The Architecture of Distributed Database System in the VANET Environment

Ján Janech, Emil Kršák and Štefan Toth
Department of Software Technologies,
Faculty of Management Science and Informatics,
University of Žilina, Univerzitná 1, 010 26 Žilina, Slovakia
E-mail: {jan.janech, emil.krsak, stefan.toth}@fri.uniza.sk

Keywords: VANET, architecture, DDBS, communication protocols, security

Received: December 20, 2013

This paper describes principles of the data communication in the distributed database system AD-DB developed by the authors. The database system is designed to function properly in such a complex and dynamic network as the VANET is. That way, vehicles connected to the VANET could distribute traffic related data to the others. The paper concludes by proposing a solution for security problems by introducing cross-certificate to our system.

Povzetek: Predstavljena je izvirna arhitektura porazdeljenih podatkovnih sistemov v okolju VANET.

1 Introduction

VANET (Vehicular Ad-hoc NETwork) is a field of important research nowadays [25]. Many researchers are trying to develop new principles to make it possible to distribute information through this network. Applications for VANET could be divided into two categories: *safety applications* and *comfort applications*. Safety applications are more important ones. They are focusing on distributing information about traffic accidents, obstacles and other safety hazards to as many vehicles as possible [13, 14].

VANET is defined to be a special case of MANET (Mobile Ad-hoc NETwork) where network nodes are represented by vehicles in a road traffic. But problems with distributing data in VANET are completely different from the MANET ones. MANET nodes as computers with limited power source and limited computing resources have to communicate in small time frames to preserve as much power as possible. All research of the MANET communication is about minimizing communication and computing time and about conserving node power.

On the other hand, almost all of VANET nodes (vehicles in road traffic, road infrastructure) have good power source. So research in this area is focusing on the best way to distribute information for all nodes that are interested in it.

2 State of the art

2.1 Classic architecture of distributed database system

Architecture of DDBS (Distributed Database System) from data organization point of view is shown in figure 1. It is simple layered model with four layers. Each one of them represents some view on data itself.

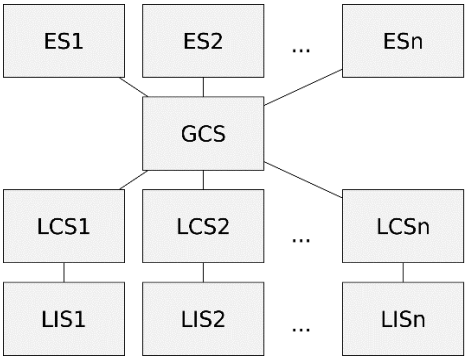

Figure 1: DDBS reference architecture [1].

There are four layers of distributed database system, each modeling one kind of view on distributed database [1]:

1. *LIS (Local Internal Schema)* represents physical representation of data stored at one node. It is analogy of internal schema from centralized databases.
2. *LCS (Local Conceptual Schema)* describes logical organization of data at one node. It is used to handle data fragmentation and replication.
3. *GCS (Global Conceptual Schema)* represents logical organization of data in whole distributed database system. This layer is abstracting from the fact that the database system is distributed.
4. *ES (External Schema)* represents user view into distributed database. Each external schema

defines which parts of database user is interested in.

The fact that the user is using only global conceptual schema through views defined in external schema, assures that the user can manipulate the data regardless of its position in the distributed database system. Therefore it is necessary to have a mapping from every local to the global conceptual schema. This mapping, named GD/D (Global Directory/Dictionary), is defined as part of the distributed database system.

The main role of GD/D is to provide access to mapping between local conceptual schemas and the global conceptual schema. So it has to be accessible from every node sending queries to the system. There are several ways to ensure it [4, 5]:

1. *Centralized directory* – whole GD/D is stored centrally at one node. The advantage of this solution is that it makes GD/D manipulation simpler. However, one central node represents single point of failure for the whole distributed system and can be a bottleneck as well.
2. *Fully redundant directory* – replication of the whole GD/D is stored on every node. That way it can be quickly accessed whenever needed. But its modifications are more complicated due to its multiple occurrences in the system.
3. *Local directory* – every node stores only its own part of the GD/D, so its management is very simple. On the other hand, global query requires communication with other nodes to make possible to create the query plan.
4. *Multiple catalog* – in the clustered distributed database system it is possible to assign whole GD/D replication to one node in each cluster. It is combination of first two ways.
5. *Combination of 1. and 3.* – every node has its own GD/D replication and there is one global replication as well. Each of this possibilities has its pros and cons. But they have all something in common: the system needs to recognize all of its parts.

Whether the GD/D is stored at one node or somewhat distributed through the system, there needs to be some way how to access it as a whole. This is not possible in VANET as there is no way to ensure communication between all of the nodes. In this situation, GD/D cannot be used to locate requested data.

As of the present time there is no solution designed specifically for VANET known to the authors. But there are few solutions for MANET, so we will describe them in next sections of the article.

2.2 TriM protocol

The TriM protocol is the one of first attempts for solving the problem of data distribution in MANET environment the generic way. It was designed as part of a PhD thesis at the University of Oklahoma [6]. The main focus of the protocol is to minimize power consumption and to utilize all three modes of communication [7]:

- *Data Push* represents data distribution using broadcast messages.
- *Data Pull* represents on demand data distribution.
- *Peer-to-peer communication* for querying data.

The main disadvantage of the TriM protocol is its requirement to have same data on all nodes. This requirement makes it practically unusable in the VANET environment.

2.3 HDD3M protocol

HDD3M protocol tries to solve TriM protocol problems. As in the original protocol, HDD3M aims to use all three modes of communication and to conserve as much power as possible. The main difference from the TriM protocol is a possibility to manage database fragments and to modify the distributed database via transactions.

HDD3M divides nodes into 3 categories:

- *Requesting node (RN)* is sending queries to distributed database system.
- *Database node (DBN)* is containing database fragments.
- *Database directory (DD)* stores GD/D for distributed database.

This protocol must solve problems with distribution GD/D. There is no guarantee that all of database directory nodes receive the GD/D update request. Some of the nodes could be inaccessible through MANET or shut down due to lack of energy. When the network is fragmented, keeping the data accurate and actual might be impossible.

The biggest problem for the deployment of distributed databases in the VANET environment is the necessity of the knowledge of all the nodes beimg available in the system. This problem persists in this solution as well because the GD/D is still used.

3 Principles of proposed solution

So the only way to ascertain the use of the distributed database system in the VANET environment is to remove the GD/D from the system and replace it with a different principle. As it has been said already, the GD/D describes the mapping between the local and global conceptual schemes. Without the mapping the system does not know where the data are located and how to query them.

Using the GD/D in the VANET environment is impossible because it requires knowledge of the whole system (*global directory*). In VANET every node knows only its immediate surroundings. So querying a distributed database is fairly limited in such environment. The only nodes which can be addressed to using queries are those in the immediate surroundings in the network. So the system naturally creates virtual clusters of nodes that can communicate with each other. The clusters might overlap, so each of the nodes of the cluster can communicate with another set of nodes.

The only possibility to introduce principles of distributed database systems into VANET environment

lies in allowing to query data only from clusters containing the query node. That way we can replace GD/D with another principle – CD/D (Cluster Directory/Dictionary). But there is still question, how to store CD/D and how to distribute it throughout the database system. The possibilities are same as they were for storing GD/D. They were described in the subsection 1. in section 1 of this paper.

Best possibility for VANET seems to be storing their own part of CD/D at each of network nodes. Other possibilities would be complicated to implement due to highly dynamic nature of VANET.

This is the way distributed database management system AD-DB (AD-hoc DataBase) is working [24]. AD-DB was created as the result of a PhD thesis at University of Žilina [2] by one of authors.

4 Query processing in AD-DB

As we already said, it is impossible to keep CD/D as a whole and distribute it throughout the VANET. Instead of that, AD-DB is using broadcast messages for data communication and lets each data node to decide whether it has requested data or not by looking to its own part of CD/D.

AD-DB supports two methods of communication each based on slightly different principle:

- *Pull method* is an application of pull mode of data communication into AD-DB database. It allows each node to query data from cluster.
- *Push method* is an application of push mode of data communication into AD-DB database. It allows to share own data to other nodes without any prior query.

4.1 The Pull method

The Pull method represents the standard method of query processing in classic distributed database systems. One of the nodes sends query to the system and waits for the results.

The method could be used in such a situation where a client does not have to update the data periodically but needs to query it once instead. One time search for nearby cinemas could be taken as an example of such a situation.

It is also possible to use the pull method as a means for data replication but it is much more ineffective than using the push method [15].

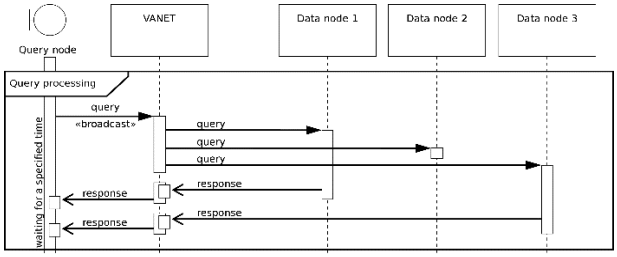

Figure 2: Query processing using the pull method [2, 3].

The query principle is shown on Fig. 2. Communication is done in the following steps:

1. *Global query optimization*. It is important to optimize a query to minimize the size of resulting data.
2. *Sending a query*. The Query node sends optimized query using broadcast message. That way all of the data nodes in cluster receive the query. The query node waits for the specified time.
3. *Query fragmentation*. Every data node which receives the query fragments searches for subqueries that the node is able to execute.
4. *Local subquery optimization*. A data node optimizes each of the found subqueries and prepares it for execution.
5. *Subquery execution*. The data node executes each of subqueries.
6. *Sending the result*. The data node sends back the resulting data together with the identification of executed subquery using the unicast message.
7. *Results evaluation*. After the specified time runs out, the query node evaluates all results received from the data nodes and merges them to one complete result.

4.2 The push method

Given that the organization of the network structure is changing rapidly in VANET, it is clear that sometimes there is a need for querying the same data repeatedly. A possibility of using the push method of data communication in AD-DB can be handy in such situation.

This is also the reason why the push method is more effective to be used in data replication algorithms than the pull method [15].

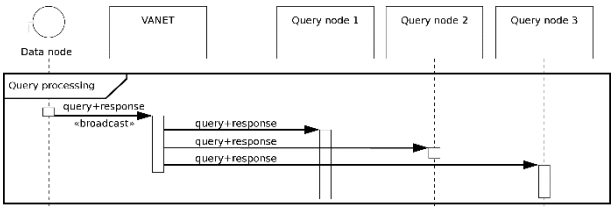

Figure 3: Schematic illustration for push method [3].

The schematic principle of push method is shown in Fig. 3. The Communication is done in the following steps:

1. *Local query optimization*. The data node optimizes the query and prepares it for execution.
2. *Query execution*. The data node executes the optimized query.
3. *Sending the data*. The data node sends resulting data packed with the query through VANET as broadcast message.
4. *Results evaluation*. When the query node receives the data, it analyzes the attached query to determine whether it needs the data or not. If

it needs the data, it forwards the data to the user application to process it.

5 High level communication protocol for AD-DB

Schematic representation of the communication protocol used in AD-DB is shown in Fig. 4. The data node can process the processQuery message. This is sent by a query node in the form of a broadcast message in the pull method of communication.

The query message has following structure [2]:

- *Schema uuid* is a unique identifier of the current database schema. It is important to include this for the data node to be able to determine whether it should process the query or not.
- *Serialized query* represents the query itself. The best way to transfer the query is in a form of serialized abstract syntax tree, as it is easy to process by the data node.

There is no need to transfer the session identifier of any kind, because the query and uuid could be used as a unique identifier of the request.

The response message structure is as follows [2]:

- *Schema uuid* as part of response unique identifier.
- *Serialized query* as part of response unique identifier. It is possible to use the query as part of unique identifier because the query processed by the database system is expected to be simple and short. If this assumption was not true, it would still possible to use value computed from the query by some hash function instead.
- *Query part* is the identifier of the processed subquery.
- *Data* as a collection of the resulting objects.

Using the schema uuid and query pair as a unique identifier of a request has one advantage over using surrogate identification number. This way the response message format can by the same for the pull and the push methods of communication.

Important part of a response message is the query part identification. It represents a unique identifier of query part processed by a data node as a subquery. This identifier is needed by a query node to be able to merge all responses from all responding data nodes.

There are two possibilities how to use the same system of numbering for all query parts by both the query and the data node:

- Inserting the identifier directly into the serialized query. The process of identifier inserting is done directly by the query node after the global optimization. Example of a query with identifiers (syntax of the query language used by AD-DB is published in multiple publications by authors [2, 3, 8]):

$$^{(1)} \bowtie (^{(2)}projects//(\lambda x|x \lhd name{=}\text{KANGO}),^{(3)}employees)$$

where (1) identifies the whole query as one part, (2) identifies collection of all projects with the name KANGO, and (3) identifies collection of all employees.

- Automatic numbering of all operations by theirs priority. The priority of an operation can not change as it is defined by the query language, so the numbering will be same on both query and data node. This system is preferred and it is used by AD-DB as it does transfer slightly smaller quantity of data between the query and data node.

The push method is using the processResponse message. It is sent by the data node in the form of broadcast message.

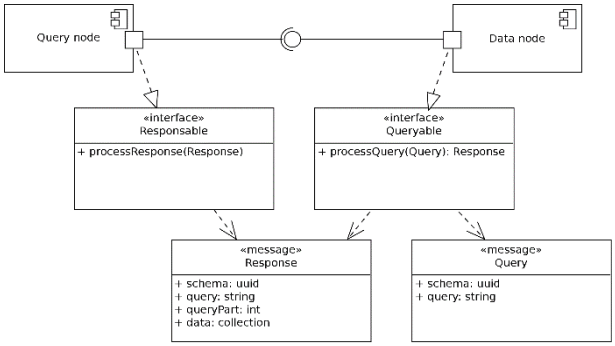

Figure 4: Schematic representation of communication protocol used by AD-DB [2].

6 The OSACP protocol

OSACP (Object Structure Aware Communication Protocol) is an application protocol designed specifically to transfer structured data through VANET. It is designed as part of PhD thesis at University of Žilina [9]. OSACP is using UDP transport protocol on top of IPv6 network protocol. Its design allows it to transfer any structured data through VANET and reconstruct it on the other side even if part of data was not transferred correctly [10, 3].

Missing parts of the structure are replaced by special object UNKNOWN to indicate incomplete message. It is up to the user of the distributed database (person, or another application) to decide whether it can process the message or not.

7 Security of DDBS

Since VANET is a very dynamically changing system, we must pay special attention to the communication security. Connections between network elements are constantly changing according to the current position of the elements and the impact of their communication devices. Looking at VANET as a distributed system without the possibility of at least partial centralization presents a high risk of abuse (threats model, authentication, privacy, secure identification of the position, etc.). It is necessary for network elements to create an appropriate security architecture that will protect the participants from various types of attacks.

Safety aspects that are required in networks can be divided into 2 areas:

- Availability

- Authentic information and privacy

On that basis, we can define commonly known security threats and divide them into the following categories.

7.1 Threats to availability

Denial of service – overload the node resources so it is not able to perform other important and necessary tasks. It occurs very often as distributed denial of service. Similar situation can be made by jamming the channel.

Black hole attack – an attack by misbehavior of the node. It can forcibly redirect network traffic to a non-existent node by falsifying the routing information, causing the data to be lost.

Malware – malware attacks, such as viruses in VANETs, have the potential to cause serious disruption to its normal operation. Malware attacks are more likely to be carried out by a malicious node. The attacks may be introduced into the network when VANET units in cars and roadside station receive software updates or special plugins.

GPS spoofing – using GPS simulators to generate fraudulent signals so nodes believe they are in different location or in different time.

Broadcast tampering – an attacker can inject false traffic safety message into the network. Broadcasting this message can cause accidents or manipulating the flow of traffic to clear chosen route.

Spamming – spam messages on VANETs elevate the risk of increased transmission latency. The lack of centralized administration causes serious problems in VANET.

7.2 Threats to authentication

Masquerading – an attacker presents itself as legitimate node in the vehicular network by using false information or by using message fabrication, message alteration, or message replay. For example, an attacker acts as an emergency vehicle to mislead other vehicles to slow down and yield [16].

Replay attack – this attack happens when an attacker replay the transmission of earlier information to take advantage of the situation of the message at time of sending [17].

Sybil attack – in this attack type, a node sends multiple messages to other nodes and each message contains a different fabricated source identity in such a way that the originator is not known [18, 19]. The basic goal of the attack is to provide an illusion to other nodes about a traffic jam, and force them to take an alternate route

Message Tampering – this type of attack is changing messages while being transferred from a source to theirs destination. Everyone within the same zone in VANET can listen to all messages sent by other users. Thus, malicious users can modify the contents of a message before it is received by real destination [20].

ID Disclosure – is about disclosing the identity information. Using this method, attacker can track the current location of his target [20].

7.3 Threats classification

From this point of view we should define which attack described in sections 7.1 and 7.2 is mapped to which layer of our distributed database system. All of them should be expected only on two lowest layers. One can expect these type of attack form group of *Threats to availability* to the LIS layer:

- Denial of service
- Black hole attack
- Broadcast tampering
- Spamming

From group of *Threats to authentication* come into consideration:

- Masquerading
- Reply attack

On the *second level LCS* (Local Conceptual Schema) it can be form group of *Threats to availability*:

- Malware
- GPS spoofing

And from group of *Threats to* authentication:

- Sybil attack
- Message Tampering
- ID Disclosure

Many of these attacks can be avoided by using PKI (Public Key Infrastructure). In conventional database systems PKI and cryptography ensure the highest layers. The VANET situation is reversed. We need to ensure the lowest layers.

Classic PKI is composed of CA (Certification Authority) tree with one root CA. Each CA has a network of RA (Registration Authority) to verify the applicant's identity certificate. Security is based on cryptography. The parent node in the hierarchy "Node certificate - CA certificate – root CA certificate" guarantees the public key and identifies subordinate by form of a certificate that is signed electronically document. In the case of private key compromising the certificate relevant node or CA needs to be revoked.

When using PKI in VANET, we must take into account the specificities that make it impossible to use certain features and protocols used in PKI.

7.4 Local storage certificates of root CAs

Many CA roots are expected in VANET and therefore fluctuation of CA is very likely. In environment, with many root CAs, it is very complicated to store every root certificate in local storage in every node. On the other hand it is possible to use cross-signed certificates so a node in its local storage must have only one or few root certificates. When node receives signed message, it receives also chain of certificates of CA hierarchy and if there is a root CA that believes it (in its local storage), message is valid.

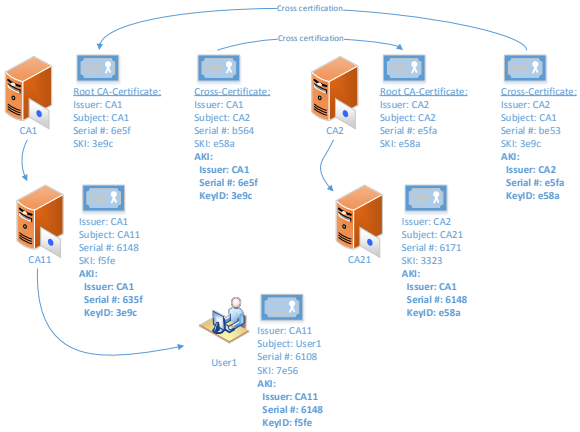

Figure 5: N tier cross certificate [21].

7.5 Revocation of certificates

In standard PKI there is a mechanism to revoke certificate by issuing CRL (Certification Revocation List) by CA. It contains a list of every revoked certificate. It can be the absolute list or the differential list of revoked certificates from last published CRL. There is also OCSP protocol (Online Certificate Revocation Protocol), which is used for online checking of validity the certificate. In VANET OCSP has very low performance [22, 23] and is not very useful.

For our distributed database system we recommend to use the cross-certificate and the absolute CRL. Using the absolute CRL we risk very large CRL. It depends on the length of certificate validity. The short period of validity of the certificate reduces probability of compromising the private key, but increases overhead and re-applying for the issuance of the certificate. For that process it can help implementation of automatic renewal of certificates by CA. Choosing a suitable length of validity of the certificate depends on the particular use of DDBS.

8 Conclusion

There is no known distributed database systems that would be possible to operate in the VANET environment. There are some attempts to do so for MANET, but they are unusable for VANET.

The paper presented the communication system of the distributed database system AD-DB. The database system is designed to be used in the VANET environment and so its basic principles had to be altered for such usage.

In the nearest future we would like to focus on enhancing the query optimization algorithms, but there are many other areas which would be interesting to explore. For example, many of the data in VANET are of highly temporal character, e.g. current weather, traffic flow speed, traffic obstacles, etc. It would be interesting to have a possibility to query current state of those temporal data.

We have some accomplishments in this area even now. We have designed system to query visual objects recognized by vehicle cameras through VANET [11, 12], so the next logical step would be to integrate this system into distributed database system AD-DB.

Acknowledgment

This contribution/publication is the result of the project implementation:

Centre of excellence for systems and services of intelligent transport II., ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

References

[1] M. T. Ozsu, P. Valduriez (2011). *Principles of Distributed Database Systems*. 3rd ed.

[2] J. Janech (2010). *Riadenie procesov pri distribúcii databáz* (Data distribution process control). PhD disertation, Dept. of Software Technologies, University of Žilina, Žilina, Slovakia.

[3] J. Janech, T. Baca, A. Lieskovsky, E. Krsak, K. Matiasko (2013). Distributed Database Systems And Data Replication Algorithms For Intelligent Transport Systems. *Communications: Scientific Letters of the University of Žilina*. Vol. 15, No. 2.

[4] P. Sokolovský, J. Pokorný, J. Peterka (1992). *Distribuované databázové systémy* (Distributed Database Systems). 6th ed.

[5] C. J. Date (2003). *An Introduction to Database Systems*. 8th ed.

[6] L. D. Fife (2005). *TriM: Tri-Modal data communication in mobile ad-hoc network database systems*. Ph.D. disertation, University of Oklahoma.

[7] L. D. Fife, L. Gruenwald (2003). Research issues for data communication in mobile ad-hoc network database systems. *SIGMOD Rec.*, Vol. 32, No 2.

[8] J. Janech, A. Lieskovský, E. Kršák (2012). Comparation of Strategies for Data Replication in VANET Environment. *26th International Conference on Advanced Information Networking and Applications Workshops* (WAINA).

[9] T. Bača (2012). *Optimalizácia prenosu správ v ad hoc sieťach* (Optimization of Message Distribution in Ad-hoc Networks). PhD dissertation, Dept. of Software Technologies, University of Žilina, Žilina, Slovakia.

[10] T. Bača (2012). Optimisation of message distribution in Ad-hoc networks. *Information Sciences and Technologies : bulletin of the ACM Slovakia*, Vol. 4, No. 4.

[11] Š. Toth, J. Janech, E. Kršák (2013). Query Based Image Processing in the VANET. *5th International Conference on Computational Intelligence, Communication Systems and Networks* (CICSyN2013).

[12] Š. Toth (2013). *Spracovanie obrazu s využitím dopytov v prostredí VANET* (Query Based Image Processing in the VANET). PhD dissertation, Dept. of Software Technologies, University of Žilina, Žilina, Slovakia.

[13] E. Kršák, P. Hrkút, P. Vestenický (2012). Technical infrastructure for monitoring the transportation of oversized and dangerous goods. *Federated Conference on Computer Science and Information Systems* (FedCSIS 2012).

[14] S. Badura, A. Lieskovsky (2010). Intelligent traffic system: Cooperation of MANET and image processing. *1st International Conference on Integrated Intelligent Computing* (ICIIC 2010).

[15] T. Bača (2011). Data replication in distributed database systems in VANET environment. *Proceedings of 2011 IEEE 2nd international conference on software engineering and service science*, Beijing, China.

[16] A. K. K. Aboobaker (2010). *Performance analysis of authentication protocols in vehicular ad hoc networks (VANET)*. Technical Report, Dept. of Mathematics, University of London.

[17] B. Parno and A. Perrig (2005). Challenges in securing vehicular networks. *The Fourth ACM Workshop on Hot Topics in Networks* (HotNets-IV).

[18] F. Sabahi (2011). Vehicular Ad - hoc Networks Security Analysis. *International Conf. on Computer Engineering and Applications* (ICCEA).

[19] S. Zeadally, R. Hunt, Y. S. Chen, A. Irwin, and A. Hassan (2010). Vehicular Ad Hoc Networks (VANETS): Status, Results, and Challenges. *Telecommunication Systems.* Vol. 50, Issue 4, pp 217-241.

[20] F. Sabahi (2012). Impact of Threats on Vehicular Adhoc Network Security. *International Journal of Computer Theory and Engineering*, Vol. 4, No. 5.

[21] C. L. Mankowski (2012). Answer on post Can a certificate have multiple chains and multiple self-signed roots? http://security.stackexchange.com/questions/15562/can-a-certificate-have-multiple-chains-and-multiple-self-signed-roots

[22] J. Serna-Olvera, V. Casola, M. Rak, J. Luna, M. Medina, and N. Mazzocca (2010). Performance Analysis of an OCSP-Based Authentication Protocol for VANET, *Int. J. Autonomous and Adaptive Communications Systems*. Vol. 3, No. 2, pp 19-45.

[23] K. Papapanagiotou, G. F. Marias, P. Georgiadis (2007). A Certificate Validation Protocol for VANETs. *IEEE Globecom Workshops.* Washington. pp: 1-9.

[24] J. Janech, Š. Toth (2013). Communication in Distributed Database System in the VANET Environment. *Federated Conference on Computer Science and Information Systems* (FedCSIS 2013). pp. 795–799.

[25] K. Mäkká, J. Dicová (2013). Possibility of using the software product AIMSUN in the process of modelling transport. *Proceedings in ITS 2013 - Intelligent Transportation Systems 2013, Virtual Conference*.