Towards Real-Time Middleware for Applications of Vehicular Ad Hoc Networks

René Meier, Barbara Hughes, Raymond Cunningham, and Vinny Cahill

Distributed Systems Group, Department of Computer Science,
Trinity College Dublin, Ireland
{rene.meier, barbara.hughes, raymond.cunningham,
vinny.cahill}@cs.tcd.ie

Abstract. Applications of inter-vehicle and vehicle-to-roadside communication that make use of vehicular ad hoc networks (VANETs) will often require reliable communication that provides guaranteed real-time message propagation. This paper describes an event-based middleware, called RT-STEAM, designed to meet these requirements. Unlike other event systems, RT-STEAM does not rely on a centralized event broker or look-up service while still supporting event channels providing hard real-time event delivery. RT-STEAM event filtering can be based on subject, content and/or proximity. Proximity filters define geographical areas within which events are delivered. To guarantee real-time communication, we exploit proximity-based event propagation to guarantee real-time constraints within the defined proximities only. The proximity within which real-time guarantees are available is adapted to maintain time bounds while allowing changes to membership and topology as is typical of VANETs. This Space-Elastic Model of real-time communication is the first to directly address adaptation in the space domain to guarantee real-time constraints.

1 Introduction

Many Ad hoc wireless networks comprise sets of mobile nodes connected by wireless links that form arbitrary wireless network topologies without the use of any centralized access point or infrastructure. Ad hoc wireless networks are inherently self-creating, self-organizing and self-administering [1].

While most research in ad hoc networks has assumed a random waypoint mobility model in a network of a particular shape, e.g., rectangular, [2], the specific patterns of vehicle movement make inter-vehicle and vehicle-to-roadside networks distinctive. In particular, the potential for high speeds and the limited dimensionality afforded by a confined roadway, differentiates vehicular ad hoc networks (VANETs) from other ad hoc networks.

By enabling inter-vehicle and vehicle-to-roadside communication a broad range of applications in the areas of cooperative driver assistance and mobile information systems are being developed. One of the more sophisticated applications for inter-vehicle communication is the platooning of vehicles [3]. For example, the lead

L. Kutvonen and N. Alonistioti (Eds.): DAIS 2005, LNCS 3543, pp. 1 – 13, 2005.

vehicle in such an application may broadcast sensor information to coordinate movement and potentially reduce the consumption of fuel.

A possible application of vehicle-to-roadside communication is in next generation urban traffic control (UTC) systems. A vehicle (or set of vehicles) approaching a junction could inform a traffic light controller at the junction of its pending arrival. The traffic light controller could then change the traffic light sequence to allow the approaching vehicle to pass through the junction without stopping. When a number of vehicles are approaching the junction, and if the traffic light controller is informed of the presence of the approaching vehicles, then the controller could optimize the traffic flow across the junction. This time constrained communication between vehicles and traffic light controllers should continue reliably during peak times when a large number of vehicles approach the junction from a number of different directions. The potential for contention increases as the number of vehicles communicating with the controller increases.

Both the vehicle platooning and UTC applications require a communication paradigm that supports dynamic changes to the topology of the underlying wireless network, for example to accommodate vehicles joining and leaving a platoon, as well as delivery guarantees for time-critical messages. In the UTC application scenario described above, the traffic light controller needs to be informed of the presence of the approaching vehicle in sufficient time to allow it to change the flow of traffic across the junction.

This paper describes an event-based middleware, called RT-STEAM, designed for ad hoc networks. Unlike other event systems, RT-STEAM does not rely on centralized services while still supporting event channels providing hard real-time event delivery.

RT-STEAM is based on an implicit event model [4] and has been designed for mobile applications and wireless ad hoc networks. RT-STEAM differs from other event services in that it does not rely on the presence of any centralized components, such as event brokers or look-up services, and supports distributed techniques for identifying and delivering events of interest based on location. RT-STEAM supports decentralised approaches for discovering peers, for routing events using a distributed naming scheme, and for event filtering based on combining multiple filters. Filters may be applied to the subject and the content of events, and may be used to define geographical areas within which events are valid. Such proximity-based filtering represents a natural way to filter events of interest in mobile applications.

RT-STEAM provides a programming model based on the concept of event channels. A number of event channel classes with different temporal and reliability attributes are available to integrate real-time support into the event channel model. Depending on the guarantees available from the underlying network, the proximity characteristically associated with an event channel may require adaptation to maintain the required real-time guarantees while allowing changes to membership and topology as is typical of VANETs. This Space-Elastic model is the first to directly address adaptation in the space domain to maintain real-time guarantees.

An underlying assumption of the Space-Elastic model is that a real-time application in a VANET can specify and interpret specified bounds in space (the proximity) where real-time guarantees are critical. This assumption relates to the observations in [3], that the relevance of data to a specific geographical area is one of the key features of an inter-vehicle network and that mission-critical (e.g., emergency braking notification), and non-critical (e.g., weather reports), communication competes for the limited available resources. Thus, Space-Elastic applications must be space-aware, i.e., operate correctly in a dynamic proximity, and information-sensitive, i.e., aware of the criticality of different sources of information (event channels). In the vehicle platooning scenario, the proximity where real-time communication is critical may bound the platoon and, for example, vehicles within the vicinity of the platoon, moving in the same direction. In the UTC scenario described, the critical proximity may be the minimum area within which the controller has sufficient time to change the flow of traffic following communication with an approaching vehicle.

The reminder of this paper is structured as follows: Section 2 introduces RT-STEAM's programming model and architecture. Section 3 presents our Space-Elastic Model of real-time communication and describes its approach to exploiting proximity-based event propagation for maintaining time bounds. Section 4 outlines RT-STEAM's communication architecture. Finally, section 5 concludes this paper and outlines our future work.

2 RT-STEAM

The design of the RT-STEAM architecture is motivated by the hypothesis that there are applications in which mobile components are more likely to interact once they are in close proximity. For example, a vehicle is interested in receiving emergency braking notifications from other vehicles only when these vehicles are within close proximity. Similarly, traffic light controllers are only interested in arrival notifications from vehicles that are located within a certain range of their own locations. This means that the closer event consumers are located to a producer the more likely they are to be interested in the events that it produces. Significantly, this implies that *events are relevant within a certain geographical area surrounding a producer*.

Event Types, Proximities, and Channels. RT-STEAM implements an implicit event model [4] that allows event producers to publish events of a specific *event type* and consumers to subscribe to events of particular event types. Producers may publish events of several event types and consumers may subscribe to one or more event types.

To facilitate the kind of location-aware application described above, RT-STEAM supports a programming model that allows producers to bound the area within which their events are relevant and to define Quality of Service (QoS) attributes describing the real-time constraints of these events. Such a combination of event type, geographical area and QoS is called an *event channel*. Producers *announce* event channels, i.e., they announce the type of event they intend to *raise* together with the geographical area, called the *proximity*, within which events of this type are to be disseminated with the required QoS constraints. Thus, an event channel announcement bounds event propagation to a defined proximity where required QoS constraints are guaranteed. Events are delivered to consumers only if they reside within a proximity where the QoS constraints for the event type are satisfied.

Producers may define proximities independently of their physical transmission range with an underlying group communication system routing event messages from producer to consumer using a multi-hop protocol. Proximities may be of arbitrary shape and may be defined as nested and overlapping areas. Nesting allows a large proximity to contain a smaller proximity subdividing the large area. Fig. 1 depicts two overlapping proximities of different shape and illustrates that multiple consuming and producing entities may reside inside a proximity. These proximities have been associated with events of *type A* and *type B* as well as *QoS A* and *QoS B* respectively. Consequently, consumers handling these event types receive events if they reside inside the appropriate proximity. An example of overlapping proximities might include a vehicle disseminating an emergency braking notification within the vicinity of a traffic light controller that is also receiving arrival notifications from approaching vehicles.

Supporting Mobility. RT-STEAM has been designed to support applications in which application components can be either stationary or mobile and interact based on their geographical location. This implies that the RT-STEAM middleware as well as the entities hosted by a particular machine are aware of their geographical location at any given time. RT-STEAM includes a location service that uses sensor data to compute the current geographical location of its host machine and entities. To suit outdoor applications, for example, in the traffic management domain, RT-STEAM exploits a version of the location service that uses a GPS satellite receiver to provide latitude and longitude coordinates.

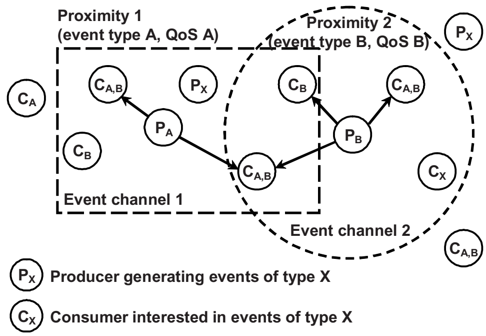

Fig. 1. Disseminating events using event types, proximities and event channels

In addition to supporting stationary and mobile entities RT-STEAM allows proximities to be either stationary or mobile. A stationary proximity is attached to a fixed point in space whereas a mobile proximity is mapped to a moving position represented by the location of a specific mobile producer. Hence, a mobile proximity moves with the location of the producer to which it has been attached. This implies that mobile consumers and producers may be moving within a mobile proximity. For example, a group of platooning vehicles might interact using a proximity that has been defined by the leading vehicle. Such a proximity might be attached to the position of the leader moving with its location.

Subscribing to Event Types. Consumers must subscribe to event types in order to have the middleware deliver subsequent events to them when located inside any proximity where events of this type are raised, until they unsubscribe. A consumer may move from one proximity to another without re-issuing a subscription when entering the new proximity. Thus, subscriptions are persistent and will be applied transparently by the middleware every time a subscriber enters a new proximity. This implies that a subscription to a specific event type applies to all proximities handling these events even though the subscriber may only receive a subset of these events at any time. A single subscription may result in events of a particular event type raised by different producers in multiple proximities and with different QoS being delivered over time. Hence, the set of events received by a subscriber at a certain time depends on its movements as well as on the movements of producers and proximities.

Defining Event Channels. Applications define event channels to specify the functional and non-functional attributes of the events they intend to disseminate. A RT-STEAM event channel is defined by a *subject* and a set of attributes, i.e., its event type, as well as of a proximity description and QoS constraints representing its non-functional attributes. The subject defines the name of a specific event type and the content defines the names and types of a set of associated parameters. RT-STEAM event instances are defined in a similar manner by specifying a subject and content parameter values. Producers and consumers must use a common vocabulary defined by the application to agree on the name of an event type. An event type and an event instance that have the same subject must have an identical content structure, i.e., the set of parameter names and types must be consistent. As described in more detail below, distributed event filters may be applied to the subject, content, and proximity attribute of an event.

Applying Event Notification Filters. RT-STEAM supports three different classes of event filter, namely *subject filters*, *content filters*, and *proximity filters*. These filters may be combined and a particular event is only delivered to a consumer if all filters match. Subject filters match the subject of events allowing a consumer to specify the event type in which it is interested. Content filters contain a filter expression that can be matched against the values of an event's parameters. Content filters are specified using filter expressions describing the constraints of a specific consumer. These filter expressions may contain equality, magnitude and range operators as well as ordering relations. They may include variable, consumer local information. Proximity filters are location filters that define the geographic scope within which events are relevant and correspond to the proximity attribute associated with an event type.

Locating Proximities. Instead of requiring a naming service to locate entities that wish to interact, RT-STEAM provides a discovery service that uses beacons to discover proximities. Once a proximity has been discovered, the associated events will be delivered to subscribers that are located inside the proximity. This service is also responsible for mapping discovered proximities to subscriptions and to the underlying *proximity-based communication groups* [5]. Hence, it causes the middleware to join a proximity-based multicast group of interest, i.e., for which it has either a subscription or an announcement, once the host machine is within the associated proximity-bound and to leave the proximity group upon departure from the area.

RT-STEAM's Distributed Naming Scheme. There are two essential issues that need to be addressed when mapping announcements and subscriptions to proximity groups. Firstly, a naming scheme for uniquely identifying groups is required and secondly, a means for consumers and producers to obtain the correct group identifiers needs to be provided. These issues are traditionally addressed by employing approaches based either on statically generating a fixed number of unique and well-known group identifiers or on using a (centralized) lookup service for generating and retrieving group identifiers. However, neither of these approaches suffices for applications that use VANETs due to their dynamic nature and their inherently distributed infrastructure.

The RT-STEAM event model exploits a decentralized naming scheme in which identifiers representing groups can be computed from event channel descriptions. Each combination of event type, proximity, and QoS is considered to be unique throughout a system assuming that there is no justification for applications to define multiple identical event channels. A textual description of a subject, proximity, and QoS triple is used as input to a hashing algorithm to dynamically generate hash keys that represent identifiers using local rather than global knowledge. Upon discovery of a proximity and the associated event type and QoS, consumers compute the corresponding group identifier if the subject is of interest. This naming scheme allows consumers to subsequently use these identifiers to join groups in which relevant events are disseminated. Moreover, it allows consumers that are not interested in certain events to avoid joining irrelevant groups and consequently, from receiving unwanted events even though they might reside inside the proximity associated with a group.

3 The Space-Elastic Model

The Hard real-time event communication in a highly dynamic VANET is challenging. We scope the problem by exploiting the proximity filters defined by RT-STEAM to reduce the area of a VANET where real-time communication is required to within the defined proximity bounds only. However, the dynamics of a VANET also impacts the real-time guarantees available within the proximity bound. For example, in the vehicle platooning scenario, movement through city environments, with increased obstacles such as tall buildings, may impact the ability to guarantee real-time communication within the entire proximity defined. In this case, we dynamically adapt the proximity bound to maintain the required real-time guarantees. This dynamic proximity, or space-elastic, adaptation is at the core of our Space-Elastic model.

Challenges to Real-Time Communication. Prior to introducing the Space-Elastic model we present an overview of the challenges to hard real-time communication in a dynamic VANET. The challenges to MANETs discussed in [6], e.g., dynamic connectivity, unpredictable latency and limited resource availability are exacerbated in VANETs, for the following reasons.

Limited Communication Duration. In [2], an analytical investigation of topological dynamics based on classical vehicular traffic theory [7], presents valuable information about the relationship between velocity and communication duration. It is shown that in the case of oncoming traffic, and a high average velocity of 130km/h the probability of maintaining connectivity for a communication duration of less than 30s is less than 0.1. Furthermore, as investigated in [3], the transmission range of the radio system significantly influences these duration figures, and that for radio systems supporting a distance less than 500m, the communication duration must be reduced to below 10s. The highly dynamic communication duration expected in a high velocity

VANET, directly impacts, and must be reflected, in the design of medium-access and routing protocols.

Diverse Range of Information. Combining safety, e.g. emergency braking notification, with comfort, e.g. traffic jam notification, in VANETs, leads to a diversity of information criticality and thus required communication guarantees. This diversity influences resource allocation and routing, e.g., prioritized bandwidth usage for emergency notification, and must be reflected in policies available to the middleware protocols.

Impact of Traffic Density on Multi-Hop Connectivity. Both the vehicle platooning and UTC scenarios may require multi-hop connectivity to coordinate driver assistance. As shown in [3], traffic dynamics, in terms of the velocity of vehicles and the density of traffic, impacts the maximum number of hops available for information exchange. For example, 5 hop communication is available with a probability of more than 90% only if the radio system supports at least a range of 1km, and the traffic volume is high. In low-density traffic volumes, with the same radio support, the probability of achieving 5 hops falls significantly to below 20%. To reduce the impact of limited multi-hop connectivity on real-time communication, we investigate dynamically adapting the geographical area within which real-time guarantees are available, and thus within which route discovery is necessary. This is an example of space-elastic adaptation which we discuss in the following sections.

Limitations of Time-Elastic Adaptation. Given the challenges outlined, achieving hard real-time guarantees in a VANET is potentially impossible without restricting the characteristics of the real-time applications, the environment or both. Traditional hard real-time systems have restricted the application, for example, by assuming a known upper bound on the number of participants or assuming known connectivity to intermediate components [8, 9]. These static assumptions are not applicable in a dynamic environment.

The Timely Computing Base (TCB) [10], addresses the problem of achieving synchrony in dynamic environments with uncertain timeliness. In [10], Veríssimo and Casimiro introduce the idea that the probability of achieving real-time communication in dynamic environments changes over time. This observation motivated the specification of the time-elastic class of applications [11]. Time-elastic applications are the subset of real-time applications where it is more beneficial to execute in a degraded mode (e.g., extended time latencies), than to stop, or more critically, have unexpected failures due to a change in the assumed coverage. Inter-vehicle and vehicle-to-roadside applications that require hard real-time guarantees are not time-elastic. Thus the TCB model and time-elastic classification are not applicable. To achieve hard real-time communication in a dynamic VANET, we look instead at the proximity, or space, within which the guarantees are required.

Space-Elastic Adaptation. The timeliness properties of hard real-time applications are invariant. Adapting the terminology of [10], hard real-time applications require guarantees that P(within T from t(e)) = 1, i.e., the probability that a time-dependent execution will complete within (a time interval) T from (start time) t(e), is guaranteed for all timeliness properties. A model of real-time communication must observe this definition of timeliness properties.

In the Space-Elastic model, we look at the space, or proximity, within which real-time properties are required, which is related to the observation in [3], that one of the key features in an inter-vehicle network is the relevance of data to a particular geographical area. For example, in emergency braking notification, the critical proximity where real-time guarantees are required, is within a bound geographical area of the braking vehicle. Beyond this proximity, a "safe distance" exists, e.g., determined by the speed limitations of the road and standard braking distances of vehicles [12], where the notification is not critical and real-time guarantees do not apply. The Space-Elastic Model exploits this rationale to guarantee real-time constraints within specific proximity-bounds only.

Guaranteed hard real-time communication within a specified proximity only, requires a reduction in the scope of the timeliness property specified previously. Assuming space(E, size) is a function that bounds a geographical space of a defined size in relation to entity E, (a mobile or stationary node in a VANET), the timeliness property becomes P(within T from t(e) over space (E, size) only) = 1.

In Fig. 2, the circle represents the proximity-bound for real-time guarantees. The entity, E in this case, may represent a traffic light, in the UTC example, or a mobile vehicle, where the proximity-bound is associated with, and moves with, the vehicle and represents the critical proximity for real-time communication with this vehicle.

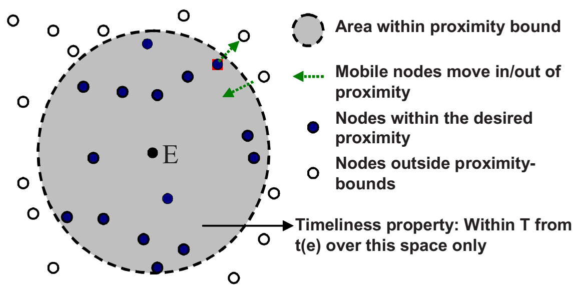

Fig. 2. Timeliness properties within a proximity-bound only

The Space-Elastic model assumes that real-time applications are space-aware. In addition, given the potential for increased network topology dynamics, the limited communication duration, and influence of traffic density on multi-hop route discovery, a further assumption of the Space-Elastic model is that defined proximity bounds are adaptable, e.g., reducing the size, or changing the shape, to guarantee desired real-time communication, regardless of the dynamics of the VANET within the proximity. Thus, Space-elastic real-time applications must tolerate dynamic proximity-bound adaptation. For example in the UTC example, an emergency vehicle (e.g., an ambulance) moving towards a traffic light may have a desired proximity within which real-time communication with the roadside is guaranteed. However, given the challenges to communication in VANETs previously discussed, real-time communication within the desired proximity may not be possible, and a reduced proximity, representing the minimum geographical area, within the desired proximity, where real-time communication is currently achievable, is available only. In this case, it is assumed that the space-elastic application knows the minimum proximity where real-time guarantees are critical and adaptation between the maximum (desired) and minimum (critical) proximity is tolerated. Failure to guarantee real-time communication in the critical proximity is a real-time failure, the consequences of which and the actions to arise are determined by the characteristics of the real-time class. For example, failure to guarantee hard real-time properties in the *critical* region is a critical failure, and transition to a fail-safe or fail-operational [13] state may be possible.

Fig. 3 illustrates proximity adaptation, in this case motivated by route failures in the desired proximity rendering real-time communication no longer possible. In this example, real-time guarantees are available in the area highlighted as the adapted proximity-bound only.

For a space-elastic application to benefit from proximity adaptation the behavior of the space-elastic application must also be adaptable. For example, in vehicle platooning, a desired proximity for inter-vehicle real-time communication for vehicle coordination, may span driving lanes used by vehicles moving in the opposite direction, e.g. for oncoming accident notification [14], to the platoon. In this case, although the desired proximity may contribute rich context about traffic conditions to the platoon, the critical proximity for real-time inter-vehicle communication, bounds the platoon and vehicles within a vicinity of the platoon, e.g. determined by the speed of the platoon, driving in the same direction only. Thus, proximity (or space) adaptation to encompass the critical proximity only is advantageous. In addition, given the reduction in the proximity where real-time guarantees are achievable, the platoon may adapt behavior accordingly, e.g., reducing speed, until such a time when real-time communication within the original desired proximity is achievable.

Space-elastic (Sε) real-time applications are defined as follows: Given an application A, represented by a set of properties P_A, A belongs to the space-elastic class iff none of the duration properties (T) derived from any property P of A, require an invariant space.

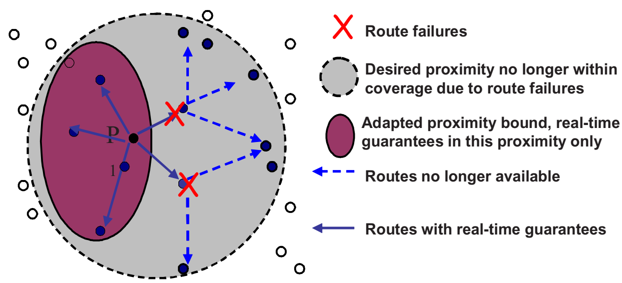

Fig. 3. Adaptive space-elasticity

$\forall A \in S\varepsilon, \forall P \in P_A$: space(A, T) is not invariant

In practical terms, space-elastic applications (A) are those that can accept a bounded proximity (space(A, T)) within which specific real-time application properties (T) are guaranteed. Space-elastic applications must be able to execute correctly in an adaptable proximity or space.

Finally, it is interesting to investigate the role of space-elastic adaptation in a partitioned VANET, i.e., where a subset of nodes (a stationary or mobile participant in a VANET), are no longer contactable i.e., within one-hop or multi-hop connectivity range, for example, due to buildings in a city environment. A partitioned VANET adversely impacts real-time communication, e.g. established routes may no longer be available, new routes may not be discovered due to inaccessible (partitioned) nodes and resources previously reserved for real-time communication, may no longer be available. Furthermore, real-time communication spanning the desired proximity may no longer be possible. Using the space-elastic model, the proximity where real-time properties are guaranteed is adapted to reflect the partition. Real-time communication is guaranteed in a reduced local proximity (at a minimum the critical proximity) only, until such time as a merge occurs with other proximities in the desired proximity.

Fig. 4 (a) illustrates the desired proximity for real-time communication, by an entity, represented by E. Initially real-time event communication is guaranteed within the desired proximity. Fig. 4 (b) represents the network topology after an interval, *T*, where, for example, vehicle movement has caused the VANET, and thus the desired proximity, to partition. The desired proximity has been adapted (Proximity_b), to the area where real-time guarantees are available only. Also illustrated, is the scenario where other local proximities (i.e., Proximity_c), possibly with different real-time constraints, co-exist within the original desired proximity bounds.

A space-elastic real-time application specifies both desired and critical proximity-bounds coupled with associated real-time guarantees, by combin ing RT-STEAM proximity filters with associated event channels, yielding real-time event-based communication in a defined proximity-bound only. To achieve real-time guarantees, the communication architecture used by RT-STEAM combines a proactive, mobility-aware routing and resource reservation protocol, at the network layer, with a predictable time-bounded medium-access control protocol. We outline this communication architecture in the following section.

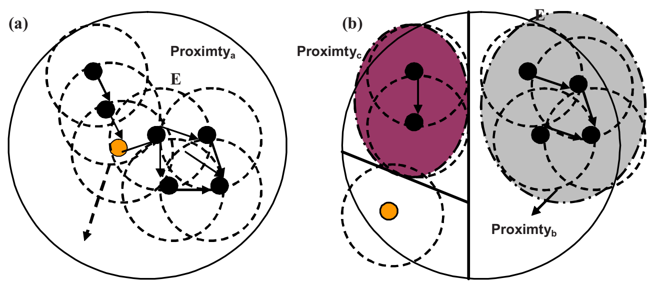

Fig. 4. Dynamic proximity adaptation in a partitioned network

4 Real-Time Communication Architecture

We have completed an implementation and evaluation of RT-STEAM, including proximity-bounds specification, distributed naming etc. for non real-time channels [15] In this section, we present our work in progress on medium-access control and routing protocols to extend our implementation to include real-time channels.

Our medium access control protocol, TBMAC [16] provides, with high probability, time bounded access to the wireless medium to mobile hosts in a multi-hop ad hoc network. The TBMAC protocol is based on time division multiple access with dynamic but predictable slot allocation. TBMAC uses a lightweight atomic multicast protocol to achieve distributed agreement on slot allocation.

To reduce the probability of contention between mobile hosts, the geographical area occupied by the mobile hosts is statically divided into a number of cells in a similar approach to [17]. Each cell is allocated a particular radio channel to use. Each mobile host is required to know its location (using GPS) and from this which cell it is in and what radio channel to use to communicate with other mobile hosts in the cell.

In a similar way to the IEEE 802.11 standard, the TBMAC protocol divides access to the wireless medium within a cell into two distinct time periods:

- Contention Free Period (CFP)
- Contention Period (CP)

Both the CFP and the CP are divided into a set of slots. A CFP followed by a CP constitute a round of the TBMAC protocol. Dividing access to the medium into these two well-known time periods requires the clocks of all the mobile hosts in the network to be synchronized (e.g., each host using a GPS receiver [18]). Once a mobile host has been allocated a CFP slot, it has predictable access to the wireless medium until it leaves the cell or fails. Mobile hosts that do not have allocated CFP slots contend with each other to request CFP slots to be allocated to them in the CP.

The motivation for our routing protocol is to combine proactive routing with mobility-awareness in order to reduce the unpredictability of a dynamic VANET. Our routing and resource reservation protocol (PMRR) attempts to discover and maintain real-time constrained routes within a proximity-bound, e.g., as specified by the Space-Elastic Model. Given the observations in [2], that in “normal” traffic scenarios, the probability of a stable topology varies between 90 and ~100%, and that the topology remains absolutely stable for up to 60s in cases of lower densities and lower relative velocities, our use of proactive routing and resource reservation to guarantee critical communication, appears justified in a VANET.

PMRR executes in two phases. The route discovery phase attempts to proactively discover real-time constrained routes and reserve resources within a defined proximity, providing timely failure notification if routes or resources are not available. The route maintenance phase uses mobility and link quality prediction [19] to assist in proactive route maintenance decisions, e.g., route repair prior to a route break occurring.

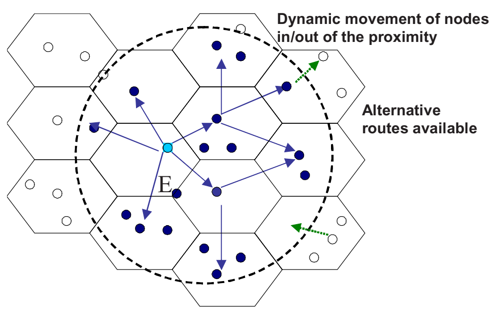

Fig. 5. Proactive route discovery and maintenance

The PMRR maintains one-hop neighbor routing information in routing tables at each hop, e.g., similar to AODV [17], to reduce routing overhead. Furthermore, the routing tables store all alternative routes (i.e., those routes where real-time constraints are guaranteed) from a node, i.e., either the originator or an intermediate node, thus reducing route maintenance latency. Similar to the work of Chen et al. [18], the PMRR utilizes selective probing. However, what is distinctive in this case is that traditional shortest path metrics are substituted by required real-time guarantees.

Fig. 5 illustrates the discovery of proactive routes for real-time communication with E, within a defined proximity-bound. E initially transmits in their allocated slot in the current cell. Using inter-cell slot allocation [16], the transmission is forwarded to neighbouring cells satisfying required real-time constraints only. This process continues for all cells in the proximity-bound, or until a cell is reached where the real-time constraints are no longer guaranteed, e.g., an empty cell.

Depending on the dynamics of the environment within the proximity and the space-elasticity of the application, proximity adaptation may be performed if real-time guarantees cannot be maintained within the entire proximity.

5 Conclusions

We have proposed our middleware for providing real-time communication in VANETs. An implementation and evaluation of our non real-time event-based middleware has been performed. We are currently implementing with the intent of evaluating our real-time middleware. We have currently developed a simulation of TBMAC, a distributed clock synchronisation protocol and a real-time driver for wireless communication using 802.11b.

Acknowledgements. The work described in this paper was partly supported by the Irish Higher Education Authority's Programme for Research in Third Level Institutions cycle 0 (1998-2001) and by the FET Programme of the Commission of the European Union under research contract IST-2000-26031 (CORTEX).

References

[1] S. Chakrabarti and A. Mishra, "QoS Issues in Ad Hoc Wireless Networks," *IEEE Communications Magazine*, vol. 39, pp. 142-148, 2001.

[2] M. Rudack, M. Meincke, and M. Lott, "On the Dynamics of Ad Hoc Networks for Inter Vehicle Communications (IVC)," in *Proceedings of the International Conference on Wireless Networks (ICWN02)*. Las Vegas, USA, 2002.

[3] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer, "Position-Aware Ad Hoc Wireless Networks for Inter-Vehicle Communications: The Fleetnet Project," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*. Long Beach, CA, USA, 2001, pp. 259-262.

[4] R. Meier and V. Cahill, "Taxonomy of Distributed Event-Based Programming Systems," in *Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS'02)*. Vienna, Austria: IEEE Computer Society, 2002, pp. 585-588.

[5] M. O. Killijian, R. Cunningham, R. Meier, L. Mazare, and V. Cahill, "Towards Group Communication for Mobile Participants," in *Proceedings of ACM Workshop on Principles of Mobile Computing (POMC'2001)*. Newport, Rhode Island, USA, 2001, pp. 75-82.

[6] B. Hughes and V. Cahill, "Towards Real-time Event-based Communication in Mobile Ad Hoc Wireless Networks," in *Proceedings of 2nd International Workshop on Real-Time LANS in the Internet Age 2003 (ECRTS/RTLIA03)*. Porto, Portugal, 2003, pp. 77-80.

[7] W. Leutzbach, "Introduction to the Theory of Traffic Flow," *Springer-Verlag*, 1988.

[8] J. Kaiser and M. Mock, "Implementing the Real-Time Publisher/Subscriber Model on the Controller Area Network (CAN)," in *Proceedings of the 2nd International Symposium on Object-oriented Real-time distributed Computing (ISORC99)*. Saint-Malo, France, 1999, pp. 172-181.

[9] Object Management Group, *CORBAservices: Common Object Services Specification - Event Service Specification*: Object Management Group, 1995.

[10] P. Verissimo and A. Casimiro, "The Timely Computing Base Model and Architecture," *IEEE Transactions on Computers - Special Issue on Asynchronous Real-Time Systems*, vol. 51, pp. 916-930, 2002.

[11] A. Casimiro and P. Verissimo, "Using the Timely Computing Base for Dependable QoS Adaptation," in *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems*. New Orleans, USA, 2001, pp. 208-217.

[12] U.S. Department of Transportation, "Amendment for Brake Performance," in Motor Carrier Safety Improvement Act 1999, Federal Motor Carrier Safety Association (FMCSA), 2002.

[13] H. Kopetz and P. Verissimo, *Real-time and Dependability Concepts*, 2nd ed: ACM-Press, Addision-Wesley, 1993.

[14] L. Briesemeister and G. Hommel, "Role-based Multicast in Highly Mobile but Sparsely Connected Ad Hoc Networks," in *Proceedings of Mobihoc 2000*. Boston, USA, 2000.

[15] R. Meier and V. Cahill, "Exploiting Proximity in Event-Based Middleware for Collaborative Mobile Applications," in *Proceedings of the 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'03)*, *LNCS 2893*. Paris, France: Springer-Verlag, 2003, pp. 285-296.

[16] R. Cunningham and V. Cahill, "Time Bounded Medium Access Control for Ad Hoc Networks," in *Proceedings of the Second ACM International Workshop on Principles of Mobile Computing (POMC'02)*. Toulouse, France: ACM Press, 2002, pp. 1-8.

[17] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," IETF Internet Draft June 2002.

[18] S. Chen and K. Nahrstedt, "Distributed Quaility-of-Service Routing in High-Speed Networks Based on Selective Probing," in *Proceedings of the 23rd Conference on Local Computer Networks (LCN98)*. Boston, USA: IEEE Computer Society, 1998.

[19] G. Gaertner and V. Cahill, "Understanding Link Quality in 802.11 Mobile Ad Hoc Networks," *IEEE Internet Computing*, vol. 8, pp. 55-60, 2004.