Geographical routing implementation in NS3

António Fonseca
Inesc-ID
Av. Professor Cavaco Silva
2744-016, Porto Salvo
afonseca@tagus.inesc-id.pt

André Camões
Inesc-ID
Av. Professor Cavaco Silva
2744-016, Porto Salvo
camoes82@gmail.com

Teresa Vazão
Inesc-ID
Av. Professor Cavaco Silva
2744-016, Porto Salvo
tvazao@inesc-id.pt

ABSTRACT

Simulation is an essential tool for computer networks research. In this regard, NS-3 is considered a very promising tool due to its possibility to simulate complex scenarios and integrate with real networks. This simulator was recently released to with a view to replace the popular NS-2 tool, but NS-3 still supports only a limited set of technologies.

Geographical routing protocols appear to be an effective strategy for vehicular networks and many protocols of this type have been developed. Greedy Perimeter Stateless Routing is one of the most representative protocols of this kind of approach since it serves as a base for some of other protocols. In this article we describe and assess our NS-3 implementation of GPSR and the concept of location-service that is needed for such protocols.

Categories and Subject Descriptors

I.6 [**Simulation and Modelling**]: General, Model Development, Model Validation and Analysis; C.2.2 [**Computer Communication Networks**]: Network Protocols—*routing protocols*

Keywords

Implementation, Routing, VANET, simulation, NS-3, GPSR

1. MOTIVATION

In the past years vehicular networks have been a relevant research topic both in scientific and industrial areas, giving origin to consortia such as C2CCC [1] and CarTALK 2000 [18], that solve the problem in a multi-disciplinary manner.

Data traffic routing in Vehicular Ad-hoc Networks (VANET) present great challenges, since traditional MANET solutions are not adequate. There are some proposals with focus on specific layers, mainly in medium access control (MAC) [5] and in routing [15]. Some new recommendations for dedicated short range communications (DSRC) were also proposed [8].

Due to well-defined differences between mobile ad-hoc networks (MANET) and vehicular ad-hoc networks (VANET), MANET routing solutions do not work properly in VANET [4]. Due to that many routing protocols have been proposed for VANET. Simulations have shown that geographical routing protocols outperform traditional MANET protocols [20].

Until recently, NS-2 [13] was one of the simulators more commonly used for this purpose. More recently a new version, NS-3, was released and is undergoing active development. The latest stable version NS-3.12 was released with no geographical routing protocol implemented. In this paper, we describe our implementation of Greedy Perimeter Stateless Routing (GPSR) [7], on NS-3.

The rest of the paper is structured as follows: In section 2 an introduction to geographical routing and then a more detailed introduction to GPSR is given. In section 3 we present the implemented module architecture, section 4 describes the simulations scenario and both greedy and recovery mode tests. In section 5 a conclusion is made.

2. INTRODUCTION

2.1 Geographical routing

Position-based protocols use the position of the destination node to make routing decisions. One of the first position-based routing protocols [19] to be proposed is based on the concept of progress to the destination node. An enhanced solution of the same has been proposed in [2], in which the selection is based on the geographical distance to the destination node. In this class of protocols, it is always assumed that each node has position information available, which is a valid assumption since nowadays most of cars have a GPS device. For these protocols, the node needs to know the location of the destination, which is done using a location service, such as the Grid Location Service (GLS) [14], the Reactive Location Service (RLS) [11] or the Hierarchical Location Service (HLS) [10]. From that moment on, the location of the destination is carried in the packet so that the retransmitting nodes do not need to use the location service again, reducing the overhead.

Experimental evaluations have shown that routing protocols that do not use geographic location, such as topology-based protocols, are not scalable [6]. More recent performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WNS3 2012, March 23, Desenzano del Garda, Italy

Copyright © 2012 ICST 978-1-936968-47-3

DOI 10.4108/icst.simutools.2012.247688

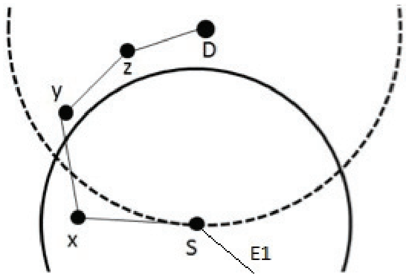

Figure 1: Local-maximum situation

evaluation work [9] shown that neither OLSR nor AODV are able to provide acceptable packet delivery ratio in a VANET scenario

So far, position-based protocols have been demonstrated to be more efficient, as they have less overheads than the existing protocols of the other classes [12].

One of the most simple but yet effective position-based routing protocols is GPSR [7], and many subsequent routing proposals have been made based on GPSR, like GPSR-L [17] or GPCR [15]. Thus, the protocol chosen for our implementation will be GPSR, since this can serve as a base for other VANET protocol implementations.

2.2 Greedy Perimeter Stateless Routing

The routing solutions used by GPSR is known as *greedy forwarding*. In this approach, a node that wants to send a packet chooses its neighbour geographically closer to the destination as its next hop.

A node can calculate this distance locally since *Hello messages* have the position of the nodes and the position of the destination is provided by a location service.

To mitigate the problem of neighbours generating *Hello messages* at the same time, a random jitter was added. The jitter is at best equal to half the *Hello interval*. In this manner, neighbours that happen to send *Hello messages* at the same time that collide will not continue having collisions for subsequent *Hello messages*.

Greedy forwarding strategies can run into a situation called *local maximum* or *local optimum*, in which the sending vehicle is closer to the destination than all of its neighbours, and the destination is not reachable by one hop. However, this does not mean that there is no connectivity to the destination (Figure 1) and so, when a local maximum occurs, a recovery strategy is used.

The recovery-mode strategy used by GPSR is the *right-hand rule*, commonly used to traverse graphs. In [7] the authors refer to this recovery-mode strategy as *Perimeter mode*. As per this rule, if node n receives a packet from edge E_1, it sends the packet through its next edge counter-clockwise about n. The routing protocol switches back to forwarding mode once the forwarding node is closer to the destination than the node that triggers the recovery-mode strategy.

The GPSR algorithm is described using pseudo-code in Algorithm 1, wherein:

- R, is the node receiving a packet p for Destination D.
- N, is the set of one-hop neighbours of R.
- n, is a node of the set N that is used to forward the packet.
- D, is the destination of the packet.

Algorithm 1 GPSR Pseudo-code

```
if ∃ n ∈ N: Distance (n, D) ≤ Distance (R, D) then
   {Greedy forwarding}
   n = Min_Distance(N, D)
   Forward_packet (p, n)
   Return
else
   {local-maximum, use right-hand rule}
   n = Right_Hand_Rule(N)
   Forward_packet (p, n)
   Return
end if
```

3. IMPLEMENTATION OF GPSR IN NS-3

3.1 About NS-3

NS-3 is a discrete event-driven network simulator used for teaching and research. It is being developed under GNU GPLv2 licence and it is freely available for both use and development.

Some of the differences from the well-known NS-2 include [16]:

- Python bindings in replacement of OTcl in the simulation configuration.
- Possibility to analyse simulation traces in Wireshark.
- Possibility to integrate with real systems in order to provide better support for researchers to transition from simulations to real tests.
- Inclusion of well documented attribute system.

Some studies have shown that NS-2 needs more CPU time and more memory than NS-3 [21].

3.2 Module architecture

The main components of our implementation (Figure 2) are the location service, neighbour maintenance and next hop selection. To serve the needs of a location service, we implemented an all-knowing service (*God Location Service*).

GPSR was implemented in the class *ns3::gpsr::RoutingProtocol* and extends the abstract class *ns3::Ipv4RoutingProtocol*. Classes *ns3::gpsr::GpsrHeader* and *ns3::gpsr::HelloHeader* extend from *ns3::Header*. We made the class *ns3::gpsr::PositionTable* to store information about neighbours. To store the packets queued we use the class *ns3::gpsr::RequestQueue*. Each entry in the queue is an object of the class *ns3::gpsr::QueueEntry*. An abstract class (*ns3::LocationService*) was created for the location service so that any geographical routing algorithm can use any available location service implementation.

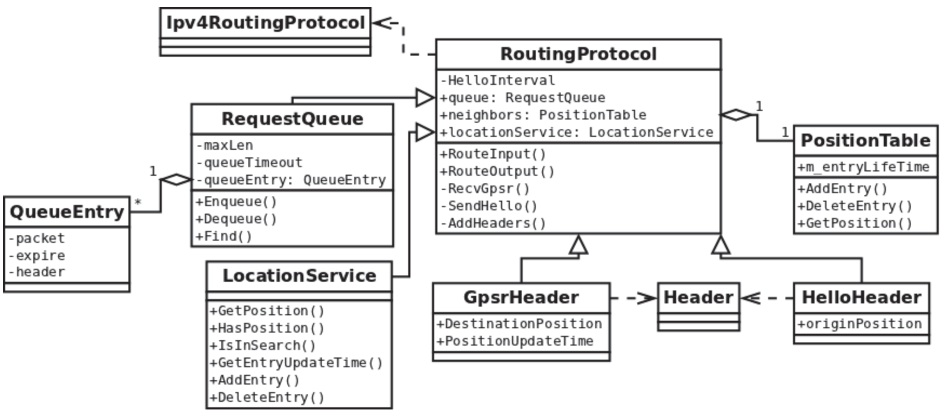

Figure 2: GPSR class diagram

PositionTable. The *PositionTable* has one entry for each neighbour. Each entry has the neighbour identification, its coordinates, and the timestamp at which the last *Hello message* was received.

Header. There are two types of message headers: *Hello messages* to keep the neighbour list up-to-date, and *GPSR messages* to convey the position of the destination to nodes that will forward the message.

Hello messages (*ns3::gpsr::HelloHeader*) are 8 Bytes long. The message contains the position of the node that sent the message. The time interval for these messages is parameterizable and the default value is 1 second since this was the value chosen by the authors in [7]. Thus, along with the random jitter, the time between two *Hello messages* is a random time between 0,5 and 1,5 seconds.

GPSR messages (*ns3::gpsr::GpsrHeader*) is 16 Bytes long, and includes the position of the destination and a timestamp to check for the position's freshness.

RequestQueue. When the route is asked for a given packet and the information is not available yet, either because the location of the destination is still unknown or because at the moment no neighbours are available, the packet is stored in the *RequestQueue* which is periodically reviewed to check if any packet has information to answer the pending requests.

For now, the only location service implemented was an all-knowing one, but an abstract class was made to permit the implementation of more relevant location services. The abstract class was made with interfaces to fit major location services at the moment. This way when more location services are implemented the interface and already existing location services don't have to be changed.

4. SIMULATION

Functional validity tests were made with the parameterisation present in Table 1. To simplify the analysis, a simple topology was used, composed by a 10x10 nodes grid, 100m apart. The transmission range was set to 100m so that nodes on the diagonal are not neighbours. MAC 802.11 DCF was used without RTS/CTS. The propagation model used was free-space propagation model. One data session was created between the top left and the bottom right nodes using Constant Bit rate (CBR) traffic that generates a packet that is retransmitted by the receiver to the sender.

Table 1: Simulation parameters

Parameter	value
Simulator	NS-3.12
Simulation time	30 s
Number of nodes	100
Node topology	10x10 grid
Node spacing	100 m
Transmission rate	2 pkt/s
Packet size	128 b
Hello rate	1 pkt/s
MAC layer	802.11 DCF
Propagation model	Free space propagation
Transmission range	100 m

Different tests were made to test greedy forwarding mode and recovery mode in which the right-hand rule is used. In the following subsections, the tests for both modes will be explained.

4.1 Greedy mode tests

To test the movement, once the mobility models for VANET are not yet wide developed, some nodes positions were set to out-of-reach places. The test has only one communicating pair: the top-left node which talks to the bottom-right node. In the middle of the simulation, the nodes from 4D to 7G (Fig. 3) disappear from the range of all the other nodes to test if the protocol adapts to the change in the topology. Fig. 3 show the paths used by the communication. Up to the point that the topology changes, the communication pass through the green dashed line. After the topological change, the communications go through the arched red line. The shaded area represents the nodes that get out of reach.

To assess if the protocol is working properly one must study

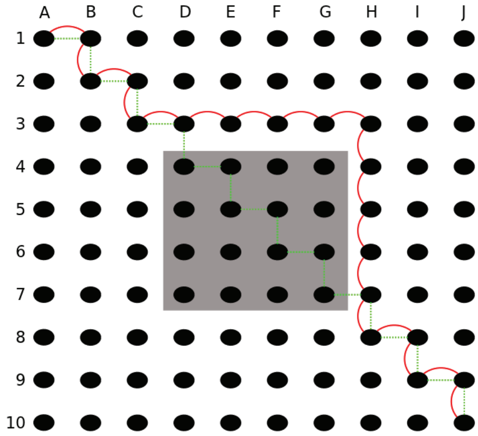

Figure 3: Simulation grid with communications

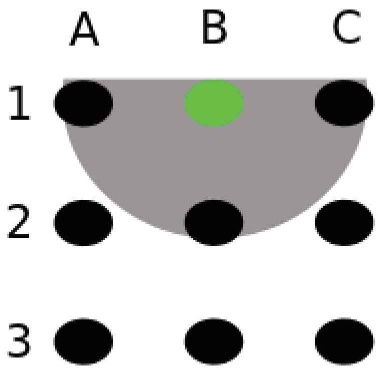

Figure 4: Choice of next hop

if the path used was the correct one. Analysing Algorithm 1, one can see that the sender node (first row, first column) chooses the neighbour closest to the destination. The neighbours are the nodes in the first row, second column and the node in the second row, first column. Those two nodes are at the exact same distance from the destination and so the one that appears first in the neighbour table will be chosen, since the table is ordered by the nodes ID, the node on the left of the sender node was chosen as expected.

Now the node that has the packet has three neighbours, and the one that is closer to the destination is the node below him. This is represented in Fig. 4 where the node with the packet is represented in green while the neighbours are within the shadowed area. The process would repeat and so the path would be left, down and repeat this order until the destination which is what happens in the simulations.

In Table 2 are presented the values of the number of hops, Round Trip Time (RTT), overhead per node and Packet Delivery Ratio (PDR) of the simulation in the two scenarios, with and without movement.

Table 2: Simulation results

Metrics	Without movement	With movement
Hops	18	18
RTT	55ms	55ms
PDR	100%	92.86%
Overhead/node	640-896b/s	640-896b/s

As can be verified in Fig. 4, the number of hops is 18 before and after the nodes move. Thus the average RTT should be the same amongst the tests which are compared by the results. The PDR decreases in the tests with movement since the nodes that change position, move with speeds such that the packets that were with those nodes when they moved were lost/dropped. The overhead was the expected value since each node sends one hello packet per second (640 bits) and two messages per second which include a position header (128 bits). The nodes that do not participate in the transmission only have 640 b/s of overhead since they do not send the two position headers each second.

Since no path is established when the nodes leave there is no adaptation time needed. A re-transmission may occur if the hello has not expired when trying to send a packet in that vicinity. Since the nodes instantly move to somewhere very far the packets that were held by those nodes at the moment where lost but this is not realistic so no packet should be lost if there is enough connectivity from the nodes that are moving away.

4.2 Recovery mode tests

Some of the tests made to recovery-mode were meant to debug the implementation and for a simple analysis of how it was working. They will thus not be described in this paper. A more complex scenario is presented which uses a topology represented in Figure 5. The round green arrows represent communication in greedy mode whilst straight red arrows represent communications in recovery mode. In this test the message is generated in the node at position 3A and sent to the node in position 4D. In this test we make sure that the algorithm works correctly when:

- Recovery mode is working correctly when entering in a random node.
- Recovery mode is working correctly when entering in the sender node, since the reply comes from a node that will enter in recovery-mode.
- A packet passes through a node for the second time due to being in recovery mode.

Apart from the topology the parameters for this test are equal to the ones described in the greedy mode tests section which are presented in Table 1.

As can be seen in Figure 5 the sender node forwards the packet in greedy mode to node 3B, this node has no neighbour closer to the destination than himself, so enters recovery mode and sends the packet back to 3A which sends to the first neighbour counter clockwise from the edge to 3B,

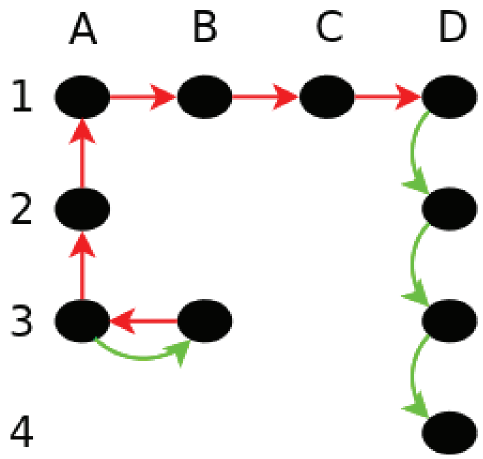

Figure 5: Recovery-mode test scenario

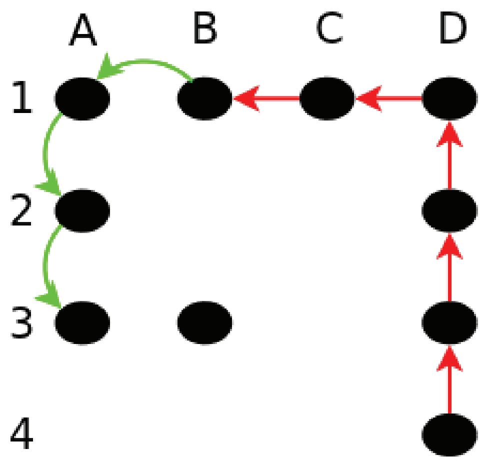

Figure 6: Recovery-mode test scenario response path

which is node 2A. All node send to the first node counter clockwise from the edge that the packet traversed until it reaches node 1D, which is closer to the destination than node 3A (in which the packet entered in recovery mode) so the packet returns to greedy mode. Figure 6 shows the hops used by the response packets. Since all of node 4D's neighbours (node 3D) are further away from the destination than himself, the packet enters in recovery mode. The packet follows the only available path until node 1B, which is closer to the destination than where the packet entered recovery mode, and it goes to greedy mode again until it arrives the destination, node 3A.

5. CONCLUSION

In this article we presented the implementation of GPSR routing protocol in NS-3 and the introduction to the location service concept in the simulator. We analysed our implementation and concluded that it works as expected, since the paths were correctly chosen according to the protocol specifications, both in greedy and recovery mode.

The implementation has already been submitted for review towards inclusion in NS-3 [3].

6. ACKNOWLEDGEMENTS

This work was supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds.

7. REFERENCES

[1] C2CCC. Car2car communication consortium. http://www.car-to-car.org.

[2] G. G. Finn. Routing and Addressing Problems in Large Metropolitan-Scale Internetworks. Technical Report ISI/RR-87-180, Information Sciences Institute, Mars 1987.

[3] A. Fonseca. Code review. http://codereview.appspot.com/5401042.

[4] A. Fonseca, A. Varela, and T. Vazão. An all-ip vanet architecture supported on r2v and v2v communication through 802.11. June 2010.

[5] Y. Inoue and M. Nakagawa. Mac protocol for inter-vehicle communication network using spread spectrum technique. In *Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994*, pages 149 –152, August 1994.

[6] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communications*, 8:48–57, 1999.

[7] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM.

[8] A. Khan, S. Sadhu, and M. Yeleswarapu. A comparative analysis of dsrc and 802.11 over vehicular ad hoc networks.

[9] I. Khan and A. Qayyum. Performance evaluation of aodv and olsr in highly fading vehicular ad hoc network environments. In *Multitopic Conference, 2009. INMIC 2009. IEEE 13th International*, pages 1 –5, dec. 2009.

[10] W. Kiess, H. Füßler, J. Widmer, and M. Mauve. Hierarchical location service for mobile ad-hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 8(4):47–58, Oct. 2004.

[11] M. Ksemann, M. Kasemann, H. Fler, H. Fuler, H. Hartenstein, H. Hartenstein, M. Mauve, and M. Mauve. A reactive location service for mobile ad hoc networks, 2002.

[12] K. C. Lee, U. Lee, and M. Gerla. *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*, chapter 8, pages 149–170. Mohamed Watfa, University of Wollongong, UAE, 2010.

[13] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. In *Vehicular Technology Magazine, IEEE*, volume 2, pages 12–22, 2007.

[14] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, New York, NY, USA, 2000. ACM.

[15] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein. Geographic routing in city scenarios. In *SIGMOBILE Mob. Comput. Commun. Rev.*, volume 9, pages 69–72, New York, NY, USA, 2005. ACM.
[16] NS-3. Ns-3. http://www.nsnam.org/.
[17] S. Rao, M. Pai, M. Boussedjra, and J. Mouzna. Gpsr-l: Greedy perimeter stateless routing with lifetime for vanets. In *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, pages 299 –304, oct. 2008.
[18] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz. Cartalk 2000: safe and comfortable driving based upon inter-vehicle-communication. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 545–550 vol.2, June 2002.
[19] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *Communications, IEEE Transactions on*, 32(3):246 – 257, Mar. 1984.
[20] C. A. T. H. Tee and A. C. R. Lee. Survey of position based routing for inter vehicle communication system. In *Distributed Framework and Applications, 2008. DFmA 2008. First International Conference on*, pages 174–182, 2008.
[21] E. Weingartner, H. vom Lehn, and K. Wehrle. A performance comparison of recent network simulators. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1 –5, june 2009.