Abstract: This paper defines a novel semantic for multicast in vehicular ad hoc networks (VANETs) and it defines a middleware, the Localized Vehicular Multicast Middleware (LVMM) that enables minimum cost, source-based multicast communications in VANETs. The middleware provides support to find vehicles suitable to sustain multicast communications, to maintain multicast groups, and to execute a multicast routing protocol, the Vehicular Multicast Routing Protocol (VMRP), that delivers messages of multicast applications to all the recipients utilizing a loop-free, minimum cost path from each source to all the recipients. LVMM does not require a vehicle to know all other members: only knowledge of directly reachable nodes is required to perform the source-based routing.

Key-Words: Vehicular networks, ad hoc networks, routing protocols

1 Introduction
Multicasting [3, 4] with QoS guarantees [3, 10] cannot be efficiently adopted in wide highly mobile wireless networks or even small networks with unconstrained mobility of independent nodes. In fact, due to the mobility of hosts there could be no partners for a multicast session: disconnections and different mobilities of hosts may prevent lasting multicast communications among nodes. Furthermore, classical ID-based multicast in MANET should face with additional problems related to mobility: fixed areas cannot be defined (like it is done in Geomulticasting [1, 5]), hosts do not have fixed positions, and multicast mechanisms have to respond to network dynamics in addition to group dynamics [3, 7, 8].

In order to provide multicast support in a highly mobile environment, some degree of network stability is fundamental. Without a stable physical topology no multicasting with QoS guarantees is feasible, neither lasting multicast communications among nodes.

Even when the underlying network has an overall unstable topology, looking at geographic proximity of some hosts over intervals of time, a stable sub-topology of the whole network can be identified. This is especially true in vehicular ad hoc networks (VANETs) [2], where vehicles continuously move (and that is why a stable multicast group cannot be identified in a fixed geographic area) but by exploiting vehicular mobility patterns and hosts proximity, a stable mobile multicast group for some of the nodes can be identified. Thus, although in the overall network topology, this multicast group is continuously moving (and thus changing its geographic coordinates), if taken in isolation from the rest of the network, it forms a stable group in which hosts’ movements are similar and the hosts form a relative topology which is time-stable. We will call this topology a Mobile Multicast Group (MMG).

In this paper we focus on multicast in VANET, and, based on the concept of MMG, we introduce a middleware layer (the Localized Vehicular Multicast Middleware or LVMM) that, from a highly mobile network, finds sets of vehicles suitable and willing to form MMGs. The nodes in a MMG have the property that they will form a stable topology for a medium/long interval of time. This group of nodes runs a source-based multicast protocol (VMRP) that delivers messages of multicast applications to all the nodes in the group using

1 In this paper the terms node, host and vehicle will be used as synonyms.
a loop-free, minimum cost path from each source to all the recipients.

LVM’s main function is thus to build and maintain MMGs and to route multicast packets among groups’ participants. LVM allows groups of vehicles to exchange data about road and traffic conditions in an ad hoc manner to enable applications such as cooperative adaptive cruise control, but also video-conferencing, real-time messaging, distributed games, chats, etc.

The rest of the paper is organized as follows. Section 2 presents the system model, Section 3 describes the LVM architecture, and Section 4 describes the multicast protocol. The analysis of the protocol is presented in Section 5 and Section 6 discusses possible LVM extensions. The conclusions are drawn in Section 7.

2 System Model and Assumptions

The Localized Vehicular Multicast Middleware (LVM) is a middleware enabling vehicles to execute distributed multicast applications in VANETs. It assumes that each vehicle is equipped with a GPS (or a D-GPS or any other position tracking device that allow an instantaneously knowledge of its positions), a radio transceiver and computational power.

LVM manages multicast groups and defines and maintains multicast routes among members. To this purpose it creates and maintains Mobile Multicast Groups (MMGs) among vehicles on a roadway and it allows multicast communications among the members of a MMG.

A MMG is a group of mobile nodes that, for a given probability α maintain a stable topology for an interval of time of at least τ. A MMG is a time stable configuration of neighbouring mobile nodes. These nodes show similar mobility patterns and mobility rate and are located in a mobile surface. Each node member of a MMG can be the source of multicast packets; a MMG is an all-to-all multicast group: when a node sends a packet all other members receive it.

Let us consider a bi-dimensional plane with the coordinates x and y and, without loss of generality, let us assume that a roadway develops along the x axis.

LVM models every vehicle vi by the 4-tuple <IDi, Posi, Veli, R > , where IDi is a unique identifier of the vehicle, R is a transmission range that is assumed to be the same for all the vehicles and which is used to determine links among vehicles, Posi is the position of the vehicle (obtained for example from the GPS), and Veli is its velocity. The parameters Posi and Veli of each node are utilized by LVM to build and maintain the MMGs.

Under the assumption that the roadway develops along the x axis, without loss of generality we represent the stretch of roadway containing a MMG with a straight segment, that is the roadway’s width is negligible (y axis) and the roadway’s shape is negligible (a segment parallel to the x axis).

We can omit roadway’s width and shape because transmission ranges cover all roadway’s width and vehicles utilize wireless radio transmissions. This allows us to consider all vehicles as lying on a straight line and their GPS devices allow us to order all vehicles on the line. Thus Posi represents a position on the x axis. For this reason in the rest of the paper we use notation xi to denote the value of Posi. For the sake of simplicity we also assume that all xi are also different from each other. If it happens that in a time interval two or more xi are equals then we can utilize other values from the GPS devices to order the vehicles or in another way find an order.

Since the vehicles are associated to points on the x axis, a MMG at time t can be modelled in a more rigorous way as follows.

Let n be the number of members of the given MMG at time t, and M be the set of members of the MMG (they have already joined the group). Let also ϕ :Identifier→Position be a function that, given the identifier of a vehicle, returns its position, thus ϕ(IDi)=xi, and let (xi,xj) represent a bidirectional link between the vehicle with position xi and the vehicle with position xj.

LVM models a MMG at time t as the graph MMG=(X, E) (called the MMG-graph), where X = \{ xi; ∀IDi ∈ M ∧ ϕ(IDi) = xi \} is the set of points on the x axis that correspond to the positions of the members of the MMG, and

E = \{(xi, xj) \mid |xi - xj| < R, xi ∈ X, xj ∈ X, (i ≠ j) \} is the set of bi-directional edges which represent communication links among the corresponding vehicles.

Note that, although in some cases communication links may be unidirectional, we deliberately ignore these links in the construction of the MMG-graph. Note that a MMG-graph models a MMG in its entirety, and we assume that a MMG-graph is a sub-case of the widely known unit disc model. Thus for each pair xi,xj there exists a link between the correspondent vehicles if and only if the Euclidean distance between them is less than the transmission radius R. We also assume that each link (xi,xj) has the same associated cost.

Due to vehicles mobility, a MMG-graph always changes with time, thus a MMGt=(X,E) represents a MMG in a given interval of time, and it should be recomputed.
when topology changes are detected. Note that topology changes can be easily detected either by the MAC layer or by using periodic hello messages. Figure 1 shows an example of a \textit{MMG-graph} of a MMG at a given interval of time \( t \) which is made up of five vehicles, where the set of members of the MMG is \( M = \{ ID_1, ID_2, ID_3, ID_4, ID_5 \} \).

Summarizing, a \textit{MMG-graph} has the following properties:

1) it is a \textit{dynamic} graph; a MMG changes over time due to the mobility of its nodes and a \textit{MMG-graph} is valid only for a given interval of time;
2) it is a \textit{one-dimensional} graph: roadway can be modelled with a straight segment, omitting roadway’s width;
3) it is a graph of \textit{ordered nodes}: each node ID has a position \( x_i \) that uniquely identifies the node on the segment;
4) it models a MMG in its \textit{entirety}: a \textit{MMG-graph} includes all nodes of a MMG and all links existing between each pair of members;

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{A MMG-graph modelling the MMG of 5 vehicles.}
\end{figure}

3 The architecture of LVMM

The architecture of the Localized Vehicular Multicast Middleware addresses all relevant aspects from the physical network to the multicast protocol definition. The LVMM architecture comprises 2 sub-layers: the TPG sub-layer and the MMG sub-layer, which address the tasks of network monitoring, multicast groups’ maintenance and multicast protocol definition.

The TPG sub-layer and the Neighbourhood Service - The consequences of nodes’ mobility suggest the inclusion of a quantitative measure of mobility directly in the nodes’ selection process. The group of stable vehicles is discovered by having each node monitoring all its directly reachable neighbours. This monitoring is performed by analyzing special beacons that are periodically broadcasted by all vehicles to their neighbours. Each beacon contains information about the sending vehicle; in particular its ID, its position and velocity, and it also may contain statistics about the past speed of the vehicle itself.

Each vehicle collects the beacons received by its neighbours in order to construct a set of stable neighbours. In particular this set will contain neighbours which, for a given interval of time, have velocities close to the velocity of the monitoring node. In this way the group of neighbouring vehicles has the property that relative distances between members will not show great differences on a given interval of time. However the proximity property among vehicles might hold for intervals of time that can be very long, depending on the way taken by nodes and on traffic conditions.

The \textit{Neighbourhood Service} of a vehicle collects the beacons and records the neighbours of the vehicle in real time. This level utilizes a list (called \textit{NeighboursList}) which contains all directly reachable nodes: each entry of this list contains the identity of each neighbour along with the data received within each beacon message: its position, its velocity and a timestamp. Let \( v_i \) be the monitoring vehicle, and let \( R \) be the transmission radius of every vehicle. The \textit{NeighboursList} of a monitoring node can be defined as follows:

\[ \text{NeighboursList}(v_i) = \{ j \mid v_i - x_j < R \} \]

Upon the \textit{Neighbourhood Service} lies the TPG sub-layer, which selects the set of directly reachable vehicles that are suitable to form a MMG: we refer to such a set of stable neighbours as a \textit{Time-Proximity Group} (TPG). A TPG is the building block of a MMG.

More specifically, the TPG(\( \tau, \alpha \)) of a given node \( v_i \) (also denoted with TPG(\( v_i), \tau, \alpha \)) is defined as the set including all the nodes inside \( v_i \)’s transmission range that will remain within \( v_i \)’s proximity (i.e. will be directly reachable) for at least a period of time \( \tau \) with a probability of at least \( \alpha \).

The TPG sub-layer of each vehicle keeps updated the TPGs for each given pair (\( \tau, \alpha \)) by working on the following input data:

- the \textit{NeighboursList} from the Neighbourhood Service,
- the two parameters \( \tau \) and \( \alpha \) which are specified by the application layer.

The \textit{TPG sub-layer} utilizes a mobility prediction algorithm to select the neighbours suitable to form TPGs from the \textit{NeighboursList}: these nodes are filtered by the mobility prediction algorithm depending on \( \alpha \) and \( \tau \).

At each node \( v_i \), the \textit{TPG sub-layer} proactively updates a TPG(\( v_i), \tau, \alpha \)) based on the changes to the \textit{NeighboursList}; this way a TPG always reflects the actual topology; it can be defined in the following way: \( \text{TPG}(v_i), \tau, \alpha) = \text{mobility\_prediction}(\tau, \alpha, \text{NeighboursList}(v_i)) \)

In a real environment, a multicast application (for example a video-chat or generally an infotainment
application) may require, for example, that nodes in the
time proximity group be always reachable for at least one
minute with a probability of 0.70. Then it means that, at
each instant, the mobility prediction algorithm at each
node will filter out all neighbours that, from sample
observations, will not be judged as directly reachable
after an interval of time of at least \( \tau = 1 \) minute with a
probability \( \alpha = 0.70 \). It should be observed that the
algorithm for mobility prediction may take into account
only vehicles position and speed, or it may also consider
the road topology and statistics about the behaviour of
the vehicles which should be communicated in the
periodic beacons. Note however that the actual
implementation of the algorithm for mobility prediction
is out of the scope of this paper.

Note that parameter \( \tau \) does not specify the minimum
required length of time for a communication, but the
minimum length of time of the future stability that is
always required to every node: at each instant, only those
nodes that, with probability \( \alpha \) will remain stable
throughout the successive interval of time (whose length
is at least \( \tau \)), enter the TPG(\( \tau,\alpha \)). Thus also holds
\( TPG(\tau',\alpha) \subseteq TPG(\tau,\alpha) \) if \( \tau' < \tau \).

It should be observed that a TPG holds both proximity
and temporal properties on a per-node basis and those
properties, when made global to the MMG, determine
group’s properties. Note also that, although proximity
could be defined on 2-hops neighbours (or more), this
would require additional overhead and intermediate
(non-stable) nodes not belonging to the group might be
involved in group maintenance to forward the beacons to
the 2-hop neighbours and in routing of applications’
messages. Utilizing non stable nodes might lead to a less
robust multicasting, making difficult to achieve the
desired degree of stability (and thus availability). For this
reason a MMG is built upon a subset of all nodes in each
TPG. Note also that not all nodes in the set of all TPGs
should be involved in a MMG, but only some selected
nodes. This selection is driven by the application.

The MMG sub-layer and the CTPG Service - The
MMG sub-layer is responsible for the creation and
maintenance of MMGs. It periodically broadcasts special
purpose packets to advertise to the neighbouring vehicles
that a MMG is present in the area, so that new nodes can
join an existent MMG. This layer also runs the protocol
for admission of new vehicles in a MMG.

This sub-layer is also responsible for the maintenance of
the multicast routing tables and the execution of the
multicast routing. It receives updated values about the
underlying local topology by the TPG sub-layer and it
updates the routing tables of the vehicle based on this
information.

When a node receives a multicast packet, this is
processed by the MMG sub-layer, the packet is passed to
the Vehicular Multicast Routing Protocol (VMRP)
which is embedded in the MMG sub-layer and which
forwards the packet utilizing the updated routing table
correspondent to the MMG to which the packet belongs.
At this sub-layer each vehicle knows the identities of all
the members of a given MMG: each vehicle maintains a
view of the global membership with a “soft state”: a
table contains an entry for each member of a given
MMG and a timeout is associated to each member. If a
vehicle does not refresh its membership, with a special
purpose multicast packet, it is no more considered a
member by the other nodes when the timeout expires.
The “soft state” mechanism relies on the underlying
multicast routing protocol to exchange messages of
membership refresh.

Fundamental to the MMG sub-layer is the CTPG Service. It applies a second filter over the neighbouring
nodes by selecting from the set of nodes contained in a
given TPG only those nodes involved in multicast
communications. In fact, in general, the instance of an
application \( P \) on a node \( v \), that wants to establish an
MMG with parameters \( \alpha \) and \( \tau \) will not include in the
group all the nodes that are contained in the TPG(\( v, \)
\( \tau, \alpha \)), but only a subset. The set of the stable neighbours
of \( v \) that join \( M \) is the CTPG(\( v, \tau, \alpha, M \)).

In other words if \( M \) is the Mobile Multicast Group for an
application \( P \) and TPG(\( \tau, \alpha \)) is the Time-Proximity Group
of each vehicle relative to \( M \) then, each member of \( M \) has a Chosen Time-Proximity Group, denoted
CTPG(\( v, \tau, \alpha, M \)), that holds the nodes contained in its
TPG(\( \tau, \alpha \)) that have joined \( M \).

A CTPG(\( v, \tau, \alpha, M \)) can be defined as follows:

\[
CTPG(v, \tau, \alpha, M) = \{ v_j : v_j \in TPG(v, \tau, \alpha) \land ID_j \in M \}
\]

Where \( ID \in M \) means that vehicle \( ID \) joined \( M \). A CTPG is
built upon a TPG and it is a set containing the stable
neighbours from the TPG that have joined a given
MMG. It is important to note that a TPG(\( v, \tau, \alpha \)) is
unique: it contains the neighbours suitable for a
communication with the requirements of stability
specified by the parameters \( \alpha \) and \( \tau \), thus it is application
independent. On the other hand, each application \( P \) has
its own Mobile Multicast Group and, on each node, it
defines its own CTPG which is thus application
dependent.
4 The VMRP Protocol

The *MMG-graph* models a global view of the multicast group, hence, in order to build a *MMG-graph*, each node should have updated information about all other members. With such global knowledge, each vehicle could, independently from each other, compute the routing tree to perform multicast. The construction of the *MMG-graph*, although possible, introduces a high overhead because each node in a multicast group should pro-actively maintain an updated view about all other members of the group, and this problem may severely limit the scalability of this approach.

In our approach a global knowledge of the *MMG-graph* is not needed. Instead, each node makes its routing decisions using only its view of the CTPG associated to a MMG; still the final result is a least-cost source-based (a tree for each source) multicast routing. Avoiding the explicit construction of the MMG graph by all nodes in a MMG eliminates much overhead without affecting the final result.

LVMM utilizes a proactive routing protocol, the *Vehicular Multicast Routing Protocol* (VMRP), which keeps routes updated. VMRP stores route information similar to routing protocols for static networks (essentially, a routing table has an entry for each node v in a CTPG): the routing table contains the nodes in the CTPG to which a vehicle has to forward the packets received from any given neighbour in the CTPG. This way, the routing path followed by each multicast packet is a sequence of nodes that is not explicit: it corresponds to a next hop table lookup at each vehicle along the route.

Upon receiving a packet each node knows if it has to forward the packet to other nodes and it knows the nodes to which it must forward the packet. No on-demand actions must be performed. Having updated routing tables expedites packets forwarding and leaves space in packets for application layer data because the destinations of each packet must not be inserted in the packets’ headers. A proactive state makes forwarding faster than approaches without state keeping and allows packets to carry more application data.

Summarizing, VMRP has the following features: it is proactive; utilizes only local knowledge; and it utilizes the *least number of transmissions* to deliver a packet from each source to all the destinations.

**Creating and Joining a Group** - When a source node sends a packet to a multicast group MMG, it must be delivered to all members of the group. Let M be the MMG for an application P and let n be the number of its members. For each member v_i of M, let C_i denotes the CTPG associated to M.

When a node v_j creates a new group, it firstly informs its TPG layer about the parameters α and τ related to the new group (the parameters above can also be obtained as a function of the velocity and position of the node). The TPG layer periodically sends a special create control packet to inform the set of stable neighbours of v_j about the existence of the group. This packet includes the identity of the group and its parameters (α and τ), this way, when a new vehicle enters the ‘coverage area’ of a group, it is provided with information about the group and an opportunity to join. The protocol does not perform a network wide broadcast, since we would restrict the set of nodes that could join the newly created group only to stable neighbours of v_j.

The “join” phase of the protocol is performed with a soft-state approach. That is, when a node v_j wish to join a group, it sends a *join* message that explicitly informs the nodes in CTPG(v_j, α, τ) about its membership to the group.

In principle the MMG could grow unbounded since new nodes can incrementally be added to the group. However, the new nodes added to the group should be stable and thus should guarantee connectivity to the group with a given probability over a certain period of time. This, in practice, may force a limit to the size of the group. Secondly, the nodes may not have interest to participate in the group if the source of the group is too far.

Another problem to face with is related to vehicles exiting from the group. When a vehicle exits from the group it may remains stable with the other vehicles (for example it stops running the application interested in the group but it keeps its velocity), or it may drastically change its velocity and thus become unstable. As long as the MMG remains connected, this fact does not have strong implications on LVMM and it may be easily tolerated. Note also that, if a node exits from a group but it remains stable, it may still provide connectivity to the group and thus participate to routing but not to the application as is briefly discussed in Section 6.

On the other hand, if the MMG becomes disconnected, then it might be impossible using only stable nodes to keep connectivity and the unstable nodes are not reliable to ensure connectivity on the long time. In this case the MMG splits in two parts. In many applications such as distributed cruise control or distributed gaming which are not critical this is not a problem. On the other hand, critical applications which need the vehicles communicate also beyond ad hoc connectivity of the VANET should rely on some other communication infrastructure.
Multicasting - Let us assume that, at a given time $t$, the nodes in the MMG $M$ are $v_1,...,v_L$ and, without loss of generality let us assume that the position of $v_i$ is $x_i$ with $x_i > x_j$ for each $i<j$. Recall also that each node $v_j$ defines a CTPG $C_j$ associated to the MMG $M$ which contains the set of neighbours of $v_j$ belonging to $M$ with their respective position. When a source $v_i$ of multicast traffic wishes to send a multicast packet to $M$, it sends the packets to all nodes belonging to $C_i$. A node $v_j$ in position $x_j$ upon receiving a packet $p$ from a node $v_i$ at position $x_i$, follows rules (a) and (b):

(a) if $x_i < x_j$ and EuclideanDistance($x_i, x_j$)$>R$ forwards packet $p$ to all nodes in $C_j$ with position smaller than $x_i$, (i.e. nodes to the left of $x_i$).

(b) if $x_j < x_i$ and EuclideanDistance($x_i, x_j$)$>R$ forwards packet $p$ to all nodes in $C_i$ with position greater than $x_j$, (i.e. nodes to the right of $x_j$).

Rule (a) establishes that, when a node with position $x_i$ receives a packet $p$ from a vehicle in front of it, then, if and only if it is the leftmost node in the CTPG of $x_i$ it forwards $p$ to all the vehicles in its CTPG that are behind itself, and only to those vehicles; otherwise it does not forward the packet.

Rule (b) establishes that when a node with position $x_i$ receives a packet $p$ from a vehicle behind it, then, if and only if it is the rightmost node in the CTPG of $x_i$ it forwards $p$ to all the vehicles in its CTPG that are in front of itself, and only to those vehicles; otherwise it does not forward the packet.

Note that, because of (a) and (b), packets sent by a member travel only along two directions: toward the leftmost extreme or toward the rightmost extreme. Packets travelling toward the leftmost extreme are always sent by a vehicle to vehicles behind it; they form a left flow of packets. In a left flow each vehicle receives a packet from a vehicle in front of it (at its right) and if necessary forwards that packet to vehicles behind it (at its left). Packets travelling toward the rightmost extreme are always sent by a vehicle to vehicles in front of it; they form a right flow of packets. In a right flow each vehicle receives a packet from a vehicle behind it (at its left) and if necessary forwards that packet to vehicles in front of it (at its right).

5 Conclusions

This paper described a new solution to multicasting in VANETs. The proposed solution is twofold. First, a novel concept of multicasting has been applied to vehicular networks in order to guarantee a minimum degree of availability and stability to communications; second, a middleware, LVMM, was introduced that addresses the tasks of multicast communications and deals with the underlying mobile network. Future work includes the extension of the group admission protocol to include parameters aimed at bounding the size of the group, simulation of LVMM and of VMRP to study their overhead and latency.

References: