



HAL
open science

Evaluation de l'influence d'un réseau de communication sans fil sur la commande d'un SED

Gilbert Habib, Pascale Marangé, Jean-François Pétin, Thierry Divoux

► **To cite this version:**

Gilbert Habib, Pascale Marangé, Jean-François Pétin, Thierry Divoux. Evaluation de l'influence d'un réseau de communication sans fil sur la commande d'un SED. Modélisation des systèmes réactifs, Nov 2009, nantes, France. pp.855-870. hal-00438902

HAL Id: hal-00438902

<https://hal.science/hal-00438902>

Submitted on 5 Dec 2009

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.

Evaluation de l'influence d'un réseau de communication sans fil sur la commande d'un SED

Gilbert Habib, Pascale Marangé, Jean François Pétin, Thierry Divoux

*Centre de Recherche en Automatique de Nancy (CRAN),
UMR 7039 Nancy Université - CNRS,
Campus scientifique, BP 70239, 54506 Vandœuvre-lés-Nancy, France
{gilbert.habib, pascale.marange, jean-francois.petin, thierry.divoux}@cran.uhp-nancy.fr*

RÉSUMÉ. Cette communication porte sur la conception de la commande de SED dont l'implantation est distribuée sur des automates programmables et/ou boîtiers d'entrées/sorties déportés communiquant via un réseau sans fil Wi-Fi. L'intégration de la communication sans fil dans ces applications apporte de nombreux avantages tels qu'une plus grande mobilité, une réduction des coûts de câblage, mais aussi plusieurs inconvénients tels que le délai de transmission, les phénomènes de gigue et les pertes des messages. L'objectif est de poser les premiers éléments d'une démarche de co-conception de ce type d'application permettant de prendre en compte la qualité de service d'une communication sans fil de type 802.11b et son impact sur les propriétés comportementales du système commandé en réseau, notamment en termes de réactivité et de déterminisme.

ABSTRACT. This communication focuses on the design of DES control whose implementation is distributed on PLCs and/or remote I/O communicating through a wireless Wi-Fi network. Using wireless communication in these applications has many advantages such as greater mobility, reduced wiring costs, but also many drawbacks such as the transmission delay, jitter phenomena and messages loss. The objective of this paper is to lay the foundations of a co-design approach that takes into account the quality of 802.11b communication service and its impact on the behavioral properties of the controlled system, especially in terms of reactivity and determinism.

MOTS-CLÉS : Système discret commandé en réseau, communication sans fil, performances temporelles, simulation

KEYWORDS : Networked Discrete Control System, wireless communication, temporal performance, simulation.

1. Introduction

Les nouvelles technologies de l'information et de la communication sont aujourd'hui intégrées dans la plupart des offres en automatismes industriels et conduisent à des architectures de commande distribuées basées sur la coopération d'objets logiciels dans lesquelles la communication joue un rôle prépondérant. Cette communication porte sur la conception de la commande de systèmes à événements discrets (SED) dont les entrées/sorties sont des signaux logiques et dont l'implantation est distribuée sur des automates programmables et/ou boîtiers d'entrées/sorties déportés communiquant via un réseau sans fil Wi-Fi. L'objectif est de poser les premiers éléments d'une démarche de co-conception de ce type d'application, qualifiée de Systèmes Discrets Commandés en Réseau (SDCR), permettant de prendre en compte les incertitudes sur la qualité de service de la communication de type 802.11b et leurs impacts sur les propriétés comportementales du SDCR, notamment en termes de réactivité et de déterminisme.

L'approche proposée contribue à cet objectif :

- en identifiant les principaux paramètres influant (temps de cycle des équipements de commande, période de scrutation des entrées/sorties, taille des paquets, nombre de retransmission autorisée en cas de pertes...) ;
- en quantifiant, de manière analytique, les temps de réponse dans les pires cas (minimum et maximum) sur le cycle complet d'un système réactif (réception d'un stimuli en provenance des capteurs et/ou des interfaces de commande, traitement, et émission d'un ou plusieurs signaux à destination des actionneurs) en intégrant les caractéristiques du protocole 802.11b ;
- en validant les résultats obtenus de manière analytique par un ensemble de simulations réalisées sous l'environnement Matlab et en exploitant la bibliothèque TrueTime puis en évaluant l'influence des paramètres du réseau sur les temps de réponse d'un cas d'étude.

La section 2 présente les principales caractéristiques des dispositifs composant une architecture industrielle et présente les travaux menés, d'une part sur l'évaluation de la qualité de service des réseaux de communication et d'autre part sur la prise en compte du comportement induit par les réseaux dans la modélisation des lois de commande d'un SED. La section 3 présente une estimation analytique du délai maximum et minimum de bout en bout entre la source et la destination d'un message. La section 4 aborde l'approche par simulation qui complète cette étude et présente les différents modèles utilisés. Dans la section 5, un ensemble de simulation réalisé sur un cas d'étude permet de valider les résultats analytiques et d'évaluer les temps de réponse depuis l'émission d'un message par la partie opérative (PO) jusqu'à la réception d'un message de commande émis en réaction. L'analyse des résultats obtenus a mis en évidence des plages de paramètres du réseau permettant d'atteindre les objectifs comportementaux de la commande. La section 6 présente quelques conclusions et perspectives de ce travail.

2. Système Discret Commandés en Réseau : contexte et état de l'art

Dans la suite de cet article, nous considérons les SDCR comme un ensemble de lois de commande discrètes pilotant une partie opérative dont entrées et les sorties sont constitués de signaux booléens. L'implantation des lois de commande est supportée par un ensemble d'Automates Programmables Industriels (API) et/ou de boîtiers d'entrées/sorties déportés échangeant de l'information via des réseaux de communication.

Compte tenu des algorithmes de traitement intégrés dans les différentes couches OSI d'un réseau et des problèmes de partage de ressource lors d'accès concurrents à un même médium de communication, l'échange d'information via un réseau subit inévitablement des phénomènes tels que des délais de bout en bout entre source et destination, de gigue lorsque ceux-ci sont variables, de pertes de messages dus à des problèmes de collision entre messages ou au bruit... Dans le cadre des applications de commande, l'ensemble du comportement induit par la communication se doit d'être maîtrisé si l'on veut pouvoir garantir les propriétés de sûreté, de réactivité ou de déterminisme de la commande (Greifeneder *et al.*, 2005). En particulier, ces applications sont souvent soumises à de fortes contraintes temporelles qui imposent un temps de réaction borné à un stimulus d'entrée (par exemple la réaction à un arrêt d'urgence dans le cas d'applications utilisant une logique programmée pour ses fonctions de sécurité). Dans le domaine particulier des réseaux de communication sans fil, les problèmes posés sont sensiblement équivalents. Pour le protocole Wi-Fi (802.11b), l'accès au médium est géré par un temps d'attente aléatoire compris entre deux valeurs minimale et maximale (*Backoff*) ne garantissant pas l'absence de collision (la probabilité de tirer deux *backoff* identiques peut être considérée comme faible). De plus, ces réseaux sont, par nature, très sensibles à leur environnement : champs magnétiques/électriques, stations cachées...

De nombreux travaux ont abordé la problématique de maîtrise de la qualité de service offerte par les réseaux de communications. Ils sont généralement basés sur des approches probabilistes telles que les réseaux de files d'attente ou les chaînes de Markov (Greifeneder *et al.*, 2006) pour estimer les principales caractéristiques comportementales du réseau. D'autres approches déterministes comme le Network Calculus (Georges *et al.*, 2005) permettent d'estimer les délais de transmission en se plaçant dans le pire des cas. Certains travaux se sont attachés à identifier les variables de commande du réseau (réservation mémoire, réservation des chemins, mécanismes de priorité, redondance des liens) influant sur la qualité de service. Les travaux de (Tipsuwan *et al.*, 2009) proposent une répartition équitable de la bande passante du réseau entre les équipements du système, son algorithme est appliqué à une application continue (non discret), l'utilisation de sa méthode sur des applications discrets peuvent donner des résultats différents que celle des applications continues. Une comparaison entre différents protocoles de routage a été faite par (Hasan *et al.*, 2005) pour voire l'influence de ces protocoles sur le délai et le taux de perte. Dans notre étude, on travaille avec des systèmes en mode

infrastructure, donc la couche réseau n'est pas dans notre intérêt. Dans le domaine des réseaux de communication sans fil, la solution proposée par (Lin *et al.*, 2007) définit dynamiquement une valeur du *backoff* en fonction de la moyenne des taux de collision, l'approche proposée par (Tresk *et al.*, 2006) exploitant, quant à elle, les possibilités offertes par le 802.11 pour gérer des priorités entre les flux. Ces approches ne sont généralement pas analytiques mais reposent sur des scénarii de simulation réalisées sur OPNET (http://www.opnet.com/university_program/).

De manière complémentaire à ces travaux développés au sein de la communauté scientifique des réseaux de communications, cette problématique a également été abordée sous l'angle de la commande des systèmes. Dans le domaine des systèmes continus, des résultats ont été obtenus en appliquant au réseau des approches issues de l'Automatique, la qualité de service étant considérée comme la consigne (Vatanski *et al.* 2009). Dans le domaine de l'Automatique des SED, la modélisation de la commande basée sur des modèles à états, souvent des réseaux de Petri temporisés et/ou colorés, est complétée par un modèle du réseau souvent restreint à un état d'attente représentant le délai de communication (Marsal *et al.*, 2005) (Addad *et al.*, 2008). Le principal inconvénient de ces travaux est que le réseau est représenté comme une simple place avec un retard fixe, sans prendre en considération une modélisation comportementale plus précise des protocoles et des différents paramètres qui leur sont associés.

De manière générale, l'ensemble de ces travaux aborde la problématique des SDCR avec un point de vue spécifique relatif au réseau de communication ou à la commande des systèmes. Cet article propose une tentative d'intégration de ces deux points de vue, avec une représentation réelle du réseau sans fil où le délai varie en fonction des différents paramètres du réseau tels que le temps de cycle des équipements de commande, la période de scrutation des entrées/sorties, la taille des paquets, le nombre maximal de retransmission autorisée en cas de pertes... En ce sens, il constitue une première étape vers une démarche de co-conception des SDCR.

3. Estimation analytique du délai maximum et minimum

Considérons le SDCR représenté en figure 1. L'automate programmable lit les variables d'entrées, les traite et met à jour les variables de sorties selon un cycle caractérisé par une période P_{API} . Tous les équipements en réseau sont munis d'un coupleur de communication caractérisé par un cycle de période nommé P_{carte} .

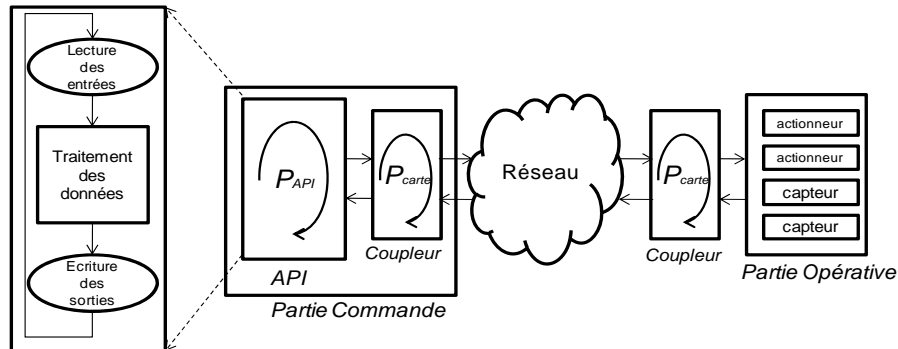


Figure 1. Description du SDCR

Les flux sont majoritairement périodiques avec quelques échanges asynchrones tels que les signaux d'alarmes. Un échange ordinaire se déroule de la manière suivante : (i) Le capteur transmet périodiquement des données vers l'API via le réseau, (ii) l'API les traite et (iii) envoie des signaux de commande aux actionneurs en fonction des lois de commande implantées.

Le délai de bout en bout, noté T , entre l'occurrence de l'événement généré par un capteur et la réception par un actionneur du signal de commande émis en réaction, est calculé en additionnant tous les retards survenus au cours de la transmission des données depuis la source (capteur) vers le destinataire (actionneur) à travers les différents équipements intermédiaires. Nous faisons l'hypothèse que n retransmissions d'un message, n étant le paramètre fixant le nombre maximal de retransmissions autorisés en cas de pertes, pourront être réalisées avant la période d'émission des messages par la PO, c'est-à-dire la période de scrutation P_{carte} associée. La figure 2 décrit les étapes pour le calcul de ce délai :

- l'occurrence d'un événement sur la PO a lieu à la date t ;
- l'émission du message associé est retardé d'un temps a représentant le temps pour que le coupleur de communication prenne en compte cet événement ;
- l'information est transmise au destinataire (API) avec un retard $d1$;
- le message est lu par l'API avec un autre retard b dû à la période de scrutation du coupleur de communication de l'API ;
- le message est ensuite traité par l'API, l'exécution des lois de commande pouvant nécessiter plusieurs cycles (k) avant l'obtention des signaux de commande,
- le message de commande subit ensuite un retard c dépendant de la période de scrutation du coupleur de communication ;
- enfin, le message arrivera à destination avec un retard $d2$ à la date T_{finale} .

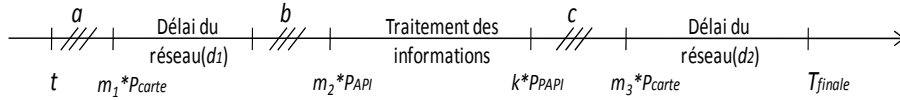


Figure 2. Les différentes étapes pour présenter le délai estimé

Le temps de réponse de bout en bout depuis l’occurrence d’un événement capteur jusqu’à la réception du message associé de commande par un actionneur sera donné par $T = T_{finale} - t$ selon :

$$T = a + d_1 + b + k * P_{API} + c + d_2 \tag{1}$$

$$\begin{cases} a = m_1 * P_{carte} \\ b = m_2 * P_{API} \\ c = m_3 * P_{carte} \end{cases} \text{ avec } 0 \leq m_i \leq 1, i = 1, 2, 3$$

où d_1 et d_2 représentent le retard causé par le réseau. Le temps de traitement est égal à $k * P_{API}$ avec $0 < k < k_1$, où k_1 représente le nombre maximum de cycle pour le traitement de l’information.

Nos précédents travaux (Habib *et al.*, 2009) ont montré que le délai de bout en bout entre émetteur et récepteur pour un réseau sans fil IEEE 802.11e en mode EDCA était toujours borné entre D_{min} et D_{max} et ont proposé une estimation de ces valeurs (le mode HCCA est inclut dans la norme mais il n’existe pas aujourd’hui d’équipements qui l’implémente). Sur la base de l’algorithme CSMA/CA avec priorités d’accès utilisé par EDCA et d’après (Standard, 2005), lorsqu’un message arrive dans la file d’attente du coupleur de communication, son accès au médium est géré comme suit :

- si la file d’attente est vide et que le médium est libre, le paquet est envoyé immédiatement ;
- sinon, un temps d’attente nommé *backoff time* est tiré de manière aléatoire entre deux valeurs minimales et maximales et est associé au message. Ces valeurs minimales et maximales dépendent de la priorité du trafic. La valeur de *backoff* est décrétementée lorsque le médium est libre et reste maintenue à sa valeur courante lorsqu’une autre station occupe le médium. Lorsque le *backoff time* atteint la valeur nulle, la station accède au médium.

Le délai minimum (D_{min}) considéré par (Habib *et al.*, 2009) est égal au temps nécessaire pour envoyer un paquet de l’émetteur vers le récepteur sans *backoff time* et sans retransmission nécessitée par d’éventuelles pertes. Le délai maximum (D_{max}) est estimé en tenant compte du nombre maximal de retransmission autorisé et en considérant que, pour chaque transmission, le *backoff time* alloué est maximal. Ces estimations conduisent à encadrer les valeurs d_1 et d_2 de l’équation [1] :

$$D_{1min} \leq d_1 \leq D_{1max}$$

$$D_{2min} \leq d_2 \leq D_{2max}$$

T est minimum lorsque l'occurrence de l'événement émis par le capteur a lieu de manière synchrone à la période du coupleur de communication à une date multiple de P_{carte} . Dans ce cas, la valeur de a sera donc nulle. Cette information traverse le réseau en un temps minimum D_{1min} . De la même manière, le temps T sera minimum si l'arrivée du message au niveau du coupleur de l'automate programmable est synchrone avec la période de ce dernier avec $b = 0$, si l'API traite cette information en un seul cycle, et enfin si le message est instantanément émis ($c = 0$). Le destinataire (PO) reçoit la réponse après un délai minimum D_{2min} . La valeur de T minimale est donc donnée par l'équation [2] suivante :

$$T \geq D_{1min} + P_{API} + D_{2min} \quad [2]$$

A l'inverse, T est maximum lorsque les coupleurs lisent où émettent les informations avec un temps maximal égal à P_{carte} . De manière similaire, les temps de transmissions (aller et retour) seront choisis respectivement égaux D_{1max} et D_{2max} . Enfin, le traitement des informations par l'API nécessitera un nombre maximum de cycle égal à $k_1 + 1$ (1 cycle pour la prise en compte des entrées et k_1 cycles pour leur traitement). Le T maximal sera donc donné par l'équation [3] suivante :

$$T \leq P_{carte} + D_{1max} + k_1 * P_{API} + P_{carte} + D_{2max} \quad [3]$$

Notons que cet intervalle dépend de P_{API} , P_{carte} , et indirectement au travers des valeurs établies D_{1min} , D_{2min} , D_{1max} and D_{2max} , qui dépendent de la charge du réseau, du nombre maximum de retransmission des paquets et de la taille des paquets (Habib *et al.*, 2009).

Si ces calculs analytiques permettent d'évaluer les temps de réaction dans les pires des cas, ils sous-estiment considérablement les possibilités du réseau et ne permettent pas de mesurer l'influence de l'ensemble de ses paramètres sur la qualité de service de l'ensemble de la solution de commande, et en particulier des propriétés comportementales que celle-ci doit respecter.

En ce sens, une étude expérimentale basée sur l'exécution de scénarii de simulation complète donc cette étude analytique. La section suivante présente les modèles développés pour la simulation, la section 5 étant consacré à l'analyse des résultats.

4. Modèles pour la simulation

La simulation d'un SDCR repose sur des modèles développés en Statecharts dans l'environnement Matlab (<http://www.mathworks.fr>) pour les parties concernant les lois de commande SED, le comportement opérationnel des équipements (API, coupleur de communication...) et une bibliothèque spécifique (TrueTime) pour la partie réseau.

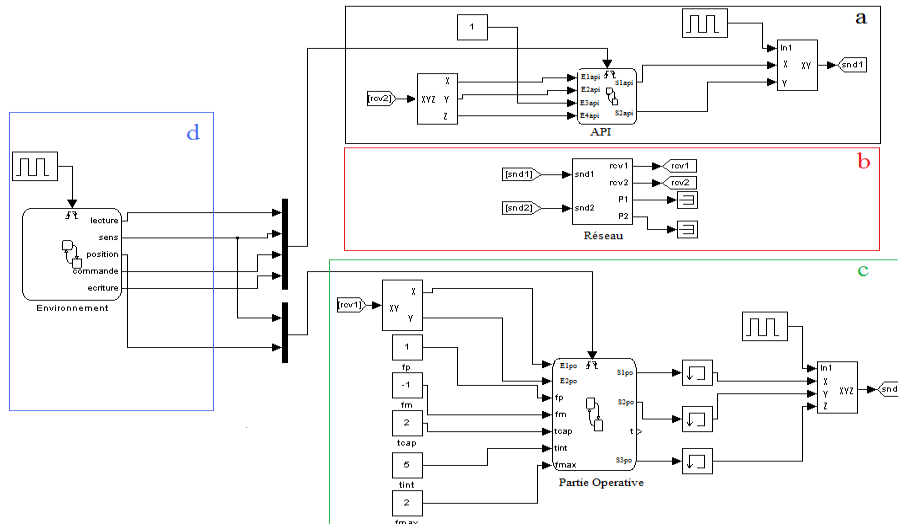


Figure 3. Modèles d'un SDCR

La figure 3 présente l'architecture construite autour de quatre principaux modèles :

- le *modèle de commande* représentant les lois de commande, le fonctionnement cyclique de l'API et la gestion des paquets par le coupleur de communication (figure 3a) ;
- le *modèle de partie opérative* représentant l'état des actionneurs, des capteurs et la gestion des paquets par le coupleur de communication (figure 3c) ;
- le *modèle d'environnement* assurant la synchronisation entre les modèles de commande et de PO (figure 3d) ;
- le *modèle du réseau* représentant le comportement induit par le réseau lors des échanges d'informations entre partie commande et PO (figure 3b).

4.1. Modèle d'environnement

Le *modèle d'environnement* (figure 3d) orchestre l'exécution du modèle de commande, du modèle de PO et l'exécution de l'API (figure 4). Les événements {*top*, *lecture*, *direction*, *position*, *commande*, *écriture*} représentent respectivement : l'horloge, la mise à jour des entrées de l'API, le calcul du sens de déplacement, l'évolution de la position de la PO, l'évolution de la commande, la mise à jour des sorties de l'API. Le modèle d'environnement représente d'une part le fonctionnement classique de l'API : lecture des entrées, évolution de la commande et écriture des sorties et d'autre part l'évolution en parallèle de la PO, c'est-à-dire le changement de positions de la PO en fonction du sens de déplacement.

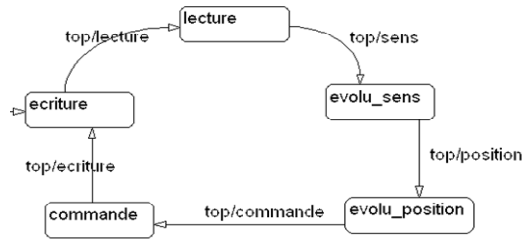


Figure 4. Modèle cyclique de l'environnement

4.2. Modèle de commande

Le modèle de commande (figure 3a) utilise les variables booléennes suivantes : les variables émises et reçues par la PO seront nommées respectivement $\{E_i, E_j, \dots; S_i, S_j, \dots\}$, les variables reçues et émises par l'API seront respectivement nommées $\{E_{iAPI}, E_{jAPI}, \dots; S_{iAPI}, S_{jAPI}, \dots\}$. L'API reçoit un message en provenance de la PO (rcv2), les extrait et les recopie en variables internes ($E_{iAPB}, E_{iAPI}, \dots$), les traite, puis encapsule les variables de sorties ($S_{iAPB}, S_{iAPI}, \dots$) dans un message envoyé périodiquement vers le réseau (snd1).

Le modèle de fonctionnement de l'API est représenté par trois modèles (figure 5):

- le modèle de lecture représente la mise à jour des variables de l'API. Quand un message arrive en entrée de l'API, l'API recopie son contenu dans ses variables internes selon "EiAPI = Ei, EjAPI = Ej ...";
- le modèle des lois de commande représente l'évolution de la commande ; ce modèle passe d'un état à l'autre lorsque la transition est vraie et qu'il réceptionne le message commande ;
- le modèle d'écriture représente les mises à jour des sorties selon "SiAPI = Si, SjAPI = Sj ...".

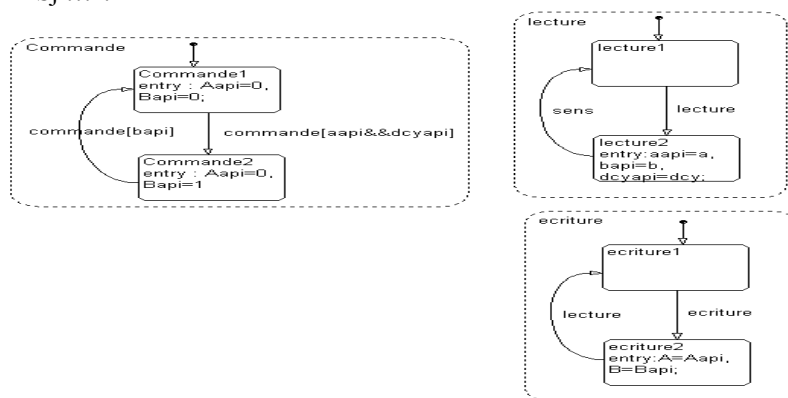


Figure 5. Modèle de fonctionnement de l'API : Lecture, Commande, Ecriture

4.3. *Modèle de partie opérative*

Le *modèle de partie opérative* (figure 3c) représente le comportement des actionneurs et des capteurs. La PO reçoit un message en provenance de l'API via le réseau (*rcv1*). Elle extrait de ce message les variables (E_{1API} , E_{2API} ...) de commande des actions qu'elle doit exécuter et les traite. Les événements émis par les capteurs (S_{1APB} , S_{2API} ...) sont encapsulés dans un message transmis vers l'API via le réseau (*snd2*).

Le *modèle de partie opérative* est défini par deux modèles : *modèle de direction* et *modèle de position*. Le *modèle de direction* calcule les actions exécutées par les actionneurs (par exemple la force pouvant causer un déplacement). Les effets causés par ces actions sont pris en compte par le *modèle de position* représentant les états de la PO et/ou du produit et générant les événements capteurs en fonction des états de la PO et/ou du produit. Pour plus de détails concernant ces modèles, le lecteur peut se reporter aux travaux de (Marangé, 2008).

4.4. *Modèle du réseau*

Le *modèle du réseau* (figure 3b) est simulé en utilisant la librairie TrueTime (<http://www.control.lth.se/truetime/>) dans Matlab. Cette librairie est développée par l'université LUND en utilisant le C++ Mex. Elle supporte deux types de réseaux : IEEE 802.11b/g (WLAN) et IEEE 802.15.4 (ZigBee) avec quelques modifications. Cette librairie simule les réseaux sans fil en prenant en considération les différents paramètres comme le nombre maximum de retransmission des paquets, la charge du réseau, la taille des paquets et *Acktimeout* (le temps attendu par l'émetteur pour recevoir le paquet d'acquittement avant d'envoyer le paquet de nouveau). Notons que dans cette architecture, il y a deux trafics cycliques. Le premier du capteur vers la PC (*rcv1*) et le deuxième du PC (*snd2*) vers la PO (*rcv2*).

5. Application

5.1. *Cas d'étude*

Les modèles de simulation présentés en section 4 sont particularisés sur un cas d'étude constitué d'un vérin pneumatique équipé d'un distributeur 5/3 et de dispositifs en réseau permettant sa commande. Notre système (figure 6) se compose d'un vérin pneumatique équipé d'un distributeur 5/3 et de trois capteurs de position S_1 , S_2 et S_3 , d'un API où sont implantés les lois de commande du système (les signaux *A* et *B* déclenchent respectivement la rentrée et la sortie du vérin), et d'un système de communication Wi-Fi permettant l'échange d'informations entre un point d'accès et deux stations (l'API et un boîtier d'entrées/sorties déporté).

L'ensemble des commandes (A et B) et des observations (S_1 , S_2 et S_3) sont émises de manière périodique.

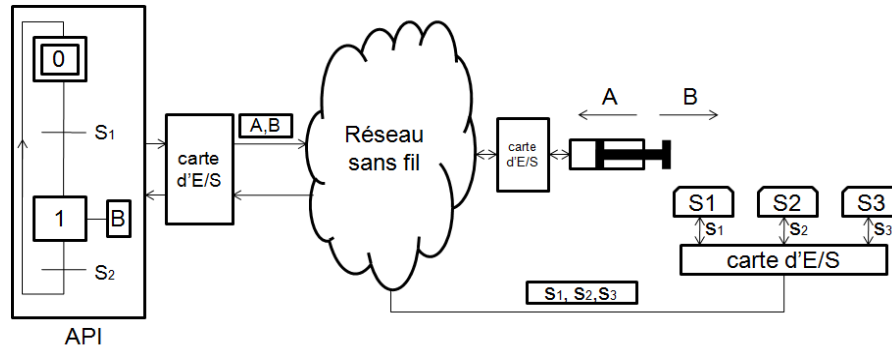


Figure 6. Cas d'étude

Les différents scénarii de simulation évaluent le délai T_s entre l'occurrence d'un événement sur les capteurs S_1 , S_2 ou S_3 et les effets produits sur la PO par la réception des signaux de commande A ou B . Ce délai tient bien sûr compte du comportement induit par le réseau de communication mais également des phénomènes d'inertie de la PO selon les propositions de (Marangé, 2008). Les paramètres des scénarii sont les suivants : la charge du réseau, la taille des paquets, le nombre de retransmission maximal, la période de scrutation des coupleurs de communication et la période de l'API. Deux objectifs successifs sont poursuivis. Dans un premier temps, il s'agit de vérifier que les temps de réponse sont bien compris entre les valeurs analytiques T_{smin} et T_{smax} de la section 3. Dans un deuxième temps, il s'agit d'évaluer les paramètres permettant de satisfaire au mieux les objectifs de la commande. Pour cela, nous avons considéré le scénario suivant : à partir de la position initiale S_1 , le vérin se déplace vers la position intermédiaire S_2 sur laquelle il doit s'arrêter. Le délai T_s que l'on cherche à évaluer correspond donc au temps compris entre l'émission d'un signal par le capteur S_2 et l'arrêt du vérin après réception d'un front descendant sur la commande B .

5.2. Validation des résultats analytiques

5.2.1. Résultats analytiques

La figure 7 présente le scénario retenu : au temps t , le vérin arrive au capteur S_2 (occurrence de l'événement s_2), un message est envoyé à l'API qui répond par un message de commande ($B=0$) avec un délai T égal à $T_{finale}-t$ selon les calculs

analytiques présentés en section 3, ce qui, au final, provoque l'arrêt du vérin après un temps d'inertie $T_{inertie}$. Ainsi

$$T_s = t + T + T_{inertie} \quad [4]$$

Le temps T_s (T_{smax}) correspond valeurs maximales des temps t , T et $T_{inertie}$. En d'autres termes, la valeur maximale de t (t_{max}) est le temps maximum requis par le vérin pour atteindre et s'arrêter sur le capteur S_2 . La valeur maximale de $T_{inertie}$ ($T_{maxinertie}$) correspond à deux cycles de l'API compte tenu des caractéristiques technologiques du capteur (plage de détection) et du vérin (vitesse de course). Pour plus de détails, se référer à (Marangé, 2008).

$$T_{maxinertie} = 2 * P_{API} \quad [5]$$

La valeur maximale de T a été calculée en section 3 par l'équation [3]. Dans notre cas, les paramètres du réseau, les priorités et les périodes de scrutation des coupleurs de communication sont identiques pour les messages émis par la PO à destination de l'API ou inversement d'où $D_{1max} = D_{2max}$. Enfin, dans la commande proposée, l'API génère la commande appropriée en un seul cycle de calcul. La valeur maximale de T peut donc être donnée par :

$$T_{max} = 2 * P_{carte} + 2 * D_{1max} + 4 * P_{API} \quad [6]$$

La valeur T_{smax} peut donc être établie comme suit :

$$T_{smax} = t_{max} + 2 * P_{carte} + 2 * D_{1max} + 2 * P_{API} + T_{inertie} \quad [7]$$

De la même manière, la valeur de T_s sera minimale (T_{smin}) pour la valeur minimale de T donnée par l'équation [2] avec D_{1min} égal à D_{2min} et pour un temps d'inertie supposé nul. Il en découle que :

$$T_{smin} = t_{min} + 2 * D_{1min} + P_{API} \quad [8]$$

5.2.2. Résultats de simulation

Dans notre cas, les pertes de paquets causées par les interférences, les obstacles et les radiations ne sont pas considérées. La charge du réseau, le nombre maximum de retransmission des paquets et P_{API} sont respectivement fixés à 8000 bits/s, 1 et 0.02s. L'influence de la P_{carte} sur T_s pour trois tailles de paquets égale à 104, 248 et 536 bits est étudiée. Les résultats présentés par la figure 7 permettent de confirmer que la valeur T_s reste bien comprise entre les valeurs T_{smin} et T_{smax} calculées de manière analytique. Une exception apparaît cependant pour la courbe concernant les tailles de paquets égales à 536 bits. Dans ce cas, la valeur de T_{smax} est inférieure à la valeur T_s simulée lorsque la période de la P_{carte} est inférieure à 0,1s. Cette différence s'explique par la charge du réseau qui est importante dans ce cas et qui peut conduire à des pertes de paquets dus à des collisions. Par conséquent, l'hypothèse considérant que la probabilité de tirage de *backoff* identiques est négligeable est trop

forte dans cette configuration. Pour lever cette hypothèse, une solution consiste à considérer que le délai maximum entre l'émetteur et le récepteur est égal à :

$$d_i = p * P_{carte} + D_{max} \tag{9}$$

où p est le nombre de périodes P_{carte} pendant lequel le signal capteur ne varie pas.

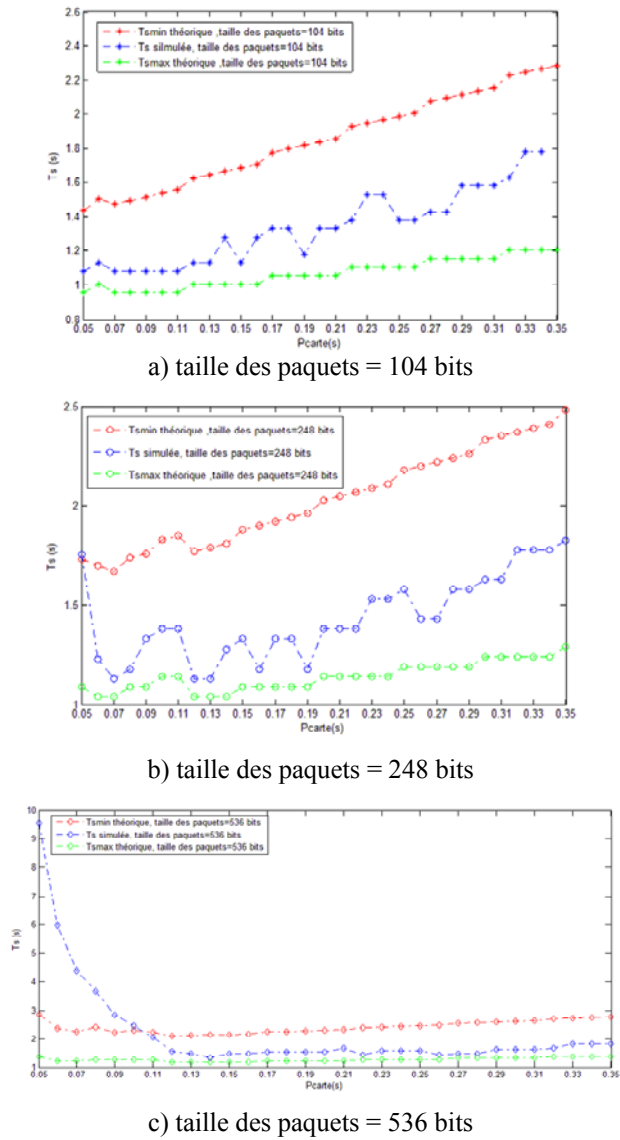


Figure 7. Valeur de T_s , T_{smin} et T_{smax} en fonction de P_{carte}

5.3. Influence des paramètres sur le comportement du système commandé

Nous nous intéressons maintenant aux propriétés comportementales du SDCR. Pour atteindre les objectifs de la commande, c'est-à-dire que le vérin s'arrête effectivement sur le capteur S2, et compte tenu des caractéristiques technologiques de la PO (vitesse du vérin, inertie, plage du capteur...), il a été mesuré que le T_s doit être inférieur à 1,33s ($T_{arr\acute{e}tcylindre}$). La valeur T_s a été évaluée par simulation pour trois tailles de paquets égale à 104, 248 et 536 bits. Les résultats obtenus sont présentés dans la figure 8.

Dans le cas où la taille des paquets est égale à 536 bits (figure 7c), trois zones peuvent être identifiées :

- $0,05 < P_{carte} \leq 0,14s$: dans cet intervalle, l'augmentation de la période du coupleur fait décroître fortement T_s ; ceci peut s'expliquer par le fait la charge générée sur le réseau est réduite, ce qui entraine une diminution des délais de transmission.
- $0,14 < P_{carte} \leq 0,35s$: dans cet intervalle la charge du réseau est faible et le temps de réponse reste relativement constant mais très légèrement supérieure au temps maximum admissible.
- $P_{carte} > 0,35s$: le cylindre ne s'arrêtera jamais parce que le capteur S2 n'envoie pas l'information indiquant la présence du cylindre ($s_2 = 1$) à la partie commande. En effet, ceci correspond au cas où la durée de présence du vérin sur la zone de détection du capteur est inférieure à la période de scrutation.

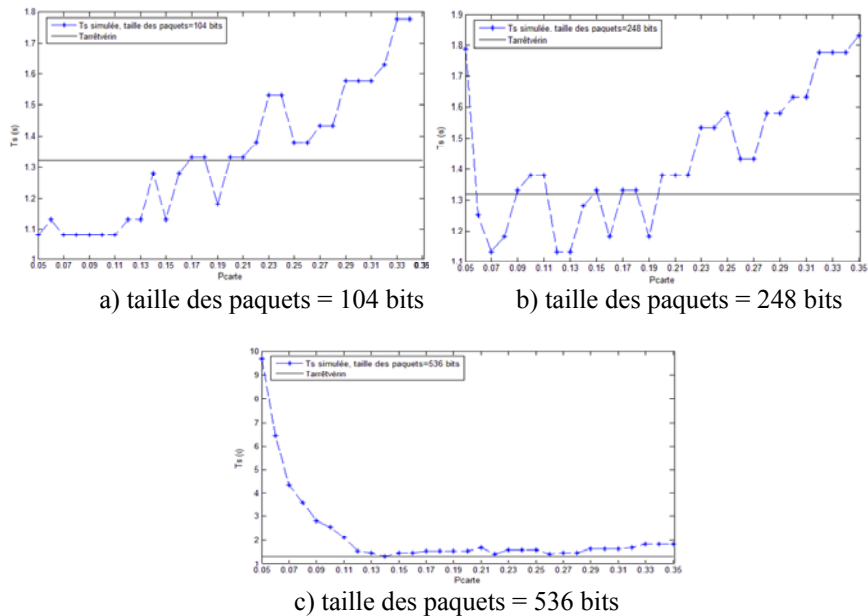


Figure 8. Variation de T_s en fonction de P_{carte}

On peut donc en conclure qu'une taille de paquet trop important (en l'occurrence dans notre cas 536 bits) peut conduire à un comportement non admissible du système commandé. En revanche, les simulations correspondant aux tailles de paquets égales à 104 et 236 bits proposent, quant à elles, quelques plages de paramètres admissibles. En effet, pour certaines valeurs de P_{carte} , la valeur de T_s est inférieur au temps maximum admissible de $T_{arrêtecylinder}$ donc vérifie l'objectif initial de la commande. C'est donc ces valeurs qu'il conviendra de choisir lors de la définition des paramètres du SDCR.

6. Conclusion et perspectives

Dans ce papier, nous avons étudié l'influence du réseau sans fil Wi-Fi (802.11e) dans les applications de commande de SED. L'objectif était de proposer une première tentative pour établir une convergence entre les approches basées sur une modélisation fine du réseau et de ses paramètres et les approches utilisées en automatique pour la modélisation des lois de commande et de la partie opérative. Les résultats ont montré la pertinence de l'approche dans la mesure où elle fournit des informations sur les performances temporelles d'un SDCR et les principales grandeurs influant sur sa qualité de service. Ces résultats doivent donc fournir aux concepteurs d'un SDCR reposant sur l'utilisation du Wi-Fi de précieuses recommandations pour le choix des différents paramètres du réseau.

Même si les paramètres étudiés sont encore incomplets, notamment au niveau de la gestion des priorités proposée par le 802.11e et de la prise en compte des pertes de paquets dus au bruit, la démarche proposée doit s'intégrer efficacement dans un processus de co-conception « commande/réseau » qui fait l'objet de travaux en cours. Ceci passe, en outre, par la modification de la bibliothèque TrueTime pour disposer d'une modélisation plus fine du protocole 802.11e (couche physique, MAC et application) et des pertes. D'autre part, nous avons supposé, dans ce travail, que la commande SED était définie et considérée comme connue, et que seuls les paramètres du réseau constituait les degrés de liberté du concepteur. Il est bien entendu qu'un travail similaire pourrait être conduit, du côté de la commande afin de définir des stratégies adaptatives notamment au travers :

- de la définition d'un estimateur qui reconstruit les informations manquantes lors de la perte des paquets ;
- de la priorisation dynamique de messages considérés comme attendus en fonction des états de la commande ;
- d'une évaluation dynamique des performances du réseau qui tienne compte des évolutions dans son environnement (bruit, nombre de stations entrant dans le champ du point d'accès...).

7. Bibliographie

- Addad B., Amari S., Modeling and Response Time Evaluation of Ethernet-based control architectures using Timed Event Graphs and Max-Plus Algebra. *4th annual IEEE Conference on Automation Science and Engineering, Washington, USA, 2008.*
- Georges J., Divoux T., Rondeau E., Confronting the performances of a switched Ethernet network with industrial constraints by using the network calculus, *International Journal of Communication Systems*, 18:877903, July 2005.
- Greifeneder J., Frey G., Probabilistic delay time analysis in networked automation systems. *ETFA'05 : 10th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 1065–1068, 2005.
- Greifeneder J., Frey G., Optimizing Quality of Control in Networked Automation Systems using Probabilistic Modèles. *ETFA'06 : 11th IEEE International Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic, Sept 2006.*
- Habib G., Divoux T., Pétrin J., Estimating Maximum and Minimum Delays for Wireless Discrete Networked Control Systems. *Wireless Telecommunication Symposium (WTS'09), Prague, Czech Republic, 2009.*
- Hasan M.S., Harding C., Yu H., Griffiths A., Modeling Delay and Packet Drop in Networked Control Systems Using Network Simulator NS2, *International Journal of Automation and Computing* 2, 187-194, 2005.
- Lin W., Wu J., Modified EDCF to improve the performance of IEEE 802.11e WLAN. *Computer Communications*, 30:841–848, 2007.
- Marangé P., Synthèse et filtrage robuste de la commande pour des systèmes manufacturiers surs de fonctionnement. *Thèse de doctorat, Université de Reims Champagne-Ardenne, 2008.*
- Marsal G., Denis B., Faure J., Evaluation des délais de réactivité des architectures de commande distribuées sur réseau Ethernet. *Revue des Sciences et Technologies de l'Automatique, e-STA*, 4, 2007.
- Standard IEEE Std 802.11e: IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, 2005.
- Tipsuwan Y., Kamonsantiroj S., Chongstitvattan P., An auction-based dynamic bandwidth allocation with sensitivity in a wireless networked control system. *Computers & Industrial Engineering* 57, 114–124, 2009.
- Trsek H., Jasperneite J., Karanam S. A Simulation Case Study of the new IEEE 802.11e HCCA mechanism in Industrial Wireless Networks. *ETFA'06 : IEEE Conference on Emerging Technologies and Factory Automation*, 20-22:921 – 928, 2006.
- Vatanski N., Georges J.-P., Aubrun C., Rondeau E., Jämsä Jounela S.-L., Networked control with delay measurement and estimation, *Control Engineering Practice* 17, 2009.