Comparison of In-Vehicle Communication Protocols for Critical Applications

Edward Robert Gundlach and Syed Masud Mahmud

Electrical and Computer Engineering Department, Wayne State University, Detroit, MI 48202

ABSTRACT

The communication network is a key enabling technology for many of the latest advances in vehicle design. As drive-by-wire, vehicle control systems (e.g., vehicle stability control), hit avoidance, pre-crash warnings, and other design advances become more pervasive, the selection of the most appropriate in-vehicle communication protocol(s) for these critical applications becomes more important. This paper provides a comparison of the leading communication protocols for these high-performance applications.

The protocols analyzed for this comparison are Time Triggered Controller Area Network (TTCAN) and FlexRay. This paper provides a summary of the features and attributes of these competing communication protocols, and it highlights the advantages and disadvantages where there are differences. In addition to the many quantitative attributes (e.g., bit rate, error detection, fault tolerant features, message payload, etc.) of these protocols, some qualitative aspects of the protocols are also provided. The goal of this paper is to provide a valuable resource when comparing and selecting communication protocols for these critical vehicle applications.

INTRODUCTION

As automotive OEMs continue to expand the level of functionality in their vehicles, the performance requirements placed on the communication network continue to increase in order to support these additional applications. In particular, drive-by-wire applications such as throttle-by-wire, brake-by-wire, and steer-by-wire, require significant enhancements to the vehicle communication network. Prior to these critical applications, the vehicle communication protocols (e.g., CAN) have typically supported event-driven messaging. However, these safety critical applications require a more deterministic message scheduling with known message latencies. In addition, the fault tolerant aspects of the communication network are very important in these safety critical applications.

This paper compares two leading protocols that address the requirements for deterministic message scheduling and fault tolerance. The first protocol is Time Triggered CAN (TTCAN), which extends the CAN protocol by specifying another layer above the data link and physical layers of the CAN standard. Due to its popularity of CAN and the widespread familiarity with CAN hardware and tools, TTCAN with its deterministic messaging capability is considered a likely selection for these critical applications. The second protocol is FlexRay, which was specifically created for these critical applications by an industry consortium that includes BMW, Bosch, DaimlerChrysler, General Motors, Motorola, and Philips as core members. With a strong list of core and associate members in the consortium, FlexRay will be considered for many of these critical applications.

Over the years, many technical papers have documented the benefits of multiplexing data onto vehicle communication networks. These papers have also captured the evolution of the vehicle communication network into categories such as SAE Class A, B, and C, and into more recently defined groupings such as X-by-Wire (2). Some papers have identified the key requirements needed for these categories of communication networks, and they have provided high-level comparisons of protocols that fall into these categories. Other papers have provided detailed analysis of the communication protocols that are the subject of this paper. Since 2001, separate SAE papers have described both the FlexRay (7) and the TTCAN (11) protocols. Subsequent papers have provided additional information on the application of each of these protocols in automotive systems.

Unlike previous high-level protocol comparisons, this paper provides a low-level comparison of how two leading protocols address the key requirements for safety critical applications such as X-by-Wire. This comparison of the TTCAN and FlexRay protocols discusses their differences, and it describes the advantages / disadvantages of each protocol in various areas. For some of the disadvantages, this paper will offer methods that may mitigate the deficiency when using the protocol. The value of this paper is in this detailed analysis of protocol differences. Similar to
other papers, a summary table that compares the protocols is also included. Because of this detailed analysis and summary, this paper should provide a valuable resource when comparing and selecting communication protocols for these safety critical applications.

This paper begins with a Background section, which provides an overview of the TTCAN and FlexRay protocols. This information shows how these protocols satisfy the requirement for deterministic message scheduling. This is followed by the main body of the paper, which is a Comparison of Protocols. This section provides a detailed analysis of the protocol differences, advantages, and disadvantages for meeting the application requirements. A summary table of the comparison is also included in this section. The paper ends with the Conclusion and References sections.

BACKGROUND

The communication procedures for both the FlexRay and the TTCAN protocols are described below. Each protocol can support the deterministic message-scheduling requirement of safety critical applications, while also supporting event-driven messages.

FLEXRAY PROTOCOL OVERVIEW

For the FlexRay protocol, media access is provided in recurring communication cycles that are triggered from a synchronized time base. Each communication cycle can contain two main phases, a static segment and an optional dynamic segment. The static segment uses a TDMA scheme to allocate static slots of uniform length to the communication network nodes. Each node has the opportunity to use its static slot to transmit a message during each recurring communication cycle. In the dynamic segment, minislots are used to establish the dynamic transmission priorities of the network nodes. The node with the highest priority has the first opportunity to transmit a dynamic message. The duration of a dynamic message can encompass the balance of the dynamic segment. One minislot after the highest priority node has completed, the next node has the opportunity to transmit a dynamic message. Depending on the utilization of the dynamic segment by the higher priority nodes, some lower priority nodes may not have an opportunity to transmit during the dynamic segment of a communication cycle. In the FlexRay protocol, arbitration is not required, because only one node has the authority to transmit at any time.

TTCAN PROTOCOL OVERVIEW

For the TTCAN protocol, media access is provided in recurring basic cycles that are triggered from a time master’s reference message. Each basic cycle allocates time to transmit messages in exclusive windows and/or arbitrating windows. An exclusive window is assigned to a specific communication network node for transmitting a periodic or an event-driven message. In an arbitrating window, the bitwise arbitration of the CAN protocol allows the highest priority node to successfully transmit a message. (The automatic retransmission feature of the CAN protocol is not allowed in exclusive or arbitrating windows.) A set of basic cycles define the system matrix of messages for the communication network. The system matrix allows the developer to specify the frequency of message transmission. In the TTCAN protocol, each node must know when it has an exclusive window to transmit a message, or when an arbitrating window is available to attempt transmitting an event-driven message.

COMPARISON OF PROTOCOLS

This main section of the paper examines the differences between the TTCAN and the FlexRay protocols, and it identifies potential advantages and disadvantages for each. The comparison begins with the communication procedures for exchanging messages described in the background section. This is followed by examining some features that improve
fault tolerance. This section concludes with a listing of some features for these two communication protocols, which are important for safety critical applications.

TIME-TRIGGERED & EVENT-DRIVEN MESSAGES

In the TTCAN communication protocol, the scheduling of periodic, time-triggered messages and the allocation of time for event-driven messages is very flexible. The exclusive window for a periodic message is scheduled at one or more positions in a basic cycle and for one or more basic cycles within the system matrix. A node may transmit an event-driven message during an exclusive window for that node or during an arbitrating window (if the node has the highest priority of those attempting to transmit). The TTCAN protocol allows arbitrating windows to be interspersed with the exclusive windows of a basic cycle, which can lower the latency of high-priority, event-driven messages. This flexibility in defining the message structure of the system matrix is an advantage for designers using the TTCAN protocol.

The message structure in the FlexRay communication protocol is very rigidly structured. For a given network, the structure of each communication cycle is identical. A cycle begins with the static segment, which contains static slots for each periodic message. A dynamic segment follows, which allocates time for the transmission of any event-based messages. Within the static segment, the time allocated for each static slot is identical.

This rigid message structure specified by the FlexRay protocol can result in inefficient utilization of the available communication bandwidth. Because the protocol requires static slots to have identical length, all static slots must accommodate the longest static message. For shorter static messages, a portion of the available payload capacity in their assigned static slots will not be used. Under-utilization would also occur when a node uses a static slot for a low-frequency periodic message. For example, if the communication cycle had a duration of 10 msec, and a node used its static slot to send a periodic message every 100 msec, then its assigned static slot would only be used once every 10 communication cycles. A more efficient alternative for transmitting this type of low-frequency periodic message would be to use the dynamic segment with a high priority minislot assignment every tenth communication cycle. Another disadvantage of the FlexRay message structure is the grouping of all static messages then all dynamic messages for each communication cycle. Unlike the TTCAN protocol, the FlexRay structure restricts the system designer from mixing dynamic messages with static (i.e., periodic) messages.

BIT ENCODING & ERROR DETECTION

Both the TTCAN and the FlexRay protocols use the Non-Return to Zero (NRZ) method for encoding the communication bit stream. The protocols differ in their solutions to the NRZ problem of clock drift when no edges are in the bit stream for synchronization. The TTCAN protocol uses bit-stuffing to force a bit transition after every five identical bit values. The FlexRay protocol uses the Byte Start Sequence (BSS) and the Frame End Sequence (FES) for bit synchronization. The BSS precedes every byte with a 1 bit followed by a 0 bit, and the FES follows the last byte with a 0 bit followed by a 1 bit. Both methods will prevent clock drift problems for communication networks operating within their specified timing tolerances. Each method adds approximately two bits of overhead for every byte of transmitted information. Both protocols will indicate timing errors when their bit synchronization techniques are violated.

Both protocols use Cyclic Redundancy Code (CRC) bits to detect corrupted information in the data stream. For TTCAN, a 15-bit CRC is added to the message to ensure the validity of the header and data information in the message. Each receiving node will calculate a CRC value based on its received data stream, and this will allow the detection of a 5-bit error. For FlexRay, an 11-bit CRC protects the message header data, and a 24-bit CRC protects the message payload data. The header CRC allows detection of a 6-bit error. The payload CRC allows detection of a 6-bit error for up to 248 data bytes or detection of a 4-bit error for payloads of 250, 252, or 254 data bytes.

Because TTCAN uses an arbitrating bus, any node that detects a timing or CRC error can immediately signal the error by transmitting an error frame on the bus. At the end of a TTCAN message, any node that has properly received the message can send an in-frame acknowledge signal. The sending node in TTCAN also monitors the bus to detect transmission errors and to receive the error or acknowledge signal from other nodes. While this type of immediate feedback can be beneficial to the system designer, there are also some drawbacks when using an arbitrating bus. The TTCAN arbitrating bus cannot use transformer coupling for improved noise immunity. The delays to support collision detection, also limit the bus speed to 1 Mbps.

The FlexRay protocol does not use an arbitrating bus. Simultaneous transmissions from multiple nodes should only occur during the network startup process. Following startup and clock synchronization, only one node should attempt to transmit on the network at any time. The lack of an arbitrating bus for FlexRay allows bus speeds up to 10 Mbps. However, an in-frame acknowledge signal may have been useful for some messages and applications. For these critical periodic messages, the system designer may choose to
allocate static slots for receiving nodes to acknowledge receipt of the original message.

STARTUP & CLOCK SYNCHRONIZATION

In the TTCAN protocol, the node assigned as the time master is responsible for transmitting the reference message, which is critical for network startup and generation of the globally synchronized clock. For a more fault tolerant system design, the protocol allows multiple nodes to be designated as potential time masters. After a reset, each potential time master checks for activity on the network and for the presence of a reference message. If a reference message is not being transmitted, then the potential time master attempts to transmit a reference message. If multiple time masters simultaneously start a reference message, the highest priority node will win arbitration. The reference message contains that time master's identifier and its local time as the global time reference. All nodes synchronize to the reference message and to the global time. At the start of any subsequent basic cycle, a higher priority time master can win arbitration and take responsibility for transmitting the reference message. If the current time master fails to send a reference message, then the other potential time masters will restart the stream of reference messages after detecting the problem.

In the TTCAN network, each node is responsible for generating a network time unit (NTU) counter. Each node divides its local system clock by a local time unit ratio (TUR) to create the node-independent NTU counter, which represents the local time. For each basic cycle, each node performs drift correction to synchronize its NTU counter properly with that of the time master. Based on the last two reference messages, the duration of the previous basic cycle in global time units is compared to the duration in local time units. The local TUR is adjusted by these values, so that the local time more closely matches the global time. The advantages of TTCAN approach are a fast startup and redundant time masters. A disadvantage of the TTCAN approach is that all nodes attempt to match the global time of the highest priority time master. Thus, a single point failure resulting in excessive variation of the global time can cause failures in the network communications.

In contrast to TTCAN, the FlexRay approach avoids this type of a single point failure for clock generation. As shown in Figure 4, the FlexRay protocol needs at least two coldstart nodes to startup the network. For a more fault tolerant system design, the protocol recommends that three nodes be designated as coldstart nodes. After a reset, each coldstart node listens for activity on the network. If FlexRay messages are not being transmitted, then the coldstart node sends a Collision Avoidance Symbol (CAS) and begins the coldstart collision resolution, which includes sending its startup frame with each cycle. Because other nodes may also attempt a coldstart, any collisions are resolved in four communication cycles. After these first four cycles, other coldstart nodes should have integrated to the leading node's schedule. If integration was successful, then it should begin sending its startup frame in cycle 4. For non-coldstart nodes, they require at least two coldstart nodes to send startup frames during four consecutive cycles for their integration to be successful. Therefore, normal communication cannot begin until at least cycle 8.

Figure 3 – TTCAN global & local time (11)

Figure 4 – FlexRay coldstart node collision resolution (11)
Figure 4 – FlexRay startup state transitions (12)
Similar to the NTU counter in TTCAN, each node in the FlexRay network generates a macrotick as the common unit of time. The clock synchronization process of the FlexRay protocol corrects the calculation of the macrotick rate as well as the offset from local time to global time. The rate correction takes place before the start of even communication cycles, and the update of the offset occurs before the end of odd communication cycles. Unlike the single timing mark of the reference message in each TTCAN basic cycle, the FlexRay protocol evaluates the timing of all messages in the static segment for the purpose of clock synchronization. These sets of corrective timing values are evaluated by a fault-tolerant midpoint algorithm to calculate the appropriate correction values. The values are sorted, and the k largest & the k smallest are discarded (see Figure 5 for values of k). Finally, the largest and smallest remaining values are averaged to derive the correction term (see Figure 6). The advantage of this algorithm is that the global time is not dependent on the clock of a single node, because the global time is synchronized based on the average clock performance of the entire network. A disadvantage of the FlexRay approach is the time required for startup and synchronization for full network communications.

<table>
<thead>
<tr>
<th>Number of values</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 2</td>
<td>0</td>
</tr>
<tr>
<td>3 – 7</td>
<td>1</td>
</tr>
<tr>
<td>&gt; 7</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 5 – Fault-tolerant algorithm parameter (12)

\[
\frac{15 + 13 + 11 + \ldots + 6 + 5}{2} = 8
\]

Figure 6 – FlexRay clock correction algorithm (12)

**NETWORK TOPOLOGY & FAULT HANDLING**

The FlexRay protocol and physical layer specifications support various configurations such as bus, active star, cascaded star, and hybrid with single or dual channels. A likely configuration for safety critical applications is shown in Figure 6, which has dual channels and cascaded active stars. This configuration eliminates the single point failures from shorts or opens. Another provision of the FlexRay specification is the Bus Guardian, which can eliminate the babbling-idiot issue. These fault tolerant features make FlexRay an easy choice for safety critical applications.

The TTCAN network would probably be designed in a single channel bus configuration. A bus is susceptible to opens cutting the bus in half and to shorts disabling the bus. However, some of the fault tolerant features of FlexRay could be incorporated into a TTCAN design. A second channel could be used for a redundant connection on critical applications, and many automotive controllers already support more than one network. In addition, the bus guardian feature could be used to a TTCAN design.

**SUMMARY**

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>TTCAN</th>
<th>FLEXRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Channels</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Configuration</td>
<td>bus</td>
<td>bus, hybrid,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>active star,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cascaded star</td>
</tr>
<tr>
<td>Connection</td>
<td>twisted pair</td>
<td>twisted pair,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fiber optic</td>
</tr>
<tr>
<td>Bus Length</td>
<td>40 m</td>
<td>depends on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>configuration</td>
</tr>
<tr>
<td>Bit Rate</td>
<td>1 Mbps</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>Data Length</td>
<td>0 - 8 bytes</td>
<td>0 - 254 bytes</td>
</tr>
<tr>
<td>Header Length</td>
<td>11 - 29 bits</td>
<td>40 bits</td>
</tr>
<tr>
<td>Error Detection</td>
<td>15-bit CRC</td>
<td>11-bit CRC &amp;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24-bit CRC</td>
</tr>
<tr>
<td>Hamming Distance</td>
<td>5</td>
<td>6 or 4</td>
</tr>
<tr>
<td>Feedback from</td>
<td>Error &amp;</td>
<td>Not in frame</td>
</tr>
<tr>
<td>Receiving Node</td>
<td>Acknowledge</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 – Summary of key protocol features

**CONCLUSION**

As expected, the TTCAN protocol has some problems handling fault tolerant issues at the physical layer. Some of these issues could be addressed with the use of a secondary bus. The deterministic scheduling provided by TTCAN should meet the initial needs of
these safety critical applications. Further study could be done to simulate the performance of the TTCAN network that uses a secondary bus for fault tolerance.

The FlexRay protocol was conceived and designed for the type of safety critical applications that were the focus of this evaluation. Therefore, its exemplary performance on issues of fault tolerance was as expected. It also has a 10 to 1 advantage versus TTCAN on baud rate. Even with this advantage, there are concerns about the rigid message structure required by FlexRay, which may cause significant underutilization of the available communication bandwidth. Further study that simulates typical automotive network workloads should be performed to assess the potential severity of this issue.

REFERENCES

1. The Requirements of Future In-Vehicle Networks and an Example Implementation; Stephen Channon and Peter Miller; SAE 2004-01-0206
2. Multiplex Bus Progression 2003; Christopher A. Lupini; SAE 2003-01-0111
3. Future Trends in Networking; Cyrilla Jane Menon; SAE 2003-01-3738
6. How to Build Automotive Applications Based on the FlexRay® Communication System; Roman Nossal, Emmerich Fuchs, Thomas M. Galla, Roland Lang, Dietmar Millinger and Michael Sprachmann; SAE 2002-01-0439
8. Analysis and Diagnostics of Time Triggered CAN (TTCAN) Systems; Richard T. McLaughlin and Chris Quigley; SAE 2004-01-0201
9. TTCAN from Applications to Products in Automotive Systems; Patrick Leteinturier, Nico A. Kelling and Ursula Kelling; SAE 2003-01-0114
10. Time-Triggered Communication on CAN; Holger Zeltwanger; SAE 2002-01-0437
11. Time Triggered CAN (TTCAN); Thomas Fuehrer, Bernd Mueller, Florian Hartwich and Robert Hugel; SAE 2001-01-0073

CONTACT

Edward Gundlach
Graduate Student
Department of Electrical and Computer Engineering
Wayne State University, Detroit MI 48202
Phone: (313) 577-3855
Fax: (313) 577-1101
Email: as5358@wayne.edu

Dr. Syed Masud Mahmud
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

ABBREVIATIONS

CAN: Controller Area Network
TTCAN: Time Triggered CAN
TDMA: Time Division Multiple Access