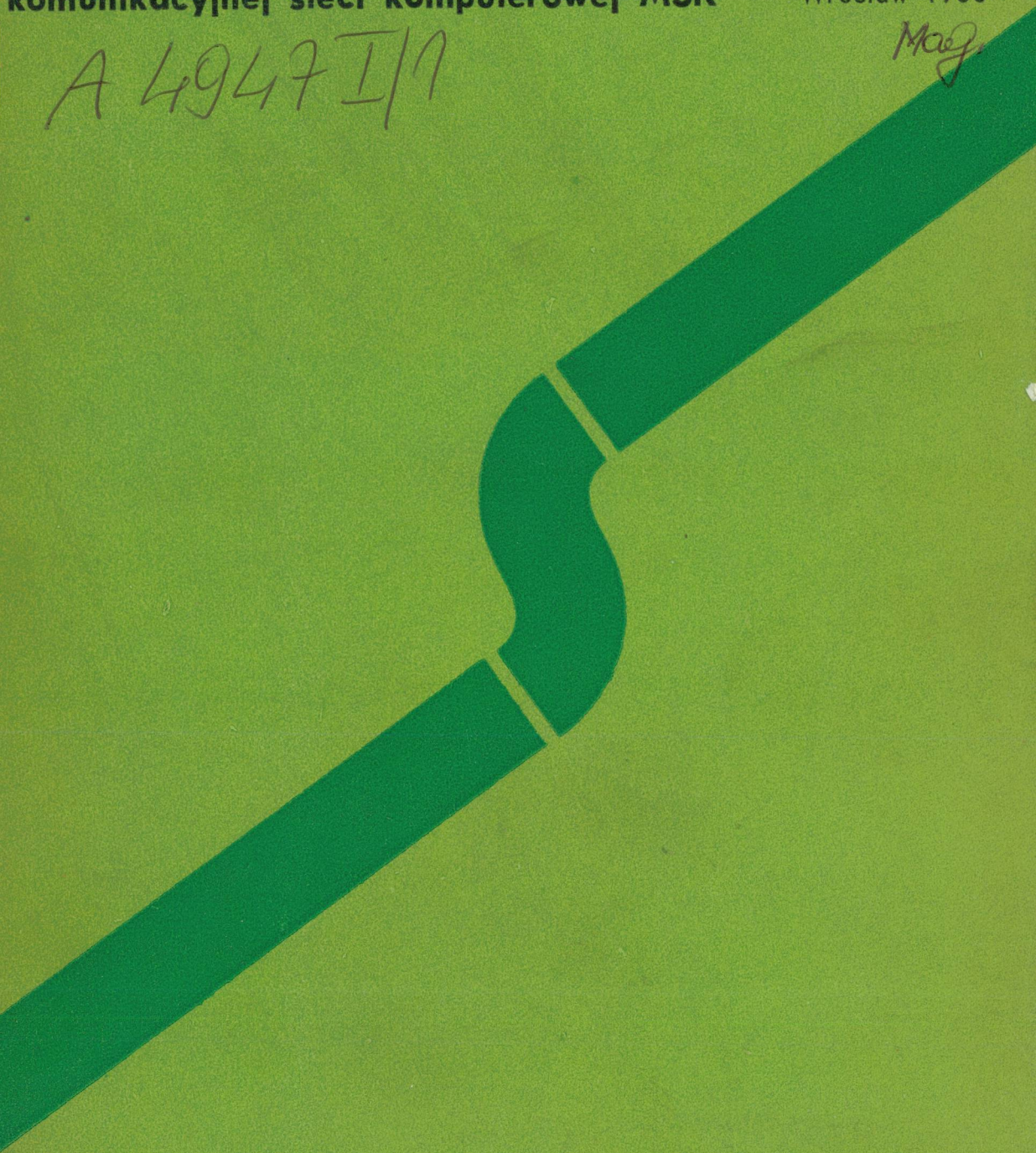


**Zagadnienia oprogramowania węzła podsieci
komunikacyjnej sieci komputerowej MSK**

Wrocław 1983

A 4947 I/1

Maciej



PRACE NAUKOWE POLITECHNIKI WROCLAWSKIEJ

Scientific Papers of the Computer Centre
of Wrocław Technical University

No 2

No 2

Studies and Research

No 2

1983

Software problems of MSK computer network
communication subnetwork node

Prace Naukowe Centrum Obliczeniowego
Politechniki Wrocławskiej

2

Seria:
Studia i Materiały

2

Zagadnienia oprogramowania węzła podsieci komunikacyjnej sieci komputerowej MSK



Wydawnictwo Politechniki Wrocławskiej · Wrocław 1983

Redaktor naczelny
Marian KLOZA

Redaktor naukowy
Antoni MAZURKIEWICZ

Opracowanie redakcyjne
Maria KOPEĆ

© Copyright by Wydawnictwo Politechniki Wrocławskiej, Wrocław 1983

WYDAWNICTWO POLITECHNIKI WROCŁAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

Nakład 150 + 50 egz. Ark. wyd. 22,5. Ark. druk. 18 + wkl. Papier offset. kl. V, 70 g, B1.
Przekazano do druku w marcu 1983 r. Druk ukończono w maju 1983 r.
Zakład Graficzny Politechniki Wrocławskiej. Zam. nr 666/83 – H-13 – Cena zł 180,-

WPROWADZENIE

Na Politechnice Wrocławskiej są prowadzone prace badawcze i projektowe zmierzające do uruchomienia sieci komputerowej obsługującej wyższe uczelnie w Polsce. Sieć ta, nazywana Międzyuczelnianą Siecią Komputerową (MSK), będzie siecią heterogeniczną - umożliwiającą włączanie w nią różnych komputerów obliczeniowych oraz siecią uniwersalną - zdolną do współpracy z innymi sieciami już działającymi na świecie lub dopiero budowanymi. Pierwszym etapem tego przedsięwzięcia jest utworzenie sieci trójwęzłowej, pozwalającej na sprawdzenie w rzeczywistych warunkach przyjętych założeń i ich realizacji; koncepcja tej sieci eksperymentalnej była znana również pod kryptonimem "Pitagoras". Na tym etapie zakłada się istnienie trzech węzłów sieci, w których są zainstalowane komputery obliczeniowe typu Odra 1300 lub/i Riad oraz połączenie tych komputerów podsiecią komunikacyjną, działającą na zasadzie komutacji pakietów.

Prace publikowane w tym zbiorze dotyczą fazy projektowej i realizacyjnej budowy podsieci komunikacyjnej. Założenia wstępne, które nie są tu omawiane, wynikają z ogólnej koncepcji sieci MSK i dotyczą standardów zachowania się podsieci oraz użytego sprzętu.

Koncepcja sieci MSK opiera się na siedmiowarstwowym modelu ISO tzw. Architektury Systemów Otwartych, w której trzy warstwy "dolne" realizuje podsieć komunikacyjna. Zgodnie z wymaganiem uniwersalności sieci MSK, jako podstawę do zdefiniowania protokołów tych warstw, wybrano zalecenie CCITT X.25 (protokół liniowy LAPB i protokół pakietowy X.25/3).

Pierwszy etap, tj. podsieć pilotową, przyjęto wykonywać na minikomputerach SM-3 wyposażonych w jednostki sterujące modemami do transmisji synchronicznej (tzw. adaptery liniowe). Minikomputery SM-3 (pojemność pamięci 28 K słów 16-bitowych, lista rozkazów zgodna ze standardową listą PDP 11) spełniają rolę węzłów podsieci komunikacyjnej - DCE (ang. Data Communication Equipment). Komputery obliczeniowe ze swoimi zasobami są dołączane do węzłów podsieci za pośrednictwem tzw. procesorów

czołowych (sieciowych), spełniających funkcje DTE (ang. Data Terminal Equipment). Również terminale użytkowników komunikują się z siecią przez procesory czołowe: lokalnie - z komputerem dołączonym do tego samego węzła lub zdalnie - przez podsieć komunikacyjną i procesor czołowy w węźle docelowym.

Zasadnicze zadanie przy budowie podsieci komunikacyjnej polega na oprogramowaniu minikomputera jako jej węzła. Ze względu na brak stosownego systemu operacyjnego i konieczność oszczędnego gospodarowania pamięcią, w Centrum Obliczeniowym Politechniki Wrocławskiej opracowano system operacyjny węzła pisząc całe oprogramowanie w języku assemblera. Przedmiotem większości prezentowanych tu artykułów są różne aspekty tego systemu operacyjnego. Artykuły były pisane przez autorów oprogramowania jednocześnie z pracami nad laboratoryjnym uruchomieniem podsieci trójwęzłowej, stąd - w wielu przypadkach - oprócz opisu przyjętego rozwiązania zawierają jego krytyczną ocenę.

Ogólny opis modularnej struktury systemu operacyjnego węzła zawiera praca Lewoca [1], będąca częstym odnośnikiem do szczegółowych rozważań innych autorów i stanowiąca m.in. przewodnik po oznaczeniach używanych w dokumentacji i opisach programów. Ten sam autor [2] opisuje jądro systemu, jakim jest tzw. koordynator węzła zapewniający właściwy przepływ sterowania między modułami programowymi. Kolejne dwa artykuły: Łanowskiej [3] i Stanisza [4] dotyczą obsługi węzła i jego zachowania się w sensie operatorskim - inicjacji pracy, restartu, sposobu komunikowania się obsługi.

Następna grupa prac wiąże się z zasadniczą funkcją węzła, tj. realizacją protokołów sieciowych. Artykuły Fryźlewicza, Dubielewicz, Kołesnik, Goetza i Mikulskiej [5-10] dotyczą protokołu pakietowego (X.25/Lewel 3) i jego implementacji w węźle zbudowanym na minikomputerze SM-3. Oprócz ogólnych zasad tego protokołu opisano poszczególne fazy transmisji danych w podsieci, podając uzasadnienia dla przyjętych ograniczeń i wyboru opcji dopuszczalnych w zaleceniach CCITT. Omówiono tu m.in. rolę tzw. "semidatagramów" wprowadzonych przez twórców podsieci, aby umożliwić komunikowanie się węzłów między sobą. Poziom liniowy (protokół HDLC w wersji LAPB) omawia Szwarz [11] podając przyjęte w podsieci opcje i wartości parametrów, a także omawiając podział zadań tego poziomu między sprzęt (adapter liniowy) i oprogramowanie węzła.

Wiele wymagań ogólnych, nie będących przedmiotem żadnego protokołu sieciowego, a koniecznych do ustalenia wewnętrznego standardu podsieci, zostało zaproponowanych przez zespół z Centrum Obliczeniowego Politechniki Wrocławskiej i przedyskutowanych na dwóch ogólnopolskich spotkaniach roboczych w 1980 i 1981 roku. Jednym z takich zagadnień jest metoda i organizacja pomiarów w podsieci - przede wszystkim do diagnos-

tyki, ale też do badań eksploatacyjnych i rozwojowych. Podstawowe założenia miernictwa sieciowego i sposoby przekazywania wyników opisano w artykułach Hudymy i Kosmulskiej-Bochenek [12-14].

Innym problemem wymagającym teoretycznego rozwiązania przed przystąpieniem do realizacji oprogramowania węzła jest sposób marszrutyzacji pakietów w podsieci - zagadnieniu temu jest poświęcona praca Lewoca [15].

Na pograniczu spraw ogólnoteoretycznych i techniczno-programowych znajdują się rozważania Muszyńskiego i Stanisza [16] dotyczące czasowego przechowywania danych w węzłach podsieci. Wybór optymalnej organizacji buforów może tu mieć istotne znaczenie dla przepustowości sieci ze względu na skąpe zasoby pamięci minikomputera SM-3 i jego niewielką szybkość.

Prace Bieleninika, Stanisza i Szwarcza [17] oraz Łanowskiej [18,19] traktują o realizacji modułów programowych obsługujących podstawowe urządzenia zewnętrzne węzła: adapter liniowy, urządzenia operatorskie i zegar.

Sprawom testowania i uruchamiania tak złożonego systemu jakim jest oprogramowanie węzła są poświęcone prace Bechera [20] i Królika [21]. W artykule [20] zawarto interesujące uwagi co do trudności występujących w zespole opracowującym nowe zadania programowe z jednoczesnym opanowywaniem nowego sprzętu i nowego języka.

Zeszyt zamyka artykuł Lewoca i Komorowskiego [22], dający przegląd prac badawczych zmierzających do optymalizacji rozwiązań węzła podsieci oraz praca Huzara [23], będąca przykładem teoretycznego ujęcia zagadnień występujących podczas budowy złożonych systemów programowych.

Można oczekiwać, że zebrane w tym zeszycie prace będą użyteczne dla tych wszystkich, którzy zajmują się inżynierią oprogramowania szczególnie w systemach komunikacyjnych - dostarczając im pewnych spostrzeżeń metodycznych i warsztatowych. Powinny być też przydatne dla przyszłych użytkowników bądź naśladowców sieci MSK, zwłaszcza tych, którzy będą tworzyli nowe wersje węzła, wprowadzali nowe udogodnienia i rozszerzali zakres usług podsieci komunikacyjnej.

W. Komorowski

Wrocław, październik 1981

*Sieć komputerowa, podsieć komunikacyjna,
węzeł, minikomputer, oprogramowanie*

Józef LEWOC^{*)}

Struktura oprogramowania węzła podsieci komunikacyjnej

Centrum Obliczeniowe Politechniki Wrocławskiej przygotowuje podsieć komunikacyjną dla Międzyuczelnianej Sieci Komputerowej, która ma zintegrować zasoby obliczeniowe różnych ośrodków akademickich w Polsce. W artykule omówiono opracowywane rozwiązanie węzła podsieci komunikacyjnej z komutacją pakietów (według zaleceń X.25 CCITT). Omówiono funkcje węzła i możliwość ich realizacji na minikomputerach SM-3. Przedstawiono koncepcję modułowej struktury oprogramowania węzła na SM-3 oraz sposób realizacji podstawowych i pomocniczych funkcji węzła. Opisano też przepływ informacji przez węzeł i czynności wykonywane przez poszczególne moduły programowe w typowej sytuacji połączenia wirtualnego.

1. WYMAGANIA OGÓLNE

W Międzyuczelnianej Sieci Komputerowej (MSK), która ma łączyć zasoby obliczeniowe różnych ośrodków akademickich w Polsce, będzie wyróżniona autonomiczna podsieć komunikacyjna z komutacją pakietów. Podsieć ma świadczyć usługi dla dołączonych do niej abonentów, którymi będą komputery komunikacyjne, a następnie terminale podłączane bezpośrednio do podsieci.

Węzły podsieci komunikacyjnej będą zbudowane na minikomputerach. Oprogramowanie węzła powinno pozwolić na wykonywanie następujących podstawowych funkcji:

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

A. Odbieranie i nadawanie informacji od/do współpracujących abonentów i innych węzłów podsieci.

B. Realizacja protokołu liniowego wymiany informacji z sąsiednimi węzłami i abonentami (komputerami komunikacyjnymi) dołączonymi do danego węzła.

C. Realizacja protokołu pakietowego na styku węzeł/komputer komunikacyjny oraz protokołu wymiany międzywęzłowej.

D. Kierowanie pakietów informacji zgodnie z wybranym algorytmem wyboru drogi.

E. Dokonywanie pomiarów wybranych charakterystyk roboczych węzła i rejestracja ważnych zdarzeń zachodzących w węźle.

F. Przekazywanie wybranych charakterystyk roboczych węzła do innych węzłów podsieci lub abonentów.

G. Badanie poprawności działania węzła i przeprowadzanie potrzebnych akcji korekcyjnych (z ewentualną częściową lub pełną odnową informacyjną).

H. Wstępna obróbka, raportowanie oraz ewentualne składowanie zebranych danych pomiarowych i diagnostycznych.

I. Współpraca z operatorem węzła.

J. Inicjowanie pomiarów charakterystyk roboczych podsieci komunikacyjnej, wstępna obróbka pomiarów sieciowych, składowanie lub raportowanie wyników.

Należy przewidywać późniejszą rozbudowę funkcji o obsługę terminali dołączonych bezpośrednio do węzła z ewentualną realizacją protokołu wirtualnego terminala.

Zarówno pełna lista funkcji węzła, jak i sposoby ich realizacji, nie mogą być sztywno ustalone w chwili projektowania oprogramowania. Wynika to z następujących przesłanek:

1. Brak doświadczeń praktycznych w dziedzinie eksploatacji sieci komputerowych nie daje gwarancji, że przyjęte założenia uściślające pozwolą na efektywne działanie sieci komputerowej.

2. Jednym z celów sieci MSK jest doświadczalne zbadanie różnych środków sprzętowych i programowych stosowanych w sieciach komputerowych.

3. Złożoność zagadnień oraz ograniczony czas na przygotowanie oprogramowania węzła podsieci komunikacyjnej (1 rok) zmuszają do realizacji możliwie prostej wersji początkowej.

Wynika stąd konieczność zastosowania modułowej organizacji oprogramowania, pozwalającej na stosunkowo łatwą rozbudowę i (lub) zmiany poszczególnych funkcji.

Dla różnych funkcji węzła będą istniały różne uwarunkowania czasowe. Najostrzejsze będą zwykle obowiązywały podczas odbierania i nadawania informacji (A). Tu bowiem oprogramowanie musi reagować w odpowiednich przedziałach czasu na dość intensywne strumienie zgłoszeń zewnętrznych.

Struktura oprogramowania węzła podsieci komunikacyjnej

Mniej pilne będą podstawowe funkcje z B÷F. Dla nich nie narzuca się sztywnych granic czasowych. Należy jednak dążyć do tego, aby czas przejścia informacji przez węzeł był możliwie najkrótszy.

Wreszcie pomocnicze funkcje G÷J powinny być realizowane tak, aby nie opóźniać wykonywania funkcji podstawowych. Zatem oprogramowanie powinno być podzielone na zadania o różnych priorytetach.

2. MOŻLIWOŚCI IMPLEMENTACJI NA MINIKOMPUTERZE SM-3

Węzeł podsieci komunikacyjnej opracowywany przez Politechnikę Wrocławską ma być zbudowany na minikomputerze SM-3 w odpowiednim zestawie. Z minikomputerem tym jest dostarczany jednoprogramowy system operacyjny DOS uniemożliwiający spełnienie powyższych wymagań modularności oprogramowania i uwarunkowań czasowych [1]. Zmusza to do opracowania nowego systemu oprogramowania ukierunkowanego na zastosowania sieciowe.

W minikomputerze SM-3 można organizować oprogramowanie w postaci zadań o różnych priorytetach bezwzględnych [2]. Priorytet jest określany na trzech bitach słowa stanu procesora (PSW) - pozwala to na wykorzystanie do 8 poziomów priorytetów. Część zadań może być wywołana przez przerwania zgłaszane na liniach BR7 (priorytet najwyższy) do BR4 (priorytet najniższy).

W minikomputerze jest stosowany wektorowy system przerwań: każdemu przerwaniu jest przyporządkowany wektor określający początek programu obsługi przerwania oraz stan procesora ustawiany dla tego programu. Słowo stanu procesora oraz bieżący licznik rozkazów przerwanego programu są automatycznie umieszczane na stosie. Dla urządzeń zewnętrznych producent określa numery linii BRysterowywanych przez te urządzenia oraz adresy wektorów przerwań. Na przykład zegar sieciowyysterowuje [3] linie BR6 oraz podaje adres wektora przerwań 100 (ósemkowo).

Zestaw minikomputera SM-3 dla węzła podsieci komunikacyjnej zawiera pamięci dyskowe typu IZOT 1370. Jednak podstawowe funkcje węzła powinny być wykonywane w czasie pojedynczych milisekund, zatem programy realizujące te funkcje nie mogą być wywoływane z pamięci dyskowej. Dlatego przyjęto, że pierwsza wersja oprogramowania węzła zostanie zrealizowana jako zbiór programów rezydujących w pamięci operacyjnej.

3. KONCEPCJA STRUKTURY PROGRAMOWEJ WĘZŁA NA SM-3

Uwzględniając możliwości organizacji oprogramowania na minikomputerze SM-3, zalecenia dotyczące priorytetów obsługi przerwań od urządzeń zewnętrznych oraz wymagania omówione w punkcie 1 - proponuje się ogólną strukturę oprogramowania węzła, przedstawioną na rys. 1.

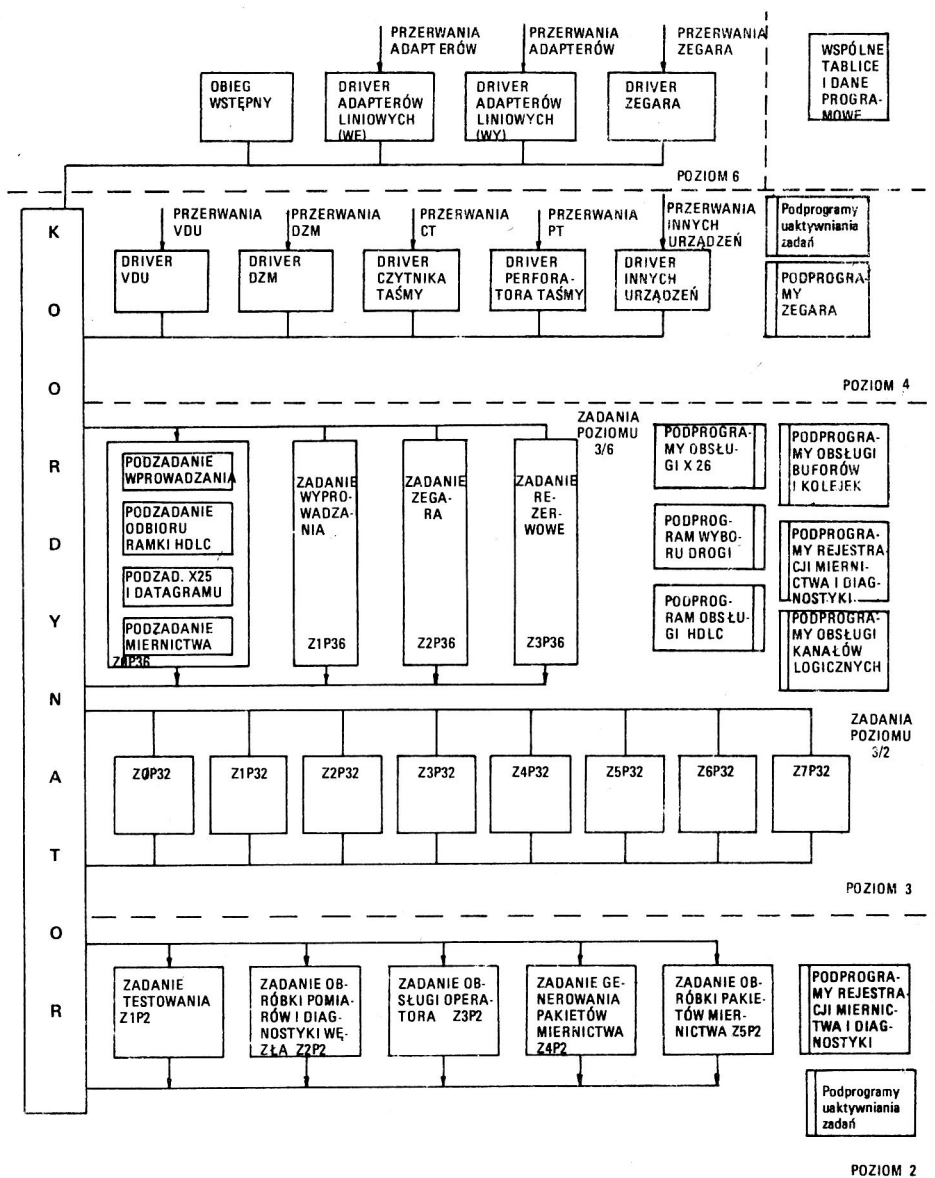
W strukturze tej wyróżniany jest koordynator - prosty system operacyjny - oraz zbiór zadań pracujących na 4 poziomach o różnych priorytetach bezwzględnych. Koordynator steruje działaniem oprogramowania węzła pośrednicząc w przekazywaniu sterowania pomiędzy różnymi zadaniami. Zadania są zamkniętymi częściami oprogramowania, które realizują pewne funkcje użytkowe węzła. Zadania są uruchamiane przez koordynator i kończą pracę oddając sterowanie do koordynatora.

Pewne zadania (drivery urządzeń zewnętrznych) mogą być uruchamiane również (lub wyłącznie) przez przerwania z urządzeń zewnętrznych. Poziom priorytetu 6 zarezerwowany jest dla zadań najbardziej pilnych. W wersji pierwszej przewiduje się tu obieg wstępny inicjujący pracę węzła, drivery adapterów liniowych oraz driver zegara. Poziom 4 jest zarezerwowany dla driverów podstawowych urządzeń wejścia/wyjścia (w wersji I monitor ekranowy, drukarka mozaikowa, czytnik i perforator taśmy).

Poziom 3 jest przeznaczony dla zadań realizujących podstawowe funkcje węzła B÷F, p. 1. Przewidziano tu 4 zadania uruchamiane na żądanie zadań pracujących na wyższym poziomie priorytetu oraz 8 zadań wywoływanych przez zadania pracujące na poziomie niższym. Poziom 2 jest przeznaczony dla 8 zadań realizujących pomocnicze zadania węzła (G÷J, p. 1). W pierwszej wersji będą przygotowane zadania testowania, obróbki pomiarów i diagnostyki oraz obsługi operatora, realizujące funkcje pomocnicze (G, H, I, p. 1). Zadania pracujące na różnych poziomach priorytetów mogą korzystać z zestawów ogólnie dostępnych podprogramów. Podstawowe podprogramy dla pierwszej wersji oprogramowania węzła przedstawiono na rys. 1.

Ze względu na stosunkowo ostre uwarunkowania czasowe występujące w węźle oraz potrzebę oszczędnego gospodarowania pamięcią (należy jak najwięcej pamięci przeznaczyć na bufory informacji), oprogramowanie węzła przygotowywane będzie w języku makroasemblera minikomputera SM-3.

Zaproponowana struktura oprogramowania węzła spełnia wymagania modularności. Każdy prostokąt przedstawiony na rys. 1 symbolizuje moduł programowy pisany i uruchamiany osobno. Moduły będą włączane do oprogramowania węzła w postaci źródłowej i kompilowane wspólnie przez makroassembler. Istnieje także możliwość łączenia modułów w postaci wynikowej za pomocą programu konsolidatora LINK.



Rys.1. Schemat ogólny programu węzła

Fig.1. Schematic Diagram of Node Software

Dla każdego modułu przewidywanego w pierwszej wersji oprogramowania określono funkcje, parametry wejściowe, sposób wywoływania oraz wyjścia [4]. Zdefiniowane symbole globalne, używane w różnych modułach, zapewniają odpowiednie przekazywanie danych i sterowanie pomiędzy różnymi modułami programowymi. Przyjęto też umowę dotyczącą nazw symboli, która eliminuje możliwość dwukrotnego definiowania tych samych symboli globalnych przez różne osoby opracowujące poszczególne moduły.

4. SPOSÓB REALIZACJI PODSTAWOWYCH FUNKCJI WĘZŁA

4.1. Przepływ informacji przez węzeł

Gdy węzeł zostanie uruchomiony, sterowanie obejmuje obieg wstępny, w którym będzie nawiązywany kontakt ze współpracującymi komputerami komunikacyjnymi oraz sąsiednimi węzłami. Nawiązanie kontaktu będzie dokonywane według protokołu liniowego LAPB. Dalsze wykonywanie podstawowych funkcji będzie inicjowane przez strumień zgłoszeń (przerwań) z adapterów liniowych. W realizowanej wersji węzła będą wykorzystywane opracowane przez Instytut Cybernetyki Technicznej adaptory liniowe synchroniczne, odpowiadające urządzeniu DUP-11 firmy Digital Equipment Corporation. Adaptory te przeznaczone dla niskich szybkości transmisji (do 9600 bit/s) współpracują z procesorem centralnym na zasadzie przerwań. Skompletowanie odbieranego znaku (w części odbiorczej) lub rozpoczęcie wysyłania nowego znaku w linię (w części nadawczej) powoduje wygenerowanie przerwania, które powinno być odebrane i obsłużone w czasie odbierania lub nadawania jednego znaku. Jeśli procesor centralny nie zdąży w tym czasie odczytać lub wyprowadzić kolejnego znaku, to znak ten zostanie zgubiony. W wyniku tego ramka będzie zniekształcona i będzie konieczne ponowne jej przesłanie. Protokół liniowy HDLC skutecznie zabezpiecza przed bezpowrotnym gubieniem informacji w podsieci. Jednak nawet wobec intensywnego strumienia zgłoszeń węzeł powinien reagować dostatecznie szybko, aby nie gubić znaków i nie zmuszać do powtórnej transmisji ramek.

Rozpatrzmy najtrudniejszy przypadek, realny dla pierwszej wersji sieci MSK: węzeł jest wyposażony w 6 adapterów liniowych współpracujących z modemami dwukierunkowymi o szybkości transmisji 2400 bit/s. Przy tych założeniach intensywność strumienia zgłoszeń wynosi:

$$\frac{2400 \text{ bit/s}}{8 \text{ bit/zgłoszenie}} \cdot 6 \text{ adapterów} \cdot 2 \text{ kierunki/adapter,}$$

czyli 3600 zgłoszeń/s. Zatem, aby nie gubić znaków, należy zapewnić obsługę jednego zgłoszenia w czasie nie większym niż $1/3600$ s, tj. około 280 μ s. W tym czasie można wykonać w SM-3 około 30 instrukcji programowych.

Jest to bardzo mało i z powyższego oszacowania wpływa następujący wniosek: drivery adapterów liniowych powinny być bardzo krótkie. Należy je ograniczyć do wprowadzenia/wyprowadzenia znaku i ewentualnego poinformowania dalszej części programu obsługi tych zgłoszeń (pracującego na niższym poziomie priorytetu) o zakończeniu lub błędzie transmisji.

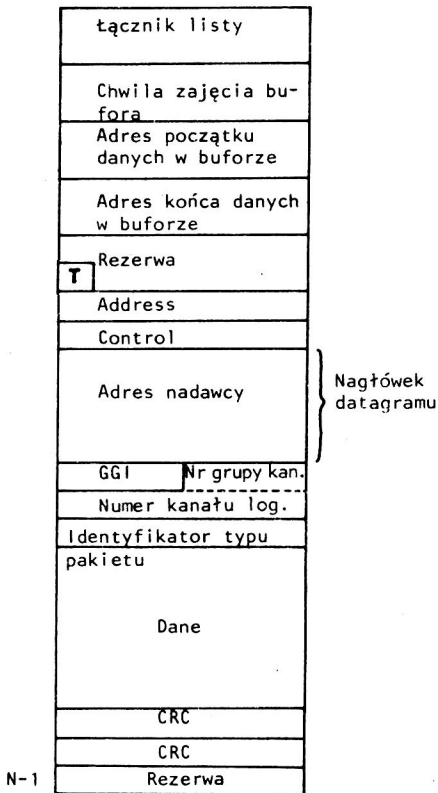
Uruchomiony driver odbiornika czyta zawartość rejestru danych odbiornika i umieszcza ją w cyklicznym buforze wejściowym danej linii (i). Dla każdej linii jest zarezerwowany 512-słowy cykliczny bufor wejściowy, pozwalający na umieszczenie ponad 3 ramek informacyjnych o maksymalnej długości. Jeśli bajt kontrolny odczytanego słowa jest niezerowy, co oznacza początek, koniec lub błąd ramki [5], driver zwiększa o 1 semafor szczegółowy dla linii (i) oraz semafor ogólny dla zadania ZØP36 pracującego na poziomie 3, po czym oddaje sterowanie do koordynatora w celu uruchomienia zadania ZØP36. Zadanie to, z uwagi na jego złożoność, jest podzielone na cztery podzadania, z których każde jest osobnym modulem programowym. Jeśli koordynator stwierdzi, że semafor zadania ZØP36 jest ustawiony, to uruchomi je od podzadania wprowadzania.

Podzadanie wprowadzania stanowi przedłużenie drivera odbiornika adaptera liniowego, pracujące na poziomie priorytetu 3. Pobiera ono bufor długości dla odbieranej ramki.

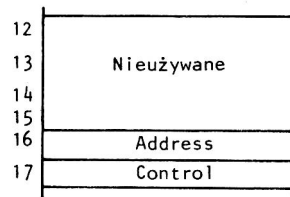
Bufory długie w węźle mają stałą długość 150 bajtów (rys. 2), co pozwala na umieszczenie w nich ramki o maksymalnej długości. Zwykle ramki mają długość znacznie krótszą. Jednak stosowanie zmiennej długości buforów prowadziłoby do dużych komplikacji podczas zwalniania buforów. Zastosowanie stałych lecz krótszych buforów zmuszałoby do dzielenia długich ramek na kilka buforów (co jest niedopuszczalne, gdyż wydłużałoby driver nadajnika adaptera liniowego). Przyjęcie różnych długości buforów dla transmisji wejściowych prowadziłoby do przepisywania długich ramek, co wprowadzałoby dodatkowe wydłużenie czasu pobytu pakietów w węźle. Wolne bufory długie w węźle są zorganizowane w pulę o strukturze listowej. Pulą tą dysponują podprogramy obsługi buforów.

W węźle istnieje także pewna liczba buforów krótkich. Są one wykorzystywane jednak wyłącznie do informacji generowanych w węźle (np. ramki nadzorcze i ramki nienumerowane [6]), gdy wiadomo, że długość bufora krótkiego jest wystarczająca. Wolne bufory krótkie są zorganizowane w pulę o strukturze listowej, podobnie jak wolne bufory długie.

Bufor z datagramem



Bufor z pakietem



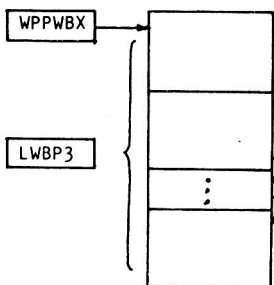
N - długość bufora (w bajtach)

$$N = \begin{cases} 150_{10} & \text{dla bufora normalnego} \\ 24_{10} & \text{dla bufora krótkiego} \end{cases}$$

T - kod typu informacji

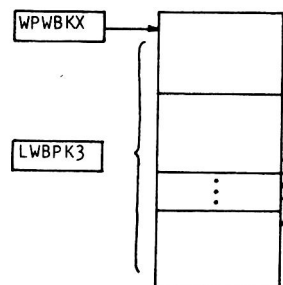
$$T = \begin{cases} 1 & \text{dla pakietu} \\ \emptyset & \text{dla datagramu} \end{cases}$$

Pola wolnych buforów długich



WPPWBX - Adres 1 wolnego bufora długiego
LWBP3 - liczba wolnych buforów długich

Pola wolnych buforów krótkich



WPWBKX - adres wolnego bufora krótkiego
LWBPK3 - liczba wolnych buforów krótkich

Rys.2. Organizacja buforów
Fig.2. Organization of Buffers

Podzadanie wprowadzania przepisuje odebrane z linii informacje z bufora wejściowego do bufora pobranego z puli, badając i odrzucając bajty kontrolne. Wykrycie niezerowego bajtu kontrolnego powoduje zmniejszenie o 1 semafora ogólnego oraz szczegółowego linii(i). Gdy jest błąd transmisji, wówczas podzadanie wprowadzania oddaje bufor do puli, natomiast po skompletowaniu poprawnej ramki uruchamia podzadanie odbioru ramki HDLC. Podzadanie to realizuje protokół LAPB [6]. Zwłaszcza, jeśli odebrana została ramka informacyjna, to uruchomi ono podzadanie X.25 i datagramu. To z kolei realizuje protokół X.25 oraz protokół wymiany międzywęzłowej.

Jeśli odebrany został pakiet CALL REQUEST, to w węźle będzie wypełniona tablica (opis kanału logicznego - rys. 3), pozwalająca na kierowanie strumienia pakietów dla zainicjowanego połączenia wirtualnego.

W każdym węźle będzie zarezerwowana pewna liczba opisów kanałów logicznych. Wolne opisy będą tworzyć pulę o strukturze listowej. Zajęte opisy kanałów logicznych będą zebrane w dwie listy (aby skrócić przeszukiwanie). Opisy kanałów logicznych są przydzielane i wyszukiwane przez podprogramy obsługi kanałów logicznych. Dla pakietu CALL REQUEST [6] podzadanie X.25 i datagramu wywołuje podprogram wyboru drogi określający numer linii (j), do której należy kierować wszystkie pakiety zainicjowanego połączenia logicznego. W pierwszej wersji zostanie zastosowany algorytm dróg ustalonych z trasą alternatywną, w którym uszkodzenie linii podstawowej prowadzi do kierowania pakietów drogą zastępczą. Algorytm taki może doprowadzić do utworzenia cyklu (trasy zamkniętej) w podsieci komunikacyjnej. Aby likwidować cykle, podzadanie X.25 i datagramu będzie sprawdzać (przed założeniem opisu kanału logicznego) czy nie istnieje już opis o tym samym numerze kanału logicznego i tym samym adresie nadawcy. Jeśli tak, to połączenie wirtualne, dla którego powstała pętla, zostanie rozłączone.

Jeśli odebrany pakiet może być przesłany dalej, to podzadanie X.25 i datagram wywołą podprogram HDLC, który dopisze nagłówek HDLC do pakietu (lub datagramu) i wywołą podprogram obsługi kolejek. Bufor zostanie dopisany do kolejki wyjściowej linii. Jeśli kolejka była pusta, to zostanie zainicjowane wyprowadzanie ramki z omawianego bufora. Podprogram obsługi HDLC oddaje sterowanie do wywołującego go podzadania X.25 i datagramu, które kończy pracę zadania ZØP36, oddając sterowanie do koordynatora. Dalsze pakiety nawiązanego połączenia wirtualnego będą przechodziły podobną drogę z tym, że będą wysyłane w linię (j) wybraną dla pakietu CALL REQUEST. Jeśli podzadanie X.25 i datagramu stwierdzi, że odebrano pakiet pomiarowy (rozpoznawany po polu General Format Identifier (GFI) [7]), to odda sterowanie do podzadania miernictwa. Podzadanie to nie będzie przygotowywane w pierwszej wersji oprogramowania.

tącznik listy									
ø	ø	ø	ø	Nr grupy kan.					
Nr kanału									
ø	ø	ø	ø	Nr grupy kan.					
Nr kanału									
p	X	X	X	X	X	X	TW		
Dł. pola nad.				Dł. pola odb.					
Adres odbiorcy									
Adres nadawcy									
Nr linii od nadawcy									
Nr linii do odbiorcy									
Stan kanału logicznego									
Rozmiar okna									
ø	P ₁ (S)								
ø	P ₂ (R)								
ø	P ₂ (S)								
ø	P ₁ (R)								
Wskaźnik fazy									
Kody przyczyn zerowania, rozłączenia lub restartu									
Licznik limitów czasowych I									
Licznik limitów czasowych II									
Chwila przestania ostatniego pakietu									
Chwila zajęcia opisu kanału logicznego									
Dług.kol.buforów wstrzymanych									
Adres pierwszego bufora wstrzymanego									
Adres ostatniego bufora wstrzymanego									

}	Numer kanału logicznego na wejściu
}	Numer kanału logicznego na wyjściu
-	Pole wskaźników P,TW
-	Długości pól adresowych

Numer kanału logicznego na wejściu

Numer kanału logicznego na wyjściu

- Pole wskaźników P,TW

- Długości pól adresowych

Rys.3. Opisy kanałów logicznych

Fig.3. Tables of Logic Channel

Przewiduje się, że będzie ono dopisywać do pakietu pewne charakterystyki robocze węzła i przesyłać pakiet pod zadany adres, albo przekazywać go zadaniu obróbki pakietów miernictwa (w węźle spełniającym rolę centrum pomiarowego podsieci komunikacyjnej). Zainicjowanie wyprowadzania informacji z bufora powoduje pojawienie się ciągu przerwań z nadajnika adaptera liniowego.

Przerwania te są odbierane przez driver nadajnika, który wyprowadza kolejne znaki z bufora. Po wysłaniu ostatniego znaku lub wskutek błędu transmisji adapter nadajnika zwiększa o 1 zawartość semafora ogólnego zadania wyprowadzania i semafora szczegółowego linii (j) oraz oddaje sterowanie do koordynatora. Ten stwierdza, że semafor ogólny zadania wyprowadzania jest ustawiony i uruchamia je. Zadanie wyprowadzania wykrywa przyczynę uruchomienia i zmniejsza zawartość swego semafora ogólnego oraz semafora szczegółowego linii (j). Jeśli transmisja ramki została zakończona bez błędu, to zadanie wyprowadzania przekazuje bufor z ramką informacyjną do kolejki retransmisji linii. Następnie pobiera bufor z kolejki wyjściowej linii (jeśli jest niepusta), umieszcza w ramce aktualne wartości zmiennych $N(R)$ i $N(S)$ protokołu LAPB I61 i inicjuje wyprowadzanie.

Jeśli w czasie transmisji wystąpi błąd, to zadanie wyprowadzania inicjuje wyprowadzanie tej samej ramki. Gdy jest zbyt duża liczba nieudanych transmisji, wówczas nadajnik adaptera linii (j) zostaje wyłączony. Wyłączenie linii jest sygnalizowane oprogramowaniu X.25 (podprogramy obsługi X.25) w celu rozłączenia wszystkich połączeń wirtualnych wykorzystujących tę linię. Zadanie wyprowadzane kończy pracę oddając sterowanie do koordynatora.

4.2. Odmierzanie czasów

Podczas realizacji protokołu liniowego i pakietowego są odmierzane czasy, po których - jeśli nie zajdą określone zdarzenia - należy podejmować pewne działania korekcyjne (np. retransmisja ramki informacyjnej wtedy, gdy nie pojawi się w ciągu określonego czasu potwierdzenie odebrania tej ramki). Podobna potrzeba istnieje np. w driverach obsługi podstawowych urządzeń zewnętrznych. Aby to umożliwić wprowadzono na poziomie 3. zadanie zegara, które można rozszerzać o tzw. fragmenty zawieszane. Mechanizm działania jest następujący:

Dowolne zadanie poziomu 3. lub 4. wywołuje podprogram zegara podając mu parametr (p), adres początkowy fragmentu zawieszonego (A) oraz liczbę sekund, po których należy uruchomić fragment zawieszony, plus 1. Podprogram zegara umieszcza te dane w tablicy fragmentów zawieszonych (TFZA), uzupełniając je o słowo stanu procesora (PSW) (pozwala to na

odmierzenie czasu dla zadań pracujących na różnych poziomach priorytetu). Następnie sterowanie zostaje oddane do programu wywołującego.

Jeśli wystąpi zdarzenie eliminujące potrzebę wykonania fragmentu zawieszonego A, to może wywołać ono inny podprogram zegara, podając mu parametr p i adres A. Wówczas pozycja tablicy zawierająca adres A i parametr p zostanie skasowana.

Wewnętrzny zegar minikomputera SM-3 wytwarza co 20 ms sygnały przerwań, uruchamiające driver zegara. Ten liczy przerwania, co 1 s ustawia (zwiększa o 1) semafor zadania zegara i oddaje sterowanie do koordynatora.

Koordynator uruchamia zadanie zegara, które wykonuje swoje funkcje podstawowe (zmniejszenie semafora, obliczenie czasu, wywołanie zadań uruchamianych cyklicznie), a następnie przetwarza TFZA, odejmując 1 od liczby sekund, po których należy uruchomić fragment zawieszony. Wyzerowanie tej liczby powoduje ustawienie stanu procesora według pamiętanego PSW oraz przekazanie pod adres A sterowania fragmentowi zawieszonemu. Fragmentowi zawieszonemu jest przekazywany też parametr p. Fragment zawieszony kończy pracę, oddając sterowanie zadaniu zegara. Skasuje ono obsłużoną pozycję TFZA i umieści na jej miejscu pozycję ostatnią (w ten sposób unika się przesuwania zawartości TFZA, a jednocześnie umożliwia się przeglądanie tylko niezbędnej liczby pozycji).

Po przetworzeniu ostatniej pozycji TFZA zadanie zegara kończy pracę, oddając sterowanie do koordynatora.

4.3. Rejestracja zdarzeń i pomiary charakterystyk

W chwili obecnej nie można jeszcze określić wszystkich zdarzeń w węźle, jakie należy rejestrować, ani charakterystyk, które trzeba mierzyć. Dlatego zdecydowano się na zastosowanie w wersji pierwszej prostego i uniwersalnego mechanizmu rejestracji zdarzeń.

Wykrycie zdarzenia rejestrowanego będzie powodować wywołanie podprogramu rejestracji i diagnostyki. Parametr przekazany temu programowi jest numerem zdarzenia. Pewna grupa numerów została przyporządkowana zdarzeniom, które będą zliczane; rejestracja polega w tym przypadku na dodaniu jedynki do zawartości pewnego licznika. Dla innych numerów zdarzeń sposób rejestracji (lub pomiaru) będzie określony indywidualnie.

Rejestracja zdarzeń będzie prowadzona zarówno w celach diagnostycznych (rejestracja błędów), jak i pomiarowych (np. rejestracja liczby różnych typów pakietów odbieranych przez węzeł). Podstawową charakterystyką mierzoną w pierwszej wersji oprogramowania będzie średni maksymalny czas pobytu informacji w węźle (od chwili zajęcia bufora do wyrowadzenia danych w linię).

5. SPOSÓB REALIZACJI POMOCNICZYCH FUNKCJI WĘZŁA

Aby uniknąć opóźnień w wykonywaniu funkcji podstawowych, funkcje pomocnicze węzła będą realizowane przez zadania pracujące na poziomie 2. W wersji pierwszej będą uruchomione trzy takie zadania.

Uruchamiane cyklicznie zadanie testowania będzie przede wszystkim kontrolować poprawność działania adapterów liniowych, sygnalizując ewentualną niesprawność przez wywołanie zadania (Z1P32) pracującego na poziomie 3, aby zainicjować rozłączenie połączeń logicznych wykorzystujących tę linię. Zadanie testowania będzie również kontrolować połączenia wirtualne likwidując opisy kanałów, w których przez zbyt długi czas nie był przesyłany żaden pakiet.

Zadanie obróbki pomiarów i diagnostyki węzła, uruchamiane cyklicznie lub na żądanie, będzie wyprowadzać raport o zarejestrowanych zdarzeniach i zebranych pomiarach.

Zadanie obsługi operatora będzie umożliwiać kontrolę i ingerencję w pracę węzła (np. włączanie i wyłączanie poszczególnych linii). Dla zadań poziomu 2. będą istnieć pewne udogodnienia uniwersalne, ułatwiające realizację różnych funkcji pomocniczych. Zadanie poziomu 2. może być wywoływane przez dowolne zadanie poziomu 3. lub 4. Zwłaszcza zadanie zegara może cyklicznie uruchamiać różne zadania poziomu 2. Do wywoływania zadań poziomu 2. służą podprogramy uaktywnienia zadań. Zadanie wywołujące może przekazywać zadaniu wywoływanemu jednosłowny parametr. Zadanie poziomu 2. może wywołać inne zadanie poziomu 2., przekazując mu jednosłowny parametr. Pozwala to na tworzenie łańcuchów zadań wywoływanych przez jedno zdarzenie (np. driver monitora ekranowego po odebraniu wiersza, wprowadzonego przez operatora, uruchamia zadanie obsługi operatora, a to - po rozszyfrowaniu polecenia - wywołuje zadanie obróbki i miernictwa w celu wyprowadzenia raportu o zebranych danych pomiarowych i diagnostycznych).

Zadanie poziomu 2. może wywołać (przez podprogram uaktywniania zadań) jedno z ośmiu (w pierwszej wersji) zadań poziomu 3. (Z0P32-Z7P32, rys. 1), przekazując mu parametr. Pozwala to na ingerencję zadań poziomu 2. w wykonywanie podstawowych funkcji węzła (np. po wykryciu awarii). Zadanie poziomu 2. może uruchamiać drivery podstawowych urządzeń zewnętrznych w celu zainicjowania transmisji wejścia/wyjścia. Zadanie określa adres pola sterującego i uruchamia koordynator instrukcją przerwania programowego (TRAP0). Zadanie otrzymuje informację o zakończeniu transmisji (słowo odpowiedzi w polu sterującym).

Zadanie poziom 2. może się zawieszać. Po zawieszeniu zadanie zostanie wznowione w chwili, gdy żadne inne zadanie poziomu 2. nie będzie czekać na wykonanie.

6. PRZEWIDYWANY ROZWÓJ OPROGRAMOWANIA WĘZŁA

Omawiana tu pierwsza wersja oprogramowania węzła została uruchomiona do końca roku 1981. W roku następnym przeprowadzono badania węzła w różnych zestawach sprzętu (węzeł-węzeł, węzeł-komputer komunikacyjny, trzy węzły, praca w sieci komputerowej).

Przewiduje się, że oprogramowanie węzła będzie rozwijane w następujących kierunkach:

- dołączenie drivera dysków w celu składowania informacji pomiarowo-diagnostycznych, a następnie przekształcenie koordynatora w nakładkowy system operacyjny z biblioteką zadań poziomu 2., przechowywaną na dysku;
- rozwój oprogramowania miernictwa i diagnostyki węzła oraz podsieci komunikacyjnej i zrealizowanie centrum pomiarowego podsieci w węźle;
- opracowanie drivera multipleksera i zrealizowanie węzła spełniającego funkcje koncentratora terminali podłączonych bezpośrednio do sieci;
- opracowanie driverów dla adapterów liniowych na wyższe szybkości transmisji;
- adaptacja oprogramowania węzła dla innych minikomputerów tej rodziny (SM-4, MERA-60);
- opracowywanie systemów liczących (na podstawie systemu operacyjnego węzła) do innego zastosowania, np. dydaktyki, laboratoriów naukowo-badawczych itp.

LITERATURA

- [1] Programnoe obespečenie SM-3, Diskovaja operacionnaja sistema obščego naznačenija, nr 4.072.088IZ, 1977.
- [2] PDP-11 04/34/45/55 Processor Handbook, Digital Equipment Corporation, Maynard, Massachusetts 1976-77.
- [3] PDP-11 Peripherals Handbook, Digital Equipment Corporation, Maynard, Massachusetts 1975.
- [4] L e w o c J. i in., Zasady przygotowywania oprogramowania użytkowego węzła, Raport Centr. Oblicz. PWr., seria SPR nr 14/80, Wrocław 1980.

Struktura oprogramowania węzła podsieci komunikacyjnej

- [5] L e w o c J. i in., Ocena możliwości realizacji węzła komunikacyjnego na bazie minikomputera SM-3, Raport Centr. Oblicz. PWr., Seria SPR nr 14/79, Wrocław 1979.
- [6] CCITT, Proposed Revision to Level 1 and 2 of Recommendation X.25, COM VII No 374-E, październik 1979.
- [7] CCITT Revised Recommendation X.25, Preface and Level 3, COM VII No 384-E, sierpień 1979.

Praca wpłynęła do Redakcji 25.11.1980 r.

Po poprawieniu 3.03.1982 r.

STRUCTURE OF COMMUNICATION SUBNETWORK NODE SOFTWARE

To integrate computational resources of various university centres in Poland a communication subnetwork for the interuniversity computer network is being prepared by the Computer Centre of the Wrocław Technical University. The paper presents the solution of packet-switching communication subnetwork (developed according to CCITT Recommendations X.25). Node functions and possibilities of their realization on the SM-3 minicomputers are discussed. A general conception of modular node software structure for the SM-3 minicomputers as well as the way in which the primary and secondary node functions can be realized are presented. The data flow through the node and actions performed by individual software modules in a typical situation of virtual call are also discussed.

Verified by Ruta Czaplińska

СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ УЗЛА ПОДСЕТИ СВЯЗИ

Вычислительный центр Вроцлавского технического университета готовит подсеть связи межузловой сети ЭВМ, которая должна объединить вычислительные ресурсы академических центров в Польше. В статье описывается разрабатываемое решение узла подсети связи пакетной коммутации (согласно стандарту X.25, принятому МККТТ). Рассматриваются функции узла и возможность их реализации с использованием SM-3. Предложена концепция модульной структуры программного обеспечения узла и метод реализации основных и вспомогательных функций узла. Описывается, кроме того, ход информации через узел и задачи выполняемые отдельными программными модулями в типических условиях виртуального соединения.

Проверила Малгожата Хейдрих

*Sieć komputerowa,
podsieć komunikacyjna,
system operacyjny, węzeł*

Józef LEWOC^{*)}

Koordinator węzła podsieci komunikacyjnej

W artykule omówiono koordinator węzła podsieci komunikacyjnej dla sieci komputerowej MSK. Przedstawiono uwarunkowanie czasowe dla różnych procesów wykonywanych w węźle. Podano sposób synchronizacji różnych procesów przez koordinator, uzasadniony przedstawionymi uwarunkowaniami czasowymi. Do synchronizacji procesów zastosowano zmodyfikowane semaforey Dijkstry oraz bufor cykliczne.

1. WSTĘP

Brak standardowego systemu operacyjnego, który mógłby być wykorzystany w węźle Międzyuczelnianej Sieci Komunikacyjnej (MSK) tworzonym na minikomputerze SM-3, zmusił Centrum Obliczeniowe Politechniki Wrocławskiej do opracowania pełnego oprogramowania węzła podsieci komunikacyjnej łącznie z systemem operacyjnym ukierunkowanym na zastosowania sieciowe. Jądro tego systemu operacyjnego stanowi koordinator, który kieruje pracą węzła, pośrednicząc w przekazywaniu sterowania pomiędzy różnymi procesami wykonywanymi w węźle. Projektując koordinator dla węzła podsieci komunikacyjnej uwzględniono uwarunkowania (zwłaszcza czasowe) występujące w tym zastosowaniu oraz udogodnienia dostarczane przez minikomputer SM-3, w tym możliwość pracy na jednym z ośmiu poziomów priorytetów bezwzględnych (z wywłaszczaniem) oraz mechanizm stosu [1].

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

Ogólną strukturę oprogramowania węzła podsieci komunikacyjnej MSK omówiono w pracy [2]. Natomiast w niniejszej publikacji opisano kordinator i jego działanie mające na celu odpowiednie zsynchronizowanie różnych procesów w węźle ze strumieniem zgłoszeń zewnętrznych, odbieranych przez jednostkę centralną węzła.

2. UWARUNKOWANIA CZASOWE DLA OPROGRAMOWANIA WĘZŁA

W podstawowej monografii na temat sieci teleinformatycznych [3] przedstawiono rozwiązanie sterowania połączeniami międzywęzłowymi, zastosowane w pierwszej wersji sieci ARPA. W rozwiązaniu tym odebranie pakietu z węzła sąsiedniego powoduje zablokowanie odbiornika adaptera liniowego aż do chwili, gdy staną się dostępne potrzebne zasoby, takie jak czas jednostki centralnej lub bufory dla następnych pakietów. Kopie wysyłanych pakietów są przechowywane w węźle źródłowym aż do chwili, gdy węzeł otrzyma potwierdzenia o poprawnym odebraniu poszczególnych pakietów. Brak takiego potwierdzenia powoduje, po określonym czasie, ponowne nadanie pakietów, dla których nie otrzymano potwierdzenia odbioru. Blokowanie odbiornika adaptera liniowego na czas zajętości jednostki centralnej węzła skutecznie eliminuje potrzebę nadążnej pracy węzła.

W sieci MSK nie można zastosować takiego rozwiązania. Zgodnie z protokołem liniowym LAPB [4], implementowanym w sieci MSK, węzeł lub abonent sieci MSK może przesyłać do współpracującego węzła lub abonenta ciąg ramek informacyjnych w liczbie określonej przez tzw. okno (w pierwszej wersji sieci MSK przyjęto szerokość okna 7). Zablokowanie odbiorników adapterów liniowych może spowodować gubienie odbieranych znaków (w adapterach liniowych stosowanych w węźle podsieci komunikacyjnej nie ma buforów dla większej liczby znaków niż dwa [5], a odbieranie poszczególnych znaków wymaga wykonania programu obsługi przerwanych pochodzących z adaptera liniowego), a co za tym idzie - konieczność ponownej transmisji źle odebranej ramki lub grupy ramek.

Z uwagi na to, że powtórne nadawanie ramek rozpocząć się może dopiero po upływie wielosekundowego limitu czasowego od chwili nadania pierwszej ramki, dla której nie zostanie odebrane potwierdzenie prawidłowego odbioru, zjawisko gubienia znaków jest bardzo niepożądane. Po pierwsze, powoduje to zasadnicze zwiększenie opóźnienia tranzytowego dla pakietów przesyłanych przez podsieć komunikacyjną (od ułamków sekundy do wielu sekund). Po drugie, zmusza do długotrwałego przechowywania kopii nadawanych ramek, w wyniku czego w węźle źródłowym bufory informacyjne są zajęte przez długi czas. Może to doprowadzić do przeciążenia węzła źródłowego i dalszego znacznego zwiększenia czasów przejścia pakietów przez ten węzeł.

Należy także zauważyć, że prawdopodobieństwo gubienia znaków i wynikającej stąd konieczności powtórnej transmisji ramek będzie oczywiście tym większe, im intensywniejszy będzie ruch pakietów w sieci. Zatem wtedy, kiedy sieć będzie najbardziej potrzebna abonentom, będzie pracować najgorzej (w sensie wprowadzania największego opóźnienia tranzytowego pakietów w podsieci komunikacyjnej). Wynika stąd potrzeba podziału funkcji realizowanych przez węzeł pomiędzy pseudospółbieżne procesy, dla których należy stosować szeregowanie z wyłuszczaniem, to znaczy pracujące na różnych poziomach priorytetów bezwzględnych ([6,7]). Przede wszystkim procesy obsługi adapterów liniowych powinny mieć możliwie najłatwiejszy dostęp do zasobów węzła, w tym do czasu jednostki centralnej. Uwarunkowania czasowe wynikające z oceny najbardziej niesprzyjających okoliczności, jakie mogą wystąpić w węźle podsieci komunikacyjnej MSK, określono w pracy [2]. Wynika z nich potrzeba ograniczenia czasu wykonania tych procesów do około 280 μ s/znak. Przy tym odcinki czasu, w których procesy obsługi adapterów liniowych nie mogą uzyskać dostępu do jednostki centralnej z powodu zaangażowania jej w wykonywanie innych procesów (np. obsługi zegara lub koordynatora), powinny być możliwie jak najkrótsze.

Procesy analizy odbieranych informacji na poziomie liniowym i pakietowym ([4,8]) wytwarzania odpowiednich pakietów i ramek odpowiedzi oraz kolejkowania powinny być również realizowane możliwie szybko, aby zminimalizować opóźnienie tranzytowe pakietów w węźle. Warunek nadążności dla tych procesów można określić na podstawie długości buforów cyklicznych wykorzystywanych przez procesy obsługi odbiorników adapterów liniowych i szybkości transmisji w stosowanych łączach. Tutaj gubienie znaków i wynikająca stąd konieczność powtórnej transmisji ramek z jej negatywnymi skutkami następują w razie przepełnienia dowolnego bufora cyklicznego.

Gdy długość buforów cyklicznych wynosi 512 słów (przyjęto tak w pierwszej wersji węzła), co pozwala na buforowanie 512 znaków (proces obsługi odbiorników adapterów liniowych wykorzystuje jedno słowo na odbierany znak, gdyż oprócz bajtu danych czyta też bajt kontrolno-sterujący wytwarzany przez adapter liniowy), i szybkość transmisji wynosi 2400 bitów/s, wówczas przepełnienie bufora może nastąpić najwcześniej po czasie:

$$t = \frac{512 \text{ znaków}}{2400 \text{ bitów/s} : 8 \text{ bitów/znak}} \approx 1,7 \text{ s.}$$

Gdy czas wykonania wynosi kilka czy nawet kilkadziesiąt milisekund na jeden proces o tym charakterze oraz liczba procesów jest ograniczona do kilkunastu, wówczas nie powinno być trudności w uniknięciu gubienia znaków z powodu przepełnienia wejściowych buforów cyklicznych.

Znacznie trudniejszy do spełnienia, jak i do badania, jest warunek, aby nie dopuścić do zajęcia zbyt dużej liczby buforów z puli, jaką dysponuje się w węźle na ramki wysyłane z tego węzła. Ocena wpływu gospodarki buforami na charakterystyki użytkowe węzła będzie jednym z tematów prac badawczych, jakie są przewidywane do przeprowadzenia dla sieci MSK.

W węźle należy także wykonywać pewne procesy o charakterze pomocniczym, np. przetwarzanie i rejestrację pomiarów i diagnostyki, dla których nie narzuca się trudnych uwarunkowań czasowych, a które powinny wywierać możliwie jak najmniejszy wpływ na czas wykonywania procesów podstawowych.

3. KOORDYNATOR PROCESÓW PODSTAWOWYCH

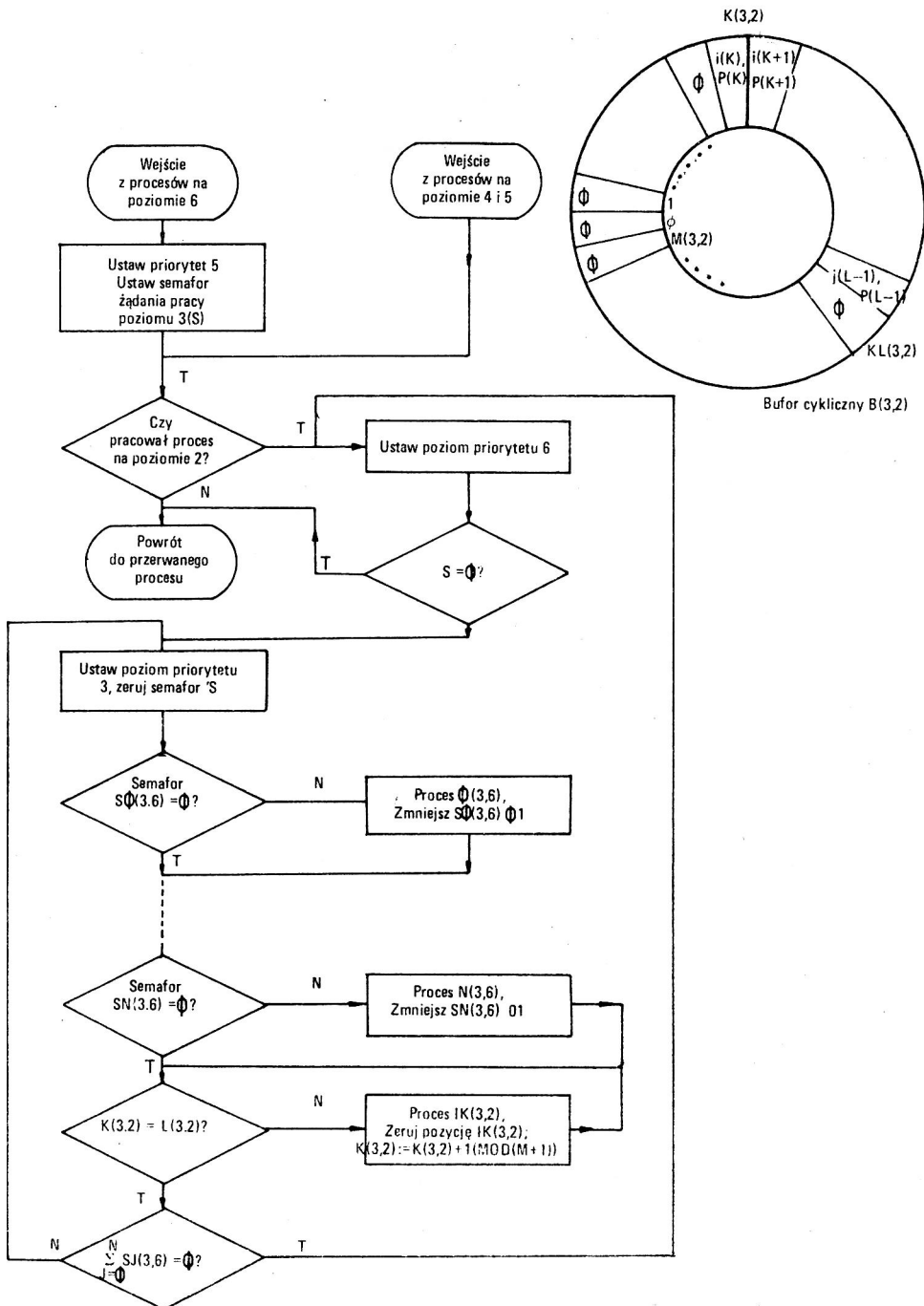
Koordinator procesów podstawowych (rys. 1) zaprojektowano w taki sposób, żeby procesy najbardziej pilne miały najwyższy priorytet bezwzględny, a synchronizacja różnych procesów powodowała możliwie jak najkrótszy czas niedostępności jednostki centralnej dla procesów najbardziej pilnych.

Na rysunku 1 pominięto procesy pracujące na najwyższym 7. poziomie priorytetu w minikomputerze SM-3. Ten poziom priorytetu jest bowiem zastrzeżony dla procesów wyjątkowych, takich jak na przykład inicjacja pracy węzła po włączeniu zasilania, czy obsługa sygnału zaniku zasilania.

Procesy pracujące na poziomie priorytetu 4÷6 są inicjowane przez zdarzenia sygnalizowane przerwaniem zewnętrznymi lub programowymi. Poziom priorytetu 6. jest przeznaczony dla najpilniejszych procesów (obsługa przerw od adapterów liniowych i zegara wewnętrznego).

Poziomy priorytetu 4. i 5. zarezerwowano dla procesów obsługi urządzeń zewnętrznych zgodnie ze standardowymi ustaleniami producenta minikomputerów SM-3. Do procesów analizy, przetwarzania i kolejkowania ramek oraz pakietów przeznaczono poziom priorytetu 3.

Podczas projektowania synchronizacji procesów najpilniejszych i mniej pilnych procesów podstawowych można było przyjąć założenie, że procesy najbardziej pilne (na poziomie 6.) inicjują pracę procesów mniej pilnych (na poziomie 3.), natomiast bezpośrednie inicjowanie procesów w kierunku odwrotnym nie jest potrzebne. Wynika to z charakteru zastosowania, w którym strumień zgłoszeń przerw z adapterów liniowych i zegara ma synchronizować pracę węzła. Dla procesów pomocniczych zastrzeżono poziom priorytetu 2. Do synchronizacji procesów na poziomie 3. przez procesy na poziomie 6. zastosowano zmodyfikowane semafory Dijkstry [3, 6, 7, 9].



Rys.1. Schemat ogólny koordynatora procesów podstawowych
 Fig.1. Schematic Diagram of Coordinator of Primary Processes

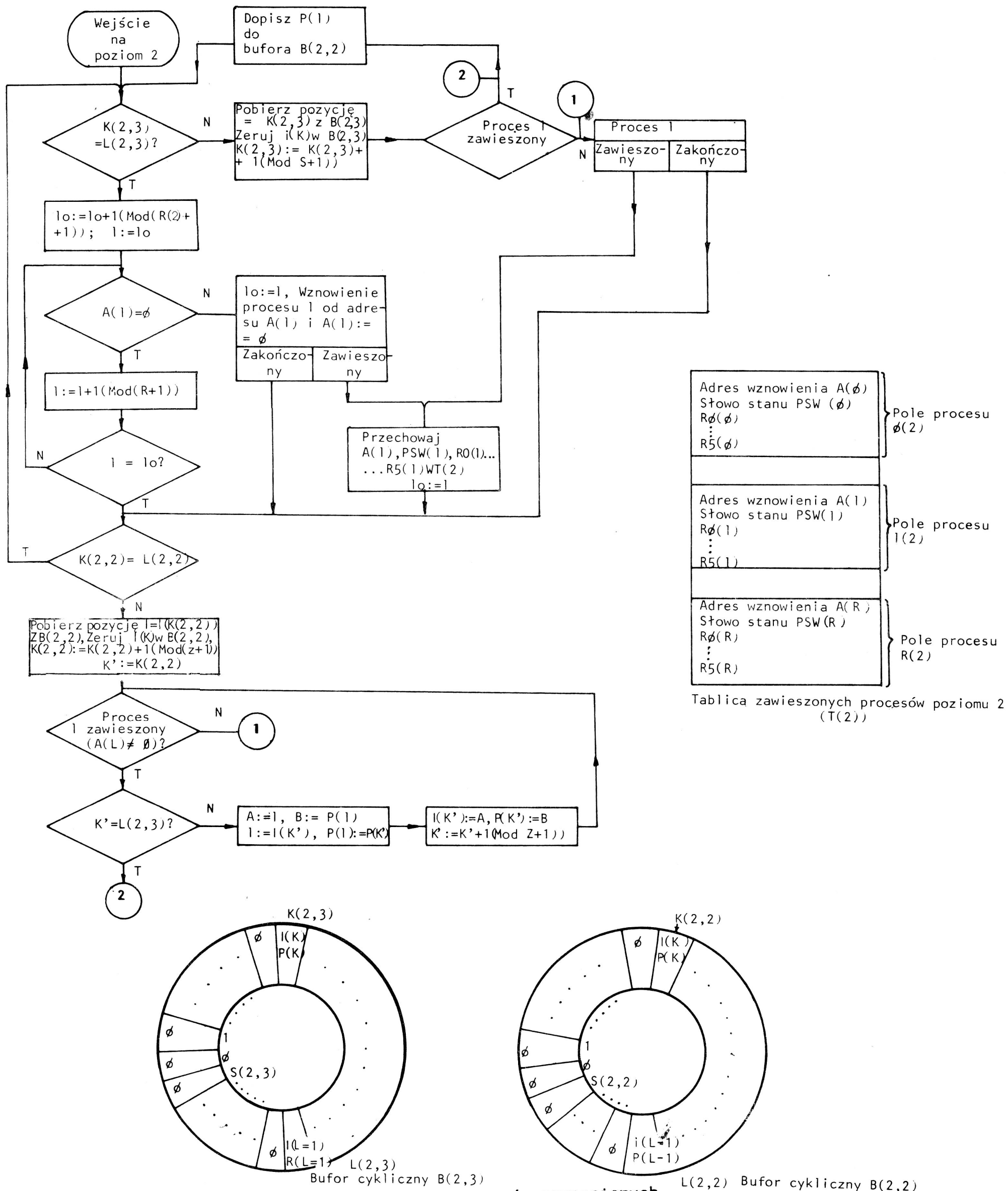
Jeśli dowolny proces na poziomie 6., zainicjowany przez sygnał przerwania zewnętrznego stwierdzi, że zdarzenie sygnalizowane przez ten sygnał wymaga zainicjowania procesu $J(3,6)$ na poziomie 3. (jak np. zakończenie transmisji ramki lub odliczenie 50 impulsów zegara wewnętrznego, tj. odmierzenie 1 s), to zwiększy o 1 zawartość semafora licznikowego $SJ(3,6)$ ($J \in \{\emptyset, 1, \dots, N\}$) (czyli wykona operację $V(SJ(3,6))$) według terminologii Dijkstry i przekaże sterowanie do koordynatora. Ten ustawia priorytet 5., umożliwiając w ten sposób dostęp do jednostki centralnej dla procesów na poziomie 6., i ustawia dwustanowy semafor S ządania pracy poziomu 3., wykonując na nim funkcję $V(S)$. Operacja ta jest wykonywana przez wielowejściowy (re-entrant) segment koordynatora, gdyż może być przerwana i ponownie uruchomiona przez proces na poziomie 6. Predykat badający czy pracował proces na poziomie 2. (przez sprawdzenie słowa stanu procesora przechowywanego na szczycie stosu, gdzie zostało ono umieszczone w wyniku odebrania ostatniego sygnału przerwania przez jednostkę centralną) wyznacza, czy należy sprawdzać potrzebę zainicjowania procesu na poziomie 3. Jeśli nie, to należy oddać sterowanie do ostatniego przerwane procesu. Jeśli tak, to rozpoczyna się wykonywanie funkcji P na semaforze S przez sprawdzenie czy $S=0$. Sprawdza się na poziomie priorytetu 6., aby uniemożliwić jego przerwanie w celu wykonania procesu, który mógłby wykonać znów funkcję $V(S)$. Mogłoby to bowiem doprowadzić do tego, że sterowanie zostałoby oddane do przerwane procesu na poziomie 2., mimo ustawionego semafora S .

Gdy jest ustawiony semafor S , to następuje przejście na poziom 3. i rozpoczęcie wykonywania funkcji V kolejno na semaforach $S\emptyset(3,6), \dots, SN(3,6)$. Jeśli semafor licznikowy $Sk(3,6)$, $k \in \{\emptyset, \dots, N\}$ jest niezerowy, to zostanie uruchomiony proces $k(3,6)$, który zmniejsza też zawartość semafora $Sk(3,6)$.

W rzeczywistości dwa procesy w węźle (zadania $Z\emptyset P36$ i $Z2P36$ [1]) wykorzystują jeszcze następny poziom semaforów licznikowych, które określają liczbę zdarzeń inicjujących dla poszczególnych łącz.

Regulamin obsługi semaforów $Sk(3,6)$, $k \in \{\emptyset, \dots, N\}$ przyjęto w taki sposób, aby zapewnić możliwie sprawiedliwy podział czasu jednostki centralnej pomiędzy poszczególne procesy i nie gromadzić nadmiernej liczby pakietów w węźle.

Procesy na poziomie 3. mogą być inicjowane także przez procesy pomocnicze wykonywane na poziomie 2. Wtedy ograniczenia czasowe nie są tak bardzo istotne, jak dla procesów na poziomie 6. i można zastosować bardziej złożony mechanizm synchronizacji. Do komunikacji między procesami na poziomie 2. a procesami na poziomie 3. wykorzystuje się bufor cykliczny $B(3,2)$. Bufor ten jest wypełniany przez proces ojca na poziomie



Rys.2. Koordynator procesów pomocniczych
Fig.2. Coordinator of Secondary Processes

mie 2., który steruje wskaźnikiem $L(3,2)$ pierwszej wolnej pozycji, a opróżniany (zerowany) przez koordynator wywołujący proces dziecka na poziomie 3. Koordynator steruje też wskaźnikiem $K(3,2)$, ustawiając go na następną pozycję bufora cyklicznego $B(3,2)$ po wykonaniu procesu dziecka. Proces ojca może przekazać procesowi dziecka jednosłowny parametr w buforze $B(3,2)$. Wskaźnikiem przepełnienia bufora $B(3,2)$ jest niezerowa pozycja, na którą proces ojca próbuje wpisywać wywołanie procesu dziecka.

Podczas komunikacji przez bufor $B(3,2)$ nie jest możliwe jego przepełnienie, gdyż podprogram ładujący go na żądanie procesu ojca przekazuje sterowanie na poziom 3. i proces dziecka zostanie wykonany przed wznowieniem pracy procesu ojca. Zastosowano jednak bufor cykliczny, aby w ten sposób umożliwić w przyszłości uaktywnianie procesów na poziomie 3. przez procesy na innych poziomach.

Jeśli $K(3,2) = L(3,2)$, to następuje sprawdzenie czy któryś z semaforów $S_i(3,6)$, $i \in 0, \dots, N$ jest niezerowy. Jeśli tak, to zostanie ponownie przebyta droga badania semaforów $S_i(3,6)$, $i \in 0, \dots, N$. Jeśli nie, to rozpocznie się wykonanie funkcji $V(S)$ na poziomie priorytetu 6.

Zastosowany dwupoziomowy mechanizm semaforów (a w rzeczywistości nawet trzypoziomowy dla pewnych procesów $i(3,6)$) pozwala na łatwe wywoływanie procesów dzieci na poziomie 3. przez procesy ojców na poziomie 6. Wymaga się bowiem tylko dodania jedynki do jednego lub dwóch semaforów. Jednocześnie zakaz przerwania poziomu 6. jest ustawiony przez koordynator wyłącznie na czas wykonania instrukcji testowania semafora S . Żadne wywołanie procesu $i(3,6)$, $i \in \{0, \dots, 1\}$ nie zostanie zgubione, mimo że funkcje P i V nie są chronione przed przerwaniem. Możliwy jest tylko pojedynczy jałowy przebieg przez kilka instrukcji pętli głównej koordynatora procesów podstawowych, jeśli zostanie wykonana funkcja $P(S)$ po wyzerowaniu semafora S (tj. zakończeniu funkcji $V(S)$), a przed ponownym rozpoczęciem wykonywania funkcji $V(S)$ (na poziomie priorytetu 6.).

4. KOORDYNATOR PROCESÓW POMOCNICZYCH

Do wykonywania procesów pomocniczych w węźle podsieci komunikacyjnej zastrzeżono poziom priorytetu 2. Część koordynatora odpowiedzialna za sterowanie procesów pomocniczych (rys. 2) ma charakter bardziej uniwersalny niż część koordynująca procesy podstawowe.

Na poziomie 2. może pracować pewna liczba procesów (do 7 w pierwszej wersji węzła). Procesy te mogą być wywoływane przez procesy na poziomie 3. (np. cykliczne przez proces zegara) lub 4. (np. przez pro-

ces obsługi klawiatury monitora ekranowego). Do synchronizacji jest wykorzystywany bufor cykliczny B(2,3) na zasadach podobnych do opisanych dla bufora B(3,2). Jednak w tym przypadku nie jest wykluczone przepełnienie bufora i podprogram uaktywniania procesów poziomu 2. przez procesy poziomu 3. lub 4. zawiera procedurę wyjątkową kasowania zaległych pozycji w buforze B(2,3).

Proces na poziomie 2. może być wykonany, jeśli ten sam proces nie został wcześniej zawieszony (na własne żądanie). Żądanie zawieszenia procesu i polega na wywołaniu koordynatora (instrukcja skoku do podprogramu), który umieszcza adres wznowienia, słowo stanu i rejestry procesu i na i-tej pozycji tablicy zawieszonych procesów na poziomie 2. Wznowienie zawieszonego procesu na poziomie 2. nastąpi wtedy, gdy koordynator procesów pomocniczych stwierdzi, że żaden inny proces na poziomie 2. nie żąda czasu jednostki centralnej.

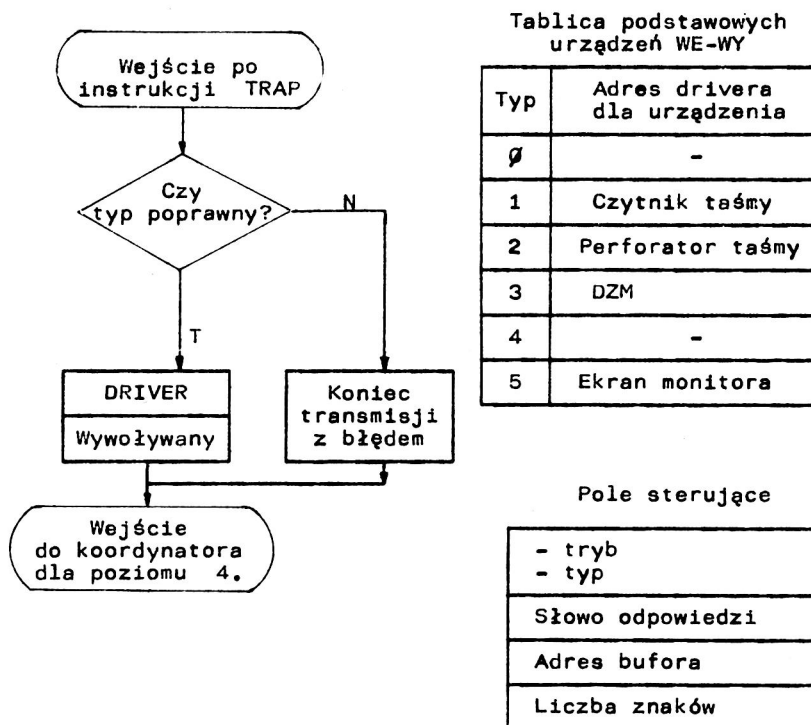
Procesy na poziomie 2. mogą tworzyć łańcuchy, wywołując inne procesy na tym samym poziomie. Proces ojca wywołuje proces dziecka dopisując (za pomocą odpowiedniego podprogramu koordynatora) pozycję L(2,2) do bufora B(2,2) na zasadach podobnych do omówionych wcześniej dla innych buforów cyklicznych. Przykładem łańcucha procesów na poziomie 2. jest wywoływanie przez proces obsługi operatora węzła innego procesu na poziomie 2., z przekazaniem jednosłowego parametru do procesu dziecka.

5. KOORDYNATOR DRIVERÓW PODSTAWOWYCH URZĄDZEŃ WE/WY

Procesy na poziomie 2. mogą żądać transmisji z/na podstawowe urządzenia zewnętrzne. W tym celu wykorzystuje się instrukcję przerwania programowego TRAP $\emptyset$. W rejestrze zerowym proces żądający zainicjowania transmisji podaje adres pola sterującego (rys. 3).

Jeśli żądanie jest prawidłowe, to koordynator uruchomi odpowiedni driver od adresu z tablicy podstawowych urządzeń WE/WY. Driver sygnalizuje zakończenie transmisji ustawiając słowo odpowiedzi. Jeśli wywołanie było błędne, to koordynator odda sterowanie do procesu żądającego transmisji, przekazując mu w słowie odpowiedzi informację o zakończeniu transmisji z błędem.

Powrót do procesu, który wysłał żądanie transmisji następuje przez koordynator według adresu i słowa stanu, zapamiętanych na stosie podczas wykonywania instrukcji TRAP $\emptyset$. Koordynator zapamiętuje także, jak i w razie przerwania zewnętrznych, odtworzenie zawartości rejestrów dla procesu żądającego transmisji.



Rys.3. Koordynator driverów podstawowych urządzeń WE/WY

Fig.3. Coordinator of Basic I/O Peripheral Drivers

6. UWAGI KOŃCOWE

Koordynator węzła podsięci komunikacyjnej, tworzonej dla sieci MSK, stanowi jądro prostego systemu operacyjnego dla minikomputera SM-3, u-kierunkowanego na zastosowanie sieciowe. Choć pierwsze wersje węzła będą posiadać koordynator dla procesów rezydujących w pamięci operacyjnej, to jednak przewidziano możliwość łatwej rozbudowy tego koordynatora i przekształcenie go w nakładkowy, dyskowy system operacyjny. W tym przypadku procesy pracujące na poziomie 2., dla których uwarunkowania czasowe nie są zbyt ostre, rezydowałyby w bibliotece na dyskach i byłyby wprowadzane do pamięci operacyjnej wówczas, kiedy byłyby potrzebne.

Przedstawiona wersja koordynatora umożliwia nadążną pracę węzła w warunkach, jakie są przewidywane na pierwszych etapach badań i wdrażania sieci MSK. Przejście na większe prędkości transmisji wymagać będzie

zmiany adapterów liniowych na adaptery z buforowaniem informacji i bezpośrednim dostępem do pamięci oraz, oczywiście, opracowania odpowiednich driverów.

Przedstawiony koordynator pozwala na tworzenie modułowych struktur programowych w celach podobnych do stawianych przed węzłem podsieci komunikacyjnej. Zwłaszcza udogodnienia dostępne dla procesów pracujących na poziomie 2. mają charakter uniwersalny, podobny do udogodnień zapewnianych przez inne systemy operacyjne czasu rzeczywistego.

LITERATURA

- [1] E c k h o u s e R.H.jr., Systemy minikomputerowe; organizacja i oprogramowanie, WNT, Warszawa 1979.
- [2] L e w o c J., Struktura oprogramowania węzła podsieci komunikacyjnej (w niniejszym zeszycie).
- [3] D a v i e s D.W., B a r b e r D.L.A., Sieci teleinformatyczne, WNT, Warszawa 1979.
- [4] CCICT, Proposed revision to level 1 and 2 of Recommendation X.25, COM VII - No 374-E, październik 1979.
- [5] B a t y c k i T. i in., Adapter liniowy AL11, dokumentacja techniczna, Raport ICT PWr., seria SPR nr 4/79, Wrocław 1979.
- [6] H a n s e n P.B., Podstawy systemów operacyjnych, WNT, Warszawa 1979.
- [7] S h a w A.C., Projektowanie logiczne systemów operacyjnych, WNT, Warszawa 1980.
- [8] CCICT, Revised Recommendation X.25, preface and level 3, COM VII - No 384-E, sierpień 1979.
- [9] D i j k s t r a E.W., Umiejętność Programowania, WNT, Warszawa 1978.

Praca wpłynęła do Redakcji 25.11.1980 r.

Po poprawieniu 3.03.1982 r.

COMMUNICATION SUBNETWORK NODE COORDINATOR

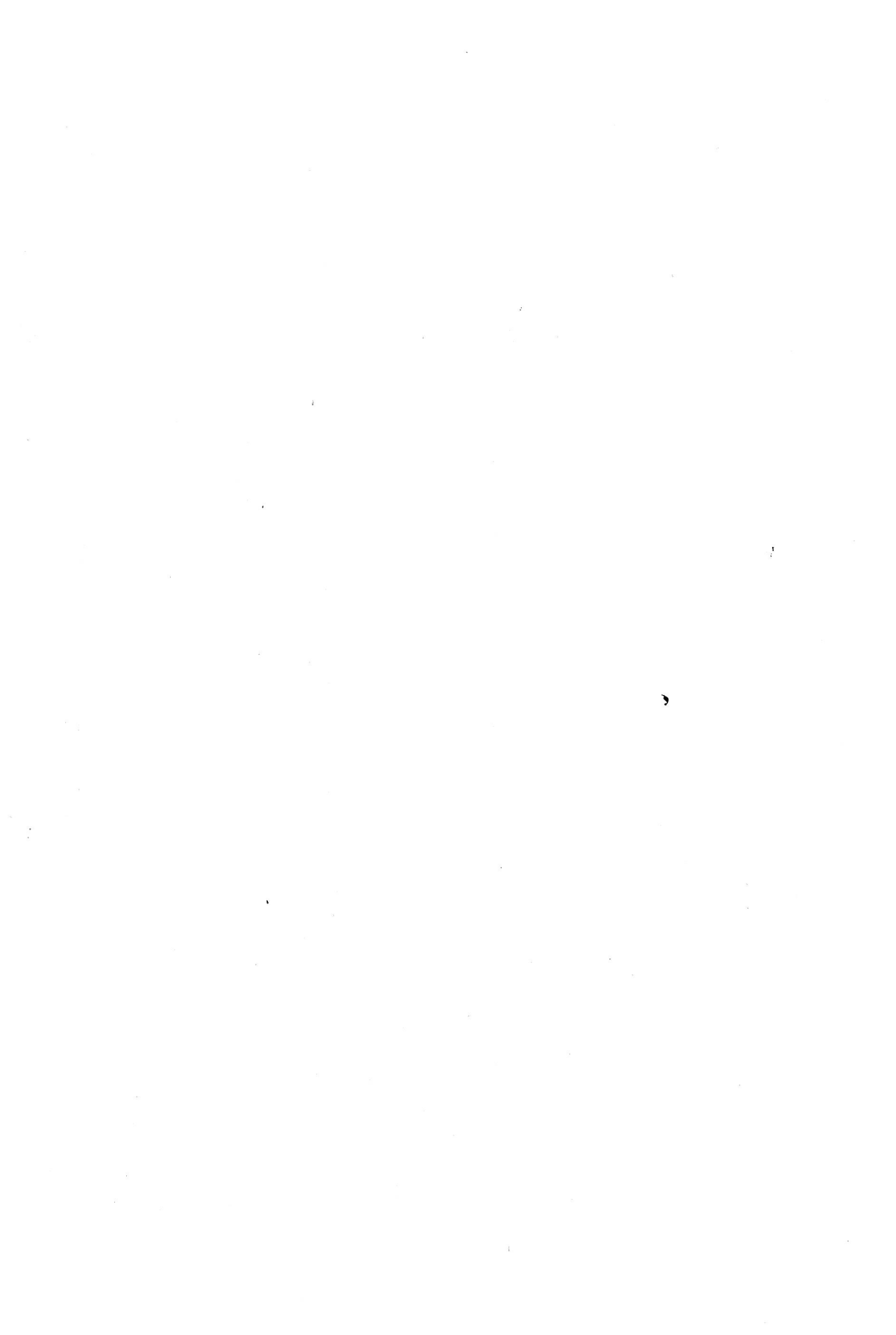
The paper presents a coordinator (monitor) of communication subnetwork node for the MSK computer network. Time limitations for various processes to be performed in the node are discussed. The way of synchronization of various processes realized by the coordinator and justified by the time limitations is presented. For synchronization of processes, the modified Dijkstra's semaphores and cyclic buffers are used.

Verified by Ruta Czaplińska

КООРДИНАТОР УЗЛА ПОДСЕТИ СВЯЗИ

В статье рассматривается координатор узла подсети связи в сети MSK. Описываются временные обусловленности для разных процессов, выполняемых в узле. Приведен, выполняемый координатором, метод синхронизации процессов, обоснованный временными условиями. Для синхронизации процессов используются модифицированные семафоры Дijkstra и циклические буфера.

Проверила Малгожата Хейдрих



*Podsieć komunikacyjna, restart automatyczny,
przerwanie, inicjacja pracy węzła*

Barbara ŁANOWSKA*)

Inicjacja pracy i restart węzła podsieci komunikacyjnej

Omówiono sprzętowy układ automatycznego restartu minikomputera SM-3 i obsługujące go programy w systemie operacyjnym węzła. Przedstawiono zasadę jego pracy i sposób inicjowania systemu operacyjnego węzła podczas wykonania automatycznego restartu.

1. WSTĘP

Dla systemów pracujących w czasie rzeczywistym ręczne uruchamianie systemu po wystąpieniu nieprawidłowości powodujących jego niewłaściwą pracę jest nie do przyjęcia. Wymaga bowiem każdorazowo interwencji operatora oraz zwiększa czas nieoperatywności systemu. Aby zwiększyć operatywność systemu należy wprowadzić automatyczny restart, który jednocześnie pozwoli na ograniczenie operatorskiego nadzoru.

W najprostszej wersji automatyczny restart zapewnia uruchomienie systemu od początku, w wersji bardziej złożonej pozwala również na wykorzystanie danych, wcześniej zachowanych w tym celu [3]. Ze względu na brak w chwili obecnej rozeznania, jakie dane węzła będą niezbędne i możliwe do przechowywania i wykorzystania po restarcie, opracowano wersję najprostszą, polegającą na automatycznej inicjacji pracy systemu w zadanych warunkach początkowych.

*) Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

Automatyczny restart systemu operacyjnego węzła zawiera dwa elementy: sprzętowy układ restartu i obsługujący go program. Oba te elementy zostaną opisane poniżej.

2. ZASADA PRACY AUTOMATYCZNEGO RESTARTU

Działanie automatycznego restartu opiera się na pobudzaniu sprzętowego układu restartu impulsami generowanymi programowo co określony stały odcinek czasu. Impulsy te świadczą o poprawnej pracy systemu operacyjnego. Brak pobudzenia sprzętowego układu automatycznego restartu po upływie kolejnego odcinka czasu powoduje generację przerwania wewnętrznego w procesorze i wywołanie podprogramu obsługi. Zadaniem tego podprogramu jest przesłanie z pamięci dyskowej do pamięci operacyjnej kopii systemu operacyjnego węzła i zainicjowanie jego pracy.

2.1. Układ automatycznego restartu

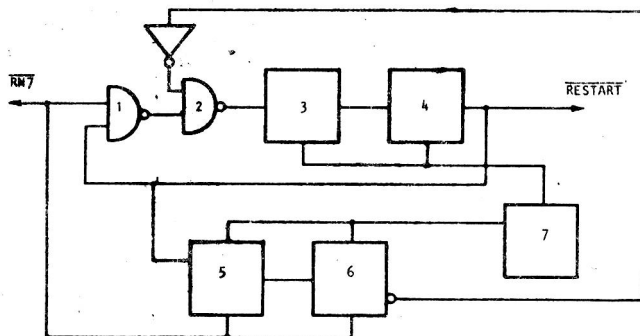
Zaprojektowany układ restartu ma postać dodatkowego pakietu umieszczonego w panelu procesora SM-3. Do pobudzania układu wykorzystano sygnał powstający po zdekodowaniu nielegalnego kodu operacji 7. W razie braku sygnału pobudzającego po zadanym odcinku czasu, co świadczy o nieprawidłowym działaniu systemu operacyjnego, układ generuje sygnał RESTART, który wywołuje przerwanie wewnętrzne i przejście do wykonywania programu, którego adres jest umieszczony w komórce 024. Układ jest wyposażony w licznik nieudanych restartów, który po wykonaniu bez powodzenia n restartów zatrzymuje pracę systemu sygnalizując awarię.

Układ skonstruowano w ten sposób, by włączał się automatycznie do pracy po wykryciu pierwszego rozkazu o kodzie 7. Rozwiązanie takie pozwala na wykonywanie przez procesor bez jakichkolwiek interwencji operatora również programów nie używających układu automatycznego restartu. W takich przypadkach układ nie jest włączany (brak rozkazu o kodzie 7).

Schemat blokowy układu przedstawiono na rys. 1. Składa się on z:

- przerzutnika monostabilnego odmierzającego czas pomiędzy dwoma kolejnymi pobudzeniami (3),
- przerzutnika monostabilnego generującego sygnał przerwania (4),
- układu ustawiania stanu początkowego (7),
- 3-bitowego licznika nieudanych restartów (5),
- przerzutnika blokady (6).

Po załączeniu napięcia zasilającego w procesorze układ ustawiania stanu początkowego (7) generuje pojedynczy impuls wynuszający za pośrednictwem przerzutnika blokady (6) stan, który zapobiega pobudzeniu prze-



Rys.1. Układ automatycznego restartu

Fig.1. Automatic restart device schematic diagram

przerzutnika monostabilnego odmierzającego czas. W ten sposób procesor może wykonywać programy, które nie posiadają procedur obsługi układu automatycznego restartu.

Układ rozpoczyna pracę po pobudzeniu pierwszym wygenerowanym programowo (po wykonaniu rozkazu nielegalnego) impulsem $\overline{RN7}$. Impuls ten zeruje licznik nieudanych restartów i zmienia stan przerzutnika blokady, pobudza przerzutnik monostabilny 3, który zaczyna generować impuls o długości nieco większej niż okres pobudzania układu restartu. Przerzutnik 3 ma możliwość ponownego wyzwolenia będąc w stanie generacji impulsu.

Jeżeli więc impulsy wyzwajające będą pojawiały się w regularnych odstępach czasu, to przerzutnik monostabilny 3 będzie zawsze w stanie generowania impulsu. Jeżeli impuls pobudzający nie pojawi się, to przerzutnik monostabilny 3 zakończy generowanie impulsu wyzwajając przerzutnik monostabilny 4. Przerzutnik ten generuje sygnał $\overline{RESTART}$ wywołujący w procesorze przerwanie wewnętrzne. Jednocześnie jest dodawana jedynka do licznika nieudanych restartów i pobudzany jest ponownie przerzutnik monostabilny 3. Jeśli restart uda się, to po przesłaniu do pamięci procesora kopii systemu operacyjnego zaczną pojawiać się generowane programowo impulsy $\overline{RN7}$, podtrzymując działanie układu. Wyzerowany zostanie licznik nieudanych restartów. Z podanej zasady wynika dobór czasu trwania impulsu generowanego przez przerzutnik monostabilny 3. Musi on być mianowicie nieco większy od sumy czasów transmisji kopii systemu do pamięci operacyjnej i inicjacji pracy systemu.

Jeśli restart nie doprowadzi do wznowienia pracy systemu operacyjnego, to pobudzanie kolejno przerzutników monostabilnych 4 i 3 będzie odbywać się do czasu, gdy przepełnienie licznika nieudanych restartów 5

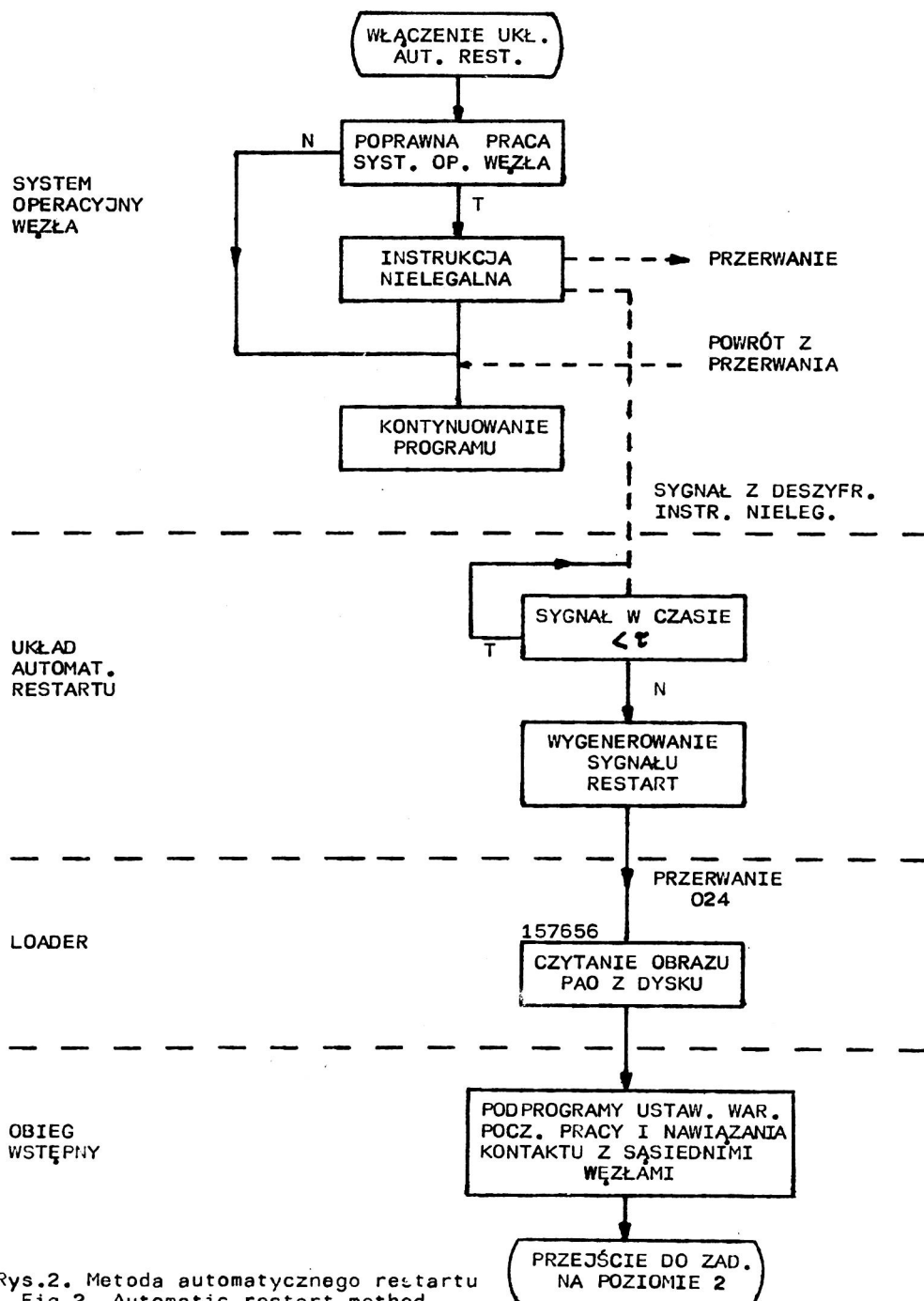
spowoduje zmianę stanu przerzutnika blokady 6, co sygnalizuje trwałe uszkodzenie procesora.

3. ODNOWA PRACY SYSTEMU OPERACYJNEGO WĘZŁA

Przedstawiona na rysunku 2 metoda automatycznego restartu zapewnia rozpoczęcie pracy systemu operacyjnego węzła od warunków początkowych w razie jego błędnej pracy.

Zadanie zegara systemu operacyjnego podczas każdego przebiegu, tj. co 1 s, modyfikuje i sprawdza liczniki kontrolne. Liczniki te są odnawiane co ustalony czas przez różne części programów systemu operacyjnego. W razie wyzerowania dowolnego z liczników uważa się, że system nie pracuje poprawnie. Wówczas nie wykonuje się rozkazu nielegalnego, generującego impuls RN7 pobudzający układ automatycznego restartu. Spowoduje to zadziałanie układu automatycznego restartu i pojawienie się sygnału RESTART. Sygnał ten powoduje przerwanie z adresem wektora przerwania 024. Jest to takie samo przerwanie jak podczas zaniku i ponownego pojawienia się napięć. Dlatego też przed rozpoczęciem transmisji dyskowej należy zaczekać około 30 ms, co odpowiada czasowi ustalania się napięcia stabilizowanego od momentu włączenia sieci. Po zainicjowaniu transmisji następuje przepisanie z pamięci dyskowej do pamięci operacyjnej kopii systemu operacyjnego węzła. Gdy transmisja jest niepoprawna, wówczas jej powtarzanie może odbywać się do 8 razy.

Po poprawnym przepisaniu systemu rozpoczyna się obieg wstępny. Jest to część programu systemu operacyjnego ustawiająca warunki początkowe. W obiegu wstępnym zostaje wysłana instrukcja nielegalna w celu przekazania układowi restartu sygnału poprawnej pracy oraz zostaje wysłany komunikat o restarcie na drukarkę mozaikową i ekran monitora. Następnie jest wykonywany podprogram inicjowania adapterów liniowych, zapewniający przygotowanie adaptera do odbioru i nadawania danych. Podprogram ten ustawia w rejestrach każdego z adapterów liniowych warunki początkowe pracy, żądając włączenia adaptera. Następnie odmierzany jest czas 1 s. Po tym czasie sprawdzany jest stan każdego z adapterów. Jeżeli stan jest poprawny, to ustawione zostają zezwolenia na przerwanie odbiornika oraz wskaźnik sprawności linii. Gdy brak potwierdzenia, wówczas zostaje ustawione w adapterze zezwolenie na przyjmowanie przerw od modemu oraz do wektora przerw jest wpisany adres programu obsługi przerw od modemu. Wskaźnik sprawności jest wyzerowany. Jeżeli po pewnym czasie adapter po drugiej stronie linii stanie się gotowy do pracy, sam wyśle "wywołanie", co spowoduje przerwanie. Obsługa tego przerwania będzie polegała na wykonaniu takiej sekwencji rozkazów, jak



Rys.2. Metoda automatycznego restartu
Fig.2. Automatic restart method

po otrzymaniu "potwierdzenia" oraz na końcu na zmianie wektora przerwania na wektor właściwy do obsługi przerwań odbiornika w systemie operacyjnym węzła, tj. na wektor związany z driverami adapterów liniowych. Po takim utworzeniu kanałów transmisji dla poszczególnych linii jest wykonywany w obiegu wstępnym podprogram inicjowania opisów kanałów logicznych.

Dalsza część obiegu wstępnego obejmuje wykonanie podprogramu inicjowania puli buforów oraz podprogram inicjowania połączeń linii dla poziomu liniowego HDLC. Następnie ustawiane jest zezwolenie na przerwanie dla zegara i klawiatury monitora ekranowego oraz po ustawieniu poziomu priorytetu 2 przejście do zadania na poziomie 2. System jest gotowy do pracy.

4. UWAGI KOŃCOWE

Zastosowany układ automatycznego restartu jest prosty. Zbudowany jest z czterech układów średniej skali integracji, a jego zastosowanie wymaga zmiany kilku połączeń na pakiecie jednostki centralnej. Zaproponowane rozwiązanie umożliwia ewentualną przyszłą modyfikację oprogramowania polegającą na zapamiętywaniu pewnych informacji dotyczących pracy węzła i wykorzystaniu ich po restarcie. Należy jednak pamiętać, że po zakończeniu transmisji dyskowej trzeba w odpowiednim czasie wysłać sygnał poprawnej pracy do układu restartu, tak aby czas τ był większy od czasu transmisji dyskowej zwiększonego o wykonywaną sekwencję rozkazów obiegu wstępnego do momentu wystąpienia instrukcji nielegalnej 7. Czas trwania transmisji z dysku dla 32 k słów wynosi około 220 ms. Przyjęty czas τ wynosi 1 s.

LITERATURA

- [1] Processor SM 2103.62 Techničeskoe opisanie čast 2, čerteży, schemy, mikroprogramy, nr 3.055.005-OI, TOI, 1978.
- [2] L e w o c J. i in., Zasady przygotowania oprogramowania węzła, Raporty Centr. Oblicz. PWr., Seria SPR nr 14/80, Wrocław 1980.
- [3] L e w o c J. i in., Automatyczna odnowa pracy systemu czasu rzeczywistego dla zestawu ODRA 1325/SMA, Informatyka, nr 6, 1979.

Praca wpłynęła do Redakcji 2.12.1980 r.

Po poprawieniu 3.03.1982 r.

INITIATION OF OPERATION AND RESTART FOR COMMUNICATION
SUBNETWORK NODE

The paper presents a hardware device for automatic restarting of the SM-3 minicomputer as well as service routines for the device in the node operating system. The principle of operation of the device and the way in which the node operating system is initiated when performing automatic restart are presented.

Verified by Ruta Czaplińska

ИНИЦИИРОВАНИЕ И РЕСТАРТ УЗЛА ПОДСЕТИ СВЯЗИ

Рассматривается аппаратная реализация автоматического рестарта МИНИ-ЭВМ SM-3 и обслуживающие ее программы операционной системы узла. Описывается принцип работы и метод инициирования операционной системы узла во время автоматического рестарта.

Проверила Малгожата Хейдрих

*Sieć komputerowa, węzeł, minikomputer,
oprogramowanie, operator*

Andrzej STANISZ^{*)}

Współpraca operator-węzeł w sieci komputerowej MSK

Omówiono funkcje operatora węzła sieci komputerowej. Wskazano czynniki ograniczające zakres funkcji, wynikające z konfiguracji węzła. Podano listę komunikatów operatorskich przyjętą w oprogramowaniu węzła realizowanego na minikomputerze SM-3 dla sieci MSK.

1. WPROWADZENIE

Mimo automatyzacji przepływu informacji w sieci komputerowej wiele funkcji przypada człowiekowi - operatorowi. Jego rola polega na włączaniu i wyłączaniu urządzeń, dysponowaniu zasobami sieci i kontroli jej pracy. Zakres funkcji operatora w znacznym stopniu zależy od przeznaczenia sieci, jej wielkości, dostępnych udogodnień, a w przypadku węzła - także od jego konfiguracji.

W artykule omówiono funkcje operatora węzła sieci komputerowej oraz czynniki decydujące o zakresie i sposobie ich realizacji w konkretnych przypadkach. Na tym tle przedstawiono rozwiązanie przyjęte dla węzła sieci MSK, realizowanego na minikomputerze SM-3.

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

2. FUNKCJE OPERATORA WĘZŁA

Działania operatora w węźle sieci komputerowej są związane z następującymi grupami funkcji.

A. Podstawowe funkcje ruchowe. Obejmują one przygotowanie węzła do pracy (uruchomienie programu węzła, włączenie linii, wprowadzenie aktualnej daty i czasu, modyfikację pewnych parametrów programowych) i działania awaryjne (włączanie i wyłączanie linii, zatrzymanie i restart programu węzła i in.).

B. Ingerencje w funkcjonowanie sieci. W bardziej rozbudowanych sieciach, dostosowanych do dynamicznych zmian, operator może modyfikować strukturę sieci (przez wprowadzanie lub zmiany zasobów sieci, włączanie/wyłączanie użytkowników itp.) oraz jej funkcjonowanie (zmiany udogodnień sieciowych itp.).

C. Funkcje kontrolne. Kontrola pracy węzła sprowadza się do reagowania na błędy sygnalizowane przez program węzła, inicjowania testów diagnostycznych podczas pracy węzła (jeżeli takie testy przewidziano), sprawdzania pewnych parametrów ruchowych węzła i inicjowania wyprowadzania raportów zawierających syntetyczne informacje nagromadzone podczas pracy węzła. Zakres funkcji tej grupy w znacznym stopniu zależy od stopnia rozbudowy warstwy diagnostyki i pomiarów w programie węzła.

Współpraca operatora z węzłem odbywa się z reguły poprzez urządzenie operatorskie (dalekopis, monitor ekranowy) z wykorzystaniem odpowiednio zdefiniowanego zestawu komend i odpowiedzi, stanowiącego - w skrajnym przypadku - specjalny język. Propozycję takiego języka zawiera praca [1]. W opracowaniu systemu komend operatorskich istotną rolę odgrywają takie czynniki, jak: wielkość i możliwości rozbudowy sieci, potrzeby użytkowników, system rozliczeń, doświadczenie projektantów itp.

Istotnym ograniczeniem może być konfiguracja węzła: obsługa dialogu operator-węzeł wymaga bowiem stosunkowo złożonych programów do analizy, kontroli poprawności wprowadzanych komend, a także ich realizacji. Wobec ograniczonej zawsze pamięci operacyjnej komputera w węźle i raczej sporadycznego charakteru pracy tych programów, często może okazać się konieczne stosowanie pamięci zewnętrznej i nakładanie programów. Dlatego w konfiguracjach węzła bez pamięci zewnętrznej zakres funkcji operatora musi być z konieczności ograniczony.

Należy podkreślić, że zakres funkcji operatora powinien być znany już w chwili projektowania struktury programu węzła, a szczególnie danych sieciowych, na których operuje węzeł. Umożliwia to wybór struktur danych, najbardziej odpowiednich zarówno z punktu widzenia przetwarzania

informacji w węźle, jak i łatwości dostępu oraz modyfikacji przez programy współpracy z operatorem.

3. ROZWIĄZANIE DLA WĘZŁA SIECI MSK

W konfiguracji węzła sieci MSK (minikomputer SM-3 z pamięcią operacyjną 28 k słów, monitor ekranowy, drukarka mozaikowa, czytnik i perforator taśmy papierowej) w zasadzie nie przewiduje się wykorzystania pamięci zewnętrznej. Wyjątek stanowi użycie pamięci dyskowej jako urządzenia, z którego ładuje się program podczas startu węzła lub automatycznego restartu [3].

Fakt ten, a także prototypowy charakter sieci MSK, zdecydowały o wyborze ograniczonego zestawu funkcji operatora, zapewniającego dostęp do węzła w zakresie niezbędnego minimum, które obejmuje:

- startowanie i restartowanie węzła,
- wprowadzanie aktualnej daty i czasu,
- odczyt wartości danych pomiarowo-diagnostycznych zbieranych na bieżąco (liczniki błędów różnego typu, uśrednione wartości niektórych parametrów ruchowych itp. [5]),
- wyprowadzanie raportów cyklicznych i na żądanie operatora z możliwością wyboru urządzenia (drukarka, perforator taśmy, monitor ekranowy) i wstrzymanie raportowania,
- modyfikację niektórych parametrów węzła (tylko dla personelu uprawnionego).

Przyjęto, że kontakt operatora z węzłem odbywać się będzie przez monitor ekranowy z wykorzystaniem prostego systemu komunikatów.

3.1. Lista komunikatów

Poniżej opisano komunikaty operatorskie dostępne w realizowanej wersji oprogramowania węzła. W podanych opisach parametry liczbowe są liczbami ósemkowymi, najwyżej 6-cyfrowymi, natomiast ∇ , $\langle \text{LF} \rangle$ i $\langle \text{CR} \rangle$ oznaczają odpowiednio klawisze spacji, zmiany wiersza i powrotu karetki.

Wprowadzanie/wyprowadzanie daty (DA). Aktualną datę wprowadza się komunikatem

DA ∇ dd-mmm-rr $\langle \text{CR} \rangle$,

gdzie: dd - dzień miesiąca, mmm - miesiąc (początkowe 3 litery nazwy miesiąca w języku polskim), rr - rok (dwie ostatnie cyfry). Komunikat o postaci

DA $\langle \text{CR} \rangle$

powoduje wyprowadzenie aktualnie ustawionej daty.

Wprowadzanie/wyprowadzanie czasu (TI). Aktualny czas wprowadza się komunikatem

TI ▽ gg:mm:ss <CR> ,

gdzie: gg - godzina, mm - minuty, ss - sekundy. Komunikat o postaci

TI <CR>

powoduje wyprowadzenie aktualnego czasu.

Zmiana uprawnień (LO, FI). Dostęp operatora do niektórych funkcji możliwy jest dopiero po wprowadzeniu hasła komunikatem o postaci:

LO hasło <CR> ,

gdzie hasło jest ciągiem znaków w kodzie ASCII, ustalonym w chwili generowania oprogramowania węzła. Komunikat

FI <CR>

przywraca pierwotny stan programu i blokadę zastrzeżonych funkcji.

Wyprowadzanie/zmiana zawartości komórki (MO). W normalnym stanie uprawnień (przed wprowadzeniem komunikatu LO) operator może zażądać wyprowadzenia zawartości ciągu komórek rozpoczynającego się od podanego adresu, wprowadzając komunikat MO, a następnie znaki <LF> i <CR> w sekwencji jak niżej (informacje wyprowadzane przez program podkreślono):

MO ▽ ad1 <CR>

ad1/zaw1 <LF>

ad2/zaw2 <LF>

.....

adn/zawn <CR>

gdzie ad1 jest adresem pierwszej komórki do wyprowadzenia, zaw1 - jej zawartością, zaś ad2, zaw2, ..., adn, zawn oznaczają odpowiednio adresy i zawartości kolejnych słów wyprowadzanego obszaru.

W stanie rozszerzonych uprawnień (po wprowadzeniu komunikatu LO) operator może zmienić zawartość komórki w odpowiedzi na wyprowadzenie dotychczasowej, jak pokazuje sekwencja:

MO ▽ ad1 <CR>

ad1/zaw1: ▽ nazw1 <LF>

ad2/zaw2: ▽ nazw2 <LF>

.....

adn/zawn: ▽ nazwn <CR> ,

gdzie: ad1, zaw1, ..., adn, zawn mają takie samo znaczenie jak poprzednio, a nazw1, ..., nazwn oznaczają nowe zawartości komórek. Pominięcie nowej zawartości (tj. wprowadzenie znaku <LF> lub <CR> bezpośrednio po wyprowadzanej dotychczasowej zawartości) spowoduje, że zawartość odpowiedniej komórki pozostanie bez zmian.

W obu postaciach komunikatu znak <LF> jest traktowany jako żądanie przejścia do następnej komórki, a <CR> kończy sekwencję komunikatu.

Uaktywnienie zadania użytkowego (ME). Operator może uaktywnić zadanie użytkowe (zadanie z poziomu 2 w ogólnej strukturze programu węzła [2]) i przekazać parametry, określające bliżej żądane działanie zadania, wprowadzając komunikat

ME_q nrz, p1, p2, p3, p4 <CR>,

gdzie nrz - numer zadania użytkowego, a p1, p2, p3, p4 - parametry dla zadania.

Wybór urządzenia wyjściowego (AS). Operator może przyporządkować zadaniu użytkowemu urządzenie wyjściowe, na które ma być wyprowadzana informacja z tego zadania, przez komunikat

AS_q nrz, nru <CR>

gdzie: nrz - numer zadania, nru - numer urządzenia (0 - żadne, 2 - perforator, 3 - drukarka, 5 - monitor ekranowy). Komunikat powoduje wpisanie odpowiedniego kodu do tablicy dostępnej dla zadania użytkowego, ale jej wykorzystanie jest fakultatywne: zadanie może organizować wyprowadzanie zgodnie z tą informacją lub ją ignorować.

3.2. Realizacja programowa

W przyjętej strukturze programu węzła [2] monitor ekranowy jest wyróżniony jako urządzenie operatorskie. Wszystkie informacje wprowadzane z klawiatury są kompletowane w buforze przez odpowiedni driver [3], a następnie - po wykryciu symbolu końca komunikatu (<CR> lub <LF>) - przekazywane (za pośrednictwem koordynatora) do zadania obsługi operatora, będącego jednym z zadań użytkowych. Zadanie to analizuje wprowadzony komunikat pod względem poprawności formatu i możliwości realizacji, a następnie bądź realizuje wprost ten komunikat (dla komunikatów DA, TI, LO, FI, MO, AS), bądź inicjuje odpowiednią akcję przez koordynator (dla komunikatu ME).

3.3. Wykorzystywanie komunikatów podczas pracy węzła

Opisany system komunikatów zapewnia operatorowi dostęp do węzła w niezbędnym zakresie. Po załadowaniu i zainicjowaniu programu węzła (z pulpitu jednostki centralnej) operator wprowadza aktualną datę i czas (komunikaty DA, TI), a następnie - jeśli zachodzi potrzeba - zmienia przyporządkowanie urządzeń wyjściowych (komunikat AS) i ustawia pewne parametry węzła (sekwencja komunikatów LO, MO, FI). W czasie pracy

węzła operator może żądać wyprowadzania różnych raportów i inicjować działania specjalne (komunikaty ME), sprawdzać zawartości wybranych komórek, np. liczników błędów (komunikat MO), a także dokonywać zmian w przyporządkowaniu urządzeń oraz modyfikować parametry (komunikaty AS, LO, MO, FI).

Dostęp do informacji w programie (przez komunikat MO) wymaga - w celu poprawnej interpretacji - znajomości fizycznych adresów i budowy tablic w programie węzła, a także pewnej znajomości jego struktury i funkcjonowania. Dlatego komunikat MO będzie używany przez "zwykłego operatora" raczej sporadycznie, natomiast może być środkiem bardzo przydatnym dla "operatora z rozszerzonymi uprawnieniami", znającego strukturę i funkcjonowanie węzła w czasie uruchamiania węzła w warunkach ruchowych i jego wstępnej eksploatacji w sieci. Konieczność znajomości i wprowadzenia hasła podczas modyfikacji zawartości pamięci stanowi pewne zabezpieczenie przed niepowołanymi zmianami, mogącymi dezorganizować program węzła.

4. UWAGI KOŃCOWE

Przedstawione rozwiązania współpracy operator-węzeł cechuje - mimo prostoty wynikającej z ograniczonej pamięci - pewna uniwersalność, umożliwiająca rozszerzenie funkcji operatora w ramach istniejącej struktury programu (przez rozbudowę zadań użytkowych i odpowiednie definicje parametrów w komunikacie ME), a także zdefiniowanie nowych komunikatów (dzięki uniwersalnej strukturze zadania obsługi operatora).

LITERATURA

- [1] J a r o s ł a w s k a M., Moduł obsługi konwersacji operator-sysem operacyjny węzła-operator, maszynopis IPI PAN, Warszawa 1979.
- [2] L e w o c J., Struktura oprogramowania węzła podsieci komunikacyjnej (w niniejszym zeszycie).
- [3] Ł a n o w s k a B., Inicjacja i restart węzła podsieci komunikacyjnej (w niniejszym zeszycie).
- [4] Ł a n o w s k a B., Oprogramowanie podstawowe urządzeń wejścia-wyjścia w systemie operacyjnym węzła podsieci komunikacyjnej (w niniejszym zeszycie).
- [5] H u d y m a E., K o s m u l s k a - B o c h e n e k E., Edycja rejestrowanych pomiarów w sieci komputerowej ISK (w niniejszym zeszycie).

Praca wpłynęła do Redakcji 7.01.1981 r.

Po poprawieniu 3.03.1982 r.

OPERATOR-COMPUTER COMMUNICATION IN THE NODE
OF THE MSK COMPUTER NETWORK

The functions of the computer network node operator are described. The factors concerning the node computer configuration and limiting the implemented functions are discussed. The list of the operator commands for SM-3 minicomputer to be applied as a node computer in the MSK computer network is given.

Verified by Ruta Czaplińska

СООБЩЕНИЕ ОПЕРАТОРА С УЗЛОМ В СЕТИ ЭВМ MSK

Рассматриваются функции оператора узла сети ЭВМ. Определяются факторы, следующие из конфигурации узла и ограничивающие функции, принимаемые для реализации. Дается описание операторских команд, доступных в программе для мини-ЭВМ SM-3, используемой в качестве узла в сети MSK.

Проверила Малгожата Хейдрих

*Sieć komputerowa, węzeł,
protokół pakietowy*

Zbigniew FRYŻLEWICZ^{*)}

Protokół pakietowy w sieci komputerowej MSK

W artykule opisano protokół poziomego pakietowego zrealizowany w węźle sieci MSK. Przede wszystkim podano:

- typy i parametry implementowanych pakietów protokołu X.25,
- zasadę komunikacji międzywęzłowej,
- strukturę modułu programowego realizującego protokół pakietowy,
- struktury informacyjne wymagane dla podanego modułu programowego.

1. WSTĘP

Architektura w Międzyuczelnianej Sieci Komputerowej (MSK) jest wzorowana na 7-warstwowym modelu Architektury Systemów Otwartych [1]. Realizacja takiego projektu wymaga jednak co najmniej:

- wyboru opcji dopuszczalnych w modelu,
- definicji szczegółów implementacyjnych pomijalnych w modelu, a nieodzownych na etapie realizacji.

W niniejszej pracy przedstawiono protokół 3. warstwy, zwanej warstwą sieci. Jest nim protokół pakietowy znany jako X.25/3. Zalety tego protokołu, uzasadniające jego wybór jako obowiązujący w MSK, przedstawiono w pracy [4]. Implementacja tego protokołu wymaga zdefiniowania szczegółów, które można określić mianem cech zewnętrznych. Najważniejsze z nich to:

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

- typy zestawianych połączeń,
- podział i gospodarka kanałami,
- typy i formaty pakietów,
- zasady adresowania,
- parametry sterowania przepływem.

Wymienione cechy są opisane w rozdz. 2. Ponieważ protokół X.25/3 obowiązuje tylko na styku DTE-DCE, więc aby zestawić i utrzymać połączenie logiczne wewnątrz podsieci komunikacyjnej należy określić dodatkowy protokół międzywęzłowy. Jego zasady opisano w rozdz. 3.

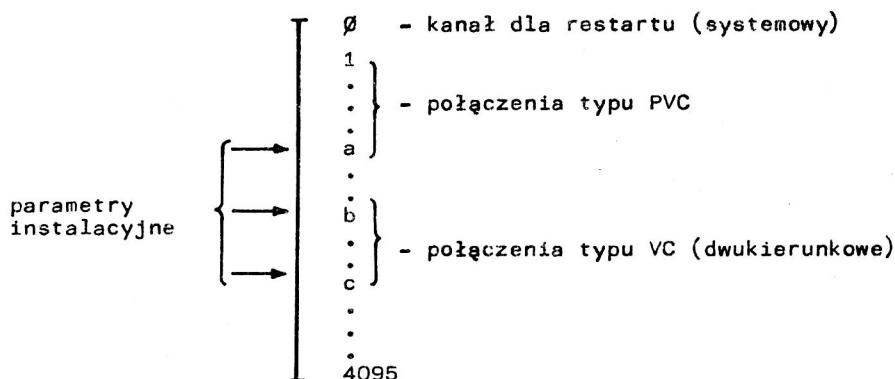
O jakości implementacji protokołu warstwy sieci decyduje także ogólna struktura oprogramowania węzła komunikacyjnego. Wybrane aspekty oprogramowania węzła, tj. gospodarka buforami, powiązania zadań, mechanizmy ich uaktywniania i struktury informacyjne wymagane do realizacji protokołów X.25/3 i protokołu międzywęzłowego opisano w rozdz. 4. Rozdział ten zawiera także opis dekompozycji zadania realizującego omawiane protokoły na submoduły programowe wraz ze schematem ich wzajemnych powiązań.

2. CECHY ZEWNĘTRZNE PROTOKOŁU X.25/3

Protokół pakietowy X.25/3 w sieci MSK zrealizowano na podstawie rekomendacji CCITT [10]. W wielu punktach rekomendacja ta określa tylko ramy dopuszczalnych rozwiązań pozostawiając projektantom oprogramowania węzła zdefiniowanie i wybór określonych opcji. Wybór ten powinien zapewniać użytkownikowi zewnętrznemu stabilną, niezawodną i dostosowaną do jego możliwości odbioru/nadawania transmisję danych z każdym innym elementem sieci. W związku z tym w sieci MSK zrealizowano typy usług określane w rekomendacji jako standardowe z poszerzeniem tego zbioru o realizację pewnych ułatwień. I tak, w omawianej sieci są implementowane połączenia virtualne stałe, tzw. PVC i dynamiczne, tzw. VC. Wykorzystanie kanałów logicznych styku DTE-DCE do realizacji różnych typów połączeń przedstawiono na rys. 1.

W tabeli 1 zestawiono natomiast typy pakietów uważanych za dopuszczalne w zrealizowanej wersji protokołu. Szczegółowe struktury tych pakietów wraz z określeniem przedziałów wartości przenoszonych parametrów podano w pracach [2-7].

Z przytoczonej w tabeli 1 postaci pola GFI łatwo zauważyć, że do kontroli przepływu zastosowano numerację modulo 8. Podsieć komunikacyjna akceptuje pakiety danych o polu informacyjnym nie dłuższym niż 128 oktetów i nie krótszym niż 1 oktet. Podsieć żąda także, aby również odbiorca dysponował możliwością odbioru pakietów danych o podanej



Rys.1. Podział kanałów logicznych styku DTE-DCE
 Fig.1. Division of logical channels of the interface DTE-DCE

T a b e l a 1

Typy pakietów implemmentowane w sieci MSK

Typ pakietu		PTI	GFI
od DCE do DTE	od DTE do DCE		
Zestawianie i zamykanie połączenia			
INCOMING CALL	CALL REQUEST	0 0 0 0 1 0 1 1	0 X 0 1
CALL CONNECTED	CALL ACCEPTED	0 0 0 0 1 1 1 1	0 0 0 1
CLEAR INDICATION	CLEAR REQUEST	0 0 0 1 0 0 1 1	0 0 0 1
DCE CLEAR CONF.	DTE CLEAR CONF.	0 0 0 1 0 1 1 1	0 0 0 1
Dane i przerwanie			
DCE DATA	DTE DATA	X X X X X X X 0	X X 0 1
DCE INTERRUPT	DTE INTERRUPT	0 0 1 0 0 0 1 1	0 0 0 1
DCE INTER. CONF.	DTE INTER. CONF.	0 0 1 0 0 1 1 1	0 0 0 1
Sterowanie przepływem i reinicjacja			
DCE RR	DTE RR	X X X 0 0 0 0 1	0 0 0 1
DCE RNR	DTE RNR	X X X 0 0 1 0 1	0 0 0 1
RESET INDICATION	RESET REQUEST	0 0 0 1 1 0 1 1	0 0 0 1
DCE RESET CONF.	DTE RESET CONF.	0 0 0 1 1 1 1 1	0 0 0 1
Restart			
RESTART INDICATION	RESTART REQUEST	1 1 1 1 1 0 1 1	0 0 0 1
DCE RESTART CONF.	DTE RESTART CONF.	1 1 1 1 1 1 1 1	0 0 0 1

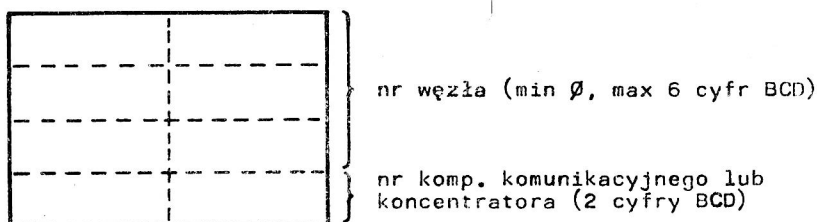
PTI - identyfikator typu pakietu.

GFI - podstawowy identyfikator formatu.

X - bit przyjmujący alternatywnie wartość 0 lub 1.

długości pola informacyjnego. Na etapie nawiązywania połączenia jest natomiast możliwe wynegocjowanie wielkości okna obowiązującej niezależnie dla każdego z dwóch kierunków transmisji styku DTE-DCE. Jeśli pakiety zestawiające połączenie nie zawierają tego typu negocjacji, wielkość okna ma wartość standardową równą dwa.

Ważną cechą protokołu X.25/3 jest stosowany mechanizm adresowania. Z rozważań ogólnych [1] wynika, że winien on odzwierciedlać strukturę samej sieci oraz być na tyle elastyczny, aby nie należało go zmieniać w miarę wzrostu samej sieci. Wydaje się, że w sieci MSK oba powyższe cele zostały spełnione, mimo że z powodów oszczędnościowych ograniczono dopuszczalną w ramach rekomendacji wielkość pola adresowego. Przyjęto, że dowolne z pól adresowych będzie mieć postać przedstawioną na rys. 2.

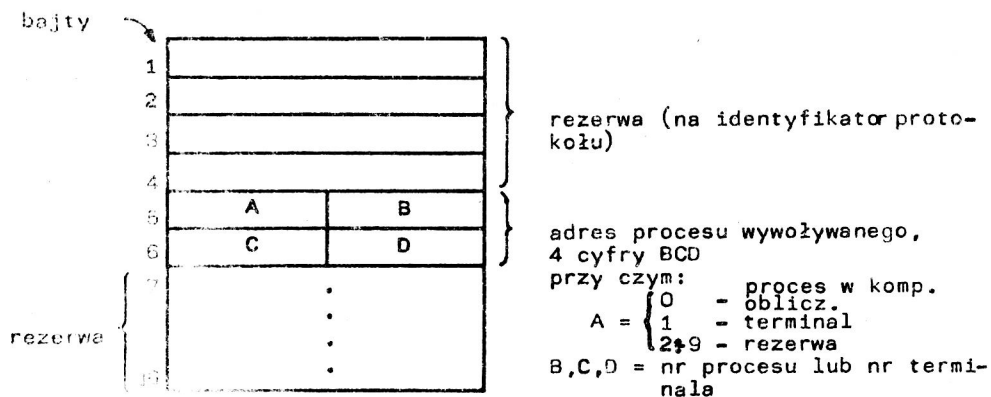


Rys.2. Budowa pola adresowego

Fig.2. Structure of the addressing field

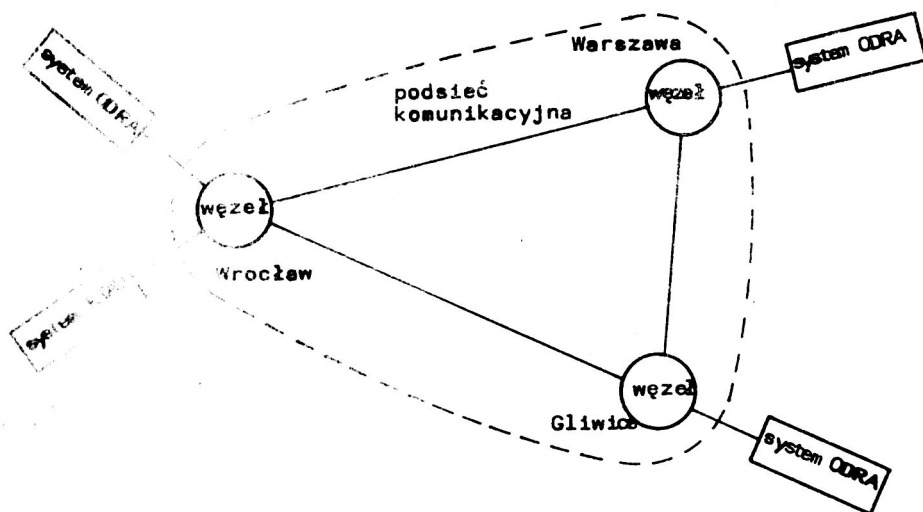
Struktura pola adresowego przedstawiona na rys. 2 umożliwia adresowanie w sieci składającej się z 1 mln węzłów, z których każdy współpracuje z co najwyżej 99 komputerami komunikacyjnymi lub koncentratorami. Dodatkowo adres ma swoją kontynuację w polu CALL USER DATA. To ostatnie pole jest jednakże przezroczyste dla podsieci komunikacyjnej. Proponowane jego wykorzystanie przedstawiono na rys. 3. W konfiguracji pilotowej (rys. 4) przyjęto jako obowiązujące adresy zestawione w tab. 2.

W sieci MSK jest możliwe stosowanie także tzw. skróconego adresowania. Ten typ adresowania stosować mogą DTE związane lokalnie przez wspólny węzeł. W polu adresowym wówczas jest możliwe pominięcie części adresu określającej numer węzła.



Rys.3. Wykorzystanie pola CALL USER DATA

Fig.3. Use of the CALL USER DATA field



Rys.4. Konfiguracja pilotowa sieci MSK

Fig.4. Pilot configuration of the MSK network

Adresacja w pilotowej wersji sieci MSK

Miejscowość	Logiczny nr węzła	Nazwa DTE	
		użytkowa	logiczna
WROCLAW	01	system ODRA	0101
WROCLAW	01	system RIAD	0102
WARSZAWA	02	system ODRA	0201
GLIWICE	03	system ODRA	0301

3. KOMUNIKACJA WEWNĄTRZ PODSIECI

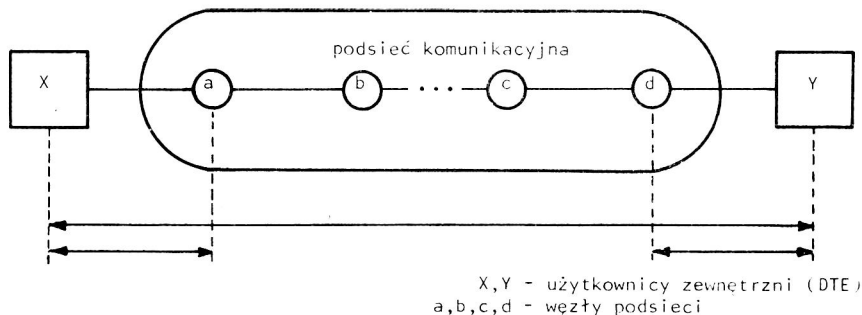
Zasady protokołu X.25/3 obowiązują na styku DTE-DCE. Jednakże droga logiczna pomiędzy dwoma użytkownikami korzystającymi z usług sieciowych przez DTE składa się z co najmniej trzech odcinków, tj. DTE-DCE, DCE-DCE, DCE-DTE. Do zestawienia i utrzymania całej tej drogi wymagane jest więc określenie pewnego zbioru zasad postępowania, zwanego dalej protokołem międzywęzłowym. Zdefiniowany dalej protokół zakładał, że:

1. Protokół pakietowy X.25/3 ma dostatecznie rozbudowane mechanizmy sterowania przepływem. Mechanizm ten wystarcza do obrony podsieci przed akceptacją zbyt dużej liczby pakietów danych, a zagrażającej gwałtownym wyczerpaniem zasobów w węzłach brzegowych.

2. Istniejący protokół liniowy dostarcza skutecznych mechanizmów zapobiegających gromadzeniu się pakietów w węzłach końcowych dla danego połączenia w razie wstrzymania odbioru przez DTE. W rezultacie nie ma konieczności sterowania przepływem pakietów na poziomie międzywęzłowego protokołu pakietowego.

Podane względy zadecydowały, że dla całej podsieci komunikacyjnej (dla jej poziomu logicznego) przyjęto schemat sterowania przepływem podany na rys. 5. Ma on wiele korzystnych właściwości [4], a przy założeniach 1. i 2. winien gwarantować stabilną i niezawodną transmisję danych pomiędzy każdą parą DTE. Dodatkowo, aby osiągnąć ten cel przyjęto, że transmisje wewnątrz podsieci będą realizowane na bazie połączeń wirtualnych a nie datagramowych [4]. W rezultacie jedyną cechą, którą musi realizować protokół międzywęzłowy, jest przesyłanie pakietów pomiędzy dwoma skrajnymi DCE przez raz utworzoną i stałą w czasie drogę logiczną. W sensie programowym wymaga to utworzenia pewnych struktur informacyjnych. Dla połączeń VC proces ten jest inicjowany w momencie

odebrania pakietu CALL REQUEST. Dla połączeń PVC struktury te są tworzone w czasie generacji systemu.



Rys.5. Schemat sterowania przepływem w sieci MSK na poziomie przesłań logicznych

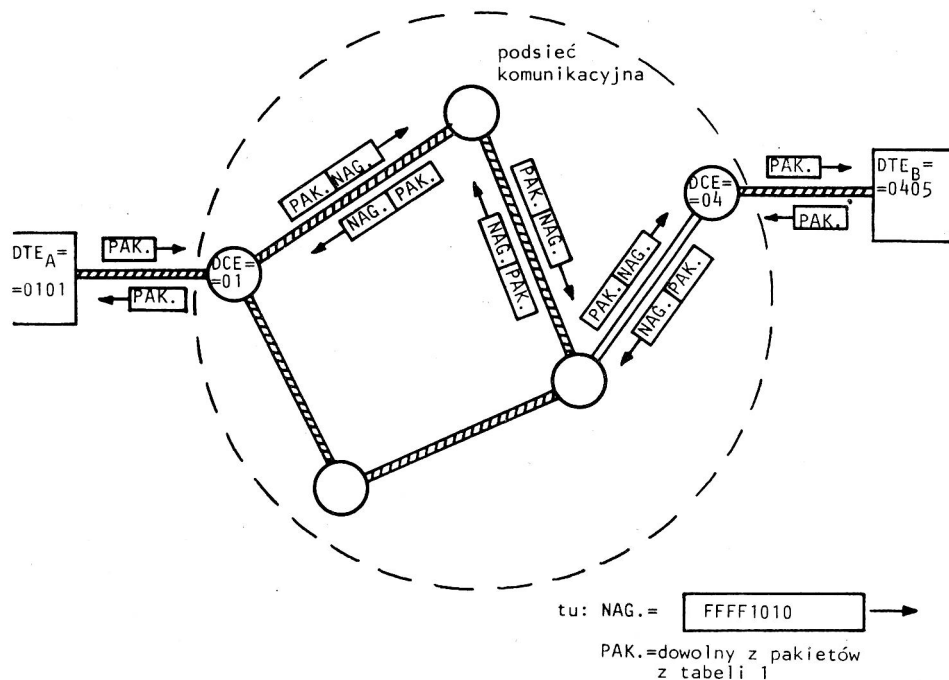
Fig.5. Scheme of flow control in ICN network on the logical level

W sensie protokolarnym utworzenie tej drogi, a następnie identyfikacja pakietów i kierowanie ich do właściwych węzłów wymaga utworzenia jednostki danych protokołu międzywęzłowego. Jednostką tą jest pakiet (w sensie X.25/3) wraz z dodatkowym nagłówkiem. W obecnie przyjętej wersji protokołu nagłówek ma następujące cechy:

- stałą długość wynoszącą 8 półoktetów,
- zapisany jest w nim adres DTE inicjującego nawiązanie połączenia (dla połączeń PVC istnieje odpowiednia umowa),
- gdy adres DTE jest krótszy niż 8 półoktetów, wtedy brakujące pola są wypełniane cyfrą F_{16} .

Ilustracją podanych założeń jest rys. 6, który obrazuje przebieg transmisji pomiędzy parą użytkowników A i B, aktywnych odpowiednio poprzez DTE_A i DTE_B . Przyjęto, że DTE_A o adresie 0101 jest członem inicjującym połączenie z DTE_B o adresie 0405.

Należy zauważyć, że połączenie virtualne pomiędzy dowolną parą użytkowników A i B aktywnych przez DTE_A i DTE_B , zestawiane w różnych momentach czasu, przebiegać może przez różne węzły pośrednie.



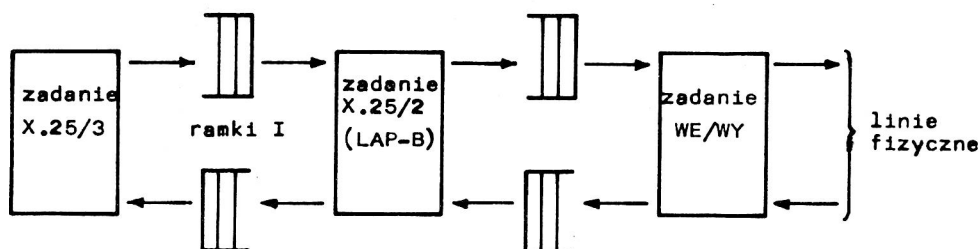
Rys.6. Formaty informacji przesyłanej wewnątrz i na zewnątrz podsieci komunikacyjnej

Fig.6. Formats of information sent in and out the communication sub-network

4. STRUKTURA OPROGRAMOWANIA WĘZŁA

Węzeł podsieci komunikacyjnej jest systemem pracującym w czasie rzeczywistym. Jego realizacja wymaga zbudowania na bazie SO czasu rzeczywistego odpowiednio powiązanych zadań użytkowych. Wymagane są co najmniej trzy takie zadania i ich wzajemne powiązania ilustruje rys. 7.

Program operacyjny węzła dostarcza także mechanizmów synchronizujących wymienione zadania. Są one aktywowane asynchronicznie i mają na ogół różne priorytety. O jakości pracy węzła decyduje także zrealizowana w nim koncepcja gospodarki buforami. Z rozważań teoretycznych [9] wynika, że najkorzystniejszą strategią jest zastosowanie tzw. wspólnej puli buforów. Oczywiście są możliwe inne koncepcje strukturalnego powiązania zadań użytkowych węzła i różna od wymienionej gospodarka buforami. Nie gwarantuję one - zdaniem autora - osiągnięcia maksymalnej przepustowości, a także, jak w projekcie [8] przez zastosowanie bufora



Rys.7. Powiązania strukturalne pomiędzy zadaniami węzła
Fig.7. Structural links between the tasks in the network node

cyklicznego w zadaniu WE rozwiązania te są wrażliwe na chwilowe, większe od średniej, zmiany gęstości strumienia informacji. Prowadzi to do sztucznego odrzucania ramek wejściowych zamiast ich akumulowania i przygotowania do późniejszego przetwarzania. Inny jest też mechanizm synchronizacji zadań. Związkiem zadanie X.25/3 jest aktywowane jednocześnie tylko dla jednego bufora z ramką I. Zadanie to jest nieprzerywalne, a więc procedury tego zadania nie muszą być wielowejściowe (ang. reentry).

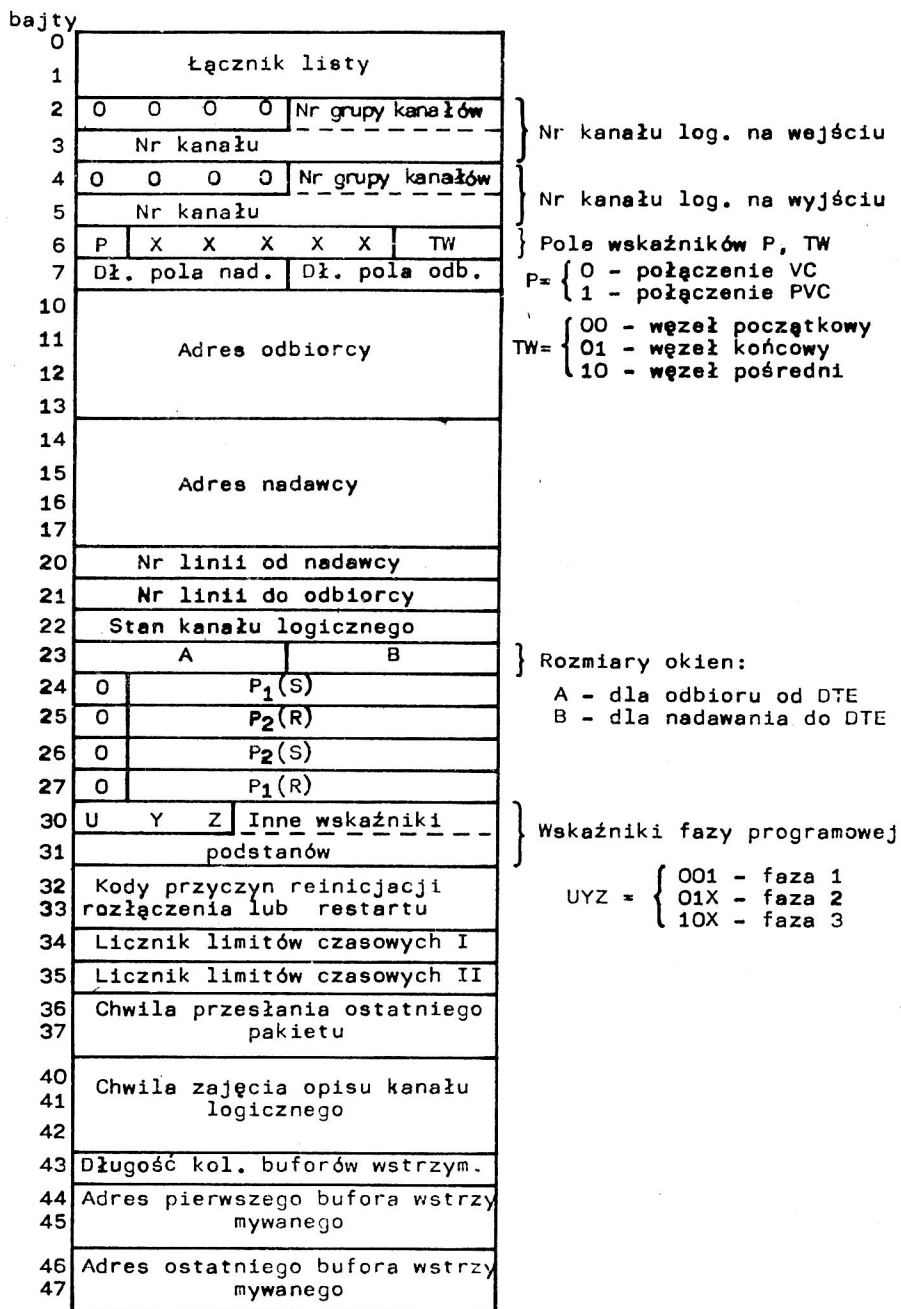
Jak nadmieniono w rozdz. 3 w celu zestawienia i utrzymania połączenia logicznego pomiędzy komunikującymi się ze sobą DTE w każdym z węzłów uczestniczących w tym połączeniu wypełniana jest pewna struktura informacyjna zwana Opiskiem Kanału Logicznego (OKL). Postać i wykorzystanie pól tej struktury ilustruje rys. 8.

W OKL są zawarte niezbędne informacje pozwalające zarówno na prowadzenie transmisji w danym kanale zgodnie z wymaganiami protokołu, a także na rozdzielanie obsługi tego kanału pomiędzy różne submoduły programowe w ramach zadania X.25/3. Zadanie to składa się zasadniczo z 4 submodułów odpowiadających za:

- zestawienie drogi logicznej (faza 1),
- transmisję danych i reinicjację (faza 2),
- kasowanie (zamknięcie) drogi logicznej (faza 3),
- restart.

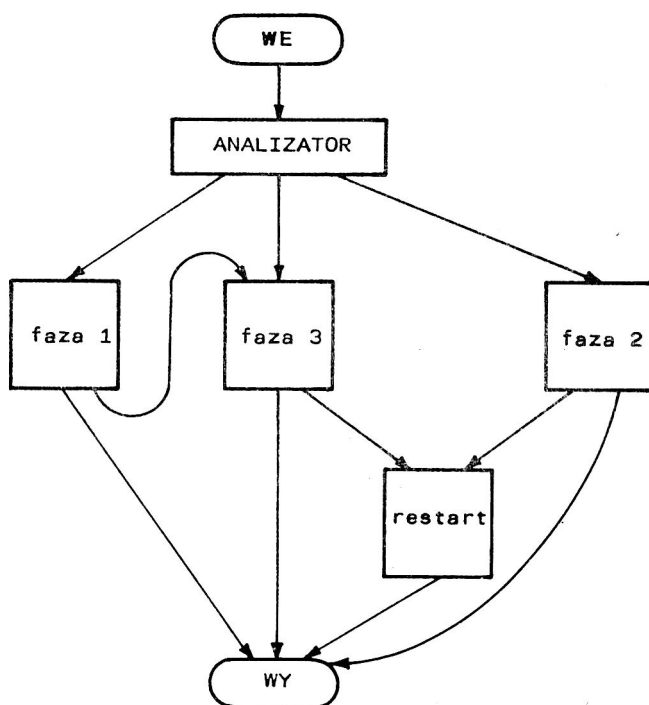
Główne strukturalne powiązania między tymi submodułami przedstawiono na rys. 9.

Wewnętrzna struktura 4 najważniejszych submodułów oraz ich szczegółowe funkcje są opisane odpowiednio w pracy [5], [7], [3], [6]. Wyjaśnienia wymaga natomiast rola ANALIZATORA i sam mechanizm przekazywania sterowania między submodułami. ANALIZATOR rozpoznaje wstępnie informację zawartą w buforze z ramką typu I i na tej podstawie znajduje OKL, a tym samym kanał logiczny, w którym odbywa się transmisja. OKL może nie istnieć tylko dla fazy 1 i wtedy będzie w tej fazie założony. Przez



Rys.8. Struktura informacyjna opisu kanału logicznego

Fig.8. Information structure of logical channel description



Rys.9. Powiązania strukturalne submodułów w zadaniu X.25/3

Fig.9. Structural links in the X.25/3 task

wskaźniki programowe umieszczone w OKL ANALIZATOR odnajduje odpowiedziany za dany stan kanału submoduł i przekazuje mu sterowanie. Odnaleziony submoduł może być niewystarczający dla pełnej obsługi odebranego bufora. Wówczas wygeneruje on żądanie zmiany kontekstu przez wskazanie dodatkowej fazy. Żądanie to ma odzwierciedlenie w zmianie wskaźników programowych w OKL i w ten sposób przyszłe przetwarzanie w tym kanale logicznym będzie przez ANALIZATOR przekazywane zawsze aktywnej fazie.

LITERATURA

- [1] D a n i e l s D., B a r b e r D.L.A., Sieci teleinformatyczne, WNT, Warszawa 1980.
- [2] D u d z i e l e w i c z I., Kasowanie połączeń logicznych w sieci komputerowej MSK (w niniejszym zeszycie).

- [3] D u b i e l e w i c z I., K o l e ś n i k K., Reinicjacja transmisji danych w sieci komputerowej MSK (w niniejszym zeszycie).
- [4] F r y ż l e w i c z Z., Protokoły pakietowe w sieciach komputerowych, Raporty Centr. Oblicz. Pwr., Seria PRE nr 44/80, Wrocław 1980.
- [5] F r y ż l e w i c z Z., M i k u l s k a E., Zestawianie połączeń logicznych w sieci komputerowej MSK (w niniejszym zeszycie).
- [6] G o e t z J., Restart połączeń logicznych w sieci komputerowej MSK (w niniejszym zeszycie).
- [7] K o l e ś n i k K., Transmisja danych w sieci komputerowej MSK (w niniejszym zeszycie).
- [8] L e w o c J., Ł a n o w s k a B., S t a n i s z A., S z w a r c M., Zasady przygotowania oprogramowania węzła, Raporty Centr. Oblicz. Pwr., Seria SPR 11/80, Wrocław 1980.
- [9] P a t e l A., P u r s e r M., Systems Programming for Data Communications on Minicomputers, Software-Practice And Experience, Vol. 10, 1980, s. 283-305.
- [10] Draft Revised Recommendation X.25, Introduction And Packet Level, CCITT Study Group VII, Contribution No 439, Geneva, Nov. 79.
- [11] Reference model of Open Systems Architecture, ISO/TC97/SC16/N117.

Praca wpłynęła do Redakcji 30.12.1980 r.

Po poprawieniu 3.03.1982 r.

PACKET PROTOCOL IN THE MSK COMPUTER NETWORK

In this paper the packet level protocol realized in the MSK node is described. It is given in particularity:

- types and parameters of the X.25/3 protocol packets implemented,
- principle of internode communication,
- software structure of the module which realizes protocol packet task,
- data structures required for programming the module described in the paper.

Verified by Ruta Czaplińska

ПАКЕТНЫЙ ПРОТОКОЛ В СЕТИ ЭВМ MSK

В статье описан протокол пакетного уровня, который был реализован в узле сети ЭВМ MSK. В особенности представлены:

- типы и параметры имплементированных пакетов уровня X.25,
- принцип действия межузловой коммуникации.
- структура программного модуля, реализующего пакетный протокол,
- информационные структуры, требуемые для представленного программного модуля.

Проверила Малгожата Хейдрих

*Sieć komputerowa,
węzeł, protokół pakietowy*

Zbigniew FRYŻLEWICZ^{*)}, Elżbieta MIKULSKA^{**)}

Zestawianie połączeń logicznych w sieci komputerowej MSK

W artykule przedstawiono realizację fazy nawiązywania połączenia według protokołu pakietowego X.25/3. Przede wszystkim opisano:

- struktury i parametry dopuszczalnych w tej fazie pakietów,
- struktury danych wymagane dla programowej realizacji fazy,
- strukturę i akcje modułu programowego realizującego tę fazę,
- typy zdarzeń rejestrowanych w celach diagnostyki i miernictwa.

1. WSTĘP

W przebiegu transmisji zrealizowanym na podstawie protokołu X.25/3 wyróżnia się zasadniczo 3 fazy:

- a) nawiązywania połączenia,
- b) transmisji danych,
- c) kasowania połączenia.

Opis faz b), c) dla sieci komputerowej MSK przedstawiono odpowiednio w pracach [4] i [1]. Celem niniejszej pracy jest natomiast opis akcji podejmowanych w fazie a).

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

^{**)} Instytut Organizacji i Zarządzania Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

Jak opisano w pracy [3], w sieci MSK istnieć mogą połączenia permanentne (PVC) lub też dynamiczne (VC). Oczywiście faza nawiązywania połączenia ma sens w przypadku tych ostatnich. Połączenia te mogą być zakładane wyłącznie na żądanie DTE. Żądanie takie w formie przesłanego pakietu jest analizowane przez DCE i po sprawdzeniu jego syntaktycznej i semantycznej poprawności powoduje, że podsieć generuje żądanie do wskazanego odbiorcy. Niektóre parametry obu żądań mogą być modyfikowane przez podsieć. W ten sposób podsieć chroni zasoby węzła przed wyczerpaniem, a całą sieć przed przeciążaniem.

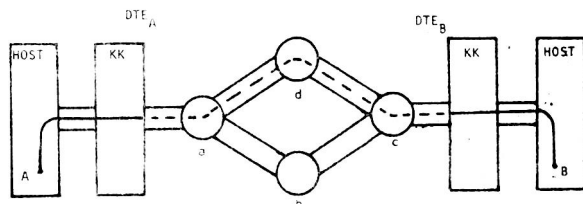
Szczegółowy opis funkcji podejmowanych w fazie zestawiania połączenia przedstawiono w rozdz. 2. W rozdz. 3 opisano poprawne dla tej fazy formaty pakietów z tabelami i grafami przejść, przedstawiającymi zachowanie wyróżnionego kanału logicznego. Opis wykorzystywanych w tej fazie struktur danych jest natomiast zawarty w kolejnym, tj. czwartym rozdziale. Tam też pokazano działanie na tych strukturach w kolejnych węzłach uczestniczących w zestawianym połączeniu. Następny rozdział zawiera wykaz sygnalizowanych w tej fazie typów zdarzeń z podziałem na kategorie informacyjne. Całość kończą uwagi o strukturze programowej omawianego modułu.

2. FUNKCJE FAZY NAWIĄZYWANIA POŁĄCZENIA

Celem tej fazy jest zestawienie połączenia logicznego pomiędzy dwoma użytkownikami sieci. Przebiega ono (zakładając, że każdy z DTE ma umowę ze związanym z nim DCE, precyzującą pewną liczbę dwukierunkowych kanałów logicznych, będących do ich wzajemnej dyspozycji) w następujących etapach:

1. DTE zgodnie z istniejącym podziałem kanałów wybiera wolny kanał logiczny i przesyła w nim pakiet CALL REQUEST do DCE.
2. DCE po sprawdzeniu poprawności takiego żądania przesyła ten pakiet do DCE odbiorcy.
3. DCE odbiorcy wybiera wolny kanał logiczny i po nim przesyła żądanie nawiązania połączenia w formie pakietu INCOMING CALL.
4. Jeśli DTE wywołujący akceptuje odebrane żądanie, to powiadamia o tym związane z nim DCE przysyłając pakiet CALL ACCEPTED.
5. Podsieć komunikacyjna transmituje odebrany pakiet potwierdzenia z powrotem do DTE wywołującego, przekazując to potwierdzenie w formie pakietu CALL CONNECTED.
6. Jeśli dwa DTE równocześnie będą żądały zestawienia między sobą połączenia logicznego, to będą zestawione dwa takie połączenia. Nie jest strukturalnie rozwiązany problem kolizji [2].

Odebranie żądania nawiązania połączenia powoduje na wstępie sprawdzenie kontekstu i formalnej poprawności żądania. Kolejno są analizowane stany zasobów węzła, m.in. sprawdza się dostępność buforów wymaganych do założenia opisu połączenia, drożność dróg fizycznych do nadawcy itp. Pozytywny wynik wszystkich kontroli powoduje przesłanie otrzymanego żądania przez podsieć w kierunku odbiorcy. W każdym z węzłów pośrednich są przeprowadzane również powyższe kontrole, a ich pozytywny wynik powoduje kolejne przesłania żądania, które w rezultacie trafia do adresata (DTE wywoływane). Na drodze przejścia tego żądania w każdym z węzłów jest zapamiętywana pewna ilość informacji o tworzonym połączeniu w strukturach danych, zwanych Opisaniami Kanałów Logicznych (OKL). W węzle końcowym, tj. bezpośrednio związanym z wywołującym DTE, podsieć kontroluje ponadto czy nie został przekroczony limit oczekiwania na odpowiedź pozytywną lub negatywną. Zestawiona w fazie nawiązywania połączenia droga logiczna (rys. 1) pozostaje nie zmieniona na czas przesyłania wszelkich innych pakietów pomiędzy użytkownikami do momentu rozłączenia.



Rys.1. Przykład drogi logicznej zestawionej w fazie nawiązywania połączenia pomiędzy użytkownikami A i B przez DTE_A-a-d-c-DTE_B: KK - komputer komunikacyjny, a, c - węzły brzegowe, d - węzeł pośredni

Fig.1. Example of virtual way made during call set-up phase between the users A i B through DTE_A-a-d-c-DTE_B

Jeśli na którymkolwiek etapie zestawienia połączenia stwierdzona zostanie niemożność jego realizacji, np. z powodu:

- chwilowego braku wymaganych zasobów,
- wykrycia przez podsieć stanu określanego jako zapętlenie,
- przekroczenia limitu czasowego itp.,

to podjęte zostaną działania w celu likwidacji zestawionej do danej chwili drogi logicznej. Akcje rozłączające są podejmowane także w momencie przesłania przez którykolwiek z DTE pakietu CLEAR REQUEST. W tym ostatnim przypadku sterowanie przejmie moduł opisany w pracy [1].

Fakty otrzymania żądania zestawienia połączenia logicznego, pozytywnej lub negatywnej realizacji tego żądania, a w tym ostatnim przypadku również przyczyn tego stanu, są rejestrowane. Rodzaj pamiętanej informacji i sposób jej wykorzystania w celach miernictwa i diagnostyki opisano szczegółowo w pracy [6].

3. STRUKTURA PAKIETÓW, GRAFY I TABELE PRZEJŚĆ

W celu zestawienia połączenia logicznego wykorzystuje się 4 typy pakietów:

- CALL REQUEST,
- INCOMING CALL,
- CALL ACCEPTED,
- CALL CONNECTED,

których strukturę przedstawiono na rys. 2 i 3. W tabeli 1 zamieszczono natomiast dopuszczalne w obecnej wersji kody i parametry pola ułatwień. Poprawny przebieg fazy nawiązywania połączenia przedstawia graf z rys. 4.

Typ pakietu	Część*) wspólna	Długość adresów	Adresy	Długość pola ułatwień	Pole ułat- wień	Dane
CALL REQUEST	K	K	K	K	O	O/16/
CALL ACCEPTED	K	O	O	O	O	Z

*) zawiera GFI, LCGN, LCN, PTI [5],

K - pole konieczne,

O - pole opcjonalne,

O/X/ - pole opcjonalne, maksymalna długość X bajtów,

Z - pole niedopuszczalne.

Rys.2. Formaty pakietów od DTE: CALL REQUEST i CALL ACCEPTED

Fig.2. Formats of CALL REQUEST and CALL ACCEPTED packets

Typ pakietu	Część*) wspólna	Długość adresów	Adresy	Długość pola ułatwień	Pole ułat- wień	Dane
INCOMING CALL	K	K	K	K	O	O/16/
CALL CONNECTED	K	K	K	K	O	Z

*) - zawiera GFI, LCGN, LCN, PTI [5],

K - pole konieczne,

O - pole opcjonalne,

O/X/ - pole opcjonalne, maksymalna długość X bajtów,

Z - pole niedopuszczalne.

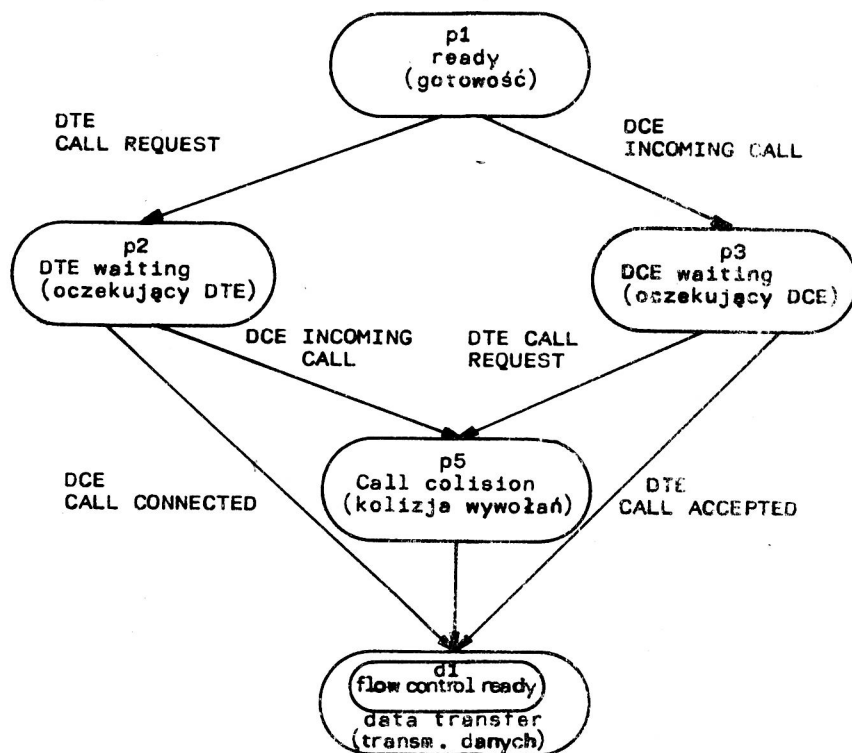
Rys.3. Formaty pakietów od DCE: INCOMING CALL i CALL CONNECTED

Fig.3. Formats of INCOMING CALL and CALL CONNECTED packets

T a b e l a 1

Kody i parametry pola ułatwień

Nazwa parametru x	Kod parametru (dwójkowo)	Wartość parametru x	Uwagi
Rozmiar okna	01000011	$0 \leq x \leq 7$	W razie braku parametru ustala się wartość standardową $x = 2$
Parametr rodzaju kontroli P(R)	00000101	0000000X	Obecna wersja realizuje sterowanie przepływem typu end-to-end niezależnie od przenoszonej wartości (dopuszczalnej P(R))



Rys.4. Graf przejścia fazy nawiązywania połączenia
 Fig.4. Diagram of states for the call set-up phase

Jak z niego wynika w fazie tej wyróżnia się 5 stanów kanałów logicznych, gdzie:

- p1 - kanał w stanie wolnym gotowy do podjęcia akcji;
- p2 - stan kanału, w którym odebrano od DTE żądanie zestawienia połączenia (odebrano pakiet CALL REQUEST);
- p3 - stan kanału, w którym przesłano z podsieci do DTE żądanie nawiązania połączenia (przesłano pakiet INCOMING CALL);
- p4 - kanał w stanie gotowości do transmisji danych (w węźle początkowym dla danego wywołania jest to stan, w który przechodzi kanał po przesłaniu do DTE pakietu CALL CONNECTED; w węźle końcowym - po odebraniu od DTE pakietu CALL ACCEPTED);
- p5 - stan kolizji wywołań. W tym przypadku DCE zwolni kanał i przydzieli go do obsługi żądania otrzymanego od DTE.

Tabela 2 stanowi uzupełnienie grafu z rys. 4 i przedstawia zmiany stanów kanałów logicznych w wyniku otrzymania przez DCE pakietów od DTE z nim związanego.

T a b e l a 2

Zmiany stanu kanału logicznego w DCE

Pakiet otrzymany od DTE	Stan kanału logicznego w DCE		
	p1	p2	p3
CALL REQUEST	p2	błąd p7	p1 lub p2
CALL ACCEPTED	błąd*) p1	błąd p7	p4 (d1)
CLEAR REQUEST	błąd*) p1	p6	p6
CLEAR CONFIRMATION	odrzućcie p1	błąd p7	błąd p7
dane, RESET, przerwanie	błąd*) p1	błąd p7	błąd p7
RESTART REQUEST lub DTE RESTART CONFIRMATION nie w kanale o nr ≠ 0	p1	błąd p7	błąd p7
Pakiet zbyt krótki lub nie rozpoznany przez DCE	p1	błąd p7	błąd p7

*) Błędny pakiet w stanie p1 powoduje wysłanie do jego nadawcy pakietu CLEAR INDICATION z kodem diagnostycznym 36_{10} lub 20_{10} (patrz tabela kodów diagnostycznych wykorzystywanych w fazie nawiązywania połączenia).

4. PRZEBIEG FAZY ZESTAWIANIA POŁĄCZENIA LOGICZNEGO

4.1. Struktury danych

Do opisu zestawianego połączenia logicznego w każdym węźle wchodzącym w jego skład jest tworzona struktura danych zwana Opiszem Kanału Logicznego. W opisywanej fazie tylko wybrane pola tej struktury są zapisywane i wykorzystywane. I tak, w węźle początkowym i końcowym dla danego wywołania są zapisywane następujące pola:

- a) 12-bitowe pole "Numer kanału logicznego na wejściu" (NKLWEE),
- b) 12-bitowe pole "Numer kanału logicznego na wyjściu" (NKLWYE),
- c) 1-bajtowe pole wskaźników (PTWE),
- d) 1-bajtowe pole "Długości pól adresowych DTE" (DPADE),
- e) 4-bajtowe pole "Adres wywoływanego DTE (odbiorcy)" (AODBE),
- f) 4-bajtowe pole "Adres wywołującego DTE (nadawcy)" (ANADE),
- g) 1-bajtowe pole "Nr linii od nadawcy" (LIWEE),
- h) 1-bajtowe pole "Nr linii do odbiorcy" (LIWYE),
- i) 1-bajtowe pole "Stan kanału logicznego" (SKLE),
- j) 1-bajtowe pole "Rozmiary okien dla odbioru i nadawania" (ROZKE),
- k) 2-bajtowe pole "Programowe wskaźniki fazy" (WFAZE),
- l) 2-bajtowe pole "Chwila wysłania ostatniego pakietu" (TOSTPE).

Strukturę tych pól przedstawiono na rys. 5. W węzłach pośrednich pola i), j), k) nie są zapisywane ani analizowane.

4.2. Działania w fazie zestawiania połączenia logicznego

Akcje podejmowane przez DCE w fazie zestawiania połączenia logicznego są zależne od:

- stanu kanału,
- typu odebranego pakietu,
- stanu zasobów węzła.

Istnieje jednak kilka akcji standardowych, podejmowanych niezależnie od podanych warunków. Należą do nich:

- rejestracja zdarzenia,
- badanie syntaktycznej i semantycznej poprawności pakietu,
- badanie kontekstu uzyskanego pakietu,
- formowanie semidatagramu^{*)} przez dopisanie nagłówka do pakietu lub odwrotnie - usunięcie nagłówka w celu uformowania pakietu.

^{*)} Semidatagram - jednostka danych protokołu międzywęzłowego [3]. Składa się z nagłówka zawierającego adres DTE inicjującego połączenie i pakietu w sensie X.25/3.

modyfi- kator	nr bajtu (ósemkowo)		
	0		
	1		
NKLWEE	2	0 0 0 0	nr gr kan log
	3		nr kanału logicznego
NKLWYE	4	0 0 0 0	nr gr kan log
	5		nr kanału logicznego
PTWE	6	P X X X X X	TW
APADE	7		dł. pola nad. dł. pola odb.
ADDRB	10		nr węzła

			nr KK
ANADE	14		nr węzła

			nr KK
LIWEE	20		nr linii od nadawcy
LIWYE	21		nr linii do odbiorcy
SKLE	22		stan kanału logicznego
ROZOKE	23	DTE-DCE	DCE-DTE
	24	0	} rozm. okien
	25	0	
	26	0	
	27	0	
WFAZE	30	W F P H	w s k a z n i k i
	31	I C H	f a z y
TOSTPE	36		chwila wysłania
	37		ostatniego pakietu
TZAJE	40		chwila zajęcia OKL
	46		
	47		

Rys.5. Struktura OKL: pola wykorzystywane w fazie nawiązywania połączenia

Fig.5. Structure of OKL: fields used in the call set-up phase

Zestawianie połączeń logicznych w sieci...

Pozostałe działania są działaniami kontekstowymi. I tak:

1. W stanie p1 otrzymanie pakietu CALL REQUEST powoduje:
 - przydzielenie wolnej struktury danych na założenie OKL,
 - zapisanie w niej informacji o zakładanym połączeniu,
 tzn. wypełnienie pól opisanych w punkcie 4.1,
 - utworzenie semidatagramu z pakietem CALL REQUEST i wysłanie go w kierunku zdalnego DCE.
2. Otrzymanie semidatagramu z pakietem CALL REQUEST w węźle końcowym powoduje:
 - przydzielenie wolnej struktury danych na założenie OKL,
 - zapisanie w niej informacji o zakładanym połączeniu,
 - odnalezienie kanału w stanie p1, utworzenie w nim pakietu INCOMING CALL, wysłanie go do DTE oraz zmianę stanu kanału na p3,
 - włączenie mechanizmu odmierzania czasu.
3. W stanie p3 odebranie pakietu CALL ACCEPTED powoduje:
 - wyłączenie mechanizmu odmierzania czasu,
 - końcowe ustalenie parametrów przepływu na styku zdalnym DTE-DCE,
 - utworzenie semidatagramu z pakietem CALL ACCEPTED i wysłanie go w stronę węzła początkowego zgodnie z danymi zapisanymi w OKL,
 - zmianę stanu kanału na p4 i wskaźnika fazy (przygotowanie kanału do programowej obsługi w fazie transmisji danych).
4. W stanie p2 odebranie semidatagramu z pakietem CALL ACCEPTED powoduje:
 - końcowe ustalenie parametrów przepływu na styku lokalnym DTE-DCE,
 - utworzenie pakietu CALL CONNECTED i wysłanie go do DTE,
 - zmianę stanu kanału na p4 i wskaźnika fazy (przygotowanie stanu kanału do programowej obsługi w fazie transmisji danych).
5. W stanie p3 odebranie pakietu CALL REQUEST powoduje:
 - wyłączenie mechanizmu odmierzania czasu,
 - chwilową zmianę stanu kanału na p5,
 - zwolnienie zasobów przydzielonych poprzednio do obsługi zdalnego wywołania z jednoczesnym powiadomieniem zdalnego DTE o niemożności zrealizowania połączenia,
 - obsługę wywołania lokalnego (patrz punkt 1).
6. Odbiór semidatagramu w węźle pośrednim powoduje:
 - a) gdy jest to semidatagram z pakietem CALL REQUEST:
 - wykonanie procedury PĘTLA wykrywającej powstawanie cyklu zamkniętego w podsieci,
 - przy pozytywnej odpowiedzi procedury PĘTLA podjęcie akcji zwrotu dotychczas przydzielonych zasobów we wszystkich węzłach pośrednich wraz z powiadomieniem odpowiedniego DTE o

- powstałej sytuacji przez przesłanie semidatagramu CLEAR INDICATION z odpowiednimi parametrami,
- przy negatywnej odpowiedzi procedury PETA;
 - i) przydział wolnej struktury danych na założenie OKL;
 - ii) zapisanie w niej informacji o zakładanym połączeniu (w wersji skróconej, punkt 4.1);
 - iii) wyznaczenie linii wyjścia w kierunku docelowego DTE i wysłanie po niej wejściowego semidatagramu;
 - b) gdy jest to semidatagram z innym pakietem:
 - przesłanie semidatagramu do kolejnego węzła zgodnie z danymi zawartymi w OKL,
 - oddanie sterowania do submodułu opisanego w pracy [1] w celu sprawdzenia czy połączenie ulega likwidacji (na podstawie informacji z OKL i typu pakietu w przetwarzanym semidatagramie),
 - w razie braku właściwego OKL podjęcie alternatywnie akcji usunięcia oscylujących semidatagramów lub wygenerowania semidatagramu (zawiera pakiet CLEAR INDICATION z odpowiednim parametrem) powodującego zwrot zasobów w innych węzłach podsieci.

4.3. Kontrola poprawności transmisji styku DTE-DCE

Wykrycie niezgodności w budowie pakietu oraz w przenoszonych przez niego parametrach, a także odbiór pakietu nie w kontekście powoduje rutynowe czynności mające na celu rozłączenie błędnego połączenia, tj.:

- zwrot dotychczas przydzielonych zasobów podsieci,
- powiadomienie użytkowników (DTE wywołującego i wywoływanego) o przyczynie rozłączenia lub niemożności zestawienia połączenia.

Powyższe akcje są przeprowadzane przez submoduł programowy zestawiający połączenie lub zadanie to jest przekazywane do wykonania submodułowi opisanemu w pracy [1]. Ten ostatni przypadek jest realizowany wówczas, gdy do zwrotu przydzielonych dla danego połączenia zasobów są wymagane potwierdzenia rozłączenia od obu związanych tym połączeniem DTE. Submoduł opisany w pracy [1] przejmuje także sterowanie, jeśli w dowolnej chwili w fazie zestawiania połączenia którykolwiek z DTE prześle pakiet CLEAR REQUEST. Kody diagnostyczne i przyczyny rozłączenia wykorzystywane w fazie zestawiania połączenia zamieszczono w tab. 3 i 4.

T a b e l a 3

Kody diagnostyczne w pakiecie CLEAR INDICATION generowanym wskutek błędów w fazie zestawiania połączenia

Lp	Przyczyna błędu	Kod (ósemkowo)
1	Błąd GFI	50
2	Kanał logiczny poza zakresem dla połączeń VC	44
3	Pakiet zbyt długi	47
4	Pakiet niewłaściwy dla stanu p1	24
5	Zbyt długie adresy DTE, adresy DTE niewłaściwe, brak adresu (ów) DTE, brak przejścia w podsieci do zdalnego DTE	100
6	Niewłaściwy kod pola ułatwień (lub niedostępny)	101
7	Niewłaściwy parametr pola ułatwień	102
8	Symbol nie w kodzie BCD w polu "adres DTE"	103
9	Adres DTE wywołującego, niezgodny z danymi systemowymi węzła	104
10	Upłynął limit czasu oczekiwania w stanie p3	61
11	Przeciążenie w węźle pośrednim	201
12	Stan RESTART-u w węźle docelowym	202
13	Przeciążenie w węźle docelowym	203
14	Pakiet niewłaściwy dla stanu p2	25
15	Pakiet niewłaściwy dla stanu p3	26
16	Inne przyczyny	200

T a b e l a 4

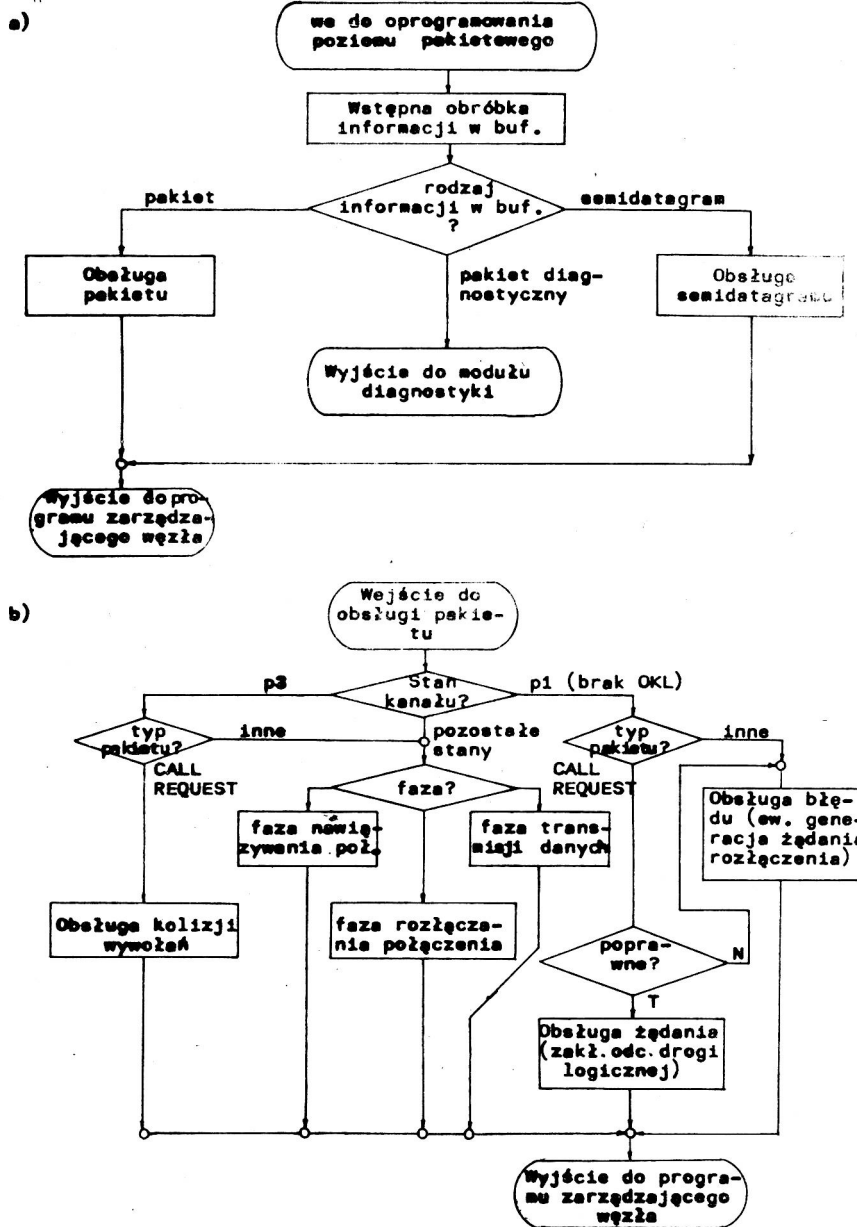
Przyczyny rozłączenia i ich kodowanie w fazie nawiązywania połączenia

Przyczyna	Kod (dwójkowo)
Błąd lokalnej procedury (Local Procedure Error)	00010011
Niedostępny zasób (Not Obtainable)	00001101
Przeciążenie sieci (Network Congestion)	00000101
Kanał zajęty (Number Busy)	00000001
Błąd zdalnej procedury (Remote Procedure Error)	00010001

5. STRUKTURA OPISOWANIA FAZY NAWIĄZYWANIA POŁĄCZENIA

W fazie nawiązywania połączenia można wyróżnić dwa etapy:

- zakładanie drogi logicznej podczas przejścia pakietu lub semidatagramu CALL REQUEST,
- działania w stanach oczekiwania na akceptację połączenia.



Rys.6. Faza nawiązywania połączenia w oprogramowaniu poziomu pakietowego: a - schemat ogólny, b - obsługa pakietu, c - obsługa semidatagramu, d - oczekiwanie na akceptację nawiązania połączenia

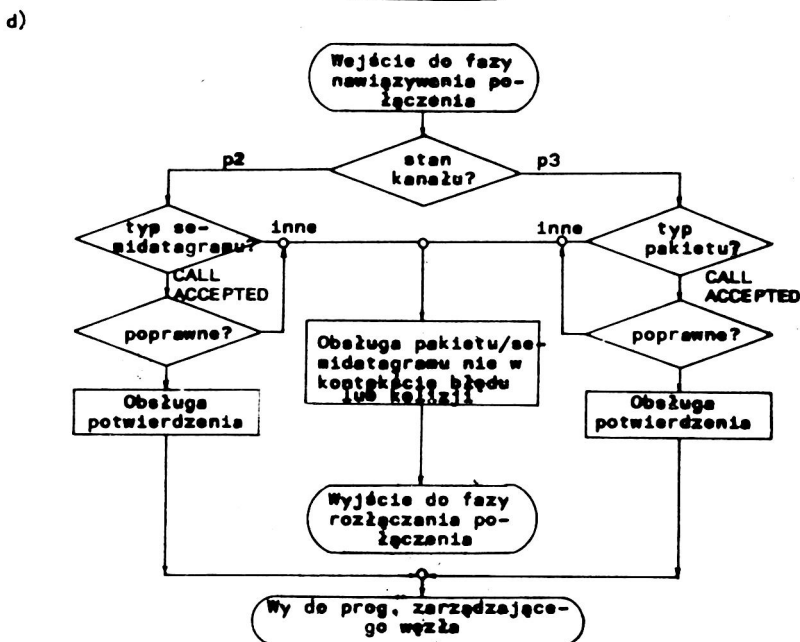
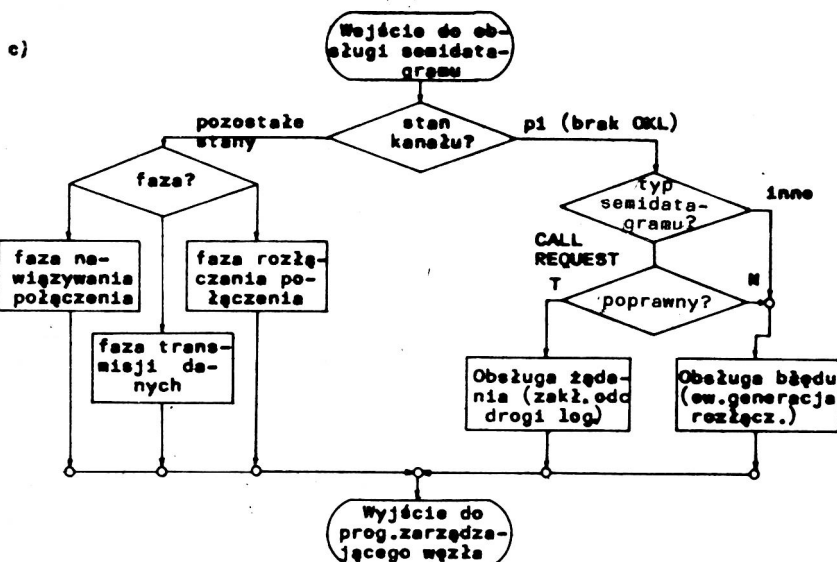


Fig.6. Call set-up phase in the software of packets level: a - general picture, b - packets service, c - semidatagram service, d - waiting for confirmation of virtual call

Strukturę programową tej fazy, w powiązaniu z oprogramowaniem poziomu pakietowego, pokazano na rys. 6a-d. Pierwszy etap jest realizowany po rozpoznaniu pakietu/semidatagramu i stwierdzeniu braku opisu kanału logicznego, po którym ten pakiet/semidatagram przyszedł (kanał w stanie p1). Wykonywane są w tym przypadku akcje wymienione w punkcie 4.2 (obsługa żądań CALL REQUEST) lub w punkcie 4.3 (obsługa błędu). Drugi etap jest realizowany programowo przez submoduł obsługujący pakiety/semidatagramy, które nadezły w stanie p2 lub p3. Wymagane jest tutaj wykonanie akcji z punktu 4.2 (obsługa potwierdzeń CALL ACCEPTED) albo z punktu 4.3 (obsługa błędu).

Następujące informacje są przekazywane w komunikacji między submodułami oprogramowania poziomu pakietowego:

- adres bufora, w którym odebrano pakiet,
- adres OKL, po którym przyszedł pakiet,
- kierunek nadejścia (od DTE lub z podsieci).

Są one przechowywane w komórkach roboczych i nie są zmieniane w czasie działań na poziomie pakietowym. W czasie tych działań są również rozpoznawane i rejestrowane zachodzące zdarzenia. Ich rodzaje w fazie nawiązywania połączenia z podziałem na kategorie i z przyporządkowanymi im wartościami liczników podano w tab. 5. Rejestracja ta jest dokonywana w celach diagnostycznych.

Działania oprogramowania poziomu pakietowego kończą się zawsze wspólnym wyjściem do programu zarządzającego węzła. Nie wszystkie działania konieczne w fazie zestawiania połączenia są jednak wykonywane tylko w opisywanym submodule. W większości bowiem przypadków wykrycia błędów i nieprawidłowości w przebiegu transmisji sterowanie jest przekazywane submodułowi kasującemu połączenie [1]. Wówczas też działanie oprogramowania poziomu pakietowego kończy się z wyjściem z tego submodule.

6. UWAGI KOŃCOWE

Warunkiem prawidłowej realizacji działań na poziomie pakietowym, a więc i w fazie nawiązywania połączenia, jest uwzględnienie, oprócz standardu, możliwie wszystkich sytuacji wyjątkowych. Wpłynie to znacznie na stabilność i niezawodność transmisji, a więc i na poprawę własności użytkowych sieci MSK. Ustalenie reguł postępowania w takich sytuacjach wymagało wprowadzenia wielu uściśleń do protokołu X.25/3. Rozwiązania te nie ograniczają jednak możliwości sieci i nie zmieniają standardowego interfejsu wymaganego do podłączenia jej do innych sieci zbudowanych również na podstawie protokołu X.25/3.

Rodzaje zdarzeń rejestrowanych w fazie nawiązywania połączenia

Lp	Zdarzenia i kategorie	Wartość licznika (w RO)
ZDARZENIA STANDARDOWE		
1	Poprawnie nawiązane połączenie VC	140
ROZŁĄCZENIA Z POWODU BRAKU ZASOBÓW		
2	Rozłączenie z powodu braku wolnych OKL. Przeciążenie podsieci	141
3	Przeciążenie podsieci. Rozłączenie z powodu braku wolnych kanałów logicznych na wyjściu (styk DTE-DCE wywołany)	142
4	Rozłączenie z powodu stanu RESTART w węźle docelowym	143
5	Przeciążenie podsieci: kolizja wywołań, rozłączenie połączenia zdalnego	144
6	Przeciążenie podsieci: kolizja wywołań + brak wolnych buforów, rozłączenie wywołania zdalnego i lokalnego	145
7	Rozłączenie połączenia z powodu próby zapętlenia	146
8	Rozłączenie połączenia zdalnego z powodu braku przejścia (awaria dróg fizycznych)	147
9	Przeciążenie podsieci, rozłączenie z powodu braku wolnych buforów	150
10	Przeciążenie podsieci, rozłączenie z innych powodów (np. odrzucenie fragmentu zawieszonego)	151
11	Zerwanie połączenia VC, głucha linia (osiągnięto limit retransmisji)	152
BŁĘDNA TRANSMISJA		
12	Odebrano błędny pakiet: - błędny formalnie - błędny w kontekście	153
13	Odebrano błędny datagram: - błędny formalnie - błędny w kontekście	154
ŻĄDANIE ROZŁĄCZENIA		
14	Wygenerowano żądania rozłączenia VC	163

Modułowa realizacja oprogramowania zapewnia łatwość wnoszenia poprawek, jak i rozbudowywania systemu przez realizację nowych typów usług definiowanych w obecnej, jak i przyszłych rekomendacjach protokołu X.25.

LITERATURA

- [1] D u b i e l e w i c z I., Kasowanie połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).
- [2] F r y ż l e w i c z Z., Protokoły pakietowe w sieciach komputerowych. Raporty Centr. Oblicz. Pwr, Seria PRE nr 44/80, Wrocław 1980.
- [3] F r y ż l e w i c z Z., Protokół pakietowy w sieci komputerowej MSK (w tym samym zeszycie).
- [4] K o l e ś n i k K., Transmisja danych w sieci komputerowej MSK (w tym samym zeszycie).
- [5] Draft Revised Recommendation X.25 Introduction And Packet Level, CCITT Study Group VII, Contribution No. 439, Nov. 79.
- [6] H u d y m a E., K o s m u l s k a - B o c h e n e k E., Edycja rejestrowanych pomiarów w sieci MSK (w tym samym zeszycie).

Praca wpłynęła do Redakcji 30.12.1980 r.

Po poprawieniu 3.03.1982 r.

THE CALL SET-UP PROCEDURE IN THE MSK COMPUTER NETWORK

In this paper the realization of the call set-up phase made according to the X.25/3 packet level protocol is presented. The following are described in particular:

- structures and parameters of the packets allowed in this phase,
- data structures required for the programming realization,
- structure and actions of module which realizes this phase,
- types of events registered for the diagnostic and surveing system.

Verified by Jutta Czaplinska

СОСТАВЛЕНИЕ ФОРМИЧЕСКИХ СВЯЗЕЙ В СЕТИ ЭВМ MSK

В статье представлена реализация фазы налаживания логической связи в соответствии с пакетным протоколом X.25/3. В особенности описаны:

- структура и допускаемые параметры пакетов в этой фазе,
- структура данных, требуемые для программной реализации фазы,
- структура и акции программного модуля, реализующего эту фазу,
- типы событий, регистрируемых для диагностических и метрологических целей.

Проверила Малгожата Хандрих

*Sieć komputerowa,
protokół pakietowy*

Krystyna KOLEŚNIK^{*)}

Transmisja danych w sieci komputerowej MSK

Opisano działanie poziomu pakietowego w węźle sieci komputerowej MSK w fazie transmisji danych. Do realizacji poziomu pakietowego przyjęto protokół pakietowy X.25. Określono zasady transmisji, omówiono akcje fazy, sposób kontroli błędów oraz implementację fazy w oprogramowaniu sieci MSK.

1. WSTĘP

Celem fazy transmisji danych jest utrzymywanie i nadzorowanie transmisji pomiędzy DCE i DTE w połączeniu logicznym trwałym lub czasowym (wirtualnym), utworzonym w fazie nawiązywania połączenia [6].

Transmisja danych polega na przesyłaniu pakietów informacyjnych i pakietów przerwań w jednym lub obu kierunkach połączenia. Wszystkie przesyłane pakiety danych i przerwań muszą być odpowiednio potwierdzone. Przewiduje się możliwość okresowego blokowania pakietów danych w węźle nadawcy lub odbiorcy w razie braku gotowości odbiorcy pakietu do odbioru.

Poprawność przesyłania i potwierdzania pakietów podlega kontroli. Wykrycie błędu podczas transmisji powoduje w zależności od rodzaju błędu reinicjację transmisji, jej zakończenie lub restart łącza.

Reinicjacja i zakończenie transmisji, a także restart łącza, mogą być spowodowane także przez jednego z użytkowników połączenia na skutek

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

przesłania odpowiedniego pakietu. Pakiety przesyłane do węzła z DTE, przed wysłaniem do podsięci, są uzupełniane do postaci datagramu o nagłówek, umożliwiający identyfikację odbiorcy pakietu w podsięci. Datagramy przesyłane do węzła z podsięci mają nagłówek obcinany i do DTE są przesyłane w postaci pakietu.

2. ZASADY TRANSMISJI

Pakiety i datagramy przychodzące do węzła są kierowane do kanału logicznego, połączonego z DTE lokalnym dla węzła, wchodzącym w skład połączenia logicznego. Numer kanału logicznego jest rozpoznawany przez fazę nawiązywania połączenia, z której jest przekazywane sterowanie do odpowiedniej fazy w zależności od stanu kanału logicznego.

Stan kanału logicznego, źródło i rodzaj pakietu określają akcje wykonywane w określonej fazie. Wykaz pakietów akceptowanych i obsługiwanych w fazie transmisji danych podano w tab. 1. Obowiązujące formaty tych pakietów pokazano na rys. 1.

T a b e l a 1

Pakiety i ich potwierdzenia akceptowane przez poziom pakietowy węzła w fazie transmisji danych

Pakiet	Potwierdzenie
DTE DATA	DCE RR DCE RNR DCE DATA
DCE DATA	DTE RR DTE RNR DTE DATA
DTE INTERRUPT	DCE INTERRUPT CONFIRMATION
DCE INTERRUPT	DTE INTERRUPT CONFIRMATION
DTE RESET REQUEST	-
DCE RESET INDICATION	-
DTE CLEAR REQUEST	-
DCE CLEAR INDICATION	-
DTE RESTART REQUEST	-

Ogólny kod formatu	Nr grupy kanałów logicznych
Numer kanału logicznego	

format nagłówka

Nagłówek			
P(R)	M	P(S)	0
Dane			

DCE lub DTE DATA

Nagłówek	
Identyfikator typu	478

DCE lub DTE
INTERRUPT CONFIRMATION

Nagłówek	
P(R)	0 00101

DCE lub DTE RNR

Nagłówek	
Identyfikator typu	378

DCE lub DTE RESET
CONFIRMATION

GFI	0
0	
Identyfikator typu	3738
Kod przyczyny restartu	
Kod diagnostyczny	

DTE RESTART REQUEST

Nagłówek	
Identyfikator typu	438
Dane o przerwaniu	

DCE lub DTE INTERRUPT

Nagłówek	
P(R)	D 00001

DCE lub DTE RR

Nagłówek	
Identyfikator typu	33.8
Kod przyczyny reinicjacji	
Kod diagnostyczny	

DCE lub DTE RESET REQUEST

Nagłówek	
Identyfikator typu	238
Kod przyczyny rozłączenia	
Kod diagnostyczny	

DCE lub DTE CLEAR REQUEST

Rys.1. Formaty pakietów

Fig.1. Packets formats

Pakiety danych są numerowane od 0 modulo 8 i wysyłane przez DTE i DCE w kolejności numeracji.

Liczba pakietów danych, które mogą być wysłane bez potwierdzenia, określona rozmiarem okna, może być różna dla obu kierunków transmisji i jest ustalana podczas nawiązywania połączenia.

Pakiet DATA musi być potwierdzony w pakiecie RR, RNR lub w pakiecie DATA z przeciwnego kierunku. Każdy pakiet INTERRUPT musi być potwierdzony pakietem INTERRUPT CONFIRMATION przez odbiorcę przed wysłaniem kolejnego pakietu tego samego typu przez tego samego nadawcę.

Wysyłanie pakietów z zakresu okna do DTE może być wstrzymywane na wniosek odbiorcy pakietu sygnalizującego chwilową niemożność odbioru przesłania pakietu RNR. W tym przypadku pakiety są blokowane w węźle lokalnym dla DTE odbiorcy sygnalizującego niemożność odbioru do czasu nadesłania przed to DTE pakietu RR kasującego ten stan.

Blokowanie pakietów odbywa się także w razie niezgodności okien nadawania i odbierania na jednym kierunku transmisji. Blokowanie pakietów odbywa się w węźle, w którym okno odbioru lokalnego DTE jest mniejsze od okna nadawania zdalnego DTE.

Nadesłanie pakietu o niewłaściwym formacie, przekroczenie rozmiaru okna, niewłaściwa kolejność przesyłanych pakietów lub potwierdzeń, nadesłanie nieoczekiwanego potwierdzenia lub pakietu niedopuszczalnego w danej fazie w połączeniu trwałym jest traktowane jako błąd i powoduje reinicjację transmisji, czyli przejście do fazy reinicjacji [3]. Na fazę transmisji danych i reinicjacji składają się 3 stany kanału:

- stan d1 - Flow Control Ready, w którym odbywa się właściwa transmisja,
- stan d2 - Reset Request, do którego kanał przechodzi po odebraniu pakietu RESET REQUEST z DTE,
- stan d3 - Reset Indication, do którego kanał przechodzi w wyniku obsługi błędu lub odebrania pakietu RESET INDICATION z DCE.

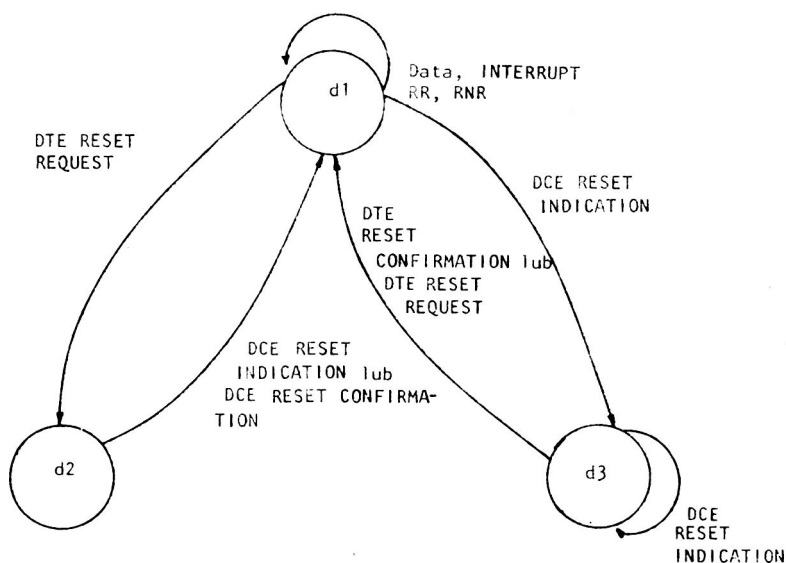
Zasady zmiany stanu kanału w zależności od stanu kanału i odebranego pakietu ilustruje rys. 2.

Odebranie pakietów CLEAR REQUEST w połączeniu wirtualnym, a także pakietu nie w fazie, tzn. innego niż pakiety akceptowane, powoduje przejście do fazy rozłączania [2]. Odebranie pakietu RESTART REQUEST powoduje przejście do fazy restartu [5].

Scenariusz poprawnej transmisji, tzn. wykaz przykładowych pakietów odbieranych przez węzeł podczas transmisji pakietów danych w jednym lub dwu kierunkach połączenia pokazują tabele 2 i 3.

3. AKCJE POZIOMU PAKIETOWEGO WĘZŁA W FAZIE TRANSMISJI DANYCH

Akcje węzła są uzależnione od stanu kanału, źródła i typu pakietu oraz poprawności syntaktyczno-semantycznej pakietu. Pewne akcje są wykonywane standardowo, tzn. niezależnie od typu pakietu, z tym że sposób ich wykonania może się różnić, inne zaś są specyficzne dla określonych pakietów, stanów i podstanów kanału. Skrótowy wykaz akcji podano w tab. 4.



Rys.2. Diagram stanów i przejść między stanami dla fazy transmisji danych

Fig.2. State diagram for data transmission phase

T a b e l a 2

Scenariusz przykładowej poprawnej transmisji podczas przesyłania pakietów DATA w jednym kierunku przy oknie równym 2

Numer odebranego pakietu	Źródło pakietu	Nazwa pakietu	P(S)	P(R)
1	2	3	4	5
1	DTE	DATA (1)	0	0
2	DTE	DATA (2)	1	0
3	DCE	RR	0	1
4	DTE	DATA (3)	2	0
5	DCE	RR	0	3
6	DTE	DATA (4)	3	0
7	DTE	INTERRUPT	-	-
8	DTE	DATA (5)	4	0
9	DCE	INTERRUPT CONFIRMATION	-	-
10	DCE	NR	-	5
11	DTE	DATA (6)	5	0
12	DTE	DATA (7)	6	0

1	2	3	4	5
13	DCE	RR	∅	6
14	DTE	DATA (8)	7	∅
15	DCE	RR	∅	∅
16	DTE	DATA (9)	∅	∅
17	DTE	RR	-	1
18	DTE	CLEAR REQUEST	-	-
19	DCE	CLEAR CONFIRMATION	-	-

T a b e l a 3

Scenariusz przykładowej poprawnej transmisji podczas przesyłania pakietów DATA w obu kierunkach przy oknach równych 4 i 2, odpowiednio gdy DTE odbiera i DTE nadaje

Numer pakietu	Źródło pakietu	Nazwa pakietu	P(S)	P(R)
1	2	3	4	5
1	DTE	DATA (1)	∅	∅
2	DTE	DATA (2)	1	∅
3	DCE	RR	-	2
4	DTE	DATA (3)	2	∅
5	DCE	DATA (1)	∅	2
6	DTE	DATA (4)	3	1
7	DCE	DATA (2)	1	4
8	DTE	DATA (5)	4	2
9	DTE	INTERRUPT	-	-
10	DCE	DATA (3)	2	5
11	DCE	DATA (4)	3	5
12	DCE	INTERRUPT CONFIRMATION	-	-
13	DCE	DATA (5)	4	5
14	DCE	DATA (6)	5	5
15	DTE	RR	-	4
16	DCE	DATA (7)	6	5
17	DCE	DATA (8)	7	5
18	DCE	RR	-	6
19	DCE	DATA (9)	∅	5
20	DCE	DATA (10)	1	5
21	DCE	CLEAR REQUEST	-	-
22	DTE	RR	-	2
23	DTE	CLEAR CONFIRMATION	-	-

Akcje węzła na pakiety i datagramy w stanie Data Transfer -
- Flow Control Ready

Nazwa pakietu	Akcje po otrzymaniu	
	pakietu	datagramu
CALL REQUEST CALL INDICATION	Błąd E#23 (stan p7)	Błąd E#23 (stan p7)
CALL ACCEPTED	Błąd E#23 (stan p7)	Błąd E#23 (stan p7)
CLEAR REQUEST CLEAR INDICATION	Standard (stan p6)	Standard (stan p7)
DCE, DTE lub CLEAR CONFIR- MATION	Błąd E#23 (stan p7)	Błąd E#23 (stan p7)
RESET REQUEST RESET INDICATION	Standard (stan d2)	Standard (stan d3)
DTE RESET CONFIRMATION	Błąd FCE# 27 (stan d3)	Błąd FCE# 27 (stan d3)
Dane, INTERRUPT INTERRUPT CONFIRMATION - poprawne	Standard (stan d1)	Standard (stan d1)
INTERRUPT - bez potwierdze- nia poprzedniego	ignorowany (stan d1)	ignorowany (stan d1)
Dane, INTERRUPT - błędne	Błąd FCE#1, #2, #38, #39, #40 (stan d3)	Błąd FCE#1, #2, #38, #39, #40 (stan d3)
RESTART REQUEST w kanale niezerowym	Błąd FCE#41 (stan d3)	-
INTERRUPT CONFIRMATION	Błąd FCE#43 (stan d3)	Błąd FCE#43 (stan d3)
Pakiet niedozwolony dla PVC	Błąd FCE#35 (stan d3)	Błąd FCE#35 (stan d3)
RESTART REQUEST	Standard (stan r2)	-

Uwaga #n oznacza kod diagnostyczny.

3.1. Rejestracja danych niezbędnych do realizacji akcji fazy transmisji danych

Do prawidłowej realizacji akcji fazy TD niezbędna jest rejestracja:

- numeru pakietu oczekiwanego od DTE-PS₁,
- numeru ostatniego pakietu potwierdzonego przez DTE-PR₁,

- numeru pakietu oczekiwanego od DCE-PS₂,
- numeru ostatniego pakietu potwierdzonego przez DCE-PR₂,
- adresu bufora zawierającego pierwszy pakiet z kolejki pakietów zablokowanych,
- adresu bufora zawierającego ostatni pakiet z kolejki pakietów zablokowanych,
- długości kolejki pakietów zablokowanych.

Dodatkowo jest wymagane rejestrowanie następujących podstawów:

- podstawu RNR - sygnalizującego chwilową niemożność odbioru lokalnego DTE,
- podstawu ICZ - sygnalizującego oczekiwanie na potwierdzenie pakietu przerywania ze zdalnego DTE,
- podstawu ICL - sygnalizującego oczekiwanie na potwierdzenie pakietu przerywania z lokalnego DTE.

3.2. Akcje standardowe

Do akcji standardowych wykonywanych dla każdego pakietu należą:

- kontrola formalnej poprawności pakietu (1),
- translacja pakietów na datagramy lub datagramów na pakiety w zależności od kierunku transmisji (2),
- rejestracja zdarzeń diagnostycznych dotyczących zliczania poprawnych i niepoprawnych pakietów i datagramów oraz zdarzeń blokowania i odblokowywania pakietów w węźle (3),
- wysyłanie pakietów i datagramów w linię (tylko dla pakietów i datagramów poprawnych) (4),
- reinicjacja obwodu logicznego w razie wykrycia błędu (5).

Właściwa kolejność akcji standardowych i specyficznych wymaga wykonania standardowych akcji 4 i 5 po akcjach szczegółowych.

3.3. Akcje po odebraniu pakietu lub datagramu DATA

Dla pakietów i datagramów DATA jest wykonywana:

- kontrola kolejności przesyłania,
- kontrola zakresu okna (dla pakietu),
- kontrola kolejności potwierdzania,
- aktualizacja danych dotyczących numeracji pakietów przesyłanych i potwierdzonych,
- kontrola możliwości odbioru pakietu przez DTE lokalne (dla datagramu),
- blokowanie pakietów, jeżeli okno odbioru lokalnego DTE jest mniejsze od okna nadawania zdalnego DTE (dla datagramu),

- odblokowanie pakietów, zablokowanych w węźle ze względu na przekroczenie okna odbioru, po przesunięciu okna odbioru DTE (dla pakietu).

3.4. Akcje po odebraniu pakietów lub datagramów RR i RNR

Dla pakietów i datagramów RR i RNR wykonuje się:

- rejestrację podstawu RNR dla pakietu RNR,
- kasowanie podstawu RNR dla pakietu RR,
- kontrolę poprawności potwierdzania pakietów DATA w pakietach RR i RNR,
- aktualizację danych dotyczących potwierdzania,
- rejestrację zdarzenia odblokowania pakietów z zakresu lokalnego okna odbioru,
- odblokowanie pakietów po pakiecie RR.

3.5. Akcje po odebraniu pakietu lub datagramu INTERRUPT

Po odebraniu pakietu lub datagramu INTERRUPT jest wykonywana:

- kontrola możliwości akceptowania pakietu ze względu na podstan,
- rejestracja podstawu wtedy, gdy pakiet może być przyjęty:
 - ICZ - dla pakietu,
 - ICL - dla datagramu,
- skasowanie pakietu nie akceptowanego.

3.6. Akcje po odebraniu pakietu lub datagramu INTERRUPT CONFIRMATION

Po odebraniu pakietu lub datagramu INTERRUPT CONFIRMATION wykonuje się:

- skasowanie podstawu:
 - ICL - dla pakietu,
 - ICZ - dla datagramu.

3.7. Akcje po odebraniu pakietu RESET REQUEST lub datagramu RESET INDICATION

Po odebraniu pakietów RESET REQUEST lub RESET INDICATION jest sygnalizowana zmiana fazy na życzenie użytkownika połączenia i przechodzi się do fazy reinicjacji [3].

3.8. Akcje po odebraniu pakietu lub datagramu RESET CONFIRMATION

Potwierdzenie reinicjacji w stanie d1 jest błędne. Odebranie pakietu lub datagramu RESET CONFIRMATION jest traktowane jako błąd fazy i powoduje przejście do fazy reinicjacji [3].

3.9. Akcje po odebraniu pakietu CLEAR REQUEST lub datagramu CLEAR INDICATION

Wykonywanymi akcjami są:

- kontrola możliwości akceptowania pakietu ze względu na rodzaj połączenia,
- przejście do fazy reinicjacji w połączeniu trwałym,
- przejście do fazy rozłączenia w połączeniu czasowym [2].

3.10. Akcje po odebraniu pakietu RESTART REQUEST

Po odebraniu pakietu RESTART REQUEST realizuje się:

- kontrolę możliwości akceptowania pakietu ze względu na numer kanału,
- przejście do fazy reinicjacji dla pakietu RESTART w kanale niezerowym,
- przejście do fazy restartu [5] podczas restartu w kanale zerowym.

3.11. Akcje po odebraniu pakietu RESTART CONFIRMATION

Pakiet RESTART CONFIRMATION w stanie d1 jest pakietem błędnym.

Do wykonywanych akcji należą:

- rozpoznanie rodzaju błędu ze względu na numer kanału,
- przejście do fazy rozłączania z sygnalizacją błędnego pakietu fazy - gdy kanał jest niezerowy,
- przejście do fazy restartu z sygnalizacją błędu restartu - gdy kanał jest zerowy.

4. KONTROLA POPRAWNOŚCI TRANSMISJI I DIAGNOSTYKA BŁĘDÓW

W fazie transmisji danych przeprowadza się liczne kontrole mające na celu wykrycie nieprawidłowości w transmisji i podjęcie odpowiedniej akcji w razie wykrycia błędów.

4.1. Kontrola formalnej poprawności pakietów

Kontrola obejmuje:

- kontrolę pola GFI pakietu:

$$GFI = \begin{cases} xx01 & \text{- dla pakietu DATA,} \\ 0001 & \text{- dla pakietów innych akceptowanych w fazie;} \end{cases}$$

- kontrolę długości pakietu - DP (w oktetach):

$$DP = \begin{cases} = 3 & \text{dla pakietów: INTERRUPT CONFIRMATION, RR, RNR,} \\ & \text{RESET CONFIRMATION,} \\ > 3 & \text{dla pakietów DATA,} \\ = 4 & \text{dla pakietu INTERRUPT,} \\ = 4 \text{ lub } 5 & \text{dla pakietu RESET REQUEST} \end{cases}$$

4.2. Kontrola kolejności przesyłania

Kontrola polega na sprawdzeniu poprawności $P(S)$ z pakietu na spełnienie warunku:

$$PR_2 + WODB > P(S) = PS_1 \quad \text{- dla pakietu,}$$

$$PR_1 + WNAD > P(S) = PS_2 \quad \text{- dla datagramu,}$$

gdzie WODB - okno dla odbioru z lokalnego DTE,

WNAD - okno dla nadawania do lokalnego DTE.

4.3. Kontrola kolejności potwierdzenia

Kontrola polega na sprawdzeniu poprawności $P(R)$ z pakietu na spełnienie warunku:

$$PS_2 \geq P(R) > PR_1 \quad \text{- dla pakietu}$$

$$PS_1 \geq P(R) > PR_2 \quad \text{- dla datagramu.}$$

4.4. Kontrola możliwości akceptowania pakietów INTERRUPT i INTERRUPT CONFIRMATION

Warunkiem akceptowania jest podstan:

$$ICZ = 0 \quad \text{- dla pakietu INTERRUPT,}$$

$$ICL = 1 \quad \text{- dla pakietu INTERRUPT CONFIRMATION,}$$

$$ICL = 0 \quad \text{- dla datagramu INTERRUPT,}$$

$$ICZ = 1 \quad \text{- dla datagramu INTERRUPT CONFIRMATION.}$$

4.5. Wykrywanie pakietów powodujących błąd transmisji (błąd FCE)

Do pakietów tych należą:

- pakiet lub datagram RESET CONFIRMATION,
- pakiet CLEAR REQUEST lub datagram CLEAR INDICATION w połączeniu trwałym,
- pakiet RESTART REQUEST w kanale innym niż zerowy,
- pakiet RESTART CONFIRMATION w kanale innym niż zerowy,
- pakiety nie zidentyfikowane w połączeniu logicznym trwałym.

4.6. Wykrywanie pakietów nie akceptowanych w fazie transmisji danych

Pakiety nie zidentyfikowane jako akceptowane w fazie powodują błąd ERROR (błąd E). Pakiet RESTART CONFIRMATION w kanale zerowym nie jest akceptowany w stanie d1 i powoduje błąd restartu RESTART ERROR (błąd RE).

4.7. Obsługa błędów

Błędy wykrywane w fazie transmisji danych powodują zmianę fazy. Omówione typy błędów są obsługiwane w następujących fazach:

- błąd FCE - w fazie reinicjacji,
- błąd E - w fazie rozłączania,
- błąd RE - w fazie restartu.

W tabeli 5 podano kody przyczyn błędów przesyłane w pakietach RESET INDICATION, CLEAR INDICATION lub RESTART REQUEST.

T a b e l a 5

Kody przyczyn błędów wysyłane do lokalnego i zdalnego DTE

Źródło błędu	Rodzaj błędu	Kod przyczyny dla lokalnego DTE	Kod przyczyny dla zdalnego DTE
Pakiet	FCE E RE	5 23 ₈ 1 błąd lokalny	3 21 ₈ - błąd zdalny
Datagram	FCE E	3 błąd zdalny 21 ₈	5 błąd lokalny 23 ₈

4.8. Rejestracja zdarzeń

Po wykryciu błędu w fazie transmisji danych rejestruje się, w celach diagnostyki i statystyki pracy węzła, następujące zdarzenia:

- błędny pakiet lub datagram danych,
- błędny pakiet,
- błędny datagram,
- blokowanie lub odblokowywanie pakietów lub datagramów w węźle.

Rejestracja zdarzeń polega na wywołaniu podprogramu diagnostyki zdarzeń.

5. IMPLEMENTACJA FAZY TRANSMISJI DANYCH W OPROGRAMOWANIU POZIOMU PAKIETOWEGO WĘZŁA SIECI MSK

Faza transmisji danych (TD) wykorzystuje dane o połączeniu logicznym zarejestrowane w fazie nawiązywania połączenia w tzw. opisie kanału logicznego (OKL). Wymagane dla fazy TD dane opisane w p. 3.1 (rys. 3) również umieszczono w OKL. Polom przewidzianym dla tych danych przypisano następujące modyfikatory:

P1SE - numer pakietu oczekiwanego od DTE - 1 bajt,
P1RE - numer ostatniego pakietu potwierdzonego przez DTE - 1 bajt,
P2SE - numer pakietu oczekiwanego od DCE - 1 bajt,
P2RE - numer ostatniego pakietu potwierdzonego przez DCE - 1 bajt,
APBWE - adres bufora zawierającego pierwszy zablokowany pakiet - 2 bajty,
AOBWE - adres bufora zawierającego ostatni zablokowany pakiet - 2 bajty,
DKBWE - długość kolejki pakietów zablokowanych - 1 bajt,
ROZKE - rozmiar okna odbioru i nadawania od i do DTE - 1 bajt.

Podstany są rejestrowane w dwubajtowym polu OKL oznaczonym WFAZE - na pozycjach określonych na rysunku.

Oprogramowanie fazy transmisji danych składa się z jednego modułu wykorzystującego 10 podprogramów własnych modułu X.25 oraz 3 podprogramy udostępnione przez oprogramowanie podstawowe węzła. Strukturę oprogramowania pokazano na rys. 4.

Moduł fazy ma 2 wejścia:

- z fazy nawiązywania połączenia - DATRE,
- z fazy reinicjacji - DATRIL

i 7 wyjść:

- do fazy reinicjacji w celu obsługi stanu d2 - REREQF,
- do fazy reinicjacji w celu obsługi stanu d3 - REINDF,
- do fazy reinicjacji w celu zmiany fazy - ORESEF,
- do fazy rozłączenia - CLERE,
- do fazy restartu w celu zmiany fazy - RESTRH,
- do fazy restartu w celu obsługi błędu restartu - OBSR2H,
- do wyjścia z oprogramowania poziomu pakietowego WYX25E.

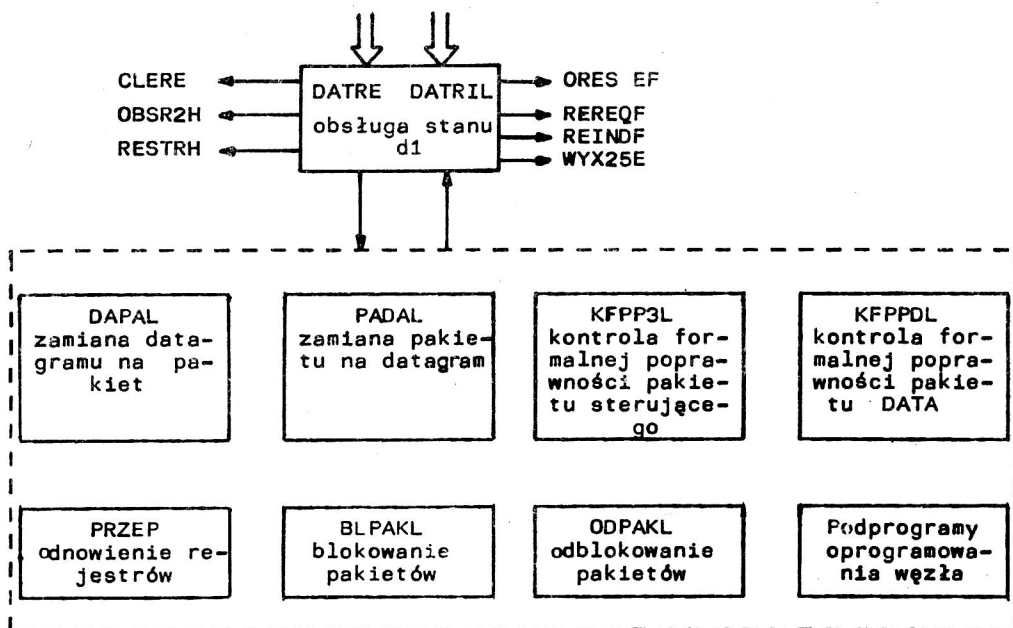
okno odbioru		okno nadawania		ROZ0KE
PS ₁				P1SE
PR ₂				P2RE
PS ₂				P2SE
PR ₁				P1RE
				WFAZE
	IZW	ILH	RNRH	
długość kolejki				DKBWE
adres pierwszego bufora zablokowanego				APBWE
adres ostatniego bufora zablokowanego				AOBWE

Rys.3. Pola w OKL wymagane dla fazy transmisji danych

Fig.3. Logical Channel Description fields designated for data transmission phase

Po zmianie fazy w rejestrze RØ jest przekazywany parametr identyfikujący akcję w danej fazie. W razie zmiany fazy na skutek odebranego poprawnego pakietu RØ = Ø, w razie zmiany fazy w wyniku błędu bardziej znaczący bajt RØ zawiera kod diagnostyczny, a mniej znaczący bajt RØ zawiera przyczynę błędu wysyланą do lokalnego DTE.

Moduł fazy transmisji danych, podobnie jak moduły innych faz, jest wykonywany jako fragment nieprzerywany. Struktury danych OKL są odrębne dla każdego połączenia, co umożliwia wykorzystywanie jednej kopii programu dla różnych połączeń. Używane dane robocze są wykorzystywane w okresie działania fazy i nie jest wymagane ich zachowanie do powtórzonego wywołania.



Rys.4. Struktura oprogramowania fazy transmisji danych

Fig.4. Programming structure for data transmission phase

LITERATURA

- [1] Draft Revised Recommendation X.25 Introduction and Packet Level, CCITT Study Group VII, Contribution, No 439, Geneva, Nov. 79.
- [2] D u b i e l e w i c z I., Kasowanie połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).
- [3] D u b i e l e w i c z I., K o l e ś n i k K., Reinicjacja transmisji danych w sieci komputerowej MSK (w tym samym zeszycie).
- [4] F r y ż l e w i c z Z., Protokół pakietowy w sieci komputerowej MSK (w tym samym zeszycie).
- [5] G o e t z J., Restart połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).
- [6] F r y ż l e w i c z Z., M i k u l s k a E., Zestawianie połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).

Praca wpłynęła do Redakcji 18.12.1980 r.

Po poprawieniu 3.03.1982 r.

DATA TRANSMISSION IN THE MSK COMPUTER NETWORK

The paper deals with packet level in the Data Transmission Phase in the MSK node.

The packet level works with the X.25 protocol. The principles of data transmission, actions of the phase and ways of error detection are described.

Programming implementation is also presented.

Verified by Ruta Czaplińska

ПЕРЕДАЧА ДАННЫХ В СЕТИ ЭВМ MSK

В статье представлена работа пакетного уровня в узле сети ЭВМ MSK во время передачи данных. Пакетный уровень работает с протоколом X.25. Определены принципы передачи данных, рассмотрены фазовые действия, метод проверки ошибок, а также имплементация в программном обеспечении протокола.

Проверила Малгожата Хейдрих

*Sieć komputerowa,
protokół pakietowy*

Iwona DUBIELEWICZ^{*)}, Krystyna KOLEŚNIK^{*)}

Reinicjacja transmisji danych w sieci komputerowej MSK

W pracy opisano działanie poziomu pakietowego w węźle sieci komputerowej MSK w fazie reinicjacji transmisji danych. Określono zasady reinicjacji transmisji danych, omówiono akcje fazy, sposób kontroli błędów oraz implementację fazy w oprogramowaniu sieci MSK.

1. WSTĘP

Celem fazy reinicjacji jest obsługa reinicjacji transmisji na skutek błędu w fazie transmisji danych lub na życzenie jednego z użytkowników połączenia. Reinicjacja transmisji może być dokonywana w połączeniu trwałym lub czasowym. Reinicjacja kasuje aktualną transmisję i w kanałach należących do połączenia ustala stan początkowy. W razie reinicjacji transmisji na skutek błędu przeprowadzana jest kontrola czasu potwierdzenia reinicjacji (limit czasowy - ang. time out).

Po przekroczeniu czasu jest dokonywana retransmisja w odpowiednim kanale. Po wyczerpaniu limitu retransmisji połączenie wirtualne jest rozłączane. W połączeniu stałym przekroczenie limitu retransmisji powoduje przejście do fazy transmisji danych. W obu przypadkach jest sygnalizowana niesprawność połączenia.

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

2. ZASADY TRANSMISJI W FAZIE REINICJACJI

Reinicjacja na żądanie jednego z użytkowników połączenia generującego pakiet RESET REQUEST powoduje zmianę stanu lokalnego kanału logicznego na stan DTE Reset Request (d2) oraz zmianę stanu zdalnego kanału logicznego na stan DCE Reset Indication (d3). Obydwa kanały wracają do stanu Data Transfer (d1) po odebraniu pakietu RESET CONFIRMATION wygenerowanego przez zdalne DTE lub po odebraniu pakietu RESET REQUEST nadesłanego z przeciwnego kierunku, co traktowane jest jako potwierdzenie kolizyjne.

Reinicjacja transmisji po wykryciu błędu polega na wysłaniu pakietów RESET INDICATION do nadawcy i odbiorcy pakietu z podaniem wykrytej przyczyny i źródła błędu. W tym przypadku obydwa kanały logiczne wchodzące w skład połączenia przechodzą do stanu d3. Kanał zdalny zachowuje się w tym przypadku tak jak po otrzymaniu RESET INDICATION na żądanie użytkownika. Kanał lokalny, który wykrył błąd i wygenerował stan d3 wraca do stanu d1 po otrzymaniu potwierdzeń lub żądań kolizyjnych z obu kierunków. Nadesłanie pakietu potwierdzenia na pakiet RESET INDICATION wygenerowany w węźle jest ograniczone czasem. Po upływie wyznaczonego limitu czasowego jest wykonywana retransmisja w kierunku, który nie nadesłał odpowiedzi. Retransmisje są powtarzane aż do wyczerpania całkowitego czasu lub liczby retransmisji, po czym w kanale wchodzącym w skład połączenia czasowego jest generowane rozłączenie [2], zaś przy połączeniu stałym kanał przechodzi do stanu d1. Reinicjacja transmisji w kanale (zarówno zdalnym, jak i lokalnym) polega na skasowaniu w kolejkach buforów wszystkich pakietów należących do wznowianej transmisji oraz na wyzerowaniu struktur danych związanych z fazą transmisji danych [5] (tj. PS_1 , PS_2 , PR_1 , PR_2). Pakiety kierowane do kanału będącego w fazie reinicjacji (stan d2 lub d3) są kasowane, jeżeli przychodzą z kierunku przeciwnego do żądającego reinicjacji lub też są kasowane z sygnalizacją błędu, jeżeli przychodzą z kierunku, na którego żądanie jest realizowana reinicjacja.

Jeżeli po zakończeniu reinicjacji na skutek błędu w węźle lokalnym lokalne DTE zacznie przysyłać pakiety danych przed nadejściem oczekiwanego potwierdzenia reinicjacji ze zdalnego DTE, pakiety te po standardowej obsłudze jak w stanie d1 będą blokowane w węźle lokalnym do czasu otrzymania potwierdzenia z podsieci (blokowanie datagramów).

Jeżeli po zakończeniu reinicjacji w węźle zdalnym, zdalne DTE zacznie przysyłać pakiety danych przed zakończeniem reinicjacji w lokalnym DTE (lokalnym dla węzła prowadzącego obsługę błędów), pakiety ze zdalne-

Reinicjacja transmisji danych w sieci komputerowej MSK

co DTE nadesłane z podsieci będą blokowane w węźle prowadzącym obsługę błędów (także po obsłudze datagramu odebranego jak w stanie d1) do czasu otrzymania potwierdzenia reinicjacji od lokalnego użytkownika kanału (blokowanie pakietów).

3. AKCJE POZIOMU PAKIETOWEGO WĘZŁA W FAZIE REINICJACJI

W fazie reinicjacji akcje są uzależnione od rodzaju wywołania, parametrów wywołania fazy, stanu i podstawu kanału logicznego oraz źródła i typu pakietu. Akcje standardowe są wykonywane jak w fazie transmisji danych. Skrótowy wykaz akcji podano w tabeli 1.

Akcje szczegółowe opisano poniżej.

3.1. Rejestracja danych niezbędnych do realizacji akcji fazy reinicjacji

Do rejestrowanych danych fazy reinicjacji należą:

- licznik liczby retransmisji do lokalnego DTE,
- licznik liczby retransmisji do zdalnego DTE,
- przyczyna reinicjacji,
- podstan RZH sygnalizujący oczekiwanie na potwierdzenie pakietu reinicjacji ze zdalnego DTE,
- podstan RLH sygnalizujący oczekiwanie na potwierdzenie pakietu reinicjacji z lokalnego DTE.

3.2. Akcje podczas reinicjacji na żądanie jednego z użytkowników połączenia

Wykonywanymi akcjami są:

- zmiana stanu na d2 (po pakiecie RESET REQUEST) lub d3 (po pakiecie RESET INDICATION),
- zerowanie wskaźników stanu nadawania i potwierdzania (PS_1 , PS_2 , PR_1 , PR_2) w opisie kanału logicznego danego połączenia,
- kasowanie kolejki pakietów-zablokowanych ze zwrotem buforów do puli buforów wolnych i zerowaniem danych o kolejce w OKL,
- kasowanie pakietów należących do reinicjowanego połączenia we wszystkich kolejkach węzła.

3.3. Akcje podczas reinicjacji na skutek błędu

Do akcji wykonywanych po wykryciu błędu należą:

- zmiana stanu kanału na d3,

T a b e l a 1

Akcje podejmowane przez węzeł w odpowiedzi na pakiety i datagramy w stanie
Data Transfer dla stanów Reset Request i Reset Indication

Rodzaj pakietu	Reset Request		Reset Indication			
	po odebraniu pakietu	po odebraniu datagramu	lokalny		zdalny	
			po odebraniu pakietu	po odebraniu datagramu	po odebraniu pakietu	po odebraniu datagramu
CALL REQUEST CALL INDICATION CALL ACCEPTED DTE CLEAR CONFIRMATION	błąd FCE # 23 (stan d3)	błąd FCE # 23 (stan d3)	błąd FCE # 23 (stan d3)	błąd FCE # 23 (stan d3)	błąd FCE # 23 (stan d3)	błąd FCE # 23 (stan d3)
CLEAR REQUEST CLEAR INDICATION	standard (stan p6)	standard (stan p7)	standard (stan p6)	standard (stan p7)	standard (stan p6)	standard (stan p7)
RESET REQUEST RESET INDICATION	odrzucony lub standard po li- micie czasu (stan d2)	standard (stan d1)	standard (stan d3 lub d1 po potwierdzeniu zdalnym)	standard (stan d3 lub d1 po potwierdzeniu lokalnym)	standard (stan d1)	odrzucony lub standard po li- micie czasu (stan d3)
DCE lub DTE RESET CONFIRMATION	błąd FCE # 28 (stan d3)	standard (stan d1)	standard z ewentualnym od- blokowaniem pa- kietów (stan d1 po potwierdze- niu zdalnym lub d3)	standard z ewentualnym od- blokowaniem da- tagramów (stan d1 po potwier- dzeniu zdalnym lub d3)	standard (stan d1)	standard (stan d1)
Data INTERRUPT INTERRUPT CONFIRMATION	błąd FCE # 28 (stan d3)	odrzucony (stan d2)	odrzucony lub blokowany po lokalnym po- twierdzeniu (stan d3)	odrzucony lub blokowany po zdalnym po- twierdzeniu (stan d3)	odrzucony (stan d3)	błąd FCE (stan d3)
RESTART REQUEST w kanale # 0	błąd FCE # 41 (stan d3)	błąd FCE # 41 (stan d3)	odrzucony (stan d3)	odrzucony (stan d3)	błąd # 41 (stan d3)	błąd # 41 (stan d3)
Pakiet niewłaściwy dla PVC	błąd FCE # 35 (stan d3)	błąd FCE # 35 (stan d3)	odrzucony (stan d3)	odrzucony (stan d3)	błąd # 35 (stan d3)	błąd # 35 (stan d3)

PVC - kanał stały,

FCE - błąd sterowania przepływem.

Reinicjacja transmisji danych w sieci komputerowej MSK

- ustawienie podstanów RLH i RZH,
- wysyłanie datagramu RESET INDICATION z podaniem źródła i przyczyny błędu,
- pobieranie bufora i wygenerowanie pakietu RESET INDICATION z podaniem źródła i przyczyny błędu,
- w razie braku bufora wygenerowanie fragmentu zawieszzonego oczekującego na bufor,
- po wysłaniu pakietu i datagramu wygenerowanie odpowiednich fragmentów zawieszonych sygnalizujących upływ limitu czasowego,
- akcje dotyczące zmiany stanu kanału jak podczas reinicjacji na żądanie (p. 3.2).

Uwaga - dla kanału lokalnego jest ustawiany tylko limit czasowy dla lokalnego DTE i podstan RLH.

3.4. Akcje w stanie Reset Indication w razie, gdy stan Reset Indication jest wygenerowany

Po otrzymaniu datagramów RESET INDICATION lub RESET CONFIRMATION traktowanych jako potwierdzenia (w tym RESET INDICATION jako potwierdzenie podczas kolizji) są wykonywane następujące akcje:

- kasowanie podstanu RZH,
- kasowanie fragmentu zawieszzonego na limit czasowy dla zdalnego DTE,
- rejestracje zdarzenia kasowania limitu czasowego,
- zwrot bufora,
- zmiana stanu na d2, jeżeli RLH = $\emptyset$,
- jeżeli RLH = $\emptyset$, ewentualne odblokowanie datagramów nowej transmisji, którą rozpoczęło lokalne DTE.

Po otrzymaniu pakietów RESET REQUEST lub RESET CONFIRMATION, także traktowanych jako potwierdzenia, są wykonywane następujące akcje:

- kasowanie podstanu RLH,
- kasowanie fragmentu zawieszzonego na limit czasowy dla lokalnego DTE,
- rejestracja zdarzenia kasowania limitu czasowego,
- zwrot bufora,
- zmiana stanu na d1, jeżeli RZH = $\emptyset$,
- gdy RZH = $\emptyset$ ewentualne odblokowanie pakietów nowej transmisji, którą już rozpoczęło zdalne DTE.

Uwaga: Po odebraniu pakietu lub datagramu DATA, INTERRUPT lub RNR akcje są następujące:

- dla podstanów RLH = 1 lub RZH = 1 odpowiednio pakiety lub datagramy są kasowane,
- dla podstanu RZH = 1 i po odebraniu pakietu pakiet jest przetwarzany tak jak w stanie d1 [5] i jako datagram jest blokowany w węźle,

- dla podstawu $RLH = 1$ i po odebraniu datagramu, datagram jest przetwarzany tak jak w stanie d1 i jako pakiet jest blokowany w węźle.

Po odebraniu pakietu CLEAR REQUEST lub CLEAR INDICATION następuje przejście do fazy rozłączenia [2]. Odebranie pakietu RESTART REQUEST powoduje przejście do fazy restartu [4]. Odebranie w stanie d3 pakietów lub datagramów innych niż wymienione jest traktowane jako błąd i powoduje wysłanie pakietu RESET do nadawcy pakietu.

3.5. Akcje w stanie Reset Indication ustalonym na skutek odebrania datagramu RESET INDICATION

Po odebraniu pakietu RESET CONFIRMATION lub RESET REQUEST, traktowanych jako potwierdzenie, następuje:

- zmiana stanu kanału na d1,
- wykonanie akcji standardowych związanych z wysłaniem potwierdzenia do podsieci.

Pakiety DATA, RR, RNR, INTERRUPT i INTERRUPT CONFIRMATION jako dopuszczalne pakiety starej transmisji są kasowane i następuje zwrot bufora. Odebranie pakietu CLEAR REQUEST powoduje przejście do fazy rozłączenia [2]. Odebranie pakietu RESTART REQUEST powoduje przejście do fazy restartu [4]. Inne pakiety są traktowane jako błędy sterowania przepływem (ang. FCE) i powodują obsługę błędu w fazie reinicjacji.

Z datagramów jest akceptowany RESET INDICATION nadesłany po upływie limitu czasowego. Powoduje on akcję taką, jak w stanie Flow Control Ready po odebraniu pakietu RESET INDICATION. RESET INDICATION nadesłany przed upływem limitu czasowego jest kasowany. Inne datagramy są traktowane jako błędne.

3.6. Akcje w stanie Reset Request

Po odebraniu datagramu RESET CONFIRMATION lub RESET INDICATION, traktowanych jako potwierdzenia, następuje:

- zmiana stanu kanału na d1,
- akcje standardowe związane z wysłaniem datagramu.

Nadesłanie datagramu CLEAR INDICATION powoduje przejście do fazy rozłączenia. Datagramy DATA, RR, RNR, INTERRUPT, INTERRUPT CONFIRMATION są kasowane.

Wszystkie pakiety, z wyjątkiem RESET INDICATION po limicie czasowym, są traktowane jako błędne i powodują przejście do obsługi błędu reinicjacji. RESET INDICATION nadesłany przed upływem limitu czasowego jest kasowany, po upływie limitu czasowego jest obsługiwany tak jak RESET INDICATION w stanie d1.

3.7. Akcje wykonywane po wywołaniu fragmentu zawieszono- go na skutek upływu limitu czasowego dla lokalnego DTE

Fragment zawieszony wykonuje następujące akcje:

- kontrolę liczby retransmisji lokalnych lub limitu czasowego na retransmisję,
- gdy limit nie jest wyczerpany, wygenerowanie pakietu RESET INDICATION i wykonywanie akcji związanych z wysłaniem pakietu do lokalnego DTE jak w przypadku obsługi błędu (p. 3.3),
- gdy limit jest wyczerpany, następuje sygnalizacja niesprawności linii i przejście do fazy rozłączenia dla połączenia czasowego, a do fazy transmisji danych dla połączenia trwałego.

3.8. Akcje wykonywane po wywołaniu fragmentu zawieszono- go na skutek upływu limitu czasowego dla zdalnego DTE

Fragment zawieszony wykonuje następujące akcje:

- kontrolę liczby retransmisji zdalnych lub limitu czasowego na retransmisję,
- gdy limit nie jest wyczerpany, wygenerowanie datagramu RESET INDICATION i wykonywanie akcji związanych z wysłaniem datagramu do zdalnego DTE jak w przypadku obsługi błędu,
- gdy limit jest wyczerpany, następuje sygnalizacja niesprawności linii i przejście do fazy rozłączenia dla połączenia czasowego, a do fazy transmisji danych dla połączenia trwałego.

4. KONTROLA POPRAWNOŚCI FAZY REINICJACJI POŁĄCZENIA

Na kontrolę poprawności przebiegu fazy reinicjacji połączenia składają się:

- a) kontrola poprawności otrzymanego pakietu/datagramu
 - formalna,
 - semantyczna,
- b) kontrola poprawności kontekstu dla otrzymanego pakietu/datagramu,
- c) kontrola upływu limitu czasowego,
- d) kontrola limitu retransmisji.

Dwie pierwsze kontrole są prowadzone dla każdego pakietu/datagramu w każdym stanie kanału logicznego. W razie stwierdzenia niepoprawności pakietu/datagramu lub uzyskania go nie w kontekście dokonuje się:

- zapisu informacji o błędzie w polu CAUSE-OKL,
- zmiany stanu kanału na d3, jeśli stan był inny (por. tab. 1) i są prowadzone akcje jak podano w punkcie 3.3),
- ignoruje się wykryty błąd, jeśli jest stan d3.

Kontrola upływu limitu czasowego jest możliwa dzięki prowadzeniu odliczania czasu przez system sterujący węzła. Po stwierdzeniu upływu limitu czasowego dokonuje się retransmisji pakietu/datagramu, aktualizując odpowiedni licznik retransmisji.

Stan liczników retransmisji jest porównywany z limitem retransmisji (parametr instalacyjny podobnie jak limit czasowy).

Jeśli limit zostanie przekroczony, to w zależności od typu połączenia dokonuje się:

- dla połączenia trwałego (PVC) zmiany fazy na fazę transmisji danych (stan kanału d1),
- dla połączenia czasowego (VC) zmiany fazy na fazę rozłączenia połączenia (stan kanału p7).

W trakcie opisanej kontroli poprawności fazy reinicjacji połączenia, jak również dla celów statystyki i diagnostyki pracy węzła i podsieci, prowadzi się rejestrację następujących zdarzeń, stwierdzonych w fazie reinicjacji:

- błędny datagram,
- błędny pakiet,
- reinicjacja łącza,
- odbiór pakietu RESET REQUEST,
- odbiór datagramu RESET INDICATION,
- brak wolnego bufora,
- rozłączenie z powodu wyczerpania limitu retransmisji,
- rozłączenie z powodu przeciążenia węzła,
- zakończenie fazy reinicjacji dla DTE i podsieci.

Informacje o zaistniałych zdarzeniach podaje się programom diagnostycznym węzła.

5. OPROGRAMOWANIE FAZY REINICJACJI POŁĄCZENIA

Oprogramowanie fazy reinicjacji połączenia składa się z 3 podmodułów, wykorzystujących 7 podprogramów własnych modułu X.25 oraz 6 podprogramów udostępnionych przez oprogramowanie podstawowe węzła.

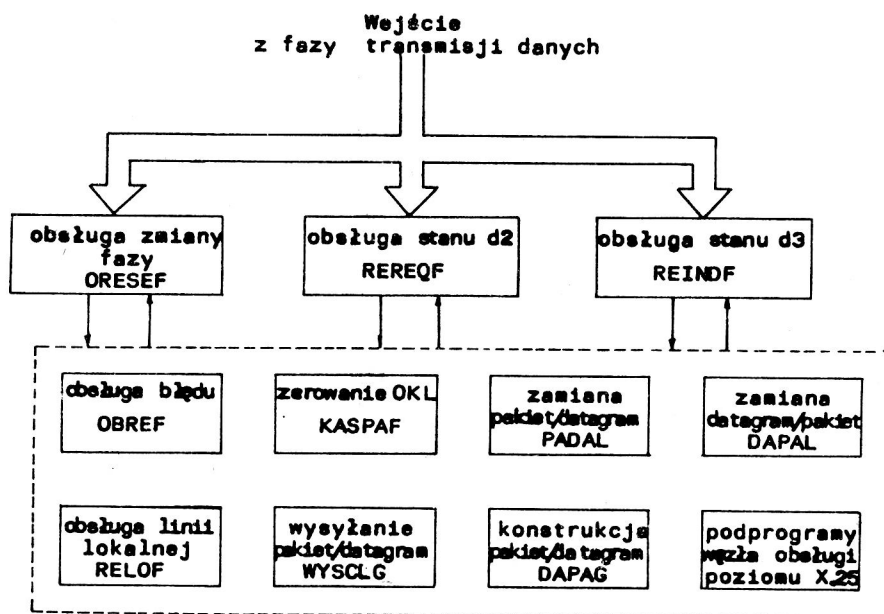
Sterowanie do dowolnego z 3 podmodułów jest przekazywane bezpośrednio z podmodułu fazy transmisji danych, a dane są podawane w rejestrach ogólnych procesora węzła.

Podmoduły realizują następujące funkcje:

- obsługę zmiany fazy - ORESEF,
- obsługę stanu d2 - REREQF,
- obsługę stanu d3 - REINDF.

Każdy z podmodułów jest jednowejściowy.

Strukturę oprogramowania przedstawiono na rys. 1.



Rys.1. Struktura oprogramowania fazy reinicjacji łącza
Fig.1. Programming structure of reset transmission phase

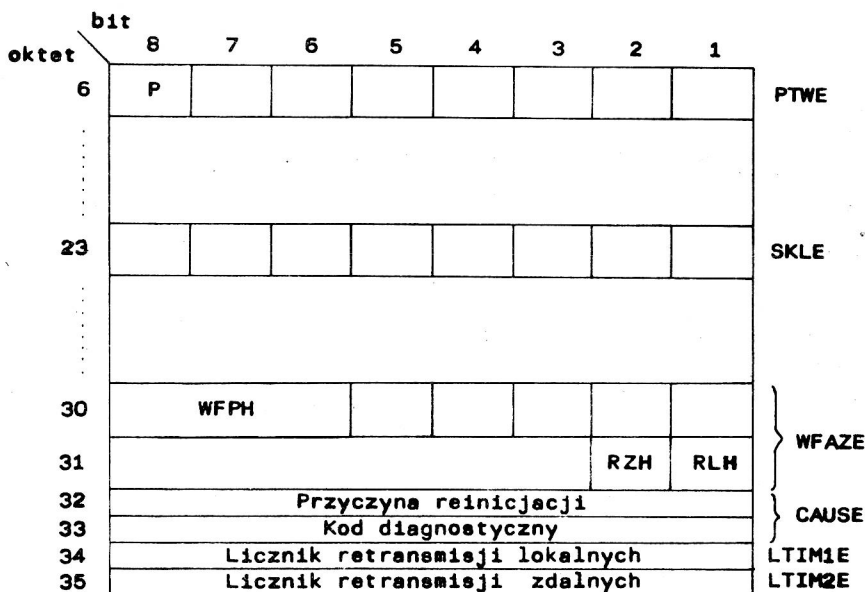
Dane między podmodułami są przekazywane przez wykorzystanie wspólnego obszaru danych, na który składają się:

- dane instalacyjne węzła,
- rejestry ogólnego przeznaczenia minikomputera SM-3,
- opis kanału logicznego OKL,
- bufor z datagramem/pakiem.

Ta sama zasada komunikacji została przyjęta dla współpracy podmoduły-podprogramy.

Struktury danych zawarte w opisie kanału logicznego, które są konieczne i wykorzystywane w fazie reinicjacji połączenia, przedstawiono na rys. 2.

Stan kanału SKLE oraz wskaźniki WFAZE są wykorzystywane do przekazywania sterowania do odpowiedniego podmodułu. W polu WFAZE wykorzystuje się dwa bity oznaczone FLH i RZH, które oznaczają odpowiednio prowadze-



Rys.2. Struktury danych OKL wykorzystywane w fazie reinicjacji
Fig.2. Data fields of LCD used in reset phase

nie retransmisji pakietów RESET INDICATION (do DTE) i datagramów RESET INDICATION (do podsieci). Pole CAUSE przechowuje informację o przyczynie reinicjacji, a pola LTIM1E i LTIM2E zawierają odpowiednio liczniki retransmisji do DTE i podsieci.

Istotny dla dazy reinicjacji jest również wskaźnik P pola PTWE, który informuje o typie połączenia, z którym jest związany OKL ($P=1$ - kanał trwały PVC, $P=0$ - kanał czasowy VC). Wskaźnik ten decyduje o sposobie zakończenia fazy reinicjacji w przypadku osiągnięcia limitu retransmisji:

- $P=1$ - następuje przejście do fazy transmisji danych,
- $P=0$ - przejście do fazy rozłączenia połączenia.

Przyjęte struktury danych są wystarczające do poprawnej realizacji fazy reinicjacji, zgodnej z podanymi w pracy [5] grafami przejść między stanami i tabelą przejść (tab. 1) po otrzymaniu określonego pakietu z DTE.

Aby jednak zrealizować wymaganie retransmisji pakietów/datagramów, system sterujący węzła musi udostępnić następujące zasoby:

- bufory dla konstrukcji datagramu/pakietu,
- odliczanie czasu,

oraz dane globalne węzła:

- limit czasowy dla d3-TD3E,
- czas oczekiwania na bufor - TBUE.

Wymaganie odliczania czasu jest związane z koniecznością określenia limitu czasowego, po którym należy dokonać retransmisji. Realizacja retransmisji wykorzystuje tzw. fragment zawieszony, będący podprogramem podmodułu REINDF, który jest inicjowany po podanym czasie. Należy zwrócić uwagę, że w razie odliczania limitów czasowych dla wielu połączeń moment inicjowania konkretnego fragmentu zawieszonego może być opóźniony w stosunku do czasu żadanego. Prowadzić to może do resynchronizowania prowadzonych retransmisji.

Niemожność uzyskania buforów do retransmisji powoduje również żądanie odliczania czasu, po upływie którego ponawia się żądanie bufora. Jednocześnie zwiększa się odpowiednio licznik retransmisji, jeśli czas oczekiwania na bufor jest równy krotności limitu czasowego.

Wyłączność (synchronizacja) dostępu do wspólnych danych jest zapewniona przez nieprzerywalność wykonywania poszczególnych podmodułów/podprogramów. Umożliwia to wykorzystywanie oprogramowania X.25 współbieżnie dla wielu połączeń.

Aby uzyskać pewną mobilność oprogramowania, jak również ze względu na niezdefiniowanie pewnych wielkości przez protokół X.25 w oprogramowaniu fazy rozłączenia, przyjęto jako parametry następujące dane:

- identyfikator ogólny formatu pakietu,
- limit retransmisji pakietów/datagramów RESET INDICATION,
- limit czasowy (DCE-TD3E) dla stanu d3,
- czas oczekiwania na bufor.

Dane te są dostępne jako parametry instalacyjne węzła we wspólnym obszarze danych i mogą być zmienione w kolejnych generacjach oprogramowania węzła podstawowego.

LITERATURA

- [1] Draft Revised Recommendation X.25 Introduction and Packet Level, CCITT Study Group VII, Contribution No 439, Geneva, Nov 79.
- [2] D u b i e w i c z I., Kasowanie połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).
- [3] F r y ż l e w i c z Z., Protokół pakietowy w sieci komputerowej MSK (w tym samym zeszycie).
- [4] G o e t z J., Restart połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).
- [5] K o l e ś n i k K., Transmisja danych w sieci komputerowej MSK (w tym samym zeszycie).

- (6) Mikulska E., Fryźlewicz Z., Zestawianie połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).

Praca wpłynęła do Redakcji 18.12.1980 r.

Po poprawieniu 3.03.1982 r.

RESET OF DATA TRANSMISSION IN THE MSK COMPUTER NETWORK

The paper deals with packet level in the Data Transmission Phase in MSK node. The rules of reset of data transmission, actions of the phase and ways of errors detection are described. The programming implementation is also presented.

Verified by Ruta Czaplińska

ПОВТОРНОЕ ИНИЦИИРОВАНИЕ ПЕРЕДАЧИ ДАННЫХ В СЕТИ ЭВМ MSK

В статье представлена работа пакетного уровня в узле сети ЭВМ MSK во время повторного инициирования передачи данных. Определены принципы повторного инициирования данных, рассмотрены фазовые действия, метод проверки ошибок, а также имплементация в программном обеспечении протокола.

Проверила Малгожата Хейдрих

*Sieć komputerowa, węzeł,
protokół komunikacyjny*

Iwoña DUBIELEWICZ^{*)}

Kasowanie połączeń logicznych w sieci komputerowej MSK

W artykule przedstawiono przebieg fazy rozłączenia drogi logicznej w sieci komputerowej MSK na poziomie protokołu pakietowego X.25. Podano przyczyny i sposób realizacji rozłączenia w węzle sieci MSK. Omówiono implementację programową fazy rozłączenia, rozważając zalety przyjętego rozwiązania.

W przebiegu transmisji danych między dwoma użytkownikami wyróżnia się trzy etapy:

- faza nawiązania połączenia,
- faza transmisji danych,
- faza rozłączenia połączenia.

Dwie pierwsze fazy dla sieci komputerowej MSK zostały przedstawione odpowiednio w pracach [6] i [4]. Celem niniejszej pracy jest prezentacja realizacji fazy rozłączenia na poziomie protokołu pakietowego X.25 w sieci MSK.

Punktem wyjścia do przeprowadzenia fazy rozłączenia jest istnienie połączenia logicznego (drogi logicznej), założonego w fazie nawiązania połączenia. Połączenie to w sieci MSK jest stałe przez cały okres prowadzenia transmisji danych. W skład połączenia wchodzi dwaj użytkownicy (DTE) oraz węzły podsieci komunikacyjnej. Wśród węzłów wyróżnia się węzeł początkowy - DCE, który jest związany z użytkownikiem - DTE żądającym połączenia, węzeł końcowy - DCE związany z użytkownikiem - DTE,

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

wywoływany do transmisji. Inne węzły wchodzące w skład połączenia, to węzły pośrednie; węzły połączone z DTE, będą nazywane węzłami związanymi z DTE. W sieci MSK transmisja danych w podsieci jest realizowana przez wysyłanie datagramów, przez które rozumie się rozszerzone o nagłówek (nadawcę) pakiety protokołu X.25.

1. CHARAKTERYSTYKA I FUNKCJE FAZY ROZŁĄCZENIA

Celem fazy rozłączenia jest likwidacja połączenia logicznego (drogi logicznej), zestawionej między dwoma użytkownikami przez pakiety REQUEST/INCOMING CALL. Rozłączenie odbywa się na żądanie dowolnego użytkownika połączenia lub przez podsieć komunikacyjną w razie stwierdzenia nieusuwalnych błędów transmisji, np. odbiór nieznanych pakietów, niemożność reinicjacji łącza, awaria linii transmisji itp.

Rozłączenie połączenia logicznego przebiega w dwu etapach:

- przesyłanie pakietu/datagramu z informacją o żądaniu rozłączenia, tzn. CLEAR REQUEST lub CLEAR INDICATION, opcjonalnie z przyczyną rozłączenia,

- przesyłanie pakietu/datagramu potwierdzenia przyjęcia żądania rozłączenia, tzn. CLEAR CONFIRMATION lub CLEAR INDICATION.

Zarejestrowanie przez podsieć komunikacyjną takiej wymiany informacji powoduje likwidację połączenia logicznego, tzn. usunięcie opisów tej drogi z węzłów: początkowego, końcowego i pośrednich, wchodzących w skład tego połączenia.

Istnieje również "awaryjna" możliwość likwidacji połączenia logicznego. Jeśli w ustalonym czasie nie stwierdzi się wspomnianej wymiany informacji, to podsieć automatycznie może dokonać likwidacji tego połączenia.

W razie awarii elementu podsieci, jeśli jest to możliwe, przed dokonaniem rozłączenia połączeń wykorzystujących uszkodzony element, podsieć przesyła informacje o awarii do użytkowników połączeń, a następnie likwiduje wszystkie połączenia.

2. PRZEBIEG FAZY ROZŁĄCZENIA

Żądanie rozłączenia połączenia może wysłać dowolny użytkownik - DTE w dowolnej chwili przez wysłanie pakietu CLEAR REQUEST. Odebrany w węźle związanym z DTE (węzeł źródłowy) pakiet zostaje zamieniony na datagram CLEAR INDICATION i przesyłany po ustalonej drodze do drugiego użytkownika - DTE. Jednocześnie informacja o zainicjowaniu fazy rozłączenia

nia zostaje zarejestrowana w opisie połączenia (w którym uzyskano pakiet CLEAR REQUEST) jako nowy stan tego połączenia.

Węzły pośrednie, przez które następnie przesyła się datagram CLEAR INDICATION, również przechowują informację o zapoczątkowaniu rozłączenia. Węzeł związany z DTE adresatem rozłączenia (węzeł docelowy) po odebraniu datagramu CLEAR INDICATION zamienia go na pakiet CLEAR INDICATION i przesyła do użytkownika - DTE. Użytkownik po odebraniu tego pakietu przesyła zwrótnie potwierdzenie rozłączenia w postaci pakietu CLEAR CONFIRMATION, który zamieniony w węźle docelowym na datagram CLEAR CONFIRMATION zostaje skierowany do węzła źródłowego. Jednocześnie w węźle docelowym likwiduje się opis połączenia.

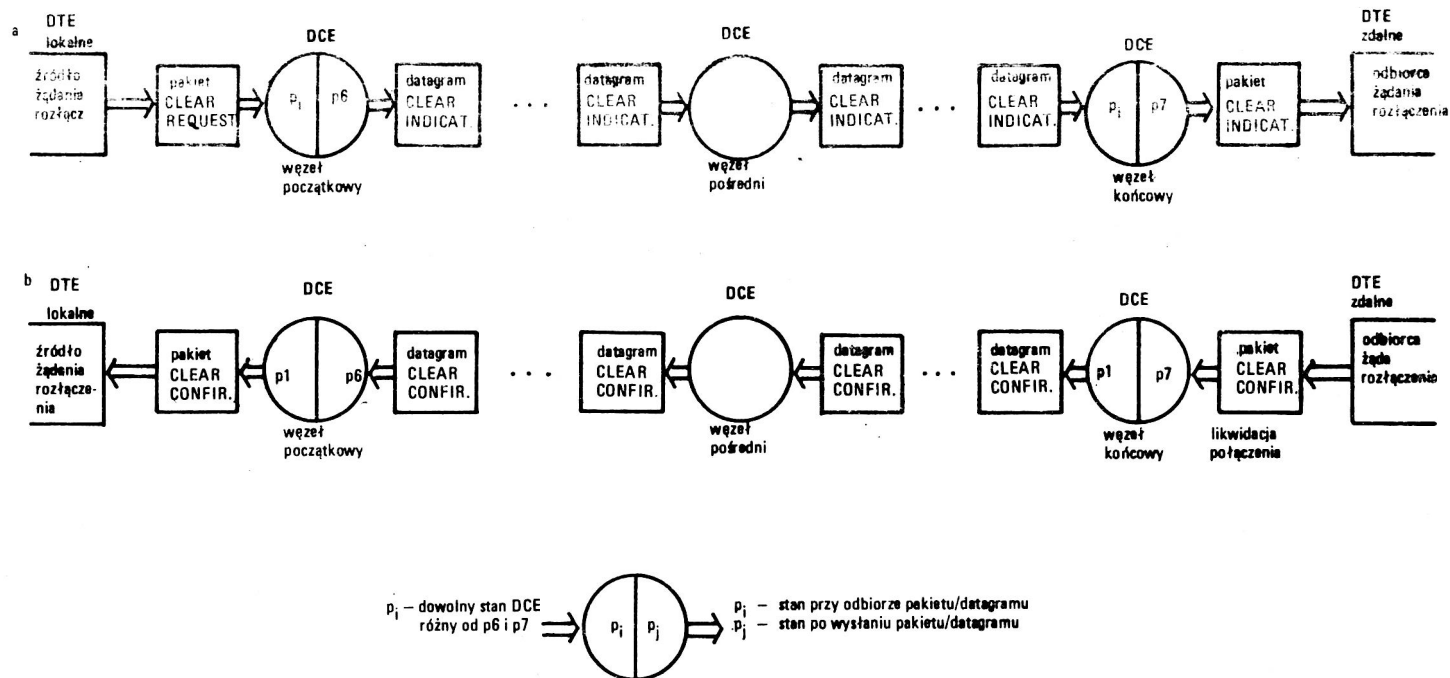
Jako potwierdzenie rozłączenia jest traktowane również odebranie w węźle docelowym pakietu CLEAR REQUEST (zamiast CLEAR CONFIRMATION). Zdarzenie takie jest określane mianem kolizji wywołań rozłączenia. Uzyskany pakiet CLEAR REQUEST zostaje zamieniony na datagram CLEAR INDICATION i jest wysyłany do węzła źródłowego, a opis połączenia jest likwidowany. W każdym węźle pośrednim, przez który jest transmitowany zwrótnie datagram CLEAR CONFIRMATION lub CLEAR INDICATION, dokonuje się likwidacji opisu połączenia.

W węźle źródłowym otrzymany datagram CLEAR CONFIRMATION lub CLEAR INDICATION zostaje zamieniony na pakiet i przesłany do użytkownika - DTE, po czym likwiduje się opis połączenia. Faza rozłączenia została zakończona - zlikwidowano opis połączenia w całej podsieci. Jeśli DTE - inicjator rozłączenia nie uzyska potwierdzenia realizacji swojego żądania, to po upływie limitu czasowego retransmituje pakiet CLEAR REQUEST, który jest przesłany w przedstawiony sposób.

Liczba retransmisji jest również ograniczona i po jej przekroczeniu DTE podejmie ewentualnie decyzję o restarcie łącza, którym jest związane z podsiecią i na którym istnieje zlikwidowane połączenie (5). Opisany przebieg fazy rozłączenia przedstawiono schematycznie na rys. 1.

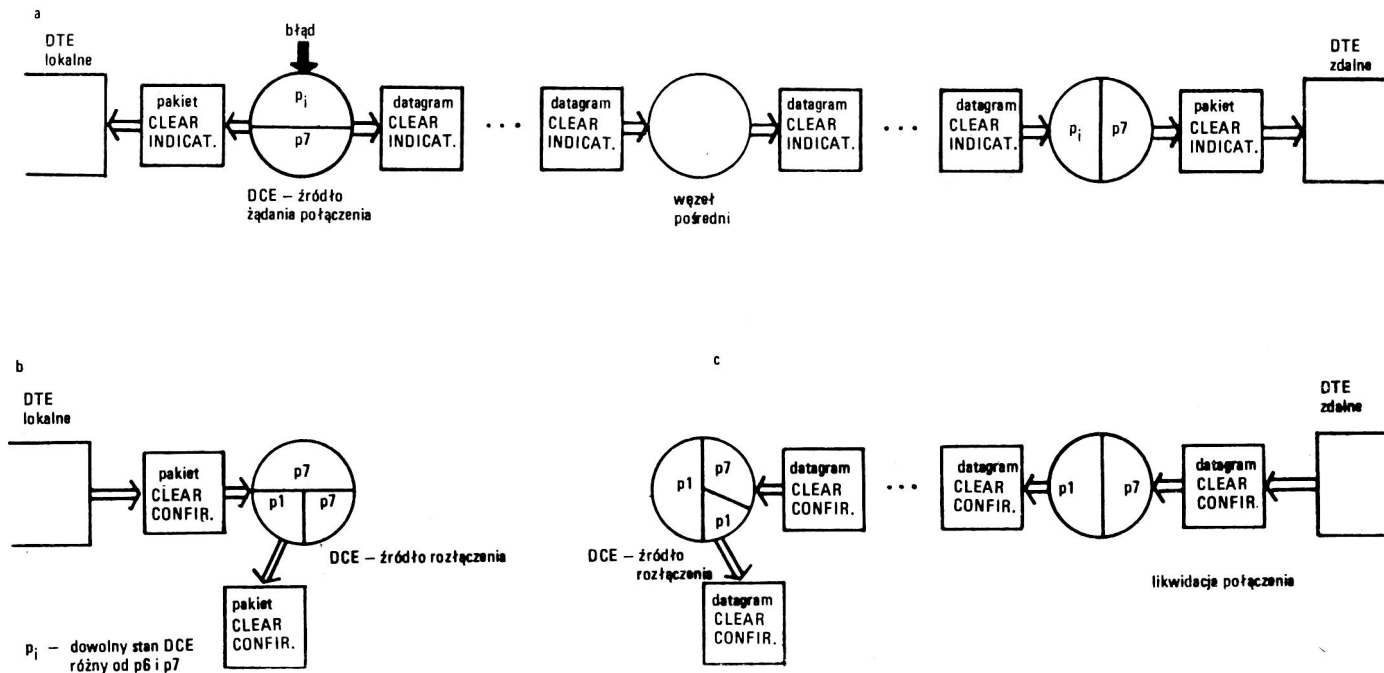
Odmienne jest realizowane rozłączenie dokonywane na żądanie podsieci, a konkretnie dowolnego z węzłów podsieci związanych z DTE. Żądanie rozłączenia jest wysyłane jednocześnie w postaci pakietu i datagramu CLEAR INDICATION odpowiednio do obu użytkowników połączenia (do lokalnego - pakiet, do zdalnego - datagram).

Węzeł - inicjator rozłączenia oczekuje na potwierdzenie od obu użytkowników, a nieotrzymanie dowolnego z nich w określonym czasie powoduje retransmisję odpowiednio pakietu lub datagramu (lub obu). Uzyskanie potwierdzeń jest jednoznaczne z zakończeniem fazy rozłączenia i połączenie w podsieci jest zlikwidowane. Po wyczerpaniu limitu retransmisji ulegają likwidacji opisy połączenia w węzłach związanych z obydwoma użytkownikami - DTE. Schematycznie przebieg tego rozłączenia przedstawiono na rys. 2.



Rys.1. Przebieg poprawnej transmisji rozłączenia na żądanie użytkownika: a - transmisja żądania rozłączenia, b - transmisja potwierdzenia rozłączenia

Fig.1. Transmission process in disconnection phase - on user request: a - transmission of disconnection request, b - transmission of disconnection acknowledge



Rys.2. Przebieg poprawnej transmisji rozłączenia na żądanie podsięci; a - transmisja żądania rozłączenia, b - transmisja potwierdzenia rozłączenia przez DTE lokalne, c - transmisja potwierdzenia rozłączenia przez DTE zdalne

Fig.2. Transmission process in disconnection phase - on subnetwork request; a - transmission of disconnection request, b - transmission of disconnection acknowledge by local DTE, c - transmission of disconnection acknowledge by remote DTE

Istnieje również możliwość dokonywania "awaryjnego" rozłączenia połączenia w razie uszkodzenia linii transmisji lub żądania przez dowolnego użytkownika - DTE restartu całego łącza, którym jest związany z podsiecią. W takiej sytuacji po wszystkich połączeniach realizowanych na uszkodzonej lub restartowanej linii przesyła się w przeciwnym kierunku pakiet lub datagram CLEAR INDICATION z informacją o występującym zdarzeniu. Nie oczekuje się potwierdzenia takiej informacji, a połączenie likwiduje się natychmiast po wysłaniu pakietu czy datagramu [5].

3. STRUKTURA PAKIETÓW, GRAFY I TABELLE PRZEJŚĆ

Do realizacji fazy rozłączenia wykorzystuje się 3 typy pakietów:

- CLEAR REQUEST,
- CLEAR INDICATION,
- CLEAR CONFIRMATION,

których strukturę przedstawiono na rys. 3.

a)

Oktet	8	7	6	5	4	3	2	1	
1	Format Ogólnny Pakietu (GFI)				Nr Grupy Kanału Logicznego (LCGN)				Identyfikator Typu Pakietu (PTI)
2	Nr Kanału Logicznego (LCN)								
3	0	0	0	1	0	0	1	1	
4	Kod Przyczyny (CC)								
5	Kod Diagnostyczny (DC)								

b)

Oktet	8	7	6	5	4	3	2	1	
1	Format Ogólny Pakietu (GFI)				Nr Grupy Kanału Logicznego (LCGN)				
2	Nr Kanału Logicznego (LCN)								
3	0	0	0	1	0	1	1	1	Identyfikator Typu Pakietu (PTI)

Rys.3. Struktura pakietów fazy rozłączenia: a - format pakietu CLEAR REQUEST lub CLEAR INDICATION, b - format pakietu CLEAR CONFIRMATION

Fig.3. Structure of packets for the disconnection phase: a - CLEAR REQUEST/CLEAR INDICATION packet format, b - CLEAR CONFIRMATION packet format

Kodowanie pola przyczyny rozłączenia podano w tab. 1, natomiast wykorzystywane kody diagnostyczne w tab. 2.

T a b e l a 1

Kody przyczyn rozłączenia

Bit y	8 7 6 5 4 3 2 1	Kod ósemkowy
Rozłączenie przez DTE	0 0 0 0 0 0 0 0	0
Zajęty numer	0 0 0 0 0 0 0 1	1
Awaria DCE	0 0 0 0 1 0 0 1	11
Błąd procedury zdalnej	0 0 0 1 0 0 0 1	21
Odbiornik nieodpowiednio wybrany	0 0 1 0 0 0 0 1	41
Zadanie niewłaściwego udogodnienia	0 0 0 0 0 0 1 1	3
Dostęp zastrzeżony	0 0 0 0 1 0 1 1	13
Błąd procedury lokalnej	0 0 0 1 0 0 1 1	23
Przeciążenie sieci	0 0 0 0 0 1 0 1	5
Niedostępny DCE	0 0 0 0 1 1 0 1	15

T a b e l a 2

Kody diagnostyczne fazy rozłączenia

Bit y	8 7 6 5 4 3 2 1	Kod ósemkowy
<u>Nieważny typ pakietu</u>	0 0 0 1 0 0 0 0	20
dla stanu p6	0 0 0 1 1 0 0 1	31
dla stanu p7	0 0 0 1 1 0 1 0	32
<u>Niedozwolony pakiet</u>	0 0 1 0 0 0 0 0	40
niezdefiniowany pakiet	0 0 1 0 0 0 0 1	41
pakiet zbyt krótki	0 0 1 0 1 1 0 0	46
pakiet zbyt długi	0 0 1 0 1 1 1 1	47
nieważny identyfikator formatu ogólnego	0 0 1 0 1 0 0 0	50
<u>Przekroczony czas</u>	0 0 1 1 0 0 0 0	60
dla CLEAR INDICATION	0 0 1 1 0 0 0 1	61

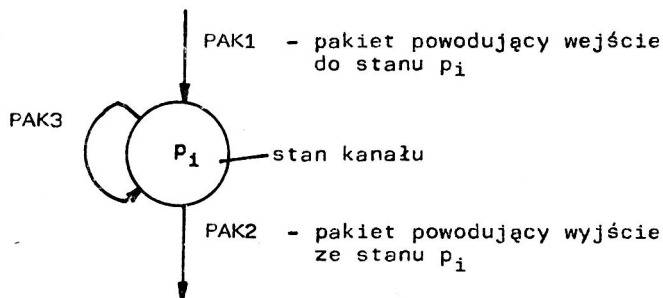
Uwaga. Pole kodu diagnostycznego nie jest wymagane w pakiecie CLEAR REQUEST; jeśli żaden z podanych kodów nie jest odpowiedni dla wykrytego w DCE błędu, to pole kodu diagnostycznego może zawierać 0.

Jak wynika z grafu (rys. 5) w fazie rozłączenia wyróżniono dwa stany kanału: p6 i p7. Pierwszy z nich oznacza odebranie przez DCE pakietu CLEAR REQUEST, a drugi odebranie przez DCE datagramu CLEAR INDICATION i wysłanie przez DCE pakietu i datagramu CLEAR INDICATION odpowiednio do DTE i do podsieci.

Tabela 3 stanowi uzupełnienie grafu i przedstawia zmianę stanów w DCE pod wpływem otrzymywanych z DTE pakietów. Działanie oznaczone "błąd" należy rozumieć jako wysłanie do DTE i do podsieci odpowiednio pakietu i datagramu CLEAR INDICATION.

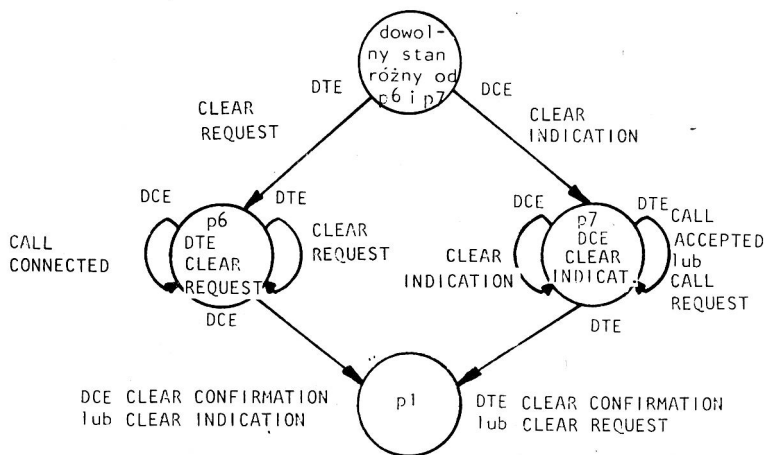
Podane liczby #N są liczbami ósemkowymi i oznaczają kody diagnostyczne wysyłane w pakiecie/datagramie CLEAR INDICATION po wykrytym błędzie.

Poprawny przebieg fazy rozłączenia przedstawia graf z rys. 5. Stosowane w grafie symbole mają interpretację jak na rys. 4.



Rys.4. Znakowanie grafu przejść

Fig.4. Marking of transition graph



Rys.5. Graf przejść dla fazy rozłączenia

Fig.5. Transition graph for disconnection phase

Tabela przejść dla fazy rozłączenia

Otrzyma- ny pakiet od DTE	Stan ka- nału DCE	Dowolny stan z wyjątkiem p6 i p7	p6	p7
CALL REQUEST		błąd dla stanów p2 → p7, # 21 p4 → p7, # 23	błąd p6 → p7 # 25	odrzućcie p7 → p7
CALL ACCEPTED		błąd dla stanów p1 → p7 # 20 p2 → p7 # 21 p4 → p7 # 23 p5 → p7 # 24	błąd p6 → p7 # 25	odrzućcie p7 → p7
CLEAR REQUEST		p _i → p6	odrzućcie p6 → p6	p7 → p1
DTE CLEAR CONFIRMATION		błąd p _i → p7 # 21 # 22 # 23 # 24 # 25	błąd p6 → p7 # 25	p7 → p1
Dane, RESET INTERRUPT		błąd dla stanów p1 → p7 # 20 p2 → p7 # 21 p3 → p7 # 22 p5 → p7 # 24	błąd p6 → p7 # 25	odrzućcie p7 → p7
RESTART REQUEST lub DTE RESTART CONFIRMATION z bitami 1÷4 oktetu 1 lub bitami 1÷8 oktetu 2÷0		jak wyżej	błąd p6 → p7 # 25	odrzućcie p7 → p7
Pakiet krótszy od 1 oktetu lub nieznany DCE		jak wyżej	błąd p6 → p7 # 25	odrzućcie p7 → p7

4. AKCJE WĘZŁA - DCE W FAZIE ROZŁĄCZENIA

Aby poprawnie przeprowadzić rozłączenie, węzeł - DCE musi mieć odpowiednie struktury danych i sterowania, które zapewnią opisane w punkcie 3 zachowanie DCE w różnych sytuacjach.

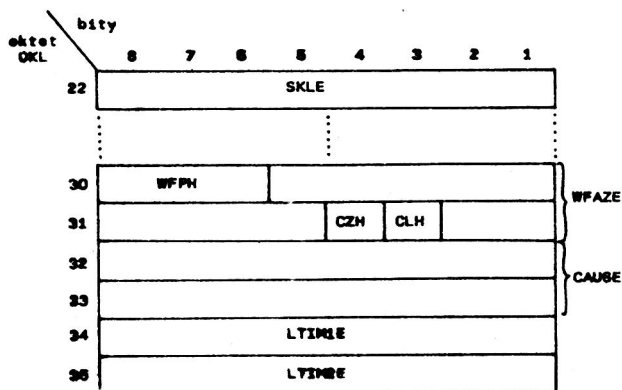
4.1. Struktury danych

Przyjęto, że w celu opisu połączenia (drogi logicznej) w każdym węźle wchodzącym w jego skład datagram INCOMING CALL tworzy strukturę danych, zwaną opisem kanału logicznego - OKL.

W fazie rozłączenia tylko niektóre pola OKL są wykorzystywane. I tak, do zapisu i odczytu wykorzystuje się:

- 2-bajtowe pole przyczyny rozłączenia CAUSE,
- 2-bajtowe pole wskaźników podfazy WFAZE,
- 2-bajtowe pole liczników retransmisji LTIM1E i LTIM2E,
- 1-bajtowe pole stanu kanału SKLE.

Strukturę tych pól przedstawiono na rys. 6.



Rys.6. Pola OKL wykorzystywane w fazie rozłączenia

Fig.6. Fields of Logical Channel Description structure using in disconnection phase

Dodatkowo korzysta się z informacji przechowywanych na innych polach OKL (tylko odczyt), np. linia odbiorcy i nadawcy itd. Strukturę pól oraz sposób konstrukcji ich zapisu można znaleźć w pracy [31].

Pola SKLE oraz WFPH są wykorzystywane podobnie jak w innych fazach, tzn. przechowują one aktualny stan kanału (10₈ dla stanu po i 11₈ dla

stanu p7) i wskazują na aktualnie pracującą fazę (4₈ - faza rozłączenia). Natomiast wyłącznie przez fazę rozłączenia są wykorzystywane pozostałe pola z rys. 6 i służą one do przechowywania specjalnych informacji o prowadzonym rozłączeniu.

Pole CAUSE zawiera na jednym (młodszym) bajcie informację o przyczynie rozłączenia (odpowiednik pola CC pakietu - por. pkt 3), natomiast na drugim (starszym) bajcie kod diagnostyczny (odpowiednik pola DC pakietu).

Pole liczników retransmisji jest podzielone na dwa liczniki retransmisji:

- do zliczenia wysyłanych pakietów LTIM15,
- do zliczenia wysyłanych datagramów LTIM21.

Bity pola WFAZE, oznaczone CZH i CLH, są wskaźnikami prowadzenia retransmisji odpowiednio datagramów i pakietów.

Jeśli przyjmują one wartość 1, to oznaczają, że jest prowadzona retransmisja. Pola te są zerowane albo po pomyślnym zakończeniu retransmisji (uzyskanie potwierdzenia), albo po wyczerpaniu limitu retransmisji.

Struktura danych wykorzystywana w konstrukcji pakietu i datagramu ma postać struktury podanej dla bufora w pracy [3].

4.2. Działanie DCE w fazie rozłączenia

Akcje podejmowane przez DCE w fazie rozłączenia są zależne od:

- a) stanu kanału,
- b) otrzymanego pakietu/datagramu,
- c) dostępności zasobów węzła.

Istnieje jednak kilka akcji standardowych, które są podejmowane niezależnie od podanych warunków. Do akcji tych należą:

- a) rejestracja zdarzenia,
- b) badanie formalnej i semantycznej poprawności pakietu/datagramu,
- c) badanie kontekstu uzyskanego pakietu/datagramu,
- d) w razie pozytywnego wyniku kontroli b) i c) zamiana pakietu na datagram (lub odwrotnie), wysyłanie przygotowanego pakietu/datagramu (wstawienie do kolejki wyprowadzania w linię).

Działania pozostałe są działaniami kontekstowymi i należą do nich:

4.2.1. Odbiór pakietu

1. Dla stanu p6:

- badanie czasu uzyskania CLEAR REQUEST; jeśli po upływie limitu czasowego, to przesyła się datagram CLEAR INDICATION; przeciwnie - zwrot bufora,

- ignorowanie uzyskanych pakietów CALL REQUEST lub CALL CONNECTED i zwrot bufora,
- rejestracja błędu w razie uzyskania innych pakietów; zmiana stanu na p7; generacja pakietu i datagramu CLEAR INDICATION i ich wysłanie; inicjowanie odliczania obu limitów czasowych i oczekiwanie na potwierdzenie.

Jeśli niemożliwe jest uzyskanie bufora do konstrukcji pakietu lub węzeł jest przeciążony (nie można inicjować odliczania limitu czasowego), to likwiduje się opis kanału logicznego - połączenie jest zlikwidowane.

2. Dla stanu p7:

- badanie czy odebrany z DTE pakiet jest potwierdzeniem dla źródłowego DCE (tzn. czy $CLH = \emptyset$), jeśli tak, to pakiet jest wysyłany w postaci datagramu; zostaje skasowany opis kanału logicznego, połączenie jest zlikwidowane; przeciwnie - odebrany pakiet należy traktować jako potwierdzenie dla DCE na wysłane uprzednio żądanie rozłączenia, zeruje się wskaźnik CLH i bada, czy zerowy jest również wskaźnik CZH; jeśli jest zerowy, to oznacza, że obaj użytkownicy połączenia potwierdzili rozłączenie i likwiduje się opis kanału logicznego, jeśli natomiast CZH = 1, to nie zmieniając stanu kanału oczekuje się na potwierdzenie z podsięci.

3. Dla dowolnego stanu kanału różnego od p6 lub p7:

- otrzymany pakiet CLEAR REQUEST jest wysyłany jako datagram CLEAR INDICATION, stan kanału ustala się na p6,
- pakiety inne niż CLEAR REQUEST nie mogą być przekazane do fazy rozłączenia, bo nie są w kontekście.

4.2.2. Odbiór datagramu

1. Dla stanu p6:

- otrzymane datagramy CLEAR INDICATION lub CLEAR CONFIRMATION są wysyłane jako pakiety do użytkownika - DTE, likwiduje się opis kanału logicznego, połączenie przestaje istnieć,
- ignoruje się datagramy INCOMING CALL i CALL CONNECTED, zwrot bufora,
- otrzymanie innych datagramów jest traktowane jako błąd (nie są w kontekście), zmienia się stan kanału na p7; wysyła się pakiet i datagram CLEAR INDICATION z podaniem przyczyny błędu, inicjuje się odliczanie limitu czasowego i oczekuje na potwierdzenie; jeśli nie jest możliwa konstrukcja pakietu (w konstrukcji datagramu wykorzystuje się uzyskany bufor z datagramem), to likwiduje się opis kanału logicznego i połączenie przestaje istnieć.

2. Dla stanu p7:

- badanie, czy uzyskany datagram CLEAR INDICATION lub CLEAR CONFIRMATION należy traktować jako potwierdzenie rozłączenia węzła docelowego

(tzn. czy CZH=1); jeśli tak, to zwalnia się bufor i kontroluje się czy uzyskano również potwierdzenie od DTE związanego; likwiduje się opis kanału logicznego, jeśli uzyskano oba potwierdzenia; w razie gdy CZH=0, a więc nie jest to potwierdzenie rozłączenia, datagram CLEAR INDICATION jest retransmitowany i należy go po przekształceniu na pakiet wysyłać do związanego DTE;

- inne datagramy, jako nie uzyskane w kontekście, ignoruje się.

3. Dla dowolnego stanu różnego od p6 lub p7:

- uzyskany datagram CLEAR INDICATION jest traktowany jako żądanie rozłączenia przez drugiego użytkownika nie związanego z DCE; przesyła się pakiet CLEAR INDICATION do związanego DTE, stan kanału ustala się na p7,

- uzyskanie innych datagramów jest niemożliwe, bo nie są one poprawne dla fazy rozłączenia.

4.2.3. Obsługa błędów

W razie stwierdzenia w fazie innej niż faza rozłączenia błędu kwalifikującego do rozłączenia połączenia (np. przekroczony limit zerowań połączenia) informacja taka jest przekazywana do fazy rozłączenia, stan kanału ustala się na p7, generuje się pakiet i datagram CLEAR INDICATION podając przyczynę błędu, inicjuje się odliczanie limitu czasowego i oczekuje na potwierdzenie. Jeśli węzeł jest przeciążony (nie ma buforów do konstrukcji pakietu lub nie można mierzyć czasu), to likwiduje się opis kanału logicznego, połączenie przestaje istnieć.

5. KONTROLA POPRAWNOŚCI FAZY ROZŁĄCZENIA

Na kontrolę poprawności przebiegu fazy rozłączenia składają się:

1. Kontrola poprawności pakietu/datagramu, formalna i semantyczna.
2. Kontrola poprawności kontekstu odebranego pakietu/datagramu.
3. Kontrola upływu limitu czasowego.
4. Kontrola limitu retransmisji.

Dwie pierwsze kontrole są prowadzone dla każdego pakietu/datagramu w każdym stanie kanału logicznego.

W razie stwierdzenia niepoprawności pakietu/datagramu lub nieuzyskania go w poprawnym kontekście, dokonuje się:

- zapisu informacji o błędzie w polu CAUSE OKL,
- zmiany stanu kanału na p7 (jeśli stan był inny - por. tab. 3) i dalsze akcje jak podano w punkcie 4,
- ignoruje się wykryty błąd (jedynie rejestruje się jego wystąpienie w celu statystyki), jeśli stan kanału jest p7.

Kontrola upływu limitu czasowego jest możliwa dzięki prowadzeniu przez system sterujący węzła odliczania czasu. Po stwierdzeniu wystąpienia limitu czasowego dokonuje się retransmisji pakietu/datagramu i zwiększa się odpowiedni licznik retransmisji. Stan tego licznika porównuje się z limitem retransmisji (parametr instalacyjny, podobnie jak limit czasowy). Jeśli jest przekroczony, to likwiduje się połączenie przez usunięcie opisu kanału logicznego.

W związku z przedstawioną kontrolą poprawności fazy rozłączenia, a także dla statystyki i diagnostyki pracy węzła (podsieci) prowadzi się rejestrację następujących zdarzeń, stwierdzonych w fazie rozłączenia:

- błędny datagram,
- błędny pakiet,
- odbiór pakietu CLEAR REQUEST,
- odbiór datagramu CLEAR INDICATION,
- brak wolnego bufora,
- rozłączenie z powodu limitu retransmisji,
- rozłączenie z powodu braku bufora,
- rozłączenie z powodu przeciążenia węzła,
- zakończenie fazy rozłączenia.

Informacje te są przekazywane podprogramom diagnostycznym węzła.

6. OPROGRAMOWANIE FAZY ROZŁĄCZENIA

Na oprogramowanie fazy rozłączenia składa się 6 podmodułów, wykorzystujących 9 podprogramów własnych modułu X.25 oraz 6 podprogramów udostępnionych przez oprogramowanie podstawowe węzła.

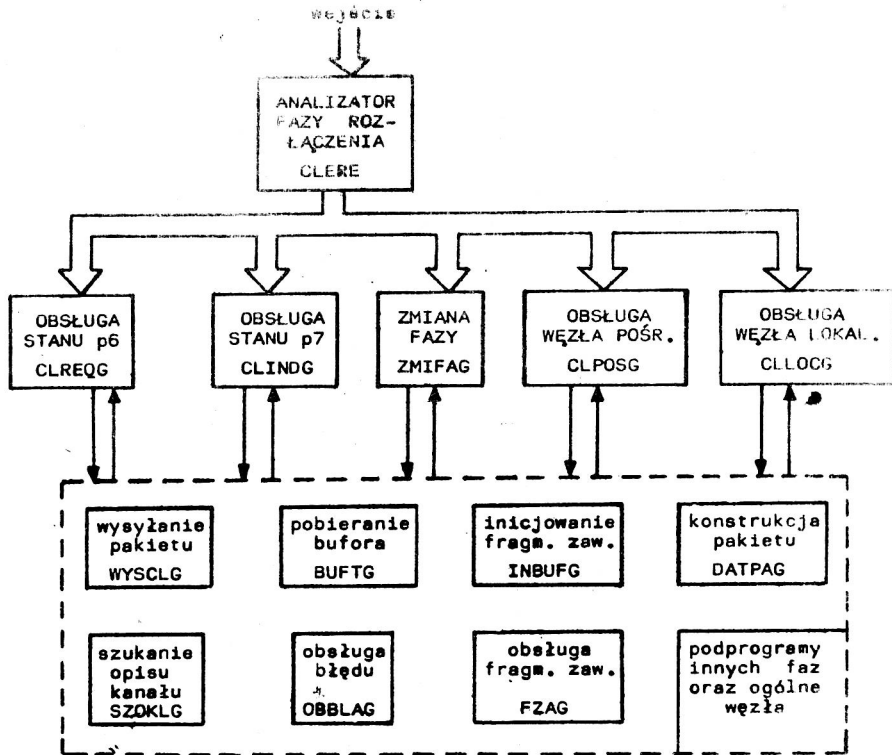
Komunikacja oprogramowania fazy rozłączenia z innymi fazami odbywa się za pośrednictwem jednego podmodułu - analizatora, który uzyskuje dane dla działania fazy i zgodnie z ustalonymi warunkami oraz danymi przekazuje sterowanie do jednego z 5 podmodułów, realizujących następujące funkcje:

- obsługę stanu p6,
- obsługę stanu p7,
- obsługę zmiany fazy,
- obsługę węzła pośredniego,
- obsługę węzła lokalnego.

Każdy z 6 podmodułów jest jednowyjściowy i z wyjątkiem podmodułu analizatora - jednowyjściowy.

Strukturę oprogramowania fazy rozłączenia przedstawiono na rys. 7.

Komunikacja między podmodułami, tzn. analizator-dowolny podmoduł, jest realizowana przez wykorzystanie wspólnego obszaru danych, na który składają się:



Rys.7. Struktura oprogramowania fazy rozłączenia

Fig.7. Programming structure of disconnection phase

- rejestry ogólnego przeznaczenia minikomputera SM-3,
- opis kanału logicznego,
- bufor z datagramem/pakiemem,
- dane instalacyjne węzła.

Ta sama zasada komunikacji została przyjęta dla współpracy podmoduły-podprogramy.

Oprogramowanie fazy rozłączenia w celu poprawnego działania wymaga udostępnienia przez system sterujący węzła następujących zasobów:

- buforów do konstrukcji datagramu/pakietu,
- odliczania czasu.

Wymaganie odliczania czasu jest związane z koniecznością pomiaru limitu czasowego i przeprowadzenia retransmisji pakietu/datagramu po

jego upływie. Po upływie limitu czasowego jest inicjowany (przez program sterujący węzła) tzw. fragment zawieszony, który realizuje akcje retransmisji pakietu/datagramu. Fragment zawieszony stanowi w zasadzie pewien fragment-podprogram właściwego oprogramowania fazy rozłączenia.

Oba zasoby są dostarczone przez specjalne podprogramy oprogramowania podstawowego węzła i są inicjowane w dowolnym z podmodułów fazy rozłączenia. Podprogramy te zwrotnie podają informację o możliwości realizacji żądania.

Wyłączność (synchronizacja) dostępu do wspólnych danych węzła jest zapewniona przez nieprzerwywalność wykonywania poszczególnych podmodułów/podprogramów.

Aby uzyskać pewną mobilność oprogramowania, jak również ze względu na niezdefiniowanie pewnych wielkości przez protokół X.25 w oprogramowaniu fazy rozłączenia, przyjęto jako parametry następujące dane:

- identyfikator ogólny formatu pakietu,
- limit retransmisji pakietów/datagramów CLEAR INDICATION,
- limit czasowy DCE i DTE związanego,
- czas oczekiwania na bufor.

Dane te są dostępne jako parametry instalacyjne węzła we wspólnym obszarze danych i mogą być zmieniane w kolejnych generacjach oprogramowania podstawowego węzła.

LITERATURA

- [1] Draft Revised Recommendation X.25 Introduction and Packet Level. CCITT Study Group VII, Geneva, November 1979.
- [2] Dubielewicz I., Koleśnik K., Reinicjacja transmisji danych w sieci komputerowej MSK (w tym samym zeszycie).
- [3] Fryźlewicz Z., Protokół pakietowy w sieci komputerowej MSK (w tym samym zeszycie).
- [4] Koleśnik K., Transmisja danych w sieci komputerowej MSK (w tym samym zeszycie).
- [5] Goetz J., Restart połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).
- [6] Fryźlewicz Z., Mikulska E., Zestawianie połączeń logicznych w sieci komputerowej MSK (w tym samym zeszycie).

Praca wpłynęła do Redakcji 15.12.1980 r.

Po poprawieniu 3.03.1982 r.

DISCONNECTION PHASE IN THE MSK COMPUTER NETWORK

This paper presents the phase of disconnection of virtual connection in the MSK computer network on the X.25 packets protocol level. The causes and the way of realization of disconnection are given.

The programming system realizing in the node disconnection phase is described and the advantages of implemented solution are discussed.

Verified by Ruta Czaplińska

РАЗЪЕДИНЕНИЕ СОЕДИНЕНИЯ В СЕТИ ЭВМ MSK

В статье описана фаза разъединения соединения в сети ЭВМ MSK на уровне пакетного протокола X.25. Представлены принципы и способ реализации разъединения. Описана программная система, управляющая разъединением в узле сети и показаны достоинства принятого решения.

Проверила Малгожата Хейдрих

*Sieć komputerowa, protokół
pakietowy, połączenie w trybie*

Józef GOETZ^{*)}

Restart połączeń logicznych w sieci komputerowej MSK

Przedstawiono zasady obsługi oraz funkcję restartu protokołu pakietowego X.25 poziomu 3. Opisano implementację restartu dla węzła sieci komputerowej MSK.

1. CHARAKTERYSTYKA FAZY RESTARTU

Celem restartu jest przywrócenie łącza (interfejsu) DTE/DCE poziomowi pakietowemu do ponownego startu, tj. założenia połączeń logicznych. W fazie restartu następuje rozłączenie wszystkich czasowych i stałych połączeń logicznych, związanych z określonym lokalnym DTE (tj. końcowym sieciowym urządzeniem transmisji danych).

Faza restartu może być wywołana przez lokalne DTE bądź przez DCE (tj. sieciowe urządzenie transmisji danych lub inaczej węzeł DCE).

Faza restartu, wywołana od lokalnego DTE, polega na przesłaniu do DCE odpowiedniego pakietu (RESTART REQUEST), na co DCE powinien odpowiedzieć potwierdzeniem otrzymania tego pakietu. Przesyłanie od lokalnego DTE do DCE innych pakietów, zanim otrzyma się potwierdzenie na wcześniej wysłany pakiet, prowadzi do wykrycia błędu restartu. Wykrycie niepoprawności w przesłaniu pakietów prowadzi do tego, że DCE przejmuje inicjatywę w wysyłaniu odpowiedniego pakietu (RESTART INDICATION) do DTE i oczekuje na jego potwierdzenie określony czas.

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

Faza restartu jest wywołana przez DCE wtedy, gdy na poziomie fizycznym lub liniowym nastąpi usunięcie uszkodzenia na łączu DTE/DCE. W tym przypadku DCE wysyła do lokalnego DTE i do zdalnych DTE połączonych z DCE połączeniami stałymi odpowiednie pakiety z informacją o tym zdarzeniu. Informacje (pakiety) są ustaloną liczbę razy retransmitowane tylko do lokalnego DTE, gdy źródłem (inicjatorem) pakietów jest DCE i brak jest potwierdzenia od DTE.

Możliwość retransmisji pakietów zrealizowano w celu zapewnienia stabilności transmisji oraz dążności do przeprowadzenia zapoczątkowanych akcji do końca. Przyjęto zasadę, że retransmisji pakietów dokonuje źródło generacji tych pakietów. Czas, po którym dokonuje się retransmisji jest zwany limitem czasowym. Zarówno limit retransmisji, jak i limit czasowy zależą od rodzaju urządzenia w sieci i rodzaju generowanego pakietu/datagramu. Przyjęto w pracy, że ogólnie różne wartości limitów są przyporządkowane każdemu łączu DTE/DCE.

Niezależnie od źródła wywołania fazy restartu, z DCE do sieci wysyła się odpowiednie informacje w datagramach (tj. pakietach uzupełnionych o adres nadawcy) CLEAR INDICATION i RESET INDICATION odpowiednio w połączeniach czasowych (VC) i stałych (PVC), związanych z danym łączem DTE/DCE. DCE wysyłające datagramy nie czeka na ich potwierdzenie.

2. ZASADY OBSŁUGI W FAZIE RESTARTU

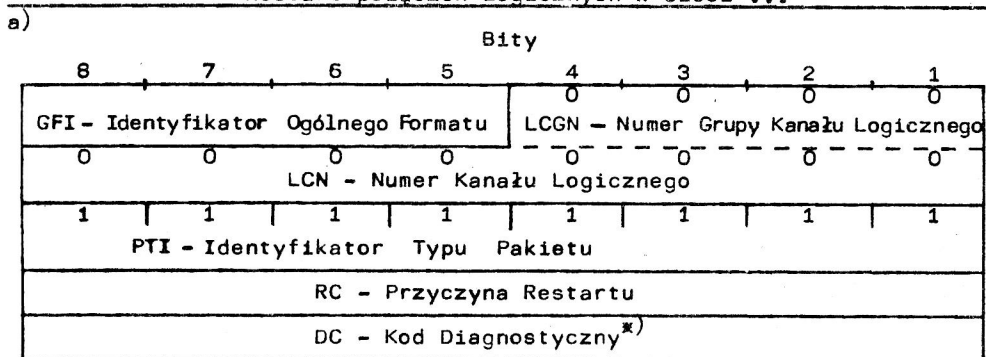
2.1. Struktura pakietów, graf stanów i tabela przejść

Wykaz pakietów akceptowanych w fazie restartu jest następujący: RESTART REQUEST, RESTART INDICATION i pakiety potwierdzające DCE RESTART CONFIRMATION oraz DTE RESTART CONFIRMATION. Pozostałe pakiety poziomu X.25 są ignorowane, bądź są przyczyną wywołania obsługi błędu restartu.

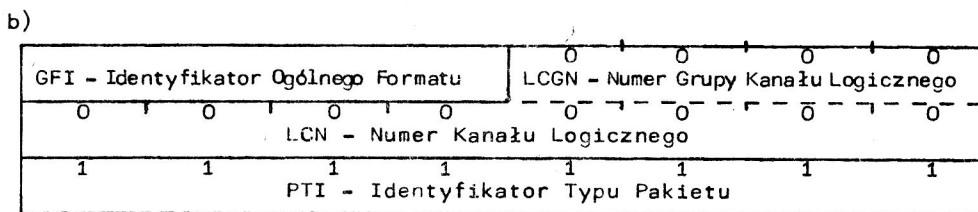
Na rysunku 1 przedstawiono strukturę pakietu fazy restartu. Pakiety mają ustawione na 0 bity 4, 3, 2 i 1 pierwszego oktetu oraz wszystkie bity drugiego oktetu. Oznacza to, że pakiety te będą przesyłane w zerowym kanale logicznym. Kanał ten, nazywany systemowym, jest związany ze stałym połączeniem, tzn. jest typu PVC. Każde łącze DTE/DCE ma swój systemowy kanał logiczny. Systemowe kanały logiczne istnieją tylko na łączach lokalnych DTE/DCE.

Oktet 4 pakietu RESTART INDICATION zawiera kod przyczyny restartu. Kodowanie tego pola jest następujące:

- 1 - błąd lokalny,
- 3 - przeciążenie sieci,
- 7 - sieć operatywna.



^{*)} - pole nie jest wymagane w pakiecie RESTART REQUEST



Rys.1. Struktura pakietów fazy restartu: a - format pakietu RESTART REQUEST i RESTART INDICATION, b - format pakietu DTE i DCE RESTART CONFIRMATION

Fig.1. Structure of packets for the restart phase: a - RESTART REQUEST and RESTART INDICATION packet format, b - DTE and DCE RESTART CONFIRMATION packet format

Przyjęto, że bity pola Przyczyna Restartu w pakiecie RESTART REQUEST będą ustawione na zero.

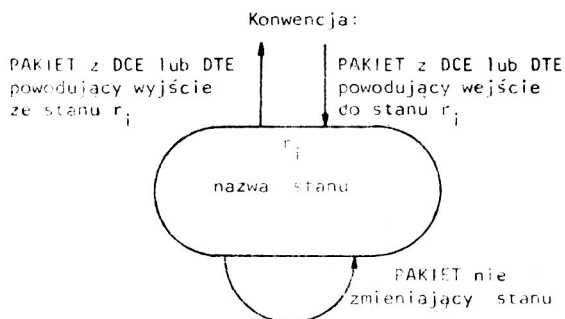
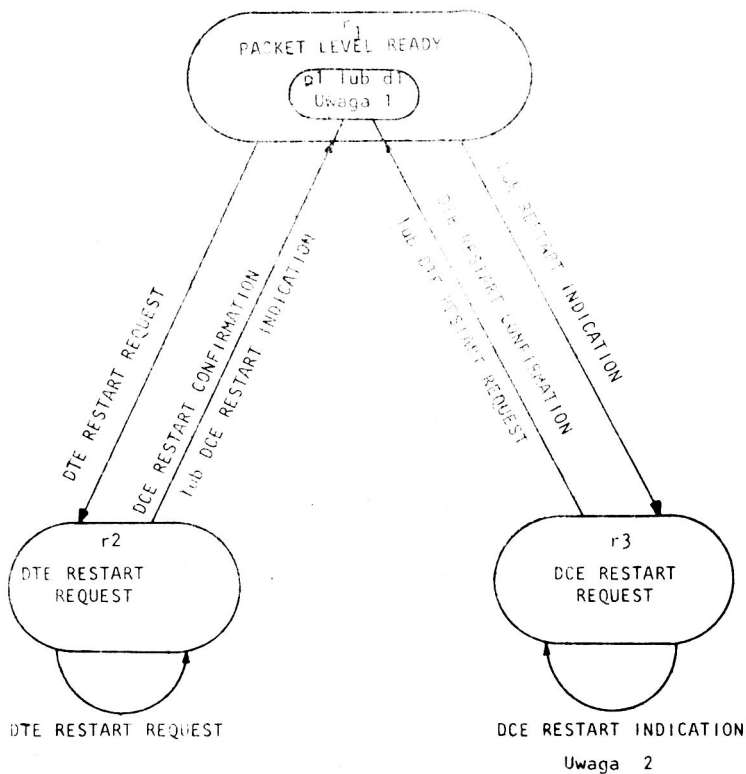
Oktet 5 zawiera dodatkowo informacje o przyczynie restartu. Dla pakietu RESTART INDICATION kod diagnostyczny jest ustawiany na zero, kiedy nie są wyspecyfikowane dodatkowe informacje o restarcie.

Przejęcie do fazy restartu może nastąpić przez wysłanie pakietu RESTART REQUEST, bądź DTE RESTART CONFIRMATION, bądź po usunięciu uszkodzenia na określonym łączu DTE/DCE.

Na rysunku 2 przedstawiono graf stanów do poprawnego przebiegu fazy restartu. Znaczenie poszczególnych uwag na rys. 2 jest następujące: uwaga 1 - stan p1 lub d1 odpowiednio dla kanałów logicznych związanych z połączeniami czasowymi lub stałymi, uwaga 2 - transmisja może nastąpić po upływie limitu czasowego.

Akcja prawidłowego rozłączenia połączeń logicznych jest poprzedzona dwoma zdarzeniami:

- przesłaniem pakietu RESTART REQUEST lub DCE RESTART INDICATION odpowiednio z żądaniem rozłączenia lub z przyczyną rozłączenia i wysłaniem odpowiednich informacji w sieć w datagramach CLEAR INDICATION i RESET INDICATION,



Rys.2. Graf stanów dla fazy restartu
Fig.2. Graph of states for the restart phase

- przesłaniem odpowiadającego pakietu potwierdzenia: DCE RESTART CONFIRMATION lub DTE RESTART CONFIRMATION/RESTART REQUEST.

Przyjęto rozwiązanie, że w stanie r_2 pojawienie się pakietu innego niż RESTART REQUEST powoduje wysłanie do DTE pakietu RESTART INDICATION, które nie jest potwierdzeniem wcześniej wysłanego pakietu do DTE (obsługa błędnego restartu).

Tabela 1 wyszczególnia akcje podejmowane przez DCE w fazie restartu w odpowiedzi na pakiety.

T a b e l a 1

Akcje podejmowane w fazie restartu przez DCE w odpowiedzi na pakiety

Pa- kiet z DTE z przypo- rządkowanym ka- nałem logicznym	Stan kanału logicznego w DCE	PACKET LEVEL READY (dowolny stan od p1 do p7) r1	DTE RESTART REQUEST r2	DCE RESTART INDICATION r3
RESTART REQUEST		Normalne (r2)	Pominięcie (r2)	Normalne (p1 lub d1) Uwaga 3
DTE RESTART CONFIRMATION		Obsługa błędu (r3) # 17	Obsługa błędu (r3) # 18	Normalne (p1 lub d1)
Dopuszczalny pakiet ak- ceptowany w innych fa- zach (oprócz fazy res- tartu). Uwaga 4		Tabela w [2,5,6]	Obsługa błędu (r3) # 18	Pominięcie (r3)
RESTART REQUEST lub DTE RESTART CONFIRMATION z bitami # 0 dla oktetu 1 (bity 1-4) i dla ok- tetu 2 (bity 1-8)		Tabela w [2,5,6]	Obsługa błędu (r3) # 41	Pominięcie (r3)
Pakiety mające identyfi- kator typu pakietu kró- tszy niż 1 oktet lub pa- kiety nie akceptowane przez DCE		Tabela w [2,5,6]	Obsługa błędu (r3) # 38 # 33	Pominięcie (r3)
RESTART REQUEST lub DTE RESTART CONFIRMATION przekracza dozwoloną długość		Tabela 2 lp. 2 i 5	Obsługa błędu (r3) # 18 Uwaga 5	Obsługa błędu (r3) # 39

Uwaga 1. Stany w nawiasach są nowymi stanami, w które wchodzi przy-
porządkowany kanał logiczny.

Uwaga 2. #N - oznacza kod diagnostyczny wysyłany w pakiecie RES-
TART INDICATION. Tylko jeden kod diagnostyczny może być
wysłany.

Uwaga 3. Stan p1 lub d1 odpowiednio dla kanałów logicznych związa-
nych z czasowymi lub stałymi połączeniami.

Uwaga 4. Reakcja DCE w stanach r2 i r3 jest niezależna od formatu
błędu, który może być zawarty w otrzymanym pakiecie.

Uwaga 5. Dla RESTART CONFIRMATION; RESTART REQUEST pominięty (r2).

W fazie restartu kanał logiczny, do którego zostaje skierowany pa-
kiet, może znajdować się w stanach r2 lub r3. Stan r2, tj. DTE RESTART
REQUEST jest stanem, do którego przechodzą kanały logiczne związane z

łączem DTE/DCE, po którym odebrano pakiet RESTART REQUEST w zerowym kanale logicznym. Stan r3, tj. DCE RESTART INDICATION jest natomiast stanem, do którego przechodzą kanały logiczne związane z naprawionym łączem DTE/DCE lub związane z łączem DTE/DCE, po którym odebrano niewłaściwy pakiet - jest to obsługa błędnego restartu. Obsługa błędnego restartu następuje w wyniku odebrania w stanie r2 albo r3 pakietu o niewłaściwym formacie (błędy syntaktyczne) albo niewłaściwej treści (błędy semantyczne), lub w niewłaściwym kontekście, a także w wyniku odebrania w stanie r1 pakietu DTE RESTART CONFIRMATION w zerowym kanale logicznym.

Wyjście ze stanów r2 i r3 dla wszystkich kanałów logicznych, związanych z określonym łączem DTE/DCE, następuje po otrzymaniu odpowiednio potwierdzenia od DCE i DTE w zerowym kanale logicznym, związanym z wyżej określonym łączem, przy czym otrzymanie w stanie r3 pakietu RESTART REQUEST w zerowym kanale logicznym jest interpretowane jako potwierdzenie.

Maksymalny czas trwania stanu r2 wyraża się przez $t(r2) = (n_2 + 1)t_2$, gdzie: t_2 - maksymalny czas, w którym oczekuje się potwierdzenia od DCE na pakiet RESTART REQUEST, n_2 - liczba dopuszczalnych retransmisji pakietu RESTART REQUEST z lokalnego DTE.

Czas oczekiwania na potwierdzenie (t_3) na wysłany od DCE pakiet RESTART INDICATION jest ograniczony. Po wyczerpaniu tego czasu następuje ponowna retransmisja informacji o retransmisji. W stanie r3 jest ograniczona liczba dopuszczalnych retransmisji n_3 pakietu RESTART INDICATION do danego lokalnego DTE. Do każdego łącza DTE/DCE przewiduje się inny zbiór wartości n_2 , t_2 , n_3 i t_3 .

Przekroczenie wartości $t(r2)$ lub n_3 powoduje, że DCE ustawia wszystkie kanały logiczne typu VC i PVC związane z danym łączem DTE/DCE odpowiednio w stan p1 (READY) i stan d1 (FLOW CONTROL READY), tzn. rozłącza czasowe połączenia, a stałe połączenia przygotowuje do transmisji (kanały logiczne stałych połączeń mogą być tylko w fazie transmisji danych lub reinicjacji, tj. w stanach d1, d2 lub d3).

3. DZIAŁANIE I AKCJE PODEJMOWANE PRZEZ DCE W FAZIE RESTARTU

Akcje podejmowane przez DCE w fazie restartu są zależne od:

- stanu kanału logicznego,
- źródła i typu pakietu/datagramu,
- poprawności syntaktyczno-semantycznej pakietu,
- dostępności zasobów DCE (wolnych buforów, stosu do zainicjowania fragmentu zawieszonego).

Pewne akcje są wykonywane standardowo, np. wszystkie datagramy oraz niektóre pakiety w fazie restartu są pomijane albo w razie odebrania

Restart połączeń logicznych w sieci ...

niewłaściwego pakietu następuje przejście do standardowej obsługi błędnego restartu.

3.1. Restart przez DTE

DTE może w dowolnym czasie żądać restartu przez transmisję przez łącze DTE/DCE pakietu RESTART REQUEST w zerowym kanale logicznym. Kod tego pakietu zostanie zidentyfikowany w fazie transmisji danych [6], po czym sterowanie zostanie oddane do fazy restartu. W tej fazie, w przypadku poprawności pozostałych pól pakietu, zostaną ustawione w stan r2 (DTE RESTART REQUEST) wszystkie kanały logiczne związane z łączem od DTE, po którym przyszedł pakiet RESTART REQUEST.

Dalsze akcje, to skasowanie kolejek buforów zablokowanych (pakietów i datagramów) kanałów logicznych związanych z danym łączem oraz wysłanie datagramów CLEAR i RESET z przekopiowanym kodem przyczyny restartu i kodem diagnostycznym z otrzymanego pakietu RESTART REQUEST.

Do wszystkich połączeń czasowych, związanych z danym łączem DTE/DCE, zostaną wysłane datagramy CLEAR INDICATION, a do wszystkich połączeń stałych datagramy RESET INDICATION. W razie braku buforów do wysłania powyższych datagramów zostaje zawieszona na czas trwania limitu czasowego akcja wysłania datagramów CLEAR lub RESET.

W razie wysłania wszystkich datagramów CLEAR i RESET przed czasem $t(r2)$ węzeł DCE potwierdza restart przez transmisję pakietu DCE RESTART CONFIRMATION i ustawia kanały logiczne typu VC i PVC, związane z danym łączem, odpowiednio w stan p1 (przez skasowanie czasowych kanałów logicznych) i w stan d1. Przekroczenie czasu $t(r2)$ powoduje akcje takie jakie są przewidziane po wysłaniu pakietu DCE RESTART CONFIRMATION.

3.2. Obsługa fazy restartu w stanie r2

Podczas, gdy w stanie r2 nie została dokończona akcja wysyłania datagramów CLEAR i RESET z powodu braku buforów, mogą pojawiać się różne pakiety. Jeśli przyjdzie pakiet RESTART REQUEST w zerowym kanale logicznym, to jest on pomijany, ponieważ jest to ponowienie wcześniejszego zdarzenia. Pojawienie się innych pakietów powoduje ustawienie odpowiedniej do poprawności i rodzaju pakietu przyczyny błędu restartu i kodu diagnostycznego, a następnie przejście do obsługi błędnego restartu.

3.3. Obsługa błędnego restartu

W razie pojawienia się błędnego restartu wszystkie kanały logiczne, związane z danym łączem DTE/DCE, ustawia się w stan r3. Następnie zostają wysłane datagramy CLEAR INDICATION i RESET INDICATION odpowiednio

do wszystkich połączeń czasowych i stałych, związanych z danym łączem DTE/DCE. We wszystkich datagramach wskazuje się błąd procedury zdalnej i kod diagnostyczny. Do lokalnego DTE zostaje wysłany pakiet RESTART INDICATION z kodem diagnostycznym odpowiednim do błędu.

W polu przyczyny błędu restartu tego pakietu wskazuje się błąd procedury lokalnej. Czas nadesłania pakietu potwierdzenia DTE RESTART CONFIRMATION jest ograniczony wartością t_3 . Po tym czasie występuje retransmisja pakietu RESTART INDICATION w zerowym kanale logicznym. Retransmisje są powtarzane, aż do wyczerpania limitu retransmisji, po czym w kanałach związanych z danym łączem DTE/DCE następuje standardowa akcja kończąca, tj. kasowanie czasowych kanałów logicznych (przejście w stan p1), natomiast stałe kanały logiczne zostają przeprowadzone w stan d1.

3.4. Obsługa fazy restartu w stanie r3

W stanie r3 oczekuje się w zerowym kanale logicznym pakietu DTE RESTART CONFIRMATION bądź RESTART REQUEST, które są interpretowane jako potwierdzenie na wcześniej wysłany pakiet RESTART INDICATION. Jeżeli otrzymanym pakietem był DTE RESTART CONFIRMATION, to jeżeli był on odpowiedzią na wysłanie z DCE RESTART INDICATION, który z kolei został zainicjowany wywołaniem restartu łącza, to nastąpi wysłanie datagramów RESET INDICATION po wszystkich stałych połączeniach z informacją, że zdalne DTE jest operatywne. W razie otrzymania oczekiwanego w tej fazie pakietu następuje standardowa akcja kończąca, związana z danym łączem DTE/DCE. Datagramy są pomijane, a reakcje na inne pakiety podaje tab.1.

3.5. Obsługa restartu łącza DTE/DCE

Wszystkie kanały związane z naprawionym łączem DTE/DCE zostają ustawione w stan r3. Do lokalnego DTE zostaje wysłany pakiet RESTART INDICATION ze wskazaniem przyczyny restartu - sieć operatywna. Zasada oczekiwania na potwierdzenie, tj. wysłanie pakietu DTE RESTART CONFIRMATION, jest identyczna jak w punkcie 3.3. Po otrzymaniu przez DCE pakietu DTE RESTART CONFIRMATION zostają wysłane datagramy RESET INDICATION po wszystkich połączeniach stałych ze wskazaniem przyczyny restartu zdalnego DTE operatywnego.

4. KONTROLA, DIAGNOSTYKA I REJESTRACJA ZDARZEŃ

W fazie restartu jest prowadzona kontrola i diagnostyka ze względu na:

- poprawność syntaktyczno-semantyczną pakietu,

Restart połączeń logicznych w sieci ...

- poprawność kontekstową (tj. badanie czy pakiet/datagram jest dopuszczalny w stanie kanału logicznego),
- przekroczenie limitu czasowego,
- przekroczenie limitu retransmisji.

Kontrola ze względu na dwie pierwsze pozycje jest prowadzona w każdym stanie kanału logicznego. W wyniku tej kontroli zostaje:

- pominięty pakiet lub datagram (zwrot bufora, w którym jest zapisany do puli buforów wolnych),
- zapisana odpowiednia informacja o restarcie lub błędzie restartu, tj. zostaje ustawiony i wpisany kod przyczyny restartu i kod diagnostyczny do odpowiednich pól.

Przekroczenie limitu czasowego jest kontrolowane przez zadanie zegara wchodzące w skład oprogramowania węzła [7]. Po stwierdzeniu upływu limitu czasowego, w zależności od okoliczności, ponawia się pytanie o bufor potrzebny do wysłania pakietu RESTART INDICATION lub datagramów CLEAR INDICATION, RESET INDICATION. W przypadku otrzymania bufora dla pakietu RESTART INDICATION retransmituje się. Jednocześnie zwiększa się odpowiednio licznik liczby pytań o bufor i licznik retransmisji. Stany tych liczników porównuje się z limitami liczby pytań o bufor i limitami retransmisji. Limity te, podobnie jak limit czasowy, mogą być ogólnie inne dla każdego łącza DTE/DCE. Jeżeli jest przekroczony limit, to rozłącza się wszystkie czasowe i stałe połączenia logiczne związane z łączem DTE/DCE, dla którego nastąpiło przekroczenie limitu. Wymienione czynności po upływie limitu czasowego są zakodowane we fragmencie zawieszonym, który jest po tym czasie aktywowany przez zadanie zegara.

W wyniku kontroli w fazie restartu identyfikuje się błędy przytoczone w tabeli 2. Kod diagnostyczny w tabeli odpowiada diagnostyce i jest umieszczony w pakiecie lub datagramie w okcie 5.

W pakiecie (datagramie) zawartość pola kodu diagnostycznego nie zmienia znaczenia zawartości pola przyczyny. Niewyspecyfikowany kod diagnostyczny w pakiecie powoduje akceptację pola przyczyny przez DTE.

Może być więcej niż jeden błąd związany z pakietem. Porządek kontrolowania nie jest znormalizowany, dlatego kod pierwszego napotkanego błędu jest ustawiony.

W wyniku pojawienia się różnych zdarzeń w fazie restartu, w wyszczególnionych w tabeli 3 pakietach i datagramach są ustawione (w okcie 4) kody przyczyny restartu.

Do akcji standardowych wykonywanych w fazie restartu należą wyżej wymienione rodzaje kontroli oraz rejestracja zdarzeń odpowiednia do stanu kanału logicznego, źródła, typu pakietu oraz jego poprawności. W celu statystyki i diagnostyki pracy węzła są rejestrowane następujące zdarzenia:

Wykaz błędów wykrywanych w fazie restartu

Lp.	Typ pakie- tu, w któ- rym wykry- to błąd (zdarzenie)	Diagnostyka	Kod przy- czyny Kod diagno- styczny	Typ pakie- tu, w któ- rym umiesz- cza się ko- dy	Stan, w którym wykry- wa się błąd	Rodzaj akcji inicjo- wanej
1	2	3	4	5	6	7
1	RESTART REQUEST	Niewłaściwy identyfika- tor formatu ogólnego GFI	5/40	RESET INDICATION	p4	Wysłanie w zero- wym ka- nale lo- gicznym do lokal- nego DTE
2	RESTART REQUEST	Przekrocze- nie dopusz- czalnej dłu- gości pake- tu	5/39	RESET INDICATION	p4	"
3	RESTART REQUEST	Ustawienie pola przy- czyny (w okcie 4) niepoprawne	5/35	RESET INDICATION	p4	"
4	RESTART REQUEST	Pakiet zbyt krótki (bez pola przy- czyny i diag- nostyki)	5/38	RESET INDICATION	p4	"
5	DTE RE- START CON- FIRMATION	Niepoprawny pakiet w stanie p4	1/17	RESTART INDICATION	r1	Obsługa błędne- go res- tartu
6	DTE RESTART CONFIRMA- TION lub dopuszczal- ny pakiet akceptowa- ny w innych fazach	Niepoprawny pakiet w stanie r2	1/18	RESTART INDICATION	r2	"
7	Dowolny pakiet	Niewłaściwe GFI	1/40	RESTART INDICATION	r2	"
8	RESTART RE- QUEST lub RESTART CON- FIRMATION	Restart w niezerowym kanale lo- gicznym	1/41	RESTART INDICATION	r2	"
9	Pakiet spo- za dopusz- czalnych w protokole X.25	Niezidenty- fikowany pa- kiet	1/33	RESTART INDICATION	r2	"

1	2	3	4	5	6	7
10	RESTART REQUEST lub RESTART CONFIRMATION	Przekroczenie dopuszczalnej długości pakietu	1/39	RESTART INDICATION	r3	Obsługa błędnego restartu
11	Brak odpowiedzi	Upłynął limit czasowy	7/52 lub 1/52	RESTART INDICATION	r3	Wysłanie pakietu RESTART INDICATION

T a b e l a 3

Wykaz kodów przyczyny restartu wykorzystywanych w fazie restartu

Lp.	Typ zdarzenia	Kod przyczyny restartu	Znaczenie kodu	Typ pakietu lub datagramu, w którym umieszcza się kod przyczyny	Stan, w którym wykrywa się przyczynę restartu	Rodzaj inicjowanej akcji
1	Niewłaściwy pakiet RESTART REQUEST	5	Błąd lokalny	RESET INDICATION	p4	Wysłanie w zerowym kanale logicznym do DTE lokalnego
2	RESTART REQUEST w zerowym kanale logicznym	∅	Inicjacja od DTE	CLEAR INDICATION	p4	Wysłanie w sieć datagramów CLEAR i RESET
		∅	Inicjacja od DTE	RESET INDICATION	p4	
3	Dowolny pakiet oprócz prawidłowego RESTART REQUEST	17 3 1	Błąd zdalny błąd zdalny błąd lokalny	CLEAR INDICATION RESET INDICATION RESTART INDICATION	r2 r2 r2	Wysłanie w sieć datagramów CLEAR i RESET oraz pakietu RESTART do lokalnego DTE
4	Usunięcie uszkodzenia na łączu DTE/DCE	9 7	Zdalne DTE operatywne Sieć operatywna	RESET INDICATION RESTART INDICATION	Restart łącza	Wysłanie datagramów RESET w sieć, a pakietu RESTART INDICATION do lokalnego DTE

- odebrano żądanie restartu w zerowym kanale logicznym,
- odebrano formalnie błędny pakiet lub pakiet jest niezgodny ze stanem,
- przekroczono limit retransmisji,

- zakończono fazę restartu z określeniem liczby retransmisji pakietu RESTART INDICATION,
- restar łącza,
- brak bufora do wysłania pakietu lub datagramu,
- odrzucenie żądania uruchomienia po określonym czasie fragmentu programu, który np. uruchamia retransmisję pakietu RESTART INDICATION lub żąda przydzielenia bufora do wysłania pakietu/datagramu.

Rejestracja dwóch ostatnich zdarzeń wynika ze standardowych kontroli dostępności odpowiednich zasobów węzła DCF.

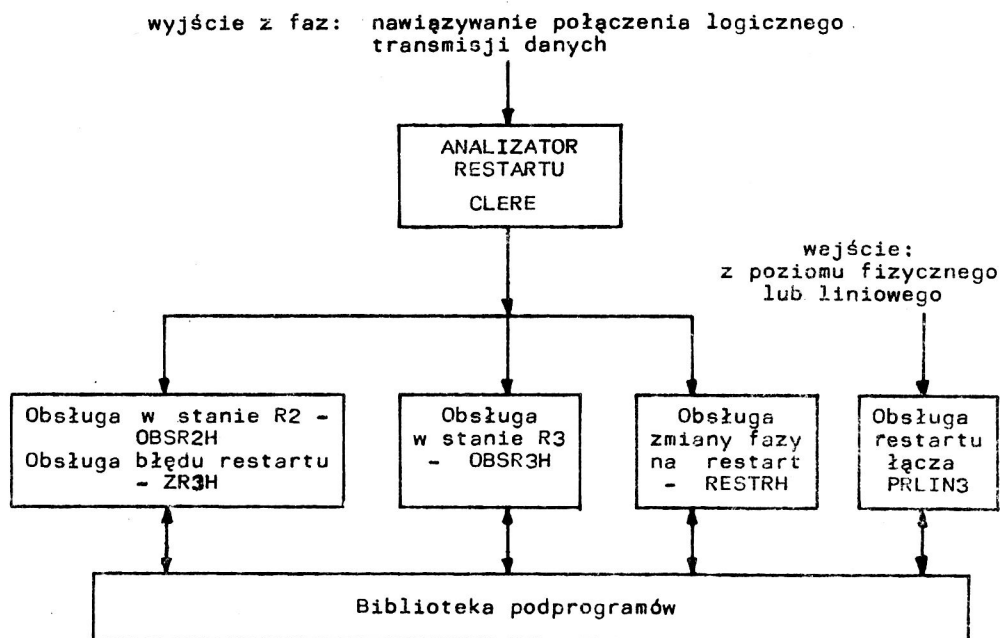
5. OPROGRAMOWANIE FAZY RESTARTU

5.1. Struktura oprogramowania

Oprogramowanie fazy restartu składa się z 4 podmodułów oraz podmodułu analizatora, wykorzystujących 10 własnych podprogramów i 3 fragmenty zawieszone oraz wiele podprogramów oprogramowania protokołu X.25 i wiele podprogramów globalnych (2,3,5-7) oprogramowania węzła (rys. 3). W skład oprogramowania fazy restartu wchodzi następujące własne podprogramy i fragmenty zawieszone:

- SZOKLH - szukaj OKL według $\emptyset$ -wego kanału logicznego i ustalonej linii (tj. łącza),
- WYSCRH - sterowanie wysyłaniem CLEAR-ów i RESET-ów w sieć,
- FZCRH - fragment zawieszony na bufor potrzebny do wysłania CLEAR-ów, RESET-ów i RESTART-u,
- WRESTH - uformowanie i wysyłanie pakietu RESTART INDICATION lub CONFIRMATION,
- RESTRF - wysyłanie RESET-ów w sieć,
- RESCLG - wysyłanie CLEAR-ów w sieć,
- SZOKLG - poszukiwanie opisu dla linii (tj. łącza),
- ODMINH - odmowa inicjacji fragmentu zawieszonego lub przekroczenia limitu retransmisji,
- RIHH - fragment zawieszony w celu ponownego wysłania pakietu RESTART INDICATION,
- WYRESH - wysłanie RESTARTU INDICATION oraz zainicjowanie fragmentu zawieszonego,
- FBUH - fragment zawieszony na bufor dla RESTARTU INDICATION,
- P1D1H - ustawienie stanów p1 i d1,
- KASKOH - kasowanie kolejek i fragmentów zawieszonych.

Przejęcie do fazy restartu odbywa się za pośrednictwem analizatora wspólnego podmodułu z fazą rozłączenia, który uzyskuje dane dla działania fazy zgodnie z ustalonymi warunkami i przekazuje sterowanie w razie



Rys.3. Struktura oprogramowania fazy restartu
Fig.3. Software structure of the restart phase

restartu, do jednego z trzech następujących podmodułów:

- obsługa fazy restartu w stanie r2 i obsługa błędu restartu,
- obsługa fazy restartu w stanie r3,
- obsługa zamiany fazy na restart (obsługa pakietu RESTART REQUEST).

Inny podmoduł - obsługa restartu łącza DTE/DCE - zostaje wywołany z poziomu fizycznego lub liniowego z przekazaniem w rejestrze R0 numeru łącza i formalnie jest traktowany jako podprogram.

5.2. Struktura danych

Komunikacja między podmodułami oraz między każdym podmodułem a wykorzystywanymi przez niego podprogramami odbywa się przez zastosowanie wspólnego obszaru danych. W skład wspólnego obszaru danych wchodzi:

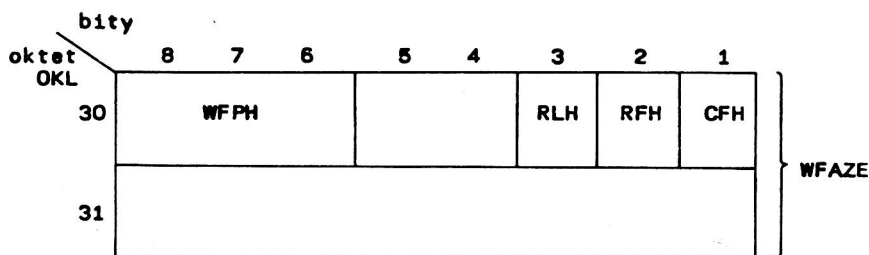
- opis kanału logicznego,
- bufor z datagramem/pakiemem,
- rejestry ogólnego przeznaczenia R0, R1, ..., R7,
- zmienne i dane robocze fazy restartu.

W fazie nawiązywania połączenia do opisu połączenia logicznego zostaje utworzona struktura danych, zwana opisem kanału logicznego - OKL [4,5].

W fazie restartu są wykorzystywane następujące pola:

- NKLWEE - nr kanału logicznego na wejściu, 2-bajtowe,
- NKLWYE - nr kanału logicznego na wyjściu, 2-bajtowe,
- PTWE - pole 1-bajtowe wskaźników, bit 8 równy 1, to połączenie typu PVC, lub typu VC - gdy równy 0,
- ANADE - adres nadawcy, 4-bajtowe,
- LIWEE - nr linii od nadawcy, 1-bajtowe,
- LIWYE - nr linii od odbiorcy, 1-bajtowe,
- SKLE - stan kanału logicznego, 1-bajtowe,
- WFAZE - pole wskaźników fazy, 2-bajtowe,
- CAUSE - kod przyczyny restartu, 2-bajtowe - młodszy bajt zawiera informację o przyczynie restartu, starszy zawiera kod diagnostyczny,
- LTIM1E - licznik limitu czasowego, 1-bajtowe - używane do zliczania liczby retransmisji pakietów RESTART INDICATION,
- LTIM2E - licznik limitu czasowego, 1-bajtowe - używane do zliczania pytań o wolne bufory,
- TOSTPE - chwila wysłania ostatniego pakietu, 2-bajtowe.

Strukturę pola WFAZE przedstawiono na rys. 4.



Rys.4. Struktura pola WFAZE dla fazy restartu
Fig.4. WFAZE field structure for the restart phase

Znaczenie wybranych bitów jest następujące:

- WFPH - faza pracująca, dla restartu = 101,
- RFH, CFH - wskaźniki prowadzonych transmisji; ustalone na 1, gdy trwa lub jest nie zakończona akcja wysłania odpowiednio datagramów RESET INDICATION i CLEAR INDICATION, w przeciwnym razie wyzerowane,
- RLH - wskaźnik ustawiony na 1, gdy został zapoczątkowany restart określonego łącza i wyzerowany w momencie gotowości węzła do nawiązywania połączeń na tym łączy (tzn. po wysłaniu informacji o gotowości węzła DCE do DTE lokalnego i DCE zdalnych).

Restart połączeń logicznych w sieci ...

Prawidłowa transmisja w fazie restartu jest prowadzona na łączu DTE/DCE w zerowym kanale logicznym. Opis OKL dla zerowego kanału logicznego zawiera NKLWEE = NKLWYE oraz LIWEE = LIWYE. Informacja o restarcie lub błędnym restarcie w datagramach CLEAR INDICATION i RESET INDICATION jest wysyłana w sieć w kanałach logicznych, związanych z odpowiednim zerowym kanałem logicznym (tj. związanych z odpowiednim łączem DTE/DCE). OKL dla zerowego kanału logicznego jest jakby reprezentantem pozostałych OKL z nim związanych.

Zatem pola: WFAZE (wskaźniki RLH, RFH i CFH), CAUSE, LTIM1E, LTIM2E służą zarówno do odczytu i zapisu tylko w zerowym OKL. Pozostałe pola, oprócz SKLE, są wykorzystywane we wszystkich kanałach logicznych tylko do odczytu zawartych w nich informacji i to przede wszystkim do wyboru kierunku transmisji. Pola SKLE i WFPH są wykorzystywane do zapisu i odczytu i tak, jak w innych fazach przechowują odpowiedni stan kanału ($r_2 = 10$, $r_3 = 11$) oraz aktualnie pracującą fazę.

Struktura danych bufora z datagramem i z pakietem jest opisana w pracy [4].

5.3. Parametry zewnętrzne i dodatkowe parametry lokalne

Faza restartu wykorzystuje parametry globalne, przekazywane jej w następujących rejestrach i odpowiadających ich komórkach:

- R0 - młodszy bajt zawiera przyczynę restartu, starszy bajt zawiera kod diagnostyczny,
- R2, R2E - adres OKL,
- R3, R3E - numer linii, po której przyszedł pakiet/datagram,
- R4, R4E - typ informacji, tj. pakiet/datagram,
- R5, R5E - adres bufora, w którym znajduje się pakiet/datagram.

W razie poprawnej transmisji informacji rejestr R0 = 0. Komórki R2E, ..., R5E są wykorzystywane do odnawiania parametrów globalnych.

Przyjęto zasadę, że parametry są przekazywane dla podprogramów przez rejestry maszynowe ze względu na szybkość wykonywania operacji na rejestrach.

W celu wielokrotnego uaktywniania w tym samym przedziale czasu tych samych fragmentów dodatkowo (oprócz OKL) zostały wykorzystane następujące zmienne parametryzowane numerem łącza DTE/DCE:

- zmienna, pod którą przechowuje się aktualny kod przyczyny restartu i kod diagnostyczny związany z danym łączem DCE/DTE w celu przesłania w datagramach CLEAR INDICATION,
- zmienna, pod którą przechowuje się kody jak wyżej, do przesłania w datagramach RESET INDICATION.

Warto zaznaczyć, że aktualny i związany z danym łączem DTE/DCE kod przyczyny restartu i kod diagnostyczny wysyłany w pakiecie RESTART INDICATION jest przechowywany w odpowiadającym zerowym opisie kanału logicznego i jest dostępny fragmentom zawieszonym, "pracującym" na tym OKL-u (nr OKL jest parametrem przekazywanym podczas inicjowania fragmentów zawieszonych).

Ze względu na to, że pewne wielkości nie są zdefiniowane przez protokół X.25 [1] oraz ze względu na założoną dużą elastyczność oprogramowania protokołu X.25, przyjęto jako parametry zewnętrzne:

- limit retransmisji pakietów RESTART INDICATION dla każdego łącza DTE/DCE,
- limit pytań o bufor w stanie r2 w celu wysłania datagramów CLEAR INDICATION i RESET INDICATION dla każdego łącza DTE/DCE,
- limit pytań o bufor jw. w stanie r3,
- limit czasowy w stanie r3 dla każdego łącza DTE/DCE,
- limit czasowy do ponowienia pytania o bufor w celu wysłania datagramów CLEAR INDICATION i RESET INDICATION i wysłania pakietu RESTART INDICATION dla każdego łącza DTE/DCE.

Za każdym razem, jeżeli czas oczekiwania na bufor w stanie r3 do wysłania pakietu RESTART REQUEST przewyższy limit czasowy dla tego stanu przyjęto, że nastąpi dynamiczne zmniejszenie limitu retransmisji pakietu RESTART REQUEST o 1.

5.4. Zasoby udostępniane fazie restartu

Oprogramowanie fazy restartu wymaga udostępnienia przez system sterujący węzła następujących zasobów (oprócz biblioteki podprogramów):

- buforów do uformowania pakietu/datagramu,
- obszaru pamięci do uformowania danych potrzebnych do aktywowania fragmentu zawieszonego po upływie limitu czasowego.

Wymienione zasoby są dostarczane przez specjalne podprogramy, które mogą być wywoływane w dowolnym podmodule z odpowiednimi parametrami. Jeżeli brak jest odpowiedniego zasobu, to te podprogramy zwrótnie o tym informują. W tych przypadkach zostaje zainicjowany odpowiedni fragment zawieszony lub nastąpi przejście do stanu rozłączenia czasowych i stałych połączeń logicznych, związanych z określonym lokalnym DTE. Również są dostępne podprogramy, które pozwalają zwracać wymienione zasoby do puli wolnych zasobów [7].

5.5. Zasady współpracy z innymi fazami

Podmoduły fazy restartu zostają uruchomione przez podmoduł fazy zestawienia połączenia, podmoduł fazy transmisji danych lub moduł poziomu HDLC. W fazie restartu sterowanie jest przekazane odpowiedniemu podmodułowi, który podejmuje stosowne akcje opisane wyżej bez przerywania swojego działania. W ten sposób jest zapewniona synchronizacja (wyłączność) działania w oprogramowaniu protokołu X.25, co gwarantuje dużą niezawodność oprogramowania. Podmoduły, do których wchodzi się z analizatora restartu, kończą swoje działanie przechodząc do wspólnej etykiety podprogramu, skąd przechodzi się do wyjścia z zadania.

6. UWAGI KOŃCOWE

Z tego względu, że protokół X.25 nie precyzuje wszystkich reguł postępowania, w wielu przypadkach należało je uściślić. Uściślenia przyjęte w pracy nie powodują ograniczeń w sensie możliwości sieci i nie są sprzeczne z tym protokołem.

Przyjęcie modułowego oprogramowania umożliwia wnoszenie poprawek i rozbudowanie o nowe funkcje.

LITERATURA

- [1] Draft Revised Recommendation X.25 Introduction and Packet Level, International Telegraph and Telephone Consultative Committee (CCITT), Nov. 1979.
- [2] D u b i e l e w i c z I., Kasowanie dróg logicznych w sieci komputerowej MSK (w tym samym zeszycie).
- [3] D u b i e l e w i c z I., K o l e ś n i k K., Reinicjacja transmisji danych w sieci komputerowej MSK. (w tym samym zeszycie).
- [4] F r y ź l e w i c z Z., Protokół pakietowy w sieci komputerowej MSK (w tym samym zeszycie).
- [5] F r y ź l e w i c z Z., M i k u l s k a E., Zestawienie połączeń logicznych w sieci komputerowej MSK. (w tym samym zeszycie).
- [6] K o l e ś n i k K., Transmisja danych w sieci komputerowej MSK (w tym samym zeszycie).
- [7] L e w o c J., L a n o w s k a B., S t a n i s z A., S z w a r c M., Zasady przygotowania oprogramowania węzła, Raporty Centrum Obliczeniowego PWR. Seria SPR nr 11/80, Wrocław 1980.

Praca wpłynęła do Redakcji 9.01.1981 r.

RESTART OF VIRTUAL CALL IN THE MSK COMPUTER NETWORK

This paper describes the principles of the services and the restart functions for the packet-switched recommendations of X.25 level 3. The restart software implementation for the MSK computer network node is also described.

Verified by Ruta Czaplińska

РЕСТАРТ УСТАНОВЛЕНИЯ ЛОГИЧЕСКИХ СВЯЗЕЙ
В МЕЖВУЗОВСКОЙ СЕТИ ЭВМ MSK

Представлены принципы обслуживания и функции рестарта пакетного протокола X.25 3-го уровня. Описано программное обеспечение рестарта для узла сети ЭВМ.

Проверила Малгожата Хейдрих

*Sieć komputerowa, węzeł,
protokół HDLC, protokół X.25*

Michał SZWARC*)

Implementacja protokołu liniowego w sieci komputerowej MSK

Przedstawiono protokół liniowy implementowany w Międzyuczelnianej Sieci Komputerowej - MSK. Omówiono jego realizację programową w węźle podsieci komunikacyjnej i wartości parametrów tego protokołu. Przedyskutowano ponadto kierunki przyszłych badań nad ulepszeniem protokołu liniowego oraz rozwiązań programowych.

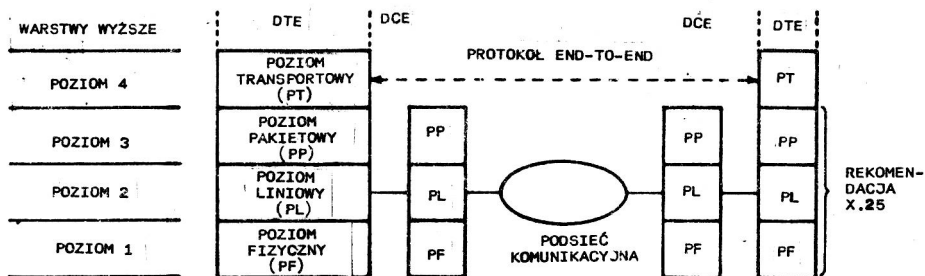
1. UWAGI WSTĘPNE

Od kilku lat Centrum Obliczeniowe Politechniki Wrocławskiej realizuje Międzyuczelnianą Sieć komputerową (MSK), mającą w wersji pilotowej objąć swoim zasięgiem trzy uniwersyteckie ośrodki obliczeniowe w Polsce: Wrocław, Warszawę i Gliwice [1].

Węzły podsieci komunikacyjnej będą zbudowane na radzieckich minikomputerach SM-3 oraz adapterach liniowych ALS-11, będących odpowiednikami adapterów DUP-11 produkowanych przez firmę DEC [2]. Zdecydowano, że będzie to sieć z komutacją pakietów, w której trzy najniższe warstwy będą oparte na zaleceniach X.25 CCITT [3] (rys. 1).

W pracy przedstawiono protokół liniowy zastosowany w powstającej sieci komputerowej MSK, omówiono jego realizację programową w węźle podsieci komunikacyjnej oraz wartości parametrów tego protokołu. Przedyskutowano również kierunki przyszłych badań protokołu liniowego i rozwiązań programowych.

*) Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.



Rys.1. Poziomy zdefiniowane zaleceniem X.25: DTE - DATA TERMINAL EQUIPMENT - urządzenie końcowe transmisji danych, DCE - DATA CIRCUIT EQUIPMENT - urządzenie komunikacyjne transmisji danych

Fig.1. Layers Covered by X.25 Recommendation: DTE - DATA TERMINAL EQUIPMENT, DCE - DATA CIRCUIT EQUIPMENT

2. PROTOKÓŁ LINIOWY SIECI MSK

2.1. Zastosowana wersja protokołu

W warstwie liniowej sieci MSK zastosowano protokół HDLC [4] w wersji zaleconej przez CCITT (choć jeszcze formalnie nie zatwierdzonej). Procedura ta, zwana procedurą dostępu do łącza (Link Access Procedure - LAP), dotyczy wymiany danych na styku DCE/DTE. Ze względu na symetryczność procedura ta będzie obowiązywała również wewnątrz podsieci komunikacyjnej na styku DCE/DCE. W węźle podsieci komunikacyjnej będzie zastosowana tylko wersja oznaczona symbolem LAPB, czyli wersja dla łącza zrównoważonego (balanced) z asynchronicznym trybem odpowiedzi, odpowiadająca procedurze BA-ISO z opcjami 2 i 8 [5,6].

2.2. Format i funkcje ramek

Wymiana informacji pomiędzy stacjami odbywa się przez ramki mające jednolity format pokazany na rys. 2. Transmisja każdej ramki jest poprzedzona i zakończona co najmniej jedną sekwencją flag $F = 01111110$. Flaga końcowa może być jednocześnie flagą początkową następnej ramki.

Pole adresowe A zawiera informację o miejscu powstania ramki (DCE lub DTE) oraz o rodzaju ramki ("komenda" lub "odpowiedź").

Pole sterujące C służy do oznaczania ramek następującego typu:

- informacyjnych,
- nadzorczych,
- nienumerowanych.

FLAGA	ADRES	POLE STERUJĄCE	POLE INFORMACYJNE	POLE KONTROLNE FCS	FLAGA
F 01111110	A 8 BITÓW	C 8 BITÓW	I N-BITÓW	FCS 16-BITÓW	F 01111110

Rys.2. Struktura ramki HDLC

Fig.2. HDLC Frame Structure

FORMAT	KOMENDY	ODPOWIEDZI	KOD			
INFORMACYJNA	I		0	N(S)	P	N(R)
NADZORCZE	RR	RR	1000	P/F	N(R)	
	RNR	RNR	1010	P/F	N(R)	
	REJ	REJ	1001	P/F	N(R)	
NIENUMEROWANE		DM	1111	F	000	
	SABM		1111	P	100	
	DISC		1100	P	010	
		UA	1100	F	110	
		FRMR	1110	F	001	

Rys.3. Typy ramek HDLC

Fig.3. HDLC command and responses

Formaty pola sterującego dla powyższych ramek pokazano na rys. 3.

Ramki informacyjne służą do przenoszenia danych, ramki nadzorcze do sterowania transmisją ramek informacyjnych, natomiast nienumerowane - do nawiązywania i rozłączania połączenia na poziomie liniowym.

Pole FCS jest sekwencją kontrolną ramki, służącą do sprawdzania transmisji ramki przez kanał transmisyjny.

2.3. Opcje protokołu i przyjęte parametry

Implementacja protokołu liniowego wymaga określenia jego parametrów oraz opcji. Przyjęcie konkretnego rozwiązania jest zależne od stosowanego sprzętu, m.in. od rodzajów modemów, jakości łączy, pojemności pamięci węzłów.

W zaleceniu X.25 pozostawia się pewne parametry i opcje niustalone, zaznaczając, że przyjęcie konkretnych rozwiązań pozostawia się do decyzji projektantów sieci lub umowy z administracją poczty. Na ogół przyjmuje się je arbitralnie na podstawie badań symulacyjnych prowadzonych na modelu, w wyniku żmudnych wyliczeń matematycznych, lub wreszcie na zasadzie analogii, przez porównanie podobnych wersji zrealizowanych sieci. Po przyjęciu ustaleń wstępnych przechodzi się do badań istniejącej wersji, których celem jest taka modyfikacja rozwiązania, która prowadziłaby do poprawy jakości oraz niezawodności przesyłanej informacji przez sieć.

Tak postąpiono również w pilotowej sieci MSK, gdzie oparto się na zaleceniach dla sieci w krajach RWPG [7] i spotkaniu roboczym "Ustalenie norm nt. podsieci komunikacyjne sieci komputerowych" w Szklarskiej Porębie w marcu 1980 r.

Podstawowymi parametrami i opcjami przyjętej procedury liniowej są:

- a) czas oczekiwania na potwierdzenie - T_1 ,
- b) maksymalna liczba niepotwierdzonych ramek informacyjnych (okno) - k ,
- c) maksymalna liczba powtórzeń transmisji - N_2 ,
- d) maksymalna liczba oktetów w ramce informacyjnej - N_1 ,
- e) ustawienie bitu P i F ,
- f) zasada potwierdzania przyjmowanych ramek informacyjnych.

2.3.1. Czas oczekiwania na potwierdzenie

Czas oczekiwania T_1 jest to maksymalny czas oczekiwania reakcji na wysłaną ramkę lub grupę ramek typu informacyjnego, tzn. czas, po którym następuje retransmisja tej ramki, przejście do odpowiedniej procedury rozłączania albo ponownego nawiązania połączenia, lub też ustalenia stanu łączy komendą nadzorczą.

Implementacja protokołu liniowego w sieci ...

Minimalny czas oczekiwania na potwierdzenie wyniku z czasu przesyłania najdłuższej dopuszczalnej ramki (139_{10} oktetów w sieci MSK) przez kanał o ustalonej przepustowości, czasu na jej przetwarzanie oraz czasu trwania transmisji ramki potwierdzającej wraz z jej przetwarzaniem w stacji nadawczej. W decydujący sposób na wartość tego parametru ma wpływ szybkość linii.

We wstępnej realizacji dla sieci MSK przyjmuje się, że czas oczekiwania będzie wynosił 4 s. Jest to zgodne z wartością tego parametru, przyjętą przez Łódzką Akademię Nauk [7] dla transmisji nie mniejszej niż 600 bit/s.

Przewiduje się, że w przyszłości możliwe będzie wprowadzenie szybszych transmisji. W związku z tym istnieje potrzeba, aby węzły podsieci komunikacyjnej były tak zrealizowane, aby mogły reagować w czasie mniejszym niż czasy podane w tabeli 1.

T a b e l a 1

Zależność czasów oczekiwania od szybkości transmisji w sieci
MSK

Szybkość transmisji (bit/s)	Czas oczekiwania (s)
600-1200	5
1200-2400	4
2400-4800	3
ponad 4800	2

Podczas eksploatacji systemu zostaną przeprowadzone odpowiednie badania symulacyjne, służące do wyznaczenia optymalnej wartości czasu oczekiwania na odpowiedź przy różnych szybkościach transmisji.

2.3.2. Maksymalna liczba niepotwierdzonych ramek

Maksymalna liczba niepotwierdzonych ramek informacyjnych (k) może wynosić 7, ponieważ procedura X.25 LAPB nie przewiduje rozszerzenia numeracji ramek informacyjnych ponad zakres 0-7^{*}).

Zbyt małą liczbą k może sztucznie powstrzymywać wysyłanie kolejnych ramek informacyjnych z danymi do chwili otrzymania potwierdzenia. Gdy liczba k jest mała, wówczas potwierdzenia będą musiały być wysyłane częściej, co będzie obciążało kanał transmisyjny, który w sieci MSK jest bardzo wolny - 2400 bitów/s. Gdy zaś liczba k jest zbyt duża, wówczas trzeba retransmitować dużą liczbę ramek informacyjnych w razie, gdy nastąpi przekłamanie w jednej z początkowych ramek w sekwencji.

^{*}) W procedurze HDLC istnieje możliwość rozszerzania okna do 128 ramek [8].

Konkluzja z tych pobieżnych rozważań jest następująca: większy rozmiar okna można będzie zastosować w razie korzystania z kanałów transmisyjnych lepszej jakości oraz w węzłach z odpowiednią pamięcią na buforowanie ramek, kiedy prawdopodobieństwo retransmisji jest stosunkowo małe. Mniejszy rozmiar - dla kanałów gorszej jakości, gdy spodziewamy się, że mechanizm retransmisji będzie musiał być częściej stosowany.

W sieci MSK z zastosowaniem kanałów dwukierunkowych o szybkości 2400 bitów/s oraz pamięci na bufory - 5k słów, przyjęty wstępnie maksymalny dopuszczalny rozmiar okna wynosi 7.

2.3.3. Maksymalna liczba retransmisji ramek

Retransmisją ramki jest próba wyjścia z sytuacji wyjątkowej. Duża liczba retransmisji może być stosowana w łączach z małą stopą błędów. Nie można niestety wykluczyć szczególnego przypadku, np. przerwania łączności, nagłego pogorszenia właściwości transmisyjnych łącz, utraty synchronizacji itp., gdy nieograniczona liczba powtórzeń w procedurze kontroli poprawności transmisji może doprowadzić do stanu bez wyjścia. Zbyt mała liczba retransmisji mogłaby natomiast doprowadzić do wyłączenia sprawnej linii, w której mogło dojść do chwilowego tylko, przemijającego zakłócenia, co jest zjawiskiem częstym w łączach komutowanych.

W sieci MSK przyjęto zgodnie z zaleceniem [7] wartość tego parametru $N2 = 3$.

2.3.4. Maksymalna liczba oktetów w ramce informacyjnej

Maksymalna liczba oktetów w ramce informacyjnej $N1$ jest parametrem systemowym zależnym od maksymalnego pola danych, transmitowanego przez styk DCE i DTE lub DCE i DCE. Długość tego pola zależy praktycznie tylko od pojemności pamięci oraz od charakterystyk występowania błędów w kanale transmisyjnym. Biorąc pod uwagę te zależności wyznaczono wstępnie wartość tego parametru,

Zakłada się, że liczba $N1$ wynosić będzie 135 oktetów (128 oktetów pola informacyjnego oraz 7 oktetów organizacyjnych) - do przesyłania pomiędzy DTE i DCE lub 139 oktetów (128 oktetów pola danych, 7 oktetów organizacyjnych oraz 4 oktety zawierające adres nadawcy i odbiorcy) - do przesyłania w podsięci komunikacyjnej.

Nie przewiduje się na razie implementacji opcji FAST SELECT (poziom 3 protokołu X.25), w której dopuszcza się transmisję dłuższych ramek niż 139 oktetów.

2.3.5. Zastosowanie bitu P oraz F

Bit P będzie ustawiony podczas retransmisji ramek SABM, DISC w komendach nadzorczych ustalających stan łącza oraz w ramach informacyj-

nych. Ustawienie bitu P w komendach zgodnie z procedurą X.25 LAPB wymusza potwierdzenie takiej ramki odpowiedzią z ustawionym bitem F.

W węźle sieci MSK przyjęto, że bit P ustawia się ponadto w ramce informacyjnej ostatniej w kolejce wyjściowej do danej linii lub ostatniej w oknie, w celu wymuszania potwierdzenia od stacji odbiorczej.

2.3.6. Zasada potwierdzania przyjmowanych ramek informacyjnych

W procedurze X.25 LAPB nie istnieje konieczność potwierdzenia ramek informacyjnych z nieustawionym bitem P. Wybór momentu generowania potwierdzenia pozostawia się do uznania projektantowi.

W tym przypadku należy wybierać pomiędzy częstym potwierdzeniem, które zwiększa ruch w kanałach, a zbyt rzadkim - mogącym powodować chwilowe powstrzymywanie dalszej transmisji ramek informacyjnych do chwili nadejścia potwierdzenia.

W węźle komputerowej sieci MSK zastosowano metodę potwierdzania co trzeciej ramki przez transmisję ramki nadzorczej wtedy, gdy w kolejce wyjściowej brak jest ramek informacyjnych również przenoszących potwierdzenie.

3. OBSŁUGA RAMKI HDLC W WĘZLE SIECI MSK

W fazie projektowania sieci komputerowej jest ustalany sposób realizacji protokołu liniowego. Funkcje tej warstwy mogą być realizowane sprzętowo lub programowo. Przyjęto, że w sieci MSK adapter liniowy ALS11 [9] będzie realizował sprzętowo następujące funkcje protokołu liniowego X.25 LAPB:

1. Synchronizację oraz fazowanie.
2. Zabezpieczenie przed błędami transmisji:
 - a) sprawdzenie ciągu kontrolnego ramki (FCS),
 - b) przerywanie transmisji (znaki ABORT),
 - c) zapewnienie przezroczystości transmisji przez wstrzykiwanie zer (bit stuffing).

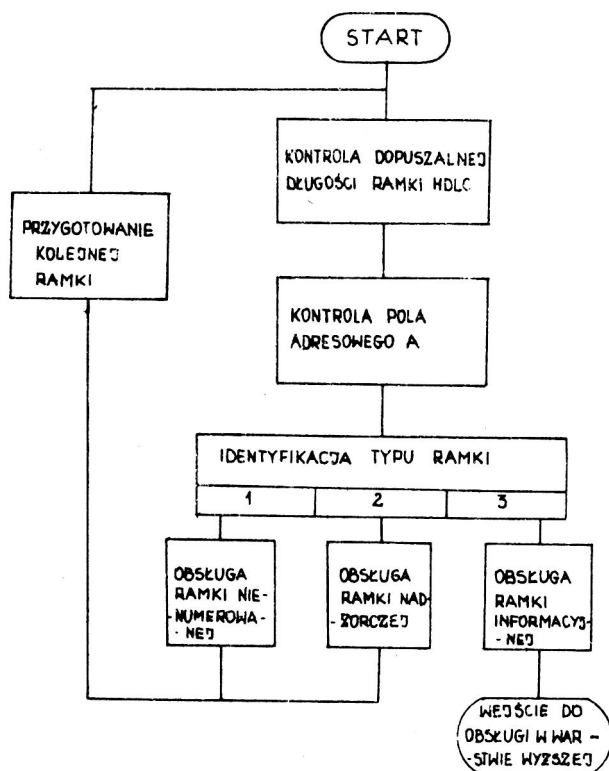
Pozostałe funkcje tej warstwy będą realizowane przez oprogramowanie operacyjne węzłów oraz komputerów komunikacyjnych.

3.1. Fazy obsługi ramki HDLC

Ogólny schemat blokowy obsługi ramki HDCC w węźle sieci MSK, zgodnie z przyjętą procedurą X.25 LAPB, pokazano na rys. 4.

Wstępna obsługa ramki HDLC polega na kontroli poprawności kodu adresu, po którym rozpoznaje się czy ramka przychodząca z danej linii jest

komendę czy odpowiedź. Przyjęto, że komendy przesyłane na styku DCE/DCE będą posiadały kod adresu typu (00000011), a odpowiedzi typu A kod typu B (00000001). Kody stosowane na styku DTE/DCE pozostaną natomiast bez zmian zgodnie z protokołem [3].



Rys.4. Schemat blokowy obsługi ramki HDLC w węźle sieci MSK

Fig.4. Structure of HDLC frame service task in the node of MSK computer network

Po tej analizie następuje kontrola minimalnej długości ramki. Ramki mające błędny kod adresu lub błąd długości są odrzucane i nie są dopuszczane do dalszego przetwarzania.

Po przejściu przez tę fazę odbywa się identyfikacja typu ramki oraz wejścia do odpowiedniego modułu obsługi, gdzie odbywa się dalsza obróbka polegająca m.in. na zmianie stanu kanału transmisji zgodnie z tabelą kontroli przepływu ramek informacyjnych (patrz tabela 2), wysyłaniu ramek potwierdzających itp.

Tablica zmian stanu linii po odebraniu ramki HDLC

Ramka \ Stan linii	SABM	DISC	DISCP	FRMR	ITP	ACS
SABM	SABM (UA)	DISCP (DM)	ITP (UA)	ITP (UA)	ITP (UA)	ITP (UA)
DISC	DISCP (DM)	DISC (UA)	DISCP (DM)	DISCP (UA)	DISCP (UA)	ACS
UA	ITP	DISCP	DISCP	FRMRC	DISCP (DM)	ACS
DM	SABM (SABM)* lub ACS	SABM (SABM)	DISCP	SABM (SABM)	SABM* (SABM)	ACS
FRMR	SABM* (SABM) lub ACS	SABM* (SABM)	DISCP	FRMRC* (FRMR) lub SABM (SABM)	SABM (SABM)	ACS
RR (komenda) RNR (komenda) REJ (komenda) I (z P = 1)	SABM	DISC	DISCP (DISC)	FRMRC* (FRMR) lub SABM (SABM)	ITP	ACS
RR (odpowiedź) RNR (odpowiedź) REJ (odpowiedź) I (z P = 0)	SABM	DISC	DISCP	FRMR	ITP	ACS

Uwaga 1. Bez nawiasów podano stany linii, w nawiasach - typy ramek wysyłanych po przejściu linii do nowego stanu (brak tego oznacza niewysyłanie żadnej ramki).

Uwaga 2. Gwiazdką oznaczono rozwiązania arbitralnie przyjęte przez autora wtedy, gdy protokół nie wskazuje konkretnego rozwiązania lub podaje rozwiązanie alternatywne.

Jeżeli odebrana i przetworzona przez poziom liniowy ramka jest ramką informacyjną z niezerowym polem danych, to następuje przejście do obsługi na wyższym poziomie pakietowym. W przypadku przeciwnym powraca się do programu wprowadzającego informację.

4. PRACA PO PRZEKROCZENIU CZASU OCZEKIWANIA NA ODPOWIEDŹ

4.1. Czas oczekiwania na odpowiedź

W skład oprogramowania poziomu liniowego wchodzić będą tzw. fragmenty zawieszone, uruchamiane po upływie czasu limitu oczekiwania na określoną ramkę. Ich funkcja sprowadza się przeważnie do retransmisji tej samej ramki, dla której ostatnio było sygnalizowane przekroczenie czasu oczekiwania na odpowiedź. Założono, że odmierzenie tego czasu dokonywane będzie w węźle dla wszystkich komend, ramek DM, FRMR oraz po zablokowaniu współpracującej sąsiedniej stacji DCE lub DTE z powodu braku pamięci.

Po przejściu czasu oczekiwania na ramkę informacyjną przed retransmisją będzie wysyłana komenda nadzorcza w celu ustalenia stanu sąsiedniej stacji. Dopiero potem ewentualnie następować będzie retransmisja ramek informacyjnych. Szczególna sytuacja nastąpi po trzech przejściach czasu oczekiwania na ramkę FRMR. Wysłana będzie wtedy komenda SABM w celu nawiązania połączenia liniowego. Po przejściu czasu na okres zablokowania sąsiedniej stacji następować będzie próba ustalenia stanu tej stacji przez wysłanie do niej komendy nadzorczej.

Interesującą propozycję na temat innego sposobu reakcji po przejściu czasu oczekiwania na odpowiedź przedstawiono w pracy [10].

5. KIERUNKI BADAŃ ORAZ UWAGI KOŃCOWE

Należy podkreślić, że protokół HDLC jest coraz częściej brany pod uwagę przez instytucje realizujące sieci komputerowe. Jest on na przykład zrealizowany w kanadyjskiej sieci DATAPAC, jest uwzględniany w architekturze sieci komputerowej EIN oraz minikomputerowej sieci firmy Honeywell. Ponadto JIS (Japanese Industrial Standards) zdecydowało się wykorzystać protokół HDLC w swojej sieci DCNA (Data Communication Network Architecture) ze względu na jego efektywność i niezawodność [11]. Protokół ten jest również brany pod uwagę przez Łódzką Akademię Nauk, realizującą sieć z komutacją pakietów [7].

Przyjęta wersja protokołu liniowego jest jedną z możliwych do przyjęcia. Nie jest ona na pewno pozbawiona wad. Jej poprawność, a także poprawność przyjętych wartości parametrów zostanie zweryfikowana podczas eksploatacji systemu. W sieci MSK są planowane badania analityczne i doświadczalne, które będą prowadzone na obiekcie symulowanym i na obiekcie rzeczywistym, w wyniku których powinno otrzymać się optymalne

wartości parametrów i optymalną strukturę procedury ze względu na przyjęte podstawowe kryterium jakości, jakim jest opóźnienie tranzytowe pakietu. Zaletą obecnego rozwiązania programowego jest to, że wykazuje ono strukturę modułową, dzięki której wprowadzanie lub usuwanie pewnych fragmentów nie będzie w zasadzie żadnym problemem.

Myśli się obecnie, aby w przyszłości - wobec postępu technologicznego - większość funkcji protokołu było spełnianych przez mikroprocesory.

LITERATURA

- [1] B a r c h a ń s k i I., B a z e w i c z M., Projekt wstępny międzyuczelnianej sieci komputerowej, cz. I., Dokumentacja Międzyuczelnianej Sieci Komputerowej, Seria SC2, Dok. nr 001, Raporty Inst. Cybern. Techn. PWr., Nr 268, Wrocław 1978.
- [2] PDP - 11 Peripherals Handbook, Digital Equipment Corporation, Maynard, Massachusetts 1975.
- [3] CCITT Orange Book, vol. UIT, Geneve 1977.
- [4] Proposed Revision to Level 1 and 2 of Recommendation X.25 as Approved by Working Party, 2-5 April 1979, CCITT - October 1979.
- [5] ISO Addendum 1 TC 97 SC 6 N 1801 - 1979.
- [6] ISO Addendum 2 Proposed Draft TC 97 SC 6 N 1804 - 1979.
- [7] P o d v y s o c k i j I., R o t a r o v V., Bit orientirovannyj protokol upravlenija informacionnym kanalom dostupa v seti s kommutaciej paketov, Latvijskaja Akademijskaja Nauk, Riga 1979.
- [8] B a z e w i c z M. (red.), Własności i funkcje sieci komputerowych cz. I. Protokoły, Wyd. PWr., Wrocław 1979.
- [9] B a t y c k i T. i in., Adapter liniowy synchroniczny ALS 11. Dokumentacja techniczna, Raporty Inst. Cybern. Techn. PWr., Seria SPR nr 26/81, Wrocław 1981.
- [10] B i e l e n i n i k E., S t a n i s z A., S z w a r c M., Obsługa adapterów liniowych w węźle podsieci komunikacyjnej (w niniejszym zeszycie).
- [11] A b e M a s h i o M., N a k a t a H., T a j i m a T., Protocol Design for Data Communication Network Architecture - Review of the Electrical Communication Laboratories, vol. 27, May-June 1979.

Praca wpłynęła do Redakcji 15.12.1980 r.

Po poprawieniu 3.03.1982 r.

IMPLEMENTATION OF LINE LEVEL PROTOCOL IN THE MSK COMPUTER NETWORK

The paper presents a line level CCITT X.25 protocol implemented in the MSK computer network. The structure of HDLC frame service task, the accepted values of the parameters of the protocol and implemented version are given. Directions of future investigations on the improvement of line level protocol implemented in MSK computer network are also discussed.

Verified by Ruta Czaplińska

ВНЕДРЕНИЕ ЛИНЕЙНОГО ПРОТОКОЛА СЕТИ ЭВМ MSK

Описывается линейный протокол, используемый в межузловой сети ЭВМ (MSK). Представлены его программная реализация в узле подсети связи, величины параметров и принятая версия. Рассматриваются направления будущих исследовательских работ, связанных с улучшением линейного протокола и его программных решений.

Проверила Малгожата Хейдрих

*Sieć komputerowa,
pomiar, diagnostyka*

Elżbieta HUDYMA^{*)}, Elżbieta KOSMULSKA-BOCHENEK^{*)}

Rejestracja pomiarów w sieci komputerowej MSK

Przedstawiono koncepcję rejestracji zdarzeń w węźle pomiarowym sieci komputerowej MSK. Zarejestrowane zdarzenia stanowią bazę pomiarową niezbędną dla statystyki, sterowania siecią diagnostyki itp. Opisana jest również organizacja danych niezbędna do realizacji rejestracji zdarzeń.

1. WSTĘP

Sieci komputerowe stanowią dość skomplikowany mechanizm współdziałania elementów sprzętowych i programowych. Zarówno uruchomienie sieci, jak i jej eksploatacja i modernizacja, wymagają dokładnej znajomości procesów zachodzących w sieci pod względem jakościowym i ilościowym. Możliwe to jest przez wbudowanie już na etapie projektowania sieci integralnie związanego z nią systemu pomiarowego.

W systemie pomiarowym zaprojektowanym dla sieci komputerowej MSK wydzielono niezależny podsystem miernictwa dotyczący podsieci. W praktyce oznacza to przede wszystkim rejestrowanie poszczególnych aspektów pracy węzła w podsieci, a następnie współpracy między węzłami składającymi się na podsieć.

Przetworzone wyniki pomiarów służą do sterowania, diagnostyki, oceny efektywności sieci i obliczania kosztów. Na ich podstawie można oszacować takie istotne parametry, jak np.:

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

- czas przebywania pakietu w węźle,
- przepustowość węzła,
- niezawodność podsieci itp.

Do oszacowania tych i innych parametrów konieczne jest posiadanie bazy pomiarów elementarnych, składających się z rejestracji poszczególnych zdarzeń zachodzących w węźle w trakcie normalnej pracy sieci oraz stanów w jakich się on znajduje. Celem niniejszego artykułu jest przedstawienie podsystemu rejestracji tworzącego bazę pomiarową dla dalszego przetwarzania.

Do rejestracji jest wytypowany arbitralnie pewien zbiór zdarzeń oraz stanów będący w chwili projektowania sieci, zdaniem autorów, zbiorem reprezentatywnym stanowiącym o stanie i pracy sieci. Zbiór ten nie jest zbiorem zamkniętym i w trakcie eksploatacji może w razie potrzeby ulegać zmianom. Jednym z podstawowych kryteriów w projektowaniu systemu miernictwa była jego możliwie duża elastyczność, zwłaszcza w zakresie:

- liczby rejestrowanych zdarzeń,
- zbioru rejestrowanych zdarzeń,
- sposobu rejestracji dodatkowych parametrów charakteryzujących zdarzenia.

System rejestracji podzielono na 2 grupy:

- rejestrację zdarzeń,
- rejestrację stanów.

Kryterium stanowi tu przede wszystkim tryb wykonywania pomiarów. Pierwsza grupa jest realizowana w sposób **permanentny** w trakcie całego czasu pracy sieci, druga - w sposób **okresowy**.

2. REJESTRACJA ZDARZEŃ

Pracę węzła można rozpatrywać z punktu widzenia rozpoznawania i obsługi zdarzeń napływających do węzła z otoczenia oraz generowania pewnych zdarzeń do otoczenia. Do rejestracji wybrano określony zbiór zdarzeń charakteryzujących pracę węzła w sieci. Każde rejestrowane zdarzenie ma przyporządkowany jednoznacznie numer. W tabeli podano przyporządkowanie numerów do określonych zdarzeń. Każde wystąpienie zdarzenia z danego zbioru powoduje w programie obsługującym dane zdarzenie wywołanie podprogramu miernictwa, którego zadaniem jest zarejestrowanie danego zdarzenia. Po zarejestrowaniu, które musi trwać tak krótko, aby możliwie mało wpływało na zmniejszenie efektywności pracy, węzeł powraca do wykonywania normalnych funkcji. Najprostsza rejestracja zdarzenia polega na zwiększeniu o 1 licznika przyporządkowanego danemu zdarzeniu. Rejestracja złożona składa się z rejestracji prostej, czyli

Tabela aktualnie rejestrowanych zdarzeń

Numer zdarzenia	Informacja dodatkowa	Typ bloku informacyjnego NPROGK	Znaczenie
1	2	3	4
0	-	-	Rezerwa
1	-	-	Zerowy adres fragmentu zawieszonego w tablicy limitów czasowych
2	-	-	Przepełnienie tablicy limitów czasowych w podprogramach
3	-	-	Żądanie kasowania limitów czasowych nie istniejących w tablicy
4	-	-	Żądanie zegara - wywołane przy zerowym semaforze
5	-	-	Pusta kolejka transmisji dla monitora ekranowego podczas wejścia do fragmentu zawieszonego
6	-	-	Błąd transmisji w sterowaniu DZM
7	-	-	Pusta kolejka transmisji dla DZM podczas wejścia do fragmentu zawieszonego
8	-	-	Błąd transmisji w sterowaniu PT
9	-	-	Błąd transmisji w sterowaniu CT
10+15	-	-	Rezerwa
16	-	-	Próba oddania do puli bufora o adresie spoza obszaru puli buforów
17	-	-	Próba kierowania opisu kanału logicznego nie znajdującego się na liście OKL
18	-	-	Rezerwa
19	-	-	Rezerwa
20	-	-	Wywołanie podprogramu DKWYP3 dla nie istniejącej linii
21	-	-	Wywołanie podprogramu DKWYK3 dla nie istniejącej linii
22	-	-	Wywołanie podprogramu DKWYI3 dla nie istniejącej linii
23	-	-	Wywołanie podprogramu OKWY3 dla nie istniejącej linii

1	2	3	4
24	-	-	Wywołanie podprogramu LKWY3 dla nie istniejącej linii
25÷31	-	-	Rezerwa
32	-	-	Błędne wywołanie zadania poziomu 3/2
33	-	-	Instrukcja TRAP z niezerowym młodszym bajtem lub błędnym typem albo trybem
34	-	-	Przepełnienie tablicy oczekujących zadań poziomu 2 (nowe zadanie odrzucone)
35	-	-	Błędne uaktywnienie poziomu 2/3
36	-	-	Przepełnienie tablicy uaktywnień zadań poziomu 2/3 (stare zadania odrzucone)
37	-	-	Przepełnienie tablicy uaktywnień zadań poziomu 3/2
38÷61	-	-	Rezerwa
62	-	-	Błędne wywołanie M17
63	-	-	Brak linii z ustawionym semaforem
64	-	-	Za krótka ramka
65	-	-	Abort w ramce
66	-	-	Błąd adaptera odbiornika - zapalone bity zarezerwowane
67	-	-	Błąd CRC
68	-	-	Błąd za długiej ramki
69	-	-	Błąd ramki z niedopuszczalnym polem informacyjnym
70	-	-	Błąd OVERRUN
71	-	-	Ramka RNR
72	-	-	Błąd N(R)
73	-	-	Błąd N(R) (odbiór REJ)
74	-	-	Nieimplementowany protokół APB, gdy $B = \emptyset$
75	-	-	Niedopuszczalny kod pola adresowego

1	2	3	4
76	-	-	Licznik nadawań ramki SABM wyzerowany
77	-	-	Licznik nadawań ramki FRMR wyzerowany
78	-	-	Retransmisja ramek sterujących
79	-	-	Retransmisja ramek informacyjnych z powodu przekroczenia limitu czasowego
80	-	-	Odbiór ramki FRMR
81	-	-	Wyłączenie linii
82-95	-	-	Rezerwa
96	-	-	Poprawnie nawiązana transmisja
97	-	-	Rozłączenie z braku wolnego OKL
98	-	-	Rozłączenie z braku wolnych kanałów na wyjściu
99	-	-	Rozłączenie z powodu stanu restartu na wyjściu
100	-	-	Rozłączenie z powodu kolizji wywołań na wyjściu (rozłączenie połączenia zdalnego)
101	-	-	Rozłączenie z powodu kolizji wywołań i braku wolnych buforów na wyjściu (rozłączenie połączenia zdalnego i lokalnego)
102	-	-	Rozłączenie z powodu zapętlenia
103	-	-	Rozłączenie ze względu na brak przejścia (awaria dróg fizycznych)
104	-	-	Rozłączenie z braku wolnych buforów
105	-	-	Rozłączenie z innych przyczyn, np. odrzucenie żądań przyjęcia fragmentu zawieszonego
106	nr linii	1	Rozłączenie z powodu osiągnięcia limitu retransmisji (głucha linia)
107	-	-	Błędny pakiet (formalnie lub w kontekście)
108	-	-	Błędny datagram (formalnie lub w kontekście)

1	2	3	4
109	Numer linii	1	Wygenerowanie restartu na poziomie X.25
110	-	-	Odebranie żądania restartu od PS węzła
111	Numer linii	1	Odebrano żądanie restartu od KK
112	Liczba retransmisji	3	Zakończenie fazy restartu
113	-	-	Odebranie pakietu CLEAR REQUEST (CLR)
114	-	-	Odebranie datagramu CLEAR INDICATION
115	-	-	Wygenerowanie CLEAR INDICATION (błędny pakiet, błędny datagram lub inne)
116	Liczba retransmisji CLI-pakiet	3	Zakończenie generowane w podsięci fazy Clear - kierunek lokalny KK
117	Liczba retransmisji CLI - datagram	3	Zakończenie generowanej w podsięci fazy Clear - kierunek zdalny KK
118	-	-	Odebranie pakietu RESET REQUEST
119	-	-	Odebranie datagramu RESET INDICATION
120	-	-	Wygenerowanie pakietu/datagramu RESET INDICATION z powodu błędu w przepływie danych
121	-	-	Wygenerowanie pakietu/datagramu RESET INDICATION
122	Liczba retransmisji	3	Zakończenie generowanej w podsięci fazy Reset - kierunek lokalny KK
123	Liczba retransmisji	3	Zakończenie generowanej w podsięci fazy Reset - kierunek zdalny KK
124	1 - wstawić do kolejki -1 - usunąć z kolejki	6	Zmiana długości kolejki zablokowanych pakietów/datagramów
125	-	-	Zarejestrowanie braku wolnego bufora
126	-	-	Zarejestrowanie pakietu CAR z niedopuszczalnymi adresami (niepoprawna długość lub treść)
127	-	-	Rezerwa
128	-	-	Odebranie pakietu, datagramu, datagramu miernictwa

1	2	3	4
129	-	-	Odebranie pakietu informacyjnego
130	-	-	Odebranie datagramu informacyjnego
131			Rezerwa
132	Numer błędu diagnostyki i pomiarów	3	Błąd rejestracji zdarzeń
133÷158	-	-	Rezerwa
159	Chwila zajęcia bufora (R1-młodsza część, R2 - starsza)	5	Oddanie bufora do puli
160	-	-	Odebranie ramki HDLC
161	-	-	Odebranie ramki informacyjnej
162÷190	-	-	Rezerwa
191	Chwila załadowania OKL (R1-młodsza część, R2 - starsza)	5	Zwolnienie bloku OKL

zliczania zdarzeń, oraz specjalnej rejestracji pewnych parametrów związanych z tym zdarzeniem. Numery przyporządkowane zdarzeniom nie są uporządkowane w stosunku do ich treści merytorycznej. Dzięki temu możliwa jest w przyszłości dowolna zmiana zbioru rejestrowanych zdarzeń. Wyróżniono jednak 2 grupy zdarzeń.

Pierwsza grupa zdarzeń ma przyporządkowane numery od 1 do 127. Charakteryzuje się tym, że przewidywana częstotliwość pojawiania się zdarzenia jest na tyle mała, że odpowiedni licznik rejestrujący zdarzenie jest zbudowany na 1 słowie. Stąd maksymalna jego pojemność wynosi $2^{16}(177777_8)$ zdarzeń.

Druga grupa zdarzeń ma przyporządkowane numery od 128 do 192. Liczniki rejestrujące tę grupę zdarzeń zbudowano na 2 słowach, a maksymalna ich pojemność wynosi $2^{31}(3777777777_8)$ zdarzeń. Są one przystosowane do dużej częstości pojawiania się danego zdarzenia. W zależności od przewidywanej częstości występowania, każde zdarzenie zostaje umieszczone w pierwszej lub drugiej grupie numerów.

Istnieje jednak możliwość przekroczenia licznika na przykład na skutek wadliwej pracy sieci lub z powodu błędnego oszacowania częstości występowania danego zdarzenia. Brak rejestracji tego zjawiska mógłby doprowadzić do uzyskania fałszywych wyników pomiarów. Aby temu zapo-

biec przewidziano obszar przekroczeń OPK, którego wielkość ustalono wstępnie na 20 bajtów. Numer zdarzenia, którego licznik uległ przepełnieniu, zostaje wpisany na kolejne bajty obszaru przekroczeń. Przeglądając tablicę przekroczeń można łatwo odtworzyć rzeczywisty stan licznika, dodając do aktualnej wartości licznika 2^{16} .

Niektóre zdarzenia wymagają, oprócz zliczenia, dodatkowej rejestracji. Zdarzenia te, wywołując podprogram rejestracji, oprócz swego numeru identyfikacyjnego, podają drugi parametr związany ze zdarzeniem, który jest przedmiotem dodatkowej rejestracji. Przewidziano 3 zasadnicze typy dodatkowej obsługi omówione poniżej. W razie potrzeby z dużą łatwością można rozszerzyć system na inne typy obsługi.

1. Czasowy typ obsługi

Typ ten służy do rejestracji czasu trwania określonego połączenia logicznego w sieci lub czasu zajętości danego bufora. Parametrem dodatkowym jest czas pojawienia się zdarzenia. Czas końca zdarzenia jest chwilą wywołania podprogramu rejestracji z podaniem numeru identyfikacyjnego zdarzenia. W obsłudze tego typu, prócz normalnego licznika zdarzeń, rejestruje się na bieżąco średni czas trwania zdarzenia oraz maksymalny czas trwania zdarzenia.

2. Parametryczny typ obsługi

Typ ten służy do rejestracji określonego parametru związanego z danym zdarzeniem. Przykładem może być zdarzenie retransmisji, z którym jest związany parametr określający liczbę powtórzeń transmisji do chwili, gdy transmisja zakończy się powodzeniem. Parametr ten musi być przekazany do podprogramu rejestracji jako dodatkowy obok numeru identyfikacyjnego zdarzenia. W tym typie obsługi oprócz licznika zdarzeń rejestruje się średnią wartość parametru oraz jego maksymalną wartość.

3. Bitowy typ obsługi

Typ ten stosuje się wtedy, gdy pod jednym numerem identyfikacyjnym zdarzenia kryje się pewien podzbiór zdarzeń możliwych i koniecznych do odróżnienia, przy czym zliczanie zdarzeń w poszczególnych grupach nie jest konieczne. Każdy podzbiór zdarzeń ma przyporządkowany określony bit w słowie. Jeśli wystąpi przynajmniej jedno zdarzenie z danego podzbioru, to przyporządkowany mu bit przyjmuje wartość 1. Jeśli zdarzenie nie wystąpi, wówczas bit ma wartość 0.

Przykładem tego typu zdarzeń może być uszkodzenie linii. Parametrem dodatkowym przekazanym do podprogramu rejestracji, oprócz numeru identyfikacyjnego zdarzenia, jest numer uszkodzonej linii. Numer ten odpowiada odpowiedniemu numerowi bitu w słowie.

W niektórych typach obsługi jest przewidziana rejestracja wartości średniej parametru. Algorytm obliczania średniej, zastosowany w podsystemie rejestracji, ze względu na swoją specyfikę wymaga dodatkowego wyjaśnienia.

Podstawowym kryterium jakości podprogramu rejestracji jest, oprócz spełniania funkcji rejestracji, krótki czas działania. Natomiast typowy algorytm obliczania na bieżąco średniej wymaga wykonania kilku dzielen i mnożeń. Dzielenie w SM-3 musi być zrealizowane przez specjalny podprogram i wymaga dużo czasu. Zdecydowano się więc tak zmodyfikować algorytm, aby podczas rejestracji przy liczeniu średniej wykonywać tylko jedno dzielenie przez potęgę 2.

Dodatek A ilustruje działanie iteracyjnego algorytmu obliczania na bieżąco średniej bez udziału dzielenia: gdzie: $\bar{X}_j^n$ - j-ta średnia wyliczona dla $n=2^j$ składników.

Z powyższego algorytmu widać, że średnia jest obliczana dla liczby zdarzeń będącej potęgą dwójki. Jeśli licznik zdarzeń nie jest potęgą dwójki, to następuje tylko sumowanie parametrów od ostatniego zdarzenia będącego potęgą 2. Oczywiście w każdej chwili w razie potrzeby można odtworzyć dokładną wartość średniej, ponieważ odpowiednie argumenty są pamiętane (patrz baza danych). Podczas odtwarzania dokładnej wartości średniej naturalnie konieczne jest wykonanie dzielenia. Działanie to jednak w najgorszym razie jest wykonywane raz przez podprogram wyprowadzania, który nie ma tak ostrych uwarunkowań czasowych.

W systemie pomiarowym przyjęto założenie, że średnia wartość rejestrowanych parametrów jest pamiętana na bieżąco. Każdy z algorytmów obliczania średniej na bieżąco wymaga co najmniej jednej operacji dzielenia. Na przykład można obliczać średnią według poniższego wzoru:

$$Sr_k = Sr_{k-1} + \frac{a_k - Sr_{k-1}}{k} ,$$

gdzie: Sr_k - średnia z k składników,
 Sr_{k-1} - średnia z $k-1$ składników,
 a_k - k-ty składnik średniej.

Na podstawie liczby wykonywanych instrukcji (uwzględniając iteracje) oszacowano czas obliczania wartości średniej na podstawie powyższego wzoru oraz algorytmu aktualnie przyjętego w systemie. W pierwszym przypadku czas wykonania jest około 10-krotnie dłuższy. Narzucone na system ostre uwarunkowania czasowe nie pozwalają na przyjęcie tego rodzaju rozwiązania, nawet w razie znacznego zoptymalizowania programu dzielenia. Znaczne skrócenie czasu obliczeń okupione zostało tym, że nie dla każdego pomiaru otrzymuje się aktualną wartość średniej. Częstość obliczania średniej zmienia się od bardzo dużej dla niewielkiej liczby pomiarów do małej w miarę zwiększania się tej liczby. Średnia jest liczona dla następującego ciągu liczby pomiarów: 1,2,4,8,16,32,64 itd. Przy założeniu stabilności parametru rozrzut średniej powinien zbliżać się do 0 w miarę zwiększania się liczby pomiarów. O ile średnia z 2 pomia-

rów może się bardzo różnić od średniej z 4 pomiarów, o tyle średnia z 512 nie powinna wiele się różnić od średniej z 1024 pomiarów. Gdy parametr jednak wykazuje określony kierunek zmiany wartości, wtedy powyższe rozważania nie są prawdziwe. W przyszłości przewiduje się wprowadzenie dodatkowego parametru, którego celem byłoby wykazywanie kierunku zmian wartości mierzonej.

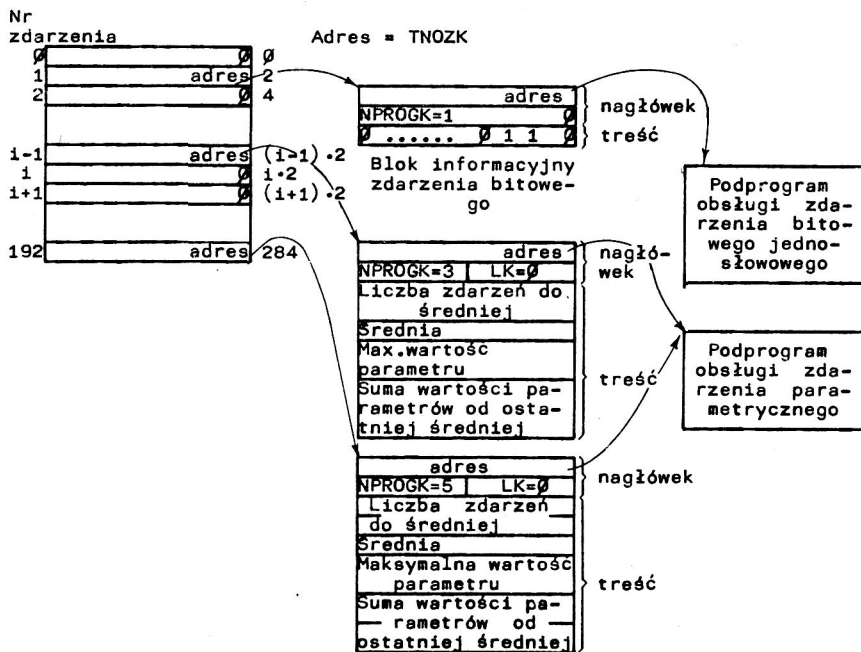
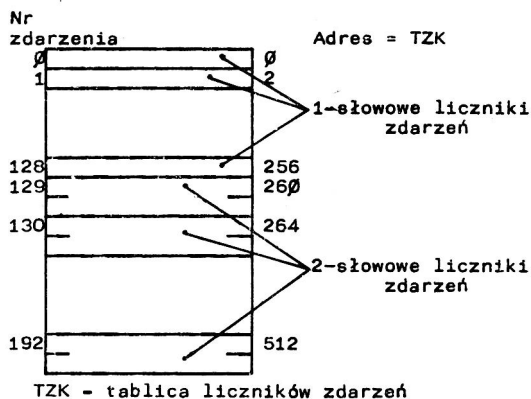
Obecnie - jeśli jest to konieczne - można uzyskać aktualną wartość średniej używając jednej z opcji wyprowadzania wyników (w formacie rozkazu wyprowadzania drugim parametrem komendy powinno być 6). W trakcie wstępnej eksploatacji przewiduje się analizę wartości średnich przez porównywanie wartości podawanej przez system według obecnego algorytmu oraz wartości liczonej dokładnie dla każdej liczby pomiarów. Analiza taka pozwoli na ostateczną ocenę proponowanego sposobu liczenia średniej dla każdego mierzonego parametru.

3. BAZA DANYCH DO REJESTRACJI ZDARZEŃ

Do realizacji przedstawionej koncepcji rejestracji zdarzeń konieczna jest odpowiednia struktura danych (rys. 1). Podstawowym obszarem w pamięci, zarezerwowanym w celu rejestracji, jest tablica liczników zdarzeń TZK indeksowana numerem zdarzenia. Do zliczania zdarzeń o numerach od 1 do 128 zarezerwowano jedno słowo, a do zliczania zdarzeń od numerów 129 do 192 - dwa słowa. Jest to 256-słowna tablica liczników wszystkich rejestrowanych zdarzeń. W przyszłości, w razie potrzeby, tablica ta może zostać powiększona. Zliczanie zdarzeń prowadzone jest w kodzie binarnym.

Pewna liczba zdarzeń, oprócz zliczania, wymaga rejestracji określonych parametrów dodatkowych. Rozpoznawanie tego typu zdarzeń możliwe jest za pomocą tablicy TNOZK indeksowanej numerem zdarzenia. Dla każdego zdarzenia jest zarezerwowane w tej tablicy jedno słowo. Jeśli zdarzenie nie wymaga dodatkowej obsługi, to w słowie tym jest pamiętane $\emptyset$. W przeciwnym razie w słowie tym jest pamiętany adres początku bloku informacyjnego zdarzenia.

Dla każdego zdarzenia wymagającego dodatkowej rejestracji w dowolnym miejscu pamięci jest umieszczony blok informacyjny zdarzenia. Blok ten składa się z nagłówka i z treści. Nagłówek zajmuje dwa słowa. Pierwsze słowo zawiera adres podprogramu obsługi dodatkowej danego zdarzenia. W trzecim bajcie nagłówka jest pamiętany typ danego zdarzenia NPROGK. Typ ten jednoznacznie określa rodzaj i wielkość parametrów pamiętanych w treści bloku informacyjnego. Czwarty bajt nagłówka o nazwie LK jest zmienną binarną, która blokuje program przetwarzający dany



Tablica TNOZK wskazująca adresy bloków informacyjnych zdarzeń wymagających dodatkowej obsługi oraz przykłady bloków informacyjnych.

Adres = OPK

142	0
54	1
0	2
0	
...	
0	256

Obszar przekroczeń, w którym kolejno wpisany został 142 i 54 numer zdarzenia. Oznacza to, że w tablicy TZK liczniki zdarzeń 142 i 54 zostały przekroczone.

Rys.1. Baza danych do rejestracji zdarzeń

Fig.1. Data base for event registration

blok informacyjny z powodu np. przekroczenia jednego z parametrów. W razie braku takiej blokady mogłoby dojść do nieśensownych wyników, np. z powodu zerowej liczby zdarzeń. Treść bloku informacyjnego zawiera parametry, których liczba, a także liczba słów zarezerwowanych dla każdego pamiętanego parametru jest zmienna i należy do typu bloku informacyjnego (jednoznacznie określa go parametr nagłówek NPROGK).

Dla NPROGK = 1 występuje blok informacyjny o bitowym typie obsługi o treści składającej się z jednego parametru 1-słowego, w którym każdy z 16 bitów ma własną interpretację zależną od zdarzenia NPROGK=2.

Blok informacyjny o bitowym typie obsługi o treści składającej się z jednego parametru 2-słowego, w którym każdy z 32 bitów ma własną interpretację zależną od zdarzenia.

Dla NPROGK = 3,4,5,6 utworzone są bloki informacyjne o czasowym lub parametrycznym typie obsługi, których treść składa się z 4 parametrów:

- liczba zdarzeń do obliczania średniej,
- średnia,
- maksymalna wartość,
- suma składników nie wziętych do obliczania średniej (patrz algorytm obliczania średniej).

Bloki te różnią się między sobą liczbą słów rezerwowanych dla każdego parametru. Zależne jest to od przewidywanej liczności danego zdarzenia, a także od przewidywanej wielkości parametru, dla którego liczymy wartość średnią i maksymalną. Wartość średnia jest pamiętana zawsze na dwu słowach, z których młodsze jest wartością ułamkową średniej, a starsze - częścią całkowitą. Dla poszczególnych typów bloków informacyjnych podano w nawiasach liczbę słów przewidzianych dla poszczególnych parametrów w kolejności, w której zostały wymienione powyżej:

NPROGK = 3 (1,2,1,2)

NPROGK = 4 (2,2,1,2)

NPROGK = 5 (1,2,2,3)

NPROGK = 6 (2,2,2,3)

Bloki informacyjne zdarzeń mogą być dowolnie definiowane w zależności od konkretnych potrzeb. Liczba ich typów jest ograniczona zakresem zmiennej NPROGK, czyli może być rozszerzona do $2^8(256_{10})$.

4. REJESTRACJA STANÓW (FOTOGRAFIA)

Ta grupa pomiarów w przeciwieństwie do rejestracji zdarzeń nie jest realizowana w trybie permanentnym. Pomiary te są przeprowadzone cyklicznie i inicjowane przez program zegara. Pod względem merytorycznym pomiary te można podzielić na 2 rodzaje.

Pierwszy rodzaj rejestrowanych pomiarów dotyczy stanu węzła w chwili wykonywania pomiaru. Informacje te mają charakter bieżący i główne zastosowanie w diagnostyce bieżącej.

Drugi rodzaj rejestrowanych pomiarów ma charakter uśrednianych statystyk. Typowym przykładem tego typu pomiarów jest liczba wolnych buforów, na podstawie której, analogicznie jak w razie rejestracji niektórych zdarzeń, oblicza się na bieżąco średnią oraz maksymalną liczbę wolnych buforów. Parametry te można by również uzyskać przez rejestrację zdarzeń, przyjmując jako zdarzenie pobranie lub oddanie bufora. Przyjęto jednak, że w stosunku do pewnych parametrów pomiar okresowy będzie szybszy, a dokładność zostanie zachowana.

Ta grupa pomiarów opierać się będzie przede wszystkim na wspólnych danych programowych, które zostaną poddane pewnemu przetwarzaniu i przepisane do obszaru rejestracji. W przeciwieństwie więc do rejestracji zdarzeń proces pomiarów nie stanowi nawet w małej części integralnej składowej bloków programowych, składających się na funkcjonalne oprogramowanie węzła. Dlatego też realizacja tego typu pomiarów będzie realizowana później. Tak więc szczegółowy wykaz rejestrowanych stanów, jak i baza danych podprogramu rejestracji oraz typy przetwarzania pomiarów, zostaną podane po realizacji tej części systemu miernictwa.

Przedstawiono jeden z istotnych elementów systemu pomiarowego podsieci Pitagoras. Dotyczy on zbierania i rejestracji danych w węźle podsieci. Dane te są bazą, na której opiera się działanie pozostałych elementów składających się na całość systemu pomiarowego, tzn.:

- sposoby i cele monitorowania pomiarów,
- przetwarzanie pomiarów,
- transfer informacji między węzłami (pakiety pomiarowe),
- centralne zarządzanie pomiarami w podsieci (centrum pomiarowe).

Praca wpłynęła do Redakcji 12.12.1980 r.

Po poprawieniu 03.03.1982 r.

Dodatek A

Iteracyjny algorytm obliczania średniej

$$\bar{x}_1^2 = \frac{x_1 + x_2}{2} = \frac{\sum_{i=1}^2 x_i}{2} ,$$

$$\bar{x}_2^4 = \frac{x_1 + x_2 + x_3 + x_4}{4} = \frac{\bar{x}_1^2}{2} + \frac{\sum_{i=3}^4 x_i}{4} ,$$

$$\bar{x}_3^8 = \frac{(x_1 + x_2 + x_3 + x_4) + x_5 + x_6 + x_7 + x_8}{8} = \frac{\bar{x}_2^4}{2} + \frac{\sum_{i=5}^8 x_i}{8},$$

•
•

$$\bar{x}_j^n = \frac{\bar{x}_{j-1}^{n/2}}{2} + \frac{\sum_{i=\frac{n}{2}+1}^n x_i}{n} \quad \text{dla } n = 2^j.$$

REGISTRATION FOR THE MEASUREMENT DATA IN THE MSK COMPUTER NETWORK

In the paper a method of events registration in the node is described. There are in the node a database and diagnostic control system for measurement and statistical purposes.

Verified by Ruta Czaplińska

РЕГИСТРАЦИЯ ИЗМЕРИТЕЛЬНЫХ ДАННЫХ В СЕТИ ЭВМ MSK

Рассматривается концепция регистрации событий в измерительном узле, составляющем измерительную базу, использующую цели статистики управления сети, диагностики и т. д. Описывается база данных, необходима для регистрации событий.

Проверила Малгожата Хейдрих

*Sieć komputerowa,
rejestracja informacji*

Elżbieta HUDYMA^{*)}, Elżbieta KOSMULSKA-BOCHENEK^{*)}

Edycja rejestrowanych pomiarów w sieci komputerowej MSK

Przedstawiono sposoby wyprowadzania informacji diagnostyczno-pomiarowych zgromadzonej w pamięci w sieci komputerowej MSK. Dane są wyprowadzane w postaci dwóch głównych typów raportów: cyklicznego i standardowego. Ten ostatni daje kilka możliwości informowania operatora o aktualnym stanie sieci.

1. WSTĘP

Przydatność zgromadzonych w węźle danych pomiarowych zależy w dużym stopniu od sposobu przekazania ich obsłudze systemu. Podstawowymi kryteriami dobrego monitorowania, oprócz jakości wyprowadzanej informacji, są: czytelność, przejrzystość i jednoznaczność. Czytelność zapewnia dziesiętna postać liczb, przejrzystość uzyskuje się przez tabelaryzację danych, jednoznaczność natomiast można uzyskać uzupełniając dane liczbowe opisami słownymi. Takie ostre kryteria organizacji wydruku wymagają zarówno dużego obszaru pamięci, jak i długiego czasu przetwarzania, co wobec skromnych możliwości hardwarowych i softwarowych maszyny obsługującej węzeł w sieci komputerowej MSK jest trudne do przyjęcia.

Jednocześnie wiadomo, że wymagania stawiane monitorowaniu zarejestrowanych informacji są uzależnione od kwalifikacji osoby, dla której są przeznaczone. Przyjęto, że odbiorcą informacji ma być inżynier, co znacznie łagodzi wymienione kryteria. Zrezygnowano więc z dziesiętnej

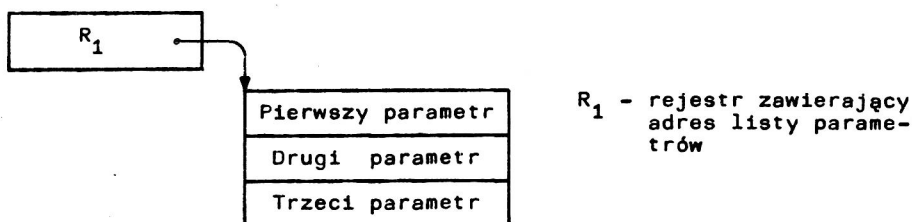
^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

postaci wyników, zastępując ją formą ósemkową oraz z wszelkich opisów słownych, zachowując przy tym sztywny format wydruków.

Prawidłowej oceny systemu można dokonać na podstawie analizy aktualnych danych oraz historii systemu. Wyłania się też drugi problem związany z monitorowaniem: dostępność i składowanie danych. Przyjęto dwa podstawowe typy raportów: raport bieżący (na żądanie) oraz raport cykliczny (standardowy). Zadaniem pierwszego jest natychmiastowe przekazanie aktualnych danych dotyczących zdarzeń wyselekcjonowanych przez operatora. Raport cykliczny ma na celu systematyczne przekazywanie wszystkich informacji okresowo zbieranych na urządzenie pamięci (np. dysk) w celu dalszego przetwarzania. Raporty te szczegółowo omówiono w punkcie 3 i 4.

2. ZAŁOŻENIA EDYCJI

Monitorowana informacja jest sterowana komendą operatora, która bazuje na bloku słów zawierającym odpowiednie parametry. Obecnie są ustalone trzy parametry, a w przyszłości lista ta może być rozszerzona (rys.).



Rys. Struktura komendy operatorskiej

Fig. Structure of operators command

Pierwszy parametr komendy określa zadanie do zrealizowania. Ustalono 5 zadań, którym odpowiadają numery 1-5:

- 1 - zmiana urządzenia wyjściowego (standardowo urządzeniem tym jest monitor),
- 2 - wyprowadzenie raportu bieżącego, dającego opis jednego zdarzenia lub grupy zdarzeń,
- 3 - wyprowadzenie raportu bieżącego zawierającego wszystkie niezerowe zdarzenia poczynwszy od zadanego,
- 4 - wyprowadzenie raportu standardowego,
- 5 - wprowadzenie daty i czasu rozpoczęcia pomiarów.

Drugi parametr komendy określa początkowy numer zdarzenia, dla którego są żądane informacje.

Parametr trzeci określa liczbę zdarzeń do wyprowadzania. Gdy parametr nie jest podany, wydrukowane zostanie tylko jedno zdarzenie.

3. RAPORT STANDARDOWY

Raport standardowy można uzyskać dwoma sposobami: jako raport cykliczny uruchamiany zegarem lub przez komendę operatora.

Treścią raportu standardowego są wszystkie rejestrowane informacje. Informacje te pochodzą z dwóch źródeł. Jedno źródło, to rejestracja wytypowanych zdarzeń, a drugie - to rejestracja stanu węzła w zadanej chwili, czyli tzw. fotografia węzła. Szczegółowy opis rejestrowanych zdarzeń zawiera praca [1]. Na obecnym etapie prac jest zrealizowana tylko rejestracja zdarzeń, natomiast dane z fotografii zostaną dołączone w terminie późniejszym.

Ze względu na ograniczoną pojemność bufora wyjściowego (256 słów) całość raportu została podzielona na trzy części (edycje). Każda edycja zaczyna się 10-słowym nagłówkiem, na który składa się data, czas oraz numer edycji (1, 2 lub 3). Pozostały obszar jest przeznaczony na dane z rejestracji. Podstawową wielkością mierzoną są liczniki zdarzeń, które mogą być jednosłowe lub dwusłowe. Niektóre zdarzenia, oprócz notowania liczby wystąpienia, wymagały jeszcze dodatkowych informacji. Informacje te są zapisane w blokach słów związanych z danym zdarzeniem, zwanymi blokami informacyjnymi zdarzeń.

Na pierwszą edycję składają się dwie grupy danych:

1. 128 liczników jednosłowych odpowiadających zdarzeniom o numerach od 0 do 127.
2. Bloki informacyjne zdarzeń o rozszerzonym przetwarzaniu z zakresu od 0 do 127.

Druga edycja zawiera trzy grupy danych:

1. 64 liczniki dwusłowe dla numerów zdarzeń 128-191 (razem 128 słów).
2. Bloki informacyjne zdarzeń o rozszerzonym przetwarzaniu z zakresu od 128 do 191.
3. 20-bajtowy obszar przekroczeń, gdzie są spisywane numery zdarzeń, dla których liczniki zostały przepełnione.

Na trzecią edycję składają się dane z fotografii węzła.

Po skompletowaniu jednej edycji, zawartość bufora zostaje wyprowadzana na dysk lub dodatkowo na inne przydzielone urządzenie.

Przeznaczeniem raportu cyklicznego jest przechowanie zarejestrowanej informacji w celu późniejszego jej przetworzenia. Dlatego też składowane dane na dysku są zachowane w postaci binarnej. Aczkolwiek jest

przewidziane wyprowadzenie raportu standardowego na monitor lub drukarkę (pierwszy parametr komendy operatora = 4), korzystanie bezpośrednie z tych danych wymaga znajomości kodu wewnętrznego maszyny.

4. RAPORT BIEŻĄCY

4.1. Treść raportu

Zadaniem raportu bieżącego jest wyprowadzenie na żądanie operatora selektywnej informacji w czytelnej postaci. Dla pierwszego parametru komendy operatora = 2 może to być wyprowadzenie licznika (oraz dodatkowo bloku informacyjnego zdarzenie, jeśli zdarzenie takie zawiera) dla jednego zdarzenia, którego numer podano w drugim słowie komendy operatora (trzecie słowo niespecyfikowane) lub wyprowadzenie informacji dotyczących kolejnych zdarzeń, gdzie początkowy numer zdarzenia podano w drugim słowie komendy operatora, natomiast trzecie słowo określa liczbę zdarzeń do wyprowadzenia.

Gdy pierwszy parametr przesłany od operatora wynosi 3, wówczas są wyprowadzone informacje dla kolejnych zdarzeń, począwszy od zdarzenia, którego numer jest określony drugim parametrem komendy operatorskiej. Koniec listowania jest warunkowany końcem strony urządzenia, np. monitor pozwala na wyświetlenie 15 wierszy 80-znakowych. Dane są wyprowadzane w kodzie ASCII.

W celu przekazania maksymalnej informacji we wszystkich przypadkach bieżącego raportu drukuje się tylko dane o zdarzeniach, które wystąpiły. Jeżeli licznik zdarzenia ma wartość zero, wyszukiwane jest zdarzenie o numerze następnym. Wykonanie żadanego zadania jest sygnalizowane symbolem końca wydruku o postaci: cr, lf, *.

4.2. Organizacja wydruku raportu bieżącego

Zarówno ograniczona wielkość bufora wyjściowego, jak i przypadkowość numeracji zdarzeń uniemożliwiały tabulację wyników oraz wprowadzenie opisów słownych.

Przyjęto zasadę sekwencyjnego wyprowadzania par: numer zdarzenia, licznik. Separatorem kolejnych par jest podwójna liczba spacji w odniesieniu do liczby spacji wewnątrz pary. Prawidłowość tę zaburzają zdarzenia o rozszerzonym przetwarzaniu. Dla tych zdarzeń od nowego wiersza jest wyprowadzany numer zdarzenia, a po nim blok informacyjny dla tego zdarzenia i następuje zmiana wiersza.

Binarna postać danych jest tłumaczona na kod ASCII, przy czym każdorazowo jest kompletowany jeden wiersz wydruku (maksymalnie 80 znaków) i przekazany na wyjście.

5. REALIZACJA KOMEND O PARAMETRACH 1 i 5

Jeżeli pierwszy parametr komendy operatora ma wartość 1, należy zmienić urządzenie wyjściowe na nowe, którego numer jest określony w drugim słowie komendy. Realizacja komendy polega na wypełnieniu nową wartością słowa Typ/Tryb [2] pola sterującego urządzeniem. Daje ona możliwość utrwalenia danych przez wykorzystanie drukarki lub chwilowe przeglądanie informacji na monitorze.

Odczyt czasu rozpoczęcia pomiarów (pierwszy parametr komendy = 5) jest bardzo istotny w ocenie ilościowej w czasie mierzonych parametrów.

6. UWAGI KOŃCOWE

Konstrukcja opracowanego programu wyprowadzania wyników pomiaru pozwala na rozszerzenie jego możliwości w kierunku zwiększonej ilości monitorowanej informacji oraz w kierunku przyjmowania i analizy dalszych komend od operatora. Przewiduje się edycję przetworzonej informacji składowanej na dysku w postaci raportów całłościowych oraz selektywnych, sterowanych różnymi parametrami.

LITERATURA

- [1] H u d y m a E., K o s m u l s k a - B o c h e n e k E., Koncepcja systemu pomiarowego w sieci komputerowej, Raporty Centrum Obliczeniowego PWr., Seria SPR nr 4/80, Wrocław 1980.
- [2] L e w o c J., Ł a n o w s k a B., S t a n i s z A., S z w a r c M., Zasady oprogramowania węzła, Raporty Centrum Obliczeniowego PWr., Seria SPR nr 14/80, Wrocław 1980.

Praca wpłynęła do Redakcji 12.12.1980 r.

Po poprawieniu 3.03.1982 r.

EDITION OF THE MEASUREMENT DATA IN THE MSK COMPUTER
NETWORK

The a way for obtaining the reports of the collected measurement and diagnostic data is given. The report may be generated as a cyclic one or for an operator demand. Using the report, the operator has a possibility to get an actual data regarding the state of the network.

Verified by Ruta Czaplińska

РЕДАКТИРОВАНИЕ ИЗМЕРЕНИЙ В СЕТИ ЭЕМ MSK

В статье представлены методы выдачи собранных в памяти диагностически-измерительных данных о работе сети ЭЕМ MSK. Данные можно получить в виде двух главных рапортов: циклического и стандартного. Стандартный рапорт позволяет в некоторых случаях информировать оператора относительно статуса сети.

Проверила Малгожата Хейдрих

*Pakiet diagnostyczno-pomiarowy,
protokół sieciowy*

Elżbieta HUDYMA^{*)}, Elżbieta KOSMULSKA-BOCHENEK^{*)}

Pakiety miernictwa w sieci komputerowej MSK

Przedstawiono koncepcję przesyłania informacji pomiędzy centrum pomiarowym i poszczególnymi węzłami. Nośnikami tych informacji są specjalne pakiety pomiarowe wysyłane z centrum pomiarowego, odpowiednio obsługiwane w węzłach i odesyłane z powrotem do centrum pomiarowego. Omówiono ogólny format pakietu pomiarowego oraz szczegółowo przedstawiono kilka typów takich pakietów różniących się zbiorem przenoszonych informacji.

1. WSTĘP

Zarówno w fazie uruchamiania podsieci, jak i podczas normalnej pracy sieci, bardzo przydatne są pakiety pomiarowo-diagnostyczne. Za pomocą takich pakietów można stwierdzić czy sieć jest sprawna oraz określić ewentualne przyczyny niesprawności sieci. Dobierając odpowiednie typy pakietów można uzyskać w miarę dokładny obraz pracy sieci.

Pakiety diagnostyczne są generowane przez centrum pomiarowe (CP), przy czym takie centrum może być zrealizowane w dowolnym z węzłów podsieci komunikacyjnej. Zadaniem centrum pomiarowego jest wysłanie odpowiedniego pakietu do odbiorcy, odebranie odpowiedzi oraz przetworzenie informacji zawartej w pakiecie.

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

2. POSTAĆ PAKIETU

Pakiety pomiarowe od pakietów użytkowych krążących w sieci różnią się kodem GFI. Każdy pakiet, mający kod GFI różny od standardowych, jest kierowany do podzadania miernictwa, gdzie następuje najpierw badanie czy prawidłowy jest nagłówek pakietu, a następnie, jeżeli pakiet zostaje zakwalifikowany jako pakiet miernictwa, odszyfrowuje się typ tego pakietu. Za prawidłowy uważa się nagłówek zawierający na polu GFI kod równy czterem jedynkom (GFI = 1111). Typ pakietu jest zapisany na drugiej części pierwszego bajtu nagłówka i jest zawarty pomiędzy 0 i 15. Kod typu pakietu równy 0000 identyfikowany jest jako błąd. Nagłówek pakietu miernictwa przedstawiono na rys. 1. Zawiera on oprócz identyfikacji i typu pakietu adresy nadawcy i odbiorcy, czas wysłania pakietu oraz znacznik odbicia. Znacznik odbicia jest binarną informacją zapisaną na bajcie sygnalizującą, że pakiet dotarł do odbiorcy.

GFIK	TYP
REZERWA	
Adres węzła docelowego	
Adres węzła nadającego	
Czas wysłania pakietu	
Znacznik odbicia	

Rys.1. Nagłówek pakietu miernictwa i diagnostyki
Fig.1. Head of measurement and diagnostic packet

3. TYPY I FUNKCJE PAKIETÓW POMIAROWYCH

Informacje do pakietów pochodzą z pól diagnostycznych, gdzie są wpisywane rejestrowane okresowo liczniki zdarzeń oraz odpowiednio przetworzone dane, charakteryzujące stany [1]. Wyróżnia się 7 typów pakietów pomiarowych.

3.1. Pakiet zbierania danych ogólnych

Pakiet wysyłany jest z centrum pomiarowego do wybranego węzła, gdzie są wpisywane odpowiednie dane i pakiet zostaje skierowany do nadawcy. Dane te są brane z pól diagnostycznych węzła i dotyczą przede wszystkim wykorzystania buforów, linii oraz ważniejszych informacji czasowych o podsięci, a zwłaszcza:

- średniego czasu pobytu pakietu w węźle,
- maksymalnego czasu pobytu pakietu w węźle,

- średniego czasu trwania połączenia wirtualnego,
- maksymalnego czasu połączenia wirtualnego,
- średniej liczby zajętych buforów,
- maksymalnej liczby zajętych buforów,
- średniej liczby założonych połączeń logicznych,
- maksymalnej liczby założonych połączeń logicznych,
- średniej długości ramki informacyjnej,
- średniej liczby retransmisji pakietów zakończonych z pozytywnym skutkiem,
- maksymalnej liczby retransmisji pakietów zakończonych z pozytywnym skutkiem,
- średniej liczby retransmisji ramek zakończonych z pozytywnym skutkiem,
- maksymalnej liczby retransmisji ramek z pozytywnym skutkiem,
- wykazu aktualnie pracujących linii węzła,
- wykazu linii węzła,
- liczby pakietów informacyjnych odebranych w węźle,
- liczby ramek odebranych w węźle,
- liczby ramek informacyjnych odebranych w węźle.

3.2. Pakiet sonda

Zadaniem tego pakietu jest wysłanie pakietu powrotnego z każdego węzła drogi do centrum pomiarowego. Pakiet powrotny zawiera numer węzła, z którego jest wysłany, numer linii wyjściowej oraz numer węzła docelowego. Centrum pomiarowe notuje czas przybycia pakietu z każdego węzła i sprawdza znacznik odbicia.

Pakiet sonda umożliwia kontrolę drogi od węzła pierwotnego do docelowego. Analizując czas przybycia pakietu powrotnego można ocenić opóźnienie jakie daje kolejny węzeł.

3.3. Pakiet echo

Jest to typowy pakiet dla każdej sieci komputerowej. Wysłany z centrum pomiarowego do określonego węzła docelowego powoduje odwrócenie adresu, ustawienie znacznika odbicia i przesłanie pakietu z powrotem do nadawcy. Centrum pomiarowe notuje czas wysłania pakietu oraz czas przybycia pakietu powrotnego. Pakiet echo może być pomocny w wykrywaniu niedrożności sieci. Gdy brak odpowiedzi po upływie założonego limitu czasu, należy sieć uznać za niesprawną.

3.4. Pakiet rejestracji drogi

Pakiet rejestracji drogi daje obraz drogi jaką przebył pakiet od węzła nadawcy (centrum pomiarowe) do węzła docelowego. Kolejne węzły pośredniczące wpisują do pakietu swoje numery oraz numery linii wejściowych i wyjściowych. Węzeł docelowy odwraca adresy i odsyła pakiet do nadawcy. W centrum pomiarowym notowany jest też czas wysłania i powrotu pakietu.

3.5. Pakiet kontroli drogi wymuszonej

Droga pakietu jest dokładnie ustalona przez CP i z każdego kolejnego węzła drogi są wysyłane do CP pakiety powrotne. Funkcje pakietu są podobne jak dla pakietu sonda z tym, że droga jest wymuszona. Pakiet powrotny zawiera numer węzła i numer linii wyjściowej.

3.6. Pakiet zbierania danych X.25

Treścią pakietu są dane charakteryzujące protokół X.25. Wielostronna analiza tych danych pozwala na ocenę efektywności pracy sieci na tym poziomie. Informacje zbierane przez węzeł:

- liczba wszystkich pakietów odebranych przez węzeł,
- liczba pakietów informacyjnych odebranych przez węzeł,
- liczba pakietów z błędnym nagłówkiem,
- maksymalna liczba retransmisji pakietów zakończonych z pozytywnym skutkiem,
- średnia liczba retransmisji zakończonych z pozytywnym skutkiem,
- liczba zgubionych pakietów,
- liczba pakietów, które nie dotarły do odbiorcy,
- liczba zdarzeń polegających na przeciążeniu węzła,
- liczba zdarzeń polegających na braku wolnych kanałów logicznych na wejściu,
- liczba kolizji wywołań,
- średnia liczba retransmisji pakietów zależna od limitu czasowego,
- liczba pakietów o niepoprawnym formacie.

3.7. Pakiet zbierania danych poziomu HDLC

Z określonych pól diagnostycznych są wpisywane dane charakteryzujące protokół HDLC, zwłaszcza:

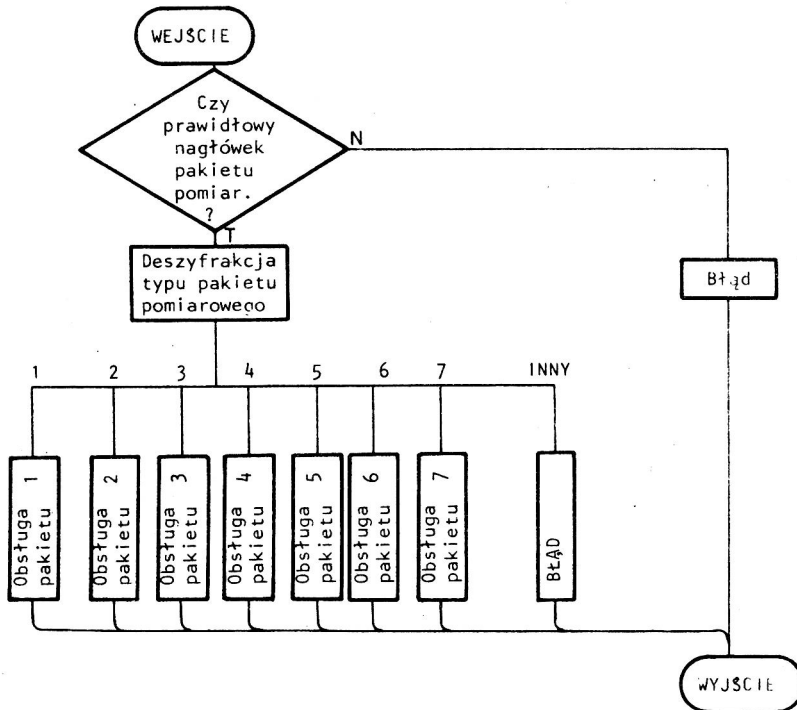
- wykaz linii węzła,
- wykaz aktualnie sprawnych linii węzła,

- liczba wszystkich ramek odebranych przez węzeł,
- liczba ramek informacyjnych odebranych przez węzeł,
- średnia i maksymalna liczba retransmisji ramek sterujących,
- średnia i maksymalna liczba retransmisji ramek informacyjnych spowodowanych odrzuceniem ramki,
- średnia i maksymalna liczba retransmisji ramek informacyjnych zależna od limitu czasowego,
- liczba wysłanych ramek FRMR w wyniku przekroczenia długości ramki informacyjnej,
- liczba wysłanych ramek FRMR w wyniku odbioru ramki nieimplementowanej lub za krótkiej,
- liczba wysłanych ramek FRMR w wyniku odbioru ramki z niedopuszczalnym polem informacyjnym,
- średnia liczba ramek nie potwierdzonych,
- liczba wysłanych ramek niegotowości odbioru.

4. STEROWANIE PAKIETAMI DIAGNOSTYCZNO-POMIAROWYMI

Oprogramowanie dodatkowe węzła pełniącego funkcję centrum pomiarowego musi uwzględniać obsługę pakietów. Program obsługi pakietów składa się z dwóch podstawowych części. Pierwsza - sterująca jest wspólna dla wszystkich pakietów, natomiast w skład drugiej wchodzi procedura obsługi kolejnych pakietów. Na sterowanie pakietami składa się analiza nagłówka pakietu oraz deszyfracja typu pakietu (rys. 2). Zadaniem programów obsługujących pakiety jest analiza kolejnych pól pakietu i odpowiednie ich przetworzenie. Efektem przetwarzania pakietów zbierających dane (ogólne, X.25, HDLC) są zbiory statystyk dających miarodajne informacje na temat pracy podsieci. Przetworzenie innych pakietów (echo, sonda, kontrola drogi automatycznej i wymuszonej) polega na analizie pól odpowiedzi wysłanych pakietów, z czego można zorientować się o stanie węzłów sieci.

Oprócz oprogramowania specjalistycznego centrum pomiarowego każdy węzeł sieci musi być wyposażony w dodatkowe oprogramowanie, umożliwiające obsługę pakietu pomiarowego. Obsługa ta polega na przyjęciu pakietu, wykonaniu czynności wynikających z żądań pakietu (zapełnienie pól pomiarowych, ustawienie znacznika odbicia itd.) oraz odesłaniu go do centrum pomiarowego lub innego węzła.



Rys.2. Uproszczony schemat obsługi pakietów diagnostyczno-pomiarowych
 Fig.2. Simplified diagram of service of the diagnostic-measurement packets

LITERATURA

- [1] Hudyma E., Kosmulska-Bochenek E., Rejestracja pomiarów w sieci komputerowej MSK (w tym samym zeszycie).

Praca wpłynęła do Redakcji 12.12.1980 r.

Po poprawieniu 3.03.1982 r.

MEASUREMENT PACKETS IN THE MSK COMPUTER NETWORK

The paper presents a method of swapping diagnostic data between the separate nodes of the network and the Measurement Center. The data are recorded on the special measurement packets, which are controlled by Measurement Center. Measurement Center sends a special packet to the rece-

iving node, acknowledges the receipt and processes the packet information. The types of the measurement packets are described in detail.

Verified by Ruta Czaplińska

ИЗМЕРИТЕЛЬНЫЕ ПАКЕТЫ В СЕТИ ЭВМ MSK

Рассмотрена концепция обмена информацией, собираемой во время регистрации событий между отдельными узлами сети и Измерительным центром. Данные записываются на специальных измерительных пакетах, посылаемых в Измерительный центр. Описан общий формат измерительного пакета. Детально рассмотрены несколько видов таких пакетов.

Проверила Малгожата Хейдрых

*Sieć komputerowa, podsieć
komunikacyjna, oprogramowanie, wybór drogi*

Józef LEWOC*)

Metoda wyboru drogi w sieci komputerowej MSK

Sposób wyboru drogi ma znaczny wpływ na charakterystyki użytkowe sieci z komutacją pakietów. Przedstawiono metody wyboru drogi w sieciach komputerowych oraz przesłanki uzasadniające zastosowanie metody dróg ustalonych z drogami alternatywnymi. Opisano implementację zastosowanej metody, podając pewne informacje szczegółowe o rozwiązaniach programowych zastosowanych w węźle sieci MSK.

1. WSTĘP

W sieciach komputerowych z komutacją pakietów, do jakich zalicza się Międzyuczelniana Sieć Komputerowa (MSK), sposób wyboru drogi dla przesyłanych pakietów ma zasadniczy wpływ na podstawowe charakterystyki użytkowe, jakimi są średni i maksymalny czas przesyłania pakietów od nadawcy do adresata. Sposób ten ma także wpływ na globalną przepustowość sieci, gdyż określa liczbę łączy wykorzystywanych w przesyłaniu pakietów.

W sieciach komputerowych spełniających zalecenia X.25 CCICT [1] (a taką ma być sieć MSK) przesyłane pakiety nie zawierają informacji podających drogi, jakie mają one przebywać. W związku z tym w każdym węźle podsieci komunikacyjnej, stanowiącej autonomiczną część sieci komputerowej spełniającej zalecenia X.25, należy podejmować decyzje, do jakich łączy przesyłać odebrane pakiety. Takie zdecentralizowane meto-

*) Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

dy wyboru drogi są stosowane w większości znanych dużych sieci komputerowych [2,3].

Przyjmując metodę wyboru drogi dla sieci komputerowej MSK należy uwzględnić ograniczenia, jakie będą występować podczas realizacji tej sieci.

Po pierwsze, należy się spodziewać stosunkowo małej niezawodności, sprzętu w porównaniu ze znanymi sieciami. Wynika to stąd, że część urządzeń komunikacyjnych, np. adaptery liniowe, będzie miała charakter prototypowy, a doświadczenia uzyskane podczas eksploatacji standardowego sprzętu przeznaczonego do pracy w sieci MSK również wskazują na gorsze charakterystyki niezawodnościowe w porównaniu ze sprzętem wykorzystywanym w innych sieciach.

Po drugie, przepustowość łączy telekomunikacyjnych będzie czynnikiem znacznie bardziej ograniczającym niż w innych znanych sieciach. Modemy, jakie będą dostępne do wykorzystywania w pierwszej wersji sieci MSK, będą pracowały z szybkością - w najlepszym razie - 2400 b/s lub 4800 b/s w porównaniu z szybkościami transmisji międzywęzłowej rzędu co najmniej kilkudziesięciu tysięcy bitów na sekundę, jakie są stosowane w znanych sieciach.

Po trzecie, sieć MSK będzie stopniowo rozbudowywana od prostej, trójwęzłowej konfiguracji początkowej do złożonej sieci łączącej co najmniej wszystkie ważniejsze ośrodki akademickie w kraju. Zatem metoda wyboru drogi powinna dać się łatwo adaptować do nowych topologii sieci.

Po czwarte, system pomiarowy dla sieci MSK będzie tworzony z pewnym opóźnieniem w stosunku do pierwszej wersji tej sieci. Wynika to z braku doświadczeń praktycznych dla sieci tworzonych przy podobnych ograniczeniach sprzętowych, a także z ograniczonych zasobów (zwłaszcza pamięci operacyjnej) minikomputerów SM-3, na jakich mają być zbudowane pierwsze węzły sieci MSK.

Zatem w pierwszym etapie tworzenia sieci MSK nie będą dostępne, jeżeli w ogóle będą określone charakterystyki, które są potrzebne do rozsądnej optymalizacji przepływu pakietów. Określenie i pomiary tych charakterystyk staną się możliwe po pewnym okresie eksploatacji i badań sieci MSK. Zatem sposób wyboru drogi powinien dać się stosunkowo łatwo adaptować do nowej sytuacji, w której możliwa i celowa będzie dynamiczna (tzn. uwzględniająca zmiany istotnych charakterystyk sieci) optymalizacja przepływu pakietów.

W dalszej części niniejszego artykułu zostaną omówione znane metody wyboru drogi ze szczególnym uwzględnieniem metody przyjętej dla sieci MSK.

2. METODY WYBORU DROGI STOSOWANE W SIECIACH KOMPUTEROWYCH

Metody wyboru drogi przesyłania pakietów można podzielić ogólnie na metody statyczne i metody dynamiczne [4]. W metodach statycznych drogi przesyłania pakietów są określone z góry podczas tworzenia poszczególnych węzłów sieci. Natomiast metody dynamiczne uwzględniają zachodzące w czasie zmiany konfiguracji i charakterystyk sieci. Droga dla pakietu jest wyznaczana na podstawie pewnej oceny aktualnego (chwilowego) stanu sieci.

Do metod statycznych należą:

- metoda rozptylowa,
- metoda losowa,
- metoda dróg ustalonych,
- statyczne metody optymalne.

Wśród metod dynamicznych wyróżnia się metody następujące:

- metody heurystyczne,
- metody adaptacyjne,
- metodę hierarchiczną,
- dynamiczne metody optymalne.

2.1. Metoda rozptylowa

Metoda rozptylowa, nazywana też metodą zatrzymania binarnego [4] polega na tym, że węzeł otrzymujący pakiet przesyła go do wszystkich jego sąsiadów. Pakiety są określane jednoznacznie przez kolejny numer oraz węzeł źródłowy. Pakiety są wysyłane z węzła, jeśli nie jest to węzeł docelowy. Pakiety, które były wcześniej przyjęte w danym węźle, zostają odrzucone.

Metoda rozptylowa zapewnia przekazanie pakietu do węzła docelowego najkrótszą możliwą drogą i jest bardzo odporna na uszkodzenia węzłów oraz łączy telekomunikacyjnych. Jednak z oczywistych względów bardzo zwiększa ruch w sieci i nie może być wykorzystywana w sieciach o intensywnym przepływie pakietów.

2.2. Metoda losowa

Tą metodą wyboru drogi węzeł wysyła odebrany pakiet, który nie jest do niego skierowany, do losowo wybranego sąsiada. Jeśli wybór będzie wyznaczony przez odpowiednio dobrane prawdopodobieństwo, to można doprowadzić do sensownego rozkładu obciążeń poszczególnych łączy sieci. Jednak losowy wybór drogi, choć chroni przed skutkami uszkodzeń łączy i

węzłów, może w pewnych przypadkach doprowadzić do długotrwałego błędzenia pakietów w sieci przed osiągnięciem węzła docelowego. Skutki takiej metody byłyby szczególnie niepożądane dla sieci MSK, w której szybkości transmisji przez łącza telekomunikacyjne są stosunkowo małe.

2.3. Metoda dróg ustalonych

W metodzie tej jest wykorzystywana informacja o topologii sieci. W każdym węźle przechowuje się tablicę, w której dla każdego węzła docelowego w sieci jest określony węzeł, gdzie należy kierować dany pakiet. Jest to metoda prosta i skuteczna, ale bardzo nieodporna na awarie w sieci. Zwykle stosuje się więc dla tej metody pewną modyfikację, dzięki której oprócz drogi podstawowej jest określona droga zastępcza (lub kilka dróg zastępczych), wykorzystywana podczas awarii na drodze podstawowej.

Metoda dróg ustalonych z drogami zastępczymi prowadzi do przepływu pakietów po drogach najkrótszych w warunkach normalnej pracy sieci (bez awarii), a także pozwala na skuteczne nawiązywanie łączności między nadawcą informacji a ich odbiorcą w razie awarii na drodze najkrótszej. Ponadto, jeśli zawartość tablicy wyboru drogi będzie zmieniana dynamicznie w zależności od aktualnego stanu sieci, to można stosunkowo łatwo rozwinąć tę metodę w bardziej złożoną metodę dynamiczną.

2.4. Statyczne metody optymalne

Metody te wyznaczają drogi dla pakietów na podstawie rozwiązania specjalnych zadań optymalizacji określonych dla danej sieci z uwzględnieniem ograniczeń wynikających z konfiguracji sieci, szybkości transmisji w poszczególnych łączach oraz przewidywanego ruchu w sieci. Dla optymalizacji statycznej przyjmuje się na ogół jeden z poniższych celów:

- możliwie najlepsze wykorzystanie łączy przez równomierne rozłożenie ruchu,
- minimalizacja czasu przejścia pakietu przez sieć,
- minimalizacja długości drogi przebywanej przez pakiety,
- minimalizacja liczby węzłów przez jakie przechodzą pakiety.

Po rozwiązaniu zadania optymalizacji statycznej zostaje opracowana tablica dróg jak w metodzie poprzedniej.

2.5. Metody heurystyczne

W tych metodach za cel przyjmuje się minimalizację opóźnienia transzytowego. Przykładem może być metoda Barana [5], w której opóźnienia na drodze od węzła źródłowego i do węzła docelowego T szacuje się

na podstawie opóźnień transmisji w kierunku przeciwnym. Wartości opóźnień mogą być przekazywane przez dowolny pakiet z węzła T do węzła I.

W praktyce stosowanie tej metody jest dość trudne. Dokładny pomiar opóźnień wymaga synchronizacji zegarów w różnych węzłach, co jest trudne i kosztowne. Można szacować opóźnienia na podstawie np. długości kolejek czy liczby wolnych buforów informacji, ale wprowadza to błędy, które mogą doprowadzić do wyników nieprawidłowych. Ponadto opóźnienie na drodze od węzła I do węzła T może być różne od opóźnienia na tej drodze, lecz w kierunku przeciwnym.

2.6. Metody adaptacyjne

Przykładem metody adaptacyjnej jest metoda wykorzystująca wektory opóźnieniowe obliczone dla drogi prowadzącej dla każdego węzła docelowego przez łącza danego węzła. Wektory te są aktualizowane za każdym razem, gdy pakiet zostanie umieszczony w kolejce lub opuszcza węzeł.

W wyborze drogi wykorzystuje się też informacje o wektorach opóźnieniowych dla węzłów sąsiadujących z danym węzłem. Wektory opóźnieniowe do węzłów sąsiadujących mogą być przesyłane cyklicznie (metoda synchroniczna) lub po pewnych zdarzeniach, np. gdy wektor opóźnieniowy zmieni się o więcej niż o określoną wcześniej wartość Δt (metoda asynchroniczna).

Zastosowanie tej metody może prowadzić do lepszego wykorzystania podsieci komunikacyjnej. Jednak, aby dobrze sterować przepływem, należy opracować właściwy sposób szacowania wektorów opóźnieniowych, do czego potrzeba doświadczeń praktycznych dla sieci pracujących przy takich ograniczeniach, jakie będą występowały w sieci MSK.

2.7. Metoda hierarchiczna

W tej metodzie węzły sieci są grupowane w grona o strukturze hierarchicznej. Pojedynczy węzeł stanowi grono poziomu 0, grupa pewnej liczby sąsiadujących węzłów - grono poziomu 1, zbiory gron poziomu 1 tworzą grona poziomu 2 itd. W węzle są przechowywane szczegółowe dane o wierzchołkach grona poziomu 1, do którego ten węzeł należy i mniej informacji o węzłach bardziej oddalonych. Wadą tej metody jest wydłużenie drogi pakietów w sieci, a zresztą - podobnie jak dla metody poprzedniej - trudno określić na etapie projektowania sieci jakie informacje pozwolą na wybór najlepszych dróg.

2.8. Dynamiczne metody optymalne

Dynamiczne metody optymalne polegają na rozwiązywaniu statycznego zagadnienia optymalizacji ze względu na określone kryterium, ale z uwzględnieniem aktualnych danych o stanie sieci. Zadanie to można rozwiązywać cyklicznie albo po stwierdzeniu zmian w sieci. Metoda ta, oprócz konieczności zbierania i przesyłania odpowiednich (a trudnych do określenia na tym etapie) informacji pomiarowych ma tę wadę, że wymaga udoskonalenia znacznych zasobów obliczeniowych komputerów w węzłach sieci.

3. METODA WYBORU DROGI W SIECI MSK

3.1. Uwagi wstępne

Uwzględniając omówione na wstępie ograniczenia istniejące w sieci MSK postanowiono zastosować metodę dróg ustalonych z drogami alternatywnymi. Podczas tworzenia tablic wyboru drogi w poszczególnych węzłach będzie się przyjmować kryterium minimalnej liczby węzłów, przez jakie mają przechodzić pakiety do węzła przeznaczenia. Przy równomiernym rozkładzie ruchu pakietów pomiędzy różne węzły sieci i przy podobnych rozwiązaniach węzłów oraz identycznych szybkościach transmisji w różnych łączach, co można przyjąć dla pierwszego etapu budowy sieci MSK, kryterium to zapewni minimalizację opóźnień tranzytowych, które są charakterystyką najbardziej istotną dla użytkowników sieci.

Zastosowanie metody dróg ustalonych z drogami alternatywnymi nie powoduje konieczności dokonywania pomiarów określonych charakterystyk ruchowych w sieci MSK w celu sterowania drogami pakietów. Niemniej jednak, jeśli takie charakterystyki zostaną określone na etapie badań pierwszej wersji sieci MSK i będą mierzone, to będzie możliwe dynamiczne modyfikowanie zawartości tablic wyboru drogi przez proces w węźle, niezależny od samego procesu wyboru drogi. Dzięki temu będzie możliwa zmiana metody wyboru drogi bez konieczności zmiany wcześniej uruchomionego i sprawdzonego oprogramowania wyłącznie przez dodanie nowego fragmentu oprogramowania.

3.1. Opis implementacji metody wyboru drogi w węźle sieci MSK

Proces wyboru drogi w węźle sieci spełniającej wymagania X.25 należy do warstwy oprogramowania pakietowego (poziom 3 według podziału ISO [6]). Proces wykonano w postaci osobnego podprogramu, stanowiącego niezależny moduł oprogramowania węzła [7], co ma umożliwić łatwe zmiany pro-

cesu wyboru drogi i jednocześnie uniezależnić od protokołu pakietowego stosowanego w sieci. Podprogram ten otrzymuje na wejściu adres docelowy, a na wyjściu podaje numer linii (łącza), do którego należy przesłać dany pakiet.

Szczegóły implementacji metody wyboru drogi zostały podyktowane przez wymagania X.25, a także przyjętą koncepcję struktury oprogramowania węzła. W osobnym module oprogramowania węzła, tzw. module konfiguracji, który jest generowany indywidualnie dla każdego węzła w sieci i zmieniany (bez konieczności zmian samego podprogramu wyboru drogi, co spełnia podane we wstępie wymaganie łatwej adaptacji metody wyboru drogi dzięki zmianom topologii sieci) wraz ze zmianami konfiguracji sieci, są zawarte tablice wyboru drogi. Zgodnie z wymaganiami X.25 [1] abonenci sieci (DTE - data terminal equipment) mogą być określani przez jednoznaczny adres o długości nie przekraczającej 15 półoktetów. Dla pierwszej wersji sieci MSK przyjęto [8] adresy DTE o długości 4 półoktetów, z których dwa pierwsze określają numer węzła, a dwa ostatnie numer komputera komunikacyjnego podłączonego do tego węzła. Należy się jednak liczyć z tym, że w przyszłości sieć MSK będzie współpracować z innymi sieciami krajowymi i/lub zagranicznymi, w związku z czym będą stosowane adresy o innej długości. Dlatego przyjęto strukturę tablic wyboru drogi uniezależniającą działanie podprogramu wyboru drogi od długości adresów DTE (rys. 1).

Indeks tablic wyboru drogi podaje adresy tablic wyboru drogi dla długości adresów w półoktetach: $2 < k_1 < k_2, \dots, k_m \leq 15$, dopuszczalnych w danej wersji sieci MSK. Ograniczenie dolne wynika z zarezerwowania dwóch półoktetów na numer komputera komunikacyjnego dołączonego do węzła, a górne z maksymalnej długości pola adresu DTE, przewidywanej w zaleceniach X.25.

Tablica wyboru drogi dla adresów DTE o długości - przykładowo - k_m zawiera na początku maksymalny numer węzła dla tej długości adresów DTE, a następnie t pozycji opisujących drogi do węzłów o numerach $n_1^m < n_2^m < \dots < n_t^m$. Zauważmy, że różne tablice wyboru drogi mogą mieć różną długość, a numery węzłów o długości $k_u < k_v$ nie muszą stanowić części numeru węzła o długości k_v , co jest zgodne z zasadami skróconego adresowania dopuszczalnego przez zalecenie X.25.

Każda pozycja w omawianej tablicy wyboru drogi zawiera numer węzła n_w^m (przy czym $n_1^m < n_2^m, \dots, < n_w^m, \dots, n_t^m$) oraz numer linii podstawowej i_w^m prowadzącej niekoniecznie bezpośrednio do węzła n_w^m oraz numer linii alternatywnej j_w^m . W celu uproszczenia przetwarzania przez podprogram wyboru drogi uzupełnia się numery węzłów o nieparzystej liczbie półoktetów kodem F_{16} do pełnych 8-bitowych bajtów. Dobre uporządkowanie tablicy wyboru drogi według rosnących numerów węzłów służy również do skrócenia przeszukiwania tej tablicy.

Indeks tablic wyboru drogi

Maksymalna długość adresu DTE (k_m)
Adres tablicy wyboru drogi dla adresów o długości k_1
$\vdots$
Adres tablicy wyboru drogi dla adresów o długości k_m

Tablica wyboru drogi

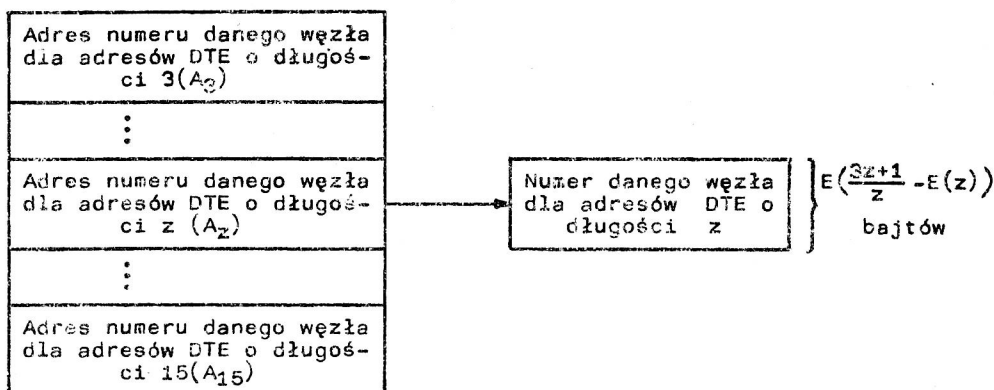
Maksymalny numer węzła sieci n_s^1
Numer węzła n_1^1
Numer linii podstawowej i_1^1
Numer linii alternatywnej j_1^1
$\vdots$
Numer węzła n_s^1
Numer linii podstawowej i_s^1
Numer linii alternatywnej j_s^1
$\vdots$
Maksymalny numer węzła sieci n_t^m
Numer węzła n_1^m
Numer linii podstawowej i_1^m
Numer linii alternatywnej j_1^m
$\vdots$
Numer węzła n_t^m
Numer linii podstawowej i_t^m
Numer linii alternatywnej j_t^m

Rys.1. Tablice wyboru drogi w węźle sieci MSK

Fig.1. Routing tables in MSK network node

Droga prowadząca do węzła docelowego n^m jest określana nie przez numer węzła sąsiedniego, jak się podaje w literaturze [3], lecz przez numer linii (łącza i_w^m lub j_w^m prowadzące do tego węzła). Dzięki temu tablice wyboru drogi są krótsze (numer węzła zajmuje nie mniej niż jeden bajt, a numer linii zawsze jeden bajt), a poza tym numer linii jest parametrem wygodniejszym do komunikacji z warstwą oprogramowania poziomu liniowego (w sieci MSK realizuje ona protokół LAPB [9] stanowiący podzbiór protokołu HDLC [10]), która jest dzięki temu uniezależniona od adresowania DTE.

Indeks numerów danego węzła



Tablica DTE danego węzła

Maksymalny numer DTE r_p
Numer DTE r_1
Numer linii i_1
⋮
Numer DTE r_p
Numer linii i_p

Rys.2. Tablice danego węzła

Fig.2. Tables of a node

Tablice wyboru drogi pozwalają na określenie linii prowadzących do innych węzłów. Natomiast do określenia drogi prowadzącej do DTE, podłączonego do danego węzła, są wykorzystywane inne tablice (rys. 2).

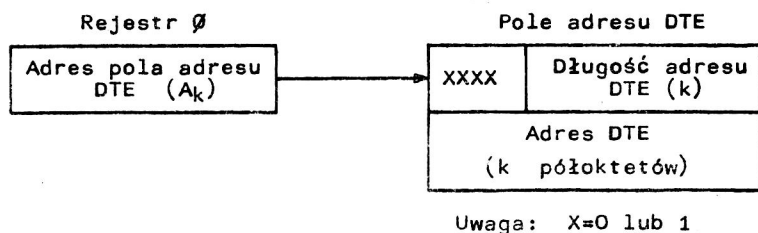
Indeks numerów danego węzła zawiera 13 pozycji, które zawierają adres numeru danego węzła A_z ($3 \leq z \leq 15$), przy czym $A_z = 0$, jeśli długość adresu DTE równa z nie jest dopuszczalna w danej wersji sieci MSK. Numery danego węzła są uzupełniane kodem F_{16} do całkowitej wielokrotności bajtów.

Numery DTE podłączone do danego węzła (tj. numery komputerów komunikacyjnych według pracy [8]) $r_1 < r_2 < \dots < r_p$ oraz prowadzące do nich linie $i_1, i_2, \dots, i_p$ są określone w jednej tablicy DTE podłączonych do danego węzła, gdyż numeracja ich będzie identyczna dla każdej z dopuszczalnych długości adresów DTE. W pierwszej wersji węzłów sieci

MSK przyjęto 2 jako dopuszczalne długości adresów DTE (pozwala na uzyskiwanie połączeń pomiędzy różnymi DTE podłączonymi do danego węzła) oraz 4.

3.3. Algorytm realizowany przez podprogram wyboru drogi

Podprogram wyboru drogi może być wywołany przez podzadanie X.25 i datagramów [11], stanowiące zasadniczą część warstwy oprogramowania poziomu 3 w wyniku odebrania pakietu CALL REQUEST (ewentualnie INCOMING CALL) [11], albo przez podzadanie miernictwa obsługujące pewien pakiet miernictwa [8]. Podprogram wyboru drogi sprawdza, czy podana w parametrach wejściowych (rys. 3) długość adresu DTE, k , jest dopuszczalna w danej wersji sieci. Zachodzi to wtedy, gdy pozycja indeksu numerów danego węzła dla długości k (rys. 2) jest niezerowa lub $k = 2$. Jeśli tak, to podprogram wyboru drogi sprawdza, czy dany węzeł jest węzłem docelowym ($k = 2$ lub zgodność numeru według parametrów wejściowych z numerem danego węzła). Gdy jest spełniony warunek, wówczas podprogram określi linię do DTE na podstawie tablicy DTE podłączonych do danego węzła (rys. 2).



Rys.3. Parametry wejściowe podprogramu wyboru drogi

Fig.3. Input parameters of routing subroutine

Natomiast jeśli pakiet jest zaadresowany do innego węzła, to numer linii zostaje określony przez przeszukanie tablicy wyboru drogi dla adresów DTE o długości k (rys. 1). Jeśli znaleziona linia i jest niesprawna, to znaczy, że nie znajduje się w stanie nawiązanego połączenia na poziomie liniowym (stan Information Transfer (9)), to zostaje wyznaczona linia zastępcza j .

Niesprawność linii prowadzącej do DTE podłączonego do danego węzła, obu linii na drodze do innego węzła lub brak odpowiednich pozycji w tablicach wyboru drogi (rys. 1) lub DTE podłączonych do danego węzła (rys. 2) powoduje odpowiedź negatywną podprogramu wyboru drogi (zerowy numer linii). Może to oznaczać brak możliwości nawiązania połączenia wirtualnego albo informację, że pakiet pomiarowy dotarł do węzła docelowego.

3.4. Zabezpieczenie przed cyklami w sieci

Zastosowana metoda wyboru drogi może w niektórych wypadkach awarii w sieci doprowadzić do powstania pętli zamkniętej (cyklu) w podsieci komunikacyjnej. W razie nawiązywania połączenia wirtualnego sytuację taką wykrywa się, jeśli w węźle znajduje się już tablica opisująca to połączenie (tzw. opis kanału logicznego (OKL) [11]). Jeśli tak, to węzeł wyśle zwrotnie (tj. do sąsiada nadającego pakiet CALL REQUEST lub INCOMING CALL) pakiet CLEAR powodujący rozłączenie tego nieudanego połączenia wirtualnego. Spowoduje to skasowanie odpowiednich OKL we wszystkich węzłach od danego węzła do węzła nadawcy.

W pakietach miernictwa sam pakiet zawierać będzie informacje umożliwiające stwierdzenie przepływu po cyklu i odrzucenie tego pakietu.

4. UWAGI KOŃCOWE

Metoda wyboru drogi dla pierwszej wersji sieci MSK, tj. metoda dróg ustalonych z drogami alternatywnymi spełnia wymagania narzucone przez omawiane zastosowanie. Jest odporna na awarie sieci (całkowicie dla pierwszej wersji trójwęzłowej sieci znanej pod nazwą PITAGORAS). Kryterium ustalania drogi podstawowej, zapewniającej osiągnięcie adresata po przejściu minimalnej liczby węzłów, wydaje się sensowne dla pierwszej wersji sieci MSK. Zmiany topologii sieci, a nawet zasad adresowania DTE, nie mają wpływu na sam podprogram wyboru drogi. I wreszcie, gdy zostaną określone charakterystyki ruchowe najistotniejsze dla użytkowników sieci i będą możliwe pomiary tych charakterystyk, wówczas metodę wyboru drogi można będzie stosunkowo łatwo zmienić na bardziej złożoną (np. dynamiczną) przez dodanie procesu aktualizującego tablice wyboru drogi na podstawie dokonanych pomiarów, lub też przez zmianę podprogramu wyboru drogi, który stanowi indywidualny moduł programowy węzła.

LITERATURA

- [1] CCITT; Draft Revised Recommendation X.25 Introduction and Packet Level, Com VII - No. 439E, listopad 1979.
- [2] Praca zbiorowa, Wprowadzenie do teleinformatyki, WKiŁ, Warszawa 1979.
- [3] Davies D.W., Barber D.L.A., Sieci teleinformatyczne, WNT, Warszawa 1979.

- [4] W e s o ł o w s k i I., Metody wyboru dróg w sieciach teleinformatycznych z komutacją pakietów, Praca IPI PAN, Nr 310, Warszawa 1978.
- [5] B a r a n P., B o e h m S., S m i t h P., On distributed communications, Raport Rand Corporation, sierpień 1964.
- [6] ISO, Reference Model of Open Systems Interconnection, ISO/TC97/SC16/N227, sierpień 1979.
- [7] L e w o c J., Struktura oprogramowania węzła podsieci komunikacyjnej (w tym samym zeszycie).
- [8] Praca zbiorowa, Propozycja norm dla podsieci komunikacyjnej międzyuczelnianej sieci komputerowej, Szklarska Poręba, marzec 1980 (materiały nie publikowane).
- [9] CCITT, Proposed Revision to Level 1 and 2 of Recommendation X.25, COM VII No 374-E, październik 1979.
- [10] ISO, Data communication - High Level data link control procedures - Elements of procedures (independent numbering), ISO/ITC 97, wrzesień 1976.
- [11] L e w o c J. i in., Zasady przygotowywania oprogramowania węzła, Raporty Centr. Oblicz. PWr., Seria SPR nr 14/80, Wrocław 1980.
Praca wpłynęła do Redakcji 25.11. 1980 r.
Po poprawieniu 3.03.1981 r.

ROUTING METHOD IN THE MSK COMPUTER NETWORK

The routing method has major influence upon operational characteristics of any packet-switching network. The paper presents routing methods used in computer networks and the reasons justifying the application of the fixed route method with alternate routes. Implementation of the method used is described and some detail information about software solutions applied in the MSK network node given.

Verified by Ruta Czaplińska

МЕТОД ВЫБОРА ПУТИ В СЕТИ ЭВМ MSK

Метод выбора пути влияет на характеристики сетей пакетной коммутации. В статье рассматриваются методы выбора пути, используемые в сетях ЭВМ, а также обоснования использования метода определенных путей с альтернативными путями. Описывается внедрение использованного метода и приводятся некоторые детальные программные решения, принятые в узле сети MSK.

Проверила Малгожата Хейдрих

*Sieć komputerowa, węzeł, minikomputer,
oprogramowanie, informacja, bufor*

Janisław MUSZYŃSKI^{*)}, Andrzej STANISZ^{**)}

Gospodarka buforami w węźle sieci komputerowej MSK

Rozpatrzono pracę węzła podsieci komunikacyjnej z punktu widzenia buforowania informacji. Opisano różne metody organizacji i przydzielania buforów oraz oceniono ich wpływ na efektywność programu węzła (wyrażoną zużyciem czasu procesora i stopniem wykorzystania pamięci). Zaprezentowano rozwiązanie problemu buforowania przyjęte w programie dla węzła sieci MSK, opartego na minikomputerze SM-3.

1. WPROWADZENIE

Działanie węzła sieci komputerowej, polegające na ciągłym odbiorze informacji napływających z linii i nadawaniu tych informacji po przetworzeniu w odpowiednie linie, wiąże się z potrzebą stałego buforowania informacji w węźle. Szybkość linii jest wielokrotnie mniejsza od szybkości jednostki przetwarzającej. Zastosowanie pamięci buforowej pozwala wyrównać różnice między tymi szybkościami. Jednostką przesyłania i przetwarzania informacji w węźle jest ramka - porcja informacji o zmiennej długości (od kilku do kilkuset oktetów). Ramki muszą być buforowane w węźle podczas odbioru i nadawania (ze względu na szeregowy sposób przesyłania informacji w liniach oraz podczas pobytu w węźle) w różnych

^{*)} Zakład Elektronicznej Techniki Obliczeniowej, ul. Ofiar Oświećcimskich 7/13, 50-069 Wrocław.

^{**)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

kolejkach: wejściowej, do przetwarzania, wyjściowej, a także w kolejkach związanych z mechanizmami retransmisji. Pamięć buforowa jest więc niezbędna do właściwego zestrojenia szybkości linii i jednostki przetwarzającej węzła oraz umożliwia współbieżność realizacji procesów odbioru ramek z linii, ich przetwarzanie oraz wysyłanie w linię. W węźle pamięć buforowa spełnia następujące zadania:

- magazynuje kolejne znaki dostarczane przez linię dla programów kompletujących je w ramki;
- okresowo przechowuje skompletowane ramki dla programów użytkowych węzła;
- stanowi obszary robocze dla programów wyprowadzających, z których wysyła się w linię kolejne znaki.

Sposób buforowania i związana z nim organizacja oprogramowania węzła ma - oprócz innych czynników - istotny wpływ na właściwości użytkowe sieci. W artykule omawia się wymagania funkcjonalne dla węzła, związane z buforowaniem informacji, a następnie - po krótkim przeglądzie znanych technik buforowania - przedstawia rozwiązanie przyjęte dla węzła sieci MSK, realizowanego na minikomputerze SM-3.

2. PRZEPŁYW I BUFOROWANIE INFORMACJI W WĘZLE

Wszelkie informacje w sieci komputerowej są przesyłane przez łącza synchroniczne blokami o zmiennej długości, mającymi format tzw. ramek [1]. Pojedyncza ramka może zawierać od kilku do kilkuset oktetów danych, przy czym struktura oraz maksymalna i minimalna długość ramki są określone przez protokół przyjęty w sieci.

W sieciach z komutacją pakietów stosujących protokół X.25 według zaleceń CCITT [1] wyraźnie wyróżniają się dwa typy ramek:

- a) ramki krótkie (od kilku do kilkunastu oktetów), używane do celów organizacji przepływu danych, np. ramki nadzorcze i nienumerowane poziomu liniowego, pakiety sterujące itp.;
- b) ramki długie (od kilkunastu do kilkuset oktetów), przenoszące właściwe dane przesyłane w sieci, tj. ramki informacyjne zawierające pakiety danych.

Większość ramek krótkich, odbieranych przez węzeł (wszystkie ramki nieinformacyjne i niektóre informacyjne), nie wymaga dalszego przesyłania, choć może powodować generowanie przez węzeł pewnych ramek odpowiedzi (również krótkich). Ramki długie (z danymi) i niektóre krótkie zazwyczaj są przesyłane dalej w sieć (po przetworzeniu w węźle).

Długość ramek dochodzących do węzła nie jest znana wcześniej, ale dopiero w momencie zakończenia odbioru. Długość prawidłowej ramki nie

przekracza jednak pewnej maksymalnej liczby oktetów. Ramki mogą docierać do węzła "paczkami" - z zachowaniem minimalnego odstępu pomiędzy ramkami, tzn. pojedynczego oktetu FLAG, lub "pojedynczo" - rozdzielone większymi przerwami. W związku z tym węzeł powinien stale dysponować pamięcią buforową, pozwalającą na odebranie z każdej linii pewnej liczby ramek i przechowanie ich do czasu przetworzenia oraz wysłania w sieć. Brak miejsca w pamięci buforowej prowadziłby do gubienia ramek i dodatkowego angażowania mechanizmów retransmisji protokołu liniowego, a w konsekwencji - do obniżenia efektywnej szybkości przesyłania danych w sieci.

Długość nadawanych ramek jest znana w węźle w chwili rozpoczęcia transmisji. Dotyczy to zarówno ramek przechodzących przez węzeł (na ogół długich), jak i ramek generowanych w węźle (najczęściej krótkich). Pozwala to na umieszczanie ramek generowanych w węźle w obszarach buforowych o wielkości dostosowanej do długości tych ramek. Z drugiej strony prostota przetwarzania wymaga nadawania ramek przechodzących przez węzeł z tych samych obszarów buforowych, do których wprowadzono je podczas odbioru (przepisywanie ramek pomiędzy buforami w węźle doprowadziłoby do znacznego narzutu w przetwarzaniu).

Organizacja buforowania ramek w węźle powinna zapewnić łatwość manipulacji na buforach związanych z odbiorem, przetwarzaniem i nadawaniem ramek (tj. tworzenie obszarów buforowych do odbioru ramek, organizacja kolejek i przenoszenie ramek pomiędzy kolejkami, odzyskiwanie obszarów buforowych po wykorzystaniu itp.). Zdaniem autorów racjonalne rozwiązanie problemów gospodarki buforami w węźle wymaga znalezienia kompromisu pomiędzy dążeniem do pełnego wykorzystania pamięci dostępnej dla buforów - w warunkach zmiennej długości ramek - a szybkością i wielkością programu.

Problemy gospodarki buforami są szczególnie ważne w węźle nie wykorzystującym pamięci zewnętrznej, w którym wszystkie potrzebne bufory muszą być zorganizowane w ograniczonym obszarze pamięci operacyjnej. Bufory te mogą być wykorzystywane jako:

- bufory wejściowe, w których programy obsługi adapterów liniowych składają ramki z kolejnych oktetów odbieranych z linii,
- bufory robocze, w których są przechowywane ramki w różnych fazach przetwarzania w węźle,
- bufory wyjściowe, z których programy obsługi adapterów liniowych pobierają kolejne oktety podczas nadawania ramki.

Rozwiązaniom możliwym w tym zakresie oraz ich krytycznej analizie jest poświęcona dalsza część artykułu.

3. METODY BUFOROWANIA

Istnieje wiele różnych metod buforowania bloków informacji (wiadomości, ramek itp.) w komputerach komunikacyjnych. Koncepcje te różnią się rodzajem buforów, sposobami ich przydzielania i zwalniania, a także złożonością algorytmów manipulacji na buforach. Omówimy je krótko.

3.1. Rodzaje i sposób przydzielania buforów

Z punktu widzenia organizacji zapisu danych w pamięci można rozróżnić trzy rodzaje buforów [2,3]:

- bufory liniowe, w których kolejne oktety ramki są umieszczane w kolejnych bajtach pamięci,
- bufory cykliczne, w których informacja jest wpisywana podobnie jak dla buforów liniowych z tym, że po osiągnięciu górnej granicy bufora dane są zapisywane ponownie od początku bufora,
- bufory łańcuchowe, w których informacja jest umieszczana w kilku blokach pamięci, powiązanych odsyłaczami.

Przydział obszaru buforowego do urządzeń lub modułów programowych może odbywać się w sposób statyczny lub dynamiczny. Przydział statyczny polega na wydzieleniu w pamięci przeznaczonych na bufory, obszaru na stałe przyporządkowanego określonej urządzeniu lub modułowi programowemu. Przy przydziale dynamicznym pamięć przeznaczona na bufory podzielona jest na bloki (zwykle o jednakowej długości), tworząc tzw. pulę buforów. Bloki pobierane z puli są przydzielane do modułów programowych, tworząc ich obszary buforowe.

Wyróżnić można kilka koncepcji organizacji pamięci buforów i dynamicznego przydzielania buforów do programu przetwarzającego:

1. Pojedyncza pula buforów o stałej długości i liniowej organizacji zapisu (długość bufora jest określona przez maksymalną długość ramki). Zaletą tego rozwiązania jest prostota oprogramowania i oszczędności czasowe uzyskane kosztem niezbyt efektywnego wykorzystania pamięci.

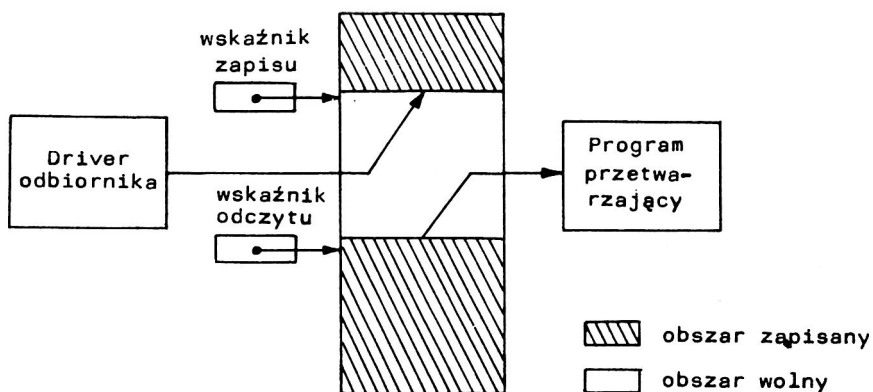
2. Kilka puli buforów o stałych długościach (w obrębie puli) i liniowej organizacji zapisu. W porównaniu z poprzednim rozwiązaniem wykorzystanie pamięci zwiększa się, ale podczas przydzielania bufora trzeba znać długość ramki, co komplikuje oprogramowanie.

3. Pula buforów o stałej długości i łańcuchowej organizacji zapisu; wzrasta przy tym wykorzystanie pamięci, jednakże kosztem istotnych komplikacji programowych.

3.2. Organizacja buforów wejściowych węzła

Każdej linii węzła musi być przyporządkowany obszar buforowy, zdolny pomieścić pewną liczbę ramek nadchodzących z linii i przechować je do czasu rozpoczęcia przetwarzania. Przetwarzanie odbieranych ramek powinno odbywać się w taki sposób, by w wejściowym obszarze buforowym linii stale istniała wolna przestrzeń, pozwalająca na odebranie przynajmniej jednej ramki o maksymalnej długości. Możliwe są różne organizacje buforów wejściowych.

Bufory cykliczne. Każdej linii jest przydzielony na stałe bufor cykliczny (rys. 1), do którego program obsługi adapterów liniowych wpisuje kolejne oktety ramki wraz z dodatkową informacją wskazującą granice ramek i ewentualne błędy wykryte podczas odbioru. Miejsce zapisu informacji w buforze określa wskaźnik zapisu aktualizowany po każdym zapisie. Odczyt informacji następuje w programie przetwarzającym z miejsca określonego przez wskaźnik odczytu. Program przetwarzający przepisuje informację z bufora, składając ją w ramki w buforach roboczych, gdzie jest ona dalej przetwarzana. Program obsługi adapterów może kontrolować położenie wskaźników i wykrywać ewentualne przepełnienie bufora, spowodowane nienadążaniem programu przetwarzającego.



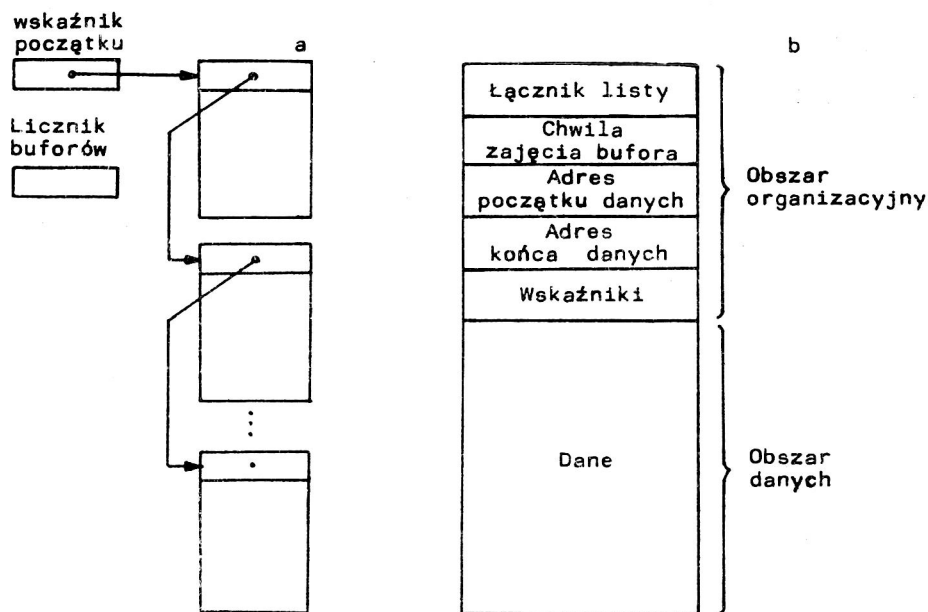
Rys.1. Koncepcja bufora cyklicznego
Fig.1. The cyclic buffer concept

Zaletą takiego rozwiązania jest prostota działania programu obsługi adapterów liniowych (uzyskana jednak kosztem przerwania pewnych działań na program przetwarzający). Jest to szczególnie istotne dla adapterów generujących przerwania dla przesłania każdego oktetu: duża inten-

sywność przerwań wobec znacznych szybkości transmisji wymaga bardzo prostej obsługi przerwań [5].

Metoda podwójnego buforowania. Każdej linii przydziela się dwa bufor o liniowej organizacji zapisu, wykorzystywane na przemian. Program obsługi adapterów liniowych wpisuje kolejne informacje do jednego z buforów, a po jego wypełnieniu - do drugiego. Informacje z wypełnionego bufora są pobierane przez program przetwarzający, aż do całkowitego opróżnienia bufora. Rozwiązanie to różni się od poprzedniego jedynie organizacją zapisu i odczytu danych w buforach, nie wnosząc istotnych korzyści.

Koncepcja buforów uniwersalnych. W obu poprzednio opisanych rozwiązaniach przydział pamięci na bufor wejściowy jest statyczny, co powoduje konieczność przepisywania informacji z bufora wejściowego do pewnego bufora roboczego. Możliwa jest jednak także taka organizacja buforowania, w której odbieranie, przetwarzanie, a następnie nadawanie informacji odbywa się bez przepisywania (tzn. bufor wejściowy są zarazem buforami roboczymi i wyjściowymi). Zakłada się przy tym zwykle dynamiczny przydział buforów z puli (rys. 2). Wielkość takiego bufora uniwersalnego jest dobierana w taki sposób, by pomieścił on ramkę o maksymalnej dopuszczalnej długości.



Rys.2. Organizacja puli buforów: a - lista wolnych buforów, b - struktura bufora

Fig.2. Buffer pool organization: a - free buffer list, b - buffer structure

Bufory wejściowe linii mogą być przy takiej koncepcji zorganizowane przez utworzenie kolejki (listy) z kilku buforów uniwersalnych i odpowiednie działania podczas odbioru ramek. Program obsługi adapterów kompletuje wtedy kolejne odbierane oktety w ramki, umieszczając ramki pojedynczo, w kolejnych buforach z listy. Program przetwarzający pobiera z kolejki bufory z odebranymi ramkami, a na końcu kolejki dołącza puste bufory.

Koncepcja taka wymaga pewnej rozbudowy programów obsługi adapterów w stosunku do poprzednich rozwiązań (w tym wykrywania początków i końców ramek, przełączania na następny bufor z kolejki, analizy błędów itp.), ale ma wiele istotnych zalet, z których najważniejszymi są:

- wyeliminowanie przepisywania każdej odebranej ramki (ramki są przetwarzane, a następnie nadawane z tego samego bufora uniwersalnego),
- możliwość dynamicznego ustalania pojemności bufora wejściowego (przez dodawanie nowych buforów uniwersalnych) w zależności od intensywności ruchu w linii,
- pewne oszczędności pamięci dla buforów wejściowych w stosunku do koncepcji z buforem cyklicznym (gdzie ze względu na konieczność zapamiętywania dodatkowej informacji kontrolnej - dotyczącej granic ramek i błędów - całkowita wielkość bufora wejściowego okazuje się większa niż pamięć potrzebna do zapamiętania tej samej liczby ramek w koncepcji buforów uniwersalnych).

3.3. Bufory robocze i wyjściowe

Obsługa ramek w węźle jest złożonym procesem realizowanym przez wiele współdziałających ze sobą modułów programowych. Przetwarzanie informacji odbywa się jednocześnie z odbiorem i nadawaniem ramek, w związku z czym w węźle zawsze istnieje pewna liczba ramek znajdujących się w różnych fazach obsługi i dotyczących różnych połączeń (kanałów) logicznych, utrzymywanych w węźle.

Organizacja buforowania w węźle powinna zapewnić możliwość manipulowania ramkami w celu przekazywania ich pomiędzy modułami programowymi, tworzenia kolejek itp. Jakkolwiek przetwarzanie ramek może odbywać się w obszarach buforowych organizowanych indywidualnie w poszczególnych modułach programowych, to jednak bardziej racjonalne jest dynamiczne przydzielenie buforów z puli i używanie ich jako buforów roboczych oraz wyjściowych.

3.4. Buforowanie i obsługa ramek

Obsługę ramek w węźle można podzielić na następujące 3 fazy:

- wprowadzanie,

- przetwarzanie,
- wyprowadzanie.

Na rysunku 3 podano schemat użytkowania buforów oparty na koncepcji dynamicznego przydziału buforów i obejmujący wymienione fazy. Dla uproszczenia pokazano tam sytuację dotyczącą pojedynczego połączenia logicznego, przechodzącego przez węzeł.

Faza wprowadzania. Kolejne znaki odbierane w adapterze liniowym są zapisywane przez driver w buforze wejściowym. Skompletowane ramki są w fazie wprowadzania kolejkwane (tworząc kolejkę wejściową do fazy przetwarzania).

Przy zorganizowaniu obszaru buforowego linii w formie bufora cyklicznego przydzielonego na stałe, bufor ten spełnia rolę kolejki wejściowej. Moduł wprowadzania pobiera wtedy ramkę z bufora cyklicznego, umieszcza w buforze roboczym pobranym z puli i przekazuje bezpośrednio do przetwarzania.

Faza przetwarzania. W fazie tej występują dwa poziomy przetwarzania:

- liniowy (poziom protokołu HDLC),
- pakietowy (poziom protokołu X.25).

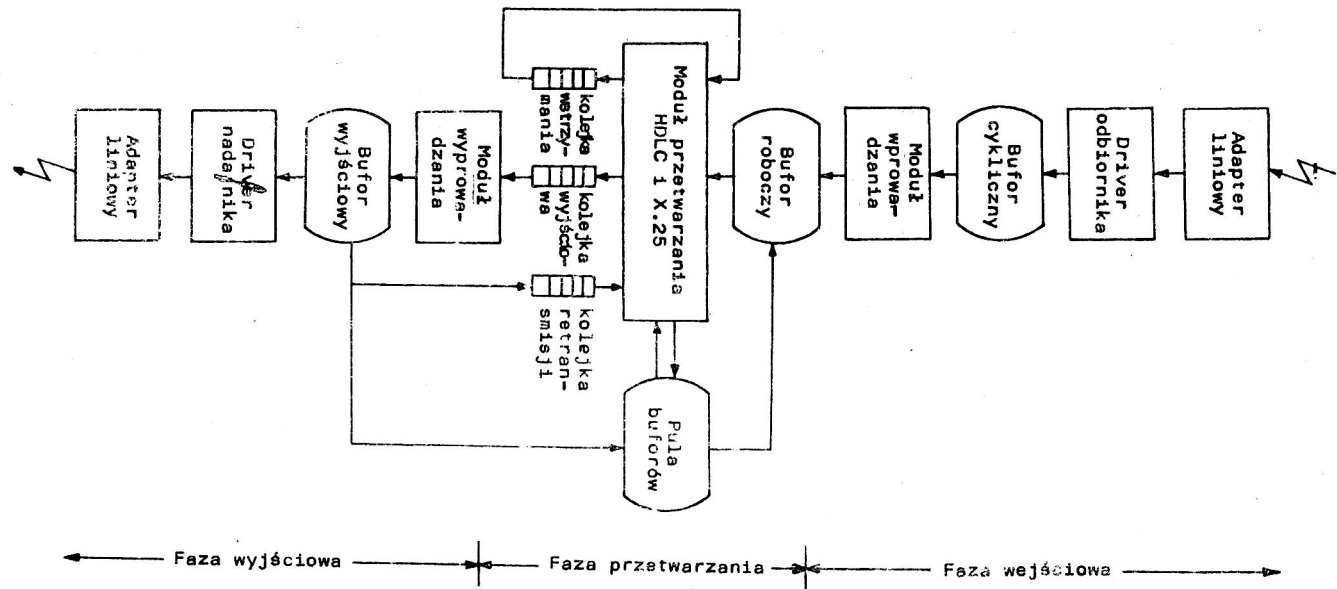
Moduł przetwarzania otrzymuje ramki z trzech źródeł:

- z kolejki wejściowej (z fazy wprowadzania),
- z kolejki wstrzymania, organizowanej przez poziom X.25,
- z kolejki retransmisji (tworzonej w fazie wyprowadzania).

Ramki z kolejki wejściowej są przetwarzane najpierw na poziomie HDLC. Poprawnie odebrane ramki informacyjne są kierowane do przetwarzania na poziomie X.25. Bufory zawierające ramki nieinformacyjne po przetworzeniu na poziomie HDLC są oddawane do puli. W razie odebrania ramki wymagającej odpowiedzi poziom ten tworzy w buforze pobranym z puli wymaganą ramkę i umieszcza w kolejce wyjściowej. Poziom HDLC obsługuje również kolejkę retransmisji przez oddawanie buforów do puli w miarę odbierania potwierdzeń lub - w razie braku potwierdzeń - kierowanie ich do kolejki wyjściowej w celu powtórnego nadawania.

Przetwarzanie na poziomie X.25 powoduje zwykle skierowanie ramki do kolejki wyjściowej odpowiedniej linii. W pewnych przypadkach - jeśli dalsze przesyłanie ramki nie jest chwilowo możliwe - poziom X.25 umieszcza ją w osobnej kolejce wstrzymania, z której zostanie skierowana do kolejki wyjściowej po odebraniu pakietu zawierającego odpowiednie potwierdzenie.

Obydwa poziomy przetwarzania mogą generować ramki i umieszczać je w kolejce wyjściowej nie tylko w odpowiedzi na aktualnie odebraną ramkę, ale także wskutek upływu limitów czasowych. Poziom X.25 może generować pakiety sterujące i kierować je do dwóch linii - w obu kierunkach połą-



Rys.3. Schemat wykorzystywania buforów w węźle
 Fig.3. Scheme of the buffer using in the node computer

czenia, a nawet - w razie restartu DTE - do wielu linii i dla różnych połączeń.

Bufory potrzebne do tych celów są pobierane z puli, przy czym podczas generowania pojedynczej ramki odpowiedzi zwykle może być użyty bufor zawierający aktualnie przetwarzaną ramkę.

Faza wyprowadzania. Bufory z kolejki wyjściowej są pobierane pojedynczo przez moduł wyprowadzania i służą jako bufory wyjściowe: driver nadajnika pobiera z nich kolejne znaki wysyłane w linię. Po zakończeniu nadawania bufory z ramkami informacyjnymi są kierowane do kolejki retransmisji, natomiast pozostałe - zwracane do puli.

4. ROZWIĄZANIE PROGRAMOWE

Dla węzła sieci MSK, kierując się kryterium minimalnego czasu przejścia ramki przez węzeł oraz warunkiem niegubienia znaków podczas odbioru [5,7], przyjęto następujące rozwiązanie. Na etapie wejściowym przydział pamięci jest statyczny: każdej linii jest przyporządkowany na stałe bufor cykliczny. Na etapie przetwarzania i wyjściowym różne moduły programowe korzystają z obszarów przydzielanych dynamicznie z puli buforów.

Z założeń przyjętych dla sieci MSK (maksymalna długość pola danych w pakiecie - 128 oktetów, przesyłanie datagramów z nagłówkiem długości 4 oktetów) wynika, że długości ramek przesyłanych w sieci wynosić będą 4-139 oktetów (z uwzględnieniem pola CRC), przy czym tylko niektóre ramki (przede wszystkim zawierające pakiety danych) mogą przekraczać długość 13 oktetów.

Na tej podstawie w oprogramowaniu węzła przyjęto, że będą używane dwa typy buforów, różniące się długościami:

- bufory długie (o całkowitej długości 150 oktetów),
- bufory krótkie (o długości 24 oktetów).

Bufory te będą zorganizowane w dwie niezależne pule o strukturze listowej (rys. 2a) i przydzielone dynamicznie w programie przetwarzającym.

Początkowy obszar buforów obu typów zawiera informacje organizacyjne (rys. 2b) - łącznie 10 oktetów, reszta bufora jest przeznaczona na dane.

Obie pule buforów są tworzone podczas obiegu wstępnego programu węzła: podprogram inicjacji dzieli dostępną pamięć na bloki i tworzy strukturę listy jednokierunkowej (pierwsze słowo bufora zawiera adres kolejnego bufora z listy lub $\emptyset$ jako wskaźnik końca listy). Liczba buforów obu typów jest ustalana podczas generacji oprogramowania. Według

wstępnych danych, dotyczących wielkości pamięci dla poszczególnych modułów programu węzła ocenia się, że do zorganizowania buforów dostępna będzie pamięć wielkości około 6-10 k słów. Pozwala to utworzyć np. bufory cykliczne dla 4 linii o wielkości 512 słów każdy oraz dwie pule buforów zawierające 40 buforów krótkich i ok. 50-100 długich.

Podczas pracy programu węzła bufory są pobierane z puli (co sprowadza się do prostej zmiany wskaźnika początku listy) i wykorzystywane jako bufory robocze lub wyjściowe. Po przetworzeniu ramki bufor na ogół jest kierowany do kolejki wyjściowej (zorganizowanej także w formie listy), a następnie - po wysłaniu ramki - zwracany do puli. Operacje te również wymagają prostych zmian odpowiednich łączników (konsekwentne wykorzystanie struktur listowych znacznie upraszcza manipulacje na buforach, eliminując potrzebę przepisywania ramek).

Dostęp programu do puli buforów i manipulacje na nich odbywają się przez następujące podprogramy:

- podprogram pobierania bufora długiego, który sprawdza istnienie wolnych buforów długich, pobiera pierwszy dostępny bufor, usuwa z puli i dostarcza jego adres; w razie braku wolnych buforów długich program wywołujący otrzymuje odpowiedź negatywną;
- podprogram pobierania bufora krótkiego, dostarczający adres pierwszego dostępnego bufora krótkiego, a w razie braku takowych - bufora długiego;
- podprogram oddawania bufora, który rozpoznaje typ bufora i włącza go do odpowiedniej puli.

Podprogramy pobierania buforów ustawiają aktualny czas w polu "chwila zajęcia bufora" w obszarze organizacyjnym (rys. 2b). Dzięki temu w chwili oddawania bufora można wyznaczyć czas zajęcia bufora i w ten sposób zbierać informacje statystyczne dotyczące czasu pobytu ramek (pakietów) w węźle [6].

Bufory puli nie są używane jako bufory wejściowe. Kierując się dążeniem do skrajnego uproszczenia obsługi adapterów liniowych, bufory wejściowe zorganizowano jako bufory cykliczne (rys. 1). Program obsługi adapterów wpisuje do bufora odpowiedniej linii informację wielkości jednego słowa na każdy odebrany oktett: informacja ta jest kopią zawartości rejestru buforowego adaptera i oprócz informacji odebranej z linii (młodszy bajt słowa) zawiera bity kontrolne, informacje o początku, końcu ramki, lub o wykryciu błędu (starszy bajt słowa). Informacje z bufora cyklicznego są pobierane przez początkowy fragment programu przetwarzającego (podzadanie wprowadzania [7]), który analizuje bity kontrolne, kompletuje ramki, wpisując je do bufora z puli i w tej postaci przekazuje do dalszego przetwarzania.

Długość bufora cyklicznego dla każdej linii ustalono na 512 słów, co pozwala zmieścić w nim 3-4 ramki długości zbliżonej do maksymalnej.

5. UWAGI KOŃCOWE

Rozwiązanie problemów buforowania, przyjęte dla węzła sieci MSK, cechuje prostota przetwarzania, ale wykorzystanie pamięci może być odległe od maksymalnego, jeśli udział ramek krótkich w strumieniach danych przesyłanych w sieci będzie znaczny. Pewne oszczędności pamięci przynosi możliwość stosowania buforów krótkich dla ramek generowanych w węźle, a także dla ramek krótkich odbieranych z linii i dodatkowo przepisywanych.

Przedstawione rozwiązanie zostało przyjęte na podstawie racjonalnych przesłanek projektowych (minimalizacja czasu przejścia ramki przez węzeł, wymaganie, aby nie gubić informacji w najbardziej niekorzystnych ramkach pracy węzła), ale bez znajomości własności statystycznych przepływu informacji w sieci. Dlatego problemy gospodarki buforami w węźle pozostają w dalszym ciągu otwarte i wymagają dalszych badań.

LITERATURA

- [1] Orange Book, vol. VIII, CCITT, Genewa 1977.
- [2] Mills D.L., Communication software, Proceedings of the IEEE, 1972, 11.
- [3] Rupińska M., Gospodarka buforami w węźle komunikacyjnym, maszynopis, IPI PAN, Warszawa 1979.
- [4] Sieci telekomunikacyjne komputerów (pod red. N. Abramsona i F.F. Kuo), WNT, Warszawa 1978.
- [5] Bieleninik E., Stanisław A., Szwarc M., Obsługa adapterów liniowych w węźle sieci komputerowej MSK (w niniejszym zeszycie).
- [6] Hudyma E., Kosmulska-Bochenek E., Rejestracja pomiarów w sieci komputerowej MSK (w niniejszym zeszycie).
- [7] Lewoć J., Struktura oprogramowania węzła podsieci komunikacyjnej (w niniejszym zeszycie).

Praca wpłynęła do Redakcji 7.01.1981 r.

Po poprawieniu 3.03.1982 r.

BUFFER MANAGEMENT IN THE NODE OF MSK COMPUTER NETWORK

The operation of a communication network node computer is discussed from view the information buffering point. Various methods of buffer organization and allocation are described and evaluated, a special attention being given to their effect on the node computer program performance (in terms of time consumption and memory occupation). The solution of buffering problems for the SM-3 minicomputer to be applied in the University Computer Network MSK as a node computer is presented.

Verified by Ruta Czaplinska

УПРАВЛЕНИЕ БУФЕРАМИ В УЗЛЕ СЕТИ ЭВМ MSK

Рассматривается работа узла сети связи ЭВМ с точки зрения буферизации информации. Описываются различные методы организации и распределения буферной памяти, а также оценивается их влияние на эффективность работы программы узла (измеряемую временными затратами и занятостью памяти). Представляется решение проблем буферизации в программе для мини-ЭВМ SM-3, используемой в качестве узла в сети ЭВМ MSK.

Проверила Малгожата Хейдрих

*Sieć komputerowa, węzeł, adapter liniowy,
transmisja danych, oprogramowanie*

Edward BIELENINIK^{*)}, Andrzej STANISZ^{*)},
Michał SZWARC^{*)}

Obsługa adapterów liniowych w węźle podsieci komunikacyjnej

Sformułowano ogólne wymagania dla obsługi adapterów liniowych, wynikające z funkcji węzła i założeń przyjętych dla sieci MSK. Przedstawiono rozwiązania zastosowane w wersji oprogramowania węzła, realizowanej na minikomputerze SM-3. Rozważono ewentualne modyfikacje eliminujące pewne wady przyjętej struktury.

1. WPROWADZENIE

Jedną z głównych funkcji komputera spełniającego rolę węzła sieci komputerowej [1] jest przesyłanie informacji przez kanały transmisyjne. Rodzaj zastosowanego kanału determinuje typ urządzenia (tj. adaptera liniowego), poprzez które komputer współpracuje z tym kanałem, a także odpowiednie rozwiązania programowe.

W sieci MSK będą wykorzystywane linie telefoniczne dzierżawione, 4-przewodowe o szybkości transmisji 2400-9600 bit/s i adaptery synchroniczne [6], odpowiadające urządzeniom DUP-11 firmy Digital Equipment Co. [5]. Adaptery te umożliwiają transmisję synchroniczną z szybkością do 9600 bit/s w trybie półdupleksowym lub pełnodupleksowym i realizują bi-towo-zorientowany protokół liniowy HDLC wraz z kontrolą CRC według CCITT [7]. Ze względu na obsługę programową wymagają one reakcji na przerwania pojawiające się w czasie transmisji po każdym znaku (okciecie).

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

W artykule przedstawiono rozwiązanie programowe obsługi adapterów w węzle sieci MSK (realizowanym na minikomputerze SM-3) na tle ogólnych wymagań funkcjonalnych węzła. Dokonano analizy rozwiązania i rozważono możliwość jego modyfikacji.

2. WYMAGANIA OGÓLNE

Poprawna praca węzła sieci komputerowej w znacznym stopniu zależy od przyjętego sposobu obsługi adapterów liniowych. Przyjęte rozwiązanie programowe powinno zapewnić efektywną pracę węzła w warunkach średnich obciążeń oraz poprawne funkcjonowanie podczas chwilowych przeciążeń. Przede wszystkim należy:

- zapewnić ciągłość odbioru i nie gubić znaków w odbieranych ramkach, aby uniknąć dodatkowego angażowania mechanizmów retransmisji protokołu liniowego;
- zapewnić ciągłość nadawania ramek w celu pełnego wykorzystania linii (bez wtrącania dodatkowych oktetów FLAG pomiędzy ramkami);
- właściwie ustalić proporcje pomiędzy czasem procesora zużywanym na obsługę transmisji a czasem przeznaczonym na przetwarzanie ramek, aby uniknąć - w warunkach średniego obciążenia - blokady węzła wywołanej nadmiernym wzrostem liczby ramek oczekujących na przetwarzanie;
- zapewnić zdolność węzła do rozładowywania przeciążeń polegającą na przyjmowaniu informacji w okresach maksymalnego ruchu w liniach nawet kosztem wstrzymania przetwarzania i nadawania.

2.1. Warstwy oprogramowania

Podane wymagania oraz sposób obsługi programowej zastosowanych adapterów liniowych prowadzą do wyróżnienia w oprogramowaniu węzła dwóch warstw:

- warstwy transmisji (drivery adapterów);
- warstwy przetwarzania.

W skład warstwy transmisji wchodzi programy obsługujące przerwanie od adapterów liniowych. Realizują one czynności niezbędne do odbioru i nadawania ramek. W czasie odbioru kolejne znaki dostarczane przez adapter są wpisywane do wcześniej przygotowanych buforów wejściowych, natomiast podczas nadawania - pobierane z buforów wyjściowych i wysyłane do adaptera. Zakończenie odbioru(nadawania) ramki oraz wykrycie błędu podczas transmisji jest sygnalizowane do programów warstwy przetwarzania.

W skład warstwy przetwarzania wchodzi programy zarządzające odbiorem i nadawaniem ramek, procedury protokołu liniowego i pakietowego oraz

programy pomocnicze. Programy te reagują na zdarzenia sygnalizowane z warstwy transmisji, przetwarzają odebrane ramki oraz inicjują nadawanie.

2.2. Uwarunkowania czasowe

Programy obsługi przerwania od adapterów muszą zapewnić reakcję na przerwanie w czasie nie większym niż czas nadawania(odbioru) jednego oktetu (ok. 3,3 ms przy szybkości transmisji 2400 bitów/s). W związku z tym powinny mieć priorytet w oprogramowaniu węzła. Ogólny bilans czasu procesora i wspomniane wymagania nakładają istotne ograniczenie na zakres i złożoność funkcji realizowanych przez drivery. I tak np. wychodząc z warunków maksymalnego obciążenia linii (6 linii w węźle, jednoczesna praca odbiorników i nadajników przy szybkości 2400 bitów/s) można oszacować, że aby uniknąć gubienia znaków, należy zastosować czas obsługi przerwania od adaptera nie większy niż ok. 280 μ s [3]. W mini-komputerze SM-3 można w tym czasie wykonać ok. 30 rozkazów.

Duża intensywność przerwania od adapterów liniowych powoduje, że na ich obsługę jest zużywany znaczny procent czasu procesora. Zakładając, że średnim obciążeniem węzła jest praca 4 linii z jednoczesnym nadawaniem i odbiorem z szybkością 2400 bitów/s, a obsługa jednego przerwania mieści się we wspomnianym ograniczeniu 280 μ s, można - podobnie jak w pracy [3] - dojść do wniosku, że obsługa adapterów liniowych zajmuje 2/3 czasu procesora, pozostawiając 1/3 dla programów warstwy przetwarzania.

W rzeczywistości czas obsługi przerwania adapterów nie jest stały i zależy od fazy transmisji, przy czym maksymalne wartości osiąga w sytuacjach wyjątkowych takich, jak zakończenie odbioru(nadawania), wystąpienie błędu, konieczność uaktywnienia warstwy przetwarzania itp. Dlatego należy oczekiwać, że rzeczywisty procent czasu na przetwarzanie będzie większy niż w przytoczonym oszacowaniu.

2.3. Kolejki

Aby zapewnić ciągłość odbioru i nadawania ramek z jednocześnie odbywającym się przetwarzaniem, konieczne jest wprowadzenie licznych kolejek, a mianowicie:

- kolejek wejściowych linii (dla ramek odbieranych),
- kolejek wyjściowych linii (dla ramek oczekujących na nadawanie),
- kolejek retransmisji (na potrzeby protokołu liniowego).

Ze względu na strukturę oprogramowania i podział funkcji może być celowe wprowadzenie innych kolejek.

3. ROZWIĄZANIE PROGRAMOWE

Wychodząc z przedstawionych wymagań funkcjonalnych przyjęto rozwiązanie (rys. 1a), w którym wyróżnia się następujące elementy:

- a) drivery odbiorników i nadajników,
- b) zadanie przetwarzania, podzielone na:
 - podzadanie wprowadzania,
 - podzadanie protokołu liniowego HDLC,
 - podzadanie protokołu pakietowego X.25,
 - podzadanie miernictwa i diagnostyki,
- c) zadanie wyprowadzania.

Elementy te stanowią moduły programowe w strukturze oprogramowania węzła, omówionej dokładniej w pracach [2,4].

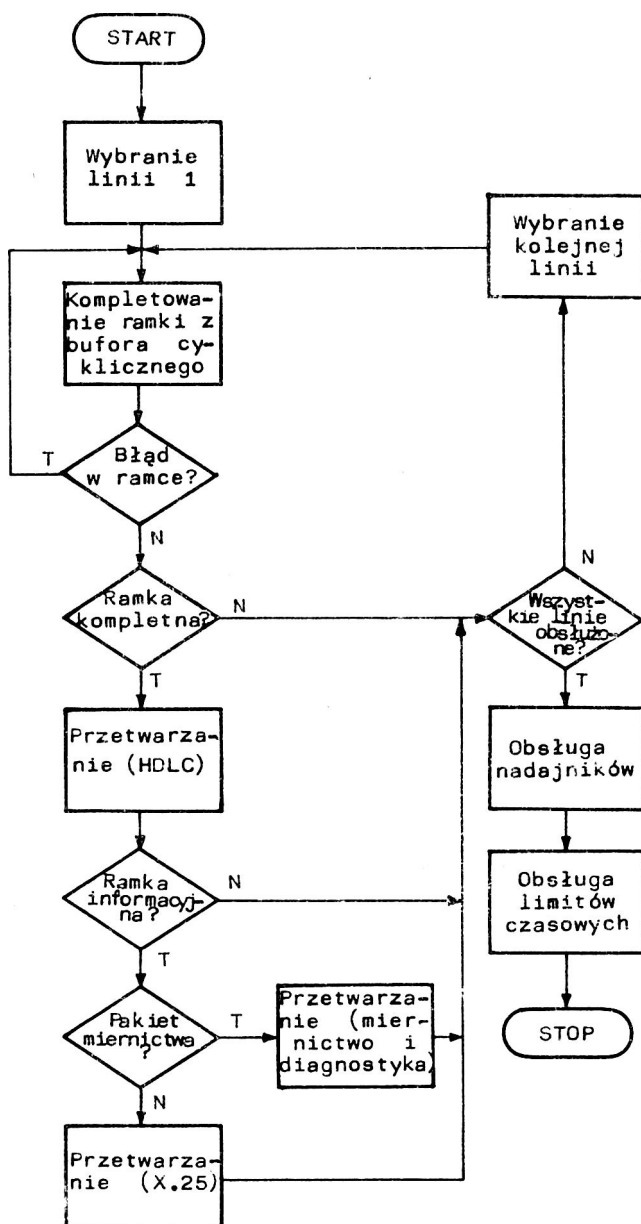
Warstwę transmisji tworzą moduły wymienione w punkcie a, a warstwę przetwarzania - w punkcie b i c. Komunikacja pomiędzy warstwami odbywa się poprzez koordynator z wykorzystaniem mechanizmu semaforów. Każdemu zadaniu przyporządkowany jest semafor (ogólny), ustawiany przez driver w celu zasygnalizowania koordynatorowi o potrzebie uaktywnienia tego zadania. Semafor jest zdejmowany przez zadanie po obsłudze uaktywnienia. Do sygnalizacji potrzeby obsługi poszczególnych linii wykorzystuje się podobny mechanizm z zastosowaniem semaforów indywidualnych (lokalnych) dla każdej linii - osobno dla nadajnika i odbiornika.

Działanie warstwy przetwarzania, wynikające ze struktury programowej (rys. 1a) oraz sposobu obsługi zadań przez koordynator, pokazano na rys. 2.

3.1. Odbiór informacji

Informacje są odbierane przez driver odbiornika oraz podzadanie wprowadzania.

Szeregowy strumień danych przychodzący z linii jest analizowany przez adapter liniowy, który wyróżnia początki i końce ramek, kompletuje informacje w oktety i generuje przerwania po odebraniu każdego oktetu. Obsługa takiego przerwania przez driver odbiornika polega na przepisaniu odebranego znaku wraz z bitami kontrolnymi (starszy bajt bufora odbiornika adaptera) do bufora cyklicznego spełniającego rolę kolejki wejściowej linii. Niezerowa zawartość bajtu kontrolnego (oznaczająca początek lub koniec ramki bądź wystąpienie błędu) powoduje uaktywnienie zadania przetwarzania i sygnalizację obsługi danej linii (przez ustawienie semafora ogólnego i lokalnego). Przepelnienie bufora cyklicznego



Rys.2. Uproszczony schemat przetwarzania informacji w węźle
 Fig.2. Simplified diagram of the data processing in communication sub-network node

powoduje zablokowanie przerwań odbiornika adaptera i uaktywnienie warstwy przetwarzania.

Zawartość bufora cyklicznego jest analizowana przez podzadanie wprowadzania (rys. 3), które przepisuje kolejne ramki do bufora pobranego z puli, z jednoczesną kontrolą poprawności odbioru, w formacie używanym podczas dalszego przetwarzania. Pojawienie się ramki przekraczającej dopuszczalną maksymalną długość powoduje inicjowanie odpowiedniego działania protokołu HDLC. Ramki poprawne są kierowane do dalszego przetwarzania.

Podzadanie wprowadzania rozpoczyna pracę zawsze od obsługi pierwszej linii, a po zakończeniu przetwarzania poprawnie odebranej ramki kontynuuje działanie dla następnej linii. Taki sposób postępowania został podyktowany dążeniem do równomiernej obsługi linii.

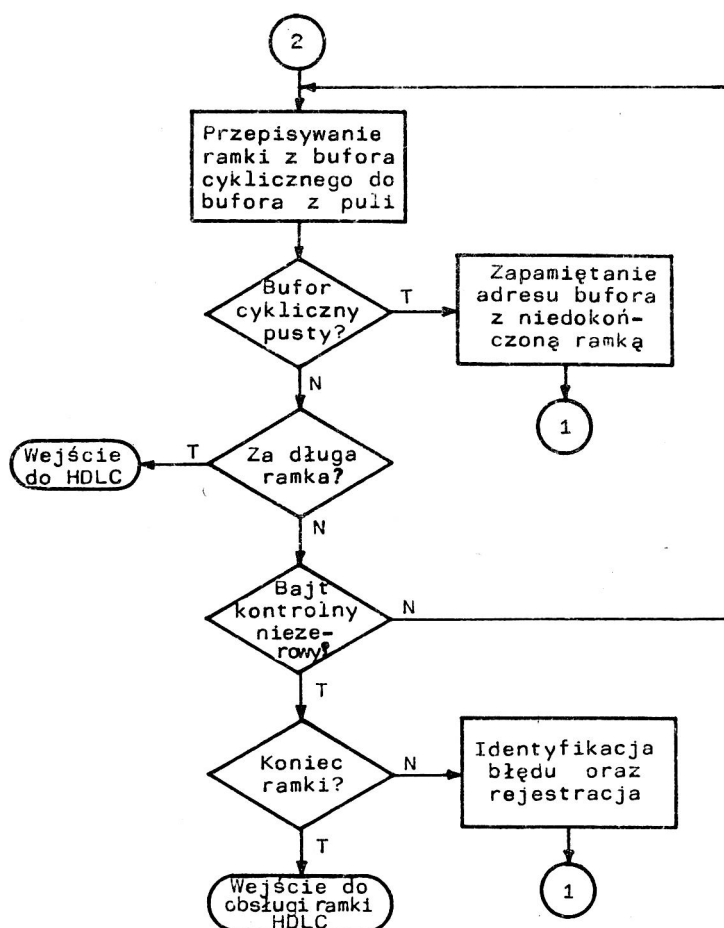
Po odebraniu ramki z przepełnionego bufora cyklicznego następuje odblokowanie przerwań umożliwiające dalszy odbiór informacji w tej linii. Po obsłudze ostatniej linii następuje wejście do koordynatora, który uruchamia pozostałe zadania warstwy przetwarzania (m.in. zadanie wyprowadzania).

3.2. Nadawanie informacji

Nadawanie informacji jest realizowane przez zadanie wyprowadzania i driver nadajnika.

Programy przetwarzające poziomu liniowego i pakietowego umieszczają ramki przeznaczone do nadawania w kolejce wyjściowej odpowiedniej linii. Zainicjowanie transmisji następuje w zadaniu wyprowadzania lub bezpośrednio po umieszczeniu ramki w kolejce (jeśli była ona dotychczas pusta). Powoduje to pojawienie się ciągu przerwań od nadajnika adaptera liniowego. Driver nadajnika w odpowiedzi na te przerwania umieszcza kolejne znaki w buforze nadajnika. W efekcie w linię jest wysyłany szeregowy strumień informacji tworzący ramkę. Zakończenie nadawania ramki lub wystąpienie błędu powoduje zablokowanie przerwań nadajnika i uaktywnienie zadania wyprowadzania dla odpowiedniej linii (przez ustawienie semaforów).

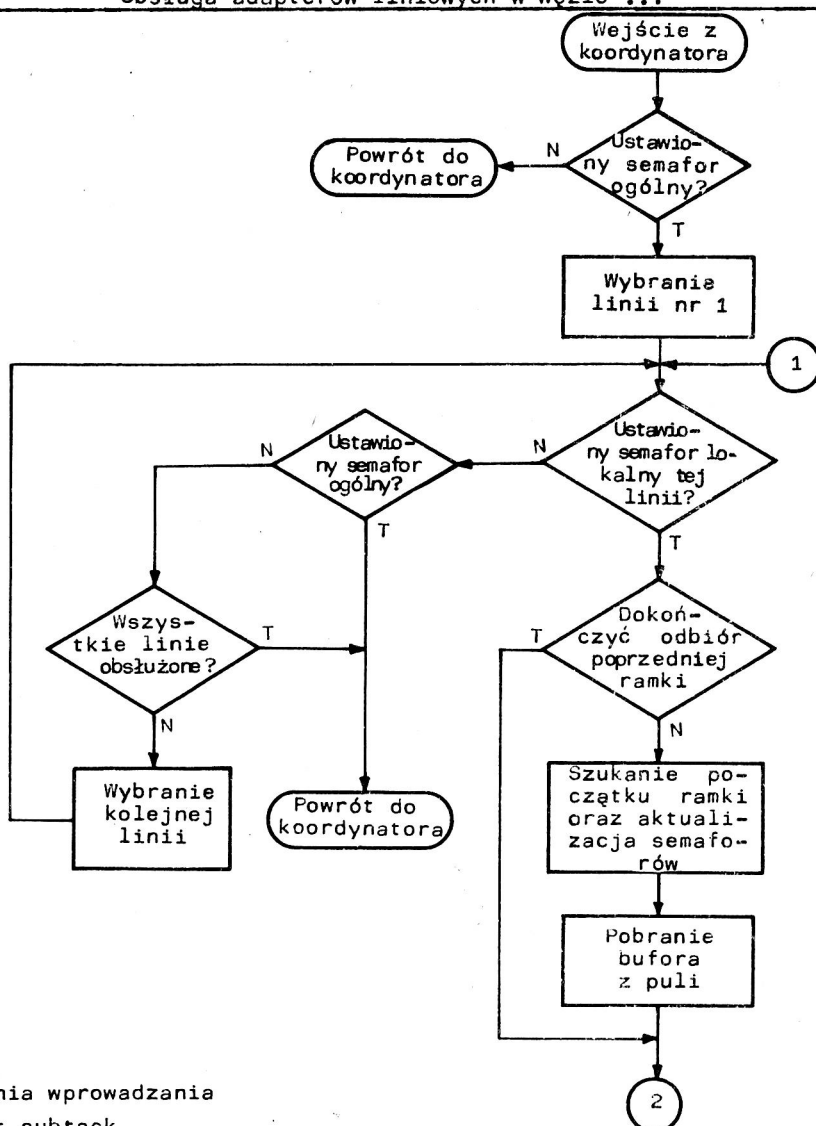
Zadanie wyprowadzania (rys. 4), po uruchomieniu przez koordynator, sprawdza sposób zakończenia poprzedniej transmisji. Po poprawnym jej zakończeniu przechodzi się do zainicjowania transmisji kolejnej ramki z kolejki wyjściowej po uzupełnieniu jej aktualną informacją protokołu HDLC. Po zakończeniu nadawania bufor jest oddawany do puli lub kierowany do kolejki retransmisji (dla ramek informacyjnych). W razie błędnego zakończenia transmisji nadawanie ramki jest powtarzane. Jeżeli w wyniku określonej liczby powtórzeń ramka nie zostanie nadana poprawnie, to następuje wyłączenie linii.



Rys.3. Schemat blokowy

Fig.3. Block diagram

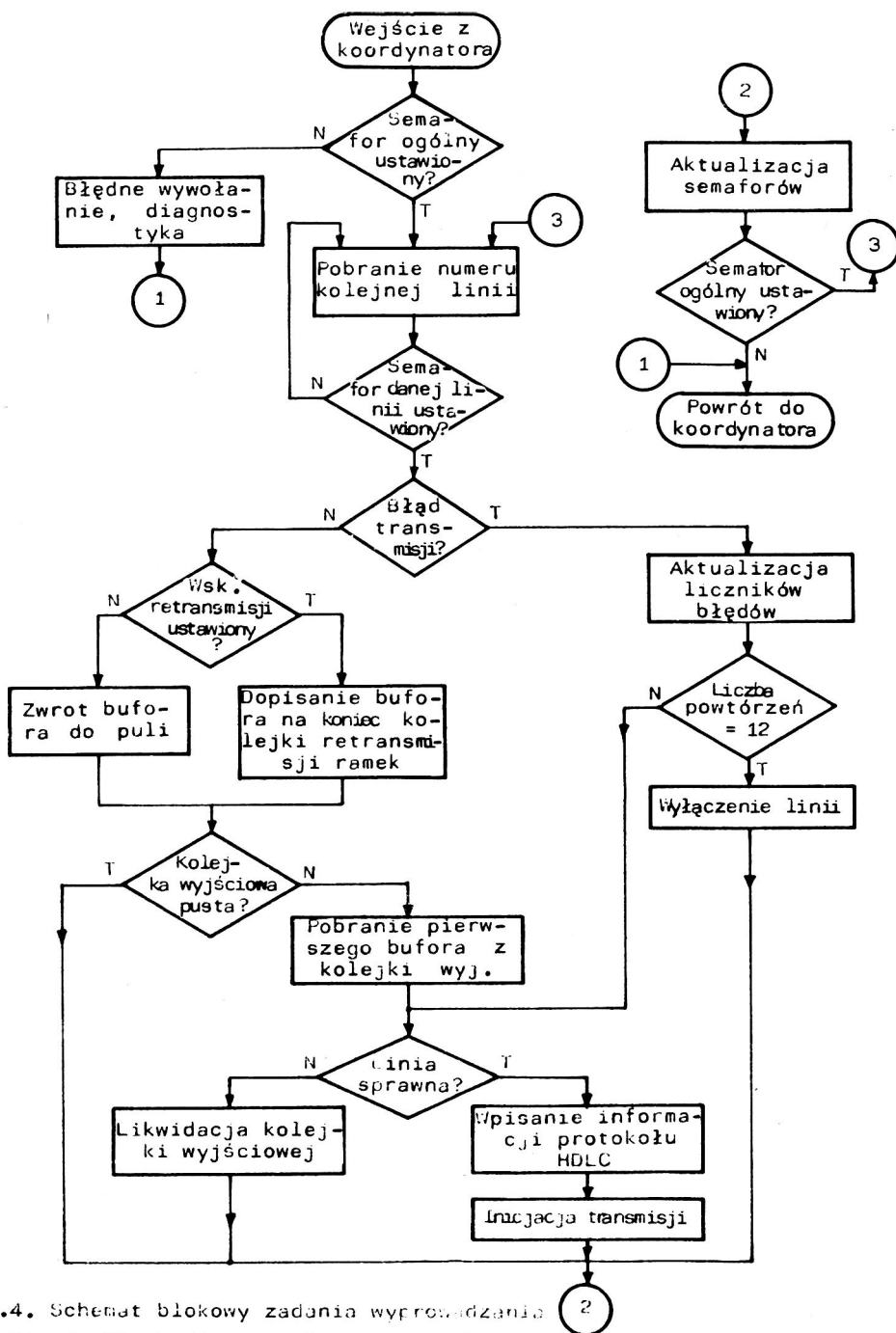
Opisane czynności są powtarzane dla kolejnych linii, aż do obsłużenia wszystkich uaktywnień sygnalizowanych przez semaforey. Aby zapewnić równomierną obsługę linii przyjęto, że po ponownym wejściu do zadania wyprowadzania obsługa rozpoczyna się od następnej linii do ostatnio obsłużonej.



podzadania wprowadzania
of input subtask

4. ANALIZA I PROFONOWANE MODYFIKACJE

Przedstawiona koncepcja obsługi adapterów liniowych opiera się na prostej strukturze systemu operacyjnego (koordynatora), przewidującej cykliczne uruchamianie zadań w warstwie przetwarzania w celu równomiernej obsługi linii i zadań. Dzięki wykorzystaniu koncepcji bufora cyklicznego jako formy kolejki wejściowej uzyskano skrajne uproszczenie ob-



Rys.4. Schemat blokowy zadania wyprowadzenia

Fig.4. Block diagram of output task

Obsługa adapterów liniowych w węźle ...

sługi przerwań adapterów i przerzucenie bardziej skomplikowanych działań związanych z transmisją do warstwy przetwarzania.

Wydaje się jednak, że rozwiązanie takie ma pewne wady, a zwłaszcza:

- konieczność przepisywania każdej odebranej ramki (z bufora cyklicznego do bufora z puli);
- znaczne prawdopodobieństwo nadawania dodatkowych oktetów FLAG pomiędzy ramkami, wynikające z braku kolejek nadajników i długiego czasu reakcji programów z warstwy przetwarzania na Jąktywnienia sygnalizowane przez driver nadajnika (zadanie wprowadzania może dojść do głosu i zainicjować kolejną transmisję dopiero po zakończeniu cyklu przetwarzania odebranych ramek);
- potencjalną nieaktualność informacji o stanie połączenia (na poziomie protokołu liniowego), spowodowaną przyjętą strategią obsługi linii (podczas kierowania ramek do nadawania nie korzysta się z informacji zawartych w ramach odebranych przez węzeł, ale oczekujących na przetwarzanie w buforach cyklicznych).

Wspomniane wady mogą być wyeliminowane przez pewne rozszerzenie funkcji realizowanych w driverach adapterów liniowych i zmiany w strukturze warstwy przetwarzania.

4.1. Proponowane zmiany

Istota zmian polega na wprowadzeniu dodatkowych kolejek (kolejki nadajników oraz kolejki do przetwarzania dla protokołu pakietowego) oraz innej strategii obsługi linii i przetwarzania (rys. 1b). Proponuje się rezygnację z koncepcji buforów cyklicznych i wprowadzenie kolejek do odbiorników i nadajników, utworzonych z buforów puli połączonych w listy. Przy takiej organizacji driverzy adapterów liniowych wprowadzają (wyprowadzają) informację wprost do(z) buforów z kolejki odbiornika(nadajnika), a po zakończeniu odbioru(nadawania) kontynuują działanie dla następnej ramki, wykorzystując kolejne bufory z odpowiedniej listy, aż do jej wyczerpania - bez potrzeby angażowania warstwy przetwarzania. Wykrycie przez driver błędu transmisji powoduje odpowiednie działania już w samym driverze: pojawienie się błędu w czasie odbioru powoduje umieszczenie odpowiedniego kodu błędu w części organizacyjnej bufora i rozpoczęcie odbioru kolejnej ramki. W razie pojawienia się błędu w czasie nadawania driver powtarza nadawanie ramki określoną liczbę razy. Zakończenie odbioru lub nadawania ramki, wymagające dalszej obsługi kolejki odbiornika lub nadajnika, jest sygnalizowane do warstwy przetwarzania z wykorzystaniem mechanizmu semaforów.

Działanie warstwy przetwarzania związane z obsługą adapterów liniowych obejmuje:

- umieszczanie w kolejkach odbiornika odpowiednio przygotowanych, pustych buforów,
- pobieranie z kolejki odbiornika buforów z odebranymi ramkami i ich przetwarzanie (dotyczy tylko protokołu liniowego: ramki informacyjne, wymagające przetwarzania protokołu pakietowego są kierowane do kolejki przetwarzania),
- umieszczanie w kolejkach nadajników buforów z odpowiednich kolejek wyjściowych (po uzupełnieniu ich informacją, wynikającą z aktualnego stanu połączenia liniowego) i inicjacja transmisji,
- pobieranie z kolejki nadajnika buforów, dla których zakończono nadawanie i kierowanie ich do kolejki retransmisji lub zwrot buforów do puli.

Realizacja wymienionych działań odbywa się (rys.5) w początkowej części programu warstwy przetwarzania, kolejno dla każdej linii w pętli, aż do całkowitego obsłużenia wszystkich uaktywnień sygnalizowanych przez driver. Dalsze przetwarzanie (realizacja protokołu pakietowego, limity czasowe) odbywa się po zakończeniu obsługi adapterów liniowych, przy czym do pętli obsługi adapterów wraca się po przetworzeniu każdego pakietu.

Obsługa limitów czasowych jest rozdzielona na dwie fazy. W pierwszej fazie sygnalizuje się upływanie zadanych odcinków czasu przez ustawienie odpowiednich wskaźników w programie, a w drugiej fazie (wykonywanej po przetworzeniu wszystkich ramek odebranych przez węzeł) następuje właściwa realizacja limitów czasowych.

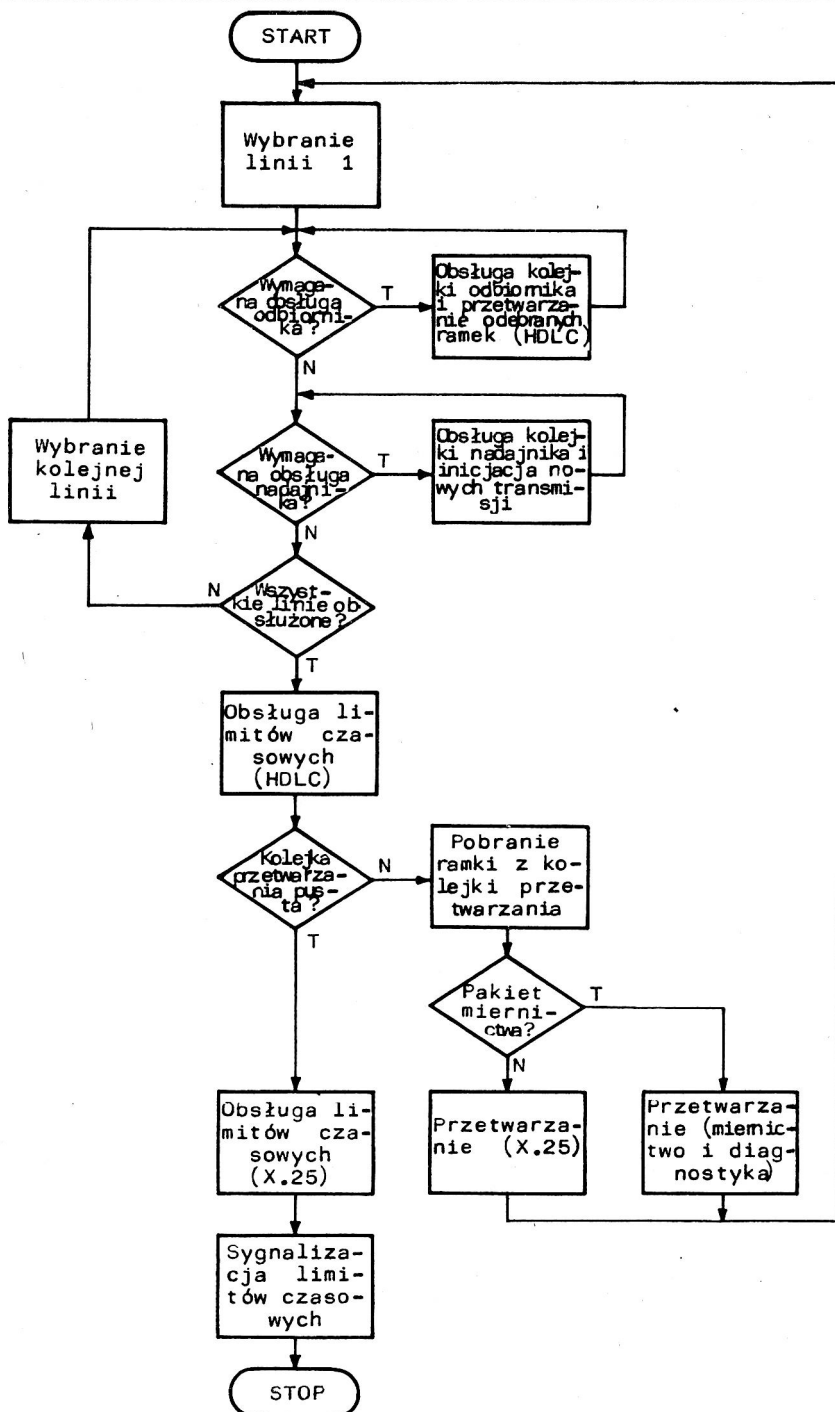
Takie rozwiązanie eliminuje możliwość podejmowania akcji protokołu liniowego i pakietowego po upływie limitu czasowego wtedy, gdy węzeł odebrał już odpowiednie informacje, ale nie zostały one jeszcze przetworzone.

4.2. Korzyści i ograniczenia

Przedstawiona propozycja modyfikacji struktury programowej powinna - jak się wydaje - doprowadzić do istotnego usprawnienia obsługi adapterów liniowych i bardziej efektywnej pracy węzła. Oprócz wyeliminowania wymienionych na wstępie wad rozwiązania opisanego w punkcie 3, przynosi ona dodatkowe korzyści w postaci:

- pewnych oszczędności pamięci w buforowaniu ramek odbieranych (w porównaniu z koncepcją bufora cyklicznego),

Rys.5. Schemat przetwarzania informacji w węźle - propozycja modyfikacji
Fig.5. Information processing diagram in communication subnetwork node -
- modified version



- możliwości dynamicznego ustalania długości kolejek odbiorników w zależności od obciążenia linii,
- zapewnienie potencjalnej ciągłości odbioru i nadawania informacji,
- usprawnienia pracy węzła przez skrócenie czasu reakcji na poziomie protokołu liniowego.

Osiągnięcie tych korzyści odbywa się kosztem pewnej komplikacji driverów w stosunku do wcześniej opisanego koncepcji, ale wydaje się, że całkiem realne jest takie zakodowanie obsługi przerwań adapterów liniowych, by czas obsługi mieścił się w ograniczeniach podanych w punkcie 2.2.

5. UWAGI KOŃCOWE

Poprawna praca węzła w sieci zależy nie tylko od przyjętego sposobu obsługi adapterów liniowych (w sensie realizacji odbioru i nadawania ramek), ale także od struktury warstwy przetwarzania. Przedstawione w pracy rozwiązania programowe mają wspólną cechę: opierają się one na jednoprogramowej strukturze warstwy przetwarzania powodującej, że kolejność działań jest ściśle określona (przez schematy z rys. 2 i 5).

Możliwe są oczywiście i inne rozwiązania, zwłaszcza wieloprogramowe, o bardziej złożonej strukturze. Celowość ich stosowania powinna być oceniana z punktu widzenia funkcjonowania węzła jako całości przez porównanie korzyści wynikających z większej efektywności węzła w stosunku do kosztów, związanych z komplikacjami struktury programowej. Problem taki wymagałby dokładniejszej analizy, a także lepszej znajomości warunków pracy węzła i sieci, skorzystania z doświadczeń uzyskanych w czasie wstępnej eksploatacji węzła w sieci.

LITERATURA

- [1] Davies D., Barber D., Sieci teleinformatyczne, WNT, Warszawa 1979.
- [2] Lewoc J., Struktura oprogramowania węzła podsieci komunikacyjnej (w niniejszym zeszycie).
- [3] Lewoc J., Łanowska B., Stanisław A., Szwarz M., Ocena możliwości realizacji węzła komunikacyjnego na bazie minikomputera SM-3, Raporty Centrum Oblicz. PWr., Nr 14/79, Wrocław 1979.
- [4] Lewoc J., Łanowska B., Stanisław A., Szwarz M., Zasady przygotowania oprogramowania węzła, Raporty Centrum Oblicz. PWr., Nr 14/80, Wrocław 1980.
- [5] PDP-11 Peripherals Handbook, Digital Equipment Corporation, Maynard, Massachusetts, 1975.

- [6] B a t y c k i T. i in., Adapter liniowy synchroniczny ALS11. Dokumentacja techniczna (wersja poprawiona), Raporty Inst. Cybern. Techn., PWr., Seria SPR 26/81, Wrocław 1981.
- [7] Orange Book, vol. VIII, CCITT, Genewa 1976.

Praca wpłynęła do Redakcji 8.01.1981 r.

Po poprawieniu 3.03.1982 r.

LINE ADAPTER SERVICE IN COMMUNICATIONS SUBNETWORK NODE

General requirement for line adapter service have been defined, basing on communications subnetwork node functions and assumptions accepted for the MSK computer network. Solutions applied in the software system realized on SM-3 minicomputer are presented. Possible modifications eliminating some inconveniencies of the structure proposed are disussed.

Verified by Ruta Czaplińska

ОБСЛУЖИВАНИЕ ЛИНЕЙНЫХ АДАПТЕРОВ В УЗЛЕ КОММУНИКАЦИОННОЙ ПОДСЕТИ

Сформулированы общие требования для процедуры обслуживания линейных адаптеров, вытекающие из функций узла и структуры MSK. Представлено решение, использованное в версии программного обеспечения узла, реализованной на мини-ЭВМ SM-3. Исследованы модификации, устраняющие некоторые недостатки принятой структуры.

Проверила Малгожата Кейлих

*Podsieć komunikacyjna, węzeł, zegar
sieciowy, program obsługi*

Barbara ŁANOWSKA^{*})

Moduł zegara w systemie operacyjnym węzła podsieci komunikacyjnej

Omówiono programy i podprogramy obsługi przyrostowego zegara sieciowego w systemie operacyjnym węzła podsieci komunikacyjnej. Proces obsługi, który nazywa się bedaniem zegara, jest wykonywany na 3. i 4. poziomie priorytetu.

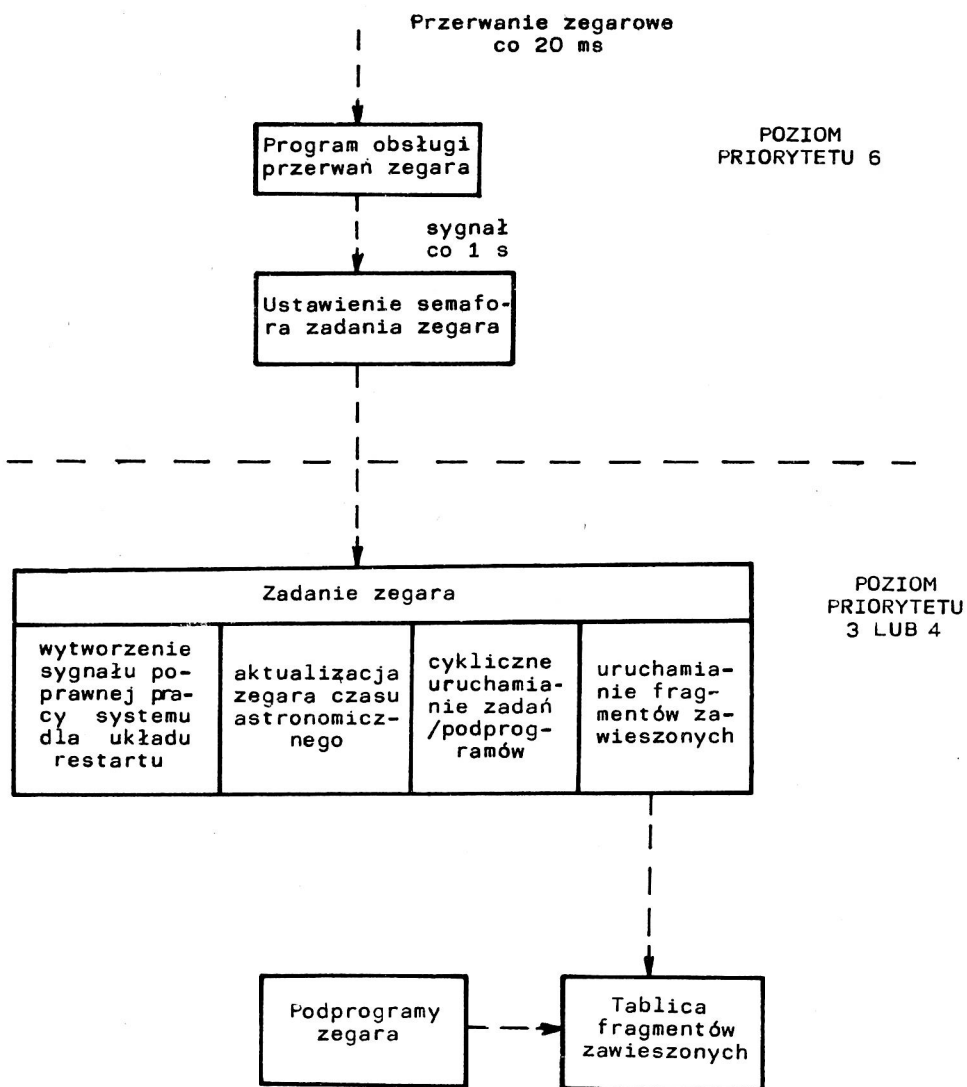
1. WSTĘP

W minikomputerze SM-3 upływanie odcinka czasu jest sygnalizowane przerwaniem zegarowym. Przerwanie to pojawia się w odstępach 1/50 s (20 ms), co wynika z częstotliwości sieci zasilającej. W celu zaimplementowania zegara przyrostowego dla systemu operacyjnego węzła opracowano następujące programy: program obsługi przerwań zegara, zadanie zegara, podprogramy zegara. Programy te spełniają następujące funkcje:

a) służą koordynatorowi do wyznaczania kwantów czasu pracy niektórych programów użytkowych. Kwanty te są wielokrotnością 1s,

b) spełniają funkcje "budzika" zabezpieczając system przed wykonywaniem zadań użytkowych bądź programów systemu operacyjnego przez czas nieskończenie długi [1]; każdemu z zadań przypisuje się dopuszczalny maksymalny czas działania i w razie przekroczenia tego czasu zachodzi proces powrotu za pośrednictwem fragmentu zawieszzonego. Schemat funkcjonalny dla powyższych programów przedstawiono na rysunku.

^{*}) Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.



Rys. Schemat funkcjonalny modułu zegara
Fig. Functional Diagram of Line Clock Module

2. PROGRAM OBSŁUGI PRZERWAŃ ZEGARA

Układ zegara generuje przerwanie co 20 ms. Zadaniem programu obsługi przerwań pracującym na poziomie 6 jest wytworzenie dla zadania ze-

gara sygnału o okresie 1 s. W tym celu odbywa się programowe zliczanie 50 przerwań zegarowych. Gdy nastąpi przepełnienie licznika (1 s), zwiększana jest o 1 zawartość odpowiedniego licznika zadania zegara (semafora zegara), sygnalizująca systemowi operacyjnemu, że należy to zadanie uruchomić.

3. ZADANIE ZEGARA

Zadanie zegara jest programem pracującym na poziomie priorytetu 3 i 4, uruchamianym co 1 s. Do jego funkcji należą:

- sprawdzanie poprawności pracy systemu operacyjnego węzła przez kontrolę tzw. liczników kontrolnych i generowanie sygnału poprawnej pracy dla układu restartu [3],

- modyfikacja zegara czasu astronomicznego,
- modyfikacja sekundnika systemowego,
- cykliczne uruchamianie pewnych zadań/podprogramów,
- uruchamianie tzw. fragmentów zawieszonych.

Omówimy kolejno poszczególne funkcje.

Dane zadania zegara zawierają 4 liczniki kontrolne, których zawartość jest odnawiana co określony czas przez wyznaczone programy systemu operacyjnego. Zawartości tych liczników są zmniejszane o 1 przez zadanie zegara. Gdy zawartości wszystkich liczników są niezerowe, uznaje się, że system operacyjny działa poprawnie i wysyłany jest przez zadanie zegara sygnał poprawnej pracy w postaci rozkazu nielegalnego 7, wytwarzającego następnie sygnał potwierdzający do układu restartu [3]. W razie wyzerowania dowolnego z liczników rozkaz 7 nie jest wykonywany. Powoduje to zadziałanie układu restartu i regenerację systemu operacyjnego.

Program zegara modyfikuje binarny sekundnik systemowy (SEKUNX) oraz aktualizuje pole (CZASX) pamiętające w kodzie ASCII bieżący czas astronomiczny. Kolejną funkcją zadania zegara jest uruchamianie zadań cyklicznych z poziomu 2 lub podprogramów cyklicznych z poziomu 3. Odbywa się to na podstawie danych zawartych w tablicy zadań (TZUCB) podprogramów (TPUCB) uruchamianych cyklicznie. Każda z tablic zawiera do 7 pozycji 4-słowych. Każda z kolei pozycja zawiera: licznik, okres uruchamiania, przekazywany parametr, numer zadania lub adres podprogramu. Zadanie zegara modyfikuje licznik, a w razie wyzerowania odnawia jego zawartość przez wpisanie okresu uruchamiania oraz przez podprogram uaktywniania zadań poziomu 2 powoduje uruchomienie zadania [4] lub podprogramu z przekazaniem parametru w rejestrze 1.

Obsługa fragmentów zawieszonych, polega na wywoływaniu sekwencji programu pracującej na poziomie priorytetu 3 lub 4 kończącej się powrotem do wspólnego miejsca w zadaniu zegara (ZADZGX). Uruchamianie fragmentów zawieszonych odbywa się na podstawie tablicy fragmentów zawieszonych (TFZAB). Tablica ta może zawierać do 128 pozycji 4-słowych. Każda pozycja zawiera licznik zadanego limitu czasowego, adres początkowy fragmentu zawieszonego, parametr, słowo stanu dla uruchamianego fragmentu zawieszonego. Dowolna część programu pracująca na poziomie 3 lub 4 może zadać limit czasowy (dopisać za pośrednictwem podprogramu zegara pozycję do TFZAB). Zadanie zegara przy każdym swoim przebiegu odejmuje 1 od liczników w poszczególnych pozycjach. W razie wyzerowania licznika (odmierzenia zadanego czasu) następuje uruchomienie fragmentu zawieszonego według adresu podanego w tablicy z przekazaniem mu w rejestrze R0 parametru i z zadaniem w tablicy słowem stanu [2]. Wszelkie działania na tablicy TFZAB są przeprowadzane w zadaniu zegara na poziomie priorytetu 4, aby nie dopuścić do jednoczesnego działania na zawartość tablicy TFZAB podprogramów zegara.

Przed uruchomieniem fragmentu zawieszonego odpowiadająca mu pozycja zostaje wyzerowana, a na jej miejsce - gdy nie jest to pozycja ostatnia - przepisuje się pozycję znajdującą się na końcu tablicy. Po przeszukaniu całej tablicy zadanie kończy się przez oddanie sterowania do koordynatora. Zaproponowane rozwiązanie zezwala na korzystanie we fragmentach zawieszonych z podprogramów zegara, a tym samym na zmiany w tablicy TFZAB. Obsługa po zakończeniu fragmentu zawieszonego polega na ponownym ustawieniu poziomu priorytetu 4 i poszukiwaniu zerowych liczników w tablicy, aż do sprawdzenia wszystkich pozycji.

4. PODPROGRAMY ZEGARA

Podprogramy zegara obejmują podprogramy zadawania limitu czasowego oraz podprogramy kasowania zadanego limitu czasowego. Mogą być wywoływane przez programy pracujące na poziomie 3. lub 4. Podprogramy zegara pracują na poziomie 4. Podprogram zadawania limitu czasowego powoduje dopisanie na końcu tablicy TFZAB pozycji o wartościach przekazanych w rejestrach R0, R1, R2 po wywołaniu podprogramu. Dopisanie pozycji do tablicy TFZAB jest równoznaczne z rozpoczęciem odmierzania na potrzeby programu odcinka czasu o zadanej długości. Podprogramy kasowania zadanego limitu czasowego pozwalają na skasowanie jednej lub wszystkich pozycji o podanym adresie fragmentu zawieszonego w tablicy TFZAB. Na miejsce kasowanej pozycji jest przepisywana pozycja ostatnia.

5. UWAGI KOŃCOWE

Opisany system pomiaru czasu nie zapewnia zbyt dużej dokładności, ponieważ opiera się na przerwaniach pochodzących z generatora synchronizowanego przebiegiem sieciowym. W związku z wahaniami częstotliwości sieci uchyb zegara może sięgać około 6 min na dobę. Zastosowanie generatora kwarcowego bez zmian w oprogramowaniu zwiększyłoby znacznie dokładność pomiaru czasu. Wszystkie procedury zegara są krótkie, tak że ich działanie nie wnosi błędów pomiarowych oraz nie jest przyczyną opóźnień w pracy innych programów.

LITERATURA

- [1] Y o u r d o n E., Projektowanie systemów o działaniu bezpośrednim, WNT, Warszawa 1976.
- [2] L e w o c J. i in., Zasady przygotowania oprogramowania węzła, Raporty Centr. Oblicz. PWr. Seria SPR nr 14/80, Wrocław 1980.
- [3] Ł a n o w s k a B., Inicjacja pracy i restart węzła podsieci komunikacyjnej (w tym samym zeszycie).
- [4] L e w o c J., Struktura oprogramowania węzła podsieci komunikacyjnej (w tym samym zeszycie).

Praca wpłynęła do Redakcji 2.12.1980 r.

Po poprawieniu 3.03.1982 r.

LINE CLOCK MODULE OF COMMUNICATION SUBNETWORK NODE OPERATING SYSTEM

The paper presents incremental line clock servicing routines in the operating system for the communication subnetwork node. The process known as the clock task is performed at the priority levels 3 and 4.

Verified by Ruta Czaplińska

МОДУЛЬ ИНКРЕМЕНТНЫХ ЧАСОВ В ОПЕРАЦИОННОЙ СИСТЕМЕ
УЗЛА ПОДСЕТИ СВЯЗИ

Описываются программы обслуживания инкрементных часов в операционной системе узла подсети связи. Представлена программа названная задачей часов, выполняемая на уровнях приоритетов 3 и 4.

Проверила Малгожата Хейдрих

*Driver, przerwanie, limit
czasowy, fragment zawieszony*

Barbara ŁANOWSKA^{*)}

Oprogramowanie podstawowych urządzeń wejścia-wyjścia w węźle podsieci komunikacyjnej

Omówiono programy sterujące urządzeń podstawowych wejścia-wyjścia, zwane driverami. Przedstawiono sposób komunikowania się pozostałych programów z driverami. Omówiono strukturę driverów. Zwrócono uwagę na opóźnienia czasowe informacji przesyłanej przez węzeł w razie jednoczesnego korzystania z więcej niż jednego drivera.

1. WSTĘP

Driver stanowi zestaw krótkich programów dotyczących jednego urządzenia zewnętrznego. Standardowy dostęp do driverów został zdefiniowany w opracowaniu [6].

Drivery podstawowych urządzeń we/wy są uruchamiane za pośrednictwem koordynatora - prostego systemu operacyjnego lub przez przerwania z urządzeń zewnętrznych. Zarezerwowano dla nich pracę na poziomie priorytetu 4. W pierwszej wersji oprogramowania przygotowano drivery następujących urządzeń: monitora ekranowego, drukarki mozaikowej, czytnika i perforatora taśmy. Każdy z driverów stanowi odrębny moduł, dla którego określono funkcje, parametry wejściowe i wyjściowe oraz sposób wywołania.

Urządzenia zewnętrzne są widziane przez program jako zbiór rejestrów adresowanych tak samo jak komórki pamięci operacyjnej.

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

Z każdym urządzeniem są związane dwa rodzaje rejestrów: stanu i danych. Liczba rejestrów i postać informacji w nich zależy od konkretnego urządzenia. W większości urządzeń występuje rejestr sterowania i stanu, który zawiera następujące bity: błędu, gotowości, zezwolenia na przerwanie i startu.

Programowo są ustawiane bity: zezwolenia na przerwanie (w razie ządania pojawienia się przerwania po wykryciu błędu lub przejścia urządzenia w stan gotowości) oraz startu. Bit gotowości jest ustawiany przez urządzenie, gdy jest ono gotowe przyjąć nową informację lub gdy zakończyło operację i ma dane gotowe do przekazania.

Programowanie urządzeń zewnętrznych w systemie SM-3 jest proste i nie wymaga specjalnej klasy instrukcji wejścia/wyjścia.

Obsługa transmisji w systemie SM-3 może odbywać się w dwojaki sposób:

- przez programowe badanie stanu gotowości urządzenia,
- za pośrednictwem przerw generowanych przez urządzenie.

Zrealizowane drivery wykorzystują drugi sposób. Pojawienie się przerwania powoduje przechowanie na stosie aktualnej zawartości licznika rozkazów i słowa stanu procesora. Nowe zawartości tych rejestrów są ustawiane na podstawie zawartości dwu słów sterujących, znajdujących się w pamięci operacyjnej. Z każdym urządzeniem jest związana jednoznaczna para takich słów, wybierana za pośrednictwem adresu (wektora przerwania) generowanego przez przerywające urządzenie.

2. KOMUNIKACJA MIĘDZY DRIVEREM I PROGRAMEM SYSTEMU

Łączność pomiędzy programem wywołującym i driverem odbywa się przez pole sterującą transmisją.

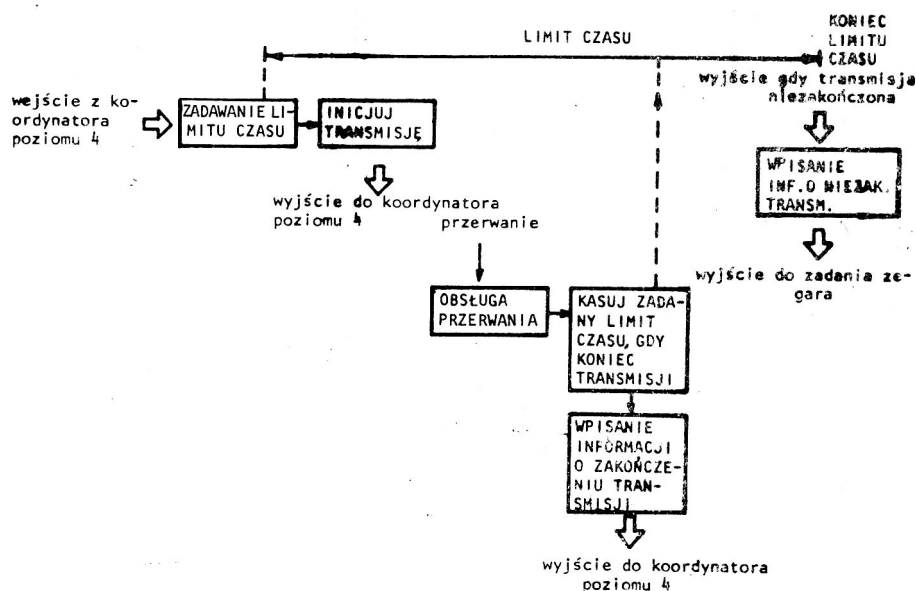
W opracowanej wersji oprogramowania systemu operacyjnego węzła transmisja z/do urządzeń wejścia/wyjścia będzie inicjowana przez zadania pracujące na poziomie priorytetu 2. Zainicjowanie transmisji wymaga określenia adresu pola sterującego transmisją (umieszczenia tego adresu w rejestrze ogólnym R0) oraz uruchomienia koordynatora instrukcją przerwania programowego (TRAP 0). Program żądający transmisji ustawia w polu sterującym następujące parametry: tryb/typ, adres bufora, licznik bajtów dla transmisji [6]. W zależności od podanego typu urządzenia koordynator uruchamia odpowiedni driver. Driver nie zmienia zawartości powyższych parametrów podczas ich wykorzystywania. Tryb określa rodzaj transmisji (czytaj, pisz). Uruchomiony driver realizuje transmisję i po jej zakończeniu podaje dla zadania wywołującego informację o poprawnym lub błędnym jej wykonaniu w słowie odpowiedzi pola sterującego. Po

zakończeniu działania driver przekazuje sterowanie do koordynatora poziomu 4. Niezależnie od wykonywanej części programu drivera zawartości rejestrów ogólnych pozostają nie zmienione. Zawartość tych rejestrów na/ze stosu przekazuje koordynator poziomu 4.

3. OGÓLNA STRUKTURA DRIVERA

Ogólną strukturę drivera przedstawiono na rysunku. Driver zawiera 3 krótkie programy: inicjowania transmisji, obsługi przerwania, obsługi w razie niezakończenia transmisji w założonym czasie.

Poniżej omówiono etapy realizacji transmisji.



Rys. Struktura ogólna drivera

Fig. General structure of driver

3.1. Program inicjowania transmisji

Po wywołaniu, zapoczątkowanym przez program wywołujący zgodnie z opisem w punkcie 2, driver inicjuje konieczne działanie ustawiając odpowiednio rejestry urządzenia [2-4]. Program inicjujący sprawdza popraw-

ność parametrów przekazanych w polu sterującym, tj. poprawność typu transmisji oraz niezbędne parametry; adres bufora i licznik bajtów umieszcza do dalszego wykorzystania w swoim roboczym obszarze. Ponadto jest zadawany limit czasowy (time out), czyli jest ustawiany za pośrednictwem programów obsługi zegara czas, w którym powinna zakończyć się transmisja. Następnie zostaje zainicjowane urządzenie przez ustawienie zezwolenia na przerwanie w jego rejestrze stanu. W urządzeniach start-stop (czytnik taśmy) jest ustawiany ponadto bit startu.

Po wykonaniu tych czynności driver przekazuje sterowanie do koordynatora. Dalszy przebieg transmisji jest wynikiem pojawienia się przerwania od urządzenia.

3.2. Obsługa przerwania

Przerwanie z danego urządzenia powoduje rozpoczęcie obsługi przerwania na podstawie zawartości słów sterujących, wskazywanych przez wektor przerwania tego urządzenia. Obsługa przerwania od urządzeń obejmuje właściwą transmisję danych. Najpierw program określa przyczynę przerwania. W razie przerwania nie spowodowanego błędem następuje przesłanie znaku między buforem urządzenia a buforem transmisji, określonym przez adres bieżący bufora. Po przesłaniu ostatniego znaku transmisja kończy się i w słowie odpowiedzi jest ustawiana informacja o zakończeniu transmisji. Zerowane jest ponadto zezwolenie na przerwanie w urządzeniu oraz przerywa się wcześniej ustawione odmierzanie czasu dla danej transmisji (kasowanie zadanego limitu czasowego). Gdy przerwanie zostało spowodowane błędem transmisji, następuje zakończenie transmisji z odpowiednią sygnalizacją w słowie odpowiedzi. Ponadto jest przekazywana informacja do podprogramów diagnostyki oraz jest zerowane zezwolenie na dalsze przerwanie. Kasowany jest również zadany limit czasowy. Obsługa przerwania zawsze kończy się standardowym wyjściem do koordynatora.

3.3. Zakończenie transmisji

Po poprawnym zakończeniu transmisji w słowie odpowiedzi są ustawione bity "transmisja zakończona" i "transmisja poprawna".

Jeżeli stwierdzi się błędy transmisji, to w słowie odpowiedzi zostaną ustawione bity "transmisja zakończona" i "transmisja z błędem".

Gdy w żądanym czasie transmisja nie zostanie zakończona, program zliczania czasu systemu operacyjnego węzła (zadanie zegara) powoduje uruchomienie odpowiedniej części programu w driverze, tzw. fragmentu zawieszonego. Program ten powoduje wyzerowanie zezwolenia na przerwanie

oraz ustawienie w słowie odpowiedzi bitów: "transmisja zakończona" i "transmisja z błędem". Program kończy się wyjściem do zadania zegara.

4. WYMAGANIA DODATKOWE DLA DRIVERÓW

Oprócz spełniania ogólnych funkcji omówionych powyżej, niektóre driverzy mają dodatkowe możliwości (driver DZM, ekranu monitora) lub różnią się od przyjętej ogólnej struktury (driver klawiatury monitora).

Driverzy drukarki mozaikowej i ekranu monitora mają możliwości przyjmowania kolejnych żądań transmisji, gdy jedna transmisja została już zainicjowana. Każdy z tych driverów zawiera tablicę 8-słową, pozwalającą na wpisanie do 8 kolejnych adresów pól sterujących. Po zakończeniu bieżącej transmisji są inicjowane kolejne transmisje, aż do opróżnienia kolejki. Gdy w zadanym czasie bieżąca transmisja nie zakończy się, jest kasowana cała kolejka i do wszystkich pól sterujących jest wpisywana informacja o zakończeniu transmisji z błędem.

Kasowanie kolejki następuje również, gdy zainicjowana zostanie transmisja dla nieoperatywnego monitora lub gdy podczas wyprowadzania informacji na ekran operator napisze znak CTRL/C.

Należy zwrócić uwagę na odmienną budowę drivera klawiatury monitora ekranowego.

Podczas inicjowania pracy systemu operacyjnego węzła jest ustawiane zezwolenie na przerwanie dla klawiatury monitora. Driver klawiatury nie zawiera części programu inicjującej transmisję za pośrednictwem koordynatora. Driver ten współpracuje z zadaniem operatora przesyłając doń komunikaty wprowadzane przez operatora systemu. Driver zawiera dwa wykorzystywane na przemian bufor o pojemności 80 bajtów (1 linia ekranu) każdy. Pierwsze słowo każdego bufora pełni rolę sterującą i zawiera licznik wprowadzonych bajtów i wskaźnik zajętości. Wskaźnik ten jest ustawiany przez driver po wykryciu pierwszego znaku nowego komunikatu i zerowany po wykorzystaniu informacji przez zadanie operatora. Każdy wpisany przez operator komunikat kończy się znakiem <CR>. Znak ten nie jest zliczany przez licznik wprowadzanych bajtów, ale jest przekazywany do zadania operatora.

Istnieje możliwość kasowania ostatniego zapisanego do bufora znaku (DEL) oraz całej linii (CTRL/U). Wprowadzane znaki o kodach mniejszych od 40 są przez driver ignorowane. Podczas transmisji przekraczających 80 znaków zawartość bufora jest kasowana i nie zostaje przesłana do zadania operatora. Tak samo postępuje driver, gdy w trakcie wprowadzania komunikatu w czasie 1 min nie naciśnięto żadnego klawisza.

Każdy wprowadzany znak zostaje natychmiast po naciśnięciu klawisza wyprowadzany na ekran. Jednakże, gdy w czasie wprowadzania komunikatu zostanie zainicjowana przez system operacyjny transmisja na ekran, uzyskuje ona pierwszeństwo. Wprowadzany komunikat zostaje skasowany i na czas trwania transmisji wyjściowej zostaje zablokowane przerwanie od klawiatury.

5. WPŁYW PRACY DRIVERÓW URZĄDZEŃ PODSTAWOWYCH NA OPÓŹNIENIA INFORMACJI PRZECHODZĄCEJ PRZESŁANEJ PRZESŁANEJ

Należy zwrócić uwagę, że zainicjowanie transmisji na urządzenia zewnętrzne powoduje uaktywnienie pracy programu na poziomie 4. i zablokowanie pracy poziomu 3. najistotniejszego z punktu widzenia użytkowego węzła. Czas obsługi informacji przesyłanej przez węzeł i obsługiwaną na poziomie 3. wydłuża się wtedy o czas pracy programów poziomu 4, tj. driverów.

Zaleca się więc korzystanie z driverów tych urządzeń tylko podczas niezbyt dużego obciążenia węzła obsługą przesyłanej informacji z/do adapterów liniowych.

LITERATURA

- [1] Programnoe obespečenie SM-3. Instrukcja po rozrabotke drajverov dla vnešnich ustrojstv nr 4.072.088 IE 14, 1978.
- [2] Videoterminal alfavitno-cifrovij SM 7205, OI, nr 263.043.014-IO TO, 1978.
- [3] Ustrojstvo vvoda i vyvoda perfolentočnoe SM 6202.01, nr 3.043.017-01 TO, 1978.
- [4] Ustrojstvo pečatajuščee alfavitno-cifrovoe SM 6300.01, nr 3.043 015-0ITO, 1978.
- [5] PDP-11 Peripherals Handbook, Digital Equipment Corporation, Maynard, Massachusetts 1975.
- [6] L e w o c J. i in., Zasady przygotowania oprogramowania węzła. Raporty Centr. Oblicz. PWr., Seria SPR nr 14/80, Wrocław 1980.

Praca wpłynęła do Redakcji 2.12.1980 r.

Po poprawieniu 3.03.1982 r.

SOFTWARE FOR BASIC INPUT/OUTPUT PERIPHERALS IN
COMMUNICATION SUBNETWORK NODE

In the paper basic peripheral control routines, known as drivers, are discussed. The way of communication between individual node routines and the drivers is presented. The structure of the drivers is described. Attention is given to time delays for data transferred through the node when more than one driver are used simultaneously.

Verified by Ruta Czaplińska

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ УСТРОЙСТВ ВВОДА-ВЫВОДА
УЗЛА ПОДСЕТИ СВЯЗИ

Рассматриваются управляющие программы устройств ввода-вывода, называемые драйверами. Описывается метод связи программ узла и драйверов, а также структура драйверов. Обращается внимание на временные задержки информации, передаваемой узлом в случае одновременного использования более одного драйвера.

Проверила Малгожата Хейдрик

Program, błąd, testowanie

Adam BECHER^{*)}

Zasady uruchamiania i testowania modułów programowych na przykładzie węzła podsieci komunikacyjnej

Występowanie błędów jest powszechnym zjawiskiem towarzyszącym programowaniu. W artykule opisano błędy popełniane przez programistów, sposoby ich unikania i usuwania oraz metody testowania programów. Podano również krótki przykład błędów wykrytych podczas kodowania i uruchamiania programu koordynatora węzła podsieci komunikacyjnej sieci komputerowej MSK.

1. KONIECZNOŚĆ TESTOWANIA

Testowanie modułów programowych jest konieczne, ponieważ każdy z nich po uruchomieniu zawiera pewną liczbę błędów. Program bezbłędny praktycznie nie istnieje, o czym łatwo jest się przekonać podczas testowania go w rozmaitych warunkach i dla różnych danych wejściowych. Rozważania podjęte w artykule dotyczą błędów popełnianych przez programistów, sposobów ich unikania i usuwania oraz metod testowania programów. Podano również krótki przykład błędów popełnianych podczas kodowania i uruchamiania programu koordynatora węzła podsieci komunikacyjnej.

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

2. RODZAJE BŁĘDÓW

Każdy programista liczy się z możliwością popełniania błędów określonego typu. Najprostszym przykładem są błędy syntaktyczne, zależne od konkretnego języka. Istnieją jednak błędy niezależne od maszyny, a tym samym niewykrywalne w pierwszym okresie uruchamiania. Mogą to być następujące błędy:

1. Niewłaściwy algorytm.

Z chwilą sformułowania zadania programista zaczyna szukać algorytmu lub metody rozwiązania problemu. Wobec ostrych ograniczeń czasowych będzie to prawdopodobnie pierwszy, nie zawsze najlepszy algorytm, dający rezultaty prawidłowe, ale zbyt wolno lub przy dużym marnotrawstwie pamięci. Po zakodowaniu programu i jego uruchomieniu nikt już nie będzie w stanie rozpocząć programu od nowa, czyli od wyboru algorytmu. Niestety często nie można stwierdzić, że algorytm jest zły zanim się go nie wypróbuje. Trochę czasu i wysiłku jednak warto włożyć w wybór algorytmu, aby nie trzeba było poprawiać każdego programu.

2. Niezgodność programu ze schematem blokowym.

Poprawki dokonywane w czasie uruchamiania programu z pewnością pochłoną programistę do tego stopnia, że po osiągnięciu sukcesu nie skoryguje on schematów blokowych. Utrudni to znacznie zadanie testowania.

3. Błędy wynikające z nieznanomości maszyny.

Bardzo trudno jest uniknąć błędów, które są następstwem specyficznej interpretacji niektórych konstrukcji przez maszynę. Przykładem niech będą następujące nie wyjaśnione sytuacje dla makroassemblera minikomputera SM-3:

- niebezpieczeństwa błędnej adresacji dla wysokich adresów dotyczą oprócz rejestrów również słowa stanu procesora; o ile obliczanie adresu $177776-2=177774$ jest akceptowane przez maszynę, o tyle czynność odwrotna, a mianowicie $177774+2$, nie jest wykonywana wskutek odwołania się do nie istniejącej komórki o adresie 177774; program jest natychmiast przerywany;

- pewne pozornie poprawne pod względem składni instrukcje nie są akceptowane przez makroassembler; użycie instrukcji

MOV Rn,-(Rn) n = 0,1,...,6

spowoduje błąd typu "działanie instrukcji nieokreślone"; próba wykonania tej instrukcji zakończy się jednak pomyślnie z następującym wynikiem:

(Rn-2) ← Rn-2,

choć cellem jej stosowania było uzyskanie wyniku:

$$(Rn-2) \leftarrow Rn;$$

godny uwagi jest fakt, że instrukcja odwrotna:

MOV (Rn)+,Rn n = 0,1,...,6

nie powoduje wymienionego błędu.

4. Błędy adresacji.

W trakcie przesuwania programów w pamięci mogą wyniknąć błędy związane ze stosowaniem niewłaściwych trybów adresacji. Podobnie w razie wstawiania fragmentów programów między rozkazy skoków typu BRANCH, adres skoku może stać się zbyt odległy, powodując błąd adresacji wykrywany przez makroassembler. Najczęściej spotykanym błędem adresacji jest jednak błąd parzystości adresu polegający na stosowaniu instrukcji słowowych zamiast bajtowych.

5. Inne błędy:

- wybór złej drogi w rozgałęzieniu,
- przeoczenie podczas kodowania bloków ze schematu blokowego,
- skok do złej etykiety,
- niewłaściwe otwarcie lub zamknięcie pętli,
- zła liczba powtórzeń pętli,
- pętle nieskończone,
- nierozważenie wszystkich możliwych typów danych,
- niezgodność przewidywanych formatów z formatami danych,
- stosowanie zmiennej bez wartości początkowej,
- pozostawienie starej wartości licznika lub akumulatora,
- pozostawienie nieocyszczonej tablicy,
- niepoprawne przekazywanie parametrów do podprogramów,
- użycie niepoprawnej stałej maszynowej, np. ósemkowej zamiast dziesiętnej.

Błędy wymienione w punktach 1-4, choć wydają się oczywiste, są nieuniknione. Wystąpienie kilkunastu błędów podczas pierwszego przebiegu tłumaczenia nie należy do rzadkości nawet u doświadczonych programistów.

Po wyeliminowaniu elementarnych błędów rozpocznie się wykrywanie błędów bardziej subtelnych. Przykładem może być nieświadome wykonanie instrukcji zdjęcia ze stosu w ciele podprogramu, a tym samym spowodowanie powrotu z podprogramu z błędnym adresem lub nawet brak powrotu. Podobnie mogą się zakończyć manipulacje na stosie podczas obsługi przerwania.

Omówione błędy są charakterystyczne dla dowolnego typu programów - stosowanych w handlu i administracji, organizacyjnych, obliczeń naukowo-technicznych, jak również sieci komputerowych. W ostatnim przypadku

należy wspomnieć jednak o błędach występujących w systemach czasu rzeczywistego. Błędy te, bardzo związane ze sprzętem i zależnościami czasowymi, mogą być szczególnie złośliwe i trudne do wykrycia [3]. Na przykład program mający za zadanie powtarzanie znaków z klawiatury na drukarce, uruchomiony pod systemem operacyjnym DOS, powtarzał co piąty, szósty znak, w najlepszym razie co drugi. Ten sam program uruchomiony za pomocą absolutnego programu ładującego działał bezbłędnie.

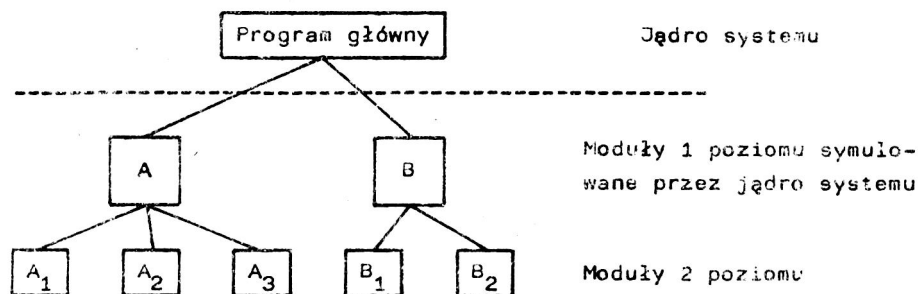
3. ORGANIZACJA TESTOWANIA MODUŁÓW PROGRAMOWYCH

Przed omówieniem ważniejszych metod testowania należy zwrócić uwagę na to, że programy testowe powinny być opracowane oddzielnie i niezależnie. Powinna je pisać inna grupa ludzi niż ta, która zajmuje się wdrażaniem systemu. Istnieje wiele czynników uzasadniających ten pogląd. Najważniejsze jest to, że ludzie związani z projektowaniem i wdrożeniem systemu opracowaliby swoje programy testowe dopiero po opracowaniu systemu, podczas gdy oddzielna grupa "programistów testów" może opracować programy testowe w tym samym czasie, w którym opracowuje swoje programy grupa wdrożeniowa. Tego rodzaju praca jednoczesna pozwala również uzyskać na czas gotowe programy testujące. Innym powodem opracowania programów testujących przez inną grupę jest to, że testy opracowane przez samych programistów będą prawdopodobnie bardzo proste. Zasugerowani opracowanym przez siebie algorytmem nie są w stanie dostrzec rozmaitych sytuacji, które spowodują, że program nie będzie pracował prawidłowo. Zwykle będą sprawdzane tylko najbardziej oczywiste przebiegi sterowania. Warto zauważyć, że podczas testowania mogą zostać popełnione te same błędy logiczne, które powstają podczas pisania programów [2].

3.1. Etapy testowania

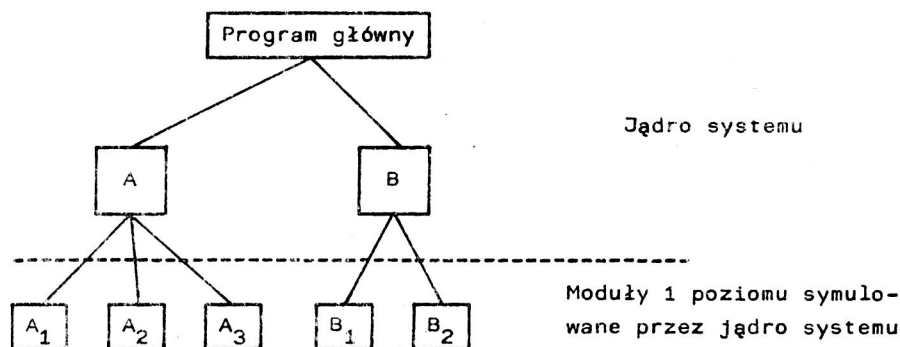
Konieczność testowania etapami wynika z dużej złożoności oprogramowania węzła. Istnieją dwie zasadnicze organizacje testowania: "dół-góra" (bottom-up) oraz "góra-dół" (top-down). Sposób "dół-góra" składa się z trzech etapów: testowania modułów, testowania podsystemów i testowania systemu. Testowanie modułów obejmuje każdy moduł z osobna. Testowanie podsystemu oznacza sprawdzenie powiązań między modułami. Wreszcie testowanie systemu, to testowanie całego konglomeratu programów. Sposób "góra-dół" oznacza wydzielenie programu głównego oraz jednego lub dwóch poziomów podprogramów jako jądra systemu i sprawdzenie poprawności działania jądra. Jest oczywiste, że moduły niższych poziomów będą modułami fikcyjnymi. Etapy testowania metodą "góra-dół" ilus-

trują rysunki 1 i 2. Ze względu na liczne wady obu sposobów testowania są one na ogół stosowane łącznie.



Rys.1. Testowanie metodą "góra-dół" - etap I

Fig.1. "Top-down" testing - stage I



Rys.2. Testowanie metodą "góra-dół" - etap II

Fig.2. "Top-down" testing - stage II

3.2. Uruchamianie i testowanie modułów programowych

Obecnie zostanie podanych wiele uwag i wskazówek dotyczących uruchamiania i testowania modułów.

Przygotowanie nośnika z programem źródłowym jest na ogół wykonywane nie przez samego programistę.

Konieczne jest zatem sprawdzenie poprawności przygotowania nośnika.

Pierwsze uruchomienie programu kończy się zwykle jednym z trzech błędów:

a) program kończy swój przebieg pod nieokreślonym adresem,

- b) program wpada w nieskończoną pętlę,
- c) program coś robi, ale nie bardzo wiadomo co.

Programy obliczeń naukowo-technicznych, o przeznaczeniu do handlu, statystyki itp., prawie zawsze produkują łatwe do identyfikacji dane wyjściowe (zbiory na nośnikach, wyniki na drukarce). Wystąpienie jednego z błędów a, b, c jest zwykle nietrudne do wykrycia. W razie oprogramowania węzła sieci komputerowej praca programów polega na sterowaniu pracą innych programów w zależności od warunków na wejściach i aktualnego stanu wspólnych danych programowych. Celowe byłoby stworzenie metody śledzenia przebiegu programu w postaci wydruków kontrolnych. Wydruki takie, oprócz śladu sterowania, dostarczałyby również informacji o zawartości rejestrów i danych programowych. Ślady dostarczają wystarczającej informacji, aby znaleźć błąd. Ich najcenniejszą zaletą jest jednak umożliwienie programiście prześledzenia wydruków w ciszy i spokoju, a nie przy maszynie, która w dodatku stoi bezczynnie przez długi czas w oczekiwaniu na jego działanie.

Dla każdego większego programu można użyć specjalnego modułu testującego, którego zadaniem byłoby symulowanie zdarzeń i przychodzących danych oraz drukowanie śladu programu sprawdzanego. Do węzła została zastosowana ta właśnie metoda.

Problem wystarczającej liczby testowań nie jest rozwiązany. Można uznać, że przed zakończeniem testowania każda instrukcja powinna być wykonana przynajmniej raz. Dane testowe powinny zapewnić wypróbowanie wszystkich sprawdzeń poprawności danych. Należy przetestować każdy skok. Współdziałające programy powinny być przetestowane za pomocą danych powodujących drobiazgowe zbadanie współdziałania.

Należy testować trzy rodzaje sytuacji: normalną, krańcową i wyjątkową. Sytuacje te powinny wymyślać osoby testujące, a nie sami programiści.

Testowanie modułów może być łatwiejsze, gdy umieści się dane w module najwyższym w hierarchii. Zapewnia to łatwiejsze testowanie, ponieważ wszystkie dane są wtedy zewnętrzne w stosunku do modułów znajdujących się niżej. Innym sposobem jest stosowanie instrukcji nadających wartości początkowe. Sposób ten pozwala uniknąć problemów z faktycznym przekazywaniem wartości modułowi. Następnie należy przetestować błędy mogące wystąpić podczas przekazywania.

Wszelkie poczynania podczas uruchamiania i testowania programów należy notować. Zapamiętanie wszystkich popełnionych błędów jest niemożliwe, a ich wykaz jest bardzo przydatny podczas uruchamiania kolejnego modułu. Notowanie czynności, wbrew pozorom, skraca efektywny czas uruchamiania. Mimo swych zalet jest na ogół lekceważone jako zajęcie niepoważne.

Korzystanie wyłącznie z dynamicznego programu uzdatniającego ODT jest zajęciem żmudnym i nieefektywnym. Po pierwsze: sprawdzanie rozgałęzień w programie jest dokonywane po omacku, zmiany zawartości komórek pamięci i rejestrów nie mogą być w żaden sposób uchwycone. Po drugie: w momencie błędnego zakończenia programu na ogół zawiesza się system. Sprawdzenie powodu błędu jest wówczas niemożliwe. Wreszcie po trzecie: po kilku godzinach ślęczenia nad klawiaturą programista nie jest już w stanie podjąć bardziej sensownej decyzji niż wyłączenie maszyny. Z drugiej jednak strony dla niewielkich programów ODT jest bez wątpienia bezcenną pomocą w uruchamianiu.

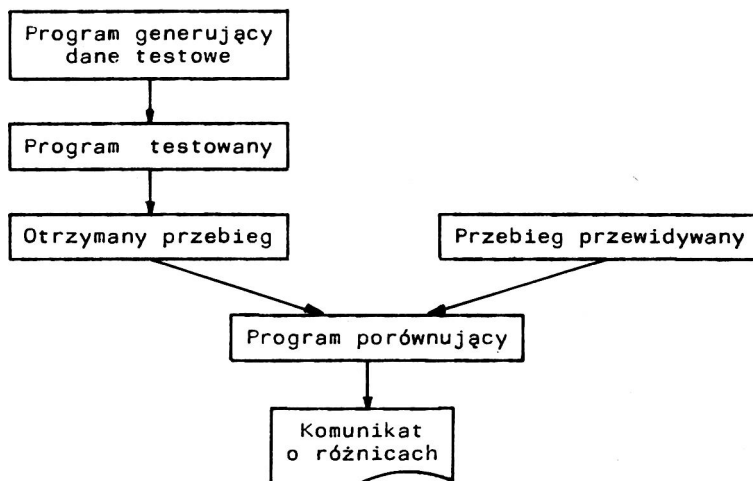
Naturalną tendencją w testowaniu programów jest automatyzacja testowania. Umożliwia ona wykonywanie maksimum testów bez udziału człowieka. Najważniejszym argumentem jest tu fakt, że ręczna kontrola błędów sama staje się ich źródłem! Ponadto ręczne testowanie przebiega wolniej i jest bardziej żmudne. Kilka metod automatycznego testowania opisano poniżej [4].

1. Automatyczna generacja danych testowych

Jej zadaniem jest dostarczenie dużej liczby różnorodnych danych. Przykładowymi danymi mogą być: napływ zgłoszeń wraz z parametrami, ramki o różnych długościach i różnych zawartościach, przychodzące przerwania itp. Konieczny jest zatem program produkujący niezależne dane testowe.

2. Automatyczna kontrola wyników

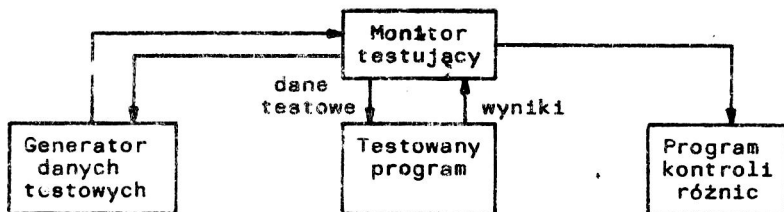
Dla poszczególnych wektorów danych wejściowych należy określić przewidywany przebieg programu i dane wyjściowe. Specjalny program będzie wykrywał różnice i wyprowadzał na drukarkę odpowiednie komunikaty (rys.3).



Rys.3. Automatyczna kontrola wyników
Fig.3. Automatic checking of results

3. Automatyczne sterowanie testowaniem (monitor testujący)

Program sterujący testowaniem pełni rolę programu głównego. Formuje dane testowe, podaje je do programu testowego, a następnie sprawdza poprawność wyników. Można pokusić się o zbudowanie programu uniwersalnego dla wszystkich modułów (rys. 4).

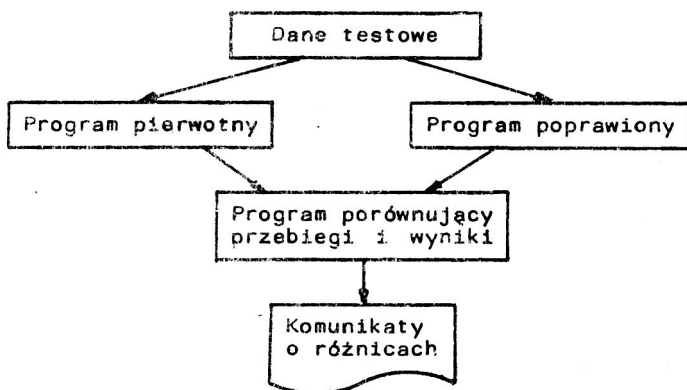


Rys.4. Automatyczne sterowanie testowaniem

Fig.4. Automatic control of testing

4. Automatyczne testowanie wielokrotne

Po wprowadzeniu najmniejszej poprawki program może pracować zupełnie inaczej. Korzystne będzie zatem porównanie pracy dwóch ostatnich wersji programu (rys. 5).



Rys.5. Powtórne testowanie

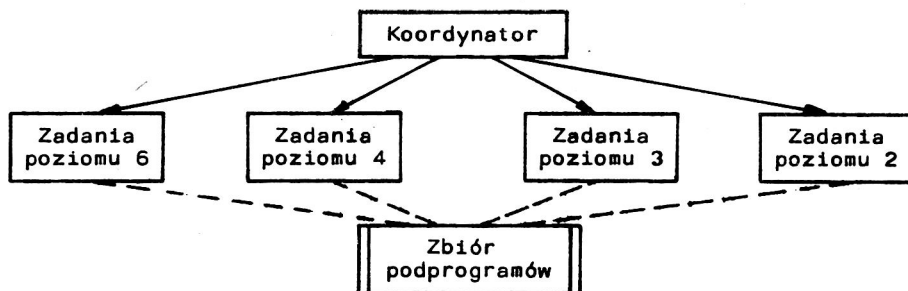
Fig.5. Repeated testing

Dla niektórych modułów programowych węzła należy zachować ostrożność w stosowaniu metod 2-4. W uwarunkowaniach czasowych jeden moduł może mieć kilka różnych przebiegów przy tych samych danych wejściowych. Dla takich modułów należy wyodrębnić podzbiór danych wejściowych, odpornych na zależności czasowe.

Niektóre dotychczasowe rozważania zostały potwierdzone w praktyce podczas realizacji programu koordynatora węzła. Następny rozdział jest poświęcony opisowi problemów napotkanych podczas uruchamiania i testowania programu koordynatora.

4. PRZYKŁADOWY PRZEBIEG URUCHAMIANIA I TESTOWANIA

Schemat ogólny oprogramowania węzła pokazano na rys. 6 [1].



Rys.6. Schemat ogólny oprogramowania węzła

Fig.6. Schematic diagram of node software

Poziom oznacza tu priorytet ustawiony w słowie stanu procesora. Dla testowania zadań poszczególnych poziomów zastosowano metodę "dół-góra", natomiast dla programu koordynatora metodę "góra-dół". W celu uruchomienia koordynatora napisano osobny moduł testujący, symulujący środowisko pracy programu koordynatora, po czym przystąpiono do uruchamiania programu. Przedstawienie kolejnych faz pracy będzie najlepszą ilustracją rozważań zawartych w poprzednich rozdziałach.

25.04.80 Pierwsza próba uruchomienia. Program kończy się błędnie. Próby poprawek nie powiodły się.

29.04.80 Wykryto przestawienie stron podczas przygotowywania nośników. Ze względu na pominięcie ręcznego etapu sprawdzania programu stracono kilka godzin na próby uruchomienia błędnego programu. Po zmianie kolejności stron znów błędne zakończenie.

16.05.80 Program niszczy własny licznik rozkazów i stos. Wykryto błędne tryby adresacji. Uruchamianie pod ODT. Program kończy przebieg pod nieznanym adresem.

20.05.80 Nie osiągnięto żadnego postępu. Nowo zatrudniony programista od razu proponuje metodę wydruków kontrolnych. Propozycja spotyka się z dużą nieufnością członków zespołu. W końcu jednak został napisany program drukowania zawartości rejestrów.

- 27.05.80 Pierwsze uruchomienie programu koordynatora ze śladem. Po wykonaniu kilku stron wydruków kontrolnych program stanął. Maszynę udostępniono następnemu użytkownikowi, a wydruki przeanalizowano z dala od maszyny.
- 30.05.80 Napisano program wydruku zawartości wybranych komórek pamięci. Program znów zakończył się błędnie. Przez następne dwa tygodnie znaleziono kilkanaście bardzo prostych błędów kodowania.
- 17.06.80 Został napisany moduł testujący dla koordynatora. Po wykonaniu 40 stron wydruków program znów zakończył się błędnie. Wykryto kolejną pomyłkę w adresacji.

18.06.80 Koordynator całkowicie uruchomiony i wstępnie przetestowany.

Opisane próby świadczą o tylko pozornej trywialności uwag zawartych w rozdziale 2. Kilkadziesiąt godzin pracy było wynikiem popełnienia elementarnych błędów, z których ponad 90% polegało na błędach adresacji. Okazuje się również, że ręczne sprawdzenie kodu, wymagające żmudnej i męczącej pracy, procentuje później w postaci braku błędów podstawowych. Stosowanie śladu programu pozwoliło na zaoszczędzenie wielu godzin siedzenia nad klawiaturą. Ładowanie tablicy uaktywnień zadań różnymi danymi pozwoliło stwierdzić, że koordynator prawidłowo reaguje na sytuacje normalne i nienormalne (np. na błędny numer zadania). Moduł testujący symulował również wzajemne wywołanie zadań różnych poziomów i sprawdzał prawidłowość obsługi przerwań przez koordynator. Starannie prowadzone notatki pozwalały uniknąć strat czasu na odtwarzanie z pamięci czynności wykonywanych podczas poprzedniej próby.

5. WNIOSKI KOŃCOWE

Doświadczenia zdobyte podczas testowania programu koordynatora zostaną wykorzystane i poszerzone podczas uruchamiania kolejnych modułów. W zadaniach operujących na wielkiej liczbie danych (zadania wyprowadzania, odbioru ramek protokołu HDLC, obsługi protokołu X.25, datagramu i inne) testowanie ręczne będzie niemożliwe. Wydruki kontrolne będą bardziej ukierunkowane na prawidłowe przetwarzanie danych, natomiast wejścia do programów będą generowane w sposób automatyczny.

LITERATURA

- [1] L e w o c J. i in., Zasady przygotowania oprogramowania węzła, Raporty Centr. Oblicz. PWR., Seria SPR nr 14/80, Wrocław 1980.
- [2] v a n T a s s e l D., Praktyka programowania, WNT, Warszawa 1979.

- [3] Y o u r d o n E., Projektowanie systemów o działaniu bezpośrednim, WNT, Warszawa 1976.
- [4] Y o u r d o n E., Techniques of Program Structure and Design, Prentice-Hall, New Jersey 1975 (dostępny przekład w języku rosyjskim).

Praca wpłynęła do Redakcji 7.01.1981 r.

Po poprawianiu 3.03.1982 r.

SOFTWARE MODULE DEBUGGING AND TESTING RULES ON THE EXAMPLE OF COMMUNICATIONS SUBNETWORK NODE

Program bugs are a general phenomenon accompanying programming. The errors committed by programmers, the ways permitting to avoid or eliminate such errors and program testing methods are described. A brief example of errors detected during debugging and testing of the coordinator routine of the communications subnetwork node for the MSK computer network is also given.

Verified by Ruta Czaplińska

ПРИНЦИПЫ ЗАПУСКА И ДИАГНОСТИРОВАНИЯ ПРОГРАММНЫХ МОДУЛЕЙ ПО ОБРАЗЦУ УЗЛА ПОДСЕТИ СВЯЗИ

Рассмотрены ошибки, допущенные программистами, методы их устранения и диагностирования программ. Описываются примеры ошибок, обнаруженных в процессе кодирования и запуска программы координатора узла подсети связи сети MSK.

Проверила Малгожата Хейдрих

*Sieć komputerowa, węzeł, minikomputer,
adapter liniowy, oprogramowanie*

Marek KRÓLIK^{*)}

Oprogramowanie diagnostyczne adaptera liniowego ALS11

Podano krótki opis synchronicznego adaptera liniowego ALS11, opracowanego w Instytucie Cybernetyki Technicznej Politechniki Wrocławskiej. Omówiono jego własności oraz możliwości i sposoby testowania. Przedstawiono ogólną strukturę oprogramowania diagnostycznego adaptera i podano przykład szczegółowego modułu diagnostycznego.

1. WSTĘP

Węzeł podsieci komunikacyjnej dla Międzyuczelnianej Sieci Komputerowej jest realizowany na podstawie procesora SM-3. Producent tego minikomputera nie dostarcza adapterów liniowych, niezbędnych do realizacji podsieci komunikacyjnej. W związku z tym było konieczne zaprojektowanie i wdrożenie adaptera liniowego na potrzeby sieci. Adapter taki (ALS11) opracowuje Instytut Cybernetyki Technicznej Politechniki Wrocławskiej opierając się na założeniach funkcjonalnych adaptera DUP 11, produkowanego przez firmę Digital Equipment Corporation [1,2].

Centrum Obliczeniowe Politechniki Wrocławskiej, które przygotowuje węzeł podsieci komunikacyjnej, nie ma dostępu do standardowego oprogramowania diagnostycznego firmy DEC. Dlatego jest konieczne przygotowanie własnego systemu diagnostycznego dla adaptera ALS11.

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

2. OPIS ADAPTERA LINIOWEGO

Adapter liniowy synchroniczny ALS11 służy do komunikacji dwukierunkowej procesora węzłowego z innymi węzłami sieci oraz z urządzeniami końcowymi transmisji podłączonymi do tego węzła [3].

Adapter ALS11 ma interfejs zgodny z zaleceniem CCITT V.24 [4]. Wyjściowe znaki z magistrali UNIBUS [1] są przesyłane w postaci równoległej do ALS11, skąd przez modem są przenoszone szeregowo do linii komunikacyjnych. Wejściowe znaki przychodzące z modemu są przenoszone do ALS11 i mogą być dostępne dla programu. Skompletowanie znaku w odbiorniku adaptera ALS11 powoduje wytworzenie sygnału przerwania wywołującego odpowiedni program obsługi. Interfejs pomiędzy procesorem a adapterem ALS11 tworzą wymienione poniżej rejestry adaptera dostępne dla programu:

- rejestr stanu nadajnika TxCSR,
- bufor danych nadajnika TxDBUF,
- rejestr stanu odbiornika RxCSR,
- bufor danych odbiornika RxDBUF,
- rejestr stanu parametrów PARCSR.

2.1. Współpraca SM-3 z adapterem ALS11 pracującym w trybie nadajnika

Część nadawcza adaptera jest konwerterem kodu równoległego na szeregowy dla danych dostarczanych z minikomputera do magistrali UNIBUS. Część ta może wykonywać poniższe podstawowe operacje:

A. Inicjacja połączenia

Przed transmisją należy nawiązać kontakt z modemem [4]. Po wymianie sygnałów z modemem program obsługi przerwań z adaptera powinien ustawić bit umożliwiający transmisję (SEND) w rejestrze TxCSR. Następnie program ustawia bit inicjujący transmisję (TSOM) w rejestrze TxDBUF.

Nadajnik będzie nieaktywny przez czas nadawania dwóch bitów, po czym zapala bit "nadajnik aktywny" (TxACT) i bit gotowości bufora danych nadajnika do wysłania nowego znaku (TxDONE) w rejestrze TxCSR. Do modemu będzie wysłana szeregowo sekwencja FLAG (0111110), rozpoczynająca początek przesyłanej ramki danych [3].

B. Transmisja znaków ramki

Gdy bit TxDONE zostanie ustawiony po rozpoczęciu wysyłania sekwencji FLAG, program obsługi przerwań z adaptera powinien wyzerować wspomniany bit inicjujący transmisję w rejestrze TxDBUF i wprowadzić pierwszy znak ramki do rejestru danych TxDBUF. Wpisanie znaku do rejestru TxDBUF spo-

woduje wyzerowanie bitu TxDONE. Bit TxDONE zostanie ponownie ustawiony, gdy bieżący znak zacznie być transmitowany w linię. Program obsługi przerwań z adaptera kontaktuje się z rejestrem TxDBUF tylko w odpowiedzi na ustawienie bitu TxDONE.

C. Przypadek nieudanej transmisji

Program obsługi przerwań musi wpisywać następny znak do rejestru TxDBUF w czasie nadawania poprzedniego znaku. W przeciwnym razie zostanie ustawiony odpowiedni bit w rejestrze TxCSR, wskazujący na błąd przekroczenia czasu. Gdy zajdzie ten przypadek, wówczas adapter wysyła w linię tzw. sekwencję ABORT (11111111). Dalsze sekwencje ABORT będą wysyłane aż do chwili, gdy program obsługi przerwań ustawi bit inicjujący transmisję, aby rozpocząć wysyłanie nowej ramki, lub wyzeruje bit SEND. Po wyzerowaniu bitu SEND zostanie wysłana bieżąca sekwencja ABORT, po czym linia przejdzie do stanu nieaktywnego.

D. Zakończenie transmisji ramki

Gdy program obsługi przerwań wyprowadzi ostatni znak danych do rejestru TxDBUF, a bit TxDONE zostanie ustawiony sygnalizując rozpoczęcie wysyłania tego znaku w linię, wówczas program ten powinien ustawić bit końca transmisji (TEOM), aby poinformować nadajnik ALS11 o zakończeniu wysyłania danych. Dalsze wydarzenia zależą od programu.

E. Przerwanie połączenia

Jeśli program obsługi przerwań żąda przerwania transmisji, to powinien wyzerować bit SEND po ustawieniu bitu TEOM. Nadajnik ALS11 wysyła ostatnią sekwencję, następnie wysyła dwa bloki sekwencji kontrolnych (CRC) i końcową sekwencję FLAG. Linia wtedy przejdzie w stan nieaktywny. Gdy stan ten będzie trwał połowę czasu nadawania jednego bitu, wyzerowany zostanie bit TxACT, a TxDONE zostanie ustawiony, wskazując, że cała ramka została wysłana w linię.

F. Transmisja nowej ramki

Jeżeli program żąda transmisji ramki po ramce wysłanej poprzednio (przy czym ramki te mają być przedzielone znakiem FLAG), to powinien czekać na ustawienie bitu TxDONE wskazującego, że rozpoczyna się wysyłanie bloku kontrolnego. Następnie program powinien wyzerować bit TEOM i ustawić bit TSOM, co wyzeruje TxDONE. TxDONE zostanie zapalony, gdy rozpocznie się wysyłanie znaku FLAG. Program powinien zerować TSOM i wyprowadzić pierwszy znak nowej ramki do rejestru TxDBUF. Ta procedura zainicjuje obliczanie CRC dla nowej ramki.

G. Działanie łącza w przerwach między ramkami

Jeżeli wymaga się nadawania ciągu znaków FLAG pomiędzy ramkami, to bit TEOM powinien pozostać zapalony po zapaleniu TxDONE.

2.2. Współpraca SM-3 z adapterem ALS11 pracującym w trybie odbiornika

Część odbiorcza adaptera wykonuje konwersję szeregowo-równoległą 8-bitowych bajtów danych oraz wylicza i kontroluje blok znaków kontrolnych. Część ta może wykonywać następujące podstawowe operacje:

A. Inicjacja połączenia

Po wymianie sygnałów z modemem program obsługi adaptera ustawia parametry w rejestrze PARCSR. Odbiornik staje się aktywny, jeśli program ustawi bit zezwolenia odbioru (RCVEN) w rejestrze RxCSR.

B. Rozpoczęcie odbioru

Wielokrotne znaki FLAG na początku ramki są ignorowane. Ramka rozpoczyna się po końcu ostatniego inicjującego znaku FLAG. Gdy pierwszy znak zostanie załadowany do rejestru danych odbiornika RxDBUF, odbiornik zapali bit "początek komunikatu" (RSOM) w rejestrze RxDBUF oraz bit "odbiornik aktywny" (RxACT) i bit "znak w buforze danych odbiornika" (RxDONE) w rejestrze RxCSR.

Program obsługi adaptera powinien odczytać zawartość RxDBUF i zapisać ją do bufora wejściowego danej linii w odpowiedzi na zapalenie bitu RxDONE, co wyzeruje RxDONE (czytanie RxDBUF przez program obsługi przerwań zawsze zeruje RxDONE). Bit RSOM zostanie wyzerowany, gdy następny znak danych zostanie zmontowany lub jeśli zostanie odebrana sekwencja FLAG lub ABORT.

C. Opóźnienie w odbiorze znaku

Następne znaki przychodzące z linii będą ładowane do rejestru RxDBUF i dostępne dla programu przez zapalenie RxDONE. Jeśli program nie przeczyta RxDBUF w czasie, gdy zakończy się zestawienie następnego znaku, to zapali się bit przekroczenia czasu w rejestrze RxDBUF.

D. Zakończenie odbioru ramki

Odbiornik ALS11 rozpozna koniec ramki, gdy odbierze końcowy znak FLAG. Bit "odbiornik aktywny" zostanie wyzerowany i zostanie zapalony bit "koniec komunikatu" w rejestrze RxDBUF. Zostanie jak zwykle zapalony bit RxDONE i jeśli CRC wskazuje błąd, wtedy także zostanie ustawiony bit błąd CRC w rejestrze RxDBUF.

E. Odebranie błędnej ramki

Program obsługi adaptera powinien po odbiorze ramki sprawdzać stan bitu "błąd CRC" w rejestrze RxDBUF. Jeśli bit ten jest ustawiony, program powinien zignorować bufor danych.

F. Czytanie następnej ramki

Odbiornik ALS11 może przyjmować następne ramki natychmiast po odebraniu pojedynczego lub wielokrotnego znaku FLAG.

Oprogramowanie diagnostyczne adaptera liniowego ALS11

G. Rozpoznanie sekwencji ABORT

Odbiornik ALS11 może odebrać sekwencję ABORT (osiem jedynek), gdy pojawia się ona w środku ramki. Odebranie sekwencji ABORT jest równoznaczne z rozłączeniem odbiornika. Program obsługi przerwań powinien odrzucić częściowo zmontowaną ramkę. Odbiornik ALS11 powraca do poszukiwania synchronizującej flagi.

3. ORGANIZACJA OGÓLNA OPROGRAMOWANIA DIAGNOSTYCZNEGO

3.1. Podstawowe założenia dla oprogramowania diagnostycznego

Projektując organizację ogólną oprogramowania diagnostycznego przyjęto następujące założenia:

1. System diagnostyczny powinien umożliwiać łatwe i dokładne zbadanie adaptera liniowego oraz zlokalizowanie błędów z dokładnością do podzespołu adaptera ALS11.

2. System diagnostyczny powinien być łatwy do rozbudowania, ponieważ oprogramowanie diagnostyczne jest przygotowywane jednocześnie z opracowywaniem adapterów liniowych. W związku z tym brak jest obecnie doświadczeń praktycznych dotyczących potrzeb diagnostycznych tego urządzenia. Gdy takie potrzeby zostaną rozpoznane, wówczas prawdopodobnie zajdzie potrzeba włączenia nowych procedur diagnostycznych do systemu oprogramowania diagnostycznego.

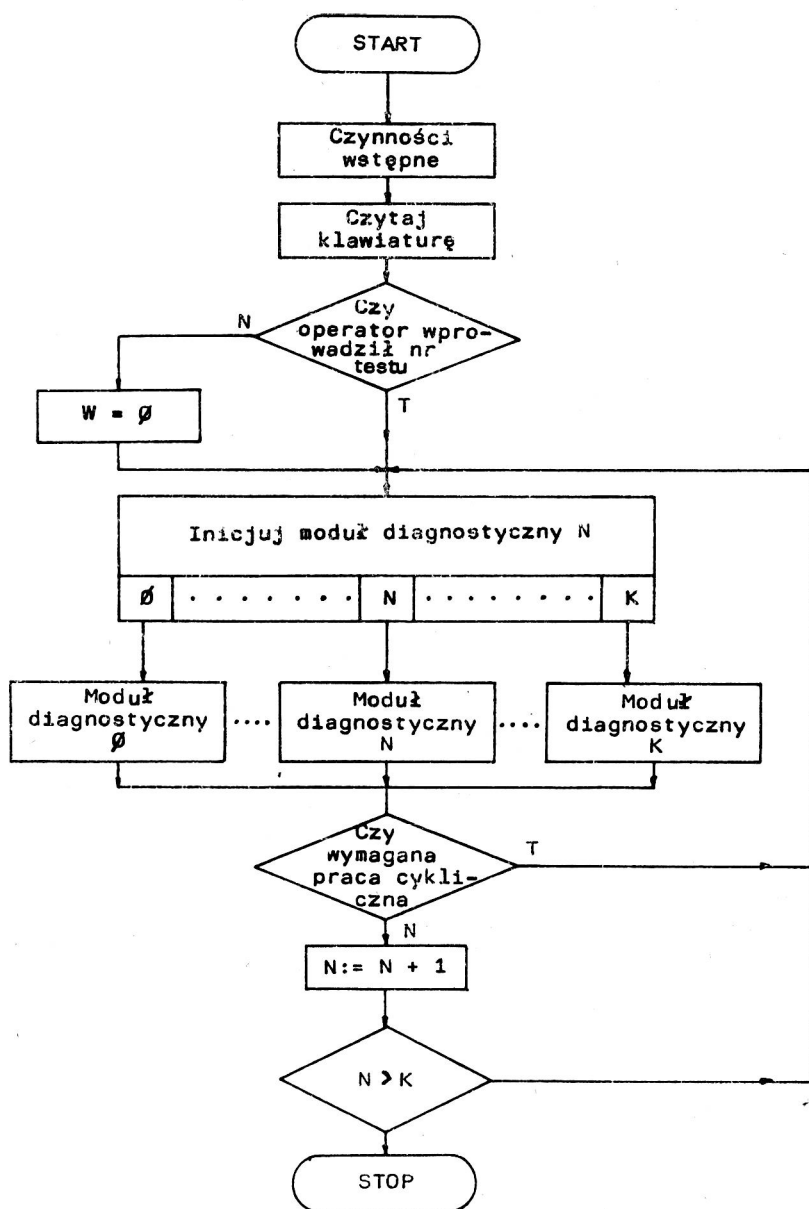
3. System diagnostyczny ma być przydatny zarówno do stwierdzenia poprawności działania adaptera liniowego, jak i do lokalizacji miejsca uszkodzenia, np. przez obserwację przebiegów w charakterystycznych punktach adaptera.

3.2. Struktura ogólna

Realizując podane założenia zaprojektowano strukturę ogólną oprogramowania diagnostycznego, przedstawioną na rys. 1. Oprogramowanie diagnostyczne składa się z modułu pętli głównej oraz z K+1 szczegółowych modułów diagnostycznych.

Po uruchomieniu systemu diagnostycznego następuje wykonanie czynności wstępnych (ustawienie wektorów przerwań, zerowanie liczników błędów oraz liczników poprawnych przebiegów), po czym program czyta klawiaturę z pulpitu operatora. Następnie zostanie zainicjowane wykonywanie modułu diagnostycznego N (jeśli operator określił numer N) lub moduł 0 (w przeciwnym razie).

Uruchomiony moduł wykonuje wymaganą procedurę diagnostyczną, po czym kończy pracę oddając sterowanie do pętli głównej.



Rys.1. Organizacja ogólna oprogramowania diagnostycznego
Fig.1. General structure of diagnostic software

Oprogramowanie diagnostyczne adaptera liniowego ALS11

Jeśli moduł diagnostyczny wykryje błąd, to zarejestruje go w odpowiednim liczniku błędów i zatrzyma wykonywanie programu lub odda sterowanie do pętli głównej zależnie od tego, czy operator ustawił tryb zatrzymania po błędzie. Praca w trybie bez zatrzymywania po błędzie pozwala na obserwację przebiegów różnych sygnałów w adapterze liniowym ALS11, nawet w razie błędnego działania adaptera. Powinno to być szczególnie pomocne podczas uruchamiania i badań prototypów adaptera liniowego.

Po otrzymaniu sterowania od modułu diagnostycznego N pętla główna sprawdza, czy operator ustawił tryb cyklicznej pracy modułu testowego M. Jeśli tak, to moduł ten zostanie zainicjowany ponownie. W przeciwnym razie zostanie zainicjowany następny moduł diagnostyczny, jeżeli moduł N nie był ostatnim modulem K.

Ten ostatni tryb pracy pozwala na łatwe sprawdzenie poprawności działania adaptera liniowego przez wykonanie ciągu modułów diagnostycznych.

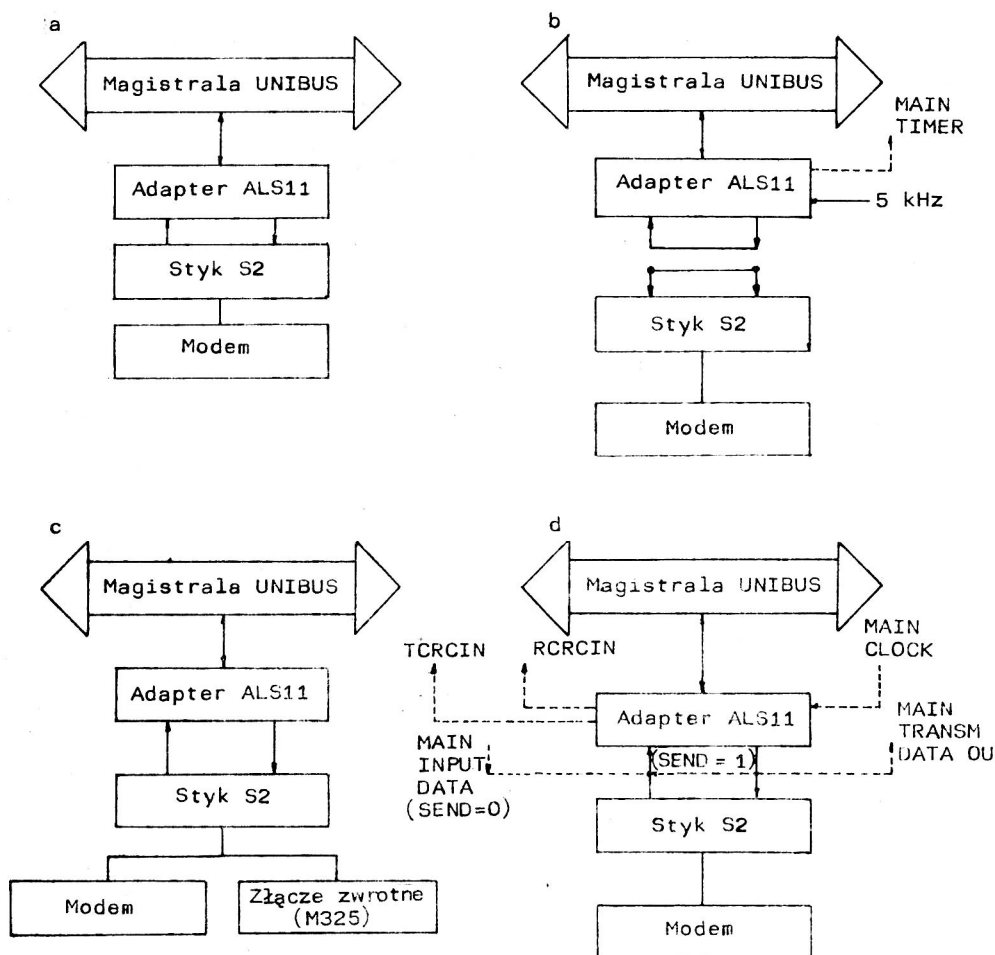
Przedstawiona struktura ogólna pozwala na łatwe rozbudowywanie systemu diagnostycznego przez zmiany modułów diagnostycznych lub dołączanie nowych modułów diagnostycznych. W razie dołączania nowych modułów diagnostycznych nie jest wymagana kompilacja całego systemu diagnostycznego. Można osobno skompilować i przebadać nowy moduł diagnostyczny, a następnie dołączyć go do systemu oprogramowania diagnostycznego w fazie konsolidacji tego systemu (za pomocą programu konsolidatora LINK).

4. SZCZEGÓŁOWE MODUŁY DIAGNOSTYCZNE

4.1. Rodzaje modułów diagnostycznych

Adapter ALS11 może pracować podczas testowania w układach pokazanych na rys. 2. W zależności od trybu pracy adaptera podczas testowania, moduły diagnostyczne możemy podzielić na testy wewnętrzne, testy zewnętrzne jednomaszynowe i testy dwumaszynowe.

Testy zewnętrzne pracują bez rozłączania modemu, sprawdzają jedną linię, podstawa czasu jest symulowana programowo i nie używają przerwań. Sprawdzają one poprawność działania adaptera podczas nadawania i odbioru oraz symulują różne możliwe sytuacje, jakie mogą wystąpić podczas pracy adaptera. Test nadawania wysyła informację z bufora wyjściowego i - obserwując wyjście nadajnika - składa ramkę w buforze wejściowym. Po nadaniu i odebraniu każdej sekwencji ośmiobitowej, zawartości obu buforów są porównywane ze sobą, przy czym niezgodność oznacza błąd. Ramka może mieć różną długość i może zawierać różne zestawy informacji. Test



Rys.2. Tryby pracy adaptera ALS11: a - tryb normalny, b - tryb testowy, c - tryb zewnętrzny, d - tryb wewnętrzny

Fig.2. ALS11 Adapter Operating Modes: a - normal operation, b - system test, c - external maintenance, d - internal maintenance

odbioru podaje informacje z bufora wyjściowego na wejście odbiornika, obserwuje stan odbiornika, składa ramkę w buforze odbioru i porównuje z nadaną. Ponadto testy wewnętrzne sprawdzają poprawność wysłania i odbioru sekwencji ABORT, symulują opóźnienia w nadajniku i odbiorniku, obserwując w tym czasie zachowanie się adaptera oraz sprawdzają poprawność obliczania bloku kontrolnego CRC.

Do grupy testów wewnętrznych należą również testy systemowe. W testach tych podstawa czasu jest generowana przez wewnętrzny zegar adaptera. Można nimi sprawdzać prawidłowość działania mechanizmu przerwań.

Testy zewnętrzne pracują w zewnętrznym trybie technicznym z wymuszoną podstawą czasu. Wymagają odłączenia modemu i włączenia specjalnego złącza zwrotnego. Testy te operują bajtami i mogą pracować z przerwaniami lub bez przerwań.

Testy dwumaszynowe pracują w trybie normalnym na dwóch maszynach połączonych linią przez modemy. Działają według programu identycznego dla dwóch maszyn. Operują na poziomie ramek, sprawdzają zgodność transmisji w pętli zamkniętej (nadawanie-odbiór-porównanie) w dwóch kierunkach (łącze symetryczne). Wykrywają także typowe błędy transmisji.

W następnym punkcie podano przykład rozwiązania szczegółowego modułu diagnostycznego.

4.2. Moduł diagnostyczny nadawania ramki

Moduł diagnostyczny nadawania ramki sprawdza poprawność działania nadajnika ALS11 podczas wysyłania ramki. Po uruchomieniu moduł ten ustawia kolejno bity w rejestrach adaptera w ten sposób, aby umożliwić inicjację połączenia, transmisję znaków ramki oraz zakończenie transmisji. Tak więc w module tym znajduje się program obsługi nadajnika. Ponieważ jest to test wewnętrzny, podstawa czasu jest symulowana programowo przez ustawianie i zerowanie bitu zegara testowego w rejestrze TxCSR oraz przez wprowadzanie opóźnień czasowych. W module tym obserwuje się dokładnie zachowanie nadajnika adaptera w czasie jego pracy. Duża szczegółowość sprawdzania poprawności działania adaptera możliwa jest dzięki programowej symulacji podstawy czasu. Po każdym takcie (impulsie) zegara następuje sprawdzenie prawidłowości ustawienia stanów rejestrów adaptera, charakterystycznych dla tej chwili. W razie stwierdzenia niezgodności następuje zarejestrowanie błędu w odpowiednim liczniku błędów i zatrzymanie programu lub dalsze jego wykonywanie, zależnie od tego czy operator ustawił tryb zatrzymania po błędzie. Oprócz tego jest śledzone szeregowe wyjście nadajnika. Sprawdza się, czy jest prawidłowo wysyłana początkowa sekwencja szesnastu zer, czy sekwencja FLAG jest prawidłowa, czy znak wychodzący w linię jest identyczny z nadanym oraz czy prawidłowo są wprowadzane bity o wartości zero po ciągach pięciu jedynek w danych wysyłanych w linię.

Rozpatrywany moduł diagnostyczny wysyła w linię ramkę długości 256 znaków, zakończoną sekwencją FLAG. Do modułu tego możemy wprowadzić parametry z układu sterowania. Parametrami są: typ ramki, tzn. rodzaj informacji wysyłanej w linię, oraz parametr sterujący wysyłaniem ramki

"bez końca", tzn. wysyłanie danych w linię w celu ułatwienia obserwacji przebiegów sygnałów w adapterze w czasie nadawania.

5. UWAGI KOŃCOWE

Omówiony w artykule system diagnostyczny powinien umożliwić stosunkowo łatwe, a przy tym dokładne badanie adapterów liniowych ALS11. Modułowa struktura systemu pozwala na łatwe rozszerzanie jego funkcji. Praca cykliczna pozwala na obserwowanie na ekranie oscyloskopu kształtu przebiegów istotnych sygnałów w adapterze liniowym. Jest to możliwe również wtedy, gdy pracuje on nieprawidłowo. Cecha ta powinna być szczególnie cenna w badaniach prototypów adaptera.

LITERATURA

- [1] PDP-11 Peripherals Handbook, Digital Equipment Corporation, Maynard, Massachusetts, 1975.
- [2] B a t y c k i T. i in., Adapter liniowy synchroniczny ALS11. Dokumentacja techniczna (wersja poprawiona), Raporty Inst. Cybern. Pwr., Seria SPR nr 26/81, Wrocław 1981.
- [3] Proposed Revision to Level 1 and 2 of Recommendation X.25 as Approved by Working Party, 2-5 April 1979, CCITT - October 1979.
- [4] Orange Book, vol. VIII, CCITT, Genewa 1976.

Praca wpłynęła do Redakcji 13.12.1980 r.
Po poprawieniu 3.03.1982 r.

DIAGNOSTIC SOFTWARE FOR SYNCHRONOUS LINE ADAPTER ALS11

A short description of a synchronous line adapter developed by Institute of Cybernetics of the Wrocław Technical University is given. Its basic operation which can be performed and ways of testing are discussed. The general structure of diagnostic software for the ALS11 adapter is discussed and an exemplary specific diagnostic module described.

Verified by Ruta Czaplińska

ДИАГНОСТИЧЕСКОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЛИНЕЙНОГО АДАПТЕРА ALS11

В работе дается краткий обзор линейного синхронного адаптера ALS11, разработанного Вроцлавским техническим университетом, а также основных операций и методов диагностики. Описывается общая структура программ диагностирования адаптера ALS11, а также пример конкретной диагностической модули.

Проверила Малгожата Хейдрих

*Sieć komputerowa, podsieć komunikacyjna,
węzeł, odnowa informacyjna,
miernictwo i diagnostyka*

Józef LEWOC^{*)}, Witold KOMOROWSKI^{*)}

Problematyka badawcza węzła podsieci komunikacyjnej

Przedstawiono niektóre tematy badawcze, które wyłoniły się podczas projektowania i budowy oprogramowania węzła podsieci komunikacyjnej MSK. Większość proponowanych zadań zmierza do poprawy jakości działania podsieci przez optymalizację przyjętych rozwiązań programowych i założonych parametrów protokołów. Naszkicowano również tematykę badań i prac implementacyjnych dotyczących tworzenia narzędzi projektowych dla systemów operacyjnych czasu rzeczywistego.

1. WSTĘP

Rozwiązania różnych problemów projektowych stosowane w pierwszej wersji węzła podsieci komunikacyjnej dla Międzyuczelnianej Sieci Komputerowej MSK przyjmowano w większości przypadków na podstawie intuicji inżynierskiej, popartej niekiedy uproszczoną analizą. Podstawowe przyczyny takiego podejścia były następujące:

- krótki czas wyznaczony na przygotowanie i uruchomienie węzła (1 rok);
- konieczność podziału zadania (przygotowanie i uruchamianie oprogramowania węzła) na podzadania realizowane przez kilka grup wykonawców działających w pewnym stopniu autonomicznie; wynikła stąd konieczność dokonania ustaleń zasad przygotowania oprogramowania węzła, które są

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

niezbędne w takim sposobie realizacji dużych systemów oprogramowania we wczesnej fazie prac, gdy zespół prowadzący te prace ma małe doświadczenie praktyczne w zakresie realizowanych tematów (pierwszą wersję zasad opracowano w styczniu 1980, a wersję poprawioną i rozszerzoną wydano w kwietniu tego roku [1,2]);

- brak w zespole przygotowującym węzeł podsieci komunikacyjnej osób mających doświadczenia praktyczne na temat zagadnień sieciowych;
- brak doświadczeń praktycznych w eksploatacji sprzętu jaki ma być wykorzystywany do zbudowania podsieci komunikacyjnej;
- brak skutecznych metod analitycznych pozwalających na rozwiązywanie problemów napotykanych podczas projektowania węzła podsieci komunikacyjnej.

Podczas projektowania oprogramowania węzła przyjmowano na ogół zasadę stosowania w pierwszej wersji rozwiązań prostych, a więc z dużym prawdopodobieństwem skutecznych. W rozwiązaniach, które budziły największą wątpliwość (np. sposób wyboru drogi [3]), zwracano szczególną uwagę na to, by ewentualna ich zmiana była w przyszłości jak najłatwiejsza.

W niniejszym artykule zostaną przedstawione pewne problemy projektowe, dla których przyjęto rozwiązania na podstawie intuicji projektantów, a które - jak sądzimy - wymagają dalszych prac badawczych. Powinny to być prace zarówno analityczne, jak i doświadczalne, przez co rozumiemy badania symulacyjne i badania przeprowadzane na rzeczywistym, złożonym narzędziu komputerowym, jakim jest przygotowany węzeł podsieci komunikacyjnej.

Pracom badawczym w omawianym temacie należy postawić następujące cele:

- A. Zbadanie jakości rozwiązań przyjętych dla podsieci komunikacyjnej MSK.
- B. Optymalizacja podsieci komunikacyjnej ze względu na uzasadnione kryteria.
- C. Opracowanie i zbadanie propozycji metod projektowania sieci komputerowych, przydatnych przede wszystkim w warunkach krajowych.
- D. Ocena jakości międzynarodowych zaleceń dla sieci komputerowych, optymalizacja przyjętych protokołów i opracowanie ewentualnych uściśleń i(lub) modyfikacji dla takich zaleceń.

2. PODSTAWOWA CHARAKTERYSTYKA WĘZŁA PODSIECI KOMUNIKACYJNEJ

Z czterech grup charakterystyk sieci, cytowanych w pracy [4], tj.:

- jakość przesyłania informacji,
- koszt budowy kanałów i węzłów,

- koszt eksploatacji kanałów i węzłów,
- niezawodność,

zainteresowani jesteśmy głównie grupą pierwszą i grupą ostatnią. Charakterystyki z dwóch pozostałych grup są wyznaczone przede wszystkim przez wykorzystywany sprzęt, a ten w warunkach krajowych jest dobierany w praktyce prawie wyłącznie na podstawie kryterium dostępności.

W podstawowej literaturze na temat sieci komputerowych [4-7] bada się indywidualnie różne problemy projektowe, np. gospodarkę buforami, metody wyboru drogi, szeregowanie procesów, przy czym dla różnych problemów przyjmuje się różne charakterystyki. Zwykle problemy są stawiane i rozwiązywane w sposób probabilistyczny, a kryterium optymalizacji stanowi najczęściej wartość oczekiwana pewnej zmiennej losowej w stanie ustalonym. Przykładami takich zmiennych losowych mogą być opóźnienia tranzytowe, długości kolejek, czy obciążenia linii.

Dla użytkownika takie podejście wydaje się mieć dość ograniczoną wartość. Już dawno Feller zwrócił na to uwagę w podstawowej monografii z rachunku prawdopodobieństwa [8]. Podany przez niego przykład (t. 2, s. 190) wyników analizy prostego systemu kolejkowego wykazuje bardzo dużą wartość wariancji w porównaniu z wartością oczekiwaną badanych zmiennych losowych. I tak, jeśli wartość oczekiwana czasu zajętości obsługującego wynosi 5, a wariancja tej zmiennej losowej wynosi 225, to dla wartości oczekiwanej 10 wariancja wynosi już 1900. Uproszczenia wprowadzane do modelu probabilistycznego prowadzą do bardzo dużych fluktuacji charakterystycznych zmiennych losowych, znacznie wyższych niż występujące w dobrze zaprojektowanych systemach cyfrowych.

Praktyczne skutki stosowania takiego podejścia widać wyraźnie na poniższych przykładach rozwiązań w systemach komputerowych o funkcjach podobnych do przewidywanych dla sieci komputerowej MSK.

W systemie Odra 1305, pracującym pod kontrolą systemu operacyjnego GEORGE, zadania otrzymują priorytet programowy od programu High Level Scheduler na podstawie, między innymi, zapotrzebowania zadania na czas jednostki centralnej [9]. W wyniku tej strategii zadania o czasie przebiegu powyżej kilku godzin mogą przebywać w systemie przez czas wielodniowy, a nawet wielotygodniowy (na co ma wpływ także stosunkowo duża awaryjność m.c. Odra 1305). Użytkownika oczywiście mało interesuje fakt, że wartość oczekiwana czasu pobytu zadania w systemie jest dość mała w sytuacjach, w których wyniki obliczeń mogą się stać nieaktualne, gdy wreszcie zostaną uzyskane.

W systemie automatycznego przetwarzania informacji dla okręgowych dyspozycji mocy podstawowym urządzeniem dwustronnego kontaktu dyspozytorów z systemem komputerowym są semigraficzne monitory ekranowe. Z uwagi na dużą pojemność ekranu monitora (dwa tysiące znaków w kodzie ISO)

stosowano jeden bufor długi o pojemności 1 K słów oraz jeden bufor krótki dla priorytetowych komunikatów wyjściowych [10]. Bufor długi był wykorzystywany do wprowadzania informacji od dyspozytorów, jak i do wyprowadzania danych (najczęściej schematów sieci elektroenergetycznej), przy czym decyzje dotyczące przydzielenia bufora dla programu wyprowadzającego lub do realizacji tak zwanego niewymuszonego odczytu były podejmowane przez szeregowe obieganie tablicy aktualizowanych obrazów, które było uruchamiane przez sygnały przerwania z jednostki sterującej monitorów ekranowych. Przerwania te były generowane po zakończeniu wcześniej zainicjowanej transmisji i (lub) upłynięciu zadanego limitu czasowego, w którym nie była realizowana żadna transmisja. Przyjęte rozwiązanie zapewniało (gdy limit czasowy wynosił około 2,5 s) wartość średnią czasu oczekiwania dyspozytora na obsługę komunikatu wprowadzanego z klawiatury monitora mniejszą niż 5 s, ale gdy była duża liczba monitorów wyświetlających obrazy uaktualniane cyklicznie, czas ten mógł przekroczyć 20 s. Takie fluktuacje czasu oczekiwania były nie do przyjęcia przez dyspozytorów. W związku z tym skrócono limit czasowy około pięciokrotnie, decydując się na znaczące zwiększenie obciążenia jednostki centralnej, aby zapewnić ograniczenie czasu oczekiwania w warunkach najbardziej niesprzyjających do około 4 s.

Z podanych przykładów widać, że dla zastosowań sieciowych o podobnym charakterze należy określać charakterystyki inne niż wartość oczekiwana lub wyższe momenty rozkładów pewnych zmiennych losowych. Istnieje także potrzeba przyjęcia prostej charakterystyki dla różnych problemów jako wspólne kryterium optymalizacji dla rozwiązań szczegółowych węzła podsieci komunikacyjnej (zwłaszcza interesujące są te rozwiązania projektowe, dla których stosunkowo niewielkie zmiany oprogramowania i (lub) sprzętu mogą wywierać znaczny wpływ na wybraną charakterystykę węzła).

Dobrą charakterystyką sieciową jest maksymalne opóźnienie tranzytowe jakie może wystąpić dla pakietów [11] przepływających przez węzeł, tj. czas od chwili pojawienia się pierwszego znaku ramki informacyjnej w węźle do chwili skutecznego wysłania ostatniego znaku informacyjnej ramki wyjściowej [12], zawierającej pakiet odebrany i przekazywany do następnego węzła lub DTE [11] w najbardziej niesprzyjających okolicznościach. Posługując się terminologią probabilistyczną stawiamy postulat, by opóźnienie tranzytowe, tj. zmienna losowa, nazywana często czasem pobytu zgłoszenia, miała rozkład skoncentrowany na skończonym przedziale czasowym $\langle 0, T \rangle$ (lub, ogólniej, $\langle S, T \rangle$, $0 \leq S \leq T$). W ten sposób formułujemy warunek nadążności dla węzła. Jeśli powyższy postulat nie będzie spełniony, to będzie istniała możliwość realizacji tej zmiennej losowej przekraczającej dowolne skończone ograniczenie, czyli możliwość czasowego "zatkania" węzła. Optymalizacja węzła powinna prowadzić do minimaliza-

cji majoranty T powyższej zmiennej losowej. Przykłady zagadnień optymalizacji poszczególnych rozwiązań węzła ze względu na przyjęte minimum majoranty T podano w następnym rozdziale.

3. PRZYKŁADY ZAGADNIEŃ OPTIMALIZACYJNYCH WĘZŁA

3.1. Gospodarka pamięcią

W wersji węzła podsieci komunikacyjnej, zrealizowanej na minikomputerze SM-3, program wraz z danymi zajmuje około 20,5 K słów pamięci. Na buforów informacji przepływających przez węzeł lub generowanych w węźle i nadawanych w linii wyjściowej pozostaje 7,5 K słów lub 9,5 K słów po rozszerzeniu pamięci operacyjnej, dostępnej dla procesora SM-3. Zastosowano trzy rodzaje buforów:

- buforów cykliczne wypełniane przez drivery odbiorników adapterów liniowych ALS11, odpowiadających pod względem funkcjonalnym adapterom DUP-11 firmy Digital Equipment Corporation [13],
- buforów długie o pojemności pozwalającej na zmieszczenie ramki informacyjnej z najdłuższym pakietem danych według przyjętych ustaleń dla zaleceń X.25 [11] oraz niezbędnych informacji kontrolno-sterujących, wykorzystywane dla pakietów przepływających przez węzeł;
- buforów krótkie o pojemności pozwalającej na zmieszczenie ramki informacyjnej z pakietem (innym niż pakiet danych) generowanym w węźle lub ramki nieinformacyjnej wraz z niezbędnymi danymi kontrolno-sterującymi.

W przyjętym rozwiązaniu można optymalizować podział dostępnego obszaru pamięci operacyjnej pomiędzy różne rodzaje buforów. Rozwiązanie takie przyjęto ze względu na prostotę wypełniania i opróżniania buforów, niezbędne do zapewnienia nadążności węzła na przyjętym sprzęcie. Możliwe są także inne rozwiązania gospodarki buforami [14], np. używanie buforów krótkich dla pakietów wejściowych z ewentualnym przepisywaniem danych do buforów długich lub jeszcze inne. W bardziej złożonych systemach gospodarki buforami można bardziej efektywnie wykorzystać dostępny obszar pamięci, minimalizując w ten sposób T . Jednak rośnie wówczas szansa gubienia odbieranych znaków w linii, co powoduje natychmiast znaczny skokowy wzrost T ze względu na konieczność wykorzystania mechanizmu odnowy przez retransmisję danych, przewidywanego przez implementowane protokoły [11,12].

3.2. Szeregowanie podstawowych procesów w węźle

Do szeregowania procesów podstawowych, które obejmują odbieranie ramek i pakietów, ich przetwarzanie, kolejkovanie oraz inicjację wysyłania do linii wyjściowych [15,16], wykorzystuje się trzy rodzaje semaforów:

- semafor globalny informujący o potrzebie wykonania jednego lub większej liczby procesów podstawowych,
- semafony ogólne informujące o potrzebie wykonania poszczególnych procesów podstawowych,
- semafony szczegółowe informujące, której linii dotyczą wywoływane procesy podstawowe.

Mechanizm szeregowania ma duży wpływ na T . Gdyby na przykład przyjąć popularną strategię szeregowania bez wywłaszczania na zasadzie ustalonych priorytetów względnych, wówczas T dla pakietów odbieranych z ostatniej linii wejściowej lub nadawanych do tej linii wyjściowej byłoby stosunkowo duże. Z drugiej strony przekazywanie sterowania do procesu dla kolejnej linii po zakończeniu procesu dla linii poprzedniej może powodować wzrost T wskutek nadmiernego wzrostu liczby operacji "biurokratycznych", związanych z jałowym wykonywaniem operacji P według terminologii Dijkstry [5,17] dla linii nieaktywnych.

W pierwszej wersji węzła przyjęto zasadę inicjowania kolejnych procesów dla pewnej linii, dopóki dotyczą one ramek nieinformacyjnych, a więc wymagają krótkiego czasu procesora. Natomiast po zakończeniu procesu dla ramki informacyjnej następuje przejście do badania za pomocą zmodyfikowanej operacji P semaforów szczegółowych dla kolejnych linii.

Zbadanie wpływu tego mechanizmu i innych mechanizmów konkurencyjnych oraz optymalizacja mechanizmu najlepszego wymaga dalszych badań.

3.3. Optymalizacja poziomu 2 protokołu X.25

3.3.1. Dobór limitu czasowego dla retransmisji ramek

Długości limitu czasowego, kiedy nadawca powinien ponownie nadać ramkę lub grupę ramek, dla której nie otrzymał potwierdzenia od odbiorcy, jest ważnym parametrem implementowanego w sieci MSK protokołu LAPB [12]. Dla coraz krótszych limitów czasowych T może wzrastać z uwagi na zbędne retransmisje ramek. Dla dłuższych limitów czasowych T może wzrastać z powodu dłuższego czasu, po którym zostanie rozpoczęta retransmisja ramki odebranej nieprawidłowo, np. na wskutek przekłamania w linii. Optymalny dobór tego parametru wymaga dalszych badań.

3.3.2. Liczba L wolnych buforów, przy której jest wysyłana ramka RNR do aktywnej linii

Protokół LAPB pozwala na nadawanie liczby ramek informacyjnych nie większej od tzw. okna (w implementowanej wersji 7 [18]) bez otrzymania potwierdzenia od odbiorcy. Odbiorca może potwierdzić prawidłowy odbiór ramek informacyjnych, np. L ramek. Gdy L maleje, T może wzrastać z uwagi na wzrost jałowego przepływu ramek potwierdzających. Dla większych L, T może wzrastać z uwagi na niepotrzebne wstrzymywanie przepływu ramek informacyjnych w rozpatrywanej linii.

3.3.3. Sterowanie ustawianiem bitu P w ramach nadawanych

Bit P w ramach nadawanych wymusza bezzwłoczne potwierdzenie ramek odebranych [12,18]. Częstsze ustawianie bitu P może prowadzić do wzrostu T w wyniku wzrastającego przepływu w linii. Natomiast rzadsze ustawianie bitu P może powodować wzrost T na zasadzie podobnej do większego L (zob. pkt 3.3.2).

3.4. Optymalizacja poziomu 3. protokołu X.25

3.4.1. Analiza zachowania się węzła podczas sterowania wielkością okna, tj. liczby pakietów nie potwierdzonych

Optymalizacja musi uwzględniać tu co najmniej dwa czynniki: czas przebywania pakietu w węźle i zajętość buforów. Gdy jest zbyt duże okno, może następować wyczerpywanie zasobów pamięciowych, co powoduje blokowanie węzła i obniżanie jego przepustowości. Zmniejszenie okna natomiast prowadzi do wzrostu częstości potwierdzeń i zmniejszenia udziału informacji użytkowej w ogólnym strumieniu danych, przenoszonych przez sieć.

3.4.2. Minimalizacja liczby rozłączeń awaryjnych wymuszanych przez węzeł na skutek wyczerpania zasobów

Parametrami wpływającymi na liczbę rozłączeń są: liczba limitów czasowych, które można rejestrować w systemie, wielkość przydzielonych limitów czasowych i wartość limitów retransmisji. Pierwszy z tych czynników zależy od przyjętych ograniczeń w systemie operacyjnym węzła, dwa pozostałe są parametrami protokołu. Optymalny dobór wartości tych parametrów zależy w decydującym stopniu od jakości sprzętu i linii transmisyjnych. Po zmniejszeniu szybkości działania i wierności transmisji skuteczność przesyłania pakietów wymaga zwiększania limitów czasowych i retransmisji powodując - przy ograniczonych zasobach - ograniczenie liczby użytkowników.

3.5. Automatyczne włączanie(wyłączanie) adapterów liniowych

Uszkodzony adapter liniowy może być źródłem zgłoszeń, tzn. przerwań nie związanych z poprawnym przesyłaniem informacji. Na przykład awaria przerzutnika RxDONE [13], sygnalizującego odbieranie kolejnych znaków z linii, może powodować intensywny strumień zgłoszeń jałowych, które niepotrzebnie inicjują różne procesy w węźle. Dlatego w węźle następuje automatyczne wyłączanie adapterów liniowych, z których odebrano ciąg N kolejnych zgłoszeń jałowych.

Małe N może powodować zbyt wczesne wyłączanie adapterów liniowych podczas przemijających zakłóceń, jakie są obserwowane w łączach komutowanych. To oczywiście powoduje wzrost T. Duże N może powodować wzrost T na skutek obsługi zgłoszeń jałowych.

Aby umożliwić skuteczną pracę węzła bez konieczności ciągłego nadzoru operatorskiego, należy umożliwić automatyczne włączanie wyłączonych adapterów po upływie pewnego czasu P. P wywiera podobny wpływ na T co N, choć oczywiście dotyczy to dłuższych okresów pracy nienormalnej.

3.6. Automatyczna odnowa informacyjna węzła

Potrzeba pracy węzła bez ciągłego nadzoru operatorskiego zmusza do automatycznego dokonywania odnowy informacyjnej po stwierdzeniu, że węzeł pracuje nieprawidłowo. Przewidujemy zastosowanie elektronicznego urządzenia wykrywającego nieprawidłową pracę węzła i inicjującego pełną lub częściową odnowę informacyjną [19]. Parametrami, które mają wpływ na T, są częstość i liczba danych składowanych w celu umożliwienia późniejszej odnowy częściowej oraz czas, po upływie którego układ elektroniczny zainicjuje odnowę informacyjną, jeśli nie odbierze sygnału poprawnej pracy węzła.

3.7. Uwagi ogólne

Przedstawione przykłady zadań badawczych, związanych ze zrealizowaną wersją węzła podsieci komunikacyjnej, nie wyczerpują całości tematyki jaka kształtowała się w okresie studialnym i projektowym budowy podsieci komunikacyjnej. Na uwagę zasługuje zwłaszcza problem weryfikacji przyjętych protokołów, które wprowadzić są zalecane przez autorytatywne organizacje międzynarodowe i są skutecznie stosowane w wielu sieciach komputerowych, lecz kryją w sobie pewne błędy "syntaktyczne". Możliwość takich błędów, również dla protokołu X.25, wykazały np. badania Zurychskiego Laboratorium firmy IBM [20]. Analiza protokołów pod względem ich poprawności wymaga prac teoretycznych związanych między innymi z opraco-

waniem właściwych formalizmów przydatnych do opisu procedur określonych przez te protokoły (por. np. [21]). Z zadań badawczo-projektowych istotne wydaje się stworzenie języka programowania wygodnego do budowy systemów pracujących na bieżąco (real-time). Do oprogramowania węzła sieci MSK z konieczności zastosowano język Assembler maszyny SM-3. Przeniesienie oprogramowania na inne maszyny będzie wymagało powtórzenia części prac lub budowy "translatora" dla konkretnej nowej maszyny. W razie dysponowania odpowiednim językiem wyższego poziomu techniczne zadanie zamiany sprzętu przestaje być problemem jednostkowym, wymagającym przeprogramowania systemu operacyjnego węzła sieci. Języki algorytmiczne czasu rzeczywistego ułatwiają i przyspieszają zaprojektowanie oraz uruchomienie systemu operacyjnego (por. np. CONCURRENT-PASCAL [22]), a uzyskana dokumentacja systemu staje się bardziej czytelna dzięki wyższemu poziomowi abstrakcji samego języka. Pierwsze próby budowy tego typu języka i analizy występujących problemów były już czynione - w związku z zastosowaniem krajowych minikomputerów w teleprzetwarzaniu [23,24].

Podjęcie niektórych spośród wymienionych tematów jest możliwe w zespole mającym nie tylko znajomość podstaw teoretycznych, lecz również szczegółową wiedzę o przedmiocie uzyskaną podczas realizacji dzieła inżynierskiego, jakim jest system operacyjny węzła podsieci komunikacyjnej MSK.

LITERATURA

- [1] L e w o c J. i in., Zasady przygotowania oprogramowania węzła, Raporty Centr. Oblicz. PWr., Seria SPR nr 11/80, Wrocław 1980.
- [2] L e w o c J. i in., Zasady przygotowania oprogramowania węzła. Wersja poprawiona i rozszerzona, Raporty Centr. Oblicz. PWr., Seria SPR nr 14/80, Wrocław 1980.
- [3] L e w o c J., Metoda wyboru drogi w sieci komputerowej MSK (w tym samym zeszycie).
- [4] Praca zbiorowa, Wprowadzenie do teleinformatyki, WKiŁ, Warszawa 1979.
- [5] D a v i e s D.W., B a r b e r D.L.A., Sieci teleinformatyczne, WNT, Warszawa 1979.
- [6] S e i d l e r J., Analiza i optymalizacja sieci łączności dla systemów teleinformatycznych, WNT, Warszawa 1978.
- [7] K l e i n r o c k L., Queueing Systems, A Wiley - Interscience Publication, Wiley and Sons, New York-London-Sydney-Toronto 1975.
- [8] F e l l e r W., Wstęp do rachunku prawdopodobieństwa, PWN, Warszawa 1966.

- [9] High Level Scheduler, University College of Swansea, PRB. pbm., Swansea 1979.
- [10] Lewoc J., Wołska J., Struktura członu 1 Systemu operacyjnego SOSAPI. Prace IASE, nr A 31, str. 42-55, Wrocław 1977.
- [11] CCITT, Revised Recommendation X.25, Preface and Level 3, COM VII No 384-E, sierpień 1979.
- [12] CCITT, Proposed Revision to Level 1 and 2 of Recommendation X.25, COM VII No 374-E, październik 1979.
- [13] PDP1104/34/45/55 Processor Handbook, Digital Equipment Corporation, Maynard, Massachusetts 1976-1977.
- [14] Muszyński J., Stanisław A., Gospodarka buforami w węźle sieci komputerowej MSK (w tym samym zeszycie).
- [15] Lewoc J., Koordynator węzła podsieci komunikacyjnej (w tym samym zeszycie).
- [16] Bieleninik E., Stanisław A., Szwarz M., Obsługa adapterów liniowych w węźle podsieci komunikacyjnej (w tym samym zeszycie).
- [17] Coffman E.G. i in., Teoria szeregowania zadań, WNT, Warszawa 1980.
- [18] Szwarz M., Implementacja protokołu liniowego w sieci komputerowej MSK (w tym samym zeszycie).
- [19] Łanowska B., Inicjacja i restart węzła podsieci komunikacyjnej (w tym samym zeszycie).
- [20] Zafriropulo P. i in., Towards Analyzing and Synthesizing Protocols, "IEEE Trans on Communications", Apr. 1980.
- [21] Sunshine C., Formal Techniques for Protocol Specification and Verification "Computer", Sept. 1979.
- [22] Hansen P.B., The Architecture of Concurrent Programs Prentice-Hall, Inc., Englewood Cliffs, 1977.
- [23] Huzar Z., Projekt makrojęzyka czasu rzeczywistego dla mini-komputera Mera 300, Inst. Cybern. Techn. PWr., Komunikat 564, Wrocław 1977.
- [24] Huzar Z., Nowe podejście do opisu semantyki komunikujących się procesów (w tym samym zeszycie).

Praca wpłynęła do Redakcji 20.12.1980 r.

Po poprawieniu 3.03.1982 r.

SOME RESEARCH PROBLEMS RELATED TO COMMUNICATION SUBNETWORK NODE REALIZATION

The paper presents some research problems that appeared during realization of operating system for the MSK communication subnetwork node. The significance of the maximum transition delay in the node is shown from the user's point of view. The optimization of the operating system and implemented protocols (X.25 level 2 and level 3) is proposed.

Verified by Ruta Czaplińska

ПРОБЛЕМЫ ИССЛЕДОВАНИЙ, СВЯЗАННЫЕ С РАЗРАБОТКОЙ УЗЛА ПОДСЕТИ СВЯЗИ

Описываются некоторые темы исследований, выступившие в проектировании и реализации узла подсети связи сети МСК. Целью большинства предложенных задач является повышение качества работы подсети с помощью оптимизации принятых программных решений и параметров протоколов. Предлагаются темы исследований, связанные с реализацией операционных систем реального времени.

Проверила Малгожата Хейдрих

Zbigniew HUZAR^{*)}

Nowe podejście do opisu semantyki komunikujących się procesów

Omówiono zasady komunikacji (wymiany informacji) pomiędzy procesami równoległymi, bazujące na uogólnionym protokole Hoare'a [7]. Uogólnienie protokołu polega na wprowadzeniu restrykcji czasowych na transfery informacji realizowane przez procesy. Podano schemat opisu semantyki komunikujących się procesów, wykorzystujący zmodyfikowane podejście Dajkesa [3]. Modyfikacja polega na związaniu z transformacjami językowymi ciągów oddziaływań zachodzących pomiędzy procesami. Rozważania zilustrowano na prostym przykładzie pary komunikujących się procesów.

1. WPROWADZENIE

Rozwój systemów rozproszonego przetwarzania informacji uzasadnia potrzebę prowadzenia badań równoległych procesów, które współpracują ze sobą dzięki wzajemnej wymianie informacji. Ostatni okres dostarcza przykładów wielu prac, które stanowią propozycję nowych konstrukcji językowych [7,26] lub języków programowania [4,11], przeznaczonych do programowania rozproszonych procesów. Propozycjom tym towarzyszą na ogół dodatkowe rozważania związane z opisem semantyki omawianych konstrukcji. Możliwość efektywnej semantycznej charakterystyki konstrukcji językowych jest jednym z ważniejszych elementów oceny ich przydatności.

Jedną z najbardziej interesujących dróg opisu semantyki komunikujących się procesów wyznaczają prace zainicjowane przez Hoare'a [5,7,8].

^{*)} Centrum Obliczeniowe Politechniki Wrocławskiej, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

Ogólna idea opisu semantyki polega na wyznaczeniu - w pierwszym etapie - semantyki pojedynczych procesów z założeniem ich izolacji od pozostałych procesów, a następnie - w drugim etapie - na "sparametryzowaniu" wyznaczonych niezależnych semantyk przez ciągi komunikatów (informacji), jakie mogą wymieniać między sobą procesy. Zasadniczym elementem, na którym opierają się rozważania, jest przyjęty protokół komunikacji pomiędzy procesami ("rendez-vous" protokół). Protokół polega na tym, że proces nadający (odbierający) informacje do (od) innego wskazanego procesu oczekuje tak długo, aż wskazany proces będzie gotowy do odebrania (nadania) komunikatu. Założenie takiego protokołu jest dużym ograniczeniem, gdyż pojedyncze procesy mogą pozostawać nieaktywne w oczekiwaniu na inne procesy przez niekontrolowane odcinki czasu, a nawet, w przypadku szczególnym, może dojść do trwałego zastoju procesów (np. gdy dwa procesy jednocześnie oczekują na odbiór informacji od partnera). Osłabienie tego założenia i przyjęcie zasady oczekiwania jednego procesu na drugi proces przez określony odcinek czasu (wprowadzanie czasów przeterminowania - limitów czasowych) powoduje komplikacje w opisie semantyki.

Przedstawione w artykule rozważania dotyczą opisu semantyki procesów komunikujących się z założeniem protokołu stanowiącego uogólnienie protokołu "rendez-vous". Uogólnienie polega właśnie na wprowadzaniu czasów przeterminowania. Do zapisu procesów używa się języka zdefiniowanego w pracy [9]. Opis semantyki pojedynczych procesów opiera się na modyfikacji podejścia Dijkstry [3], przedstawionej w pracy [9]. W artykule ograniczono się do przedstawienia nowego podejścia do opisu semantyki procesów komunikujących się na podstawie analizy prostego przykładu pary tych procesów.

2. JĘZYK PROGRAMOWANIA I ZASADY KOMUNIKACJI PROCESÓW

Język programowania procesów zawiera instrukcje [3]: zostaw, zerrwij, podstawienie, alternatywę, iterację oraz złożenie sekwencyjne instrukcji. Instrukcje alternatywy i iteracji są zmodyfikowane następująco [9]:

$$\underline{\text{if}} \ B_1; C_1 \longrightarrow SL_1 \square \dots \square B_n; C_n \longrightarrow SL_n \ \underline{\text{fi}} \quad (2.1)$$

$$\underline{\text{do}} \ B_1; C_2 \longrightarrow SL_1 \square \dots \square B_n; C_n \longrightarrow SL_n \ \underline{\text{od}} \quad (2.2)$$

gdzie $B_1, \dots, B_n$ są warunkami (dozorami) logicznymi, $SL_1, \dots, SL_n$ - dowolnymi ciągami instrukcji, natomiast $C_1, \dots, C_n$ są nowymi konstrukcjami, tzw. warunkami (dozorami) transferowymi. Warunki transferowe stanowią czasowo uwarunkowane instrukcje wejścia/wyjścia i mogą mieć jedną z postaci:

cond $d \ ? \ v \ \underline{\text{while}} \ T$ (2.3)

cond $d \ ! \ E \ \underline{\text{while}} \ T$ (2.4)

cond $d \ ? \ \underline{\text{nonwhile}} \ T$ (2.5)

cond $d \ ! \ \underline{\text{nonwhile}} \ T$ (2.6)

Warunek (2.3) jest czasowo uwarunkowaną operacją wejścia. Realizacja warunku polega na próbie odebrania z urządzenia d pewnej informacji (komunikatu) z otoczenia procesu (od innego procesu) i podstawienie tej informacji pod zmienną wewnętrzną procesu v . Próba takiego transferu jest wykorzystywana w odcinku czasu $[t, t+T]$, gdzie t jest momentem rozpoczęcia realizacji warunku, a T jest wyrażeniem o wartościach nieujemnych. Jeżeli w czasie trwania tego odcinka otoczenie procesu będzie gotowe do przekazywania informacji, to nastąpi podstawienie odpowiedniej wartości pod zmienną v i realizacja warunku (2.3) zakończy się pozytywnie. Jeżeli w czasie trwania tego odcinka otoczenie jest niegotowe, to zmienna v nie zmieni swej wartości i realizacja (2.3) zakończy się negatywnie.

Warunek (2.4) jest czasowo uwarunkowaną operacją wyjścia. Jego realizacja przebiega analogicznie do realizacji warunku (2.3) z tym, że transfer informacji polega na przekazaniu do otoczenia procesu wartości wyrażenia E przez urządzenie d .

Warunki (2.5), (2.6) są warunkami dopełniającymi w stosunku do warunków (2.3)-(2.4) i spełniają funkcję testowania gotowości otoczenia procesu do transferu informacji w odcinku czasu $[t, t+T]$. Warunek (2.5) kończy się pozytywnie, jeżeli w tym odcinku czasu urządzenie d jest niegotowe do odbioru informacji z otoczenia oraz kończy się negatywnie w przypadku przeciwnym. Analogicznie realizuje się warunek (2.6) z tym, że testuje on gotowość urządzenia do nadawania.

Intuicyjny sens instrukcji alternatywy jest następujący:

Realizacja instrukcji rozpoczyna się od obliczania wartości warunków logicznych $B_1, \dots, B_n$. Jeżeli żaden z nich nie jest prawdziwy, to następuje zerwanie obliczeń. Jeżeli pewne warunki są prawdziwe, to rozpoczyna się jednoczesna realizacja tych warunków transferowych C_i , dla których odpowiadające warunki B_i są prawdziwe. Realizacja warunków transferowych trwa do momentu, w którym zakończy się pozytywnie jeden z warunków. W tym momencie zostaje przerwana realizacja pozostałych warunków (oznacza to, że zostaje wykonana co najwyżej jedna operacja wejścia/wyjścia). Jeżeli jako pierwszy pozytywnie zakończył swe działanie warunek C_j , to jako następną wykonuje się instrukcję SL_j , po czym wykonanie instrukcji alternatywy kończy się. Gdyby wszystkie warunki transferowe zakończyły się negatywnie, wówczas nastąpiłoby zerwanie obliczeń.

Wykonanie instrukcji iteracji przebiega podobnie. A więc rozpoczyna się od obliczenia wartości warunków logicznych. Gdy żaden z tych warunków nie jest prawdziwy, następuje zakończenie instrukcji (a nie zerwanie jak w poprzedniej instrukcji). Gdy pewne warunki logiczne są prawdziwe, rozpoczyna się jednoczesna realizacja odpowiednich warunków transferowych, a po pozytywnym zakończeniu jednego z nich - wybór odpowiedniej instrukcji SL_j (tak jak w instrukcji alternatywy), po czym przedstawiony cykl obliczeń powtarza się i następuje ponowne obliczenie nowych wartości warunków logicznych.

W razie pomyślnego wykonania instrukcji alternatywy bądź pojedynczego cyklu instrukcji iteracji zachodzi dokładnie jedno oddziaływanie (pkt 3 - co najwyżej jedna operacja wejścia/wyjścia) generowane przez warunki transferowe $C_1, \dots, C_n$. Aby zachodzące oddziaływanie jednoznacznie powiązać ze zrealizowanym pomyślnie warunkiem transferowym przyjęto założenie, że w zbiorze warunków $C_1, \dots, C_n$ nie mogą wystąpić dwa warunki typu cond - while albo dwa warunki typu cond - nonwhile dotyczące tego samego urządzenia wejścia/wyjścia. Dopuszcza się także, że w instrukcjach (2.1), (2.2) może nie wystąpić pewien warunek B_i lub C_i . Brak B_i będzie potraktowany tak jakby $B_i \iff T$ (T - predykat zawsze prawdziwy), a brak C_i będzie potraktowany jako warunek transferowy, który do pomyślnej realizacji wymaga pustego oddziaływania. Wynika stąd w szczególności, że jeżeli wszystkie warunki $C_1, \dots, C_n$ są puste, to wybór odpowiedniej instrukcji SL_i dla jednego spośród warunków B_i jest dokonywany w sposób niedeterministyczny.

Formalną semantykę zmodyfikowanych instrukcji alternatywy i iteracji przedstawiono w pracy [9].

Różne procesy mogą komunikować się ze sobą przez wspólne urządzenia wejścia/wyjścia. Przekazywanie wartości wyrażenia E od procesu P_1 do procesu P_2 przez urządzenie d podstawienia tej wartości pod zmienną v wymaga takiej realizacji warunków:

cond $d ! E$ while T_1 w procesie P_1

oraz

cond $d ! v$ while T_2 w procesie P_2 ,

aby niepusty był przekrój odcinków czasu $[t_1, t_1+T_1]$ oraz $[t_2, t_2+T_2]$, gdzie t_1, t_2 oznaczają momenty rozpoczęcia realizacji odpowiednich warunków transferu. Jest to tzw. warunek asocjacji dozorów transferowych.

3. ZASADY OPISU SEMANTYKI

Opis semantyki procesów komunikujących się wymaga wprowadzenia dodatkowego formalizmu opisującego transfery informacji między procesami.

Obecnie będzie rozpatrywana najprostsza sytuacja komunikacji między dwoma procesami. Niech D oznacza zbiór wspólnych urządzeń wejścia/wyjścia, przez które procesy komunikują się ze sobą. Z pojedynczym urządzeniem $d \in D$ będzie związany zbiór oddziaływań

$$\left. \begin{aligned} \Gamma_d &= \Gamma_d^- \cup \Gamma_d^+ , \\ \Gamma_d^- &= \{d^-\} , \\ \Gamma_d^+ &= \{v_d^1, v_d^2, \dots, v_d^{n(d)}\} . \end{aligned} \right\} \quad (3.1)$$

gdzie

Elementy zbioru Γ_d nazywane oddziaływaniami służą do oznaczania różnych sytuacji jakie mogą zajść podczas wykonywania warunków transferowych (2.3)-(2.6). Symbol d^- oznacza, że urządzenie d pozostawało niegotowe do transferu przez odcinek czasu określony w odpowiednim warunku. Symbole zbioru Γ_d^+ oznaczają, że urządzenie d było gotowe do transferu we wskazanym przez warunek transferowy odcinku czasu i wskazuje wartość, która została przesłana.

Zbiór $\Gamma = \bigcup_{d \in D} \Gamma_d$ nazywa się zbiorem oddziaływań zewnętrznych, a ciągi postaci

$$\gamma = \gamma_1 \gamma_2 \dots \gamma_n \quad (3.2)$$

gdzie $\gamma_i \in \Gamma$ nazywa się ciągami oddziaływań. Ciąg γ dzieli się na dwa rozłączne podciągi γ^- oraz γ^+ . Podciąg γ^- , zwany ciągami transferów negatywnych, zawiera wszystkie te wyrazy γ_i ciągu γ , które należą do zbioru $\Gamma^- = \bigcup_{d \in D} \Gamma_d^-$. Podciąg γ^+ , zwany ciągami transferów pozytywnych, zawiera wszystkie pozostałe wyrazy γ_j ciągu γ , tzn. te, które należą do zbioru $\Gamma^+ = \bigcup_{d \in D} \Gamma_d^+$.

Ciągi oddziaływań γ oraz δ obserwowane przez dwa współpracujące procesy P1 oraz P2 mogą być oczywiście różne, ale muszą być zgodne, tzn. muszą zawierać identyczne podciągi transferów pozytywnych, czyli $\gamma^+ = \delta^+$.

Ciąg oddziaływań γ jest klasy SYN(k), jeżeli dwa kolejne elementy podciągu γ^+ są w ciągu γ przedzielone najwyżej k elementami podciągu γ^- . Para ciągów γ, δ jest klasy SYN($k, 1$), jeżeli są one zgodne oraz γ jest klasy SYN(k), zaś δ - klasy SYN(1). Łatwo zauważyć, że

$$\text{SYN}(k, 1) \subset \text{SYN}(m, n) \quad \text{dla} \quad k \leq m, \quad 1 \leq n \quad (3.3)$$

$$\text{SYN}(k, 1) \cup \text{SYN}(m, n) = \text{SYN}(\max(k, m), \max(1, n)) \quad (3.4)$$

$$\text{SYN}(k, 1) \cap \text{SYN}(m, n) = \text{SYN}(\min(k, m), \min(1, n)) \quad (3.5)$$

W dalszym ciągu będzie rozpatrywana współpraca dwóch procesów równoległych (oznaczenie $P1 \parallel P2$) z założeniem, że są one synchronizowane w taki sposób, iż obserwowane przez nie ciągi oddziaływań należą do klasy $SYN(k,1)$. Problem w jaki sposób należy te procesy synchronizować, aby zapewnić ciągom oddziaływań pozostawanie w tej klasie, stanowi oddzielną, nie poruszaną tutaj kwestię.

Niech $\Gamma_{(k,1)}(P1, x \parallel P2, y)$ oznacza zbiór tych par ciągów oddziaływań, które należą do klasy $SYN(k,1)$, i które mogą wystąpić podczas współpracy procesów $P1, P2$ z założeniem, że rozpoczynają one swe obliczenia od stanów początkowych x oraz y . Definicja semantyki procesów komunikujących się opiera się na następującym schemacie. Niech R oznacza predykat wyznaczający pewien zbiór par stanów końcowych (x, y) procesu $P1$ oraz procesu $P2$. Należy wyznaczyć taki zbiór par stanów początkowych (x, y) procesów $P1$ i $P2$, które mają tę właściwość, że obliczenia rozpoczynające się w tych stanach początkowych pod wpływem dowolnych par ciągów oddziaływań należących do zbioru $\Gamma_{(k,1)}(P1, x \parallel P2, y)$ kończą się w podzbiorze par stanów końcowych wyznaczonych przez predykat końcowy R . Zbiór, który należy wyznaczyć przedstawia się następująco:

$$\left\{ (x, y) \mid \left(\bigvee_{(y, \delta) \in \Gamma_{(k,1)}(P1, x \parallel P2, y)} (P1(x, \gamma), P2(y, \delta)) \in R \right) \right\}. \quad (3.6)$$

Występujące oznaczenia mają sens następujący: $P1(x, \gamma)$ - wyznacza stan końcowy jaki osiągnie proces $P1$ rozpoczynając swe obliczenia ze stanu początkowego x przy założeniu, że w trakcie jego realizacji znajdzie ciąg oddziaływań γ ; analogicznie $P2(y, \delta)$ wyznacza odpowiedni końcowy stan procesu $P2$. Oznaczenie $(a, b) \in R$ należy czytać: para stanów (a, b) , należąca do zbioru par stanów wyznaczonych przez predykat R .

Predykat, który wyznacza zbiór (3.6) będzie nazywany najsłabszym warunkiem wstępnym dla klasy oddziaływań $SYN(k,1)$ warunki końcowe R i będzie oznaczany przez $sww_{(k,1)}(P1 \parallel P2, R)$. Zdefiniowany słabszy warunek wstępny spełnia znane właściwości (3):

$$sww_{(k,1)}(P1 \parallel P2, F) \iff F, \quad (3.7)$$

$$(Q \implies R) \implies (sww_{(k,1)}(P1 \parallel P2, Q) \implies sww_{(k,1)}(P1 \parallel P2, R)), \quad (3.8)$$

$$sww_{(k,1)}(P1 \parallel P2, R) \wedge sww_{(k,1)}(P1 \parallel P2, Q) \iff sww_{(k,1)}(P1 \parallel P2, Q \wedge R) \quad (3.9)$$

$$sww_{(k,1)}(P1 \parallel P2, R) \vee sww_{(k,1)}(P1 \parallel P2, Q) \implies sww_{(k,1)}(P1 \parallel P2, R \vee Q) \quad (3.10)$$

gdzie F oznacza predykat zawsze fałszywy, a R, Q są dowolnymi predykatami.

Dodatkowe własności najsłabszego warunku wstępnego wynikają z własności klas ciągów oddziaływań:

$$sww_{(k,l)}(P1 || P2, R) \Rightarrow sww_{(m,n)}(P1 || P2, R) \text{ dla } k \geq m, l \geq n, \quad (3.11)$$

$$sww_{(k,l)}(P1 || P2, R) \wedge sww_{(m,n)}(P1 || P2, R) \Leftrightarrow sww_{(\max(k,m), \max(l,n))}(P1 || P2, R), \quad (3.12)$$

$$sww_{(k,l)}(P1 || P2, R) \vee sww_{(m,n)}(P1 || P2, R) \Leftrightarrow sww_{(\min(k,m), \min(l,n))}(P1 || P2, R). \quad (3.13)$$

Często zachodzi potrzeba wyznaczenia semantyki jednego z komunikujących się procesów w sposób możliwie uzależniony od własności innych procesów. W tym celu wprowadza się pojęcie niedeterministycznego procesu zastępczego (NPZ).

Niech $P1, P2$ będą procesami komunikującymi się w klasie oddziaływań $SYN(k,l)$. Procesem zastępczym procesu $P2$ dla procesu $P1$ jest proces NPZ , który:

- (i) zapewnia procesowi $P1$ pozostanie w klasie synchronizacji $SYN(k)$,
- (ii) w każdym momencie pozytywnego transferu informacji kierowanej do procesu $P1$ przez urządzenie d może wygenerować dowolną z wartości zbioru Γ_d^+ .

Z definicji wynika, że proces zastępczy NPZ procesu $P2$ dla procesu $P1$ jest niezależny od procesu $P2$ oraz od jego klasy synchronizacji. Stąd też zamiast pisać

$$sww_{(k,l)}(P1 || NPZ, R)$$

będzie się pisać

$$sww_{(k)}(P1, R).$$

Niech predykat R będzie określony tylko na zbiorze stanów procesu $P1$, co zapiszemy $R(x)$. Wtedy dla dowolnego procesu $P2$ i dowolnego l :

$$sww_{(k)}(P1, R(x)) \Rightarrow sww_{(k,l)}(P1 || P2, R(x)). \quad (3.14)$$

Dalej, niech $Q(x,y) \Leftrightarrow R(x) \wedge S(y)$, wtedy z (3.9) oraz (3.14) wynika, że

$$sww_{(k)}(P1, R(x)) \wedge sww_{(l)}(P2, S(y)) \Rightarrow sww_{(k,l)}(P1 || P2, Q(x,y)). \quad (3.15)$$

Wzór (3.15) pokazuje związek między "niezależnymi" semantykami procesów $P1, P2$ a semantyką "zależną" procesów $P1 || P2$.

4. PRZYKŁAD

Dalsze rozważania zostaną oparte na przykładzie dwóch współpracujących procesów $P1 \parallel P2$. Procesy te są przedstawione w następującej szkieletowej postaci:

P1: if $B_{11}^1; C_{11}^1 \longrightarrow x := E_{11}^1 \square$
 $B_{12}^1; C_{12}^1 \longrightarrow x := E_{12}^1 \square$
 $B_{13}^1; C_{13}^1 \longrightarrow x := E_{13}^1$
 fi;
 if $B_{21}^1; C_{21}^1 \longrightarrow x := E_{21}^1 \square$
 $B_{22}^1; C_{22}^1 \longrightarrow x := E_{22}^1 \square$
 $B_{23}^1; C_{23}^1 \longrightarrow x := E_{23}^1$
 fi.

 P2: if $B_{11}^2; C_{11}^2 \longrightarrow y := E_{11}^2 \square$
 $B_{12}^2; C_{12}^2 \longrightarrow y := E_{12}^2 \square$
 $B_{13}^2; C_{13}^2 \longrightarrow y := E_{13}^2$
 fi;
 if $B_{21}^2; C_{21}^2 \longrightarrow y := E_{21}^2 \square$
 $B_{22}^2; C_{22}^2 \longrightarrow y := E_{22}^2 \square$
 $B_{23}^2; C_{23}^2 \longrightarrow y := E_{23}^2$
 fi.

Procesy komunikują się przez jedno wspólne urządzenie wejścia/wyjścia d. Zakłada się, że warunki transferowe C_{j1}^i ($i, j = 1, 2$) mają postać (2.3), warunki C_{j2}^i - postać (2.4), a warunki C_{j3}^i - postać (2.5) lub (2.6). Na inne potrzeby wprowadza się dodatkowe oznaczenie: $x(C)$ symbolizuje zmienną procesu, która jest elementem warunku transferowego C ; podobnie $E(C)$ oznacza wyrażenie, które jest elementem warunku C .

Transfery informacji między $P1$ a $P2$ wymagają odpowiedniej asocjacji warunków transferu. W klasie oddziaływań $SYN(0,0)$ zbiór możliwych transferów jest wyznaczony przez następujące asocjacje:

$$(c_{11}^1, c_{12}^2)(c_{21}^1, c_{22}^2), \quad (4.1)$$

$$(c_{11}^1, c_{12}^2)(c_{22}^1, c_{21}^2), \quad (4.2)$$

$$(c_{12}^1, c_{11}^2)(c_{21}^1, c_{22}^2), \quad (4.3)$$

$$(c_{12}^1, c_{11}^2)(c_{22}^1, c_{21}^2). \quad (4.4)$$

Klasy SYN(0,1) oraz SYN(1,0) nie wyznaczają dodatkowych asocjacji, gdyż wymaganie braku nieudanych transferów w jednym z procesów narzuca brak nieudanych transferów w drugim z procesów. W klasie SYN(1,1) pojawiają się dodatkowe możliwości asocjacji:

$$(c_{11}^1, c_{12}^2), \quad (4.5)$$

$$(c_{11}^1, c_{22}^2), \quad (4.6)$$

$$(c_{12}^1, c_{11}^2), \quad (4.7)$$

$$(c_{12}^1, c_{21}^2), \quad (4.8)$$

$$(c_{21}^1, c_{12}^2), \quad (4.9)$$

$$(c_{21}^1, c_{22}^2), \quad (4.10)$$

$$(c_{22}^1, c_{11}^2), \quad (4.11)$$

$$(c_{22}^1, c_{21}^2), \quad (4.12)$$

Klasy SYN(1,2) oraz SYN(2,1) nie wyznaczają dodatkowych asocjacji.

Dalsze klasy synchronizacji SYN(k,l) dla $k, l \geq 2$ wyznaczają dodatkowo jedną asocjację pustą (wszystkie warunki typu cond-while kończą się negatywnie).

Z każdą asocjacją jest związany zbiór ciągów oddziaływań, stanowiący pewien podzbiór ciągów oddziaływań należących do danej klasy synchronizacji. Niech

$$sww_{(k,l)}^i(P1 || P2, R) \quad (4.13)$$

oznacza najśłabszy warunek wstępny dla podklasy oddziaływań generowanej przez i-tą asocjację w klasie SYN(k,l). Zatem zachodzi wzór:

$$sww_{(k,l)}(P1 || P2, R) \iff \bigvee_{i \in I} sww_{(k,l)}^i(P1 || P2, R), \quad (4.14)$$

gdzie I oznacza zbiór wszystkich asocjacji w klasie SYN(k,l).

Wzór (4.14) pozwala na etapowe wyznaczenie warunku $sww_{(k,l)}$ przez wyznaczenie wszystkich warunków $sww_{(k,l)}^i$ $i \in I$, natomiast wyznaczenie tych ostatnich warunków stanowi zagadnienie klasyczne [3].

Przykładowo rozpatrzmy warunki $\text{sww}_{(0,0)}^1(P1 || P2, R)$ dla $i=1,2,3,4$, odpowiadających odpowiednio asocjacjom (4.1)-(4.4). Dla zwięzłości dalszego zapisu wprowadza się następujące oznaczenia: jeżeli $R(x,y)$ jest predykatem, $E_i(x,y)$ pewnymi wyrażeniami, to

$$R(x,y) \Big|_{E_1 \rightarrow x, E_2 \rightarrow y}$$

jest predykatem, który powstaje z R przez zastąpienie wszystkich wystąpień zmiennych x oraz y w R odpowiednio przez E_1 oraz E_2 .

Dodatkowo

$$R(x,y) \Big|_{\begin{array}{l} E_1 \rightarrow x, E_2 \rightarrow y \\ E_3 \rightarrow x, E_4 \rightarrow y \end{array}}$$

będzie skrótem zapisu

$$(R(x,y) \Big|_{E_1 \rightarrow x, E_2 \rightarrow y}) \Big|_{E_3 \rightarrow x, E_4 \rightarrow y}$$

W przyjętej konwencji oznaczeń:

$$\text{sww}_{(0,0)}^1(P1 || P2, R) \iff$$

$$\iff B_{12}^2 \wedge B_{11}^1 \wedge (B_{22}^2 \wedge B_{21}^1 \wedge R \Big|_{\begin{array}{l} E_{21}^1 \rightarrow x, E_{22}^2 \rightarrow y \\ E(C_{22}^2) \rightarrow x(C_{21}^1) \end{array}} \Big|_{\begin{array}{l} E_{11}^1 \rightarrow x, E_{12}^2 \rightarrow y \\ E(C_{12}^2) \rightarrow x(C_{21}^1) \end{array}}) \quad (4.15)$$

$$\text{sww}_{(0,0)}^2(P1 || P2, R) \iff$$

$$\iff B_{12}^2 \wedge B_{11}^1 \wedge (B_{21}^2 \wedge B_{22}^1 \wedge R \Big|_{\begin{array}{l} E_{22}^1 \rightarrow x, E_{21}^2 \rightarrow y \\ E(C_{22}^1) \rightarrow y(C_{21}^2) \end{array}} \Big|_{\begin{array}{l} E_{11}^1 \rightarrow x, E_{12}^2 \rightarrow y \\ E(C_{12}^2) \rightarrow x(C_{21}^1) \end{array}}) \quad (4.16)$$

$$\text{sww}_{(0,0)}^3(P1 || P2, R) \iff$$

$$\iff B_{11}^2 \wedge B_{12}^1 \wedge (B_{22}^2 \wedge B_{21}^1 \wedge R \Big|_{\begin{array}{l} E_{21}^1 \rightarrow x, E_{22}^2 \rightarrow y \\ E(C_{22}^2) \rightarrow x(C_{21}^1) \end{array}} \Big|_{\begin{array}{l} E_{12}^1 \rightarrow x, E_{11}^2 \rightarrow y \\ E(C_{12}^1) \rightarrow y(C_{21}^2) \end{array}}) \quad (4.17)$$

$$swv^4_{(0,0)}(P1 || P2, R) \iff$$

$$\iff B_{11}^2 \wedge B_{12}^1 \wedge (B_{21}^2 \wedge B_{22}^1 \wedge R \mid \begin{array}{l} E_{22}^1 \rightarrow x, E_{21}^2 \rightarrow y \\ E(C_{22}^1) \rightarrow y(C_{21}^2) \end{array} \mid \begin{array}{l} E_{12}^1 \rightarrow x, E_{11}^2 \rightarrow y \\ E(C_{12}^1) \rightarrow y(C_{11}^2) \end{array}) \quad (4.18)$$

Konjunkcja prawych stron wzorów (4.15)-(4.18) wyznacza $swv_{(0,0)}(P1 || P2, R)$. W celu wyznaczenia najsłabszych warunków wstępnych dla innych klas synchronizacji należałoby wyznaczyć najsłabsze warunki wstępne dla dalszych asocjacji. Ostatnim przykładem będzie najsłabszy warunek wstępny dla pustej asocjacji, który przedstawia się wzorem.

$$B_{13}^1 \wedge B_{13}^2 \wedge (B_{23}^1 \wedge B_{23}^2 \wedge R \mid \begin{array}{l} E_{23}^1 \rightarrow x, E_{23}^2 \rightarrow y \end{array} \mid \begin{array}{l} E_{13}^1 \rightarrow x, E_{13}^2 \rightarrow y \end{array}) \quad (4.19)$$

5. UWAGI KOŃCOWE

Zakres przedstawionych rozważań ogranicza się do podania ogólnych zasad opisu semantyki procesów komunikujących się i wyjaśnienia wprowadzonych koncepcji na prostym przykładzie. Pominęto wiele ważnych kwestii wymagających dalszego rozpatrzenia. Należą do nich między innymi:

- (i) zasady generacji $swv_{(k,l)}(P1 || P2, R)$ na podstawie opisu syntaktycznego procesów $P1$ i $P2$, zwłaszcza dla procesów zawierających pętle;
- (ii) zasady synchronizacji procesów gwarantujące przebieg współdziałania procesów w określonej klasie oddziaływań.

Przedstawioną propozycję można uważać za jedną z prób semantycznej charakteryzacji procesów komunikujących się w czasie rzeczywistym. Wprowadzona postać warunków transferu oraz pojęcie klasy oddziaływań pozwala abstrahować od bezwzględnej wartości czasu rzeczywistego. Tym samym została umożliwiona adaptacja metod opisu semantyki programów sekwencyjnych do opisu semantyki procesów komunikujących się.

LITERATURA

- [1] B i l l e A., A survey of input-output semantics and program verification. Inst. Podst. Inf. PAN, Rap. 344, 1979.
- [2] B r i n c h H a n s e n P., Distributed processes: a concurrent programming concept, Comm. ACM, vol. 21, 11, 1978 (934-941).

- [3] D i j k s t r a E.W., Guarded commands, nondetermincy and formal derivation of programs, Comm. ACM, vol. 18, 8, 1975 (453-457).
- [4] F e l d m a n J.A., High level programming for distributed computing, Comm. ACM, vol. 22, 6, 1979 (353-368).
- [5] F r a n c e z N., H o a r e C.A.R., R o e v e r W.P., Semantics of nondeterminism, concurrency and communication. Proceedings of Symposium "Mathematical Foundations of Computer Science 1978", ed. by J. Winkowski, Springer Verlag, 1978.
- [6] H e w i t t C.E., A t k i n s o n R.R., Specification and proof techniques for serializers, IEEE Trans. Softw. Ing. vol. SE-5, 1, 1979 (10-23).
- [7] H o a r e C.A.R., Communicating sequential processes, Comm. ACM, vol. 21, 8, 1978 (666-667).
- [8] H o a r e C.A.R., A relational trace-oriented semantics for communicating sequential processes. Workshop on Verification of Parallel processes. Aarhus, Denmark 1977.
- [9] H u z a r Z., Semantyka programów z operacjami wejścia/wyjścia, Podstawy Sterowania, t. 11, z. 1, s. 33-50, 1981.
- [10] M i l n e r R., Synthesis of communicating behaviour, Proceedings of Symposium "Mathematical Foundations of Computer Science", ed. by J. Winkowski, Springer Verlag, 1978.
- [11] S c h u t z H.A., On the design of a language for programming real-time concurrent processes, IEEE Trans. Softw. Ing., vol. SE-5, 3, 1979 (248-255).

Praca wpłynęła do Redakcji 25.06.1980 r.

Po poprawieniu 3.03.1982 r.

A NEW APPROACH TO THE DESCRIPTION OF SEMANTICS OF COMMUNICATING PROCESSES

Principles of communication (information exchange) between parallel processes, based on generalized Hoare's protocol [7] are discussed. The generalization of protocol consistain introduction of time-outs for information exchange made by processes. A description semantics of using modified Dijkstra's approach [3] is given in schematic form. The modification consists in matching predicates transformers with the interaction sequences occurring while the inter processes are communicating. The considerations are illustrated by a simple example of a pair of communicating processes.

Verified by Ruta Czaplińska

НОВЫЙ ПОДХОД К ОПИСАНИЮ СЕМАНТИКИ КОММУНИКАЦИОННЫХ ПРОЦЕССОВ

Представлены принципы коммуникации (обмена информацией) между параллельными процессами, базирующие на обобщенном протоколе Хоара [7]. Это обобщение заключено во введении временных ограничений обмена информацией между процессами. Представлена схема описания семантики коммуникационных процессов, использующая модифицированный подход Дийкотри [3]. Эта модификация заключена в связи трансформаторов предикатов с секвенцией трансверсов информации между процессами. Рассуждения иллюстрированы простым примером пары коммуникационных процессов.

Проверила Малгожата Хейдрих

SPIS RZECZY

Wprowadzenie	3
1. Józef LEWOC, Struktura oprogramowania węzła podsieci komunikacyjnej	7
2. Józef LEWOC, Koordynator węzła podsieci komunikacyjnej	23
3. Barbara ŁANOWSKA, Inicjacja pracy i restart węzła podsieci komunikacyjnej	35
4. Andrzej STANISZ, Współpraca operator-węzeł w sieci komputerowej MSK	43
5. Zbigniew FRYŻLEWICZ, Protokół pakietowy w sieci komputerowej MSK	51
6. Zbigniew FRYŻLEWICZ, Elżbieta MIKULSKA, Zestawianie połączeń logicznych w sieci komputerowej MSK	63
7. Krystyna KOLEŚNIK, Transmisja danych w sieci komputerowej MSK	79
8. Iwona DUBIELEWICZ, Krystyna KOLEŚNIK, Reinicjacja transmisji danych w sieci komputerowej MSK	95
9. Iwona DUBIELEWICZ, Kasowanie połączeń logicznych w sieci komputerowej MSK	107
10. Józef GOETZ, Restart połączeń logicznych w sieci komputerowej MSK	125
11. Michał SZWARC, Implementacja protokołu liniowego w sieci komputerowej MSK	143
12. Elżbieta HUDYMA, Elżbieta KOSMULSKA-BOCHENEK, Rejestracja pomiarów w sieci komputerowej MSK	155
13. Elżbieta HUDYMA, Elżbieta KOSMULSKA-BOCHENEK, Edycja rejestrowanych pomiarów w węźle sieci komputerowej MSK	169
14. Elżbieta HUDYMA, Elżbieta KOSMULSKA-BOCHENEK, Pakiety miernictwa w sieci komputerowej MSK	175
15. Józef LEWOC, Metoda wyboru drogi w sieci komputerowej MSK	183
16. Janisław MUSZYŃSKI, Andrzej STANISZ, Gospodarka buforami w węźle sieci komputerowej MSK	195
17. Edward BIELENINIK, Andrzej STANISZ, Michał SZWARC, Obsługa adapterów liniowych w węźle podsieci komunikacyjnej	209
18. Barbara ŁANOWSKA, Moduł zegara w systemie operacyjnym węzła podsieci komunikacyjnej	225
19. Barbara ŁANOWSKA, Oprogramowanie podstawowych urządzeń wejścia-wyjścia w węźle podsieci komunikacyjnej	231
20. Adam BECHER, Zasady uruchamiania i testowania modułów programowych na przykładzie węzła podsieci komunikacyjnej	239

21. Marek KRÓLIK, Oprogramowanie diagnostyczne adaptera liniowego ALS11	251
22. Józef LEWOC, Witold KOMOROWSKI, Problematyka badawcza węzła podsieci komunikacyjnej	261
23. Zbigniew HUZAR, Nowe podejście do opisu semantyki komunikujących się procesów	273

CONTENTS

Preface	3
1. Józef LEWOC, Structure of communication subnetwork node software	7
2. Józef LEWOC, Communication subnetwork node coordinator . . .	23
3. Barbara ŁANOWSKA, Initiation of operation and restart for communication subnetwork node	35
4. Andrzej STANISZ, Operator-computer communication in node of the MSK computer network	43
5. Zbigniew FRYŻLEWICZ, Packet protocol in the MSK computer network	51
6. Zbigniew FRYŻLEWICZ, Elżbieta MIKULSKA, Call set-up procedure in the MSK computer network	63
7. Krystyna KOLEŚNIK, Data transmission in the MSK computer network	79
8. Iwona DUBIELEWICZ, Krystyna KOLEŚNIK, Reset of data transmission in the MSK computer network	95
9. Iwona DUBIELEWICZ, Disconnection phase in the MSK computer network	107
10. Józef GOETZ, Restart of virtual call in the MSK computer network	125
11. Michał SZWARC, Implementation of line level protocol in the MSK computer network	143
12. Elżbieta HUDYMA, Elżbieta KOSMULSKA-BOCHENEK, Registration for the measurement data in the MSK computer network	155
13. Elżbieta HUDYMA, Elżbieta KOSMULSKA-BOCHENEK, Edition of the measurement data in the MSK computer network	169
14. Elżbieta HUDYMA, Elżbieta KOSMULSKA-BOCHENEK, Measurement packets in the MSK computer network	175
15. Józef LEWOC, Routing method in the MSK computer network . . .	183
16. Janisław BUSZYŃSKI, Andrzej STANISZ, Buffer management in the node of MSK computer network	195

17. Edward BIELENINIK, Andrzej STANISZ, Michał SZWARC, Line adapter servicing in communications subnetwork node	209
18. Barbara ŁANOWSKA, Line clock module of communication subnetwork node operating system	225
19. Barbara ŁANOWSKA, Software for basic input/output peripherals in communication subnetwork node	231
20. Adam BECHER, Software module debugging and testing rules on example of communications subnetwork node	239
21. Marek KRÓLIK, Diagnostic software for synchronous line adapter ALS11	251
22. Józef LEWOC, Witold KOMOROWSKI, Some research problems related to communication subnetwork node realization	261
23. Zbigniew HUZAR, A new approach to semantics description of communicating processes	273

PRACE NAUKOWE CENTRUM OBLICZENIOWEGO

Nr 1, Studia i Materiały nr 1, *Zagadnienia oprogramowania systemów
komputerowego wspomagania dydaktyki*, Wrocław 1980

20,—

Cena zł 180.—

BIBLIOTEKA
Politechniki Wrocławskiej

A 4947 I/1
1983 nr 2/2

Subscription rates and orders should be addressed to:

OR PAN, PKiN, 00-901 Warszawa, POLAND

Bank account number: NBP, VIII OM nr 1550-6-81574 Warszawa

Orders from abroad can be placed with

"ARS POLONA"

Krakowskie Przedmieście 7, 00-068 Warszawa

Wydawnictwa Politechniki Wrocławskiej
ma stałe na składzie Księgarnia Wr 49

Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

oraz Wojewódzka Księgarnia Techniczna

ul. Świdnicka 8, 50-067 Wrocław