12AE-0196

Enabling Exponential Growth of Automotive Network Devices while Reducing the Wired Communication Infrastructure with Security, Reliability, and Safety

Author, co-author
(Do NOT enter this information. It will be pulled from participant tab in MyTechZone)
Affiliation (Do NOT enter this information. It will be pulled from participant tab in MyTechZone)

ABSTRACT

The CAN protocol has served the automotive and related industries well for over twenty-five (25) years now; with the original CAN protocol officially released in 1986 followed by the release of CAN 2.0 in 1991. Since then many variants and improvements in CAN combined with the proliferation of automotive onboard microprocessor based sensors and controllers have resulted in CAN establishing itself as the dominant network architecture for automotive onboard communication in layers one (1) and two (2). Going forward however, the almost exponential growth of automotive onboard computing and the associated devices necessary for supporting said growth will unfortunately necessitate an equivalent growth in the already crowded wired physical infrastructure unless a suitable wireless alternative can be provided.

While a wireless implementation of CAN has been produced[1], it has never obtained real traction within the automotive world. Other alternative methodologies for providing wireless connectivity have been much more pervasive and accepted, but none of them provide anything more to CAN interfaces than a CAN-to-Wireless Bridge; with Wi-Fi, Blue Tooth, and GSM being the primary wireless network architectures bridging to CAN.

What is proposed within this paper is more than simply a wireless extension of CAN in that it does more than extend CAN into the wireless domain (as was the case with CANRF). As pure wireless CAN with no accommodations for heavy utilization would only exacerbate CAN's primary deficiency of starving out lower priority messages; since there would be no way to isolate devices in sub-networks as could be done with a wired infrastructure.

Rather, the proposal within this paper would attack this deficiency head-on by modifying the newly defined wireless network protocol and architecture, DQWA (Distributed Queuing Wireless Arbiter) to not only extend CAN into the wireless domain, but also addresses CAN's more prominent shortcomings.

DQWA is a solution that provides both security and reliability within a wireless framework, while maintaining CAN's distributed network communication methodology and implicit avoidance of single points of failure within the network.

PREFACE

Before discussing the merits of this extension it is important to understand the basic technology behind it. What that means is in order to understand the extension both the intricacies and inadequacies of CAN itself must be clearly delineated along with that of the core architecture behind the extension; namely, DQSA (Distributed Queuing Switch Architecture).

[1] The CANRF module produced by Automation Artisans, Inc. (see http://www.canrf.com).

DQSA was originally designed as a Layers One (1) and Two (2) broadcast network architecture for cable TV networks that provided deterministic access to the transmission queue while simultaneously limiting collisions to a finite window within the DQ Transmission Frame. Subsequently, DQSA has been extended by Luis Alonso, PhD and Jesus Zárate, both of the University Polytechnic of Catalonia (UPC), into the wireless arena; focusing mostly on the Link Layer (i.e. Layer Two (2)) with only minimal direction regarding the Physical Layer (i.e. layer two (2)). Subsequently, the company Ether2 has further extended the wireless nature of DQ with the newly defined, Distributed Queuing Wireless Arbiter (DQWA), with most of the specification dealing with the Link Layer while also providing only minimal direction for the Physical Layer.

DQWA is a hybrid of a traditional "hub and spoke" network architecture with that of a peer-to-peer MESH network architecture. The primary area of focus of the DQWA specification is that of the Link Layer, although a key and critical aspect of it successful implementation, the Contention Window and associated Min-Slots, is heavily dependent upon the Physical Layer in that successful implementation of a unique Collision Detection mechanism (see [1] and [2]) must be implemented.

Finally, the emphasis of the paper to enable the reader's preliminary understanding of the subject technology's general applicability and extensibility into the automotive world. The specific details within this paper focus on DQWA's encryption capabilities along with its network integrity architecture as the primary means of providing security within a DQ Service Set[2] (DQSS).

1 INTRODUCTION

1.1 Why Not CAN?

Tradition CAN utilizes a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) technique similar to that of Ethernet but with frames that are relatively small by networking standards in that the largest possible frame can be no more than 128-bits (i.e. 16-Bytes, including the maximum of 8-bytes for the payload), whereas the smallest Ethernet Frame is 64-bytes, going all the way to 1,536-bytes for the largest Ethernet Frame. Unlike Ethernet however, there is no loss of data as a result of collisions. This is because of CAN's unique non-destructive message arbitration methodology that guarantees high-priority messages access to the CAN bus with no fear of collision or loss of data; hence, no need for retransmission.

However, the same feature that is CAN's strength (its non-destructive collision resolution methodology) is also its weakness in that as a CAN bus approaches its utilization capacity so does its propensity for indefinite starvation of lower priority messages. Given that a CAN message cannot arbitrarily change its priority; the CAN protocol is completely inflexible under heavy loads for successfully ensuring that lower-priority messages reach their destination.

The traditional methodology for resolving this problem has been in the separation of CAN nodes within a particularly vehicle into multiple CAN sub-networks. However, such delineation can often be the source of frustration when attempting to discern the most efficient means for dividing the devices into disparate CAN networks while still affording cross network communication through various backhaul communication technologies.

Given the proliferation of network devices into our daily lives[3], it is only logical to deduce a similar growth pattern within vehicles. As that growth pattern continues, it will become increasingly difficult to depend so heavily on a wired infrastructure for providing communication connectivity within the vehicle. Of greater significance will be the proliferation of automotive onboard devices that will be expected to communicate externally; particularly with respect to both personal data derived from the human passengers as well as vehicular data exchanged with vehicular traffic management technology both fixed and potentially with other vehicles.

It is clear for many reasons, both because of the physical limitations, difficulty, and expense of installing and maintaining wired bus infrastructures that the necessity of a wireless alternative is inevitable.

[2] The DQ Service Set consists of all the nodes with a given DQ Distributed Network.

[3] The number of devices is expected to total over 1-Trillion (or 140-devices per person on earth) by 2013 [4].

1.2 What challenges must be overcome for CAN to be successful deployed within a Wireless Environment?

As mentioned above, the primary weakness in attempting to utilize CAN within a heavily utilized bus is the propensity for lower priority messages to be starved out and hence never sent; or sent too late to be of any use. Obviously, if CAN is to be deployed within a wireless environment then this weakness becomes a severe problem given that it will become difficult for CAN nodes to form a sub-network within the same vehicle; not to mention potential interference from external sources, including CAN nodes broadcasting on the same frequency in nearby vehicles.

Even if adequate RF shielding and filtering techniques are utilized within the vehicle chassis to maintain successful RF communication; given the limited number of available frequencies, a methodology would still need to be employed that would facilitate coexistence with other nodes broadcasting on the same frequency within the vehicle; particularly with respect to access to the bus' transmission queue. Also, given the real-time, mission and safety critical nature of automotive communication, reliability and robustness must be key considerations in any deployed networking methodology supporting automotive communication.

Last, given that by definition wireless communication is ubiquitously broadcast, security becomes a crucial concern. Examples of such concern consists both of those from listening in violating both privacy and network security as well as those attempting to gain unwanted access over the network devices within the network (ex. either by either directly manipulation of the devices or by indirect manipulation via the spoofing of existing devices within the network).

1.3 Why Wireless and Why DQWA?

1.3.1 Why Wireless?

Recognizing the proliferation of devices with network connectivity within vehicles is only going to continue escalating; it is only logical to look for a means to facilitate this expansion without an equivalent expansion in wired infrastructure. Anyone who has looked under the hood of a vehicle from the 70's and then compared that to what is under the hood today must wonder where the space for any additional infrastructure is going to come from.

The same is true for under the dashboard and/or in the trunk with respect to entertainment systems. Consumers want more space, not less; they want their technologic advances without paying the price in either comfort or cost. The only foreseeable path to that end is a wireless one.

It is this path that brings fewer wires; lower costs; easier installation; greater capabilities for expansion.

1.3.2 Why DQWA?

Like CAN, DQWA is essentially a distributed architecture with respect to communication. However, for control, DQWA is static for a given point in time; specifically, it is static for the duration of a DQ Transmission Frame. The designated central control point may transition to other nodes upon completion of the current DQ Transition Frame; which is why DQWA can be viewed as a hybrid between a pure MESH ad-hoc architecture and that of a traditional Hub-and-Spoke architecture.

The hybrid nature of the DQWA network architecture provides flexibility for adaptation to a CAN Wireless Extension in that communication is distributed while enabling a central authority to elevate priorities of messages as needed providing a QoS aspect to DQWA that CAN severely lacks. Also, because the central authority may shift from DQ node to DQ node if desired (i.e. enabled to do so), traffic patterns are automatically localized with respect to control. Thus, reducing latency when and where needed[4]; according to the traffic pattern.

Last, because all communication can be encrypted at the MAC layer, including the headers; security is maintained at all times in spite of the fact that all traffic is broadcast wirelessly.

[4] NOTE: The expectation for reduced latency is implied from the fact that if the Cluster Head moves to the center of the traffic pattern, the distance (and therefore the time) is reduced between when the Cluster Head receives the frame and responds with a feedback frame. This is particularly important in a noisy RF environment; given that the further away the signal is the more prone it is to be interfered with.

2 DQWA for CAN

Distributed Queuing Wireless Arbiter (DQWA) is ideal for applications requiring distributed communication and control, of which the automotive world certainly falls into that category. In short, DQWA adds the ability to simplify intra-vehicle connectivity while expanding overall communication capabilities. Hopefully, the reader will walk away with this as their resultant impression.

2.1 Background

The DQWA Protocol is based on the Distributed Queue Switch Architecture (DQSA) developed at the Illinois Institute of Technology. The heart of this technology is a medium access control (MAC) that allows an arbitrary number of stations to share a common communications channel over any distance and operating at any data rate. DQSA can operate over virtually any topology and will also provide a Quality of Service (QoS) at the MAC layer that includes the ability to temporarily elevate priorities in order to prevent starvation (as can occur in traditional CAN).

The key feature of DQSA is that all control resides in the stations, no central control is required. The network state is maintained at all times by each station in just two (2) binary counters per DQSS, providing it with all the information necessary to make decisions as to when to transmit for that specific DQSS. A DQ Transmission Frame is divided into three separate time periods or segments; with the three segments listed below:

1) Contention Window, utilized as part of the Access Request Sequence (ARS) to the Transmission Queue with three (3) control mini-slots acting as a finite sized Contention Queue;
2) Data and Control Window consisting of a single DQ Data and Control Frame; and,
3) Feedback Window, consisting of the DQ Feedback Frame with Synchronization Beacon,.

A synchronization beacon must be transmitted to all stations prior to the start of each segment from which all stations must synchronize with for every transmission frame so that they may participate in the DQSS. The DQ Feedback Frame and associated Synchronization Beacon can come from any node within the DQSS, but is always sent by a single node at any given time and from which the node is typically chosen as one of a set of nodes designated for accessing gateways beyond the DQSS. Within a wireless environment, this central point would normally be referred to as the Base Station, Access Point, or Hub; the DQWA nomenclature for this central authority is the Cluster Head.

Variable length DQ Messages may be segmented into multiple data slots without requiring any further overhead. Qualities of Service (QoS) Priorities are available and it is possible for a higher priority DQ Data & Control Frame to preempt a lower priority DQ Data & Control Frame during transmission within a period of one DQ Message[5].

Segments can be allocated to a specific station thus providing time-division-multiplex (TDM) channels, commingled with normal DQ Frame traffic. The overall utilization within a wireless environment, i.e., ratio of data content to the channel capacity ranges from over 95% down to 80%;, depending upon frame size[6] and overall network utilization. Given CAN's relatively small payload size (no more than 8-bytes), customization of DQWA for automotive communication would need to done in order to achieve similar efficiency for reduced size payloads.

Again, as mentioned in Section 1.3 above, because access to communication within a DQSS consists solely of member nodes, the entire contents within a MAC layer frame, including the header, may be encrypted; thus ensuring the both security and privacy.

2.2 The Contention Window (CW)

The purpose of the CW's ARS is twofold:

1. To afford current members of the DQSS with an opportunity to request communication privileges with one or more of the other nodes (including the Cluster Head) within the network.

[5] A DQ Message can span one or more DQ Transmission Frames.

[6] These numbers were calculated assuming constant data rates, preamble length of 96-bits, various interspacing fields of 8-bits, 16-bits, and 24-bits respectfully, as well as minimum sized payloads varying between 256-bytes and 4,096-bytes.

2. To simultaneously mitigate the potential for MAC & Data Payload collisions and hence, dropped frames resulting from corruption.

The latter is achieved by limiting the contention for access to the channel to a finite and predictable period of time. With the exception of the Cluster Head, all nodes must utilize this mechanism in order to access the MAC & Data Payload segment of the DQ Transmission Sequence.

2.2.1 ARS Mechanics

The ARS Segment is divided into three (3) sub-parts, termed, Mini-Slots (MS) (see below).

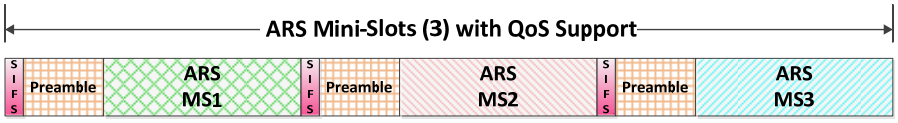

Figure 1 – DQ Access Request Sequence Segment Structure with QoS Support

This number was initially chosen based upon research [1] (i.e. (Xu & Campbell, 1992)) showing that the collision resolution process can be made to work faster than the data transmission process when the number of MS is restricted to three (3). Increasing the number of MS beyond three (3) may introduce additional delay as well as adding increased overhead to the overall protocol resulting from the added delay.

The collision resolution process referenced above utilizes unique patterns transmitted by each soliciting device and a summation of those patterns in the event of a collision as a means for detecting collisions.

The operation of DQWA is based on the *m*-ternary feedback information on the state of each of the mini-slots. The Cluster Head must be able to distinguish between the three states:

- Idle,
- Success,
- and, Collision,

for each mini-slot; as this information is crucial for the application of the protocol rules at the end of each frame. Adopting a patented technology [2] (i.e. (Campbell & Xu, 2001)) each node is assigned a unique bit pattern that has the property that when two or more ARS collide, the pattern of the overlapping signal is distinguishable from the original pattern of any single ARS; hence, the Cluster Head can detect the collision.

The preferred example patterns referenced in the paper are binomial coefficients; however, this number can be modified to meet the requirements of the targeted environment

Each node accepted into the network is assigned both a Node Address and a constant size Code Word of constant Hamming Weight (see below).

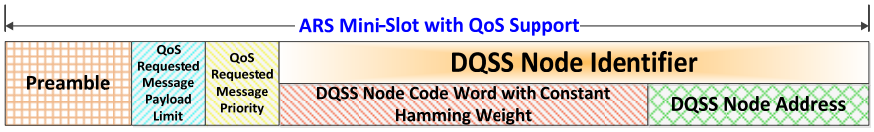

Figure 2 – Expanded (QoS Enabled) ARS Mini-Slot (MS) Structure

When a collision does occur, and particularly within an RF environment, it is a relatively straightforward process to determine a collision has occurred since the collision will more than likely make the interpretation of the combined signal unintelligible. Further, even if the resultant collided signal does result in an intelligible result, the resulting Hamming Weight will almost certainly be something other than the selected constant value. When taking into account that the correct associated DQSS node address must accompany the code word of constant hamming weight, the detection of a collision is all but guaranteed.

DQWA has an additional validation mechanism within the DQ Feedback Frame that protects against the unlikely occurrence of an illegitimate, but valid Code Word and DQSS Node Address combination resulting from a collision; however, that discussion would go well beyond the scope of this paper.

The aforementioned ternary decision described above can be subsequently determined as follows:

- Idle (i.e. no signal in ARS Mini-Slot) – Received Signal is below the RSSI (Noise) Threshold.
- Success – A demodulation resulting in the correct hamming weight and correct code word value and node address combination and again validated within the DQ Feedback Frame.
- Collision – Any signal detected above the noise (RSSI) threshold not resulting in a translation into the digital domain of a code word with the correct hamming weight and correct code word value and node address combination.

The Cluster Head will respond with the collision results as part of the DQSS Management Segment in order to clarify any potential ambiguities.

2.2.2 DQSS Node Addressing within the ARS and applicability to CAN Message Addresses

Standard DQSS Network addresses are 12-bits in length, however, only the lower 10-bits are assignable for the dynamic portion of a valid address; as the upper two bits have special meaning.

Both bits along with the rest of the DQSS Network Address are provided below:

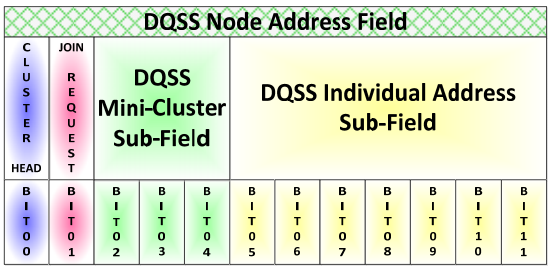

Figure 3 – DQ Node Network Address Field

That said, there is nothing special or specific about the 10-bits selected for the actual node address; thus, this address could be readily modified to an 11-bit CAN address.

While it is true that CAN utilizes a message based architecture, which is in sharp contrast to most network architectures (of which DQWA falls into the same category), a DQWA implementation for CAN would facilitate this feature by viewing standard CAN message addresses as DQWA Multicast Addresses; while still extending CAN so that individual nodes addressing is still valid (although potentially not necessary in the short term).

2.2.2.1 DQSS Node Cluster Bit

NOTE: This bit is NOT used within the ARS and should ALWAYS be zero during the ARS; it is explained here since this bit is part of the DQSS Node Address Field.

The MSB of the address is reserved for the Cluster Head. This is particularly helpful if the Network Topology moves and the Cluster Head moves with it. Thus, allowing any node to maintain its original identity both before and after assuming the duties of the Cluster Head. In this way, the DQSS table maintains consistency regardless of which node is currently in charge of the network.

2.2.2.2 DQSS Node Join Request Bit

The next most significant bit (bit 1) is used by nodes wishing to join the network. In order for an unknown node to be considered for admittance to the DQSS, it must satisfy the below two conditions:

1) The “Join Request” Bit shown in Figure 3 must be set within the DQSS Node Address Field. It is this bit (i.e. the Join Request Bit) that allows for parts to be installed within a particular vehicle network architecture with little to any actual configuration in that “newly” installed parts can automatically request for inclusion in the desired vehicle’s DQSS.

2) The “DQSS Mini-Cluster” Sub-Field must set ‘7’ (i.e. “111b”).

The “DQSS Individual Address” Sub-Field may be a value between ‘0’ and “127” (i.e. a span of 128-values).

2.2.2.3 DQSS Mini-Cluster Sub-Field

This is an important field in that it explicitly affords specific portions of a DQSS to be segmented into individual address spaces for the purpose of multi-cast addressing as well as enabling CAN sub-networks within a specific DQSS. The addition of a Message Bit to the DQSS Node Address Field (as alluded to in the previous section) would enable further enforcement of messages being restricted to specific CAN sub-networks.

2.2.2.4 DQSS Individual Address Sub-Field

These seven bits are used for assigning individual addresses, with any value between ‘0’ and “126” assignable for an individual DQSS Network Address. The only time “127” may be used during the ARS is during a “Join Request.” As “127” is otherwise set aside for “Directed Broadcasts” and regular “Broadcasts” for all Mini-Cluster Sub-Field values except for ‘7’ (i.e. “111b”).

2.3 The DQ Service Set and its relationship to ensuring Security

A key component of the DQ Service Set concept is network security and the rules by which nodes may become members of a specific DQ Service Set. A DQSS can operate in one of three operational modes listed below the operational modes listed in decreasing order of centralized membership control:

- Static Association Mode;
- Semi-Manual Association Mode;
- Promiscuous Mode;

Each of the modes will now be individually discussed in detail.

2.3.1 DQSS Static Association Mode

In Static Association Mode, the DQSS is completely pre-configured. New nodes may not request to join and can only become part of the DQSS either by directly adding nodes to an existing DQSS Configuration Database or by installing a completely new DQSS Configuration Database containing the desired nodes.

In response to the fact that a DQSS configured in Static Association Mode cannot add nodes in real time (doing so only through configuration); any attempt to submit a DQSS Membership Request Code Word during the ARS segment will be ignored.

2.3.2 DQSS Semi-Manual Association Mode

A DQSS configured to be in Semi-Manual Mode has all of the capabilities of a Static Association Mode DQSS as well as the additional ability to add nodes in real time. There are two methods for which a node may acquire inclusion within a DQSS configured in DQ Semi-Manual Association Mode.

The first method for acceptance for a given node into a DQSS while in DQSS Semi-Manual Association Mode is via manual configuration as part of a DQSS Configuration Database. The second method utilizes a two-step process for any node outside of the current DQSS membership and described below:

1) First, the Candidate Node must issue a request for DQSS Inclusion.
2) Second, an external confirmation of the request from either an operator (i.e. service technician or factory installation personnel) or configuration robot utility must explicitly accept the Candidate Node into the DQSS; presumably based upon some criteria established for admission.

It is the latter act that serves as the basis for the moniker, “DQSS Semi-Manual Association Mode” since confirmation of inclusion requires an explicit action from an external source[7].

[7] Presumably via an operator (i.e. service technician or factory installation personnel) or configuration robot utility

2.3.3 DQSS Promiscuous Association Mode

A DQSS configured to be in Promiscuous Association Mode has two methods for DQSS membership inclusion. As with all modes, the first method for inclusion into a DQSS is through configuration.

The second method for inclusion into an existing DQSS is similar to the second inclusion method listed for DQSS Semi-Manual Association Mode; however, no operator intervention is required except for the case of an operator explicitly desiring to exclude a node from the DQSS.

Thus, the only time external[7] intervention occurs during a DQSS operating in Promiscuous Association Mode is when an operator wishes to explicitly "blacklist" a candidate node; adding it to either a permanent blacklist or a blacklist that can be aged out.

An example of a situation in which permanent blacklisting may be desired would be if a paid subscriber for XM Radio or other paid electronic subscription service was delinquent in paying their subscriber fees and/or had exceeded their usage. The subscriber could then be explicitly blacklisted until they brought their account current again and/or purchases additional time.

An example of temporary blacklisting could occur as a result of a background task monitoring network usage. If there was a limit as to the daily network activity for a particular subscriber and that subscriber had exceeded their limit, the Candidate Node of the subscriber could be placed on a blacklist that expired whenever their "lease" renewed again.

While there are certainly other, potentially more cogent examples, each of the above examples sufficiently illustrates the viability of the blacklist exclusion capability.

2.4 DQSS Encryption Modes

Encryption may be used in any mode and can be implemented such that there is little, if any affect, as to how each Association Mode operates. There are two different types of encryption used within DQWA:

- Encrypted Private Key Mode.
- Encrypted Public Key Mode.

Both of these encryption methodologies will now be discussed in relation to their effects on operating modes.

2.4.1 DQSS Encrypted Private (Shared) Key Mode

A DQSS configured to be in Encrypted Private Key Mode utilizes a symmetric encryption methodology with respect to both encrypting outgoing messages and decrypting incoming messages.

Because both sides know what the decryption algorithm is, both sides may transmit the entire message encrypted, including the header.

The clear implication with this mode is that the encryption/decryption algorithms must be done within the PHY in hardware in order for the three operating modes (Static, Semi-Manual, and Promiscuous) to operate oblivious to the effects of encryption performed on the encapsulated data.

2.4.2 DQSS Encrypted Public Key Mode

A DQSS configured to be in Encrypted Public Key Mode utilizes an asymmetric encryption methodology with respect to the encryption of outgoing messages and decrypting incoming messages.

Specifically, the shared (i.e. private) key is used for decrypting messages, but the public key must be utilized for encrypting messages. In this way, the entire message may be encrypted (as is done with Private (Shared) Key Mode), but the public key must be known in order to encrypt an outgoing message.

Thus, nodes wishing to "join" the network, regardless of the configuration must "listen" to the Feedback Packet in order to get the Public Key before they can transmit. The cogent point here is that although the public key is broadcast, it is done so in encrypted form using the "Private" key; thus adding an additional layer of security to this process.

The key advantage to this encryption mode to the automotive industry is that the public key could be provided to all legitimate parts vendors without sacrifice of security. The designated Cluster Head within a specific vehicle could then validate the part as valid or invalid according to the default configuration within the vehicle database. Not only would this serve the purpose of providing security to the vehicle insofar as normal traffic is concerned, it also ensures that only authorized parts may be used for a given vehicle type.

2.5 Dynamic Clustering

DQ supports Dynamic Clustering for the Control Point of DQ Network Topology. If Dynamic Clustering is disabled, the Cluster Head serves as the static control point of the vehicle DQSS network. Thus, if the static DQSS Cluster Head goes down, so does the DQ Network. However, if Dynamic Clustering is enabled, the Dynamic Cluster Head Designation Order will be included within the DQSS and updated separately on a periodic basis.

There are multiple events that may trigger a Cluster Head Transition including traffic loading, hardware and/or power failures, energy consumption fairness criteria, or simply user discretion are a few of the more prominent events. Therefore, in order to support the various types of event triggers, there are multiple selections for the type of Cluster Topology configuration.

The different Cluster Topology configuration types are listed below:

- **Clustering Disabled** – The network is complete static, with one and only one node designated as the central control and arbitration point. Thus, if the Cluster Head fails, then the overall network connectivity also fails.
- **Clustering Enabled for Backup Only** – So long as the network is operating normally, the network is completely static; with a single node designated as the Cluster Head. However, in the event the designated Cluster Head fails, a succession of backup Cluster Heads have been previously identified within the DQSS Table and thus assume the role of the Cluster Head according to their priority order and online status (i.e. the node that is both "online" and has the highest designated priority status becomes the Cluster Head if the current Cluster Head fails; if the highest designated priority status node is not online then the duty falls to the next lower designated priority status node). In the event there are no nodes that are online and have been designated as a backup Cluster Head, the network connectivity fails.
- **Limited Clustering Enabled** – Normal Clustering is enabled for the network with this setting; however, only a limited set of designated nodes may participate as Cluster Heads.
- **Clustering Enabled** – Normal Clustering is enabled for the network, with all nodes eligible for Cluster Head designation.

As alluded to above, for clustering to occur within a DQSS not only must the overall Cluster Topology be specified, but so must the Clustering Methodology.

2.5.1 Clustering Methodologies

At present there are three distinct Clustering Methodologies:

1. Static Clustering;
2. Traffic Flow Clustering; and,
3. Traffic Flow with Topology Coverage Clustering.

More Clustering Methodologies may be added over time; but these three represent the initial set. Each of the three Clustering Methodologies will now be discussed.

2.5.1.1 Static Clustering

Regardless of the setting of the Cluster Topology for a given DQSS, if the Cluster Methodology is set to "Static Clustering", then Dynamic Cluster is completely disabled. This is the only setting allowed for the "Clustering Disabled" and "Clustering Enabled for Backup Only" Cluster Topologies.

If this setting is used for either the "Limited Clustering Enabled" or "Clustering Enabled" topologies, then the net effect is to force the overall network topology into that of "Clustering Enabled for Backup Only".

2.5.1.2 Traffic Flow Clustering

Traffic Flow Clustering enables the Cluster Head to be located at the node providing the most efficiency with respect to being a "gate keeper" of the traffic flow. Because all communication and control is distributed and is not routed through a central spoke in order to communicate with other nodes within the DQSS, the only real advantage to the Cluster Head moving as the flow moves would be if the gateway can move with it. Meaning, the Cluster Head nodes have dual functionality with one port servicing the DQSS and other ports servicing one or more gateways.

2.5.1.3 Traffic Flow with Topology Coverage Clustering

Traffic Flow with Topology Coverage Clustering enables the Cluster Head to be located at the node providing the greatest coverage for the current traffic flow. The distinction between this mode and standard "Traffic Flow Clustering" is that the former does not take into account the overall range of coverage of the client nodes within the DQSS.

Similar to standard "Traffic Flow Clustering", because all communication and control is distributed and is not routed through a central spoke in order to communicate with other nodes within the DQSS, the only real advantage to the Cluster Head moving as the flow moves would be if the gateway can move with it. Thus, as above, in order for this mode to be effective, Cluster Head nodes must have dual functionality with one port servicing the DQSS and other ports servicing one or more gateways.

2.6 Additional DQSS Rules and their impact on Security

The Cluster Head distributes the DQSS table on a periodic basis. No node may communicate with another node unless both nodes are contained within the same DQSS. Because of the strict adherence to this policy, in order for a node to join and subsequently communicate with other nodes, including the Cluster Head, within the DQSS, the following sequence of events must occur:

a) The Cluster Head must explicitly acknowledge and admit a node for inclusion into the DQSS;
b) The Cluster Head must then add it to the DQSS and perform either a complete or partial[8] DQSS update of the DQSS Table to the nodes within the DQSS.

In short, the Cluster Head must first admit the node in the network and then secondarily inform the other nodes in the DQSS of the joining node's admission into the DQSS.

The format of the DQSS Table includes the following:

- DQSS Configuration Data; providing information specifying the functional and operational makeup of the DQSS. Information included would be the DQSS Mode (i.e. Static, Manual, Promiscuous, Promiscuous-Shared Key), Encryption Indication, DQ Gateway Information, Maximum DQ Frame and DQ Packet Sizes,
- 48-Bit MAC Address of every Node within the DQSS.
- 12-Bit DQSS Address; this address is assigned by the Cluster Head to the individual nodes within the DQSS as a means of reducing the amount of overhead within the transmission stream.
- Static Sized Code Word, assigned by the Cluster Head, and used for Access Requests to the Transmission Queue. This value is coupled with the DQSS Address on all access requests.
- Active or Inactive Indicators for Every DQ Member

Given that the primary purpose of the DQSS Table is to maintain the integrity of the network, a DQSS Table should be viewed as an Object Oriented Encapsulation of a specific DQ Network.

[8] When possible, the Cluster Head will update the DQSS Table through update distributions as a means of saving time and bandwidth. There are few instances in which a complete DQSS distribution will occur, with the nominal occurrence being during initialization and start-up of the DQSS.

2.7 Additional DQWA Features

2.7.1 DQWA and Fixed-Bandwidth

The bandwidth in DQWA can be divided into fixed-size segments and groups of contiguous segments are allocated to each DQ Frame but many applications, such as a fuel injection module would be better served with the equivalent of a TDM channel. DQWA supports this feature; a node requests that a segment be allocated on a recurring basis resulting in an isochronous (TDM) channel of the desired bandwidth. This feature is of true significance since it means that DQWA can satisfy with equal facility both packet and fixed-bandwidth requirements. This is described in detail by Wu and Campbell in [3].

2.7.2 Deterministic Transmit Queue Access

The collision resolution mechanism [5] for the Transmit Queue Contention Queue guarantees the number of collisions will not exceed $\log_x Y$[9] before achieving a successful resolution (i.e. the admittance of the first node into the transmission queue). This same mechanism also insures that all nodes involved within the initial collision set will be serviced within a finite, deterministic amount of time.

2.7.3 Energy Efficiency

Each DQ Data & Control Frame contains the total number of bytes within the frame at the beginning of the header; thus non-essential devices may go into a power save sleep mode for the period of the DQ Data & Control Frame transmission; awaking in time for the DQ Feedback Frame and inclusive DQ Transmission Beacon.

2.7.4 Utilization

There is no congestion in a DQSA network thus networks can be designed for average loading of 90%. The surges over 100% that cause chaos in conventional routers just mean that the distributed queues get longer, temporarily. There are no lost packets except for those lost due to Line Error. If only a single node has packets to send, that node can utilize 100% of the available capacity, when a second node desires to transmit, the available capacity is split automatically without any central control input, evenly between the two stations. And so on for an arbitrary number of stations. Priorities can be utilized to negate this inherent fairness.

2.7.5 Mission and Safety Critical Architecture

The distributive and non-static control aspect of DQWA affords DQWA to be used “As Is” within environments requiring mission critical and/or fail-safe architectures and without any additional redundancies in the network. Unlike conventional Hub-and-Spoke architectures, the current DQWA control node within a given DQWA network may fail without affecting the communication abilities of the remaining nodes within the DQSA network. In short, DQWA eliminates the single point of failure, which is common in all commercial network architectures deployed today.

This is huge benefit that Mission and Safety Critical applications a built-in mechanism within the network architecture for supporting their specific application. A DQWA network becomes part of the Mission and/or Safety Critical Solution and not another problem for which a work-around must be found (usually involving duplicate and/or alternative hardware and communication paths).

3 Conclusions

The distributive and non-static (i.e. transitional) control aspect of DQWA affords DQWA to be used “As Is” within environments requiring mission critical and/or fail-safe architectures (like that necessitated within the automotive domain) and without any additional redundancies in the network. Further, given the increasing security needs of automotive onboard network devices and the ubiquitous and pervasive nature of CAN; DQWA would be an excellent complimentary technology for wireless CAN networks; particularly as a wireless CAN backhaul topology.

[9] Where ‘x’ is the number of contention windows and Y is the number of nodes attempting to access the Contention Queue in the initial attempt.

Additionally, as more and more automotive modules require intra-vehicle network connectivity, wireless becomes the only viable alternative. The challenge is to enable the transition to wireless connectivity, reliably, safely, and most of all securely. DQWA provides the answer to this increasingly important and difficult problem.

REFERENCES

[1] Xu, Wenxin, & Campbell, Graham (1992). A Near Perfect Stable Random Access Protocol for a Broadcast Channel. *In Proc. of the IEEE International Conference on Communications 1992 (ICC'92)*, (pp. 370-374 Vol. 1).

[2] Campbell, Graham, & Xu, Wenxin. (2001). *Patent No. 6292493.* USA.

[3] C. T. Wu and G. Campbell "CBR Channels on a DQRAP-based HFC Network", *SPIE '95 (PhotonicsEast),* Philadelphia, PA Oct. 1995.

[4] Littleson, Randy, Senior VP, Marketing, Flexera Software,"Anyone for 1 quadrillion intelligent, connected devices on the Internet?", July 2011, http://blogs.flexerasoftware.com/ecm/2011/07/anyone-for-1-quadrillion-intelligent-connect-devices-on-the-internet.html

[5] Alonso-Zárate, J., Verikoukis, C., Kartsakli, E., Cateura, A., & Alonso, L. (2008). A near-optimum cross-layered distributed queuing protocol for wireless LAN [medium access control protocols for wireless LANs]. *Wireless Communications, IEEE , 15* (1), 48–55.

CONTACT INFORMATION

Barton Shields, bshields@ether2.com, 951.522.3540

Jonathan Gael, jgael@ether2.com, 323.874.4235