



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

8 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:

- Automatykacja wykonywanych czynności
- Przyspieszenie obliczeń
- Autokorekcja w przypadku, kiedy dojdzie do częściowej awarii systemu

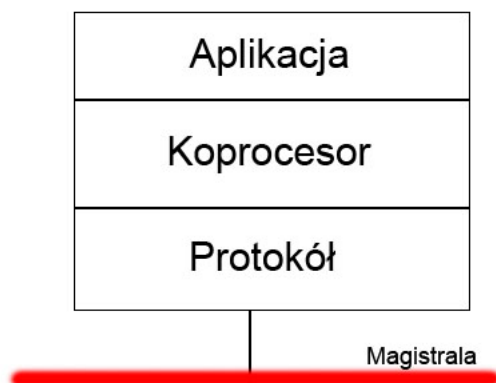
Przy stosunkowo niewielkim koszcie (zaprojektowanie, zaprogramowanie, instalacja takiego systemu), problem pojawił się przy dalszej rozbudowie branży przemysłowej (co dla części informatycznej oznaczało przede wszystkim rosnącą złożoność obliczeniową zastosowanych algorytmów czy większą liczbę urządzeń 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 również 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 wieloprocessorowych), 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 najczęściej pełni 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 transmisyjne), 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 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. 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). Naistotniejsze jest, aby czas dostępu do pamięci został obciążony krytycznym czasem granicznym.

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 jej 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.

4 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

4.1 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 należy 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.

4.2 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.

Odłączenie 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 start 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.

4.3 Parametryzacja wymian

Literatura

- [1] Andrzej Kwiecień, *Analiza przepływu informacji w komputerowych sieciach przemysłowych*, Gliwice 2002
- [2] Wojciech Mielczarek, *Szeregowe interfejsy cyfrowe*, Helion, Gliwice 1997