ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1995

Seria: INFORMATYKA z. 28 Nr kol. 1270

Andrzej KWIECIEŃ
Piotr GAJ

SIEĆ FIP. WSTĘP DO ANALIZY CZASOWEJ

Streszczenie. W artykule przypomniano najważniejsze cechy sieci FIP. Przedstawiono budowę wszystkich typów ramek oraz oszacowano ich długość i czas transmisji. Podano również czas elementarnego cyklu.

FIP NETWORK. INTRODUCTION TO TIME ANALYSIS

Summary. The general information about FIP Protocol, structure of variables and Producer-Distributor-Consumer (PDC) model of real-time data base are presented. The structure of each type of trame is presented and time of thers transfering also. At the end, the typical cycle of exchanging data in FIP network is calculated.

RESEAU FIP. LE TEMPS DE LA TRANSMISSION

Résumé. Dans cet article on a rappelé des signes les plus importants de réseau FIP. On a présenté la construction de touts les types des cadres, et aussi on a évalué une longueuz d' eux et le temps de la transmission. On a clonné aussi le temps d' un cycle élémentaire.

1. Sieci typu FIP

Sieć FIP (Factory Instrumentation Protocol) jest siecią terenową (field network) realizującą niższe warstwy procesów sterowania i produkcji "dyskretnej". W obszarze realizacji tych warstw rozróżnia się różne rodzaje wymian informacji:

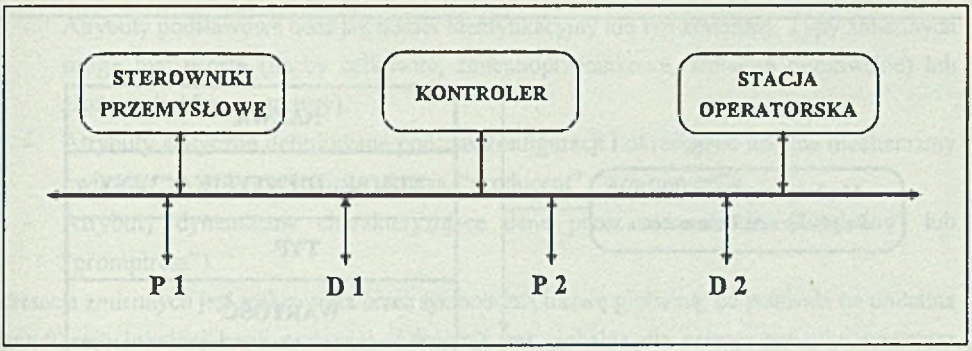
- informacje (dane) pochodzące z procesu tworzone przez układy wejścia (inicjatory),
- rozkazy tworzone przez system i zwracane do procesu i realizowane przez układy wyjścia (układy wykonawcze),
- informacje wymieniane przez różnych abonentów sieci i przez różne systemy sterowania w kontekście zastosowanej aparatury takiej jak: sterowniki przemysłowe, klasyczne lub "inteligentne" inicjatory i układy wykonawcze,
- informacje niezbędne do monitorowania, sterowania i pomiarów.

Pierwsze trzy typy wymian dotyczą komunikacji poziomej w sieci, gdzie następuje podział informacji od inicjatorów, do układów wykonawczych czy sterowników przemysłowych. Te rodzaje wymiany mają również miejsce w trakcie aktualizacji rozproszonej bazy danych. Ostatni natomiast rodzaj wymian można określić jako "pionowy" ruch danych, służący monitorowaniu i sterowaniu procesem. Te wymiany są mniej krytyczne z punktu widzenia wymagań czasowych. Sieć FIP stara się jednocześnie zaspokoić wymagania wynikające z "pionowej" i "poziomej" wymiany danych ze szczególnym uwzględnieniem periodycznej wymiany obwarowanej czasem krytycznym i aperiodycznej komunikacji z wyższymi warstwami oprogramowania.

2. Model typu PDC (PRODUCENT - DYSTRYBUTOR - KONSUMENT)

"Producent" jednostki danych jest odpowiedzialny na poziomie aplikacji za "produkowanie" danych, które mogą być periodyczne lub aperiodyczne oraz synchroniczne lub asynchroniczne, z innymi aplikacjami lub "produkcjami" danych. "Dystrybutor" danych jest natomiast odpowiedzialny za transfer danych od "Producenta" do wszystkich "Konsumentów".

"Konsumenci" danych są aplikacjami, które będąc wykonywane żądają danych. Użycie danych może być periodyczne lub aperiodyczne oraz synchroniczne lub asynchroniczne, z innymi aplikacjami. Dystrybucja danych powinna gwarantować czasową charakterystykę "produkcji" danych uwzględniającą "konsumpcję" i dostępność danych zgodnie z wymaganiami "konsumentów". Dystrybucja powinna być również periodyczna lub nie, synchroniczna lub nie, z innymi aplikacjami. Model PDC stawia na pozycji uprzywilejowanej "konsumentów". Poza tym, funkcja dystrybucji, statycznie skonfigurowana i zaprojektowana, może być modyfikowana, satysfakcjonując czasowe wymagania konsumentów. Model PDC doskonale nadaje się do zaadaptowania w poziomych wymianach i również może być zadowalający dla określenia niektórych wymian pionowych.



Rys. 1. Model PDC
Fig. 1. PDC model

2.1. Zgodność danych

Na rys.1 przedstawiono przykład aplikacji zawierający w sobie element zgodności danych. Dwa sensory P1 i P2 dokonują pomiarów ciśnienia, a sensory D1 i D2 dokonują pomiarów przepływu. Kontroler periodycznie żąda wyników wszystkich pomiarów. Sterownik przemysłowy i stacja operatorska również używają tych danych. Istotne jest, aby wszyscy odbiorcy w tym samym momencie dysponowali pomiarami zebranymi jednocześnie. Pojawiają się zatem następujące terminy:

- Czasowa zgodność produkcji danych

Jest to własność oznaczająca, że dane zostały wyprodukowane w tym samym przedziale czasowym. Pozwala to zapewnić próbkowanie wszystkich danych w tym samym momencie. Może się to okazać konieczne do zapewnienia spójności algorytmu w aplikacji użytkownika.

- Czasowa spójność transmisji

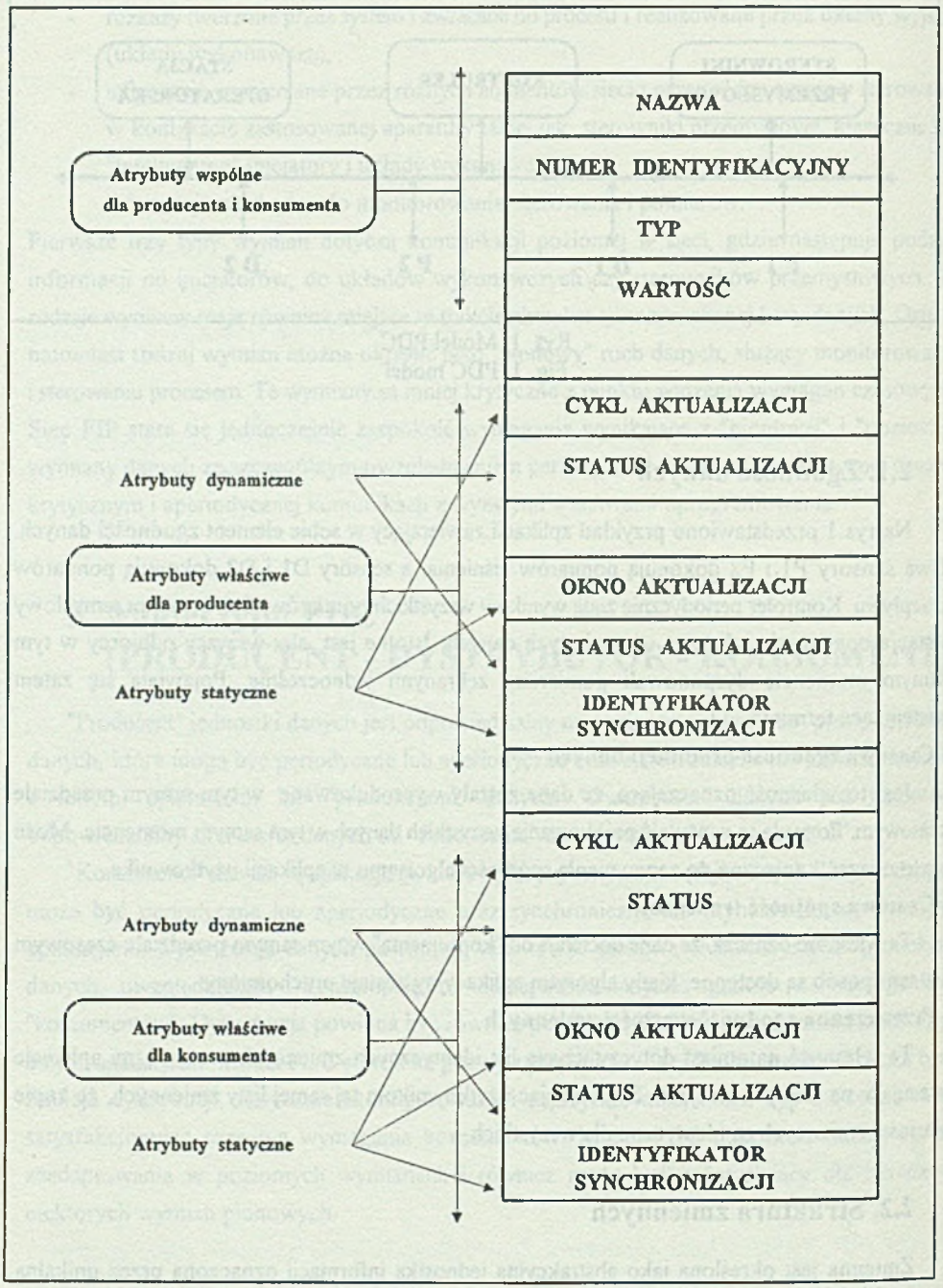
Ta własność oznacza, że dane docierają do "konsumenta" w tym samym przedziale czasowym i w ten sposób są dostępne, kiedy algorytm aplikacyjny zostaje uruchomiony.

- Przestrzenna zgodność wartości zmiennych

Ta własność natomiast dotyczy użycia list identycznych zmiennych przez różne aplikacje pracujące na różnych stacjach. Sygnalizując użytkownikom tej samej listy zmiennych, że kopie wartości zmiennych są identyczne dla wszystkich.

2.2. Struktura zmiennych

Zmienna jest określona jako abstrakcyjna jednostka informacji oznaczona przez unikalną nazwę w systemie sterowania. Jej atrybuty są związane z nazwą i są następujące (rys. 2):



Rys. 2. Struktura danych
Fig. 2. Data structure

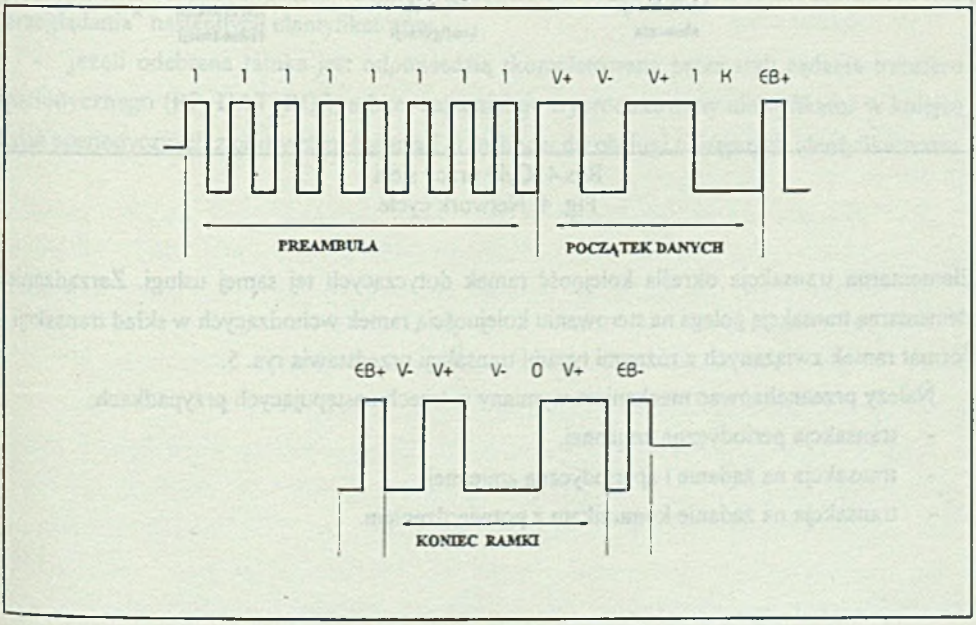
- Atrybuty podstawowe takie jak numer identyfikacyjny lub typ zmiennej. Typy zmiennych mogą być proste (liczby całkowite, zmiennoprzecinkowe, zmienne boolowskie) lub złożone (tablice, struktury).
- Atrybuty statyczne definiowane podczas konfiguracji i określające lokalne mechanizmy związane ze zmienną (synchronizacja, "producent", "konsument").
- Atrybuty dynamiczne charakteryzujące dane przez same siebie ("updating" lub "promptress").

Adresacja zmiennych jest realizowana przez symboliczną nazwę globalną, co pozwala na unikalną identyfikację lokalnej kopii zmiennej. Adresacja jest globalna dla całego systemu wewnątrz obszaru roboczego mechanizmu dystrybucyjnego. Baza danych jest rozproszona na wszystkie stacje (urządzenia) podłączone do sieci i ta sama nazwa oznacza lokalny obraz odpowiadający jej zmiennej w całym systemie.

2.3. Krótki opis techniczny usług oferowanych przez sieć FIP

2.3.1. Warstwa fizyczna

Dane transmitowane po sieci FIP są kodowane za pomocą kodu Manchester II (split - phase - code). Kod ten daje korzyści wynikające z równoległego przesyłania zegara, jak i danych. Rys.3 przedstawia zależności czasowe dla początku i końca ramki. Warstwa fizyczna również definiuje mechanizm redundancyjnego kanału transmisyjnego.



Rys. 3. Sposób kodowania ramki
Fig. 3. Frame coding

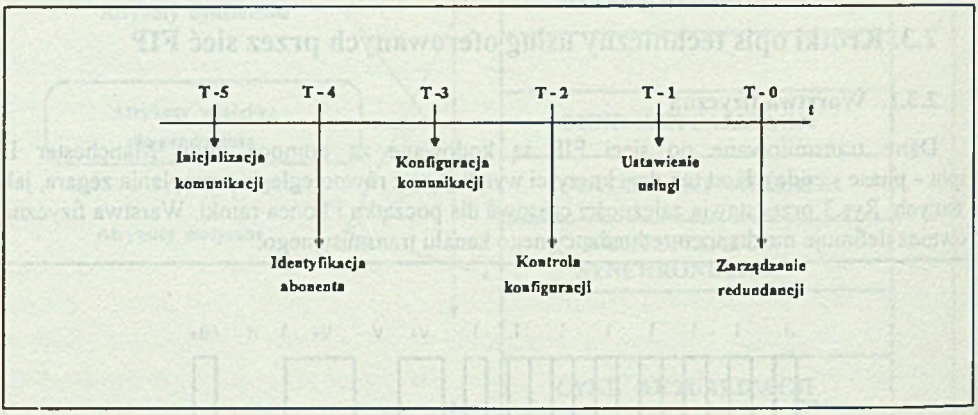
2.3.2. Warstwa łączenia

Jak już wspomniano, wszystkie wymiany danych na magistrali są zarządzane przez arbitra magistrali. Arbiter magistrali realizuje trzy podstawowe funkcje:

- periodyczny odczyt zmiennych,
- odczyt zmiennych na żądanie,
- transmisja na żądanie komunikatów.

Poza tym arbiter magistrali może zapewnić realizację funkcji synchronizacji gwarantującą stały czas trwania cyklu "odpytywania producentów" zmiennych.

Każdy typ "odpytywania" abonentów ma miejsce, odpowiednio, w periodycznym oknie i aperiodycznym oknie dla zmiennych oraz w aperiodycznym oknie dla transmisji komunikatów i synchronizacji. Cztery okna określają elementarny cykl "odpytywania". Postać cyklu elementarnego oraz czas jego trwania są definiowane przez klienta podczas konfiguracji - rys 4.



Rys.4. Cykl pracy sieci
Fig. 4. Network cycle

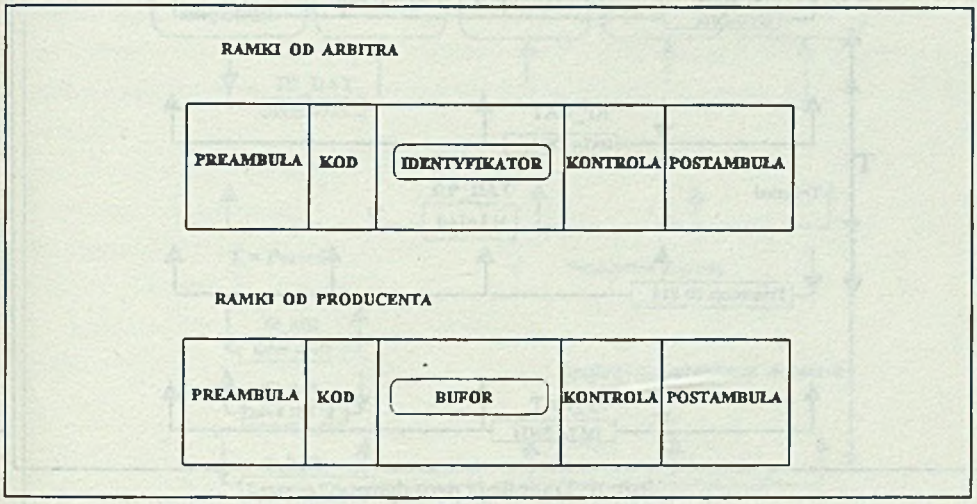
Elementarna transakcja określa kolejność ramek dotyczących tej samej usługi. Zarządzanie elementarną transakcją polega na sterowaniu kolejnością ramek wchodzących w skład transakcji. Format ramek związanych z różnymi typami transakcji przedstawia rys. 5.

Należy przeanalizować mechanizm wymiany w trzech następujących przypadkach:

- transakcja periodyczna zmiennej,
- transakcja na żądanie i aperiodyczna zmiennej,
- transakcja na żądanie komunikatu z potwierdzeniem.

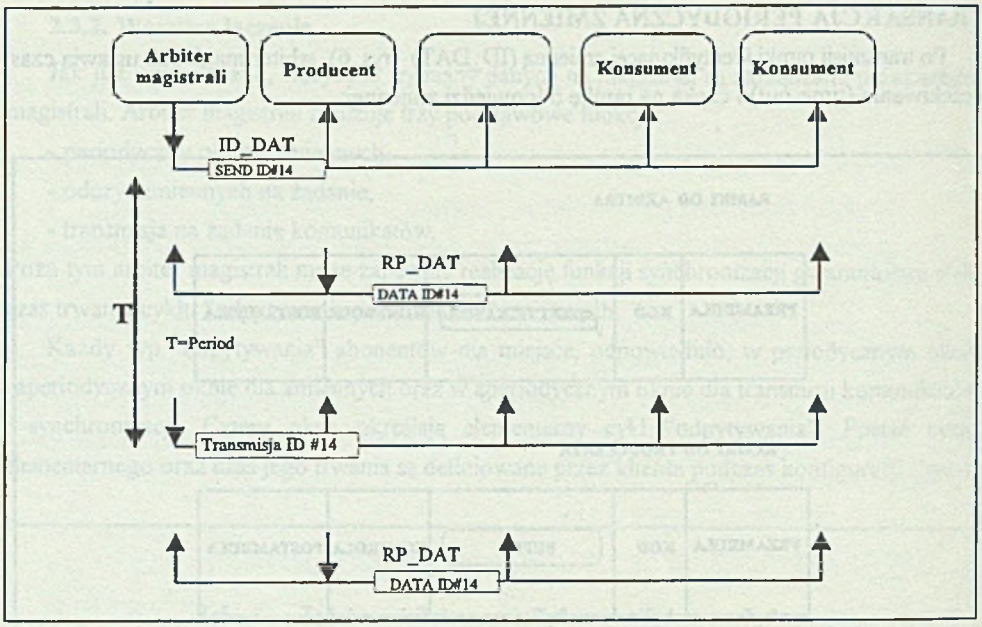
TRANSAKCJA PERIODYCZNA ZMIENNEJ

Po transmisji ramki identyfikującej zmienną (ID_DAT) (rys. 6), arbiter magistrali ustawia czas oczekiwania (time-out) i czeka na ramkę odpowiedzi zmiennej:



Rys.5. Podstawowe typy ramek
Fig.5. Types of frames

- jeżeli odebrana ramka jest ramką odpowiedzi RP_DAT, to arbiter przechodzi do "przeglądania" następnych identyfikatorów,
- jeżeli odebrana ramka jest odpowiedzią skompletowaną przez tryb żądania transferu aperiodycznego (RP_DAT_RQi), arbiter zapamiętuje wyprodukowany identyfikator w kolejce żądań aperiodycznych z priorytetem żądania i przechodzi do obsługi następnych identyfikatorów.



Rys. 6. Transakcja periodyczna
Fig. 6. Periodic transaction

Indeks "i" oznacza:

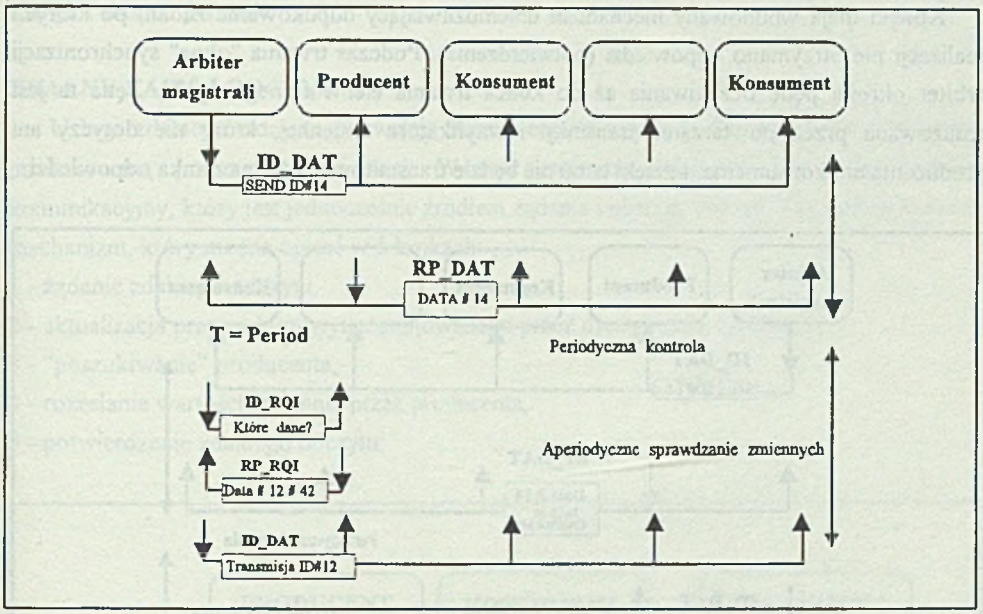
- i = 1 żądanie pilne,
- i = 2 żądanie standardowe,

- jeżeli odebrana ramka jest odpowiedzią skompletowaną przez żądanie transmisji komunikatu (RP_DAT_MSG), to arbiter zapamiętuje identyfikator tego żądania w kolejce komunikatów i przechodzi do obsługi następnych identyfikatorów,
- jeżeli odebrana ramka odpowiedzi została skompletowana przez żądanie transferu aperiodycznego (przez żądanie transferu komunikatu - RP_DAT_RQi_MSG), to arbiter zapamiętuje identyfikator tego żądania w odpowiednim zbiorze i przechodzi do obsługi kolejnego identyfikatora.

ŻĄDANIE I APERIODYCZNA TRANSAKCJA ZMIENNEJ

Kiedy zakończone jest przeglądanie identyfikatorów żądań periodycznych (rys.7.), arbiter przechodzi do obsługi listy żądań. Transmituje pierwszą ramkę identyfikatora żądań (ID_RQ1) z kolejki pilnych transferów aperiodycznych. Jeżeli kolejka jest pusta, arbiter przechodzi do obsługi kolejki żądań standardowych (niepilnych). Jeżeli kolejka ta nie jest pusta, arbiter ustawia czas oczekiwania (time-out) i oczekuje na ramkę odpowiedzi żądania. Producent, który rozpoznał swój identyfikator, transmituje w ramce RPQ i zmienne, których dotyczyło żądanie. Arbiter

zapamiętuje szereg identyfikatorów żądań (ramki typu RP_RQi) w kolejce i przechodzi natychmiast do obsługi pierwszego identyfikatora żądanej zmiennej.



Rys. 7. Żądania i aperiodyczne transakcje zmiennych
Fig. 7. Aperiodic transaction

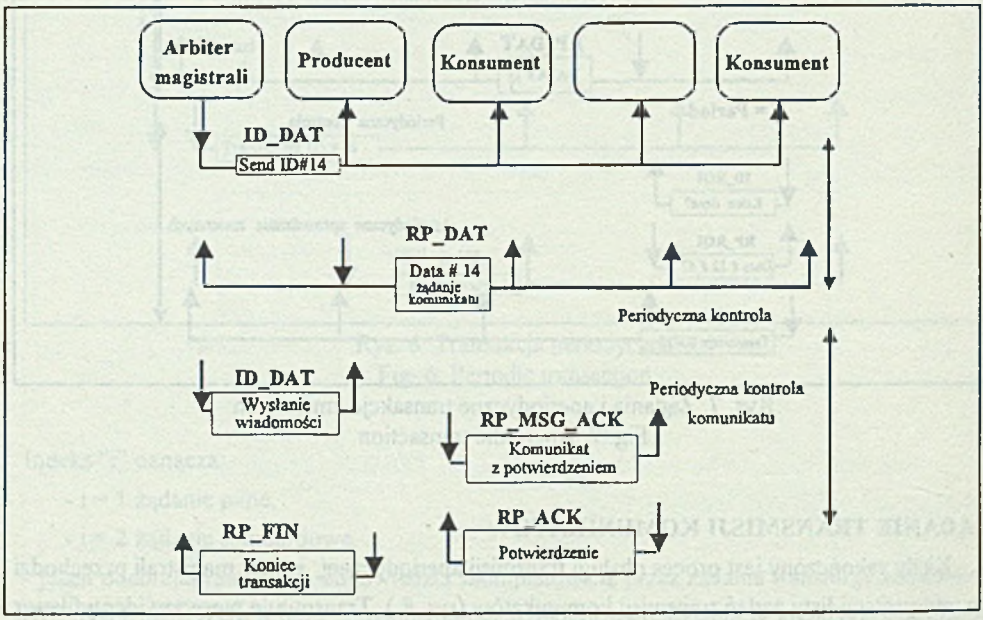
ŻĄDANIE TRANSMISJI KOMUNIKATÓW

Kiedy zakończony jest proces obsługi transmisji aperiodycznej, arbiter magistrali przechodzi do interpretacji listy żądań transmisji komunikatów (rys. 8.). Transmituje pierwszy identyfikator "komunikatu" (ID_MSG) z kolejki komunikatów i jeśli nie jest ona pusta, ustawia czas oczekiwania i oczekuje na ramkę typu "End of transaction". Abonent, który rozpoznał swój identyfikator, transmituje komunikat z listy w formie ramki RP_MSG_ACK. Jeżeli uzyska ramkę potwierdzenia RP_ACK lub jeśli nie uzyska żadnej odpowiedzi, po próbie kilku retransmisji wysyła do arbitra magistrali ramkę typu RP_FIN. Jeżeli arbiter magistrali odbierze ramkę typu RP-FIN, przechodzi do obsługi następnego identyfikatora. Jeżeli natomiast odebrana przez arbitra magistrali ramka jest typu RP_MSG_ACK lub RP_ACK, to arbiter ustawia dodatkowy czas oczekiwania (time-out) i oczekuje na koniec transakcji, czyli ramkę typu RP_FIN. W omawianych trzech typach transmisji mogą zaistnieć inne przypadki:

- jeżeli odebrana przez arbitra ramka jest innego typu niż omawiane, to arbiter wykrywa błąd i przechodzi do obsługi kolejnego identyfikatora,
- jeżeli odebrana ramka zawiera niepoprawne pole kontrolne (FCS), arbiter wykrywa błąd transmisji i przechodzi do obsługi kolejnego identyfikatora,

- jeżeli czas oczekiwania (time-out) został przekroczony arbiter sygnalizuje "zagubienie" ramki odpowiedzi i przechodzi do obsługi kolejnego identyfikatora.

Kolejki mają wbudowany mechanizm uniemożliwiający duplikowanie żądań, po których realizacji nie otrzymano odpowiedzi (potwierdzenia). Podczas trwania "okna" synchronizacji arbiter określa pętlę oczekiwania aż do końca trwania elementarnego cyklu. Pętla ta jest realizowana przez powtarzanie transmisji identyfikatora zmiennej, który nie dotyczy ani producenta ani konsumenta, i dzięki temu nie będzie transmitowana żadna ramka odpowiedzi.



Rys. 8. Żądania i transmisje komunikatów z potwierdzeniem
Fig. 8. Message transmision with acknowlegement

2.3.3. Warstwa aplikacji

Różne usługi warstwy aplikacji umożliwiają dostęp do zmiennych i umożliwiają synchronizację aplikacji użytkownika operujących na kilku abonentach sieci.

LOKALNY ZAPIS I ODCZYT

Funkcje automatycznego sterowania mają w swych interfejsach komunikacyjnych lokalny obraz produkowanych lub konsumowanych zmiennych. Są do dyspozycji dwie metody wymiany informacji pomiędzy abonentami w sieci. Wielkość skonsumowana, dla odczytu lokalnego, pochodzi z ostatniego procesu aktualizacji bazy realizowanego przez mechanizm dystrybucji zarządzany przez dystrybutora. Wartość wyprodukowana, dla lokalnego zapisu, zostaje umieszczona, jako depozyt systemu komunikacyjnego dystrybutora, do rozpowszechniania dla

wszystkich konsumentów. Lokalny odczyt i zapis są mechanizmami niezależnymi od aktywności sieci. Lokalny odczyt daje możliwość dostępu do listy zmiennych, które mogą nadchodzić od różnych abonentów sieci.

ZDALNY ZAPIS I ODCZYT

Usługi zdalnego odczytu pozwalają konsumentowi uzyskać aktualną wartość zmiennej produkowaną przez abonenta sieci. Wartość zmiennej odbierana jest przez interfejs komunikacyjny, który jest jednocześnie źródłem żądania i obsługi. Na rys. 9 przedstawiono ten mechanizm, który można opisać w 5 krokach:

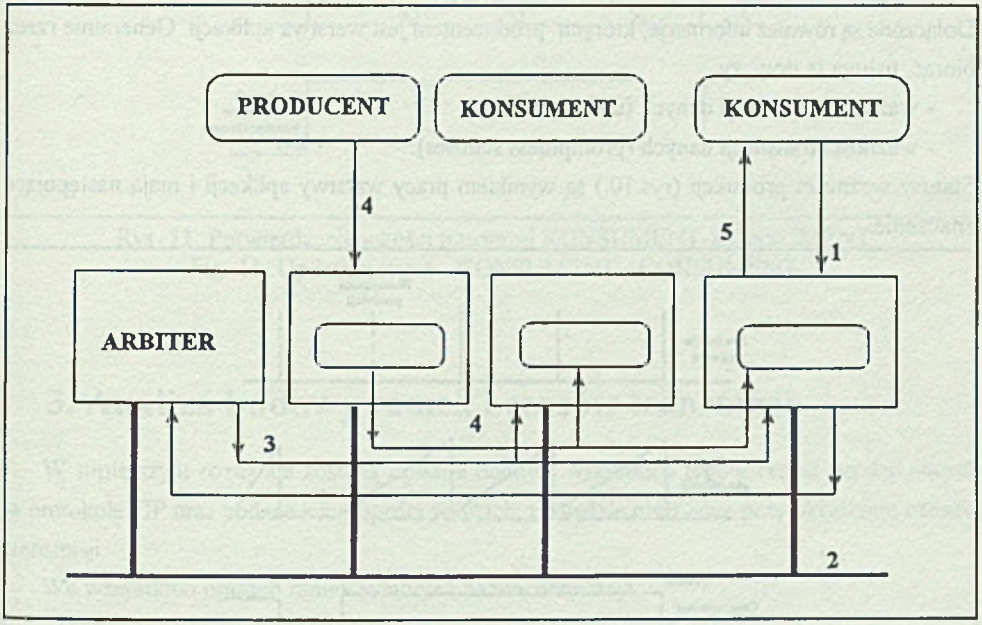
1 - żądanie zdalnego odczytu,

2 - aktualizacja przez arbitra wytransmitowanego przez dystrybutora żądania,

3 - "poszukiwanie" producenta,

4 - rozesłanie wartości zmiennej przez producenta,

5 - potwierdzenie zdalnego odczytu.



Rys. 9. Zdalny odczyt i zapis
Fig. 9. Remote write and read

USŁUGI AKTUALIZACJI

Żądania lokalnego zapisu bądź odczytu są odpowiednio inicjalizowane przez producentów i konsumentów zmiennych. Usługi aktualizacji są osiągalne, udostępnione abonentowi, który stanowi tak zwaną "trzecią osobę". "Trzecia osoba" może być producentem lub użytkownikiem zmiennej, ale może nie być zainteresowana ani konsumpcją, ani produkcją zmiennej.

USŁUGI SYGNALIZACJI

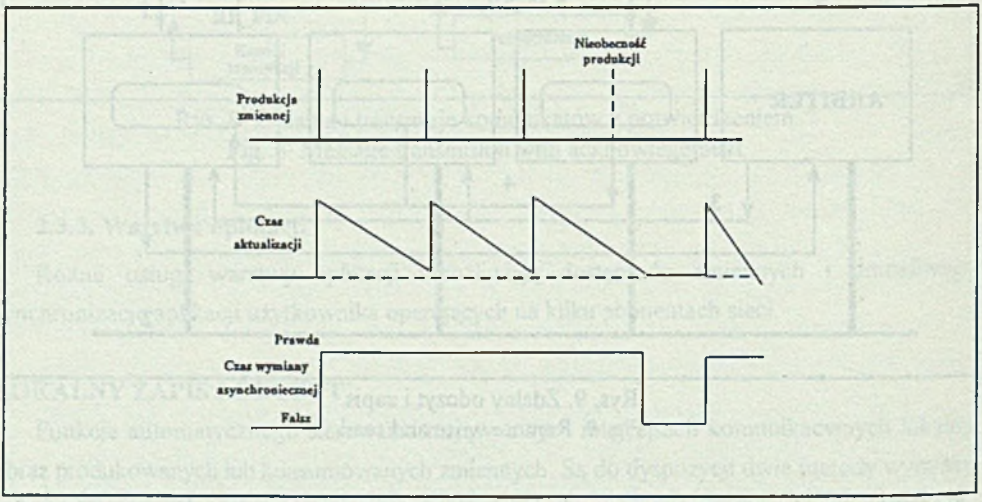
Opcjonalne usługi sygnalizacji informują konsumentów o odbiorze zmiennych, które znajdują się w ich interfejsach komunikacyjnych lub informują producentów o tym, że należy wytransmitować wyprodukowaną zmienną. Usługa ta może być użyta do sprawdzenia poprawności wykonywania operacji przez interfejs komunikacyjny, synchronizacji aplikacji użytkownika związanej z odbiorem lub nadawaniem zmiennych.

USŁUGI OKREŚLANIA JAKOŚCI ODBIERANYCH I NADAWANYCH ZMIENNYCH

Usługi te informują konsumenta o ważności wyprodukowanych i skonsumowanych danych. Dołączone są również informacje, których producentem jest warstwa aplikacji. Generalnie rzecz biorąc, usługa ta dotyczy:

- ważności produkcji danych (update statuses),
- ważności transmisji danych (promptuess statuses).

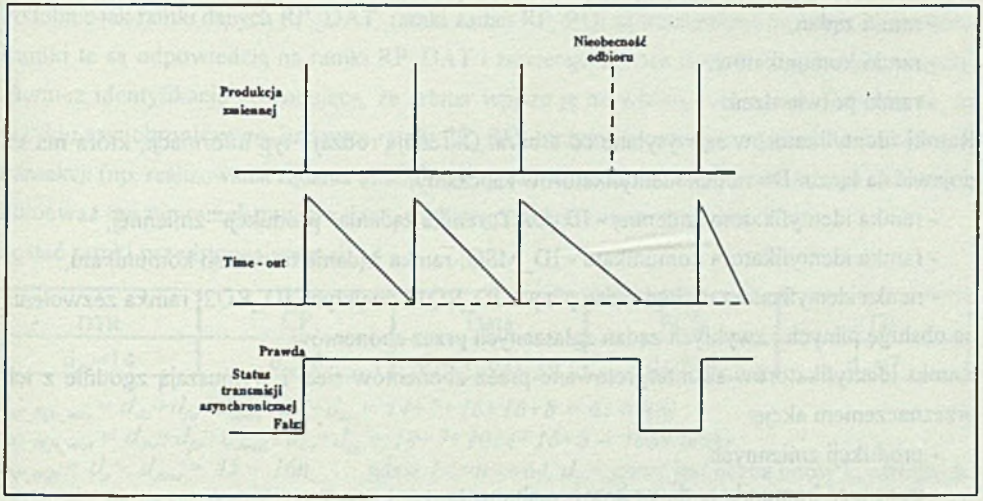
Statusy ważności produkcji (rys.10.) są wynikiem pracy warstwy aplikacji i mają następujące znaczenie:



Rys. 10. Potwierdzenie "produkcji" - statusy aktualizacji PRODUCENT - KONSUMENT
Fig. 10. Updating status PRODUCER - CONSUMENT

- stan statusu "PRAWDA" informuje konsumenta danych, że ich producent był "obserwowany" przez cały cykl produkcji danych. Innymi słowy, stan ten daje pewność konsumentowi, że dane są aktualne,
- stan statusu "FAŁSZ" oznacza, że dane nie zostały wyprodukowane w ostatnim cyklu (na skutek nieobecności producenta).

Statusy ważności transmisji (rys. 11) są również wynikiem pracy warstwy aplikacji i informują konsumenta zmiennych, że odpowiednie zmienne były wytransmitowane przez sieć wcześniej niż czas transmisji.



Rys. 11. Potwierdzenie jakości transmisji KONSUMENT-KONSUMENT
Fig. 11. Updating status - CONSUMENT - CONSUMENT

3. Analiza budowy ramek i czasów transmisji

W niniejszym rozdziale zostaną opisane budowy wszystkich typów ramek występujących w protokole FIP oraz podane ich długości w bitach, co będzie niezbędne przy określaniu czasów transmisji.

We wszystkich opisach ramek symbole i nazwy oznaczają:

- DTR - początek ramki,
- FTR - koniec ramki,
- FCS - kontrola poprawności ramki,
- CP - pole kontrolne ramki - kod operacji,
- ID - identyfikator zmiennej,
- Data - dane transmitowane od abonenta ("producenta" zmiennej),

- Destination - adres odbiorcy komunikatu,
- Source - adres nadawcy komunikatu
- Message - znakowa postać komunikatu.

Pod rysunkiem przedstawiającym budowę ramki umieszczono wyrażenie określające długość danej ramki liczoną w bitach.

Jak opisano w poprzednich rozdziałach, w sieci FIP można wyodrębnić następujące typy ramek:

- ramki identyfikatora,
- ramki danych,
- ramki żądań,
- ramki komunikatów,
- ramki potwierdzeń.

Ramki identyfikatorów są wysyłane od arbitra. Określają rodzaj i typ informacji, która ma się pojawić na łączu. Do ramek identyfikatorów zaliczamy:

- ramka identyfikatora zmiennej - ID_DAT; ramka żądania "produkcji" zmiennej,
- ramka identyfikatora komunikatu - ID_MSG; ramka żądania transmisji komunikatu,
- ramka identyfikatora obsługi żądań pilnych ID_RQ1 i zwykłych ID_RQ2; ramka zezwolenia na obsługę pilnych i zwykłych żądań zgłaszanych przez abonentów.

Ramki identyfikatorów są interpretowane przez abonentów sieci i wymuszają zgodnie z ich przeznaczeniem akcję:

- produkcji zmiennych,
- transmisji komunikatu do wybranego abonenta,
- przesłania przez abonenta do arbitra opisu żądania

Postać ramek wszystkich identyfikatorów jest identyczna. Przedstawia ją rysunek 12.

DTR	CP	ID	FCS	FTR
d_{dtr}=14	d_{cp}=8	d_{id}=16	d_{fcs}=16	d_{ftr}=7

$$d_{ID_DAT} = d_{ID_MSG} = d_{ID_RQi} = d_{dtr} + d_{ftr} + d_{id} + d_{fcs} + d_{cp} = 14+7+16+16+8=61 \text{ bitów} = const$$

Rys. 12. Budowa ramek identyfikatorów
Fig. 12. ID frames

Ramki danych RP_DAT są transmitowane przez producentów zmiennych (rys. 13). Ich długość jest zmienna. Generalnie w ramce występuje pole stałej długości d_c oraz pole zmiennej długości d_{data} zawierające dane "wyprodukowane".

DTR	CP	ID	Data	FCS	FTR
d_{dtr}=14	d_{cp}=8	d_{id}=16	d_d=(1..128)*8	d_{fcs}=16	d_{ftr}=7

$d_{RP_DAT_min} = d_{dtr} + d_{ftr} + d_{dmin} + d_{id} + d_{fcs} + d_{cp} = 14+7+8+16+16+8 = 69$ *bity*
$d_{RP_DAT_max} = d_{dtr} + d_{ftr} + d_{dmax} + d_{id} + d_{fcs} + d_{cp} = 14+7+1024+16+16+8 = 1085$ *bitów*
$d_{RP_DAT} = d_c + d_{data} = 61 + 8n$ gdzie $1<=n<=128$, $d_c = const$ jest liczbą bitów kontrolnych

Rys. 13. Budowa ramek danych

Fig. 13. Data frames

Podobnie jak ramki danych RP_DAT, ramki żądań RP_RQi są transmitowane przez abonentów. Ramki te są odpowiedzią na ramki RP_DAT i zawierają oprócz danych "wyprodukowanych" również identyfikację powodującą, że arbiter wpisze je na własną listę żądań. Oznacza to, że w cyklu asynchronicznym nadawca ramki RP_RQi otrzyma możliwość wykonania dodatkowej transakcji (np. realizowania żądania aktualizacji zmiennej produkowanej przez innego abonenta). Ponieważ ten typ ramek zawiera dane, to dlatego są one zmiennej długości.

Postać ramki przedstawia rysunek 14.

DTR	CP	Data	FCS	FTR
d_{dtr}=14	d_{cp}=8	d_d=(1..64)*16	d_{fcs}=16	d_{ftr}=7

$d_{RP_RQi_min} = d_{dtr} + d_{ftr} + d_{dmin} + d_{fcs} + d_{cp} = 14+7+16+16+8 = 61$ *bitów*
$d_{RP_RQi_max} = d_{dtr} + d_{ftr} + d_{dmax} + d_{fcs} + d_{cp} = 14+7+1024+16+8 = 1069$ *bitów*
$d_{RP_RQi} = d_c + d_{data} = 45 + 16n$ gdzie $1<=n<=64$, $d_c = const$ jest liczbą bitów kontrolnych

Rys. 14. Budowa ramek RP_RQi

Fig. 14. RP_RQi frames

Ramki komunikatów mają postać jak na rysunku 15.

DTR	CP	Destination	Source	Message	FCS	FTR
d_{dtr} = 14	d_{cp} = 8	d_{adr} = 24	d_{adr} = 24	d_d=(1..256)*8	d_{fcs}=16	d_{ftr}=7

$d_{RP_MSG_ACK_min} = d_{dtr} + d_{ftr} + 2d_{adr} + d_{dmin} + d_{fcs} + d_{cp} = 14+7+48+8+16+8 = 101$ *bitów*
$d_{RP_MSG_ACK_max} = d_{dtr} + d_{ftr} + 2d_{adr} + d_{dmax} + d_{fcs} + d_{cp} = 14+7+48+2048+16+8 = 2141$ *bitów*
$d_{RP_MSG_ACK} = d_{RP_MSG_NOACK} = d_{cm} + d_{data} = 93 + 8n$
gdzie $1<= n<=256$, $d_{cm} = const$ jest liczbą bitów kontrolnych

Rys. 15. Budowa ramek komunikatów

Fig. 15. Message frames

Ramki komunikatów służą do wymiany danych pomiędzy abonentami, przy czym może to odbywać się z żądaniem potwierdzenia odbioru lub bez (RP_MSG_ACK, RP_MSG_NOACK).Ostatnim typem ramek są ramki potwierdzeń, których budowę przedstawia rysunek 16.

DTR	CP	FCS	FTR
d_{dtr}=14	d_{cp}=8	d_{fcs}=16	d_{ftr}=7

$d_{RP_ACK} = d_{RP_FIN} = d_{dtr}+d_{ftr}+ d_{fcs}+d_{cp} = 14+7+16+8 = 45$ *bitów = const*

Rys. 16. Budowa ramek potwierdzeń

Fig. 16. Acknowledge frames

3.1. Analiza czasu trwania podstawowych wymian w sieci FIP

W niniejszym rozdziale podjęto próbę analizy wpływu podstawowych czynników na czas podstawowych wymian w sieci.

Czasy transmisji

Czasy transmisji znaków synchronizacyjnych jest stały dla każdej ramki i wynosi:

672 [μs] dla prędkości 32.25 [kbps]

21 [μs] dla prędkości 1.0 [Mbps]

8.4 [μs] dla prędkości 2.5 [Mbps]

Czasy transmisji ramek (znaki synchronizacyjne oraz dane) w zależności od ich długości (min./max.) oraz prędkości transmisji przedstawia poniższa tabela.

typ ramki	identyfikator	dane		żądania		komunikaty	
kod ramki	ID_DAT; ID_MSG; ID_RQi	RP_DAT		RP_RQ1; RP_RQ2		RP_MSG_NOACK; RP_MSG_ACK	
P_{tr} [kbps]	61 bitów	69 bitów	1085 bitów	61 bitów	1069 bitów	101 bitów	2141 bitów
31,25	1,95 [ms]	2,20 [ms]	34,72 [ms]	1,95 [ms]	34,21 [ms]	3,23 [ms]	68,51 [ms]
1000	61,00 [ms]	69,00 [μs]	1,08 [ms]	61,00 [μs]	1,06 [ms]	0,10 [ms]	2,14 [ms]
2500	24,40 [ms]	27,60 [μs]	0,43 [ms]	24,40 [μs]	0,42 [ms]	40,40 [μs]	0,85 [ms]

Dla ramek potwierdzeń czasy te wynoszą:

typ ramki	potwierdzenia
kod ramki	RP_ACK; RP_FIN
P_{tr} [kbps]	45 bitów
31.25	1.440 [ms]
1000	45.000 [μs]
2500	18.000 [μs]

Pozostałe parametry czasowe

Na czas trwania wymian mają wpływ parametry globalne wykorzystywane przez poszczególne warstwy sieciowe. Należą do nich:

Czas oczekiwania (milczenia sieci ang. *link network silence time–out*), popularnie zwany *timeout*, jest to maksymalny czas oczekiwania arbitra na odpowiedź abonenta.

$T_{tout} = const$

Czas odpowiedzi (ang. *turnaround network silence time–out*), jest to minimalny czas ciszy na sieci pomiędzy transmisją kolejnych ramek.

$T_{tr} = const$

przy czym:

$$10\ T_{mac} <= T_{tr} <= 70\ T_{mac}$$

gdzie T_{mac} jest czasem potrzebnym na transmisję jednego z symboli synchronizacyjnych warstwy fizycznej, czyli *de facto* czasem potrzebnym na wytransmitowanie jednego bitu.

Poniżej przedstawiono wartości czasu transmisji jednego bitu T_{mac}, w zależności od prędkości transmisji P_{tr}.

dla P_{tr} = 31.25 [kbps] $T_{mac} = 1/P_{tr}$ = 32.0 [μs]
dla P_{tr} = 1.0 [Mbps] $T_{mac} = 1/P_{tr}$ = 1.0 [μs]
dla P_{tr} = 2.5 [Mbps] $T_{mac} = 1/P_{tr}$ = 0.4 [μs]

Poniżej przedstawiono przedziały wartości czasu odpowiedzi T_{tr}, w zależności od prędkości transmisji P_{tr}.

dla P_{tr} = 31.25 [kbps] 320.0 [μs] <= T_{tr} <= 2.24 [μs]
dla P_{tr} = 1.0 [Mbps] 10.0 [μs] <= T_{tr} <= 70 [μs]
dla P_{tr} = 2.5 [Mbps] 4.0 [μs] <= T_{tr} <= 28 [μs]

Maksymalna liczba wymian

Z powyższych danych wynikają ograniczenia na liczbę wymian cyklicznych w sieci. Oczywiście teoretycznie liczba wymian ograniczona jest maksymalną ilością identyfikatorów. Należy jednak zauważyć, iż wzrost liczby wymian cyklicznych prowadzi do konieczności wydłużania okna periodycznego. Zatem, o ile chcemy zachować stały maksymalny okres trwania obsługi aperiodycznej i synchronizacji, konsekwencje naszych poczynań będą dwie. Pierwsza to wydłużenie cyklu elementarnego, druga wynikająca z poprzedniej to wydłużenie okresu odświeżania zmiennych.

Zakładając wartość czasu odświeżania na poziomie 0.25 sekundy (250ms), prędkość transmisji równą 1Mbps, oraz fakt, iż mamy do czynienia tylko z wymianami cyklicznymi, z czasem oczekiwania (timeout) 240μs, otrzymujemy:

$$L_{exch} = 0.25 / (T_{ID_DAT} + 2T_{tr_min} + T_{tout} + T_{RP_DAT})$$

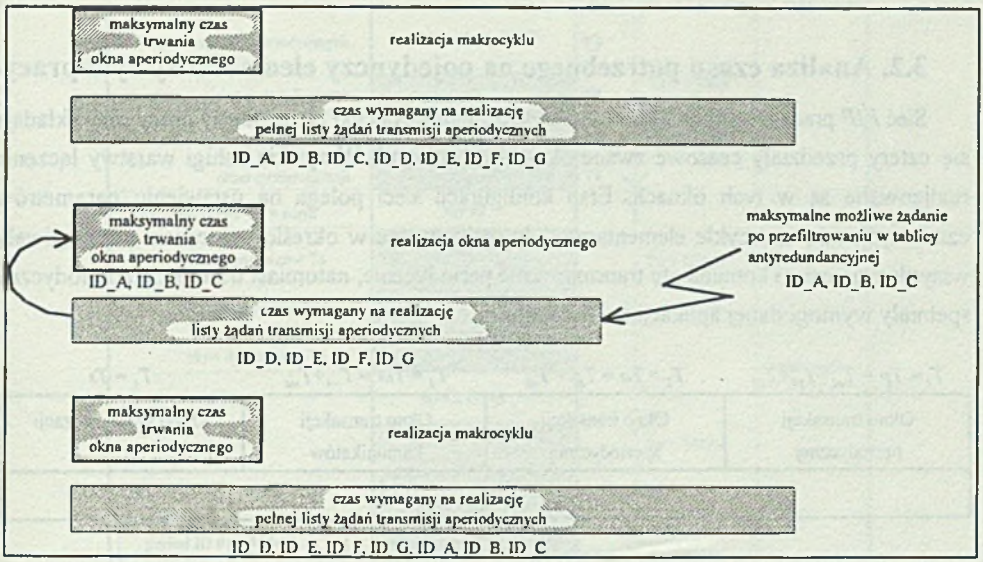
$$L_{exch} = 0.25 / (61*10^{-6} + 20*10^{-6} + 240*10^{-6} + 1.085*10^{-3})$$

$$L_{exch} = 177.8$$

Zmniejszanie wartości czasu odpowiedzi (T_{tout}) do zera daje graniczną wartość liczby wymian dla prędkości transmisji 1Mbps, równą 214. Dla tej liczby wymian i powyższych przykładowych parametrów niemożliwe jest realizowanie wymian z periodem 0.25 sekundy dla identyfikatorów z listy periodycznej.

Jeżeli chodzi o wymiany aperiodyczne, sprawa wygląda nieco inaczej. Teoretyczne, ograniczenie liczby wymian aperiodycznych w jednym cyklu elementarnym wynika tak jak poprzednio z maksymalnej ilości identyfikatorów. Oczywiście zakładając dowolnie długi okres trwania obsługi. Istnieje jednak silniejsze ograniczenie wynikające z ilości abonentów. Jeden abonent nie może zgłosić w jednym cyklu elementarnym więcej jak 64 żądania transmisji aperiodycznej. Zatem w przypadku gdy mamy np. dwóch abonentów i każdy z nich żąda maksymalnej ilości uaktualnień aperiodycznych, otrzymujemy maksymalną liczbę zgłoszeń w cyklu równą 128. Analizując przypadek skrajny, dla 256 abonentów i maksymalnej liczby różnych żądań, ilość zgłoszeń wynosi 16384.

Do tej pory nie zwracaliśmy uwagi na czas trwania okna obsługi aperiodycznej. Czas ten jest jednak ograniczony przez wymogi związane z czasem trwania makrocyklu elementarnego. Może zaistnieć sytuacja, iż część lub całość zgłoszonych żądań nie będzie obsłużona w najbliższym oknie aperiodycznym. Ponieważ jednak arbiter zapamiętuje zgłoszenia na dynamicznej liście, nie ma obawy o zagubienie zgłoszenia. Żądania nieobsłużone zostaną obsłużone w następnym oknie aperiodycznym. Wspomniana lista jest więc swoistą dynamiczną kolejką typu *FIFO*. Pojawia się jednak pytanie, co będzie, gdy w każdym makrocyklu wystąpi liczba żądań przekraczająca możliwości realizacji w najbliższym oknie czasowym transmisji aperiodycznych. Otóż w związku z faktem istnienia mechanizmu antyredundancyjnego realizowanego przez arbitra magistrali, niemożliwe jest, aby ilość żądań, zapamiętana na liście, przekroczyła maksymalną liczbę identyfikatorów obsługiwanych aperiodycznie. Inaczej ujmując, po zapełnieniu listy żądaniami uaktualnień wszystkich zmiennych obsługiwanych aperiodycznie, niemożliwe jest, aby zostało dopisane do niej więcej żądań, niż zostało obsłużonych w ostatnim oknie aperiodycznym. Opisany problem zilustrowany jest na poniższym rysunku.



Rys. 17. Schemat realizacji żądań
Fig. 17. The chart of request realisation

Użytkownik konfigurujący sieć powinien zatem sprawdzić, jak duża może być lista zgłoszeń obsługi aperiodycznej, a co za tym idzie - jak będzie kształtował się pesymistyczny czas aperiodycznego odświeżania, oraz problem podstawowy, czy pamięć arbitra jest dostatecznie duża do alokacji tej listy.

Wpływ liczby abonentów na liczbę wymian

Maksymalna liczba wymian cyklicznych przedstawiona powyżej nie ma związku z ilością abonentów na sieci. Taka ilość wymian może przebiegać zarówno dla jednego, jak i szeregu abonentów. Zależy to od danej aplikacji. Zatem liczba abonentów nie wpływa na obciążenie sieci i nie ma bezpośredniego związku z ilością wymian.

Wyjątek stanowi zwiększenie liczby abonentów obsługujących nowe zmienne lub komunikaty w transakcjach cyklicznych. Wzrost liczby transakcji cyklicznych musi pociągnąć za sobą wydłużenie okna periodycznego, co może kolidować z czasem trwania okien pozostałych, o ile czas trwania całego makrocyklu chcemy pozostawić bez zmian.

Przy założeniu że mamy do czynienia ze stałą liczbą (listą) identyfikatorów obsługiwanych periodycznie, można stwierdzić, iż zwiększenie bądź zmniejszenie liczby abonentów nie wpływa na obciążenie sieci.

3.2. Analiza czasu potrzebnego na pojedynczy elementarny cykl pracy

Sieć *FIP* pracuje wykonując tzw. cykle elementarne. Na cykl elementarny pracy sieci składają się cztery przedziały czasowe zwane oknami czasowymi. Wszystkie usługi warstwy łączenia realizowane są w tych oknach. Etap konfiguracji sieci polega na ustawieniu parametrów czasowych tak, aby cykle elementarne były realizowalne w określonym czasie i obsługiwały wszystkie zmienne i komunikaty transmitowane periodycznie, natomiast transakcje aperiodyczne spełniały wymogi danej aplikacji i układu arbitra.

$T_1 = Tp = T_{p1}+T_{p2}+T_{p3}$	$T_2 = Ta = T_{a1} + T_{a2}$	$T_3 = Tm = T_{m1}+T_{m2}$	$T_4 = Ts$
Okno transakcji periodycznej	Okno transakcji aperiodycznej	Okno transakcji komunikatów	Okno synchronizacji
T_{cycle}			

Tp, Ta, Tm, Ts — czas trwania transakcji periodycznych, aperiodycznych, komunikatów, synchronizacji

T_{p1} — czas trwania transakcji periodycznej zmiennej

T_{p2} — czas trwania transakcji periodycznej komunikatu bez potwierdzenia

T_{p3} — czas trwania transakcji periodycznej z potwierdzeniem

T_{a1} — czas trwania transakcji aperiodycznej pilnej

T_{a2} — czas trwania transakcji aperiodycznej normalnej

T_{m1} — czas trwania transakcji komunikatu bez potwierdzenia

T_{m2} — czas trwania transakcji z potwierdzeniem

Czas potrzebny na zrealizowanie pełnego cyklu wynosi:

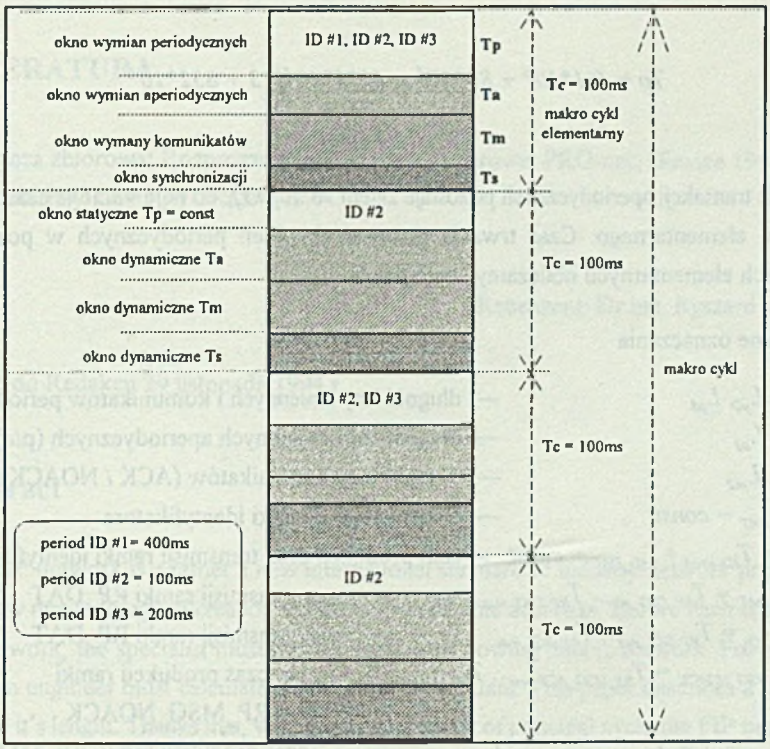
$$T_{cycle} = \sum T_i \quad i=1..4$$

Parametr T_{cycle} jest definiowany przez użytkownika i zależy od wymogów danej aplikacji.

Przykład

Na poniższym rysunku przedstawiono schemat makrocyklu transmisji we wszystkich czterech oknach czasowych cyklu elementarnego, dla trzech zmiennych transmitowanych periodycznie z różnym periodem.

Czas trwania okna wymian periodycznych jest stały w każdym cyklu elementarnym, lecz różny w poszczególnych cyklach należących do całości makrocyklu. Czas trwania okien dynamicznych może być zmienny w każdym cyklu elementarnym, jednak nie może przekroczyć zdefiniowanej wartości maksymalnej.



Rys. 18. Schemat elementarnego cyklu pracy sieci
Fig. 18. The chart of elementary cycle

Czas trwania okien periodycznych jest prosty do wyliczenia, wynosi on:

$$Tp = (T_{ID_DAT} + 2T_{TR} + T_{RP_DAT})Lp$$

$$Tp = (T_{ID_DAT} + 2T_{TR} + d_c * T_{moc})Lp + T_{data}$$

Założenia:

ID #1 jest tablicą 16 bajtów;

ID #2 jest liczbą zmiennoprzecinkową zakodowaną na 24 bajtach

ID #3 jest łańcuchem 64 znaków jednobajtowych

$T_{TR} = 40$ [μs]

dla okna periodycznego z pierwszego cyklu elementarnego parametr T_{data} (czas transmisji danych) wynosi

$$T_{data} = (16*8 + 24*8 + 64*8)T_{moc} = 832*1\ [\mu s] = 832\ [\mu s]$$

stąd dalej

$$Tp = (61*10^{-6} + 80*10^{-6} + 61*1*10^{-6})*3 + 832*10^{-6}$$

$$Tp = 1.44\ [ms]$$

Na realizację transakcji aperiodycznych pozostaje zatem *98.56[ms]*, co daje *98.56%* czasu trwania makrocyklu elementarnego. Czas trwania pozostałych okien periodycznych w pozostałych makrocyklach elementarnych obliczamy analogicznie.

Wykorzystane oznaczenia

L_{p1}, L_{p2}, L_{p3} — długość listy zmiennych i komunikatów periodycznych
L_{a1}, L_{a2} — długość listy zmiennych aperiodycznych (p/n)
L_{m1}, L_{m2} — długość listy komunikatów (ACK / NOACK)
$T_{ID_DAT} = const$ — czas transmisji ramki identyfikatora
$T_{ID} = T_{ID_prod} + T_{ID_DAT} = const$ — czas produkcji oraz transmisji ramki identyfikatora
$T_{RP_DAT} = T_{RP_DAT_min} .. T_{RP_DAT_max}$ — czas transmisji ramki RP_DAT
$T_{RP_RQi} = T_{RP_RQi_min} .. T_{RP_RQi_max}$ — czas transmisji ramki RP_DAT
$T_{RP_MSG_NOACK} = T_{RP_MSG_ACK_min} .. T_{RP_MSG_ACK_max}$ — czas produkcji ramki RP_MSG_NOACK
$T_{RP_MSG_ACK} = T_{RP_MSG_ACK_min} .. T_{RP_MSG_ACK_max}$ — czas produkcji ramki RP_MSG_ACK
$T_{RP_ACK} = const$ — czas produkcji ramki RP_ACK
$T_{RP_FIN} = const$ — czas produkcji ramki RP_FIN

4. Wnioski końcowe

Sieć FIP stała się już standardem. Ale ciągle są prowadzone badania dotyczące szeregu elementów, w tym również zakresu stosowalności, co bardzo silnie jest związane z analizą czasową przepływu informacji. Niniejsze opracowanie jest jednym z trzech kolejnych planowanych do wydania materiałów, które będą w całości zawierać analizy przepływu danych i jak sądzimy, pozwolą stworzyć model matematyczny sieci FIP. Powstanie takiego modelu umożliwi opracowanie programu symulacji pracy sieci FIP, a tym samym otworzy drogę do bezpiecznego projektowania przemysłowych systemów sieciowych i tworzenia nadrzędnych stacji wizualizacyjnych i monitorujących. Wyniki, prowadzonych w tym kierunku badań, pozwolą na szybką implementację w polskim przemyśle nie tylko systemów sieciowych, ale również dadzą szansę na produkcję kart sieciowych dla urządzeń wykonawczych produkowanych przez polski przemysł.

LITERATURA

[1] Praca zbiorowa: Rozproszone systemy komputerowe. PRO-net, Gliwice 1994.
[2] Materiały klubu „WORLD-FIP"

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 29 listopada 1994 r.

Abstract

The FIP network is a rather a new international standard of industry network protocol. The FIP is a new methodology of creation of distribiuted real time data base. Before each configuration of FIP network, the specialist must made a analysis of flowing data in network. For solving this problem an engineer must calculate a time of exchange data. This paper describes a structure of frames and it's lenght. Thanks that, we can calculate a time of principal cycle the FIP network, and we can also show the limits and area of aplication the FIP network.