COLLECTION OF DEDICATED INFORMATION IN VEHICULAR AD HOC NETWORKS

Murat Caliskan[1]*, Bernd Rech[1], Andreas Lübke[1], Martin Mauve[2]
1. Volkswagen AG / Group Research Electronics
Letter Box 1776/0
D-38436 Wolfsburg, Germany
Phone: + 49-5361-9 29425 – Fax: + 49-5361-9 72837
murat.caliskan @volkswagen.de
2. University of Düsseldorf, Germany
Institute of Computer Networks and Communication Systems

ABSTRACT

In recent years, the application of mobile ad hoc networking among vehicles has become an important research field for the automotive industry. Vehicular Ad Hoc Networks (VANETs) are formed by vehicles that are equipped with wireless communication devices based on the standard IEEE 802.11 Wireless-LAN. Embedding mobile ad hoc network devices into vehicles will pave the way for future Advanced Driver Assistance Systems. These could be used to improve road safety and comfort for the driver as well as the passengers. Consider the situation in which you drive into an unfamiliar city. Your vehicle navigation system checks the traffic conditions on-demand and directs you quickly through the city avoiding busy roads and road blocks. In the meanwhile your vehicle navigation system requests parking information in the vicinity of your destination, so that you can drive straight to a free parking space. In this paper we discuss the feasibility of dedicated information collection through parameterized queries in VANETs. As an application example, we analyze the use of personal messages to guide vehicles towards suitable parking spaces available nearest to their destination area. The evaluation shows that it is possible to use a VANET to collect on-demand information from a target area within a city. To prove this a search strategy is proposed, implemented and evaluated in an interactive and realistic simulation environment.

KEYWORDS

Vehicle-to-Vehicle communication, Wireless LAN IEEE 802.11, Applications for VANETs, Simulation of VANETs, Travelling Salesman Problem

INTRODUCTION

Targeted information search in VANETs

In future Vehicular Ad Hoc Networks (VANETs) each vehicle will send inquiries actively to other vehicles, in order to receive individual information about a specific and pre-defined region. Multifarious inquiries possessing special parameters of the sender, which give them a personalized character, are possible. Contents of the inquiries may change in accordance with the driving situation. Hence each vehicle is expected to send different and unpredictable inquiries to the VANET tailored to seek special information.

Examples of inquiries triggered by the vehicle or the driver are: the search for a suitable service station along the travel route, finding out traffic conditions along the travel route, search for information about road conditions in a target area, search for offered leisure activities in a city (e.g. theatre, cinema, social events) or the search for free parking spaces in a target area. Especially “free parking space search” is an application that requires many personalized parameters from the driver, e.g. target road, tolerable walking distance from exact destination and specific wishes like reserved parking spaces for handicapped persons or those with cheapest parking fare. In short, the target area, the kind of information sought and even the search parameters of the information are individually different for every vehicle.

Existing mechanisms for decentralized information dissemination in VANETs cannot be utilized for personal inquiries triggered by the driver. This is because existing mechanisms are based on broadcast messaging and the distributed information (e.g. traffic information) is of general interest. No personalized character could thus be embedded into the broadcasted data packets. For these reasons new approaches are needed to solve the problem of targeted information search in VANETs, i.e. sending personal inquiries into a pre-defined target area, searching and collecting the requested information in the target area and finally receiving back the results of the inquiries.

The question “Whether a free parking space exists in a parking area or not?” can be answered with the help of multi-hop Vehicle-to-Vehicle communication without the need of additional message-relay infrastructure. Such an application based on Vehicle-to-Vehicle communication would extend the functionality of today's navigation systems and fill the gap between the vehicle navigation system and the autonomously parking vehicle. Hence investigations in this paper focus on this reference problem.

Reference Problem: Search for free parking spaces in urban traffic conditions

Searching for free parking spaces in urban traffic conditions is becoming a serious traffic problem. [1] provides results of a study regarding the free parking spaces problem in the district Schwabing of Munich. This study shows the extent of the annual damage resulting from searching parking space traffic (values given per year for Schwabing):

- Total of 20 million Euros economical damage
- 3.5 million Euros for gasoline and diesel, which are wasted for search for free parking spaces
- 150 000 hours of waiting time
- The proportion of park search traffic amounts to 44% of the entire traffic, i.e. nearly every second vehicle is in search for a free parking space. (The statistically determined vehicle traffic in Schwabing is about 80 000 km per day.)

Projected on large cities in Germany with similar districts, a total economical damage from two to five billion Euro per year is estimated.

RELATED WORK

Investigated applications with focus on connecting a vehicle with its surroundings are based on centralized approaches. Communication among vehicles as well as the communication of a vehicle with its surroundings is possible via a roadside infrastructure-network. For example in classical Traffic-Travel-Information-Systems (TTI) [2], the information is collected over sensors installed on road sides, evaluated in a central station and distributed via a dedicated radio channel (e.g. Traffic Message Channel in Europe (TMC)) back to the vehicles. Other applications like Internet-Browsing, Short Message Service (SMS), E-mail or Navigation-systems available in vehicles are also using centralized networks like Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), Universal Mobile

Telecommunications System (UMTS) or satellite systems like Global Positioning System (GPS). Since these applications are organized in a centralised manner, they cannot be utilized as such in a decentralized vehicular ad hoc network.

Even though the application of digital radio techniques for ad hoc networking among vehicles in a decentralized way is quite new, some applications already exploit the benefits of VANETs. One such application using the IEEE 802.11 Wireless-LAN standard is the Self Organizing Traffic Information System (SOTIS) [2]. In SOTIS vehicles broadcast periodically messages into their direct neighbourhood. Each vehicle combines the information contained in the received message with data from its knowledge-base and generates from this a new message, and broadcasts it. Although applications like SOTIS follow a decentralized approach for information collection, they are targeted to disseminate information which is of general interest. Since each node broadcasts periodically pre-defined information and no network-wide routing possibilities are implemented, neither personalized inquiries can be generated nor can a certain region or a certain vehicle be addressed with this kind of applications. Hence, we present a first concept for network-wide personalized queries to overcome the limitations in applications like SOTIS.

CONCEPT FOR TARGETED INFORMATION SEARCH IN VANETS

The presented concept for information collection in a pre-defined area divides the problem into 4 steps. Below these steps are introduced and the applicability of existing approaches for each step is discussed in detail.

Step 1: Determining the search area and computing the optimal route for the data packet

The road network comprises roads and junctions. Depicting this road network as a graph, the roads represent edges and the junctions represent vertices. Since in the “worst case” the complete target search area must be scanned by the data packet, graph algorithms described in [6], [7] can be used to describe the search area as well as to determine the optimal data packet route.

Given that two vehicles in communication range form a part of a possible route that the data packet follows, the optimization process can be described as follows: Pre-calculate the road sequence with as much as possible forwarding vehicles for the data packet and maximize the possibility that the data packet can be forwarded through all roads in the given network.

So far, no investigations are known for searching the complete target area as well as the determination of the optimal route for a data packet in VANETs. Therefore the target area search problem is dealt-with in more detail in this paper. For a first approach the Travelling-Salesman Problem (TSP) which pre-calculates the optimal search route for the data packet is applied. The introduced TSP-Based Information Collection Algorithm (TIC) uses traffic data available offline for this pre-calculation.

Step 2: Forwarding the data packet up to the borders of the target search area

Step 3: Perform the requested search within the target area

Step 4: Re-locate the inquiring vehicle after the search is accomplished

In order to accomplish the steps 2-4 Unicast-Mechanisms in combination with a location lookup service to re-locate the inquiring vehicle can be used. Unicast-Mechanisms, particularly position based Unicast describes the transmission of a message from a sender to a known receiver [3], [4], [5].

Additionally to the steps above, a broadcasting mechanism to improve the performance of targeted information search can be implemented. Broadcasting data packets in a network means that data packets are transferred from one point to all participants. Broadcast in

Vehicle-to-Vehicle networks is used in two variants. One variant is flooding a region with messages through Multihop-Broadcasting and the other one is beaconing which means 1-Hop broadcasting. In the presented reference problem a 1-Hop-Broadcast-Mechanism can be used to generate pre-information about the target search area. The working principle of such a mechanism is similar to the one described in [2].
In this paper the investigations are limited to problems on data communication level with respect to the free parking space search problem in VANETs. The emphasis of the work is to answer from a data communication point of view, whether or not and in which quality a targeted information search can be realized within VANETs.
For this reason all further considerations presume that specific problems not related to data communication concerning targeted information search are solved.

Hence it is presumed that:

- Vehicle-to-Vehicle communication radio technology is deployed into vehicles allowing vehicles to communicate among each other.
- Vehicles can recognize free parking spaces in their direct environment or at roadsides. The recognition can be done by using Radio Frequency Identification Tags (RF-ID), pressure sensors or image recognition mechanisms or by any other technology.
- Each vehicle possesses a navigation system or a digital map and vehicles can code the coordinates of the detected free parking spaces into the data packet, so that other vehicles can drive to these coordinates.

TSP-BASED INFORMATION COLLECTION ALGORITHM

We use the well known optimization problem “Travelling-Salesman-Problem (TSP)” [6], [7]. TSP appears to be a good candidate for an exemplarily first approach. The TSP-Based Information Collection (TIC) Algorithm, instead of visiting the roads within a search area in an undefined order, a data packet will traverse the roads according an optimal (or near optimal) route. The proceedings of the algorithm and the calculation of the optimal route are discussed in the following paragraphs. Then how the traversing costs of edges are determined is explained. Finally needed stop criteria for a successful search are presented.
In the given solution for determining the optimal route, it is to be noticed that TSP is NP-Complete [6]. NP-Completeness is described in detail in [6] and means that in general it is not possible to find algorithms which can solve a given problem instance within acceptable time and memory boundaries. For this reason it must be examined whether the storage capacity and computing power existing in vehicles are sufficient for executing such kind of algorithms, i.e. whether or not the calculation can be done in acceptable time. If not appropriate heuristics guaranteeing low computation time and memory consumption have to be applied.

Methodology

The Travelling-Salesman-Problem (TSP) requires finding the shortest path visiting each of a given set of nodes and returning to the starting node. The shortest path describes the optimal route through the given network. The optimization criteria are costs assigned to each edge in the given graph. The calculation of the optimal route is done by minimizing the total cost value based on pre-assigned edge costs. In a given road network all roads should be visited at least once. However TSP visits all nodes in a given graph, a transformation of the given road network into its TSP-Representation must be done before TSP can run. For this a new graph is generated, roads are transformed into vertices and junctions are transformed into edges. Formally, the given road network within the search range is considered as a graph G = (V, E), the crossings are the vertices V, and the roads E are the edges of the graph G.
Next, we illustrate the problem of determining the optimal route for the data packet under urban traffic conditions using the TSP-Problem. For that, the problem of finding an optimal

route for the data packet has to be transformed into a TSP compatible representation. The transformation process comprises two steps. First, the transformation of the road network: The graph G is converted into G ' = (V', E') with V' = E and E' = V from the graph G = (V, E), i.e. the edges (roads) of the origin graph (road network) become vertices and the origin vertices (crossings) become edges. Second, the assignment of cost values to roads: In the road network the cost for each edge is calculated through an assigned quality value RQ(r) (Road Quality Value for Road r) for each road. In the TSP-Representation of the road network, this value is weighted by a special metric and assigned to edges in the TSP-Graph.

Transformation: Road-Network ↔ TSP-Graph

The following rules define the transformation function and are applied gradually to all vertices and edges of the road network. The transformation rules are:

1. Each road r_i within the search area becomes a vertex v_i.
2. Each possible transition from road i to road j over a crossing becomes the edge i⟶j in the TSP-Graph.

The target road, the route calculated by the navigation system and the search area are marked. **Figure 1** shows the road network with enumerated roads. After applying the presented rules the enumerated road network changes into the TSP-Graph representation in **Figure 2**.

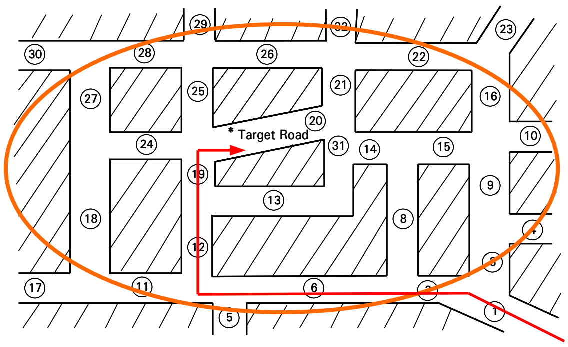

Figure 1 – Enumerated Road Network

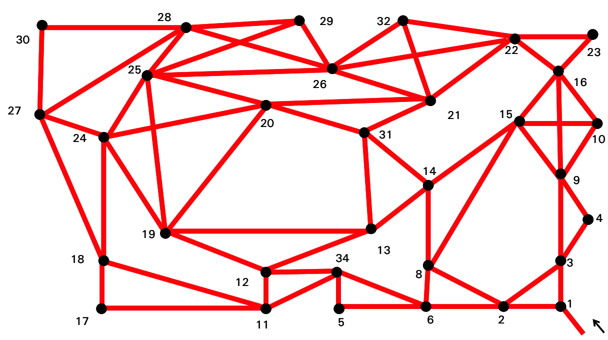

Figure 2 – Transformed Road Network

A simple example of the transformation is given in the following section.

Calculation of the Optimal Route

A data packet should successfully go through its search area. Therefore an optimal route for the data packet must first be determined. In the calculation process actual road conditions and road characteristics have to be taken into account.

The description of an optimal route for a data packet is:

- The optimal route is given by a path in the search area. This is described through a sequence of edges, i.e. roads r_1, $r_{2...}$ $r_{n.}$
- All roads within the search area are visited by the query packet.
- If possible, no road is visited twice.
- The search begins with the most suitable road for parking and ends with the less suitable ones.

The search is successful when the inquiry coded into the packet could be answered. This happens when either the maximum number of requested free parking spaces is found, or the complete target area is scanned but insufficient numbers of parking spaces are found. Otherwise the inquiry is not successful (e.g. because of exceeding the maximum allowed time to reply) and the packet is rejected.

In the defined search area, every road needs to be visited at least once. This problem is very similar to the TSP problem, which describes that the shortest tour in a network is to be found

that visits every vertex of a graph at least once. The optimization criteria are costs assigned to every edge.
The difference with the TSP problem is that the Information Collection Algorithm tries to visit every road or edge of the network and the TSP problem visits every junction or vertex. Therefore the graph is transformed into a line graph, which describes the TSP-Graph representation of the road network. In this graph every vertex represents an edge or road from the original network. Weights which represent cost factors are assigned to the edges to reflect the quality of the corresponding roads. Heuristic methods are used to calculate a sequence of roads, which are to be traversed by the data packet. Because routing decisions must be made on junctions and not on roads, the resulted tour is projected on the original graph to extract a list of junctions. This process is shown in the following steps:

Step 1: A graph representation of the road network is constructed:

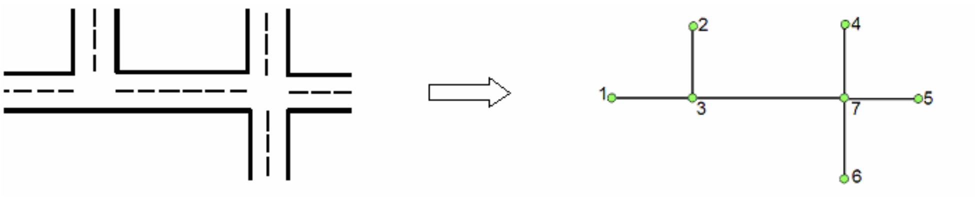

Figure 3 - Transformation from an example road network to a graph representation

Step 2: The Graph representation of a road map is then converted into a line graph. Every edge in the original graph becomes a junction in the line graph. An edge is added between two vertices (roads) when in the original graph a vertex (junction) exists between the two edges (roads).
Step 3: Cost values are assigned to every edge in the transformed graph. These values reflect the costs to traverse from one road to another.

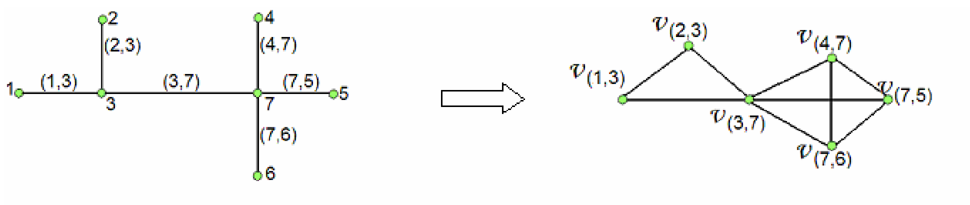

Figure 4 - Transformation from a road graph to a line graph

Step 4: Since the TSP-Problem requires a fully connected input graph, extra edges are added to the transformed graph to make it fully connected. The cost values of the new edges represent the minimum costs of the path between the new edges.
Step 5: The Christofides heuristic [7] is applied on the transformed graph. This results in a Hamiltonian tour[1], which contains a sequence of roads. An example could be $v(1,3)$, $v(2,3)$, $v(3,7)$, $v(4,7)$, $v(7,5)$, $v(7,6)$, $v(3,7)$, $v(1,3)$.

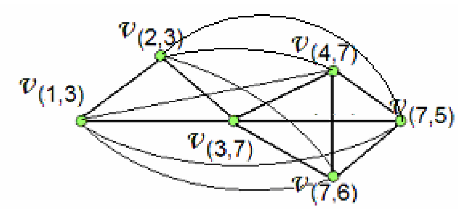

Figure 5 - Fully connected line graph

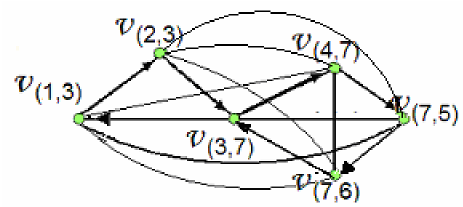

Figure 6 - Example tour

Step 6: The sequence of roads is then reflected on the original graph. When the route traverses one of the added edges from step 4, these have to be resolved using a shortest path algorithm

[1] A Hamiltonian cycle is a tour through a graph, which visits every node exactly once.

[6]. For example, when the tour traverses from $v(2,3)$ to $v(7,6)$ the edge $e(v(2,3),v(7,6))$ could be resolved to $v(4,7)$, $v(3,7)$, $v(4,7)$, $v(7,6)$,(in this case very unlikely). The example mentioned in step 5, would result in the following sequence of junctions: 1, 3, 2, 3, 7, 4, 7, 5, 7, 6, 7, 3 **(Figure 7)**.

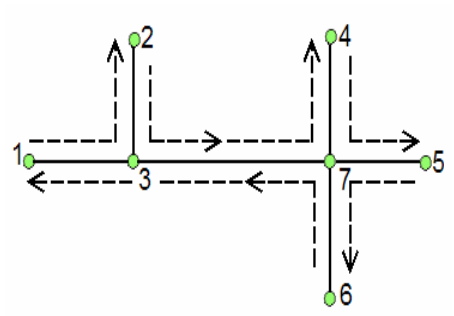

Figure 7 - Example TSP tour reflected on road map

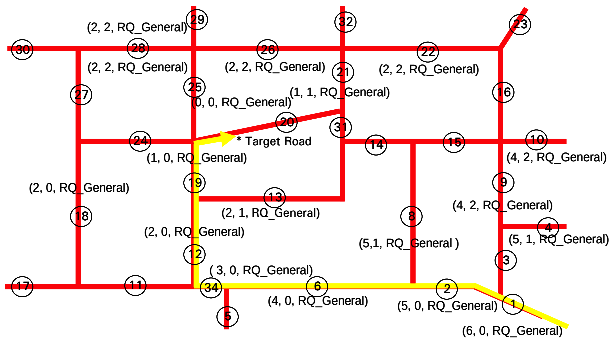

Figure 8 – Road Weightings

Determination of the Weighting Function

To be able to calculate an optimal route, costs must be assigned to edges. In the road graph, the assigned cost values represent the quality of the road. “Good roads” are roads that are likely to be traversed easily by a data packet. Factors that are used to determine the weighting for each road are:

- Average number of vehicles: roads with high traffic density have more forwarding opportunities.
- Length of a road: the distance a data packet can travel from one node to another is bound by the nodes radio ranges. Longer roads result in more hops.
- Type and relative position: It can be expected that dead ends and one way roads offer only few forwarding possibilities. Roads relatively close to the target road are attractive and worth to be traversed by the data packet more than once.
- Speed limit: higher relative velocities between nodes reduce the performance of an underlying routing protocol.

The quality value of a given road RQ(r) expresses the relevance of the road for the search in the target area and is transformed to a certain cost value. Formally the value RQ(r) is described by the triple i.e. RQ(r) = (i, j, RQ_General), where:

- i: relative position of the road to the travel route
- j: relative position to the target road
- RQ_General: Estimated value for the road quality. This value comprises parameters e.g. : Number of estimated vehicles in this road, number of available parking spaces, forwarding quality of the road dependent on traffic density, mobility of nodes, traffic lights, road capacity, channel capacity of the road etc.. Values for these parameters are based on preceding observations or measurements.

In **Figure 8**, the quality RQ(r) of some roads in the example road network is deduced based on above explanations. For example, the value RQ(26) = (2, 2, RQ_General) reflects road quality parameters for road 26. The road quality parameters for each road RQ(r) are mapped to edge costs during the optimization process. The cost value derived by the quality value of roads RQ(r) in the given road network is used to assign weights to edges in the TSP-Graph.

After generating the input for the TSP-Problem one proceeds as follows:

1. In the TSP-Graph a path with minimum total costs min(C_{total}) for the vertices 1, 2..., n is determined, i.e. the defined TSP-Instance is solved.

2. The solution for the TSP-Graph with input TSP(vertices, edges, cost function) is transformed back to the origin graph.
3. The resulting path represents the optimal route for the given road network.
4. The resulting starting node corresponds to the entrance road for the vehicle in the target search area respectively in the origin road network.

The values for the parameters are based on estimated values and vague data. These values must be available before the calculation of the optimal route in the inquiring vehicle starts. The needed data could be firmly coded on the Navigation-DVD or can be distributed a priori via broadcast-communication medium to all vehicles. In this case the scoring of the received data is done by the receiving vehicles and is then used to assign costs to the roads in the road network.

Traversal of the search area

After an optimal route is calculated, i.e. order and frequency of roads to be visited by the data packet are obtained; this route is packed into a data packet and sent towards the search area. This route is used by vehicles receiving the data packet in order to decide the direction in which the data packet should be forwarded within the next step.

In the search area the packet will traverse the roads by using the vehicles as relays, until a stop criterion is fulfilled. One stop criterion could be when the requested information is collected. Another stop criterion could be that the packet has completed a full tour. The latter is the worst case scenario. When the stop criterion is fulfilled, the data packet returns to the inquiring vehicle. This can be done by existing mobile ad hoc routing protocols such as GPSR[8] or DSR[9] in combination with a location lookup service for the inquiring vehicle.

Rules and Break Criteria of the Algorithm

In order to terminate, the algorithm must obey certain rules. First, the algorithm has to ensure that the data packet is forwarded between vehicles, i.e. vehicles know that other vehicles are in their communication range. Second, the algorithm has to ensure that the data packet traverses the optimal route systematically. For example, if a data packet moves out of the pre-calculated optimal route, a next optimal candidate should be determined previously and the packet should be forwarded to this vehicle in time.

In case, the inquiring vehicle does not get an answer within a pre-defined time frame, a new inquiry should be generated. In such a case the data packet with the old inquiry is rejected. If additional break criteria are coded into the data packet (e.g. maximum number of parking spaces to be searched) the data packet is sent back to the inquiring vehicle upon fulfilment of this condition and the search inquiry is finished successfully.

SIMULATION

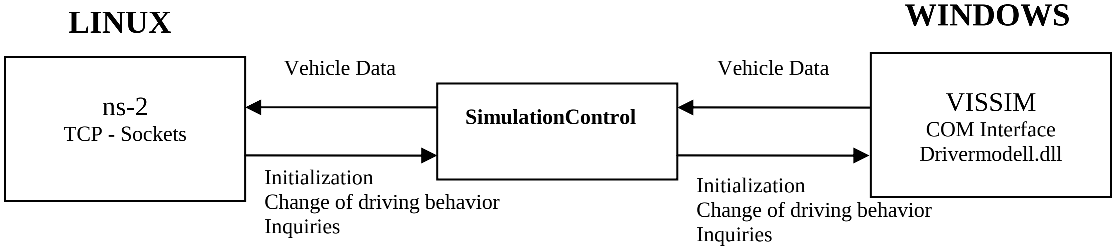

Figure 9 – Interaction between ns-2 and VISSIM

For the implementation and evaluation of the presented concept a simulation environment was implemented. Main parts of the simulation environment are the microscopic traffic simulator, VISSIM [11] and the well known network simulator ns-2 [10]. To perform realistic interactive simulations VISSIM and ns-2 are coupled through a coupling program, called SimulationControl. The interaction between ns-2 and VISSIM is drafted in **Figure 9**

- Synchronize the exchanged data between VISSIM and ns-2
- Perform interaction (Change of driving behaviour, Request for VehicleData etc.) with VISSIM via the offered interfaces
  - COM (Component Object Model) and
  - Drivermodel.dll

To have a realistic simulation scenario an existing and reliable city model of the German city Braunschweig is used for the evaluation of the TSP-Based Information Collection Algorithm (TIC). In the used Braunschweig VISSIM model, the simulated traffic is realistically modelled from 6.00 h till 12.00 h for an average weekday.

Figure 10 – Modell of the city of Braunschweig

t = time step
C = Capacity
T = Tolerance
$R_{(t)}$ = Filling rate at time step t
$FP_{(t)}$ = Free places at time step t

$$FP_{(t)} = C - \left(R_{(t)} + random(-T, T)\right)$$

Formula 1 - Uniform distribution to simulate parking place availability

Additionally, parking places are integrated into the simulation. Since the focus lies on the TIC-Algorithm, only the location and the capacity of each *big* public parking place in Braunschweig is taken into account. To calculate the filling rate at a certain time, a *uniform distribution function* is used. This function generates every 20 seconds a random filling rate based on a uniform distribution with the expected number of vehicles and variance value specific for each parking place.

Implementation details of the TSP-Based Information Collection Algorithm

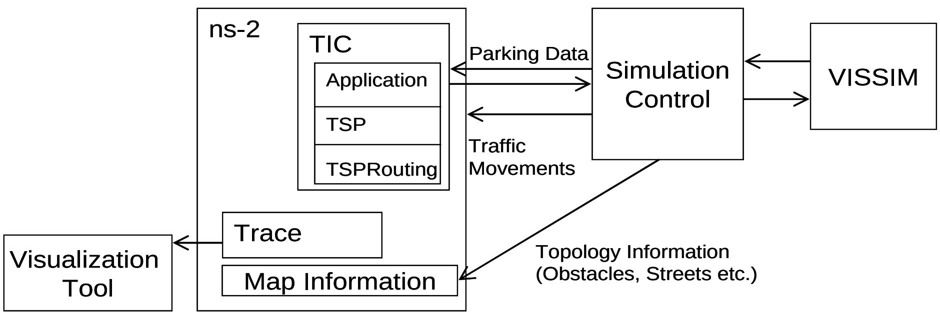

Figure 11 – Implemented architecture of the TIC-Algorithm

The TIC implementation is depicted in **Figure 11** and distinguishes three parts:

1. Access to resources: The TIC implementation needs access to a variety of resources.

- Most obvious are the movements of the vehicles. These are sent from the VISSIM traffic simulator to the ns-2 network simulator, which feeds these movements to the right mobile nodes.
- It is presumed that every vehicle is equipped with a navigation system; this gives the mobile node access to detailed map information. This map information is implemented

in the ns-2 simulator, and every mobile node has access to this map. This data is static and is generated based on data from VISSIM.

- Every vehicle must be able to communicate with parking places. The occupation statistics of the parking places are modelled in the ns-2 simulator and are based on real field data. This data is available to every vehicle, when in communication range of the parking place.

2. Simulation of the information collection: The simulation of the TSP-Based Information Collection Algorithm is performed using the ns-2 network simulator. The behaviour of this algorithm is influenced by the collected data. The algorithm can be divided into three parts or layers:

- The application is specific for the kind of information requested by the user. The function of the application is two-fold. It offers an interface to the user as well as it collects and aggregates the data during the information search. For the example case of free parking place search; the application collects parking data information obtained from parking places in communication range.
- The TSP layer has access to map information and calculates the optimal route for the search packet. This layer determines to which route the data packet will traverse from junction to junction. For this the TSP layer needs access to the movement data of its own mobile node and the map information.
- In order to be able to decide which nodes are in communication range, the movement data is processed. The TSPRouting layer sends the search packet to the next vehicle.
- After completed search the TSP layer re-locates the inquiring vehicle by using a location lookup service and sends the data back to the originator.

3. Analysis and visualization of the results.

RESULTS

Figure 12 shows the time taken by the data packet to hop from junction to junction traverses junctions**.** As can be seen from this figure, a successful search in the city of Braunschweig takes 80 seconds. This result corresponds to the simulation model of Braunschweig that has 112 roads around the city centre with approximately 1400 vehicles during weekday rush hour. Within these 80 seconds, the packet searches all roads in the target area for available parking spaces and returns with the collected results to the originating vehicle.

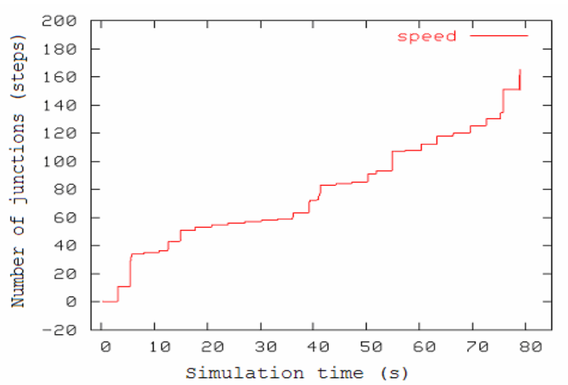

Figure 12 – Time and TIC-Algorithm steps during a successful search

The configuration for the simulations of the TIC-Algorithm is shown in **Table 1**. The simulations are started between 06:00 and 09:30 for an average weekday. Accordingly to defined 22 different simulation configurations the accumulated results of the simulations are listed in **Table 2.**

Simulation Configuration				
Protocol parameters			Simulation parameters	
Parameter	Value		Parameter	Value
Maximum search time (s)	240		Search query start time	*adjustable*
Beaconing interval (s)	1,5		Penetration rate (%)	100
Maximum attempts if no forwarder is available	30		Random seed[2]	*adjustable*
Communication range (m)	240		Maximum number of available nodes	5000
Acknowledgement[3] (s)	1,5		Roadmap size (km²)	12
Number of parking place searching vehicles at same time	1		Number of modelled junctions	95
Search area size(km²)	12		Number of modelled roads	119

Table 1 - Results of each simulation together with corresponding configuration

Search statistics	# Simulation configurations		Success ratio
	22		32%
Search time	Min. search time	Max. search time	Avg. search time
	79 s	216 s	155 s
Number of hops in one successful search	Min. hops	Max. hops	Avg. hops
	258	570	462

Table 2 – Aggregated results of the simulation

CONCLUSION AND OUTLOOK

We have presented a TSP-Based Information Collection strategy that is capable of collecting dedicated information from a pre-defined area of the city. The algorithm is developed and implemented in a simulation environment, which makes it is possible to simulate realistic city scenarios with up to 5000 mobile nodes (vehicles).

The simulation results show that it is possible to route a data packet through a defined area of the city using a Vehicular Ad Hoc Network within acceptable search times. A search strategy that relies on only one single data packet is bandwidth efficient; however it turned out to be very fragile. When a search packet gets lost, all collected information is lost. The reason for this is that network partitioning occurs quite often because of the clustered movement patterns of the vehicles. Therefore it is necessary to have a robust recovery strategy, to prevent packets from getting lost. The success of a complete search depends strongly on the vehicle movement scenario defined. We observed that simulations around 06:00 mostly failed in contrary simulations around 07:30 were mostly successful. The efficiency can be improved by relying on more then one packet. One approach could be to first route the search packet to the centre of the search area, because the information from this part is probably the most important. From here, the packet starts traversing in a circular way. After every, say three roads, the packet is cloned, and an intermediate result is sent back to the initiator. Starting the search in the centre of the search area, results in the most relevant search results, being returned first. When the inquiring vehicle does not receive a reply within a predefined time, the packet is probably lost and should be dropped by the forwarding vehicle. The inquiring vehicle re-sends a new data packet that continues from the point were the intermediate search packet was received. Another approach is to divide the search area into separate parts, which are searched by individual search packets. This means that an inquiring vehicle sends the search packet to the centre of a search area. Then the packet is cloned and sent towards every sub section.

[2] The seed value influences the starting values of the vehicle traffic generating distribution functions.
[3] Maximum tolerated time to get an acknowledge-message between a sender and a receiver in communication range.

Position information and map data are sufficient resources to let a data packet traverse through a city scenario.

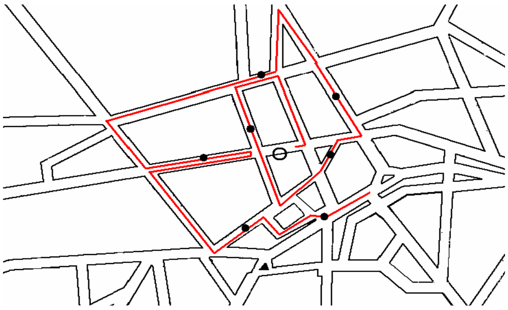

Figure 13 – Circular search, with intermediate report

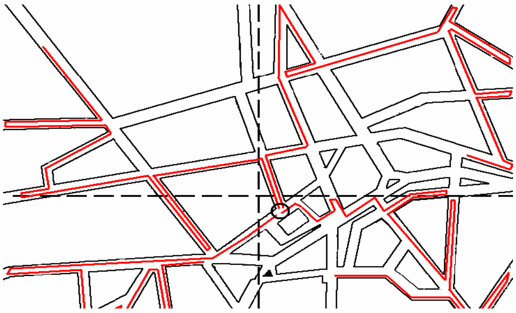

Figure 14 – Parallel search in parts of the search area

The implemented algorithm aims to let a data packet visit the geographical position of every junction of the optimal route. The pre-calculated route for the data packet is based on average traffic conditions which need to be present in the vehicle. Our future research will focus on information search algorithms that rely on more than one data packet and do not need pre-calculation of a certain route for the data packet.

REFERENCES

[1] http://focus.msn.de/F/FT/FTB/FTB113/ftb113.htm – *Parkplatznotstand und Ordnungs-Widrigkeiten*

[2] Wischhof L,; Ebner A.; Rohling H. *SOTIS – A Self Organizing Traffic Information System based on Car-to-Car Communication: Prototype Implementation.* In Proceedings of the 1st International Workshop on Intelligent Transportation (WIT 2004), Hamburg, Germany, March 2004.

[3] Mauve M; Hartenstein H; Füßler, H.; Widmer, J.; Effelsberg, W. *Positionsbasiertes Routing für die Kommunikation zwischen Fahrzeugen.* it+ti (Informationstechnik und Technische Informatik 44), Oldenburg Verlag, May 2002.

[4] Lochert, C.; Hartenstein, H.; Tian, J.; Füßler, H.; Hermann, D.; Mauve, M. *A Routing Strategy for Vehicular Ad Hoc Networks in City Environments.* IEEE Intelligent Vehicles Symposium (IEEE IV 2003), Columbus, Ohio, June 2003.

[5] E.M. Belding-Royer, C.-K. Toh. *A review of current routing protocols for ad-hoc mobile wireless networks.* IEEE Personal Communications Magazine, pages 46--55, April 1999

[6] Hromkovic, J. *Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics.* Springer Press, 2001.

[7] Aarts, E and Lenstra J.- K. *Local Search In Combinatorial Optimization.* Princeton University Press, 2003.

[8] Karp B.-N., *Geographic Routing for Wireless networks*, The Division of Engineering and Applied Sciences. Harvard Universiy, Cambridge Massachusetts (2000)

[9] Josh Boch, David B. Johnson, and David A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet-Draft, draft-ietf-manet-dsr-03.txt, October 1999. Work in progress. Earlier revisions published June 1999, December 1998, and March 1998.

[10] Kevin Fall and Kannan Varadhan, editors. ns Notes and Documentation. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997. Available from http://www-mash.cs.berkeley.edu/ns/.

[11] PTV traffic mobility logistics. http://www.ptv.de.