Vandenberghe *et al. EURASIP Journal on Wireless Communications and Networking* 2012, **2012**:316
http://jwcn.eurasipjournals.com/content/2012/1/316

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Vehicular ad hoc networking based on the incorporation of geographical information in the IPv6 header

Wim Vandenberghe[1*], Erwin Van de Velde[2], Chris Blondia[2], Ingrid Moerman[1] and Piet Demeester[1]

Abstract

Several approaches can be identified in the domain of vehicular ad hoc networks (VANET). Internet Protocol version 6 (IPv6) networking and non-IP geographical networking can each fulfill a subset of the application requirements. In general, a combination of both techniques is proposed to meet all of the application requirements. In this case, packets of one VANET routing protocol are encapsulated inside packets of another. This tunneling, together with the position service required for non-IP geographical unicasting, makes such a combined solution rather complex, and hence more challenging to implement, debug, and maintain. In this article, a new VANET approach is presented that relies on the key assumptions that geo-anycast functionality is not required by the applications, and that geographic unicasting is not needed when IP-based unicasting is provided. This enables the adoption of an IPv6-only VANET solution, removing the need for tunneling and position services. New techniques are required to support IPv6-based geo-broadcasting. In this article, it is described how addresses should be assigned, how geographical data can be incorporated in the IPv6 address, how the other IPv6 header fields can be used to contain additional VANET information, and how routing should be handled to guarantee that no modifications are required to the application units. The implementation of the proposed techniques is described, and the correct functionality of the solutions is experimentally demonstrated. Finally, to prove the added value compared to current state-of-the-art propositions, the presented solution is stacked up against the recently released ETSI standards TS 102 636-4-1 (geographical addressing and forwarding) and TS 102 636-6-1 (transmission of IPv6 packets over GeoNetworking protocols).

Keywords: Vehicular ad hoc networks, VANET, IPv6, Geographical networking

Introduction

Intelligent transport systems (ITS) are ICT systems that enable a more efficient and safer traffic through the use of diverse technologies. In the ITS domain, cooperative systems allow innovative applications that rely on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure communication to increase the time horizon, the quality, and reliability of information available to the drivers about the road conditions and other vehicles in their immediate environment. To enable such forms of interaction, vehicles equipped with local wireless communication interfaces are interconnected in vehicular ad hoc networks (VANET).

Numerous research efforts have already been put into the VANET domain. Routing and broadcasting protocols have been developed in several initiatives, using Internet Protocol version 6 (IPv6) as well as non-IP-based geonetworking solutions, each providing different functionalities for data exchange and dissemination. To cover all communication requirements imposed by the applications, several efforts have focused on combining IPv6 and geonetworking into a common ITS communication system architecture. This is typically based on the encapsulation of packets from one VANET routing protocol inside packets of another. This tunneling, together with the position service required for non-IP geographical unicasting, makes current combined solution rather complex.

However, in the domain of software development, one of the rules of thumb is that simplicity should be a key

*Correspondence: wim.vandenberghe@intec.ugent.be
[1] Department of Information Technology (INTEC), Ghent University, IBBT, Ghent, Belgium
Full list of author information is available at the end of the article

goal in design, and that unnecessary complexity should be avoided. This design principle is often referred to with the famous acronym KISS (Keep it Simple, Stupid!). The advantage of this principle is that it strives to make code easier to implement, debug, and maintain. The result is a positive influence on both software quality and development cost. An example of a software system where the KISS principle is rigorously and successfully applied is the Unix operating system [1]. Another example is the Internet, for which the architectural principles are described in RFC 1958 [2]. In that document, it is literally mentioned that a general design issue is to keep it simple. When in doubt during design, one should choose the simplest solution. Based on these two examples and previous experiences in design and development of ad hoc network protocols, the authors of this article firmly believe that the pursuit of simplicity in designs would be of great benefit in the domain of VANETs. This conviction is further strengthened by the observation that VANETs will be very heterogeneous, consisting of vehicles of many different brands, each possibly equipped with other implementations of the applicable VANET protocols and standards. For a satisfactory operation of the VANET, it is of the utter importance that each of these implementations is robust and unambiguous. Adoption of the KISS design principles facilitates this goal.

The goal of this article is to present an approach to VANET that fulfills all imposed application requirements in a simple yet effective manner. It is an evolution of previous study [3]. Compared to this previous publication, the approach presented in this article is characterized by an overhauled mechanism to support delay tolerant networking and a novel approach to intra-station dissemination of VANET packets. Several other new elements were added, such as a detailed implementation description, an experimental validation of the functionality and a thorough comparison with the applicable European Telecommunications Standards Institute (ETSI) standardization efforts. These ETSI specifications regarding geographical addressing and forwarding [4] and transmission of IPv6 packets over GeoNetworking protocols [5] were released recently, and represent the current state-of-the-art in combined VANET networking solutions. The article is organized as follows: the "VANET networking techniques" section introduces the different available approaches to VANET networking. The "Communication requirements" section performs an analysis of the required data dissemination functionalities from an application point of view, and recites other design requirements imposed by the typical ITS architecture. The "Solution description" section describes the technical details of the proposed solution, and the "Implementation details" section presents the actual implementation. An evaluation of the proposed approach to VANET networking is performed in the "Evaluation" section. Finally, the article is concluded with a Conclusions section.

VANET networking techniques

The numerous VANET networking techniques that have been developed in the past research efforts can be divided in three classes: IPv6-based networking techniques, non-IP solutions, and combined approaches.

IPv6 networking

Applying IPv6 for VANET networking has some significant advantages [6,7]. First of all, IP can support all types of ITS applications, while allowing developers to rely on established networking APIs. IP can also bring (legacy) Internet applications (web browsing, video streaming, peer-to-peer file sharing, online gaming, etc.) to the vehicles. Since it is the *de facto* standard for data exchange, IP ensures interoperability of ITS communication systems with other communication systems. Using IP, applications can run transparently over diverse underlying communication media.

The most important reason to adopt IPv6 in the VANET domain instead of the common IPv4 protocol is the fact that IPv4 does not provide a sufficient amount of available IP addresses. Because IPv4 addresses are 32-bits long, the size of the entire address space is 2^{32} or approximately 4.3 billion, of which the major part has already been assigned. On a global level, the Internet Assigned Numbers Authority (IANA) allocated the last available addresses on February 3, 2011. On a regional level, the unallocated address pool is already exhausted for one regional Internet registries (APNIC which is responsible for example China and India) and it is estimated that the other regions will follow within a few years [8]. IPv6 addresses have a length of 128 bits, resulting in an address space size of 2^{128}, resolving the address exhaustion problem. Other advantages of IPv6 are the inherent auto-configuration capabilities and network mobility support.

A disadvantage of IPv6 is that it has no built-in notion of geographical information. This means that it does not support concepts such as geocasting where data are disseminated to vehicles within a given geographical area. Therefore, routing protocols have to rely on topology information instead of geographic information. Typically, IPv6 VANET routing protocols extend existing ad hoc protocols with techniques to improve performance and reliability. Several publications exist that focus on enhancing reactive ad hoc routing protocols such as the Ad Hoc On-Demand Distance Vector Routing protocol. The notion of link and route lifetime estimates has been introduced, based on velocity vectors and other movement information [9,10]. Other studies focus on restricting the flooding of the route requests [11,12]. Proactive ad hoc

routing protocols such as the Optimized Link State Routing Protocol (OLSR) were also extended with VANET optimizations. The movement predication-based routing framework adjusted OLSR to prefer most stable paths instead of shortest paths [13], while the authors of [14] proposes DHT-OLSR, combining OLSR with techniques from the domain of peer-to-peer networking: dynamic clustering and distributed hash table routing. The third class of existing ad hoc protocols, the hybrid routing protocols such as the Zone Routing Protocol (ZRP), have also been optimized for the VANET scenario. The adaptive ZRP enhances the performance of ZRP with the use of a variable zone radius for every node, based on a metric called route failure rate [15], while the Sharp Hybrid Adaptive Routing Protocol monitors traffic patterns and local network characteristics such as link failure rate and node degree to determine zone sizes [16].

Non-IP networking

Topology broadcasts

Topology broadcast protocols disseminate packets from a source node to all nodes located at a specific distance, in terms of hops. WAVE Short Message Protocol (WSMP) and CALM FAST are the two most important topology broadcast protocols that aim to achieve higher repetitive broadcasting efficiency by substituting the IP protocol. WSMP is standardized by IEEE as part of the IEEE 1609.3 standard [17]. It defines a short message header, containing information such as WSM length, version number, security info, application class, application data and transmission power, rate, and channel. The length of the packet is 9 bytes plus the variable byte size of the application context data. WSMP only supports single-hop broadcasting, not multi-hop. CALM FAST is a networking protocol currently being standardized by ISO [18], combining networking and protocol layer functionalities. It is based on a two-octet network header containing the source and destination address of the packet. The protocol is primarily designed for single-hop communications, although it supports *n*-hop broadcasts in the optional FAST transport protocol extension [18].

Geographic networking

The basic idea behind geographic networking is that nodes can be addressed using geographic concepts such as locations and areas, and routing decisions can be based on inter-node distance, relative movement, etc. Depending on the destination type, several geo-routing schemes may be used. Geo-unicast routes data from a source node to a single destination node which is identified by its exact geographical location. Since this location will change over time due to the mobility inherent to VANET nodes, a position service is required that maintains a mapping in real-time between vehicle identity and exact location. Geo-anycasting refers to the situation where data are routed from a source node to one random node that is located within a defined geographical broadcasting area. Geo-broadcasting is used when data is routed from a source node to all nodes located within a defined geographical area.

The different publications in the domain of geographic networking can be classified in a set of common techniques. One of them that is applied by many is opportunistic broadcasting, where the probability that a node B will retransmit a broadcast message sent by node A is dependent of the distance between A and B: the greater the distance, the higher the probability that B will rebroadcast [19-21]. Another common technique is irresponsible forwarding, where the probability that node B will rebroadcast the broadcast message of A is dependent of the neighborhood density [22-24]. Greedy forwarding lets the sender node A itself select the next node B that has to rebroadcast the message, aiming to achieve a maximum traveling distance per rebroadcast [25-27]. In urban environments, intersection routing strategies are often utilized [28-30].

Combined solutions

It is very likely that VANETs will have to support the different functionalities provided by both the IPv6 and the non-IP solutions for the actual deployment of cooperative applications. This vision was extensively researched in the GeoNet project. The project investigated how IPv6 connectivity can be provided in combination with the non-IP-based networking protocols CALM FAST and the C2C-CC geographic networking protocol. It was chosen to encapsulate IPv6 packets in C2CNet packets to transport them within the GeoNet domain [31]. During the course of the project, this tunneling technique was implemented, validated, and evaluated [32]. It could be concluded that based on this approach all requirements can indeed be met in a satisfactory fashion.

However, tunneling makes the solutions more complex compared to a single-protocol solution. As elaborated in the introductory section 2, unnecessary complexity makes designs more challenging to implement, debug, and maintain in future products. Higher levels of complexity can also raise interoperability issues. In the "Complexity analysis" section, this aspect is discussed in more detail.

The GeoNet work was continued in two different tracks. The first is the further implementation in the CarGeo6 initiative. This is a joint effort of the Tunisian school ESPRIT and the French institute of research in computer science, INRIA. In contradiction to the original proprietary GeoNet implementations, CarGeo6 is an open-source implementation of IPv6 GeoNetworking, conforming with GeoNet specifications. A validation of CarGeo6 can be found in [33]. The second track is

the standardization of the GeoNet specifications by the ETSI. In the following subsections, the two corresponding standards are introduced.

ETSI technical specification TS 102 636-4-1

The ETSI technical specification TS 102 636-4-1 [4] standardizes a non-IP approach to geographical addressing and forwarding. One of the focal points is the definition of the appropriate packet headers. As illustrated in Figure 1, in total seven different headers are defined. Every header type begins with a common header, followed by an extended header which is different for every type. The header length varies between 36 and 88 bytes, dependent of the specific extended header. The header can contain geographical information about the network nodes. It is included in the long position vectors for source and sender, for which not only the coordinates but also speed, heading, and altitude are always given, together with information regarding acquisition time and accuracy. The destination area in case of geo-casting can be specified using the GeoBroadcast/GeoAnycast extended header. The shape of area can be a circle, a rectangle, or an ellipse. Regarding the unique identification of the network nodes, the standard combines station type, station subtype, country, and a unique identification derived from the MAC address. The standard also describes mechanisms that allow all network nodes to maintain a local location table which contains information about other ITS stations in the neighborhood. This is especially useful in case of geo-unicast traffic. To support the communication of IPv6 packets over the VANET, the standard requires the GeoNetworking to IPv6 Adaption Sub-Layer (GN6ASL) which is discussed in the following subsection.

ETSI technical specification TS 102 636-6-1

To support the transmission of IPv6 packets over the ETSI GeoNetworking protocols described in the previous subsection, the GeoNetworking to IPv6 Adaption Sub-Layer (GN6ASL) is proposed by ETSI in TS 102 636-6-1 [5]. It is an additional layer at the networking level that is positioned between the GeoNetworking layer and the IPv6 networking layer. It integrates the functionalities of the IPv6 and GeoNetworking implementations in a transparent way. This means that from the IPv6 point of view, GN6ASL is just a link-layer protocol which is responsible for the communication of an IPv6 packet between two IPv6 nodes connected to the same link. From the GeoNetworking point of view, GN6ASL is just a higher layer that produces data packets that have to be geo-unicasted or geo-broadcasted to a specific destination. No adaptations to any of the two implementations are required, both protocols are not even aware of each others existence.

The main concept behind GN6ASL is the virtual link. A virtual link can span multiple physical links. Two types are defined. A geographical virtual link (GVL) is a multicast-capable virtual link with geographically scoped boundaries. It is associated with one single GeoNetworking GeoBroadcast/GeoAnycast area (also called geoarea). A topological virtual link (TVL) is a non-broadcast multi-access virtual link with topologically scoped boundaries. IPv6 multicast traffic may not be exchanged through a TVL. Both types of virtual links are associated to physical interfaces in a slightly different manner: one physical interface can be associated to multiple GVLs but to only one TVL. The GN6ASL layer provides the virtual interfaces to IPv6 in the form of virtual network interfaces. Such virtual interfaces can be assigned to either GVLs or TVLs, but a single virtual interface can only be associated to one GVL or one TVL.

To configure the virtual links and the IPv6 network on top of them, standard IPv6 stateless address autoconfiguration (SLAAC) is applied. IPv6 routers broadcast Router Advertisements on their configured GVLs using the corresponding virtual interfaces. This is translated by GN6ASL to geo-broadcast ETSI GeoNetworking packets with a destination area equal to the corresponding GVL area. Upon the reception of such an advertisement by other nodes within that area, these nodes will check if they have already configured this GVL. If not, they will create it using the destination area of the GeoBroadcast packet, configure a new corresponding virtual interface, and forward the packet over that interface to the IPv6 layer for further SLAAC execution. This approach is most suitable in the situation where roadside units (RSUs) provide wireless coverage. In that case, each consecutive RSU is responsible for its own area, and will continuously broadcast Router Advertisements on the corresponding GVL. Vehicles will then continuously join and leave GVLs based on their current position. The overhead on the wireless medium is limited to the broadcasts of the RSUs. However, on roads where no such RSU coverage is available, vehicles are themselves responsible for creating GVLs that enable them to communicate with the surrounding vehicles. If all vehicles start defining a GVL around its own position, and hence continuously broadcast router advertisements, this will result in a significant load on the wireless medium. However, if too few nodes try to set up a GVL, this could lead to situations where vehicles are within each others' communication range but fail to actually communicate IPv6 packets. A suitable protocol that takes these trade-offs into account is required, but not yet available.

Based on these virtual links, IPv6 packets can be encapsulated in GeoNetworking packets. In the case of traffic originating from the IPv6 layer (outbound traffic), the IPv6 layer selects the outgoing interface itself. If this interface is one of the virtual interfaces associated to a GVL or TVL, the packet is processed by the GN6ASL. The packet

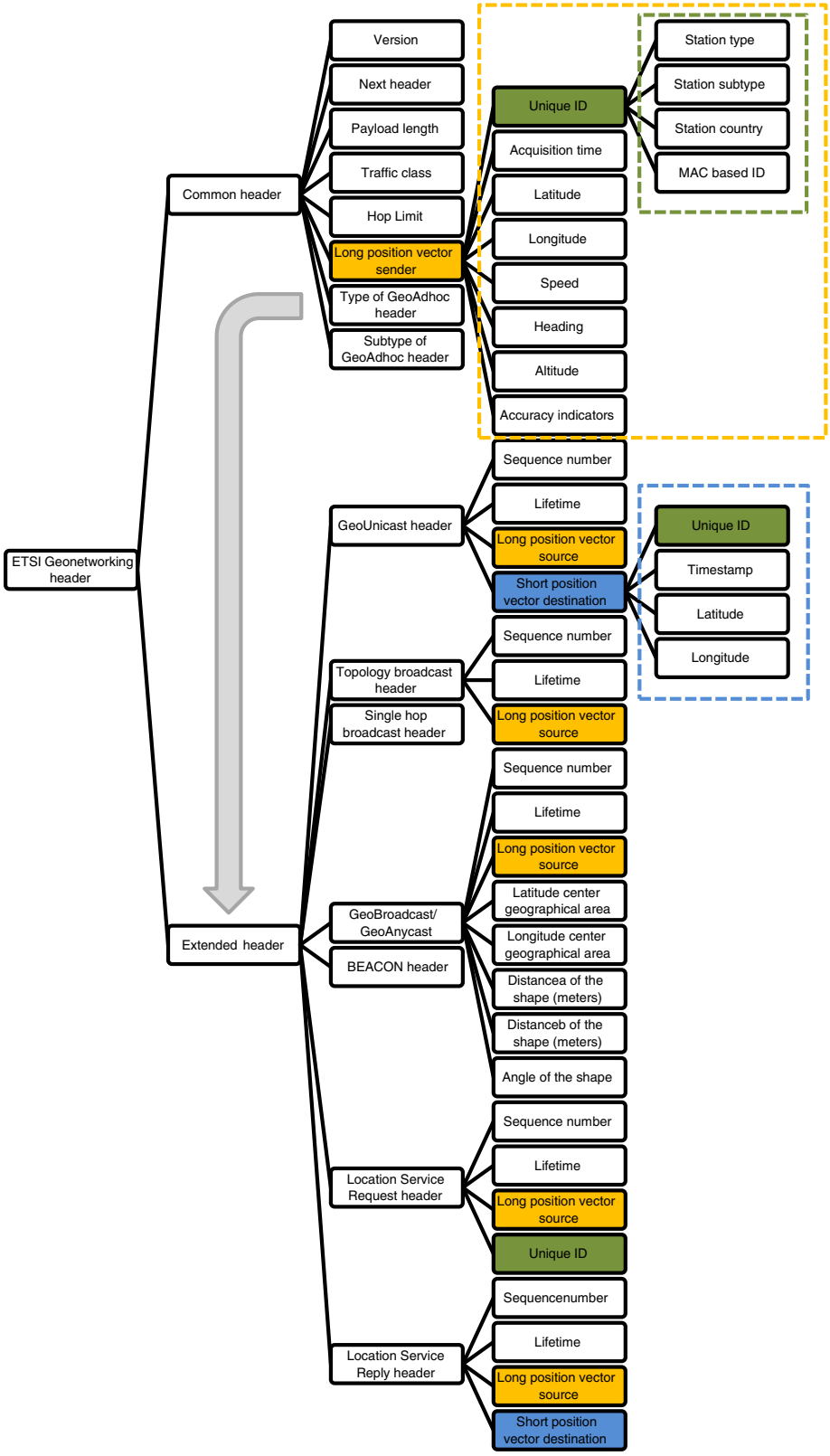

Figure 1 Summary of ETSI GeoNetworking header structure as defined in TS 102 636-4-1. For clarity reasons, the subelements of the long and short position vectors and of the unique ID are given only once.

will be encapsulated in a GeoNetworking packet. In case of a multicast IPv6 packet, this will be a GeoBroadcast packet with a destination area equal to the GVL area that corresponds with the virtual interface that received the packet. In case of unicast traffic, this will be a GeoUnicast packet for which the destination address is directly derived from the interface identifier (IID, last 64 bits of the IPv6 address) of the destination IPv6 address. This kind of address resolution without neighbor discovery is possible because the TS 102 636-6-1 standard states that a compliant ITS station should use IPv6 addresses containing IIDs that directly resolve to its GeoNetworking address. Quite similar is the processing by GN6ASL of received GeoNetworking packets transporting IPv6 packets (inbound traffic). The main challenge there is the determination of the virtual link that the packet belongs to. In case of GeoBroadcast messages, a GVL is searched for which the GVL area matches the destination area of the GeoBroadcast packet. In case of a GeoUnicast header, a slightly more complex procedure takes link scope and the relation between the position of the source and the available GVLs into account.

Communication requirements

Application requirements

An enumeration of common cooperative ITS applications is given by different standardization organizations such as the C2C-CC and ETSI [34,35]. An analysis of their different requirements regarding supported networking techniques can result in a common list of requirements imposed on any generic VANET networking solution.

The C2C-CC investigated a large number of use cases. Based on that analysis, the consortium was able to define six generic applications that together can support all use cases. "Vehicle-2-Vehicle Cooperative Awareness" supports the requirement for applications to share information with each other without any persistent communication link between the vehicles [36]. The corresponding messages are called Cooperative Awareness Messages (CAM). "Vehicle-2-Vehicle Unicast Exchange" enables a communication link between two vehicles for the exchange of information. "Vehicle-2-Vehicle Decentralized Environmental Notification" provides information about events and roadway characteristics that are probably interesting to drivers for a certain time in a certain area [37]. The corresponding messages are called Decentralized Environment Notification Messages (DENM). "Infrastructure-2-Vehicle (One-Way)" supports the communication from RSUs to vehicles without a persistent communication link between vehicles and RSUs. "Local RSU connection" supports use cases where data between a vehicle and a RSU need to be sent from the vehicle to the RSU or bi-directionally. The last application, "Internet Protocol Roadside Unit Connection", supports services that are offered to the driver by servers located in the Internet. A technical analysis of all six applications, containing among others the required communication techniques, is described in the C2C-CC Manifesto [34]. An overview is given in Table 1. The results are in line with the results obtained in the ETSI application analysis which deduced seven basic applications from a larger set of use cases. From Table 1, it can be concluded that from an application point of view, a VANET should support some form of unicasting (IP- or geographical-based), topology broadcasting and geo-broadcasting.

Design requirements

The European ITS Communication Architecture described in the COMeSafety project [38] defined the ITS station as the core component in the four different instantiations (vehicle, personal, roadside, and central station). ITS Vehicle Stations and ITS Roadside Stations consist of a Communication & Control Unit (CCU) and one or more Application Units (AU). The CCU shall be equipped with at least a single ITS external communication interface to provide connectivity to external networks. This will typically be a short-range wireless network interface for VANET communication, often accompanied by a mobile data network interface for continuous Internet connectivity. Both the CCU and the AUs are also equipped

Table 1 General capabilities for C2C-CC applications

Application name	V2V Cooperative awareness	V2V unicast exchange	V2V decentralized environmental notification	Infrastructure 2 vehicle (one-way)	Local RSU connection	Internet protocol roadside unit connection
Communication type	Topology broadcast, Geo-broadcast	Unicast	Topology broadcast, Geo-broadcast	Topology broadcast, Geo-broadcast	Unicast	Unicast
Communication range	300 m to 1 km	0 m to 5 km	300 m to 20 km	300 m to 5 km	0 m to 1 km	0 m to full radio range
Roadside units	N/A	N/A	Not required but can assist	Required	Required	Required
Security	V2V trust	V2V trust	Originator trust	Vehicle must trust RSU	RSU/OBU must trust each other	Internet security (IPSec, appl. layer)

with an ITS internal communication interface for data exchange between the different ITS Station components, typically an Ethernet interface. Important communication requirements for this architecture are given in [39]:

- It must remain as close to the IP standard as possible, no strong modifications to the IP stack of the involved components should be required.
- Because of radio throughput limitations, the introduced overhead should be kept as low as possible.
- No modifications should be required in the AUs, since these can be IP standard legacy devices.

Solution description

In the previous sections an overview of possible VANET networking approaches was given. Based on an analysis of cooperative applications it was determined which of the networking techniques have to be supported by all VANET solutions. These requirements were supplemented with requirements imposed by the European ITS Communication Architecture. Based on this set of requirements, the VANET networking solution presented in this article was designed.

In the "Application requirements" section, it was determined that unicasting, topology broadcasting, and geo-broadcasting should be supported. However, no requirements are imposed on the unicast technique: as long as it is possible to address and route data from a source to one well-defined destination, it does not matter if this destination is defined on an IP basis, or on a geographical basis. Combined solutions based on packet encapsulation (section "Combined solutions") support both, but introduce a complexity in the routing process that could be avoided. If a networking solution would choose to only support IPv6 unicasting, and not geographical unicasting, it would still fulfill the communication requirements, but it would not need to support the more complex tunneling. Another advantage is that the position service for geographical unicasting is not required in IP unicasting. In a similar manner, support for geo-anycasting can also be omitted without violating the application requirements, since only geo-broadcasting is demanded.

Based on these observations, the VANET networking solution presented in this article is designed as a pure IPv6 solution. This way, unicasting implementation is straightforward, and topology broadcasting can easily be supported by combining multicasting with correct usage of the Hop Limit header field. Geo-broadcasting however requires further refinement of the proposed solution. A mechanism is required to incorporate all required geographical data in the standard IPv6 header. In the following subsections our VANET networking solution is described in more detail.

Automatic address assignment

The main idea behind the chosen approach for automatic address assignment is the fact that the CCU receives a valid IPv6 address block from its operator or ISP (if the CCU has no mobile Internet uplink, this is performed once in a special configuration session at home, in the garage, etc.), divides this in smaller subnets, dedicates a subnet to every attached network (VANET, Internet uplink, internal station network, etc.) and correctly configures its own interfaces. Once these steps are performed, it starts broadcasting IPv6 router advertisements on its local network interfaces, just like any other IPv6 router would, enabling IPv6 SLAAC for all connected AUs [40].

The IETF recommends that mobile networks such as vehicles or mobile phones with an additional network interface (such as Bluetooth or 802.11) should receive a static/64 prefix to allow the connection of multiple devices through one subnet [41]. However, this range is too small for modern vehicular networks where the CCU is a mobile router interconnecting multiple subnetworks over its different network interfaces. Since the IPv6 Addressing Architecture defines that all global unicast addresses other than those that start with binary 000 have a 64-bit IID field [42], the CCU cannot divide the received prefix in smaller subnets. Therefore, we propose that the CCU should follow the general case described in [41], and receive a /48 prefix. This way, the CCU can construct the global unicast address: the first 48 bits contain the received static prefix, the next 16 bits define the subnets for the different connected networks, and the last 64 bits are the IID which is constructed using standard stateless configuration mechanisms.

This simple approach needs no adjustments to the AUs. The addresses used for the VANET networking interface of the CCU are dependent on the static prefix assigned to the CCU, therefore guaranteed to be unique. This means that there is no need to support the Neighbor Discovery protocol within the VANET, which would require more complex techniques to be supported in the VANET domain [6]. The only partial functionality of the Neighbor Discovery protocol that our solution does require on the VANET is address resolution to translates the next hop IPv6 address to the corresponding 802.11p MAC address. However, in an IPv6 ad hoc network, the next hop during multi-hop forwarding (which is determined by an active VANET routing protocol) will always be a node within communication range of the sender. This means that the neighbor solicitation and advertisement messages can be limited to single-hop broadcasts, which can directly be translated to MAC broadcast messages.

IPv6 geographical addressing scheme

This article defines a VANET networking solution based entirely on IPv6. Since geo-broadcasting is a crucial

requirement for many cooperative applications, a mechanism that incorporates the necessary geographical data is indispensable. To define this mechanism, a more profound insight in the required geographical data is required. Kovacs [31] defines a position vector containing the following data fields: MAC id, C2C NET ID, timestamp, position in latitude, longitude and altitude, and speed and heading. Similar data fields are defined in the long position vector defined in [4] (see "Comparison with ETSI technical specification TS 102 636-4-1" section). Since this article chooses to work with IPv6 only, the MAC id and C2C NET ID can be dispensed. Position in latitude and longitude is absolutely required for geocasting. A timestamp allows delay tolerant nodes to destroy queued packets or to keep valid messages alive in the geographical destination area, without being aware of the applied upper layer protocols. Heading information can be used to limit rebroadcasting to certain areas, which could be a valuable mechanism to tackle the broadcast storm problem in dense networking conditions. Speed and altitude are considered by the authors of this article as less decisive parameters for the addressing of nodes in a VANET context, and will not be included in the proposed solution. This decision is motivated by the observation that the applications analyzed in the "Application requirements" section rarely include speed or altitude information when defining the destination for their messages. Hence, it makes no sense to incorporate these concepts on the networking level. In the exceptional case that such functionality would be required, a more suitable approach is to apply a filter mechanism on the application layer that takes speed and/or altitude into account.

Different techniques to include this geographical information in VANETs are introduced by Gordillo [39] and Khaled [43]. Besides the tunneling approach that we wish to avoid in this article, they proposed two other solutions. The first approach is to include it in the application layer, e.g., using an extended DNS that is capable of storing geographic information. This could easily be implemented, but it is not really adapted to a mobile environment, and the scalability of this approach is unclear. The second approach is to include all information in the IPv6 protocol. This can be done in three different ways: all information can be put in the IPv6 destination address using a special addressing scheme, it can be put in the existing IPv6 header fields by redefining their interpretation, or it can be encoded in a newly defined IPv6 Extension Header.

Applying a new IPv6 Extension Header allows a comprehensible integration of the required geographical data into the IPv6 packets. However, the downside is that is causes additional overhead on top of the 40 bytes standard IPv6 header. Since one of the design requirements is that overhead should be kept as low as possible, it is preferred to define an addressing scheme that can incorporate all required geographical data in the standard IPv6 header. A suitable addressing scheme is depicted in Table 2. It has some similarities with the format of the IPv6 multicast address for a circular area presented in [44]. However, the approach presented in Table 2 uses another technique to encode area coordinates, and it omits the group ID of [44] in favor of the introduction of two headings to define the broadcasting zone within the circular area, and a timestamp to support delay tolerant networking.

As in all IPv6 addresses, the first 8 bits define the address type [42]. For geo-broadcasting we will use multicast packets, hence the value of these bits should be 0xFF. The next 4 bits, the flags, indicate various properties: if an address of a multicast rendezvous point is embedded in the multicast address, if the multicast address is assigned based on the network prefix, and if the multicast address is permanently assigned by the IANA. In our case, the response to those three statements is negative, therefore the appropriate value for this field is 0×1. The next 4 bits define the scope. Since this addressing scheme will be used to address other nodes within the VANET, none of the available assigned values such as link-local, site-local, or global seems entirely appropriate. It was chosen to use one of the unassigned scope values, since these are defined as available for administrators to define additional multicast regions [42]. In our solution, a scope value of 0×6 indicates VANET (geo- or topology) broadcast. The next 32 bits contain the latitude of the center of the geocast area as a signed integer of 32 bits, represented in micro-degrees. The following 32 bits contain the longitude. The next 16 bits represent an unsigned integer that defines the radius of the circular communication area in meters. This allows ranges up to 65 km. The following 4 bits contain Heading 1. It is an unsigned integer, leading to 16 possible values. Therefore, the actual heading value of the GPS receiver should be mapped to the closest fixed heading as depicted in Figure 2. Together with Heading 2 (the next 4 bits), the broadcast area is defined as the section between the circular area and the zone between headings 1 and 2 in

Table 2 Geographical IPv6 addressing scheme

8 bits	4 bits	4 bits	32 bits	32 bits	16 bits	4 bits	4 bits	24 bits
Multicast	**Flags**	**Scope**	**Latitude**	**Longitude**	**Radius**	**Heading 1**	**Heading 2**	**Expiration time**
0xFF	0x1	0x6	int32	int32	uint16	uint4	uint4	uint24

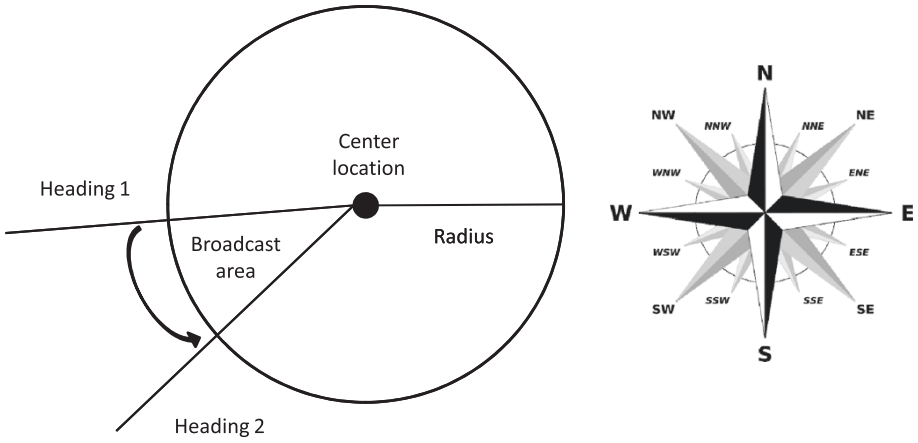

Figure 2 Headings concept.

a counterclockwise direction. Circular geo-broadcasting can be achieved by putting the 0×0 value in both heading fields.

The last 24 bits are used as an expiration timestamp. They represent the least significant bits of the corresponding Unix time, a 32-bit value defining the number of seconds elapsed since midnight of January 1, 1970. To handle buffer overflows introduced by this reduction in bitsize (where the value of a timestamp in the future can become smaller than the current time), we define the maximum supported difference between current time and expiration time as half of the interval, or 2^{23} s. Nodes will only consider a packet as expired when both of the following requirements are met (all time values are in the 24 bits format): the current system time is equal to or higher than the packet expiration timestamp and the absolute difference between the current time and the expiration timestamp is lower than the maximum supported difference. This way, timestamp overflow will not result in the unnecessary expiration of the packet. Hence, our solution supports a packet validity of up to 97 days, with a granularity of 1 s. To allow geo-broadcasting without limitations in the time domain, 0×0 is defined as a special value for the packet expiration timestamp field.

Interpretation of IPv6 header parameters

The proposed addressing scheme allows to define receiver nodes in both place and time. In practical implementations, additional information will be required. This is added to the standard IPv6 header by reinterpreting some existing IPv6 header fields, as illustrated in Table 3. A notion of packet ID, not available in the standard IPv6 header, can assist flooding mechanisms to avoid double retransmission of the same packet. Such an ID is stored in the Flow Label field of the IPv6 header. The triplet source address, destination address, and flow label is then used to identify a unique packet. For every new VANET packet created on a host, the used Flow Label value has to be increased with 1. If applications on a host would be responsible for defining the value of the Flow Label field themselves, it is possible that they would be using the same values at the same time. If these different messages are sent to the same destination (e.g., topology broadcast), they will be considered as equal by the other nodes of the VANET. This way, new messages could wrongfully be discarded. Therefore, the configuration of the Flow Label field has to be coordinated by a global service on the host.

To support topology broadcasts, correct values need to be used in the Hop Limit field in combination with a special destination address equal to FF16::1. The traffic class value is used to signal the IEEE 802.11p (or ITS-G5A) MAC layer which priority to be used. The source of any (geo-)broadcast packet always has to be the global unicast IPv6 address of the sender, since multicast addresses may not be used as source addresses [42]. This also allows unicast communication with nodes that were discovered through (geo-)broadcast announcement.

Table 3 Interpretation of IPv6 header parameters

4 bits	8 bits	20 bits	16 bits	8 bits	8 bits	128 bits	128 bits
Version	Traffic class	Flow label	Payload length	Next header	Hop limit	Source address	Destination address
0x6	QoS class 802.11p	Packet ID			Topology broadcast: # hops	Global VANET unicast address	Topology broadcast: FF16::1

Table 4 Addressing scheme for intra-station multicasting of VANET packets (based on RFC 4489)

8 bits	4 bits	4 bits	8 bits	8 bits	64 bits	32 bits
Multicast	Flags	Scope	Reserved	Plen	IID	Group ID
0xFF	0x3	0x2	0x00	0xFF	IID of CCU interface on intra-station subnet	0x0000FF16

Routing methodology

The presented VANET solution requires no changes on the AUs IP stack. Applications can translate their communication needs to specific destination addresses and header values, and create the correct packets to forward to the CCU. To send this data, it is only required that the AU routing table contains an entry indicating that the gateway for packets in the FF16::/16 domain is the CCU. To receive certain data, standard multicast groups cannot be applied over the VANET, since the used addresses do not correspond to predefined geographical zones. Hence, the applications cannot determine multicast group IDs to join. To solve this problem, one specific link-local multicast group is defined within the ITS station on every connected subnet. The CCU will multicast a copy of every received VANET packet to these intra-station groups. The address of a group is determined using the method described in RFC 4489 [45]. As depicted in Table 4, the address will start with FF32:00FF, followed by the interface ID of the CCU intra-station interface connected to that subnet. The last 32 bits can be chosen by the host, we define the value 0x0000FF16 to indicate this special multicast service. Its configuration is straightforward: the CCU provides standard multicast management functions according to the specified addressing scheme on all connected subnets. Application units interested in receiving VANET traffic derive the appropriate multicast address from the IID of their default gateway (which is the CCU interface on their local network). To join this group they can rely on their standard multicast capabilities.

The CCU routing functionality requires some adjustments on the IP stack. It has to interpret the used geocast destination addresses, and has to know how to forward them on the VANET. The other way around, it has to be able to decide if it's within the broadcast range of a geocast message received on the VANET interface, and to duplicate and forward it to the appropriate intra-station multicast groups. To decide when and how to rebroadcast or route messages, it can apply any desired broadcasting or ad hoc networking protocol. This flexibility is an important advantage of the proposed solution. It facilitates the fast adoption of novel VANET protocols which are regularly published in academic literature. Such protocols can be both VANET (geo)broadcasting schemes and IP-based networking protocols which are specifically optimized for the context of VANETs.

This routing methodology is illustrated in Figure 3. In this example, the CCU of every ITS station received a

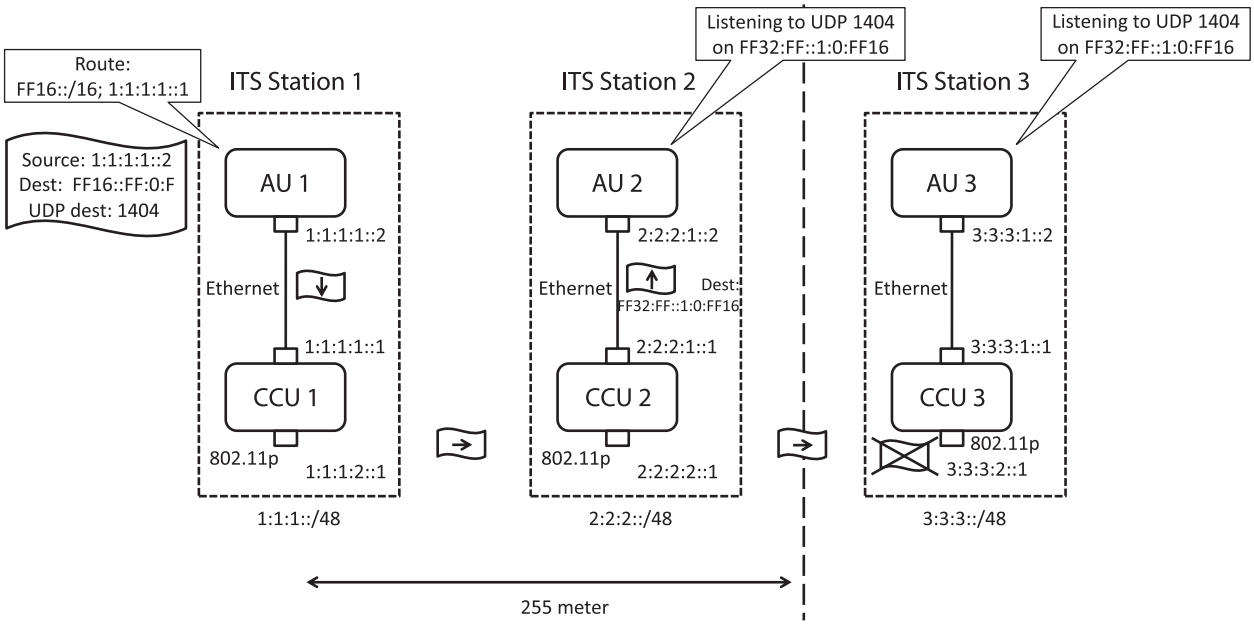

Figure 3 Illustration of routing methodology.

static /48 prefix. As described in the "Automatic address assignment" section, this prefix is divided in several subdomains (here LAN and VANET) using the next 16 bits, and IPv6 SLAAC ensures the correct configuration of the AU units connected to the CCU. To maintain clarity in this example, the IID part of the addresses is simplified. The VANET routing protocol implemented on the CCU is simple flooding. A cooperative safety application is active on all AU nodes, in this example it uses the UDP port 1404 to exchange messages. To receive VANET packets, the application listens to this UDP port on the appropriate intra-station multicast group. The routing table of every AU describes its CCU as the next hop destinations in the FF16::/16 domain.

The example starts when the cooperative safety application running on the AU of ITS Station 1 generates a warning that has to be geographically broadcasted. It creates an UDP/IPv6 packet with its own IPv6 global unicast address (1:1:1:1::2) as source address. The destination address is constructed using the scheme presented in the "IPv6 geographical addressing scheme" section. To keep this example clear, the location of ITS Station 1 is defined as 0 degrees latitude and 0 degrees longitude. It is chosen to define a circular communication area (hence headings 1 and 2 are both 0×0), with a radius of 255 m (0xFF). The expiration time is 15 s (0xF). Hence, the destination address is FF16::FF:0:F. The UDP destination port of the packet is 1404. When this packet is transferred to the (standard) routing layer of the AU, it looks up the gateway for that packet, which is CCU 1, and forwards it to that CCU over the Ethernet interface. When CCU 1 receives this packet, its modified IPv6 routing layer recognizes this packet as a geo-broadcast IPv6 packet, and broadcasts it on the VANET 802.11p interface. This broadcasted packet is received by the CCU of ITS Station 2, which again recognizes it as a geo-broadcast packet. Based on its own location, it determines that it is positioned within the destination communication area. The CCU 2 networking stack multicasts a copy of the packet to every subnet within the ITS station. In this case, this is the Ethernet LAN. In the example, we use simplified IIDs, for the CCU 2 LAN interface this is ::1. Therefore, the multicast address for disseminating VANET packets on this LAN is FF32:FF::1:0:FF16. After delivery of a copy to all subnets, the packet is given to the VANET routing protocol for further relaying. The simple flooding of this example decides to rebroadcast the original packet through its VANET interface. This packet is then received by CCU 3. Based on its location, this CCU decides that it is outside of the destination area, and destroys the packet.

The above example focused on the geo-broadcasting aspect of the proposed VANET solution. However, support for the other required communication modes is quite similar. If AU 1 would have chosen to disseminate the message using topology broadcasting with a hop limit of 3, the difference would be that the destination address of the generated packet becomes FF16::1, as described in the "Interpretation of IPv6 header parameters " section. Then the packet is be given to CCU 1, which reduces the hop limit value with 1 (new value = 2) and then broadcasts it on the VANET interface. CCU 2 receives this packet, but performs no geographical filtering, it reduces the hop limit (new value = 1), forwards a copy to AU 2 and rebroadcasts the original on its VANET interface. This packet is then received by CCU 3, which will destroy the packet because any relay action by CCU 3 results in the value of hop limit becoming 0. In case of unicast traffic, the destination address of the packet becomes the global unicast address of the receiver. To support this form of communication, it is required that a unicast VANET ad hoc networking protocol is active on every CCU. This way, routes between the different CCUs are maintained. In the routing table of every AU, its CCU is configured as default gateway. This way, the created unicast packet will be forwarded by the (standard) routing layer of the AU to the CCU, which hands it over to the unicast ad hoc protocol which can deliver the packet of the VANET to the CCU which corresponds with the destination AU. This CCU will then forward the packet to the destination AU.

Implementation details

The approach to VANET networking presented in the previous sections was validated with an actual implementation. Two frameworks were applied: the Click Modular Router was used to implemented the networking stack, while the applications were developed in Java. The implementation of both the networking layer and the demonstrator applications could be performed within limited manpower constraints, while providing all required functionality. This illustrates the benefits of reducing the level of complexity in VANET design. In the next subsections, more details are given regarding both aspects of the implementation.

Networking layer

As mentioned, the networking aspect of our solution was implemented in the Click Modular Router framework. This is a modular software router platform originally developed by MIT with subsequent development by a broad research community [46]. Its main strength is that the modular design enables very efficient prototyping of networking protocols. In a Click configuration, modules providing basic functionality (called elements) are interconnected. Packets flow through this chain of elements. The configuration that corresponds with the presented VANET solution is depicted in Figure 4. At first sight it might seem rather extensive, but the applied color coding eases the process of grasping the configuration details. All

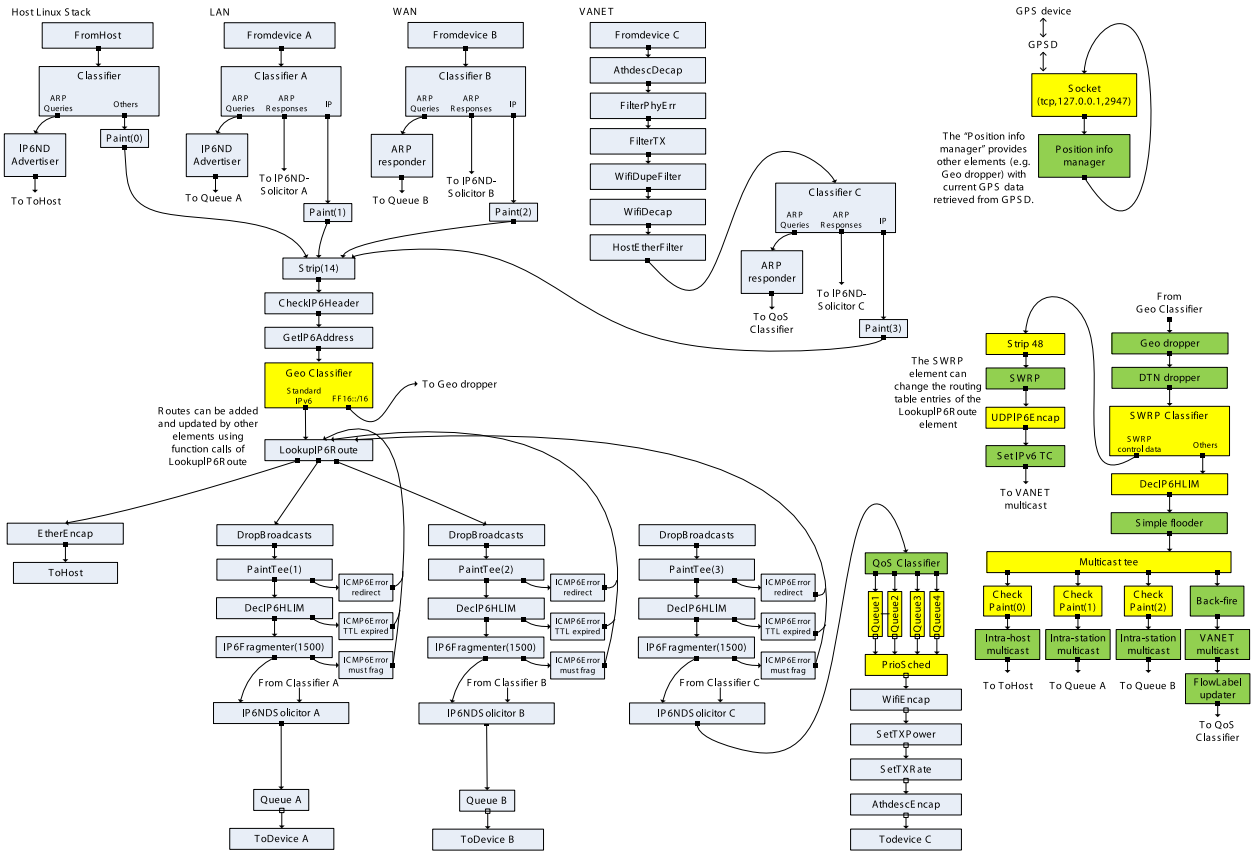

Figure 4 Click modular router configuration of the implemented networking stack. Light grey elements correspond with a standard IPv6 router, yellow elements are additional VANET elements which are part of the click standard library, green elements were specifically developed for the VANET networking stack.

elements in light grey are the elements included in a standard IPv6 router. The yellow elements are elements which are part of the Click standard library but where inserted in the configuration to implement our VANET solution. The green elements are novel elements that were specifically implemented in the context of this VANET research. Hence, in the entire figure, only the green elements had to be programmed to obtain a full implementation of our solution.

The starting point of our implementation was the configuration corresponding with a standard IPv6 router. As mentioned, all required elements and the corresponding configuration are part of the Click standard library. In our specific case, the router is connected to the Linux stack of the host, and three network interfaces: an intra-station Ethernet LAN and Wi-Fi WLAN, and the VANET IEEE 802.11p interface. To provide maximum control of all VANET transmission aspects, the VANET interface (which relies on the Madwifi wireless driver) was configured in a special mode called "monitor mode". This mode requires the presence of some additional elements to perform error checks normally executed in the driver itself. Examples are the FilterPhyErr, FilterTx, and WifeDupeFilter elements.

Two adjustments were then made to the standard IPv6 router configuration. First of all, on the outgoing VANET interface, packets are classified according to their annotated QoS class and put in different queues. A priority scheduler is placed after them. As a result, every time that the VANET interface is ready to transmit a new packet, packets with the highest priority will be selected for transmission first. Only if the queue corresponding with the highest QoS class is empty, packets will be retrieved from the second highest QoS class queue. Only if that queue is also empty, packets can be retrieved from the third highest QoS class queue, and so on. The second adjustment was the introduction of a classifier in the forwarding chain of the router. If messages in the FF16::/16 domain pass this classifier, they do not flow into the standard lookup table of the router, but to the part on the right of the figure which provides all geo-networking functionality.

The entry point of this geo-networking implementation is the Geo dropper element. When it receives a packet, it investigates the IPv6 destination address of the packet. It

also requests the current node location at the position info manager (an element that periodically polls the Linux daemon GPSD to retrieve the current position and GPS time). If the processed packet is a topology broadcast packet, it will forward it to the next element. If it is a geo-broadcast packet, it will checked if the current position of the node is part of the destination area. If so, the packet is passed to the next element, otherwise the packet is destroyed. The DTN dropper is the next element in the chain. It has a similar function as the Geo dropper element, but it focuses on time instead of location. It destroys expired geo-broadcast packets and forwards the other packets to the next element. This next element is a filter (based on the usage of a dedicated UDP port) for control messages of the Simplified Wireless Routing Protocol (SWRP). This protocol is a simplified version of the wireless routing protocol [47]. It is a pro-active ad hoc routing protocol that exchanges routing tables between one-hop neighbors. It uses the topology broadcast address FF16::1 together with a hop limit of 1 to disseminate the routing table of a node to its peers. The SWRP element provides the core functionality of the protocol, it relies on other elements placed before and after it to add and remove network and transport headers and to assign highest priority to its messages.

Geo-networking packets not destined for the SWRP element are passed to the DecIP6HLIM element, which reduces the value in the IPv6 Hop Limit field by one. In contradiction to the similar element in the standard IPv6 section of the Click configuration, there is no connection to an element that sends ICMP6 error messages to the source network interface. On the VANET, such functionality would cause unnecessary capacity waste. The next element in the forwarding chain is the Simple flooder element. It maintains a list of all packets that it recently forwarded. As described in the "Interpretation of IPv6 header parameters" section, the triplet source address, destination address, and flow label are used to uniquely identify each packet. If the Simple flooder element receives a packet that is already included in its list, then the packet is destroyed. Else the corresponding ID is added to the list, and the packet is passed to the next element: the Multicast Tee. This element does nothing more then duplicating a message received on its input to all its outputs. This means that a copy of the geo-networking packet is forwarded to the Linux networking stack of the host, the intra-station LAN and WLAN, and the VANET interface. The check paint elements on the outgoing connections of the Multicast Tee make sure that the packet is not forwarded to the router input from which it originated (host, LAN, or WLAN). On the VANET interface such an inspection is not foreseen since in this context message relaying requires that packets can be forwarded on the same network interface as the one that they were received on. Before the packets are put in the outgoing queue of each network interface, the appropriate MAC headers are placed in front of the IPv6 geo-networking packet. This is carried out by the different multicast elements. In case of intra-station multicasting, the applied addresses correspond with the scheme described in the "Routing methodology" section. In case of VANET traffic, the destination MAC address always is the broadcast address FF:FF:FF:FF:FF:FF. On the VANET, a last check is also performed before handing the packet over to the VANET interface. The value of the Flow Label field is inspected, if no value has been set and the packet originated from this ITS station, the field is updated with the appropriate value.

Application layer

The implementation of the networking layer was demonstrated on public roads with different common cooperative ITS applications: road works warning, emergency vehicle warning, bicycle collision alert, and broken down vehicle warning. For the interested reader, video footage of this demonstration can be found on YouTube [48]. In the first application, a message was continuously broadcasted by a RSU which was temporary installed at a road works site. The onboard unit (OBU) of the demo vehicles received these messages and warned the driver when the hazardous location was approached. The emergency vehicle warning was an example of V2V communication, allowing the driver of the emergency vehicle to electronically notify the surrounding vehicles to give way. The bicycle collision alert demonstrated the value of ad hoc communication between vehicles and vulnerable road users. In case of limited visibility at a junction, the cooperative application warned the driver in case of a collision course with the bicycle. A similar use case was demonstrated with the last application: broken down vehicle warning. In this case the goal was again to inform the driver about hazards in case of limited visibility, but the communication pattern was V2V instead of vehicle-to-bicycle.

These applications were implemented in the regular Java SE Platform. No additional libraries were required except for one common Java class that can convert a desired geo-networking communication form into the corresponding IPv6 destination address. Such a class was straightforward to implement, and can be made publicly available if the need would arise from the VANET community. During application development, it was sufficient to configure the correct destination address, hop limit, and traffic class for a new packet based on the desired communication behavior. This can easily be done using the MulticastSocket class available in the Java SE SDK, in combination with the address generator introduced above. To illustrate this programming methodology, a coding example based on our own application code is shown below.

```
//In this example, all required variables to define the communication behavior are
//command line arguments
String myVanetAddress=args[0];
String destinationVanetAddress=args[1]; //unicast, FF16::1 or determined by the library
int toPort=Integer.parseInt(args[2]);
int trafficClass=Integer.parseInt(args[3]);
int hopLimit=Integer.parseInt(args[4]);

//create the socket
InetSocketAddress myAddress = new InetSocketAddress(myVanetAddress,0);
MulticastSocket senderSocket = new MulticastSocket(myAddress);
senderSocket.setTrafficClass(trafficClass);
senderSocket.setTimeToLive(hopLimit);

//create the message
testMessage = new TestMessage(); //dummy object, implements the interface Serializable
ByteArrayOutputStream bos = new ByteArrayOutputStream();
ObjectOutputStream oos = new ObjectOutputStream(bos);
oos.writeObject(testMessage);

//send the message
InetAddress toAddress = InetAddress.getByName(destinationVanetAddress);
DatagramPacket msg =
    new DatagramPacket(bos.toByteArray(), bos.toByteArray().length, toAddress, toPort);
senderSocket.send(msg);
```

Evaluation

The main goal of the VANET networking solution presented in this article is to fulfill all imposed requirements in a simple yet effective manner. In this section, it is evaluated to which degree this goal has been achieved.

In the "Communication requirements" section it was concluded that a VANET should support some form of unicasting, topology broadcasting, and geo-broadcasting. Since our solution supports IPv6 unicast, IPv6 topology broadcasting, and geo-broadcast using the standard IPv6 header, this indeed is the case. Another requirement was that the proposed solution should remain as close as possible to the IP stack, should limit the overhead and should not require modifications to the AUs. These requirements are also fulfilled. So, in theory, the solution presented in this article fulfills all imposed requirements. To justify this conclusion, an experiment was performed on the IBBT w-iLab.t wireless testbed. This experiment is presented in the "Experimental validation of the provided functionality" section.

Assuming that the support of all required functionality has been experimentally proven, it is still unclear how our solution holds up against the alternative VANET techniques introduced in the "VANET networking techniques" section. In almost all available publications processed in the corresponding literature study, no information was given about the practical details regarding the applied network header format, addressing scheme, and so on. In general, the publications focused on the defined routing or broadcasting schemes, assuming that a networking stack (being IP based or geographical networking based) is available to support their solutions. The main exceptions were all publications related to the GeoNet project. This project pioneered the domain of combined VANET solutions. During the course of the project, several publications focused on the different design and implementation issues, touching topics such as network headers, addressing schemes, and so on. No similar literature could be found anywhere else.

Since the work of the GeoNet project (which was executed between 2008 and 2010) was continued in the recently released ETSI standards, it seemed most appropriate to focus on these standards when evaluating our solution against the related state-of-the-art. Therefore, in the "Comparison with ETSI technical specification TS 102 636-4-1" and "Comparison with ETSI technical specification TS 102 636-6-1" sections our approach is compared with the corresponding ETSI standards for geographical networking and IPv6 encapsulation. To conclude the evaluation of our solution, the "Complexity analysis" section will focus on the bigger picture, evaluating the reduction in complexity of our solution compared to the tunneling approach.

Experimental validation of the provided functionality

Because of the small number of VANET nodes involved in the demonstrator on public roads, in practice all communication types were limited to single-hop geo-broadcasts. To validate that the implementation of our proposed solution supports all required forms of communication, an additional experiment was performed. It was executed on real hardware, more specific on the wireless testbed IBBT w-iLab.t. This is a general purpose wireless testbed, which has been identified as a useful additional tool for VANET research in previous study. For more details regarding this testbed in the context of VANET research we refer the interested reader to [49]. As depicted in Figure 5, the topology of the nodes is a rectangular grid, the nodes are successively numbered from right to left. All IEEE 802.11 interfaces are equipped with fixed 10 dB attenuators to enable both large collision domains (at 23-dBm transmit power) as multi-hop experimentation (at 3-dBm transmit power). On the testbed we defined an experiment with 106 nodes representing a highway scenario with three

Figure 5 Topology of the third floor of the IBBT w-iLab.t wireless testbed. On the figure the physical nodes are depicted. On each of these devices, two IEEE 802.11 a/b/g interfaces are available. For each of those interfaces an entirely independent VANET networking stack is deployed on the physical node. Hence every depicted node actually corresponds with 2 VANET nodes during the experiment.

different communication patterns: single-hop topology broadcasting of CAM messages by all nodes, multi-hop geo-broadcasting of DENM messages originating from six different nodes (ID 1–6), and IPv6 unicast VoIP communication between two other nodes (from nodes 99 to 101). Based on [49], a transmit power of 13 dBm was chosen for this experiment. As shown in the coding example found in the "Application layer" section, applications can select the different types of communication by configuring the appropriate destination address, hop limit, and traffic class. Table 5 summarizes the values applied in the experiment.

To capture end-to-end performance statistics such as delay and packet success rate (PSR), a testing application was created. This application can be configured to emulate data flows according to the desired application characteristics. Tunable parameters are the data size of the created messages, interval between two consecutive messages, number of messages to be created, and a range of message identification numbers for which the performance metrics should be recorded. The actual content of the communicated messages is random dummy data. In contradiction to our demo applications which were implemented in Java, this testing application was implemented within the Click Modular Router framework. This way, the same tests can be performed on actual hardware and in the NS-2 network simulator (see "Experimental comparison with TS 102 636-4-1" section). To mimic CAM data traffic, messages with 300 bytes of application data were sent every 100 ms. The same characteristics were applied to emulate DENM traffic. These parameter values correspond with the results of [49]. In the VoIP case, messages with a data size of 80 bytes were sent every 30 ms. This corresponds with the behavior of the iSAC codec which Skype uses by default [50]. The results of the experiment are depicted in Figures 6 and 7.

Figure 6 illustrates the reception of CAM messages. On the left part of the figure, every point corresponds with the tuple (sender node ID, receiver node ID). The color of the point indicates the percentage of CAM broadcast messages originating from the sender that were actually received by the receiver. It varies between blue (no communication possible) and red (perfect communication). On the diagonal the sender and receiver are the same. On this point the value corresponds with the number of messages that the node actually created during the experiment. This way it can easily be inspected if the request data load was generated during the experiment. If so, the entire diagonal should be dark red. It can be observed that for every sender node, the nodes that received the broadcasted CAM messages are situated around the diagonal. This corresponds with nodes that are in the vicinity of the sender. The diversity in the amount of receiver nodes per sender was nearly identical over several runs of the experiment, and can be explained by the fact that the w-iLab.t testbed is an indoor testbed. The nodes are spread

Table 5 Parameter values applied in the experimental functionality validation

Application	Node ID	Destination address	Hop limit	Traffic class
CAM	1–110	FF16::1	1	3
DENM	1	FF16:424C:2521:406F:11C8:002D:0000:0000	255	3
DENM	2	FF16:424C:2521:406F:11C8:002D:0000:0000	255	3
DENM	3	FF16:424C:250E:406F:11DC:002D:0000:0000	255	3
DENM	4	FF16:424C:250E:406F:11DC:002D:0000:0000	255	3
DENM	5	FF16:424C:2504:406F:11C8:002D:0000:0000	255	3
DENM	6	FF16:424C:2504:406F:11C8:002D:0000:0000	255	3
VOIP	99	2001:2001:0101:0002:0280:48FF:FE54:BCC8	255	3

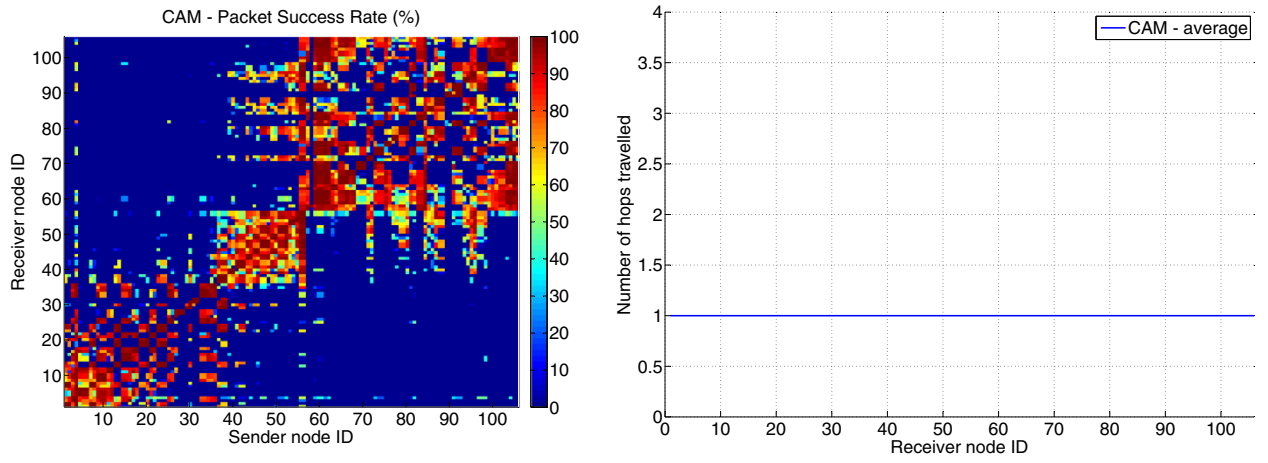

Figure 6 Reception of CAM messages on the IBBT w-iLab.t testbed. The color of each point indicates the percentage of CAM broadcast messages originating from the sender that were actually received by the receiver.

around an entire floor of an office building. Due to the presence of obstacles with different propagation characteristics (wooden office partition walls, concrete elevator shaft, brick walls around kitchen, and toilets), communication properties of links can vary greatly based on the specific location of sender and receiver node. To indicate that the CAM messages were limited to single-hop communication, the right part of Figure 6 depicts the average number of hops traveled before a CAM message was received. As desired, the value is 1 for all nodes. From the above, it can be concluded that the topology broadcast functionality is provided by our solution.

The left part of Figure 7 illustrates the reception of the DENM and VOIP messages. The X-axis corresponds with the receiver node ID, the Y-axis with the PSR. Every series on the figure corresponds with a different sender node. So one series on the figure illustrates to which degree the messages created by a single source could be received by all other nodes. To avoid overloading the figure, only half of the DENM sources are depicted. When focusing on the DENM results, it should be mentioned that the length of the testbed grid topology is about 90 m. The destination address of the DENM applications where configured in such a way that they define a circular area around the position of the source node, with a radius of 45 m (see Table 5). The source nodes 1–6 are all located at one end of the topology, close to each other. The used Hop Limit value is 255, allowing for multi-hop dissemination of the messages. As can be seen from Figure 7, the desired communication characteristics are achieved: the DENM messages are only received by half of the nodes, corresponding with all nodes located within the geo-broadcast

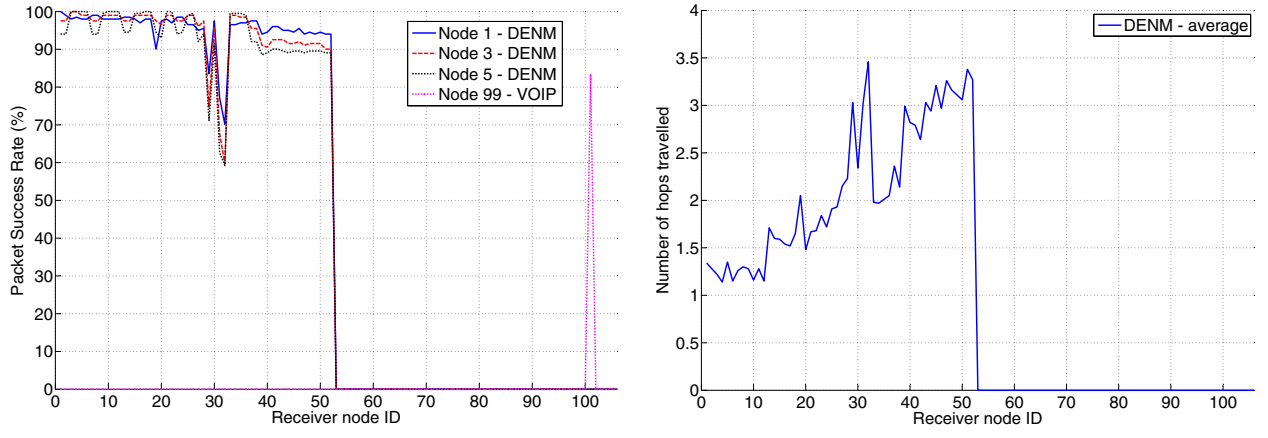

Figure 7 Reception of DENM and VoIP messages on the IBBT w-iLab.t testbed. On the left figure, one series illustrates to which degree the messages created by a single source could be received by all other nodes. The right figure illustrates the multi-hop nature of the DENM messages, depicting the average number of hops traveled for all DENM messages sent by nodes 1–6.

destination area. To indicate that the DENM messages were in fact relayed by intermediate nodes, the right part of Figure 7 depicts the average number of hops traveled before a DENM message originating from node 1–6 was received. This value gradually increases from 1.2 to 3.4. When focusing on the VOIP results, again the desired behavior is observed on the left part of Figure 7: only node 101 received the unicast messages sent out by node 99.

Comparison with ETSI technical specification TS 102 636-4-1

The comparison of the proposed solution with the ETSI TS 102 636-4-1 standard was approached both from a theoretical and an experimental point of view. In the following subsections both aspects are presented.

Theoretical comparison with TS 102 636-4-1

The geonetworking techniques described in TS 102 636-4-1 are quite similar to those proposed in this article. The most important difference between both approaches lies in the packet headers. As elaborated in the "ETSI technical specification TS 102 636-4-1" section, the ETSI standard defines seven different headers. This means that the networking layer has to implement seven different interpretations of the geonetworking header. In contradiction, as illustrated in Figure 8, the header format defined in this article is more straightforward. There is only a single header format, all information is always available at the same position in the header. During packet processing, the only point of attention is the interpretation of the destination address. If it is the special address FF16::1, it should be handled as a topology broadcast packet. If it is not this special address, but does start with 0xFF16, then it should be handled as a geo-broadcast packet according to the addressing scheme proposed in this article. Otherwise, the packet is handled as a standard IPv6 packet.

Other differences can also be observed. They are summarized in Table 6. The ETSI header contains geographical information about the network nodes. This is not the case in the proposed solution, which only applies

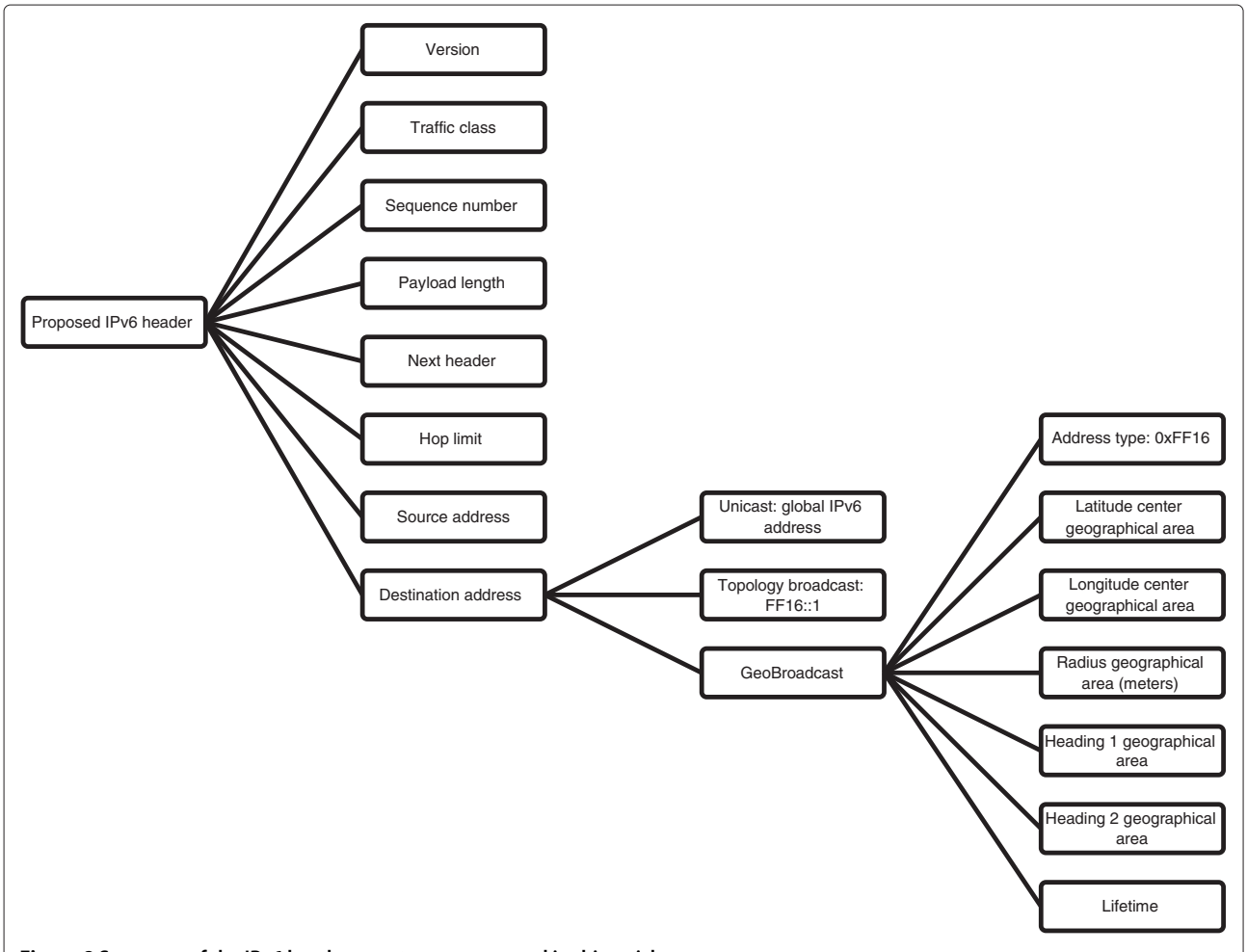

Figure 8 Summary of the IPv6 header structure as proposed in this article.

Table 6 Differences between ETSI TS 102 636-4-1 and the approach presented in this article

	ETSI TS 102 636-4-1	Vandenberghe et al.
Supported communication types	Unicast, topology broadcast, geo-broadcast	Unicast, topology broadcast, geo-broadcast
Number of packet header types	7	3
Header length	36–88 bytes, depending on type	40 bytes
Geographical information about network nodes	Latitude, longitude, speed, heading, altitude, accuracy indicators	None
Supported geo-broadcast area shape	Circle, rectangle, ellipse	Circle, wedge
Identification of network node	Station type + station subtype + station country + MAC derivative	Received global IPv6/48 prefix + MAC derivative
IP-compatibility	Requires GeoNetworking to IPv6 Adaption Sub-Layer (GN6ASL)	Natively

geographical information to define destination areas for geo-broadcasting. However, the absence of this information does not lead to reduced functionality compared to the ETSI standard since both support the same communication forms. These are unicast traffic between any two nodes in the network, topology broadcasting, and geographical broadcasting. It is true that the ETSI standard produces a local location table which contains information about other ITS stations in the neighborhood, and that this information could be useful for higher-layer applications. However, keeping the aim for a KISS approach into account, the authors of this article are convinced that if this information is required by the higher layers, that it should be collected and maintained on those higher layers. This opinion is strengthened by the Internet design guidelines described in RFC 1958 [2] (Architectural Principles of the Internet) which not only recommends the KISS principle, but also literally states that modularity is good, and that if you can keep things separate, you should do so. The most feasible approach seems to assign the task of creating a local location table to a higher layer service on every ITS station which makes this information available to all other applications. This approach would be in line with the European ITS Communication Architecture [38] which describes a facility layer between the networking layer and the application layer. One of the considered facilities is the support for a Local Dynamic Map, which corresponds with the creation of a local location table.

Another difference lies in the length of the header. In the ETSI solution, this varies between 36 and 88 bytes, while the IPv6 solution has a fixed header length of 40 bytes. Main reason for this difference lies in the fact that the ETSI header includes some more information than the IPv6-based proposition. Most of this information is included in the long position vectors for source and sender (the source is defined as the node that originates a packet, the sender as the node that has sent the packet, which will be different from the source during multi-hop forwarding). For both nodes, not only the coordinates but also speed, heading and altitude are always given, together with information regarding acquisition time and accuracy. In the proposed solution, no geographical information is given regarding source and sender nodes, since this is not required to provide all required communication forms. Similar to the remark regarding the location table, it can be stated that if an application needs to know location information regarding the source of a message, it is more convenient that the source application includes this information in the message payload. This approach is in line with the ETSI standards that describe message payload formats for cooperative applications (CAM [36] and DENM [37]). In both message types location information is included that describes the location of the event or source node. This information includes latitude and longitude, altitude, speed and heading.

This removal of redundant data leads to shorter header lengths, which is a valuable characteristic since VANETs can suffer from capacity problems under high vehicle densities [51,52]. If we consider that the majority of cooperative ITS applications will rely on CAM and DENM messages to exchange information, it can be stated that the typical payload for a secure VANET packet will be 300 bytes [49]. When taking the different header lengths into account (40 bytes IPv6, 36–88 bytes ETSI), it can be concluded that in case of the shortest ETSI header the medium occupation time for every message in our proposed solution is 1% longer than in the ETSI approach. However, in case of the longest ETSI header, the medium occupation time for of the ETSI approach becomes 14% longer than in our proposed solution. To evaluate the impact of this increase, an experiment using our implementation is presented in the "Experimental comparison with TS 102 636-4-1" section.

Some other small differences between the ETSI standard and the proposed solution can be observed. The shape of geo-broadcast areas is different: both techniques support the definition of circles, the ETSI standard also supports rectangles and ellipses, while the proposed solution also supports wedges. Both approaches allow the

adequate definition of broadcast zones, and this difference can be classified as negligible. Regarding the unique identification of the network nodes, the ETSI standard combines station type, station subtype, country, and a unique identification derived from the MAC address. In the proposed solution, the unique global IPv6 unicast address is derived from the /48 prefix received from its operator or ISP (see "Automatic address assignment " section), and an interface ID derived from the MAC address. The final difference between the ETSI standard and the proposed solution lies in the fact that the ETSI standard requires the GeoNetworking to IPv6 Adaption Sub-Layer (GN6ASL) [5] for the communication of IPv6 packets over the VANET, while this is natively supported in the proposed solution. This is one of the most important differences in complexity between both solutions, and will therefore be more elaborated on in the "Comparison with ETSI technical specification TS 102 636-6-1" section.

It can be concluded that the ETSI technical specification TS 102 636-4-1, which describes a non-IP approach to geographical addressing and forwarding, and the solution proposed in this article are quite similar. They provide the same forms of communication towards the upper layers, but the proposed solution is characterized by a more straightforward header design. This design leads to a reduced complexity in packet processing. The consequence of this design is that less geographical information regarding events and neighboring nodes is available in the proposed solution compared to the ETSI standard. However, this information is not required at the networking layer since it is already available at the facility and application layers of the European ITS Communication Architecture. On the other hand, the removal of this redundant data could potentially improve network performance. This claim is experimentally investigated in the next section.

Experimental comparison with TS 102 636-4-1

In this section, we present an experiment that was designed to investigate the performance gain introduced by our solution. Compared to the ETSI standards, our packets are 4 bytes longer for CAM messages (single hop topology broadcast), but 44 bytes shorter for DENM messages (multi-hop geo-broadcast). This difference in size is the result of our specific header design and removal of unnecessary data redundancy. In the preceding section, it was mentioned that this is a valuable characteristic since VANETs can suffer from capacity problems under high vehicle densities. This experiment quantifies this statement.

Since the VANET scalability problem is only clearly noticeable in large networks, the experiment was not performed on the IBBT w-iLab.t testbed, but in the NS-2 network simulator. The same test application is used as the one described in the "Experimental validation of the pro vided functionality" section. To improve simulation accuracy, the overhauled PHY/MAC layer implementation (included in NS-2 since version 2.34) was applied [53]. It introduces accumulative noise calculation during the entire packet transmission period. This greatly improves accuracy of the simulation in the context of the hidden node problem, one of the main drivers behind the VANET scalability issues. The simulated VANET interfaces were configured according to the IEEE 802.11p specifications. The simulation represents a 5-km long highway (6-lane) with a fixed average inter-vehicle distance. Three different distances were considered: 160 m (low traffic intensity), 80 m (normal traffic intensity), and 40 m (intense but flowing traffic) [49]. The experiment is divided in two simulations. The VANET solution presented in this article is deployed in both of them. In the first simulation, all CAM and DENM messages are 300 bytes long. In the second one the CAM messages are 296 bytes long and the DENM messages are 344 bytes long. All nodes create a CAM message every 100 ms, DENM messages are only created by the first six nodes occurring after the 1000-m mark. DENM generation interval is 100 ms, the destination area covers the entire simulated site.

The forwarding technique for geo-broadcasting is the same in both scenarios: opportunistic broadcasting. This technique expands the simple flooding technique (in which a node relays every unique packet once) with a mechanism to avoid redundant transmissions. As explained in the "Geographic networking" section, the probability that a node B will retransmit a broadcast message sent by node A is dependent of the distance between A and B: the greater the distance, the higher the probability that B will re-broadcast. This behavior is implemented by assigning different backoff times to the relaying nodes B. This time is proportional to the packet SNR value: the higher the SNR value, the longer the node has to wait before forwarding the packet. All chosen backoff times should have a value between 0 and 75 ms. If a node overhears the relaying of the message by another node during its own backoff time, it cancels its own forwarding of the packet. This approach results in a large geographical gain per message retransmission, while ensuring that in sparse networking conditions all neighbors get the opportunity to relay the message when necessary.

To assess the impact of the differences in packet size on the networking performance, first the PSR and delay values of both solutions were compared. In the scenarios with an inter-vehicle distance of 160 and 80 m the difference was hardly noticeable. The solution presented in this article performed a little better than the ETSI TS 102 636-4-1 standard, but the difference is negligible. However, in the scenario with an inter-vehicle distance of 40 m, the difference in DENM performance was significant. The average

PSR dropped from 92 to 70%, but the delay increased quite dramatically as depicted in Figure 9 (mind the logarithmic scale, average rises from 68 to 8281 ms). On this figure, only the nodes situated between 1000 m (node 150) and 4000 m (node 600) are taken into account. Nodes situated in the first and last kilometer of the simulated 5 km scenario are not considered. This is to avoid the unwanted inclusion of border effects in the performance analysis. In other words, we only take nodes into account for which the interference by neighboring nodes is entirely modeled (since all interfering nodes located before and after it are part of the simulation).

To explain the large difference between our solution and the ETSI standard under intense traffic circumstances, we investigated the amount of DENM packets transmitted by each node. The results are given in Figure 10. Nodes 150–155 are the DENM sources. Because they both create DENM packets and forward some DENM packets of the other sources, they transmit most DENM packets in the scenario corresponding with our solution (almost 600 messages in total, while each source creates 350 unique DENM messages). Because of the applied opportunistic broadcasting technique, all other nodes relay approximately 400 DENM messages. Note that in the case of simple flooding (where every node retransmits every received message once), this value would be equal to the amount of unique DENM messages ($6 \times 350 = 2100$). However, in the case of longer DENM messages as proposed by ETSI TS 102 636-4-1, it seems that the correct functioning of the opportunistic broadcasting algorithm is distorted. The amount of relayed DENM messages is much higher for nodes 150–400.

This distortion is the result of exceeding a critical channel congestion threshold when increasing the DENM message size with 44 bytes. As discussed by Ma and Chen [54], congested IEEE 802.11 ad hoc networks can suffer from a phenomenon called consecutive freeze process (CFP). This effect can be observed when a station that has just completed a transmission and that has a new packet to send chooses zero as its initial MAC backoff timer. It will start to transmit right after a DIFS, giving other stations no chances to back off. The impact of CFP is higher in case of broadcasting, leading to longer packet delays under network congestion. This longer delay causes more intermediate nodes to rebroadcast received DENM messages, since their backoff timer defined by the opportunistic forwarding scheme becomes empty before the nodes located further away get the opportunity to relay the message. This again leads to more load and hence congestion on the wireless channel, leading to longer CFP effects, leading to a further distortion of the opportunistic forwarding scheme, and so on.

To avoid this significant performance degradation, it is required that the load on the wireless channel is reduced in such a way that the critical threshold is no longer crossed. This can be done by reducing the DENM message generation rate, which was 10 Hz in all previous simulations (a common value for VANET-based applications [34,35]). As illustrated in Figure 11, reducing this to 6 Hz when applying the ETSI packet format results in a

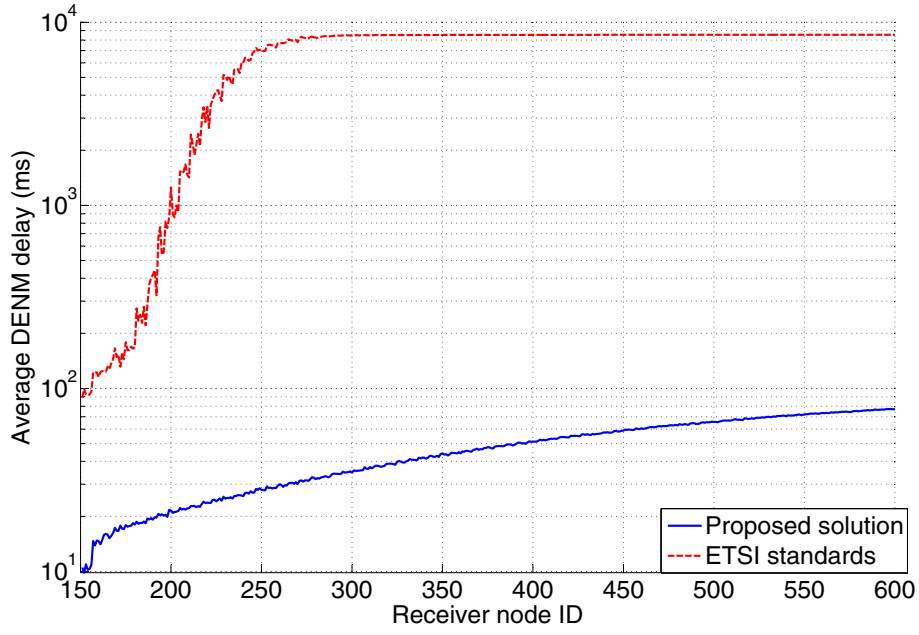

Figure 9 Comparison of the average DENM delay between the solution presented in this article and the ETSI approach.

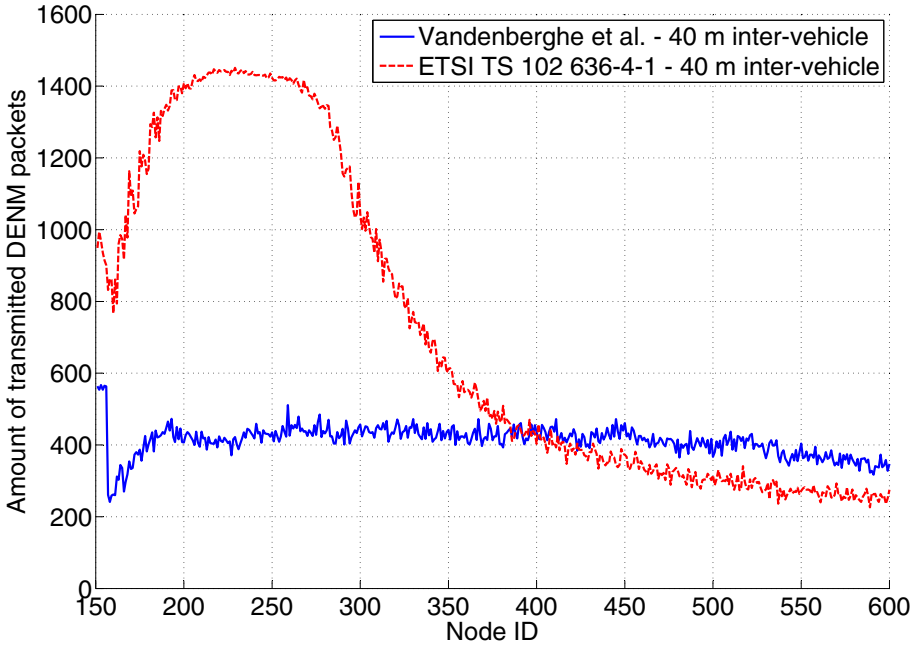

Figure 10 Comparison of the number of transmitted DENM delay between the solution presented in this article and the ETSI approach. The additional load caused by the longer DENM messages results in the exceeding of a critical channel congestion threshold. This causes the backfire broadcasting scheme to malfunction. This is illustrated by the significant rise in the amount of forwarded DENM messages (note that 2100 unique DENM messages are created in total during the experiment).

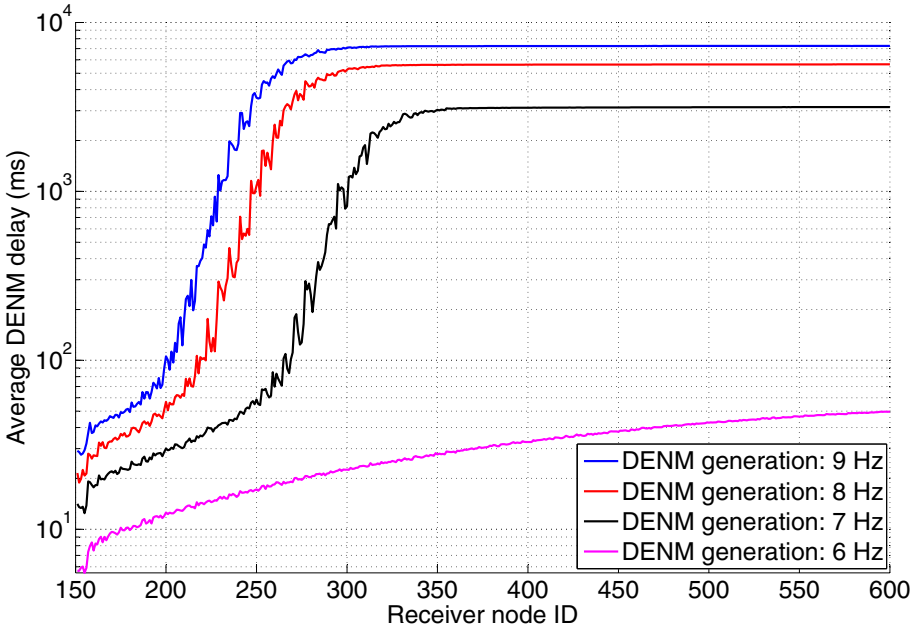

Figure 11 Effect of reducing the DENM packet generation rate in case of packet sizes as defined in ETSI TS 102 636-4-1.

DENM performance which is similar to that of the solution presented in this article at 10 Hz. Another approach could be to extend the opportunistic forwarding scheme with other VANET optimization techniques which further reduce the load on the wireless channel. Examples are probabilistic forwarding, dynamic adjustment of transmission power and rate, optimization of the applied MAC parameters, and so on. However, the exploration of such additional techniques is outside the scope of this article, and will be the subject of our future study.

To summarize, when comparing the solution presented in this article with the approach of the ETSI technical specification TS 102-636-4-1, no significant difference in performance could be observed in case of low and normal traffic intensity. However, in the case of intense (but flowing) traffic, a significant drop in DENM PSR and delay was noticed. After careful analysis, it could be concluded that this performance decrease is caused by the crossing of a critical network congestion threshold, activating a snowball effect that distorts the applied opportunistic forwarding scheme. To overcome this issue when applying the ETSI packet format, the DENM message generation rate has to be reduced from 10 to 6 Hz. Implementing additional VANET optimization techniques could be an alternative to overcome this issue without lowering the DENM generation rate, and will be investigated in future study.

Comparison with ETSI technical specification TS 102 636-6-1

A detailed description of the GN6ASL sub-layer was given in the "ETSI technical specification TS 102 636-6-1" section. Based on that description, it is now possible to compare the ETSI approach with the solution presented in this article (Table 7). The most important difference lays in the fact that GN6ASL requires no changes to the implementation of the IPv6 protocol, while in the proposed solution a small adjustment is required: the protocol has to be aware of the special meaning of FF16::/16 packets, and should hand them over to a VANET routing protocol that can process them appropriately. However, this adjustment is only required in the CCU nodes, and there is no need for the implementation of an entire sub-layer such as GN6ASL. The ETSI approach does not request any assignment or reservation of IPv6 prefixes or suffixes for special purposes. In our solution we rely on an unassigned scope value to indicate VANET broadcasting, but as defined in RFC 4291 [42] we have the freedom to do so. The GN6ASL layer performs address resolution without neighbor discovery to save wireless resources. In our solution we do perform neighbor discovery address resolution on the VANET. However, in an IPv6 ad hoc network, the next hop during multi-hop forwarding always has to be a node within communication range. This means that the neighbor discovery broadcast messages can be limited to single-hop broadcasts. This restricts the overhead on the wireless medium. On the other hand, GN6ASL has to create and maintain GVL areas, which also consumes wireless resources.

It can be concluded that GN6ASL sub-layer and the VANET solution proposed in this article both provide all the required functionality to support the transmission of IPv6 packets over the VANET. No significant differences between both approaches could be identified, except the fact that our solution does not require the implementation of an entire sub-layer between the applied networking and transport layers.

Complexity analysis

It was shown in the previous sections that the solution presented in this article meets all imposed requirements. When compared to the most relevant state-of-the-art, it was concluded that our solution is quite similar to the alternatives, but our approach is considered to be less complex. This is important, since one of the main elements in this article is the adoption of the KISS principle. However, it is hard to quantify this characteristic objectively. If the focus would have been put on the comparison of specific algorithms, then asymptotic analysis techniques could be applied. If two different implementations of a single piece of software should be compared, cyclomatic complexity would be a suitable metric. However, the pure IPv6 solution presented in this article is a quite different networking paradigm than the alternative of communicating IPv6 packets as payload in a geographic network. They cannot be compared in terms of complexity based on any of the mentioned techniques. Hence, the complexity level of our solution cannot be experimentally determined. Therefore, to illustrate that the proposed solution indeed follows the KISS principle more closely, an elaborate argumentation is presented instead.

Table 7 Differences between ETSI TS 102 636-6-1 and the approach presented in this paper

	ETSI TS 102 636-6-1	Vandenberghe et al.
Required implementation	Entire sub-layer	Small adjustment to IPv6 stack
Changes to the IPv6 implementation	None	CCU: handover of FF16::/16 packets to VANET routing protocol
Reservation IPv6 prefix	None	Use of unassigned scope value 0×6 (allowed)
Management overhead on wireless medium	Creation and maintenance of GVL	Neighbor discovery address resolution (single hop)

Implementations of the IPv6 networking stack are already available for many years. When it is assumed that the implementation of the geographic networking stack still has to be commenced by manufacturers of CCU nodes, our claim of reduced complexity is supported by the fact that our approach only needs a small adjustment to the available IPv6 stack. The alternative approach on the other hand requires the implementation of an entire geographic networking stack and of an additional sub-layer to be put between the stack and the transport layer.

However, although not as widespread as the IPv6 stack, some implementations of geonetworking protocols are already available (e.g., the CAR-2-X protocol stack implemented by NEC [55]). When assuming that CCU manufacturers will build their products on top of such an existing geographical networking stack, at first sight it becomes less obvious that our solution still follows the KISS approach more closely. In this case, both approaches can start from an existing networking stack. In the solution presented in this article, an adjustment has to be made to the IPv6 stack of the CCU, while no changes are required to the available geographical networking stack in the alternative approach. On the other hand, the implementation of the GN6ASL sub-layer is required in the case of tunneling, which is not the case in our proposed solution. Based on this information, a clear distinction in terms of complexity cannot be made. However, when taking a closer look at the practical implications of the tunneling approach, it becomes clear that our approach is less complex than the tunneling approach.

As mentioned in the "Design requirements" section, no modifications may be required in the application units. This means that geographical networking will not be deployed on the AUs, and that except in the VANET itself, IPv6 will be applied (e.g., intra-station LAN and Internet backbone). Hence, in the case of geographical anycast or broadcast, some mechanism is required to allow these units and their applications to indicate to which geographical area their messages are targeted. The fundamental idea of our solution is that if you already provide such a mechanism, the KISS approach is to extend it to the VANET instead of translating it to another networking stack.

In our approach, a CCU can be implemented starting from the IPv6 networking stack in which only one adjustment has to be made: in case of forwarding packets in the FF16::/16 domain, packets have to be handed over to the geographic routing protocol instead of being processed by the standard IPv6 stack. This adjustment requires practically no implementation effort. In the case of the Click Modular Router platform that was used (see "Implementation details" section) this is only a matter of inserting one Classifier element which hands these packets over to the geographical routing protocols. In case a kernel space IPv6 stack in combination with user space geographic routing protocols (similar to CarGeo6), this is only a matter of connecting these protocols to a virtual interface and adding an entry to the routing table of the IPv6 router. This entry ensures that all FF16::/ traffic is forwarded to this virtual interface.

When implementing the tunneling approach, the required effort and complexity of the solution increases. At the starting point of the tunnel, some service is required that performs the translation between the IPv6 geographical annotation and the geonet destination. An example is the mechanism to configure GVLs as defined in ETSI TS 102 636-6-1, or the Geo-destination module of the CarGeo6 implementation presented by Toukabri et al. [33]. The same publication also mentions that an IP Next Hop cache should be implemented, since in case of unicast traffic resolving the next IPv6 hop over the C2CNet leads to end-to-end performance degradation. As mentioned in the "Automatic address assignment " section, such an additional element does not need to be implemented in our proposed solution. Since we rely on native IPv6 ad hoc protocols for unicast traffic, the next hop for a given destination will always be a node within transmission domain. Hence, we can rely on the standard IPv6 address resolution functionality. This will broadcast a single-hop neighbor solicitation message once on the VANET interface, when the first packet arrives for which the MAC address of the corresponding IP address is not yet known. A third element mentioned by Toukabri et al. is the fact that the Location Service mechanisms implemented at the C2CNet level causes high round-trip time and packet loss values in case of multi-hop tunneling of unicast traffic. It is suggested to implement a new multi-hop beaconing mechanism to counter this problem. Since our proposed solution does not require any Location Service mechanisms, it does also not require the implementation of such novel beaconing mechanisms.

In the above we compared the case where a geonetworking implementation is already available on the CCUs (and IPv6 tunneling has to be added) with the case where the standard IPv6 stack is available on the CCU (but should be modified to support the approach presented in this article). In both cases a mechanism has to be provided that allow IPv6 based applications to indicate the geographical area to which their multicast packets are addressed. In both cases, ad hoc routing and geo-broadcast protocols have to be provided on top of the used networking stacks to define the appropriate forwarding actions. Besides these common functionalities, the native IPv6 approach requires only the insertion of a filter in the standard IPv6 stack. However, in the case of tunneling the IPv6 packets as geonetworking data, this requires the implementation of a service to translate the IPv6 geographical

annotation into the appropriate geonet destination, of an IP Next Hop Cache and of a new multi-hop beaconing mechanism to counter the introduced performance issues. Based on these observations, we conclude that the approach presented in this article more closely follows the KISS principle than the tunneling approach, even in the case that an existing geographical networking stack can be used as the starting point of the implementation.

Conclusions

In this article, an approach to VANET networking was presented that incorporates geographical data in the standard IPv6 header. Starting from an overview of possible networking techniques and the requirements imposed by the applications, it was shown that in a simple yet effective manner, a VANET networking solution can be constructed that is entirely based on IPv6 and supports all required communication forms and design requisites.

The strength of our approach is that it is based on a simpler design compared to the VANET approach which combines IPv6 and geographic networking solutions (as for instance adopted by ETSI). First of all there is no need to provide a non-IP geographic networking stack. Such a stack does not only define specific addressing mechanisms based on node location, it also requires additional functionality such as a position service. Only a few implementations of such networking stacks exist. Instead, our solution relies on the standard IPv6 stack, which is widely available for years now. Our solution also does not require additional protocol translation mechanisms for tunneling, nor required performance optimizations as identified during practical implementations of the combined approach. Such a reduction in complexity makes it easier to implement, debug en maintain (future) VANET networking stacks. The presented packet header is also more straightforward than in the corresponding ETSI standard. This leads not only to a reduction in processing complexity at the network layer, but also to a performance improvement in terms of PSR and latency for multi-hop broadcast messages.

The downside of our solution is that the combined approach has already gained quite a lot of momentum. Significant implementation efforts were made in the GeoNet project. Since then, these results have been standardized by ETSI and are made publicly available by the CarGeo6 open-source implementation. The solution presented in this article on the other hand has only been taken up by ourselves until now. Applying this work would require stepping back to a clean-slate implementation of the VANET networking stack. Although the straightforward design of our solution results in a manageable implementation effort, it seems unfeasible to expect that all extensive VANET standardization and implementation efforts of the past will be entirely abandoned. However, the goal of this publication is to introduce the concepts that allowed us to provide all required functionality while rigorously striving to follow the KISS principle. Ideally, this would inspire the VANET community to evaluate these concepts critically, and possible apply them in future iterations of VANET standards and mainstream implementations.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Information Technology (INTEC), Ghent University, IBBT, Ghent, Belgium. [2]Department of Mathematics and Computer Science, University of Antwerp, IBBT, Antwerp, Belgium.

Received: 19 December 2011 Accepted: 14 August 2012
Published: 19 October 2012

References

1. ES Raymond, *The Art of Unix programming*, 1st edn. (Addison-Wesley Professional, Boston, 2003)
2. B Carpenter (ed.), Architectural principles of the Internet, IETF RFC, 1958, 1996
3. W Vandenberghe, D Carels, I Moerman, P Demeester, E Van de Velde, J Bergs, C Blondia, in *Proceedings of the 10th international conference on ITS telecommunications (ITST),* VANET addressing scheme incorporating geographical information in standard IPV6 header (Kyoto, Japan, 2010), pp. 1–6
4. ETSI ITS Working Group 3, Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media independent functionalities, ETSI TS 102 636-4-1, 2011
5. ETSI ITS Working Group 3, Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols, ETSI TS 102 636-6-1, 2011
6. J Choi, Y Khaled, M Tsukada, T Ernst, in *Proceedings of the 8th international conference on ITS telecommunications (ITST),* IPv6 support for VANET with geographical routing (Phuket, Thailand, 2008), pp. 222–227
7. T Ernst, Why IPv6 Geonetworking is needed for ITS? Presentation given at GeoNet final workshop. Paris, 2010. http://www.geonet-project.eu/?download=GeoNet-IPv6_for_ITS.pdf. Accessed 1 August 2011
8. G Huston, IPv4 Address Report. http://www.potaroo.net/tools/ipv4/index.html. Accessed 2 August 2011
9. V Namboodiri, L Gao, Prediction-based routing for vehicular ad hoc networks. IEEE Trans. Veh. Technol. **56**(4), 2332–2345 (2007)
10. T Taleb, E Sakhaee, A Jamalipour, K Hashimoto, N Kato, Y Nemoto, A stable routing protocol to support ITS services in VANET networks. IEEE Trans. Veh. Technol. **56**(6), 3337–3347 (2007)
11. Y-B Ko, NH Vaidya, Location-aided routing (LAR) in mobile ad hoc networks. Wirel. Netw. **6**(4), 307–321 (2000)
12. W Wang, F Xie, M Chatterjee, Small-scale and large-scale routing in vehicular ad hoc networks. IEEE Trans. Veh. Technol. **58**(9), 5200–5213 (2009)
13. H Menouar, M Lenardi, F Filali, in *Proceedings of the 7th International Conference on ITS Telecommunications (ITST),* Improving proactive routing in VANETs with the MOPR movement prediction framework (Sophia Antipolis, France, 2007), pp. 1–6
14. E Baccelli, Schiller J, in *Proceedings of the 8th International Conference on ITS Telecommunications (ITST),* Towards scalable MANETs (Phuket, Thailand, 2008), pp. 133–138
15. S Giannuolis, C Katsanos, S Koubias, G Papadopoulos, in *Proceedings of the IEEE International Workshop on Factory Communication Systems (WCFS),* A hybrid adaptive routing protocol for ad hoc wireless networks (Vienna, Austria, 2004), pp. 287–290
16. V Ramasubramanian, ZJ Haas, EG Sirer, in *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing*

(MobiHoc), SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks (Annapolis, USA, 2003), pp. 303–314
17. IEEE Vehicular Technology Society, IEEE standard for wireless access in vehicular environments (WAVE)—networking services, IEEE Std 1609.3-2010, 2010
18. International Organization for Standardization, Intelligent Transport Systems—Communications access for land mobiles (CALM)—Non-IP networking, ISO DIS 29281, 2009
19. N Mariyasagayam, H Menouar, M Lenardi, in *Proceedings of the Vehicular Networking Conference (VNC),* An adaptive forwarding mechanism for data dissemination in vehicular networks (Tokyo, Japan, 2009), pp. 1–5
20. A Laouiti, P Mühlethaler, Y Toor, in *Proceedings of the 9th International Conference on ITS Telecommunications (ITST),* Reliable opportunistic broadcast VANETs (R-OB-VAN) (Lille, France, 2009), pp. 382–387
21. M Torrent-Moreno, in *Proceedings of the 4th Annual IEEE/IFIP Conference on Wireless On Demand Network Systems and Services (WONS),* Inter-vehicle communications: assessing information dissemination under safety constraints (Obergurgl, Austria, 2007), pp. 59–64
22. S Busanelli, G Ferrari, S Panichpapiboon, in *Proceedings of the Global Communications Conference (Globecom),* Efficient broadcasting in IEEE 802.11 networks through irresponsible forwarding (Hawaii, USA, 2009), pp. 1–6
23. K Ibrahim, MC Weigle, M Abuelela, in *Proceedings of the IEEE 69th Vehicular Technology Conference (VTC),* p-IVG: probabilistic inter-vehicle geocast for dense vehicular networks (Barcelona, Spain, 2009), pp. 1–5
24. OK Tonguz, N Wisitpongphan, F Bai, DV-CAST: a distributed vehicular broadcast protocol for vehicular ad hoc networks. IEEE Wirel. Commun. **17**(2), 47–56 (2010)
25. SA Rao, M Pai, M Boussedjra, J Mouzna, in *Proceedings of the 8th International Conference on ITS Telecommunications (ITST),* GPSR-L: greedy perimeter stateless routing with lifetime for VANETs (Phuket, Thailand, 2008), pp. 299–304
26. M Zhang, RS Wolff, Routing protocols for vehicular ad hoc networks in rural areas. IEEE Commun. Mag. **46**(11), 126–131 (2008)
27. A Festag, R Baldessari, H Wang, in *Proceedings of the 5th International Workshop in Intelligent Transportation (WIT),* On power-aware greedy forwarding in highway scenarios (Hamburg, Germany, 2007), pp. 1–6
28. G Korkmaz, E Eikici, F Ozgüner, Black-burst-based multihop broadcast protocols for vehicular networks. IEEE Trans. Veh. Technol. **56**(5), 3159–3167 (2007)
29. J Zhao, G Cao, VADD: vehicle-assited data delivery in vehicular ad hoc networks. IEEE Trans. Veh. Technol. **57**(3), 1910–1922 (2008)
30. Y Ding, L Xiao, SADV: static-node-assisted adaptive data dissemination in vehicular networks. IEEE Trans. Veh. Technol. **59**(5), 2445–2455 (2010)
31. A Kovacs (ed.), D2.2 Final GeoNet Specification. Deliverable of the project GeoNet STREP No. 216269, 2010, http://www.geonet-project.eu. Accessed 5 August 2011
32. IB Jemaa, M Tsukada, H Menouar, T Ernst, in *Proceedings of the 10th International Conference on ITS Telecommunications (ITST),* Validation and evaluation of, NEMO in VANET using geographic routing (Kyoto, Japan, 2010), pp. 1–6
33. T Toukabri, M Tsukada, T Ernst, L Bettaieb, in *Proceedings of the 11th International Conference on ITS Telecommunications (ITST),* Experimental evaluation of an open source implementation of, IPv6 GeoNetworking in VANETs (Saint-Petersburg, Russia, 2011), pp. 237–245
34. CAR-2-CAR Communication Consortium, CAR 2 CAR Communication Consortium Manifesto (2007)
35. ETSI ITS Working Group 1, Intelligent Transport Systems; Vehicular Communications; Basic set of applications; Definitions, ETSI TR 102 638, 2009
36. ETSI ITS Working Group 1, Intelligent Transport Systems; Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, ETSI TS 102 637-2, 2010
37. ETSI ITS Working Group 1, Intelligent Transport Systems; Vehicular Communications; Basic Set of Applications; Part 2: Specification of Decentralized Environment Notification Basic Service, ETSI TS 102 637-3, 2010
38. M Bechler (ed.), European ITS Communication Architecture; Overal Framework; Proof of Concept Implementation. Deliverable DEL31 of the COMeSafety project, 2010, http://www.comesafety.org. Accessed 4 July 2011
39. A Gordillo, M Calderon, CJ Bernardos, in *Proceedings of the 9th international conference on ITS telecommunications (ITST),* Enabling, IP geomulticast services for vehicular networks (Lille, France, 2009), pp. 292–297
40. S Thomson, T Narten, T Jinmei, IPv6 stateless address autoconfiguration, IETF RFC 4862, 2007
41. InternetEngineeringSteeringGroup Internet Architecture Board, IAB/IESG recommendations on IPv6 address allocations to sites, IETF RFC 3177, 2001
42. R Hinden, S Deering, IP Version 6 Addressing Architecture, IETF RFC 4291, 2006
43. Y Khaled, M Tsukada, T Ernst, in *Proceedings of the 9th international conference on ITS telecommunications (ITST),* Geographical information extension for, IPv6: application to VANET (Lille, France, 2009), pp. 304–308
44. Y Khaled, IB Jemaa, M Tsukada, T Ernst, in *Proceedings of the 9th international conference on ITS telecommunications (ITST),* Application of, IPv6 multicast to VANET (Lille, France, 2009), pp. 198–202
45. J-S Park, M-K Shin, H-J Kim, A method for generating link-scoped IPv6 multicast addresses, IETF RFC 4489, 2006
46. E Kohler, The Click modular router, PhD thesis, Massachusetts Institute of Technology, 2000
47. S Murthy, J Garcia-Luna-Aceves, An efficient routing protocol for wireless networks. Mob. Netw. Appl. **1**(2), 183–197 (1996)
48. IBBT NextGenITS Demo—Cooperative Systems, http://www.youtube.com/watch?v=cSP9xlTDY3o&fmt=22
49. W Vandenberghe, I Moerman, P Demeester, H Cappelle, in *Proceedings of the 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT),* Suitability of the wireless testbed w-iLab.t for VANET research (Gent, Belgium, 2011), pp. 1–6
50. M-D Cano, F Cerdan, Subjective QoE analysis of VoIP applications in a wireless campus environment. Telecommun. Syst. **49**(1), 5–15 (2012)
51. S Eichler, in *Proceedings of the 66th IEEE Conference on Vehicular Technology (VTC-2007),* Performance evaluation of the IEEE 802.11p WAVE communication standard (Baltimore, USA, 2007), pp. 2199–2203
52. N Wisitpongphan, OK Tonguz, JS Parikh, P Mudalige, F Bai, V Sadekar, Broadcast storm mitigation techniques in vehicular ad hoc networks. IEEE Wirel. Commun. **14**(6), 84–94 (2007)
53. Q Chen, F Schmidt-Eisenlohr, D Jiang, M Torrent-Moreno, L Delgrossi, H Hartenstein, in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM),* Overhaul of IEEE 802.11 modeling and simulation in NS-2 (Chania, Greece, 2007), pp. 159–168
54. X Ma, X Chen, Performance analysis of IEEE 802.11 broadcast scheme in Ad Hoc Wireless LANs. IEEE Trans. Veh. Technol. **57**(6), 3757–3768 (2008)
55. A Festag, R Baldessari, W Zhang, L Le, in *Proceedings of the IEEE Vehicular Networking and Applications Workshop - Future Wireless Technologies for Vehicle Infrastructure Integration (VII) Applications (VehiMobil 2009),* CAR-2-X communication SDK—a software toolkit for rapid application development and experimentations (Dresden, Germany, 2009), pp. 1–5

doi:10.1186/1687-1499-2012-316
Cite this article as: Vandenberghe *et al.*: **Vehicular ad hoc networking based on the incorporation of geographical information in the IPv6 header.** *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:316.