An improved CAN-switch with CANopen-management interface

Heikki Saha, Magnus Wikman, Patrik Nylund, Bertil Bäck

Complex machines use today control-systems consisting of multiple buses because any single bus can not support enough nodes. Moreover, any fatal failure in a single bus system will stop the whole system. With CAN, requirement of linear topology limits efficient networking in some systems. CAN-switch has successfully been used to solve those challenges. Benefits of a switch have been proved in real systems during last four years.

This paper presents latest improvements in CAN-switch technology. Most significant improvement has been made in forwarding. A new state-of-the-art TX-buffering scheme has been adopted from Ethernet and ATM-switches. With the new buffering packet loss no more exists at full 1Mbps bit rate in 4-port switch. According to the measurements, forwarding delay remains constant for all CAN-IDs due to table-based forwarding rules. The forwarding delay of the switch has been significantly reduced from the delay of the first prototype.

Another significant improvement is CANopen-based management interface. It enables seamless integration of CAN-switch into CANopen networks by offering object-dictionary based managing. Fault monitoring exceptions are provided as EMCY-objects and heartbeat producer enables existence monitoring of a switch. Support of existing CANopen design flow and network design tools has made use of a switch as easy as any other CANopen node.

Introduction

In traditional industrial, office and home networks switches have been used for long time to improve net performance of the networks by dividing the network into segments and keep as much transmissions as possible local to those segments [5] [12]. Moreover use of switches in the network requires bidirectional point-to-point connections between switches and network nodes, which improve the overall performance. Typical switches operate at store-and-forward principle, where incoming telegram is completely received before transmission [4]. If compared with the traditional CSMA-networks, switched networks need flow-based analysis and design [4] [12] instead of telegram-based analysis and design.

The idea of utilize switches in CAN-networks has been introduced in [1]. Results of the first experiments were extremely encouraging – use of COTS gateway hardware as a switch revealed the same benefits as advertised in Ethernet and ATM networking. The biggest benefits in CAN networks are increased bus coverage and support of star- or tree-topologies without a need to reduce baud rate or deviate from standard termination scheme. Additional benefit was a possibility to run different baud rates in different parts of a network. Timeliness was improved, because a switch divides a bus into several error domains. Error frames occurred inside single part of a network is not visible outside that part and can not mess up the schedule of the whole logical network.

As documented in [12] for communication networks used in distributed computing, also CAN-networks used for industrial controls need to have predictable performance. Characteristics of the CAN switch prototype has been documented in [2]. The forwarding delay was slightly acceptable for the first control application, but needed to be improved together with small packet loss ratio.

This paper presents the improved switch structure covering switch HW, software forwarding principles and queue structure. Main targets are at least 25% bus load at 1Mbps without lost telegrams and forwarding delay below 1ms for all CAN-IDs independent of forwarding direction. Simultaneously with those improvements, the old ASCII-terminal based management and diagnostics interface was replaced with CANopen implementation to provide standardized, simple and easy to use for management and diagnostics.

Switch hardware

The COTS gateway hardware used in the switch prototyping had high-performance hybrid controller with only one on-chip CAN-controller. Two external CAN-interfaces were implemented with SJA1000 stand-alone CAN-controllers, where biggest performance bottleneck was multiplexed host processor interface. Another performance bottleneck was lack of multistage automatic prioritization in transmission buffer, which had to be replaced with software.

A new microcontroller with 5 on-chip CAN-controllers and integrated communication coprocessor was selected to avoid affect of slow external bus interfaces. New on-chip CAN-controllers are connected with parallel on-chip memory bus and have 3-stage automatically prioritizing transmission buffers reducing the time-consuming and performance hungry SW buffer processing.

Improved forwarding

The first prototype was actually a gateway where filtering and forwarding of incoming telegrams were not fast and reliable enough for switching. According to literature [7], SW-lookup is inefficient and Content-Addressable Memory (CAM) based HW-lookup is used in IP-networks. Because CAM is a complex device [8] [11] and too expensive for cost-sensitive mobile control systems, alternative solution was required.

Fortunately the first applications for CAN-switch are CANopen-systems using only 11-bit ID space. It enabled table-based SW filtering and forwarding of incoming telegrams. Route table with only 2048 words is needed. Table-based forwarding gives forwarding delay independent of CAN-ID of the telegram to be forwarded, which was one of the key requirements.

The switch retrieves an entry from the table using CAN-ID of the incoming telegram as an index. Entry structure is presented in Figure 1. After getting the CAN-ID specific entry, source port defines which nibble will be used to determine the target ports. Every nibble consists of boolean flags controlling whether the received telegram will be forwarded to the corresponding target port or not.

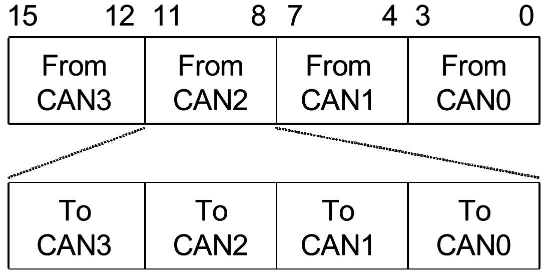


Figure 1: Route table entry structure

Improved queue structure

10 different telegrams can be received from and transmitted to single port in the first prototype. There are own buffers for every received telegram, which may cause lost reception events and furthermore lost telegrams. In the new switch, the improved queue structure has been adopted from high-performance ATM switches.

The selected new HW platform is based on high-performance communication coprocessor. Instead of parallel switching network, it offers internal speedup [9]when compared with maximum port speed. According to the [5], local TX and RX buffers are needed to avoid message losses caused by internal service order. CAN-controllers in the selected HW offer 5-stage FIFO-buffer for reception and automatically prioritizating transmission buffer supporting 3 telegrams.

Input buffering has been found to be the most simple [6] [10] [12] but major problem occurs when the first received telegram in the input queue can not be forwarded and that telegram blocks the whole queue [6].

It is called head-of-line (HOL) blocking [45] [10] [12] and it reduces the throughput significantly [6]. HOL-blocking can be solved by implementing virtual output queues (VOQ) in every input port [10]. In addition a scheduler is needed for selecting, from which VOQ the next message will be moved into transmission buffer. The major problem of input queuing is complexity. FIFOs are degrading performance [12], which leads to more complex queue structures. Every input port requires an own VOQ [9] [10], which increases total amount of queues. A scheduler is always needed for moving telegrams from input buffers or VOQs to the output ports [6] [9] [10].

Output queuing solves the major problems of input queuing [6]. All received telegrams can be immediately forwarded to output ports and inputs are never blocked. When telegrams can not be transmitted into target bus, only TX queue of the corresponding port will overflow.

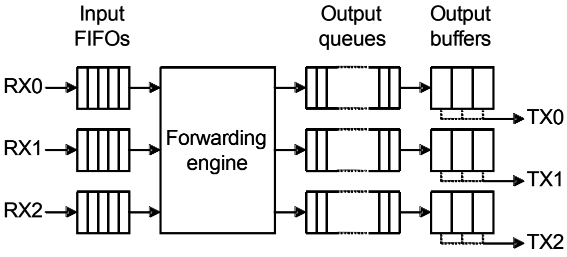


Figure 2: Block diagram of the improved CAN switch

The structure of the new switch is presented in Figure 2. Input FIFOs and output buffers are provided by the new HW. Forwarding engine SW reads telegrams from input FIFOs and write them into output SW queues according to the route table. Telegrams are moved from output queues into output buffers, where they are sent to the target buses.

CAN-ID in the CAN-telegram defines priority of a telegram. If more than one telegram is transmitted simultaneously, one with smallest CAN-ID wins the arbitration and remains unchanged. To keep the nature of switched CAN networks in line with standard CAN networks, the same behavior is implemented by HW output buffers. SW-queues operate as FIFOs.

Single-telegram measurements

The first test phase is to send single telegram from any single port to any another single port. This test gives the minimum possible forwarding delay. The best case delay has been measured with single telegrams, when the target bus is always free and arbitration in the target bus does not cause extra delay.

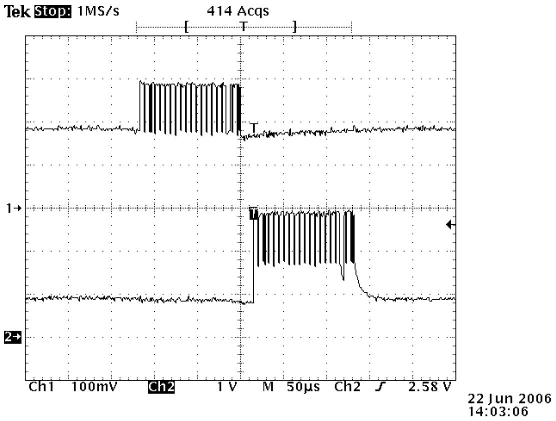


Figure 3: Best case forwarding delay for single telegram

According to the Figure 3, best case forwarding delay is 15µs instead of 135 µs measured from the first prototype [2]. Unlike with first prototype, the improved switch offers equal performance for all directions.

System scale test

Table 1: Forwarding rules for system scale performance test

Port 0		Port 1		Port 2	
RX	TX	RX	TX	RX	TX
0x1B2	0x000		0x000	0x000	0x1B2
	0x222	0x1B3		0x222	0x1B3
	0x223	0x1B4		0x223	0x1B4
	0x224	0x1B5		0x224	0x1B5
	0x225	0x1B6		0x225	0x1B6
	0x226	0x1B7		0x226	0x1B7
	0x227	0x1B8		0x227	0x1B8
	0x228	0x1B9		0x228	0x1B9
	0x229	0x1BA		0x229	0x1BA

To get overall results comparable with results measured form prototype, system level test setup equal to the prototype test [2] has been used. The forwarding rules are presented in Table 1. Port2 operates at 1Mbps and other ports operate at 500kbps. The test setup contains the most essential transmissions from the first target application.

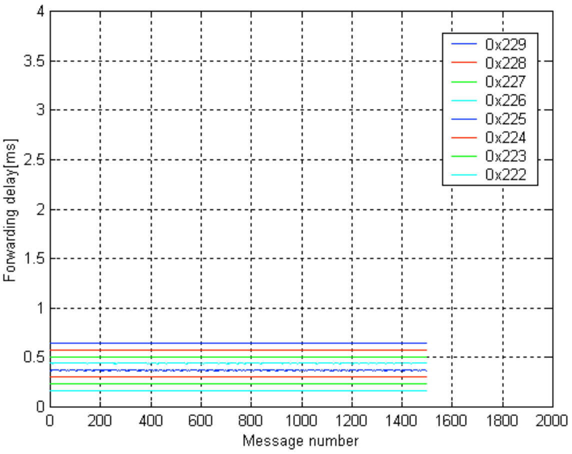


Figure 4: Forwarding delays from 1Mbps Port2 to 500kbps Port0

Port2 to Port0 forwarding delays in Figure 4 show, how fast the improved filtering and forwarding work. The small delay increase between the telegrams with increasing CAN-IDs is caused by transmission delay longer in the slower target bus than reception delay in the faster source bus. Readers shall notice that any higher priority telegram sent in the target bus will increase the delay and therefore generic performance values for all applications can not be given.

The results in Figure 5 show the constant performance independent of direction and CAN-ID. A small difference between delays of 0x222 and 0x1B2 comes from baud rate differences. In Port0 to Port2 forwarding, target bus is faster than source bus and baud rate difference slightly reduces the forwarding delay instead of increasing like in Figure 4 where target bus is slower than source bus.

Like in Figure 5, also in Figure 6 the destination bus is faster than source bus. Forwarding delay does not depend on the CAN-ID, because previously received telegram has been sent completely before receiving the next telegram. The minimum delay equals with the minimum level measured from other directions.

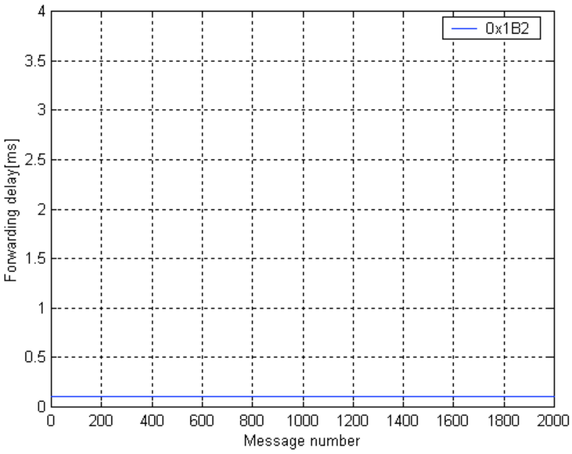


Figure 5: Forwarding delay from 500kbps Port0 to 1Mbps Port2

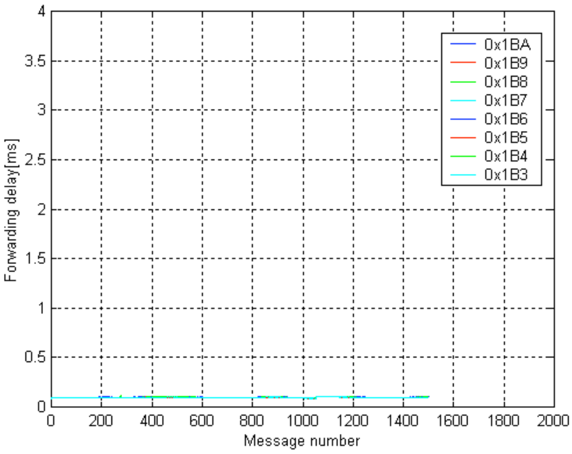
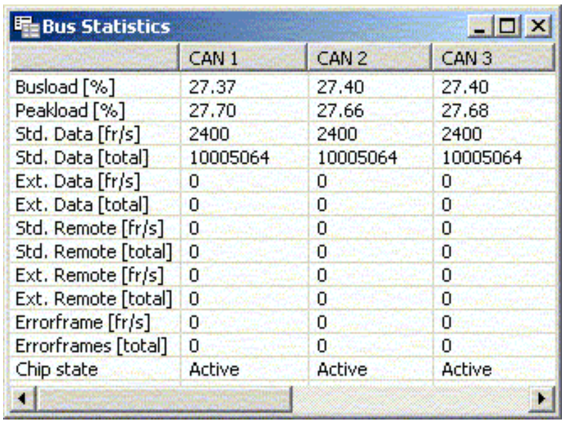


Figure 6: Forwarding delays from 500kbps Port1 to 1Mbps Port2

The queue improvements have removed the delay variations and improved the maximum forwarding delay from 1.7ms down to 200µs, which is a lot below the target. According to the results, arbitration can cause much more delay to the transmission than single CAN switch.

Stability

Long-term stability has been tested by transmitting bursts of 8 telegrams at fixed 10ms interval into 3 ports. In single burst, telegrams have been transmitted with minimum interframe space to stress test the queues. All received telegrams were forwarded to other two ports to simulate normal CAN bus. The resulting average bus load was 27% and after 10 million telegrams none of them was lost as presented in Figure 7. Improvement is significant when compared to results in /2/.



	CAN 1	CAN 2	CAN 3
Busload [%]	27.37	27.40	27.40
Peakload [%]	27.70	27.66	27.68
Std. Data [fr/s]	2400	2400	2400
Std. Data [total]	10005064	10005064	10005064
Ext. Data [fr/s]	0	0	0
Ext. Data [total]	0	0	0
Std. Remote [fr/s]	0	0	0
Std. Remote [total]	0	0	0
Ext. Remote [fr/s]	0	0	0
Ext. Remote [total]	0	0	0
Errorframe [fr/s]	0	0	0
Errorframes [total]	0	0	0
Chip state	Active	Active	Active

Figure 7: Stability test results

Integrated diagnostics

Integrated diagnostic features were designed well already in the first prototype. Queue improvements enabled improved queue overflow indications not supported by the first prototype. Error messages are transmitted into separate management interface covered later in details. All errors are indicated as CANopen EMCY telegrams. Following errors are supported by forwarding engine:

Port error warn: A warning is generated if serious communication problems exist in a port (see warning of serious transmission problems in CAN fault-confinement [3]).

Port error passive: A warning is generated if serious communication problems exist in a port (see error-passive state in fault-confinement [3]). A port in error passive state has decreased its transmission capability and may generate other problems too (TX buffer overrun of target port in error passive state).

Port bus-off: Port goes to bus-off if any message can not be sent to or received from a port (see bus-off state in fault-confinement [3]).

RX overrun: RX overrun error is generated when the switch can not read the messages from receive HW buffer fast enough. As long as the HW is not damaged and program memory not corrupted, the RX overrun error should not exist.

TX overrun: If no other nodes are connected to the target port or target bus contains fatal error, TX overrun error has been generated by the switch to tell that messages can not be sent successfully to the target bus fast enough.

Inhibit time applies for EMCY transmissions and counter value is attached into manufacturer specific part of the EMCY telegram to offer amount of errors since last EMCY.

Because the management port of the switch is CANopen device, it supports heartbeat producer. It enables NMT master to monitor existence of the switch in the network. Switch enters automatically into NMT operational state to enable as fast startup as possible.

CANopen management

Instead RS-232 management and diagnostics interface used in the first prototype, CANopen [13] management interface has been developed to improved version. Management interface use its own port, which can be left open or connected to any other interface. Typically management port is connected into a switching port connected with same bus segment than primary error message consumer in the system or subsystem. CANopen management interface offers following benefits:

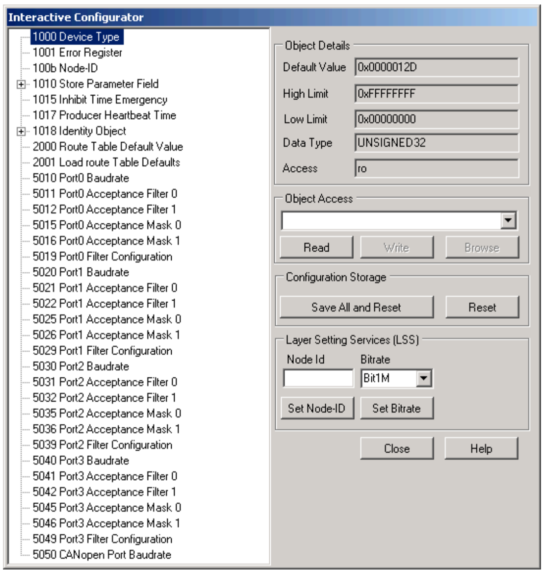
Uniform onboard diagnostics: The switch supports heartbeat and emergency diagnostic services like in any CANopen device. Additional protocols and/or software are not needed.

Standard tool support: All standard CANopen network design, configuration download and monitoring tools required by rest of the system can be used also with the CAN switch.

Standard process support: Standard CANopen design process can be utilized for management of switch route table and interface parameters.

Interface parameters like baud rates and HW filter settings can be adjusted via manufacturer specific objects. To help system integrators, two alternative methods are supported to manage the route table:

Simple method: To keep the process simple and Electronic Datasheet (EDS) file [14] compact the switch supports two objects, through which the whole route table can be configured with two expedited SDO transactions. At first the route table entry value for all CAN-IDs shall be written into object "Route Table Default Value" and then writing keyword "load" into object "Load Route Table Defaults" fills the whole route table with the specified value. Example view of simple method has been presented in Figure 8.

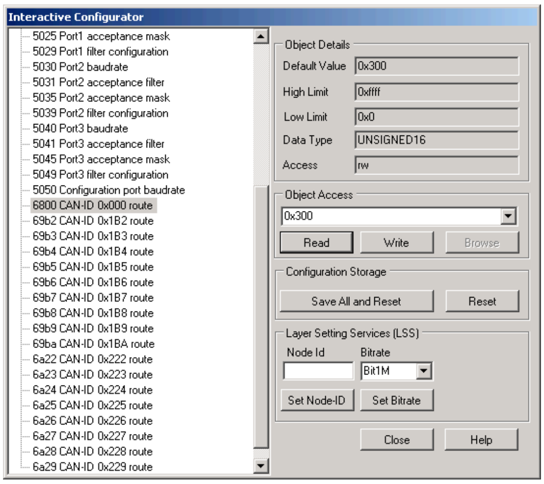


1000 Device Type
1001 Error Register
100b Node-ID
1010 Store Parameter Field
1015 Inhibit Time Emergency
1017 Producer Heartbeat Time
1018 Identity Object
2000 Route Table Default Value
2001 Load route Table Defaults
5010 Port0 Baudrate
5011 Port0 Acceptance Filter 0
5012 Port0 Acceptance Filter 1
5015 Port0 Acceptance Mask 0
5016 Port0 Acceptance Mask 1
5019 Port0 Filter Configuration
5020 Port1 Baudrate
5021 Port1 Acceptance Filter 0
5022 Port1 Acceptance Filter 1
5025 Port1 Acceptance Mask 0
5026 Port1 Acceptance Mask 1
5029 Port1 Filter Configuration
5030 Port2 Baudrate
5031 Port2 Acceptance Filter 0
5032 Port2 Acceptance Filter 1
5035 Port2 Acceptance Mask 0
5036 Port2 Acceptance Mask 1
5039 Port2 Filter Configuration
5040 Port3 Baudrate
5041 Port3 Acceptance Filter 0
5042 Port3 Acceptance Filter 1
5045 Port3 Acceptance Mask 0
5046 Port3 Acceptance Mask 1
5049 Port3 Filter Configuration
5050 CANopen Port Baudrate

Default Value: 0x0000012D
High Limit: 0xFFFFFFFF
Low Limit: 0x00000000
Data Type: UNSIGNED32
Access: ro
Node Id:
Bitrate: Bit1M

Figure 8: Example view of simple configuration method

Detailed method: Route table can be configured CAN-ID wise if needed. Simple method is recommended method for loading default values. Then a special EDS with CAN-ID specific forwarding rules shall be created. Size of the resulting EDS-file strongly depends on the number of CAN-IDs needing forwarding information. Practice has been shown that efficiency of detailed method can be increased by composing the EDS-file from CANopen Profile Databases (CODB) [15] instead of using an EDS containing full route table support. The two most important CODBs contain communication profile area and manufacturer specific area objects existing in the EDS-file used in simple method. Then additional CODB containing the application specific route table definitions are included into application specific CODB. Finally the EDS file for the detailed configuration method is composed by just combining the previously defined CODBs. Example of detailed route table configuration is presented in Figure 9.

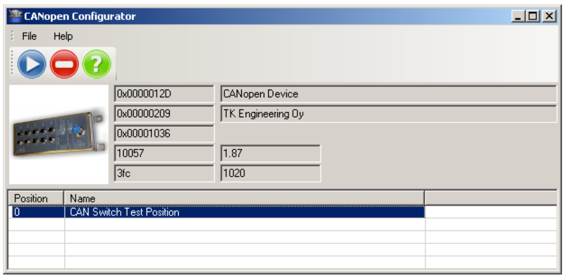


5025 Port1 acceptance mask
5029 Port1 filter configuration
5030 Port2 baudrate
5031 Port2 acceptance filter
5035 Port2 acceptance mask
5039 Port2 filter configuration
5040 Port3 baudrate
5041 Port3 acceptance filter
5045 Port3 acceptance mask
5049 Port3 filter configuration
5050 Configuration port baudrate
6800 CAN-ID 0x000 route
69b2 CAN-ID 0x1B2 route
69b3 CAN-ID 0x1B3 route
69b4 CAN-ID 0x1B4 route
69b5 CAN-ID 0x1B5 route
69b6 CAN-ID 0x1B6 route
69b7 CAN-ID 0x1B7 route
69b8 CAN-ID 0x1B8 route
69b9 CAN-ID 0x1B9 route
69ba CAN-ID 0x1BA route
6a22 CAN-ID 0x222 route
6a23 CAN-ID 0x223 route
6a24 CAN-ID 0x224 route
6a25 CAN-ID 0x225 route
6a26 CAN-ID 0x226 route
6a27 CAN-ID 0x227 route
6a28 CAN-ID 0x228 route
6a29 CAN-ID 0x229 route

Default Value: 0x300
High Limit: 0xffff
Low Limit: 0x0
Data Type: UNSIGNED16
Access: rw
Object Access: 0x300
Node Id:
Bitrate: Bit1M

Figure 9: Example view of detailed configuration method

Support of standard design flow and file formats enable storing switch position specific configuration into Device Configuration File [14]. Configuration can be easily downloaded in production line from DCF into a brand new switch with any standard CANopen configuration download tool like the one presented in Figure 10.



0x0000012D CANopen Device
0x00000209 TK Engineering Oy
0x00001036
10057 1.87
3fc 1020

Position	Name
0	CAN Switch Test Position

Figure 10: Configuration download into switch with standard configuration download tool

Discussion

Four years successful evaluation of switched CAN buses in real applications proved the benefits of the concept. The most significant bottlenecks were recognized during the tests and the first production version was developed.

The most significant improvement was selection of a new HW platform serving better CAN switching by offering more processing power and 5 on-chip CAN-controllers with good HW message buffers. The structure of a new hardware platform enabled moving part of the buffering from SW to HW, which improved the performance. Remaining SW queue structures and forwarding principles were adopted from ATM switches to further improve the switch performance. Due to the improved internal structure, more detailed diagnostics could be implemented.

With previously listed improvements performance target was exceeded. Forwarding delay was reduced 89% down to 15μs. Based on stress test using bursts of 8 telegrams, 27% bus load was reached without lost frames, which also exceeds target value set for the upgrade.

The management and diagnostics interface was improved too. To serve well especially CANopen networking, management is made via object dictionary and error messages are using CANopen EMCY protocol. CANopen nodes in the same network can monitor existence of a switch with heartbeat protocol. When compared with the first prototype, standard CANopen design process, design-, configuration download- and analysis-tools can be used also with CAN-switches.

Future development will concentrate on improvement of physical layer diagnostics and increasing number of switching ports. Biggest challenges will be finding more possible HW platforms and feasible licensing of CAN-controller IP-blocks to be able to evaluate FPGA-implementation.

References

[1] H. Saha, Comparison of system level networking solutions with high-speed CAN networks, Proceedings of 9:th international CAN Conference, CiA, October 2003, pp. 09-01..09-08

[2] Saha H., CAN-switch performance and fault-tolerance evaluation, CAN Newsletter 4[2005, CiA, 2005, pp. 46..50

[3] Road Vehicles - Controller Area Network - Part 1: Data link layer and physical signaling, ISO 11898-1:2003(E), ISO, 2003, 52 p.

[4] Jasperneite J., Neumann P., Theis M., Watson K., Deterministic real-time communication with switched Ethernet, 4th IEEE International Workshop on Factory Communication Systems, IEEE, 2002, pp. 11-18

[5] Elsaadany A., Singhal M., Ming T. Liu, Performance study of buffering within switches in local area networks, Proceedings of Fourth International Conference on Computer Communications and Networks, 1995, pp. 451-455

[6] M. J. Karol, M. G. Hluchyj, S. P. Morgan, Input Versus Output Queuing on a Space-Division Packet Switch, IEEE Transactions on Communications, vol. COM-35, NO. 12, 1987, pp. 1347-1356

[7] P. Berube, M. MacGregor, J. Nelson Amaral, FPGA implementation and experimental evaluation of a multizone network cache, Microprocessors and Microsystems 28 (2004), pp. 237-252

[8] Pagiamtzis K., Sheikholeslami A., Content-addressable memory (CAM) circuits and architectures: a tutorial and survey, IEEE Journal of Solid-State Circuits, Volume 41, Issue 3, pp. 712- 727, ISSN 0018-9200

[9] J. Chao, Saturn: A Terabit Packet Switch Using Dual Round-Robin, IEEE Communications Magazine, IEEE, December 2000, pp. 78-84

[10] M. Nabeshima, Performance Evaluation of a Combined Input- and Crosspoint-Queued Switch, IEICE Transactions of Communication, Vol.E83-B, NO.3 March 2000, pp. 737-741

[11] Hanzawa S., Sakata T., Kajigaya K., Takemura R., Kawahara T., A large-scale and low-power CAM architecture featuring a one-hot-spot block code for IP-address lookup in a network router, IEEE Journal of Solid-State Circuits, Volume 40, Issue 4, 2005, pp. 853- 861, ISSN 0018-9200

[12] T. E. Anderson, S. S. Owicki, J. B. Saxe, C. P. Thacker, High-Speed Switch Scheduling for Local-Area Networks, ACM Transactions on Computer Systems (TOCS), Volume 11, Issue 4, ACM, 1993, pp. 319-352, ISSN 0734-2071

[13] CANopen Application Layer and Communication Profile, CiA DSP-301 rev. 4.1.1, CiA, 2006

[14] Electronic Datasheet Specification for CANopen, CiA 306 rev. 1.3, CiA, 2005

[15] Profile Database Specification for CANopen, Appendix to CiA 306 rev. 0.3, CiA, 2000

Heikki Saha
Research Engineer
UHM Research, Automation Systems
Sandvik Mining and Construction Oy
P.O. BOX 100, 33311 TAMPERE, Finland
Tel: +358 (0)400 346 537
Fax: +358 (0)205 44 120
Email: heikki.saha@sandvik.com
http://smc.sandvik.com/

Magnus Wikman
Sales Manager
TK Engineering Oy
Wolffintie 36F2]III, FIN-65200 VAASA, Finland
Tel: +358 (0)50 511 9735
Fax: +358 (0)6 357 6320
Email: magnus@tkengineering.fi
http://www.tkengineering.fi/

Patrik Nylund
Project Manager, SW
TK Engineering Oy
Wolffintie 36F2]III, FIN-65200 VAASA, Finland
Tel: +358 (0)40 910 0473
Fax: +358 (0)6 357 6320
Email: patrik@tkengineering.fi
http://www.tkengineering.fi/

Bertil Bäck
Project Manager, HW
TK Engineering Oy
Wolffintie 36F2]III, FIN-65200 VAASA, Finland
Tel: +358 (0)50 362 7517
Fax: +358 (0)6 357 6320
Email: bertil@tkengineering.fi
http://www.tkengineering.fi/