IP over CAN, Transparent Vehicular to Infrastructure Access

Conference Paper *in* Conference Record - IAS Annual Meeting (IEEE Industry Applications Society) · February 2008

DOI: 10.1109/ccnc08.2007.175 · Source: IEEE Xplore

3 authors:

Per Lindgren
Luleå University of Technology
78 PUBLICATIONS **798** CITATIONS
SEE PROFILE

Simon Aittamaa
Luleå University of Technology
7 PUBLICATIONS **22** CITATIONS
SEE PROFILE

Johan Eriksson
Luleå University of Technology
9 PUBLICATIONS **34** CITATIONS
SEE PROFILE

IP Over CAN, Transparent Vehicular to Infrastructure Access

Per Lindgren
EISLAB
Luleå University of Technology
97187 Luleå
Email: Per.Lindgren@ltu.se

Simon Aittamaa
EISLAB
Luleå University of Technology
97187 Luleå
Email: Simon.Aittamaa@ltu.se

Johan Eriksson
EISLAB
Luleå University of Technology
97187 Luleå
Email: Johan.Eriksson@ltu.se

Abstract— **For the future we foresee each vehicle to feature wireless communication (to the Internet and/or other vehicles) over various technologies, e.g., UMTS/GPRS, and WLAN/WiFi. In this paper we show how access to such communication resources could be granted to individual components (CAN bus connected ECUs) in the car by allowing transparent data transport using the standardized Internet Protocol (IP). Our experiments show that a complete IP Over CAN implementation, providing both UDP and TCP transport over IP, running on an Atmel AT90CAN128 is capable of transfer speeds up to 200 kbits while using less than 2 kbytes of dynamic RAM.**

I. Introduction

For the future we foresee each vehicle to feature communication over various technologies, e.g., UMTS/GRPS, WLAN, and other wireless media. The introduction of such technology to vehicular applications will (arguably) be the next big thing, with impact spanning from:

- machine-to-machine issues such as testing during development, maintenance, traffic control, active safety etc.,
- man-machine issues such to enhance the driving experience by context aware information, such as heads up road signs, traffic information (congestion, road-work etc.) all the way to
- man-man issues, such as communication and sharing information with fellow drivers in the nearby.

Such functionality could indeed be accomplished by proprietary solutions, but the history so far points clearly in the direction of standardized and ”open” protocols and interfaces. Hence, the Internet Protocol (IP) will be a strong contender for standardized communication in and between vehicles. A major advantage of TCP/IP all-the-way to the end nodes (ECUs), is that the gateway (relaying inter and intra communication) can be designed to operate ignorant from the actual communicating applications. This scenario has been acknowledged e.g. by the upcoming Media Oriented System Transport (MOST [1]) protocol, endorsed e.g. by BMW, Daimler-Chrysler, and Harman/Becker, with its native TCP/IP support. However, the dominant intra vehicle communication technology (for the last decade and for the imminent future), the CAN bus, commonly lacks support for IP transport.

II. IP all-the-way

A driving scenario urging for IP all-the-way is end-to-end applications where (at least) one of the endpoints is an in-car unit (ECU). IP-based communication has the advantage of being a mature and well-established standard, for which techniques for authorization, security/encryption and addressing/multi-homing has been developed over the last 30 years. However, the CAN bus, as being the dominant intra vehicle communication technology (for the last decade and for the imminent future), commonly lacks support for IP transport and hence is a hurdle for the development of IP-based end-to-end applications.

Providing IP all-the-way from in-car ECUs to the Internet or to other vehicles, is not solely a question of how to facilitate IP communication over the in-car network(s), but also raises the questions of vehicle-to-infrastructure, and vehicle-to-vehicle communication. These subjects are research topics in their own rights and as such out of scope of this paper.

III. Lightweight Implementation

The work in this paper is based on an existing lightweight modular stack (MODEM-IP) [2] developed at Luleå University of Technology. For the purpose of developing a lightweight IP Over CAN implementation only the IP, TCP, and UDP layers from MODEM-IP were used. The Data-Link-Layer and the Physical-Layer is replaced by IP Over Can and the physical CAN-Bus, respectively. The IP Over CAN was based on the proposed protocol in [3].

Each ECU is given a number of predefined static IDs. From the IDs a CAN ID is produced (Figure 1). As an option, to maintain a symmetric priority for reply, the source and destination fields can be swapped (SwapSD bit set ’1’). However, this requires two mailboxes, corresponding to the SwapSD = ’0’/’1’. The control bit is used to create a control channel for pre-allocating bandwidth/buffer space. It could also be used for various other tasks, such as DHCP etc.

The CAN ID assignment is slightly different than the proposed protocol, the only major difference is the SwapSD. The data layout is slightly changed to exclude the sequence counter, mainly to reduce the overhead and code complexity. Most of the functionality provided by the sequence counter is implemented using the control channel.

1-4244-1457-1/08/$25.00 © IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.

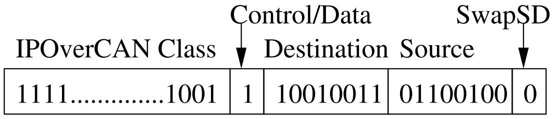

Fig. 1. IP Over CAN ID Layout.

A similar effort to standardize IP Over CAN has been proposed to IETF but is still considered a work in progress [4]. However, the protocol proposed to IETF utilizes almost all of the address bits, making it quite inflexible and could easily interfere with the real-time traffic on the network.

The current implementation uses less than 2 kbytes dynamic RAM and 48 kbytes flash (not optimized for code-size).

IV. Experimental Setup

The IP-Over-CAN protocol implementation has been validated on a set of CAN bus connected ECUs on a Formula SAE car developed at Luleå University of Technology [5]. The ECUs are based on the Atmel AVR AT90CAN128 micro controller clocked at 16MHz, featuring a total of 4 kbyte RAM, and 128 kbyte Flash Memory [6]. The ECUs run a minimalistic real-time Timber kernel featuring [7]:

- Concurrently executing state protected objects
- Time constrained synchronous and asynchronous messages
- Deadline scheduling of message execution

The MODEM-IP and its IP-Over-CAN protocol has been developed to exploit object orientation for modularity, and timed messages to realize internal communication between objects and to capture the timing constraints of the specification.

A set of experiments have been conducted on the reference implementation:

- Setting up and tearing down end-to-end TCP-IP sessions over the CAN network.
- Max throughput tests for IP over CAN has been conducted under a set of different CAN bus load conditions.

The number of connections manageable is limited only by the amount of memory assigned to the TCP layer. The current implementation requires 67 Bytes for storing the per-session TCP state, and another 40 Bytes for an ACK network buffer. As a connection is allowed to be immediately reused after termination, the stack imposes no performance limitation (other than that of the CPUs and network) to setting up and tearing down connections. The three-way-handshake of a TCP connection setup takes roughly 40 CAN-frames, about 50000 bits and 0.05seconds.

The ping test has been conducted using different buffer sizes and periodicities.

The communication is done using UDP, packets are sent in one direction only. Our tests show that the current stack implementation running on the given platform supports transfer speeds up to approximately 200 kbits payload. During the measurements an external bus-load of approximately 400 kbits was present.

Payload	Period	Total Bandwidth	
UDP and IP (bytes)	(ms)	Utilization	Overhead
512	20	45%	5.5%
256	10	43%	11%
128	20	37%	22%
128	10	37%	22%
64	20	30%	44%
64	10	30%	44%
64	5	30%	44%

TABLE I
Experimental throughput test results.

The CAN bit-rate used in this test is 1Mbit/s. From the test we can see the bandwidth utilization almost reaches 50% as the Payload size is increased (see Table I). The limiting factor is the CAN bus overhead (SOF, arbitration, control, CRC, ACK and EOF fields) of 64 bits for each packet. The UDP and IP overhead is fixed and does not scale with the payload size.

The measurements have been conducted by connecting a PC to the CAN Network, using a USB to CAN interface (Kvaser USBCANII) and a CAN debugging software from ATI (ATI CANLab). The external bus-load is also generated from the PC.

V. Conclusions

In this paper we have shown an IP all-the-way solution for CAN networks. The problem of mapping IP transport over the CAN bus with minimal protocol overhead has been discussed, and the IP-Over-CAN protocol proposal has been presented. Furthermore, we have presented a generic modular, lightweight IP stack implementation suitable for embedded platforms. We have shown how the IP-Over-CAN protocol can be seamlessly integrated in the modular stack architecture. The proposed protocol has been evaluated by experiments conducted on a reference implementation running on a set of CAN bus connected ECUs. Our experiments show that a complete IP-Over-CAN implementation, providing both UDP and TCP transport over IP, can operate efficiently using less than 2 kbytes of dynamic RAM.

References

[1] "MOST Cooperation Homepage." [Online]. Available: http://www.mostcooperation.com/

[2] Isak Rova, Simon Aittamaa, "A modular TCP/IP stack for embedded systems with a tinyTimber interface," EISLAB, Lulea University of Technology, 2007:169:ISSN:1402-1617, 2007.

[3] Michael Ditze, Reinhard Bernhardi, Guido Kamper, Peter Altenbernd, "Porting the Internet Protocol to the Controller Area Network," in *RTLIA*, 2003.

[4] Petr Cach, Petr Fiedler, "IP Over CAN," Internet Draft: Expired, 2001.

[5] J. Eriksson, P. Lindgren, and J. van Deventer, "A Distributed Engine Management System for Formula SAE," in *SAE Journal, 2007, document number 2007-01-1602*, 2007.

[6] "Atmel product web:CAN Networking." [Online]. Available: http://www.atmel.com/products/CAN/

[7] P. Lindgren, J. Nordlander, and J. Eriksson, "Robust Real-Time Applications in Timber," in *Sixth IEEE International Conference on Electro,Information Tech, EIT*, 2006.