Remote Vehicle Diagnostics over the Internet using the DoIP Protocol

Mathias Johanson
Alkit Communications AB
Mölndal, Sweden
mathias@alkit.se

Pål Dahle
Volvo Car Corporation
Gothenburg, Sweden
pdahle@volvocars.com

Andreas Söderberg
SP Technical Research
Institute of Sweden
Borås, Sweden
Andreas.Soderberg@sp.se

***Abstract*—Next generation vehicles will provide powerful connectivity and telematics services, enabling many new applications of vehicle communication. We will in this paper study the opportunities of performing remote vehicle diagnostics, where the diagnostic tool (test equipment) and the vehicle are separated by an internetwork, e.g., the Internet. The development of a prototype system for remote vehicle diagnostics, based on the emerging Diagnostics over IP (DoIP) ISO standard, is presented and early usage experiments with synchronous remote diagnostic read-out and control are described. A number of safety related issues are identified that will need closer study before a broad deployment of remote diagnostics services is feasible. Furthermore, a classification of vehicle diagnostics applications is provided, which is intended to elucidate the differences between synchronous (online) and asynchronous (offline) operation in local and distributed settings.**

Keywords-vehicle diagnostics, vehicular communication

I. Introduction

Access to diagnostic data from Electronic Control Units (ECU) in vehicles is of great importance in the automotive industry, both from a life cycle support perspective and during product development. Through diagnostic services, the state-of-health of components and subsystems can be monitored to detect and prevent failures by means of predictive maintenance, which improves operational availability and lowers support costs. For pre-series test vehicles, diagnostic services are crucial in order to be able to track problems as early as possible in the development process, preventing serious faults to pass undetected into production vehicles or as a tool during verification and validation activities. In the aftermarket, diagnostics form an important part of the service and maintenance process, with Diagnostic Trouble Codes (DTC) routinely being read out from customer vehicles during service for state-of-health monitoring and fault tracing. Automotive manufacturers rely on diagnostic systems in order to improve customer satisfaction by increasing the service technicians' ability to diagnose and remedy problems in the increasingly complex electronically controlled vehicles. As an added value for the automotive manufacturer, the diagnostic data retrieved during service can be uploaded to the manufacturer's database over the Internet. Statistical analysis of collected DTCs is important in order to monitor the quality of components and subsystems, to prioritize in which order problems should be addressed and to find correlations between different faults, or between faults and the operating environment.

A. Remote vehicle diagnostics

With the tremendous proliferation of wireless communication networks, telematics systems and services have been designed that make it possible to access diagnostic data from vehicles remotely, without requiring physical access to the vehicle. Presently, telematics services for diagnostics of general purpose passenger cars are mainly used during testing and validation of pre-series vehicles, but aftermarket services are also emerging in premium segments, for improved service and maintenance offerings [1]. Next generation vehicles will have sophisticated on-board connectivity equipment, providing wireless network access to the vehicle for infotainment and other telematics services. This will make it possible to realize remote diagnostic services for large-scale collection of diagnostic data from ECUs at level previously unattainable. Furthermore, this will enable many new aftermarket services and will also improve the opportunities of collecting diagnostic data for use during product development.

B. Integrated automotive diagnostics

Since automotive diagnostic systems are important both for aftermarket services and during many stages of product development, a common framework for capture, analysis and management of diagnostic data is highly desirable. Campos et al. argue that previous generations of diagnostics systems have not been well integrated, resulting in unnecessary duplication of effort in developing different diagnostic applications, each with its own infrastructure, components and software [2]. This leads to inefficient use of resources and high costs for development and maintenance of the diagnostics applications.

The key to realizing integrated diagnostic systems is to rely on standardized interfaces for communication and systems integration and to base the diagnostic software development on a component-based software architecture. This facilitates re-use of software components and makes integration of components and subsystems from many different vendors possible in an interoperable way.

Automotive diagnostics has a long history of standardization efforts, driven both by industrial inter-operability initiatives and legislation. One recent such effort is the emerging DoIP standard.

II. The DoIP standard

The standardization of automotive diagnostics technology was initiated by legislative regulations for emission control. These initiatives have led to numerous

 ISBN: 978-1-61208-166-3

standardization efforts of automotive diagnostic services, on virtually all technological levels, from hardware interfaces to communication protocols and software APIs. The perhaps most visible and influential initiative to date is the OBD-II specification issued by the California Air Resource Board (CARB), which is now mandatory for all cars sold in the US and the EU. Building further on this, the United Nations has initiated work on a new standardization framework called WWH-OBD (World Wide Harmonized On-Board Diagnostics), with the aim of rendering regional standards of vehicle diagnostics for emission control unnecessary and to replace them with a global standard. Moreover, this new standard will be a great leap forward in terms of new technology and protocols, enabling entirely new applications and services. One of the results of the WWH-OBD effort is the choice of using the Internet Protocol (IP) for communication between off-board and on-board diagnostic systems and for this purpose the Diagnostics over IP (DoIP) protocol is being developed by ISO, the International Organization for Standardization, under the formal name ISO 13400 [3].

The main motivation for introducing IP into the family of automotive diagnostics protocols is that the recent developments of new in-vehicle networks has led to the need for communication between external test equipment and on-board ECUs using many different data link layer technologies. To avoid having to implement, maintain and optimize transport and data link layer protocols for each new communication equipment development, and to easily be able to introduce new physical and data link layer technologies, a common internetworking protocol is needed, which is exactly what IP was designed for.

There is however a very interesting side-effect of this choice of network protocol, since it will improve the opportunities of interconnecting in-vehicle networks with the Internet for many new applications, including online, remote automotive diagnostics, which is the focus of this paper.

A. DoIP protocol overview

The ISO 13400 standard consists of four parts:

- Part 1: General information and use case definition
- Part 2: Transport protocol and network layer services
- Part 3: IEEE802.3 based wired vehicle interface
- Part 4: Ethernet-based High-speed Data Link Connector

In Part 1, the use cases that have guided the design of the protocol are outlined and a number of typical communication scenarios are described. Five main use case clusters are identified: (i) Pre-defined information request (such as state-of-health monitoring or road-worthiness assessment), (ii) vehicle inspection and repair (e.g., vehicle diagnostic fault tracing or vehicle readiness qualification), (iii) vehicle/ECU software reprogramming (i.e., firmware upgrade of ECUs during service or manufacturing), (iv) vehicle/ECU assembly line inspection and repair (similar to (ii) but in a manufacturing environment) and (v) multi-purpose data transfer from and to the vehicle, which involves non-diagnostic data exchange between vehicle and external equipment, including mobile customer equipment such as smart phones or PDAs.

The use case descriptions and communication scenarios described in ISO 13400-1 shows a considerable focus on communication between an in-vehicle network and external equipment in the immediate vicinity of the vehicle, such as test equipment connected through an Ethernet cable or a local area network (LAN), or mobile devices connected through wireless LAN (WLAN) technology. Uses cases such as the one focused in this paper, i.e., the opportunity of doing vehicle diagnostics with the external test equipment (or mobile device) being arbitrarily far away from the vehicle, interconnected by a true internetwork (i.e., a routed, packet-switched network like the Internet) is not specifically discussed. This is also reflected in the design of the DoIP communication protocol itself, for instance in the reliance on subnet broadcasts for vehicle announcements.

Part 2 defines network and transport layer protocols and services for vehicle diagnostics. This includes IP address assignment, vehicle announcement and vehicle discovery, connection establishment, communication protocol message format, data routing to in-vehicle nodes, status information and error handling. The focus on applications where the external test equipment is in the immediate vicinity (i.e., on the same subnetwork) as the vehicle is manifest primarily in the mechanism designed for vehicle announcement and discovery. This mechanism is intended to make external test equipment aware of the IP address and Vehicle Identification Number (VIN) of the vehicles connected to the same subnetwork. This is performed through subnet broadcasts of Vehicle Announcement and Vehicle Identification Request messages. Once the external test equipment has learned the IP address of a vehicle, a direct TCP connection to the vehicle's gateway node can be established, and the diagnostic data (or other data) exchange can be initiated. The message format designed for carrying the data is a lightweight message format based on a generic header and a payload specific header. The 8 byte generic header contains the DoIP protocol version number, payload type identifier and payload length field. The payload format for diagnostic data exchange adds a 4 byte header containing the 16-bit source and destination addresses (identifying the test equipment and ECU respectively), followed by the variable length data (up to 4 Gbytes). The connection set-up and data exchange can be carried out according to the DoIP specification regardless of whether the external test equipment and the vehicle are on the same local network or separated by an internetwork, providing that some mechanism external to DoIP is used for vehicle discovery. This is the basis for the remote online diagnostics application that will be described in detail in Section V.

Parts 3 and 4 of the standard specifies the data link layer and physical layer requirements, which are based on the Ethernet (IEEE 802.3) protocol and the ISO 15031-3 (SAE J1962) connector.

Note that, despite the name Diagnostics over IP, the DoIP protocol specifies several payload types that are not directly related to diagnostics in terms of the ISO 14229 scope [4]. (Only payload types 8000 and 8001 are intended for ISO14229 diagnostics.)

 ISBN: 978-1-61208-166-3

III. Remote Online Diagnostics

We use the term Remote Online Diagnostics to refer to data communication for vehicle diagnostics between one or more in-vehicle network nodes and an external test equipment that are interconnected by an internetwork. Thus, "remote" here means that the communication endpoints are not required to be connected to the same local subnetwork. This means that the physical distance between the external test equipment and the vehicle can be arbitrarily large, providing there is a network infrastructure available. We use the "online" qualifier to characterize our intended use of the DoIP protocol to perform diagnostic data exchange synchronously over a TCP connection set up between the endpoints. This is in contrast to "offline" or asynchronous diagnostic data exchange being performed by an on-board test equipment that performs the read-out locally in the vehicle, possibly remotely triggered, and then uploads the result to a server at a suitable time using whatever network connection is available.

A. A classification of vehicle diagnostics

It will be useful to study the different modes of diagnostics a bit more closely, to identify possible applications and to distinguish the technological solutions needed to implement them, their advantages and drawbacks. We will start this by classifying vehicle diagnostics applications according to whether the diagnosis is performed with the vehicle and the external test equipment being in the same place (connected to the same local subnetwork) or in different places (connected by an internetwork) and whether the diagnostic data exchange is performed synchronously (at the same time) or asynchronously (at different times). This classification, inspired by the classic time/space taxonomy of Groupware by Ellis et al. [5] is shown in Figure 1.

	Same time (synchronous)	*Different times (asynchronous)*
Same place (local network)	Traditional Diagnostics	Local Offline Diagnostics
Different places (internetwork)	Remote Online Diagnostics	Remote Offline Diagnostics

Figure 1. Time / space taxonomy of automotive diagnostics applications

The *same place / same time* case is the "traditional" diagnostic application, wherein a service technician (or automotive engineer) connects an external tester to the vehicle's OBD-II connector, reads out and analyzes diagnostic data for fault tracing or state-of-health purposes.

The *different places / same time* case is the application we focus on in this paper (remote online diagnostics), which gives the possibility for a service technician or engineer to do the same diagnostic read-out and fault tracing without being at the same place as the vehicle. A specific use scenario might be that a customer detects a malfunction in a vehicle and calls a service technician for support. The technician can then perform the fault tracing remotely and online, detecting and possibly solving the problem, and instructing the customer on how to proceed.

The *different places / different times* case is a remote offline diagnostic application. A typical example of when this type of service is useful is when large scale diagnostic data collection from a fleet of test vehicles (or possibly customer vehicles) is set up to gather performance data or statistics for use in product development. In such a scenario, a batch of diagnostic queries is scheduled for download to a number of vehicles. At a suitable time (when they have come online), the vehicles' telematics systems download and execute the diagnostic queries, assemble the responses, and upload the results to a central database, possibly at a much later time.

The *same place / different times* case does not have as immediately obvious applications as the others, but one can envision a situation where a service technician (or an automotive engineer) performs time consuming diagnostic tests of vehicles available locally, by downloading a diagnostic script file to an onboard tester that performs the tests, assembles the results, and then sends the results back to the test equipment (or a server), notifying the technician when the process is done.

The most interesting case for analysis in our present context is the distinction between the online and offline modes of remote diagnostics.

B. Diagnostic read-out versus diagnostic control

A distinction must be made between diagnostic read-out and diagnostic control. The purpose of diagnostic read-out is to query the status of the ECUs, typically by reading out DTCs for fault tracing or state-of-health applications. In diagnostic control applications, diagnostic commands that may alter the behavior of the vehicle are generated, for instance to turn the lights on and off. Thus, diagnostic read-out is a read-only operation, whereas diagnostic control is read/write.

C. Wireless versus wired diagnostics

Note that our definition of remote versus local diagnostics does not depend on whether the communication is performed using wired or wireless networks. A wireless local diagnostic application is for instance when a service technician connects to a locally present vehicle over a short-range wireless communication technology such as Bluetooth or IEEE 802.11 for diagnostics. In the wireless remote diagnostics case, some wide area wireless network technology is used (such as GPRS, 3G or 4G), or a combination of short range wireless communication and wired networks.

D. Online versus offline diagnostics

Although elements of the DoIP standard could be used to implement both the online and offline modes of remote diagnostics described above, it is clear that the DoIP protocol has been primarily designed with synchronous operation in mind. Since the main use cases that governed the design of the protocol are actually in the *same place / same time* category of Figure 1, this is not surprising. An interesting point to observe is that systems designed for *same place / same time* applications can, if implemented using the DoIP protocol, with very minor changes be used also for *different places / same time* applications, i.e., for remote online diagnostics. For instance, a traditional diagnostic read-out tool used in a service repair shop for fault tracing could with small modifications be used to remotely diagnose a vehicle on another continent. A drawback of using the online approach for remote

 ISBN: 978-1-61208-166-3

diagnostics is that applications that perform a complete diagnostic read-out of DTCs from all ECUs typically generate a large number of query/response transactions. With a considerable round-trip delay, as is often unavoidable in internetwork configurations, this can lead to a long total read-out time. The obvious remedy for this is to instead download a batch of queries, perform them locally in the vehicle, assemble the responses and send back. This is the offline approach described above. However, it is not always easy to design a generic batch of diagnostic queries, since the choice of which queries to include depends on the answer to previous queries. This means that a lot of logic needs to be present in the onboard tester in order to be able to execute the diagnostics properly in all situations. It is generally beneficial to keep this complexity at the infrastructure (server) side, rather than in the vehicle.

The main technological difference between the synchronous and the asynchronous case is that in the synchronous case the diagnostic queries or commands are sent by the external test equipment and directly responded to by the ECUs, whereas in the asynchronous case there is a time difference between query and response, and the network connection is not required to be kept alive during this time interval in the asynchronous case. The division between the two is not clear-cut however, and one can imagine hybrid approaches combining the two modes.

E. Remote online diagnostics using DoIP

As previously discussed, the core of the DoIP protocol can be used unmodified for remote online diagnostics, provided that the vehicle discovery and identification mechanism is supported by some additional means. Recall that the problem of the DoIP-mechanism for vehicle announcement and discovery is that it relies on subnet broadcasts, and thus these messages will not be accessible outside the local IP subnet the vehicle is connected to. One approach to overcome this problem is to establish a Virtual Private Network (VPN) connection from the vehicle to some enterprise network from where the operation of remote testers is supported. Alternatively, the VPN connection is terminated at a proxy server that listens to the vehicle announcements and keeps track of the IP addresses and VIN identifiers of the connected vehicles. The test equipment also connect through VPN to the proxy server, send vehicle identification requests, and receive the VIN identifiers and IP addresses of the currently connected vehicles. Clearly this approach will have scalability implications, when a very large number of vehicles are connected, typically for aftermarket applications. Performance scalability issues at the server side can be easily resolved by scaling up the number of proxy servers for load-balancing, using some simple heuristic method for deciding which server handles which subset of vehicles (e.g., based on IP subnet masks or similar). The problem that will appear at the external tester side is that the tester might get overly many responses to an unqualified vehicle identification request (i.e., a vehicle identification request message without VIN or EID). This can be resolved by only allowing vehicle identification request messages with EID or VIN at the proxy servers. Another problem is that all vehicle announcement messages will be propagated to the connected external testers, which might cause network connection congestion or processing overload. This can be solved by filtering out vehicle announcement packets from the VPN connections of the external testers. A side benefit of using a VPN based approach is the resolution of several security issues.

An alternative to the VPN approach to the vehicle identification problem is to develop a dedicated vehicle identification mechanism for remote online diagnostics applications. In the prototype application development described in Section V, a very simple vehicle identification mechanism is used, wherein each vehicle that comes online connects using TCP to a proxy server, reports its VIN number and then waits for a DoIP session to begin (keeping the TCP connection alive). The external tester connects to the proxy, queries for a particular VIN and if the vehicle is connected to the proxy the two TCP connections are interconnected and the DoIP session can begin. An additional benefit of this approach is that it also solves the problem that appears if the vehicle is not assigned a public IP address, due to Network Address Translation (NAT) firewalls being used.

For security reasons, and practical reasons, it might be desirable to let the vehicles use private IP addresses. This is often the case with addresses being assigned to mobile network devices in commercial wireless Internet access services. The problem with this is that private IP addresses are not reachable from outside; all communication sessions must be initiated from the mobile device (the vehicle in our case). Both mechanisms for vehicle discovery described above avoid this problem by having the VPN and DoIP TCP connections respectively initiated from the vehicle side.

IV. Safety aspects of remote diagnostic operations

Introducing the possibility to remotely control a vehicle using diagnostic operations creates a new range of safety related problems to address.

Safety can generally be divided into two main cases; safety in normal operation and safety for a system that is under influence of one or several system faults. The former, safety in normal operation, mainly addresses the task of creating a system that is safe with respect to usage, whereas the latter is about what is generally referred to as functional safety or system safety. This involves building more reliable or even fault tolerant systems and addresses issues about the process of reducing faults due to systematic (i.e., design) errors.

A. Normal operation

By introducing a remote diagnostic function, even if used by trained multi-skilled technicians, we may have introduced the possibility of the following new safety implications:

- the mechanic cannot directly observe the situation that the vehicle is in,
- the mechanic may not get visual feedback on what is really happening with the vehicle when it is under diagnostic control,
- the mechanic cannot interact with the vehicle in any other ways than using the terminal and an established communication session,

 ISBN: 978-1-61208-166-3

- the connection between the operator and the vehicle may be unreliable in terms of latency and bandwidth,
- there might be significant (non-deterministic) delays between the issuing of a diagnostic command and the moment when action is taken in the vehicle,
- there may be persons nearby or even inside the vehicle, e.g., the driver of the vehicle.

In connection to the prototype development of a remote diagnostics system described in Section V, a safety mechanism involving the remote user in diagnostic actions has been designed. In this solution we have concluded that

- the user of the vehicle needs to confirm her or his presence at the vehicle,
- the user needs to understand and subsequently approve the action to be taken,
- the user needs to be in charge of triggering of the remote action.

The mechanic, with diagnostic and service expert knowledge, is initiating the diagnostic request by down-loading a diagnostic task to the vehicle. The mechanic has to be in contact with the remote user (e.g., by phone) to be able to give instructions and get confirmation of understanding and approval to proceed. Presence control can easily be achieved by interacting with the vehicle (e.g., entering a code in the vehicle). Finally a trigger device (e.g., the remote key-fob) connected to the vehicle will trigger the diagnostic action to be taken.

It is believed that pure diagnostic read-out poses no safety risk, whereas only a limited set of diagnostic control actions can be considered safe under all circumstances. A large amount of actuators in the vehicle are risk related, especially in certain situations, such as when the vehicle is moving. Approval of safety limited synchronous diagnostic control therefore leads to a complex task of actuator safety classification. Furthermore, combinatory effects between sensors and other actuators complicate this matter even further.

B. Functional safety

A soon to be finalized ISO standard being applied intensively by many vehicle OEMs, ISO26262 [6], that addresses functional safety for E/E systems within passenger cars is the natural starting point when studying the system safety aspects of the diagnostic (sub-)system. The standard, which comes in 10 parts, has been jointly developed within a global automotive engineering community for the last 5-10 years. It is expected to become the de facto platform for system safety within the automotive domain, since it spans the fields of system engineering, hardware and software development, but also is specifically tailored to fit how automotive development is traditionally organized by OEMs and suppliers.

Specifically, we have done work within the "concept phase" (part 3 of the standard) by considering the diagnostic sub-system as the system under focus in the *Item definition*. This has proven to be difficult considering the natural characteristics of the diagnostic system: it contains limited functionality, but spans virtually all (electrical) sensors and actuators in the vehicle. Moreover, the system is constantly expanding as new sensors and actuators are introduced in the vehicle and it is hard to predict what the function developers will introduce in the future. Thus, the key has been to find a generalized way to analyze the system faults instead of looking at specific actuators that may be involved in the cause of the hazard. The general findings need then be applied at the various subsystems that use diagnostics as a tool, by considering faulty diagnostics as a source of hazards as well as any other root cause.

Note that nothing of the above makes any difference between traditional off-board diagnostics and remote diagnostics. The diagnostic subsystem is present even in today's vehicles. However there is one specific difference: the test equipment that is traditionally connected to the OBD connector in the vehicle would now usually (from a business case point of view) be integrated within the vehicle and is always present even if inactive. This *internal tester* needs special attention when it comes to the analysis of the source of any hazards.

V. Prototype system implementation and experiments

In order to gain practical experiences from remote online diagnostics and to explore how this can be realized using the DoIP protocol, a prototype system was implemented and tested in a controlled environment. Since no vehicle with an on-board DoIP gateway was available, it was decided that a DoIP gateway would be implemented on a Linux-based telematics system that could be connected to a standard vehicle's CAN buses through the OBD-II connector. The telematics platform has GPRS, EDGE and WLAN network interfaces as well as Ethernet interfaces. The DoIP entity implemented in the telematics unit handles the routing of diagnostic data between the in-vehicle (CAN) networks and the DoIP TCP connection on the wireless network interfaces.

To avoid having to develop a full-fledged diagnostics application from scratch the aftermarket diagnostics software VIDA, developed by Volvo Cars, running on an ordinary Windows PC was used as the external tester. Since there were no DoIP functionality implementation in VIDA at the time of this work, and since the implementation of this in VIDA itself was deemed not to be feasible within the time frame of the project, the client side DoIP interface was implemented in a dynamically liked library (DLL) that VIDA can access through the J2534 interface. This way we were able to develop an online remote vehicle diagnostics system without modifying the vehicle or the actual diagnostics tool.

With this approach, the diagnostics application (VIDA) on the PC will communicate ISO 14229 diagnostic messages through the J2534 DLL in the same way as if the PC was connected directly to the vehicle's CAN bus. What really happens is that the DLL encapsulates the ISO 14229 messages in DoIP messages that are transmitted over the IP network to the DoIP gateway in the vehicle, that decapsulates them, relays them onto the CAN bus, reads and assembles the responses (if any) and returns over the DoIP connection back to the DLL that forwards the result to the application. The diagnostics application can then process the response and go on to send the next query. The DoIP protocol is in this situation completely transparent to the diagnostic application.

Except for being a resource efficient way to implement our prototype system for experimentation, demonstration

 ISBN: 978-1-61208-166-3

and proof-of-concept, this approach is also interesting in that it provides a way to integrate diagnostics software completely unmodified into a DoIP-based infrastructure. This could help migration towards DoIP of the large installed base of tools and services based on J2534. A drawback of the design is that some of the complexities of the transport protocol used for implementing ISO 14229 diagnostics services over CAN (i.e., ISO 15765-2), such as the management of flow control filters, needs to be duplicated between the DLL and the DoIP gateway in the vehicle.

A. Experiments

The experiments carried out with the remote online diagnostics system prototype was first of all to demonstrate that a complete diagnostic read-out session could be performed over a wireless Internet connection, using the GPRS interface of the telematics unit. The PC was located in an office environment connected to the Internet using a LAN connection. A complete read-out of DTCs and additional data from the approximately 20 ECUs on the two CAN buses of a Volvo V70 takes around 10 minutes over a GPRS network connection. This is primarily due to the significant round-trip delay in GPRS networks. When using a WLAN connection, significantly shorter read-out times were measured: around 3 minutes, which is similar to local read-out using a directly connected CAN device.

In addition to the DRO experiments, diagnostic control commands were also tested, for instance recording of pedal positions, with real-time visualization of the pedal positions in the diagnostic application. Commands requiring write access were also tested, but limited to relatively safe operations, in the context of the experiments, like turning the engine fan or the lights on and off.

In principle, remote ECU reprogramming should also be possible to do in this way, but this was not tested, due to practical obstacles. In practice, remote reprogramming of ECUs is much more likely to be implemented based on a remote offline diagnostics model. This is because reprogramming of ECUs is typically time consuming, and the requirement to keep an online connection alive throughout the reprogramming will in many cases be failure prone. If the connection is disrupted during the reprogramming, the entire session will have to be rolled back. A better alternative is to download the software update to the vehicle asynchronously, perform the reprogramming in offline mode, and then reestablish the connection to report the status. Such an approach is described by Nilsson and Larson [7].

VI. Conclusions

In this paper we have shown how remote online vehicle diagnostics can be realized based on the DoIP protocol. To define what we mean by remote online diagnostics, we performed a classification of automotive diagnostics applications, based on whether the diagnosis is performed over a local network or over an internetwork spanning an arbitrarily large distance, and whether the diagnostic session is synchronous or asynchronous. We then outlined the salient features of the DoIP protocol, which has been designed first and foremost for synchronous, local applications. However, since DoIP is using the IP protocol, which is also the network protocol of the Internet, truly remote diagnostic applications are possible. The feasibility of designing such remote, online diagnostic applications was demonstrated through a prototype implementation, wherein a legacy vehicle diagnostics system was adapted to use the DoIP protocol. Experiments with the prototype shows that remote diagnostic read-out over relatively narrowband wireless internetworks is possible. Remote diagnostic control applications were also demonstrated.

One of the biggest challenges for introducing remote vehicle diagnostic services at a large scale is how to ensure the safety of the users of the vehicles. Our safety analysis shows that pure diagnostic read-out can be safely implemented, whereas diagnostic control applications in the general case are problematic. A related critical issue is how to protect a remote diagnostic service from illicit malevolent access. A comprehensive analysis of security issues in remote vehicle diagnostics is currently being conducted in relation to the work being presented here. The outcome of this analysis will have a profound impact on the design of the remote diagnostic system.

Our main conclusion from this work is that the DoIP protocol, when deployed broadly throughout the automotive industry, will enable many new applications of remote vehicle data access and control. This will pose many challenges in terms of performance, scalability, security, safety and resource management, but will at the same time give rise to very interesting new added-value services for the customers, and will also bring great opportunities to improve automotive product development.

Acknowledgment

This work was supported by the SIGYN-II project co-funded by VINNOVA, the innovation agency of Sweden.

References

[1] Hiraoka, C. "Technology Acceptance of Connected Services in the Automotive Industry," Gabler, ISBN 978-3-8349-1870-3, Wiesbaden, 2009.

[2] Campos, F.T., Mills, W.N. and Graves, M.L. "A reference architecture for remote diagnostics and prognostics applications," Proceedings of Autotestcon, pp. 842-853, ISBN 0-7803-7441-X, Huntsville, USA, October 2002.

[3] ISO/CD 13400, "Road vehicles — Diagnostic communication between test equipment and vehicles over Internet Protocol (DoIP)," 2009.

[4] ISO 14229-1, "Road vehicles - Unified diagnostic services (UDS) -- Part 1: Specification and requirements," 2006.

[5] Ellis, C., Gibbs, S. and Rein, G. “Groupware - Some Issues and Experiences,” Communications of the ACM, Vol. 34 No. 1, pp. 38-58, 1991.

[6] ISO/FDIS 26262, "Road vehicles – Functional safety," Parts 1-10, 2010.

[7] Nilsson, D. and Larson, U. "Secure Firmware Updates over the Air in Intelligent Vehicles," Proceedings of the First IEEE Vehicular Networks and Applications Workshop (Vehi-Mobi), pp. 380-384, Beijing, People's Republic of China, May 2008.

 ISBN: 978-1-61208-166-3