Survey of Fault Tolerance Techniques in Automotives

Divya Jhalani and Shikha Dhir
Department of Electrical and Computer Engineering
University of Wisconsin, Madison
{jhalani, dhir}@wisc.edu

Abstract

This paper presents a survey of some of the fault tolerance techniques in automotives. Automotives need to be highly reliable with various safety-related systems in place. This increases the overall automotive and passenger safety by liberating the driver from handling the routine tasks and also assisting the driver during critical situations. One of the major trends to achieve this is by replacing the hydro-mechanical systems in the automotive with electrical systems. These systems, called x-by-wire, are fault-tolerant distributed systems which are fail-operational and can maintain a reliable and safe state at all times. These systems communicate with each other through a time-triggered architecture which is deterministic and fault-tolerant as well. This dual fault tolerance achieves an overall automotive safety critical to the well-being of everybody involved.

1. Introduction

There is an ongoing trend in the automobile industry to move towards more reliable and safer systems. These systems liberate the driver from handling the routine tasks and also assist the driver during critical situations by providing a dependable service and fault tolerance at all times. Various consortiums have been founded in the recent years to address this trend. For instance, the Brite EuRam III Project "Safety Related Fault Tolerant Systems in Vehicles" consisting of organizations such as Daimler-Benz Research, Fiat Research Center, Ford Europe, Volvo, Bosch, Magneti Marelli, Mecel, University of Chalmers, and the Vienna University of Technology, has come up with a new framework that introduces safety related fault-tolerant electronic systems without hydro-mechanical backup in automotives [1]. These systems are called x-by-wire systems.

The x-by-wire technology was previously seen in fly-by-wire systems, where the aerospace industry employed electronic systems in the aircrafts. These replaced the original hydro-mechanical systems and made the aircrafts more reliable and fault-tolerant. The application of this in automotives is, hence, apparent. Significant research has been done in this area and many of these techniques are in place today in numerous cars, such as Mercedes-Benz and BMW.

The x-by-wire systems are distributed, fault-tolerant and connected by a fault-tolerant real-time communication medium as well. This guarantees a system level fault tolerance necessary for automotives. These systems are also cheaper to manufacture with reduced packaging problems, no brake fluids or bleeding, and simple maintenance [1, 2]. These systems are easy to integrate with other well established systems such as anti-lock braking system (ABS) and electronic stability system (ESP).

Automotives consist of various components that interact with each other in a well defined manner. [1] presents a possible way to categorize these components according to their requirements and functions. The consumer electronics category comprises of features such as mobile phone, navigation system and WWW access. These electronics need high bandwidth and availability, but they are not critical to safety. The body electronics category consists of functions such as air-conditioning, car alarm, display, door module and light module. These need soft real-time communication requirements, i.e., perform the task without errors most of the time with occasional deadline misses acceptable. However, they need to maintain a fail-safe state at all times, i.e., even during faults the automotive continues to operate in a safe manner with possibly reduced functionality. Lastly, the system electronics category contains functions such as ABS, shift-by-wire and engine control which need to be fail-safe and have hard real time communication requirements, i.e. all tasks need to be finished within a time period. This category also consists of fail-operational systems like brake-by-wire and steer-by-wire, which need to maintain a fully operational state at all times. These are highly dependable and fault-tolerant systems.

This paper focuses on the fail-operational systems of the third category. The paper is organized as fol-

lows: Section 2 describes requirements for an automotive. These requirements can be met by the x-by-wire systems and the communication mechanisms, described in Section 3. Section 4 presents the fault tolerance techniques present in these systems. Section 5 presents an overall evaluation, and section 6 concludes the paper by providing some recent applications that use these new x-by-wire systems and time-triggered communication architecture.

2. Automotive requirements

Automotives have many constraints and requirements that need to be satisfied for their successful operation described in [1, 3, 4, 5] and summarized as following. They need to be operational in harsh environments with varying temperatures and noise levels. They also need to be eco-friendly, which has become an important factor in today's society. The reliability, availability, maintainability and lifetime of the automotives need to stay the same or increase with introduction of any new technology.

Automotives are produced at a very large scale and, hence, need to satisfy all mass production constraints such as low costs, system modularity and feasibility. Implementation of a new technology in the automotive should be compatible with the existing system to make it feasible. Automotives consist of components developed at different companies which are integrated in varying configurations to make the entire system. Therefore, nothing should be left to chance and all the parameters and subsystem behaviors need to be predefined.

The on-board diagnosis is one of the key features of today's automotives. They provide diagnostic information for maintenance purposes. For some intermittent faults, the system must be able to memorize the error code, which advices the driver to perform certain kind of maintenance. It should also be able to provide information about its internal status at all times.

For x-by-wire and other electronic features, the automotive must be able to provide higher data rates, deterministic behavior and support for fault tolerance. Therefore, safety of the automotive and the passengers should not be compromised. The system should still continue to perform in a fail-safe state even in the presence of one arbitrary fault. For example, if there is a fault in the steering system, the steering functionality must be maintained for a time long enough to reach a safe parking area. Therefore, safe (and possibly reduced) operation of the automotive needs to be guaranteed at all times. Furthermore, the fault should not propagate and affect the rest of the system.

All these automotive requirements are essential and need to be satisfied by the x-by-wire systems and other new technologies. The following sections survey some of these new technologies.

3. X-by-wire systems

The trend in the automotive industry to go towards safer systems has led to a lot of research on mechanisms to realize safer vehicles. In order to increase the overall automotive safety, the driver needs to be liberated from routine tasks and also be provided with assistance in critical situations. Hydro-mechanical systems originally present in automotives can not achieve this intelligent behavior. Therefore, they need to be replaced by electronic controls, which are fail-operational, dependable and cost effective for mass production.

The X-By-Wire project deals with this issue and presents a solution and a prototype utilizing various x-by-wire features, such as brake-by-wire, steer-by-wire and throttle-by-wire. For example, in brake-by-wire technology, traditional components such as the pumps, hoses, fluids, belts and brake boosters are replaced with electronic sensors and actuators [6]. The communication mechanism also plays an important role in these systems. It provides real-time communication channel for the data to move between the subsystems.

3.1. Case study: brake-by-wire

X-by-wire technology can be applied to multiple automotive systems. Brake-by-wire is an example of one of these systems. It replaces the hydraulic and mechanical systems with electrical wirings and sensors, that provide better fault tolerance and increases the reliability of the system. The brake-by-wire systems are also known as electro-mechanical brakes (EMB) and are easier to manufacture and maintain, and are environment friendly. This is because the brakes are controlled by computer chips rather than brake fluids and complex mechanical parts.

A possible architecture of an EMB system is shown in Figure 1, as proposed in [4]. For each brake, there is an Electronic Control Unit, or ECU, which processes the tasks for that brake, and some other local processors and sensors such as wheel speed sensors, steering sensors and motor sensors. There are also some actuators, power supply, motor and memory unit in this system. The communication happens on a channel employing a time-triggered architecture, which is centrally connected to a controlling unit, also known as BBW (brake-by-wire) manager. Figure 2 shows a detailed layout of this system employing brake-by-wire technology [6]. The controller is connected to the time-

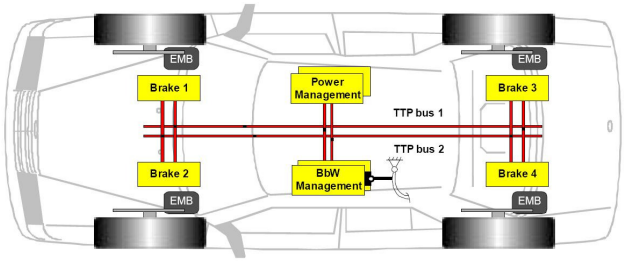

Figure 1. Brake-by-wire architecture
Source: [4]

triggered communication channel through which it interacts with the ECUs, power manager, actuators and sensors.

The following is a summary of the behavior of the EMB system, as described by [6]. When the brake is applied by the driver through the brake pedal, each brake's ECU generates an independent brake command. It gets sent to the brake's caliper via a time-triggered communication network. This message path is shown in Figure 2. The information analyzed by the controller in the caliper also includes sensor values from anti-lock braking system (ABS) and vehicle stability control (VSC), and other sensory data such as revolution counters of the wheels, clamp force and internal temperature [4]. Figure 2 shows that the brake pedal information is directly sent to the caliper as well, through a separate data bus.

The controller analyzes the brake command sent by the ECU in addition to information provided by the other sensors, in order to compute the drive control commands for the power management module. The power management module applies the required pressure or brake force on the brake actuators. In case there is a failure in the brake actuator, it can go into a fail-safe state, defined as open brake, with no brake torque [4].

Because the brakes are safety critical parts of an automotive, care needs to be taken when sampling the sensory data. A technique for compensating for missing samples is described in [8].

3.2. Communication mechanisms

Today's automotive systems employ complex networks connecting multiple subsystems together. Therefore, there needs to be some communication mechanism present to connect the subsystems together. Since 1980s, there has been a widespread use of Controller Area Network (CAN), which uses an event-triggered architecture. Some of the other commonly used networks are A-BUS, VAN, J1850-DLC, and J1850-HBCC. However, studies have shown that these fall short when it comes to building fault-tolerant x-by-

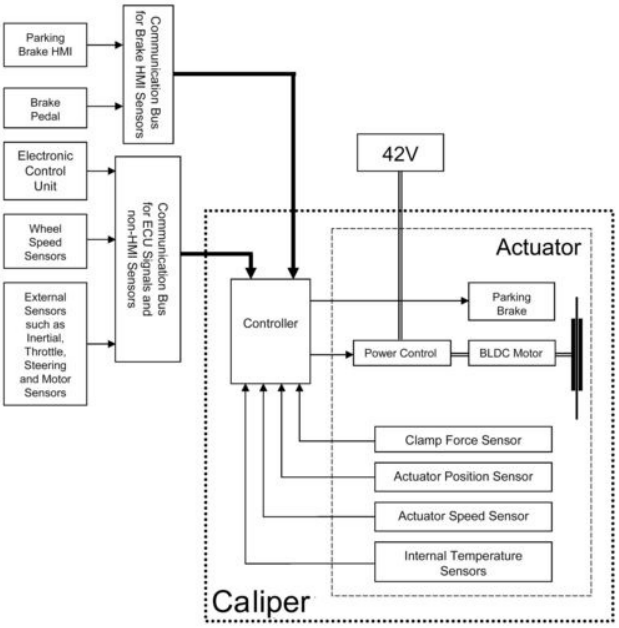

Figure 2. Detailed brake-by-wire architecture
Source: [6]

wire systems, such as brake-by-wire [11]. Although these systems are low-cost and provide medium bandwidth, they do not have deterministic behavior, synchronization and fault tolerance [2].

Research has been done since the introduction of x-by-wire systems to provide synchronized predictable communication mechanisms. These protocols consist of a time-triggered architecture, instead of the event-triggered architecture present in the original protocols such as CAN. Some examples of these protocols are Time-Triggered Protocol for Class C (real-time) applications (TTP/C) and FlexRay.

3.2.1. Controller area network (CAN)

Controller Area Network (CAN) is one of the most widely used communication network. It is an event-driven architecture in which events are assigned priorities in order to communicate over the network. It transmits message frames at speeds of up to 1 Mbps, with some systems commonly using 500 Kbps, such as engine control and ABS systems [12].

CAN is based on a message-oriented transmission protocol. Every message is unique in the network as it defines content and a priority, which is used for bus arbitration. The priorities present in the system are predefined, with higher priority assigned to tasks which need urgent communication such as engine load. Bus access conflicts are resolved by bit-wise arbitration, and all the nodes with lower priority lose the

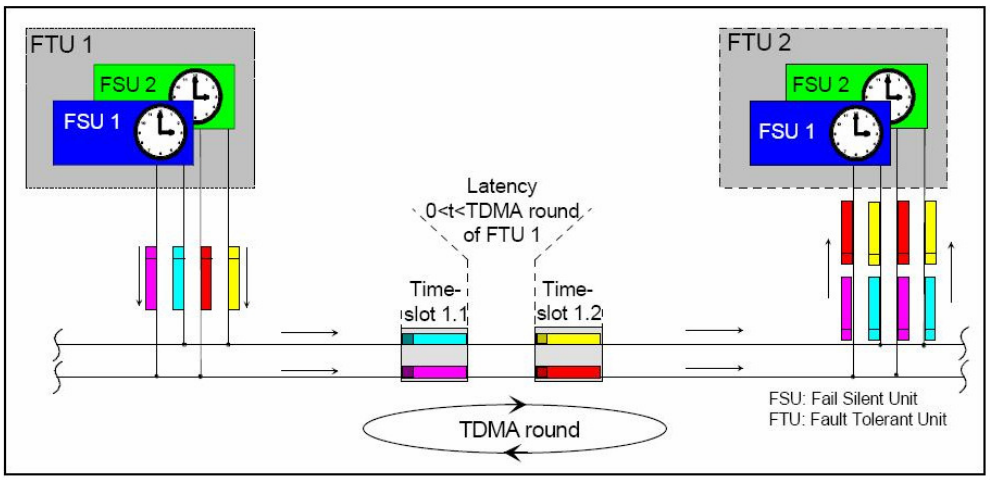

Figure 3. Time-triggered Protocol Architecture
Source: [2]

competition for the bus. They compete for the bus again as soon as the message gets delivered [13].

3.2.2. Time-triggered Protocol (TTP/C)

X-by-wire systems, such as brake-by-wire system, need a reliable time-triggered architecture for communication. Time-Triggered Protocol (TTP/C) is one of the protocols implementing these features developed by the University of Vienna and Daimler-Benz Research [2].

TTP/C utilizes Time Division Multiple Access (TDMA) technology, where each node connected to the bus has a predefined time slot when it can send a message. The length of the time slots may be different for different nodes. A TDMA round is a period of time in which all nodes send one message each. Figure 3 describes the connectivity of the nodes to the bus [2]. The overall TTP/C architecture consists of two kinds of subsystems. The host subsystem is run in a node, which executes the real-time application. The communication subsystem provides reliable real-time message transmission service. It consists of two channels that transmit information independently. The interface between these two subsystems is called Communication Network Interface (CNI), which is a dual-ported RAM with an interrupt line between the host subsystem and the controller [2, 14]. When the host wants to send something over the network, it "pushes" the information onto the CNI memory. The communication system transports the data provided in this memory at the appropriate time to the receiver node's CNI memory. The receiver node then "pulls" the information from the CNI memory. The interrupt line between the CNI and the host subsystem is used to signal the ticks of the global synchronized time and to inform the node about the occurrence of significant events such as mode changes or transmission errors [15].

The TTP/C protocol supports multiple modes to make the communication system behave differently at different points in time [14]. For example, in the start-up mode, only node synchronization information is exchanged, whereas in the normal mode of operation all the application-related data is exchanged.

3.2.3. FlexRay

FlexRay is a new automotive network communication protocol developed by the FlexRay Consortium. This consortium consists of members like Volkswagen, BMW, DaimlerChrysler, General Motors, Robert Bosch GmbH, NXP Semiconductors and Freescale [16]. FlexRay provides high data rates (10 Mbps), time-triggered behavior and redundancy, safety and fault tolerance.

The FlexRay protocol is specifically designed for automotive networking, with x-by-wire systems and other electronic systems in place. However, it also works in conjunction with already well-established systems such as CAN [17], instead of replacing these existing networks. The FlexRay protocol is used by critical applications such as brake-by-wire. It also provides a backbone providing determinism for the engine control and other applications in order to make the entire system fault-tolerant [17].

FlexRay supports various network topologies [18] in order to make a reliable communication network which is also redundant and fault-tolerant. The nodes can be configured as a single- or dual-channel bus network. Network can also be configured in a star topology, which contains two star couplers that can support redundant communication channels. Each network must be free of closed rings, and can only

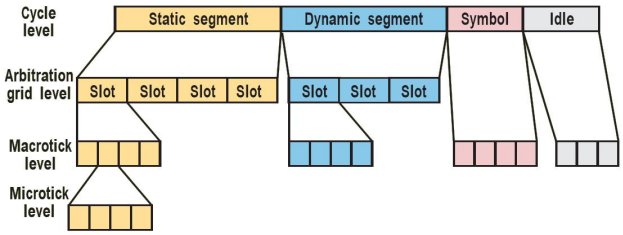

Figure 4. FlexRay communication cycle
Source: [19]

support two channels or two star couplers. The incoming signal received by the star coupler gets propagated to all the other connected nodes [18]. Hybrid topologies are also possible, with both bus and star configurations. When constructing the communication network in an automotive, redundancy and node connectivity can be varied in order to get the best possible fault tolerant network.

The FlexRay system consists of a bus and node processors (Electronic Control Units, or ECUs) connected in various topologies as mentioned above. Each ECU has an independent clock and, hence, all these clocks must be synchronized with each other. The maximum allowable clock drift is 0.15% compared to the reference clock. Therefore, the difference between the slowest and the fastest clock is a maximum of 0.3%. The clock synchronization process is frequent enough to assure that this drift gets minimized making the system function normally at all times [16].

In FlexRay protocol, each cycle consists of a static segment, a dynamic segment, a symbol window and network idle time. Figure 4 taken from [19] shows this division of the communication cycle in the four segments. These segments are defined in terms of a timing hierarchy [18]. The segments consist of arbitration grids, which, in turn, consist of macroticks. This arbitration grid ensures that the communication is collision-free in both the static and dynamic segments. In the lowest level of hierarchy, the cycle is divided up into fixed length microticks, which allows a strict timing for reliable communication. The static segment implements a static time division multiple access (TDMA) scheme to arbitrate transmissions. The dynamic segments implements a dynamic mini-slotting based scheme to arbitrate transmissions. The symbol window is a communication period in which a symbol can be transmitted on the network to allow the realization of redundant communication path. Finally, the network idle time is a communication-free period which concludes the cycle [18]. The static and the dynamic portions do not interfere with each other and the system can run in either static or dynamic or a hybrid mode. The time-critical applications use the static scheduling scheme with predefined time slots to achieve a time-triggered architectural mode. Other applications in the system can use the dynamic segment to model an event-triggered architectural mode.

4. Fault Tolerance

Since there is no hydro-mechanical backup system in place in the x-by-wire systems, it is very important for these systems to be fault-tolerant and reliable at all times. This is achieved by fault-tolerant communication protocols and hardware redundancy. The x-by-wire system also needs to be composable to make it easier to add a system to an existing automotive with varying technologies, and integrate multiple technologies together.

One of the major requirements is that the system needs to be functional even in the presence of one arbitrary fault. This fault should not propagate and affect the rest of the system. This is done by making the automotive subsystems fail-silent which provides an overall reliable fault-tolerant service. A fail-silent component detects all transient and permanent faults and encapsulates them, resulting in a self-disabling of the component. Therefore the fault doesn't get propagated to the rest of the system.

4.1. X-by-wire subsystems

X-by-wire systems employ redundancy as a key mechanism to achieve fault tolerance. For example, for the brake-by-wire case study seen earlier, Figure 2 showed that the brake pedal information is directly sent to the caliper as well, through a separate data bus. This provides a redundancy mechanism in case a fault occurs in the communication channel or ECU.

Fault containment region is required to achieve a non-faulty system [22]. An ECU constitutes a fault-containment region since an error in a part of the ECU will affect the entire ECU. The ECUs are replicated with the use of triple modular redundancy (TMR). Other n-modular redundancy (NMR) techniques can also be employed. An example of such system is shown in Figure 5. The outputs of the redundant units are checked with a voter to get a majority vote. Various voter techniques can be used to achieve this. Some techniques that have been used in this area are: n-input

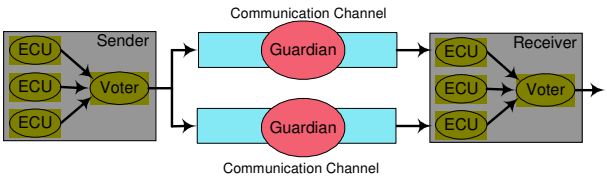

Figure 5. Hardware redundancy with TTP/C

majority voter, plurality voter, median voter, weighted average voter, and fuzzy voter [23].

N-input majority voter provides a correct result if the majority of the inputs agree to a value. If there is a tie, then an exception flag is turned on, which makes the system go into a fail-safe mode. Plurality voter is another extension of majority voter. Here, a strict majority is not required to get a correct answer. A number a less than the strict majority is acceptable. In median voter scheme, mid-value is selected among the incoming inputs. Given an odd number of inputs, this scheme can be used to successfully eliminate the outlying values. The weighted average voter calculates the weighted mean of the inputs to get the final result. Finally, the fuzzy voter employs a voting technique where a fault is gradually removed from the system by the use of weights. This provides a higher noise tolerance than other voting schemes.

4.2. Communication subsystems

Controller Area Network (CAN) has some message error detection capabilities such as Cyclic Redundancy Check (CRC) and acknowledgement (ACK) error detection. If a message is invalid due to bit failure, the associated CRC will show this inconsistency. Since CAN is an event-triggered protocol, there are also ACKs that get sent out by the receiver after the transmission completes. The protocol checks for transmission errors by looking at the ACK values. The CAN protocol provides a statistical mechanism for distinguishing sporadic errors from permanent errors and local failures at the station. This enables a possible shut-down of the affected node and, hence, the network is not negatively affected if the fault is correctly identified [13].

However, these error detection capabilities are not enough to provide an overall fault-tolerance for x-by-wire technologies. Since the CAN protocol is an event-driven protocol, it fails to be deterministic and provide reliable service.

Time-Triggered Protocol (TTP/C) and FlexRay protocol, on the other hand, provide a deterministic behavior ideal for real-time communication. They can tolerate any single arbitrary fault without any effects to the overall system behavior. The synchronization of the ECUs in the system is done on the basis of a global sparse time. An event occurring within a delta value of the clock tick is considered to be at the same time. This removes the errors caused due to clock skews in the system.

Figure 5 shows TTP/C communication channel containing a unit called guardian. This guardian is an independent unit that has predefined timings available for all messages that get sent over the network. The two duplicated communication channels have their own guardians. The sender node's message gets copied over to the two channels and then independently checked for timing errors by the two guardians. If the guardian detects an untimely message, the message is invalidated. At the receiving end, if the two copies of the message are both correct, then whichever copy arrives first is used in further computations. In other words, the sending node is judged as operational as long as it generates at least one timely message. In FlexRay, the guardian is associated with the ECU, and hence, comes under the fault containment region of the ECU.

[22] defines five failure modes that can exist in the communication mechanism:

- Babbling idiot failure
- Masquerading failure
- Slightly-off-specification (SOS) failure
- Crash/Omission (CO) failure
- Massive transient disturbances

The first four kinds are subcategories of the arbitrary failure mode. It is assumed that only one failure can occur at a time in the system.

The babbling idiot failure is when the ECU sends untimely message. This can potentially cause communication disruption between the working nodes. The TTP/C protocol deals with this failure with the use of the guardian unit. If an untimely message is detected, it will be immediately invalidated. The FlexRay protocol, on the other hand, may allow babbling idiot failures to happen during its dynamic transmission segment. However, note that the static segment is predefined and used for real-time critical applications and the guardian present in the ECU checks for the message transmit times during this period. Hence, for x-by-wire systems, the babbling idiot failure doesn't pose a threat.

The masquerading failure can occur if an erroneous node assumes the identity of another node. The system can be harmed if this occurs as the erroneous node may then send messages to the system, without the receiving node being able to detect it. This can potentially propagate the errors to the rest of the system. TTP/C does not allow for this failure to occur as the timings and the sender-receiver information of all the messages are known beforehand. FlexRay protocol's static segment does not allow for this failure to occur either because of similar reasons. However, the dynamic segment is susceptible to them.

The SOS failure can occur during analog to digital conversion. Since the subsystems are produced by different manufacturers, they might deviate from the specifications a little in timing, frequency or voltage. Therefore, the node's receive window can be made wider to accept these slight deviations to ensure that inputs get received correctly. However, if an erroneous

node produces an incorrect signal with SOS failure, it would not be detected due to this wider window, causing the error to propagate into the system. The TTP/C protocol does not allow this failure to occur with the use of the guardian unit. This unit accepts all the messages and checks for timing failures. It also checks for SOS failures by comparing the message format to the predefined specifications and doing other tests such as message CRC. If an error is detected, the message is invalidated. The FlexRay protocol also achieves this for the static segment with the use of the guardian present in the node.

The CO failures happen when a node crashes or the communication channel fails to operate. It is one of the most common failures in the distributed systems setting. These failures are detected by the TTP/C protocol with the use of membership management service. Every node knows about the state of every other node in the distributed system. Thus, if CO failure occurs, it gets detected by the other nodes immediately. The FlexRay protocol does not provide detection mechanism for this failure. Therefore, it is left to the application running on the nodes to detect it.

The massive transient disturbances such as electromagnetic emission can cause failures in the automotives. This results in temporary communication loss in the system. These failures can be avoided with engineering principles such as shielding the copper cables or replacing copper wire with fiber optics. These failures are very rare, but when they occur, fast detection is needed to make the non-faulty nodes remain in their correct states. One way to achieve this is by freezing the nodes to a state before the failure occurred, if the communication loss is for less than 50 msec. The TTP/C protocol detects these failures with the use of membership management service and the clique avoidance algorithm. The FlexRay protocol does not provide any mechanism to deal with this failure. The detection is left to the application running on the nodes.

The TTP/C protocol also has some other services which help in the fault tolerance of the entire system [14]. It allows for a temporarily failed ECU to get reintegrated back into the system by receiving the history state from the other ECU in the node. The clock synchronization in TTP/C protocol is achieved by running a fault-tolerant averaging algorithm that ensures fault-tolerant synchronization within microsecond range.

The FlexRay protocol, on the other hand, provides some services which help in the fault tolerance as well [17]. There is a startup service that provides an autonomous startup procedure. The wakeup service addresses the power management needs of the system. The diagnosis service tests the bus guardian on the physical layer. This helps detect faults in the bus guardian. In this protocol, the synchronization of the global time happens at the macrotick level, with the use of a cluster-wide clock synchronization algorithm. This algorithm ensures that the precision of the clocks, i.e., the worst-case deviation between the corresponding macroticks, of any two ECUs is within the 0.3% limit [20]. This clock synchronization algorithm continues to operate even in the event of an ECU failure in the system, unlike a master-slave synchronization algorithm. Each ECU runs this algorithm independent of the host processor, and hence, is not affected by the states of the other ECUs [20].

5. Evaluation

5.1. Evaluation of brake-by-wire

Brake-by-wire systems are meant to be used to replace the hydro-mechanical systems currently present in automotives. They are also not supported by any backup hydro-mechanical systems, which take over in the presence of a fault. Therefore, the brake-by-wire systems need to be highly reliable and fail-operational at all times. They are designed such that they can support one arbitrary fault and still be in an operative state.

[4, 6] describe the brake-by-wire system's key features that make it fault-tolerant. The actuators in the brake-by-wire system can function with no braking torque, in case there is a power loss. The sensors have triple redundancy with a voting mechanism in place. There are also redundant copies of some of the signals that are essential to ensure safety, such as displacement and force measurements of the brake pedal. Other redundant subsystems include power supply and ECUs. For example, in case one ECU fails, the other ECU can take over without affecting the rest of the system.

Some of the sensory data is very critical for the automotive's function and, by extension, human lives. Such sensors include the brake pedal and wheel speed sensors. The brake pedal sensor informs the system of the driver's intention of how much braking he or she wants. The wheel speed sensor informs the system of the current speed of the wheel, in order to avoid skidding. Missing data from these sensors gets compensated by algorithms such as time delay neural network (TDNN) presented in [8].

A brake-by-wire system is capable of reacting more quickly than a conventional hydro-mechanical braking system [7]. This results in shorter stopping distance, increasing the automotive's reliability and safety. Some aesthetic issues such as vibrations due to braking are also resolved as the brake-by-wire systems

are silent with no mechanical parts involved. They also consume less physical space.

As [9] suggests, the brake-by-wire system can also be used to automatically apply the parking brake. Therefore, the traditional manual parking brake is no longer needed. Furthermore, these systems are eco-friendly and can be integrated with the rest of the system in the automotive without any compromise to fault tolerance, reliability and safety [10].

5.2. Comparative assessments of communication protocols

Both the Time-Triggered Protocol (TTP/C) and the FlexRay protocol consist of a time-triggered architecture where communication takes place in predefined time slots. On the other hand, the Controller Area Network (CAN) protocol consists of event-driven architecture where the communication is priority-driven. For x-by-wire systems, the protocols implementing time-triggered architecture are better suited since they provide a deterministic behavior and have various fault tolerance features.

[14] lists various advantages that TTP/C has over event-triggered protocols such as CAN. The separation of data and communication tasks within an Electronic Control Unit (ECU) makes sure that a fault in the processing part of the node can not affect the time at which messages are transmitted on the bus. This feature prevents fault propagation from the faulty node to the rest of the system. Another feature that TTP/C provides is composability. This is very important in automotives since the various subsystems may be produced by different manufacturers. It ensures that the properties of a subsystem stay the same even after its integration into the rest of the system. This makes the testing and verification of individual components easier.

Another feature of TTP/C is the predictability of the message latency. Because of the predefined time-slots, the variations in message latency, or jitter, can be made really small. On the other hand, event-triggered protocols may have variable message latencies, depending upon the current number of higher priority messages in the system waiting to be serviced. Furthermore, a synchronized global clock provides a consistent view of the system at all times.

In CAN message collisions are normal and need to be resolved with a bit-by-bit arbitration mechanism. This requires that the bit rate on the bus is below a certain limit. However, in TTP/C, the bus bandwidth is not limited providing much higher bit rates. Finally, TTP/C is conceptually simpler and predictable and, hence, easier to analyze than event-driven protocols like CAN. This is really helpful in diagnosing errors. TTP/C also provides a cleaner platform for introduction of fault tolerance mechanisms, increasing the reliability of the entire system.

The FlexRay protocol is also compared to the event-driven protocols, specifically the CAN protocol, in [20]. As mentioned above, CAN fails to provide large bandwidths due to message collision detection mechanism. FlexRay, on the other hand, can provide high data bandwidths of up to 10 Mbps. The FlexRay protocol also has deterministic communication just like TTP/C. It provides bounded latencies and small jitter. In CAN, integrating the subsystems together can cause unexpected side effects due to increased bus load. However, FlexRay provides composability, which makes system integration easy to perform.

Fault tolerance issue is not dealt with in CAN. The inherent protocol can not guarantee application level fault tolerance at all times. For example, a faulty node may affect the entire system by congesting the bus. FlexRay, on the other hand, provides fault tolerance at various levels. It provides deterministic communication mechanism, redundant channels, fault-tolerant clock synchronization, and bus guardian ability. FlexRay also provides higher performance, more extensibility and lower complexity, making the detection of faults and error diagnosis easier and faster.

Both TTP/C and FlexRay protocol are good candidates for communication in the x-by-wire systems. Although TTP/C provides more fault tolerance in the hardware, FlexRay provides the flexibility with both the time-triggered and event-triggered architectures in-built. An application-level fault tolerance mechanism is expected in both the protocols, to provide more reliability.

6. Conclusion

Electronic systems provide more fault tolerance than the hydro-mechanical systems traditionally used in automotives. They are safer, more reliable and dependable and can be integrated with the other existing systems easily. These x-by-wire systems need real-time fault-tolerant communication mechanism as well. The two time-triggered communication mechanisms surveyed in this paper, TTP/C and FlexRay, provide features that help maintain a fault tolerant state in the entire system at all times.

Various automotives have come out in recent years that utilize some form of fault tolerance techniques surveyed in this paper. For example, the Mercedes Benz E-class and SL models, Toyota Estima, some models of Nissan and BMW use a version of brake-by-wire technology. Fiat and Volkswagen will also introduce brake-by-wire in their future automotives [7]. The 2006 BMW X5 uses FlexRay communi-

cation mechanism for the pneumatic damping system. The full use of FlexRay will be present in 2008 in this model [16].

References

[1] E. Dilger, T. Führer and B. Müller, "Distributed Fault Tolerant and Safety Critical Applications in Vehicles – A Time-Triggered Approach," *Proc. of the 17th International Conference on Computer Safety, Reliability and Security*, pp. 267-283, 1998.

[2] T. Ringler, J. Steiner, R. Belschner and B. Hedenetz, "Increasing System Safety for By-Wire Applications in Vehicles by Using a Time Triggered Architecture," *Proc. of the 17th International Conference on Computer Safety, Reliability and Security*, pp. 243-253, 1998.

[3] A. Manzone, A. Pincetti and D. De Costantini, "Fault Tolerant Automotive Systems: An Overview," *Proc. of the 7th International On-Line Testing Workshop*, pp. 117-121, 2001.

[4] B. Hedenetz and R. Belschner, "Brake-by-wire without Mechanical Backup by Using a TTP-Communication Network," *SAE Congress Proceedings*, 1998.

[5] Günter Heiner and Thomas Thurner, "Time-Triggered Architecture for Safety-Related Distributed Real-Time Systems in Transportation Systems," *Proc. of the 28th Annual International Symposium on Fault-Tolerant Computing*, pp. 402-407, 1998.

[6] Reza Hoseinnezhad, "Position Sensing in Brake-by-Wire Callipers Using Resolvers," *IEEE Transactions on Vehicular Technology*, 2006.

[7] "What is Brake by Wire?," [Online; accessed 21-April-2007]. Available: http://www.brakebywire.com/brake-by-wire.html

[8] Reza Hoseinnezhad and Alireza Bab-Hadiashar, "Missing Data Compensation for Safety-Critical Components in a Drive-by-Wire System," *IEEE Transactions on Vehicular Technology*, pp. 1304-1311, 2005.

[9] Tom Ripley, "Brake by wire," Nov. 2006, [Online; accessed 22-April-2007]. Available: http://www.drivers.com/article/906/

[10] Freescale Semiconductor, "Electromechanical Braking (Brake-By-Wire)," [Online; accessed 22-April-2007]. Available: http://www.freescale.com/files/shared/doc/selector_guide/SG2008.pdf

[11] Markus Krug and Anton V. Schedl, "New Demands for Invehicle Networks," *23rd EUROMICRO Conference*, pp. 601, 1997.

[12] Thomas Nolte, Hans Hansson, Lucia Lo Bello, "Implementing Next Generation Automotive Communications," *1st Embedded Real-Time Systems Implementation Workshop in conjunction with the 25th IEEE International Real-Time Systems Symposium*, Dec. 2004.

[13] CAN in Automation, "Controller Area Network (CAN) – Protocol," [Online; accessed 21-April-2007]. Available: http://www.can-cia.org/can/protocol/index.html

[14] X-by-Wire Safety Related Fault Tolerant Systems in Vehicles, Project No BE 95/1329, Contract No: BRPR-CT95-0032 Final Report Version 2.0.0, Nov. 1998.

[15] H. Kopetz and T. Thurner, "TTP – A New Approach to Solving the Interoperability Problem of Independently Developed ECUs," *SAE Congress Proceedings*, 1998.

[16] Wikipedia, "FlexRay --- Wikipedia, The Free Encyclopedia," 2007, [Online; accessed 21-April-2007]. Available: http://en.wikipedia.org/w/index.php?title=FlexRay&oldid=124300758

[17] Freescale Semiconductor, "FlexRay Network," [Online; accessed 21-April-2007]. Available: http://www.freescale.com/webapp/sps/site/application.jsp?nodeId=02Wcbf12813975124F

[18] FlexRay Communications System Protocol Specification, v2.1 Revision A, FlexRay Consortium, December 2005.

[19] W. Wong, "Design FlexRay FAQs," Jan. 2006, [Online; accessed 21-April-2007]. Available: http://www.elecdesign.com/Files/29/11820/11820_01.pdf

[20] J. Berwanger, et. al., "FlexRay hits the road," Nov. 2006, [Online; accessed 21-April-2007]. Available: http://www.automotivedesignline.com/194400425;jsessionid=K54YXMUGSLDHMQSNDLQSKHSCJUNN2JVN?printableArticle=true

[21] Kermit Whitfield, "Solve for X," *Automotive Design and Production*, Dec. 2001, [Online; accessed 21-April-2007]. Available: http://findarticles.com/p/articles/mi_m0KJI/is_12_113/ai_81147817

[22] H. Kopetz, "Fault Containment and Error Detection in TTP/C and FlexRay," *Research Report 23/2002*, v1.5, Technical University of Vienna, 2002.

[23] Reza Hoseinnezhad and Alireza Bab-Hadiashar, "Fusion of Redundant Information in Brake-by-Wire Systems Using a Fuzzy Voter," *Journal of Advances in Information Fusion*, pp. 52-62, 2006.