



## Implementacja algorytmów diagnostycznych w sieciach o strukturze sześciangu

JAN CHUDZIKIEWICZ

Wojskowa Akademia Techniczna, Wydział Cybernetyki, Instytut Teleinformatyki i Automatyki,  
Zakład Systemów Komputerowych, 00-908 Warszawa, ul. S. Kaliskiego 2

**Streszczenie.** W pracy przedstawiono sposób implementacji algorytmów diagnozowania sieci komputerowych (wieloprocesorowych). W wykorzystanym rozwiązaniu zastosowano sterownik protokołu. Sterownik ten współpracuje z biblioteką DLL implementującą algorytm diagnostyczny. Przedstawiono również, bazując na systemie operacyjnym klasy Windows®, narzędzia i mechanizmy wbudowane w system, które ułatwią implementację algorytmów. Ponadto zaprezentowano sposób modelowania sieci typu  $n$ -wymiarowego sześciangu w notacji UML.

**Słowa kluczowe:** hipersześciang, samodiagnozowanie, systemy tolerujące błędy, algorytmy diagnostyczne

**Symbole UKD:** 681.324

### Wprowadzenie

Systemy zdolne do wykonywania określonych w specyfikacji zadań, pomimo wystąpienia pewnych niezdatności, nazywa się systemami tolerującymi uszkodzenia [1, 9, 10]. Jednym z warunków, które muszą spełniać struktury wykorzystywane w systemach tolerujących uszkodzenia, jest redundancja elementów systemu, czyli zastosowanie struktur nadmiarowych (patrz określenie 2.4). Przykładem struktury, która zapewnia odpowiednią liczbę linii komunikacyjnych, jest struktura  $n$ -wymiarowego hipersześciangu  $H^n$  [1],  $n$ -wymiarowym hipersześciangiem binarnym jest graf zwykły  $G$  ( $G = \langle E, U \rangle, |E| = 2^n, |U| = n \cdot 2^{n-1}$ ). Kolejnym warunkiem tolerowania uszkodzeń jest ich poprawna diagnostyka. Jej jakość ma decydujące znaczenie dla przywrócenia zdatności systemu poprzez wymianę uszkodzonych jednostek bądź też izolację niezdatnych jednostek i przeprowadzenie rekonfiguracji

zadań (łagodna degradacja systemu) [2, 3, 9]. Wymaga to stosowania efektywnych metod diagnozowania [4, 7]. W przypadku systemów z przetwarzaniem rozproszonym można zastosować metody wykorzystujące rezultaty wzajemnego testowania się elementów systemu [14, 15, 16]. Wnioskowanie o stanie niezawodnościowym systemu odbywa się w tym przypadku na podstawie wyników wszystkich możliwych testowań (opiniowanie diagnostyczne). Po zidentyfikowaniu wszystkich niezdatnych jednostek systemu może nastąpić jego rekonfiguracja (wyznaczenie nowego przydziału zadań dla jednostek przetwarzających) z uwzględnieniem aktualnego stanu niezawodnościowego systemu.

W niniejszym artykule przedstawiono rozwiązanie bazujące na protokole NDIS dla implementacji algorytmów diagnozowania sieci komputerowych (wieloprocesorowych) [3]. W zastosowanym rozwiązaniu wykorzystano sterownik protokołu współpracujący z biblioteką DLL implementującą algorytm diagnostyczny.

## 2. Pojęcia podstawowe

**Określenie 2.1.** Mówimy, że graf  $G$  ( $G = \langle E, U \rangle$ ) należy do klasy  $G^n$  grafów zwykłych, jeżeli  $|E| = 2^n$ ,  $\mu(e) = n$  ( $e \in E$ ) oraz  $|C^k(G)| = \binom{n}{k} \cdot 2^{n-2}$ ,

gdzie:  $C_k(G)$  oznacza zbiór cykli prostych w grafie  $G$ , z których każdy zawiera  $k$  krawędzi ( $3 \leq k \leq |E|$ ).

**Własność 2.1.** Węzły grafu klasy  $G^n$  można zaetykietować  $n$ -wymiarowymi wektorami binarnymi tak, że odległość Hamminga między wektorami opisującymi węzły przyległe będzie równa 1.

**Określenie 2.2.**  $n$ -wymiarowym sześcianem binarnym nazywamy graf zwykły  $G'$  ( $G' = \langle E, U' \rangle$ ,  $|E| = 2^n$ ,  $|U'| = n \cdot 2^{n-1}$ ) o  $2^n$  węzłach, z których każdy opisany jest odpowiednim wektorem binarnym  $z$  ( $z = (z_1, \dots, z_n)$ ,  $z_i \in \{0, 1\}$ ,  $1 \leq i \leq n$ ,  $z \in Z^n$ ,  $|Z^n| = 2^n$ ) oraz o  $n \cdot 2^{n-1}$  krawędziach, łączących te węzły, których opisujące je wektory odległe są o 1 według miary Hamminga.

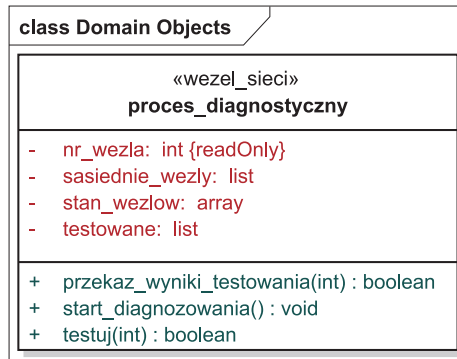
**Określenie 2.3.** Mówimy, że sieć komputerowa jest typu  $n$ -wymiarowego ( $n \geq 2$ ) sześcianu binarnego lub że jest osadzona (zagnieżdżona) w  $H^n$ , jeżeli graf  $G$  ( $G = \langle E, U \rangle$ ) opisujący jej strukturę logiczną jest spójnym grafem częściowym grafu  $H^n$ , takim że:

$$2^n - 1 \leq |U| \leq n \cdot 2^n - 1.$$

**Określenie 2.4.** Nadmiarową strukturą logiczną sieci komputerowej nazywamy strukturę, której graf  $G$  ( $G = \langle E, U \rangle$ ) spełnia warunek:  $|U| \geq |E|$ .

### 3. Modelowanie systemu diagnostycznego

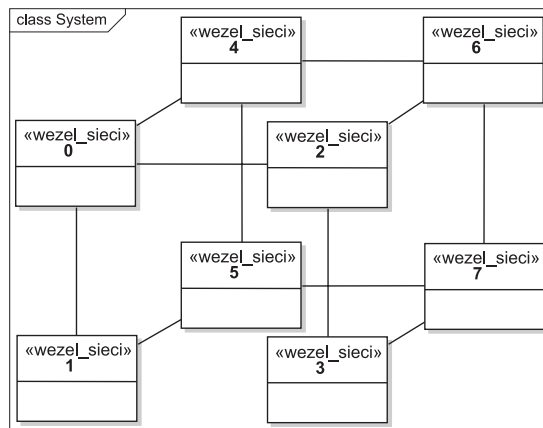
W [17] zaproponowany został sposób wykorzystania diagramów klas UML do modelowania topologii sieci wykorzystywanych w systemach rozproszonych, w tym również sieci o strukturze logicznej sześcianu. Podstawowym elementem systemu z punktu widzenia projektu jest proces diagnostyczny w każdym z węzłów sieci, dlatego też zdefiniowano klasę *proces diagnostyczny* jak na rysunku 1 [7].



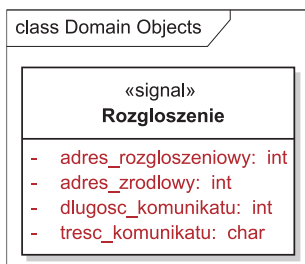
Rys. 1. Definicja klasy reprezentującej proces diagnostyczny

Mając zdefiniowany węzeł sieci, można przedstawić strukturę logiczną sieci typu sześcianu. Na rysunku 2 przedstawiono przykład zamodelowania struktury sieci dla sześcianu 3-wymiarowego (patrz określenie 2.3), bazując na zdefiniowanej klasie węzła.

Komunikacja pomiędzy węzłami realizowana jest poprzez przesyłanie komunikatów. Komunikat traktowany jest jako sygnał, który przekazywany jest przez linie komunikacyjne. Definicja komunikatów przedstawiona jest na rysunku 3.

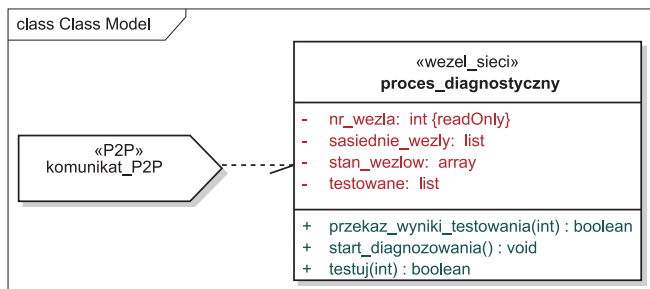


Rys. 2. Przykład zamodelowania struktury sieci typu 3-wymiarowego sześcianu

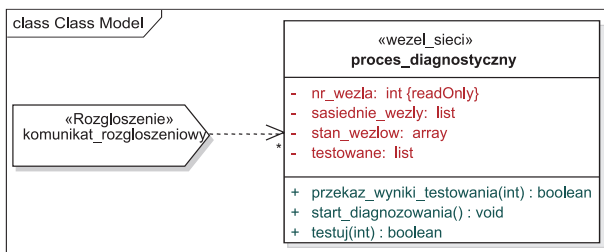


Rys. 3. Klasy reprezentujące komunikaty przesyłane przez sieć

Na rysunku 4 oraz 5 przedstawiono sposób zamodelowania komunikatów typu P2P oraz rozgłoszeniowego.

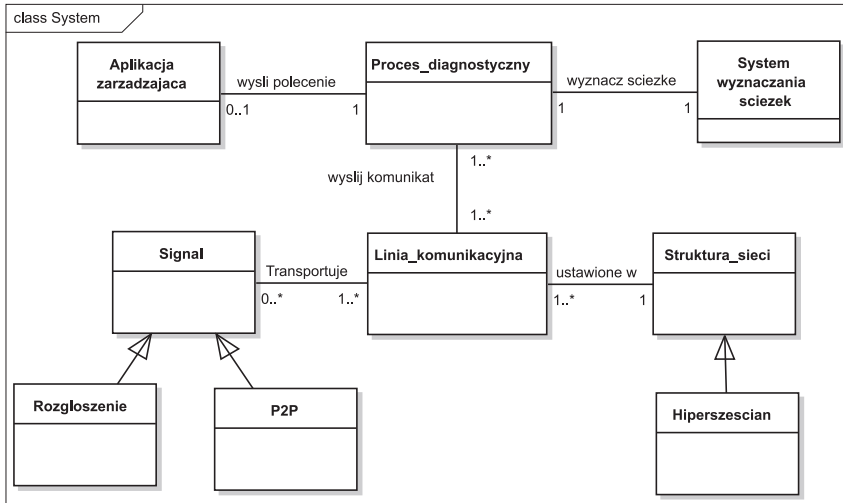


Rys. 4. Przykład modelowania komunikatu typu P2P



Rys. 5. Przykład modelowania komunikatu typu rozgłoszeniowego

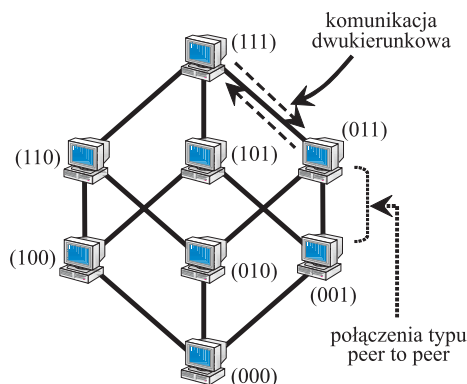
Natomiast na rysunku 6 przedstawiony został ogólny diagram klas z uwzględnieniem komunikacji w sieci o strukturze typu sześciangu.



Rys. 6. Ogólny diagram klas systemu diagnostycznego

#### 4. Implementacja w systemie Windows®

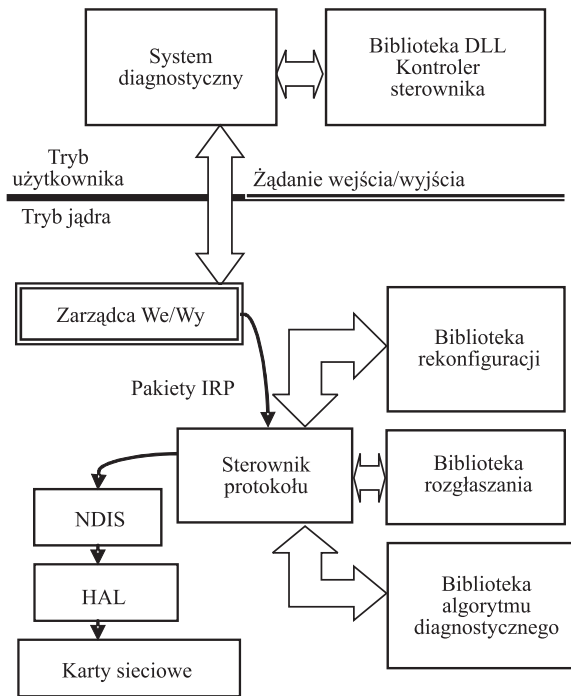
Sieć komputerowa o strukturze sześciangu jest zaliczana do sieci typu punkt-punkt (rys. 7), co powoduje, że komunikacja pomiędzy komputerami, których odległość Hamminga jest większa niż 1 (patrz określenie 2.2), wymaga pośrednictwa innych komputerów.



Rys. 7. Rozwiązanie komunikacji w sieci komputerowej o strukturze sześciangu

Rozwiązanie takie wymaga zastosowania komputerów wyposażonych w  $n$ -interfejsów sieciowych, gdzie  $n$  jest wymiarem struktury sześciangu (patrz określenie 2.3). Aby zapewnić możliwość komunikacji z wykorzystaniem wszystkich

interfejsów, należało opracować odpowiedni sterownik [3] oraz bibliotekę dynamiczną, włączając ją jednocześnie w architekturę sieciową systemu Windows® (rys. 8). Istotne znaczenie z punktu widzenia obsługi odwołań do sprzętu w architekturze Windows® odgrywa warstwa HAL (ang. *Hardware Abstraction Layer*) [11, 12]. Jest to warstwa uniezależnienia od sprzętu, jest ona warstwą kodu uniezależniająca jądro, sterowniki urządzeń oraz pozostałą część centrum wykonawczego Windows od zróżnicowanej warstwy sprzętowej. Natomiast do wymiany danych ze sterownikiem pracującym w trybie jądra wykorzystywane są struktury danych znane jako pakiety IRP (ang. *I/O request packet*).



Rys. 8. Architektura systemu Windows klasy NT

Zapewnienie efektywnego mechanizmu komunikacji na poziomie przesyłanych danych wymagało opracowania dedykowanego protokołu warstwy sieciowej. Pakiet zdefiniowano w następujący sposób: do numeracji węzłów wykorzystane zostały liczby 16-bitowe, nagłówek protokołu zawiera informacje o logicznym numerze (adresie) źródła, celu oraz o typie pakietu. Pozostałą część pakietu stanowią dane, których interpretacja zależy od typu pakietu. Postać pakietu dla opracowanego protokołu została przedstawiona w tabeli 1.

TABELA 1

Postać pakietu protokołu systemu diagnozowania

Źródło (2 bajty)	Cel (2 bajty)	Typ (1 bajt)	Dane
---------------------	------------------	-----------------	------

Użyty sterownik protokołu NDIS eksportuje funkcje ProcolXxx będące interfejsem dla niżej położonych w stosie sterowników.

Sterownik współpracuje z leżącymi niżej w stosie sterownikami miniportu lub pośrednimi, które eksportują zestaw funkcji MiniportXxx [3].

Komunikacja ta odbywa się za pośrednictwem biblioteki NDIS przez wywołanie odpowiednich jej funkcji. Na przykład dla wysyłania pakietów mogą być wykorzystane funkcje NdisSend oraz NdisSendPackets. Warto nadmienić, że biblioteka NDIS eksportuje szereg funkcji (o przedrostkach Ndis) wymaganych przy implementacji sterowników sieciowych, w znacznej mierze są to odpowiedniki funkcji systemowych, nie ogranicza to jednak możliwości korzystania z funkcji dostępnych dla wszystkich sterowników trybu jądra.

Sterowniki protokołu muszą wywołać funkcję NdisRegisterProtocolDriver w celu zarejestrowania jako protokół w bibliotece NDIS. Przekazywana jako parametr struktura NDIS\_PROTOCOL\_DRIVER\_CHARACTERISTICS powinna być zainicjalizowana przez sterownik protokołu, w szczególności ustawione muszą zostać referencje do funkcji stanowiących punkty wejścia dla NDIS.

Zgodnie z wymaganiami systemowymi sterownik implementuje wszystkie wymagane funkcje sterownika protokołu oraz funkcje pomocnicze.

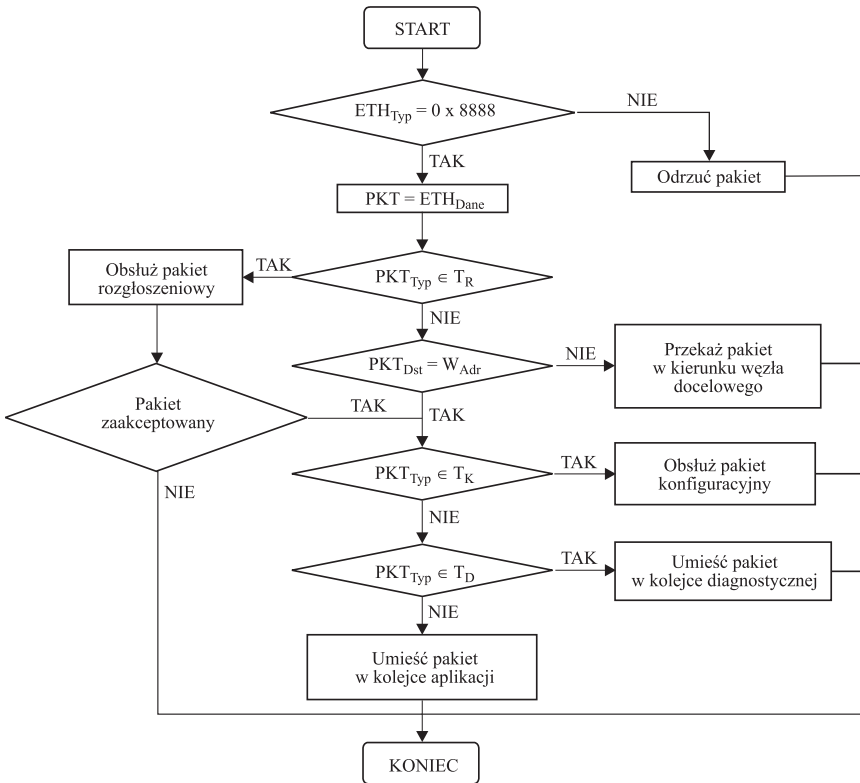
Proces inicjowania sterownika realizowany jest z poziomu systemu diagnostycznego. Po odebraniu danych konfiguracyjnych sterownik inicjalizuje struktury danych odpowiedzialne za kolejowanie otrzymywanych i wysyłanych pakietów oraz inne struktury odpowiedzialne za przechowywanie danych specyficznych dla określonego urządzenia sieciowego. Następnie zostaje ustawiony filtr pakietów — filtr ten określa, jakie pakiety powinny być przekazywane od sterownika miniportu do sterownika protokołu.

Proces filtrowania pakietów odbywa się na podstawie pola typ nagłówka protokołu.

Sterownik z innymi elementami sieci, jak już wcześniej wspomniano, komunikuje się na zasadzie przesyłania pakietów.

Do kolejki pakietów w węzle przesyłane są pakiety, które mają wartość DST\_ADDR równą numerowi logicznemu danego węzła lub pakiety rozgłoszeniowe. Wyróżnia się dwa rodzaje kolejek komunikatów: kolejka podsystemu diagnostycznego oraz kolejka aplikacji. Do pierwszej przesyłane są pakiety: diagnostyczne, konfiguracyjne oraz rozgłoszeniowe. Do kolejki aplikacji przesyłane są pozostałe pakiety, które są traktowane jako pakiety danych. Pakiety konfiguracyjne wykorzystywane są do

przesyłania poleceń do zdalnego kontrolowania stanu systemu diagnozowania. Pakiety, które są kierowane do innych węzłów, trafiają do oddzielnej kolejki obsługiwanej przez moduł routera zaimplementowany w sterowniku. Algorytm obsługi pakietów przychodzących do węzła przedstawiony jest na rysunku 9 [7].



Rys. 9. Algorytm odbierania pakietów

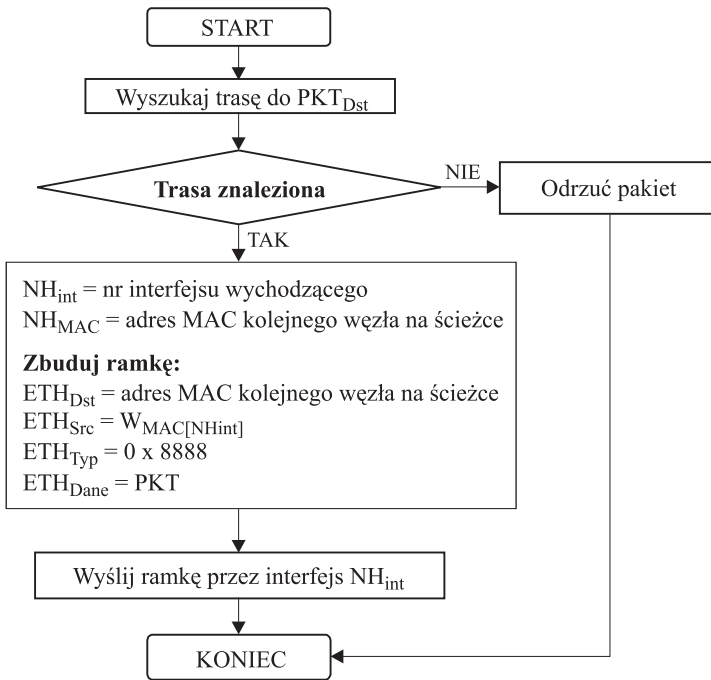
Użyte w zaprezentowanym schemacie oznaczenia należy interpretować w następujący sposób:

- $T$  — zbiór typów pakietów;
- $T_K$  ( $T_K \subset T$ ) — zbiór typów pakietów konfiguracyjnych;
- $T_D$  ( $T_D \subset T$ ) — zbiór typów pakietów diagnostycznych;
- $T_R$  ( $T_R \subset T$ ) — zbiór typów pakietów rozgłoszeniowych;
- $PKT$  — pakiet;
- $PKT_{Dst}$  = pole adresu docelowego pakietu;
- $PKT_{Src}$  — pole adresu źródłowego;
- $PKT_{Typ}$  — pole typ pakietu;
- $ETH_{Src}$  — pole adresu źródłowego ramki Ethernet;



- $ETH_{Dst}$  — pole adresu docelowego ramki Ethernet;
- $ETH_{Typ}$  — pole typ ramki Ethernet;
- $ETH_{Dane}$  — dane ramki Ethernet;
- $W_{Adr}$  — nr logiczny lokalnego węzła;
- $W_{MAC[i]}$  — nr MAC interfejsu lokalnego węzła, i nr interfejsu.

Opracowany sterownik operację wysyłania pakietów realizuje zgodnie z algorytmem przedstawionym na rysunku 10 [7].



Rys. 10. Algorytm wysyłania pakietów

Pakiet jest pobierany z kolejki pakietów do wysłania, z jednoczesną aktualizacją informacji wymaganych do jego przesłania (adresy MAC odbiorcy i nadawcy oraz typ ramki) i następnie jest kierowany do odpowiedniego urządzenia. Wybór urządzenia realizowany jest na podstawie tablicy węzłów sąsiednich oraz tablicy routowania. Tablica routowania zrealizowana jest jako wektor wskaźników o wymiarze  $2^n$ . Pojedynczy wskaźnik identyfikuje strukturę opisującą węzeł sąsiedni, do którego mają być kierowane pakiety. Tablica ta adresowana jest przez numer węzła docelowego, dzięki czemu wybór trasy wymaga pojedynczego odwołania do niej.

Omówiony proces przekazywania pakietów jest charakterystyczny dla transmisji typu punkt–punkt (pojedynczej emisji) stanowiącej jeden z podstawowych prymitywów komunikacyjnych.

Podsystem diagnostyczny został zaimplementowany w postaci osobnego wątku trybu jądra. Oczekuje on na pojawienie się pakietów w kolejce diagnostycznej. Pojawienie się nowego pakietu sygnalizowane jest zdarzeniem, co powoduje wzbudzenie czekającego wątku diagnostycznego. Odczytuje on pakiet diagnostyczny z kolejki i podejmuje czynności zgodnie z zaimplementowanym algorytmem diagnostycznym. Efektem końcowym działania algorytmu jest uaktualnienie macierzy przyległości oraz wektora opisującego stan węzłów sieci. W przypadku wystąpienia zmiany struktury sieci wywoływana jest procedura wyznaczania ścieżek do wszystkich węzłów sieci.

## Podsumowanie

Zaprezentowane rozwiązanie charakteryzuje się dużą uniwersalnością. Uniwersalność została zapewniona przez wykorzystanie bibliotek dynamicznych do implementacji algorytmu diagnozującego sieć. Zmiana algorytmu lub zmiana założeń jego użycia wymaga jedynie modyfikacji biblioteki. Ponadto wykorzystanie sterownika protokołu niezależnie opracowane rozwiązanie od medium transmisyjnego. Pozwala to na wykorzystanie różnych typów interfejsów sieciowych, między innymi połączeń kablowych (karty Ethernet, standard USB) oraz bezprzewodowych (Wi-Fi, Bluetooth).

Przedstawione właściwości opracowanego rozwiązania są szczególnie istotne z punktu widzenia systemów podlegających procesowi „łagodnej degradacji” [1]. W systemach tego typu zakłada się brak możliwości wymiany uszkodzonego elementu, a zadania, które on realizował, przejmują inne elementy systemu. Powoduje to równocześnie zmianę struktury systemu, która staje się podgrafem częściowym struktury  $H^n$  [3].

Obecnie prowadzone są prace nad implementacją przedstawionego rozwiązania w systemach wbudowanych (sieciach wieloprocesorowych), których elementy pracują pod kontrolą systemu Windows CE®.

Artykuł wpłynął do redakcji 14.12.2009 r. Zweryfikowaną wersję po recenzji otrzymano w grudniu 2009 r.

## LITERATURA

- [1] J. CHUDZIKIEWICZ, *Sieci komputerowe o strukturze logicznej typu hipersześcianu*, Instytut Automatyki i Robotyki, Wydział Cybernetyki WAT, Warszawa, 2002.
- [2] J. CHUDZIKIEWICZ, *Metoda wyznaczania m-optymalnych struktur opiniowania diagnostycznego dla sieci komputerowych typu hipersześcianu*, V Krajowa Konferencja Diagnostyka Urządzeń i Systemów, Ustroń, 2003.
- [3] J. CHUDZIKIEWICZ, K. MURAWSKI, *Wyznaczanie bezkolizyjnych dróg przesyłania danych w sieci teleinformatycznej o strukturze typu hipersześcianu*, Diagnostyka, 3, 39, 2006.

- [4] J. CHUDZIKIEWICZ, Z. ZIELIŃSKI, Wyznaczanie *m*-diagnostozowalnych struktur typu PMC w systemach o zwiększonej odporności na uszkodzenia, Materiały X Konferencji SCR' 2003, Ustroń, 15-18 września 2003.
- [5] Dokumentacja: Microsoft Windows 2003 Driver Development Kit, Microsoft, 2003.
- [6] ELIAS PROCÓPIO DUARTE JR., TAKASHI NANYA, A Hierarchical Adaptive Distributed System-Level Diagnosis Algorithm, IEEE Transaction On Computers, 47, 1, 1998.
- [7] A. GLINKA, Projekt i implementacja systemu diagnozowania sieci komputerowej o strukturze logicznej typu hipersześcianu, praca dyplomowa WAT, 2009.
- [8] S. L. HAKIMI, A. T. AMIN, Characterization of connection assignment of diagnosable systems, IEEE Transaction On Computers, C-23, Jan. 1974.
- [9] R. KULESZA, Podstawy diagnostyki sieci logicznych i komputerowych, Instytut Automatyki i Robotyki, Wydział Cybernetyki WAT, wyd. II, Warszawa, 2000.
- [10] R. KULESZA, Struktury samodiagnostozowalne w systemach cyfrowych, V Krajowa Konferencja Diagnostyka Urządzeń i Systemów, Ustroń, 2003.
- [11] Microsoft Developer Network Library for Visual Studio 2008, ENU.
- [12] W. ONEY, Programming the Microsoft Windows Driver Model, wyd. II, Microsoft Press, 2003.
- [13] T. ROBERTS, DLLs in Kernel Mode, Windows Driver Developer's Digest, 1, 3, 2003.
- [14] M. STAHL, R. BUSKENS, R. BIANCHINI, Simulation of the Adapt On-Line Diagnosis Algorithm for General Topology Networks, Proceedings of the 11th Symposium on Reliable Distributed Systems, 1992.
- [15] M. STAHL, R. BUSKENS, R. BIANCHINI, The Adapt2 On-Line Diagnosis Algorithm for General Topology Networks.
- [16] Z. ZIELIŃSKI, Model symulacyjny procedur diagnostycznych w rozproszonych systemach sieciowych, Diagnostyka, 3, 39, 2006.
- [17] S. VIPIN, R. DEEPA, Modeling of Computer Network Topologies Through UML, The Icfai University Journal of Systems Management, 6, 4, November 2008, 45-55.

J. CHUDZIKIEWICZ

### Implementation of diagnostic algorithm in cube networks

**Abstract.** In this paper, the author presents the method of implementation of algorithms for computer network diagnosis. This method takes advantage of a protocol driver. The protocol driver cooperates with DLL library implementing a diagnostic algorithm. Moreover, for Windows® operating systems, the tools and mechanisms which facilitate implementation of algorithms are presented. Besides, the manner of cube computer network modelling in UML is presented.

**Keywords:** hypercube, auto-diagnosis, faulty tolerance systems, diagnostic algorithms

**Universal Decimal Classification:** 681.324

