ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 34 Nr kol. 1381

Stefan SENCZYNA
Politechnika Śląska, Katedra Podstaw Systemów Technicznych

MODEL SYMULACYJNY SIECI KOMPUTEROWEJ W OPARCIU O STANDARD VHDL

Streszczenie. Analiza wydajności sieci komputerowej może być przeprowadzona metodą pomiarów realizacji sprzętowej i aplikacyjnej oraz metodą analizy modelu sieci. Znanym modelem sieci jest model kolejkowy dla analizy statystycznej. W artykule przedstawiono model zdarzeniowy realizowany za pomocą opisu funkcjonalnego i symulacji w oparciu o normę VHDL. Model przeznaczony jest do analizy ilościowej struktury sieci.

A SIMULATIVE MODEL OF COMPUTER NETWORK UNDER VHDL STANDARD

Summary. The analysis of computer network performance may be done by measurements of network hardware and application performance or by analityca models, e.g. queuening network models. This article presents the event-driven model based on functional description and simulation of VHDL standard. Presented model is a suitable tool to quantitative analysis of a network structure.

1. Zagadnienie budowy modelu sieci komputerowej

Sieć komputerowa składa się z takich elementów, jak serwery i stacje robocze oraz układy transmisyjne i arbitrażu, które łączą stacje i serwery ze sobą.

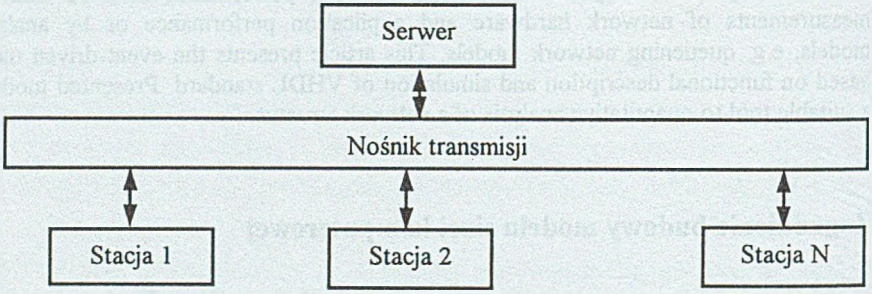
Model sieci komputerowej należy rozpatrywać w dwóch płaszczyznach: metod stosowanych w procesie projektowania oraz metod analizy właściwości pewnej struktury sieci. W procesie projektowania budowane są schematy strukturalne komponentów sieci oraz opracowywane protokoły współpracy serwerów i stacji z układem transmisyjnym. Celem procesu projektowego jest opracowanie szczegółowej struktury (logicznej) modułów oraz

oprogramowania realizującego określone protokoły. Stopień złożoności modeli stosowanych w procesie projektowania powoduje, że złożoność modelu sieci (składającego się z modeli projektowych komponentów) można określić jako znaczą.

Analizę właściwości sieci i komponentów można przeprowadzić budując modele typu moduł wejścia - wyjścia, opisany za pomocą zbioru relacji wejścia - wyjścia. Sieć komputerowa jest ogólnie strukturą połączonych procesorów, pamięci i interfejsów. W tym przypadku model strukturalny sieci jest budowany metodą odwzorowania poszczególnych komponentów za pomocą modułów wejścia - wyjścia. Formułując odpowiednio relacje wejścia - wyjścia modułów można przeprowadzić analizę realizacji podstawowego zadania sieci polegającego na udostępnianiu stacjom zasobów serwerów. Tego typu model jest budowany metodą zstępującą od określenia struktury sieci (zbioru i relacji między modułami) do specyfikacji wejścia - wyjścia modułów (relacji wejścia - wyjścia modułów).

W procesie projektowania układów logicznych stosowane są języki opisu sprzętu, tak samo można zastosować język opisu do budowy modelu sieci komputerowej. W tym przypadku analizę zachowań modelu można przeprowadzić za pomocą symulatora (zdarzeniowego) układów logicznych.

Symulacja zdarzeniowa pokazuje zachowania modelu w funkcji czasu w postaci wartości (sygnałów) wejść i wyjść komponentów. Zliczając określone zachowania modelu można przeprowadzić analizę ilościową modelu. Przyporządkowując relacjom modułów wartości „opóźnienia relacji" można zbudować model dynamiczny sieci pozwalający oszacować czas potrzebny na komunikację między stacjami i serwerami. Model sieci przedstawiający komunikację między stacjami i serwerem przedstawia rysunek 1.



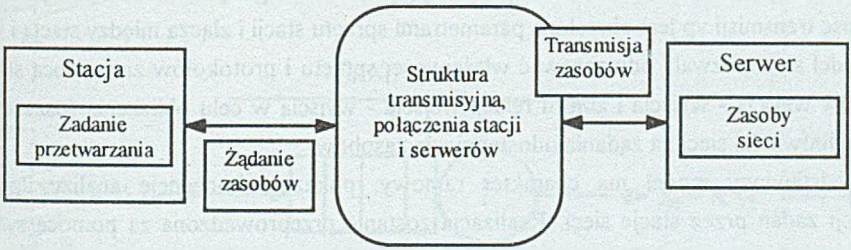
Rys. 1. Model sieci komputerowej
Fig. 1. A model of computer network

Interakcja między stacją i serwerem polega na wysłaniu przez stację - klienta sieci ciągu żądań do serwera aż do otrzymania żądanej ilości zasobów. Rzeczywista komunikacja jest określana protokołami sieciowymi, realizowanymi zarówno w warstwie sprzętu

i oprogramowania. Budowa modelu sieci wymaga odwzorowania funkcji protokołów za pomocą relacji wejścia - wyjścia. Przyjmując szereg uproszczeń można zbudować model o niskiej złożoności (mierzonej liczbą modułów i relacji modułów), który pozwoli przeprowadzić analizy ilościowe z zastosowaniem symulacji zdarzeniowej.

Zadanie analizy ilościowej modelu sieci można określić następująco:

- model sieci pokazuje interakcję między wybraną stacją i serwerem - rysunek 2,
- stacja do realizacji przetwarzania wymaga zasobu o określonej liczbie jednostek,
- protokół komunikacji zapewnia przesłanie żądania zasobu do serwera oraz jednostki zasobu do stacji,
- do żądań, jednostek zasobów oraz operacji protokołu są przyporządkowane czas ich realizacji,
- za pomocą modelu symulacyjnego sieci obliczane są liczbą żądań, jednostek zasobów oraz czas wykonania żądań dla danego zadania realizowanego przez stację,
- obliczenia za pomocą modelu mają na celu oszacowanie prędkość transmisji między stacją i serwerem dla szczególnych przypadków zadań realizowanych przez stację,
- należy założyć, że model jest przeznaczony do analizy przybliżonej zbioru przypadków szczególnych zadań realizowanych przez stację,
- model symulacyjny sieci jest strukturą złożoną, to znaczy że pewna liczba stacji i serwerów może wykonywać swoje zadania współbieżnie.



Rys. 2. Model interakcji stacja - serwer
Fig. 2. A model of workstation - server interaction

Dla określonego zadania realizowanego przez stacje można podać żądanie zasobów w postaci ciągu:

$$Z = \{ tp_1 + tz_1, \ldots tp_i + tz_i \ldots tp_N + tz_N \}$$

gdzie: tp - czas, po którym nastąpi kolejne żądanie,

tz - czas trwania żądania,

N - ilość żądań, taka że suma Tp czasów żądań zapewni transmisję zapotrzebowanego zasobu w liczbie Zw do przetwarzania,

$Tp = \sum tp_i$, dla $i = 1,...,N$

przy czym: $Zw < Tp * vp$,

Zw - zapotrzebowanie zasobu w jednostkach bezwzględnych (bajty), wymagana przez proces przetwarzania,

vp - prędkość transmisji złącza stacja - sieć (jednostki zasobu/jednostkę czasu).

Miarę efektywności transmisji dla danego procesu przetwarzania wymagającego zasobu Zw można podać następująco:

$a = Tw / Tn$,

gdzie: $Tw = Zw / vp$ czas transmisji zasobu w przypadku transmisji ciągłej,

$Tn = \sum tp_i + \sum tz_i$ dla $i = 1,...,N$ czas transmisji zasobu za pomocą ciągu żądań Z.

Stacja, która realizuje przetwarzanie, kieruje żądania do sieci i oczekuje transmisji potrzebnego zasobu Zw. W trakcie realizacji zadania generowana jest taka ilość N wyrazów ciągu Z, aż żądany zasób Zw zostanie dostarczony. Podział zasobu na elementy $tp_i + tz_i$ jest określony przez proces przetwarzania oraz protokół transmisji. Liczba N elementów ciągu Z jest uwarunkowana obciążeniem sieci powodującym, że szereg żądań może być odrzuconych. Prędkość transmisji vp jest określona parametrami sprzętu stacji i złącza między stacją i siecią.

Model sieci pozwala odwzorować właściwości sprzętu i protokołów za pomocą struktury modułów wejścia - wyjścia i zbioru relacji wejścia - wyjścia w celu obliczenia (oszacowania) czasu odpowiedzi sieci na żądania udostępnienia zasobów.

Przedstawiony model ma charakter ramowy, pokazuje koncepcję analizy ilościowej realizacji zadań przez stacje sieci. Realizacja zostanie przeprowadzona za pomocą symulacji zdarzeniowej. Ciąg żądań Z zostanie odwzorowany za pomocą wartości sygnałów logicznych, natomiast protokoły transmisji za pomocą operacji na sygnałach logicznych. Liczba elementów ciągu Z jest obliczana jako liczba cykli symulacji zdarzeniowej.

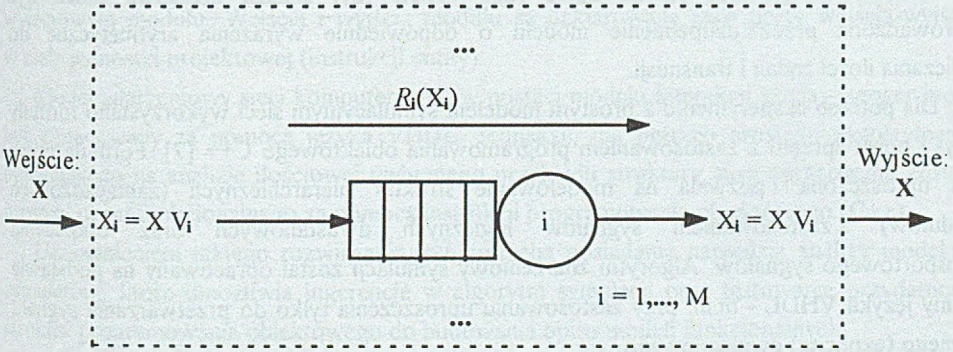
1.1. Elementy modelu kolejkowego systemu komputerowego

Do analizy wydajności systemów komputerowych stosowane są modele kolejkowe. W modelu kolejkowym analizowany jest czas (średni) obsługi zgłoszeń na stanowisku obsługi. Zgłoszenia napływające do stanowiska obsługi są ustawiane w kolejce (rysunek 3). Strumień wejściowy jako ciąg odcinków czasu między zgłoszeniami oraz stanowisko obsługi są modelowane za pomocą rozkładów prawdopodobieństwa.

Za pomocą modelu statystycznego [2], po wprowadzeniu formuł modelujących protokoły transmisji w sieci, można budować model statystyczny sieci komputerowej. Taki model jest zbiorem równań opisujących strukturę sieci, kolejki, stanowiska obsługi oraz przepływy zgłoszeń między stanowiskami i kolejkami. Następnie analiza danego modelu może być przeprowadzona za pomocą takich narzędzi, jak: QNAP, SMURPH, AMOK, AMOR [3]. Należy zauważyć, że modele statystyczne są również analizowane za pomocą metod symulacyjnych - ilościowych [3].

Wybranym przykładem elementu modelu statystycznego jest „sieć otwarta M stanowisk", (MVA) rysunek 3. Przepustowość X_i dowolnego stanowiska i, i = 1, ... M. jest określona natężeniem strumienia zewnętrznego X oraz liczbą wizyt V_i (współczynnik rozdziału strumienia X na stanowiska) składanych przez klientów w tym stanowisku, $X_i = XV_i$. Przepustowość X_i nie zależy natomiast od długości kolejki i czasu oczekiwania na danym stanowisku. Niech N_i oznacza średnią liczbę klientów w stanowisku i, $\underline{B}_i$ - średni czas obsługi, $\underline{R}_i$ - średni czas pobytu w stanowisku (łączny czas czekania i czas wykonania zadania). Nowy klient przychodząc do stanowiska i widzi przed sobą $\underline{N}_i$ klientów, z których każdy będzie obsługiwany przez czas $\underline{B}_i$, zanim zacznie się obsługa obserwowanego klienta. Czas odpowiedzi stanowiska, złożony z czasu czekania i czasu obsługi wynosi:

$\underline{R}_i = \underline{N}_i\underline{B}_i + \underline{B}_i = \underline{B}_i(\underline{N}_i + 1)$ [3].



Rys. 3. Sieć otwarta M stanowisk [3]
Fig. 3. An open network of M service stations [3]

2. Model zdarzeniowy sieci komputerowej

Przedstawiona metoda budowy modelu sieci komputerowej w oparciu o interakcję stacja - serwer (rysunek 2) może być realizowana za pomocą modelu kolejkowego. Natomiast

w oparciu o metody projektowania układów logicznych można zaproponować model zdarzeniowy sieci komputerowej. Taką metodą jest język opisu sprzętu, za pomocą którego budowany jest model układu logicznego w postaci zbioru operacji na sygnałach logicznych.

W definicji języka opisu sprzętu (w tym przypadku VHDL [5]) zdarzenie jest definiowane jako zmiana stanu sygnału logicznego, tą definicję można zaadoptować dla potrzeb modelu zdarzeniowego sieci komputerowej.

W modelu zdarzeniowym składniki struktury sieci są modułami wejścia - wyjścia. Połączenia między modułami są odpowiednio sygnałami logicznymi, natomiast żądania oraz zasoby są wartością sygnałów. W tym przypadku wystąpienie żądania oraz transmisja zasobów jest zdarzeniem. Model zdarzeniowy sieci komputerowej oparty jest na znanej metodzie formułowania opisu funkcjonalnego dla modułów układów logicznych. W prosty sposób żądanie oraz transmisję zasobu można modelować za pomocą wartości sygnału logicznego np.: „High", natomiast protokół transmisji za pomocą wyrażeń logicznych oraz modeli opóźnień sygnałów logicznych.

Model układu podany w języku opisu sprzętu uzupełniony o zbiór pobudzeń jest modelem symulacyjnym. W przypadku modelu sieci zbiór pobudzeń jest ciągiem żądań dla określonego zadania przetwarzania. Typowe symulatory modeli układów logicznych posiadają szereg narzędzi analizy, zwykle graficznej, wejść i wyjść modelu. W przypadku modelu sieci wykonanego za pomocą języka opisu (np. VHDL [5]) analiza ilościowa może być wprowadzona przez uzupełnienie modelu o odpowiednie wyrażenia arytmetyczne do obliczania ilości żądań i transmisji.

Dla potrzeb eksperymentu z prostym modelem symulacyjnym sieci wykorzystano formuły języka opisu sprzętu z zastosowaniem programowania obiektowego C++ [7]. Formuła opisu jest uproszczona i pozwala na modelowanie struktur hierarchicznych (zagnieżdżonych modułów) z zastosowaniem sygnałów logicznych dwustanowych oraz opóźnienia transportowego sygnałów. Algorytm zdarzeniowy symulacji został opracowany na podstawie normy języka VHDL - m.in. przy zastosowaniu uproszczenia tylko do przetwarzania sygnału jawnego (explicite) dwustanowego.

Opracowane narzędzie do budowy modelu zdarzeniowego sieci pozwala na przeprowadzenie analizy ilościowej interakcji stacja - serwer dla danej (prostej) struktury sieci.

2.1. Opis funkcjonalny w języku VHDL

Opis funkcjonalny jest narzędziem realizacji modeli typu zbiór relacji wejścia - wyjścia. Ma na celu badanie zachowania modułów układów logicznych bez konieczności projektowania ich architektury wewnętrznej.

W języku VHDL opis (projekt) układu logicznego jest formułowany w postaci struktury

hierarchicznej, zagnieżdżonych modułów. Poziomem najwyższym w hierarchii jest jednostka projektowa (entity), która stanowi moduł wejścia - wyjścia. Struktura jednostki (modułu) może być zbiorem alternatywnych architektur (architecture). Poszczególne architektury są opisem strukturalnym lub funkcjonalnym jednostki. Opis strukturalny jest zbiorem połączonych funktorów i modułów logicznych. Opis funkcjonalny jest zbiorem relacji wejścia - wyjścia, stanowiących operacje na sygnałach logicznych.

W języku VHDL opis architektur jest budowany za pomocą zbioru instrukcji współbieżnych (concurrent_statement). Natomiast podstawową instrukcją opisu funkcjonalnego jest instrukcja „process" [5], której uproszczona postać jest następująca:

etykieta:

process (*lista_czułości*) **is**

instrukcje_deklaracji

begin

instrukcje_modelu_we_wy

end process;

W instrukcji process *lista_czułości* jest zbiorem wejść modułu - sygnałów logicznych, które generują zdarzenia. Wystąpienie zdarzeń na wejściu powoduje wykonanie bloku instrukcji: „*instrukcje_modelu_we_wy*", które mogą przypisać nowe wartości sygnałom wyjściowym modułu. Wejścia i wyjścia modułu są deklarowane jako porty wejścia-wyjścia w ciele jednostki projektowej (instrukcji entity).

Model zdarzeniowy sieci komputerowej, w postaci modelu interakcji stacja - serwer może być opracowany za pomocą języka VHDL. Jednakże, dla potrzeb prostego eksperymentu polegającego na analizie ilościowej wybranego przykładu struktury sieci opracowane zostały formuły opisu funkcjonalnego za pomocą instrukcji programowania obiektowego (C++).

Uzasadnieniem takiego rozwiązania jest potrzeba posiadania narzędzia analizy modelu - symulatora, które umożliwia ingerencje w algorytm symulacji oraz testowanie przydatności struktur programowania obiektowego do budowania opisu modeli funkcjonalnych.

Jako wzorzec dla formuł opisu obiektowego jest postać instrukcji process (uruchamianie instrukcji: *instrukcje_modelu_we_wy* zależnie od zdarzeń na wejściu do modelu). Natomiast algorytm symulacji został opracowany (w wersji uproszczonej) na podstawie definicji języka VHDL.

2.2. Opis obiektowy modelu zdarzeniowego

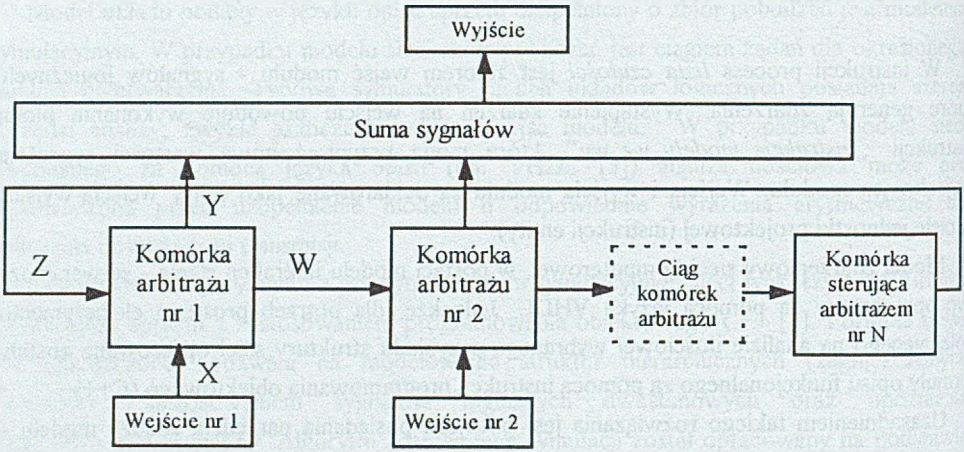
Opis obiektowy dla potrzeb eksperymentu jest realizowany wyłącznie za pomocą instrukcji i struktur danych języka C++. Takie rozwiązanie pozwala na zastosowanie dostępnych

kompilatorów C++ w celu generacji modelu symulacyjnego sieci komputerowej.

Na podstawie definicji języka VHDL opracowane zostały metody formułowania opisu modułów jako obiektów deklarowanych instrukcją class [7]. Metody określają, w jaki sposób należy budować opis hierarchiczny, opis modułów obejmujący wejścia - wyjścia, listę czułości oraz operacje na sygnałach wyjściowych.

Opis obiektowy jest budowany w postaci zbioru zmiennych obiektowych oraz określonej sekwencji wywołania metod obiektów, która stanowi główną pętlę symulacji. Dla uproszczenia opisu obiektowy zostanie przedstawiony dla wybranego przypadku struktury sieci - rysunek 1, z zastosowaniem metody opisu funkcjonalnego.

Przedstawiony model sieci jest złożony z modułu (jednego) serwera, modułów stacji (N modułów) oraz modułu nośnika transmisji. Moduły serwera i stacji realizują funkcje udostępnienia i żądania zasobów - dla uproszczenia, w modelu przesyłane będą tylko żądania od stacji do serwerów. Moduł nośnika transmisji modeluje połączenia stacji do serwera oraz (uproszczony) protokół przekazywania żądań stacji do serwera.



Rys. 4. Schemat strukturalny nośnika transmisji
Fig. 4. A structure scheme of transmisson carrier

Funkcje oraz wejścia - wyjścia poszczególnych modułów są następujące:

serwer: jedno wejście dla sygnałów żądań,
nie realizuje funkcji,

stacja: jedno wyjście sygnału żądań, funkcja generacji skończonego ciągu żądań,
funkcja generacji skończonego ciągu żądań,

nośnik: wejścia sygnałów żądań, arbitrażu, wyjścia sygnałów żądań, arbitrażu,
funkcje transmisji żądań, funkcje arbitrażu żądań.

Model sieci przedstawiony na rysunku 1 jest podany za pomocą schematu złożonego z symboli modułów i połączeń. Ta postać modelu jest nazywana modelem strukturalnym (sprzętu). Wejścia i wyjścia poszczególnych modułów są sygnałami logicznymi. Moduły mogą być dalej rozwijane metodą modelu strukturalnego aż do struktury złożonej z połączonych funktorów logicznych odpowiadających jednostkom sprzętowym oraz metodą opisu funkcjonalnego za pomocą zbioru relacji wejścia - wyjścia (funkcji logicznych).

Do budowy opisu modelu sieci (z rysunku 1) zostały zastosowane modele strukturalne i funkcjonalne. Rozwinięcie strukturalne modułu nośnika podaje rysunek 4. Moduł serwera nie wymaga dalszego rozwinięcia. Moduły stacji i komórki nośnika zostaną podane za pomocą modelu funkcjonalnego, który obejmuje również zależności dynamiczne między sygnałami - opóźnienia.

2.3. Opis funkcjonalny modułów - model komórki arbitrażu

Podstawowym składnikiem modelu nośnika transmisji jest komórka arbitrażu. Model funkcjonalny komórki arbitrażu nośnika można przedstawić następująco:

zbiór wartości sygnałów: $W = \{0, 1\}$

zbiór wartości zdarzeń: $Z = \{0 \rightarrow 1, 1 \rightarrow 0\}$

zbiór wartości stanów sygnału: $Sk = \{0 \rightarrow 1, 1 \rightarrow 0\} \cup \{0, 1\}$

sygnały wejścia: X - sygnał żądania
Z - sygnał arbitrażu

sygnały wyjścia: Y - sygnał żądania
W - sygnał arbitrażu

zbiór relacji wejścia - wyjścia: R = { jeżeli $Z = 0 \rightarrow 1$ i $X = 1$ to $Y = X$,
jeżeli $X = 0$ to $Y = 0$,
jeżeli $Z = 0 \rightarrow 1$ i $X = 0$ to $W = 1$,
jeżeli $Y = 0 \rightarrow 1$ to $W = 1$,
jeżeli $W = 0 \rightarrow 1$ to po czasie t1 $W = 0$,
pozostałe przypadki nie zmieniają wartości wyjść,
}

Dla modułu, jakim jest komórka arbitrażu, opracowano opis funkcjonalny za pomocą deklaracji *class* języka C++. Wyjścia modułu, sygnały logiczne są zmiennymi obiektowymi Y, W typu *signal*, gdzie deklaracja typu obiektowego *signal* została opracowana na podstawie (w uproszczeniu) definicji „driver" oraz definicji cyklu symulacji normy VHDL. Wyjścia modelu są definiowane za pomocą parametrów formalnych typu *signal* niektórych metod obiektu. Metody obiektu, stanowiącego opis modułu komórki arbitrażu, zapewniają realizację następujących zadań:

- ustawienie wartości początkowych obiektu,
- przetwarzanie obiektu w cyklu symulacji,
- odwzorowanie relacji wejścia-wyjścia modelu, analogicznie do struktury instrukcji „*process*" języka VHDL.

Należy zaznaczyć, że struktura deklaracji obiektu stanowiącego opis modułu została opracowana na podstawie normy języka VHDL, jednakże nie stanowi jednoznacznego odpowiednika instrukcji języka VHDL (w szczególności instrukcji *process*). To znaczy, że wykorzystano niektóre, główne założenia normy, odwzorowane w strukturze deklaracji obiektu.

Struktura hierarchiczna (zagnieżdżone moduły) modelu sieci zostanie odwzorowana za pomocą zagnieżdżonych deklaracji zmiennych obiektowych. Wszystkim modułom występującym na poziomach od najwyższego do najniższego rozwinięcia hierarchicznego modelu muszą być przyporządkowane zmienne obiektowe. Na najwyższym poziomie hierarchii odwzorowuje zmienna obiektowa typu „model sieci", na najniższym poziomie występuje zbiór obiektów typu *signal*, które odwzorowują sygnały logiczne.

Model komórki arbitrażu jest dwupoziomowy - sygnały logiczne są deklarowane jako zmienne obiektowe „wewnątrz" deklaracji obiektu komórki arbitrażu. Deklaracja obiektu dla tego modelu jako szczególne rozwiązanie stanowi wzorzec dla realizacji opisu obiektowego (funkcjonalnego) dla poszczególnych modułów modelu sieci komputerowej. W oparciu o deklarację - wzorzec zostały opracowywane deklaracje obiektów dla pozostałych modułów przykładowej struktury sieci - rysunek 1.

```
// deklaracja obiektu - modelu komórki arbitrażu
class cell01 {
 private:
    unsigned int f1;                    // parametry opóźnienia

 public:
    int Y1 ();                          // wyjścia "pomiarowe"
    int Y2 ();                          // wyjścia "pomiarowe"
    signal Y, W;
    int input1 (signal In1)             // deklaracja "formalna" portu typu input, "value"
        { return In1.out_val; };
    int in_event1 (signal In1)          // deklaracja "formalna" portu typu input, "event"
        { return In1.event; };
    cell01 ();
    cell01 (unsigned int ff1);          // par. stałe opóźnienia
```

```
    unsigned long mtime ();              // wyszukanie minimalnego czasu transakcji sygnałów
    void analyze_s ();                   // analiza transakcji bieżących sygnałów
    void update_s ();                    // aktualizacja sygnałów i statusów zdarzeń
    void delete_s ();                    // kasowanie transakcji aktywnych sygnałów
    void block (signal Z, signal X);     // obliczenia relacji modelu
    void cykl_0 (signal Z, signal X);    // cykl nr 0, obliczenia relacji modelu
} // cell011

cell01::cell01 ()   { f1 = 20; /* konstruktor */ }
cell01::cell01 (unsigned int ff1)  { f1 = ff1; /* konstruktor */ }

unsigned long cell01::mtime ()  // wyszukanie wartości minimalnej w buforach transakcji
 { unsigned long tt;
   tt = Y.tval_trz ();
   if (tt > W.tval_trz ()) tt = W.tval_trz (); return tt;    } // mtime

void cell01::analyze_s ()  // analiza zdarzeń na sygnałach - ustawienie statusu „sygnał aktywny"
 { Y.acv_trz ();      W.acv_trz ();  }

void cell01::update_s ()  // aktualizacja sygnałów aktywnych oraz statusów zdarzeń
 { Y.out_trz ();  W.out_trz ();  }

void cell01::delete_s ()  // kasowanie transakcji "aktywnych"
 { Y.del_trz ();  W.del_trz ();  }

void cell01::block (signal Z, signal X)  // relacje wejścia - wyjścia modułu
 { if (in_event1 (Z))
      { if ((input1 (Z) = = High) && (input1 (X) = = High)) Y.in_trz (input1 (X),0);
        if ((input1 (Z) = = High) && (input1 (X) = = Low)) W.in_trz (High, 0);
      }
   if (in_event1 (X))  if  (input1 (X) = = Low) Y.in_trz (Low,0);
   if ((Y.event) && ( Y.out_val = = Low )) W.in_trz (High, 0);
   if ((W.event) && (W.out_val = = High))  W.in_trz (Low, f1);
 } // block

void cell01::cykl_0 (signal Z, signal X)  // instrukcje  wykonywane tylko w cyklu "zerowym"
 { Y.in_trz (input1 (X),0);  W.in_trz (Low, 0);
```

```
Y.in_trz (Low,0); W.in_trz (High, 0);   }
// wyjścia "pomiarowe"
int cell01::Y1 ()  { return Y.out_val; };
int cell01::W1 ()  { return W.out_val; };
```

3. Analiza ilościowa modelu symulacyjnego sieci komputerowej

Przedstawiony model zdarzeniowy został opracowany w oparciu o metodę opisu funkcjonalnego układu logicznego. W procesie projektowania opis funkcjonalny służy do analizy zachowań poszczególnych modułów układu logicznego przez porównanie z założonymi wzorcami pobudzeń wejściowych i wyjściowych.

Model zdarzeniowy sieci jest opisem funkcjonalnym wybranego przypadku struktury sieci oraz zbioru protokołów komunikacji stacja - serwer. Zakładając, że wejściami do modelu są żądania zasobów generowane przez stacje (aplikacje stacji), tak jak na rysunku 2, to ciąg żądań należy odwzorować za pomocą pewnego ciągu zdarzeń. Model symulacyjny jest modelem zdarzeniowym i zbiorem ciągów zdarzeń podawanych na jego wejścia.

Analiza ilościowa modelu symulacyjnego obejmuje obliczanie zdarzeń na wyjściach modelu dla określonych pobudzeń wejściowych. Obraz zachowań modelu sieci jest zbiorem wskaźników ilościowych dla danego zbioru ciągów zdarzeń wejściowych.

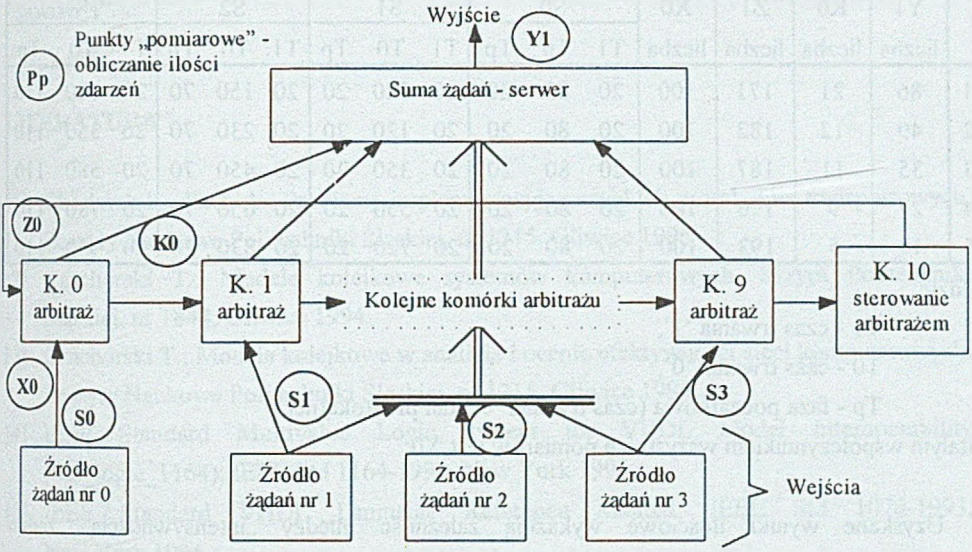
Realizacja analizy ilościowej modelu zostanie przedstawiona dla danego przypadku modelu sieci - rysunek 1, dla którego został opracowany model funkcjonalny (model komórki arbitrażu) i symulacja dla zadanych ciągów pobudzeń wejściowych.

Model symulacyjny danego przypadku sieci jest złożony z następujących obiektów:

Obiekt	Funkcje obiektu
stacja	źródło żądań
serwer	suma wyjść z komórek arbitrażu
nośnik transmisji	arbitraż żądań stacji

Przedstawiony model, w znacznym uproszczeniu, odwzorowuje podstawowe zadanie sieci stanowiące przydział zasobów serwera stacjom na podstawie arbitrażu żądań stacji. W szczególności model nośnika transmisji nie stanowi odwzorowania rzeczywistych protokołów stosowanych w sieciach komputerowych. Model pokazuje metodę - opis funkcjonalny oraz narzędzie - programowanie obiektowe (C++, VHDL), za pomocą których można zbudować model zdarzeniowy sieci (dla wybranego przypadku) w celu analizy ilościowej. Analiza ilościowa danego przypadku sieci zostanie przeprowadzona w warunkach, gdy jedna stacja wysyła stałą ilość żądań, pozostałe „zakłócają" arbitraż ciągami o stopniowanej ilości żądań.

Żądania są modelowane za pomocą fali prostokątnej o określonym współczynniku wypełnienia. Analiza ilościowa obejmuje obliczanie liczby żądań i zgłoszeń w ustalonym odcinku czasu. Obliczenia obejmują zgłoszenia, które pojawiły się na wejściu serwera oraz ilość żądań wystawionych przez stacje i zaakceptowanych przez arbitraż. Strukturę modelu dla danego przypadku sieci przestawia rysunek 5.



Rys. 5. Struktura analizowanego modelu
Fig. 5. A structure of analyzed model

3.1. Pobudzenia wejściowe i odpowiedzi wyjść

Pobudzenia wejściowe są generowane przez źródła nr 0 do nr 3 (rysunek 5) i podawane na wejścia komórek arbitrażu nr 0 do nr 9. Źródło nr 0 i komórka nr 0 są wyróżnionymi elementami dla potrzeb analizy ilościowej. Format żądań został uproszczony do fali prostokątnej. Czas trwania żądań jest jednakowy dla wszystkich źródeł.

Analiza wyjść obejmuje obliczanie ilości zdarzeń na wyjściach Y1, K0 oraz dodatkowo ilość zdarzeń na wejściu Z0 komórki nr 0 dla odcinka czasu określonego przez 100 żądań źródła nr 0 (S0). Wejście Z komórek arbitrażu jest „zezwoleniem" (żetonem) na transmisję żądania z wejścia X na wyjście Y, gdy wejście X otrzymało żądanie. Po pewnym, określonym czasie zezwolenie jest przekazywane do sąsiedniej komórki (rysunek 4, oraz model funkcjonalny komórki arbitrażu).

Przyjmuje się, że skala czasu analizowanego modelu jest bezwymiarowa (nie określa się współczynnika skali).

3.2. Wyniki analizy ilościowej modelu

Tabela 1

Wyniki analizy ilościowej dla modelu podanego na rysunku 5

Nr	Wyj. Arbitr.	Wyj. Kom.	Wejścia komórki nr 0					Wejścia komórek nr 1 do 9								
	Y1	K0	Z1	X0	S0			S1			S2			S3		
	liczba	liczba	liczba	liczba	T1	T0	Tp	T1	T0	Tp	T1	T0	Tp	T1	T0	Tp
1	86	21	171	100	20	80	20	20	120	20	20	150	70	20	180	110
2	49	12	182	100	20	80	20	20	120	20	20	230	70	20	350	110
3	35	11	187	100	20	80	20	20	350	20	20	450	70	20	580	110
4	27	9	190	100	20	80	20	20	550	20	20	650	70	20	780	110
5	15	5	193	100	20	80	20	20	720	20	20	830	70	20	950	110

Gdzie:

T1 - czas trwania '1',

T0 - czas trwania '0',

Tp - faza początkowa (czas trwania '0') fali prostokątnej.

Stałym współczynnikiem wszystkich pomiarów jest X0.

Uzyskane wyniki ilościowe wykazują zależność między „intensywnością" żądań generowanych przez źródła S1, S2, S3 a pracą komórki nr 0. Liczba transmisji (K0) wykonanych przez komórkę nr 0 jest mniejsza, im więcej żądań (w jednostce czasu) jest generowanych przez źródła. Ta zależność jest specyficzna dla analizowanego modelu, pokazuje ewentualne efekty, jakie można uzyskać za pomocą analizy ilościowej modelu zdarzeniowego sieci.

4. Wnioski

We wnioskach należy zaznaczyć, że przedstawiony model zdarzeniowy nie stanowi alternatywy dla modelu statystycznego. Model zdarzeniowy jest narzędziem analizy ilościowej określonych przypadków struktur sieci i protokołów transmisji, a uzyskane wyniki mogą być wskazaniem do badań statystycznych. Możliwe jest również generowanie ciągów żądań na podstawie rozkładów statystycznych, a wyniki przetwarzać analizą statystyczną. W takim przypadku model zdarzeniowy byłby szczególnym przypadkiem modelu statystycznego.

Przedstawiona metoda analizy modelu zdarzeniowego, obejmująca opis funkcjonalny, symulację i analizę ilościową, jest narzędziem analizy zachowań modelu w warunkach

określonych ilościowo pobudzeń. Metoda ilościowa może mieć zastosowanie do badania wydajności sieci w warunkach realizacji przez stacje określonego typu aplikacji - generujących „regularne" ciągi żądań zasobów. Wynikiem analizy będą współczynniki ilościowe charakteryzujące wydajność połączenia stacja - serwer.

Dalsze badania nad metodą analizy ilościowej będą prowadzone w kierunku opracowania modeli funkcjonalnych podstawowych protokołów sieci oraz struktur sieci typu „magistrala" i „gwiazda".

LITERATURA

[1] Chodorek R., Pach A.: Problemy implementacyjne szybkich protokołów transportowych. Zeszyty Naukowe Politechniki Śląskiej, nr 1315, Gliwice 1996.

[2] Czachórski T.: Modele kolejkowe systemów komputerowych. Skrypt Politechniki Śląskiej, nr 1844, Gliwice 1994.

[3] Czachórski T.: Modele kolejkowe w analizie i ocenie efektywności sieci komputerowych. Zeszyty Naukowe Politechniki Śląskiej, nr 1315, Gliwice 1996.

[4] IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164); IEEE Std 1164-1993, New York 1993.

[5] IEEE Standard VHDL Language Reference Manual; IEEE Std 1076-1993, New York 1994.

[6] Kacprzak M.: Metoda oceny charakterystyk czasowych adapterów komunikacyjnych dla sieci komputerowych. Zeszyty Naukowe Politechniki Śląskiej, nr 1222, Gliwice 1993.

[7] Lippman S. B.: Podstawy języka C++ . WNT, Warszawa 1994.

[8] Tannenbaum A. S.: Sieci komputerowe. WNT, Warszawa 1988.

[9] Tyszer J.: Symulacja cyfrowa. WNT, Warszawa 1990.

[10] Walran J.: An Introduction to Queueing Networks. Prentice Hall, Englewood Cliffs, New Jersey 1988.

Recenzent: Dr hab. inż. Tadeusz Czachórski, Prof. Politechniki Śląskiej

Wpłynęło do Redakcji 23 grudnia 1997 r.

Abstract

The computer network is a set of hardware components and sofware of systems and application. The evaluation of network performance may be done by analysis some hardware - measuring performance e. g. transfer rate or using network analityca model for estimating performance. A event-driven model of network is presented in this article.

A model is built for simply structure of network - fig. 1. The task of resource requirements is considered using client-server model - fig. 2.

A description hardware language (tool for IC design) is suitable for event-driven model and VHDL is main standard in this domain. For modeling the was choosing hierarchical and functional description. Fig. 4 presents a network structure for witch the description is built using object oriented programming (C++). A method of object description is work out under VHDL standard (subset). This gives simulative model of computer network (e.g. fig. 5). The results of quantitative analysis is presented in table 1.

Modeling hardware and protocols of computer network using event-driven model and object programing we have posibility to evaluate the client-server resource transfer rate.