HAL
open science

Embedded data on intelligent products – impact on real-time applications

Sylvain Kubler, William Derigent, Eric Rondeau, André Thomas, Kary Främling

To cite this version:

Sylvain Kubler, William Derigent, Eric Rondeau, André Thomas, Kary Främling. Embedded data on intelligent products – impact on real-time applications. 10th International Conference on Mobile Web Information Systems, MobiWIS 2013, Aug 2013, Paphos, Cyprus. pp.1-10. hal-00852087

HAL Id: hal-00852087

https://hal.science/hal-00852087v1

Submitted on 8 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Embedded data on intelligent products – impact on real-time applications

Sylvain Kubler[a], William Derigent[b], Éric Rondeau[b], André Thomas[b], and Kary Främling[a]

[a] Aalto University, School of Science, Espoo, Finland.
P.O. Box 15400, FI-00076 Aalto, Finland.
[b] Université de Lorraine, CRAN, UMR 7039, Vandœuvre-lès-Nancy, F-54506, France
CNRS, CRAN, UMR 7039, Vandœuvre lès Nancy, F-54506, France
corresponding author: sylvain.kubler@aalto.fi

Abstract. New challenges and opportunities arise with the Internet of Things (IoT), making it possible to link any objects of the real world with the virtual one. In recent years, IoT has become increasingly popular in industrial applications (e.g. for the inclusion of data related to the product history). It might therefore be asked what is the impact on real-time applications when accessing data from the object instead of accessing it from the database. To assess that impact, this paper develops an approach that uses jointly two simulators: CPN Tools® and OPNET Modeler®. This approach is then applied on a benchmark scenario.

Keywords: Internet of Things; Intelligent product; Real-time system; Data dissemination; Product life cycle

1 Introduction

Today, the study of the product life cycle (PLC) is an integral part of the company strategy to plan, design and manage the life of their products more effectively. In general, the PLC consists of three main phases as depicted in Fig. 1: Beginning of Life (BoL), Middle of Life (MoL) and End of Life (EoL). The different applications and actors from the PLC require specific product-related data to fulfill their mission (e.g. production orders, maintenance orders, traceability data) [1,2]. Product-related data is of great importance since it is all that people can work with when the product does not physically exist in their environment. Making product-related data sharable among all actors of the PLC is a major challenge [3].

With the advance of new technologies (nanotechnology, RFID, cloud computing), objects of the real world can be linked with the virtual one, thus enabling connectivity anywhere, anytime and for anything [4]. Such objects are also referred to as "smart" or "intelligent products", and are the essence of concepts as ubiquitous/pervasive computing and Internet of Things (IoT). These concepts refer to a world where physical objects and beings, as well as virtual data and

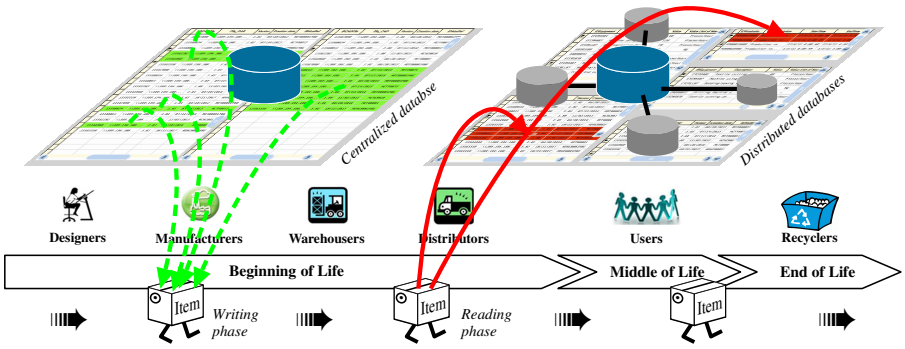

Fig. 1. Illustration of a *writing* and *reading* operation of the product during the PLC

environments, all interact with each other in the same space and time [5]. It can be stated that, today, IoT applications are mainly defined in MoL (smart house, smart space,...) but become increasingly popular in BoL and EoL [6]. For instance, much research work has been carried out in that direction by the IMS (Intelligent Manufacturing System) community over the last decade, which promotes the use of intelligent products to make systems more efficient and flexible [2]. Several definitions and classifications of an "intelligent product" are admitted, but the common denominator is that the product should be able to communicate effectively with its environment and should be able to optimize its operations, usage and other behavior. Meyer et al. [7] provide a comprehensive survey on intelligent products and related applications. Among their statements, the authors reveal that intelligent products often only provide a network pointer (e.g. via a RFID tag) to a linked database and a decision making software agent. However, in recent years, it is not uncommon to see products with larger data-storage capacities with faster data writing and reading [8,9].

Since products can carry more data than old ID technologies, it is quite natural to store, at important stages of the PLC, a subset of useful data from some databases onto the product as illustrated in Fig. 1 (*cf. Writing phase*). Therefore, this data is always available with the object to which it is attached, regardless of the network availability, and it could be retrieved in any other upstream PLC stages (*cf. Reading phase* in Fig. 1). For instance, Suzuki et al. [10] explain that intelligent products are used in the airline industry for the inclusion of aircraft part history records, which may be very useful in maintenance operations. In line with this work, some scholars developed approaches aiming at selecting the most appropriate subset of data for its inclusion onto the product [9,11]. These approaches are particularly interesting because they take into account the context of use of the data and the product (user's expectations, geographic location of the product,...) for such a data selection. However, to the best of our knowledge, no work has yet studied the impact on real-time performance when accessing data from the product instead of accessing it from the database.

This paper proposes a preliminary study to assess such an impact. In our context, the product undergoes different operations through its PLC and each operator requires particular data fragments (i.e. information subsets) to fulfill his/its mission. These data fragments are assumed to be stored/accessed either from a database or from the product itself (e.g. a product fitted with a RFID tag), assuming that both devices are always available in the application. Based on this approach, it is then possible to assess each data access schema from a time perspective. Petri Net models (using CPN Tools) are used for modeling the sequence of operations undergone by the product, used together with the OPNET Modeler for taking into account realistic time behaviors for accessing data considering a specific communication network. This approach is described in section 2, and is then applied on a benchmark scenario in section 3. This proposal could further be used as a criterion for selecting the appropriate information to be stored on the product, at different stages of the PLC. For instance, in [9], this appropriateness is based on a degree of data relevance computed by taking into account several criteria. These criteria reflect the context of use of the product (actor's expectations, product's location,. . .), and new ones may be defined in this approach (e.g. a criterion reflecting the impact on the time when accessing data from the product instead of accessing it from the database).

2 Assessment approach of real-time impact

2.1 Scenario modeling using Petri Nets and network simulations

In this study, Petri Nets are used for modeling, simulating, and analyzing the sequence of operations undergone by the product (i.e. the operations to be assessed). Petri Net model is designed by working on hierarchical views, thus facilitating the sequence modeling. For instance, the upper view could model the considered operation sequence as depicted in Fig. 2 (the upper view denoted view 0) , while the bottom views (view 1, 2,. . .) could model the characteristics of each operation like the operation duration, the set of data fragments required for achieving the operation, the resource assignment (e.g. arrival frequency of the products, size of the product queue), as depicted through the zoom made on operation 4 (denoted view 1).

The second simulator - OPNET is primarily aimed at developing and analyzing communication networks, devices, protocols and applications. It is used in our study to model and simulate the network architecture[1], thus making it possible to estimate the times to perform read/write queries on/from a database and on/from a product. These times are referred to as Round Trip Times[2] (RTTs).

[1] OPNET Modeler offers a wide selection of technologies and devices, but also provides a programming platform to develop new communication models. For instance, Dongkai and Wenli [12] designed the RFID reader models and tag models according to the EPCglobal Class-1 Generation-2 Protocol.

[2] *Round Trip Time*: time between sending the first packet of a query and receiving the last packet of the response.

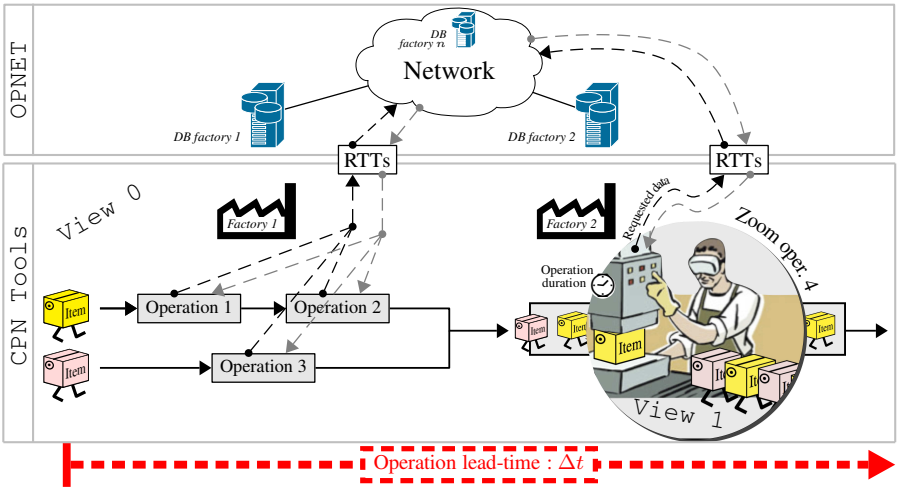

Fig. 2. Assessment approach using jointly two simulators: CPN tools & OPNET

RTTs are then injected/added in the Petri Net as symbolized by the dashed arrows in Fig. 2. More concretely, RTTs are bounded to Petri Net transitions, which represent the action of performing queries.

2.2 Real-time impact assessment

Based on Petri Net simulations, the objective is then to compare different data access schemas, where a data fragment could be either accessed from the database system or from the product. One possibility of data access schema is defined via n boolean values [F1, F2,..., Fn], with n the total number of data fragments:

- Fk means that the data fragment k is accessed from the product,
- $\overline{\text{F}k}$ means that the data fragment k is accessed from the database,

For instance, [$\overline{\text{F1}}$, $\overline{\text{F2}}$,..., Fn] informs the simulator that Fn is accessed from the product and F1, F2 from the database. In total, 2^n data access schemas are simulated and compared. The comparison is based on the time to complete the operation sequence, noted "operation lead-time" Δt in Fig. 3, which inevitably varies according to the data access schema.

3 Applicative scenario

Three operations composing a PLC are defined in this scenario, which are modeled and assessed in term of real-time performance. These three operations are modeled via CPN Tools as depicted[3] in Fig. 3, where each operator requires specific data to complete his/its tasks. In this scenario, an operation is considered

[3] Only view 0 of the hierarchical views is presented, but three levels have been defined in total in order to characterize all operation features (as explained in section 2.1).

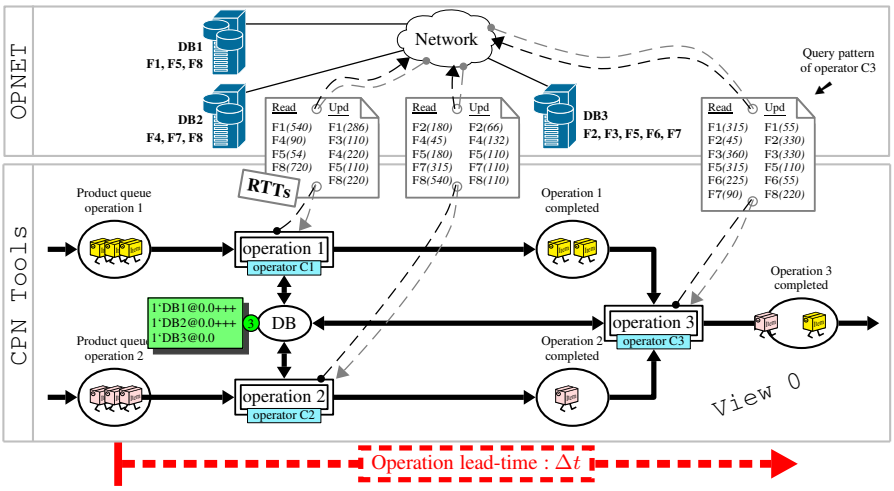

Fig. 3. View 0 of the Petri Net hierarchy: operation sequencing in the PLC

as completed once all data fragments required by the operator have been accessed/updated, and once the operation duration is ended. As a result, the time to fulfill the operation is somehow dependent on the RTTs, which are estimated via OPNET simulations based on a specific network architecture and operator's query pattern (i.e. a set of read/write query performed by each operator).

In that regard and to make the paper even more interesting, an optimal data distribution is considered as a benchmark. This benchmark is the one from [13], where a scenario considering 3 databases (DB1 to DB3), 3 clients/operators (C1 to C3) and 8 data fragments (F1 to F8) is defined. These data fragments are accessed by each operator according to a specific query pattern, and are accordingly distributed (optimally) over the 3 databases. Fig. 3 shows each operator's query pattern (the number in brackets indicating the number of bytes read/written) and the resulting data fragment allocation (F1, F5, F8 allocated to DB1, *etc.*). The network architecture, the data distribution and the operator's query patterns are configured in OPNET[4]. The RTT of each query (performed by each operator) is then estimated from OPNET simulations and injected in the appropriate Petri Net view, as symbolized by the dashed arrows in Fig. 3. By way of indication[5], the operator C3 spends $\simeq 5ms$ for reading 45 bytes of F2 and $\simeq 12ms$ for writing 330 bytes of F2. Fig. 4 shows the Petri Net model that defines the sequencing of operation 1 (i.e. the transition denoted `operation 1` in

[4] The exact network architecture is given in Fig. 5, which shows that data can be either accessed from DB1, DB2, DB3 or from the product. For information purposes, the following equipments have been used in simulations: routers *Cisco 1600*, switches *Cisco 2900XL*, server *Sun Ultra* 10 $333Mhz::1$ *CPU,* 1 *Core per CPU, Solaris.*

[5] RTTs depend on various parameters such as the hardware, the technique of data replication implemented (synchronous, asynchronous), *etc.* Due to the space limitations, all parameters are not presented.

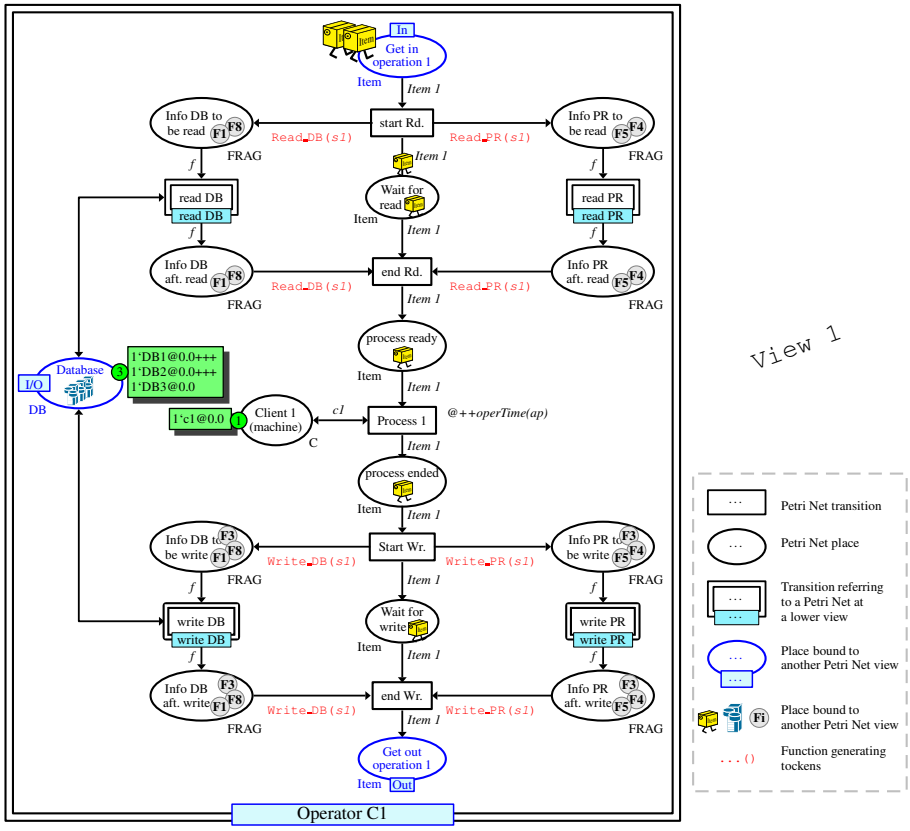

Fig. 4. Toto

view 0/Fig. 3). This Petri Net thus corresponds to a view 1 (i.e. a lower view to view 0). Two other views 1 (not detailed in this paper) are also defined regarding the transitions `operation 2` and `operation 3` of view 0. In Fig. 4, when a product arrives in the queue for being processed (i.e. in the place named `Get in operation 1`[6]), the set of data fragments that must be read by Client 1 (i.e. F1, F4, F5, F8; *cf.* Fig. 3) are immediately generated by the functions `fd1R_DB(`*sc1*`)` and `fd1R_P(`*sc1*`)`, with *sc1* the data access schema. In other words, *sc1* indicates if the data fragment Fi ($i \in \{1, .., 8\}$) is read from the database (in such a case, a color token is generated in the place denoted `Info DB to be read`) or from the product (the token is generated in the place denoted `Info PR to be read`). In the example given in Fig. 4, *sc1* specifies that F1, F8 are read from the database, while F4, F5 from the product. Once all data fragments have been read (i.e. once

[6] `Get in operation 1` is a place linked with the upper view (view 0). Likewise, places `Database` and `Get out operation 1` in Fig. 4 are respectively I/O (Input/Output) and O (Output) of this Petri Net/view.

transactions `read DB` and `read PR` are crossed[7]), products are then ready to be processed by Client 1 (see place `process ready`). Transition `Process 1` represents such a process with *@++operTime(item1)* the function that binds the time required to process the product to the transition. Once `process 1` is completed, data fragments are updated in a similar way than before, whether they are accessed from the database or from the product (see the places named `Info DB to be write` and `Info PR to be write`). Finally, products are added to the place denoted `Get out operation 1` before moving towards operation 3 (*cf.* Fig. 3).

2^8 (i.e. 256) data access schemas are simulated with CPN Tools and the resulting operation lead-times Δt are recorded (*cf.* Fig. 3). In our scenario, there are 8 data access schemas considering all possibilities of accessing one data fragment from the product (see Table 1); 28 schemas when accessing two data fragments from the product, and so on. Sections 3.1 and 3.2 present and compare the operation lead-times resulting from simulations, when using respectively a reliable network and a network with traffic overload.

Table 1. Possibilities of data access schemas considering 8 data fragments

	Solution	Detailed Solution		Solution	Detailed Solution
1/8 fragment on the product	1	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ **F8**]	2/8 fragment on the product	1	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ **F7** **F8**]
	2	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ **F7** $\overline{F8}$]		2	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ **F6** $\overline{F7}$ **F8**]
	3	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ **F6** $\overline{F7}$ $\overline{F8}$]		3	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ **F5** $\overline{F6}$ $\overline{F7}$ **F8**]
	4	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ **F5** $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]		⋮	⋮
	5	[$\overline{F1}$ $\overline{F2}$ $\overline{F3}$ **F4** $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]			
	6	[$\overline{F1}$ $\overline{F2}$ **F3** $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]		26	[**F1** $\overline{F2}$ $\overline{F3}$ **F4** $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]
	7	[$\overline{F1}$ **F2** $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]		27	[**F1** $\overline{F2}$ **F3** $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]
	8	[**F1** $\overline{F2}$ $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]		28	[**F1** **F2** $\overline{F3}$ $\overline{F4}$ $\overline{F5}$ $\overline{F6}$ $\overline{F7}$ $\overline{F8}$]

3.1 Reliable network

In our experiments, two product throughputs (i.e. the transmit rate of the wireless communication between the product and the wired network) are defined: 1Mb/s (high-speed RFID technology) and 11Mb/s (802.11b standard). The operation lead-times Δt resulting from the 256 simulations, considering both product throughputs, are synthesized in a box and whisker diagram in Fig. 5. It gives the min, the 1st-3th quartile, the median and the max Δt (see the y-axis) resulting to the different possibilities for accessing 0 data fragment out of 8 from the product (0/8 in Fig. 5; i.e. all data fragments are accessed from the database only), 1 data fragment out of 8 from the product (1/8; i.e. 7 data fragments

[7] RTTs estimated from OPNET are injected in a lower view (view 2), which represents the step sequencing of the transaction `read DB` of Fig. 4/view 1.

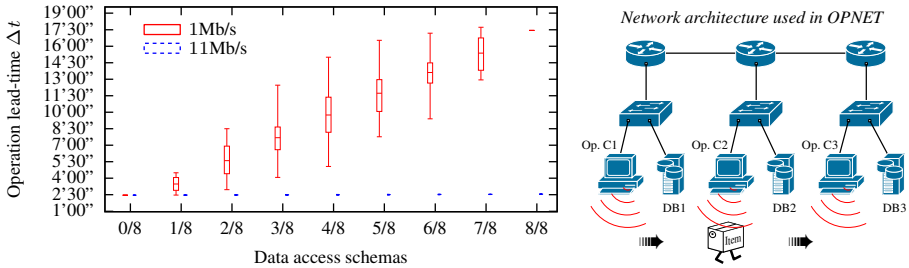

Fig. 5. Impact on times when accessing data from the product: reliable network

are accessed from the database only), and so on. The data access schema 0/8 indicates it took 2'30" (Δt) to complete the three operations[8]. The data access schema 1/8, with a product throughput of 1Mb/s, indicates that one possibility among the 8 data access schemas listed in Table 1 (*cf.* table "1 fragment from the product") requires $2'30''$ to complete the three operations (*cf.* the min Δt value), while another possibility out of the 8 requires $4'45''$ (*cf.* the max Δt value). More generally, it can be observed in this scenario that it is always faster to access data from the database considering a product throughput of 1Mb/s ($2'30'' \leq \Delta t \leq 17'30''$). However, in the case where products communicate at 11Mb/s, all solutions are identical (i.e. $2'30''$). In conclusion, there is no point to store/access data fragments on the product since the database is always a better or an equivalent solution.

3.2 Overload network situation

A traffic is added to the network between PC1 and PC2 as shown in Fig. 6 (PC1 sends 1500bytes/s to PC2). This overload inevitably generates delays on queries intended for, or originating from DB1. In such a situation, the whisker diagram in Fig. 6 shows that gains on Δt are obtained with the two product throughputs. In case of 11Mb/s, the higher the number of data fragments accessed from the product, the lesser Δt. In case of 1Mb/s, this statement is true until 3/8 because the three max Δt are always lower than $12'30''$, which is Δt at 0/8. However, when accessing more than 3 data fragments from the product, Δt therefore becomes higher and higher (*cf.* the median and max values), exceeding $12'30''$, even reaching $18'00''$.

Based on this study, it can be concluded that in such a situation, it is better to access the 8 data fragments from the product when they communicate at 11Mb/s. However, in case where they communicate at 1Mb/s, the solution providing the smallest Δt is a data access schema 3/8, which precisely corresponds to the following schema: [$\overline{\text{F1}}$ $\overline{\text{F2}}$ **F3** $\overline{\text{F4}}$ $\overline{\text{F5}}$ **F6** **F7** $\overline{\text{F8}}$], as emphasized in Fig. 6.

[8] The product throughput does not have any impact on Δt since no data is accessed from the product.

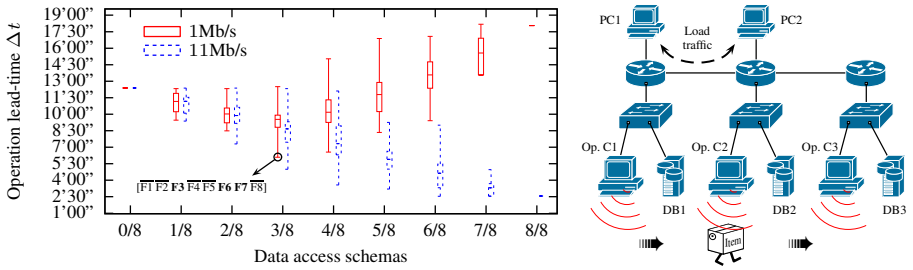

Fig. 6. Impact on times when accessing data from the product: overload traffic

4 Conclusion

Objects and products are increasingly being fitted with new technologies (RFID, sensors) that can store increasing quantities of data about themselves. In the context of product life cycle, it is a formidable challenge to link product-related data to the products themselves, thus making the information easily achievable throughout their life. However, to the best of our knowledge, no work has yet studied the impact on real-time performance about accessing data from the product instead of accessing it from the database system. Accordingly, this paper develops an approach to assess such an impact, which enables to determine in which situation(s) it is beneficial (from a time perspective) to access data from the product. A scenario is then proposed and shows that the impact on real-time strongly depends on the application features (product throughput, network state).

Results show that accessing to all information from the database does not penalize the application considering temporal aspects, unlike accessing it from the product/object. However, in situations where e.g. network disruptions occurs, results show that the application performance can be improved. In current work, Taguchi experiments are carried out to identify which factors mainly impact on the application performance (is it the data access schema? the arrival frequency of the products? the size of the product queue?...), and what is their degree of impact. These factors could further be used as a criterion for selecting the appropriate information to be stored on the product, at different stages of the PLC. For instance, in [9], the degree of data relevance is computed based on several criteria (actor's expectations, product's location,...), where new ones could be defined (e.g. a criterion reflecting the impact on the time when accessing data from the product instead of accessing it from the database).

Acknowledgement

This work is supported by OPNET via "Teaching with OPNET" program. The authors gratefully acknowledge the financial support of the CPER 2007-2013.

References

1. Wong, C.Y., Mcfarlane, D., Zaharudin, A.A., Agarwal, V.: The Intelligent Product Driven Supply Chain. In: International Conference on Systems, Man and Cybernetics. (2002) 4–6
2. McFarlane, D., Giannikas, V., Wong, A.C., Harrison, M.: Product intelligence in industrial control: Theory and practice. Annual Reviews in Control (2013)
3. Främling, K., Holmström, J., Loukkola, J., Nyman, J., Kaustell, A.: Sustainable plm through intelligent products. Engineering Applications of Artificial Intelligence **26**(2) (2013) 789 – 799
4. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer Networks **54**(15) (2010) 2787–2805
5. Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S.: Vision and challenges for realising the Internet of Things. Cluster of European Research Projects on the Internet of Things, European Commision (2010)
6. Kiritsis, D.: Closed-loop plm for intelligent products in the era of the internet of things. Computer-Aided Design **43**(5) (2011) 479–501
7. Meyer, G., Främling, K., Holmström, J.: Intelligent products: A survey. Computers in Industry **60**(3) (2009) 137–148
8. Jun, H.B., Suh, H.W.: Decision on the memory size of embedded information systems in an ubiquitous maintenance environment. Computers & Industrial Engineering **56**(1) (2009) 444–451
9. Kubler, S., Derigent, W., Thomas, A., Rondeau, E.: Embedding data on "communicating materials" from context-sensitive information analysis. Journal of Intelligent Manufacturing **10.1007/s10845-013-0745-y** (2013)
10. Suzuki, S., Harrison, M.: Data synchronization specification. aerospace-id program report. Technical report, Auto-ID Labs, University of Cambridge. (2006)
11. Chan, D., Roddick, J.: Context-sensitive mobile database summarisation. In: 26th Australasian computer science conference. Volume 16. (2003) 139–149
12. Dongkai, Y., Wenli, L.: The wireless channel modeling for rfid system with opnet. In: Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing. (2009) 3803–3805
13. Hababeh, I., Bowring, N., Ramachandran, M.: A method for fragment allocation design in the distributed database systems. In: The 6th Annual UAE University Research Conference. (2005)