Management and configuration for MilCAN vetronic systems

George Valsamakis (UoS), Periklis Charchalakis(UoS), Bob Connor (Qinetiq), Elias Stipidis(UoS), Falah Ali(UoS)

Electronic network-centric devices are being widely used in military vehicles. The environments they operate are demanding and performance delivery is crucial under harsh conditions. Dictated by application needs, customised levels of configurability control and analysis are essential. The objectives of the work presented in this paper are to provide intelligent means of connectivity for integrating diverse network technologies, and tools for configuring and monitoring military vehicle electronic systems.

Our investigation introduces the use of smart bridges that allow access to custom tools for reconfiguration and network information from the system. The target is to enhance the means of configuring these bridges, and to acquire customised diagnostics that potentially provide enhanced performance solutions. Experimental investigations utilizing MilCAN and Ethernet are being conducted, using a variety of different network topologies and configuration sets, to interconnect multiple MilCAN segments over an Ethernet backbone.

Introduction

The high dependability requirements in device networks for Military Vetronics present significant challenges to their integration. Determinism, prioritisation, synchronised transmission and QoS, are tasks that provide reliability, yet with limitations in aspects such as network bandwidth and number of devices evident in certain applications. An intelligent bridging system [1] has been developed to retain these tasks and also act as the basis for managing the whole network. The investigation involves MilCAN (Military Controller Area Network) [2] and high-speed networks to connect multiple device segments in vehicular environments, and remote configuration management.

While the current MilCAN specification provides basic management functionality at sub-system level, the integration of multiple segments introduces the need for enhanced control and management capabilities at system level.

The present bridging [1] platform enables the provision of intelligent network monitoring with error detection and recovery, implementing system resilience and fault-tolerance ranging from network level down to single device level. Furthermore, allowing network managers to design the system configuration and arrange its operation without requiring low-level technical knowledge of the system.

MilCAN

MilCAN is an open-standard interface to the CANbus [3] technology aiming to provide hard and soft real-time capabilities required by military applications. It is being developed and maintained by the MilCAN Working Group that was formed in 1999 as a sub-group of the International High Speed Data Bus - Users Group (IHSDB-UG).

Consisting of two versions, MilCAN-A and MilCAN-B, and a set of three specifications, Physical Layer, Data Link Layer, and Application Layer, MilCAN defines *compulsory* features that are required to accomplish a deterministic network operation. Additionally, *optional* features (not required for MilCAN compliance) are also discussed mainly as recommendations to allow compatibility and direct plugability between different MilCAN devices implementations.

The two versions exist for interoperability with SAE-J1939 (MilCAN-A) and CAN-Open (MilCAN-B). MilCAN-A uses 29bit Message Identifiers and a frame format similar to SAE-J1939 making it possible to share the same CAN bus. It allows both periodic and event driven message generation. MilCAN-B uses 11 bit identifiers and operates on top of CANopen using a system developed by BOFORS. For simplicity purposes in this paper, focus is only given to MilCAN-A specification.

Physical Layer

The Physical Layer specification defines the physical and electrical connectivity, the transceiver characteristics, and bit timing of the network. For the physical connectivity the gender types of cables, devices, and interconnect components are specified as *compulsory*, while different network topologies such as linear multi-drop and daisy chain are recommended.

Data Link Layer

The Data Link Layer specification defines the CAN frame format, bus access control, and message types. MilCAN-A uses only 29bit extended identifiers with a format based on the SAE-J1939 and differentiated by the Protocol type bit (bit 25). This allows sharing of the same CAN bus between MilCAN-A and SAE-J1939 nodes. The message types support Operational and Non-Operational messages, System Configuration messages, Sync Frame and Alive messages. The first two types are used for the communication of the applications, while the last two for the network operation. The System Configuration messages instruct the nodes to enter a *maintenance* mode, during which their normal operation is suspended while they are configured remotely. The current specification requires all the nodes in a segment to enter the configuration mode while one or more of them are reprogrammed, but it is envisaged that single-node configuration will be implemented in the near future to allow run-time reconfiguration with minimum network disruption.

Application Layer

The MilCAN Application Layer specification defines the higher layer communication architecture of the network, such as message identifier assignments, multi-instance addressing, data/command distribution, and system operation modes. The system modes define the functional states of a MilCAN node such as Pre-Operational, Operational, and System Configuration. Transition between the system modes depends on the status of the network and the task of the specific node within its segment. When a node enters System Configuration mode its application suspends normal operation and only responds to configuration messages. During this state its application can be configured or reprogrammed depending on the requirements, and then resume operation.

Deterministic Operation

MilCAN uses a prioritised bus access and bounded throughput to support deterministic data transmission on the network, based on the criticality of each nodes function. For each of the MilCAN priorities, a guaranteed maximum transmission latency is assigned. By limiting message generation from each node to only one within their allocated period, the network traffic can be pre-scheduled to provide this deterministic operation. These latencies along with the whole network operation are based on a synchronisation mechanism, the Primary Time Unit (PTU). Each PTU accommodates the transfer of a maximum number of frames, based on their priority, which have been scheduled for transmission on that specific PTU or they have been generated earlier but due to higher priority traffic they have been stalled. The start of a PTU is triggered by a specific frame, the *Sync Frame*, broadcasted by a single node, the *Sync Master*. The functionality of a *Sync Master* can exist in multiple nodes, along with their dedicated user applications, to prevent operation disruptions on single node failures. Using an election process one of the possible *Sync Masters* is selected while the others keep monitoring the *Sync Frame* generation on the network and takeover if the current *Sync Master* fails.

VSI Bridge

The VSI Bridge was developed in collaboration between the Vehicle Systems Integration Group at QinetiQ and the Communications Research Group at Sussex University. The main target was to provide transparent interconnection of MilCAN segments over high-speed non-deterministic networks such as Ethernet, along with the ability to monitor and filter MilCAN traffic. Being a development platform its design was based on a modular scheme to allow easy and quick implementation and evaluation of different bridging and routing techniques.

A typical usage of a VSI Bridge in a MilCAN based network consisting of multiple MilCAN segments is shown in figure 1. Based on their functionality, MilCAN nodes are separated into local segments that share a common backbone for intercommunication. However, an independent CANbus with basic MilCAN-to-MilCAN bridges that block local-segment traffic can be used as a backbone. The low bandwidth availability of CAN becomes an issue when the number of segments increase. Introducing VSI Bridges to the system, a high-speed network such as Ethernet can be used in conjunction with a CANbus to prevent bottlenecks by redirecting low-priority and non-real time traffic through the non-deterministic backbone while increasing the available bandwidth in the CANbus backbone used by the high-priority messages. Additionally, with the availability of multiple backbone links and available routing paths, a form of network redundancy is established where network failures can be detected and automatically recovered by the VSI Bridges using spare links and adjusting the routing paths of the network.

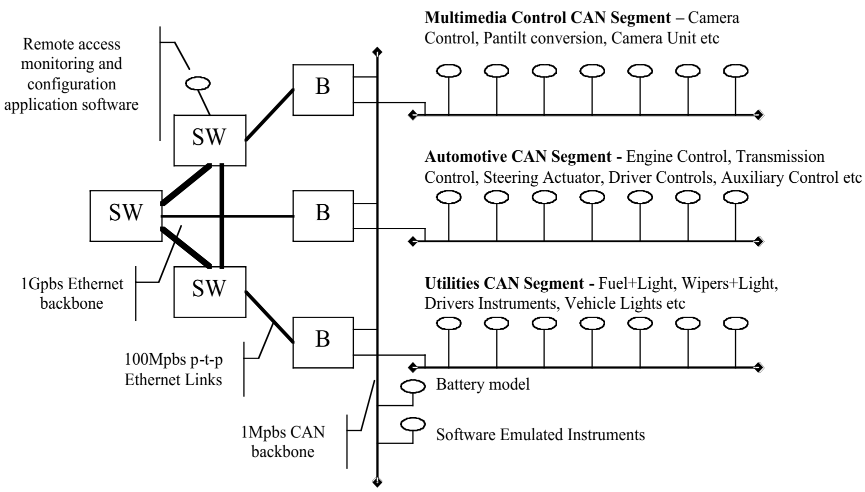

Figure 1: MilCAN Network with VSI Bridges

VSI Bridge Structure

The internal structure of the VSI Bridge consists of three main components, the *iFaces*, the *NAT*, and the *NCC*. The iFaces component handles the communication layer with the various network protocols used, such as MilCAN and Ethernet, each consisting of a pluggable sub-interface. It also manages the communication with a User Interface that remotely configures the VSI Bridge. The NAT component does the filtering and routing of the MilCAN frames based on a dynamic and flexible set of rules, while the NCC handles mainly internal tasks.

Operating in a deterministic environment the VSI Bridge has to adhere to the requirements of the network. For this reason the internal communication between the VSI Bridge components, and also the “processing” of the MilCAN frames, are handled based on their priorities. A custom interface, the *Standard Vetronics Interface* (SVI), was devised as a standardised way to handle MilCAN frames. It provides a transport mechanism of MilCAN frames within the application layers of a MilCAN system. The SVI frame format includes the structure of a MilCAN frame and additional fields to simplify its implementation in complex designs, and also improve overall performance. Using the SVI format, an abstraction layer is created between the CAN layer and the MilCAN stack, minimising hardware dependencies.

VSI Bridge Interfaces

The combination of the modular design and the SVI format, an abstraction layer within the *iFaces* component allows the use of different types of protocols as high-speed backbones transparently by the VSI Bridge, and an Application Interface to allow remote commissioning. Currently, the sub-interfaces implemented as backbone include TCP/IP and EtherNet/IP [4] over Ethernet. The Application Interface is used in conjunction with a Graphical User Interface (GUI), which communicate over Ethernet. Through the GUI a user can access, monitor, and modify the functionality of any VSI Bridge existing in the network. This includes manipulation of bridging and filtering rules, real time traffic monitoring, and operational status querying.

VSI Bridge Configuration Management

A set of tools is designed to support all interconnected VSI Bridges in a MilCAN network, and provide the basis for an on-line/off-line segment visualisation and management. This generic set of facilities allows the continuous monitoring of the system performance. The target is not only to provide the operational functions but to enhance performance and dependability by adapting network management and fault compensation methodologies.

A database system is designed to support the management tools for holding information at system and node level. It holds configuration and statistical information of MilCAN entities and their variables, and allows customisation for application specific requirements. To permit future deployment of alternative GUI systems, the core of the tools are a software library, where all the basic logic for communicating with the VSI Bridges and the database resides.

Operation

The user front-end tool (VSI-GUI) utilises the Application Interface of the local VSI Bridge to communicate with the network. The communication is based on a simple set of operations that gives users the ability to:

- Maintain a list of the VSI Bridges on the network with their configuration and topology.
- Generate MilCAN messages in SVI format and transmit to the network.
- Generate VSI Bridge configuration sets and transmitting them.
- Control and regulate network traffic and connections.
- Monitor MilCAN segments (node level).
- Retrieve and store performance statistics from VSI Bridges.

The ability to construct MilCAN messages by the GUI not only provides a basis for auditing network nodes, but also makes possible the transmission of messages to the network without the need to program and attach new nodes onto the network.

VSI Tool Design

The user interface employs a 3-tier design scheme (Figure 2) such that the important tasks for communicating with the VSI Bridges operate independently from the front-end and the data storage. This makes the design modular for future adaptation with alternative means of interaction than an application may require.

The layered structure incorporates COM (Component Object Model) Interfaces for the internal communication between the components of the system.

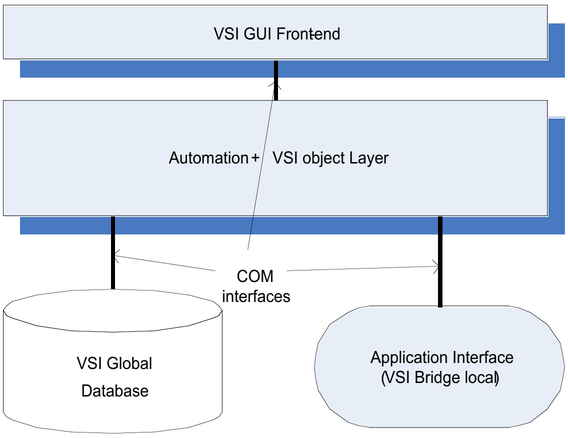

Figure 2: VSI Management Tool 3-tier structure

The Automation Layer is where the VSI Bridge functionality resides while the GUI front-end acts passively between the system and the user. It holds 3 interfaces: Network, Control and Database. The Network Interface is responsible for the data reception and transmition from and to the VSI Bridge, while the Control Interface is responsible for sending VSI Bridge commands and receiving current status and configuration information. The Database Interface is responsible for all the data concerning the data storage. The VSI Bridges use this interface to report their configuration and any other relevant information. The Application Interface of a VSI Bridge implements these interfaces of the Automation Layer to achieve communication.

VSI Bridge Online Configuration

The internal communication of the Bridge components is synchronous to the operation of the network allowing real-time reconfiguration without having to stop and restart the whole system. Commands received by the remote configuration tool are processed and executed by temporarily suspending the internal operation of the component being reconfigured. As the other components of the system are functioning normally, previously pending and newly arrived messages to be processed are queued until the reconfiguration is finished. The component then resumes normal operation with the new configuration applying it to messages that will be processed from then on (queued or new messages). The suspension of the component is kept to the minimum possible not to impose significant delays to the latency of the processed messages, and disrupt the network operation.

MilCAN Network Management

A VSI Bridge is remotely controlled by the GUI using a custom protocol designed to accommodate the currently available features, in a dynamic manner to allow future additions and enhancements. This protocol is used for the communication of the GUI with the Application sub-interface of a VSI Bridge within the network, which then redirects the commands to the target VSI Bridge. Currently, the VSI Bridge supports extensive remote configuration through the GUI. This includes modification of the filtering rules, allowing both fine and coarse control to the traffic passing through the device, and the routing paths that redirect frames to their destination via one of the available backbone links. As the VSI Bridge is highly customizable, the system can be set to mangle and modify certain fields in MilCAN messages, allowing the management system to make even more drastic changes.

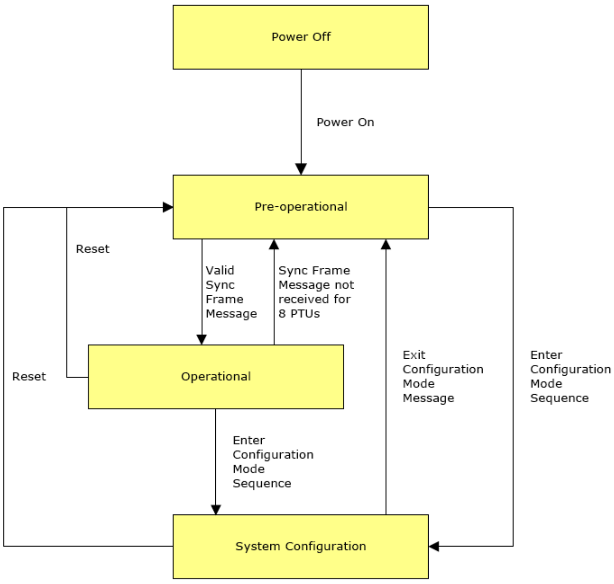

Figure 3: MilCAN Node operational modes

In a network of multiple MilCAN segments, the VSI Bridges act as network information agents. The remote configuration tools receive continuous status information from the VSI Bridges, permitting for management operations to take place. Using a single point of monitoring and control, node or network disruptions can be detected and corrected before escalating to a complete system failure. With a topology incorporating redundant backbone links, a network disruption due to link failures or high traffic loads can be corrected by reorganising the routing paths of the network. Fine grained control to the network could be further developed utilising the *System Configuration* mode of the MilCAN protocol (Figure 3). Nodes can be set to this mode for remote reconfiguration through the VSI management tool either manually or automatically.

Such tasks may be done reactively if performance drops to undesired levels according to the network resource capabilities. Performance variables such as message latencies and network utilisation are monitored on-request by the GUI for messages of various type and priority.

Conclusion and Discussion

The presented multi-objective model targets to a basis of an extendable infrastructure to support diverse application integration. Overall management aspects that arise naturally due to the introduction of complexity are dealt with by providing the basis of a network information and management system. Auditing, monitoring, and system configuration on segment level take place regardless the subjective application.

The design methodologies deployed allow the flexibility of cross-platform implementations, extensible information retrieval and remote system configuration. Further extension can provide fault tolerance on the application level with manual or automatic incident response, approaching the standards of highly reliable and resilient networks, suitable for military environments.

Acknowledgements

The work presented in this paper was funded by the UK MoD as part of the Applied Research Programme and undertaken by QinetiQ and the Communications Research Group of Sussex University.

References

[1] Periklis Charchalakis, George Valsamakis, Bob Connor, Elias Stipidis, "MilCAN and Ethernet", 9th International CAN Conference, Oct 2003, Germany.

[2] MilCAN - Adapting COTS CANBus to Military Vetronics, CIA/COTS February 2002

[3] CAN Specifications http://www.can-cia.de/

[4] Open DeviceNet Vendor Association,"EtherNet/IP", http://www.odva.org/

[5] George Valsamakis, Periklis Charchalakis, Bob Connor, Elias Stipidis "VSI Embedded network integrated systems for military vehicles based on MilCAN." to appear in Journal of Defense Science, Spring 2005, UK

Authors

George Valsamakis
University of Sussex
Falmer
Brighton
East Sussex
United Kingdom
BN1 9QT
Tel: +44 (0) 1273 678957
Fax: +44 (0) 1273 678399
G.Valsamakis@sussex.ac.uk
www.comms.scitech.susx.ac.uk

Periklis Charchalakis
University of Sussex
Falmer
Brighton
East Sussex
United Kingdom
BN1 9QT
Tel: +44 (0) 1273 678957
Fax: +44 (0) 1273 678399
P.Charchalakis@sussex.ac.uk
www.comms.scitech.susx.ac.uk

Bob Connor
QinetiQ
Cody Technology Park
Ively Road
Farnborough
United Kingdom
GU14 0LX
Tel: +44 (0) 1252 397011
Fax: +44 (0) 1252 397011
Rmconnor@QinetiQ.com
www.QinetiQ.com

Dr Elias Stipidis
University of Sussex
Falmer
Brighton
East Sussex
United Kingdom
BN1 9QT
Tel: +44 (0) 1273 678957
Fax: +44 (0) 1273 678399
E.Stipidis@sussex.ac.uk
www.comms.scitech.susx.ac.uk

Dr Falah H. Ali
University of Sussex
Falmer
Brighton
East Sussex
United Kingdom
BN1 9QT
Tel: +44 (0) 1273 678445
Fax: +44 (0) 1273 678399
f.h.ali@sussex.ac.uk
www.comms.scitech.susx.ac.uk