Self-managed and Blockchain-based Vehicular Ad-hoc Networks

Benjamin Leiding
University of Göttingen
Göttingen, Germany
benjamin.leiding@cs.uni-goettingen.de

Parisa Memarmoshrefi
University of Göttingen
Göttingen, Germany
memarmoshrefi@informatik.uni-goettingen.de

Dieter Hogrefe
University of Göttingen
Göttingen, Germany
hogrefe@informatik.uni-goettingen.de

UbiComp/ISWC '16 Adjunct, September 12-16, 2016, Heidelberg, Germany
ACM 978-1-4503-4462-3/16/09.
http://dx.doi.org/10.1145/2968219.2971409

Abstract

Combining Vehicle Ad-hoc Networks (VANETs) and Ethereum's blockchain-based application concepts enables transparent, self-managed and decentralized system which are self-regulating and in no need of a central managing authority.

Author Keywords

Self-managed VANET; Ethereum; Blockchain

ACM Classification Keywords

I.2.9 [Robotics]: Autonomous vehicles; K.4.4 [Electronic Commerce]: Cybercash; H.3.4 [Systems and Software]: Information networks; H.3.5 [Online Information Services]: Data sharing

Introduction

Most commonly proposed vehicular-ad-hoc-networks (VANET) concepts focus on centralized network structures which are similar to the road traffic regulations of the pre-VANET-age, controlled by governmental institutions. As often in computer networks, centralized control structures result in several downsides: Central control units are single points of failures and especially worthwhile targets for attackers. Besides outside attackers, centralized systems are also particularly interesting for entities which perform (non)-governmental surveillance. As a consequence, the following section introduce an alternative decentralized and

VANET Definitions

OBU: Is typically mounted onto a vehicle and used for exchanging information with RSUs or other OBUs. Information exchange is usually achieved via short range wireless communication or other radio technologies [2].

AU: The AU is closely linked to the OBU and might even reside in the same physical unit or is mobile and might be regularly removed from the vehicle (e.g smartphones). The AU runs applications that can utilize the OBU's communication capabilities [1][2].

RSU: RSUs are placed along the road side or in dedicated locations such as at crossroads. Usually, RSUs provide short range communication based on IEEE 802.11p radio technology but can also be equipped with other network devices in order to provide communication within the infrastructural network [1].

self-managed VANET structure based on the concept of Ethereum and a challenge-response-based authentication mechanism. Ethereum's contract system provides a framework for self-organization and self-management of the network using distributed applications on top of the blockchain technology.

VANETs

As illustrated in Figure 1, the basic components of VANETs are vehicles, on-board-units (OBUs), application-units (AUs) and road-side-units (RSUs).

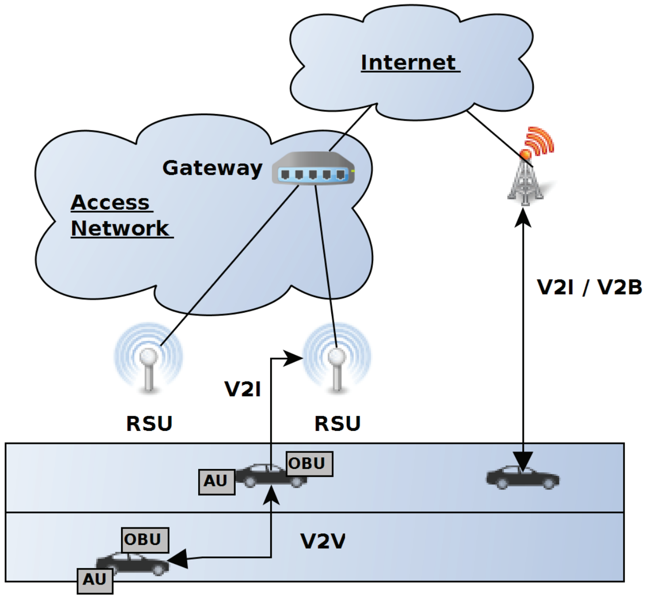

Figure 1: General VANET architecture (Based on: [2])

Communication in VANETs happens either inside a vehicle between AUs and OBU, wirelessly between different vehicles (V2V) or vehicles and infrastructure (V2I) or between vehicles and the infrastructure via broadband (V2B) [5]. For authentication purposes, each network participant is equipped with a unique public/private key pair which usually resides in a tamper-proof-device (TPD).

Ethereum

Ethereum is a decentralized platform which is based on Bitcoin's block chain concept. "What Ethereum intends to provide is a blockchain with a built-in fully fledged Turing-complete programming language that can be used to create 'contracts'" [3]. In the context of Ethereum, a contract is an instance of an application which runs on the block chain and consists of its program code, storage and an account balance [4]. Contracts are created by posting them (and their code) to the blockchain. "The contract's code is executed whenever it receives a message, either from a user or from another contract. While executing its code, the contract may read from or write to its storage file. A contract can also receive money into its account balance, and send money from its account balance to other contracts or users. The code of a contract determines how it behaves when it receives messages, under what conditions (and to whom!) it sends money out and how it interacts with other contracts by sending messages to them" [4]. Contracts are powered by "gas", Ethereum's internal fuel. Each computational step of the code execution requires a certain amount of gas and therefore prevents accidental or hostile infinite loops [3]. Contracts are implemented in one of Ethereum's high-level languages (e.g Serpent), afterwards compiled into bytecode and posted to the blockchain. The execution of the contract code itself "is part of the definition of the state transition function, which is part of the block validation algorithm, so if a transaction is added into block B the code execution spawned by that transaction will be executed by all nodes, now and in the future, that download and validate block B" [3].

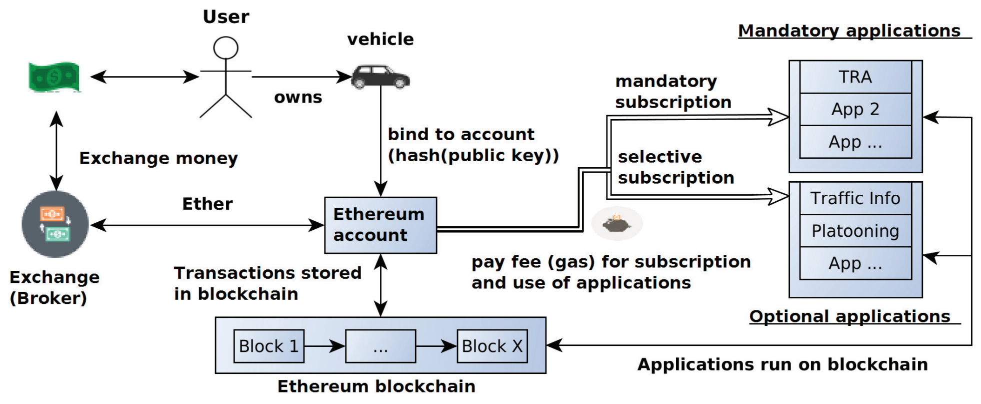

Figure 2: Ethereum-based service provision and rule enforcement in self-managed VANETs

Blockchain Definitions

Blockchain: A blockchain consists of a technically unlimited number of blocks which are chained together in a chronological order.

Transactions: Each block consists of transactions which are the actual data to be stored in the chain.

Mining: During the mining-process, miners collect valid transactions, compute a proof-of-work and the result is a new block which can be added to the chain. Each block depends on its predecessor block and therefore tampering and manipulating a block is quite difficult and requires an infeasible recalculation of all successor blocks. The mining process is used to establish a consensus of all participants of the network on a current state which will be used as ground truth for the next block.

Self-Managed VANETs

The following section now introduces the communication, organization and management of the network as well as the provided services. Our approach uses a similar network and communication structures as state of the art solutions illustrated in Figure 1. Vehicle either communicate among each other, with RSUs or, in case there is no RSU nearby, directly with the backbone system[1] of the network (e.g via GSM).
But in contrast to traditional VANET infrastructure, RSUs are not only used as a communication relay station, instead their additional purpose is the provision of Ethereum-based applications which have been deployed to the Ethereum blockchain. Users can subscribe to certain provided applications and make use of them if provided by the current RSU. Information on two example applications are provided on the next page's info box.

[1]in this context we use the term backbone as general and abstract description of all hardware which is not located directly at the roadside such as RSUs

Figure 2, illustrates the actual subscription process and usage of deployed applications for network participants. Each network entity has a Ethereum address (hash of its public key) and is registered as participant in the Ethereum blockchain. In context of this work, Ethereum powers certain applications which are used by other network entities. These applications are either used to enforce certain rules (traffic regulation) or provide different services (traffic jam notification, weather information, platooning, etc.). As a result, the network regulates itself based on the deployed applications and regulations.
Costs for running the required infrastructure and the Ethereum applications is somehow self-regulating. Changing the required real world money into Ether (Ethereum's internal cryptocurrency) works the same way as known from Bitcoin and usually involves a Ether-exchange service. Each message/transaction which involves an Ethereum application requires Ethereum-gas, the internal pricing for running a transaction or contract in Ethereum, for calculation. So each car has to pay these fees while using the mandatory

Application Examples

Traffic Regulation Application (TRA): It is also possible to implement all traffic regulations using Ethereum's high-level languages and make it mandatory for all participants. Based on exchanged traffic data, automated identification of cars, video-monitoring (eg. at crossroads) and other traffic related data, it is possible to identify misbehaving cars and punish them accordingly. This might include speeding, ignoring traffic lights, causing an accident, etc. Since the car can be directly identified due to their public key, it is possible to directly bill the the corresponding Ethereum account and punish the car owner.

Vehicle Tax: In most countries (including Germany) each owner of a vehicle is required to pay vehicles taxes every year and get insurance for each vehicle which covers varying levels of damage to the car or caused by the vehicle owner. In context of our system, paying the yearly amount of taxes could be automated and done without human interaction. As soon as a vehicle is registered to the system, the required amount of money (ether) is transferred from the Ethereum account linked to that car, to the Ethereum account of the government which collects the tax.

applications and therefore funds the required infrastructure and the computation. The side effect of this approach is that power-user which use the infrastructure more than others pay more than users which use the system less frequently. Since the network participants basically pay for the costs of running and maintaining the infrastructure, there is quite a big incentives for actually providing RSUs and necessary equipment. Ideally, RSU hardware and software is standardized up to a certain point (with optional additional modules for further services) and can be deployed by network participants without the need of a large infrastructure provider (even though that is also a scenario). An additional incentive for providing RSUs and other infrastructure components are mining-fees as known from Bitcoin. Users pay (as part of the gas for running the applications) a small fee to the miners for mining collected transactions into a new block of the block chain. Mining might be done by the RSUs, components in the backbone system or by external computing providers which earn money by mining transactions (as currently done by several mining-companies or mining-pools in the Bitcoin network).

Conclusion

In this work, we presented a novel concept of a decentralized, self-managed and blockchain-based vehicular ad-hoc network. Using Ethereum's smart contract system it is possible to deploy and run any type of application on the Ethereum blockchain which, in combination with state of the art vehicular ad-hoc networks, enables the deployment of applications for self-managed and decentralized network operation. We proposed the use of mandatory applications such as traffic regulation application, vehicle tax and vehicle insurance applications or other applications which enforce network rules and regulations. In addition, we proposed also optional applications which provide service that can be subscribed to by users if necessary. These optional applications may provide information and updates on traffic jams or weather forecasts. But due to Ethereum's Turing-complete high-level languages, it is also possible to deploy any other kind of application.

REFERENCES

1. Al-Sultan, Saif and Al-Doori, Moath M and Al-Bayatti, Ali H and Zedan, Hussien. 2014. A Comprehensive Survey on Vehicular Ad Hoc Network. *Journal of Network and Computer Applications* 37 (2014), 380–392.

2. Roberto Baldessari, Bert Bödekker, Matthias Deegener, Andreas Festag, Walter Franz, C Christopher Kellum, Timo Kosch, Andras Kovacs, Massimiliano Lenardi, Cornelius Menig, and others. 2007. Car-2-Car Communication Consortium - Manifesto. (2007).

3. Buterin, Vitalik. 2014. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. URL: `https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-White-Paper`. (2014). (Accessed May 06, 2016).

4. Delmolino, Kevin and Arnett, Mitchell and Kosba, Ahmed and Miller, Andrew and Shi, Elaine. 2015. A Programmer's Guide to Ethereum and Serpent. URL: `https://mc2-umd.github.io/ethereumlab/docs/serpent_tutorial.pdf`. (2015). (Accessed May 06, 2016).

5. Miad Faezipour, Mehrdad Nourani, Adnan Saeed, and Sateesh Addepalli. 2012. Progress and Challenges in Intelligent Vehicle Area Networks. *Commun. ACM* 55, 2 (2012), 90–100.