy, Architecting the future Internet, Online on 20/03/2012 at : http://trilogy-project.org/.
[108] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard. Networking Named Content. CoNEXT'09, 2009.
[109] V.P. Kafle, H. Otsuki, and M. Inoue. An ID/locator split architecture for future networks. Communications Magazine, IEEE, 48(2) :138 –144, february 2010.
[110] Wiki book. VIN Model Year. [Online]. http://en.wikibooks.org/wiki/Vehicle_Identification_Numbers_%28VIN_codes%29/Model_year
[111] X. Yang, D. Clark, and A.W. Berger. NIRA : A New Inter-Domain Routing Architecture. IEEE/ACM Transactions on Networking '07. IEEE/ACM, 2005..
[112] Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, and E. Lear. RFC1918 – Address Allocation for Private Internets. IETF, 1996.
[113] Z. Shelby, S. Chakrabarti, E. Nordmark, "Neighbor Discovery Optimization for Low Power and Lossy Networks (6LoWPAN)" - draft-ietf-6lowpan-nd-17, June 13, 2011
[114] Zach Shelby, Carsten Bormann "6LoWPAN: The Wireless Embedded Internet", 2009 John Wiley & Sons Ltd
[115] ZigBee Alliance, "Understanding ZigBee Gateway", September 2010.

[116] -, Ognjanovic, N. (ed.), “Description of baseline reference systems, scenarios, technical requirements & evaluation methodology”, Deliverable 2.1 (D2.1), project EXALTED, May 2011.