A PERIOD-BASED GROUP MEMBERSHIP STRATEGY
FOR NODES OF TDMA NETWORKS

Elizabeth Latronico

ECE Department
Carnegie Mellon University
Pittsburgh, PA, USA
beth@cmu.edu

Philip Koopman

ECE Department
Carnegie Mellon University
Pittsburgh, PA, USA
koopman@cmu.edu

Group membership provides strong guarantees for safety-critical fieldbus systems. However, using a single group for all messages provides an unnecessarily high risk of temporary node outages in the face of transient faults. Using groups based on virtual nodes that are divided by message period can increase the availability for the most critical, high-speed network messages with potentially reasonable bandwidth cost and without giving up the assurances of strong group membership algorithms.

Keywords: Automobile industry, Availability, Communication protocols, Embedded systems, Fault tolerance, Fieldbus, Group membership, Safety-critical, TTP/C

1. INTRODUCTION

In automotive embedded systems, all nodes are not created equal. Automotive embedded systems generally have a variety of nodes, sending messages with up to two orders of magnitude difference in period. For example, a brake activation message might be sent every 5 milliseconds while a battery status message might be sent at 1000 milliseconds (Tindell and Burns, 1994). Cost concerns often preclude point-to-point connections between each pair of nodes, leading to use of a broadcast bus for messages with such significantly different periods. Current automotive buses such as the Controller Area Network (CAN) use a priority-based message sending scheme (CAN, 1991). However, there is pressure to adopt a time-based sending scheme in order to provide a more predictable platform for safety assurance.

Time Division Multiple Access (TDMA) networks provide a statically-scheduled method of transmitting data on a broadcast bus. TMDA networks contain slots for each message to be transmitted. Messages are transmitted in frames, which also include overhead. Slots are defined according to their order in a round. Each node typically has at least one slot per round (Bauer and Paulitsch, 2000). Rounds are assembled into a cluster cycle (TTP/C, 2002). Typically a dual-redundant bus is used, and a node sends each message once on each of the two channels.

TDMA systems often allow nodes to share their views of the system with other nodes. These group membership services protect against a variety of faults including processor faults, link faults, and noise on the communication bus (Kim and Shokri, 1993). The faults may be permanent or transient. Group membership services form the basis for important services such as replication and clock synchronization. In automotive embedded systems, there is typically a single group, and recovery involves ensuring that all nodes eventually belong to that single group.

Unfortunately, with a single group strategy it is a difficult task to label a fault as permanent or transient. Each frame in an embedded network typically carries a Cyclic Redundancy Code (CRC) to detect if the frame has been corrupted. If a node receives a frame with an invalid CRC, it cannot tell from this information alone whether the fault is permanent or transient. The source of the fault is also unknown – the sender could be faulty, or the bus could be noisy. Therefore group membership services in state-of-the-art TDMA protocols, such as TTP/C, must take a pessimistic approach upon receiving a faulty frame. A node that receives faulty frames on both bus channels will consider the sender to be faulty. If enough nodes consider the sender to be faulty, the sender loses membership in the group and must reintegrate.

A more optimistic approach is possible if multiple, period-based groups are used. This paper demonstrates how the same group membership algorithms for a single group strategy can be used in a multiple group strategy. Greater tolerance is achieved for transient faults caused by noise on the communication bus at an acceptable bandwidth cost, without altering the group membership algorithm. This technique is demonstrated for a braking application based on SAE benchmark data.

Section 2 discusses domain characteristics and the Society of Automotive Engineers (SAE) workload from Tindell and Burns (1994) that is used as a reference example. Section 3 presents relevant concepts of group membership and explores a standard single group system. Section 4 presents our multiple group solution. Section 5 presents an availability and bandwidth analysis comparing the single group and multiple group strategies.

2. DOMAIN

A number of system constraints help structure the solution space. First, the SAE standard workload from Tindell and Burns (1994) is reviewed as a representative automotive workload. Next, other relevant constraints are discussed.

The SAE standard workload (Tindell and Burns, 1994) contains a set of periodic and sporadic messages sent in a prototype electric car with seven subsystems. These subsystems include the Batteries (Battery), Brakes (Brakes), Driver (Driver), Inverter/Motor Controller (I/M C), Instrument display panel (Ins), the Transmission control (Trans), and the Vehicle Controller (V/C). For our purposes it is assumed that each subsystem constitutes one node except the Brakes, where it is assumed that there will be one Brakes node per wheel for a total of four Brakes nodes. Actual systems might differ from this configuration; the workload in Tindell and Burns (1994) was originally designed for a point-to-point system. The SAE workload contains messages with six different periods: 5 ms, 10 ms, 20 ms, 50 ms, 100 ms, and 1000 ms. All of the 50 ms messages are sporadic messages, but are assumed to have a 50 ms period as Tindell and Burns assume (1994), in accordance with standard automotive practice.

Embedded systems are often highly constrained, and those constraints can be used to our advantage. In particular, the following properties are useful:

- *Harmonic periods*
  Messages are commonly scheduled with harmonic periods so it is easier to prove schedulability. Hence, messages can easily be grouped by period.
- *Period and deadline usually equal*
  A message's period is often the same as its deadline. If so, increasing the period is not an option.
- *Short payloads relative to overhead*
  Data payloads are often on the order of one to eight bytes long. Other fields in a frame typically include an ID field and Cyclic Redundancy Code (CRC) field for error checking, which can consume a few bytes. A solution must be cautious about adding overhead, but overhead is acceptable in many cases.
- *Sender has "ground truth"*
  Regardless of how many other nodes disagree, the message sender is the node that has the correct state variable value of a message being sent . Therefore it is best to treat transient errors differently than permanent errors if possible, as nodes are typically not interchangeable.

3. GROUP MEMBERSHIP CONCEPTS

This paper will show that forcing all nodes to be members of a single group is a limiting restriction. Specifically, this lowers availability when transient faults are treated in the same manner as permanent faults. A single faulty frame causes a sending node to lose membership even if the fault is due to noise on the communication bus. This paper shows that having multiple groups can ease the effects of this pessimistic restriction. There are several known group membership algorithms, with varying levels of guarantees. This work refers to the group membership algorithm of /C.

A key advantage to this approach is that the group membership algorithm remains unchanged. This allows results from existing proofs to be reused. A central design problem for any group membership service is determining when a node should lose membership, if at all. Inventing a new algorithm is difficult - there are many tradeoffs and subtle points to consider. Alternately, availability of the system can be increased by using multiple groups, operating by the same rules a single group would operate by. The next sections review relevant group membership concepts and constraints.

3.1. *Fault Model*

Group membership algorithms are usually designed to withstand node crashes, send faults, and receive faults. Algorithms handle both permanent and transient faults, typically with restrictions on fault interarrival rates (Kim and Shokri, 1993). Group membership algorithms cannot compensate for loss of network connectivity or semantically incorrect data that is syntactically correct. Group membership requires at least four nodes to tolerate one faulty node (Pfeifer, 2000). Faulty nodes that lose membership may reintegrate into the system, after the group has reached consensus on its members. Consensus is guaranteed to be reached within two rounds after a fault has been identified (Pfeifer, 2000). If a fault occurs in the group, additional faults are not tolerated while nodes in the group have inconsistent views of membership, although better fault tolerance is possible for some faults if a slightly longer time is allowed (Kim and Shokri, 1993).

3.2. *Clique Avoidance, Implicit Acknowledgment*

Clique avoidance is one of two mechanisms employed in order to ensure that a group does not partition into two or more separate groups, called cliques (Bauer and Paulitsch, 2000). Each node maintains a list regarding who it thinks the members of its group are, sometimes called a membership vector (TTP/C, 2002). Since a node considers a frame incorrect if the sender does not have the same membership vector, nodes in separate cliques would not be able to communicate with each other. Clique avoidance is also designed to identify nodes that are receive-faulty. Clique avoidance requires a node to have received more correct frames than faulty frames in the last round in order to retain membership (not counting null frames). Clique avoidance may prohibit a node from sending a frame in its next two slots following a fault, sometimes when the node was not the source of the actual fault (Bauer and Paulitsch, 2000).

Implicit acknowledgment ensures that a faulty sender will lose membership. After sending a frame, the sending node waits to see if subsequent nodes have received its frame. Protocols use some sort of a broadcast membership vector per node (either explicit or implicit) to relate a node's opinion of who is in its group. A sender will lose membership if not enough other nodes receive the frame correctly (Pfeifer, 2000).

3.3. *Performance Implications*

Due to the interaction of clique avoidance and implicit acknowledgment services, group membership requires at least one round and at most two rounds to achieve consistent membership (Bouajjani and Merceron, 2000). A node may also be prohibited from sending frames during these two rounds to ensure consensus on a single group is reached.

According to the TTP/C specification and group membership proofs, each node transmits exactly once per round. Specifically, if all nodes in the system belong to a single group then:

- Each node must transmit at least once per round (Bauer and Paulitsch, 2000)

In order for the system to reach consensus, it must hear from all member nodes. Mandating that each node must transmit at least once per round allows the guarantee of a maximum of two rounds to achieve consistent membership to be made.

- Each node may transmit at most once per round (TTP/C, 2002, p. 18)

The TTP/C specification does not list a specific reason for this; however, it can be inferred that allowing a node to transmit multiple times per cycle would give this node an unequal weight in the failed slots counter that is incremented every time a faulty frame is received. Also, a sending node must always wait for at least one subsequent valid frame to be acknowledged (Bauer and Paulitsch, 2000).

Therefore, for a system with a *single* group, each node must transmit exactly once per round (although it is not necessarily the same message that is transmitted each round). The TTP/C protocol also allows shared slots, where distinct nodes (called multiplexed nodes) may alternate sending messages in a designated slot in a round (TTP/C, 2002). Multiplexed nodes are not employed here, because the results are undesirable regardless of whether the group membership algorithm considers these nodes to be separate member nodes or a single member node. If the group membership algorithm considers the nodes sending in the shared slot to be separate member nodes, then the time to achieve consistent membership will increase as consensus requires the opinions of all member nodes. If the group membership algorithm considers the nodes as a single member node, then loss of membership for the member node implies that all of the nodes sending in the shared slot will lose membership. This work also does not consider redundant nodes with distinct sending slots due to space considerations. Redundant nodes with separate slots would consume a larger amount of bandwidth.

4. OUR SOLUTION

In order to tolerate transient faults, one can take advantage of the fact that nodes often send messages at different periods. Therefore, redundant information about the state of a node is available – a single corrupted message might be considered to be a transient failure if the next type of message from that node is correct. However, transmission of the next type of message from that node might be suppressed by the clique avoidance algorithm. Thus, the next type of message might not be sent.

In order to track different message types separately, message periods need to be the basis for group membership, not physical nodes. The obvious approach to separating messages is to try to create separate groups of physical nodes. Unfortunately, in general it is difficult to split automotive network nodes into disjoint sub-groups. For example, one cannot create two distinct groups of nodes for the SAE workload because the Vehicle Controller is either the producer or a consumer for all messages.

Table 1 shows the physical sending nodes and their sending periods in this system, and the total number of payload bits sent per period. For example, the Battery node sends 8 bits worth of data every 50 ms, 32 bits of data every 100 ms, and 17 bits of data every 1000 ms. These numbers only include payload data, not other fields in the frame, which will be discussed in the Performance Analysis section. This paper assumes four Brakes nodes instead of a single Brakes node as in Tindell and Burns (1994).

4.1. *Virtual Groups and Virtual Nodes*

Instead of anchoring our groups on physical nodes, we use *virtual groups* made up of *virtual nodes.* The algorithm for constructing virtual groups is to create one virtual group per unique message period. A virtual node is created according to the periods of messages that a physical node sends. One virtual node is created per period for each physical node. Each virtual group must have at least four members (as Pfeifer shows is required to tolerate one faulty node) (2000). If there are fewer than four distinct physical nodes sending messages at a particular period, the system designer has two choices. The designer can elect to send a message more frequently, and assign the virtual node to a smaller period virtual group. Alternatively, the designer may create additional virtual nodes by having physical nodes send placeholder messages at that period. In general, virtual nodes sending placeholder messages can be added to any virtual group if additional fault tolerance is desired.

Tables 2 through 6 show the virtual groups for the SAE benchmark system. There is one virtual group per unique message period in our system, with the exception that there is no 10 ms group. There is only one message sent at 10 ms, and a group of one node will not be fault tolerant, so this message is sent at 5 ms instead, incurring a small amount of extra bandwidth. VirtualGroup5 only had three virtual node members, so the virtual node I/MC1000 is added, sending a one bit message in this group (Table 6). There is one virtual node per message period that a physical node sends. For example, as Table 1 shows, the Battery Node sends messages at 50 ms, 100 ms, and 1000 ms. This results in three virtual nodes - Battery50, Battery100, and Battery1000 - shown in Table 4,Table 5 and Table 6.

This strategy allows us to relax the criterion that nodes must send once per round as Section 3.3 discussed. Instead, a virtual node need only send once during its virtual group's period. The virtual group will then be guaranteed to reach consensus within two times its associated period (not twice the round length). This strategy will tolerate one fault in two times the period of the virtual group. The round length will remain the same, and virtual nodes will send at most once per round, and at least once per period. Note that the period of the virtual group is always greater than or equal to the round length. One might expect this to negatively impact availability, but this is not the case as Section 5 will show.

A main benefit of this strategy is increased tolerance for transient faults, namely corrupted frames due to noise on the communication bus. Each virtual group keeps its own membership; therefore if a frame is corrupted only the virtual node will lose membership, not the physical node. This means that a physical node that sends messages at different periods will still be able to send some of its messages if one of its frames gets corrupted. For example, assume that a 100 ms frame from the physical Transmission node is corrupted by noise on the bus. This frame corresponds to virtual node Trans100 in VirtualGroup4. Assuming all virtual nodes in VirtualGroup4 detect a corrupted frame, virtual node Trans100 will lose membership and may not be able to send messages for the next 200ms (twice the period). However virtual node Trans5 (which is likely to be more critical, as it has a shorter period) is unaffected. In addition to providing increased availability, this group strategy provides some protection against critical messages being prevented from sending by non-critical message failures.

Table 1. Nodes and Sending Message Periods
From data in Tindell and Burns (1994)

Node / Period	5ms	10ms	20ms	50ms	100ms	1000ms
Battery				8	32	17
BrakesOne	16		1		8	
BrakesTwo	16		1		8	
BrakesThree	16		1		8	
BrakesFour	16		1		8	
Driver	8			13		2
I/M C	16			14		
Trans	8				8	
V/C	16	16		25		3

Note that neither a virtual group alone, nor virtual nodes alone would solve this problem. If virtual groups were created involving physical nodes, a single faulty frame from a physical node would affect all virtual groups. The physical node would lose membership in all virtual groups; thus reintegration time would not be improved. In fact, reintegration would take longer, because some of the virtual groups have periods longer than a round. If only virtual nodes were created and a single group was used, the bandwidth cost would be prohibitive as nodes must send exactly once per round to guarantee consensus occurs in two rounds.

4.2. *Other Possible Sources of Faults*

Using multiple groups provides a fairly robust way to identify transient faults. It would be unlikely for a fault other than a transient bus error to corrupt one of the messages a node sends and not the others. Since the group membership service depends on the CRC included with the frame for error detection, a node that has sent an invalid value will be deemed correct as long as all receiving nodes correctly receive that value. A permanent fail silent processor fault would affect all messages. An outgoing link failure on a node would affect all messages. An incoming link failure would affect all messages. A faulty clock would likely affect

Table 2. VirtualGroup1

VirtualGroup1 (5 ms period)	
Virtual Node Name	Total Payload Bits
BrakesOne5	16
BrakesTwo5	16
BrakesThree5	16
BrakesFour5	16
Driver5	8
I/MC5	16
Trans5	8
V/C5	*24

Table 3. VirtualGroup2

VirtualGroup2 (20 ms period)	
Virtual Node Name	Total Payload Bits
BrakesOne20	1
BrakesTwo20	1
BrakesThree20	1
BrakesFour20	1

* This includes two 8-bit 10ms messages, since the V/C was the only node to send at 10 ms and it cannot be in a group by itself

Table 4. VirtualGroup3

VirtualGroup3 (50 ms period)	
Virtual Node Name	Total Payload Bits
Battery50	8
Driver50	13
I/MC50	14
V/C50	25

Table 5. VirtualGroup4

VirtualGroup4 (100 ms period)	
Virtual Node Name	Total Payload Bits
Battery100	24
BrakesOne100	8
BrakesTwo100	8
BrakesThree100	8
BrakesFour100	8
Trans100	8

Table 6. VirtualGroup5

VirtualGroup5 (1000 ms period)	
Virtual Node Name	Total Payload Bits
Battery1000	17
Driver1000	2
I/MC1000**	1
V/C1000	3

** This virtual node was added so this virtual group would have 4 members

all messages, although additional investigation is necessary for this topic.

It is important to note that our approach only takes over the reintegration service portion that a group membership facility provides. Specifically, a clock synchronization algorithm may need to consider the system as having a single group, since the 'correctness' of a frame mandates that the sender's view of members in the group matches the receiver's (TTP/C, 2002). Since each virtual node can be mapped back to a physical node, a clock synchronization algorithm will be able to determine which nodes are functional and which are not from the virtual group information.

5. PERFORMANCE ANALYSIS

This section delves into a detailed analysis of the availability provided and bandwidth required by the single group approach and the multiple group approach. First, frame overhead and payload sizes are determined. Next, the probability of all four Brakes nodes being unavailable at the same time with both membership approaches is computed. Finally, the bandwidth required by each approach is discussed.

5.1. *Frame Overhead*

In order to estimate bandwidth and availability, one needs to determine how many bits will be sent in a frame. In addition to the data payload, each frame includes some overhead fields. The size of the overhead fields was determined according to version 1.0 of the TTP/C specification. For these estimates, the smallest size possible was used, in order to compare the new strategy to the best performance possible under the existing single group strategy.

Table 7. Estimated Frame Overhead

Field Name	Estimated Bits
CRC	24
Frame Type Identifier	2
Mode Change Request	1

Table 7 lists the additional fields (besides the data payload) sent with each frame in the TTP/C system (TTP/C, 2002). There is also a Schedule ID field calculated into the CRC but not sent explicitly. Note that a length field is not required for TDMA protocols, since this information may be placed in a Message Descriptor List (MEDL) deployed before startup on all nodes. The TTP/C Specification mandates a CRC with a minimum Hamming distance of 6 (TTP/C, 2002, p. 44). The maximum allowable data length (plus implicit C-state) depends on the CRC length (TTP/C, 2002, p. 24). A 24-bit CRC will adequately protect the maximum allowed data payload of 240 bytes. For the Frame Type Identifier, there are at least three types of frames: cold start, implicit C-state, and explicit C-state (TTP/C, 2002, p. 39-40). Therefore, at least two bits are needed to represent the Frame Type Identifier. For the Mode Change Request field, 'Each MEDL contains at least two modes, the startup mode and one application mode', so at minimum this field is one bit (TTP/C, 2002, p. 11). This gives a total of 27 bits of overhead per frame.

If an explicit membership vector is used, this will also incur overhead. The membership vector contains one entry for every node in the sending node's group, usually a true/false flag. In standard group membership, there will be one bit per node in the system. With virtual groups, a virtual node will send one bit for every node in its virtual group. A membership vector for a virtual group will always be equal to or smaller in size than a membership vector that treats the system as a single group, since each real node will have at most one virtual node in each virtual group. So the size of the single group membership vector is an upper bound for the size of a virtual group membership vector.

Membership vectors can also be sent implicitly (except during startup), consuming no bandwidth. Implicit membership vectors are assumed; therefore, the bandwidth calculations do not include overhead for the vectors. Explicit membership vectors are required for reintegration, however. For this analysis, it is assumed that explicit C-state frames are sent at regular intervals, but do not occur during the two rounds in question. Recall that a node that loses membership may not be able to reintegrate in the round following a fault, due to clique avoidance. Therefore, reintegration is not a factor in the performance analysis.

5.2. *Payload Size*

In order to determine the data payload size, recall that for single group membership a node will send exactly once per round. Table 1 lists the number of data bits a physical node has to send for each period. Because of the send-once-per-round rule, a node will need to combine payloads into one frame, or the round will need to be shorter than the minimum period. Our example assumes the payloads will be combined into a single frame, since shortening the round would result in more overhead bits total. Looking at Table 1, the shortest message period in this system is 5 ms, so the round length will be 5 ms.

The worst-case payload size for a Brakes node can be computed from the information in Table 1. For the single group strategy, a Brakes node will send out an aggregate frame exactly once per round, or every 5 ms. Each Brakes node (BrakesOne, BrakesTwo, BrakesThree, BrakesFour) has 16 bits of data at a 5 ms period. Then, each Brakes node also must send 1 bit of data every 20 ms and 8 bits of data every 1000 ms. Careful design can avoid having the 20 ms payload and the 1000 ms payload in the same aggregate frame. Therefore the worst case payload size for a Brakes node using the single group strategy is 24 bits. For our multiple group strategy, the payloads will contain data only for other messages sent at the same period. Therefore the 5 ms message will have a 16 bit payload, the 20 ms message will have a 1 bit payload, and the 100 ms message will have an 8 bit payload.

5.3. *Chance of Losing All Brake Nodes*

This section explores the probability that all four Brakes nodes lose membership due to corrupted frames, given as the probability per hour. Our fault model is random, independent noise on the bus. This probability provides a conservative estimate of the chance that the brakes will be unavailable due to a transient bus error. For the multiple node strategy, it is possible that only some of the virtual nodes lose membership. Additionally, the probability of losing a particular message from the brakes is different from the probability of losing all messages from the brakes. Cases where a node loses membership due to other types of errors are not considered (for example, a node that is receive-faulty). Since the fault model is independent noise, a conditional probability analysis would produce the same results because the errors are uncorrelated. A more inclusive fault model is an avenue for future work.

Single Group Strategy

One needs to know the worst-case size possible for two frames in a row, since reintegration can take two rounds. From the Payload section, the worst-case payload size for a Brakes node with the single group strategy is 24 bits. Therefore the worst case for the first frame is a size of 24 data bits plus 27 overhead bits for a total of 51 bits. For the next frame, the worst case occurs when the 20 ms data and the 5 ms data are sent. Adding overhead gives a frame size of 16 + 1 + 27 = 44 bits. Since the 20 ms data is sent along with every fourth 5 ms frame, and the 1000 ms data is not sent with the 20 ms data, the 20 ms data will fall either in the round after or before the 1000 ms data. The order is immaterial to the reliability equations. Therefore the largest number of bits sent by a Brakes node in 10 ms is 95 bits.

Our Multiple Group Strategy

For the virtual group strategy, recall that the consensus time will be equal to the period of that group. The Brakes nodes send messages in three different virtual groups - the 5 ms group, the 20 ms group, and the 100 ms group, as can be seen in Tables 2-6. An upper bound on the probability of losing all frames from all

Table 8. Probability Per Hour of Losing Brake Nodes

BER	**Single group, any/all message(s)**	*Multiple group, any 5 ms message*	*Multiple group, any 20 ms message*	*Multiple group, any 100 ms message*	**Multiple group, all messages**
10^{-4}	2.39 E-11 /hour	5.76 E-18 /hour	9.67 E-19 /hour	2.99 E-17 /hour	9.73 E-38 /hour
10^{-5}	2.39 E-19 /hour	5.76 E-26 /hour	9.67 E-27 /hour	2.99 E-25 /hour	9.73 E-62 /hour
10^{-6}	2.39 E-27 /hour	5.76 E-34 /hour	9.67 E-35 /hour	2.99 E-33 /hour	9.73 E-86 /hour
10^{-7}	2.39 E-35 /hour	5.76 E-42 /hour	9.67 E-43 /hour	2.99 E-41 /hour	9.73 E-110 /hour

Brakes nodes can be found by multiplying the probabilities of losing all 100 ms frames within twice that period (200 ms), all 20 ms frames within 40 ms, and all 5 ms frames within 10 ms. Note that each frame is sent twice, since there is a dual-redundant bus. Therefore a total of eight frames must be lost. Table 8 gives the probability per hour that a group will lose one type of message from all of the Brakes nodes at the same time. Equation 1 gives the formula for the first four columns:

$$\left(BER * \frac{\# bits}{frame} * \frac{\# frames}{two\_ period}\right)^8 * \frac{two\_ period}{hour} \qquad (1)$$

The first portion of the equation represents the chance that one frame will be corrupted in a window of two periods. This is raised to the power of 8, since 8 frames (two per node) must be corrupted within this window of time. Then, the second portion of the equation shows how many windows occur in an hour. For example, the first entry in the 'Multiple Group Any 5 ms Message' column was calculated as:

$$(.0001 * (16+27) * 2)^8 * \frac{1}{10ms} * \frac{3600000ms}{hour} \qquad (2)$$

The BER is 0.0001. There are 16 payload bits (from Table 1) plus 27 overhead bits (from Section 5.1) in a frame. One frame is sent per 5 ms period, giving two messages per two periods. This probability of a corrupted frame (i.e., losing all messages from the physical node in a frame) is raised to the power of 8. Then, this is multiplied by how many 10 ms rounds occur per hour.

For a single group strategy, the probability of losing a particular type of message from all nodes is the same as the probability of losing all messages from all nodes. A transient error in both redundant frames a node sends will cause that node to lose membership and it will not be able to send any message in its next slot, and possibly its next two slots. However, for the multiple group strategy, it is possible that only some of the virtual nodes lose membership and not others. Therefore some of the messages may be entirely lost without affecting other messages. The final column gives the probability that all messages will be lost for the multiple group strategy, per hour. This probability is obtained by multiplying the previous three columns together. This is a pessimistic estimate, as this is the probability that all three virtual groups will fail in an hour, not necessarily at the same time.

Observations

For this data set, all of these approaches seem reasonable for automotive applications. Automotive protocols are typically designed to have lower than a 10^{-9} probability of failure per hour (PALBUS, 2001) for a network with a BER of 10^{-5} to 10^{-6}. The estimates for all four Brakes nodes losing membership do not violate that criterion; however, designs with heavier workloads might. The results in Table 8 are not failure rate calculations and do not account for nodes that lose membership for any other reason besides a transient bus error while the node is sending a frame. These results should not be used as a sole estimate of any failure rate, but rather are intended to illustrate the possible benefits of a virtual node approach.

The multiple group strategy has a lower chance of losing a single particular message from all four Brakes nodes. This is due to the fact that the frame sent is smaller because the payload is smaller. In the multiple group strategy, less data needs to be sent in the payload, since the virtual nodes may send only once per their virtual group period instead of once per round.

The multiple group strategy has a far lower chance of losing all messages from all four Brakes nodes. Since there are virtual Brakes nodes in three virtual groups, the chance of all virtual nodes losing membership is roughly equal to the chance of the single group strategy raised to the power of three. In general, a node with an X chance of losing membership in a single group will have a X^n chance of losing membership in all n of its virtual groups. So in Table 8, the last column is roughly equal to the first column cubed. This is a significant benefit for safety-critical systems.

It is important to note that some forms of redundant components, such as shadow or backup nodes, will not improve availability in this situation that involves a transient fault. A node is not allowed to integrate into a group until the group has reached consensus on its members. So the shadow node will be prohibited from sending frames if the original node would have been prohibited from sending frames.

5.4. *Bandwidth*

This availability gain has low bandwidth cost. Determining the bandwidth needed to send frames requires computing the slot size for each node. According to the TTP/C specification, each node is assigned one slot in a round (TTP/C, 2002). The message size may vary; however, the slot size remains constant. Therefore a node's slot size must be at least as large as the largest frame the node has to send. (The actual slot size is slightly larger, but that is true for both strategies, and only serves to make this bandwidth comparison conservative.) The slot size will determine the bandwidth required – even if a portion of

Table 9. Slot Size Required Per Node (Bits)

	Battery	Brakes(4)	Driver	I/M C	Trans	V/C
Single	35	204	39	51	43	59
Multiple	35	312	66	78	70	113

the slot goes unused, other nodes are prohibited from sending until the end of the slot.

For the single-group strategy, the payloads are aggregated into a single frame because a node may only send once per round. One can do slightly better than simply adding up all payload bits given in Table 1 for a node by using the complete data listing from Tindell and Burns (1994). Since the system is statically scheduled, and some of the long period messages do not send every five milliseconds, more conservative figures can be given for the slot size. Table 9 summarizes the slot size required by each node for the single and multiple group strategies. These figures include overhead. There will be 200 of each slot in one second (since the round is 5 ms long), giving a total required bandwidth of 86,200 bits/second.

For a multiple group strategy, there is a tradeoff between error detection ability and bandwidth. If data payloads from the same physical node but different virtual nodes are sent in the same message with a single CRC, and the message is corrupted, all participating virtual nodes will lose membership in their virtual groups. For example, if the actual Transmission node sends its 5 ms and 100 ms payloads using only one CRC, both virtual nodes Trans5 and Trans100 will lose membership if the message is corrupted.

We choose to preserve the error detection ability by assuming that a physical node will send back-to-back separate, complete messages per time slot for each virtual node that needs to send. This is slightly different than the TTP/C specification approach for sending multiple messages per slot where overhead is not duplicated (TTP/C, 2002). This also represents maximum bandwidth consumption, so if the bandwidth required is too great, further points in the tradeoff space can be explored. Another option is to have some of the virtual nodes use a shared slot, since virtual nodes are not required to send a frame once per round. Both approaches incur the same bandwidth, since the same data is being sent – it is just a matter of whether a single slot is reserved, or multiple slots are reserved. It is acceptable to combine payloads for messages with the same period. These calculations assume all overhead is duplicated; it may be possible to combine some overhead fields.

The maximum slot sizes required for the multiple group virtual node strategy are listed in Table 9. There will be 200 of each slot in a second, giving a total required bandwidth of 134,800 bits/second. This represents a 1.56 times bandwidth cost compared to single group, in exchange for a dramatically reduced probability of losing group membership for the most critical messages.

6. CONCLUSION

A multiple-group, period-based group membership strategy provides an attractive way to tolerate transient errors due to bus noise in TDMA protocols. Algorithms for single group membership can be applied to virtual groups with virtual nodes without changing the existing theoretical framework. By creating a virtual group composed of virtual nodes for each period in the system, the virtual nodes are isolated from each other. Assigning a virtual node per each message period of a physical node allows the system to tolerate a corrupted message from the physical node without affecting some of the other messages the node sends.

Multiple-group, period-based group membership provides increased availability at an affordable bandwidth cost. For N virtual groups, the unavailability can be reduced by as much as a power of N over a single group strategy. The bandwidth increase is not prohibitive (1.56 times the single group bandwidth for the system studied), as the requirement that nodes send once per round can be relaxed to once per period for the virtual nodes.

ACKNOWLEDGMENTS

This work is supported in part by the General Motors Collaborative Research Laboratory at Carnegie Mellon University and by the United States Department of Defense (NDSEG/ONR).

REFERENCES

Bauer, G. and M. Paulitsch (2000). An Investigation of Membership and Clique Avoidance in TTP/C, *Proceedings of 19th IEEE Symposium on Reliable Distributed Systems SRDS-2000,*, pp. 118-24.

Bouajjani, A. and A. Merceron (2002). Parametric Verification of a Group Membership Algorithm, *Proceedings of the 7th International Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems.*

Controller Area Network (CAN) Specification v. 2.0 (1991). Robert Bosch GmbH, Stuttgart.

Kim, K.H. and E. Shokri (1993). Minimal-Delay Decentralized Maintenance of Processor-Group Membership in TDMA-Bus LAN Systems, *Proceedings of the IEEE International Conference on Distributed Computing Systems - ICDCS '93*, pp. 410-19.

Analysis and Test of Bus Systems, PALBUS Task 10.2, 10.3 (2001), SP Swedish National Testing and Research Institute.

Pfeifer, H. (2000). Formal Verification of the TTP Group Membership Algorithm, *Proceedings of FORTE XIII/PSTV XX,* pp. 3-18.

Time-Triggered Protocol TTP/C High-Level Specification Document (2002), TTTech Computertechnik AG, Schoenbrunner Strasse 7, A-1040 Vienna, Austria.

Tindell, K. and A. Burns (1994). Guaranteeing Message Latencies on Control Area Network (CAN), *Proceedings of the 1st International CAN Conference.*