

Robert CHODOREK

Politechnika Świętokrzyska, Katedra Elektroniki i Teletransmisji

MULTIMEDIA I PROTOKOŁY MULTIMEDIALNE

Streszczenie. W artykule przedstawiono podstawową charakterystykę aplikacji multimedialnych. Zaprezentowano protokoły sieci komputerowych zaprojektowane do optymalnej pracy z aplikacjami multimedialnymi.

MULTIMEDIA AND MULTIMEDIA PROTOCOLS

Summary. In this paper we present brief characteristics of multimedia applications. We also present multimedia transport protocols specially designed to provide best service to multimedia applications.

1. Wprowadzenie

Współczesne aplikacje bardzo często korzystają bezpośrednio z usług świadczonych przez protokół transportowy, z pominięciem wyższych warstw sieci. Dlatego istotną rzeczą jest dokładne zdefiniowanie usług, jakie może świadczyć dany protokół transportowy [16]. Klasyczne protokoły transportowe, takie jak TCP [13], UDP [14] czy TP4 [8], dostarczają usług wystarczających wielu aplikacjom, np. transmisję plików czy pocztę elektroniczną. Dla nowych aplikacji multimedialnych konieczne są dodatkowe, nie występujące w tych protokołach, usługi. Wynika stąd konieczność modyfikacji istniejących protokołów transportowych lub stworzenia nowych, bardziej odpowiadających współczesnym wymaganiom [12].

W rozdziale 2 przedstawiono podstawową charakterystykę aplikacji multimedialnych i ich wymagań stawianych sieciom teletransmisyjnym. Kolejne rozdziały prezentują wybrane,

interesujące protokoły transportowe, pozwalające dobrze obsłużyć aplikacje multimedialne. Protokoły te były projektowane lub dostosowywane do aplikacji multimedialnych.

2. Charakterystyka aplikacji multimedialnej

System multimedialny zawiera i integruje tekst, dźwięk, głos, grafikę, obrazy i video. Każda z tych usług posiada inne parametry ruchowe, generuje inny strumień danych o określonych parametrach. Inne są także wymagania aplikacji: przykładowo dane tekstowe czy grafika są mało wrażliwe na opóźnienia wnoszone przez sieć transmisyjną, natomiast sygnał audio czy video wymaga ścisłej synchronizacji i "ciągłej" transmisji. Wymienione aplikacje wymagają także skrajnie różnych przepustowości sieci od niewielkich dla tekstu do bardzo dużych dla transmisji obrazów o dużej rozdzielczości czy transmisji telewizyjnej HDTV.

Wymagania każdej aplikacji można zawrzeć w parametrach określających jakość transmisji QoS (*Quality of Service*). Najważniejszymi parametrami są:

- przepustowość sieci,
- opóźnienia transmisji,
- fluktuacje transmitowanego strumienia danych (*jitter*),
- stopa błędów.

Przepustowość systemu przy transmisji typowych 25 obrazów na sekundę o rozdzielczości 640x480 pikseli z 24-bitowym kodowaniem kolorów wymaga (bez kompresji) transmisji o przepustowości 175 Mb/s. Wprowadzenie kompresji pozwoli znacznie ograniczyć ten strumień danych [1]. Pojawia się wtedy inny problem - ilość transmitowanych danych zależy od dynamiki transmitowanych scen, przyjętej metody kompresji, poziomu strat informacji, jaki dopuszczamy w zastosowanej metodzie kompresji itd. Utrudnia to predykcję wymaganego pasma transmisyjnego, jak również zwiększa wpływ błędów transmisyjnych na jakość transmisji.

Opóźnienia transmisji są szczególnie krytyczne przy aplikacjach czasu rzeczywistego. Istotne jest to także w przypadku transmisji wizji czy też fonii [15]. Natomiast parametr ten jest mniej istotny dla transmisji grafiki czy obrazów.

Dla wielu usług multimedialnych ważny jest mały poziom fluktuacji czasowych transmitowanego strumienia danych. Musi on być bardzo mały dla transmisji video, tak aby nie występowały zniekształcenia transmitowanego obrazu. Istotny jest on również dla transmisji fonii.

Dla ostatniego parametru - stopy błędów możemy określić także pewną gradację wymagań. Generalnie zależy nam na bezbłędnej transmisji. Jednak są aplikacje, w których

wystąpienie błędu transmisji nie jest krytyczne, a nawet dopuszczalne (np. w transmisji głosu drobne, często niezauważalne zakłócenia są tolerowane przez użytkownika).

3. MSP

Protokół MSP (*MultiStream Protocol*) [11][10] zaprojektowany został w celu zapewnienia usług transmisyjnych dla szerokiej grupy współczesnych aplikacji. Dlatego też protokół posiada szereg mechanizmów protokołowych pozwalających obsłużyć różnorodne typy aplikacji. Istotną zaletą protokołu jest możliwość zmiany parametrów transmisji i mechanizmów protokołowych w czasie trwania połączenia. Pozwala to dynamicznie dostosowywać się do zmieniających się parametrów strumienia informacyjnego generowanego przez aplikacje. Jest to szczególnie istotne dla aplikacji multimedialnych, pozwalając obsłużyć je w sposób optymalny.

Tabela 1

Strumień	Charakterystyka strumieni			Atrybuty
	Typ usługi			
	Pakietowa/ Blokowa	korekcja błędów	odbiór danych w kolejności	
1	P	tak	tak	akceptuje pakiety tylko w kolejności
2	P	nie	tak	dopuszcza nieciągłości strumienia
3	P	tak/nie	nie	dopuszcza nieciągłości strumienia, akceptuje zagubienie pakietów
4	P	tak	tak	dopuszcza nieciągłości strumienia, akceptuje zagubienie pakietów
5	B	nie	tak	brak mechanizmów zapewniających niezawodność
6	B	tak	nie	niezawodna transmisja
7	B	nie	tak	niezawodna transmisja

Użytkownik może mieć pewną trudność w doborze odpowiedniego zestawu funkcji, optymalnego dla danej aplikacji. Dlatego też zdefiniowano profile protokołu nazywane strumieniami. Stanowią one podzbiór mechanizmów protokołowych z odpowiednim zestawem parametrów.

W protokole MSP zdefiniowano siedem strumieni informacji (tabela 1). Pierwsze cztery strumienie posiadają możliwość transmisji z wykorzystaniem klasycznych jednostek danych protokołowych (PDU), trzy następne realizują transmisję bloku danych (realizowana transmisja zbliżona jest do transmisji uzyskiwanych w protokołach SNR czy NETBLT).

Strumień numer 5 wykorzystywany jest do transmisji video. Zapewnia on wymaganą przepustowość i maksymalne uproszczenie przetwarzania protokołowego. Brak mechanizmów korekcji błędów wynika z wymagań transmisji video, które nie pozwalają na retransmisję informacji. Inne strumienie wykorzystywane są dla innych rodzajów transmisji, np. strumień numer 7 do transmisji plików.

Protokół został zaprojektowany do implementacji w architekturze równoległej. Stanowi on zestaw funkcji protokołowych, które można przetwarzać równolegle. Jest to zgodne z architekturą HOPS (*Horizontally Oriented Protocol*). W protokole wykonywane są tylko te funkcje, które w danej chwili potrzebne są aplikacji (zgodnie ze zdefiniowanym strumieniem). Pozwala to na ograniczenie przetwarzania protokołowego dla części aplikacji i mniejsze obciążenie systemu wieloprocesorowego. W ten sposób można optymalnie gospodarować zasobami systemu przy współpracy z różnymi typami aplikacji.

4. MHTP

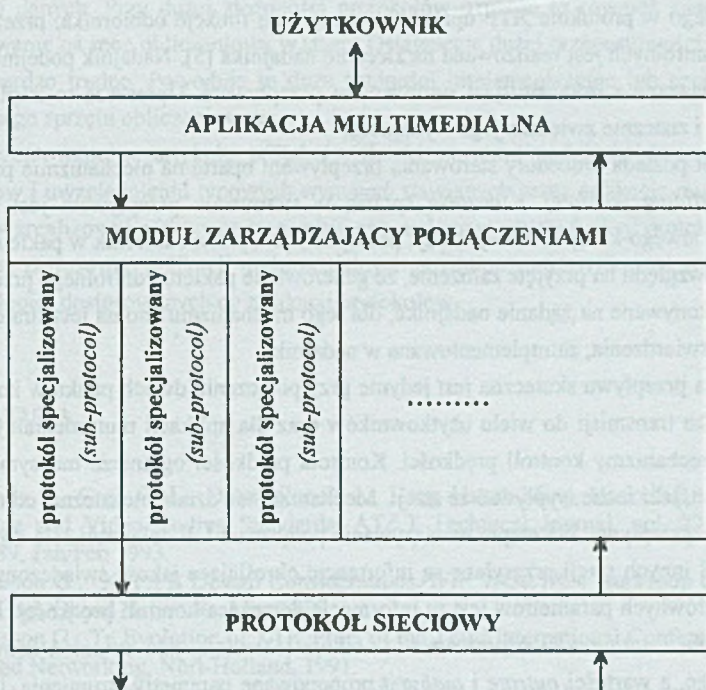
Protokół MHTP (*Multimedia High-speed Transport Protocol*) [7][9] został zaprojektowany do pracy z aplikacjami multimedialnymi. Posiada on unikalną architekturę, przedstawioną na rys. 1. W protokole MHTP jest moduł zarządzający połączeniami multimedialnymi (*multimedia connection manager*) oraz moduły protokołowe będące w istocie oddzielnie zaprojektowanymi, specjalizowanymi protokołami (*sub-protocols*) do określonych typów transmisji, np. głosu, tekstu, obrazu.

Ramka protokołu MHTP zawiera informacje pozwalające właściwie sterować poszczególnymi protokołami i optymalizować transmisje. W ramce określony jest typ transmisji określający, czy jest to transmisja dźwięku, video, tekstu, czy innych strumieni danych. Dane te uzupełnia pole przenoszące parametry jakości usług (QoS). Składają się na nią pasmo wymagane przez aplikację, tolerowana przez aplikację stopa błędów i maksymalne tolerowane przez aplikację opóźnienie. Dane o jakości usług uzupełniane są flagami bitowymi pozwalającymi selektywnie sterować dodatkowymi usługami. Pozwala to wyłączać generowanie sumy kontrolnej dla medium transmisyjnego posiadającego stopę błędów mniejszą od 10^{-9} do 10^{-12} (np. łącza światłowodowe). Zmniejsza to znacznie zapotrzebowanie na moc obliczeniową dla protokołu. Istnieje również możliwość wyłączenia retransmisji pakietów - opcja użyteczna dla transmisji dźwięku czy też obrazów.

Protokół oferuje trzy możliwe typy realizacji połączenia: transmisję połączeniową, transmisję bezpołączeniową oraz szybką transmisję połączeniową. Aplikacja może wybrać właściwy wariant transmisji, stosownie do swoich wymagań.

W szybkich protokołach realizujących transmisję dużych ilości informacji konieczne jest też zapewnienie innych mechanizmów sterowania przepływem. W protokole MHTP mechanizm sterowania przepływem bazuje na określeniu tempa nadawania, tzn. określeniu liczby bajtów, jaką można nadać w określonym czasie (*rate control*).

Protokół koryguje błędy poprzez retransmisję: grupową lub selektywną. Wybór metody zależy od wymagań aplikacji oraz od pojemności bufora odbiorczego i nadawczego.



Rys. 1. Architektura protokołu MHTP
Fig. 1. The MHTP architecture

5. XTP

Prace nad protokołem XTP (*Xpress Transport Protocol*) [2][18] rozpoczęto w 1986 r., wykorzystując doświadczenia z protokołów już istniejących, bądź projektowanych: DATAKIT, TCP, Delta-t, NETBLT, VMTP, TP4. Protokół ten zaprojektowano do pracy w szybkich sieciach teleinformatycznych, świadczących szeroki zakres usług [3][19].

Protokół transportowy XTP został zaprojektowany pod kątem efektywnej realizacji nowych zadań stawianych systemom sieciowym, ze znaczną redukcją operacji przetwarzania protokolowego i łatwą implementacją z użyciem układów VLSI.

Podczas projektowania protokołu XTP jego twórcy przeanalizowali działanie kilku typowych protokołów transportowych. Stwierdzili wówczas, że wydajność transmisji w dużym stopniu zależy od wydajności odbiornika [2][17], gdyż to właśnie odbiornik musi wykonać wiele operacji związanych z testowaniem poprawności przesyłanych danych i ich integralności. Dlatego w protokole XTP uproszczono znacznie funkcje odbiornika, przez co część operacji kontrolnych jest realizowana na zlecenie nadajnika [5]. Nadajnik podejmuje decyzje m.in. związane z retransmisją i sterowaniem przepływem. Upraszcza to implementację odbiornika i znacznie zwiększa jego wydajność.

Protokół posiada procedury sterowania przepływem oparte na mechanizmie przesuwne-go okna (*sliding window*). Kluczową sprawą w sterowaniu przepływem jest procedura przydziału nowego kredytu. Nowy kredyt przesyłany jest przez odbiornik w pakietach kontrolnych. Ze względu na przyjęte założenie, że generowanie pakietu kontrolnego przez odbiornik jest dokonywane na żądanie nadajnika, dla tego mechanizmu istotna jest strategia generowania potwierdzenia, zaimplementowana w nadajniku.

Kontrola przepływu skuteczna jest jedynie przy połączeniu dwóch punktów końcowych. W przypadku transmisji do wielu użytkowników oraz dla aplikacji multimedialnych wprowadzono mechanizmy kontroli prędkości. Kontrola prędkości ogranicza maksymalny strumień danych, jaki może wypływać ze stacji. Mechanizm ten działa niezależnie od mechanizmu kontroli przepływu. Na etapie nawiązywania połączenia lub dołączania się do trwającej konwersacji innych stacji przesyłane są informacje określające jakość świadczonych usług. Jednym z głównych parametrów jest tu informacja dotycząca kontroli prędkości: *inrate*, *inburst*, *outrate* i *outburst*. Wartości *inrate* i *inburst* określają, jaki strumień danych stacja może przyjąć, a wartości *outrate* i *outburst* proponowane parametry strumienia, jaki stacja może nadać.

Podczas transmisji danych przeprowadzona jest kontrola błędów. Dla każdego pakietu sprawdzana jest suma kontrolna nagłówka. Na żądanie aplikacji możliwe jest także kontrolowanie danych przychodzących poprzez obliczanie sumy kontrolnej także dla segmentu danych.

Z mechanizmami detekcji błędów współpracuje mechanizm korekcji błędów, oparty na zasadzie retransmisji błędnych danych. Może on pracować w dwóch trybach: retransmisji selektywnej lub retransmisji grupowej (ang. *go-back-N*). Istnieje także możliwość wyłączenia przez aplikację mechanizmu retransmisji.

Tak różnorodne typy usług realizowanych przez protokół [6] wraz ze zdefiniowanymi ostatnio rozszerzeniami protokołu o pełną negocjację jakości usług pozwalają optymalnie dostosowywać protokół do wymagań aplikacji multimedialnych.

6. Zakończenie

Aplikacje multimedialne wymagają wielu nowych usług. Wymagają również efektywnej transmisji danych. Przy dużej złożoności protokołów oznacza to również zwiększone zapotrzebowanie na moc obliczeniową w stacji. Osiągnięcie dużej przepustowości w tej sytuacji jest bardzo trudne. Powoduje to duże trudności implementacyjne lub często wymaga kosztownego sprzętu obliczeniowego.

Zaprezentowane w artykule protokoły, dzięki nowemu podejściu do projektowania protokołów i uwzględnieniu typowych wymagań stawianych przez aplikacje multimedialne, pozwalają zrealizować efektywną transmisję optymalizując użycie funkcji protokołowych.

Wzrost zastosowań aplikacji multimedialnych powinien zintensyfikować wykorzystanie nowych, lepiej dostosowanych do aplikacji protokołów.

LITERATURA

- [1] Aravin R., Cash G. L., Duttweiler D. L., Hang Hsueh-Ming, Haskell B. G., Puri A.: Image and Video Coding Standards. AT&T Technical Journal, vol. 72, no. 1, pp. 67-89, Jan/Feb 1993.
- [2] Chesson G.: XTP/PE Design Considerations. IFIP WG6.1/6.4 workshop on Protocols for High-Speed Networks, May 1989.
- [3] Chesson G.: The Evolution of XTP. Proc. of the Third International Conference on High Speed Networking, North-Holland, 1991.
- [4] Choderek R.: Sterowanie przepływem danych w protokole transportowym XTP. I Konferencja Naukowa Wpomaganie Komputerowe w Zarządzaniu, Kielce 1994.
- [5] Choderek R., Pach A.: Problemy implementacyjne szybkich protokołów transportowych. Zeszyty Naukowe Politechniki Śląskiej, Seria Informatyka z.30, s. 421-434, Gliwice 1996.
- [6] Choderek R.: Pewne aspekty zarządzania połączeniami w protokole XTP. Zeszyty Naukowe Politechniki Śląskiej, Seria Informatyka z.30, s. 523-536, Gliwice 1996.

- [7] EunGi K., SunShin A.: High-Level Error Control Mechanisms for Multimedia Transport Protocol. SUPERCOM/ICC'94, pp. 1336-1340, New Orleans, May 1994.
- [8] ITU-T: Protocol for providing the OSI connection-mode transport service. ITU-T Recommendation X.224, 1993.
- [9] Jau-Hsiung H., Sheng-Horng L.: MHTP - A Multimedia High-Speed Transport Protocol. GLOBECOM'92, pp.1364-1368, December 1992.
- [10] La Porta T. F., Schwartz Mischa: Performance Analysis of MSP: Feature-Rich High-Speed Transport Protocol. IEEE/ACM Transactions on Networking, vol. 1, no. 6, pp. 740-753, December 1993.
- [11] La Porta T. F., Schwartz Mischa: The MultiStream Protocol: A Highly Flexible High-Speed Transport Protocol. IEEE Journal on Selected Areas in Communications, vol. 11, no. 4, pp.519-530, May 1993.
- [12] La Porta Thomas F., Schwartz M.: Architectures, features, and implementation of high-speed transport protocols. IEEE Network Magazine, vol. 5, no. 3, pp. 14-22, May 1991.
- [13] Postel J.: Transmission Control Protocol. RFC 793, USC/Information Sciences Institute, September 1981.
- [14] Postel J.: User Datagram Protocol. RFC 768, USC/Information Sciences Institute, September 1981.
- [15] Różański G.: Integracja usług w lokalnych sieciach komputerowych. WAT, Warszawa 1994.
- [16] Spragins J. D., Hammond J. L., Pawlikowski K.: Telecommunications: protocols and design. Addison-Wesley Publishing Company, Inc., Massachusetts 1991.
- [17] Strayer W. T., Dempsey B. J., Weaver A. C.: Making XTP Responsive to Real-Time Needs. Computer Science Report No. TR-89-18, December 1, 1989.
- [18] Strayer W. T. ed.: Xpress Transport Protocol 4.0 Specification. XTP Forum Inc., Santa Barbara, March 1995.
- [19] Weaver A. C.: Xpress Transport Protocol Version 4. XTP Forum, 1995.

Recenzent: Dr inż. Wojciech Mielczarek

Wpłynęło do Redakcji 9 grudnia 1996 r.

Abstract

New multimedia applications require a new structure of the transport layer and a new type of services. The services provided by networks varies widely. In section 2 brief characteristics of multimedia applications was presented. Basic requirements for multimedia applications were presented.

Multimedia transport protocols specially designed to provide best service to multimedia applications was presented.

In section 3 MSP (MultiStream Protocol) protocol designed to support wide range of high speed applications was presented. MSP supports seven traffics streams (streams are defined by set of common protocol function) presented in table 1. In section 4 MHTP (Multimedia High-Speed Transport Protocol) protocol was analysed. MHTP, in abstract, consist of a multimedia connection manager and a list of sub-protocols as shown in Fig. 1. In section 5 XTP (Xpress Transport Protocol) protocol was presented. The XTP is a light weight protocol, designed for high speed networks. XTP offers key-based addressing and routing lookups, message priority, flow, burst and error control, selective retransmission or go-back-N, implicit connection setup, data pipeline, message priority, multicast capabilities, addressing schemes to support TCP or OSI-style addresses.