



POLITECHNIKA ŚLĄSKA W GLIWICACH

PRZEMYSŁOWE SYSTEMY ROZPROSZONE

**Protokoły dostępu do łącza stosowane w
sieciach przemysłowych. Sieci o dostępie
typu Master-Slave. Zachowanie reguł
determinizmu czasowego.**

AUTORZY:

Bartłomiej Buchała
Mateusz Forczmański

Informatyka, semestr V
Rok akademicki 2014/2015
Grupa 3

12 lutego 2015

1 Wstęp

Wraz z rozwojem informatyki, znalazła ona szerokie zastosowanie w różnych gałęziach przemysłu. Zastosowanie systemu informatycznego niesie ze sobą wiele korzyści, m. in:

- Automatyzację wykonywanych czynności
- Przyspieszenie obliczeń
- Autokorekcję w przypadku, kiedy dojdzie do częściowej awarii systemu

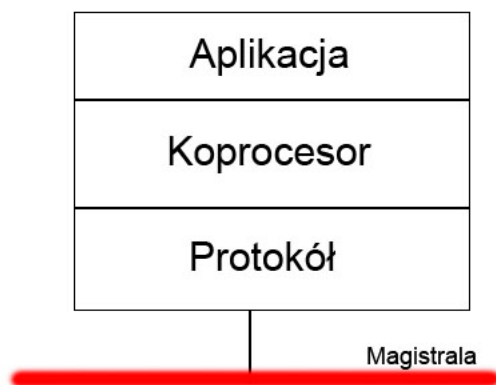
Koszt (zaprojektowanie, zaprogramowanie, instalacja takiego systemu) jest stosunkowo niewielki. Problem pojawił się przy dalszej rozbudowie branży przemysłowej. Dla części informatycznej oznaczało to przede wszystkim rosnącą złożoność obliczeniową zastosowanych algorytmów czy większą liczbę urządzeń do obsługi. Dodatkowo, pewnym utrudnieniem jest mniejsza w ostatnich latach dynamika rozwoju mocy obliczeniowej pojedynczych komputerów, co jest związane ze zbliżaniem się do pułapu miniaturyzacji elementów. Remedium na ten problem okazało się zastosowanie **systemów rozproszonych czasu rzeczywistego**.

Dla systemu czasu rzeczywistego, oprócz warunków logicznych, ważne jest także spełnienie warunków czasowych. Oznacza to, że oprócz poprawnych rezultatów musi on również zapewnić ich wykonanie w odpowiednim czasie. Można wyróżnić systemy czasu rzeczywistego, które tolerują przekroczenie ograniczeń czasowych (interwału czasowego), i takie, dla których czas operacji jest wartością krytyczną.

Podstawową cechą systemów rozproszonych jest fakt, że jest to zbiór niezależnych od siebie komputerów (lub rzadziej procesorów, jeżeli mówimy o komputerach wieloprocesorowych) połączonych siecią komputerową. Posiadają one wspólne oprogramowanie, które pozwala im na współdziałanie (podział zasobów, współbieżne wykonywanie obliczeń) w określonym przez twórcę celu. Komputery będące elementami większego systemu rozproszonego najczęściej nazywa się **węzłami**. Aby zapewnić poprawną pracę, wymagana jest odpowiednia komunikacja między węzłami.

2 Struktura węzła w sieci przemysłowej

Rolę węzła systemu rozproszonego pełni najczęściej sterownik swobodnie programowalny (PLC) lub grupa takich sterowników. PLC jest uniwersalnym urządzeniem mikroprocesorowym, którego zadaniem jest sterowanie maszyną lub urządzeniem technicznym. Realizuje on program znajdujący się w jego pamięci operacyjnej. Ponieważ węzeł jest elementem większej sieci, samo PLC nie wystarcza. Zazwyczaj pojedynczy moduł ma budowę trójwarstwową:



- Przez **aplikację** rozumiemy znajdujący się w sterowniku PLC system operacyjny czasu rzeczywistego oraz program cyklicznie realizujący określone przez programistę założenia. Ten poziom odpowiada za faktyczną pracę urządzenia i jest częściowo niezależny od innych węzłów systemu.
- **Koprocesor** jest elementem pracującym niezależnie od głównego procesora urządzenia. Jest on odpowiedzialny za komunikację pomiędzy macierzystym układem, a siecią. Posiada dwa zastosowania:
 - *Nadawanie* – otrzymując informacje od procesora głównego, kopiuje ona odpowiednie fragmenty pamięci do buforów nadawczych, następnie transformuje je do odpowiedniej formy (ramki transmisyjnej), po czym wysyła je przy użyciu nadajnika (portu UART).
 - *Odbieranie* – przechwytuje informacje z sieci skierowane do tego węzła i dekoduje je do formy mogącej być odczytaną przez sterownik. Po tym następuje skopiowanie treści do bufora odbiorczego i przepisanie do pamięci urządzenia.
- **Protokół transmisji** jest ściśle związany z koprocesorem sieci. Jest zbiorem zasad, na jakich odbywa się połączenie pomiędzy dwoma lub więcej węzłami w sieci. Dzięki temu możliwa jest komunikacja sterowników działających na różnym oprogramowaniu.

W dalszej części wypracowania skupimy się głównie na protokole transmisji.

3 Determinizm czasowy w sieciach przemysłowych

Poprzez **determinizm** działania obiektu (w naszym przypadku węzła sieci) rozumiemy, że dla każdej kombinacji stanu poprzedniego oraz zdarzenia pojawiającego się z zewnątrz, węzeł ten posiada zdefiniowany, jednoznaczny stan pracy.

Determinizm czasowy definiujemy jako zgodne z działaniem deterministycznym reakcje obiektu w określonym i skończonym czasie.

Można wyróżnić dwa różne rodzaje determinizmu czasowego:

- Określony ściśle (ostry), gdzie po upływie określonego czasu od zdarzenia inicjującego T_G następuje określona reakcja.
- Ograniczony granicznie (nieostry) - tutaj reakcja następuje nie później niż do określonego czasu T_G .

Zależności te można wyrazić poniższymi wzorami:

- Determinizm czasowy ściśle określony:

$$T_{OS} = T_G \quad (1)$$

- Determinizm czasowy granicznie określony

$$T_{OS} \leq T_G \quad (2)$$

- Dodatkowo:

$$T_{OS} = T_R + T_O \quad (3)$$

Gdzie

- T_{OS} - czas obsługi sieciowej zdarzenia
- T_G - czas graniczny obsługi
- T_R - czas reakcji na zdarzenie
- T_O - czas właściwej obsługi zdarzenia

Determinizm czasowy nie wymaga stuprocentowej niezawodności podczas przekazywania danych. Za poprawną pracę sieci przyjmuje się wtedy sytuację, kiedy nie dochodzi do wystąpienia stanu awaryjnego (zarówno z logicznego, jak i fizycznego punktu widzenia). Najistotniejsze jest, aby czas dostępu do pamięci został obciążony krytycznym czasem granicznym.

4 Rodzaje protokołów stosowane w sieciach przemysłowych

Ponieważ rozproszona sieć przemysłowa czasu rzeczywistego powinna się charakteryzować skończonym czasem przekazywania danych, będą nas interesować jedynie protokoły o zdefiniowanym (deterministycznym) w czasie dostępie.

Współcześnie można wyróżnić tylko 3 rodzaje protokołów, które spełniają powyższy warunek:

- Token
- Master - Slave
- Producent - Dystrybutor - Konsument (PDK)

Pierwsze dwa protokoły posiadają wiele implementacji i zastosowań, z kolei trzeci z nich jest dość świeży. Wszystkie inne rodzaje protokołów spełniające warunek deterministyczny są pochodnymi lub hybrydami powyższych modeli.

Protokół z użyciem Tokena polega na przesyłaniu informacji między równorzędnymi stacjami, posiadającymi dostęp do wspólnej magistrali. Aby zapewnić brak kolizji (sytuacji, w których więcej niż jedna stacja zapisuje dane), stosuje się tzw. żeton (z ang. *Token*). Jest to specjalny rodzaj informacji krążący po sieci od jednej stacji do drugiej. Odczytanie Tokenu przez węzeł powoduje jego uprzywilejowanie, co w praktyce oznacza prawo do transmisji ramek z danymi. Jednak aby zachować prawo determinizmu czasowego, prawo do zapisu trwa określony czas, a po jego upływie węzeł musi bezwzględnie przekazać żeton następnemu użytkownikowi. Token wykorzystywany jest między innymi w protokołach typu Token-Ring (pierścieniowa budowa sieci) i Token-Bus (pojedyncza magistrala, w praktyce konfiguracja gwiazdy).

Model PDK bazuje na podziale stacji na 3 rodzaje abonentów: producentów, dystrybutorów oraz konsumentów. "Producent" pełni tu rolę nadawcy: "produkuje" dane, które zostaną użyte przez inne stacje. "Dystrybutor" jest pośrednikiem - jego zadaniem jest transfer danych od producenta (lub wielu producentów) do konsumentów. Dystrybutor przechowuje scenariusz wymian i na jego podstawie określa, kiedy dana informacja ma trafić do sieci. W trakcie podejmowania decyzji, gdzie trafić mają dane pod uwagę brane są średnia konsumpcja oraz aktualne obciążenie konsumentów. "Konsument" to stacja będąca punktem końcowym dla danych, gdzie są one odczytywane i odpowiednio używane. Protokół ten jest używany np. w sieci FIP.

5 Sieci o dostępie typu Master-Slave

Protokół Master-Slave jest prostą metodą dostępu ze względu na narzucone w niej ograniczenia i sztywne reguły transmisji:

- Wymiana informacji jest dopuszczalna jedynie pomiędzy wybraną stacją nadrzędną, zwaną Master, a jedną ze stacji podrzędnych, zwanymi Slave.
- Tylko stacja Master może inicjować transakcje.
- Dwa rodzaje transmisji:
 - **Zapytanie - Odpowiedź:** polega na wymianie informacji pomiędzy Masterem, a wybraną, konkretną stacją Slave. W tym procesie uczestniczy żądanie (zdefiniowany rozkaz) oraz odpowiedź od Slave.

- **Rozgłoszenie:** żądanie jest wysyłane do wszystkich jednostek Slave w sieci, a Master nie oczekuje od nich odpowiedzi.
- Ograniczona liczba rozkazów jaką może realizować Master, czyli ściśle określony zbiór dopuszczalnych wymian.
- Wymiany informacji odbywają się cyklicznie i nie wymagają ani wstrzymania sieci, ani jej ponownego uruchomienia.

Powyższe własności sieci Master-Slave niosą ze sobą wiele zalet:

- Niskie wymagania z punktu widzenia oprogramowania.
- Łatwa konfiguracja sieci.
- Konieczność oprogramowania sprowadza się wyłącznie do koprocatora stacji Master. Stacje Slave mogą wykonywać polecenie sprzętowo, bez konieczności programowania.
- Szybkie uruchomienie systemu komunikacyjnego.

5.1 Scenariusz wymian

Scenariusz wymian to szczegółowy opis możliwych wymian danych w sieci. Znajduje się w pamięci koprocatora jednostki Master i jest to jeden z najistotniejszych elementów sieci Master - Slave. Spoczywa na nim obowiązek konfiguracji całej sieci i doprowadzenie do jej sprawnego działania. Struktura scenariusza wymian składa się z listy zmiennych jakie mogą być przesyłane, czyli ich nazw, kierunków transmisji oraz czasów odświeżania.

Aby móc wykonać scenariusz wymian wymagane jest zdefiniowanie budowy ramek jakie są przesyłane. Ponieważ w sieci Master - Slave wyróżnia się dwa rodzaje transmisji:

- Master - Slave ("żądanie" oraz "rozgłoszenie"),
- Slave - Master ("odpowiedź"),

pojawia się konieczność określenia dwóch typów ramek, gdzie każda będzie mogła wykonać zadanie jednej z transmisji.

Elementarną ramkę w tego typu sieci przedstawia poniższa tabela:

1 [B]	1 [B]	n [B]	2 [B]
Adres stacji	Kod funkcji	Dane	Kontrola błędów

- **Adres stacji** - określa do której z jednostek Slave należy wysłać dane. Pewna określona wartość w tym polu oznacza "rozgłoszenie", czyli wysłanie ramki do wszystkich stacji Slave, jednak jest ona zależna od rozwiązań firmowych.
- **Kod funkcji** - definiuje rozkaz jaki Master każe wykonać Slave.
- **Dane** - wielkość tego pola jest zależna zarówno od Kodu Funkcji, jak i od konkretnych implementacji sieci. Rola tego pola nie ogranicza się do "surowych" danych, może również zawierać takie informacje jak kod funkcji dodatkowych bądź adres danych, które zależy odczytać.
- **Kontrola błędów** - słowo kontrolne, które ma gwarantować przesył poprawnych danych. Jest zależne od implementacji, może to być m.in. CRC lub LRC.
- W zależności od konkretnego rozwiązania, w ramce można wyróżnić jeszcze znaczniki początku ramki oraz końca ramki.

Pierwsze dwa pola są identyczne dla obu ramek. W ramce "żądania" w polu Adres Stacji znajduje się albo właściwy adres stacji Slave, bądź Rozgłoszenie. Z kolei w ramce "odpowiedzi" w tym polu może znajdować się albo adres Mastera, albo adres własny ramki - pozwala to Masterowi zweryfikować, czy otrzymana ramka jest od prawidłowej stacji. W polu danych ramki "odpowiedzi" mogą znajdować się albo dane, które są wynikiem wykonania rozkazu, albo jedynie potwierdzenie realizacji.

Właściwy proces tworzenia scenariusza wymian jest zależny jedynie od upodobań konfigurującego sieć.

5.2 Dziury w sieci Master - Slave

"Dziury" są negatywnym zjawiskiem w funkcjonowaniu sieci, które objawia się poprzez przerwę w zbiorze stacji. W sieci Master-Slave źródłem dziur może być wyłącznie uszkodzenie stacji abonenckiej lub jej wyłączenie. Ogółem rzecz biorąc, dziury powstają na wskutek nieobecności stacji w sieci. Skutkiem istnienia dziur jest opóźnienie wymiany informacji pomiędzy Masterem, a stacją podrzędną, co może być niebezpieczne z punktu widzenia krytycznych parametrów sieci, m.in. gwarantowanego czasu dostarczenia informacji. Rolą programistów jest zapewnienie takiej konfiguracji, aby dziury powstawały jak najrzadziej.

Prosta topologia sieci Master-Slave oraz nieskomplikowana konfiguracji w postaci scenariusza wymian umożliwiają wykrycie dziur już na wczesnym etapie. Każda wymiana informacji, jaka znajduje się w scenariuszu, jest kierowana wyłącznie do jednego adresata. Jeżeli wystąpi błąd w konfiguracji i wymiana będzie skierowana do nieistniejącego abonenta, zostanie ona wychwycona.

Odlączenie lub uszkodzenie stacji Slave może skutkować opóźnieniem czasu pracy, który jest równy sumie czasu transmisji ramki od Mastera oraz czasu oczekiwania na odpowiedź. Co więcej, może się zdarzyć, że do nieaktywnego odbiorcy wysyłana ramka posiada wiele danych i jest bardzo długa, co dodatkowo opóźnia czas reakcji.

Jednym z mechanizmów umożliwiających minimalizację strat czasowych związanych z powstawaniem dziur jest określanie liczby cykli w czasie których stacja odbiorcza nie jest odpytywana. Po zakończeniu wszystkich cykli nawiązywana jest próba połączenia ze stacją. Jeżeli i tym razem odbiorca jest nieaktywny, następuje ponowne wstrzymanie wymian z nim oraz odczekanie wcześniej ustalonej liczby cykli. Ta metoda nosi nazwę "autousuwania" nieobecnego abonenta. Zastosowanie tego rozwiązania zależy już wyłącznie od decyzji programisty, który powinien mieć możliwość włączania i wyłączania mechanizmu autousuwania.

5.3 Parametryzacja wymian

Parametryzacja jest ważnym elementem programowania koprocatora w stacji Master. Polega na wprowadzeniu przygotowanego scenariusza wymian oraz uzupełnieniu go wymaganymi informacjami.

5.3.1 Konfiguracja tablicy wymiany

- **Numer wymiany** - jest to niezbędny parametr aby umożliwić zastosowanie mechanizmu wymian wyzwalanych, które są inicjowane przez aplikację. Liczba wymian w pojedynczym cyklu jest zależna od programisty i w zależności od konkretnej implementacji sieci, może osiągnąć nawet kilkaset.
- **Numer abonenta** - określa do której ze stacji Slave wymiana jest kierowana.
- **Kod operacji**
- **Adres pobrania danych** - definiuje miejsce w pamięci jednostki centralnej Master, które będzie źródłem danych.
- **Adres przesłania danych** - definiuje miejsce w jednostce centralnej stacji Slave, gdzie zostaną przekazane dane.
- **Liczba danych**
- **Wybór trybu przesłania** - może być *periodyczny* (zwany też *cyklicznym*), czyli wymiana wykonuje się w każdym cyklu pracy sieci, lub *wyzwalany*, czyli wymiana realizowana jest przez program jednostki centralnej.
- **Autousuwanie** - określa czy przy braku odpowiedzi ze strony abonenta zostaje uruchamiany ten mechanizm. Został opisany w poprzednim podrozdziale.
- **Adres słowa raportu wymiany z daną stacją**

5.3.2 Konfiguracja tablicy parametrów wymiany

Dla każdej wymiany z osobna należy wypełnić jej tablice parametrów, na którą składają się m.in.:

- Maksymalny czas oczekiwania na odpowiedź od stacji Slave T_{OD} - bardzo ważny parametr, którego wartość powinna być wynikiem dokładnej analizy obciążenia sieci (w szczególności czas pracy abonenta oraz czas trwania cyklu jego jednostki centralnej T_A).
- Liczba prób nawiązania łączności z abonentem - określa maksymalną liczbę prób nawiązania połączenia w przypadku braku odpowiedzi ze strony stacji Slave. Ma wielki wpływ na przepustowość sieci oraz zjawisko powstawania "dziur" [5.2].
- Maksymalna liczba cykli przy braku odpowiedzi - w czasie trwania tych cykli nie będzie prób nawiązania komunikacji ze stacją Slave, którą wykryto jako nieobecną. Ten parametr również został opisany w podrozdziale dotyczącym zjawisko "dziur" [5.2].
- Przerwa czasowa między dwiema transakcjami typu "rozgłoszenie" (*broadcast*).
- Czas opóźnienia występujący przed oraz po transmisji ramki.
- Czas autoryzacji w przypadku pracy z modemami.

5.3.3 Parametry globalne

Istnieje pewna część parametrów, które muszą zostać zdefiniowane globalnie, tzn. nadrzędnie dla każdej wymiany w sieci. Wszystkie wymiany muszą pracować zgodnie z nimi aby umożliwić prawidłową pracę sieci.

- Tryb transmisji, w przypadku sieci Modbus można wyróżnić:
 - RTU - transmisję bitową
 - ASCII - transmisję znakową

Tryby transmisji nie mogą być mieszane i w czasie trwania transakcji należy zachować tylko jeden z nich. Nie każdy protokół oparty na Master-Slave posiada kilka trybów, np. Magistrala LIN.

- Liczba bitów stopu
- Rodzaj kontroli parzystości
- Liczba bitów definiująca pojedynczy znak transmisji L_{BZ}
- Szybkość transmisji V

5.4 Zjawiska zachodzące pomiędzy koprocesorem, a jednostką centralną

Analiza procesów zachodzących na styku "koprocesor - jednostka centralna" jest naturalną koniecznością wynikającą z programowania i parametryzacji wymian w sieci. Jest to niezbędne aby precyzyjnie wyznaczyć dwa bardzo istotne parametry związane z konfiguracją sieci:

- T_{OD} - czas oczekiwania stacji Master na odpowiedź od stacji Slave
- T_{GOT} - czas oczekiwania na gotowość jednostki centralnej stacji Slave

W pierwszej kolejności należy określić parametr T_{GOT} , który ma umożliwić stacji Slave spokojne wykonanie powierzonych jej zadań. Parametr ten może zostać ustawiony albo globalnie dla wszystkich stacji Slave, albo lokalnie dla każdej z nich. Wówczas każdy z abonentów posiada indywidualną wartość T_{GOT_j} , zależną od jego potrzeb. Sam parametr można zdefiniować następującym wzorem (indeks dolny i oznacza parametr stacji Master, natomiast j parametr jednej ze stacji Slave):

$$T_{GOT_j} = T_{DR} + \max(T_{AR_j}) + \max(T_{A_j}) + \max(T_{PR_j})$$

Gdzie:

- T_{DR} - czas detekcji ramki
- $\max(T_{AR_j})$ - maksymalny czas analizy ramki
- $\max(T_{A_j})$ - maksymalny czas trwania cyklu pracy w stacji Slave
- $\max(T_{PR_j})$ - maksymalny czas przygotowania ramki odpowiedzi przez stację Slave

Innymi słowy, największy dopuszczalny parametr T_{GOT} oznacza maksymalny czas jaki jest potrzebny co najmniej jednej ze stacji Slave na wykrycie ramki od stacji Master, jej poprawne przetworzenie i udzielenie odpowiedzi.

W drugiej kolejności należy określić parametr T_{OD} , który powinien być większy od największej wartości T_{GOT} i jednocześnie powiększony o czas wykonywania wszystkich czynności niezbędnych do wysłania ramki żądania z stacji Master oraz odebrania ramki odpowiedzi ze stacji Slave. Możemy zatem T_{OD} zdefiniować następująco:

$$T_{OD} = \max(T_{GOT_j} + T_{PR_i} + T_{TR_i} + T_{TR_j} + T_{DR} + T_{AR_i} + T_{A_i})$$

Gdzie:

- T_{GOT_j} - największy czas T_{GOT} spośród czasów stacji Slave
- T_{PR_i} - czas przygotowania ramki żądania
- T_{TR_i} - maksymalny czas transmisji ramki żądania
- T_{TR_j} - maksymalny czas transmisji ramki odpowiedzi o największej możliwej długości
- T_{DR} - czas detekcji ramki odpowiedzi
- T_{AR_i} - maksymalny czas analizy ramki odpowiedzi
- T_{A_i} - maksymalny czas trwania cyklu pracy w stacji Master

Zastosowanie powyższych wzorów daje gwarancję, że w sieci Master - Slave każda transakcja zostanie poprawnie wykonana, jednak wartości te będą wpisane odgórnie. Dodatkowo uwzględniają najgorsze możliwe przypadki, przez co transakcje, które mogłyby wykonać się krócej, będą trwać tak samo długo jak inne. Czas ten można zmniejszyć posługując się mechanizmem autousuwania, ale należy korzystać z niego z rozwagą.

5.5 Czas trwania cyklu sieci

Jest to niezwykle ważny parametr, który określa czas nadawania pomiędzy stacjami, jak również możliwości sieci do wykonywania określonych zadań.

Aby móc przystąpić do określenia tego czasu potrzebne są pewne założenia i uproszczenia:

- W omawianej sieci nie występuje zjawisko "dziur" [5.2]
- Znany jest scenariusz wymian - w sieci istnieje N wymian
- Znane są długości wszystkich ramek

Przystępujemy zatem do analizy naszej sieci:

1. Czas transmisji pojedynczej ramki

$$T_{TR} = \frac{[LZT] \cdot [LBZ] + [BS]}{V} \quad (4)$$

Gdzie:

- LZT - liczba znaków transmisji
- LBZ - liczba bitów w pojedynczym znaku
- BS - liczba bitów sterujących
- V - prędkość transmisji

2. Czas transmisji ramki żądania wysyłanej przez stację Master

$$T_{TR_i} = \frac{[LZT_i] \cdot [LBZ] + [BS]}{V} \quad (5)$$

3. Czas transmisji ramki odpowiedzi wysyłanej przez stację Slave

$$T_{TR_j} = \frac{[LZT_j] \cdot [LBZ] + [BS]}{V} \quad (6)$$

4. Z tymi danymi można przystąpić do obliczenia czasu pojedynczej wymiany, która jest równa sumie czasów wysłania i wykonania żądania, przesłania i odebrania odpowiedzi oraz analizy i wpisu wyniku transakcji:

$$T_{W_i} = \underbrace{T_{PR_i} + T_{TR_i}}_{T_{Zi}} + \underbrace{T_{DR} + T_{AR_j} + T_{A_j} + T_{PR_j} + T_{TR_j}}_{T_{Oj}} + \underbrace{T_{DR} + T_{AR_i} + T_{A_i}}_{T_{RA_i}} \quad (7)$$

Gdzie:

- T_{Zi} - czas emisji żądania, składa się z czasów przygotowania i transmisji ramki żądania
- T_{Oj} - czas emisji odpowiedzi, składa się z czasów detekcji i analizy ramki żądania, przygotowania i transmisji ramki odpowiedzi oraz czasu oczekiwania na sygnał końca dla stacji Slave
- T_{RA_i} - czas analizy i wpisu wyniku wymiany, składa się z czasów wykrycia i analizy ramki oraz czasu oczekiwania na sygnał końca dla stacji Master

5. Teraz wystarczy ten czas przenieść na wykonania N wymian w sieci:

$$T_{CW} = \sum_{i=1}^N T_{W_i} = \sum_{i=1}^N T_{PR_i} + T_{TR_i} + \sum_{j=1}^N T_{AR_j} + T_{A_j} + T_{PR_j} + T_{TR_j} + \sum_{i=1}^N T_{AR_i} + T_{A_i} + 2 \cdot N \cdot T_{DR} \quad (8)$$

6. Powyższy czas może się zwiększyć jeżeli w sieci będą występować "dziury", które skutkują powtarzaniem wymian (repetycjami). Dla N_R powtarzanych transmisji w całym scenariuszu wymian (realizacja wymian periodycznych) w najgorszym możliwym przypadku dodatkowy czas wyniesie:

$$T_{REP} = \sum_{i=1}^N N_{R_i} \cdot T_{TR_i} + N_R \cdot T_{OD_i} \quad (9)$$

Gdzie:

- T_{TR_i} - czas transmisji żądania dla i -tego powtórzenia
- N_{R_i} - liczba powtórzeń do i -tego abonenta
- N - liczba abonentów
- T_{OD_i} - czas czekania na odpowiedź przez stację Master

7. Zatem łączny czas trwania cyklu powiększony o czas realizacji repetycji jest równy:

$$T_{W_{MAX}} = T_{CW} + T_{REP} \quad (10)$$

8. Na koniec należy jeszcze uwzględnić realizację wymian wyzwalanych. Ten kolejny dodatkowy czas jest całkowicie zależny od rozwiązań firmowych - należy znać liczbę wymian wyzwalanych oraz postacie ich funkcji.

Zakładając, że w sieci realizowana jest tylko jeden rodzaj wymiany wyzwalanej oraz w jednym cyklu pracy zostanie zrealizowanych N_W takich wymian, czas zwiększy się dokładnie o N_W cykli aplikacji, które je uruchomiła. Wówczas czas zwiększa się o:

$$T_{WW} = N_W \cdot T_{A_j} + \sum_{i=1}^{N_W} T_{W_i} \quad (11)$$

Gdzie:

- T_{A_j} - czas oczekiwania na sygnał końca dla stacji Slave
- T_{W_i} - czas wykonania i -tej wymiany wyzwalanej

9. Ostatecznie wzór na czas trwania cyklu sieci przyjmuje postać:

$$T_{W_{MAX}} = T_{CW} + T_{REP} + T_{WW} \quad (12)$$

Gdzie T_{CW} oznacza czas trwania cyklu dla bezbłędnie działającej sieci, a T_{REP} oraz T_{WW} określają potencjalny nakład czasowy.

5.6 Sprawność sieci i jej przepustowość

Znając całkowity czas pojedynczej transakcji danych ze wzoru (7), można przystąpić do wyliczenia wartości określających efektywność czasową przesyłania danych w sieci:

- **Sprawność** wyznacza stosunek czasu przesyłu informacji (bitów z danymi) do całkowitego czasu przesyłu danych w transakcji. Liczona w procentach.
- **Przepustowość** pozwala określić maksymalną ilość informacji, jaka zostanie przesłana w jednostce czasu. Liczona w bitach na sekundę.

Wartości te przedstawia się za pomocą poniższych wzorów

$$\eta = \frac{8 \cdot n}{T_{W_i}} \quad (13)$$

$$P = \frac{8 \cdot n}{T_{W_i}} \quad (14)$$

Gdzie:

- η - sprawność
- P - przepustowość
- n - liczba bajtów danych
- V - szybkość transmisji
- T_{Wi} - Całkowity czas trwania pojedynczej transakcji danych (wliczając ramkę żądania)

Ponieważ wartość T_{Wi} nie jest stałą zdefiniowaną globalnie, a indywidualną dla każdej ramki, wartości obu wielkości można wyrazić za pomocą przedziałów, używając w tym celu minimalnego i maksymalnego czasu trwania transakcji danych.

W praktyce dość często sprawność i przepustowość obliczana jest po odrzuceniu narzutu czasowego powstałego w wyniku dodatkowych akcji na magistrali, nie będących transmisją danych, m. in. czasu detekcji T_{DR} czy czasu transmisji ramki żądania T_{TR} . W tym celu w miejsce T_{Wi} podstawiamy równanie ze wzoru (4). Po zamianie uzyskuje się nowe wartości, obejmujące wyłącznie czas transakcji danych.

$$\eta = \frac{\frac{8 \cdot n}{V}}{\frac{[LZT] \cdot [LBZ] + [BS]}{V}} = \frac{8 \cdot n}{[LZT] \cdot [LBZ] + [BS]} \quad (15)$$

$$P = \frac{\frac{8 \cdot n}{[LZT] \cdot [LBZ] + [BS]}}{V} = \frac{8 \cdot n \cdot V}{[LZT] \cdot [LBZ] + [BS]} \quad (16)$$

5.7 Możliwości optymalizacji

Niezależnie od stosowanego podejścia, sprawność i przepustowość rośnie wraz z długością ramki. Należy jednak pamiętać, że jej zwiększanie niesie ze sobą również skutki uboczne - rośnie czas transmisji oraz cykl przetwarzania (co z kolei zwiększa czas trwania cyklu sieci).

W celu polepszenia parametrów czasowych, przede wszystkim powinno się poprawić czas trwania wymian - jest to najbezpieczniejszy sposób usprawnienia przepustowości i sprawności sieci (nie licząc zmiany sterowników i/lub koprocetorów na ich szybsze odpowiedniki). Optymalizacja powinna dotyczyć zwłaszcza jednostki nadrzędnej Master, ponieważ jest on inicjatorem lub odbiorcą każdej transakcji w systemie. Usprawnienie działania koprocetorów stacji Slave również poprawia parametry czasowe, lecz opatrzone jest mniejszym priorytetem.

Można wyodrębnić kilka zasad, które pozwalają na usprawnienie działania sieci:

- Skrócenie czasu trwania cyklu jednostki Master - można to uzyskać np. za pomocą podziału programu sterownika na mniejsze fragmenty, by zwiększyć częstotliwość komunikacji jednostki centralnej z koprocetorem.
- W przypadku, kiedy Master kilkakrotnie wymienia informacje z tą samą jednostką podrzędną, można zrezygnować z raportu wymiany. Następuje wtedy grupowanie i wysłanie jednego raportu w ramach grupy zamiast pojedynczego dla każdej zrealizowanej transakcji.
- Innym sposobem jest grupowanie danych przeznaczonych dla tego samego Slave'a już w jednostce centralnej Mastera, a następnie wysłanie ich w jednej, dłuższej ramce.
- Wysyłanie jak największej liczby informacji w trakcie wymian wyzwalanych - najlepiej, aby były to sporadycznie używane polecenia, lub dane o mniejszym priorytecie. Ponieważ wymiany wyzwalane są stosowane rzadziej w porównaniu do cyklicznych, sieć zostanie ociążona z niepotrzebnych, nadmiarowych wymian. Dodatkowo, uzyskamy więcej miejsca na pilne transakcje.
- Należy zadbać, aby wymiany cykliczne z tym samym abonentem nie następowały zaraz po sobie - po każdej transakcji Slave potrzebuje odrobiny czasu, aby zidentyfikować (zinterpretować) polecenie od arbitra sieci, skomunikować się ze swoją jednostką centralną oraz stworzyć ramkę odpowiedzi.

- Szczególny przypadek powyższego punktu: jeżeli do jednej stacji podrzędnej wysyłany jest rozkaz zapisu i odczytu, to wpierrw należy dokonać odczytu danych, a następnie zapisu. Taka kombinacja jest szybsza o jeden cykl jednostki centralnej Slave'a dla każdej takiej pary.

6 Podsumowanie

Większość współczesnych systemów informatycznych stosowanych w przemyśle ze względu na swój format jest systemami rozproszonymi czasu rzeczywistego. By zapewnić im stabilność działania i brak awarii, należy zagwarantować aby poszczególne fragmenty składające się na całość działania wykonywały się w określonym, ograniczonym czasie. Taki efekt można uzyskać przede wszystkim implementując odpowiedni protokół komunikacji.

Jednym z najpopularniejszych i najbardziej efektywnych protokołów spełniających warunek determinizmu czasowego jest Master-Slave, oparty na komunikacji pomiędzy stacją nadrzędną i wieloma stacjami podrzędnymi. Nawet w przypadku pojawienia się stanu niepożądanego ("dziur") dla danej sieci M-S istnieje możliwość obliczenia zdefiniowanego czasu transakcji posiadając informację o ilości i czasie repetycji żądania. Większość logiki jest realizowana po stronie stacji nadrzędnej, dlatego w celu optymalizacji czasowej w pierwszej kolejności tam należy wykonać usprawnienia.

Literatura

- [1] Andrzej Kwiecień, *Analiza przepływu informacji w komputerowych sieciach przemysłowych*, Gliwice 2002
- [2] Wojciech Mielczarek, *Szeregowe interfejsy cyfrowe*, Helion, Gliwice 1997
- [3] Piotr Gaj, *Zastosowanie protokołu TCP/IP do transmisji informacji dla potrzeb przemysłowych systemów kontrolno-nadzorczych. Rozprawa doktorska.*, Gliwice, 2003