HAL
open science

Analyse de performances de protocoles temps-réel basés sur Ethernet

Jérémy Robert, Jean-Philippe Georges, Eric Rondeau, Thierry Divoux

To cite this version:

Jérémy Robert, Jean-Philippe Georges, Eric Rondeau, Thierry Divoux. Analyse de performances de protocoles temps-réel basés sur Ethernet. Sixième Conférence Internationale Francophone d'Automatique, CIFA 2010, Jun 2010, Nancy, France. pp.CDROM. hal-00489939

HAL Id: hal-00489939

https://hal.science/hal-00489939v1

Submitted on 7 Jun 2010

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.

Analyse de performances de protocoles temps-réel basés sur Ethernet

Jérémy Robert[1], Jean-Philippe Georges[1], Eric Rondeau[1], Thierry Divoux[1]

[1]Centre de Recherche en Automatique de Nancy,
UMR 7039 – Nancy-Université, CNRS
Faculté des Sciences BP 239, 54506 Vandœuvre lès Nancy, France.
Prenom.Nom@cran.uhp-nancy.fr

Résumé— **Aujourd'hui, Ethernet se positionne comme la solution standard pour les communications industrielles en dépit de ses inconvénients intrinséques (particulièrement sa méthode d'accès au médium non déterministe). Comme pour les réseaux de terrains, chaque fournisseur réseaux fait la promotion de sa solution. En conséquence, de nombreuses solutions hétérogènes ont été proposées sur le marché afin de rendre Ethernet déterministe. Le défi est alors pour les industriels de sélectionner la technologie la plus appropriée tout en conservant un regard sur leur pérennité. L'objectif de ce papier est donc de comparer différents protocoles relativement à des contraintes de temps de cycle. Ce temps de cycle est calculé pour chaque solution.**

Mots-clés— **Systèmes temps-réel, analyse de performances, contrôleur d'axes, systèmes embarqués.**

I. Introduction

Historiquement, les bus de terrain étaient principalement conçus par les fournisseurs d'API (Automates Programmable Industriel). Les spécificités des contraintes de communications industrielles font que les bus de terrain requièrent un haut degré d'expertise. Ils doivent être robustes à l'environnement industriel (couche physique), déterministes afin de garantir le rafraîchissement des données dans le temps de cycle API (couche liaison de données) et intéropérables pour pouvoir échanger des informations entre tous les types d'équipements industriels (couche application). Les fournisseurs proposent donc différentes solutions à ces contraintes sans prendre en compte le dernier point, plus sensible en terme de business. Finalement, il existe une large gamme de bus de terrain ([6]).

La nouvelle tendance alors défendue par le consortium IAONA (Industrial Automation Open Networking Alliance) entre autres, est de promouvoir Ethernet comme le standard industriel. Les bénéfices sont la réduction des coûts d'installation du réseau (composants sur étagère) et la réduction des problèmes d'intéropérabilité. D'autres intérêts peuvent être cités : Ethernet est un protocole connu, largement implémenté et ses performances augmentent continuellement avec l'évolution des technologies (particulièrement sa bande passante).

Cependant, la méthode d'accès au médium s'appuie sur un algorithme non déterministe (CSMA/CD), qui utilise une méthode stochastique pour résoudre les collisions et qui ne garantit pas la transmission du message. Par conséquent, Ethernet ne peut pas être a priori implémenté dans un milieu industriel à forte contrainte temporelle. Afin de faire face à ce problème, différentes solutions ont été proposées, comme mentionnées dans [2], [4] et [5]. Chaque fournisseur fait la promotion de son produit avec pour but la standardisation de celui-ci. Même s'il existe différentes solutions basées sur Ethernet sur le marché, aucune d'entre elles n'est capable de communiquer avec les autres comme l'ont été les historiques bus de terrain.

En accord avec [9], ces différents protocoles peuvent être classés dans trois catégories. La classification se base à la fois sur les performances et sur les fonctionnalités ajoutées à Ethernet standard. La plupart des protocoles font face au problème de la couche MAC en ajoutant une sous-couche de priorité/ordonnancement dans laquelle des mécanismes de type TDMA ou maître/esclaves sont utilisés. Avec ces méthodes, le réseau transmet seulement un message à la fois et supprime ainsi tout problème de collisions. Le temps de transmission est simple à estimer, ce qui constitue le principal intérêt de ces protocoles comme le montrent [9] et [13]. Les résultats de ces articles sur la comparaison entre EtherCAT et Profinet IRT sont alors repris dans la section II et étendu aux solutions Modbus/TCP et EtherNet/IP. Ces protocoles correspondent aux solutions les plus répandues d'Ethernet industriel et l'évaluation porte sur les performances temporelles.

II. Estimation du temps de cycle minimal

A. Introduction

Cette partie présente une évaluation de performances de quatre solutions basées sur Ethernet et utilisées dans des milieux industriels, à savoir EtherCAT, Profinet IRT, Modbus/TCP et EtherNet/IP. Un scénario commun est défini : un unique calculateur agissant comme un maître collecte les différentes mesures et met à jour périodiquement les données. Les capteurs et actionneurs sont quant à eux considérés comme des esclaves. Le temps de cycle peut être divisé en trois partie : collecte des données (c'est à dire les données produites par les capteurs), calcul des actions et transmission de celles-ci. Par conséquent, le réseau doit être capable de supporter pendant un cycle automate les échanges relatifs aux entrées envoyées par les capteurs au contrôleur et les sorties envoyées par le contrôleur aux actionneurs. De ce point de vue, le temps de cycle doit être plus long que le temps nécessaire à tous ces échanges. Un temps de cycle minimum est alors défini de la façon suivante :

Definition 1(Temps de cycle minimum) : *Le temps de cycle minimum correspond au temps de communication nécessaire pour collecter et mettre à jour les mémoires de données de tous les capteurs et actionneurs.*

La définition précédente, similaire à celle proposée dans [9], sera utilisé dans cet article comme la métrique permettant de comparer la capacité des différentes solutions basées sur Ethernet à satisfaire les contraintes industrielles. Le temps de cycle minimal est calculé pour chaque protocole et exprimé en fonction de différents paramètres tels que le temps de transmission d'une trame, la latence d'un équipement réseau, la congestion ou même du temps de propagation d'un lien. Le tableau 1 décrit les notations utilisées. De plus, il sera supposé des communications sans erreurs de transmissions et un réseau dédié, c'est à dire non partagé avec d'autres applications.

TABLE I
NOTATIONS

Termes	Notation	Unités
Temps de transmission	τ	s
Latence de l'équipement	ℓ	s
Temps de propagation	δ	s
Débit	C	$bits/s$
Charge utile	x	$octets$
Nombre d'équipements (esclaves)	n	–

B. EtherCAT

EtherCAT est développé par Beckhoff (type 12 dans le standard IEC 61158, [7], [8]). En théorie, EtherCAT utilise des cartes Ethernet standard. Cependant en pratique, un matériel spécifique (FPGA : *Field-Programmable Gate Array* ou ASIC : *Application-Specific Integrated Circuit*) est utilisé pour minimiser le temps de traversée de la trame. EtherCAT ajoute un protocole Maître/esclaves au dessus d'Ethernet. La trame est envoyée par le maître et les esclaves peuvent lire et écrire à la volée (mécanisme *on the fly*). Le temps de lecture et d'écriture est indépendant de la taille de la trame et est identique pour tous les esclaves : ce temps correspond à la latence ℓ introduite par l'équipement. Un anneau logique est défini entre les esclaves tel que lorsque la trame atteint le dernier esclave (de l'anneau), elle est renvoyée au maître. Le diagramme spatio-temporel de la FIGURE 1 illustre le comportement des communications EtherCAT.

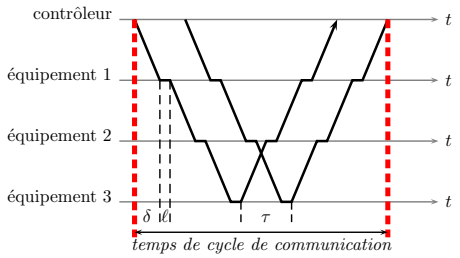

Fig. 1. Diagramme spatio-temporel d'EtherCAT

Le protocole EtherCAT supporte des topologies linéaires et en anneau. Comme la topologie linéaire (FIGURE 2) est principalement utilisée en industrie, elle sera étudiée ici.

Les datagrammes EtherCAT sont directement encapsulés dans la trame Ethernet. Une trame EtherCAT est composée d'une entête spécifiant la longueur de la trame et

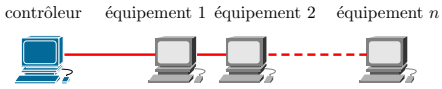

Fig. 2. Topologie linéaire d'EtherCAT

de l'ensemble des datagrammes. Un datagramme est défini pour chaque esclave et contient le type de la commande ainsi que les données associées. D'ores et déjà, il est possible de dire que le nombre d'esclaves aura un impact important sur la taille de la trame.

L'analyse des performances d'EtherCAT considère ici les hypothèses suivantes :

- topologie linéaire,
- l'étape d'initialisation est ignorée et seules les communications cycliques sont prises en considération,
- le maître envoie une seule trame par cycle,
- la charge utile x est identique pour chaque esclave.

Sous ces hypothèses, il est à présent possible d'évaluer la durée minimale du temps de cycle (FIGURE 1). Le temps de transmission est le rapport entre la taille de la trame et la capacité du lien C. La taille totale de la trame peut être divisée en deux parties :

- une partie constante égale à la somme du protocole Ethernet (26 *octets*), du trou inter-trame (équivalent à 12 *octets*) et l'entête EtherCAT (2 *octets*),
- une partie variable dépendant du nombre d'esclaves n, de la quantité de données à transporter x et de l'entête des datagrammes (12 *octets*).

Le temps de transmission est alors :

$$\tau = \frac{8\left(40 + \max\left(44, n\left(12 + x\right)\right)\right)}{C}$$

Le terme 44 dans l'équation précédente est ajouté dans le but de prendre en compte la taille minimale des données définie par le protocole Ethernet : si la longueur d'un télégramme EtherCAT est inférieure à 46 *octets*, une quantité de données de remplissage sera insérée dans la trame Ethernet. Comme un télégramme EtherCAT inclut une entête de 2 *octets*, il n'y aura donc pas de remplissage si la taille de la séquence de datagrammes est supérieure à 44 *octets*.

D'après la FIGURE 1, le temps de cycle de communication est la somme de :

- la latence pour traverser tous les esclaves (identique pour tous les esclaves)
- le temps de propagation. Le temps de propagation est supposé identique sur tous les liens.
- le temps de transmission (calculé précédemment).

L'estimation du temps de cycle peut alors être obtenue par :

$$\begin{aligned}\Gamma &= \left(2n-1\right)\ell + 2n\delta + \tau \\ &= \left(2n-1\right)\ell + 2n\delta + \frac{8\left(40 + \max\left(44, n\left(12 + x\right)\right)\right)}{C} \qquad (1)\end{aligned}$$

Notons que (1) prend en compte seulement une trame. Comme la charge utile dépend directement du nombre d'esclaves et comme la trame Ethernet ne peut excéder 1526 *octets* (et donc un champ de données de 1500 *octets*), l'équation ne peut être utilisée que si le nombre

d'équipements interconnectés au réseau est inférieur à :

$$n \leq n_{max} = \left\lfloor \frac{1500 - EH}{12 + x} \right\rfloor$$

où EH est la taille de l'entête EtherCAT (2 *octets*) et n_{max} le nombre de datagrammes de longueur x qui peuvent être inclut dans une seule trame (il est donc supposé ici que $x \leq 1486$ *octets*).

Si le nombre d'équipements sur le réseau n'est pas limité à la valeur donnée précédemment, il est nécessaire de considérer l'envoi de plusieurs trames par le contrôleur. En fait, le nombre de trames Ethernet nécessaires pour transporter les données (de charge utile x) des n équipements est donné par l'expression suivante :

$$k = \left\lceil \frac{n}{n_{max}} \right\rceil$$

Par conséquent, l'équation (1) doit maintenant intégrer les différents temps de traversée de chaque trame (le remplissage n'étant considéré si besoin que pour la dernière trame). L'expression du temps de cycle minimal devient :

$$\Gamma = (2n-1)\,\ell + 2n\delta + \frac{8}{C}\Big(40k + (k-1)\,n_{max}\,(12+x)\Big) + \frac{8}{C}\max\Big(44, (n-(k-1)\,n_{max})\,(12+x)\Big) \quad (2)$$

Comparé aux résultats de [9], le temps de cycle minimal défini en (2) prend en compte le mécanisme *on the fly* proposé par EtherCAT. L'avantage principal de ce mécanisme est qu'un équipement peut commencer à transmettre une trame avant la réception complète de celle-ci (contrairement à un mode *store and forward*), ce qui permet de réduire considérablement le délai de bout-en-bout global (FIGURE 1). De plus, comparé à [9], [13], l'équation (2) prend en compte la possibilité de remplissage définie par Ethernet et le temps pour transmettre l'information envoyée par les équipements au contrôleur (FIGURE 1). Finalement, il est important de noter que les trames Ethernet ne seront complètes (1526 *octets*) que pour une configuration particulière de la charge utile x des équipements EtherCAT.

C. Profinet IRT

Le protocole Profinet est développé par Siemens (type 10 dans le standard IEC 61158, [7], [8]). Différentes versions du protocole existent telles que RT (*Real-Time*) et IRT (*Isochronous Real-Time*). Profinet IRT est capable de supporter les communications temps-réel. Des cartes Ethernet standard ne peuvent pas être utilisées et donc Profinet requièrent un matériel spécifique sur les esclaves (ASIC de type commutateurs 2 ou 4 ports intégrés). Le protocole est basé sur la division temporelle du cycle de communication en deux modes : les modes asynchrones et isochrones sont utilisés respectivement pour des trafics non contraints et temps-réel.

Le lecteur peut consulter [10] pour une simulation de ces deux modes. Ici, il est considéré que Profinet utilise le *slipstreaming effect* (FIGURE 3) qui consiste en l'envoi par le contrôleur des trames dans un ordre particulier : de l'esclave le plus éloigné au plus proche sur la topologie linéaire (FIGURE 4). Un comportement similaire est utilisé pour le trafic généré par les esclaves au contrôleur puisqu'il est considéré des échanges simultanés et en mode *full-duplex*.

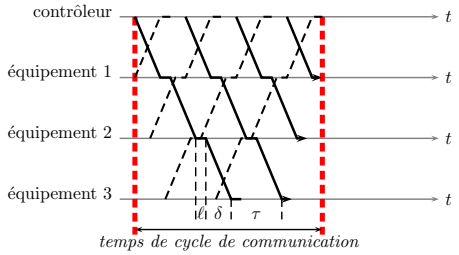

Fig. 3. Diagramme spatio-temporel de Profinet IRT avec *slipstreaming effect*

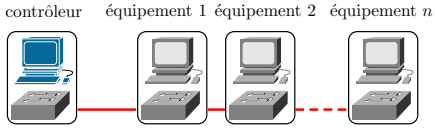

Fig. 4. Topologie linéaire de Profinet

Les équipements sont supposés synchronisés, et capables d'envoyer périodiquement leurs messages au contrôleur. De plus, le *slipstreaming effect* sera considéré seulement lorsqu'il est positif, et donc il est nécessaire que $\tau \geq \delta + \ell$ comme mentionné dans [9]. Dans ce cas, le temps de cycle, donné par [9], [13] s'exprime ainsi :

$$\Gamma = \delta + \ell + n\tau \quad (3)$$

Comparé au temps de cycle minimal pour EtherCAT (équation (2)), il est à noter que le temps de transmission est plus important dans Profinet puisqu'il est multiplié ici par le nombre d'esclave. L'équation (3) peut être développée en analysant ce temps de transmission τ. Les télégrammes Profinet sont encapsulées directement dans la trame Ethernet.

Trois champs sont ajoutés à la trame Ethernet : un identifiant (2 *octets*), les données (sous l'hypothèse suivante $x \leq 1494$ *octets*) et un statut d'information (4 *octets*). Avec le respect de la taille minimale des données Ethernet, le temps de transmission d'une trame est donné par l'expression suivante :

$$\tau = 8\frac{38 + \max(46, 6+x)}{C}$$

et l'équation (3) peut alors être complétée comme suit :

$$\Gamma = \delta + \ell + n\frac{8}{C}\Big(38 + \max(46, 6+x)\Big) \quad (4)$$

L'équation (4) sera utilisée pour le calcul du temps de cycle minimal d'un réseau Profinet donné.

D. Modbus/TCP

Modbus est un protocole de communication série proposé par Modicon en 1979. Modbus/TCP (type 15 dans le standard IEC 61158, [7], [8]) est une variante du protocole Modbus qui est adapté à l'utilisation d'Ethernet (couche physique et liaison de données [15], [11]). Modbus/TCP encapsule une trame Modbus dans une trame

TCP. Le datagramme Modbus est composé d'une en-tête applicative du protocole Modbus (MBAP), de la fonction (lecture/écriture) et des données.

Avant d'aller plus loin dans l'analyse, il est important de noter que les performances de Modbus/TCP dépendent de différents paramètres comme la topologie, les échanges applicatifs et les performances des processeurs. En effet, Modbus/TCP est une approche pragmatique dans laquelle toutes les configurations sont possibles. Bien que le protocole soit basé sur des transactions orientées connexion, Modbus/TCP peut utiliser un schéma de communication maître/esclave, producteur/consommateur ou bien client/serveur. Le mode maître/esclave est choisi dans cette étude afin d'introduire un autre principe d'accès au médium comparé à EtherCAT et Profinet. Une autre raison est que ce mode simplifie l'analyse car les équipements sont synchronisés par les événements réseau. La communication peut se décrire de la façon suivante : le maître envoie une trame par esclave, et à la réception de la trame, l'esclave envoie au maître une réponse applicative (couche 7). Le maître interroge un autre esclave dès la réception de la trame entière. Ce mécanisme de requête/réponse est montré sur la FIGURE 5.

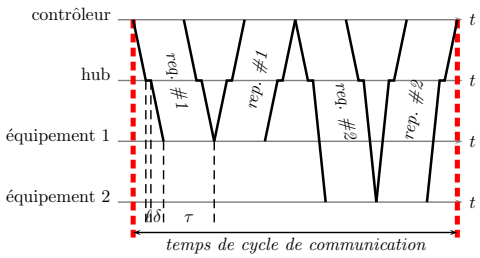

Fig. 5. Diagramme spatio-temporel de Modbus/TCP

Des échanges additionnels dûs au protocole TCP ne sont pas considérés dans la FIGURE 5. Comme Modbus/TCP repose sur le protocole TCP, son comportement doit être pris en compte. Premièrement, les sessions d'ouverture et de fermeture ne sont pas considérées, l'étude porte sur la phase périodique. Deuxièmement, TCP est responsable de l'acquittement des transmissions par segment. A la réception d'un segment, TCP peut décider d'envoyer immédiatement un acquittement, d'attendre la réception de plusieurs segments dans le but d'acquitter l'ensemble en une seule fois ou bien d'utiliser le "piggybacking" ce qui signifie ajouter un acquittement dans la prochaine transmission de données. En fait, le comportement de TCP diffère en pratique selon différents paramètres comme le système d'exploitation, les configurations TCP, ou l'activation de l'algorithme Nagle. Dans cette étude, il est considéré un temps d'envoi d'un acquittement de $0,5$ s en mode "piggybacking" (FIGURE 5). De plus, l'étude est faite dans le cas le plus favorable : les pertes ainsi que les retransmissions inhérentes ne sont pas considérées.

Modbus/TCP repose sur différentes topologies (linéaire, étoile ou boucle). Des architectures complexes basées sur des commutateurs peuvent être utilisées spécifiquement quand le réseau est partagé avec d'autres applications. Cependant, comme le schéma de communication de type maître/esclaves est considéré et que le l'utilisation de commutateurs implique des coûts additionnels, une infrastructure de type bus sera utilisée dans notre étude (FIGURE 6).

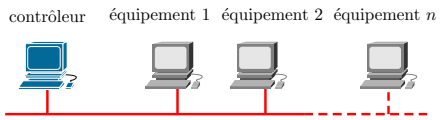

Fig. 6. Topologie linéaire de Modbus/TCP

Chaque équipement est supposé interconnecté à un concentrateur commun. Le délai d'un équipement à un concentrateur est égal à δ. Dans ces conditions, la FIGURE 5 montre que le temps de cycle est égal au nombre d'équipements n multiplié par le temps nécessaire pour interroger un équipement. Ce temps d'interrogation dépend :

- du temps de transmission de la requête et de la réponse,
- et de leur temps de propagation de la requête et de la réponse.

A partir des précédentes hypothèses, le temps de cycle minimal peut s'écrire :

$$\Gamma = n\left(\tau_{req} + \tau_{rep} + 2\left(2\delta + \ell\right)\right) \tag{5}$$

Dans le pire cas, où les acquittements TCP sont envoyés pour chaque segment, le temps de cycle minimal sera égal à $\Gamma = n\left(\tau_{req} + \tau_{rep} + 3 \times \left(2\delta + \ell\right) + 2\tau_{ack}\right)$ avec $\tau_{ack} = 672/C$ avec 672 la taille minimale en bits d'une trame Ethernet (trou inter-trame inclus). Notons que l'équation (5) diffère quand plusieurs concentrateurs sont utilisés car les temps de propagation doivent être augmentés.

Le temps de transmission des trames est composé de :

- une partie constante relative à la somme du protocole Ethernet (38 *octets* avec le trou inter-trame), l'en-tête IP (20 *octets* sans options), l'en-tête TCP (20 *octets* sans options) et l'en-tête Modbus/TCP (7 *octets*)
- une partie variable relative au type de message Modbus. En fait, le format du télégramme Modbus encapsulé dans la trame Modbus/TCP varie en fonction du type de données (code fontion) et du type de transaction (requête ou réponse). Ici, les requêtes en écriture seront considérées. Chaque message Modbus/TCP encapsule un télégramme Modbus.
- et une partie variable proportionnelle à la charge utile (comme le champ "compteur d'octet" est sur un octet, le champ de donnée est limité à $x \leq 255$ *octets*).

Comme la somme des tailles des en-têtes IP, TCP et Modbus est largement supérieure à la taille minimale des trames Ethernet, il n'y a pas besoin d'ajouter du remplissage. Le délai est directement donné par :

$$\tau_{req} + \tau_{rep} = 8\frac{91 + x}{C} + 8\frac{90}{C}$$

et l'équation (5) s'écrit :

$$\Gamma = n\left(8\frac{181 + x}{C} + 2\left(2\delta + \ell\right)\right) \tag{6}$$

E. Ethernet/IP

Ethernet/IP (IP signifie *Industrial Protocol*) est un réseau développé par Rockwell Automation en 2001 et supporté par l'ODVA (*Open DeviceNet Vendor Association*) [3], [14]. Ethernet/IP (type 2 dans le standard IEC

61158,, [7], [8]) utilise CIP (*Common Industrial Protocol*) ce qui permet l'emploi de produits sur étagère compatibles avec la pile TCP-UDP/IP. Ethernet/IP est un réseau basé sur les connexions CIP qui définissent le paquet qui sera produit sur le réseau. Deux catégories de connexions sont définies : la messagerie explicite (ou *Explicit Messaging*) et la messagerie implicite (ou *I/O messaging* ou *Implicit messaging*). Tandis que la messagerie explicite est utilisée pour des communications génériques et multiples entre deux nœuds, la messagerie d'entrées/sorties est spécifique aux applications d'entrées/sorties. Quand une application est contrainte temporellement, la messagerie d'entrées/sorties est le mode préféré car il utilise UDP plutôt que TCP. De plus, CIP utilise le modèle de communications producteur/consommateur qui requiert des échanges en diffusion (ou *broadcast*) supportés par UDP. L'analyse suivante sera réalisée dans le cas particulier de connexions de type entrées/sorties.

Comme Ethernet/IP s'appuie sur des composants sur étagère (*COTS : Components On The Shelves*), aucune topologie particulière n'est requise : des architectures basées sur des commutateurs ou des concentrateurs sont possibles. Les commutateurs sont principalement intéressants dans le sens où ils réduisent les domaines de collisions. Ils sont également utiles pour la redondance et peuvent supporter des mécanismes de réseau virtuel (ou *VLAN*) et de classification de service [12]. Des architectures commutées sont recommandées pour des messages temps-critique (ou *implicit*). Cependant, différents types d'architectures Ethernet commutées sont possibles. Dans cette étude, une topologie linéaire commutée est choisie (FIGURE 7). Nous pouvons voir celle-ci comme une extension de la configuration expérimentale considérée dans [1] où le nombre de commutateurs varie ici en fonction du nombre de ports par commutateurs.

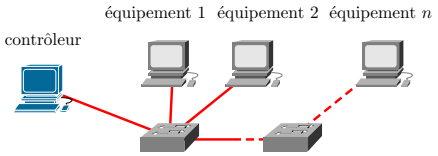

Fig. 7. Topologie linéaire commutée d'EtherNet/IP

Dans le but de garantir la comparaison avec les protocoles précédemment étudiés, il est considéré que le contrôleur envoie une trame à chaque équipement et que respectivement chaque équipement produit une information pour le contrôleur. La FIGURE 8 représente ce comportement particulier.

Initialement, EtherNet/IP ne supporte pas de synchronisation d'accès au médium comme dans une approche de type maître/esclaves utilisée dans Modbus/TCP. Cela signifie que tout équipement peut accéder au réseau en tout temps. Cependant, un mécanisme de synchronisation peut être utilisé : EtherNet/IP supporte le protocole IEEE 1588 implémenté dans le profil CIPSync. Par conséquent, il est tout à fait possible par exemple de demander aux esclaves d'envoyer leur message en même temps. Il est alors proposé ici d'utiliser comme pour Profinet le *slipstreaming effect* pour le contrôleur en plus du départ commum pour les équipements comme représenté sur la FIGURE 8. Bien entendu, ce profil correspond à un cas idéal relatif au temps de cycle minimum pour cette architecture. Premièrement, les erreurs de synchronisation mèneront à un autre profil et à un temps de cycle plus long. Deuxièmement, l'utilisation du *slipstreaming effect* requiert ici comme pour Profinet que $\tau \geq \delta + \ell$.

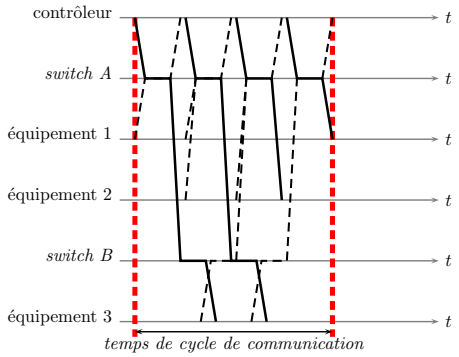

Fig. 8. Diagramme spatio-temporel d'EtherNet/IP avec des *switches* 4 ports

Considérons maintenant le profil donné par la FIGURE 8. Le temps de cycle minimum correspond à la somme :

- des latences ℓ pour traverser seulement un commutateur plus deux fois le délai de propagation δ entre le contrôleur/équipement et un commutateur,
- et le temps de transmission τ de chaque trame envoyée par le contrôleur.

Le temps de cycle minimum peut alors être écrit de la façon suivante :

$$\Gamma = 2\delta + \ell + n\tau \tag{7}$$

Notons ici que le terme donné dans l'équation 8 est seulement valide si $\tau \geq \delta + \ell$, ce qui signifie déjà que le switch ne travaille pas en mode *store and forward*. Le temps de transmission est alors calculé en considérant le format d'encapsulation des messages *I/O* d'EtherNet/IP.

18 *octets* sont ajoutés par le protocole CIP, 8 *octets* par UDP, 20 *octets* par IP et finalement 38 *octets* par Ethernet (pas de remplissage ici). Sous l'hypothèse $x \leq 1454$ *octets*, le temps de transmission est donné par :

$$\tau = 8\frac{38 + 20 + 8 + 18 + x}{C}$$

(8) peut être modifié ainsi :

$$\Gamma = 2\delta + \ell + 8n\frac{84 + x}{C} \tag{8}$$

Comme le protocole CIP permet d'encapsuler plusieurs items dans une trame, il est à noter ici que le profil donné dans la FIGURE 8 peut différer avec l'application d'un scénario particulier. Dans la section suivante, les résultats obtenus pour EtherCAT, Profinet IRT, Modbus/TCP et EtherNet/IP seront comparés relativement au scénario commun et à la bande passante.

F. Comparaisons

L'objectif de cette partie est de comparer le comportement des différentes solutions basées sur Ethernet. FIGURE 9 exprime le temps de cycle en fonction du nombre d'esclaves connectés au réseau. De plus, la comparaison est

réalisée pour deux bande-passantes (100 Mb/s et 1 Gb/s) avec une des charges utiles introduites dans [13] (16 *octets* correspondant à une valeur acceptable dans les communications industrielles). La latence des équipements ℓ comme définie dans [9], [13] est résumé dans la TABLE II sachant que pour Modbus/TCP elle est de 1 μs (*hub*) quelque soit le débit.

TABLE II
VALEURS NUMÉRIQUES

Protocol	*FastEthernet* (100 Mb/s)	*GigaEthernet* (1 Gb/s)
EtherCat	1.35 μs	0.85 μs
Profinet IRT	3 μs	0.6 μs
Modbus/TCP	1 μs	1 μs
EtherNet/IP	3 μs	0.6 μs

Il est supposé que les performances des commutateurs utilisés dans EtherNet/IP sont les mêmes que celles de Profinet IRT. Bien entendu, d'autres valeurs donneraient des autres temps de cycle (à noter également que ces résultats ne sont valides que si $\tau \geq \delta + \ell$). Le délai de propagation est de $50ns$ (pour une distance de 10 m entre les équipements).

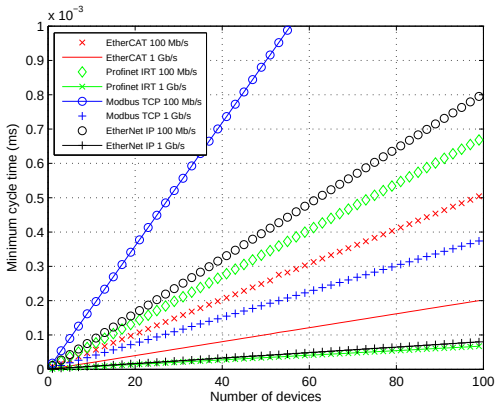

Fig. 9. Temps de cycle minimum en fonction du nombre d'équipements pour une charge utile de 16 octets par équipement

On constate que Modbus/TCP fournit de mauvais résultats : cela est dû à son mécanisme d'accès au médium. Le *polling* n'apparaît pas ici intéressant quand le temps de cycle minimum est considéré. Deuxièmement, la figure met en avant le fait que si EtherCat fournit le meilleur résultat quand FastEthernet est utilisé, il ne bénéficie pas de l'avantage de l'augmentation de la bande passante comme pour Profinet IRT ou EtherNet/IP. En effet, la bande passante a seulement un impact sur le temps de transmission τ. Comme une seule trame est envoyée dans EtherCat et comme Profinet et EtherNet/IP requièrent la transmission de n trames, le bénéfice de l'augmentation de la bande passante est plus sensible pour ces deux derniers protocoles. De plus, $2n$ latences seront considérées en EtherCat alors que les temps de cycle en Profinet et EtherNet/IP sont seulement composés d'une latence ℓ grâce au *slipstreaming effect*. C'est pourquoi Profinet et EtherNet/IP permettent des temps de cycle moins longs, et comme la différence d'un temps de propagation existe entre les temps de cycle de Profinet IRT et EtherNet/IP, Profinet fournit de meilleurs résultats à 1 Gb/s.

Les performances sont principalement les mêmes quand la charge utile augmente. La seule différence est visible pour EtherCat comme son temps de cycle minimum sera plus long que pour Profinet quand le nombre d'équipements augmentent. Cela est dû à la fragmentation pour les télégrammes EtherCat dans plusieurs trames Ethernet qui diminue les performances du protocole.

En conclusion, des solutions pleinement respectueuses de la standardisation comme Ethernet/IP peuvent fournir des résultats se trouvant parmi les meilleurs.

III. CONCLUSION

Ce papier met en avant les performances temporelles de protocoles industriels basés sur Ethernet. L'approche présentée ici permet d'obtenir une expression du temps de cycle des différents réseaux. Pour des systèmes à fortes contraintes temporelles, tous les protocoles présentés dans ce papier sont adéquats pour un débit de 1 Gb/s. A 100 Mb/s, une attention particulière est requise dans la sélection du protocole. Une étude sur l'offre commerciale (coupleur et composants applicatifs) est également à prendre en compte pour le choix des composants. Dans ce sens, la solution complètement standard semble être la solution la plus pérenne.

RÉFÉRENCES

[1] Alessandria E., Seno L. et VitturiS. Performance analysis of ethernet/ip networks. 7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems, FET, Toulouse, France, Novembre 2007.

[2] Alves M., Tovar E. et Vasques F. *Ethernet goes real-time : a survey on research and technological developments*. Groupe de recherche IPP-HURRAY, Polytechnic Institute of Porto (ISEP-IPP), Janvier 2000.

[3] Brooks P. Ethernet/ip-industrial protocol. 8th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Antibes - Juan lès Pins, France, Octobre 2001.

[4] Decotignie J.-D. A perspective on ethernet-tcp/ip as a fieldbus. IFAC International Conference on Fieldbus Systems and their Applications (FET'2001), Nancy, France, Novembre 2001.

[5] ——, Ethernet-based real-time and industrial communications. *Proceedings of the IEEE*, vol. 93, no. 9, pp. 1102-1117, June 2005.

[6] Felser M. Real-time ethernet - industry prospective. *Proceedings of the IEEE*, vol. 93, n°. 6, pp. 1118-1128, Juin 2005.

[7] IEC. *Digital data communications for measurement and control – fieldbus for use in industrial control systems : Part 3 : Data link service definition*. IEC Standard 61158, 2006.

[8] ——. *Digital data communications for measurement and control – fieldbus for use in industrial control systems : Part 4 : Data link protocol specification* IEC Standard 61158, 2006.

[9] Jasperneite J., Schumacher M. et Weber K. Limits of increasing the performance of industrial ethernet protocols. International Conference on Emerging Techonologies and Factory Automation, ETFA, Patras, Greece, Septembre 2007

[10] Jasperneite J. et Elsayed E. Investigations on a distributed time-triggered ethernet realtime protocol used by profinet. 3rd International Workshop on Real-Time Networks, RTN, Catane, Italie, Juin 2004.

[11] Modbus-IDA. *Modbus application protocol specification v1.1b.*. [Online]. Available : http ://www.modbus-ida.org/

[12] Modlovansky A. Utilization of modern switching technology in ethernet/ip networks. 1st International Workshop on Real-Time LANs in the Internet Age, RTLIA, Vienna, Austria, Juin 1998.

[13] Prytz G. A performance analysis of ethercat and profinet irt. International Conference on Emerging Techonologies and Factory Automation, ETFA, Hamburg, Allemagne, 2008.

[14] Schiffer V. *The common industrial protocol (CIP^{TM}) and the family of CIP networks*. Open DeviceNet Vendor Association - Inc. (ODVA), 2006. [Online]. Available : http ://www.odva.org

[15] The Modbus Organization. *Modbus*. [Online]. Available : http ://www.modbus.org/