Hybrid Model for Data Replication in Distributed Database Systems in VANET Environment

Final Report- November 25th, 2020

Student Details

1. Gauri V Nair, Roll No: 171CO116, Email ID: gaurivgn@gmail.com
2. Rahul Kumar, Roll No: 171CO133, Email ID: rahul.kr0324@gmail.com
3. Rishi Sharma, Roll No: 171CO135, Email ID: rishi94075@gmail.com

Abstract

Efficient data replication in the VANET environment is a demanding problem due to the robust nature of the network. Data replication is a method that aims to increase data available on mobile networks, especially on networks around VANET. To improve the performance of data backup, we reviewed data backup algorithms distributed on VANET to exchange information and calculate arbitrarily connected to the vehicle intersection network. To minimize duplication and improve network performance, we try to create a hybrid model that controls the number of duplicate messages that can be copied over the network and then proposes efficient data delivery techniques. The main idea is that the data carrier distributes the data backup tasks on several nodes to speed up the backup process. We calculate the number of communication steps for the network to reach an equilibrium state and show that the proposed distributed algorithm can be merged into a consensus at various levels of communication.

Objective

Robust data backup in a distributed network can accelerate the distribution of information in a particular area. Potential problems with regular routes include Network congestion in clustered areas that causes massive delays due to wastage of network resources and network flexibility. Since network resources may quickly become constrained, deciding how to replicate the messages plays critical roles in many routing protocols. Replication of data in VANET affects frequent network changes. Changes in network technology can prevent distributed data from reaching the cellular nodes they need. To reduce unnecessary transmission and increase communication costs, the number of messages that can be repeated when the message is distributed over a particular area should be limited. The replication delay will be reduced when the computer and communication load is distributed to the vehicle, which encourages us to combine distributed averaging and data-based distribution.

Problem statement

Therefore, we aim to provide a solution for data distribution and replication in VANET hence.

1. Develop a data replication scheme that can control the number of message copies and combine distributed averaging with data dissemination to evaluate network convergence rate through data replication. Motivated by the complementary properties of the existing models in Cluster and Grid computing, our objective is to come up with a hybrid model which takes in the best property of existing algorithms in the literature.

Analysis and Design

In order to solve the problem mentioned there is a distributed database system aimed directly at the **VANET** network, and the system is based on queries done by means of broadcast messages. It offers two ways of doing the queries:

- **Pull Method**
- **Push Method**

Using the PULL method with replications in VANET - Pull method is a method of data communication used in mobile and fixed networks. OBU sends a query, and an RSU or another OBU sends them to it in reply. This method is therefore also referred to as data on demand.

Using PUSH method with replications in VANET - For regular updates in the VANET network, it is possible to use the PUSH method. A query node does not send any request to the network. It only saves the information saying that the answer to this question is expected. A data node (RSU and OBU) carries out the data transmission. It transmits data by means of broadcast messages at regular intervals. The message packs the query being answered together with the data that reply to it. Thanks to this a query node can pair the data with the query. With this method of data distribution a query node does not set the time to receive an answer but it processes the data asynchronously as soon as it receives them.

There are some issues which are not addressed in the paper and we are also aiming to provide a solution to address them. Data replication technique for VANET databases must also deal with the following issues arising from constraints imposed by their specific environments and applications: **Replica relocation, Consistency Management, Location management, Server power consumption, Server mobility, Client mobility, Frequent disconnection of mobile hosts, Real-time applications, Network partitioning, Client power**.

Client mobility - The decision to replicate a data item in a particular server may be based on the access frequency of that data item on that server. Hence, the access frequencies must be dynamic in nature and the decision to replicate data items in appropriate servers must also be dynamic.

Replica relocation -Here, the problem of when, who, where and how to distribute copies is solved. Since mobile hosts are usually poor in resources, it is usually impossible to have copies of all data items in the network. In addition, since the mobile host can move freely and the network topology changes dynamically, there is no static optimal copy allocation. Therefore, effective replica relocation is critical to system performance.

Server power consumption - Servers in VANET run on battery power. Servers with higher power availability are expected to perform more work than those that have lower power. Servers with no power remaining would not be able to provide any more services.

Frequent disconnection of mobile hosts - Mobile hosts often get disconnected from the network due to various factors like power failure or their mobility. Servers which hold the data cannot provide services if they are disconnected from other mobile hosts.

Server mobility - Due to their mobility, servers might sometimes move to a place where they cannot be reached by other servers or clients. The replication technique should avoid replicating hot data items in such isolated servers.

As we mentioned above there are two algorithms for solving the issue mentioned in the paper, there are two basic methods for communications between Consumers and Producers: the Pull model and the Push model.
In the Push model, the initiator is the Producer. The Producer sends status information when it detects that the status changes are greater than the threshold. It is ideal for keeping maximum consistency between the Producer and the Consumer if the threshold is proper. However, if the threshold is small, minor changes result in too much information transmission, which may place strain on the network. If the threshold is large, important updating may be lost. Based on the Push model, we attempt to minimize the useless updating, and maximize information consistency between Consumers and Producers.
In the Pull model, the Consumer is the initiator. As such, the low Pulling rate consumes little network bandwidth, but may imply missing important updating during the Pulling interval, which is undesirable for Consumers. However, information at high Pulling rate is “fresher” but heavily intrusive to the original system. In a paper, we found that they used adaptive Push-Pull strategy to disseminate dynamic web data. They combine the two methods in the Web research area and present the “Push and Pull” (PaP) method as well as the “Push or Pull” (PoP) method. The PaP method simultaneously employs both Push and Pull methods to exchange data, but has

tunable parameters which determine the degree to which Push and Pull are used. The PoP method allows servers to adaptively choose between Push and Pull methods for each connection. Disseminating web data has many similarities with Cloud resource monitoring. Hence the idea of combining Push and Pull methods can also be extended to here in distributed database systems. The Push model is more accurate when threshold is appropriate, while the Pull model performs less transmission costs when inquiring interval is proper. Since the two models have complementary properties under specific conditions, a reasonable trade-off between the two models becomes possible. Thereby, we propose a combined Push and Pull model for Cloud computing.

Work done

Distributed database systems paradigm contains many shared resources, such as infrastructures, data storage, various platforms and software. Resource monitoring involves collecting information of system resources to facilitate decision making by other components. We extend the prevailing monitoring methods in literature, namely Pull model and Push model, to the paradigm of Distributed database systems. We find that in certain conditions, the Push model has high consistency but low efficiency, while the Pull model has low consistency but high efficiency. Based on complementary properties of the two models, we propose a user oriented resource monitoring model named Push & Pull (P&P), which employs both the above two models, and switches the two models intelligently according to users' requirements and monitored resources' status.

The P&P model is composed of P&P-Push algorithm and P&P-Pull algorithm.

Here is the **P&P pull** model:-

1. **Initialize Pull operation's initial query interval: PULL_INIT_INTERVAL, minimal possible inquiry interval: PULL_INTERVAL_MIN and maximal possible inquiry interval: PULL_INTERVAL_MAX**
2. **Pull_interval ⟵ PULL_INIT_INTERVAL**
3. **WHILE (TRUE)**
    a. **set Push operation identifier isPushed ⟵ FALSE**
    b. **waiting for Pull_interval**
    c. **IF isPushed equal to TRUE**
        i. **update status information (c_now) that Consumer currently holds**
    d. **ELSE**
        i. **isPulled ⟵ TRUE, Pull the Producer**
        ii. **update c_now**

e. ENDIF
f. change_degree= |c_now-c_last| /(MAX-MIN) 2
g. IF (change_degree≤UTD)
i. IF increased_Pull_interval ≤ PULL_INTERVAL_MAX
ii. Pull_interval=increased_Pull_interval
iii. ELSE keep current Pull_interval
iv. ENDIF
h. ELSE IF (change_degree>UTD)
i. IF decreased_Pull_interval≥PULL_INTERVAL_MIN
1. Pull_interval=decreased_Pull_interval

ii. ELSE
1. keep current Pull_interval
iii. ENDIF
i. ENDIF
j. c_last ⟵ c_now
4\. **ENDWHILE**

Here is P&P push model:-

1. WHILE TRUE
a. set Pull operation identifier isPulled ⟵FALSE
b. waiting for Push_interval
c. IF isPulled equal to TRUE during Push_interval
i. update status information (c_now) that Consumer currently holds
d. ELSE //examine whether need to Push
i. get sensor's current value (sensor_now) at Producer
ii. IF |sensor_now-c_now|/(MAX-MIN)≥UTD
1. isPushed⟵TRUE
2. c_now⟵sensor_now
3. Push c_now to the Consumer
iii. ENDIF
e. ENDIF
2. ENDWHILE

The P&P-Push algorithm runs at the Producer and the P&P-Pull algorithm runs at the Consumer simultaneously. The two algorithms try to make the resource monitoring system intelligently switch between Push and Pull operations according to user's requirements and status' changes of monitored resources. Then we analyzed the algorithms in four aspects:-

1. Set the "Pull" operation identifier "isPulled" and the "Push" identifier "isPushed" to be mutually exclusive to avoid simultaneous "Push" and "Pull" operations in the same time period, which can further reduce the update time, namely , If a "pull" occurs, the "push" operation in the corresponding interval is abandoned, and vice versa. Therefore, when UTD is equal to 0, all Pull operations will be prohibited, and the P&P model will be downgraded to a pure Push model. Similarly, when UTD is equal to 1, all Push operations will be prohibited, and the P&P model will be downgraded to a pure Pull model. In addition, "isPulled" and "isPushed" should be controlled by the synchronization model to avoid inconsistencies when simultaneously reading or writing.

2. When the value of UTD is relatively small, the Push method dominates. As explained in the P&P-Push model, because the conditions in the P&P-Push algorithm can be easily met, the Push operation is usually triggered. On the other hand, although the P&P-Pull algorithm tries to minimize the value of the Pull interval, when the Pull interval becomes very small , PULL_INTERVAL_MIN prevents this situation. In most cases, the "push" operation runs before the "pull" operation. Therefore, push-based methods have become mainstream. An extreme case is that the producer hardly has any state changes, so the "push" operation will not be triggered for a long time, but the P&P-Pull algorithm still pulls the producer to the maximum PULL_INTERVAL_MAX cycle to notify the producer of consumer availability.

3. When the value of UTD is large, the method based on Pull dominates. Since it is difficult to meet the condition of the Push operation is rarely triggered. However, the P&P-Pull algorithm adjusts its pull-up interval according to changes in state. In this case, the method based on Pull will dominate. In the extreme case, when the Pull interval is large, drastic changes that violate the UTD will occur during the large Pull interval. At this time, the "Push" operation will be triggered and the "abnormal" state will be pushed to the user. At the same time, the P&P-Pull algorithm tries to reduce the Pull interval, because the Push operation usually means that the conditions in the model are met.

4. When the value of UTD is relatively moderate, neither "push" nor "pull" dominates. This situation is only in the middle of the above two situations, and both Push and Pull methods often work.

Pure network performance metrics can be used for the evaluation of VANET data dissemination algorithms, such as packet loss, packet error, packet delivery ratios, end-to-end delay, normalized network load, and packet duplication. However, some techniques propose other corresponding metrics that can better evaluate specific application scenarios.

Models for traffic safety and delay-tolerant systems rely on two basic performance metrics: Mean dissemination delay and probability of successful message reception. Delivery ratio is an alternative measure for the latter that can be gathered via simulation results. In safety applications, it is required to minimize the delay and maximize data delivery. If an accident occurs on a road for example, it is required to disseminate a warning message in order to inform vehicles that are planning to visit the same road, so that they may search for an alternative route. Under high density circumstances, the arrival of more uniformed vehicles can shortly block the road. In this case, the data traffic overhead becomes a significant metric, which can be measured by the average number of duplicate messages received per vehicle.

Despite that delay-tolerant techniques accept relatively longer delay compared to traffic safety models, it is still required to keep the delay in a reasonable range. For example, if a vehicle sends a delay-tolerant query to indicate congestion status in a certain faraway road, it can wait for some time before receiving a reply. Since data is expected to travel far away from the requesting vehicle, message propagation rate measures the speed of message delivery process in delay-tolerant models.

Modeling techniques		Standard	Metrics	Road layout assumptions	Mobility assumptions	Network assumptions	Addressed challenges
Traffic safety	Time-probabilistic analysis [18]	Not mentioned	(1) Mean dissemination delay	– Linear network topology	– Fixed speed – Fixed distances between vehicles	– Synchronous transmission	– Interference
	Modeling with priority [19]	IEEE 802.11	(1) Average number of nodes that receive the high-priority message (2) The per-hop message forwarding distance.	– One-direction highway (linear topology)	– Vehicles may take over each other – Fixed-speed time intervals chosen from Gaussian distribution	– Two priority classes for messages (low, high)	– External and internal interference
	Warning delivery service [20]	Not mentioned	(1) Average delay (2) Probability of successful reception (3) Average number of duplicate messages	– One-direction multi-lane highway (linear topology)	– Lane change (overtaking) – Vehicles do not move during broadcast	– Error-free wireless channel – Worst-case (high density)	–
Delay-tolerant	Delay-tolerant message propagation [21]	IEEE 802.11p	(1) Message propagation rate	– Bidirectional highway	– Fixed speed – Fixed distances	– Delay tolerance network	–
	VADD [16]	IEEE 802.11	(1) Data delivery delay (2) Data delivery ratio (3) Data traffic overhead.	– Graph with different traffic modes: intersection, straightway and destination	–	– Sparse network	– Dynamic path finding

Figure 1. Performance modeling approaches for VANETs

The below table gives an example using the P&P model with a moderate UTD that equals to 0.25. Initially, the Pull interval is set to 50s and Push interval is set to 10s. During the first Pull interval, at 10:46:30, the Producer Pushes the status since the change degree of Push is 0.31>UTD. At 10:46:50, when the first Pull interval expires, Pull operation in this interval is forbidden since Push operations happen. When the change degree of Pull is 0.31>UTD, the Consumer decreases its next Pull interval to 40s. During the second Pull interval, there was no Push operation. Thus, at the end of the second Pull interval (time stamp: 10:47:30), the Consumer executes Pull operation and increases its third Pull interval to 50s. The next two Pull intervals also follow the rules explained above. The total updating cost is 5 including both Push and Pull operations compared to the cost 19 in the pure Push method.

Table 1

Time stamp	Sensor CPU Load (%)	Push change degree	Pull change degree	Push/ Pull operation	Value
10:46:00	53	0.53	\	Push	53
10:46:10	48	0.05	\	\	53
10:46:20	39	0.14	\	\	53
10:46:30	22	0.31	\	Push	22
10:46:40	27	0.05	\	\	22
10:46:50	18	0.04	0.31	\	22
10:47:00	30	0.08	\	\	22
10:47:10	15	0.07	\	\	22
10:47:20	8	0.14	\	\	22

UTD=0.25, Push interval=10s,

PULL_INIT_INTERVAL=50s,

PULL_INTERVAL_MIN=30s,

PULL_INTERVAL_MAX=120s,

Pull interval=Pull interval+/-10s

Two groups of experiments were conducted in the transmission pair. The first group of experiments aims to reveal the relation between updating number and UTD, and compare updating number of the P&P model with the pure
Push and pure Pull models. We set the sensor's updating period to 1s. Producer's Push interval was set to 1s, and Consumer's PULL_INIT_INTERVAL, PULL_INTERVAL_MIN, and PULL_INTERVAL_MAX were set to 5s, 3s and 12s, respectively. The Pull interval increased or decreased 1s each time.

The total updating number includes both the numbers of Push and Pull operations. As shown in Figure 3, the total updating number, Push operations' number, and Pull operations' number are closely relevant to UTD. As UTD rises, the total updating number decreases, and the number of Push operations drops dramatically, but the number of Pull operations grows slightly. The reason for this phenomenon is that the speed at which the number of Pull operations increases is much less than the speed at which the number of Push operations decreases. Another phenomenon is the proportion of Push operation decreases, while Pull operation accounts for more of the total as UTD grows. When UTD is 0, the P&P model degenerates to the pure Push model. When UTD is relatively low, take 0.15 or 0.2 for
For example, most of the operations are Push, whereas Pull operations only occupy a little percentage. So Push is dominant. If UTD (0.45 or 0.5) is relatively moderate, neither Push nor Pull operations exceed the other too much. When UTD (0.7 or 0.75) is relatively high, the number of
Pull operations are dominant. Finally, the P&P model becomes the pure Pull model when UTD is

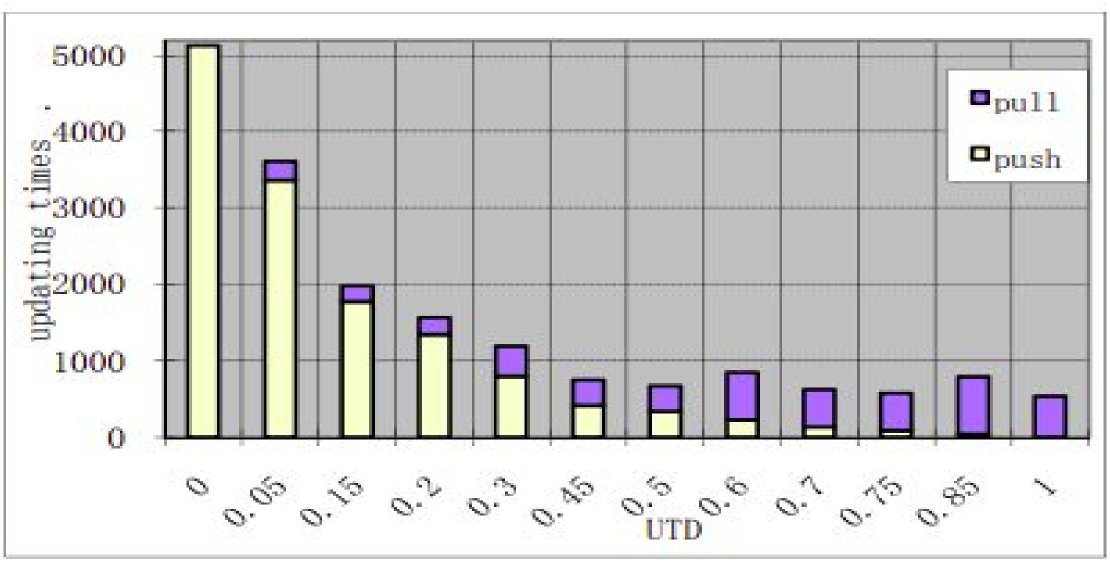

Figure 2. Updating number of P&P model at different UTDs

Also we observe that the updating number in the P&P model is much smaller than the updating number in the pure Push method (UTD=0), or the value of UTD is relatively small. This is because the threshold of the P&P Push algorithm depends on the users' requirements and reduces unnecessary Push operations. On the other side, the updating number of the P&P model is slightly greater than that of the pure Pull method (UTD=1) when UTD is relatively large since it involves a small number of Push operations.

Conclusions

In this report, we examined the algorithms for the data distribution in VANET networks. The main task was to propose suitable algorithms that can replicate the data, even in a VANET network environment. The algorithms PULL and PUSH were already present in the literature but both have some advantages and disadvantages so we tried to come up with a hybrid of PUSH and PULL algorithm which has advantages of both algorithms. The hybrid algorithm was subsequently tested in a simulation. The simulation showed that when using the hybrid algorithm, on board units are the units suitable as data distributors. The static road infrastructure units served primarily as triggers to spread new information. They also show that in most cases, the Dynamic Connectivity based Grouping method gives the highest accessibility, and the Static Access Frequency method gives the lowest traffic. In a real environment, a proper method among the three methods should be chosen based on the characteristics of the system.
In comparison with the pure Push model and pure Pull mode, the P&P hybrid model can effectively reduce updating numbers and maintain various levels of coherence in accordance with the users' requirements. In the future we are planning to test the replication methods with distributing multiple types of replications, and to solve the related problems.

References

1. Prasanna Padmanabhan, Le Gruenwald: Managing Data Replication in Mobile Ad-hoc Network Databases. In: 2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing, 17-20 Nov. 2006.
2. T. Hara: Data replication issues in mobile ad hoc networks. In: 16th International Workshop on Database and Expert Systems Applications (DEXA'05).
3. Fife, L. D. -- Gruenwald L.: TriM: Tri-Modal Data Communication in Mobile Ad-Hoc Networks. In: Lecture Notes in Computer Science. [s.l.]. Springer Berlin / Heidelberg. 2004. vol.3180. ISSN 0302-9743.
4. Rahbar, Afsaneh – Mohsenzadeh, Mehran – Rahmani, Amir Masoud: HDD3M: A New Data Communication Protocol for Heterogeneous Distributed Mobile Databases in Mobile Ad Hoc Networks. In: ICETC '09: Proceedings of the 2009 International Conference on Education Technology and Computer. Washington, DC, USA: IEEE Computer Society, 2009. ISBN 978-0-7695-3609-5.

5. Janech, Ján: Distributed Database Systems in the Dynamic Networks Environment. In: International Journal on Information Technologies and Security. Sofia: Union of Scientists in Bulgaria, 2010. n◦ 1. ISSN 1313- 8251.
6. J. Janech: Process Control in the distribution database, PhD dissertation, Faculty of Management Science and Informatics, University of Žilina, Žilina, Slovakia, 2010.
7. Micek, J. – Kapitulik, J.: Car-to-car communication system, Computer Science and Information Technology, 2009. IMCSIT '09. International Multiconference on, vol., no., pp.627-631, 12-14 Oct. 2009
8. Matiaško K – Zábovský M.: A data replication for the mobile environment. Journal of Electrical Engineering. 2008;59(5):277-80.
9. Fife, L. D. -- Gruenwald L.: Mobile ad-hoc network data communication in large geographic areas. In: Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE. 3-6 2006. vol.1.
10. Fife, L. D. – Gruenwald, L.: Research issues for data communication in mobile ad-hoc network database systems. In: SIGMOD Rec.. New York, NY, USA: ACM, 2003. vol.32, n.2. ISSN 0163-5808.
11. Bellavista, P., Corradi, A., Magistretti, E.: REDMAN: A decentralized middleware solution for cooperative replication in dense MANETs. In: International Conference on Pervasive Computing and Communications Workshops, pp. 158–162 (2005).
12. Bellavista, P., Corradi, A., Magistretti, E.: Comparing and evaluating lightweight solutions for replica dissemination and retrieval in dense MANETs. In: IEEE International Symposium on Computers and Communications, pp. 43–50 (2005).
13. Camp, T., Boleng, J., Williams, B., Wilcox, L., Navidi, W.: Performance comparison of two location-based routing protocols for Ad Hoc networks. IEEE INFOCOM, pp. 1678–1687 (2002).
14. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless Communication and Mobile Computing: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, pp. 483–502 (2002).
15. T. Hara, "Strategies for data location management in mobile ad hoc networks," Proc. of IEEE ICPADS 2005, to appear, July 2005.
16. H. Hayashi, T. Hara, and S. Nishio, "Cache invalidation for updated data in ad hoc networks," Proc. of Int'l Conf. on Cooperative Information Systems (CoopIS'03), pp.516-535, Nov. 2003.
17. H. Hayashi, T. Hara, and S. Nishio, "Replica allocation considering data update intervals in ad hoc networks," Proc. of IFIP/IEEE Int'l Conf. on Mobile and Wireless Communication Networks (MWCN 2004), pp.131- 142, Oct. 2004.
18. H. Hayashi, T. Hara, and S. Nishio, "Updated data dissemination methods for updating old replicas in ad hoc networks," ACM/Springer Personal and Ubiquitous Computing Journal, to appear, 2005