PREPRINTS

ECRTS
8-11 July
Madrid, Spain
2014

13th International Workshop on Real Time Networks RTN'14

Madrid, Spain, July 8, 2014
In conjunction with the 26th ECRTS

http://irt.enseeiht.fr/scharbarg/rtn2014.html

Chairs:

Jean-Luc Scharbarg
Université de Toulouse – IRIT/INPT/ENSEEIHT
Michael Short
Teesside University

Workshop Chairs

Jean-Luc Scharbarg, Université de Toulouse - IRIT/INPT/ENSEEIHT, France

Michael Short, Teesside University, UK Eduardo Tovar, IPP-HURRAY,
Portugal

Program Committee

Luis Almeida, University of Porto, Portugal

Moris Benham, MRTC/Mlardalen University, Vsteras, Sweden

Jean-Dominique Decotignie Swiss Center for Microtechnology, Switzerland

Lucia Lo Bello, University of Catania, Italy

Julian Proenza Universitat de les Illes Balears, Spain

Thilo Sauter, Austria Academy of Science, Austria

Ye-Qiong Song, LORIA, France

Advance Program

8:00-8:30	**Registration**
8:30-10:00	**Session 1 - Keynote Talk** *Switched Networks as Automotive Backbones - Status and Open Challenges* Rolf Ernst
10:00-10:30	**Coffee Break**
10:30-12:00	**Session 2** *Optimizing worst case delay analysis on wormhole networks by modeling the pipeline behaviour* Laure Abdallah, Mathieu Jan, Jérôme Ermont and Christian Fraboul *Towards a more distributed avionics architecture* Emilie Berard, Christian Fraboul and Fabrice Le Sergent *Worst-case backlog for AFDX network with n-priorities* Rodrigo Coelho, Gerhard Fohler and Jean-Luc Scharbarg
12:00-13:30	**Lunch**
13:30-15:00	**Session 3 - Invited talk** *Information processing for extreme dense sensing: timeliness and scalability issues* Eduardo Tovar
15:00-15:30	**Coffee Break**
15:30-16:30	**Session 4** *Low Level Error Detection For Real-Time Wireless Communications* Jeferson Luiz Rodrigues Souza and José Rufino *A networking infrastructure for small smart grids* Michael Short and Muneeb Dawood
16:30-18:00	**Panel discussion and closing remarks**

Session 1 - Keynote talk

Switched Networks as Automotive Backbones - Status and Open Challenges

Rolf Ernst
Technische Universitaet Braunschweig, Germany

Abstract

Current automotive networks consist of different field buses and gateways with new protocols and topologies in almost every generation. This federation shall now be replaced by Switched Ethernet as a common backbone standard. The talk will outline the rationale for this development, identify the status and point to open challenges.

Session 2

Optimizing worst case delay analysis on wormhole networks by modeling the pipeline behaviour

Laure Abdallah and Mathieu Jan
CEA, LIST, Embedded Real Time Systems Laboratory
F-91191 Gif-sur-Yvette, France
Email: Firstname.Lastname@cea.fr

Jérôme Ermont and Christian Fraboul
IRIT INP-ENSEEIHT, Université de Toulouse
F-31000 Toulouse, France
Email: Firstname.Lastname@enseeiht.fr

***Abstract*—Wormhole switching is widely used within Network-on-Chip, due to its improved throughput. Real-time packet schedulability analysis of wormhole networks have been studied, but without taking advantage of the pipeline transmission of flits making packets. In this paper, we illustrate through two examples that not taking advantage of this pipeline transmission of flits is a major source of pessimism in the computation of worst-case end-to-end delays of flows. The analytically computed delays are compared to the one obtained using a state-of-the-art method, showing the feasibility of greatly reducing the pessimism. We are currently developing a generic algorithm to compute these delays based on this idea of modeling the pipeline behaviour of wormhole networks.**

I. Introduction

In hard real-time systems, the worst-case end-to-end delay of all the packets generated by a flow must be lower than a predetermined deadline. Such real-time packet schedulability analysis have been done for various types of networks by taking into account the type of contentions that can occur. A flow can be classified as either being in direct or indirect contention with the analyzed flow f. A direct flow shares at least one link with f, while an indirect flow shares at least one link only with a direct flow of f. In the end, the goal of new methods is to reduce the pessimism in the values that are obtained, generally by adding assumptions on the behaviour of the analyzed system.

In this paper, we focus on wormhole networks. Wormhole switching [1] is used within different types of networks: from Network-on-Chip (NoC) of many-core systems to specific networks such as SpareWire. In these wormhole switching networks, a packet is divided in flow control digits (flits) of fixed size, which are transmitted one by one by routers. The header flit, i.e. the first flit, contains the routing information that defines the path the next flits will follow in a pipeline way.

However, current state-of-the-art methods to compute worst-case end-to-end delays (simply noted delays) in wormhole networks do not model this pipeline way of transmitting flits. Instead, the delays of all the flows that block the analyzed flows are simply added. In wormhole switching, when the last flit of a packet leaves an output port of a router, this port is available for the other flows in order to progress. In this paper, *we show that integrating this pipeline behaviour of wormhole networks in the computation of delays can greatly reduce the pessimism of the obtained values*. We illustrate that on two examples and compare analytically computed delays with the one obtained by applying a method proposed in [3].

II. Related work

Recursive Calculus (RC) [3] is a method to compute an upper-bound on the worst-case delay of a packet in a SpaceWire network. A packet delivery is assumed to be divided into two phases. In the first phase, the header of the packet is routed to its destination and creates a *virtual circuit* between the source and the destination. In the second phase, the whole packet is then transferred. The method recursively analyzes the contention in the path of the analyzed flow f. At each router, the direct flows are identified and their delays are computed by thus taking into account the indirect flows of f. As a round robin arbitration is assumed, at each input port, the maximum delay from the set of flows in direct or indirect contention with f is added.

In [6], another method is presented for computing the delays in a real time communication with priority-based wormhole switching. In our work, we instead consider Commercial off-the-shelf (COTS) wormhole switching networks with round-robin arbitration, such as what is available in [7]. In [5], the network calculus is used to model the flow control mechanism of wormhole switching as a service curve. [4] clearly showed with an illustrative example that such a modeling is pessimistic, leading to over-dimensioning the resources.

Recently, [2] identifies two main sources of pessimism and introduces constaints on the input traffic of the flows and on the behaviour of the tasks to overcome them. Using a minimum inter-release between consecutives packets of f and and upper-bound on the number of packets that f can generate in a given interval time reduce the obtained delays. However, these assumptions are at an application level, while we focus on the network behaviour.

III. Examples illustrating the pessimism

First, let us consider a deterministic X-Y routing with links of capacity C. We also consider a single input queue per channel of size equals to f_{size}, the size of a flit. We consider that all the packets are made of 3 flits. We assume a continuous generation of packets at the source, while the destination immediately consumes any flit arriving as well as forwards back a credit to the previous router. Let us note the switching delay of the header flit d_{sw_1}. It is equal to the time for the router to grant an output port for the packet. The other flits of the same packet have a lower switching delay noted d_{sw_2}. The traversal delay of a link, noted d_t, is equal to $\frac{f_{size}}{C}$, should it be for regular data or credits due to the flow control.

Figure 1 illustrates the simple NoC architecture and the flows we consider in our two examples. The analyzed flow is

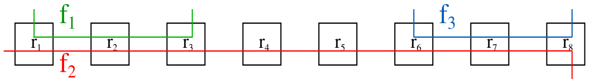


Fig. 1. A simple NoC architecture composed of 8 routers to illustrate the pessimism when not modeling the pipeline behaviour of wormhole networks.

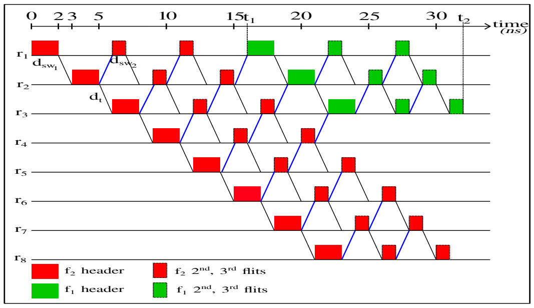


Fig. 2. Timeline of the transmission of the flows f_1 and f_2.

f_1. In our first example we only have the flow f_2 in direct contention with f_1, while in our second example we add the flow f_3 in indirect contention with f_1. In the RC method, the analyzed flow is considered to be blocked till all the flows in contention arrive to theirs destinations. The delay of the analyzed flow is therefore also made dependant on the remaining distance between its destination and the destination of the flows in contention. We now show that this can be pessimistic.

Figure 2 indeed shows the timeline of the transmission of the flits of our first example, i.e. when considering the flows f_1 and f_2 only. Each line represents a router, so once a flit is transmitted to next router (i.e. lines between r_i and r_{i+1}), a credit is also transmitted to the previous router (i.e. lines between r_i and r_{i-1}). When the third flit of f_2 crosses r_2 at time t_2, the header flit of f_1 is transmitted by r_1. In this case and independently of the progression of f_2 towards its destination, f_2 will no longer impact the delay of f_1. We now consider our second example, i.e with the additional flow f_3. Figure 3 shows the timeline of the transmission, where it can be noticed that f_3 does not block f_1. Obviously, the delay of flow f_1 in this second example is identical to its delay in the first example.

Let us note Er the number of empty routers between the destination of the analyzed flow and the source of an indirect flow. Let us further note N_f the number of flits a packet is made of. Then, the condition to check whether an indirect flow has no impact on the analyzed flow is the following: $Er \geq N_f - 1$. The number of flits considered in the condition is reduced by 1 in order to exclude the header flit. In our second example, we have $Er = 2$. The header flit of the direct flow f_2 is blocked by f_3 in the buffer of the router r_6. The remaining flits of f_2 are thus distributed on the empty routers r_4 and r_5 which are part of the path taken by f_2. As this number of empty routers is equal to $N_f - 1$, the flow f_1 is thus not blocked.

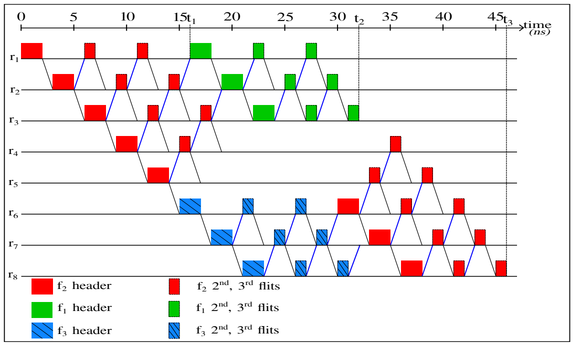


Fig. 3. Timeline of the transmission of the flows f_1, f_2 and f_3.

IV. First analysis and comparison with RC

Let assume that $d_{sw_1} = 2ns$ and $d_{sw_2} = 1ns$. The flit size is $f_{size} = 2B$ and the capacity of the links is $C = 16Gbps$, therefore $d_t = 1ns$. In the examples of Figures 2 and 3, the flow f_2 blocks the flow f_1 during $t_1 = 16ns$ (i.e. $t_1 = 4 \times d_{sw_1} + 2 \times d_{sw_2} + 6 \times d_t$). The transmission of the flow f_1 starts at this point and takes $t_2 = 32ns$ (i.e. $t_2 = t_1 + 3 \times d_{sw_1} + 4 \times d_{sw_2} + 6 \times d_t$).

When applying the RC method to the same configuration, the delay of f_2 is added when computing the delay of f_1, leading to an overall delay of $47ns$. In the second example, the delay of the indirect flow f_3 is further added, leading to a delay equals to $63ns$ for f_1 using the RC method. These delays are 1.5 and 2 times higher than the delays considering the pipeline behaviour.

V. Conclusion and future work

The method, used in [2], [3] for computing the worst-case end-to-end delays in wormhole networks, consider that an analyzed flow can be blocked by others flows while they have not reach their destinations. The method thus adds the transmission delays of the blocking flows to the analyzed flow, leading to an over-approximation of the transmission delays. As shown in the examples given in this paper, only a part of the blocking flows should be considered in order to compute tighter end-to-end transmission delays, due to the pipeline behaviour of the network. We also formulate a first property allowing to check if an indirect flow can impact the transmission delay of the analyzed flow. Some further works will be done in order to study the pipeline behaviour and to find a method to compute the worst-case end-to-end delays on wormhole networks.

References

[1] W. Dally. Virtual-channel flow control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2), Mar 1992.

[2] D. Dasari, B. Nikolić, V. Nélis, and S. M. Petters. NoC Contention Analysis Using a Branch-and-prune Algorithm. *ACM Trans. Embed. Comput. Syst.*, 13(3s):113:1–113:26, Mar. 2014.

[3] T. Ferrandiz, F. Frances, and C. Fraboul. A method of computation for worst-case delay analysis on SpaceWire networks. In *Proc. of the 4th Intl. Symp. on Industrial Embedded Systems (SIES)*, pages 19–27, Lausanne, Switzerland, July 2009.

[4] T. Ferrandiz, F. Frances, and C. Fraboul. Using Network Calculus to compute end-to-end delays in SpaceWire networks. *SIGBED Review*, 8(3):44–47, 2011.

[5] Y. Qian, Z. Lu, and W. Dou. Analysis of Worst-Case Delay Bounds for On-Chip Packet-Switching Networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(5):802–815, May 2010.

[6] Z. Shi and A. Burns. Real-Time Communication Analysis for On-Chip Networks with Wormhole Switching. In *Proc. of the 2nd Intl. Symp. on Networks-on-Chips (NOCS)*, pages 161–170, Newcastle, UK, April 2008.

[7] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. B. III, and A. Agarwal. On-chip interconnection architecture of the tile processor. *IEEE Micro*, 27(5):15–31, 2007.

Towards a more distributed avionics architecture

Emilie BERARD[1,2], Christian FRABOUL[2], Fabrice LE SERGENT[1]
[1] Airbus Group Innovations, Dept.: Electronics, Communication & Intelligent Systems, Toulouse, France
[2]University of Toulouse IRIT-INPT/ENSEEIHT, Toulouse, France

Abstract**—In current modular avionics architectures (IMA), core processing modules (CPM) are physically centralized into few avionics bays. The main objective is to share (large) CPM by many applications (classical robust partitioning problem). In many cases such modules also includes I/O capabilities (CPIOM) that can be used locally by the allocated applications. If we want to use smaller modules the applications will have to be distributed on a larger number of modules that don't necessary share I/O capabilities. Moreover a given (large) application may have to be split (parallelized) on different communicating modules. As a consequence networks architectures used for interconnecting an increased number of processing modules must take into account new communications needs. Such architectures have to be validated according to distributed and parallelized applications communication requirements. Such validated architectures can then be compared in order to select the best one in a given applications context.***

I. Introduction

Federated and Integrated Modular Architectures are often used in avionics context. In federated architecture, each application is developed independently of others, has dedicated and isolated computer resources which ensure a natural fence to fault propagation, and communicates with other applications thanks to unidirectional data buses. It entails raising the number of devices in function of applications, and freezing quickly the architecture with difficulties to modify it. In integrated architecture, computing units, called modules, are shared by several applications (each application is executed in a given partition context to allow robust spatial and temporal partitioning of the applications). Modules communicate thanks to avionics data-buses (ARINC 429) or avionics networks (AFDX) network (implementing virtual links simulating classical 429 links).

The need of air forced cooling imposes that all modules are physically centralized into few avionics bays. The aim of our work is to study the feasibility of new avionics architectures by proposing an increased distribution of devices and consequently of applications in the overall aircraft. In this paper, we present the benefits of increased distribution (part II) and the challenges (part III) this implies in order to highlight our future works (part IV). [4, 5, 6]

II. Benefits of increased distribution

Our work is motivated by the opportunity to work on a new architecture using smaller computing units which can be placed in unused area, do not require forced air cooling, but which cannot have enough capacity to receive more and more voluminous application which requests more and more resources. Thus there is an increased need for distributing the applications on a larger number of processing modules and moreover to split a given application into tasks that can be executed concurrently on different modules. Such an increased distribution implies supplementary communications over the data buses or networks. Applications also depend on Inputs/Outputs (I/Os). They may be centralized: they are directly connected to a computing unit as CPIOM (Core Processing Input/Output Module), but I/Os are allocated under constraints. They may be decentralized: I/Os go through gateway like IOM (Input/Output Module) or CRDC (Common Remote Data Concentrator) so as to reach computing units, but it would imply more traffic on the network.

III. Challenges of increased distribution

A. Integrated Modular Architecture

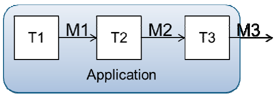
APEX, for APplication-EXecution, is an interface between a Real-Time Operating System (RTOS) and software programs which segregates temporally and spatially critical and non-critical functions thanks to partitions so as to not interfere between them. Each partition is activated periodically according to a MAjor Frame (MAF): it represents the activation scheme of all the partitions during the hyper-period of all of them contained in one core of the processor.

The communication between partitions leans on queuing or sampling ports. Two ports communicate together over the network thanks to Virtual Links (VLs) of an AFDX network. Message is encapsulated in a frame with its identifier; this one is used to find the appropriate path over the network to reach destination(s). In order to prevent interferences between traffics of several virtual links using the same physical link, all VLs are isolated by limiting the rate with the Bandwidth Allocation Gap (BAG) which is the minimum interval between two successive frames transmitted on the VL, and the maximum size of frames L_{max} that can be transmitted on one VL.

B. More distributed applications

The work detailed in [2] shows that it is easier to achieve shorter latencies but there is less flexibility on a centralized architecture than on a decentralized one. Moreover, the communication infrastructure has impact on latencies. Our work consists in comparing different architectures taking into account latency (the system answers quickly enough to a solicitation), freshness (data of the system depend on recent enough data to be pertinent) and coherence (some variables have to arrive within temporal window to be coherent) constraints according to the allocation of tasks and I/Os. [3]

In order to split an application into several computing units, the model proposed in [1] will be used. It consists in decomposing an application into a set of periodic tasks as detailed in Figure 1: this is composed of several periodic tasks with four real-time attributes: their period, their Worst Case Execution Time (WCET), their release time (Offset) and their deadline. Tasks communicate together according to their data-dependencies, which correspond to the messages M in Input and Output columns: if a task does not receive data in time, it cannot start.



Task	Period	WCET	Offset	Deadline	Input	Output	Required memory
T1	100	20	0	100		M1	200 Mb
T2	50	15	0	50	M1	M2	200 Mb
T3	50	15	0	50	M2	M3	100 Mb

Figure 1 - Application decomposition

Due to the fact that smaller computing units will be used, all tasks will not be located within one unit. Spatial allocation has to be done. Figure 2 shows examples of possible spatial allocation: tasks will be allocated according to the processors' capacity and the memory that tasks need.

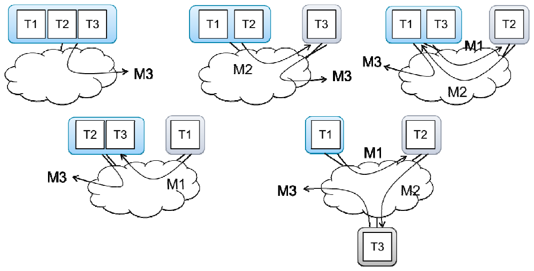


Figure 2 - Example of spatial allocation

Each task will be located into one partition within computing unit where it is located. But all partitions have to be schedulable: the MAF frame has to be built thanks to the real-time attributes of tasks contained into each core of processors as illustrated on Figure 3 (temporal allocation).

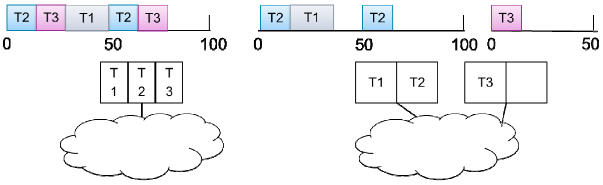


Figure 3 - Example of temporal allocation

The increasing distribution of applications also impact communication needs (as an example more VLs have to be mapped over an AFDX network) to connect the increased number of partitions. The spatial allocation of partitions may have important consequences: according to the data volume to be transferred, it may be preferable to gather some tasks in one computing unit or in another network and so avoid overconsumption of bandwidth. Moreover communication architecture can be composed of different kind of communication buses or networks (ARINC 429, AFDX: Avionics Full DupleX, AVB: Audio-Video Bridging or CAN: Controller Area Network). Proposed heterogeneous networks architecture must meet the requirements of avionics data flows such as bandwidth, latencies and jitters.

Finally, I/Os handling have also important consequences. As soon as I/Os are centralized to computing unit like CPIOM, the allocation of tasks is under constraint: indeed, tasks will be placed where connection with I/O is done; so the architecture is frozen as soon as it is implemented and then it is difficult to modify it. To decentralize I/Os, they may be connected directly to the backbone or thanks to dedicated I/O devices (IOM) or concentrators (CRDC). There is here less allocation constraints for applications but additional communication costs have to be computed and verified.

IV. WORK UNDERTAKEN

The objective of our work is to evaluate the trade-off between the benefits of an increased distribution of target physical architectures and the additional communication costs of the allocated distributed applications.

The main problem is to verify that a given target architecture respects distributed applications constraints. We must, in the first step, be able to describe and characterize processing and communication architectures (physical architecture). The second step is to describe the decomposition of each application into tasks (applications architectures). And the third step deals with the allocation (processing units and networks) of the applications on the architecture. Thus, a given allocation of given applications on a given architecture has to be evaluated in order to validate requirements of distributed applications.

Specific modeling tools are thus needed to describe a given configuration (architecture, applications, allocation description). Worst case end-to-end delay analysis has to be used for validating a given configuration. Moreover different valid solutions can be then compared using simulation tools.

We propose, during the RTN workshop, to illustrate this approach on a representative case study example.

REFERENCES

[1] Forget, J.; Boniol, F.; Grolleau, E.; Lesens, D.; Pagetti, C., "Scheduling Dependent Periodic Tasks without Synchronization Mechanisms," *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE* , pp.301 - 310, 12-15 April 2010

[2] Kanajan, S.; Pinello, C.; Haibo Zeng; Sangiovanni-Vincentelli, A., "Exploring Trade-off's Between Centralized versus Decentralized Automotive Architectures Using a Virtual Integration Environment," *Design, Automation and Test in Europe, 2006. DATE '06. Proceedings* , vol.1, pp.1 - 6, 6-10 March 2006

[3] Lauer, M.; Ermont, J.; Boniol, F.; Pagetti, C., "Latency and freshness analysis on IMA systems," *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on* , pp.1 - 8, 5-9 Sept. 2011

[4] Watkins, C.B.; Walter, R., "Transitioning from federated avionics architectures to Integrated Modular Avionics," *Digital Avionics Systems Conference, 2007. DASC '07. IEEE/AIAA 26th* , pp.2.A.1-1 - 2.A.1-10, 21-25 Oct. 2007

[5] Moore, J., Chapter 33. Advanced Distributed Architectures, Digital Avionics Handbook, Second Edition - 2 Volume Set. Dec 2000

[6] Alena, R.L.; Ossenfort, J.P.; Laws, K.I.; Goforth, A.; Figueroa, F., "Communications for Integrated Modular Avionics," *Aerospace Conference, 2007 IEEE* , pp.1 - 18, 3-10 March 2007

Worst-case backlog for AFDX network with n-priorities

Rodrigo Coelho, Gerhard Fohler
Technische Universität Kaiserslautern, Germany
{coelho, fohler}@eit.uni-kl.de

Jean-Luc Scharbarg
Université de Toulouse - IRIT/INP-ENSEEIHT
jean-luc.scharbarg@enseeiht.fr

Abstract—**In most recent avionics systems, AFDX (Avionics Full Duplex Switched Ethernet) is the network used to replace the previously employed point-to-point networks. AFDX guarantees bandwidth reservations by means of virtual links which can be classified with two priority levels. AFDX compliant switches implement output buffers at each switch output port. The stored frames leave each output port according to a fixed priority FIFO policy. Overflow of these buffers must be avoided at all cost to prevent data loss. Although the AFDX standard determines the minimum buffer size dedicated to an output port, the actual length of each priority buffer, is a designer decision.**

Previous works address the worst case backlog of ADFX buffers of one and two priorities. In this work we assume an extended AFDX network in which virtual links can be classified into n-priorities and present the problem statement to compute an upper bound on the worst case backlog faced by each buffer of each output port in each switch of the network.

I. Introduction

In most recent avionics systems, the switched ethernet network AFDX [1] (ARINC 664 Part-7), is chosen to substitute the point-to-point connections of previous avionics distributed systems. Currently, AFDX offers a network bandwidth of 100Mbps and allows for bandwidth isolation among all network traffic by employing the concept of virtual links (*VL*).

A *VL* defines a logical path from one source end-system (ES) to one or more destination ESs. The physical route for each VL is statically defined at design time and therefore the switches traversed by each *VL* are known before run-time. The predictable behavior of AFDX is further ensured by the parameters *BAG* and S_{max}, respectively bandwidth allocation gap (minimum time interval between the transmission of two consecutive frames of a *VL*) and maximum frame size associated with a *VL*. AFDX further allows for the classification of *VL*s into two priority levels: high and low.

AFDX compliant switches perform store-and-forward. In order to cope with contention for the switches output ports, each output port offers one FIFO queue per priority level. Due to the fixed priority FIFO scheduling at the output ports, switches have to send all frames with high priority before the low priority ones. Considering the non-preemptive property of frame scheduling, a switch cannot abort the transmission of a lower priority frame in favor of a higher priority one.

The AFDX standard specifies the minimum number of frames that must be buffered on the switches output ports. However, the actual output port buffer size for each priority level is left as a design decision and is used in the configuration phase of the network (section 4.7.3.2 of [1]). Thus, in order to avoid buffer overflow at the output ports and consequently packet loss, the designer must compute an upper bound for the backlog of each priority buffer.

Previous works address the computation of upper bounds for the worst case backlog of ADFX buffers considering virtual links with one and two priority levels. In this work we consider an extended AFDX network where the number of priorities assigned to *VL*s is unlimited (n-priority levels instead of two) to present the problem statement and the challenges for the computation of an upper bound for the backlog of each priority buffer on each output port of each switch of the network.

II. Related Work

[2] presents how to compute probabilistic bounds on buffer backlogs, based on stochastic network calculus (NC). [3] shows how NC leads to pessimistic results when compared to those achieved by trajectory approach (TA).

In contrast to NC, which considers each *VL* as a flow, TA analyzes the VL traffic at a finer granularity, accounting for the individual frames of the *VL*s. In [4] and [5] the authors make use of TA to compute the *e2e delay* for FIFO output buffers with *single priority* and *distinct static priority flows* respectively. [3] computes the worst case *backlog* for FIFO output buffers with *single priority flows*. and extends the previously mentioned works presenting an analysis of the pessimism intrinsic to TA.

Preliminary results for the buffer backlog of AFDX networks with two priority flows have been presented in [6]. We extend this analysis and present the analysis for the computation of an upper bound of the worst case buffer backlog assuming an AFDX network with n-priorities.

III. Worst case scenario and computation of backlog upper bound

We compute an upper bound for the worst case backlog for any output port buffer in three steps: first, we identify all *VL*s competing for the output port, second we compute the number of frames of the competing *VL*s that impact the worst case backlog of the buffer under study, and third we determine the worst case arriving sequence for these frames and compute an upper bound for the worst case backlog.

In this paper we consider AFDX virtual links with n-priorities. Consequently, we assume that n buffers, one buffer for each priority, exist on each switch output port (similarly to what [1] defines for two priorities).

Identifying the competing virtual links is straightforward: the routes used by the virtual links are computed off-line and do not change during run time. To compute the number of frames from the competing virtual links that impact the backlog encountered by each frame (f_m) with same priority as the buffer under study, we make use of the trajectory approach (TA) [4]. In principle, any other method that provides the number of competing frames can be used in our analysis.

Studying the worst case arriving sequence for the competing frames is the main contribution of this paper towards the worst case backlog computation. We first classify the competing virtual links into three sets: *VL*$^{\mathrm{S}}$, *VL*$^{\mathrm{H}}$ and *VL*$^{\mathrm{L}}$, respectively with virtual links of same (*sp*), higher (*hp*) and lower (*lp*) priority than the buffer under study. Then we analyze the impact of each of these sets into each frame of *VL*$^{\mathrm{S}}$.

We start our analysis presenting limit values for the worst case backlog upper bound. Further, we expand this analysis to compute a tighter upper bound.

A. Worst case backlog upper bound limits

If we create an imaginary scenario where the frames of the same priority set (*VL*$^{\mathrm{S}}$) are the only frames on the network, an upper bound for the worst case buffer backlog of the buffer under analysis (Buffer$^{\mathrm{S}}$) can be computed as presented in [3] and named here as bl_{max}^{onlyS}. Computing an upper bound for the worst case backlog considering all frames (of *VL*$^{\mathrm{S}}$, *VL*$^{\mathrm{H}}$ and *VL*$^{\mathrm{L}}$), can only lead a value larger than or equal to bl_{max}^{onlyS}.

The backlog of Buffer$^{\mathrm{S}}$ can never be larger than the sum of the sizes of all frames in *VL*$^{\mathrm{S}}$.

Thus, the computed upper bound for the worst case backlog for a buffer of a given priority is limited by:

$$bl_{max}^{onlyS} \leq bl_{max}^{S} \leq \sum_{\forall f_k^S \in VL^{\mathrm{S}}} s(f_k^S) \tag{1}$$

where the function $s(f)$ represents the length of a frame f.

B. Worst case backlog scenario

Figure 1 presents the arrival of *sp* and *hp* frames and how they are scheduled at the output port according to the fixed priority FIFO policy. Figure 1 further shows the backlog of the buffer under analysis (Buffer$^{\mathrm{S}}$). In this figure, frames arrive from four input links and compete for the same output port. Red frames have the same priority as the buffer under analysis (elements of *VL*$^{\mathrm{S}}$ set) and blue frames are those with higher priority (elements of *VL*$^{\mathrm{L}}$ set). The impact of lower priority frames will be considered later.

Next we present the meaning of the points in time and time lengths depicted in the figure:

- α: start of transmission of *hp* frames
- β: end of transmission of *hp* frames
- θ_{sp}: end of reception of *sp* frames
- Δ: $\theta_{sp} - \beta$

For the sake of simplicity and without loss generality, we assume that $s(f)$ units of time is the amount of time required to send a frame of length $s(f)$.

The worst case backlog faced by Buffer$^{\mathrm{S}}$ occurs after all *sp* frames arrive and the access of *sp* frames to the output port suffers the largest delay. Therefore, we compute bl_{max}^{S} at θ_{sp} and construct a scenario in which all frames from *VL*$^{\mathrm{H}}$ and *VL*$^{\mathrm{L}}$ delay the dispatch of *sp* frames the longest.

According to Figure 1, the computation of the worst case backlog for Buffer$^{\mathrm{S}}$ can be presented as the sum of all frames in *VL*$^{\mathrm{S}}$ minus the amount of data transmitted during the time interval Δ, i.e.:

$$bl_{max}^{S} = \sum_{\forall f_k^S \in VL^{\mathrm{S}}} s(f_k^S) - \mid \Delta \mid \tag{2}$$

An intuitive approach to construct a scenario that leads to the worst case backlog of Buffer$^{\mathrm{S}}$, based on previous analysis for single priority AFDX, is to assume that all *hp* frames arrive before the *sp* frames (see Figure 1). Further, frames arrive in decreasing order of size within the same set to avoid idle times at the output link [5].

According to equation (2), in order to achieve the worst case backlog for Buffer$^{\mathrm{S}}$, the arrival sequence of the frames should be such that $\mid \Delta \mid$ is minimum. Intuitively, in order to compute the shortest Δ (which is equivalent to the longest β since θ_{sp} is constant), we should compute the latest α (α_{max}) and assume that the output link will be busy with all *hp* frames until β, i.e.:

$$\beta = \alpha_{max} + \sum_{\forall f_k^H \in VL^{\mathrm{H}}} s(f_k^H) \tag{3}$$

as depicted in Figure 1. In this example $\alpha = 70$, $\sum_{\forall f_k^H \in VL^{\mathrm{H}}} s(f_k^H) = 671$ and therefore $\beta = 741$.

However, neither the scenario presented in Figure1 nor the equation (3) leads to the largest β for every set of frames.

Figure 2 presents a scenario in which one frame of *VL*$^{\mathrm{S}}$ is shorter than in Figure 1. In this case, the set *VL*$^{\mathrm{H}}$ remains unchanged and so does the result of equation (3). Nevertheless, in Figure 2, β is larger than 741, in fact $\beta = 771$ due to an idle time of 30 units of time at the output link.

Figure 3 presents a scenario with the same frames as in Figure 2, but a different arriving sequence for the second input link: one *lp* frame arrives before the sequence of *sp* frames. Again, the result of equation (3) is 741 but the actual β is equal to 812, even larger than the one of the scenario presented in Figure 2 because of the larger idle time at the output link.

We can conclude that, although the frames arrive in decreasing order of lengths per set (*VL*$^{\mathrm{S}}$, *VL*$^{\mathrm{H}}$, *VL*$^{\mathrm{L}}$), they do not arrive in decreasing order of lengths if we consider all sets together. Therefore some idle time may be present at the output link and consequently equation (3) does not hold. We propose equation (4) to account for this idle time:

$$\beta_{max} = \alpha_{max} + \sum_{\forall f_k^H \in VL^{\mathrm{H}}} s(f_k^H) + idle_{max} \tag{4}$$

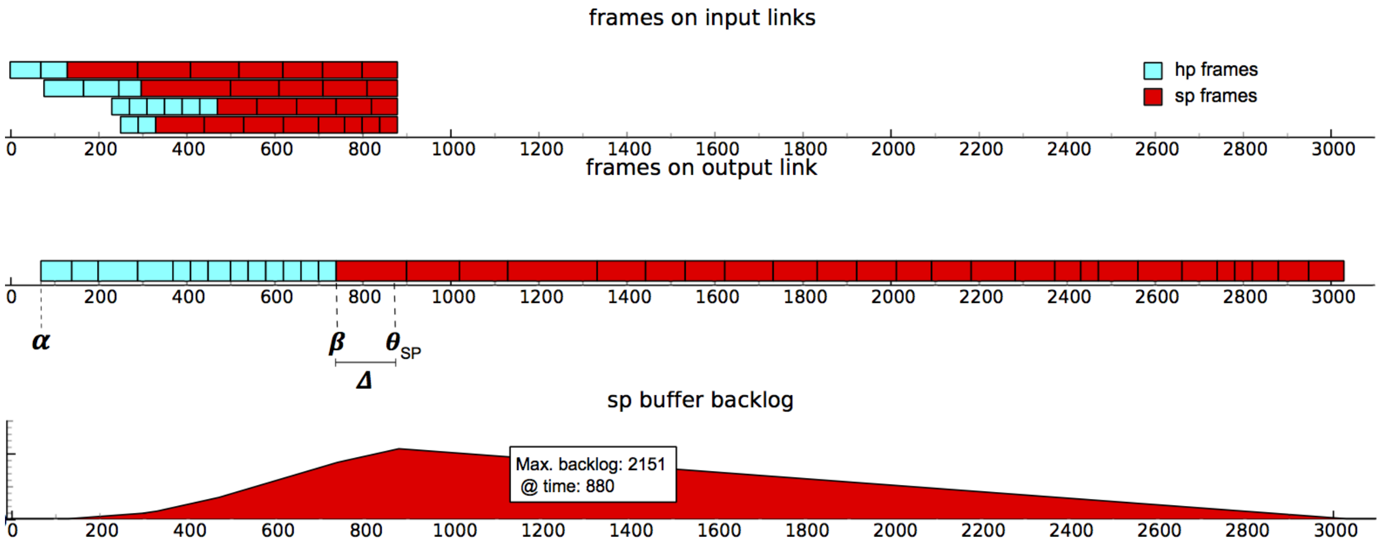


Fig. 1: Intuitive approach for the worst case scenario and computation of the worst case backlog for Buffer^S.

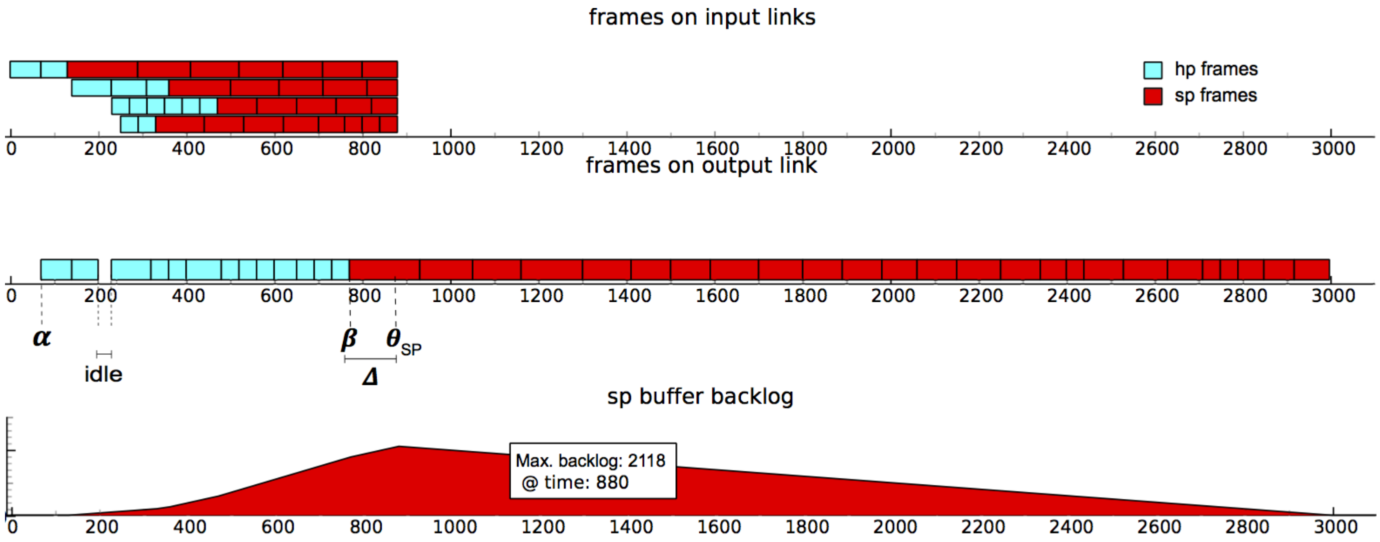


Fig. 2: Small change on previous scenario (one shorter *sp* frame) leads to idle time at the output link and β larger than the one computed by equation (3).

According to equation (4), we must achieve two goals in order to compute β_{max}:

1) compute the maximum idle time at the output link due to the non-decreasing arrival order of frames $idle_{max}$.
2) compute the latest point in time in which the set of *hp* frames start transmission (α_{max}) such that no *sp* frame is transmitted before α_{max}.

Additionally, we must take into account the impact of lower priority frames.

IV. Conclusion and Future Work

In this paper we presented the problem statement and the challenges for the computation of an upper bound for the backlog of each priority buffer on each output port of each switch of an extended AFDX network with n-priority levels (instead of two).

The challenges to compute an upper bound for the worst case backlog of each buffer, within the limits presented in (1), is summarized as follows:

- prove that equation (4) holds for any sequence of arriving frames
- compute α_{max}
- compute $idle_{max}$
- compute the impact of *lp* frames into the worst case backlog of Buffer^S

Obviously, the computation of an upper bound for the worst case backlog of output buffers for the current AFDX network (with two priorities) is a sub problem of the one presented in this paper and can, therefore, be achieved by assuming $n = 2$.

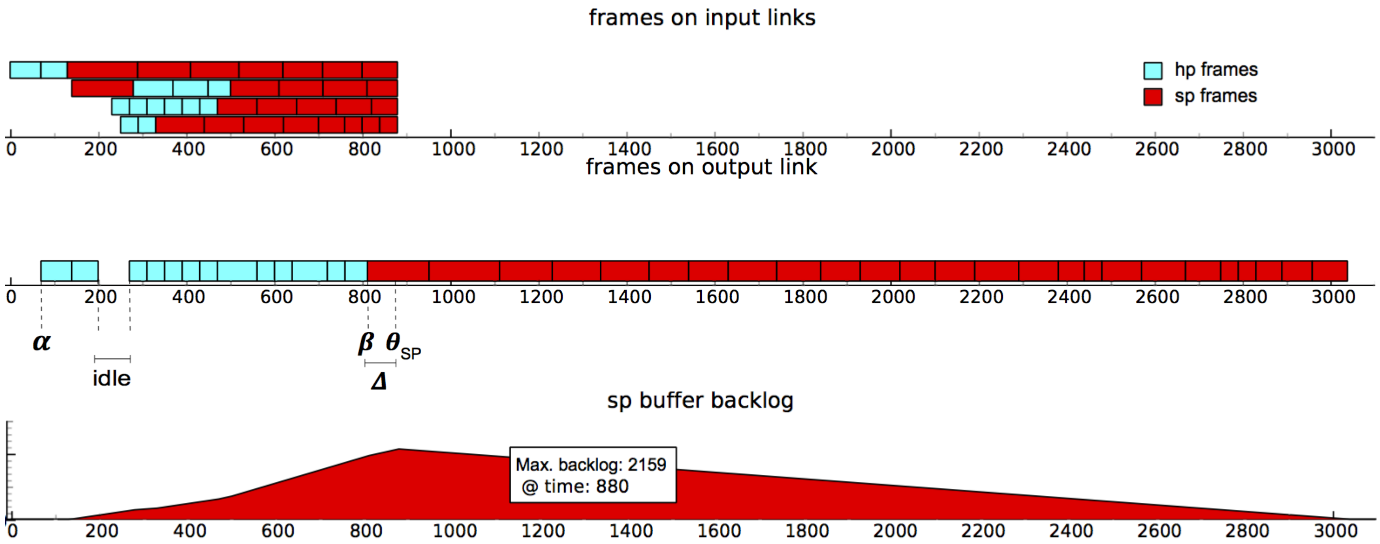


Fig. 3: One *sp* frame arrives first on the second input link leading to larger idle time at the output link and thus larger values of β.

References

[1] "ARINC specification 664 P7-1. Aircraft Data Network Part-7 Avionics Full-Duplex Switched Ethernet Network," September 2009.

[2] F. Ridouard, J.-L. Scharbarg, and C. Fraboul, "Stochastic network calculus for buffer overflow evaluation in an avionics switched ethernet," in *Junior Researcher Workshop on Real-Time Computing*, March 2007, pp. 55–58.

[3] H. Bauer, J.-L. Scharbarg, and C. Fraboul, "Worst-case backlog evaluation of avionics switched ethernet networks with the trajectory approach," in *ECRTS*, July 2012, pp. 78 –87.

[4] H. Bauer, J. Scharbarg, and C. Fraboul, "Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network," in *ETFA*, 2009, pp. 1–8.

[5] H. Bauer, J.-L. Scharbarg, and C. Fraboul, "Applying trajectory approach with static priority queueing for improving the use of available AFDX resources," *Real-Time Systems*, vol. 48, no. 1, pp. 101–133, January 2012.

[6] N. R. Garikiparthi, R. F. Coelho, and G. Fohler, "Calculation of worst case backlog for afdx buffers with two priority levels using trajectory approach," in *12th Workshop on Real-time Networks (RTN13) in conjuction with 25th Euromicro International Conference on Real-time Systems (ECRTS13)*, July 2013.

Session 3 - Invited talk

Information processing for extreme dense sensing: timeliness and scalability issues

Eduardo Tovar
CISTER/INESC-TEC, ISEP,
Polytechnic Institute of Porto, Porto, Portugal

Abstract

Large-scale and dense sensor/actuator deployments pose fundamental challenges concerning both interconnectivity and processing of huge quantities of information. Think about the simple example of obtaining the minimum value among cents of sensor readings. Or think about the more sophisticated active flow control application where through proper modulation of aircraft skin surfaces a significant reduction of drag and related fuel consumption (and emissions) may be attained. Currently available approaches for data processing in such large-scale very dense deployments of sensors lead to energy-waste and long response-times from sensing to actuation. This talk will address emerging techniques that are able to allow scalable and efficient data processing in large-scale dense cyber-physical systems, where cents of nodes may coexist within the same broadcast domain.

Session 4

Low Level Error Detection For Real-Time Wireless Communications

Jeferson L. R. Souza and **José Rufino**
Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal
LaSIGE - Navigators Research Team
Email(s): jsouza@lasige.di.fc.ul.pt, ruf@di.fc.ul.pt

***Abstract*—The use of wireless networks to support communications with real-time restrictions is becoming a common requirement within environments such as industries, autonomous vehicles, and aerospace technologies, including also the support for cyber-physical systems (CPS). An effective real-time support on the wireless realm is still an open issue, relying on the presence of dependable and fault tolerant communication services, which are built upon fundamental mechanisms such as error detection. In this paper we continue to explore the low levels of the networking protocol stack in the design of a robust foundation to efficiently support real-time on wireless networks; focused on low level error detection, we present the innovative idea to combine error protection mechanisms for enhancing the capabilities and accuracy of a wireless node in the detection of node failures. This paper shows how our low level error detection approach can be utilised to detect and differentiate node transient omission failures, node permanent failures (e.g., in the transmitter circuitry), and node crash failures in wireless communications.**

***Index Terms*—error detection; wireless communications; real-time; dependability; timeliness; wireless sensor and actuator networks**

I. Introduction

The provision of real-time guarantees on wireless communications is still an open challenge, which is derived by the lack of a robust abstract communication model and its related supporting technologies offering resilient and real-time communication services, even in the presence of disturbances, such as transient overload and network errors [1], [2].

In wired networks, enhancements in the lowest levels of the networking protocol stack had proven to be highly effective in dealing with network errors and in the provision of real-time communication services [3], [4]. A similar strategy to enhance low levels of the networking protocol stack can also be utilised in the wireless realm, where innovative solutions towards the provision of resilient real-time wireless communications are being designed and developed [1], [2], [5].

Though such solutions are restricted to one-hop wireless network segments, the benefits from that approach can be easily and effectively extended to multi-hop settings with strict temporal restrictions, such as those found (in general) in cyber-physical systems (CPS), having a positive impact in the analysis of end-to-end message schedulability guarantees [6], [7], [8] and in ensuring determinism and bounded timeliness properties in multi-hop networking communications.

Error detection plays a fundamental role in the provision of reliable networking communications, being the frame check sequence (FCS) mechanism the first error filter applied by the medium access control (MAC) sublayer to verify the integrity of incoming frames. FCS is usually a silent filter that discards erroneous frames without any notification; the absence of such error notifications hides details concerning the state of the communication channel and node error patterns.

Traditional approaches to reliable networking communications often disregard the potential of low level error detection. Assuming networking communication models where the lowest levels are unreliable, mechanisms to deal with the failure of networking components (wireless nodes included) are routinely designed using timeout-based approaches, which have just to rely on the accurate dimensioning of timers to avoid violations of the timeliness properties of the entire system.

Without diminishing the usefulness of timers and timeout-based mechanisms, this paper builds upon an extension to the standard FCS mechanism, which adds the ability of providing a management notification if an erroneous frame is received. This simple yet innovative method has proved effective in the detection of communication channel failures [5]. This paper extends such raw mechanism with three new features: with a very high coverage, the ability to extract correct information (e.g., source node address) from an erroneous frame; the ability to detect the permanent failure of source node transmit machinery (such as the FCS generator malfunction); and the ability to detect node's crash failures.

We believe that these new mechanisms will be of fundamental importance to design and develop highly effective node failure detection and membership services, traditionally non-existent in standard wireless technologies, yet instrumental to the provision of reliable real-time communication services [1].

To present our contributions this paper is organised as follows: Section II presents the system model and the detailed description of an abstract communication model dubbed wireless network segment (WnS), which is suitable for one-hop real-time wireless communications; Section III presents our proposed low level error detection mechanisms, and its benefits for error detection in resilient real-time wireless

This work was partially supported by the EC, through project IST-FP7-STREP-288195 (KARYON); by FCT/DAAD, through the transnational cooperation project PROPHECY; and by FCT, through project PTDC/EEI-SCR/3200/2012 (READAPT) and through LaSIGE Strategic Project PEst-OE/EEI/UI0408/2014.

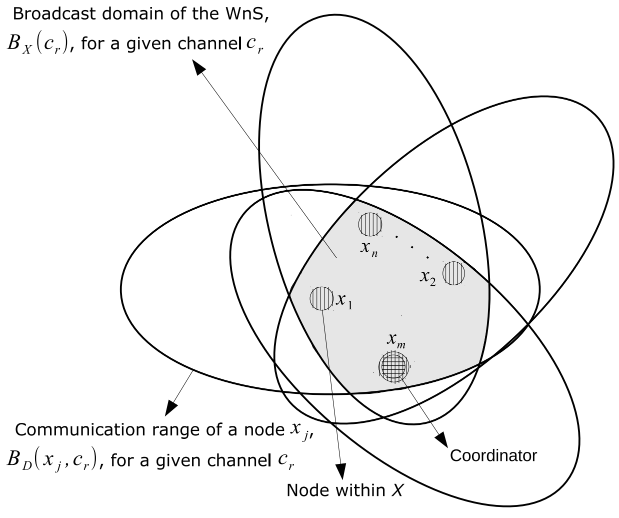

Broadcast domain of the WnS,
$B_X(c_r)$, for a given channel c_r
x_n
x_2
x_1
x_m
Communication range of a node x_j,
$B_D(x_j, c_r)$, for a given channel c_r
Coordinator
Node within X

Fig. 1. A graphical representation of the WnS communication model

communications; finally, Section IV presents the conclusions and future directions of the work presented in this paper.

II. System Model and The Wireless Network Segment Approach

All networking communications described within this paper are subjected to the characteristics of a data link layer communication model dubbed Wireless network Segment (WnS). The WnS is a broadcast network segment where communication devices are able to communicate directly and sense transmission from each other (one-hop distance). The communication devices are dubbed networking nodes, which are devices with capability to communicate throughout a shared wireless transmission medium. The terms networking node, wireless node, or simply node, are used interchangeably in the remaining of the paper.

The utilisation of the WnS communication model, introduced in [1], is herein extended and formalised. A formal definition of the WnS is expressed by a 4-Tuple, $WnS = \langle X, x_m, C, W \rangle$, where X is the set of the wireless nodes members of the WnS; x_m is the WnS coordinator, $x_m \in X$; C represents a set of communication channels; and W represents the set of access modes utilised to access the network within the WnS.

The set of WnS members is defined by $X = \{x_1, \ldots, x_n\}$, where the cardinality $\#X$ represents the number of nodes within the WnS. All communications are performed through a set of non-overlapping communication channels, $C = \{c_1, \ldots, c_z\}$, where each $c_r \in C$ is a unique channel, being $1 \leq r \leq z$. In case of a WnS using only one channel, $z = r = 1$, i.e., $\#C = 1$. The access to the network is characterised by a set of access modes $W = \{w_1, \ldots, w_s\}$, where each $w_v \in W$ is utilised by nodes, to control the network access for each channel $c_r \in C$, being $1 \leq v \leq s$, and defining its timing characteristics through φ_{w_v}. Examples of access modes include carrier sense with collision avoidance (CSMA/CA) and time division multiple access (TDMA).

Every node $x_j \in X$ is included in the set of recipients of each transmitted frame, for each channel $c_r \in C$. The broadcast domain of the WnS, for a given channel $c_r \in C$, is defined by: $B_X(c_r) = \bigcap_{j=1}^{\#X} B_D(x_j, c_r), \ \forall x_j \in X$, where $B_D(x_j, c_r)$ is a geographic region that represents the communication range of a node x_j for a given channel c_r.

Let $P(x_j, c_r)$ represent the geographic position of node x_j transmitting on channel c_r. A node $x_j \in X$ if, and only if, $\exists c_r \in C$ where $P(x_j, c_r) \subseteq B_X(cr)$. Otherwise, as a consequence of node mobility, a node $x_j \notin X$ if, and only if, $\forall c_r \in C$, $P(x_j, c_r) \nsubseteq B_X(cr)$.

Figure 1 illustrates a graphical representation of the WnS, where the ellipses characterise the communication range of nodes for a given channel $c_r \in C$, being the grey area the characterisation of the broadcast domain for a given WnS. In practice, the communication range of each node may assume irregular and complex forms [9].

A. Fault Model

Networking components (e.g., a node $x_j \in X$, or a channel $c_r \in C$) either behave correctly or crash upon exceeding a given number of consecutive omissions (the component's *omission degree bound*), f_o, following a given observation criteria (e.g., the duration of a given protocol execution, $\mathcal{T}_{rd}$). Omission faults may be inconsistent (i.e., not observed by all recipients).

In the context of networking communications, we define an omission as an error that destroys a frame. In this sense, errors derived from the presence of accidental faults are transformed into omissions, which are accounted for the purpose of monitoring networking components at different levels. For each received frame, every node $x_j \in X$ locally accounts the omissions observed at each level.

Despite of their importance we are not considering the presence of intentional faults in our fault model, being such topic addressed properly in future work.

B. WnS abstract channel properties

The characteristics of the low level layers in the wireless networking protocol stack can be abstracted by a set of correctness, ordering, and timeliness properties, which are in essence independent of each particular networking technology. In our WnS abstraction such properties are offered through the facet of an abstract single communication channel we dubbed WnS abstract channel, as illustrated in Fig. 2. A relevant set of WnS abstract channel properties is defined in Fig. 3.

Property WnS1 (*Broadcast*) formalises that it is physically impossible for a node $x_j \in X$ to send conflicting information (in the same broadcast) to different nodes, within the broadcast domain of the WnS [10], $B_X(c_r)$, for a given channel $c_r \in C$.

Properties WnS2 (*Frame Order*) and WnS3 (*Local Full-Duplex*) are common in network technologies, wireless technologies included. Property WnS2 (*Frame Order*) is imposed by the wireless transmission medium of each channel $c_r \in C$,

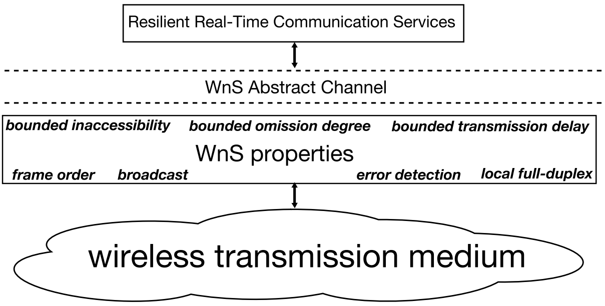


Fig. 2. WnS abstract channel

WnS1 - ***Broadcast***: correct nodes, receiving an uncorrupted frame transmission, receive the same frame;

WnS2 - ***Frame Order***: any two frames received at any two correct nodes are received in the same order at both nodes;

WnS3 - ***Local Full-Duplex***: a correct node may receive, on request, local frame transmissions;

WnS4 - ***Error Detection***: correct nodes detect and signal any corruption done during frame transmissions in a locally received frame;

WnS5 - ***Bounded Omission Degree***: in a known time interval $\mathcal{T}_{rd}$, omission failures may occur in at most k transmissions;

WnS6 - ***Bounded Inaccessibility***: in a known time interval $\mathcal{T}_{rd}$, a wireless network segment may be inaccessible at most i times, with a total duration of at most $\mathcal{T}_{ina}$;

WnS7 - ***Bounded Transmission Delay***: any frame transmission request is transmitted on the wireless network segment, within a bounded delay $\mathcal{T}_{td} + \mathcal{T}_{ina}$.

Fig. 3. WnS communication properties

and results directly from the serialisation of frame transmissions on the shared wireless transmission medium. Property WnS3 (*Local Full-Duplex*) specifies that the sender itself is also included in that ordering property, as a recipient.

Property WnS4 (*Error Detection*) has both detection and signalling facets; the detection facet, traditionally provided by the MAC sublayer, derives directly from frame protection through a frame check sequence (FCS) mechanism, which most utilised algorithm is the cyclic redundant check (CRC); the signalling facet is provided by the FCS extension introduced in [5], which is able to signal omissions detected in frames received with errors. No fundamental modifications are needed to the wireless MAC standards, such as IEEE 802.15.4 [11]. The use of such unconventional extension is enabled by emerging controller technology, such as reprogrammable technology and/or open core MAC sublayer solutions, such as the transceivers and the MAC sublayers developed by ATMEL [12]. The residual probability of undetected frame errors is negligible [13], [14].

Property WnS5 (*Bounded Omission Degree*) formalises the failure semantics introduced earlier in the fault model definition, being the abstract omission degree bound, $k \geq f_o$. The omission degree of a wireless network segment can be bounded, given the error characteristics of its wireless transmission medium [14], [15], [16].

The *Bounded Omission Degree* property is one of the most complex properties to secure in wireless communications. Securing this property with optimal values and with a high degree of dependability coverage may require the use of multiple channels. In [5] we have advanced on how this can be achieved by monitoring channel omission errors, and switch between channels upon detecting that the channel omission degree bound has been exceeded.

The time domain behaviour of a WnS is described by the remaining properties. Property WnS7 (*Bounded Transmission Delay*) specifies a maximum frame transmission delay, which is $\mathcal{T}_{td}$ in the absence of faults. The value of $\mathcal{T}_{td}$ includes the medium access and transmission delays and it depends on message latency class and overall offered load bounds [17], [18]. The value of $\mathcal{T}_{td}$ does not include the effects of omission errors. In particular, $\mathcal{T}_{td}$ does not account for possible frame retransmissions. However, $\mathcal{T}_{td}$ may include extra delays resulting from longer WnS access delays derived from subtle side-effects caused by the occurrence of periods of network inaccessibility [16].

A period of network inaccessibility is a disturbance that may be induced externally by electromagnetic interference, or by glitches in the MAC sublayer operation, such as those that may result from the omission of a MAC control frame (e.g., beacon). The network cannot be considered failed; it only enters into a temporary state where the communication service is not provided to some or all of the nodes. Hence, nodes may experience a loss of connectivity within a WnS; the loss of connectivity due to node mobility is also treated under the inaccessibility model. Therefore, the bounded transmission delay includes $\mathcal{T}_{ina}$, a corrective term that accounts for the worst-case duration of inaccessibility glitches, given the bounds specified by property WnS6 (*Bounded Inaccessibility*). The inaccessibility bounds depend on, and can be predicted by the analysis of MAC sublayer characteristics [16].

III. Low Level Error Detection For Real-Time Wireless Communications

The ability to detect errors occurred on networking transmissions is the foundation to provide reliable wireless communications, and therefore to establish a set of useful services for distributed real-time systems such as node failure detection and node membership (abstractly represented by the set X of the WnS). This section focus in presenting some fundamental low level error detection mechanisms that are utilised to augment the error detection capabilities of wireless nodes.

A. Enabling low level enhancements for error detection

Enabling low level enhancements for error detection includes the ability to know when a frame has been received, even if such frame contain errors.

The FCS extension presented in [5] introduces minor but fundamental modifications to the standard frame processing, which are essential to secure the signalling facet of the *Error Detection* property of the WnS. For self-containment purposes,

Algorithm 1: The FCS extension presented in [5]

```
1  Initialisation phase;
2  $fcs\_error$ ←false;
3  $timestamp \leftarrow -\infty$;
4  begin
5    loop
6      when Channel.indication(frame) do
7        $timestamp$ ← MAC.readClock();
8        $frame\_header$ ←MAC.get.header(frame);
9        if MAC.FCS.check(frame) is OK then
10         $fcs\_error$ ← false;
11         MAC.indication(frame) ;
12       end
13       else
14         $fcs\_error$ ← true;
15         MAC.frame.discard(frame) ;
16       end
17       MAC.Mgmt-Ext.indication($timestamp$, $frame\_header$, $fcs\_error$) ;
18     end when
19   end loop
20 end
```

let us briefly present the FCS extension [5] that is reproduced in Algorithm 1.

When a frame is received (line 6), the frame header is extracted (first highlighted modification at line 8), and the FCS check is performed at line 9. Should a frame be received without errors, the standard frame processing procedure is followed and a correct frame is delivered above MAC (line 11), at the WnS abstract channel interface (Fig. 2). If a frame is received with errors it will be simply discarded (line 15) and no data is delivered at the WnS abstract channel interface. The extensions introduced in Algorithm 1 to the standard FCS mechanisms specify that a notification is delivered at the MAC management interface, as highlighted at line 17. This means it is possible for a wireless node to be aware when it is receiving frames with errors through a given channel $c_r \in C$. Algorithm 1 specifies that a relevant part of a received frame (e.g., the frame header) is delivered at the management interface for further processing. However, one question is whether this information is useful in the event a frame has been corrupted by errors?

Surely, the simple notification that an error (transformed into an omission) has occurred is useful. In [5] we have exploited this feature to secure an innovative channel selection function. In that case, a given node $x_j \in X$ is able to issue management notifications that are used for channel monitoring purposes: channel omission failures are detected and accounted for; if the allowed number of successive channel omission failures is exceeded, a switch to a different channel is performed.

Nevertheless, the fundamental question remains: could we use the header information extracted from an erroneous frame?

B. Securing protection of relevant frame content

The use of FCS protecting the entire content of a frame as a single unit allows the detection of errors during networking communications with a very high probability [13], but one cannot know which part of the frame was corrupted, and therefore is not possible to know which bits of a frame were modified. As the corruption may be in any part of the received frame, if one wants to extract important information from a frame received with errors, the specific content to be extracted has to be protected by an additional integrity verification mechanism.

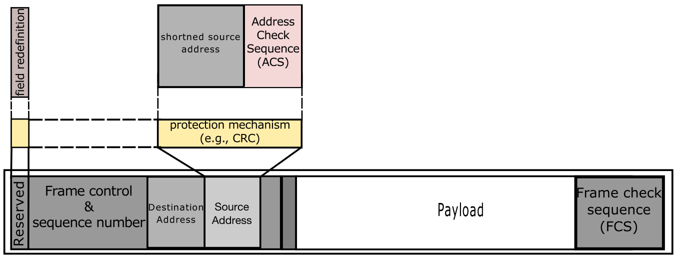

field redefinition
shortned source address
Address Check Sequence (ACS)
protection mechanism (e.g., CRC)
Reserved
Frame control & sequence number
Destination Address
Source Address
Payload
Frame check sequence (FCS)

Fig. 4. Representation of our additional integrity verification mechanism for particularly important frame contents (partial frame header protection)

How our proposed integrity verification mechanisms works: Use case of partial frame header protection

The protection provided by the integrity verification mechanism can be applied to any important part of the frame content. In an abstract perspective, the introduction of an additional integrity verification mechanism for important contents of a frame is similar to put an "extra shield" over such contents to protect them against external disturbances on the communication channel, particularly improving the dependability of its transfer.

For many networking standards, some frame formats can be extended or redefined to accommodate the required extra integrity verification data, without any change in the overall frame format, and therefore without any frame overheads.

To illustrate our approach let us use the protection of the source address and its related reserved control field as an example. In Fig. 4, a reserved control field is utilised to signal the source address field is now protected and shortened; and in the place of the source address a compound field is stored, consisting in a shortened version of the source address plus an additional integrity verification field dubbed address check sequence (ACS), which protects the source address and the reserved control field altogether. The total length of the standard source address field is unchanged, implying no extra overhead to the frame length.

In Fig. 4, when the integrity of the reserved control field and the shortened source address of a frame are not compromised, we are able to know which node transmitted the received frame, even if FCS check has failed; the additional protection of just few bits offered by ACS has the benefit of allowing a dependable extraction of important information from frames received with errors.

Algorithm 2 describes the ACS check procedure, which is utilised to materialise our dependable extra shield for practical purposes, being used as a complement of the FCS extension presented in [5]. We assume here all nodes, $\forall x_j \in X$, start to use our dependable extra shield mechanism after being confirmed as a member of a given WnS.

For all frames received and notified by the management

Algorithm 2: Frame header integrity verification

```
1  Initialisation phase;
2  $X$;                    // Representing the set of nodes members of the WnS.
3  $node\_id$;
4  $reserved\_field$;
5  begin
6      loop
7          when MAC.Mgmt-Ext.indication(timestamp, $frame\_header$, $fcs\_error$) do
8              $reserved\_field \leftarrow$ MAC.Mgmt-Ext.extractReserved($frame\_header$);
9              $node\_id \leftarrow$ MAC.Mgmt-Ext.extractNodeID($frame\_header$);
10             if $reserved\_field$ is ACS $\wedge$ MAC.ACS-Ext.check($frame\_header$) is OK $\wedge$ $node\_id \in X$ then
11                 MAC.Mgmt-Ext.indication(timestamp, $node\_id$, $fcs\_error$) ;
12             end
13         end when
14     end loop
15 end
```

Algorithm 3: Detecting node permanent failures

```
1  Initialisation phase;
2  $nodeOd[x_1, \dots, x_n] \leftarrow 0$;       // Omission failures for each node of the WnS.
3  $k_{perm}$; // Defines the threshold to detect a permanent node failure. A reasonable
   value for such bound has to be greater than the omission degree bound to detect
   channel failures [5].
4  begin
5      loop
6          when MAC.Mgmt-Ext.indication(timestamp, $node\_id$, $fcs\_error$) do
7              if $fcs\_error$ is true then
8                  $nodeOd[node\_id] \leftarrow nodeOd[node\_id] + 1$ ;
9                  if $nodeOd[node\_id] > k_{perm}$ then
10                     MLA.Mgmt.indication($node\_id$, $node\_perm\_failure$) ;
11                 end
12             else
13                 $nodeOd[node\_id] \leftarrow 0$ ;
14             end
15         end
16     end when
17 end loop
18 end
```

indication of line 7 (originated from the FCS extension), the format and integrity of the protected content is verified through the ACS field (line 10). If the integrity of any of the protected content has been not compromised, a management indication is generated on its turn (line 11) to inform the availability of such content (e.g., $node\_id$).

C. Detecting node permanent failures

Permanent failures are characterised by the consecutive reception of frames with errors from the same node. Assuming both ACS generation and checking procedures use methods that are not affected by a failure in the transmitter circuitry, a permanent node failure indicates a malfunction in the transmitter of that node, which can induce a bad FCS for outgoing frames from such transmitter. However, at a recipient, individual omission failures with origin in a transmitter node cannot be distinguished from omission failures with origin in the communication channel.

The value defined for k_{perm}, the threshold to detect a permanent node failure, must then include two distinct contributions: the allowed number of consecutive omissions with origin in the node being monitored, as produced by the node's transmitter circuitry, k_{txfail}; and the channel omission degree bound, k, as defined by property WnS5 (*Bounded Omission Degree*). Thus, $k_{perm} = k_{txfail} + k$.

Algorithm 4: Detecting node crash failures

```
1  Initialisation phase;
2  $node\_live\_period[x_1, \dots, x_n]$; // Aliveness period for each node of the WnS.
3  begin
4      loop
5          when MLA.Mgmt.indication($node\_id$, $max\_idle\_period$,
           $has\_joined\_WnS$) do
6              $node\_live\_period[node\_id] \leftarrow max\_idle\_period + \mathcal{T}_{td} + \mathcal{T}_{ina}$ ;
7              MLA.Mgmt.startTimer($node\_live\_period[node\_id]$, $node\_id$) ;
8          end when
9          when MLA.Mgmt.indication($node\_id$, $has\_left\_WnS$) do
10             MLA.Mgmt.stopTimer($node\_id$);
11         end when
12         when MAC.Mgmt-Ext.indication(timestamp, $node\_id$, $fcs\_error$) do
13             MLA.Mgmt.restartTimer($node\_live\_period[node\_id]$, $node\_id$) ;
14         end when
15         when MLA.Mgmt.indication($node\_id$, timer_expired) do
16             MLA.Mgmt.indication($node\_id$, $node\_crash\_failure$) ;
17         end when
18     end loop
19 end
```

Algorithm 3 has been designed to detect permanent failures of nodes within the WnS. In the context of Algorithm 3, an omission is accounted for a given node (line 8) by the reception of a frame with FCS error (line 7). When such node exceeds k_{perm}, a permanent failure is detected (line 9), and notified through a management notification (line 10). Otherwise, if a correct frame is received from a given node, $node\_id$, its omission degree is cleared (line 13).

D. Detecting node crash failures

Although earlier in this paper we have disregarded the utilisation of timeout-based approaches to detect transient and permanent node failures, the only possibility to detect node's crash failures (i.e., the absence of node's activity on the WnS) is using the notion of time. In concrete, we define a timeout-based approach (Algorithm 4) to detect the crash of nodes on the WnS, where timers are dimensioned consonant with the temporal behaviour of the WnS, which is dictated by the WnS properties WnS6 (*Bounded Inaccessibility*) and WnS7 (*Bounded Transmission Delay*).

In Algorithm 4, a node starts to be monitored (line 7) when it joins the WnS (line 5). To be able to detect node crash failures we assume each node of a given WnS, $x_j \in X$, has to inform other nodes of its maximum idleness period (represented by the max_idle_period parameter), which is utilised to monitor that node activity; the interval of the idleness period of a given node can be derived, for example, from the periodic transmission of heartbeat control frames for membership maintenance.

For each node $x_j \in X$ (represented by $node\_id$) a timer is started (line 7) with a timeout value (line 6) equal to the maximum time interval between two consecutive frame transmission requests from a given node plus the worst case time required to transmit the frame, which is $\mathcal{T}_{td} + \mathcal{T}_{ina}$, as specified by the WnS7 property (*Bounded Transmission Delay*).

The bounded transmission delay of the WnS (property WnS7) accounts for the worst case transmission delay, $\mathcal{T}_{td}$, and for the presence of periods of network inaccessibility (property WnS6 of the WnS), being the value assumed for $\mathcal{T}_{ina}$ the worst case duration derived from the network technology utilised.

For each frame received from a given node represented by $node\_id$ (line 12), the timer instantiated to monitor the activity of such node is restarted (line 13). If a node crashes, no more frame transmissions are received from that node and the corresponding timer expires (line 15). The crash of $node\_id$ is declared through a management notification (line 16).

E. Integration in IEEE 802.15.4 networks

The algorithms we have just discussed can be directly applied to IEEE 802.15.4 networks: the original source address field has a size of 16 bits; we define the shortened address with a size of 10 bits (allowing the identification of up to 1024 nodes), leaving 6 bits for the ACS. A CRC algorithm, such as CRC6 [19], can be used for that purpose with reasonable coverage [20]. Given we are using a standard frame format, there is no need to include any integrity verification data in the payload part of the frame and therefore no frame overhead penalties arise. Since the size of the address field and the typical length of a IEEE 802.15.4 (approx. 924 bits using 16-bit addressing) we roughly estimate that a source address extraction will not succeed only in 1% of the frames.

Additionally, we use $k_{perm} = 4$ as a reasonable value of the threshold for detecting node permanent failures in IEEE 802.15.4 networks, being derived from the sum of the IEEE 802.15.4 omission degree bound, $k = 3$, utilised to detect channel failures [5], and the number of consecutive omission produced by the same node, which is assumed $k_{txfail} = 1$.

IV. Conclusion

This paper has presented a forward looking innovative idea of an extension to the frame check sequence procedures that enables the detection and signalling to high protocol layers, such as protocols implementing resilient real-time communications, that frame have arrived with errors.

Furthermore, we have presented three additional algorithms that allows: to extract correct relevant information (e.g., node source address) from an erroneous frame; the ability to detect the permanent failure of a node (e.g., due to the failure of its transmit circuitry); and the capability to detect the crash failure of a node when it simply stops to operate.

These algorithms are particularly useful and relevant to build a node failure detection and membership services, as a part of a wider and comprehensive approach to support resilient real-time communications over wireless networks.

References

[1] J. L. R. Souza and J. Rufino, "An approach to enhance the timeliness of wireless communications," in *The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, Lisbon, Portugal, 2011.

[2] ——, "Towards resilient real-time wireless communications," in *25th Euromicro Conference on Real-Time Systems (ECRTS-WiP)*, Paris, France, July 2013.

[3] P. Veríssimo, J. Rufino, and L. Rodrigues, "Enforcing Real-Time Behaviour on LAN-Based Protocols," in *10th IFAC Workshop on Distributed Computer Control Systems*, September 1991.

[4] J. Rufino, C. Almeida, P. Veríssimo, and G. Arroz, "Enforcing Dependability and Timeliness in Controller Area Networks." in *32nd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Paris, France, Nov. 2006.

[5] J. L. R. Souza and J. Rufino, "Analysing and reducing network inaccessibility in IEEE 802.15.4 wireless communications," in *38th IEEE Conference on Local Computer Networks (LCN)*, Sydney, Australia, October 2013.

[6] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Priority assignment for real-time flows in WirelessHART networks," in *23rd Euromicro Conference on Real-Time Systems (ECRTS)*, 2011, pp. 35–44.

[7] W. Shen, T. Zhang, M. Gidlund, and F. Dobslaw, "SAS-TDMA: A Source Aware Scheduling Algorithm For Real-Time Communication In Industrial Wireless Sensor Networks," *Wireless Networks*, vol. 19, no. 6, pp. 1155–1170, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11276-012-0524-2

[8] W. Dong, C. Chen, X. Liu, K. Zheng, R. Chu, and J. Bu, "Fit: A flexible, lightweight, and real-time scheduling system for wireless sensor platforms," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 126–138, Jan 2010.

[9] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *in MobiSYS 04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM Press, 2004, pp. 125–138.

[10] O. Babaoğlu and R. Drummond, "Streets of Byzantium: Network Architectures for Fast Reliable Broadcasts," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 6, Jun. 1985.

[11] IEEE 802.15.4, "Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs) - IEEE standard 802.15.4," 2011.

[12] ATMEL, *ATMEL AVR2025: IEEE 802.15.4 MAC Software Package - User guide*, ATMEL Coorporation, May 2012.

[13] T. Fujiwara, T. Kasami, A. Kitai, and S. Lin, "On the undetected error probability for shortened hamming codes," *IEEE Transactions on Communications*, vol. 33, no. 6, Jun. 1985.

[14] D. Eckhardt and P. Steenkiste, "Measurement and analysis of the error characteristics of an in-building wireless network," in *Annual Conference of the Special Interest Group on Data Communication (SIGCOMM)*, 1996.

[15] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance study of IEEE 802.15.4 using measurements and simulations," in *Proceedings of the Wireless Communications and Networking Conference (WCNC 2006)*. Las Vegas, NV, USA: IEEE, Apr. 2006, pp. 487 – 492.

[16] J. L. R. Souza and J. Rufino, "Characterization of inaccessibility in wireless networks - a case study on IEEE 802.15.4 standard," in *IFIP 3th International Embedded System Simposium IESS*, September 2009.

[17] I. Ramachandran, A. K. Das, and S. Roy, "Analysis of the contention access period of IEEE 802.15.4 MAC," *ACM Transactions on Sensor Networks*, vol. 3, March 2007. [Online]. Available: http://doi.acm.org/10.1145/1210669.1210673

[18] M. Hameed, H. Trsek, O. Graeser, and J. Jasperneite, "Performance investigation and optimization of IEEE 802.15.4 for industrial wireless sensor networks," in *IEEE 13th International Conference on Emerging Technologies & Factory Automation (ETFA)*, September 2008.

[19] 3G 3rd Generation Partnership, "Physical layer standard for CDMA2000 spread spectrum systems - revision d - version 2.0," 2004.

[20] P. Koopman and T. Chakravarty, "Cyclic redundancy code (CRC) polynomial selection for embedded networks," in *International Conference on Dependable Systems and Networks (DSN)*, June 2004, pp. 145–154.

A networking infrastructure for small smart grids

Michael Short and Muneeb Dawood
Electronics & Control Group / Technology Futures Institute,
Teesside University,
Middlesbrough, UK.
{m.short, m.dawood}@tees.ac.uk

***Abstract*—**A smart grid is an energy distribution network that not only allows for the physical transfer of energy (usually electricity but not necessarily so), but also supports ICT interfaces that enable real-time information exchange related to the scheduling, monitoring, control and protection of the interconnected energy generating, consuming and transmission equipment. To facilitate this, the supporting ICT infrastructure must provide a wide variety of interconnectivity options for measurement, control and user interface equipment (e.g. smart meters, synchrophasors, weather measurement stations, grid inverters, building automation controllers, energy trading applications, etc). As well as providing a vast amount of interconnection options, the infrastructure must also provide varying levels of flexibility (typically to provide reconfigurable information flows) and dependability (typically reliability/availability, security and timeliness). Although several proposals have been made in this area, there is yet to emerge a single set of accepted standards and only a small number of attempts to provide realistic implementations have been achieved. In this paper, an outline for a proposed networking infrastructure for small smart grids will be described. Some initial work describing a prototype implementation and initial testing is also described.

Keywords—Smart grids; Audio Video Bridging; Automatic Generation Control; Home and building automation.

I. INTRODUCTION

Traditionally, for economic and safety reasons, the two most commonly consumed worldwide forms of energy – heat and electricity – have been generated by large fossil-fuelled generators and transported to consumers via one-way transmission and distribution networks (typically through hot water or steam pipe work and copper wires) [1]. Although the generation has always been distributed over several large generating stations, with transmission interconnections possibly spanning several countries, these generators and interconnections have been under the control of only a small number of public and private bodies [1][2]. As such the scheduling, control, synchronization and management of the overall network have been achieved using well-understood means using dedicated technologies. With respect to communications, the employed technology would typically consist of a dedicated duplex channel between a utility and the Independent System Operator (ISO) for telemetry and telecommand, with an additional voice/data channel over PSTN as a backup [3]. More recently, remote data exchanges using UDP/IP or TCP/IP over suitable media have been employed. In addition, give the large-scale nature of traditional generating plant, a large number of highly skilled operators and engineers are present to continuously monitor and adjust the state of the local systems to help keep the overall network running safely and close to its optimum state.

However the liberalization of the energy markets - combined with the drive towards a low-carbon economy - has forced a rethink in the way that energy is to be generated and distributed to consumers [4]. In particular, the emergence of small- and medium-scaled generation equipment (typically driven by renewable or alternative forms of energy conversion) embedded within the transmission and distribution networks can lead to problems of grid control, stability and protection. This is down to several reasons, but is principally because the power flows become bi-direction for the grid nodes equipped with Distributed Energy Resources (DERs), as opposed to unidirectional [4]. Although the state of the grid can be monitored locally to a DER for the purposes of control (e.g. frequency, voltage in an electrical grid), without co-ordination this has proved to be insufficient to ensure area-wide stability, as recent problems in EU grids have shown [5]. Ultimately it is desirable to have all DERs, even small privately owned ones, in a position to respond to real-time commands from the grid operator or regulator [5]. In turn this suggests that the communication links that are required for effective management of the network are required to be extended to every DER node that is capable of generation or storage, and ideally all power injections must be capable of being regulated, controlled or adapted remotely.

This has led to the concept of the smart grid, i.e. an energy distribution network that not only allows for the physical transfer of energy (usually electricity but not necessarily so), but with support for ICT interfaces that enable real-time information exchange related to the scheduling, monitoring, control and protection of the interconnected DERs. Although several proposals have been made in this area, there is yet to emerge a single set of accepted standards and only a relatively small number of attempts to provide realistic implementations have been achieved. In this paper, a proposed ICT infrastructure for small smart grids operating at the 'Neighborhood' level will be described. By neighborhood, it is intended that the communications between DERs takes place over a distance of not more than approximately 20 KM[1]. Some initial work describing a prototype implementation and initial testing is also described. The remainder of this paper is organized as follows. Section II provides some background information including a summary of the main states that an

[1] Note that this definition is based upon the observation that most district heating systems and radial distribution feeds will not extend much further than ≈ 10 KM from a medium-sized Combined Heat and Power (CHP) plant.

energy distribution system may be in, the events that may occur to occur to cause state transitions, and typical required reaction times of the interconnected physical infrastructure used to prevent transitions to unwanted or dangerous states. Following this, the Section presents a categorization of the main communication traffic types to be expected within a small smart grid. Section III provides an outline of an ICT infrastructure that may be able to meet these needs for small neighborhoods, and Section IV provides details of some initial prototype implementations and testing that have been carried out in the context of an EU-funded project[2]. A brief summary of future work is presented in Section VI.

II. Background

A. *Power System Control and Monitoring Functions*

The operational state of a power generation and distribution system will generally fall into one of five possibilities, as indicated in Fig. 1 below:

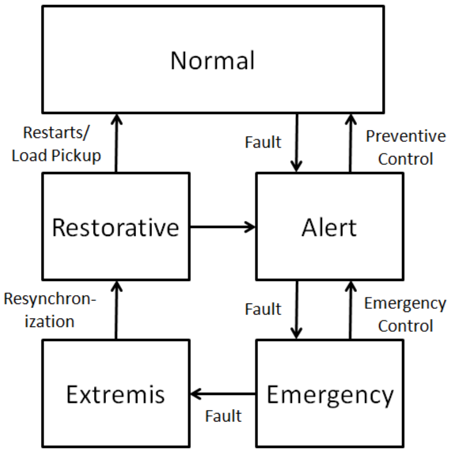


Fig. 1. Operational states of a power system

The diagram above is reproduced from Elgerd [3] and provides a good conceptual picture of the overall control requirements of a power system. For more than 99 percent of the time the system is to be found in its normal state, during which all the system power flows are satisfied and well within safe limits, and the system overall is as close as it may be to its optimal configuration economically. Assume now that the system would suffer the sudden loss of a generator (or equivalently, experience a sudden unexpected increase in load) or experience some other event that would reduce the security level of the system. Then the system would enter an alert state, in which although all the system power flows would still be stable and within safe limits, the system overall may not be optimally configured. Preventive controls (such as modulating the power output of remaining generating plant) are activated to try to return the system state to normal. Suppose that while still in the alert state, some further fault or disturbance occurs, such as the tripping of another generator: the resulting power imbalance may overload a line, and the system enters the emergency state and operates in an unsafe condition. At this point, emergency controls (such as enforced load shedding and activation of spinning reserves) must quickly be implemented to remove the overload. If these controls fail, then a series of cascading fault events (such as the tripping of transmission lines) quickly occur leading to an extremis state with increasing disturbances throughout the system. In such a state, the network typically breaks up into islands operating independently of each other, with large power imbalances throughout the system and potential blackouts/power cuts. The restorative state typically involves generator restarts and gradual load pickup/synchronization, and it may take between several hours to days to return to normal operation. Note that whatever the current state of operation, synchronization of controls and actions is normally through a central control station.

B. *Smart Grid Domains and Existing Standards*

Within a small smart grid, from a control, monitoring and ICT perspective it is possible to identify two distinct domains with different requirements and constraints. These domains of operation are the local generation and distribution (‘field’) domain and the ‘customer’ domain. Distributed Energy Resources (DERs) exist physically in both domains. Domain (ii) can be expected to contain a large amount of smart meters.

1) *Field Domain*

The field domain contains multiple DERs which may generate energy locally on a ‘medium’ (macro) scale, such as solar collector fields and CHP plant along with traditional forms of energy generation (e.g. coal or gas powered generators). This domain also contains high and medium voltage distribution and protection equipment, some heavy/medium industry and interlinks for power exchanges to other areas. ICT connectivity is mainly provided by Local, Metropolitan and Wide Area Networks along with field-level automation networks such as CAN and PROFIBUS [6]. In this domain, IEC 61850 is the standard of choice for automation and communication within DERs and substations [7][8]. IEC 61850 communications are based on the Ethernet protocol using store-and-forward switches to increase determinism, and operate at communication rates of up to 100 Mbps. The main purpose of the standard is not to strictly define the lower level of a protocol stack - this is achieved through the use of the existing Ethernet standards - but to define application layer features (principally data semantics) so that intelligent electronic devices (IEDs) from different vendors can interoperate and communicate efficiently. The standard principally supports operation, control and automation of sub-station equipment, and has limited facilities for monitoring and control by remote stations. The IEC 61850 communication architecture is given in Fig. 2.

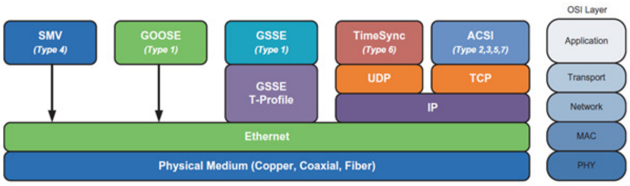


Fig. 2. Communication stack mapping for IEC 61850 messages

As may be seen in the figure, the communications architecture is only partially based upon IP protocols and

[2] Refer to http://www.ideasproject.eu/ for further details.

internet technology. The time-critical, low-level communications are directly mapped into raw Ethernet frames using specific (non-IP) protocols. These low-level communication types include GOOSE (Generic Object Oriented Substation Events), GSSE (Generic Substation State Events) and SMV (Sampled Measured Values). MMS (Manufacturing Messaging Specification) may also be mapped into the low-level communications, but MMS is now generally transferred using standard TCP/IP (this is a revision to the original MMS specifications to improve easy of connectivity). The GOOSE type supersedes the GSSE type and is now becoming the preferred and dominant mapping in IEC 61850 [7][8]. GOOSE and SMV are both directly mapped into frames using ASN.1: BER (Basic Encoding Rules) and employ IEEE 802.1Q (Priority Tagging/VLAN) at layer 2 to achieve the desired fast response times. Ethernet v2 is employed at the physical layer, normally over fiber optic cabling to achieve very high noise immunity. Normally, the communication structure of the tagged, time-critical frames is simple enough that timing analysis using well-known techniques can be applied [9].

In the context of smart grids, the recent proposal for an IEC 61850-7-420 Communications Standard for Distributed Energy Resources (DER) is of interest. This extension to the basic standard is hoped to achieve a single international standard that defines the entirety of communication and control interfaces for all DER devices. Although much of the required DER interface traffic may be carried by internet protocols such as TCP/IP over a DSL or modem link, there is a lack of needed real-time support to export low-level control and monitoring signals in real-time. This real-time support is needed for power grid management and balancing. For example, worst-case message latencies for synchrophasor signals (which are mapped into SMVs) are around 40 ms in a 50 Hz electrical system; clock synchronization between multiple PMIs (located in physically separate DERs) must be around be around the 1 μs level for measurements to be meaningful. Such real-time capabilities clearly cannot be guaranteed with standard IP technology over a MAN, WAN or DSL. For a small smart grid, large elements of IEC 61850 can be leveraged using standard IP to provide the required interconnectivity for DERs; a solution to the problem of real-time traffic that may be suitable for multiple bridged LANs is in the form of the AVB protocols [10][11]. Audio Video Bridging (AVB) is a common name for a set of interrelated technical standards developed by the IEEE AVB Task Group connected to the IEEE 802.1 standards committee. The goal of the AVB standards is to provide the specifications that will allow time-synchronized low latency streaming services through IEEE 802 networks, and consists of four main elements:

IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications (PTP). This standard describes how the best source of time in a LAN may be identified and distributed through the network to synchronize all end stations. When employed with suitably accurate clock sources, PTP enables fault-tolerant synchronization at sub-microsecond levels.

IEEE 802.1QAT: Stream Reservation Protocol (SRP). The SRP) is one of the core protocols required for AVB Systems. SRP is designed to allow the sources of AVB content (Talkers) to advertise that content (Streams) across a network of AVB Bridges, and users of the AVB content (Listeners) to register to receive the streams through AVB Bridges. Streams are only propagated through the network if there are enough resources to guarantee timely delivery of every stream packet. Once a stream is confirmed, VLANs along the route are automatically configured and bandwidth reserved.

IEEE 802.1QAV: Forwarding and Queuing for Time-Sensitive Streams (FQTSS). Once AVB streams are confirmed, some mechanisms are needed along the stream route to ensure timeliness is achieved and the reserved bandwidth is not exceeded. FQTSS defines the mechanisms by which this is enforced, and specifies a Credit Based Shaper (CBS) for online traffic shaping. The CBS uses information about the reserved amount of bandwidth for AVB streams, which is calculated by SRP.

IEEE 802.1BA: Audio Video Bridging Systems. Since the whole AVB scheme depends upon the participation of all devices between the talker and listener, any network element that does not support AVB must be identified and flagged. The procedures for this are defined in this standard.

Together, these standards define common QoS services for time-sensitive streams and mappings between different layer 2 technologies. They also enable a common endpoint interface for QoS regardless of the particular layer 2 technologies used in the path followed by a stream, effectively defining an API for QoS-related services that extended well beyond the transfer of audio and video (Teener et al. 2013). The principal end services that are offered to applications by an AVB network are two real-time streaming classes: the low-latency stream provides guaranteed end-to-end latency of 2 ms over 7 hops, and the medium latency stream provides guaranteed end-to-end latency of 50 ms over 7 hops. With appropriate enhancements to the domain of medium-scale energy generation and distribution, it is clear that AVB technology may be able to provide the real-time facilities for transporting low-level IEC 61850 traffic that are lacking in standard IP technology.

2) *Customer Domain*

The customer domain contains homes, office buildings, small business and light industry which may have local generation on a ‘small’ (micro) scale, such as photovoltaic (PV) arrays. This domain also contains low voltage distribution and protection equipment. ICT connectivity is provided by (possibly bridged) Local Area Networks and DSL links, along with building automation networks such as KNX. In this domain, IEC 62056 is the dominating standard which is for communications with smart meters; however elements of IEC 61850 may still be present within the DERs. IEC 62056 is a modern European smart metering protocol and is a superset of IEC 61107. IEC 62056 is a set of standards that defines the electrical connections/interfaces and data exchanges for meter reading, energy tariffs and load control that is published by the IEC. In terms of data exchanges, the standards describe both a Device Language Message Specification (DLMS) and a Companion Specification for Energy Metering (COSEM). IEC 61107 is a relatively simple set of standards, is well-accepted and is now widely used in the EU, and the standards include details of how DLMS and COSEM may be mapped to TCP or

UDP using standard IP. Clearly IEC 62056 can provide the required connectivity using standard IP technology for smart metering that is required for small smart grids.

3) *Traffic Summary*

Based upon the above discussions, and taking on board previous analyses of smart grid traffic (e.g. [6]), it is possible to try to generically classify the types of network traffic that may be expected in a small smart grid, and this is given in Fig. 3 below. This table of message types may be divided into three different classes based upon their latency requirements. Class 'A' traffic corresponds to the low-latency class, and consists of types 1a, 1b and 4. Class 'B' traffic corresponds to the medium-latency class, and consists of types 2 and 3. Class 'B' traffic corresponds to the high/very high-latency class, and consists of types 5 and 6. Type 7 traffic may be in either class, depending upon the accuracy of synchronization that is needed. Based upon the previous discussions, a networking infrastructure for small smart grids is described in the next Section.

Message type	Application Services	Deadlines (ms)
1A	Fast message (trip): GSE / GOOSE	3–100
1B	Fast message (other): Control, monitoring	20–100
2	Medium speed message: ACSI (MMS)	100-500
3	Low speed message: ACSI (MMS)	≥ 500
4	Periodic raw data: SMV	3–100
5	File transfer: ACSI (DLMS/COSEM, EPNMS)	≥ 1000
6	Web services	≥ 1000
7	Time synchronization TS	(Required accuracy)

Fig. 3. Smart grid traffic types

III. Proposed Networking Architecture

A. *Field Domain*

The networking architecture that is proposed for the field domain is illustrated graphically in Fig. 4. The solution architecture leverages the fact that since IEC 61850 is the dominant standard for automation within DERs and is widely accepted, one may expect that the ICT architecture within them is based upon one or more LANs implemented by switched Ethernet. Although TCP/IP may be used to provide connectivity and flexibility for Class C traffic, some additional means must be leveraged to provide flexible support for Class A and B traffic. This support is provided by leveraging the flexible, low-latency streaming capabilities of AVB to implement a fibre-optic MAN backbone operating at 1 Gbps or more[3] that may be used to bridge together the DERs and the grid control centre.

As direct fiber-optic links using non-buffered repeaters may be used, the distances required in a small smart grid may be covered (note that AVB is not yet suitable for wireless operations; however this is part of the working group's plans for extension of the protocol). Grid Class A and B traffic may be mapped into the low- and medium-latency stream facilities of AVB, and gateways used to carry low-latency traffic classes into and out from a DER/sub-station. Since an AVB network also provides TCP/IP connectivity, this provides a flexible solution for the field side of a grid that has the potential to provide re-configurability and flexibility even for low-latency traffic. A possible protocol stack (with reference to the 5-layer 'internet' version of the OSI 7-layer model) for this is as shown in Fig. 5 below.

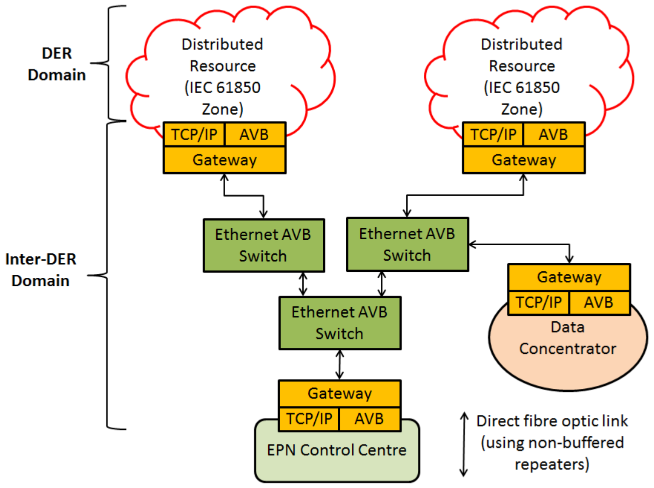


Fig. 4. Infrastructure for the field domain

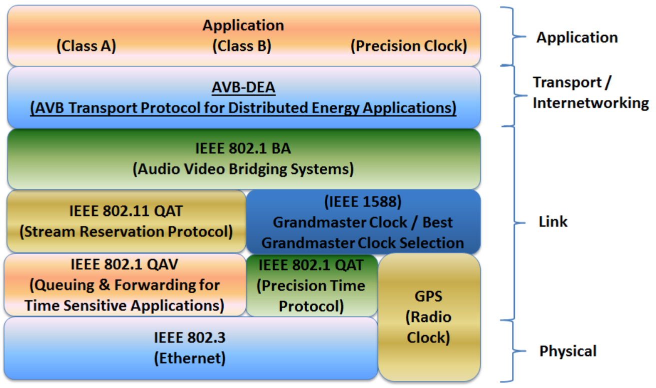


Fig. 5. Protocol stack for the field domain

At this stage, the full specification of this layer of the protocol stack cannot be stated clearly and unambiguously, as research is needed to examine the exact requirements it must possess. However it can be stated that the services that must be exposed to the application would typically include: (i) a transparent high-precision clock interface to provide local timing control to drive the scheduling of critical activities, (ii) a means to dynamically map grid traffic types from Class A and Class B into low- and medium-latency AVB streams and to both advertise and discover these grid class streams and (iii) Ingress and egress temporal firewalling and security and integrity checking.

B. *Customer Domain*

The generic solution architecture that is proposed for the customer domain is illustrated graphically in Fig. 6. The solution architecture leverages the fact that most homes and small business nowadays have private internet connections that

[3] Note that such fibre optic gigabit AVB switches are now commercially available: see for example, http://www.extremenetworks.com/libraries/solutions/avb_solution_brief.pdf.

are typically implemented via a fibre-optic backbone or a fast DSL/ISDN link operating at least 2 Mbps. TCP/IP may be used to provide connectivity and flexibility for Class C grid traffic; UDP/IP and multicast protocols such as IGMP/PIM may be leveraged to provide flexible support for Class A and B traffic. However, such support is not likely to achieve the low-latencies required for these latter traffic classes as no timing guarantees can be obtained from commercial ISPs. Instead, such traffic must be operated in a soft real-time mode, which – given that customer side DERs are normally low-voltage and non-critical - may be suitable in most cases. The required protocol stack (with reference to the 5-layer 'internet' version of the OSI 7-layer model) for the customer domain is shown in Fig. 7. As can be seen, this is pretty much a standard comm stack.

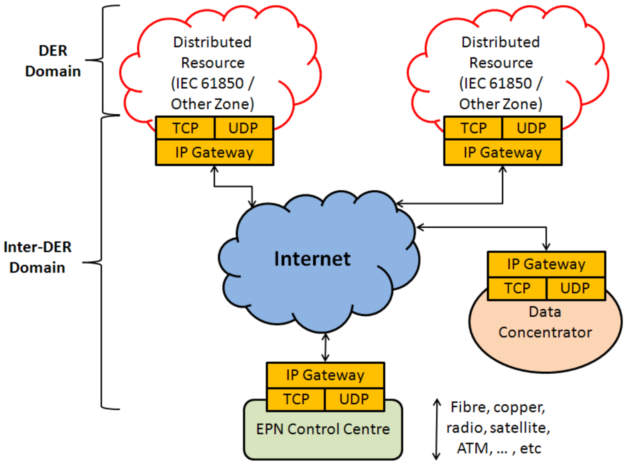


Fig. 6. Infrastructure for the customer domain

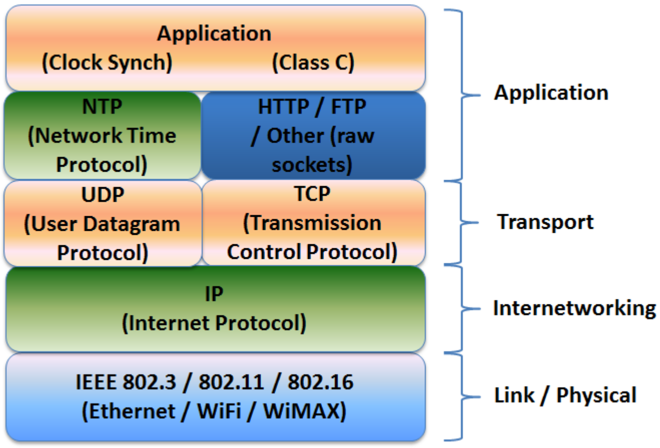


Fig. 7. Protocol stack for the customer domain

IV. TEST FACILITY

In order to test the elements of the proposed infrastructure, two main demonstrating sites have been identified and are located in France and Finland. The customer domain architecture is under test at these sites. In addition, there is another smaller demonstration site implemented in the UK for testing some of the communications concepts defined for the field domain. In this Section, progress made in this latter aspect is described. This demonstration site is intended to directly explore how aspects of dependable and re-configurable real-time communications may be implemented for class A and B traffic. A high-level overview of the test facility is shown in Fig. 9. A simulated power grid is employed for the purposes of employing Hardware-In-The-Loop (HIL) testing of the embedded communication, monitoring and control systems using specific scenarios that will be developed and fine-tuned during prototyping. The importance of HIL testing for smart grid application, including the testing and evaluation dependable communications, has been previously highlighted [12].

With respect to Fig. 4 and 5, the main elements of this demo site may be viewed as: (i) a simplified central control center implemented on a PC; (ii) multiple PC-based environments for the real-time simulation of multiple DERs and generation / distribution / protection equipment, (iii) a communications infrastructure based upon standard and AVB-based packet switched Ethernet links and (iv) a high-precision GPS radio clock. Note that the basic equipment setup is shown in the above; the test facility is flexible in that the amount of IEDs and gateways may be modified to suit the particular scenario under test.

In particular, prototype gateways are under construction to test the basic services outlined in the previous Section, and to provide a working interface between two simulated IEC 61850 switched LAN sub-domains and a central control center implemented. The gateways are implemented using embedded PCs. The suggested structure of such a remotely configured gateway is as shown in Fig. 8. TCP/IP and a custom message specification (EPNMS) are to be used in combination to allow the remote configuration of the gateways. In addition to the AVB bandwidth reservation mechanisms, temporal firewalling will be implemented in the form of Generic Cell Rate Algorithms (GCRAs) operating as 'leaky buckets' will be employed on the periphery of the DER domains, to prevent erroneous messages creating network loads that are beyond stated requirements [13].

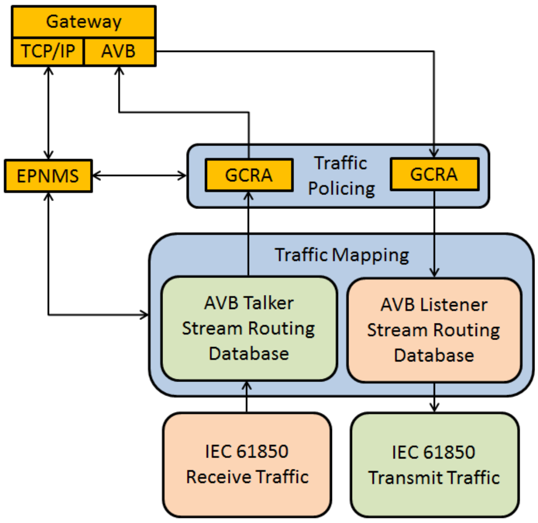
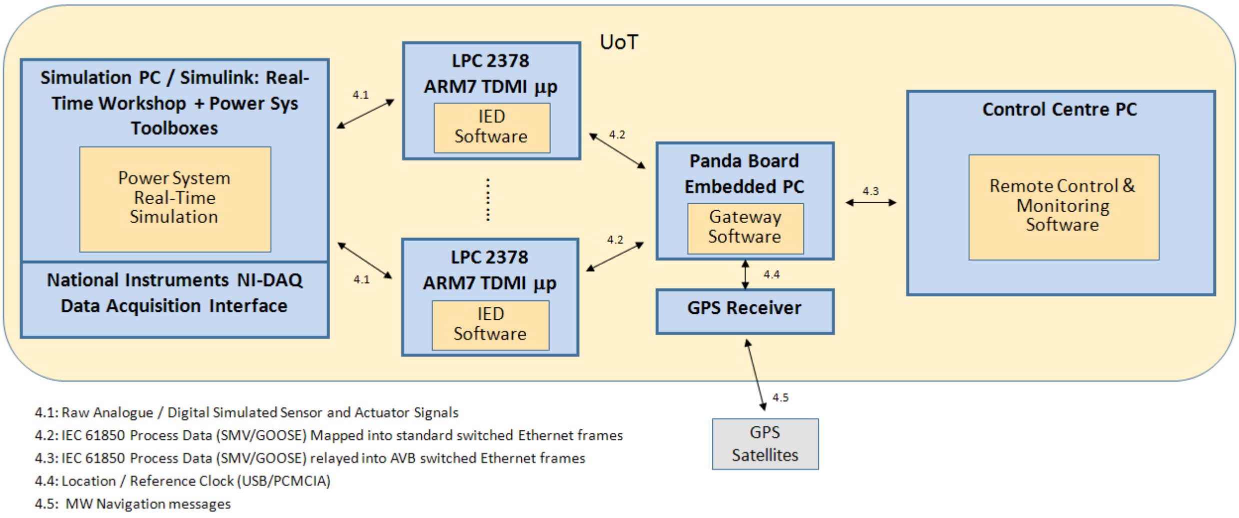


Fig. 8. Protoype gateway



UoT
Simulation PC / Simulink: Real-Time Workshop + Power Sys Toolboxes
Power System Real-Time Simulation
National Instruments NI-DAQ Data Acquisition Interface
LPC 2378 ARM7 TDMI µp
IED Software
LPC 2378 ARM7 TDMI µp
IED Software
Panda Board Embedded PC
Gateway Software
GPS Receiver
Control Centre PC
Remote Control & Monitoring Software
GPS Satellites
4.1
4.2
4.3
4.4
4.5
4.1: Raw Analogue / Digital Simulated Sensor and Actuator Signals
4.2: IEC 61850 Process Data (SMV/GOOSE) Mapped into standard switched Ethernet frames
4.3: IEC 61850 Process Data (SMV/GOOSE) relayed into AVB switched Ethernet frames
4.4: Location / Reference Clock (USB/PCMCIA)
4.5: MW Navigation messages

Fig. 9. Overview of Proposed Test Facility

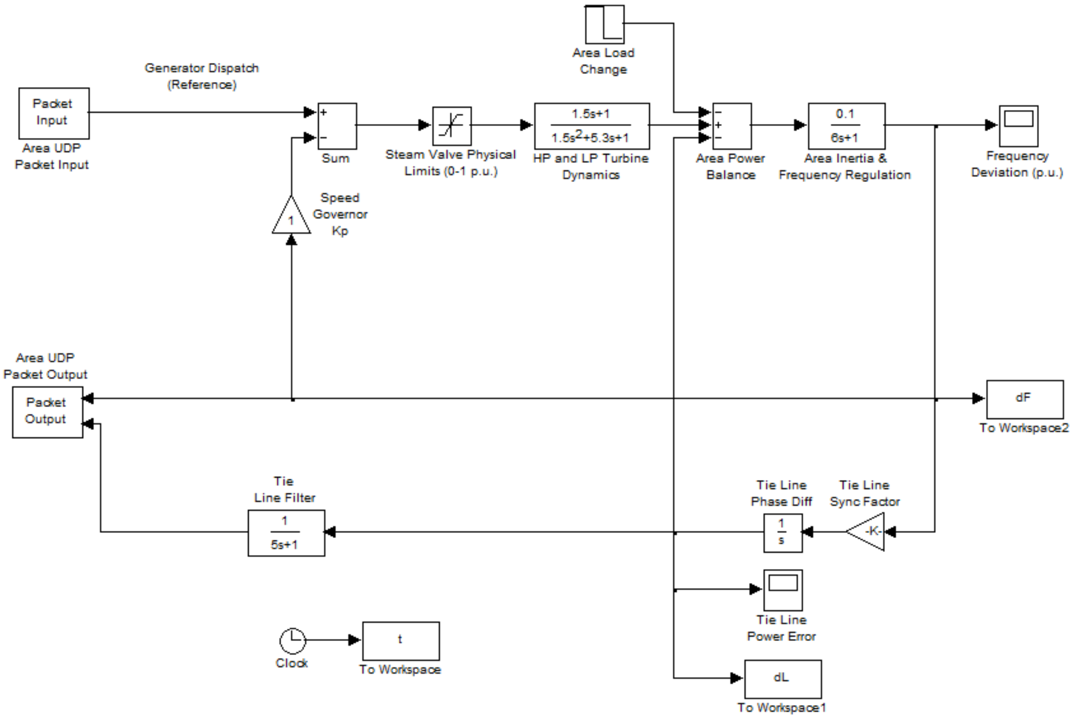


Fig. 10. Single area power flow dynamic model

V. Preliminary Tests

To provide a preliminary test of the proposed facility, a linearized per-unit (p.u.) one-area model of a power system with a tie-line interconnection to a larger network (assumed to be a national grid) was created. The linear model is commonly used in the study of power systems and is described in [3][15]. The purpose of the model was to allow the real-time testing of a simple Automatic Generation Control (AGC) scheme for performing area Load and Frequency Control (LFC).

A Simulink block diagram of the configuration is as shown in Fig. 10. The model features a single equivalent lumped generator within the area (using a representative dynamic model for a two-stage turbine with re-heater [14]) which is used – along with a scheduled power exchange to the larger grid – to supply the local power demand. The selected units were 1 p.u. power = 1 GW and 1 p.u. frequency = 50 Hz. The area configuration was set using typical representative parameters: the load was assumed to have a power dependency on frequency of 200 MW per Hz, with inertia for the equivalent generator of 6 seconds. The tie-line synchronizing co-efficient (dependent upon the reactance of the line) was taken to be 2π/15. The basic principles behind AGC for LFC are well described elsewhere (e.g. [15][2]). Basically, when an increase in area load occurs (or equivalently generation shortfalls), there is an instantaneous mismatch between power supply and demand and kinetic energy is borrowed from the turbines. This causes the rotational speed of the shafts - and hence the area supply frequency – to drop. As the supply frequency drops the power consumed by the connected loads also drops proportionately, and unless extra power is injected to restore the balance a steady-state frequency offset occurs once the transient has subsided. The primary control means to restore this balance is to use proportional speed governors on the connected generators to increase the opening of the steam valves. Although this is sufficient in most cases to keep the frequency within working limits, some offset (known as the 'droop') remains due to the lack of integral action. A value of 5% droop between no-load and full-load, which is typically used in industry [15], was used in this example by setting the governor gain appropriately. The resulting drift in frequency away from the nominal value due to this droop leads to a phase shift between the local frequency and that of the areas connected by the tie-lines occurring. This negative phase shift induces a net power-flow into the area to restore the power balance and hence frequency. In the case of decreases in area load (or equivalently, generation excess), the reverse of the process occurs: the frequency will increase in this case, and a net power flow out of the area is induced by a positive phase shift. Since in most cases it is desirable to keep tie-line power flows within safe limits and as close as possible to pre-scheduled (contracted) levels, the modification of the net power flows due to supply/demand changes is mostly unwanted.

The role of LFC is to maintain the area frequency and pre-scheduled tie-line power flows using a form of secondary control. As shown in Fig. 10, the instantaneous frequency and filtered tie line power errors are transmitted via a SCADA system to a central controller maintained by the area ISO, as shown in Fig. 11, and combined to form an Area Control Error (ACE) signal. This ACE signal is passed to a slow-acting Integral or Proportional-Integral (PI) controller, the output of which is a bias signal which is sent – again via the SCADA system – to the set points of the connected generators that have been contracted to partake in area regulation. Although the controller is shown in continuous time in the figure, in reality it is discrete and a sampling rate of 1 – 6 seconds is normally used. Recently, remote data exchanges using UDP/IP or TCP/IP have been employed to implement these data exchanges, and the stability of the closed-loop is highly dependent upon the network delays that are encountered [2]. In order to illustrate this point further, an experiment was carried out using the test facility in which the LFC controller was implemented on a physically separate PC to that used for the real-time implementation of the area model, with UDP/IP communications over a small switched Ethernet network employed as the SCADA system. A 1 second sampling time for the LFC loop was employed, and the sampling time for the area dynamics and remote controller was set to 1 ms.

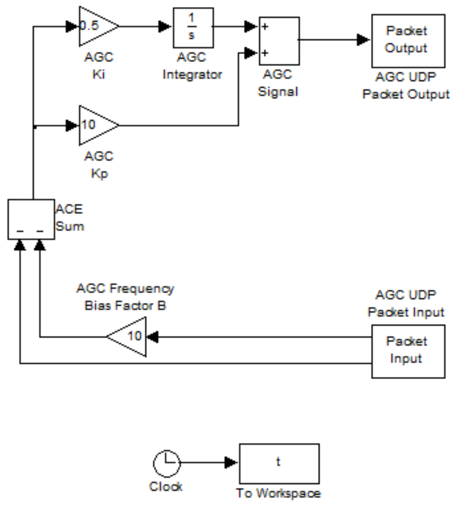


Fig. 11. Remote AGC – LFC Controller

Fig 12 shows the resulting frequency error and Fig. 13 the tie-line power errors following a 0.2 p.u. step change in demand at t = 0 seconds, followed by a -0.2 p.u. step change in demand at t= 500 seconds. From the figures, it can be seen that both frequency and power exchanges are restored to their nominal values in around 150 seconds, which is an appropriate timescale for AGC [15].

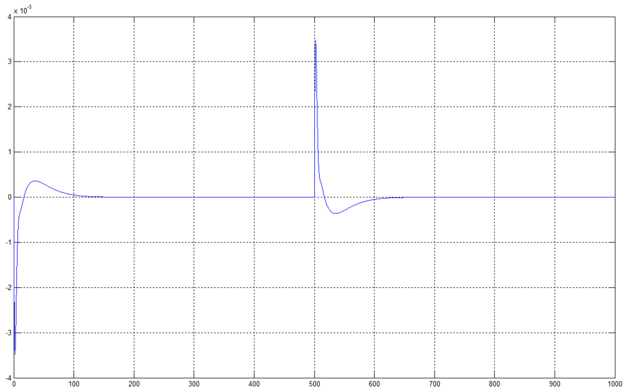


Fig. 12. Frequency deviation (p.u.) vs time, nominal delay

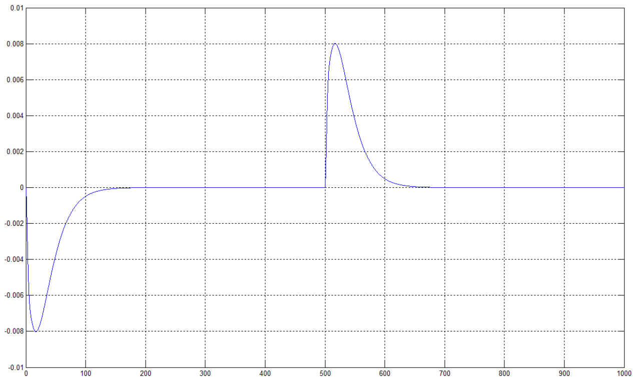


Fig. 13. Tie-line power error (p.u.) vs time, nominal delay

Next, the experiment was repeated using the same configuration, except for the deliberate injection of a 0.7 second additional delay into the packet transmissions. Note that delays of 0.7 seconds are not excessive for UDP or TCP traffic. Fig 14 shows the resulting frequency error and Fig. 15 the tie-line power errors in this case. From the figures, it can be seen that both frequency and power exchanges become oscillatory, with large undesirable changes applied to the generator.

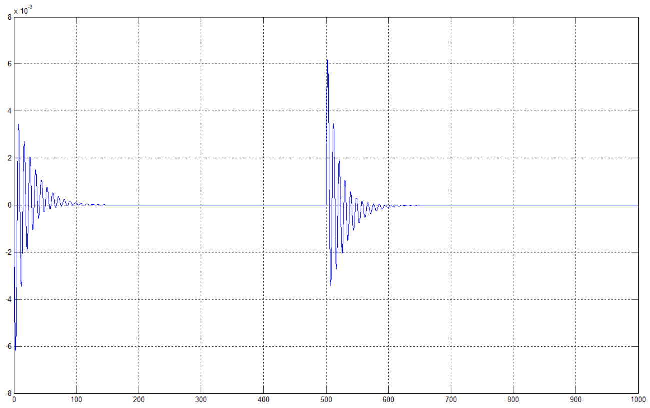


Fig. 14. Frequency deviation (p.u.) vs time, 0.7 sec additional delay

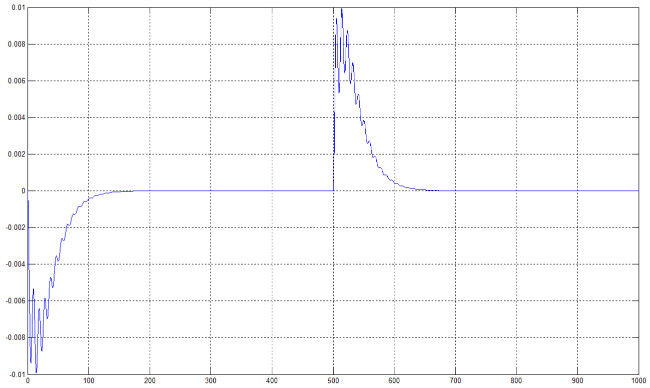


Fig. 15. Tie-line power error (p.u.) vs time, 0.7 sec additional delay

If the delay is increased to 1 second, the system becomes unstable. In reality the control gain can be de-tuned to bring stability to the system again, but this is at the expense of longer settling times and loss of performance. These results provide some initial indicators that the proposed test facility - once fully complete - should allow the evaluation of the impacts of the proposed networking infrastructure under differing operating conditions to be evaluated.

VI. Conclusions And Future Work

This paper has proposed a networking infrastructure for small smart grids, and described a test facility for the evaluation of some of the real-time aspects. Although the facility and development work is not yet complete, some basic tests to illustrate its intended usage have been described. Future work will consider more representative scenarios, and will also aim to implement AGC for LFC using the proposed AVB extensions. Comparisons will be undertaken with UDP when interfering traffic at various levels is introduced into the underlying communications networks and variable delays and QoS is present.

Acknowledgment

The IDEAS Collaborative Project (Grant Agreement No. 600071) is co-funded by the European Commission, Information Society and Media Directorate General, under the Seventh Framework Programme (FP7), Cooperation theme three, "Information and Communication Technologies". The authors wish to acknowledge the commission for their support, the efforts of the partners, and the contributions of all those involved in IDEAS.

References

[1] Masters, G. *Renewable and Efficient Electric Power Systems*. New Jersey: John Wiley & Sons, 2004.

[2] S. Bhowmik, K. Tomsovic, and A. Bose, "Communication model for third party load frequency control," *IEEE Transactions on Power Systems*, Vol. 19, No.1, pp. 543-548, 2004.

[3] C.L. Elgerd, "Control of Electric Power Systems,", *IEEE Control Systems Magazine*, Vol. 1, No. 2, pp. 4-6, 1981.

[4] J. Ekanayake, N. Jenkins, K. Liyanage, J. Wu & A. Yokoyama. *Smart Grid: Technology and Applications*. Wiley-Blackwell, 2012.

[5] Strasser, T., Andrén, F., Lehfuss, F., Stifter, M. & Palensky, P. (2013) "Online Reconfigurable Control Software for IEDs", *IEEE Transactions on Industrial Informatics,* Vol. 9, No. 3, pp. 1455-1465.

[6] Khan, R.H. & Khan, J.Y. "A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network", *Computer Networks*, Vol. 57, pp. 825–845.

[7] International Electrotechnical Comission (IEC). *IEC 61850: Communication Networks and Systems in Substations*. International Electrotechnical Comission, Geneva, 2002.

[8] Brand, K.P., Ostertag, M. & Wimmer, W. "Safety related, distributed functions in substations and the standard IEC 61850", In: *Proceedings of the 2003 IEEE Bologna Power Tech Conference*, Bologna, Italy, June 2003.

[9] K.C. Lee, S. Lee & M.H. Lee, "Worst case communication delay of real-time industrial switched Ethernet with multiple levels," *IEEE Transactions on Industrial Electronics*, Vol. 53, No. 5, pp. 1669-1676, 2006.

[10] IEEE 802.1 AVB Task Group. *IEEE 802.1 Audio/Video Bridging*. [Online]. Available: http://www.ieee802.org/1/pages/avbridges.html.

[11] Teener, M.D.J., Fredette, A.N., Boiger, C., Klein, P., Gunther, C., Olsen, D. & Stanton, K. "Heterogeneous Networks for Audio and Video: Using IEEE 802.1 Audio Video Bridging", *Proceedings of the IEEE*, Vol. 101, No. 11, pp. 2339-2354, 2013.

[12] R. Podmore and M. Robinson, "The role of simulators for smart grid development," *IEEE Transactions on Smart Grid*, Vol. 1, No. 2, pp. 205–212, Sep. 2010.

[13] Le Boudec, J.Y. & Thiran, P. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer-Verlag, April 2012.

[14] F.P. deMello, et al. "MW Response of fossil-fueled steam units", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-92, No. 2, pp 455-463, March/April 1973.

[15] N. Jaleeli et al. "Understanding Automatic Generation Control", *IEEE Transactions on Power Systems*, Vol. 7, No. 3, pp. 1106-1122, 1992.