An Improved Adaptive Data Reduction Protocol for In-Vehicle Networks

Radovan Miucic and Syed Masud Mahmud
Wayne State University

Reprinted From: In-Vehicle Software & Hardware Systems
(SP-2028)
ABSTRACT

The demand for drive-by-wire, pre-crash warning and many other new features will require high bandwidth from the future in-vehicle networks. One way to satisfy the high bandwidth requirement of future vehicles is to use a higher bandwidth bus or multiple busses. However, the use of a higher bandwidth bus will increase the cost of the network. Similarly, the use of multiple buses will increase cost as well as the complexity of wiring. Thus, neither option is a viable solution. Another option could be the development of a higher layer protocol to reduce the amount of data to be transferred. The higher layer protocol could be acceptable provided it does not increase the message latencies. The cost of implementing the protocol will be marginal because it can be done by making changes in software. Various data reduction protocols are available in the literature. We have made changes in the existing data reduction protocols to improve the performance of the protocol. Our paper will explain the improved protocol in detail and compare its performance with that of other protocols. The performance of our protocol will be shown in terms of message latency, message throughput and bus utilization.

INTRODUCTION

The amount of data can be reduced in various ways. For example, when the cruise control of a vehicle is activated the vehicle speed will vary very little. Thus, the electronic module that is responsible for sending the vehicle speed, can send only one bit instead of the actual speed to inform the recipients that the speed did not change. Similarly, when the change in speed is very small, the electronic module can send only the amount of change using a few bits rather than the actual speed using many bits. The protocol can also be made adaptive by allowing it to select different message formats for different conditions of the vehicle. For example, when the vehicle is in idle state the message format can be different from the format when it is moving at a speed of 60 mph or higher.

There are many data reduction techniques for communication protocols. We will single out two, Misbahuddin’s et al [1] suggested general data reduction technique based on repeatability of data bytes in a message. Ramteke et al [2] in his paper proposes data reduction technique based on signal stability. Both algorithms are designed to work with the Controller Area Network (CAN) [4] but can be applied to any serial protocol.

A CAN message has up to 8 bytes [4]. In-vehicle network architecture always has periodic messages. Periodic messages convey information from transmitter to receiver(s). Data bytes often time do not change from one to next message transmission. Misbahuddin exploited byte repeatability to come up with general form of data reduction technique. If all bytes changed, form one to next message transmission, original message is sent. If one or more bytes remained unchanged, from one to the next message transmission, the reduced message is sent. Reserve bit (R) in control field of CAN message indicates data reduction is taking place. Misbahuddin names reserve bit a data-compression bit (DCB). In other words, a message is sent with reduced content when DCB is set. Consider reduction of CAN message with 8-byte data field. First byte, compression code, of a new message contains compression information. Position of each bit in the compression code corresponds to data byte position of the message. A bit with a value of “1” indicates a repeated byte in the message and a bit with a value of “0” indicates the byte that is not repeated. The non-repeated bytes are placed after compression code in the data field of the actual data frame sent over the multiplexing bus. [1]

Instead of concentrating on byte value preservation as Misbahuddin did, Ramteke looks at the value of signals in the message. Signals may be represented with any number of bits. If signal does not change in value from previous to current message transmission, signal value is not sent. If signal value is slightly changed then only difference is sent over the multiplexing bus. This paper proposes improvements over Adaptive Data Reduction (ADR) suggested by Ramteke. [2]

Data reduction (DR) is not only useful in production vehicles, but in development environment as well. For example, DR can be applied in data acquisition tools that use CAN Calibration Protocol (CCP) [7] to reduce the number of messages and bus utilization. The advantages of resulting tools are increased data rate.
and ability to monitor more data. Examples of CCP instrumentation tools are Vector’s CANape [8] and ATI’s Vision [9] software.

Decreased message traffic decreases electromagnetic noise coming through on the radio. Basically, less message traffic improves EMC; more specifically radiated emission, radiated susceptibility, conducted emission and conducted susceptibility. Some vehicle manufacturers, such as Lexus, pay close attention to the bus use and messaging protocol. Unnecessary message communication is eliminated. Only messages conveying needed information are sent. [10] Therefore, data reduction concept helps EMC performance.

**IMPROVED DATA-REDUCTION ALGORITHM**

Improved Adaptive Data-Reduction (IADR) algorithm is implemented on top of the CAN protocol. However, it could be applied over any message based serial data protocols. IADR has impact on periodic messages that contain numeric signal values. Signal value is usually an output of a continuous function. Let us consider vehicle speed. Let us say that the engine Electronic Control Unit (ECU) sends message \( M_1 \) periodically with period \( T \). Message \( M_1 \) contains instantaneous value of vehicle speed. Vehicle speed at time \( t_{n-1} \) is \( s_{n-1} \) at the next time instance \( t_n = t_{n-1} + T \), vehicle speed is \( s_n = s_{n-1} + \Delta s \). Where \( \Delta s \) is the difference in values of the vehicle speed at times \( t_n \) and \( t_{n-1} \).

![Figure 1 – Signal increased value from \( t_{n-1} \) to \( t_n \) for \( \Delta s \).](image)

The difference, \( \Delta s \), is a very small value if message transmission period \( T \) is small and the value of vehicle speed is not drastically changing. For example, let us consider changing vehicle speed during acceleration of very fast vehicle. In real world, acceleration of a very fast vehicle may go up to 6.4 m/sec\(^2\) [3]. Let us say message transmission period is \( T = 5 \text{ ms} \). Maximum value of \( \Delta s \) will be in the order of 0.1152 km/h. The difference, \( \Delta s \), requires fewer bits in the message then bits representing entire vehicle speed signal.

Overview of the IADR is shown in Figure 2. Each ECU on the multiplexing bus has physical and application layer buffers. When transmitter aims to send a message, it puts the message in the application transmit buffer. The IADR component compares the message with the previously sent message and decides if and how data reduction is to be used. Compressed or not, the message is placed to physical transmit buffer. The actual message on the bus comes from the physical transmit buffer. Similarly, receiver gets the message in the physical receive buffer. The IADR component interprets the message and prepares the message that is reported to the upper application layer of receiver.

IADR is similar to ADR proposed by Ramteke[2] but with several changes. Detail description of IADR and differences between IADR an ADR follows.

1. Instead of using two message IDs, as in Ramteke’s ADR, IADR uses single ID.
2. First bit of data field in CAN message, DRC (Data Reduction Code), determines if data reduction is used or not. Ramteke uses different IDs to indicate DR is taking place. If data reduction is not used, entire signals are sent. This is the case with the very first message. Also, if for all signals the calculated difference exceeds delta size then message containing all signals in their entirety is sent.

   - **DRC=0** Message does not use data reduction. Message contains entire signals. It is no different than normal message, except that the data fields are shifted to the right by one bit.

   - **DRC=1** Message uses some form of data reduction. Setting DRC bit, turns on SiRC(Signal i Reduction Code) bits.

3. There are three levels of signal transmission.
   a. **Entire signal**
   b. **Delta of signal**
   c. **Implied signal** (signal not sent because it did not change from previous time instance.) Upon reception of the implied signal, receiver assumes previous value of the signal.

SiRC determines if the entire signal is sent or the reduction code is sent.

   - **SiRC=0** Send entire signal i.
   - **SiRC=1** Send delta or signal i is implied.
   
   Setting SiRC bit, turns on SiRT(Signal i Reduction Transmitted) bit.

SiRT determines if the delta of the signal is sent or the signal is implied.

   - **SiRT=0** Send delta of signal i.
   - **SiRT=1** Signal i is implied.

A summary of the IADR rules is presented in Figure 3.
IADR compare current and previous signals and prepare physical T_BUF

Is it time for periodic message?
Yes
No

Continue application execution loop.

Send Entire Signals

No Data Reduction Present

DRC:

Some Data Reduction Present

Send Entire Signals

Send Delta of Signal

No Change Do Not Send Anything

Figure 2 - Graphical representation of the IADR required changes for transmitter and receiver ECUs. The physical message is passed from the physical transmit buffer to the physical receive buffer.

Send Entire Signals

No Data Reduction Present

Figure 3 - Compression code implementation details.
Several message examples are shown below.

![Message Diagram](image)

**DRC:**
1 - data compression present
0 - no data compression present

**Figure 4** – Example of non-compressed message. Message contains all entire signals.

The message in Figure 4 is the first periodic message, or a message in which all signals changed beyond the size of delta.

**Figure 5** – Example of a message containing highest data compression. Signals did not change from previous message transmission.

For the message in Figure 5, all SiRTs have value of “1”. All signals have not changed from previous message transmission and no signal data is sent. Signal values are implied at the receiver’s end.

The message in Figure 6 has DRC code set. The message is compressed. All SiRCs have value of “1”. All signals are presented in the compressed form. All SiRTs have value of “0”. All compressed fields are delta values of signals.

The message in Figure 7 and its data is interpreted in the following manner:
1. Signal 1 did not change from the previous message transmission.
2. Signal 2 changed within the range of delta.
3. Signal 3 did not change from the previous message transmission.
4. Signal 4 changed inside the range of delta.
5. Signal 5 changed within the range of delta.

**Figure 6** – Example of a message containing data compression. All signal changes are represented with deltas.

**Figure 7** – Another example of a message containing data compression.
SIMULATION RESULTS

We have created a time based software simulation of a CAN network. Every CAN message is composed of overhead field and data field. The data field carries signals. Signals vary with time. An input file contains signal behavior with respect to time. Inputs are presented in the Figure 10. Our program simulates data reduction algorithm proposed by Ramteke (ADR) as well as our improved algorithm (IADR). The performance of our algorithm is measured in terms of message latency and bus utilization.

Message latency is defined as the time interval from the instant a message is generated by transmitter to the instant when the message’s last bit is received by the receiver. [1]

Bus utilization is the ratio of utilized bus time to the total time. [1]

Our simulation calculates message latency and bus utilization every 100 ms. The simulated bus baud rate is 500 kbits/second.

We have considered a periodic message with 5 ms period. The message contains the following fields:

1. Vehicle Speed
   units: km/hour
   range: -80 km/h to 300 km/h
   resolution: 0.1 km/h
   bits req: 1 (sign) + 12 (magnitude) = 13 bits

2. Engine coolant temperature
   units: Celsius
   range: -40 C to 150 C
   resolution: 0.1 C
   bits req: 1 (sign) + 11 (magnitude) = 12 bits

3. Engine RPM
   units: rotation per minute
   range: 0 RPM to 20000 RPM
   resolution: 0.1 RPM
   bits req: 18 (magnitude) = 18 bits

4. Engine Fan RPM
   units: rotation per minute
   range: 0 RPM to 1600 RPM
   resolution: 10 RPM
   bits req: 8 (magnitude) = 8 bits

5. Instantaneous Fuel Flow
   units: gram per second
   range: 0 g/sec to 100 g/sec
   resolution: 0.2 g/sec
   bits req: 9 (magnitude) = 9 bits

Data field of a normal message, message without any data reduction changes (containing entire signals), is presented in Figure 8.

Each signal delta is 4 bits wide: 1 bit for sign plus 3 bits for magnitude.

In Figures 8 and 9, a number in a circle represents entire signal and a number in a triangle represents the delta of a signal.

Consider a scenario where vehicle is accelerated from 0 km/h to some high speed. The vehicle is driven at high speed for some time. The driver applies the brakes and the speed of the vehicle decelerates to some low speed. Relative values of signals vs. time are presented in the Figure 10.

Simulation results in Figure 11 show significant bus utilization savings using IADR algorithm.

The results in Figure 11 are taken form running simulation in which messages are sent even when their signals contain no change. Bus utilization for IADR is considerably less then that of ADR.

In addition, if messages when signals did not change are not sent then, again, simulation results still show improved bus utilization savings, Figure 12. This “no change - no message transmission” feature of the DR algorithm was proposed by Ramteke in [2]. Not sending messages when signals are unchanged is not recommended because if transmitter is disconnected then receiver(s) would, for some time, imply old values of the signals.
Figure 10 – Signal variation during acceleration, steady speed, during braking and again steady speed.

Figure 11 – Bus utilization comparison.
Figure 12 – Bus load comparison for simulation run where if no data changed message is not sent.

Figure 13 – Bus load comparison for enhanced IADR simulation run.
POSSIBLE ENHANCEMENT TO IADR

To overcome, preemptively prevent, potential issues of not having synchronized signals in transmitting and receiving nodes signals can be forced to be send in entirety on rotating basis, “Cyclic Refresh”. For example, let us have a message of five signals. First message sent will contain signals in their entirety. Second message sent will forcibly send first signal in its entirety, even though first signal did not change or slightly changed sufficient for reduction. Third message will forcibly send second signal in its entirety. Next message will forcibly send third signal in its entirety, and so on. Sending signal in entirety will rotate.

ADVANTAGES AND DISADVANTAGES OF IADR

Misbahuddin’s technique is difficult to implement because many CAN peripherals on the market are forcing the reserve bit to a specific value. For example CAN peripherals of ST9 and PIC will force reserve bit to a dominant value of “0” [5 & 6]. Implementing Misbahuddin’s technique would require hardware changes to existing microcontrollers and transceivers.

Ramteke’s technique uses two message identifiers to differentiate compressed from normal message. The delta is sent in the compressed message even if a corresponding signal did not change from previous message transmission.

The IADR technique has one bit overhead in the message containing all signals, while there is no overhead in other mentioned techniques. For 5-signal message maximum overhead bits in IADR is 11 and in ADR is 5.

From results it can be seen that IADR has better bus utilization than ADR.

Figure 14 – Different signal behaviors from $t_{n-1}$ to $t_n$ time instances. Value of signal $s_1$ slightly increased. Value of signal $s_3$ slightly decreased. Value of signal $s_2$ drastically increased. Value of signal $s_4$ drastically decreased.
FUTURE CONSIDERATION

This paper presents data reduction based on the difference in signals. In other words, one level of data reduction is a result of subtraction. Other kinds of relationship, between current and previous values can be exploited in order to save space in messages. For example, if signal difference requires more bits than the size of delta, then the tangent function can be used to explain large changes in signal values. The transmitter, instead of sending delta, will send angle $\beta$. Receiver will calculate current value of signal $s(t_n)$ by adding $\tan(\cdot)$ to previous value of signal $s(t_{n-1})$. The current signal would be calculated based on Equation 1

$$s(t_n)=s(t_{n-1})+\tan(\beta)$$  (1)

Also, a look up table can be used instead of tangent function where $\beta$ would be index. Receiver will calculate current value of signal $s(t_n)$ by adding $\text{look\_up\_table\_value}(\beta)$ to previous value of signal $s(t_{n-1})$, Equation 2.

$$s(t_n)=s(t_{n-1})+\text{look\_up\_table\_value}(\beta)$$  (2)

Range 1 covers such cases where the signal difference can be explained using the delta of the signal. Range 2 covers cases where the signal difference cannot be explained using the delta of signal. From figure 14, signals $s_1$ and $s_2$ are candidates for range 1 and signals $s_3$ and $s_4$ are candidates for range 2. However, using the tangent function or a lookup table does not guarantee exactly repeated signal value at the receiver end.

CONCLUSION

Data reduction helps lower in-vehicle communication cost, EMC performance, and increase in scalability of the existing protocol. We have created time based software simulation of CAN bus network. In the simulation, we have implemented two data reduction techniques, IADR and ADR. We showed empirical results confirming advantages of IADR over ADR. Results show significant bus utilization savings for vehicle “steady state”, the state vehicle signals are not changing rapidly. Simulated inputs reflect real world data. However, our next task is to run simulation with messages taken from the actual vehicles. In addition, we plan to study “dynamic state” the state vehicle signals are changing rapidly.

REFERENCES


CONTACT

Radovan Miucic received his B.S. degree in electrical engineering in 2001 and his M.S. degree in computer engineering in 2002 both from Wayne State University, Detroit. Currently he is working on his PhD degree, and as a software engineer in automotive industry.

Phone: (313) 577-3855,
Email: radovanmiucic@hotmail.com

Syed Masud Mahmud, Ph.D.
Associate Professor
Department of Electrical and Computer Engineering
Wayne State University
Detroit MI 48202
Phone: (313) 577-3855
Fax: (313) 577-1101
Email: smahmud@eng.wayne.edu

DEFINITIONS, ACRONYMS, ABBREVIATIONS

ECU – Electronic control unit.
ADR – Adaptive data reduction.
IADR – Improved adaptive data reduction.
CAN – Control Area Network.
DCB – Data compression bit.
CCP – CAN calibration protocol.
DRC – Data reduction code.
SiRC – Signal i reduction code.
SiRT – Signal i reduction transmitted.
EMC – Electromagnetic Compatibility
Normal message – Message without data reduction technique.