ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1999
Seria: INFORMATYKA z. 36 Nr kol. 1414

Piotr GAJ
Politechnika Śląska, Instytut Informatyki

SZYBKA SIEĆ PRZEMYSŁOWA A SYSTEM WIZUALIZACJI – PROBLEM INTERFEJSU

Streszczenie. Artykuł przedstawia zagadnienia związane z połączeniem szybkiej komputerowej sieci przemysłowej z systemami wizualizacyjnymi. Główny nacisk położony jest na sposoby przekazywania szybkozmiennej informacji dla użytkownika systemu. W artykule zawarte są odniesienia do rzeczywistych rozwiązań.

HIGH SPEED INDUSTRIAL NETWORK vs. SCADA SYSTEM – PROBLEM OF THE INTERFACE

Summary. This article presents the issues coupled with connections between high sped computer industrial network and industrial visualisation system (SCADA). Main topic of this article is passing an information to system user. The real treatment links are including in this article.

1. Wstęp

Niniejsza praca traktuje o laboratoryjnych próbach praktycznej implementacji połączenia szybkiej sieci przemysłowej z systemem wizualizacji i archiwizacji zdarzeń pracującym pod kontrolą popularnego obecnie systemu operacyjnego MS Windows NT 4.0 Workstation. Całość rozważań dotyczy systemów informatycznych opartych na sieciowych systemach rozproszonych [1]. Za przykład sieci przyjęto sieć polową (*fieldbus*) typu FIP zapewniającą możliwość transmisji zdarzeń z szybkością wielokrotnie przekraczającą możliwości percepcji użytkownika oraz mogącą przysporzyć kłopotów wieloprocesowemu systemowi operacyjnemu, jakim jest MS Windows NT. W niniejszej pracy nie przedstawiono dokładnego opisu protokołu FIP, gdyż można go znaleźć między innymi w [7], [8], [9], [10]. Jako przykład systemu

wizualizacyjnego przyjęto system KRONOS. Jest to produkt krajowy dedykowany do pracy pod kontrolą systemu operacyjnego MS Windows NT, zatem wykorzystuje wszystkie mechanizmy, jakie są przez ten system oferowane. Opis i strukturę systemu KRONOS można znaleźć w [19].

2. Zagadnienie przekazywania informacji

Na wstępie przyjmijmy kilka założeń. Struktura rozpatrywanego systemu informatycznego ma postać wielowarstwową. Zobrazowane jest to na rys. 1. Podstawowy podział systemu wynika ze struktury rozproszonej obsługi procesu. Można zatem wyróżnić:

1) poziom procesu,
2) poziom transmisji,
3) poziom wizualizacji (SCADA).

Przepływ informacji, która ma zostać zwizualizowaną odbywa się od poziomu procesu przez poziom transmisji do poziomu SCADA.

Poziom procesu jest abstrakcyjnym zbiorem zawierającym wszystkich abonentów generujących zdarzenia (informacje) podlegające wizualizacji. Analiza tego poziomu z punktu widzenia tematu pracy jest niecelowa. Istotny jest jedynie strumień zdarzeń rozumiany jako rozkład występowania zdarzeń generowanych przez zbiór abonentów sieci w czasie. Inaczej mówiąc, poziom procesu stanowi generator zdarzeń o określonym rozkładzie.

Poziom transmisji jest to abstrakcyjna sieć umożliwiająca transmisję zdarzeń. Z punktu widzenia tematu pracy nieistotne są architektura, topologia, model czy też wykorzystywany protokół. Ważny jest natomiast fakt, aby poziom ten umożliwiał przetransmitowanie strumienia danych generowanych przez poziom procesu w tzw. czasie rzeczywistym oraz aby poziom ten nie zmieniał rozkładu strumienia zdarzeń. Pojęcie czasu rzeczywistego jest dość niejednoznaczne. Dlatego dla potrzeb dalszych odwołań zdefiniujmy pojęcie propagacji zdarzenia w czasie rzeczywistym jako działanie w czasie nie większym od przyjętej stałej T_R. Czyli:

$$T_A - T_G \leq T_R$$

gdzie:

T_A – czas dotarcia zdarzenia do miejsca docelowego,

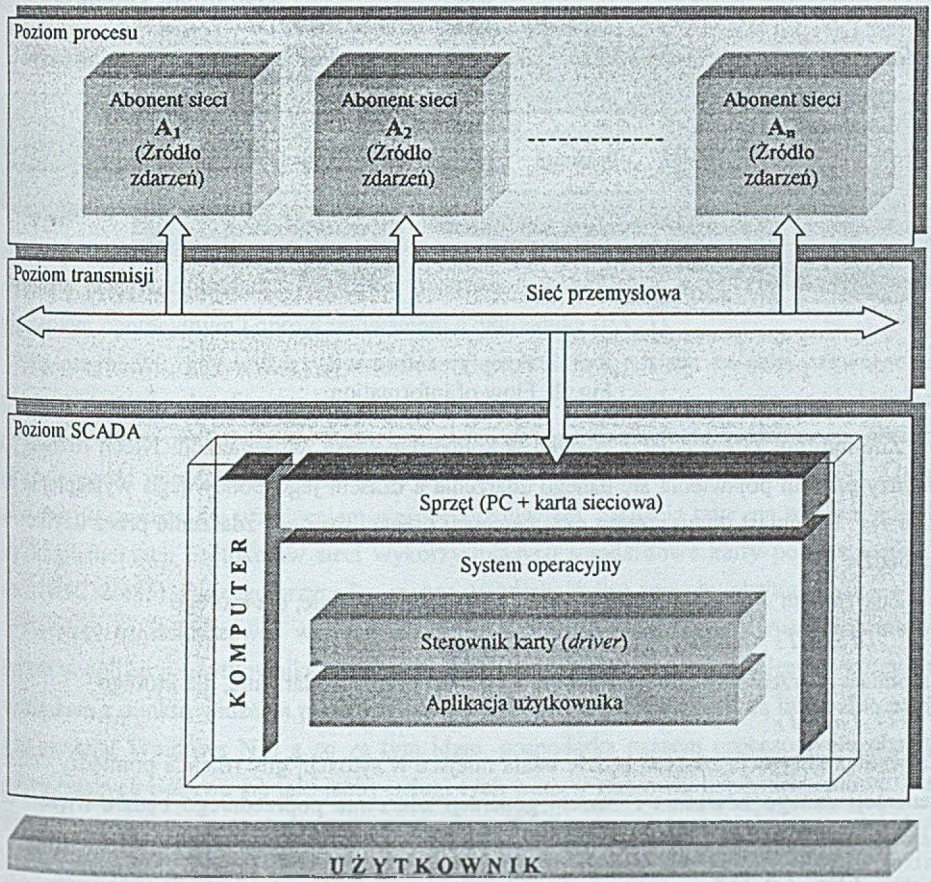
T_G – czas generacji zdarzenia przez źródło,

T_R – maksymalna wartość czasu propagacji zdarzenia ze źródła do celu.

Przypadek systemu, w którym poziom procesu generuje zdarzenia z szybkością przekraczającą możliwości ich propagacji przez poziom transmisji w czasie rzeczywistym, nie będzie dalej rozpatrywany.

Przyjmując powyższe założenia oraz dla uproszczenia problemu, można scalić poziom procesu i poziom transmisji traktując je jako generator zdarzeń o rozkładzie takim samym jak rozkład strumienia z poziomu procesu.

Najistotniejszym elementem systemu z punktu widzenia tematu pracy jest poziom wizualizacji.

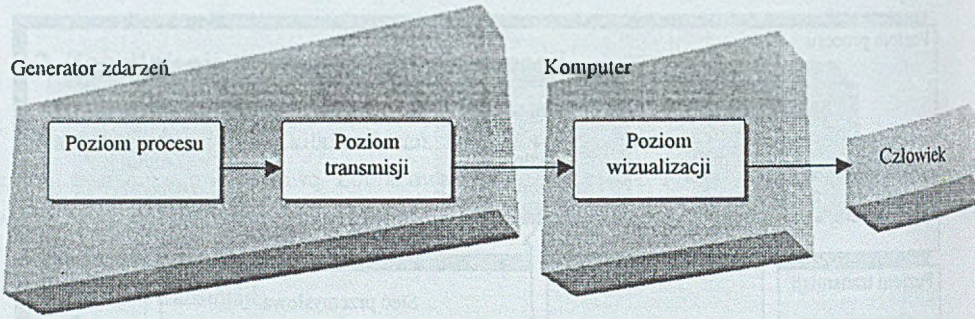


Rys. 1. Wielowarstwowa struktura przepływu informacji
Fig. 1. Multilayer structure of information flow

Po uproszczeniu mamy układ składający się z generatora strumienia zdarzeń podłączonego do urządzenia wizualizującego te zdarzenia, jakim jest komputer. Należy zwrócić uwagę, że komputer jest tutaj tylko pośrednikiem pomiędzy generatorem a użytkownikiem, dla którego prowadzona jest wizualizacja. Odbiorcą docelowym jest więc człowiek.

2.1. Charakterystyka zdarzeń

Przesłanie informacji wg schematu z rys. 2 nie stwarza problemów dla przypadków generatora zdarzeń wolnozmiennych. Problemy pojawiają się, gdy generowane są zdarzenia szybkozmienne.



Rys. 2. Przepływ informacji
Fig. 2. Flow of information

Zdefiniujmy zatem pojęcie zdarzeń w kontekście czasu ich generacji. Niech różnica pomiędzy czasem pojawienia się danego zdarzenia a czasem jego ponownego wystąpienia będzie T_{GZ}, natomiast dopuszczalny czas percepcji i reakcji na dane zdarzenie przez użytkownika będzie T_P.

Zdarzeniem wolnozmiennym będziemy określać zdarzenie, dla którego

$$T_{GZ} \geq T_P,$$

natomiast zdarzeniem szybkozmiennym będziemy określać zdarzenie, dla którego

$$T_{GZ} < T_P.$$

Lawinowa generacja zdarzeń będzie miała miejsce w sytuacji, gdy różnica pomiędzy czasem generacji danego zdarzenia i czasem generacji zdarzenia poprzedniego będzie większa od czasu T_P, czyli:

$$T_{Ei-1} - T_{Ei} < T_P.$$

Ponieważ definicja szybkozmienności zdarzeń odnosi się do charakteru występowania konkretnego zdarzenia E_i, a definicja lawinowości odnosi się do charakteru występowania grupy zdarzeń $E_k \ldots E_p$, dla uproszczenia przyjmijmy określać wszystkie te zdarzenia jako zdarzenia szybkozmienne.

Należy tutaj zwrócić uwagę, iż zdarzeniem może być otrzymanie nowych danych, np. wartości zmiennej procesowej. Niezależnie od różnicy aktualnych danych względem danych poprzednich pojawienie się ich stanowi dla obsługi kolejne zdarzenie. Innymi słowy, jeśli na poziomie procesu wykonywany jest pomiar temperatury, a sieć dostarcza jej aktualną wartość,

to niezależnie czy ta wartość uległa zmianie od poprzedniej wymiany, czy nie, mamy do czynienia ze zdarzeniem wymagającym obsługi.

2.2. Zagadnienie rozdziału warstw sieciowych pomiędzy poziomy sprzętowo–programowe komputera

Ponieważ cały czas mamy do czynienia z działaniem w czasie rzeczywistym, należy przyjrzeć się zagadnieniu przekazywania danych od generatora do warstwy aplikacji użytkownika. Jest to kanał, który może wprowadzać istotne opóźnienia a co za tym idzie zmuszać do wydłużania czasu T_R.

Sposób rozdziału warstw sieci modelu ISO / OSI pomiędzy struktury poziomu wizualizacji jest istotnym zagadnieniem doboru, jak i tworzenia systemu wizualizacyjnego. Na wstępie warto się przyjrzeć dokładniej samemu poziomowi wizualizacji, czyli komputerowi wraz z systemem operacyjnym i oprogramowaniem użytkownika (rys. 3).

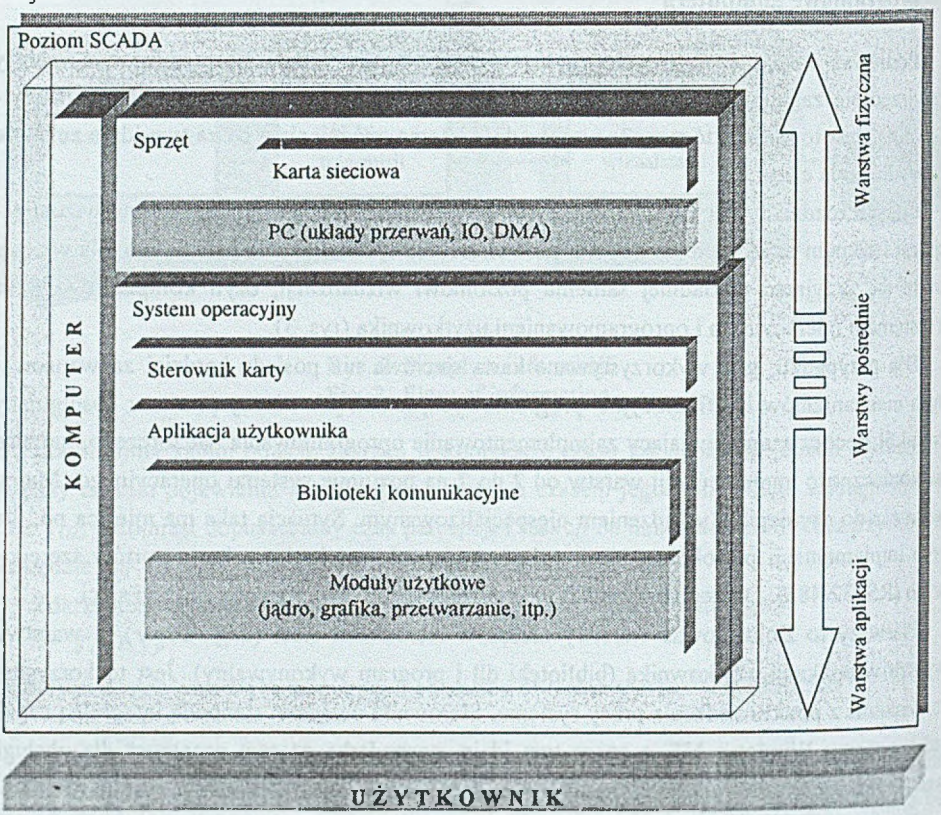
Dla przypadku, gdy wykorzystywana karta sieciowa nie posiada bardziej zaawansowanych mechanizmów konfiguracyjno–programowych, takich jak własny procesor obsługujący protokół sieci czy umożliwiający zaimplementowanie oprogramowania nadzorczego, narzuca się konieczność implementacji warstw od 2 do 7 na poziomie systemu operacyjnego. Mamy wówczas do czynienia z urządzeniem niespecjalizowanym. Sytuacja taka ma miejsca np. dla prób implementacji protokołów sieci wykorzystujących standardowe karty portów szeregowych (RS232/485). Dla tego przypadku można wyodrębnić dwa sposoby obsługi sieci.

Pierwszy to zlokalizować warstwy niższe w sterowniku sieci (ang. *driver*), a warstwy wyższe w aplikacji użytkownika (biblioteki dll i program wykonywalny). Jest to korzystne rozwiązanie z punktu widzenia pracy systemu. Sterowniki urządzeń stanowią integralną część jądra systemu Windows NT, a co za tym idzie, gospodarka czasem procesora dla obsługi warstw niższych odbywa się tak, jakby usługi tych warstw stanowiły usługi systemowe. Natomiast warstwy wyższe obsługiwane z poziomu aplikacyjnego nie wymagają pracy w dużym reżimie czasowym. Innymi słowy, przy takim rozdziale warstw czas rzeczywisty, w jakim powinna pracować obsługa sieci, został podzielony na dwie części: część systemową dla usług wymagających mniejszego T_R, oraz część aplikacyjną dla T_R o wartościach mniej krytycznych.

Drugi sposób to możliwość skorzystania ze standardowych sterowników urządzeń. Spowoduje to konieczność implementacji prawie wszystkich warstw sieciowych[1] na poziomie aplikacyjnym. Jest to rozwiązanie ze wszech miar niekorzystne. Praca wszystkich warstw sieciowych jest silnie uzależniona od obciążenia warstwy aplikacyjnej systemu operacyjnego. Pewnym rozwiązaniem problemów z tego wynikających jest wykorzystanie systemowych

[1] Oprócz warstwy fizycznej obsługiwanej przez kartę.

mechanizmów pracy wielowątkowej. Jest to jednak rozwiązanie niepewne, gdyż dowolna aplikacja z odpowiednio dużym priorytetem swoich wątków może spowodować nieokreślone w czasie wstrzymywanie przetwarzania w warstwach sieciowych. Jest chyba oczywiste, jak duże znaczenie ma taka sytuacja dla systemów przemysłowych pracujących z T_R stosunkowo małym.



Rys. 3. Rozmieszczenie warstw sieciowych
Fig. 3. Localization of network layers

Generalnie przypadek obsługi niespecjalizowanych kart do obsługi protokołów sieci przemysłowych jest z punktu widzenia stacji opartej o system Windows NT nienajlepszy. Wynika to z konieczności implementacji całości obsługi na poziomie systemu operacyjnego i braku wspomagania sprzętowego.

Znacznie lepszym rozwiązaniem jest skorzystanie z kart specjalizowanych. Urządzenie takie przejmuje wówczas znaczną część obsługi związanej z usługami wszystkich warstw sieciowych. Dla sieci przemysłowych można przyjąć za istotne rozpatrywanie tylko trzech warstw sieciowych modelu ISO/OSI, a mianowicie warstw:

- fizycznej (*layer 1*),
- łączenia (*layer 2*),
- aplikacji (*layer 7*).

Najlepszym rozwiązaniem jest karta posiadająca własny procesor komunikacyjny obsługujący protokół sieci na poziomie niezbędnych warstw 1, 2 i przynajmniej częściowo 7. Rozwiązanie takie jest wręcz idealne dla obsługi sieci w czasie rzeczywistym, gdyż karta musi korzystać z jednostki centralnej tylko do swojej konfiguracji. Cała obsługa stacji abonenckiej odbywa się niezależnie od pracy systemu operacyjnego i jego obciążenia.

Dla analizowanego przykładu sieci typu FIP zastosowano kartę ZEG-121 umożliwiającą pracę w sieci o protokole WorldFIP. Karta obsługuje warstwę fizyczną, łączenia oraz częściowo warstwę aplikacji (warstwa komunikacyjna aplikacji). Sterownik karty oraz aplikacja użytkownika przejmuje funkcje tylko warstwy 7. Do najważniejszych z nich należą:

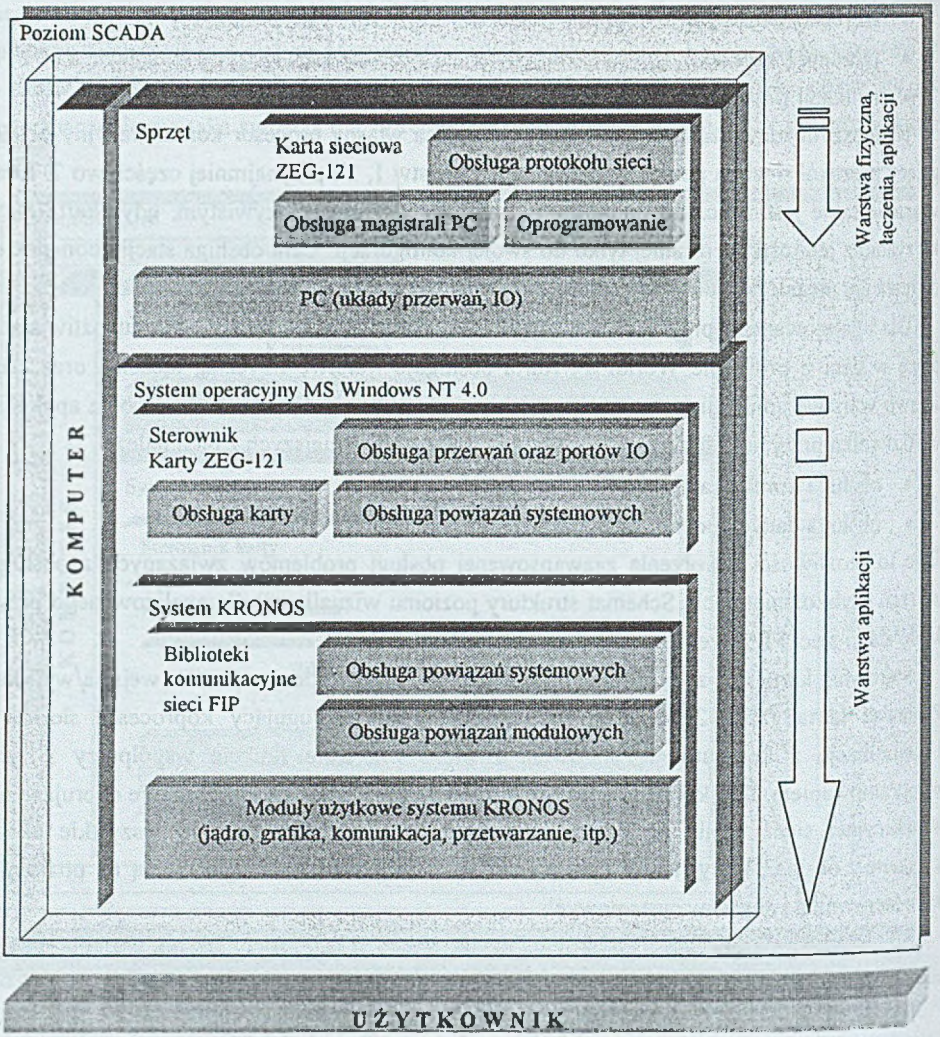
- obsługa konfiguracji karty,
- obsługa danych pochodzących z sieci.

Daje to możliwości stworzenia zaawansowanej obsługi problemów związanych z obsługą zdarzeń szybkozmiennych. Schemat struktury poziomu wizualizacji dla analizowanego przykładu stacji sieci FIP przedstawiony jest na rysunku 4.

Sterownik karty realizuje obsługę przerwań oraz zapisy i odczyty portów wejścia/wyjścia. Ponieważ karta ZEG-121 posiada specjalny program obsługujący koprocesor sieciowy i komunikację z komputerem, sterownik zawiera wszystkie funkcje współpracy z tym oprogramowaniem. Dzięki temu na zewnątrz sterownika udostępniane są funkcje operujące na abstrakcyjnej sieci, a nie na fizycznych rejestrach karty. Oprócz tego wszystkie akcje związane z obsługą karty, mające się wykonywać cyklicznie, wykonywane są na poziomie kodu sterownika i zegarów systemowych.

Biblioteki komunikacyjne po stronie użytkownika służą już tylko do scalenia usług udostępnianych przez sterownik z wewnętrzną strukturą systemu wizualizacyjnego.

Powyższe rozłożenie funkcji interfejsu komunikacyjnego sieci gwarantuje optymalne rozłożenie zadań stawianych przed tymi funkcjami, szczególnie w aspekcie czasu. Oprogramowanie i sprzęt stanowiący zasoby karty sieciowej mogą realizować najbardziej krytyczne czasowo funkcje obsługi protokołu. Oprogramowanie sterownika sieci obsługuje komunikację z kartą oraz może stanowić szybki bufor dla obsługi danych. Aplikacji użytkownika pozostaje już tylko korzystać z informacji i wydawać polecenia warstwom niższym.



Rys. 4. Rozmieszczenie warstw sieciowych dla kart specjalizowanych
Fig. 4. Localization of network layers for specialized board

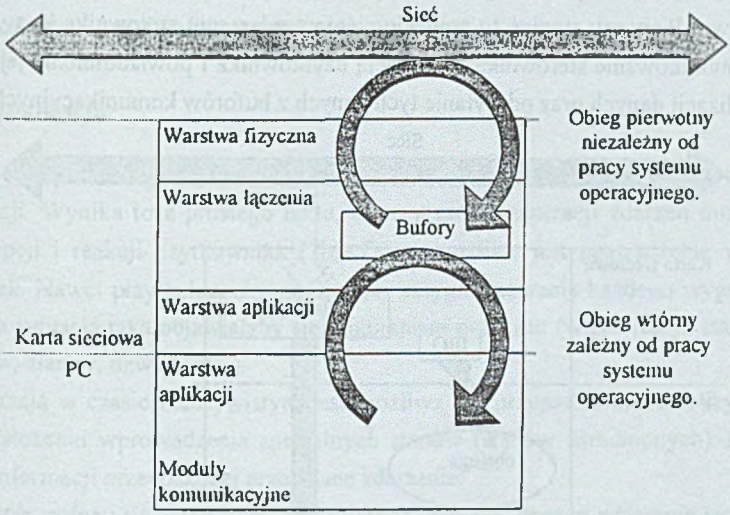
3. Problem interfejsu

Ze względu na rodzaj reakcji użytkownika na wystąpienie zdarzenia problem obsługi zdarzeń szybkozmiennych przez systemy SCADA należy rozpatrywać przynajmniej dla dwóch przypadków. Pierwszy to bezpośrednia wizualizacja zdarzenia, czyli niezwłoczne (w czasie rzeczywistym) przekazanie informacji o wystąpieniu zdarzenia dla użytkownika. Drugi to

przypadek rejestracji i archiwizacji zdarzeń. Jednakże przed przystąpieniem do analizy tych przypadków należy się zastanowić nad potencjalnymi możliwościami przetwarzania informacji tego fragmentu warstwy aplikacji, który znajduje się pod kontrolą systemu operacyjnego.

3.1. Przepustowość warstwy aplikacji

Dla omawianego w poprzednim rozdziale przypadku najkorzystniejszego rozmieszczenia poszczególnych warstw sieciowych na poziomie wizualizacji znaczna część przetwarzania musi się znaleźć po stronie nieautonomicznej, czyli po stronie systemu operacyjnego. Przyjrzyjmy się tylko jednej grupie funkcji, a mianowicie odczytowi buforów komunikacyjnych warstw niższych.



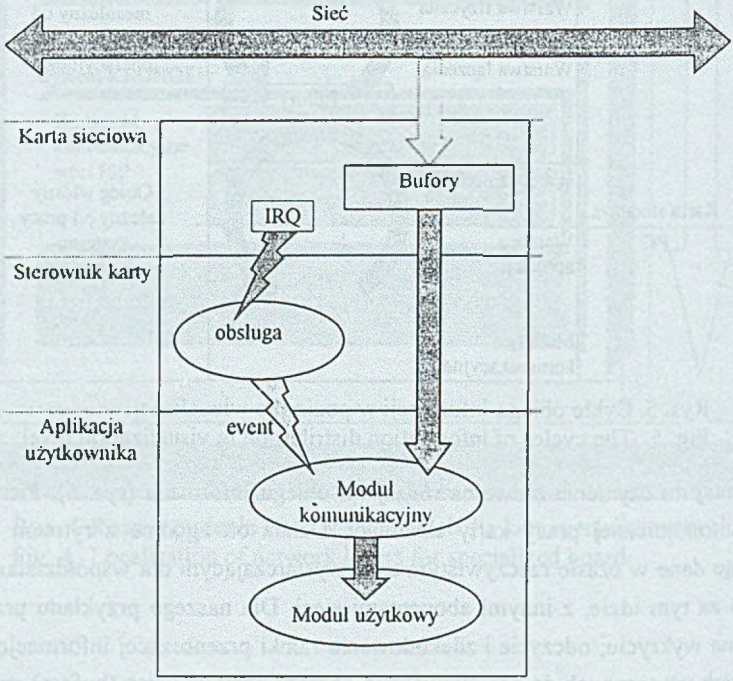
Rys. 5. Cykle obiegu informacji w poziomie wizualizacji
Fig. 5. The cycles of information distribution in visualization level

Mamy tutaj do czynienia z dwoma rodzajami obiegu informacji (rys. 5). Pierwszy obieg wynika z autonomicznej pracy karty sieciowej. Działa on zgodnie z rytmem pracy sieci przetwarzając dane w czasie rzeczywistym z T_R wystarczającym dla współdziałania z protokołem, a co za tym idzie, z innymi abonentami sieci. Dla naszego przykładu przetwarzanie polegałoby na wykryciu, odczycie i zdekodowaniu ramki przenoszącej informację oraz przepisaniu danych użytecznych do zaalokowanego obszaru pamięci karty (bufora) przechowującego aktualną wartość tych danych. W zależności od charakteru wymiany przetwarzanie takie odbywałoby się cyklicznie lub aperiodycznie.

Drugi obieg to obieg wymuszony przez warstwę aplikacji pracującej pod kontrolą systemu operacyjnego. Zadaniem tego obiegu danych jest przekazywanie danych użytecznych z warstw niższych do modułów komunikacyjnych systemu wizualizacji.

Warstwa aplikacji pracująca pod kontrolą systemu operacyjnego po każdym cyklu przetwarzania warstw niższych musi zostać powiadomiona o pojawieniu się nowych danych użytecznych. W przeciwnym razie karta pracowałaby sama dla siebie, bez żadnych mechanizmów synchronizacji obiegów pierwotnego i wtórnego. Sygnałem synchronizującym może być wygenerowanie przerwania przez kartę sieciową. Funkcja obsługi przerwań sterownika karty powinna wykrywać, od jakiego zdarzenia przyszło przerwanie i inicjować obieg wtórny. Oczywiście, nie zawsze obieg wtórny musi być tak ściśle synchronizowany z obiegiem pierwotnym i inicjalizowany od przerwania. Dla zdarzeń wolnozmiennych informacje mogą być pozyskiwane na drodze próbkowania wymuszanego przez odpowiednie moduły systemu wizualizacji.

Przekazanie informacji z karty sieciowej do aplikacji użytkownika musi się odbywać dwustopniowo. Pierwszy stopień to powiadomienie o zdarzeniu sterownika karty. Drugi stopień to skomunikowanie sterownika z aplikacją użytkownika i powiadomienie jej o konieczności aktualizacji danych oraz odczytanie tych danych z buforów komunikacyjnych (rys. 6).



Rys. 6. Przekazywanie informacji o zdarzeniu
Fig. 6. Passing the information of event

Przy obsłudze szybkozmiennych zdarzeń czas odstępu pomiędzy generacją kolejnych przerwań może być bardzo krótki. Należy zatem zoptymalizować czas trwania obsługi prze-

rwania, tak aby nie był on dłuższy od czasu jego generacji i nie mógł tym samym powodować niestabilności pracy systemu.

Ponieważ układ podwójnego powiadamiania wprowadza dodatkowe opóźnienia do czasu przetwarzania warstwy aplikacji, korzystne jest, aby szybkozmienne zdarzenia buforować na poziomie sterownika, a sygnał powiadamiający warstwę użytkownika generować dopiero przy przepełnieniu bufora. Takie przekazywanie informacji można nazwać pakietowym i jest to rozwiązanie mogące mieć zastosowanie przy transferach cyklicznych lub przy lawinach zdarzeń aperiodycznych. Wymaga ono jednak prowadzenia dynamicznych buforów z dostępem FILO na poziomie sterownika karty, a więc jądra systemu.

Generalnie sieciowa warstwa aplikacji powinna być tak skonstruowana, aby możliwy był dostęp do danych z poziomu aplikacji użytkownika w czasie rzeczywistym (przy założonym T_R i dla określonych możliwości sprzętowych).

3.2. Wizualizacja

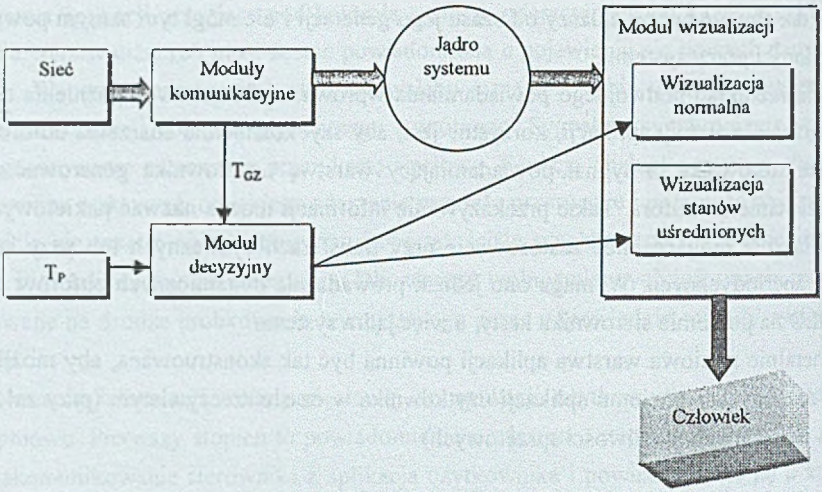
Zagadnienie prezentacji szybkozmiennych zdarzeń jest trudne lub wręcz niemożliwe do implementacji. Wynika to z prostego faktu, iż przy czasie generacji zdarzeń mniejszym od czasu percepcji i reakcji użytkownika ($T_{GZ}<T_P$) niemożliwe jest prowadzenie wizualizacji czegokolwiek. Nawet przy założeniu możliwości zasygnalizowania każdego wygenerowanego zdarzenia sytuacja taka objawiałaby się migotaniem przekazu (wizualizacja stanów, zmiany pomiarów, alarmy, dźwięki itp.).

Wizualizacja w czasie rzeczywistym jest możliwa dla przypadku zdarzeń szybkozmiennych przy założeniu wprowadzenia specjalnych stanów (stanów uśrednionych) zależnie od charakteru informacji przenoszonej przez dane zdarzenie.

Oczywiście, należy się zastanowić, czy szybkozmienność danego zdarzenia jest dla danego poziomu procesu zjawiskiem poprawnym i dopuszczalnym w normalnej pracy, czy też stanowi informację o nieprawidłowościach. Dla przykładu, transmisja wartości pomiarowych z okresem 10 ms może być wymogiem śledzenia i dokładnej archiwizacji pomiaru, natomiast wyświetlenie tej wielkości na ekranie w czasie rzeczywistym jest problematyczne. Wyświetlana musiałaby być wartość próbkowana lub poddana jakiejś metodzie uśredniania.

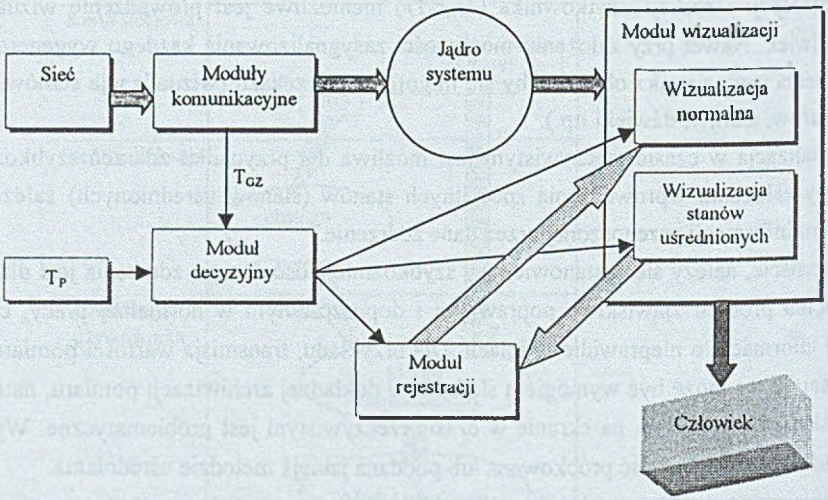
Rozwiązaniem problemu wizualizacji szybkich zmian stanu informacji binarnych mogłoby być wprowadzenie specjalnego stanu oznaczającego migotanie. Dla przypadku pomiarów można wprowadzić specjalny stan pomiaru, którego wartość będzie poddawana przetwarzaniu (np. uśrednianiu, próbkowaniu, maksimum, minimum). Jeżeli dopuszczalna jest wizualizacja „po fakcie", można przyjąć powyższe rozwiązanie z dołączonym mechanizmem rejestracji zdarzeń (zob. rozdz. 3.3) i późniejszego odtwarzania (rys. 8).

Istotne jest określenie czasu T_P, po przekroczeniu którego system nie ma możliwości skutecznie zwizualizować informacji i powinien przejść do obsługi stanów uśrednionych (rys. 7).



Rys. 7. Mechanizm wizualizacji stanów uśrednionych
Fig. 7. The middle state visualization mechanism

Wartość czasu T_P zależy od konkretnego obiektu i zdarzenia i może być dobierana tylko na zasadzie szacunkowej.



Rys. 8. Mechanizm wizualizacji stanów uśrednionych z rejestracją
Fig. 8. The visualization mechanism of the middle states with registration

3.3. Rejestracja i archiwizacja

W przeciwieństwie do zagadnienia wizualizacji proces rejestracji i archiwizacji zdarzeń szybkozmiennych jest zawsze możliwy do przeprowadzenia w systemach, gdzie parametry przepustowości warstwy aplikacji gwarantują pracę w czasie rzeczywistym.

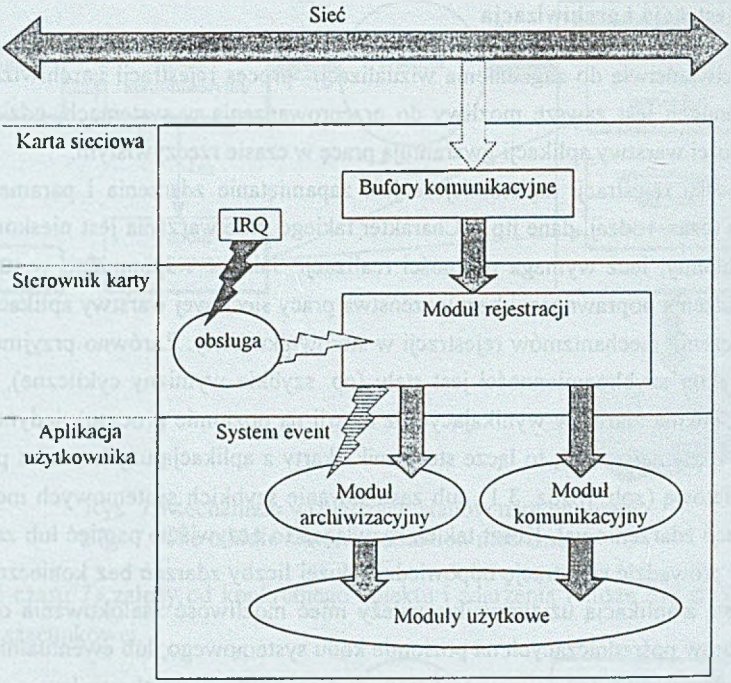
Dla procesu rejestracji najistotniejsze jest zapamiętanie zdarzenia i parametrów z nim związanych (czas, rodzaj, dane itp.). Charakter takiego przetwarzania jest nieskomplikowany i nieczasochłonny, lecz wymaga pewności realizacji. Jak już wspomniano w rozdziale 3.1, z punktu widzenia poprawności i bezpieczeństwa pracy sieciowej warstwy aplikacji korzystne jest umieszczenie mechanizmów rejestracji w sterowniku karty. Zarówno przyjmując założenie, iż charakter szybkozmienności jest stały (np. szybkie wymiany cykliczne), jak i przejściowy (np. lawina alarmów wynikających z awarii na poziomie procesu), jedyne, co można dodatkowo zoptymalizować, to łącze sterownika karty z aplikacją użytkownika przez komunikację pakietową (zob. rozdz. 3.1), lub zastosowanie szybkich systemowych mechanizmów synchronizacji zdarzeniowej. Koszt takich rozwiązań to oczywiście pamięć lub zasoby systemowe. Aby prowadzić rejestrację odpowiednio dużej liczby zdarzeń bez konieczności komunikowania się z aplikacją użytkownika, należy mieć możliwość zaalokowania odpowiednio dużych buforów pośredniczących na poziomie kodu systemowego, lub ewentualnie z niewielką stratą wydajności przemapowywać obszary pamięci aplikacji użytkownika.

Rejestracja zdarzeń nie wymagających zarejestrowania każdego wystąpienia lub zdarzeń wolnozmiennych może być zlokalizowana w aplikacji użytkownika i działać na zasadzie próbkowania.

Mechanizmy archiwizacji zwykle służą do śledzenia zdarzeń „po fakcie" i związane są z prowadzeniem archiwum na nośnikach o dostępie stosunkowo wolnym (np. dysk). Archiwizacja w czasie rzeczywistym nie ma zatem sensu. Przy współdziałaniu archiwizacji z rejestracją mechanizmy archiwizacji mogą pracować na poziomie aplikacji użytkownika, wykonując swoje funkcje z priorytetem normalnej aplikacji.

Zakładając poprawność pracy stacji wizualizacyjnej, istotne jest zapamiętywanie zdarzeń w kolejności ich wystąpień, lecz zupełnie nieważne jest, z jakim opóźnieniem zostaną one zarchiwizowane.

Widać zatem wyraźnie, że dla struktury z rys. 4 funkcje rejestrujące powinny być wyraźnie oddzielone od funkcji archiwizujących zarówno na poziomie funkcjonalnym, jak i na poziomie warstw systemowych (rys. 9).



Rys. 9. Lokalizacja modułów rejestracji i archiwizacji
Fig. 9. The location of registration and archivization module

4. Wnioski

Rozłożenie sieciowych warstw interfejsu pomiędzy poszczególne warstwy sprzętowo-programowe komputera klasy PC, pracującego pod kontrolą systemu operacyjnego MS Windows NT, znacznie upraszcza konstrukcję interfejsu sieciowego na poszczególnych warstwach i wpływa pozytywnie na wykorzystanie jednostki centralnej. Dodatkowo, upraszcza się procedura modyfikacji i testów całego interfejsu przez możliwość niezależnych testów poszczególnych jego elementów. Budowę i korzystanie z interfejsu poprawia również zastosowanie specjalizowanych kart sieciowych.

Dla potrzeb wizualizacji informacji szybkozmiennej, pozbawionej znamion mniej lub bardziej dokładnego próbkowania wartości, konieczne jest zastosowanie specjalnych modułów przetwarzających informację szybkozmienną na wolnozmienną wg określonych algorytmów.

Problem interfejsu pomiędzy szybką siecią a systemem wizualizacyjnym jest stale rozwijany w ramach prac nad systemem KRONOS i sieciami typu FIP.

Opisana metoda podziału interfejsu z sukcesem została zastosowana w Elektrociepłowni Miechowice oraz w PEC Stargard Szczeciński do zbierania stanów zabezpieczeń transformatorów 6/0.6kV zasilających przemienniki częstotliwości dużej mocy z serii GD2000E, GD3000E [17].

Aplikacja do wizualizacji stanów zabezpieczeń transformatorów oraz arbiter magistrali sieci FIP pracują pod kontrolą systemu KRONOS na platformie systemowej Windows NT4.0. Całość pracuje bez zastrzeżeń i z bardzo dużą szybkością i niezawodnością.

LITERATURA

1. Praca zbiorowa pod kier. A. Grzywaka: Rozproszone systemy komputerowe. Pronet, Gliwice 1994.
2. Grzywak A., Kwiecień A.: Perspektywy rozwoju zastosowań sieci komputerowych w górnictwie. Mechanizacja i Automatyzacja Górnictwa nr 5, 6/94.
3. Grzywak A., Kwiecień A.: Rozproszone systemy sterowania i zarządzania procesami technologicznymi. Mechanizacja i Automatyzacja Górnictwa nr 8/94.
4. Grzywak A., Kwiecień A.: Sieci komputerowe w systemach sterowania i zarządzania w górnictwie. Konferencja Międzynarodowa ICAMC'95 World Mining Congress.
5. Cupek R.: Metody hierarchizacji rozproszonych procesów przemysłowych. ZN Pol. Śl. s. Informatyka z. 28, Gliwice 1995.
6. Bigewski Z.: Optymalizacja pracy sieci przemysłowych. ZN Pol. Śl. s. Informatyka z. 28, Gliwice 1995.
7. Kwiecień A.: Sieć rozległa FIP. ZN Pol. Śl. s. Informatyka z. 24, Gliwice 1993.
8. Kwiecień A., Gaj P.: Sieć FIP, wstęp do analizy czasowej. ZN Pol. Śl. s. Informatyka z. 28, Gliwice 1995.
9. Kwiecień A., Gaj P., Grzywak A., Mrówka Z.: Rozwiązania sprzętowe i programowe sieci przemysłowej FIP. ZN Pol. Śl. s. Informatyka z. 30, Gliwice 1996.
10. Kwiecień A., Gaj P., Mrówka Z.: Optymalizacja wymian w sieci FIP. ZN Pol. Śl. s. Informatyka z. 32, Gliwice 1997.
11. Kwiecień A., Gaj P., Mrówka Z.: O pewnej implementacji sieci typu FIP. ZN Pol. Śl. s. Informatyka z. 34, Gliwice 1998.
12. Cupek R., Fojcik M.: Budowa modułów komunikacyjnych stacji nadzorczej z sieciami przemysłowymi. ZN Pol. Śl. s. Informatyka z. 32, Gliwice 1997.
13. FIPCODE Software v. 5.0 User Reference Manual; DPS 50263 b–en, Clamart 1994.
14. FIP NETWORK General Introduction; DPS 50249 aA, Clamart 1990.
15. Klubowe materiały szkoleniowe; WorldFip, Clamart 1995, 1996, 1997, 1998.

16. FIP STARTER KIT materiały klubowe WorldFip; Clamart 1995.
17. GD2000E User Manual; DPS 51374 c–en, Clamart 1997.
18. Dokumentacja techniczna stacji RM_IO 32; Proloc, Katowice 1997.
19. System KRONOS w.2.0 – podręcznik użytkownika; Proloc, Katowice 1997.

Recenzent: Dr inż. Włodzimierz Boroń

Wpłynęło do Redakcji 12 kwietnia 1999 r.

Abstract

The paper describes the issue of connection between high-speed industrial network and industrial SCADA system. The FIP type network was used as a high-speed network. The FIP is able to transmit events with speed that is problematic to service by user and operating systems. The base of software component was used KRONOS SCADA system and operating system MS Windows NT 4.0 Workstation.

The good solution of above connection is preparing the network interface with dividing functionality between each part of hardware and software component of visualization station. Only three ISO/OSI layers are useful in industrial networks: physical, link and application. These layers added to hardware part of interface (network board) stay very helpful to build rest of application layer on operating system layer.

To presentation speed changing data, the visualization system should be expanded of special conversion module.

Author based on early publications ([8], [9]) and owns experience with building, using and servicing industrial computer network and SCADA systems.