

Piotr KASPRZYK  
Politechnika Śląska, Instytut Informatyki

## MODEL SYSTEMU TRANSMISJI DANYCH O WIELKIEJ OBJĘTOŚCI

**Streszczenie.** Obecnie coraz popularniejsze jest przesyłanie multimedialnych plików o dużej objętości przez szybką sieć lokalną wysokiej jakości. Uzyskiwane prędkości transmisji danych pozwalały na wysunięcie przypuszczenia, że możliwe jest zwiększenie tej prędkości i dlatego opracowano model systemu służącego do transmisji danych. Implementacja aplikacji badawczej zgodnie z przedstawionym modelem pozwoli na przeprowadzenie serii eksperymentów, które powinny pomóc w znalezieniu przyczyny zmniejszonej efektywności transmisji.

## THE MODEL OF SYSTEM FOR LARGE DATA TRANSMISSION

**Summary.** Large multimedia files are nowadays transmitted using high quality fast LANs. A certain inefficiency in this process was observed thus a model of such transmission system was presented. Implementation of this model will allow for running several experiments aimed at locating and fixing the source of the inefficiency.

### 1. Wprowadzenie

Obecnie można zaobserwować rosnące zainteresowanie transmisją plików multimedialnych w sieciach komputerowych. Pliki takie charakteryzują się dużą objętością (co najmniej setki megabajtów), a sam proces transmisji polega na pobraniu treści pliku z twardego dysku jednego komputera, przesłaniu jej przez sieć komputerową do drugiego komputera i zapisaniu na jego twardym dysku. Dla przesłania plików o dużej objętości korzystne jest zastosowanie szybkiej sieci lokalnej, dlatego skupiono się głównie na powszechnie stosowanej sieci Fast Ethernet [2].

Można wyliczyć maksymalną, teoretycznie możliwą do uzyskania, prędkość transmisji przy zastosowaniu protokołu TCP w sieci Fast Ethernet. Na początku ramki znajduje się preambuła o długości 8 bajtów, po niej jest 14 bajtów nagłówka sieci Ethernet, 1500 bajtów transmitowanych danych, 4 bajty sumy kontrolnej FCS oraz co najmniej 12 bajtów (96 bitów) odstępu między kolejnymi ramkami [3]. Na podstawie tych wartości można założyć, że wysłanie ramki o największym dozwolonym rozmiarze zajmie co najmniej tyle czasu, ile trwa wysłanie 1538 bajtów danych. Wykorzystując do transmisji danych protokół TCP z pola danych musimy przeznaczyć po 20 bajtów na nagłówki IP i TCP, co pozostawia 1460 bajtów rzeczywiście przekazywanych danych. Z powyższych wartości można wyznaczyć maksymalną prędkość transmisji sieci Fast Ethernet:  $100000000 \text{ bitów na sekundę} * 1460 / 1538$ , co daje około 11866059 bajtów na sekundę.

Sieć Fast Ethernet w rzeczywistych warunkach umożliwi uzyskanie prędkości transmisji danych na poziomie 10 megabajtów na sekundę, a współcześnie produkowane dyski twarde pozwalają na uzyskanie prędkości zapisu i odczytu danych około 20 megabajtów na sekundę. Przy takich parametrach pracy elementów systemu można zaobserwować, że prędkość uzyskiwana podczas transmisji dużych plików między dwoma komputerami jest mniejsza od prędkości transmisji danych w sieci. Taki efekt sugeruje, że możliwe byłoby udoskonalenie pracy systemu, aby zwiększyć efektywność transmisji dużych plików między dwoma komputerami.

W celu łatwiejszej analizy pracy systemu postanowiono opracować model aplikacji badawczej. Aplikacja badawcza powinna być na tyle uniwersalna, aby móc pracować zarówno na komputerze nadającym, jak i na komputerze odbierającym dane. Użytkownik aplikacji badawczej powinien mieć możliwość obserwacji zmian prędkości transmisji danych, aby określić, czy uzyskiwana prędkość podlegała znaczącym wahaniom.

Podstawowym elementem konstrukcji powszechnie stosowanych komputerów klasy IBM PC (inaczej jest w niektórych serwerach, gdzie technologia I<sub>2</sub>O umożliwia wymianę danych bezpośrednio między urządzeniami wejścia/wyjścia [4]) jest przysłanie informacji z wykorzystaniem bufora umieszczonego w pamięci RAM. Dane pobierane z sieci lub z dysku twardego są umieszczane w takim buforze, a następnie system operacyjny komputera przekazuje te dane właściwemu procesowi lub sam je wykorzystuje. Ze względu na taki sposób pracy komputerów klasy PC centralnym elementem aplikacji badawczej będzie bufor umieszczony w pamięci RAM komputera. Aplikacja badawcza powinna umożliwiać zmianę rodzaju i parametrów pracy takiego bufora. Użytkownik aplikacji będzie mógł podać, z jakiego źródła będą pobierane do bufora dane: z programowego generatora danych, z pliku lub ze strumienia danych otrzymywanych przez sieć komputerową. Analogicznie użytkownik powinien mieć możliwość wysłania danych pobranych z bufora do pliku, do innego

komputera przez sieć komputerową lub do procedury analizującej uzyskiwaną prędkość transmisji danych. Założono tutaj pewną niesymetryczność w pracy aplikacji: dane muszą być pobierane w pewnym eksperymencie dokładnie z jednego źródła, a liczba strumieni wyjściowych jest dowolną liczbą naturalną, dodatnią. Taka niesymetryczność pozwala na dokonanie wielu różnych operacji na strumieniu tych samych danych, np. jednoczesne zapisywanie danych w pliku oraz obserwowanie prędkości zapisywania danych.

## 2. Model aplikacji badawczej

Poniżej przedstawiono szkic modelu takiej aplikacji badawczej, zapisany w MBNF [1]:

łańcuch = wejście magazynowanie.

magazynowanie = magazyn reszta.

reszta = ( wyjście | broker magazynowanie | półbroker reszta | rozdzielacz reszta reszta ).

Model aplikacji badawczej zakłada, że przetwarzane dane będą przesyłane przez łańcuch połączonych procesów. Poszczególne procesy wchodzące w skład łańcucha będziemy nazywać ogniwami. Taki łańcuch może mieć odgałęzienia, co pozwala na dostarczenie tej samej porcji informacji do dwóch różnych odbiorców.

Określono następujące klasy ogniw: magazyn, wejście, wyjście, broker, półbroker i rozdzielacz. Graficzną reprezentację tych klas przedstawiono na rys. 1.

### 2.1. Magazyn

Termin "magazyn" reprezentuje ogniwo zawierające podstawowy element aplikacji badawczej, jakim jest bufor w pamięci RAM (rys. 1a). Działanie magazynu polega na oferowaniu poprzedzającemu go ogniwu miejsca, które przez to poprzedzające ogniwo zostaje zapelnione przesyłanymi danymi. Dane umieszczone w buforze są następnie oferowane następnemu ogniwu w łańcuchu. Praca magazynu kończy się w momencie, gdy poprzedzające ogniwo zasygnalizowało koniec strumienia danych, a wszystkie dotychczas otrzymane dane zostały przekazane dalej.

### 2.2. Wejście

Termin "wejście" oznacza ogniwo, które wprowadza dane do łańcucha (rys. 1b). Zaraz za ogniwnem "wejście" musi znajdować się magazyn, który zaoferuje miejsce w buforze RAM. W zależności od wskazanego typu ogniwa "wejście" dane mogą być wczytywane z pliku o danej nazwie, odbierane ze strumienia sieciowego bądź uzyskiwane z generatora danych -

prostego procesu, który ma za zadanie wypełnić podaną liczbę bajtów bufora. Po dotarciu do końca danych (koniec pliku, strumień sieciowy został zamknięty, generator już wyemitował żądaną liczbę danych) na ofertę miejsca w buforze magazynu ogniwo "wejście" zwraca informację o braku dalszych danych i kończy działanie.

### 2.3. Wyjście

Termin "wyjście" określa ogniwo znajdujące się na końcu łańcucha (rys. 1c). W zależności od wskazanego typu ogniwa "wyjście" dane mogą być zapisywane do pliku o podanej nazwie, wysyłane przez strumień sieciowy do wskazanego komputera lub dane mogą być bezpośrednio analizowane w celu określenia chwilowej oraz ogólnej prędkości transmisji. Przed ogniwnem typu "wyjście" powinien znajdować się magazyn bądź omówione poniżej ogniwa pomocnicze "półbroker" i "rozdzielacz", pełniące funkcje magazynu dla ogniwi następujących po nich. Działanie ogniwa typu "wyjście" kończy się w momencie, gdy wcześniejszy magazyn poinformuje o końcu strumienia danych.

### 2.4. Broker

Termin "broker" określa ogniwo umieszczone między dwoma magazynami (rys. 1d). Broker nie dysponuje własnym buforem w pamięci RAM, a jego zadaniem w najprostszym przypadku jest przenoszenie danych z poprzedzającego go magazynu do magazynu następującego po nim. Broker oczekuje na ofertę danych od wcześniejszego magazynu oraz na ofertę miejsca od późniejszego magazynu w łańcuchu. Gry broker uzyska obie te rzeczy, dokonuje skopiowania danych z bufora wcześniejszego magazynu do późniejszego magazynu. Broker jest właściwym miejscem do zaimplementowania operacji kompresji i dekompresji przesyłanych danych.

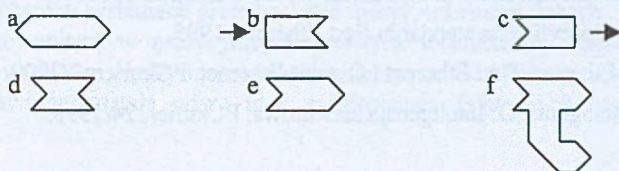
### 2.5. Półbroker

Termin "półbroker" określa ogniwo, które potrafi pobierać dane od poprzedzającego je magazynu (lub zachowującego się podobnie ogniwa typu "półbroker" lub opisany poniżej "rozdzielacz"), a po przetworzeniu tych danych oferuje rezultat przetwarzania następującemu po nim ogniwu typu "wyjście", "broker" lub "półbroker" (rys. 1e). Półbroker dysponuje zwykle własnym, niewielkim buforem w pamięci RAM. Funkcją realizowaną przez półbrokera może być np. liczenie sumy kontrolnej MD5, wykonywane w celu sprawdzenia poprawności przesyłanych danych. Dane pobierane z magazynu przez takie ogniwo służą do aktualizacji wartości sumy kontrolnej MD5, a po uzyskaniu informacji o zakończeniu

strumienia danych wyliczona wartość sumy kontrolnej jest przekazywana do następnego ogniwa w łańcuchu.

## 2.6. Rozdzielacz

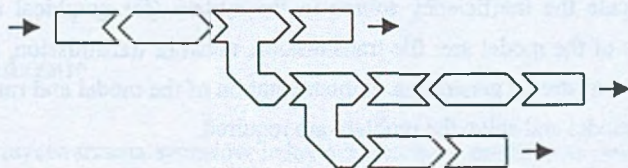
Termin "rozdzielacz" określa specyficzny rodzaj półbrokera, do którego można dołączyć równolegle dokładnie dwa ogniwa typu "wyjście", "broker" lub "rozdzielacz" (rys. 1f). Podobnie jak dla półbrokera elementem poprzedzającym rozdzielacz powinien być magazyn, półbroker lub inny rozdzielacz. Działanie rozdzielacza polega na przekazaniu porcji danych zaferowanej przez poprzedzający magazyn pierwszemu ogniwu następującemu po rozdzielaczu, a następnie ta sama porcja danych jest przekazywana drugiemu ogniwu następującemu po rozdzielaczu. Takie działanie rozdzielacza umożliwia przetwarzanie na wiele sposobów pojedynczego strumienia danych. Liczba ogniw typu "wyjście" w łańcuchu musi być dokładnie o jeden większa od liczby rozdzielaczy, aby łańcuch był kompletny.



Rys. 1. Klasy ogniw: a – magazyn, b – wejście, c – wyjście, d – broker, e – półbroker, f – rozdzielacz

Fig. 1. Classes of links: a – store, b – input, c – output, d – broker, e – half-broker, f – splitter

Figury symbolizujące klasy ogniw zostały zbudowane na bazie prostokąta zorientowanego poziomo. Każda figura symbolizuje ogniwo, które ma za zadanie przesyłanie informacji pobieranej z lewej strony i przekazywanie tej informacji w prawą stronę. Oba boki prostokąta, lewy i prawy, mogą być zmodyfikowane przez utworzenie wypukłości (ogniwo pełni funkcję magazynu) lub przez utworzenie wklęsłości (ogniwo pełni funkcję brokera). Brak modyfikacji z lewej strony oznacza, że jest to (jedynne) miejsce wprowadzania danych do systemu. Brak modyfikacji z prawej strony oznacza, że jest to (jedno z wielu możliwych) miejsce wyprowadzania danych z systemu. Na rys. 2 pokazano system z 3 wyjściami.



Rys. 2. Przykład systemu z 3 strumieniami wyjściowymi

Fig. 2. Example of system with 3 output streams

### 3. Podsumowanie

Przedstawiony powyżej model aplikacji badawczej jest na tyle modułowy, że potrafi reprezentować zarówno stację nadawczą, jak i odbiorczą. Konstrukcja modelu umożliwia przeprowadzenie samego odczytu z dysku, samego transferu sieciowego lub samego zapisu na dysk. Implementacja aplikacji badawczej według tego modelu pozwoli zweryfikować poprawność modelu, a także przeprowadzić szereg badań, mających na celu znalezienie przyczyny zmniejszonej efektywności systemu podczas transmisji dużych plików multimedialnych.

### LITERATURA

1. Iglewski M., Madey J., Matwin S.: Pascal. WNT, Warszawa 1986.
2. IEEE 802.3u, Specyfikacja standardu Fast Ethernet, 1995.
3. Nowacki K.: Ethernet, Fast Ethernet i Gigabit Ethernet. PCkurier, 2/2000.
4. Skwara J.: Intelligent I/O: Inteligencja serwerowa. PCkurier, 24/1998.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 25 marca 2002 r.

### Abstract

Up to date popular hard disks can maintain transfer speed at level of 20 MB per second. Network transfer speed of Fast Ethernet can be estimated at 10 MB per second. The process of transmitting large files between 2 computers using fast LAN is not as efficient as the lower of two presented above numbers. A model of data transmission system (in MBNF) was proposed to locate the inefficiency source in the system (for graphical representation see fig. 1). Features of the model are: file transmission, network transmission, measuring of data transfer speed, data stream generation. Implementation of the model and running experiments to validate the model and solve the problem are required.