



POLITECHNIKA ŚLĄSKA W GLIWICACH

PRZEMYSŁOWE SYSTEMY ROZPROSZONE

Analiza czasowa przepływu informacji w sieciach o protokole Master-Slave

AUTORZY:

Jakub Barbuletis

Dominik Korda

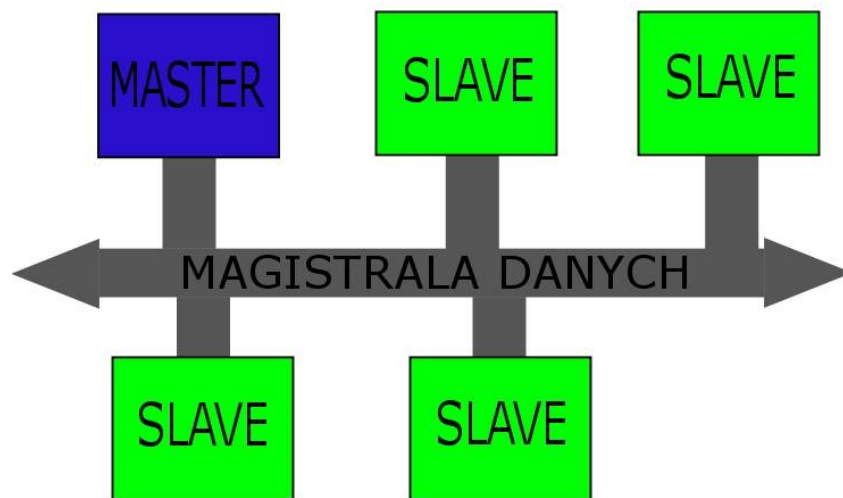
23 lutego 2015

1 Analiza tematu

Celem niniejszej pracy jest zobrazowanie zależności w przesyłach danych, między poszczególnymi stacjami działającymi w sieci opartej o protokół Master - Slave.

2 Budowa i działanie sieci Master - Slave

W sieciach opartych o model komunikacyjny Master - Slave, można wyróżnić wiele stacji typu slave, oraz jedną stację master. Jedynie stacja master może swobodnie wysyłać i żądać dane od innych stacji. Slave natomiast jeśli potrzebuje jakiejś informacji, której cykliczna transmisja nie została przewidziana na etapie projektowania sieci musi skorzystać z tzw. wymiany wyzwalanej (na które dodatkowo należy wziąć poprawkę przy projektowaniu scenariusza wymian w sieci).



Rysunek 1: Struktura sieci master-slave

2.1 Scenariusz wymian

Ze względu na swój sposób działania sieć master - slave daje nam ogromny zakres kontroli nad wymianami danych zachodzącymi w sieci. Jest tak, ponieważ każdą z nich trzeba samemu zaplanować - robi się to za pomocą tzw. *scenariusza wymian*, projektowanego równolegle z siecią i zależnemu od pracy poszczególnych komponentów systemu rozproszonego.

W scenariuszu znaleźć się muszą wszystkie wymiany, których dokonanie w określonym okresie czasu jest niezbędne dla poprawnego działania systemu (oraz spełniania przez niego wymagań czasowych).

Przyjmuje on postać tabeli, w której zapisane są takie parametry jak :

- tryb transmisji danych
 - ASCII
 - RTU
- liczba bitów stopu i parzystości
- liczba bitów przypadająca na jeden znak transmisji
- prędkość transmisji

Dodatkowo uwzględniane są parametry indywidualne dla każdej wymiany:

- numer wymiany

- adres abonenta
- kod operacji
- adres danych jednostki master do wysłania, bądź adres danych do otrzymania od stacji slave
- rozmiar danych
- wybór trybu - wymiana okresowa bądź wyzwalana
- ustalenie czy należy automatycznie usunąć abonenta przy braku odpowiedzi
- adres słowa raportu wymiany
- parametry związane raczej ze sprzętową konfiguracją wymiany:
 - graniczny czas oczekiwania na odpowiedź T_{ODP}
 - ilość prób ponownego połączenia w przypadku przekroczenia czasu T_{ODP} , określane dalej jako L_{REP}
 - liczba cykli sieci, po których podjęta zostanie ponownie próba komunikacji (w przypadku gdy żadna z prób komunikacji z punktu powyższego nie zakończyła się sukcesem), określana dalej jako L_{COCZK}
 - długość przerwy między kolejnymi transmisjami rozgłoszeniowymi
 - czas opóźnienia przed transmisją ramki oraz po jej zakończeniu.
 - czas autoryzacji mający znaczenie przy pracy z modemami

Jako, że nie każde dane należy transmitować w regularnych odstępach czasu, dodany został mechanizm tzw. *wymian wyzwalanych*. Jest to wymiana, którą może zainicjować stacja slave w pakiecie odpowiedzi - informuje ona stację slave, że potrzebuje określone dane, a master z czasem je jej dostarczy.

Warto pamiętać, że zawartość scenariusza wymian, oraz sposób konfiguracji go będzie się różnić w zależności od użytego koprocatora sieci, oraz wybranego protokołu.

3 Podstawowe parametry czasowe

Podstawowym parametrem jest długość trwania tzw. *cyklu wymian* czyli odcinka czasu, na którym pojawiają się wszystkie wymiany konieczne dla sprawnego działania systemu. Jest on zazwyczaj ograniczony od góry wymaganiami czasowymi, jakie musi spełniać system aby kwalifikował się jako system czasu rzeczywistego.

W *cyklu sieci* wyróżnić można pojedyncze *wymiany*, które należą zazwyczaj do jednej z następujących kategorii:

- *zapytanie* bądź *polecenie sterujące* wraz z odpowiedzią
- transmisja rozgłoszeniowa (w tego typu transmisji nie występują odpowiedzi od stacji slave)

3.1 Przebieg pojedynczej wymiany

Aby dowiedzieć się ile trwa cykl wymiany między stacją master a stacją slave, należy szczegółowo przeanalizować z jakich etapów składa się cykl wymiany i jak w wyniku tego należy sieć skonfigurować aby działała stabilnie i niezawodnie. Dwa podstawowe parametry, które mają na to wpływ to:

- Czas oczekiwania przez stację master na odpowiedź od stacji podrzędnej - T_{ODP}
- Czas oczekiwania na gotowość stacji nadrzędnej - T_{GOT}

Rozważmy, co się stanie, jeśli powyższe czasy zostaną źle dobrane.

Za krótki czas T_{GOT} będzie powodował, że stacja slave notorycznie będzie zgłaszała brak gotowości stacji master i niemożność zrealizowania wymiany.

Za krótki czas T_{OODP} będzie z kolei powodował częste komunikaty stacji master o braku połączenia ze stacją slave.

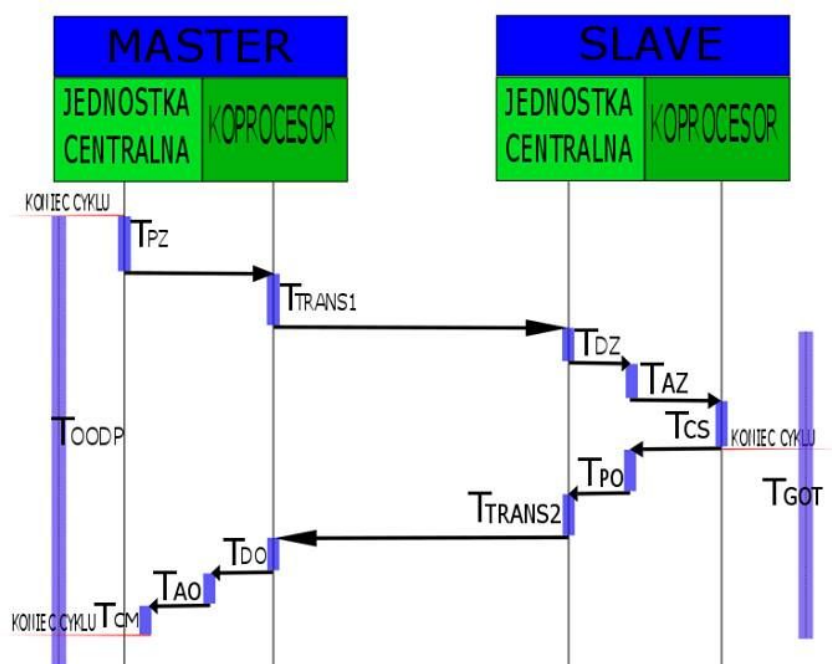
Co więcej, powyższe parametry warto dobrać z pewnym marginesem, zabezpieczających nas od wszelkich opóźnień mogących wystąpić na etapie transmisji, bądź przetwarzania danych na granicy Jednostki Centralnej i Koprocatora Sieci.

Przeanalizujemy co się dzieje w ciągu pojedynczej wymiany i jakie opóźnienia czasowe tam występują.

Przed wszystkim jednostka centralna stacji master przekazuje do koprocatora informacje jakie żądanie musi zostać zrealizowane. Po zakończeniu cyklu automatu master musi zostać przygotowana ramka żądania, bo przecież abonent docelowy musi zostać zaadresowany, należy zapewnić kontrolę poprawności danych sumą CRC itp. Określimy ten czas jako T_{PZ} .

Przygotowaną ramkę, należy przesłać łączem fizycznym do abonenta docelowego - czas przesyłu jest bezpośrednio związany z właściwością łącza i ustaloną prędkością transmisji danych. Czas propagacji danych po łączu określimy jako T_{TRZ} . Po wysłaniu ramki przez stację master, koprocator docelowej stacji slave musi wykryć tą ramkę, i odczytać jej adres, bo przecież wcale nie musi być celem transmisji - jest to czas detekcji ramki T_{DZ} . Po ustaleniu przez koprocator, że przesłane żądanie ma być zrealizowane, należy żądanie przeanalizować, i poinformować jednostkę centralną jakie działania ma podjąć, bądź też jakie dane przygotować do transmisji - określimy ten czas jako T_{AZ} . Po przekazaniu danych do jednostki centralnej musi się wykonać cykl automatu stacji slave, w którym dane zostaną przygotowane do transmisji do mastera, bądź zostanie wykonana czynność nakazana w otrzymanym żądaniu - czas cyklu można określić jako T_{CS} .

Po cyklu automatu następuje jakby "odwrócenie" procesu - stacja slave przygotowuje ramkę odpowiedzi, a master będzie ją odbierał. W skład tego procesu wejdą następujące czasy: czas przygotowania ramki odpowiedzi - T_{PO} , czas transmisji ramki odpowiedzi - T_{TRO} , czas detekcji ramki odpowiedzi - T_{DO} , czas analizy ramki odpowiedzi - T_{AO} i wreszcie czas cyklu automatu stacji master, w którym nastąpi przetworzenie odpowiedzi oraz zapisanie informacji o wymianie zakończonej sukcesem do raportu - T_{CM} .



Rysunek 2: Przebieg poszczególnych procesów podczas pojedynczej wymiany

Warto pamiętać, że dla poprawnej pracy sieci powinny zachodzić następujące zależności:

$$T_{GOT} > T_{DZ} + T_{AZ} + T_{CS} + T_{PO} \quad (1)$$

$$T_{OODP} > T_{PZ} + T_{TRZ} + T_{DZ} + T_{AZ} + T_{CS} + T_{PO} + T_{TRO} + T_{DO} \quad (2)$$

Warto zwrócić uwagę na fakt, że w przypadku braku sygnału *końca cyklu KC* w czasie T_{GOT} koprocessor stacji slave wyśle odpowiedź *NACK* powodując uznanie wymiany za nieudaną. Jest to o tyle ważne, że wpływ na to ma czas trwania cyklu automatu slave T_{CS} , którego długości nie da się wyliczyć, a jedynie zmierzyć. Dodatkowo będzie ona się prawdopodobnie zmieniała w zależności od działań podejmowanych przez stację slave, przez co może się drastycznie zmienić w wyniku komunikatów nadanych przez sieć. Takie zdarzenie może zaburzyć działanie sieci - dlatego należy doliczyć pewien margines w stosunku do przewidywanego najdłuższego czasu cyklu.

3.2 Analiza czasu trwania cyklu wymian

Na początek warto policzyć, ile czasu zajmie transmisja ramki przez łącze. Jest to dość proste, bo jest to stosunek liczby bitów w ramce do prędkości transmisji. Jedyną o czym należy pamiętać, to narzut bitów sterujących z protokołu.

Czas transmisji ramki żądania:

$$T_{TRZ} = \frac{ZTZ * BZ + BS}{V[b/s]} \quad (3)$$

Czas transmisji ramki odpowiedzi:

$$T_{TRO} = \frac{ZTO * BZ + BS}{V[b/s]} \quad (4)$$

Gdzie: ZTZ , ZTO to odpowiednio liczba znaków w danych w ramce żądania i odpowiedzi, BZ to liczba bitów w znaku, BS to liczba bitów sterujących narzucanych przez protokół (np. CRC itp.), a V to prędkość transmisji po łączy, wyrażana w *bitach/sek*.

Znając już czasy poszczególnych transmisji, można się zabrać za wyliczenie już bardziej złożonych rzeczy, jak choćby czasu trwania pojedynczej wymiany - jak łatwo się domyślić będzie to suma wszystkich składowych czasowych:

$$T_{WYMP} = T_{PZ} + T_{TRZ} + T_{DZ} + T_{AZ} + T_{CS} + T_{PO} + T_{TRO} + T_{DO} + T_{AD} + T_{CM} \quad (5)$$

Warto zwrócić uwagę, na fakt, że bardzo duży wpływ na czas trwania takiej wymiany mają cykle automatów mastera jak i slave'a. Dodatkowo są to parametry, które można zmienić w sposób dość prosty, gdyż nie wymaga to zmiany sprzętu. Natomiast zmiany parametrów detekcji ramki, czy analizy żądania bądź przygotowania odpowiedzi, są już zależne od użytego koprocessora i ich skrócenie wiązałoby się z koniecznością wymiany sprzętu.

Kolejnym parametrem, który można sobie prosto wyliczyć jest czas trwania wszystkich N wymian:

$$T_{WYMC} = \sum_{i=1}^N (T_{PZi} + T_{TRZi}) + \sum_{j=1}^N (T_{AZj} + T_{CSj} + T_{POj} + T_{TROj}) + \sum_{i=1}^N (T_{CMi} + T_{TAoi}) + N(T_{DZ} + T_{DO}) \quad (6)$$

Należy mieć na względzie, że wyliczone wyżej wartości zakładają brak jakichkolwiek retransmisji, usunięcie z sieci stacji itd. Czas wykrycia niedostępności pojedynczego abonenta wynosi:

$$T_{NA} = \sum_{i=1}^{L_{REP}} (T_{TRZ} + T_{OODP}) \quad (7)$$

Jak widać, jest to czas powtórzenia L_{REP} razy transmisji wraz z oczekiwaniem na odpowiedź. Po upływie takiego czasu, stacja master uzna abonenta za niedostępnego i ponowi próbę komunikacji po wykonaniu się $L_{CO CZK}$ cykli sieci.

W najgorszym przypadku czas wszystkich transmisji będzie równy czasowi wszystkich transmisji + maksymalnemu czasowi repetycji, czyli:

$$T_{WYMMAX} = T_{WYMC} + \sum_{i=1}^N T_{NAi} \quad (8)$$

4 Poprawa parametrów sieci oraz jej modyfikacja

4.1 Modyfikacja sieci

Każda sieć powinna obsługiwać podstawowe czynności związane z dodawaniem lub usuwaniem abonentów. Gdyby te dwa mechanizmy nie byłyby dostępne to mogłaby być po pewnym czasie cała sieć do wyrzucenia. Nie ma wtedy kontroli nad błędami w systemie. W omawianym tutaj protokole jednak występują te dwa mechanizmy i dzięki nim możemy spokojnie rozszerzać naszą sieć bądź ją zmniejszać. A w razie awarii abonenta nie powinno dojść do błędnej transmisji danych między resztą działających urządzeń

W MASTER-SLAVE pierwszy mechanizm jest realizowany w taki sposób, że to programista ma za zadanie ustalić na początku z jakimi abonentami ma być nawiązana transmisja (pod jakim adresem oni się znajdują) i w którą stronę ma odbywać się transmisja. Gdy już zostanie zdefiniowany scenariusz wymian, powinno zostać zapewnione połączenie fizyczne między slave'ami a masterem. Po uruchomieniu master wyśle ramkę rozgłoszeniową, a następnie sieć zacznie działać. W przypadku gdy chcielibyśmy dodać kolejnego abonenta do działającej już sieci, to jesteśmy zmuszeni do modyfikacji scenariusza wymian, aby slave został odpowiednio obsłużony. Jest to dosyć niewygodne rozwiązanie, ponieważ musimy przy każdej zmianie sprawdzać czy odpowiednie parametry działania sieci są poprawne. Być może dodanie kolejnego abonenta sprawi, że pojawią się błędne transmisje.

W jednym znanym zastosowaniu protokołu MASTER-SLAVE rozwiązano ten problem. Do każdego slave'a umieszczono pewną tablicę danych zwanych deskryptorami. Zawierają one informacje na temat obsługi danego slave'a. Master po wykryciu nowego slave'a po prostu wczytuje deskryptory i odpowiednio sam modyfikuje scenariusz wymian. Siecią o której teraz mowa jest popularne USB, które doczekało się już wersji 3.1 i nadal jest rozwijana. Popularność zapewne zawdzięcza temu mechanizmowi dodawania abonentów. Sprawia to, że taki system jest uniwersalny.

Innym rozwiązaniem, który można by było zastosować jest już w fazie projektowania umieszczenie dodatkowych "pustych" abonentów do sieci, których maksymalna ilość cykli sieci przy braku odpowiedzi oraz ilość prób nawiązania łączności będą na tyle duże, że w późniejszym czasie można będzie bez problemu wstawić brakującego slave'a. Jest to pewne rozwiązanie ale tu trzeba się zastanowić czy możemy sobie pozwolić na pogorszenie transmisji na rzecz rozbudowy systemu.

Drugą wspomnianą czynnością edycji sieci jest usuwanie abonenta z niej. Ten mechanizm jest bardzo potrzebny żeby sieć działała cały czas sprawnie. Najczęstszym powodem usuwania abonenta z obiegu jest nie uzyskanie od niego odpowiedzi na wysłanego do niego zapytania. Spowodowane to może być błędami związanymi z transmisją albo po prostu uszkodzeniem abonenta. W protokole MASTER-SLAVE także wymyślony sposób usuwania nieobecnych slave'ów. Najpowszechniejszym rozwiązaniem jest ustalenie dwóch ważnych parametrów o których wcześniej wspominałem. Pierwszy z nich to maksymalna liczba cykli sieci przy braku odpowiedzi. Przed ponownym wysłaniem ramki żądania do slave'a zostanie odczekana pewna liczba cykli sieci. Parametr ten poprawia znaczącą jakość działania sieci, ponieważ nie traci za każdym razem czasu na odpytywanie niedziałającego slave'a. Drugim parametrem jest ilość prób nawiązania łączności. Dzięki niemu po określonym czasie abonent zostanie po prostu wyrzucony z sieci (tzw. autousuwanie).

4.2 Poprawa parametrów sieci

Częstym zadaniem projektantów sieci jest poprawa parametrów od których zależy jej działanie. Tutaj można wyróżnić dwa sposoby poprawy parametrów:

- zwiększenie transmisji danych
- skrócenie cyklu automatu

Pierwsze rozwiązanie jest najprostsze ale nie zawsze skuteczne. Można to osiągnąć przez zwiększenie szybkości nadawania przez urządzenie albo po przez zwiększenie przepustowości łącza fizycznego. Niestety jest większe ryzyko wystąpienia błędów w takiej transmisji.

Drugi sposób jest znacznie lepszy, ponieważ nie ma zagrożeń wynikających z poprawy tego parametru. Ale jego poprawa wymaga dużo więcej czasu na analizę samego problemu. Optymalizacja kodu przez programistów lub wykorzystanie skuteczniejszych algorytmów może znacznie skrócić czas cyklu automatu.

Wymienione wyżej metody powinny poprawić parametry sieci. Ale co jeżeli wystąpią uszkodzenia linii danych albo uszkodzenia pewnych portów wejścia/wyjścia? W wyniku takich okoliczności system traci i możliwe jest, że przestanie działać prawidłowo taka sieć. Nawet najszybsza naprawa nie uchroni właściciela sieci od strat wynikających z braku działania sieci. Należy się na takie okoliczności zabezpieczyć i zastosować redundancję. Przykładowo dodatkowa linia danych może być bardzo dobrym zabezpieczeniem w razie awarii tej pierwszej. Ale niestety wiąże się to z kosztami. Nie tylko zakupu, ale także konserwacji takiej redundantnej linii. Jeżeli byłaby nieużywana w czasie działania pierwszej linii można by było pomyśleć, że jest to strata pieniędzy. Na szczęście projektant jest w stanie wykorzystać od razu drugą linię danych do zwiększenia przepustowości łącza. A jedynie w czasie awarii wykorzystywana byłaby tylko jedna linia danych. Należy więc się zastanowić jakie elementy warto powielać, a jakich lepiej nie.

5 Wnioski

Sieci używające protokół MASTER-SLAVE to idealne rozwiązanie dla tych projektantów, którzy lubią mieć dużą kontrolę nad wykonywanym projektem. Jej zalety takie jak prostota oraz uniwersalność sprawiły, że jest powszechnie używany. Nadaje się do tworzenia małych i średnich sieci, ponieważ w dość łatwy sposób zaprojektować taką sieć i można obliczyć dokładnie parametry wymagane do prawidłowego działania systemu. W dużych sieciach jest to także możliwe, aczkolwiek nakład pracy wymagany do stworzenia takiej sieci, a następnie jej utrzymanie może być nieopłacalne, a nawet niewykonalne.

Literatura

- [1] Andrzej Kwiecień *Analiza przepływu informacji w komputerowych sieciach przemysłowych*, Politechnika Śląska, Gliwice 2002
- [2] Dokumentacja sieci MODBUS firmy Modicon