

dr hab. inż. Andrzej Kwiecień  
Instytut Informatyki, Politechnika Śląska  
mgr inż. Marcin Sidzina  
Katedra Podstaw Budowy Maszyn, Akademia Techniczno-Humanistyczna

## **WPLYW CZASU TRWANIA CYKLU PRACY STEROWNIKA SWOBODNIE PROGRAMOWALNEGO NA CZAS WYMIANY DANYCH W SIECIACH PRZEMYSŁOWYCH, ZE SZCZEGÓLNYM UWZGLĘDNIENIEM ICH INTEGRACJI.**

*W pracy przedstawiono zależności pomiędzy czasem trwania cyklu pracy sterownika swobodnie programowalnego a czasem wymiany informacji w sprzęgach sieci przemysłowych, oraz w pojedynczych sieciach typu MASTER-SLAVE. Prezentowane wyniki oparto na szeregu badaniach przeprowadzanych na sterownikach programowalnych wyposażonych w niezbędne moduły komunikacyjne. Przedmiotem badań był jednokierunkowy przepływ danych pomiędzy sieciami TOKEN-RING i MASTER-SLAVE w przypadku, gdy urządzeniem sprzęgającym jest sterownik programowalny wyposażony w koprocesory komunikacyjne tych sieci. Przedstawiono wyniki pomiarów czasu wymiany danych dla podanego sprzęgu oraz sieci MASTER-SLAVE.*

### **THE PROGRAMMABLE LOGIC CONTROLLER (PLC) WORK CYCLE TIME INFLUENCE ON EXCHANGE DATA TIME IN INDUSTRIAL NETWORKS**

*The article describes dependences between PLC work cycle time and exchange data time in single or integrated networks. The results of many experiments with using PLCs in network coprocessors are presented too. The paper describes also, one-way communication from TOKEN RING to MASTER-SLAVE network in case, when PLC like interconnection device or integration device are used.*

#### **1. WSTĘP**

Większość najnowszych przemysłowych rozwiązań systemów sterowania opartych jest częściowo lub w całości na sieciach komputerowych niskiego poziomu, tworząc rozproszone systemy sterowania. Mówiąc o sieciach niskiego poziomu, należy mieć na uwadze te, które w sposób bezpośredni, za pomocą sterowników swobodnie programowalnych lub zdalnych obiektowych wejść/wyjść, są sprzęgnięte ze sterowanym bądź monitorowanym czy regulowanym obiektem technologicznym. Niemniej jednak bardzo często zdarza się, iż głównym modelem sterowania jest system

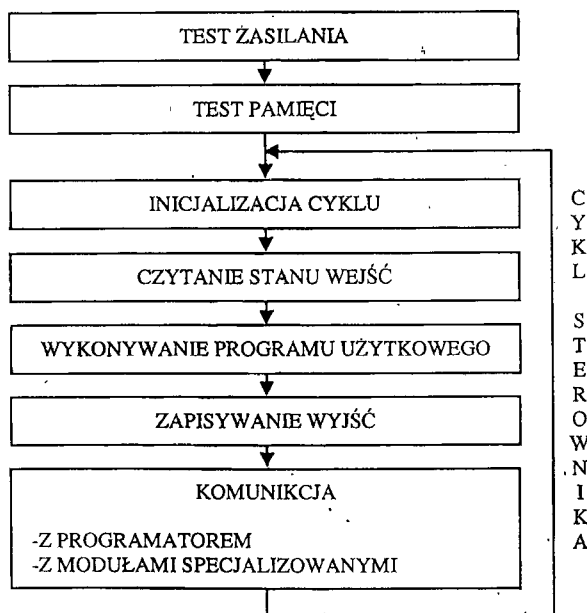
scentralizowany. Pojawianie się coraz to nowych urządzeń z zaimplementowanymi protokołami transmisji danych sprawia, że te stare systemy coraz częściej zostają zamieniane na systemy rozproszone. Duża liczba oferowanych produktów oraz protokołów transmisji przez nie używanych powoduje powstanie problemów związanych z komunikacją w danej strukturze sieciowej, lub komunikacją pomiędzy dwiema, lub więcej sieciami używającymi różne standardy komunikacji. Skłania to do zastanowienia nie tylko nad tym czy dana sieć jest lepsza od innej, ale również, czy dana implementacja sieci sprostą oczekiwaniom użytkownika [1, 5]. Innymi słowy mamy do czynienia z procesem integracji struktur sieciowych w ramach tego samego obiektu lub szeregu podobiektów.

Istnieje kilka metod integracji przemysłowych sieci komputerowych. Do ważniejszych zaliczyć można: zastosowanie specjalizowanych sterowników programowalnych z modułami komunikacyjnymi lub komputerów PC wyposażonych w specjalizowane oprogramowanie wymieniające dane pomiędzy sterownikami [5]. Badanie tych sprzęgów przy dążeniu do pełnej integracji sterowanych procesów ujawnia wady i zalety tych rozwiązań [3].

Istotnym problemem w łączeniu sieci komputerowych jest dopasowywanie wymiany danych między wszystkimi komponentami aplikacji w taki sposób, aby cykl trwania automatu (sterownika sprzęgającego) był jak najmniejszy. W dalszej części pracy cykl trwania automatu oraz cykl pracy sterownika programowalnego będą używane zamiennie.

## 2. CYKL PRACY STEROWNIKA PROGRAMOWALNEGO

Sterownik swobodnie programowalny jest powszechnie stosowanym urządzeniem automatyki, do sterowania i regulacji różnorodnych procesów przemysłowych.



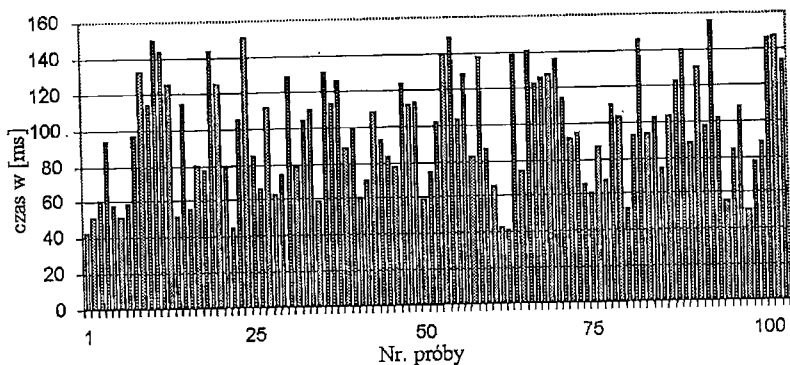
Rys. 1. Cykl pracy sterownika programowalnego (automatu) [1,4]

Ze względu na pełnione funkcje i uwarunkowania czasowe w sterowanych procesach musi być zagwarantowany pewien nieprzekraczalny maksymalny czas reakcji sterownika na zaistniałe zdarzenia w sterowanym procesie. Po uruchomieniu sterownika następuje szereg operacji koniecznych do wykonania programu logicznego. Zbiór tych operacji nazywany jest cyklem pracy sterownika programowalnego [1, 4]. (Rys.1).

Podczas trwania cyklu sterownik, oprócz programu użytkownika wykonuje pewne zadania systemowe. Większość zadań systemowych takich jak inicjalizacja cyklu, czytanie stanu wejść, zapisywanie wyjść jest obligatoryjna, a czas ich wykonania jest zależny od liczby punktów wejść/wyjść. Cykl pracy sterownika może się również zmienić w zależności od tego, czy w danej chwili jest podłączony programator lub stacja nadrzędna monitorująca sterowany proces i czy zgłaszane jest żądanie komunikacji. Czas na komunikację z programatorem jest ustalony systemowo i powinien być z góry uwzględniony na etapie projektowania systemu. To samo dotyczy komunikacji z modułami specjalizowanymi. Pod pojęciem modułów specjalizowanych rozumie się koprocesory sieciowe, moduły programowalne, szybkie liczniki itp. Należy zwrócić uwagę, że w niektórych sterownikach w trakcie jednego cyklu, jednostka centralna może odpytać tylko jeden moduł specjalizowany. Warto zauważyć, że czas na komunikację z koprocesorami sieciowymi (komunikacja systemowa) jest również systemowo ograniczony, co może spowodować, że wymiana danych pomiędzy jednostką centralną a koprocesorem sieci nie będzie obsługiwana w trakcie jednego cyklu pracy sterownika, ale w dwóch, czy nawet więcej cyklach. Dlatego przed rozpoczęciem projektowania systemu warto sprawdzić, jaki jest minimalny i maksymalny czas przesłania najmniejszej jednostki danych w sieci przy różnej długości cykli automatu.

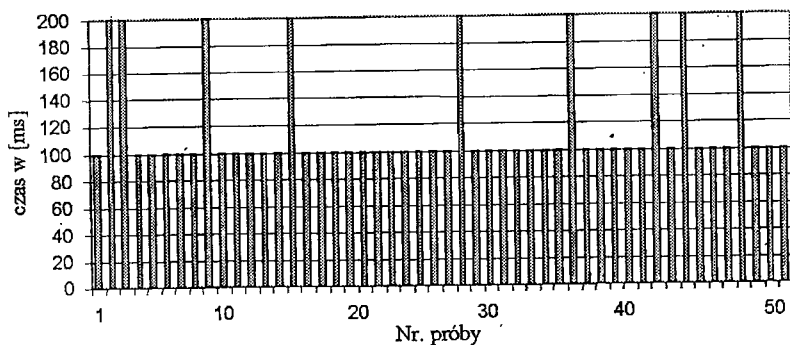
### **3. CZAS TRWANIA CYKLU PRACY STEROWNIKA SWOBODNIE PROGRAMOWALNEGO A WYMIANA DANYCH W SIECI MASTER-SLAVE**

W zależności od budowy sterownika moduł koprocesora sieciowego może być zintegrowany z jednostką centralną lub jest dołączany do niego jako moduł dodatkowy. Podczas badań eksperymentalnych zmierzono czas potrzebny na przesłanie informacji ze stacji MASTER do stacji SLAVE. Przetestowano dwa różne sterowniki będące stacjami MASTER dla sieci MASTER-SLAVE. Pierwszy to sterownik przemysłowy o budowie modułowej z dodatkowym modułem stacji MASTER, gdzie komunikacja z koprocesorem wywoływana jest programowo raz, a dalsze wymiany wykonywane są automatycznie. W drugiej konfiguracji zastosowano sterownik z wbudowanym portem stacji MASTER, w tym przypadku każda wymiana jest wyzwalana przez program użytkownika. W obu przypadkach w sterowniku zastosowano minimalny program użytkownika potrzebny do uruchomienia wymian sieciowych.



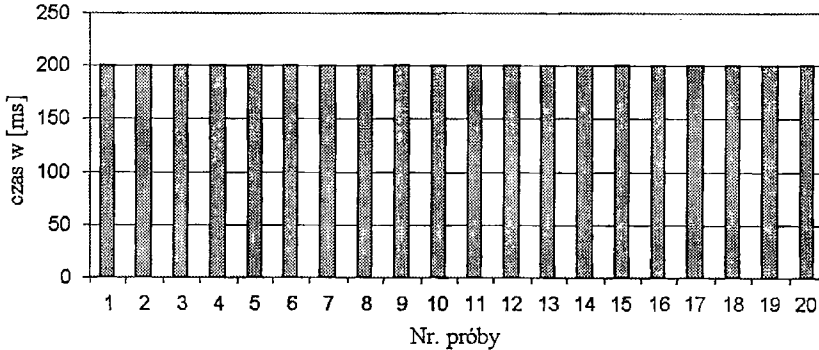
Rys. 2. Wykres czasu przepływu informacji ze stacji SLAVE do stacji MASTER przy cyklu sterownika < 10 ms

Na Rys. 2 przedstawiono wyniki pomiarów czasu przesłania 1 rejestru danych ze stacji SLAVE do stacji MASTER dla pierwszego przypadku sterownika. W eksperymencie stosowano standardowy cykl pracy sterownika (nie określono czasu jego trwania) Ze względu na prostotę programu sterującego, cykl ten zawsze mieścił się w czasie mniejszym niż 10 ms. Na wykresie (Rys. 2) widać duże wahania czasu przepływu informacji. Spowodowane jest to wysyłaniem impulsu w losowo dobranej chwili. Moment wysłania informacji określany jest względem czasu trwania cyklu sieci. W przypadku, kiedy przełączenie następuje na krótko przed „odpytywaniem” jednostki SLAVE, czas przepływu informacji jest krótki. W przypadku, gdy wysłanie informacji następuje zaraz po „odpytaniu” stacji SLAVE, czas przesłania informacji jest wydłużony prawie o cały czas trwania cyklu sieci. Na podstawie tego wykresu można wnioskować, że dokonując odpowiedniej liczby pomiarów można określić doświadczalnie maksymalny i minimalny czas trwania cyklu wymian. Otrzymane wyniki (Rys. 2) skłaniają do zastanowienia się nad tym, czy czas cyklu sieci jest tak długi, że nie można obsłużyć w krótszym czasie wymian sieci, czy też jednostka centralna sterownika swobodnie programowalnego potrzebuje kilku cykli automatu, aby przesłać całą informację. W celu wyjaśnienia tego zagadnienia przeprowadzono pomiary dla wymiany tej samej liczby danych zwiększając jednocześnie czas trwania cyklu sterownika do 100 ms (Rys. 3) i 200 ms (Rys. 4).



Rys. 3. Wykres czasu przepływu informacji ze stacji SLAVE do stacji MASTER przy cyklu sterownika wynoszącym 100 ms.

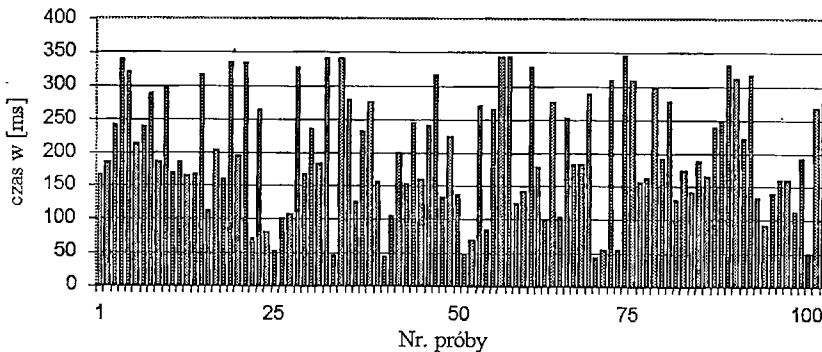
Zwiększenie czasu trwania cyklu pracy sterownika programowalnego do 100 ms spowodowało, że wymiana informacji w sieci odbywała się w dwóch odstępach czasowych 100 ms lub 200 ms. Oznacza to, że w poprzednim przypadku informacja ze stacji SLAVE była uaktualniana dla stacji MASTER w czasie nie dłuższym niż 10 ms licząc od momentu jej dotarcia do koprocesora sieciowego. Wydłużenie cyklu sterownika do 100 ms spowodowało, że wszystkie wymiany, które dotarły do koprocesora przed upłynięciem czasu cyklu, były uaktualniane w stacji MASTER po 100 ms, zaś te, które dotarły do stacji MASTER w przedziale czasowym od 101 ms do 200 ms zostały uaktualnione w ciągu 200 ms. Zwiększenie czasu trwania cyklu sterownika, do 200 ms spowodowało natomiast, że wszystkie informacje były dostarczane do stacji MASTER w ciągu 200 ms (Rys. 4).



Rys. 4. Wykres czasu przepływu informacji ze stacji SLAVE do stacji MASTER przy cyklu sterownika wynoszącym 200 ms.

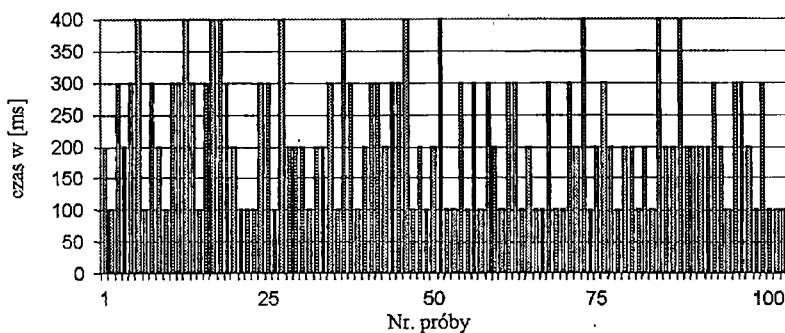
W kolejnym etapie badań, dla tej samej konfiguracji sieci, postanowiono zwiększyć liczbę wymian do trzech. Każda z trzech wymian otrzymała zadanie „odpytywania” po jednym rejestrze ze sterownika SLAVE.

Zwiększenie liczby wymian do trzech, znacznie zwiększyło czas trwania cyklu sieci (Rys. 5)



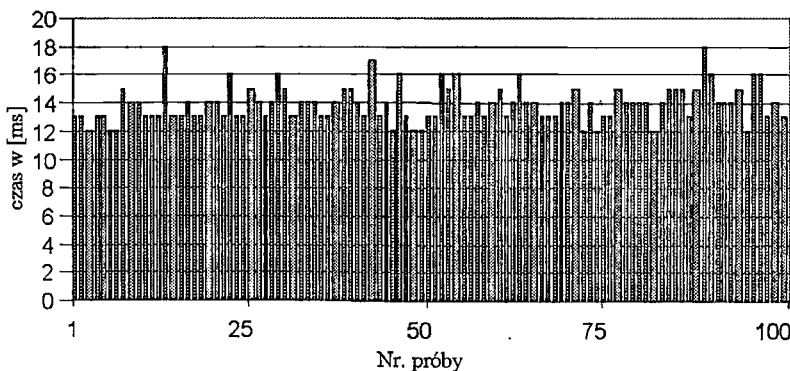
Rys. 5. Wykres pomiaru czasu dla przepływu 3 wymian informacji ze stacji SLAVE do stacji MASTER przy cyklu sterownika wynoszącym < 10 ms.

Jak wynika z Rys. 5. nastąpił znaczny wzrost czasu wymiany informacji, gdy stacja MASTER wykonuje trzy wymiany. Dotychczas, gdy stacja MASTER wykonywała jedną wymianę, czas przesłania informacji mieścił się między 40 ms a 160 ms. Zwiększenie liczby wymian do trzech, spowodowało wzrost czasu realizacji wymian do 350 ms. Przeprowadzono także badania dla 2 wymian i wówczas maksymalny czas przepływu informacji wynosił 250 ms. Oznacza to, że koprocesor potrzebuje ok. 100 ms na wykonanie 1 wymiany ze stacją SLAVE. Przyjmijmy, że cykl automatu wynosi 100 ms, a liczba wymian w stacji MASTER wynosi trzy (Rys 5). Pomiar wykazały, że sygnał może dotrzeć do celu w przedziale czasowym od 40 ms do 350 ms. Jeżeli cykl pracy sterownika wynosi 100 ms., to informacja nie zostanie uaktualniona w stacji MASTER wcześniej niż po upływie czasu trwania jednego cyklu sterownika, czyli po 100 ms. Jeżeli zmiana sygnału zostanie odnotowana w przedziale czasu między 101 ms a 200 ms uaktualnienie nastąpi po upływie czasu trwania drugiego cyklu sterownika (200 ms). Jeżeli sygnał dotrze do koprocesora w przedziale czasu 201 ms do 300 ms to uaktualnienie nastąpi po 300 ms, a gdy informacja dotrze w czasie dłuższym niż granica 300 ms to uaktualnienie nastąpi po upływie czasu trwania czwartego cyklu sterownika (400 ms). Powyższe zależności ilustruje Rys 6.



Rys. 6. Wykres czasu przepływu 3 wymian informacji ze stacji SLAVE do stacji MASTER przy cyklu sterownika wynoszącym 100 ms.

Dla drugiego sterownika przeprowadzono identyczne badania. Zaskakujący w tym przypadku był bardzo krótki czas przesłania informacji ze stacji SLAVE do stacji MASTER, wynoszący, w pojedynczych wymianach wywoływanych „wyzwalaczem”, od 12 ms do 18 ms przy czasie trwania cyklu automatu poniżej 5 ms (Rys.7).



Rys. 7. Pojedyncze wymiany, wywołane „wyzwalaczem” dla drugiego sterownika

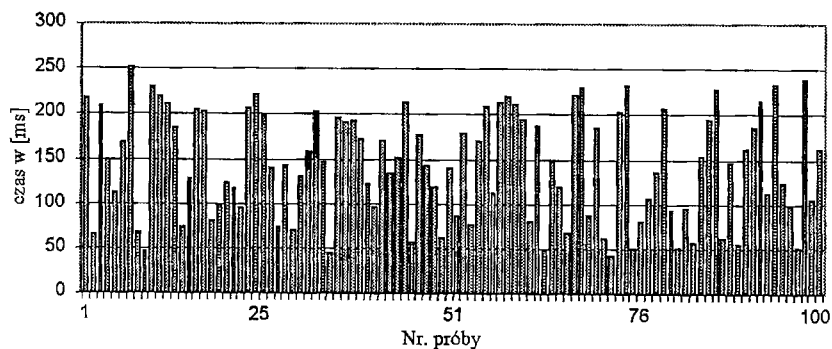
Jak wcześniej zostało wspomniane, każda wymiana w drugim sterowniku musi być wywoływana przez program. Aby zapewnić cykliczne wysyłanie informacji zastosowano układ czasowy (ang. "timer") wyzwalający wymianę danych w określonych odstępach czasu. Wyniki pomiarów dla kilku wariantów czasowych układu czasowego zostały umieszczone w Tabeli 1.

Tabela 1

MASTER	czas mierzony od ostatniego uaktualnienia zmiennej MASTER
timer 20ms	min 14ms, max 37ms,
timer 30 ms	min 21ms, max 35ms
timer 40 ms	min 32ms, max 43ms,
timer 100 ms	min 91ms, max 103ms,
timer 200 ms	min 190ms, max 204ms,

#### 4. WPŁYW CZASU TRWANIA CYKLU AUTOMATU BĘDĄCEGO URZĄDZENIEM SPRZĘGAJĄCYM NA CZAS WYMIANY DANYCH POMIĘDZY SIECIAMI

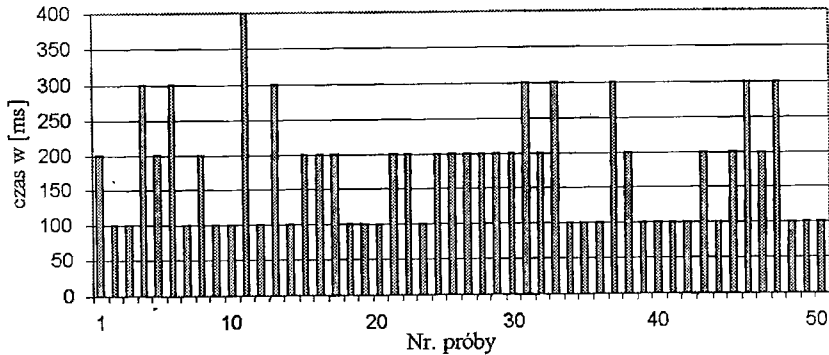
Po podłączeniu koprocatora sieci TOKEN-RING do pierwszego sterownika z koprocotorem sieci MASTER-SLAVE, uzyskano sterownik sprzęgający dwa typy sieci.



Rys. 8. Czas przepływu informacji z sieci MASTER-SLAVE do sieci TOKEN-RING dla 2 wymian w sieci M-S przy cyklu sterownika sprzęgającego  $< 10\text{ ms}$

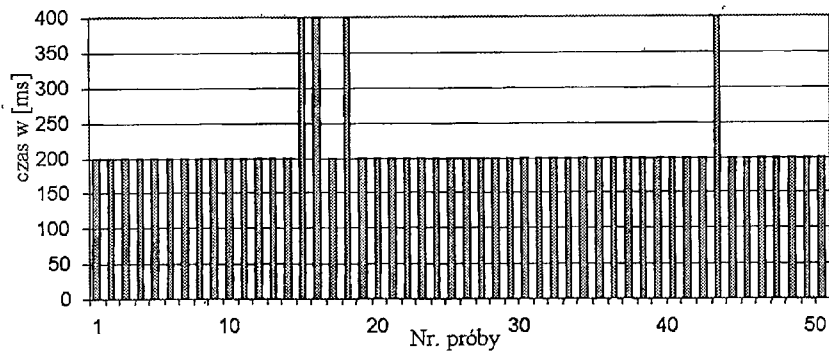
Wykonane pomiary dla 2 wymian z sieci MASTER-SLAVE do sieci TOKEN-RING wskazują, że czas trwania cyklu automatu sprzęgającego jest istotnym czynnikiem wpływającym na czas wymian. Największe straty czasowe generuje w tym przypadku wcześniej omawiany cykl sieci MASTER-SLAVE oraz cykl pracy sterownika programowalnego stacji MASTER (Rys. 8). W analizowanym przypadku wystąpił dodatkowo nowy czynnik, jakim jest drugi koprocetor sieciowy w kasecie sterownika. Jak wcześniej wspomniano, niektóre sterowniki w trakcie trwania cyklu mogą „odpytywać” tylko jeden koprocetor sieciowy, przynosząc moment „odpytania”

drugiego koprocatora, do następnego cyklu. Mechanizm ten powoduje znaczne wydłużenie czasu potrzebnego na wymianę danych między sieciami (Rys.9). Przy założeniu najmniej korzystnych warunków wymiany danych, takich że cykl pracy automatu wynosi 100 ms a maksymalny czas wymian wynosi 250 ms, uaktualnienie informacji powinno nastąpić po 3 cyklu pracy automatu. Jeśli jako pierwszy w kolejce do „odpytania” jest inny koprocator, czas potrzebny na uaktualnienie danych zwiększy się o następne 100 ms czyli tyle ile wynosi czas trwania cyklu sterownika.



Rys. 9. Czas przepływu informacji z sieci MASTER-SLAVE do sieci TOKEN-RING dla 2 wymian w sieci M-S przy cyklu sterownika sprzęgającego 100 ms

Zwiększenie czasu trwania cyklu automatu do 200 ms powoduje częste występowanie przypadku wymagającego maksymalnego czasu (400 ms) potrzebnego do uaktualnienia informacji przesyłanej z drugiej sieci (rys.10).



Rys. 10. Czas przepływu informacji z sieci MASTER-SLAVE do sieci TOKEN-RING dla 2 wymian w sieci M-S przy cyklu sterownika sprzęgającego 200 ms

Wynik ten jest analogiczny z uzyskanym dla cyklu sterownika MASTER w sieci MASTER-SLAVE.



## 5. PODSUMOWANIE

Skrócenie czasu wymiany danych w systemach rozproszonych najniższego poziomu jest jednym z najistotniejszych czynników decydujących o możliwości rozszerzenia zastosowań tych systemów. W pracy wykazano, iż czas wymiany informacji w sieci lub pomiędzy różnymi sieciami zależy głównie od czasu trwania cyklu automatu będącego newralgicznym punktem systemu. W przedstawionych przypadkach automatami newralgicznymi były: sterownik MASTER w sieci MASTER-SLAVE oraz sterownik sprzęgający sieci MASTER-SLAVE i TOKEN-RING. Zwiększenie częstotliwości odświeżania danych może być wykonane nie tylko poprzez wymianę sterownika na szybszy, ale również poprzez odpowiednie ustawienie scenariusza wymian i odciążenie tych stacji z dodatkowych operacji nie będących koniecznymi i nie wpływającymi na obsługę komunikacji.

W pracy wykazano, iż różne implementacje tych samych sieci mogą się znacznie różnić czasem wymiany informacji. Wymusza to na projektancie systemu wcześniejsze sprawdzenie konkretnych implementacji sieci, aby mógł dopasować zadania dla poszczególnych automatów.

Biorąc pod uwagę, jak duże znaczenie podczas przeprowadzonych pomiarów miał czas cyklu sterownika swobodnie programowalnego, należy zwrócić uwagę nie tylko w stronę implementacji sieci, ale również na metody zwiększające częstotliwość dostępu do procesora sieciowego.

## 6. LITERATURA

- [1] Kwiecień A., Analiza przepływu informacji w komputerowych sieciach przemysłowych. Wydawnictwo Pracowni komputerowej Jacka Skalmierskiego, Gliwice 1999.
- [2] Grandek K., Kaliczyńska M., Rajek R., Rozproszony system automatyki na bazie sterowników PLC, PAR 3/2000.
- [3] Kwiecień A., Sidzina M., Integracja przemysłowych sieci komputerowych typu MASTER-SLAVE i TOKEN-RING, X Konferencja Systemy Czasu Rzeczywistego, Ustroń wrzesień 2003.
- [4] Maczyński A. Sterowniki Programowalne PLC, Budowa systemu i podstawy programowania, ASTOR, Kraków 2002.
- [5] Tutaj R., Sterownik PLC w środowisku wielosieciowym, PAR 4/2000.
- [6] Mrówka Z., Sieci przemysłowe, Przegląd rozwiązań, zakres zastosowań, zestawienie czasów transmisji danych, Seminarium Wiśła 1999.