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

Robert CHODOREK
Akademia Górniczo-Hutnicza, Katedra Telekomunikacji

PROTOKOŁY DO TRANSMISJI DANYCH MULTIMEDIALNYCH

Streszczenie. W artykule zaprezentowano wymagania, jakie stawiają współczesne aplikacje multimedialne w stosunku do sieci teleinformatycznych. Zaprezentowano także podstawowe protokoły, jakie są używane do transmisji multimedialnej. Należą do nich: MHTP, MSP, XTP, RSVP, RTP, RTCP i RTSP.

PROTOCOLS FOR TRANSMISION OF MULTIMEDIA DATA

Summary. In this paper we present requirement provided by multimedia application. We also present protocols designed for multimedia. We present protocols: MHTP, MSP, XTP, RSVP, RTP, RTCP and RTSP.

1. Wstęp

Postęp w dziedzinie transmisji danych [1] i przetwarzania informacji umożliwia realizację wielu, jeszcze niedawno niedostępnych, usług [19]. Zaliczyć do nich możemy: wideo na żądanie, telekonferencje czy też interaktywne korzystanie z multimedialnych baz danych.

Nowe aplikacje wymagają realizacji przez sieci teleinformatyczne nowych usług. Należą do nich np.: połączenia wielopunktowe dla systemów wideokonferencyjnych czy możliwość negocjacji parametrów określających jakość transmisji.

Obecnie główną rolę odgrywają aplikacje multimedialne. Aplikacje te przesyłają tekst, dźwięk, głos, grafikę, obrazy i wideo. 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 wideo 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.

Aby optymalnie sprostać tym wymaganiom, należy zastosować nowoczesne protokoły transmisyjne przystosowane do tego typu aplikacji [12]. W artykule zaprezentowano wymagania stawiane przez aplikacje multimedialne. Wskazano na podstawowe procesy, jakie występują wewnątrz aplikacji multimedialnej i jakie są związane z nimi wymagania w stosunku do sieci transmisyjnej. Zaprezentowano również protokoły, które zostały zaprojektowane do przesyłania danych multimedialnych lub do realizacji usług dodatkowych rozszerzających możliwości tradycyjnych protokołów.

2. Aplikacje multimedialne

Aplikacje multimedialne integrują kilka rodzajów danych, np.: dźwięk, grafikę czy sekwencje wideo. Powoduje to powstanie strumienia danych o złożonej charakterystyce. Dane transmitowane nie mogą być też traktowane jako prosta superpozycja poszczególnych strumieni. Zazwyczaj dane multimedialne są od siebie ściśle uzależnione czasowo. Wymagana jest ścisła synchronizacja danych (np. dźwięku stowarzyszonego z obrazem wideo) oraz gwarantowane stale opóźnienie.

Typowa sieć charakteryzuje się zmiennym opóźnieniem zależnym od chwilowego obciążenia sieci. Nie rekompensują tego zjawiska powszechnie stosowane protokoły sieciowe (IP) czy też transportowe (TCP). Nie realizują one ponadto synchronizacji danych pochodzących z kilku strumieni. Dlatego też same aplikacje rozszerzane są o możliwości multipleksacji strumieni danych jak i synchronizację.

Typowym przykładem może być aplikacja zgodna ze standardem MPEG-2 [5][7]. Strumień danych MPEG-2 ma zazwyczaj kilka składowych, odpowiadających sygnałowi wideo, sygnałowi audio, danym pobocznym zawartym w strumieniu, a niekoniecznie związanych z treścią przekazu, informacjom sterującym wspomagającym proces dekodowania oraz informacjom synchronizującym. Wynikowy strumień tworzony jest ze strumieni elementarnych stanowiących przedmiot przekazu (audio, wideo). Ma on postać pakietów, umożliwiających poprawny transport i odtwarzanie w czasie rzeczywistym, z zachowaniem współzależności czasowych pomiędzy strumieniami elementarnymi i możliwością precyzyjnego taktowania (*ang. timming*). Poszczególne strumienie elementarne są reprezentowane w postaci pakietów o zmiennej długości, tzw. pakietów elementarnych (*ang. Packettized Elementary Streams*). Pakiety takie są multipleksowane, uzupełniane o informacje systemowe umożliwiające ich separację (*ang. Program Specific Information*) oraz uzupełnianie szczegółowymi informacjami o parametrach źródła sygnału [5][6].

W podstawowej części standardu MPEG-2 [5] zdefiniowano dwa, wzajemnie wykluczające się, formaty strumieni MPEG, przenoszących zmultipleksowane pakiety elementarne: PS (*Program Stream*) i TS (*Transport Stream*).

Strumień MPEG wykorzystujący format PS przenosi strumienie elementarne (audio, wideo, dane) należące do jednego programu. Strumień PS tworzą pakiety o zmiennej długości. Strumień PS nie posiada mechanizmów multipleksacji. Stąd nie może być multipleksowany ze strumieniami pochodzącymi z innych źródeł. Format PS został opracowany dla aplikacji multimedialnych, otrzymujacych strumień danych z lokalnego źródła - na przykład z odtwarzacza CD. Transmisja ta charakteryzuje się relatywnie niską stopą błędów, stosunkowo łatwą synchronizacją, stałym opóźnieniem oraz brakiem buforowania.

Strumień transportowy TS służy do przekazu sygnałów multimedialnych w środowisku sieciowym, w stacjach telewizji naziemnej, satelitarnej i kablowej. Poprawny transport jest możliwy dzięki wprowadzeniu do strumienia informacji synchronizującej nadajnik (koder) i odbiornik (dekoder), to znaczy informacji wyznaczającej poprawny czas dekodowania i prezentacji pokazywanego materiału filmowego. Cechą charakterystyczną strumienia transportowego TS jest możliwość multipleksacji strumieni pochodzących z wielu źródeł w jeden strumień TS, który tworzą pakiety transportowe (*ang. Transport Packets*). Pakiety te mają stałą długość 188 bajtów. Możliwa jest konwersja *Program Stream* do *Transport Stream* (dokładniej do *Single Program* TS).

Utrudnieniem w transmisji danych multimedialnych w standardzie MPEG jest założenie, że w kanale transmisyjnym pomiędzy źródłem sygnału MPEG (koderem) a odbiornikiem (dekoderem) nie następuje utrata informacji, a opóźnienie transmisyjne jest stale. Założenie stałego opóźnienia umożliwia synchronizację dekodera z koderem i eliminuje złożony problem kompensacji opóźnienia. Synchronizacja odbywa się poprzez sygnatury czasu (*ang. time stamp*) zawarte w strumieniu. Sygnatury te określają czas, jaki wskazywał zegar kodera w chwili kodowania danego pakietu. W strumieniach TS sygnatury nazywane są PCR (*ang. Program Clock Reference*) i są umieszczane w strumieniu nie rzadziej niż co 700ms.

Poprawny odbiór i dekodowanie strumienia wymaga, aby dekoder MPEG-2 pozostawał w synchronizacji z koderem. Oznacza to, że czasy, jakie upływają pomiędzy przekazaniem kolejnych pakietów TS do bloku demultipleksera w dekoderze, powinny odpowiadać czasom, jakie upływały pomiędzy wygenerowaniem tych pakietów w multiplekserze kodera. Jest to trudne do realizacji w typowych sieciach teleinformatycznych. Dlatego też istotne jest znalezienie takich mechanizmów pracy sieci, aby to zagadnienie rozwiązać.

Na przykładzie transmisji danych multimedialnych w standardzie MPEG-2 przedstawiono podstawowe trudności związane z transmisją danych multimedialnych. Konieczne jest

zapewnienie synchronizacji danych, definicji parametrów jakości usług czy zestawiania połączeń wielopunktowych.

3. Protokoły transmisyjne

W sieciach teleinformatycznych stosowanych jest wiele protokołów transportowych. Współczesne protokoły charakteryzują się specyficznymi mechanizmami, dostosowanymi do konkretnych rozwiązań [13][17][18].

Transmisja danych multimedialnych to nie tylko konieczność przesłania danych. To także konieczność zapewnienia odpowiednich parametrów danej transmisji. Potrzebne są także dodatkowe usługi jak realizacja transmisji wielopunktowej. Jest to realizowane wewnątrz protokołów transportowych (np.: XTP) lub też poprzez protokoły pomocnicze (np.: RSVP).

3.1. Protokół MSP

Protokół MSP (*MultiStream Protocol*) [15][16] zaprojektowany został dla 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.

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. 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 wideo. Zapewnia on wymaganą przepustowość i maksymalne uproszczenie przetwarzania protokołowego. Brak mechanizmów korekcji błędów wynika z wymagań transmisji wideo, 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 dla 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 aplikacjami.

3.2. Protokół MHTP

Protokół MHTP (*Multimedia High-speed Transport Protocol*) [14] został zaprojektowany do pracy z aplikacjami multimedialnymi. Posiada on unikalną architekturę. W protokole MHTP jest zdefiniowany moduł zarządzający połączeniami multimedialnymi (*multimedia connection manager*) oraz moduły protokołowe będące w istocie oddzielnie zaprojektowanymi, specjalizowanymi pod protokołami (*sub-protocols*) do określonych typów transmisji, np.: głosu, tekstu, obrazu.

Ramka protokołu MHTP zawiera informacje pozwalające właściwie sterować poszczególnymi pod protokołami i optymalizować transmisje. W ramce określony jest typ transmisji określający, czy jest to transmisja dźwięku, wideo, 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.

3.3. Protokół XTP

Protokół transportowy XTP (*Xpress Transport Protocol*) [11][20] został zaprojektowany pod kątem efektywnej realizacji nowych zadań stawianych systemom sieciowym, ze znaczną redukcją operacji przetwarzania protokołowego 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, 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. 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 przesuwnego okna oraz mechanizmie kontroli prędkości. Kontrola przepływu wykorzystująca mechanizm przesuwnego okna skuteczna jest jedynie przy połączeniu dwóch punktów. Dla zapewnienia mechanizmów kontrolnych także 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. 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 wraz ze zdefiniowanymi ostatnio rozszerzeniami protokołu o pełną negocjację jakości usług pozwalają optymalnie dostosowywać protokół do wymagań aplikacji multimedialnych.

3.4. Protokół RSVP

Protokół RSVP (*Resource ReSerVation Protocol*) [2][3] jest zbliżony do protokołu ST-II [4]. Zaprojektowany został do pracy w sieciach rozległych ze szczególnym uwzględnieniem Internetu. Protokół ten zapewnia stworzenie ścieżki wirtualnej pomiędzy komunikującymi się aplikacjami. Pozwala on dla danej ścieżki na zdefiniowanie parametrów jakościowych (QoS) strumienia informacji. Zastosowanie protokołu wymaga jednak routerów posiadających zaimplementowany protokół RSVP.

W każdym z routerów rezerwowane są odpowiednie zasoby do transmisji danych. Wymaga to określenia zasobów danego routera i możliwości alokacji pasma dla dodatkowej transmisji. Po wynegocjowaniu parametrów na całej ścieżce sieć powinna zapewnić transmisję danych o zarezerwowanych parametrach jakosciowych.

Protokół RSVP pozwala na zestawienie połączeń typu punkt-punkt oraz połączeń wielopunktowych. Współpracuje z protokołem IP w wersji 4 oraz wersji 6. Protokół RSVP może pracować w sieciach heterogenicznych. Jeżeli odbiorniki reprezentują różne technologie, to protokół negocjuje różne parametry QoS, specyficzne dla danej technologii. Protokół pozwala na dynamiczną zmianę grupy hostów wymieniających informację. Sterowanie parametrami jakościowymi jest adaptacyjne. Uwzględnia ono zmianę trasy przechodzenia pakietów oraz pogorszenie parametrów transmisyjnych.

3.5. RTP

Protokół RTP (*Realtime Transport Protocol*) [8][9] jest protokołem transportowym zaprojektowanym do transmisji danych w czasie rzeczywistym. Własności protokołu pozwalają na wykorzystanie go jako protokołu do transmisji danych multimedialnych, takich jak interaktywne usługi audio lub wideo. Najczęściej protokół RTP korzysta z usług protokołu UDP, który zapewnia multipleksacje oraz kontrolę błędów. Możliwa jest także współpraca tego protokołu z innymi realizacjami niższych warstw sieci. Szczególnie istotna jest możliwość bezpośredniej współpracy z sieciami ATM poprzez AAL5. Protokół RTP przewidziano tak dla transmisji typu punkt-punkt jak również do transmisji wielopunktowych.

RTP wewnętrznie nie zawiera mechanizmów zapewniających właściwe dostarczanie danych (z odpowiednimi parametrami czasowymi). Zapewnia jednak mechanizmy sekwencyjnego dostarczania danych oraz synchronizacji danych. Protokół RTP identyfikuje również rodzaj przesyłanych danych (audio, wideo) oraz metodę kodowania danych (np.: PCM, ADPCM).

Każdy pakiet protokołu RTP posiada znacznik czasowy. Pozwala on na realizację synchronizacji oraz określenie zmienności czasu dostarczenia pakietu - zmienności opóźnień (*jitter*). Wartość startowa znacznika czasowego jest losowa. Przy realizacji prawidłowej transmisji powinien on liniowo wzrastać.

Wraz z protokołem RTP współpracuje protokół RTCP (*Real-Time Control Protocol*) [8]. Realizuje on szereg funkcji pomocnicznych niezbednych do prawidłowej pracy protokołu RTP jak również aplikacji. Protokół RTCP monitoruje transmisję, określając jakość transmisji. Informacja ta może być wykorzystana przez aplikację. Oprócz funkcji monitorowania połączeń protokół RTCP identyfikuje uczestników konwersacji. Zapewnia także tworzenie grup konferencyjnych z pełną identyfikacją uczestników. Stan konferencji jest przekazywany na bieżąco. W tym celu protokół RTCP wysyła regularnie pakiety do wszystkich uczestników konferencji, korzystając z tych samych mechanizmów co protokół bazowy RTP.

Obecnie protokoły RTP jak i RTCP są wykorzystywane w kilku komercyjnych aplikacjach. Promuje go firma Netscape jak i firmy z nią związane. Protokół RTP wykorzystywany jest także do realizacji transmisji strumieniowej. Wymaga to jednak dodatkowych protokołów takich jak RTSP.

3.6. RTSP

Protokół RTSP (*Real-Time Streaming Protocol*) [10] jest protokołem warstwy aplikacji służącym do sterowania prezentacją multimedialną. Typowa prezentacja składa się z wielu strumieni multimedialnych. Protokół RTSP posiada mechanizmy protokołowe służące do sterowania transmisją strumieniową realizowaną w czasie rzeczywistym. Dla każdego strumienia protokół RTSP dokonuje wyboru odpowiedniego protokołu transportowego. Może on wykorzystać protokoły klasyczne, jak np.: UDP czy TCP lub też protokół RTP.

Protokół RTSP syntaktycznie zbliżony jest do protokołu HTTP 1.1. Zapożyczył on od protokołu HTTP mechanizmy ochrony informacji, schematy adresacji czy też metody rozliczeń finansowych. Został on jednak znacznie rozszerzony i przebudowany w porównaniu z protokołem HTTP. W odróżnieniu od serwera protokołu HTTP serwer protokołu RTSP musi posiadać informacje o stanie połączenia. Protokół RTSP jest także protokołem symetrycznym. W protokole RTSP zdefiniowano metody niezbędne do sterowania transmisją strumienia danych poprzez sieć. Są to m.in. metody: SETUP, REDIRECT, PLAY, PAUSE i RECORD.

Protokół RTSP pozwala na sterowanie typową transmisją punkt-punkt dla realizacji usług audio lub wideo na żądanie (*audio/video on demand*). Pozwala również na sterowanie transmisją wielopunktową dla usług typu prawie wideo na żądanie (*near media-on-demand*).

Pozwala również na sterowanie transmisjami dla systemów konferencyjnych. Przydział zasobów sieciowych dla obydwu rodzajów transmisji odbywa się z wykorzystaniem innych protokołów jak chociażby RSVP.

Zapewnia on synchronizację procesów transmisyjnych - synchronizacja wielu strumieni. Synchronizacja taka jest niezbędna dla realizacji transmisji multimedialnej z jednego serwera jak i z wielu serwerów. Poszczególne zdarzenia są synchronizowane za pomocą znaczników czasowych. W protokole RTSP są stosowane znaczniki czasowe względne NPT (czas wyznaczany względem początku prezentacji) lub znaczniki czasowe bezwzględne AT (aktualny czas).

4. Podsumowanie

Przedstawione protokoły możemy podzielić generalnie na dwie grupy: protokoły sterujące i protokoły transportowe. Do pierwszej z grup zaliczamy MHTP, MSP, XTP czy RTP. Do drugiej grupy zaliczamy RSVP i RTSP. Protokoły te zapewniają szereg mechanizmów ułatwiających realizację transmisji multimedialnej. Pozwalają na definiowanie połączeń wielopunktowych, na definicję parametrów jakości usług oraz na realizację synchronizacji (RTP). Są to najważniejsze wymagania stawiane transmisji multimedialnej. Zaprezentowane protokoły „wspólnie" realizują zadania stawiane przez aplikację multimedialną. Każdy z nich realizuje podzbiór funkcji koniecznych do realizacji takiej transmisji. Dlatego konieczne są dalsze badania i poszukiwania nowych, efektywnych protokołów lub modyfikacja tych już istniejących

LITERATURA

[1] Boisseau M., Demange M., Munier J. M.: High-speed networks. John Wiley & Sons, 1994.

[2] Braden R., Zhang L., Berson S., Herzog S., Jamin S.: Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification. RFC 2205, September 1997.

[3] Mankin A., Baker F., Braden B., BradnerS., O'Dell M., Romanow A., Weinrib A., Zhang L.: Resource ReSerVation Protocol (RSVP) -- Version 1 Applicability Statement Some Guidelines on Deployment. RFC 2208, September 1997.

[4] Delgrossi L., Berger L.: Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2. RFC 1819, August 1995.

[5] ITU-T: Information technology – generic coding of moving pictures and associated audio information: systems. H.222.0, 1995.

[6] ITU-T: Multimedia multiplex and synchronization for audiovisual communication in atm environments. H.222.1, 1996.

[7] ITU-T: Information technology – generic coding of moving pictures and associated audio information: video. H.262, 1995.

[8] Schulzrinne H., Casner S., Frederick R., Jacobson V.: RTP: A Transport Protocol for Real-Time Applications. RFC 1889. January 1996.

[9] Schulzrinne H.: RTP Profile for Audio and Video Conferences with Minimal. RFC 1890. January 1996.

[10] Schulzrinne H., Rao A., Lanphier R.: Real Time Streaming Protocol (RTSP). Internet Engineering Task Force, November 1997.

[11] Chesson Greg: XTP/PE Design Considerations. IFIP WG6.1/6.4 workshop on Protocols for High-Speed Networks, May 1989.

[12] Chodorek Robert R.: Multimedia i protokoły multimedialne, Seminarium Sieci Komputerowych, Gliwice 1997.

[13] Doshi Bharat T., Johri Pravin K., Netravali Arun N., Sabnani Krishan K.: Error and Flow Control Performance of a High Speed Protocol. IEEE Transactions on Communications, vol. 41, no. 5, pp. 707-720, May 1993.

[14] Jau-Hsiung Huang, Sheng-Horng Lee: MHTP - A Multimedia High-Speed Transport Protocol. GLOBECOM'92, pp.1364-1368, December 1992.

[15] La Porta Thomas 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.

[16] La Porta Thomas 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.

[17] Meleis Hanafy E., Serpanos Dimitrios N.: Designing Communication Subsystems for High-Speed Networks. IEEE Network Magazine, vol. 6, no. 4, pp. 40-46, July 1992.

[18] Netravali Arun N., Roome W. D., Sabnani K.: Design and Implementation of High-Speed Transport Protocol. IEEE Transactions on Communications, vol. 38, no. 11, pp. 2010-2024, November 1990.

[19] Ransom M. Niel, Spears Dan R.: Applications of Public Gigabit Networks. IEEE Network Magazine, vol. 6, no. 2, pp. 30-40, March 1992.

[20] Strayer W. T. ed.: Xpress Transport Protocol 4.0 Specification. XTP Forum Inc., Santa Barbara, March 1995.

Recenzent: Dr inż. Bartłomiej Zieliński

Wpłynęło do Redakcji 15 grudnia 1997 r.

Abstract

In article we present multimedia application. Specially we present the MPEG-2 application. Base on this application we present a requirement for telecommunication network. We also present protocols designed for multimedia. All protocols we divide to two groups: transport protocols and control protocols. The first group consist of protocols: MHTP, MSP, XTP and RTP. The second group consist of RSVP and RTSP.