Deterministic and Flexible Communication for Real-Time Embedded Systems

Supervisor
prof. Riccardo Sisto
dott. Ivan Cibrario Bertolotti

Candidate
Tingting Hu

Department of Control and Computer Engineering (DAUIN)
Politecnico di Torino

2015/02/27
Outline

- Introduction
  - Controller Area Network (CAN)

- Deterministic Communication
  - 8B9B, VHCC, ZSC

- Flexible Communication
  - IP over CAN
  - Modbus CAN

- Conclusion
As embedded systems evolve from centralized to distributed architecture, communication becomes more and more important.

- Controller Area Network (CAN) is a real-time communication network
- It is the de facto standard in automotive and gained popularity in networked embedded control systems recently.
- **Determinism** is an important feature of real-time embedded systems:
  - Delay: time taken to complete a certain task
  - Jitter: variability in delay

Jitter *impairs* determinism, and *worsens* the real-time performance.
Sources of jitter in CAN-based distributed system

Variable task execution time
OS scheduling and timing
CAN Node
Operating System
T
Device Driver
CAN Controller
T
CAN Node
CAN Node
Header
Data
CRC
software
Hardware
CAN bus

Task-level jitter
Comm.-level jitter

Bit stuffing mechanism

At the physical layer, CAN relies on bit stuffing (BS) for receiver synchronization.

- # of stuff bits depends not only on the frame length, but also on its content ⇒ Jitter in communication.
- For real-time systems with tight timing constraints, ~20% of system-wide jitter.
- BS interferes with the CRC-based error detection mechanism in CAN and jeopardizes data integrity severely.
State of the art

- **Header**: fixed & known in advance $\Rightarrow$ no communication jitter
- **Data field**: variable from message to message
- **CRC**: depends on both the header and the data field; calculated by hardware, at run time

Existing approaches just prevent BS jitter in the **data field**, by either **scrambling** or **encoding** the payload in a way that less or no stuff bits will be added by the CAN controller during transmission.

**NO** approaches available for the CRC.
BS prevention mechanisms

ZSD and ZSC prevent BS from the **data field** and the **CRC**, respectively

- **Zero Stuff-bits Data (ZSD)**
  - Fixed-length payload encoding: \(8B9B\)
  - Variable-length payload encoding: VHCC
- **Zero Stuff-bits CRC (ZSC)**
- ZSD and ZSC are **compatible** with each other

---


\(^3\) G. Cena, I. Cibrario Bertolotti, T. Hu, and A. Valenzano. “Un codec a basso jitter per reti CAN”. *Automazione e strumentazione.*
Every byte of the original payload is translated into a distinct 9-bit codeword.

- Codebook property 1)
  - ≤ 4 consecutive bits at the same value within the 9-bit codeword
  - ≤ 2 consecutive bits at the same value at the beginning & end

- Padding is needed at the end of the data field

- Achieve better encoding efficiency than existing approaches.
Instead of padding, pack sub-byte application information

- Codebook property 2)
  - Nested codebooks: 8B9B codebook can also be used for any N bit to N+1 bit encoding, $N \in [1, 8]$.
  - Nested codebooks still preserve property 1)

---

Zero Stuff-bits CRC (ZSC)

Instead of padding, use it to prevent BS in the CRC.

- Exploiting only **3 bits** at the end of the data field, it is always possible to **tune** the CRC calculation to a value that is BS free.
- **Independent** from the payload content and the encoding scheme.
- ZSD + ZSC leads to **deterministic communication**\(^5\):

Main achievements

- Theoretical/experimental results show that 8B9B and VHCC achieve better **computational** and **communication** efficiency\(^6\).
- Highly optimized and portable codec was developed for dissimilar embedded platforms.
- Residual error probability **decreases** by about two orders of magnitude\(^7\).
- ZSC: an Italian **patent**\(^8\) application was submitted and a European extension is in progress.

---


Flexible communication

Application-level protocols for CAN are mainly automotive oriented.

- SAE J1939 and ISO 11783 for communication/diagnostics among in-vehicle components.
- CANopen and DeviceNet for industrial automation
- ARINC 825 for local subsystem communication in civil aviation

Extend the flexibility of CAN by broadening the set of high-level protocols supported on it

- General-purpose protocol (IP)
- Special purpose protocol (Modbus)
General-purpose protocol support

Internet Protocol (IP): the most widespread protocol, enormous software available

Goal:
- Integrate CAN (at the field level) into Intranet
- Support non real-time activities like remote configuration, firmware update...

Existing approaches:
- Protocol translation

IP over CAN: permits IP datagrams to be transmitted on CAN
When integrating different subsystems, coexistence among them is always the main concern

- Interference to real-time performance (due to IP traffic):
  - IP messages can be modeled as real-time messages with the lowest priority
  - Worst-case jitter is bounded.

- Non real-time performance is comparable to what can be achieved on a pure CAN link (in absence of RT traffic):

<table>
<thead>
<tr>
<th>$d$ (B)</th>
<th>1</th>
<th>4</th>
<th>16</th>
<th>64</th>
<th>256</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{C\rightarrow C}$ (kB/s)</td>
<td>0.78</td>
<td>2.91</td>
<td>8.90</td>
<td>18.02</td>
<td>24.76</td>
<td>26.08</td>
</tr>
<tr>
<td>$r_{C\rightarrow E}$ (kB/s)</td>
<td>0.65</td>
<td>2.49</td>
<td>7.97</td>
<td>17.80</td>
<td>24.07</td>
<td>26.43</td>
</tr>
<tr>
<td>$r_{E\rightarrow C}$ (kB/s)</td>
<td>25.18</td>
<td>26.12</td>
<td>26.36</td>
<td>26.50</td>
<td>26.46</td>
<td>26.44</td>
</tr>
</tbody>
</table>

Table: Mean data transfer rate $r$ for 10 MB data vs data chunk size $d$. 

---

---

Special purpose protocol support

Modbus

An application-level real-time communication protocol, which is commonly used in building automation.

- Existing support for Modbus:
  - RS485: obsolete, extremely slow (19200 bps)
  - Ethernet: popular, extra cabling and intermediate devices

- CAN is a good compromise in terms of both link speed (1 Mb/s) and cost (bus architecture)

- Modbus CAN: enables Modbus traffic to be transmitted on the CAN bus
Modbus CAN\textsuperscript{10} outperforms Modbus TCP on a 100 Mb/s Ethernet, with a 1 Mb/s CAN bus.

Break-even point (for mean RTT) at \( n \approx 80 \ldots 100 \) registers when using a 500 kb/s CAN bus.

A compromise between link speed and software processing overhead

Conclusion

- Research work was carried out focusing on the determinism and flexibility of CAN communication.
- BS jitter is completely prevented all over the frame by ZSD + ZSC.
- The application scenario of CAN was largely broadened by the design and implementation of IP over CAN and Modbus CAN.
- Modbus CAN is adopted by industry for local subsystem communication and it paves the way of CAN in building automation.
2014


List of Publications (2/5)

*Effect of Jitter-Reducing Encoders on CAN Error Detection Mechanisms*, 

*Design, Verification, and Performance of a Modbus–CAN Adaptation Layer*, 

*Un codec a basso jitter per reti CAN*, 

*Limitazione del bit stuffing in una trama di comunicazione di un segnale elettronico*, 
2013

Fixed-Length Payload Encoding for Low-Jitter Controller Area Network Communication,  

[10] I. Cibrario Bertolotti, L. Durante, T. Hu, and A. Valenzano,  
A Model for the Analysis of Security Policies in Industrial Networks,  

Software-Based Assessment of the Synchronization and Error Handling Behavior of a Real CAN Controller,  

On a Family of Run Length Limited, Block Decodable Codes to Prevent Payload-Induced Jitter in Controller Area Networks,  
List of Publications (4/5)

2012

A Unified Class Model for Checking Security Policies in ICT Infrastructures,
Proc. 1st IEEE AESS European Conference on Satellite Telecommunications (ESTEL),
pp. 1–6, Oct. 2012.

Performance evaluation and improvement of the CPU–CAN controller interface for low-jitter communication,
Proc. 17th IEEE Conference on Emerging Technologies and Factory Automation (ETFA),
pp. 1–8, Sep. 2012.

Performance comparison of mechanisms to reduce bit stuffing jitters in controller area networks,
Proc. 17th IEEE Conference on Emerging Technologies and Factory Automation (ETFA),
pp. 1–8, Sep. 2012.

[16] G. Cena, I. Cibrario Bertolotti, and T. Hu,
Formal Verification of a Distributed Master Election Protocol,
Proc. 9th IEEE International Workshop on Factory Communication Systems (WFCS),
2011


THANK YOU FOR YOUR ATTENTION