Mobile Information Systems 8 (2012) 153–172
DOI 10.3233/MIS-2012-0136
IOS Press

Cooperation as a service in VANET: Implementation and simulation results

Hajar Mousannif[a,*], Ismail Khalil[b] and Stephan Olariu[c]
[a]*Cadi Ayyad University, Guéliz, Marrakech, Morocco*
[b]*Johannes Kepler University, Linz, Austria*
[c]*Old Dominion University, Norfolk, VA, USA*

Abstract. The past decade has witnessed the emergence of Vehicular Ad-hoc Networks (VANET), specializing from the well-known Mobile Ad Hoc Networks (MANET) to Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) wireless communications. While the original motivation for Vehicular Networks was to promote traffic safety, recently it has become increasingly obvious that Vehicular Networks open new vistas for Internet access, providing weather or road condition, parking availability, distributed gaming, and advertisement. In previous papers [27,28], we introduced Cooperation as a Service (CaaS); a new service-oriented solution which enables improved and new services for the road users and an optimized use of the road network through vehicle's cooperation and vehicle-to-vehicle communications. The current paper is an extension of the first ones; it describes an improved version of CaaS and provides its full implementation details and simulation results. CaaS structures the network into clusters, and uses Content Based Routing (CBR) for intra-cluster communications and DTN (Delay – and disruption-Tolerant Network) routing for inter-cluster communications. To show the feasibility of our approach, we implemented and tested CaaS using Opnet modeler software package. Simulation results prove the correctness of our protocol and indicate that CaaS achieves higher performance as compared to an Epidemic approach.

Keywords: Intelligent transportation systems, VANET, publish/subscribe, cluster, CBR, DTN routing

1. Introduction

Over the last few decades, the need for better transportation systems has grown significantly. The number of vehicles on the road has approached critical mass [32], forcing government transportation departments across more and more countries to develop Intelligent Transportation Systems [4], which refer to broad range of diverse technologies, including information processing, sensors, communications, control, and electronics. Combining these technologies in innovative ways and integrating them into the transportation system will save lives, time and resources by simplifying data exchange between roadside infrastructure and vehicles. Some of this data is collected to support real-time traveler information and traffic control, whereas other data is collected and used off-line to help characterize typical travel patterns and project future traffic conditions [10].

Recently, the emphasis in the area of Intelligent Transportation Systems has turned to cooperative systems in which the vehicles communicate with each other and/or with the infrastructure. Such cooperative systems can greatly increase the quality and reliability of information available about the

[*] Corresponding author: Cadi Ayyad University, FSTG, B.P 549, Av. Abdelkarim Elkhattabi, Guéliz, Marrakech, Morocco. Tel.: +212661471177; E-mail: hajar.mousannif@gmail.com.

vehicles, their location and the road environment. They enable improved and new services for the road users, which, in turn, will lead to greater transport efficiency, by making better use of the capacity of the available infrastructure and by managing varying demands, and increased safety, by improving the quality and reliability of information and allowing the implementation of advanced safety applications.

These cooperative systems make a combined use of both Vehicular Sensor Networks (VSN) and Vehicular Ad-hoc Networks (VANET). Unlike a traditional wireless sensor network in which optimizing energy consumption is the main challenge [26], vehicles in a vehicular sensor network are typically not affected by limitations in power, computational capacities or memory. In fact, vehicles can be easily equipped with powerful processing and storage units, multiple wireless interfaces (e.g. Wifi, Bluetooth and 2G/3G), Event Data Recorders (EDRs) [12], and sensing devices of some complexity (e.g. GPS receivers, cameras, vibration sensors and acoustic sensors). Depending on its equipment, a car 'knows' about its speed (tachometer), the actual location (GPS), distance to the next car (distance control), destination of the actual trip (navigation system) and the actual weather conditions (rain and temperature sensor) [20].

Many car manufacturers have also been installing wireless connectivity equipment in their vehicles to enable communications with roadside base stations and also between vehicles, for the purposes of safety, driving assistance, and entertainment. The two primary distinct features of vehicle networks are that: 1°) Vehicles can be highly mobile, with speed up to 30 m/s and 2°) Their mobility patterns are more predictable than those of nodes in Mobile Ad hoc Networks (MANET), extensively studied in the literature (see, for example [38], and the references therein), due to the constraints imposed by roads, speed limits, and commuting habits. Therefore, these networks require specific tradeoffs and identify a novel research area, i.e., Vehicular Ad hoc Networks (VANET).

VANETs support two types of communication: Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). While V2V deals with communication among vehicles themselves, V2I is concerned about transmitting information between a vehicle and the fixed infrastructure installed along the road, e.g. roadside base stations [11] that can be connected with each other or, depending on the deployment scenarios, can also be connected to the Internet. V2V and V2I communications are made possible via the DSRC/WAVE (Dedicated Short Range Communications/Wireless Access in a Vehicular environment) standard [19] which is a short to medium range communication technology operating in the 5.9 GHz. Readers can find a detailed overview of the DSRC standards in [19].

The potential applications of vehicular networks offer vast opportunity [14]. While the past decade has witnessed a proliferation of mainly vehicular safety applications [3], such as Electronic Brake Warning (EBW), Vehicle Stability Warning (VSW), application On-coming Traffic Warning (OTW) and Lane Change Warning (LCW), many other innovative applications can be achieved by combining high accuracy positioning, inter-vehicular communication technologies and the on-board array of sensors. Vehicle infotainment system (VIS) [17], as an example, has gained much attention recently due to its promising usage in a wide range of Internet-based services, ranging from location-aware services such those in [7], on-demand traveling information and traffic conditions, to rich media news and video distribution [13]. Vehicles can also act like mobile sensors monitoring parameters such as road and weather conditions, parking lots availability (like in [1]) or traffic density (like in [8,29]). Such information can be shared among vehicles in order to perform route optimization or adaptations of driving behavior. Internet access was one of the earliest applications proposed for VANET. The idea is to allow drivers to share their underutilized network resource with other drivers who may need to access the Internet while on the move. Free Internet access can also be combined with some form of advertisement distribution like in [21], where cars carry and distribute advertisement using mainly single-hop inter-vehicle communication. The same idea is found in [31], where the authors propose AdTorrent;

an extension of the physical billboards that allows drivers to download advertisements of interest using a location-sensitive distributed mechanism. Other applications such as RoadSpeak [39], which allows drivers to communicate on the road via voice chat messages, interactive online games [35], and video applications (video-phoning and teleconferencing) [36] might not be a driving force for VANET in the immediate future.

Motivated by finding solutions to problems such as the lost of worker productivity and fuel, and the high level of CO2 emissions due to traffic congestion, the increase of the number of fatalities directly attributable to traffic-related incidents and the huge cost related to Intelligent Transportation Systems, we described in previous papers [27,28], a new service-oriented solution for VANETs, referred to as Cooperation as a Service or CaaS, that extends the two novel types of Vehicular Cloud services: Network as a Service (NaaS) and Storage as a Service (SaaS) introduced in [33]. CaaS allows providing vehicles/drivers, which are willing to cooperate, with some sets of services using very minimal infrastructure, by taking advantage of Vehicle-to-Vehicle (V2V) communications. CaaS enables the integration of a set of improved services for the road users through a novel hybrid publish/subscribe mechanism we introduced. In [27], the proposed mechanism structures the network into clusters, and uses Content Based Routing (CBR) for intra-cluster communications and geographic routing for inter-cluster communications.

The main limitation of this previous version is the use of geographic routing for inter-cluster communications which assumes that nodes are capable of determining their own position either through GPS devices (such as Navigator Systems) mounted on the vehicles, or by deploying virtual coordinates [2]. Since our main objective is to provide all vehicles with services they subscribed to, regardless of whether they are equipped with a navigator device or not, it would be unfair to consider that all vehicles in the network need to be equipped with such devices to benefit from a certain service. In this paper, we overcome the limitations of the previous version by using DTN routing [18] for inter-cluster communications instead of geographic routing. Our choice will be justified throughout this paper.

The remainder of this paper is organized as follows: Section 2 reviews some related work in the area of data dissemination in VANETs and highlights our contribution. Section 3 describes the improved version of our cooperative solution for VANET. Implementation details and simulation results are provided in Section 4. Finally, Section 6 offers concluding remarks and directions for future work.

2. Data dissemination in VANETs

Before a VANET-based application can start to process and propagate data, local measurements need to be made. These local observations are application-dependant and are usually obtained through the car integrated sensors. Reading information from these sensors alone can indeed provide important information. As an example, reading the speedometer of a vehicle may allow conclusions to be drawn about the traffic conditions. But, integrating and combining information from different sources will make such conclusions even more accurate. Authors in [34] examine a system that aims at integrating measurements from multiple sources through sensor fusion techniques in order to provide useful information about the current road condition.

After obtaining local measurements, information has to be disseminated to interested parties. Nevertheless, due to the capacity constraints [15] in VANET, it is technically unfeasible to deliver detailed and regularly updated information to all participants in the network. The key idea is to combine information from a cooperative VANET using measurement summarizing and aggregation mechanisms which aim at reducing the generated amount of data. Further details about data fusion techniques can be found in [30]. It is also important to determine whether an event (e.g., a traffic congestion) or a resource (e.g., an available parking space) is relevant to a vehicle. Authors [6] in propose a general data management architecture for vehicular networks that allows to select, among the events received, the events that are relevant to the vehicle and so may be also relevant to the driver using geographic vectors and maps.

Depending on the application, information needs also to be shared among vehicles that are interested in it. These vehicles might need to adapt their behavior based on the received information. One way of distributing this information inside the network is to use flooding which simply consists of rebroadcasting the information by each node which receives it. Since this naïve approach may lead to severe congestions in the network, many approaches have been proposed to deal with this problem and mainly aim at influencing the forwarding behavior of vehicles by either adapting the time to forward, the geographic area where to forward [24] or by simply placing rules on whether a vehicle should forward or not. Readers can find details on the use of these flooding techniques in [37,41].

Despite the use of these flooding techniques and due to wireless signal dynamics, node mobility and vehicular networks density, especially in big cities, poor performance of flooding-based routing protocols has been noticed [43]. As an alternative, geographic routing has been chosen in many routing algorithms used for VANET. Geographic-based routing protocols exploit both local information and information about the surrounding road topology to route packets. In some scenarios, information about speed, direction or route plan can also be used. Details about the most used geographic-based routing protocols in VANET can be found in [22,23,45]. The most important assumption that almost every geographic routing protocol makes is that nodes are capable of determining their own position. This can be done either by equipping vehicles with GPS devices; a navigation system available on more and more vehicles nowadays can provide location information, or by deploying virtual coordinates, such as in [2], which consists in assigning some elected nodes in the network coordinates and letting the rest of the nodes obtain their virtual coordinates either through triangulation techniques or by averaging the coordinates of their neighbors.

In many application scenarios, another class of routing paradigms known as Content-Based Routing (CBR) is used to achieve better performances. In CBR, the sender simply injects the message in the network, which then determines how to route it according to the nodes' interests (or subscriptions). CBR is proposed as an efficient publish/subscribe approach in many Service-Oriented Architectures (SOAs) [5].

The fact that vehicular networks are highly mobile and sometimes sparse complicates finding an end-to-end connection to disseminate such data efficiently. DTN (Delay – and disruption-Tolerant Network) routing [18] is based on the principle of store-and-forward; that is, a message can be buffered at a node until an appropriate next hop appears. Then, the node forwards the message to the next such hop. Therefore, instead of waiting for a path to the destination, messages can be forwarded to intermediate nodes, which in turn would buffer these packets for some period of time and then forward them to other nodes. This process can be continued until some intermediate node eventually comes in contact with the destination node and delivers the message to it. Let us take, as an example, an alert application that propagates emergency messages when accidents occur. Just after the accident the application forwards a message that is rapidly propagated in order for the upcoming vehicles to brake and stop. There will be no need for the last vehicle which received the message to send it if there are no more neighbors to send packet to. The intelligent decision is to buffer the packet and forward it as soon as a new approaching vehicle is detected which will save another accident

In our work, we aim at integrating as many VANET-based services as possible and allowing the driver to select those he/she is interested in. Cooperation among vehicles is the key point in our framework.

This collaboration is illustrated through the novel publish/subscribe mechanism we propose for VANET; participants can act as publishers who generate information (either local or collaborative measurements) and subscribers are drivers who express their interests in a set of services and who are willing to cooperate to provide other subscribers with the information they are interested in.

Our work deals with three major challenges in VANET:

- we suppose a partially structured vehicular network;
- not all vehicles would be interested in collaborating, so they should not be affected;
- we deal with network fragmentation and the resultant lack of continuous end-to-end connectivity at any given instant.

The proposed solution will enable improved driving conditions and an optimized use of the road network. In fact, CaaS will allow:

- A decrease in the cost related to Intelligent Transportation Systems, since our solution performs well even when no infrastructure is available.
- A reduction in the severity of road congestions and CO2 pollution, at the same time, time and fuel saving. Vehicles act like mobile sensors monitoring parameters such as road and weather conditions, or traffic density. Our solution allows such information to be shared among vehicles in order to perform route optimization or adaptations of driving behavior.
- Improving driver's safety and preventing rear-ending accidents. Our solution allows vehicles to alert the surrounding cars of its braking manoeuvres for example.
- A more enjoyable driving experience (Internet access, interactive online games...).

3. CaaS: Cooperation as a service

As stated earlier, our framework deploys a publish/subscribe interaction scheme. With publish/subscribe models; participants can act as subscribers who express their interest in an event, or a pattern of events, and publishers who submit information regarding those events to the system. Readers can find a detailed survey on this communication paradigm in [9]. Such a scheme is well adapted to the loosely coupled nature of distributed interaction in large-scale networks, VANET for instance, mainly because of its decoupling properties. In Fact, publish/subscribe-based schemes achieve at least two dimensions of decoupling: 1°) Space decoupling: subscribers are interested in getting the information they want regardless of who published or how this information is published in the network and 2°) Time decoupling: publishers and subscribers do not need to interact at the same time. Our designed publish/subscribe mechanism for VANET insures both of them.

In this section, we will be discussing our proposed publish/subscribe interaction scheme from the algorithmic, the functional, and the architectural perspectives. But before doing so, we will start first by arguing our choices regarding the underlying routing protocols we use in our approach.

3.1. Discussion

A traditional publish/subscribe system model relies on an event notification service (or broker) that stores and manages subscriptions, thus, acting as a mediator between publishers and subscribers. In VANET, we cannot expect any dedicated server (or service) that will play such a role. Nodes themselves should act as mediators as well as publishers and subscribers. This makes designing a scalable publish/subscribe scheme well suited to VANET environments extremely challenging.

Two main approaches have been proposed to support the publish/subscribe paradigm in MANET in general and in VANET in particular: a structured-based approach (like [44]) and a gossip-based approach (i.e. BubbleStorm [40]). The former requires the nodes to be organized into a sort of overlay structure and builds the publish/subscribe methods on top of it. The latter uses gossiping for information exchange which supposes that a query or publication will sufficiently populate a large portion of the network so that their paths intersect at some rendezvous nodes with high probability. Although, the first approach suffers from the disadvantage of introducing an additional overhead for structure construction and maintenance, it achieves better efficiency in comparison to the second which introduces additional computation and storage costs due to the intermittent connectivity issues in VANET. Since we want to guarantee a high level of service delivery to the participants in the network without affecting non-interested parties, we favor the structure-based approach over the gossip-based one.

In most structure-based approaches proposed for VANET, Content based data dissemination, where information is routed based on the content rather than the destination address, is proposed as an efficient publish/subscribe scheme. But again CBR requires a one-tree structure network topology [25], almost impossible to maintain when the size of the network increases. The high mobility of nodes in the network may make this tree maintenance issue even worse since a tree may become partitioned into a number of trees leading to serious issues finding a path to merge those trees.

To take advantage of the benefits of CBR and reduce its disadvantages, we decide to allow more than one tree structure in the network. Each tree will represent a cluster whose size and depth are appropriately chosen to allow a proper maintenance of the trees. In our structure, we use CBR for intra-cluster communications; subscriptions of all members of the cluster are forwarded to a cluster-head and updated regularly to deal with the continuous movement of the nodes. This is done using the same algorithm we already presented in [27,28] and which we will be reviewing in the next section.

For inter-cluster communication, many possible solutions can be considered depending on whether infrastructure exists or not. If roadside stations are available, clusters will be interconnected using the infrastructure as depicted in Fig. 1. However, since a widespread presence of roadside stations cannot be guaranteed at any time and place, two options can be considered: The first is to use a flooding-based approach (i.e. Document flooding (DF) [44]) to exchange subscription summarizations and publications among clusterheads in the network. The second is to take advantage of the performance efficiency achieved by deploying DTN routing [18] to disseminate subscriptions and publications between clusters. In our structure, we decide to use DTN routing for inter-cluster communication and let cluster-heads which have heard a publication buffer the packet and forward it as soon as a new approaching cluster-head vehicle is detected or a roadside station is in its vicinity.

In summary, our approach suggests a hybrid publish/subscribe scheme for VANET where CBR is used for intra-cluster communication and DTN routing for inter-cluster communication. Figure 2 summarizes our proposed network structure.

3.2. Tree construction

As we mentioned earlier, we allow more than one tree structure to exist in the network. Each tree represents a cluster where the root of the tree (i.e. clusterhead) should maintain an up-to-date subscription summary of all members in its cluster. The clusterhead should also be able to route publications to interested vehicles using CBR. Since a node might join or leave the cluster at any time, the tree should also be able to maintain itself and, depending on the proximity of nodes, merge itself with another tree. Our intra-cluster structure considers the following roles to set up a routing infrastructure:

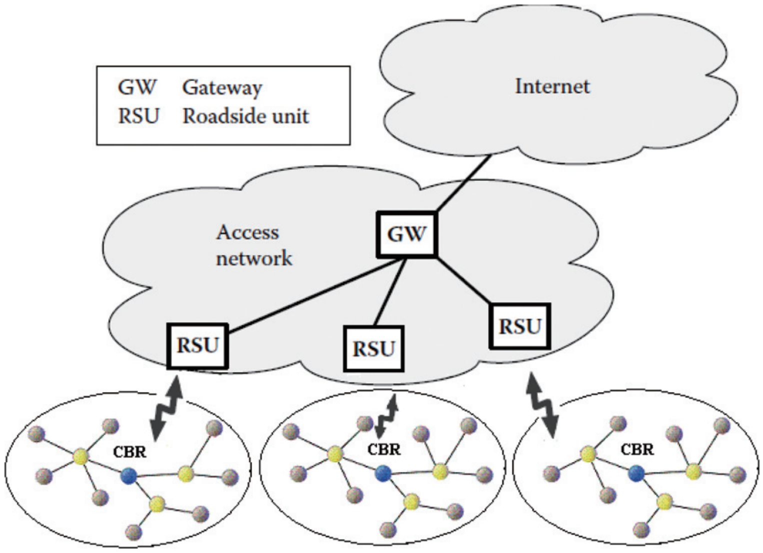


Fig. 1. Infrastructure-based vehicular network.

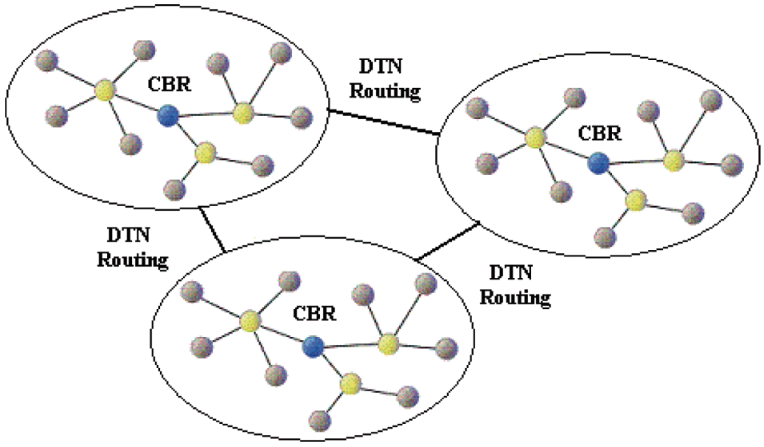


Fig. 2. Our proposed network structure for infrastructure-free vehicular network.

- Clusterhead: node responsible for summarizing subscriptions of the cluster members and forwarding them to other clusters. It is also responsible for delivering publications to interested nodes inside the cluster.
- Broker: node acting mainly as a relay. Each broker holds a subscription table used to determine how to disseminate subscriptions/publications along the tree.
- Subscriber: a node which expresses its interest in a service (or a set of services).
- Publisher: any entity in the network that publishes information about services in which vehicles might be interested.

To form a cluster, each node needs to know its neighbors' interests (i.e. subscriptions). Like other routing protocols for VANET, we consider that each node broadcasts a Hello message each Hello_Interval seconds to announce its existence to neighboring nodes. When a node is no longer receiving those

Table 1
Acronyms and definitions

NodeID$_i$	A unique identifier of node i, it could be a MAC address
ClusterID$_i$	The id of the root node of the tree to which node i belongs
Height$_i$	The maximum number of nodes separating a leaf from the root in the tree to which node i belongs. This value is the same for all members of a cluster, but could vary in time if a node joins or leaves the cluster
Max_Height	The maximum height allowed for a tree. This value is fixed to allow a proper maintenance of the trees
Level$_i$	The number of nodes separating node i from the root. The immediate neighbors of the root are on the first level
ParentID$_i$	The identifier of the neighbor parent of node i
Child_list$_i$	Identifiers of the neighboring childs of node i
CH_Flag$_i$	A flag indicating whether node i is a clusterhead or not.
Subscription Table	A list of pairs (NodeID, subscription list) maintained by each node

messages from a neighbor, it means that this neighbor disappeared or moved out of the transmission range of the current node. Hello messages will also serve to update cluster members and nodes' subscriptions. To achieve this, each Hello message should contain some necessary information to maintain the tree such as the node Id, the cluster Id, the height of the tree, the level of the node and the subscription table (or a part of it, in case a change in the nodes' subscriptions occurs) as described in Table 1. However, depending on whether the node is already a member of a cluster or not, the format of these messages might vary.

Initially, when a node enters the system, it tries to join an existing cluster by listening to the broadcasted Hello messages sent by neighbors and by considering as a parent the node whose signal is the strongest and whose level is strictly smaller than the maximum height allowed for the tree. A child cannot have two parents because the tree should not contain any cycles. If no such node exists, the node considers itself to be a clusterhead.

Each node broadcasts a hello message with the information listed above along with its interests if there are any. Upon receiving a hello message, a neighboring node will update its local information (ClusterID, level, Child_list... etc.). Moreover, each node should maintain heartbeats of its neighboring nodes (i.e. parent and children): After receiving a Hello message from a neighbor, a timer is set to a certain value. If this timer expires without receiving any hello messages from this neighbor, this neighbor will be considered unreachable and is removed from the list of neighbors maintained by the node. Particularly, if a node loses its parent, it should initialize its local parameters to the default ones (Fig. 3, initialization section) to either join another cluster or build its own tree. To avoid cycles, we also consider that only roots of the trees can issue the merging process. Figure 3 shows our proposed algorithm for processing hello messages.

3.3. Subscriptions and publications dissemination

Now that the trees are constructed and properly maintained thanks to the merging and dissociation processes explained earlier, subscriptions and publications can now be forwarded along the constructed links of the tree. Our idea is to exploit hello messages to communicate the node subscriptions to other nodes in the cluster and especially to the clusterhead. Each node (broker) will maintain a subscription table containing its interests, if any, along with the neighboring nodes' interests. Each node will send its subscription table only once; unless a change in this subscription table occurs (a node inserted a new interest or changed a subscription for example). If so, only the affected lines in the table will be sent in the following hello message.

```
Input: The Hello message received, it should contain at least the following fields: $NodeID_{sender}$, $ClusterID_{sender}$, $level_{sender}$, $Height_{sender}$, $ParentID_{sender}$, $Child\_list_{sender}$
Initialization:
//initialize local information
ClusterID=$NodeID_{current}$  //Initially, a node considers itself as the root of a tree
Level=0
Height=0
ParentID=$NodeID_{current}$  //Initially, a node is a parent of itself
Child_list=none   //Initially, no children
CH_Flag=1   //Initially, a node considers itself a clusterhead
Output:
1:IF $ClusterID_{sender}$ is different from the current ClusterID THEN
2:   IF (CH_Flag=1) && ($Height_{sender}$+Height <Max_Height) THEN
3:      //Node should join the cluster of the sender (trees merging) and local information need to be updated
4:      ClusterID=$ClusterID_{sender}$
5:      ParentID=$NodeID_{sender}$
6:      Level= $level_{sender}$+1
7:      Height=Height+$Height_{sender}$
8:      CH_Flag=0 //Current node is no longer a clusterhead since it joined a tree
9:   END-IF
10: ELSE //Sender belongs to the same cluster as that of the current node
11:   IF ($ParentID_{sender}$= $NodeID_{current}$) //The sender is a child of the current node THEN
12:      //Update child list
13:      Child_list=Child_list+{$NodeID_{sender}$}
14:   END-IF
14:   IF ($NodeID_{sender}$=ParentID) //The sender is the parent of the current node THEN
15:      execute instructions 4 , 6, and 7 // Local information update
16:   END-IF
17: END-IF
```

Fig. 3. Hello messages processing algorithm.

Publications will be routed along the path set up by subscriptions to interested subscribers only. This is the same approach as CBR. Moreover, each clusterhead will have to forward publications to other clusters either using infrastructure if available or DTN routing if no roadside stations are present.

4. Implementation details and simulation results

In order to assess the actual feasibility and prove the validity of our framework, we run exhaustive simulations and report on our proposed protocol performance over several scenarios, varying from a fully infrastructure-based scenario characterized by a large presence of roadside stations to a scenario in which no infrastructure is available as this represents the more challenging case for our protocol.

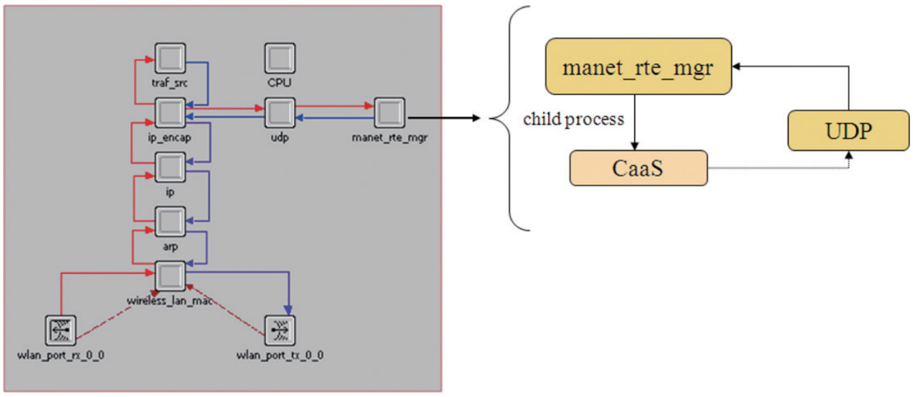


Fig. 4. CaaS model architecture in Opnet.

4.1. *Simulation environment*

As simulation platform, we used Opnet simulator, an event-driven, network simulation tool, which allows an easy implementation of all model elements. The hierarchical assembly in OPNET is done in three main layers: a) The node model which specifies the main blocks and parameters of a node and provides an interface to the network element, b) the process model which defines the states and the state transitions of the node model elements and abstracts the behavior of the network element. C/C++ code that governs each state of a process model can be rapidly customized. OPNET Kernel Procedure APIs exist to facilitate development and support common communications mechanisms, such as packets, queues, and traffic, and c) the network model which contains the set of nodes and defines links between them.

4.2. *Implementation details*

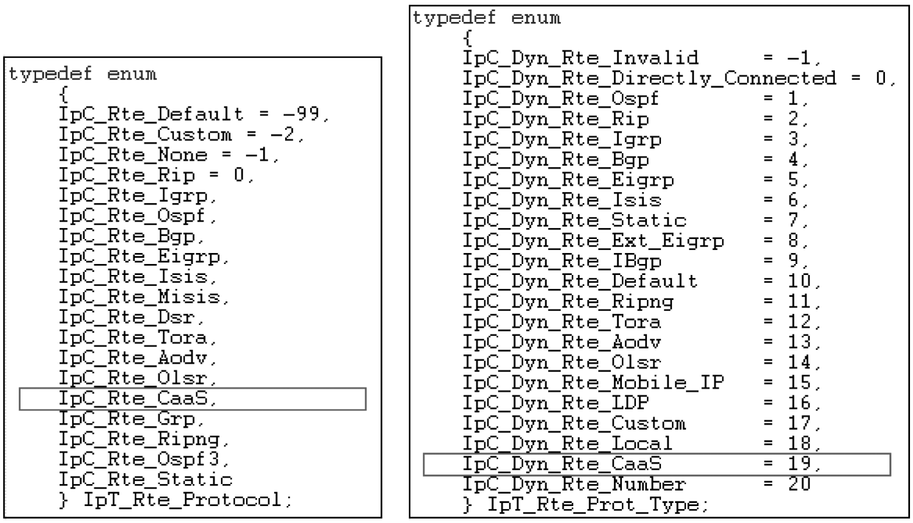
We implemented CaaS as a new custom proactive VANET routing protocol that interfaces with IP through UDP. Our protocol has the following features:

- Neighbor sensing mechanism.
- Fast topology change detection via periodic hello messages exchange.
- Neighboring nodes' subscription table maintenance and update.
- Maintenance of the cluster topology (route to members of the cluster and their subscriptions) by the clusterhead.
- DTN routing between a clusterhead and another clusterhead when no infrastructure is available.

Before explaining how these functionalities are evaluated, we need to briefly introduce the node model architecture of a VANET vehicle, running CaaS, in Opnet.

4.2.1. *CaaS architecture model*

The node model of a VANET vehicle running CaaS (or any other proactive VANET routing protocol) in Opnet is shown in Fig. 4. We created our protocol as a child process of manet_rte_mgr module which provides a common interface to multiple VANET routing protocols and is extensible for custom protocols. In Fact, manet_rte_mgr module is suitable for our situation because it is already interfaced to UDP, so no new "protocol number" is needed. IP simply thinks of the custom protocol as UDP on a port



```
typedef enum
    {
    IpC_Rte_Default = -99,
    IpC_Rte_Custom = -2,
    IpC_Rte_None = -1,
    IpC_Rte_Rip = 0,
    IpC_Rte_Igrp,
    IpC_Rte_Ospf,
    IpC_Rte_Bgp,
    IpC_Rte_Eigrp,
    IpC_Rte_Isis,
    IpC_Rte_Misis,
    IpC_Rte_Dsr,
    IpC_Rte_Tora,
    IpC_Rte_Aodv,
    IpC_Rte_Olsr,
    IpC_Rte_CaaS,
    IpC_Rte_Grp,
    IpC_Rte_Ripng,
    IpC_Rte_Ospf3,
    IpC_Rte_Static
    } IpT_Rte_Protocol;
```

```
typedef enum
    {
    IpC_Dyn_Rte_Invalid      = -1,
    IpC_Dyn_Rte_Directly_Connected = 0,
    IpC_Dyn_Rte_Ospf         = 1,
    IpC_Dyn_Rte_Rip          = 2,
    IpC_Dyn_Rte_Igrp         = 3,
    IpC_Dyn_Rte_Bgp          = 4,
    IpC_Dyn_Rte_Eigrp        = 5,
    IpC_Dyn_Rte_Isis         = 6,
    IpC_Dyn_Rte_Static       = 7,
    IpC_Dyn_Rte_Ext_Eigrp    = 8,
    IpC_Dyn_Rte_IBgp         = 9,
    IpC_Dyn_Rte_Default      = 10,
    IpC_Dyn_Rte_Ripng        = 11,
    IpC_Dyn_Rte_Tora         = 12,
    IpC_Dyn_Rte_Aodv         = 13,
    IpC_Dyn_Rte_Olsr         = 14,
    IpC_Dyn_Rte_Mobile_IP    = 15,
    IpC_Dyn_Rte_LDP          = 16,
    IpC_Dyn_Rte_Custom       = 17,
    IpC_Dyn_Rte_Local        = 18,
    IpC_Dyn_Rte_CaaS         = 19,
    IpC_Dyn_Rte_Number       = 20
    } IpT_Rte_Prot_Type;
```

Fig. 5. Set of routing protocols contributing entries to the common IP route table.

and once the port is set up, all we need is to send packets to UDP and have a receiver to get those packets from UDP. This module is either invoked by UDP through a stream interrupt (i.e., a packet for CaaS child process) or when CaaS child process directly sends packet to UDP on the connected port number. It is worth to mention that a process in Opnet is an instance of a process model and can dynamically create child processes and respond to interrupts.

In order to implement CaaS as a custom VANET routing protocol in Opnet, few steps need to be carefully performed. We need first to add CaaS to the list of dynamic routing protocol already implemented in Opnet and enumerated in the corresponding IP header files. Examples of these protocols include those listed in Fig. 5-a.

The CaaS process employs a routing table to keep track of valid routes to destination nodes. CaaS routing table is implemented as a hash table indexed by an IP address. CaaS routing table is populated and updated via Hello messages. OPNET implements packet forwarding within the IP module which uses a Common IP Routing Table. This routing table is updated and maintained by the routing protocol configured for the simulation study. Thus, CaaS and other routing protocols, in addition to maintaining their internal routing tables are also responsible for updating routing table at IP layer. Therefore, CaaS needs to be added to the list of those protocols that operate on IP Common Route Table as depicted in Fig. 5-b.

The second step is to create a new child process for CaaS and attach it to the parent manager process. This will be fully explained in the next section.

4.2.2. CaaS process model

Before proceeding with the implementation of any protocol in Opnet, a process model of this protocol needs to be created. Process Modeling Methodology (PMM) is a systematic approach to creating process models in OPNET Modeler and is considered to be the quickest and most efficient method of development thanks to the consistency of results it provides. Figure 6 shows our proposed process model for CaaS.

Our process model is composed of a forced (green) state and an unforced (red) state. The difference between the two is that the former is a non blocking state that deals with the initialization phase of nodes,

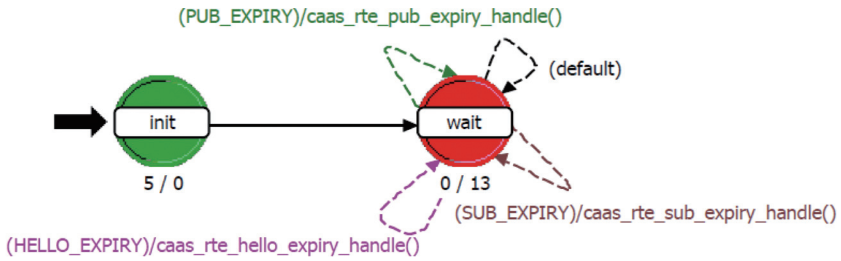
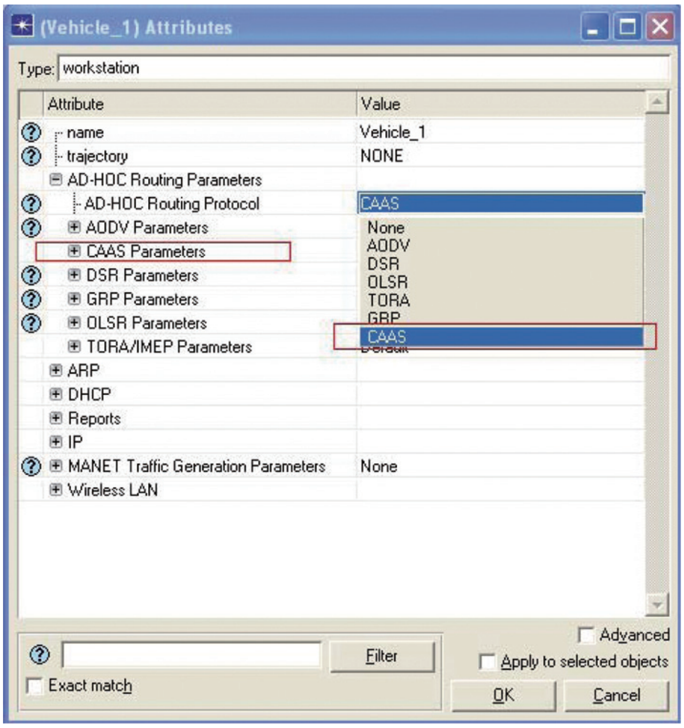

(PUB_EXPIRY)/caas_rte_pub_expiry_handle()
(default)
init
wait
5 / 0
0 / 13
(SUB_EXPIRY)/caas_rte_sub_expiry_handle()
(HELLO_EXPIRY)/caas_rte_hello_expiry_handle()

Fig. 6. CaaS process model.


(Vehicle_1) Attributes

Type: workstation

Attribute	Value
name	Vehicle_1
trajectory	NONE
AD-HOC Routing Parameters	
AD-HOC Routing Protocol	CAAS
AODV Parameters	
CAAS Parameters	
DSR Parameters	
GRP Parameters	
OLSR Parameters	
TORA/IMEP Parameters	
ARP	
DHCP	
Reports	
IP	
MANET Traffic Generation Parameters	None
Wireless LAN	

Fig. 7. A vehicle node attributes.

local and global parameters definition and attributes reading while the latter is a blocking state waiting for invocation.

The Simulation Kernel operates by maintaining an event list for the entire simulation. As the simulation executes, the Simulation Kernel manages a list of events to take place. As each event reaches the top (or head) of that list, it becomes an interrupt. Interrupts are often delivered to specific modules, and this occurrence is what activates the module's process model. All transitions in CaaS process model are from idle back to itself. Three interrupts are scheduled to occur: 1°) a publication is scheduled to be sent (PUB_EXPIRY condition), 2°) a subscription needs to be advertized (SUB_EXPIRY condition), and 3°) A simple hello message (mainly used for topology control) needs to be sent. A default transition is used because if there is a different type of interrupt, other than the three mentioned above, there must be a transition that the process model can follow. The default transition handles these different interrupt types. A transition with condition "default" is true if and only if no other conditions are true.

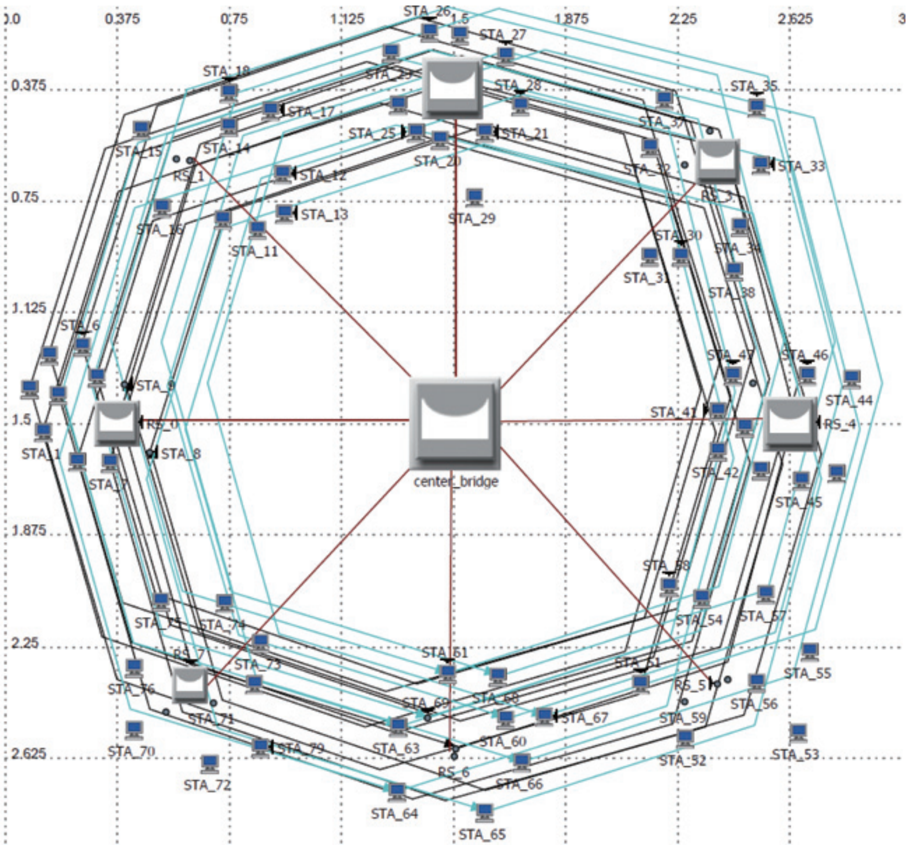


Fig. 8. CaaS network model.

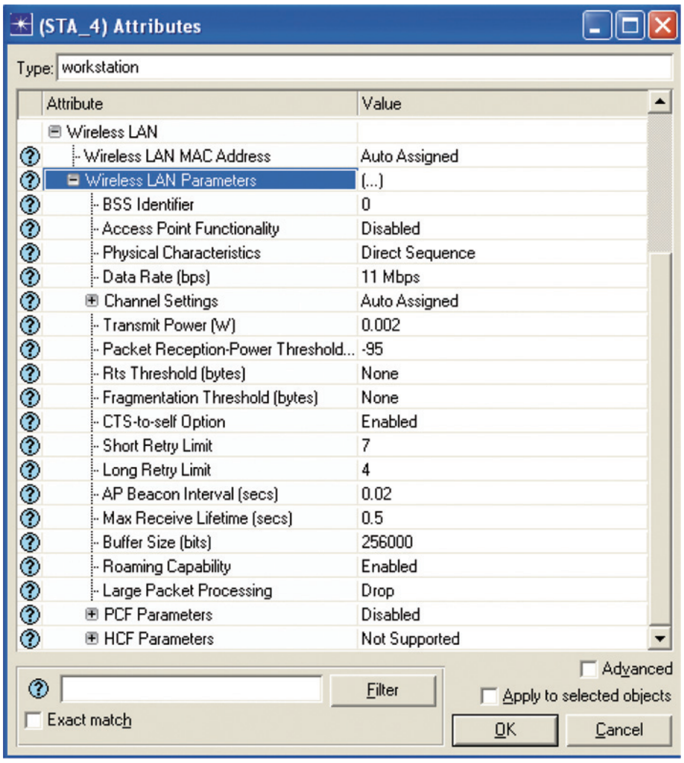
Three different timers manage those three interrupts. In the waiting state, the arrival of one of the three packet types previously sent is properly handled using the algorithm we explained in details in the previous section.

The child process can later be attached to the parent manager process. CaaS specific attributes, such as the hello interval attribute, should also be added to manet_rte_mgr model attributes (Fig. 7) so they can be seen while configuring the protocol at a node. Attributes will be parsed by the child process model.

4.3. Simulation settings

As a first experiment, we focused on a fully infrastructure-based scenario in which a large presence of roadside stations is considered. Our goal is twofold: on one hand we want to prove the correctness of CaaS, and on the other hand we want to evaluate the dissemination of publications to interested parties inside the clusters (i.e., intra-cluster structure).

The simulation scenario we used is shown in Fig. 8. Our network model consists of a 3Km x 3Km area in which 80 vehicles and 8 roadside units running CaaS are present. Some vehicles move in clockwise direction, and others move in counter-clockwise direction, both at a speed of maximum 110 Km/h. Opnet allows defining custom and realistic mobility patterns for each vehicle to meet most complex road topology requirements. We use 802.11b wireless radio interface (data rate 11 Mbps and 2 mW transmission power). Other simulation parameters are presented in Fig. 9.



Type: workstation

Attribute	Value
⊟ Wireless LAN	
Wireless LAN MAC Address	Auto Assigned
⊟ Wireless LAN Parameters	(...)
BSS Identifier	0
Access Point Functionality	Disabled
Physical Characteristics	Direct Sequence
Data Rate (bps)	11 Mbps
⊞ Channel Settings	Auto Assigned
Transmit Power (W)	0.002
Packet Reception-Power Threshold...	-95
Rts Threshold (bytes)	None
Fragmentation Threshold (bytes)	None
CTS-to-self Option	Enabled
Short Retry Limit	7
Long Retry Limit	4
AP Beacon Interval (secs)	0.02
Max Receive Lifetime (secs)	0.5
Buffer Size (bits)	256000
Roaming Capability	Enabled
Large Packet Processing	Drop
⊞ PCF Parameters	Disabled
⊞ HCF Parameters	Not Supported

Fig. 9. WLAN simulation parameters.

Each roadside station acts like a router with two interfaces: a wireless interface running CaaS and automatically subscribing to receiving all kinds of publications) and an Ethernet interface connecting the station to the backbone network. The node model of the roadside base station is shown in Fig. 10.

We, later, progressively reduce the number of roadside units to assess the impact of the additional DTN routing in such scenario.

4.4. Simulation results

In all our experiments, we measure the publication delivery ratio, expressed as the number of subscribers that successfully received publications they subscribed to receiving, the average delay between publication advertizing and their reception, and the traffic overhead defined as the number of transmissions received per minute by each vehicle. We later compare CaaS against a classic epidemic approach [42]. Results are averaged over multiple runs.

Simulation results show that the delivery ratio is 100% in a fully infrastructure-based network as depicted in Fig. 11. All subscribers (a total of ten subscribers in our simulation) have successfully received at least one publication regarding the service they subscribed to, which demonstrates the correctness of CaaS intra-cluster structure. Nevertheless, despite this important result, assuming a widespread presence of roadside units is unrealistic in many scenarios. For this particular reason, we decided to reduce progressively the number of those roadside stations and observe the impact of this on the delivery ratio.

Remarkably, publication delivery is above 90% even if no roadside stations are available (as depicted in Fig. 12). This is mainly due to DTN routing deployment in which clusterheads buffer publications

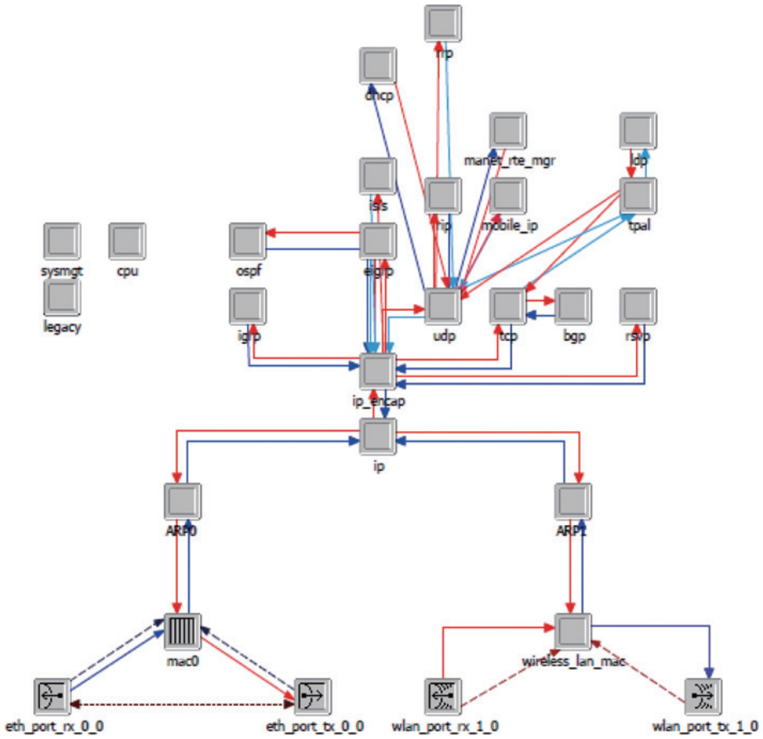


Fig. 10. Node model of the roadside base station.

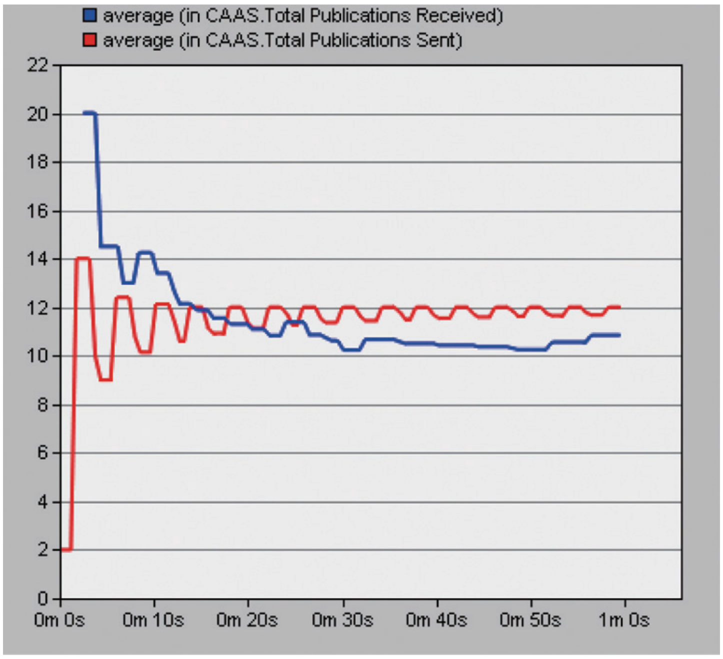


Fig. 11. Publication delivery in a fully infrastructure-based scenario.

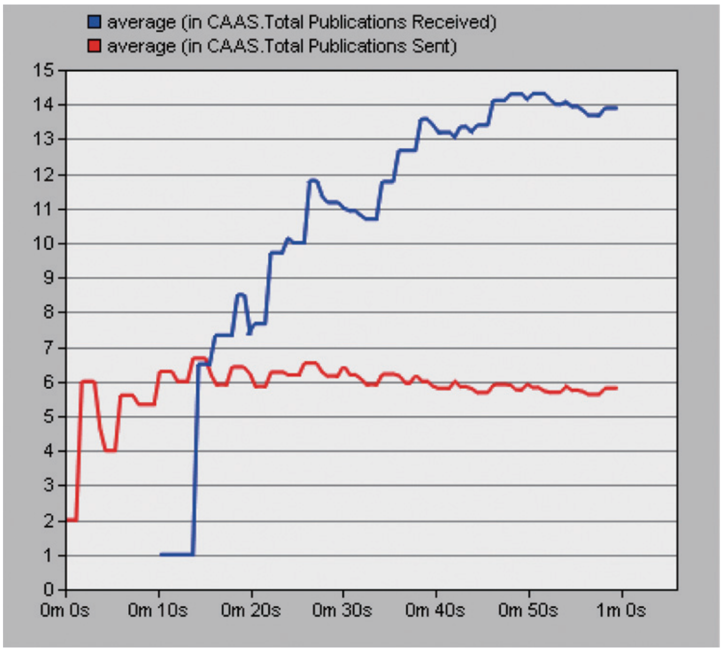


Fig. 12. Publication delivery in an infrastructure-free scenario.

and then forward them to other clusterheads when they are in their vicinity. When infrastructure is not available, transmitting publications to clusterheads that are going in the opposite direction represents the most favorable case for our protocol even if the connection between the two vehicles lasts only few seconds, as they act as information carriers that will rapidly alert vehicles going in the opposite direction about events affecting their route. The delay of publication reception in an infrastructure-free scenario is, however, higher than that of a fully infrastructure-based scenario. Figure 9-b shows that it takes at least 20 seconds to inform 90% of the subscribers about an event affecting their route while it was almost instantaneous in a fully infrastructure-based network. This is due to the multihop nature of transmission in pure VANET and intermittent connectivity issues, causing delays in packets buffering and appropriate neighbor finding.

To put our work in the context of related efforts, we also compared our protocol with an epidemic approach [42], in which all vehicles (subscribers and non-subscribers) buffer each message received and retransmit it to all neighbors which have not heard that message yet. In our experiment, we consider an infrastructure-free network since this is the most challenging situation for our protocol.

Simulation parameters are kept the same as in previous scenarios. We will not focus on comparing the delivery ratio since it is quite high for both protocols. Therefore, our main concern will be more about the network overhead induced by assuming that all communication in this experiment relies on vehicle-to-vehicle technology. Figure 13 shows that structuring the network into clusters significantly reduces the overall network overhead. This is explained by the fact that CaaS delivers publications only to interested vehicles inside the cluster, instead of all vehicles in the network. Moreover, since each clusterhead maintains a subscription table of all members of the cluster, it will not systematically forward all received publications to members of the cluster unless they are interested in receiving them, which confirm our claims of message overhead reduction.

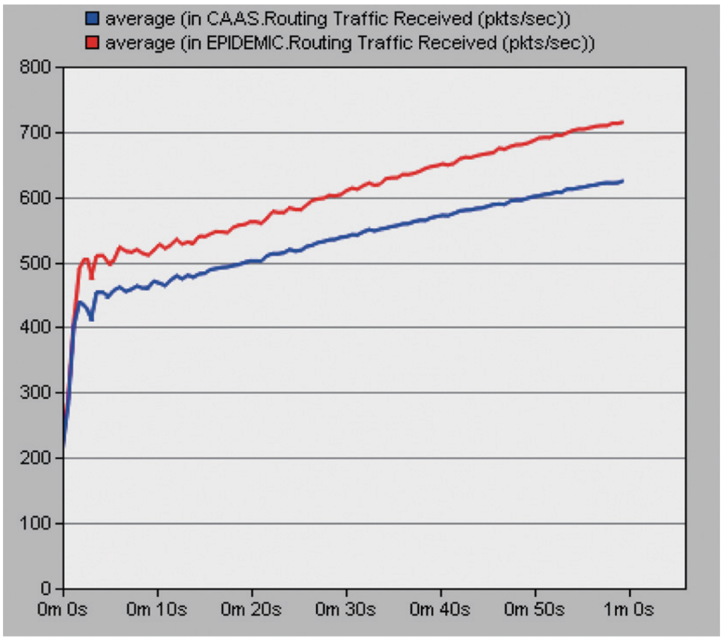


Fig. 13. Network overhead in CaaS and in epidemic routing.

5. Concluding remarks and future work

In this paper, We introduced a new solution for VANETs, referred to as Cooperation as a Service (CaaS), that allows providing vehicles/drivers, which are willing to cooperate, with some sets of services using very minimal infrastructure, by taking advantage of Vehicle-to-Vehicle (V2V) communications.

To tackle the infrastructure-less nature of these networks, organize our network into clusters that can be properly maintained and to use CBR for intra-cluster communications and DTN routing for inter-cluster ones.

We designed Cooperation as a Service (CaaS) to mainly provide drivers with services for free using vehicles' cooperation. There is one more theme that will contribute to shape CaaS in a more effective way: privacy (of location and motion patterns) and security (mainly, confidentiality and protection from DOS (Denial of Service) attacks). We did not dwell on the privacy/security topic in this paper because we chose to focus on conventional network layer aspects. However, two trends are clear. The need of a Certificate Authority (CA) will require efficient connection to Internet Servers. At the same time, to handle protection from bogus attacks in situations when they are disconnected from the Internet, or it is simply too time consuming to consult the Internet CA, the mobile users must organize in P2P communities and use majority rules and/or elect proxy mobile CAs to resolve security issues [16].

Acknowledgment

Professor Olariu was supported by NSF grants CNS 0721586.

References

[1] M. Caliskan, D. Graupner and M. Mauve, Decentralized discovery of free parking places, in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks*, (2006), pages 30–39.

[2] A. Carus, A. Urpi, S. Chessa and S. De, GPS-free coordinate assignment and routing in wireless sensor networks, in *Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM'05), Vol. 1, (2005), pages 150–160.

[3] D. Caveney, in: *VANET: Vehicular Applications and Inter-Networking Technologies*, H. Hartenstein and K. Laberteaux, eds, A John Wiley and Sons, UK, 2010, pp. 21–48.

[4] T.G. Crainic, M. Gendreau and J. Potvin, Intelligent freight transportation systems: Assessment and the contribution of operations research, *Transportation Research Part C: Emerging Technologies* **17**(6) (2009), 541–557.

[5] G. Cugola and E. Di Nitto, On adopting Content-Based Routing in service-oriented architectures, *Information and Software Technology* **50**(1–2) (2008), 22–35.

[6] T. Delot, S. Ilarri, N. Cenerario and T. Hien, Event sharing in vehicular networks using geographic vectors and maps, *Mobile Information Systems* **7**(1) (2011), 21–44.

[7] M.D. Dikaiakos, A. Florides, T. Nadeem and L. Iftode, Location aware services over vehicular ad hoc networks using car-to-car communication, *IEEE Journal on Selected Areas in Communications* **25**(8) (2007), 1590–1602.

[8] S. Dornbush and A. Joshi, StreetSmart Traffic: Discovering and disseminating automobile congestion using VANET's, in *Proceedings of the 65th IEEE Vehicular Technology Conference*, (2007), pages 11–15.

[9] P.Th. Eugster, P.A. Felber, R. Guerraoui and A.M. Kermarrec, The many faces of publish/subscribe, *ACM Computers Survey* **35**(2) (2003), 114–131.

[10] M.D. Fontaine, in: *Vehicular Networks from theory to Practice*, S. Olariu and M.C. Weigle, eds, CRC Press, USA, 2010, pages 1-1/1-26.

[11] R. Frenkiel, B. Badrinath, J. Borras and R.D. Yates, The infostations challenge: balancing cost and ubiquity in delivering wireless data, *IEEE Personal Communications* **7**(2) (2002), 66–71.

[12] D.J. Gabauer and H.C. Gabler, Comparison of roadside crash injury metrics using event data recorders, *Accident Analysis & Prevention* **40**(2) (2008), 548–558.

[13] G. Gehlen and L. Pham, Mobile Web services for peer-to-peer applications, in *Proceedings of IEEE international conference on consumer communications and networking*, (2005), pages 427–433.

[14] M. Gerla and L. Kleinrock, Vehicular networks and the future of the mobile internet, *Computer Networks*, (2010), doi:10.1016/j.comnet.2010.10.015.

[15] P. Gupta and P.R. Kumar, The capacity of wireless networks, *IEEE Transactions on Information Theory* **46**(2) (2000), 388–404.

[16] X. Hong, D. Huang, M. Gerla and Z. Cao, SAT: building new trust architecture for vehicular networks, In *proceedings of ACM SIGCOMM Workshop on Mobility in the Evolving Internet Architecture* (*MobiArch*), 2008.

[17] R.C. Hsu and L.R. Chen, An integrated embedded system architecture for invehicle telematics and infotainment system, in *Proceedings of the IEEE International Symposium on Industrial Electronics* **4** (2005), pages 1409–1414.

[18] S. Jain, K. Fall and R. Patra, Routing in a Delay Tolerant Network. In SIGCOMM '04: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press. ISBN 1-58113-862-8, (2004), pages 145–158.

[19] D. Jiang, V. Taliwal, A. Meier, W. Holfelder and R. Herrtwich, Design of 5.9 GHz DSRC-based vehicular safety communication, *IEEE Wireless Communications* **13**(5) (2006), 36–43.

[20] I. Khalil, R. Kronsteiner and G. Kotsis, A semantic solution for data integration in mixed sensor networks, *Computer Communications* **28**(13) (2005), 1564–1574.

[21] S.B. Lee, G. Pan, J.S. Park, M. Gerla and S. Lu, Secure incentives for commercial ad dissemination in vehicular networks, in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, (2007), pages 150–159.

[22] I. Leontiadis and C. Mascolo, GeOpps: Opportunistic Geographical Routing for Vehicular Networks, in *Proc. of the IEEE Workshop on Autonomic and Opportunistic Communications*, Helsinki, Finland, 2007.

[23] C. Lochert, M. Mauve, H. Fußler and H. Hartenstein, Geographic routing in city scenarios, *ACM SIGMOBILE Mobile Computing and Communications Review* **9**(1) (2005), 69–72.

[24] C. Maihofer, A survey on geocast routing protocols, *IEEE Communications Surveys and Tutorials* **6**(2) (2004) 32–42.

[25] L. Mottola, G. Cugola and G.P. Picco, A Self-repairing tree topology enabling content-based routing in mobile ad hoc networks, *IEEE Transactions on Mobile Computing* **7**(8) (2008), 946–960.

[26] H. Mousannif, H. Al Moatassime and S. Rakrak, An energy-efficient scheme for reporting events over WSNs, *International Journal of Pervasive Computing and Communications* **7**(1) (2011), 44–59.

[27] H. Mousannif, I. Khalil and H. Al Moatassime, Cooperation as a Service in VANETs, *Journal of Universal Computer Science* **17**(8) (2011), 1202–1218.

[28] H. Mousannif, I. Khalil and S. Olariu, Cooperation in static and mobile sensor-based platforms for situation, activity and goal awareness. Proceedings of the 2011 international workshop on Situation activity & goal awareness at the 13th ACM International Conference on Ubiquitous Computing (UbiComp2011), (2011), pages 5–14.

[29] T. Nadeem, S. Dashtinezhad, C. Liao and L. Iftode, TrafficView: traffic data dissemination using car-to-car communication, *ACM SIGMOBILE Mobile Computing and Communications Review* **8**(3) (2004), 6–19.

[30] E.F. Nakamura, A.A.F. Loureiro and A.C. Frery, Information fusion for wireless sensor networks: Methods, models, and classifi cations, *ACM Computing Surveys* **39**(3) (2007), 9/1–9/55.

[31] A. Nandan, S. Das, S. Tewari, M. Gerla and L. Klienrock, AdTorrent: delivering location cognizant advertisements to car networks, in *The Third International Conference on Wireless On Demand Network Systems and Services* (WONS 2006), Les Menuires, France, 2006.

[32] National Highway Traffic Safety Administration, Traffic safety facts – preliminary 2009 report, http://www-nrd.nhtsa.dot.gov/Pubs/811255.pdf, March 2010.

[33] S. Olariu, I. Khalil and M. Abuelela, Taking VANETs to the cloud, *International Journal of Pervesive Computing and Communication* **7**(1) (2011), 7–21.

[34] K. Ozbay, H. Nassif and S. Goel, Propagation characteristics of dynamic information collected by in-vehicle sensors in a vehicular network, in *Proceedings of the IEEE Intelligent Vehicles Symposium*, (2007), pages 1089–1094.

[35] C.E. Palazzi, M. Roccetti, G. Pau and M. Gerla, Online Games on Wheels: Fast Game Event Delivery in Vehicular Ad-hoc Networks, in *V2VCOM'07*, Istanbul, Turkey, 2007.

[36] N. Qadri, M. Altaf, M. Fleury and M. Ghanbari, Robust video communication over an urban VANET, *Mobile Information Systems* **6**(3) (2010), 259–280.

[37] M. Saito, J. Tsukamoto, T. Umedu and T. Higashino, Design and evaluation of intervehicle dissemination protocol for propagation of preceding traffic information, *IEEE Transactions on Intelligent Transportation Systems* **8**(3) (2007), 379–390.

[38] S. Sesay, Z. Yang and J. He, A Survey on Mobile Ad Hoc Wireless Network, *Information Technology Journal* **3**(2) (2004), 168–175.

[39] S. Smaldone, L. Han, P. Shankar and L. Iftode, RoadSpeak: Enabling Voice Chat on Roadways using Vehicular Social Networks, in *SocialNets'08*, Glasgow, Scotland, UK, 2008.

[40] W.W. Terpstra, J. Kangasharju, C. Leng and A.P. Buchmann, BubbleStorm: resilient, probabilistic, and exhaustive peer-to-peer search, in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ACM, New York, NY, USA, 2007, pages 49–60.

[41] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige and V. Sadekar, Broadcasting in VANET, in *Proceedings of the 2007 Workshop on Mobile Networking for Vehicular Environments*, 2007, pages 7–12.

[42] A. Vahdat and D. Becker, Epidemic routing for partially connected ad Hoc networks, Tech. Rep., Duke University, 2000.

[43] K.J. Wong, B.S. Lee, B.C. Seet, G. Liu and L. Zhu, BUSNet: Model and Usage of Regular Traffic Patterns in Mobile Ad Hoc Networks for Inter-vehicular Communications, in *Proc. 10th International Conference on Telecommunications* (ICT '03), 2003, pages 102–108.

[44] S. Yoo, J.H. Son and M.H. Kim, A scalable publish/subscribe system for large mobile ad hoc networks, *Journal of Systems and Software* **82** (2009), 1152–1162.

[45] J. Zhao and G. Cao, VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks, in *Proc. 25th IEEE International Conference on Computer Communications* (INFOCOM '06), 2006, pages 1–12.

Hajar Mousannif received her M.S. degree in Telecommunications and Computer Sciences from the National Institute of Posts and Telecommunications (INPT), Rabat (Morocco) in 2005. She joined, the same year, the Faculty of Sciences and Techniques Marrakech (Cadi Ayyad University, Morocco) as a telecom engineer where she was mainly responsible for managing and administrating the network. Since 2006, she is a teacher within the Department of Computer Sciences at the same Faculty. Her primary research interests include wireless sensor networks, wireless adhoc networks, and next generation internet technologies with main focus on routing protocols development and mobility management. In addition to her academic experience, she was in the program committee chair of many national conferences.

Ismail Khalil is a senior researcher and lecturer at the institute of telecooperation, Johanes Kepler University Linz, Austria since October 2002. He is the president of the international organization of Information Integration and Web-based Applications & Services @WAS. He holds a PhD in computer engineering and received his habilitation degree in applied computer science on his work on agents interactions in ubiquitous environments in May 2008.

He currently teaches, consults, and conducts research in Mobile Multimedia, Agent Technologies, and the Semantic Web and is also interested in the broader business, social, and policy implications associated with the emerging information technologies. Dr. Khalil has authored around 100 scientific publications, books, and book chapters. He serves as the editor-in-chief of the International Journal on Web Information Systems (IJWIS), International Journal on Pervasive Computing and Communication

(IJPCC), Journal of Mobile Multimedia (JMM), International Journal of Mobile Computing and Multimedia Communication (IJMcMc), Advances in Next Generation Mobile Multimedia book series, and Atlantis Ambient and Pervasive Intelligence book series. He is on the editorial board of several international journals. His work has been published and presented at various conferences and workshops. (http://www.iiwas.org/ismail/)

Professor Olariu has held many different roles and responsibilities as a member of numerous organizations and teams. Much of his experience has been with the design and implementation of robust protocols for wireless networks and in particular sensor networks and their applications. Professor Olariu is applying mathematical modeling and analytical frameworks to the resolution of problems ranging from securing communications, to predicting the behavior of complex systems, to evaluating performance of wireless networks. His research interests are in the area of complex systems enabled by large-scale deployments of sensors and vehicular networks and cloud computing.

Currently, Professor Olariu is the Associate Editor in Chief of IEEE Transactions on Parallel and Distributed Systems and an Associate Editor of IEEE Transactions on Computers.

Advances in Multimedia
The Scientific World Journal
International Journal of Distributed Sensor Networks
Journal of Industrial Engineering
Applied Computational Intelligence and Soft Computing
Advances in Fuzzy Systems
Modelling & Simulation in Engineering
Journal of Computer Networks and Communications
Advances in Artificial Intelligence
Advances in Computer Engineering
International Journal of Computer Games Technology
International Journal of Biomedical Imaging
Advances in Artificial Neural Systems
Advances in Software Engineering
Journal of Robotics
Advances in Human-Computer Interaction
Computational Intelligence and Neuroscience
International Journal of Reconfigurable Computing
Journal of Electrical and Computer Engineering