Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 795–799

Communication in Distributed Database System in the VANET Environment

Ján Janech, Štefan Toth
University of Žilina
Faculty of Management Science and Informatics
Department of Software Technologies
Univerzitná 1, 010 26 Žilina
Email: {jan.janech, stefan.toth}@fri.uniza.sk

***Abstract*—This paper describes principles of the data communication in the distributed database system AD-DB developed by paper 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 others.**

I. Introduction

VANET (Vehicular Ad-hoc NETwork) is the field of important research nowadays. Many 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.

II. State Of the Art

A. Classic Architecture of Distributed Database System

Architecture of DDBS from data organization point of view is shown at the figure 1. It is simple layered model with four layers. Each one of them represents some view on data itself.

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

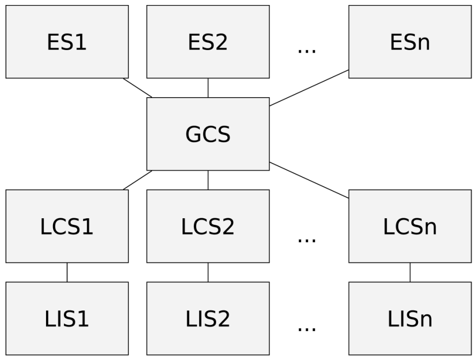

Fig 1: DDBS reference architecture [1]

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 is user 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 with 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 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 not 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.

B. 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 a part of 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.

C. 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 possibility to manage database fragments and to modify distributed database in 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 deployment of distributed databases in the VANET environment is the necessity of the knowledge of all the nodes in the system. This problem persists in this solution as well because the GD/D is still used.

III. Principles of Proposed Solution

So the only way to make sure it is possible to use 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 a VANET every node knows its immediate surroundings only. So querying of 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 A. in section I of this paper.

Best possibility for VANET seems to be to store they 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 is working. AD-DB was created as the result of PhD thesis at University of Žilina [2] by one of authors.

IV. 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.

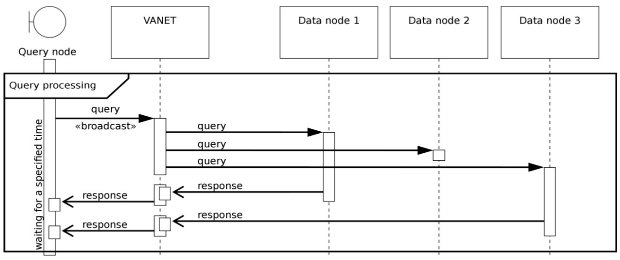

Fig 2: Query processing using the pull method [2][3]

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

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

A. Pull method

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

The method could be used in such situation where a client does not have to update data periodically and it 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 mean for data replication but it is much more ineffective than using the push method [15].

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*. 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 received the query fragments the query and searches for subqueries the node is able to execute.
4. *Local subquery optimization*. Data node optimizes each of found subqueries and prepares it for execution.
5. *Subquery execution*. Data node executes each of subqueries.
6. *Sending the result*. Data node sends back the resulting data together with identification of executed subquery using unicast message.
7. *Results evaluation*. After the specified time runs out, the query node evaluate all results received from data nodes and merges them to one complete result.

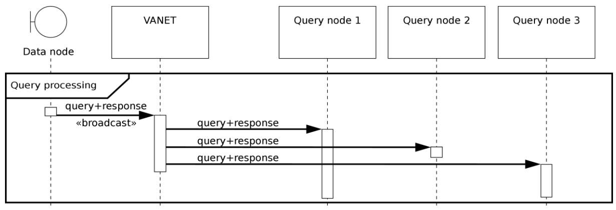

Fig 3: Schematic illustration for push method [3]

B. Push method

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

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

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

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

V. High Level Communication Protocol for AD-DB

Schematic representation of the communication protocol used in AD-DB is shown on Fig. 4. Data node can process message processQuery. That is sent by a query node in form of 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 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 of it, as it is easy to process by data node.

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

The response message structure is as follows [2]:

- *Schema uuid* as a part of response unique identifier.

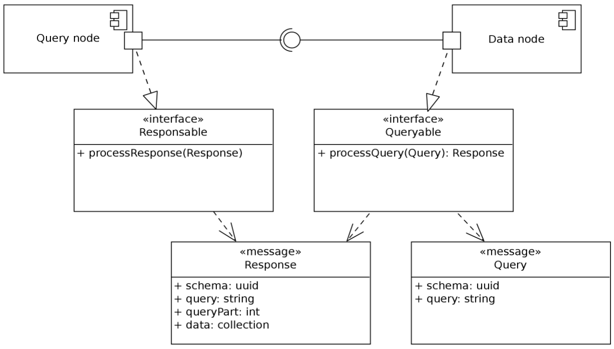

Fig 4: Schematic representation of communication protocol used by AD-DB [2]

- *Serialized query* as a part of response unique identifier. It is possible to use query as a part of unique identifier, because query processed by the database system is expected to be simple and short. If this assumption was not true, it is still possible to use value computed from a query by some hash function instead.
- *Query part* is the identifier of 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 the both query and data node:

- Inserting the identifier directly into serialized query. Process of identifier inserting is done directly by the query node after the global optimization.

  Example of 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. 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.

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

VI. 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 a 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.

VII. Conclusion

There is no known distributed database systems, that would be possible to operate in the VANET environment. There are few 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 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 query optimization algorithms, but there are many other areas which would be interesting to explore. For example, many of data in VANET are of highly temporal character, e.g. current weather, traffic flow speed, traffic obstacles, etc. It would be interesting to have 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, *Principles of Distributed Database Systems*, 3rd ed. 2011.

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

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

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

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

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

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

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

[9] T. Bača, *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. 2012.

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

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

[12] Š. Toth, *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. 2013.

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

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

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