

Henryk KRAWCZYK, Tomasz DZIUBICH

Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki

## IMPLEMENTACJA MODELU DIAGNOSTYKI SYSTEMOWEJ W SIECI KOMPUTEROWEJ

**Streszczenie.** W pracy przedstawiono nowy model diagnostyki systemowej dla typowej lokalnej sieci komputerowej. Rozpatrzono strategię grupową, w której testowanie polega na porównaniu rezultatów otrzymanych przez grupę węzłów realizujących typowe zadania obliczeniowe. Proponowaną strategię zaimplementowano i przebadano dla sieci lokalnej złożonej ze stacji roboczych typu HP.

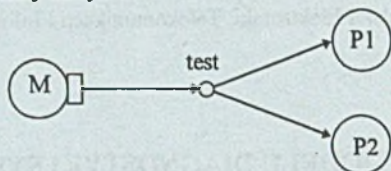
## IMPLEMENTATION OF SYSTEM-LEVEL DIAGNOSIS MODEL OF A COMPUTER NETWORK

**Summary.** This paper presents a new system-level diagnosis model for local area networks. The group diagnosis strategy is proposed and comparison testing is used to identify faulty units. This strategy was implemented and evaluated for LAN's consisted of HP workstations.

### 1. Wprowadzenie

Diagnostyka sieci komputerowej [9] jest na ogół realizowana przez administratora, który ma do dyspozycji odpowiedni zestaw testów sprawdzających różne elementy sieci, jak: łącza fizyczne, stacje robocze czy aplikacje. W literaturze znane są od dawna tzw. modele diagnostyki systemowej [4], które określają możliwości samodiagnozy systemu komputerowego poprzez ocenę jednych jednostek systemu stacji roboczych przez inne

jednostki. W pracy przyjęto uogólniony model Holta-Smitha [5], w którym testowanie odbywa się poprzez wykonanie tego samego zadania na kilku stacjach roboczych i porównaniu otrzymanych wyników. Niech  $M$  oznacza jednostkę systemu odpowiedzialną za wykonanie testowania,  $P_i$  jednostki testowane. Wówczas prosty test sprawdzający można przedstawić w sposób ukazany na rys. 1.



Rys. 1. Model testu dla sieci komputerowej

Fig. 1. Test model for a computer network

Wyniki testowania określa zaś tabela 1, gdzie stan każdej jednostki sieci (stacja robocza, węzeł sieci) określa się binarnie:

$$\text{stan jednostki} = \begin{cases} 0 - \text{gdy jednostka funkcjonuje prawidłowo} \\ 1 - \text{gdy jednostka funkcjonuje nieprawidłowo} \end{cases} \quad (1)$$

Tabela 1

Wyniki testowania dla testu z rys. 1

Stany jednostek			Decyzja diagnostyczna wynikająca z modelu
M	P1	P2	
0	0	0	wszystkie jednostki sprawne
0	0	1	P2 - uszkodzona
0	1	0	P1 - uszkodzona
0	1	1	uszkodzona P1 lub P2, lub obie
1	0	0	sytuacja nieokreślona
1	0	1	sytuacja nieokreślona
1	1	0	sytuacja nieokreślona
1	1	1	sytuacja nieokreślona

Rolę jednostki testującej (typu master) może spełniać każda stacja robocza sieci. Wówczas cała procedura testowania polega na następujących krokach.

#### Procedura diagnostyczna:

1. Utwórz grupę jak na rys. 1 i dokonaj oceny funkcjonowania sieci.



2. Przekazać wynik testowania (decyzje diagnostyczne) innym jednostkom.

3. Na podstawie zebranych wyników testowania określić stan sieci.

Procedura ta nie określa ani sposobu tworzenia grupy, ani kolejności testowania w grupie.

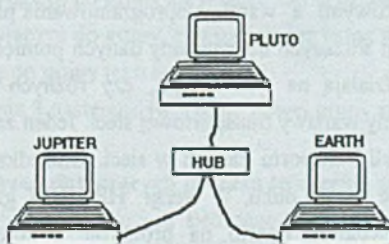
Na ogół wyróżnia się dwie strategię [5].

- 1) **grupową**, w której w sieci o liczbie stacji  $n$  wyróżnia się co najwyżej  $\lceil n/3 \rceil$  grup, które równolegle określają stan sieci,
- 2) **postępującą**, w której tworzy się jedną grupę, dla której testowanie nie wykrywa błędów i następne jednostki tej grupy są jednostkami nadrzędnymi w nowych trzech utworzonych grupach. W ten sposób w każdym kroku następnym do każdej grupy angażuje się po jednej jednostce uznanej za poprawną w kroku poprzednim.

Niniejsza praca dotyczy implementacji procedur diagnostycznych dotyczących tylko pojedynczej grupy. Opracowano programy dla jednostki zarządzającej i jednostek podrzędnych. Przeanalizowano ich zachowanie w sytuacjach pojedynczego uszkodzenia węzła bądź łącza fizycznego.

## 2. Charakterystyka środowiska sieciowego

Program diagnostyczny został wykonany i testowany w środowisku sieci lokalnej LAN składającej się z trzech stacji roboczych HP 712/60. Stacje te, wyposażone w karty sieciowe firmy 3COM, tworzą fizycznie gwiazdę (jak na rys.2). Metoda dostępu do medium oparta jest na standardzie 802.3 IEEE (Ethernet). Elementem łączącym komputery jest hub HP 2610A o ośmiu portach wejściowych. Taka konfiguracja nosi miano 10Base T (max. prędkość transmisji 10 Mb/sek, transmisja w paśmie podstawowym - cyfrowa, medium transmisyjne - skrętka miedziana). Na każdej ze stacji o nazwach: Pluto, Earth, Jupiter jest zainstalowany system HP-UX 7.0.

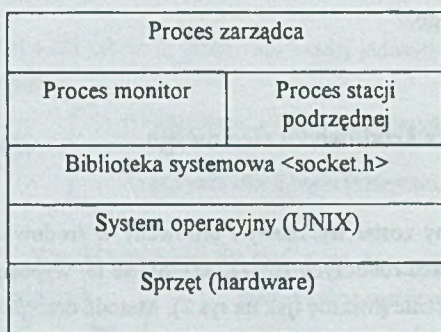


Rys. 2. Topologia środowiska sieciowego  
Fig.2. The considered network hardware architecture

Z topologią sieci wiązą się bezpośrednio rodzaje uszkodzeń fizycznych, mogących występować w sieci [8]:

- Awaria magistrali "kręgosłupa" (backbone) - co oznacza całkowity upadek sieci. Ramki wysyłane przez stacje są zagłuszane przez siebie same (zjawisko odbicia fali), ponieważ brak na jednym z końców pochłaniacza (terminatora) - awaria huba.
- Awaria między przekaźnikiem AUI a stacją lub w złączu karta sieciowa - przewód (T - connector) - stacja nie może odbierać ramek (przerwa w obwodzie). Nadawane ramki natomiast nie dochodzą do innych stacji. Pozostałe stacje pracują bez zakłóceń.
- Awaria karty sieciowej - brak możliwości nadawania i odbioru ramek. Praca innych stacji bez zmian.
- Awaria stacji roboczej - brak komunikacji z uszkodzoną stacją roboczą

Struktura przyjętego oprogramowania jest przedstawiona na rys. 3.



Rys. 3. Architektura przyjętego oprogramowania sieci  
Fig. 3. The assumed network software architecture

Oprogramowanie zostało wykonane przy użyciu gniazdek (socket) [3,6]. Gniazdko stanowi API (Application Program Interface), czyli warstwę oprogramowania leżącą pomiędzy programami użytkowymi a warstwą oprogramowania protokołu sieci. Biblioteka gniazdek jest to zbiór funkcji służących do wymiany danych pomiędzy procesami niezależnie od tego, czy procesy te działają na tym samym, czy różnych komputerach. Biblioteka gniazdek maskuje mechanizmy warstwy transportowej sieci. Jeden zapis lub odczyt z gniazdko może powodować wiele cykli transportu danych w sieci. Gniazdko pozwalają na dostęp do sieci na takiej samej zasadzie jak do pliku. W wersji HP-UNIX gniazdko są realizowane w strukturze jądra systemu. Program oparto na protokole z kontrolą połączenia. Użycie gniazdek w trybie pracy klient-serwer z kontrolą połączenia polega na tym, że:

#### KLIENT

- tworzy gniazdko,



- łączy się z serwerem podając adres jego gniazdka (na adres ten składa się adres sieci komputera serwera oraz numer portu komunikacyjnego program usługowego serwera),
- pisze lub czyta dane z gniazdka,
- zamyka gniazdko.

#### SERWER

- tworzy gniazdko,
- nadaje adres gniazdka (adres sieci komputera oraz numer portu komunikacyjnego usługi) - binding,
- czeka na żądania połączeń nadchodzące od klientów,
- dla każdego odebranego żądania połączenia ustanawia połączenie, pisze lub czyta poprzez gniazdko, zamyka gniazdko obsługujące połączenie.

Zadaniem procesu stacji podrzędnej jest przetworzenie otrzymanego od stacji Master zadania i wysłanie obliczonego wyniku z powrotem do stacji Master. Ze strony stacji Master działa proces monitor przekazujący otrzymane wyniki do procesu zarządcy. Ten, na podstawie otrzymanych rezultatów, podejmuje decyzję i ewentualnie sam przeprowadza obliczenia.

### 3. Koncepcja samodiagnozy w grupie

Program został zaimplementowany przy następujących założeniach projektowych:

- W komunikacji biorą udział po trzy stacje w grupie.
- Monitor jest stacją porównującą wyniki odebrane od pozostałych stacji.
- Podstacja jest stacją przeprowadzającą proces samotestowania i wysyłającą wyniki.
- Przez nazwę stacji należy rozumieć jej nazwę sieciową.

Zasady tworzenia grupy zostały rozbite na 3 etapy:

- a. Stacja chce się przyłączyć do grupy, ale jeszcze żadna grupa nie istnieje.
- b. Stacja chce się przyłączyć do grupy, gdy istnieje już jedna stacja w grupie.
- c. Stacja przyłącza się do grupy jako trzecia.

Przy ilości stacji większej niż 3 następuje tworzenie nowej grupy również według powyższych zasad.

#### Opis poszczególnych zdarzeń dotyczących procesu tworzenia grupy

- a. W momencie uruchomienia procesu śledzącego pracę stacji zostaje wysłane żądanie przyłączenia do grupy - CONNECT (nazwa stacji). Jednocześnie ustawiony jest Timeout. W przypadku upływu czasu i nieodebrania ramki potwierdzającej przyłączenie do grupy następuje powtórka powyższej akcji. Próba jest kontynuowana 10-krotnie.

- W przypadku braku jakiejkolwiek odpowiedzi z sieci stacja staje się MONITOREM. Następuje wówczas oczekiwanie na odbiór ramki CONNECT (nazwa stacji) od innych stacji (patrz rys.4).
- W przypadku odebrania potwierdzenia ACK\_GRP (nazwa monitora, liczba stacji) następuje przyłączenie stacji do grupy i przejście do punktu b) lub c) w zależności od odebranego parametru liczba stacji.

b. Stacja znajduje się w tym stanie, jeśli parametr liczba stacji=1. Zarówno stacja podrzędna, jak i monitor wykonują to samo zadanie, a wyniki są oceniane przez monitor. W przypadku jakiejkolwiek niezgodności następuje rozwiązanie grupy.

c. Stacja znajduje się w tym stanie, jeśli parametr liczba stacji=2. Monitor przestaje wykonywać przesyłane zadania. Stacje po wykonaniu narzuconych zadań przysyłają wyniki do monitora. Monitor ocenia i wysyła odpowiedź pozytywną lub negatywną.

- Jeśli otrzymamy obie odpowiedzi poprawne - to monitor wysyła kolejne zadania do realizacji przez stacje.
- Jeśli otrzymamy jedną odpowiedź negatywną - to nastąpi wyłączenie z grupy stacji, która przesłała złą odpowiedź. W takim wypadku monitor oczekuje na przyłączenie innej stacji.
- Jeśli żadne z dwóch otrzymanych odpowiedzi nie są jednakowe, następuje całkowite rozwiązanie grupy.

Stacja-monitor w fazie tworzenia grupy oczekuje na chęć zgłoszenia się do grupy pewnej stacji. Gdy otrzymuje sygnał CONNECT z nazwą stacji, to wysyła potwierdzenie przyłączenia do grupy ACK\_GRP ze swoją nazwą. Otrzymanie potwierdzenia dołączenia do grupy GROUP powoduje wysłanie zadania do stacji. Jednocześnie zostaje ustawiony budzik odliczający czas działania stacji jako monitora. W tym stanie można otrzymać następujące ramki:

- Result - stacja, do której wysłano zadanie, przysyła odpowiedź, a monitor zależnie od wielkości grupy podejmuje różne działania.

⇒ Grupa dwóch stacji: Monitor również sam wykonuje zadanie i swój wynik przyjmuje do porównań. Jeżeli wyniki są poprawne, następuje ponowne wysłanie zadania. Jeśli wyniki są różne, następuje wysłanie DISCONNECT i rozłączenie grupy, z jednoczesnym zastrzeżeniem, że stacje te następnym razem nie mogą być w grupie.

⇒ Grupa trzech stacji: Najpierw monitor oczekuje na otrzymanie wyniku od dwóch stacji podrzędnych. Następnie porównuje te wyniki. Jeśli są jednakowe, następuje wysłanie do obu stacji następnego zadania. Jeśli wyniki są różne, monitor wylicza zadanie i porównuje otrzymany wynik z odebranymi. Tej stacji, która przesłała taki sam rezultat, wysyła następne zadanie, drugiej ramkę DISCONNECT. Następnie przechodzi do pracy w trybie



dwóch stacji. Jeśli natomiast wszystkie 3 rezultaty są różne, wysyła do obydwóch stacji ramki DISCONNECT i przechodzi do oczekiwania na przyłączenie do innej grupy z zastrzeżeniem, że stacje te następnym razem nie mogą być w grupie.

- CONNECT - oznacza, że pewna stacja chce się przyłączyć do grupy. Jeżeli ilość stacji w grupie jest =1 lub 2, następuje wysłanie ACK\_GRP i przejście do pracy w grupie 3 stacji.

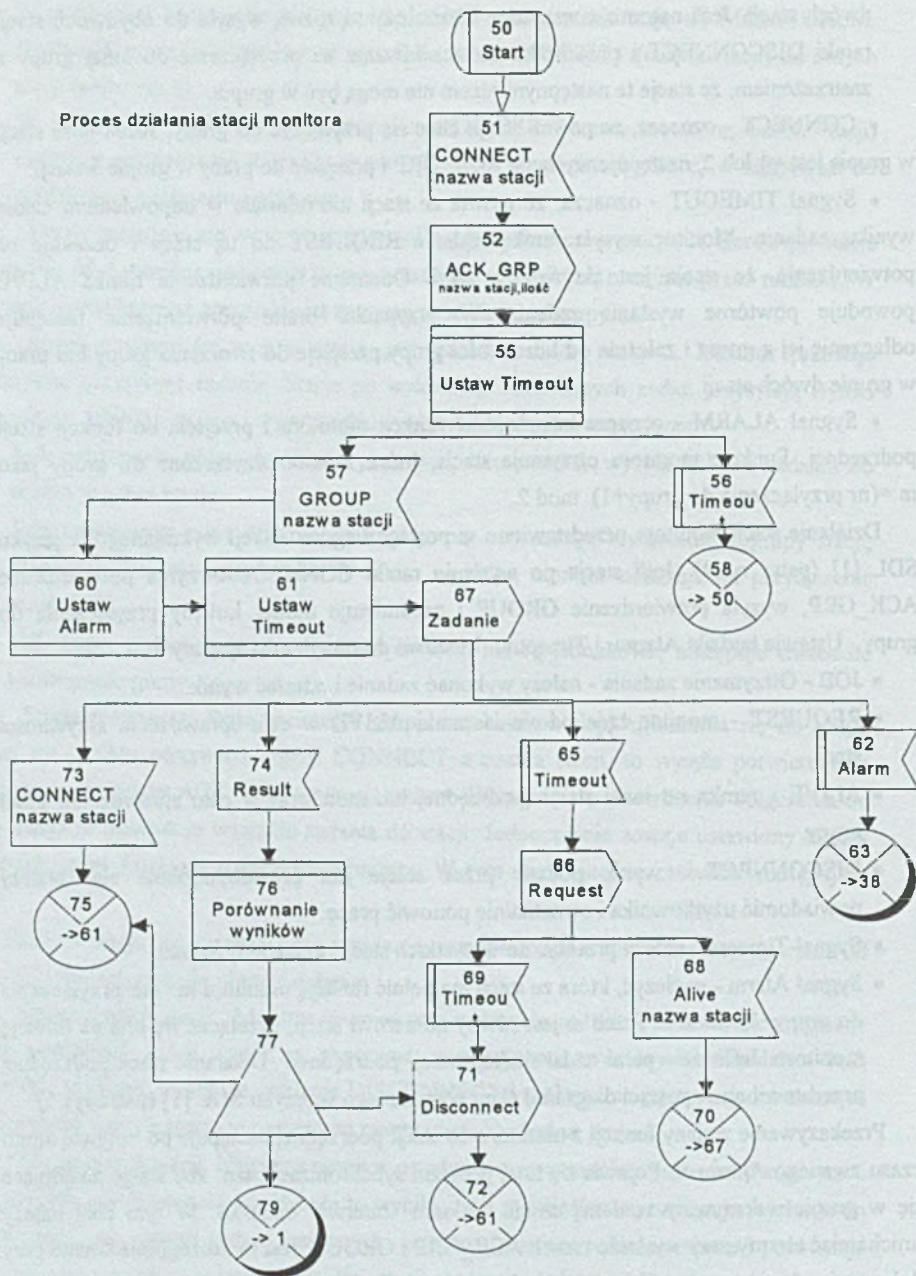
- Sygnał TIMEOUT - oznacza, że pewna ze stacji nie odesłała w odpowiednim czasie wyniku zadania. Monitor wysyła ramkę zapytania REQUEST do tej stacji i oczekuje na potwierdzenie, że stacja jest aktywna w sieci. Odebranie potwierdzenia ramka ALIVE powoduje powtórne wysłanie zadania. W przypadku braku potwierdzenia następuje odłączenie jej z grupy i zależnie od liczebności grupy przejście do tworzenia grupy lub pracy w grupie dwóch stacji.

- Sygnał ALARM - oznacza zakończenie funkcji monitora i przejścia do funkcji stacji podrzędnej. Funkcję monitora otrzymuje stacja, która została przyłączona do grupy jako  $m:=(nr \text{ przyłączenia do grupy}+1) \bmod 2$ .

Działanie stacji monitora przedstawiono w postaci diagramu akcji wykonanego w języku SDL [1] (patrz rys.4). Jeśli stacja po wysłaniu ramki CONNECT otrzyma potwierdzenie ACK\_GRP, wysyła potwierdzenie GROUP i zapamiętuje numer kolejny przyłączenia do grupy. Ustawia budziki Alarmu i Timeoutu. Możliwe do otrzymania sygnały to:

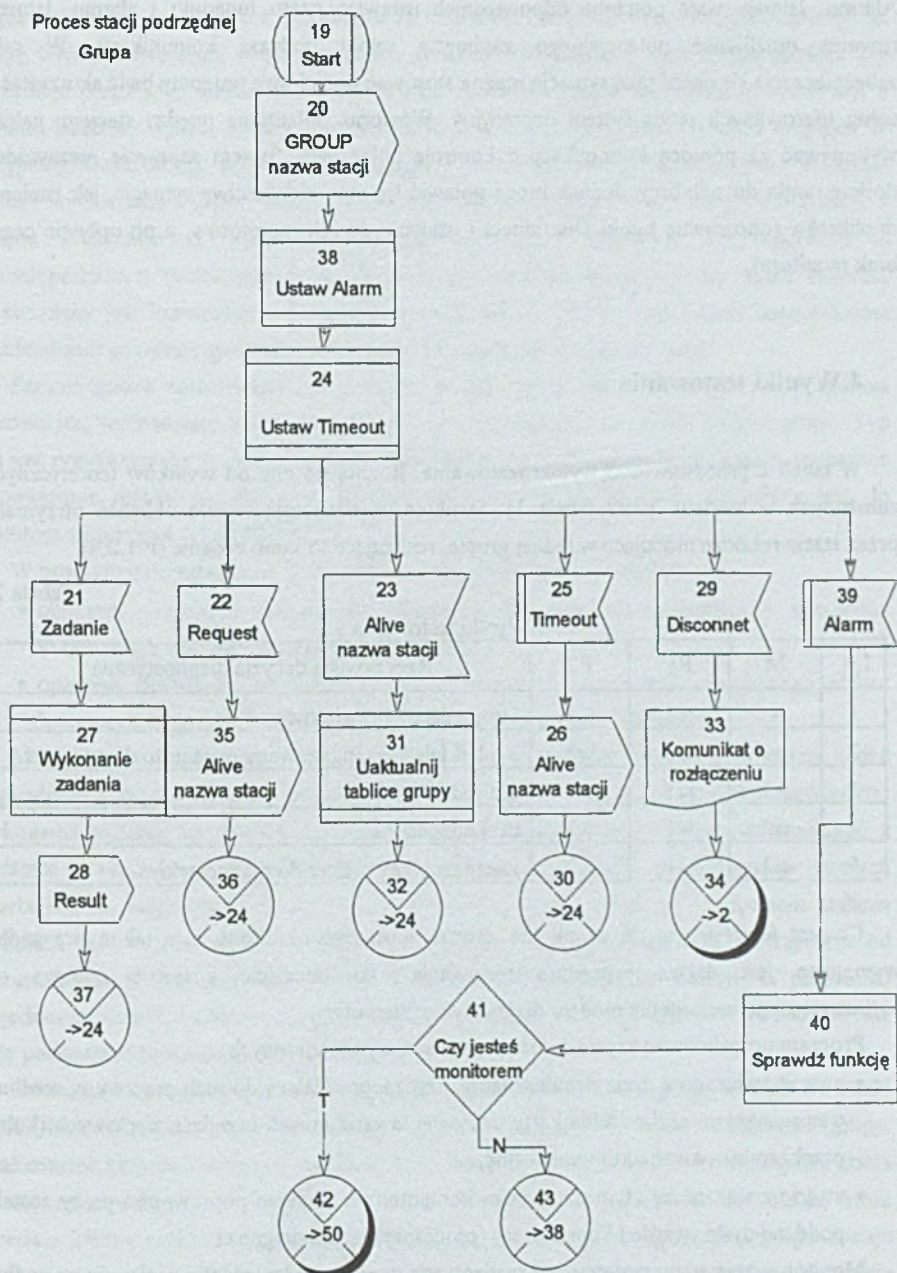
- JOB - Otrzymanie zadania - należy wykonać zadanie i odesłać wynik.
- REQUEST - monitor żąda odesłania ramki ALIVE w celu sprawdzenia aktywności stacji.
- ALIVE - ramka od innej stacji (podrzędnej lub monitora) w celu sprawdzenia stanu grupy.
- DISCONNECT - wynik podany przez stację jest prawdopodobnie zły. Należy powiadomić użytkownika i ewentualnie ponowić pracę.
- Sygnał Timeout - należy przesłać do wszystkich stacji ramkę aktywności.
- Sygnał Alarm - wyliczyć, która ze stacji ma pełnić funkcję monitora  $m:=(nr \text{ przyłączenia do grupy}+1) \bmod 2$ . Jeżeli  $m$  jest równy numerowi stacji, przełącza się ona na funkcję monitora. Jeśli nie - pełni nadal funkcję stacji podrzędnej. Działanie stacji podrzędnej przedstawiono w postaci diagramu akcji wykonanego w języku SDL [1] (patrz rys.5).

Przekazywanie zmiany funkcji z monitora do stacji podrzędnej następuje po upływie limitu czasu zwanego Alarmem. Pojawia się tutaj problem synchronizacji, tzn. aby stacje znajdujące się w grupie wskazywały w danej chwili ten sam interwał czasowy. W tym celu należy uruchamiać alarmy przy wysłaniu ramek ACK\_GRP i GROUP. Jest to szczególnie ważne przy dołączaniu do grupy trzeciej stacji, która może wyrazić chęć przyłączenia do grupy w dowolnym momencie czasu. Ramkę GROUP należy wysłać w momencie upływu czasu



Rys.4. Działanie stacji monitora - diagram w języku SDL  
 Fig. 4. Testing strategy algorithm for a monitor - SDL diagram





Rys.5. Działanie stacji podrzędnej - diagram w języku SDL  
Fig.5. Testing strategy algorithm for a substation - SDL diagram!

Alarmu. Istnieje więc potrzeba odpowiednich ustawień czasu timeoutu i alarmu. Istnieje również możliwość potencjalnego zaginięcia ramki podczas komunikacji. W celu zabezpieczenia się przed taką sytuacją można stosować dodatkowe timeouty bądź skorzystać z usług oferowanych przez system operacyjny. Wówczas połączenia między stacjami należy wykonywać za pomocą komunikacji z kontrolą połączenia. System zapewnia niezawodne dojście ramki do odbiorcy. Jednak mogą pojawić się inne niewłaściwe sytuacje, jak istnienie duplikatów (opóźnienie ramki Disconnect i istnienie dwóch monitorów, a po upływie czasu brak monitora).

#### 4. Wyniki testowania

W tabeli 2 przedstawiono wyniki testowania. Różnią się one od wyników teoretycznych założonych w modelu (patrz tabela 1). Symbole  $w_i$  oznaczają wyniki obliczeń otrzymane przez stacje robocze pracujące w jednej grupie, realizujące to samo zadanie ( $i=1,2,3$ ).

Tabela 2

Wyniki testowania

Lp.	M	P1	P2	Rzeczywista decyzja diagnostyczna
1	-	w1	w1	Grupa działa prawidłowo
2	-	w1	w2	3 lub 4 lub 5, zaangażowany master do diagnostyki
3	w1	w1	w2	uszkodzona P1
4	w2	w1	w2	uszkodzona P2
5	w3	w1	w2	zaangażowany master do diagnostyki

Co jest interesujące, że wynik nie zawsze musi być określony i w takim przypadku wymagana jest dalsza procedura testowania. To podejście znacznie odbiega od rozpatrywanego we wstępie modelu diagnostyki systemowej.

Program umożliwia wykrycie dwóch typów sytuacji wyjątkowych:

- awarie związane z uszkodzeniem łącza fizycznego. Należą do nich przerwy w medium transmisyjnym, uszkodzenia karty sieciowej, awarie innych urządzeń sieciowych (huby, przełączniki) - awarie komunikacyjne,
- awarie związane ze złym działaniem komputerów. Badaniu poprawności pracy zostały poddane dyski twarde i koprocesory (procesory) - samodiagnoza.

Monitorowanie stanu systemu jest realizowane poprzez badanie stanu zapisu do gniazdka. Jeśli jest on możliwy, to znaczy, że istnieje stacja, z którą można się połączyć. W przypadku



poprawnej pracy możliwe jest połączenie ze wszystkimi stacjami, które w posiadanej przez stację tablicy **hosts** mają ustawiony status podstacji lub monitora. Jeśli stacja monitor nie otrzymuje przez czas **alarm** (który upłynął od wysłania zadania do podstacji) informacji o wyniku zadania, wysyła żądanie potwierdzenia o aktywności w sieci - ramka Request. W przypadku trzykrotnego wysłania tego żądania i nieotrzymania odpowiedzi Alive - podstacja zostaje wykluczona z grupy. Zostaje wysłana do niej ramka Disconnect, zaś do innej stacji w grupie informacja o wyłączeniu tej podstacji oraz komunikat dla operatora o prawdopodobnym rozłączeniu stacji. W przypadku jeśli stacja odpowie na ramkę Request, wyświetlany jest komunikat o kłopotach z połączeniem. Przy dużej liczbie komunikatów administrator powinien sprawdzić stan łącz, konfigurację i obciążenie sieci.

Samodiagnoza realizowana jest poprzez wykonywanie zadań testowych. Typ zadania testowe jest wyznaczany przez stację Monitor i rozsyłany do wszystkich stacji z grupy. Typ ten jest przykazywany za pomocą ramki JOB. Stacja, po otrzymaniu typu zadania, wykonuje odpowiednie działanie zależne od rodzaju zadania. Wynik działania przysyłany jest do monitora za pomocą ramki RESULT.

W programie dopuszczalne są dwa typy zadań, sprawdzające poprawność:

- obliczeń - zadanie polega na wykonaniu operacji matematycznej i przesłaniu otrzymanego wyniku jako rezultatu działania,
- operacje dyskowe - zadanie polega na odczytaniu wielkości określonego zbioru dyskowego i przesłaniu odczytanej wielkości jako rezultatu operacji.

Możliwe jest rozszerzenie liczby rodzajów zadań testowych. W tym celu należy nadać odpowiedni (kolejny) numer typu zadania i dodać do programu odpowiednie procedury: wykonania zadania (w procedurze obsługi podstacji) oraz oceny wyniku otrzymanego z podstacji (w procedurze obsługi monitora). Zastosowana strategia samodiagnozy systemu oparta jest na diagnozie rozproszonej grupowej, tzn. każda stacja po otrzymaniu zadania wykonuje je, a wynik odsyła do stacji Monitora. Monitor porównuje wyniki otrzymane od wszystkich stacji. Jeśli otrzymane wyniki są zgodne, wysyła następne zadanie. W przypadku niejednoznaczności (każda ze stacji przesłała odmienny wynik) sam wykonuje to zadanie i na jego podstawie przyjmuje poprawny rezultat.

Stacja, która przesłała niepoprawny rezultat zostaje wyłączona z grupy (przesłanie ramki DISCONNECT). Gdy wszystkie wyniki są różne, następuje całkowite rozłączenie grupy. Realizowane zadania muszą być ponownie wykonane. Przy następnym tworzeniu grupy stacja ta nie zostanie włączona do grupy, w której poprzednio uczestniczyła, tzn. stacja, która przesłała błędny rezultat nie może pracować ponownie w tej samej grupie. Grupa zostaje także rozłączona po wielokrotnym uruchomieniu procesu monitora. Ma to wymusić tworzenie różnych grup w środowisku sieciowym o większej ilości stacji.

Tabela 3

## Sygnalizacja stanów awaryjnych

	Komunikacja			Diagnostyka	
Typ awarii	Uszkodzenie łącza transmisyjnego	Uszkodzenie kart sieciowych	Awaria koncentratora	Błędy przetwarzania	Błędy pamięci dyskowej
Sposób detekcji	Ramki IS_ALIVE, timeouty, informacja od systemu	Programy diagnostyczne producenta interfejsu	Ramki IS_ALIVE, timeouty, informacja od systemu	Zadania testujące (obliczenia, zmienno-przecinkowe)	Zadania testujące (odczyt zapis plików)
Sygnalizacja stanu awaryjnego	Informacja na ekranie stacji, na której łączy wystąpiło uszkodzenie	Komunikat na ekranie odłączenie stacji z grupy	Komunikat na wszystkich stacjach sieciowych	Komunikat na ekranie odłączenie stacji z grupy	Komunikat na ekranie odłączenie stacji z grupy

W przypadku wysłania przez monitor zadania JOB do stacji podrzędnej i nieotrzymania przez określony czas odpowiedzi (RESULT) następuje monitorowanie stanu stacji (wysyłanie ramek REQUEST i ALIVE). W przypadku braku odpowiedzi następuje wyłączenie stacji z grupy. Zestawienie wykrywalnych stanów awaryjnych, ich detekcji i sygnalizacji przedstawia tabela 3.

## 5. Uwagi końcowe

Przy implementacji testów wykonano nieskomplikowane zadania sprawdzające. W praktyce należałoby użyć większej gamy zadań testowych, o większym zakresie działania, tzn. większym współczynniku pokrycia uszkodzeń. Najprostszym sposobem byłoby uruchomienie poprzez funkcję `exec` pakietu programów testujących i odebranie kodu zakończenia procesu.

Możliwe jest również zwiększenie stacji w grupie do  $N$  jednostek. Należy wówczas powiększyć liczbę obiegów pętli w procesie zarządcy, który wywołuje odpowiednią liczbę



procesów monitorów do komunikacji z podstacjami oraz liczbę stacji, które mogą być przyłączane do grupy. Jednocześnie wymagana jest zmiana procedury decyzyjnej określającej, które z otrzymanych odpowiedzi są poprawne, czyli które ze stacji działają poprawnie, a które błędnie. Opracowanie w pełni profesjonalnego programu wymagałoby również usprawnienia komunikacji międzystacyjnej. Wykorzystać można byłoby ramki okólnikowe (broadcast), co wymaga zgody administratora. Zlikwidowałoby to utrudnienie polegające na podawaniu wszystkich stacji w sieci. Dalszą modyfikacją jest uzupełnienie programu o możliwość archiwizacji zadań i plików. Możliwe wykonanie to okresowe zapisywanie pliku „core” i archiwizacja przyrostowa plików. Należałoby także zwiększyć liczbę testów styku sieciowego (karty sieciowe, protokoły). Testy takie jednak są zależne od typu karty i jej producenta. Zazwyczaj z kartą dostarczane są sterowniki i programy testujące. Takie programy należałoby dołączyć do procedur testujących w celu dokładnego zlokalizowania błędu lub uszkodzenia.

## LITERATURA

- [1] Belina F., Hogrefe D. and Sarina A.: *SDL With Applications From Protocol Specification* Prentice Hall 1991.
- [2] Frisch E.: *UNIX – Administracja systemu. ReadMe*, Warszawa 1996
- [3] Gabassi M., Dupouy B.: *Przetwarzanie rozproszone w systemie UNIX. LUPUS* Warszawa 1995
- [4] Krawczyk H., AL-Karaawy S.: *The Intermediate Model for System Level Fault Diagnosis. Proc. of System, Modelling, Control Conference*, 1-5 May, Zakopane, 1995, pp. 451-456.
- [5] Krawczyk H.: *Rozproszona diagnoza sieci komputerowych. Materiały krajowego sympozjum telekomunikacji KST'89. Bydgoszcz, 6-8 września 1989*, s. 200-206.
- [6] Krawczyk H., Umiński P.W.: *Strategie testowania lokalnych sieci komputerowych. Materiały Krajowej Konferencji POLMAN'96. Poznań 15-17 kwietnia, 1996*, s. 285.
- [7] Rochkind M. J.: *Programowanie w systemie UNIX. WNT, Warszawa 1993.*
- [8] Sheldon T.: *Lan Times Encyclopedia of Networking. McGraw-Hill Inc. 1994.*

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 21 listopada 1996 r.

## Abstract

The paper presents a new system-level diagnosis model devoted to local area networks. This model is shown in Fig. 1 and Table 1. The comparison testing is executed inside groups of nodes. Only one node in each group plays the role of a decision unit. If outcomes obtained inside a group are unreliable, the group is canceled and a new one consisting of different combination of nodes is created. Therefore definitions of outcomes are modified and shown in Table 2. In consequence, testing strategy as adapted to a real situation occurred in the network. The considered architecture of a computer network is given in Fig. 3. The procedures of monitor and substation are presented in Fig. 4 and Fig. 5, respectively. The proposed strategy was implemented and evaluated in LAN made up of HP workstations (see Fig. 2). The general diagnostic capability of the proposed strategy are shown in Table 3.