Data-based Agreement for Inter-Vehicle Coordination

Brahim Ayari, Abdelmajid Khelil, Kamel Saffar, Neeraj Suri
Computer Science Department
Technische Universität Darmstadt
Darmstadt, Germany
Email: {brahim, khelil, m_saffar, suri}@informatik.tu-darmstadt.de

***Abstract*—Data-based agreement is increasingly used to implement traceable coordination across mobile entities such as ad-hoc networked (autonomous) vehicles. In our work, we focus on data-based agreement using database transactions where mobile entities agree on a set of coordinated tasks that need to be performed by them in an atomic way. Atomicity means that all transaction participants agree on a set of tasks which will be performed by them or no one of them is performing any task. The data about the agreed tasks and their corresponding stakeholders are kept in local databases as a proof for the obtained agreement. This proof might be needed by users and regularities/authorities involved depending on the application scenario. In this demo, we demonstrate our effort to provide for partition-aware atomic commit protocols for transactional data-based agreement.**

I. Introduction

The pervasiveness and functionality of portable devices, equipped with wireless network interfaces is continuously increasing. Mobile devices are also increasingly equipped with small-footprint databases such as Oracle Database Lite 10g [1] and IBM DB2 Everyplace [2]. For consistent mobile data management, often mobile users require transactional services which are not explicitly planned in advance. Examples include the spectrum of mobile commerce scenarios, mobile DBs, and increasingly cooperative or autonomous driving through vehicle-to-vehicle communication [3]. Often a connection to a wide-area network such as the Internet may be unavailable due to lack of infrastructure or may be inconvenient or impractical due to the costs/expenses required for such a connection. Mobile Ad-hoc Networks (MANET) are mainly deployed to maintain a certain level of service availability when an infrastructure is unavailable. For instance they are used to support transactional services, such as in Vehicular Ad-hoc Networks, where communication between different entities should be set quickly. The achievable level of transactional service delivery in MANETs depends essentially on the basic transaction services provided by atomic commit protocols. Atomic commit protocols ensure strict atomicity of transactions and consequently play a major role in transaction processing. The support of distributed atomic transactions in MANETs is a key requirement for many mobile application scenarios. In this demo we focus on transactional inter-vehicle ad-hoc coordination, and demonstrate our work to provide for atomicity in the presence of frequent network partitioning.

Research supported in part by DFG GRK 1362 (TUD GKMM), EU INSPIRE, EU CoMiFin, and CASED (www.cased.de).

II. Overview of one Atomic Data-based Agreement Technique

The main problem of existing commit protocols such as 2PC or 3PC is either low commit rates and/or blocking behavior in MANET environments. Therefore, we present in [4] the Partition-Tolerant Atomic Commit protocol (ParTAC) for MANETs. To reduce Commit/Abort decision times and prevent the protocol from blocking, ParTAC follows a best-effort strategy by defining a lifetime for every transaction after which the transaction is aborted. Further, ParTAC introduces a new coordination strategy based on a flexible pre-selection of *multiple* coordinators among the participating nodes. Thus, the failure of a single coordinator can be tolerated in the presence of network partitioning. Moreover, transactions can be aborted by *any* coordinator based on lifetime expiration. As shown in [4] through simulations, ParTAC supports a significantly wide range of mobility patterns and partitioning scenarios than existing protocols. Our solution is valid for arbitrary MANET scenarios. in the following we demonstrate it for inter-vehicle scenarios and applications.

III. Application Scenarios

Coordination across autonomous networked vehicles is an excellent scenario for atomic transaction protocols in ad-hoc networks. It presents a potential application where mobile transactions are needed for the purpose of coordination for safe and traceable navigation of unmanned autonomous networked vehicles. Like airplanes, we assume in the following example that autonomous vehicles are equipped with black boxes which are basically mobile databases. This example describes a scenario where it is useful to equip (unmanned) vehicles with such black boxes. Fig. 1 shows four unmanned vehicles at a traffic intersection. These vehicles need to agree on an order how they will pass the intersection. Prior to their actual passing, this order information needs to be agreed upon and recorded atomically to their corresponding black boxes. This information would be absolutely useful for insurance companies or police in case an accident occurs between these vehicles.

Fig. 2 shows another unmanned vehicles scenario where four unmanned vehicles are moving on a road. Two of them are moving in the opposite direction of the other two. One could imagine a mobile market scenario where these vehicles are participating in a mobile transaction (selling and buying items) started in the near past and could not succeed because

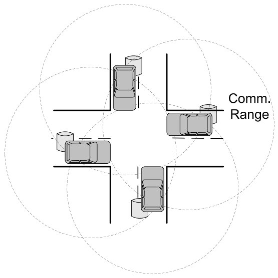

Fig. 1. Coordination across networked autonomous vehicles

the vehicles got partitioned before reaching a commit decision. When the four vehicles join the same partition before the expiration of the transaction lifetime, the final decision is agreed upon using ParTAC.

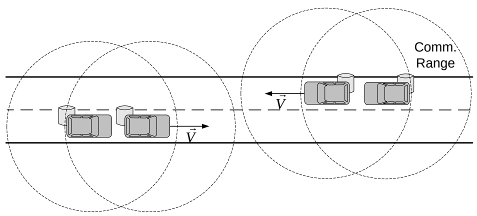

Fig. 2. Unmanned vehicles on the road

IV. Prototype Implementation

To demonstrate the inter-vehicle scenarios described above, we use four LEGO Mindstorms NXT 2.0 robots [5] equipped with ultrasonic and color sensor to play the role of the unmanned vehicles. We select the HTC Touch Diamond 2 Personal Digital Assistant (PDA) as our development platform as it provides a WiFi interface, large memory and a good development environment. Every robot is equipped with one PDA (Fig. 3) that represents the computation and communication unit of the unmanned vehicle. The PDA sends commands to the robot via Bluetooth. These commands control the movement of the robot and sensing activity of its environment. The unmanned vehicles communicate with each other using WiFi (IEEE 802.11b/g) interfaces of their corresponding computation and communication units (PDAs). We implement our application using J2ME (Java 2 Micro Edition). For this purpose, we install on every PDA a J2ME virtual machine (IBM J9 MIDP 2.0). We install on every PDA a mobile database (Mimer Mobile SQL) needed for the realization of the black boxes described in the inter-vehicle coordination scenarios in Section III. We implement the ParTAC protocol to coordinate the unmanned vehicles passage at the traffic intersection. Using their sensors, the robots are able to detect their position in the field and the information is transmitted to their corresponding PDA via bluetooth. Every vehicle in our demonstration has a unique ID.

V. Demonstration

For our demonstration we will place the vehicles (LEGO Mindstorms NXT 2.0 equipped with a PDA each) on a virtual traffic intersection as shown in Fig. 4. Using their sensors the vehicles are able to detect an intersection. The first vehicle which reaches the intersection will stop and detect the presence of other vehicles in the intersection by broadcasting a HELLO message. Every vehicle that receives this message, responds with its ID. Upon receiving the reply messages, the initiator vehicle sends a transaction to all the other vehicles present in the intersection. The order of passage of the vehicles is determined by the order of the messages received from the other vehicles present in the intersection. Using the ParTAC protocol, the application decides either to commit or abort the transaction. In case of the transaction commits, the intersection passage order of the unmanned vehicles is saved in the database and send to all other vehicles.

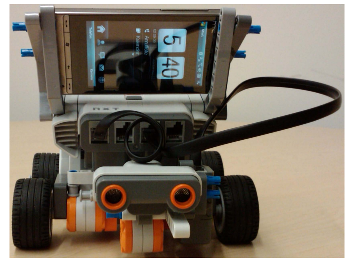

Fig. 3. LEGO Mindstorms equipped with HTC PDA

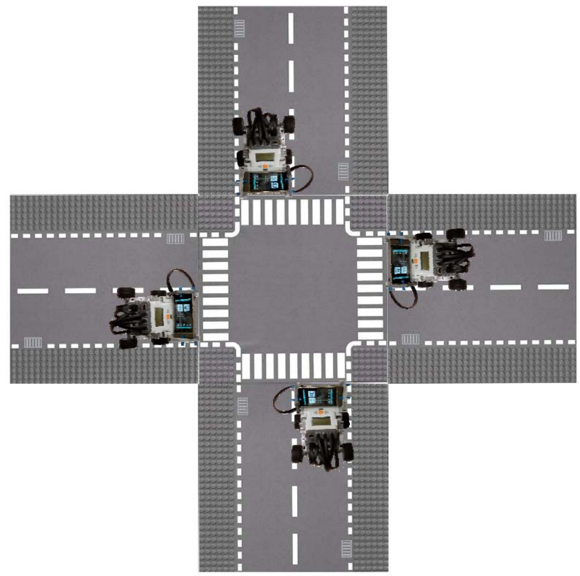

Fig. 4. Intersection scenario

References

[1] Oracle Database Lite 10g. Oracle Corporation. [Online]. Available: http://www.oracle.com/technology/products/lite/index.html

[2] IBM DB2 Everyplace. IBM Software Products. [Online]. Available: http://www-01.ibm.com/software/data/db2/everyplace/

[3] E. Huang, W. Hu, J. Crowcroft, and I. Wassell, "Towards commercial mobile ad hoc network applications: A radio dispatch system," in *Proc. of MobiHoc*, 2005, pp. 355–365.

[4] B. Ayari, A. Khelil, and N. Suri, "Partac: A partition-tolerant atomic commit protocol for manets," in *Proc. of MDM*, 2010, in press.

[5] LEGO Mindstorms. LEGO Website. [Online]. Available: http://mindstorms.lego.com