STUDIA INFORMATICA 2000
Volume 21 Number 1 (39)

Robert R. CHODOREK
Akademia Górniczo-Hutnicza, Katedra Telekomunikacji

WSPÓŁDZIAŁANIE APLIKACJI MULTIMEDIALNYCH I PROTOKOŁÓW TRANSPORTOWYCH W WARUNKACH LIMITOWANYCH ZASOBÓW SIECI DOSTĘPOWYCH

Streszczenie. W artykule przedstawiono metody transmisji multimedialnych w sieciach teleinformatycznych. Zaprezentowano metody stosowane w celu zapewnienia poprawnej transmisji przy limitowanych zasobach sieci dostępowej.

COOPERATION OF MULTIMEDIA APPLICATION WITH TRANSPORT PROTOCOLS IN ACCESS NETWORKS WITH LIMITED RESOURCES

Summary. The multimedia network application composes many different types of data. To transmit that data we must use specialized protocols or expand application functionality. In the paper an analysis of transmission multimedia data in access network with limited resources was presented.

1. Wprowadzenie

Do niedawna w Internecie dominowała transmisja prostych danych (tekst, pliki binarne, grafika). Współczesne aplikacje ewoluują w kierunku prezentacji multimedialnych. Powoduje to zmianę wymagań stawianych sieci transmisyjnej.

Użytkownicy sieci uzyskują dostęp do Internetu najczęściej poprzez sieci dostępowe. Współcześnie stosowane sieci dostępowe zapewniają szeroki zakres prędkości:

- 33,6 ÷56 kb/s dla klasycznych modemów,
- 64 kb/s lub 128 kb/s dla sieci ISDN,
- 115 kb/s dla technologii HiS (Home internet Solution),
- 1,5 ÷8 Mb/s dla technologii ADSL (*Asymmetric Digital Subscriber Line*),
- 52 Mb/s dla technologii VDSL (*Very high speed Digital Subscriber Line*).

W sumie sieci te posiadają przepustowości od 33,6 kb/s do kilku Mb/s. Przy tak dużych różnicach istnieje problem z zapewnieniem odpowiedniej jakości świadczonych usług. W wielu przypadkach należałoby dostosowywać jakość usługi do każdej sieci dostępowej.

W artykule zostanie zaprezentowana współpraca aplikacji multimedialnych z siecią teleinformatycznymi i dokonana analiza metod zapewnienia poprawnej transmisji w sieci przy ograniczonych zasobach. Przeanalizowany zostanie również wpływ transmisji multimedialnych na jakość realizacji usług klasycznych.

2. Współdziałanie aplikacji multimedialnych z siecią

Aplikacje multimedialne korzystają z wielu usług sieciowych [1]. Wymagają one efektywnej transmisji stosunkowo dużego strumienia danych (np. transmisja wideo) do wielu odbiorców oraz określonej jakości świadczonych usług transportowych (QoS) [2]. W wielu przypadkach aplikacje samodzielnie oferują dodatkowe usługi związane z transmisją w sieci. Przykładowo, w standardzie MPEG przewidziano procedury związane z pakietyzacją strumienia oraz z synchronizacją strumieni podczas transmisji poprzez sieć. Umożliwiło to zdefiniowanie współpracy standardu MPEG bezpośrednio z siecią ATM. Istnieje jednak duża grupa aplikacji wymagająca bogatego zestawu usług sieciowych.

Generalnie aplikacje multimedialne wymagają efektywnej transmisji strumienia danych. Istotne jest, aby transmisja odbywała się rytmicznie bez zmiany wielkości opóźnienia pomiędzy kolejnymi pakietami (lub grupami pakietów przenoszących jeden element przekazu, np. pojedynczą ramkę obrazu). Z tego też względu dla wielu aplikacji niedopuszczalna jest retransmisja uszkodzonych pakietów. Wynika to z faktu, iż czas pomiędzy dostarczeniem oryginalnego pakietu a dostarczeniem jego poprawnej kopii jest powyżej czasu tolerowanego przez aplikacje. Dlatego dla wielu aplikacji dopuszcza się możliwość niewielkich strat pakietów, ale nie dopuszcza się do nagłego zwiększenia opóźnienia (np. podczas transmisji wideo lub audio). Gdy jednak potrzebna jest niezawodność przekazu, można stosować inne metody korekcji błędów, np. FEC (*Forward Error Correction*) korzystające z nadmiarowości przesyłanej informacji.

Oprócz transmisji danych konieczne jest zapewnienie dodatkowych usług. Jednym z ważniejszych zagadnień jest synchronizacja wielu strumieni. Innym jest możliwość sterowania przebiegiem prezentacji multimedialnej.

Do transmisji danych wykorzystywane są protokoły transportowe. W przypadku aplikacji multimedialnych musi on świadczyć usługi w trybie multicast oraz posiadać możliwość definiowania jakości usług (QoS). Klasyczne protokoły (np. TCP) nie posiadają mechanizmów charakterystycznych dla nowszych rozwiązań, takich jak szybkie zestawianie

połączeń czy multicast [1]. Różnorodność transmisji multimedialnych powoduje, że stosowanych bywa kilka protokołów transportowych, dostosowanych do konkretnych rozwiązań. Do najpopularniejszych możemy zaliczyć: TCP, UDP oraz RTP. Do sterowania prezentacją multimedialną często wykorzystywany jest protokół RTSP.

2.1. Protokół TCP

Protokół TCP (*Transport Control Protocol*) [6] zaprojektowano dla usług, które można było zrealizować na początku lat 80: poczta elektroniczna (bez możliwości przekazów multimedialnych), transmisja plików za pomocą FTP czy interaktywna praca na zdalnym komputerze (usługa telnet). Wiele usług nie sposób było wówczas zrealizować na typowym systemie końcowego użytkownika. Również sama sieć stanowiła poważne ograniczenie. Te realia wpłynęły na ostateczny kształt protokołu TCP. Obecnie protokół TCP spełnia rolę protokołu transportowego dla przesyłania sygnałów sterujących przebiegiem sesji multimedialnej. Wykorzystywany jest on m.in. przez protokoły HTTP i RTSP.

Specyfikacja protokołu TCP została oficjalnie zdefiniowana we wrześniu 1981 roku w dokumencie RFC793 [6]. W ciągu 19 lat, jakie upłynęły od tego czasu, pierwotna specyfikacja protokołu była kilkakrotnie zmieniana. Celem tych zmian było wprowadzenie dodatkowych usprawnień, jak również poprawienie błędów pierwotnej specyfikacji protokołu [4][9]. W efekcie pojawiło się kilka wersji rozwojowych protokołu TCP, z których najbardziej znanymi są: TCP Tahoe, TCP Reno, TCP New-Reno, TCP Vegas i TCP SACK. Każda z wersji protokołu TCP posiada inne algorytmy powolnego startu, zapobiegania zatorom i inne strategie korekcji błędów. Ma to wpływ na poprawę wydajności protokołu.

W opracowanej w 1988 roku wersji TCP Tahoe wprowadzono algorytmy: powolnego startu (*Slow-Start*), zapobiegania zatorom (*Congestion Avoidance*) oraz szybkiej retransmisji (*Fast Retransmit*).

Wersja TCP Reno uzupełnia mechanizm korekcji błędów o szybkie odtwarzanie połączenia (*Fast Recovery*). W przypadku zagubienia segmentu algorytm szybkiego odtwarzania pozwala na transmisję nowych danych bez konieczności wykonania powolnego startu.

W wersji TCP New-Reno wprowadzono jedynie zmianę parametrów szybkiego odtwarzania. Pozwala to na eliminację pewnych problemów powodujących słabą wydajność TCP Reno, bez potrzeby wprowadzania trudnej w implementacji wersji SACK.

Wersja TCP SACK (*Selective ACKnowledgement*) [4] posiada algorytm selektywnej retransmisji (dotychczasowe wersje protokołu posiadały jedynie algorytm Go-back-N).

Wersja TCP Vegas oparta jest na implementacji TCP Reno. Zmianie uległ sposób estymacji pasma transmisyjnego. Zmodyfikowano również mechanizmy: retransmitowania pakietów, zapobiegania zatorom oraz powolnego startu.

TCP FACK jest to wersja TCP Reno z mechanizmem Forward Acknowledgment. Algorytm FACK kontroluje przeciążenia podczas odtwarzania połączenia poprzez dokładne określanie ilości danych pozostających w sieci (TCP Reno zakłada, że każde otrzymane zduplikowane ACK reprezentuje jeden segment, który opuścił sieć).

Niezależnie jednak od wersji protokół TCP zapewnia niezawodne przesyłanie danych korzystając z mechanizmu ACK - pozytywnego potwierdzenia. Mechanizm ten jest skuteczny dla łączy o dużej stopie błędów i stosunkowo niewielkiej przepustowości. Jednakże w przypadku łączy o dużych prędkościach transmisji, jak również dla łączy charakteryzujących się dużym opóźnieniem (np. satelitarnych) mechanizm pozytywnego potwierdzenia powoduje znaczne ograniczenie wydajności protokołu. Podobne problemy występują w przypadku mechanizmu sterowania przepływem.

TCP został zaprojektowany jako protokół transportowy ogólnego przeznaczenia, nie jest wobec tego zoptymalizowany pod kątem przenoszenia konkretnego typu danych. Obecnie jest on wykorzystywany głównie w celu niezawodnej transmisji danych. TCP jest jednym z najczęściej stosowanych protokołów transportowych. Jest on podstawowym protokołem transportowym w sieci Internet.

2.2. Protokół UDP

Najprostszym protokołem wykorzystywanym do transmisji multimedialnych jest protokół UDP (*User Data Protocol*). Jest on protokołem bezpołączeniowym, który realizuje transmisję bez mechanizmów zapewnienia niezawodności transmisji danych [7]. Pozwala na identyfikację aplikacji w systemach docelowych. Protokół może realizować detekcję błędów za pomocą sum kontrolnych, nie ma jednakże zdefiniowanych mechanizmów korekcji błędów. Nie realizuje również kontroli poprawności otrzymywanego strumienia danych pod względem kolejności otrzymywanych danych (błędna kolejność otrzymywanych danych może wystąpić np. przy innym wyborze trasy dla każdego pakietu). Podobnie jak protokół TCP protokół UDP pracuje na bazie protokołu sieciowego IP [5].

Możliwość zastosowania protokołu UDP do transmisji multimedialnych wynika z braku opóźnień spowodowanych przez mechanizmy retransmisji. Protokół UDP stosujemy również do realizacji aplikacji typu klient-serwer, w których występują krótkie i pojawiające się często komunikaty typu: zapytanie-odpowiedź. Protokół UDP świadczy również usługi dla innych protokołów transportowych jak chociażby RTP.

2.3. Protokół RTP

Protokół RTP (*Realtime Transport Protocol*) [8] 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 zarówno dla transmisji typu punkt-punkt (*unicast*), jak również do transmisji wielopunktowych (*multicast*).

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.

Z protokołem RTP współpracuje protokół RTCP (*Real-Time Control Protocol*) [8]. Realizuje on wiele funkcji pomocniczych niezbędnych do prawidłowej pracy protokołu RTP. 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.

RTCP spełnia cztery funkcje:

- monitorowanie jakości (QoS) transmisji danych. Na podstawie tej informacji można zrealizować kodowanie adaptacyjne (dostosowywanie jakości kodowanego przekazu do jakości usług świadczonych przez sieć),
- identyfikacja źródła informacji,
- skalowanie sesji. Dzięki tej funkcji każdy użytkownik posiada informacje o ilości stacji biorących udział w sesji. Liczba ta jest używana do obliczenia prędkości wysyłania pakietów,
- przenoszenie dodatkowej informacji kontrolnej sesji, np.: identyfikacja uczestnika w celu zobrazowania w interfejsie użytkownika. Jest to funkcja opcjonalna.

2.4. Protokół RTSP

Protokół RTSP (*Real Time Streaming Protocol*) jest protokołem poziomu aplikacji (*application-level*) służącym do sterowania dostarczaniem danych transmitowanych w czasie rzeczywistym. RTSP dostarcza elastyczną strukturę umożliwiającą kontrolowanie przesyłanych na żądanie danych czasu rzeczywistego, takich jak audio i video. Podczas sesji RTSP klient RTSP ma możliwość otwierania i zamykania wielu połączeń transportowych do serwera w celu wydania rozkazów RTSP.

Protokół RTSP wspiera następujące operacje:

- opis prezentacji z serwera mediów,
- zaproszenie grupy serwerów mediów do konferencji,
- dodawanie mediów do istniejących prezentacji.

Strumienie sterowane przez RTSP mogą korzystać z RTP, ale działanie RTSP nie zależy od mechanizmu transportowego używanego do przenoszenia ciągłych mediów. Można chociażby użyć protokołu UDP czy TCP. Protokół RTSP jest podobny w składni i operacjach do HTTP 1.1. Jednak protokół RTSP różni się fundamentalnie od HTTP w dostawie danych. HTTP jest asymetrycznym protokołem, gdzie klient żąda emisji, a serwer odpowiedzi. W RTSP zarówno klient media, jak i serwer media może żądać emisji.

3. Rezerwowanie zasobów sieci

Aplikacje multimedialne do poprawnego działania wymagają określonych zasobów. Potrzebne są zatem mechanizmy do ich rezerwacji. Problem rezerwacji został w znacznym stopniu rozwiązany w technologii ATM. Jest on jednak możliwy do stosowania tylko w sieci homogenicznej ATM. Aby można było zrealizować rezerwację zasobów w Internecie, opracowany został protokół RSVP.

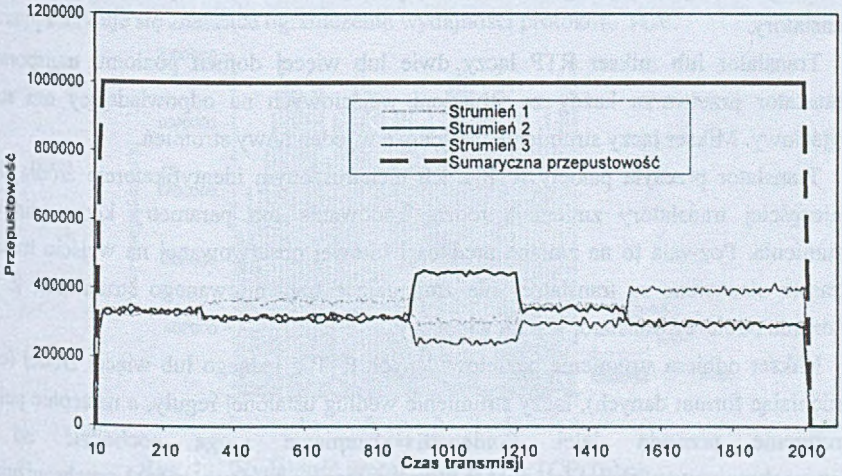
Generalnie protokół rezerwacji zasobów - RSVP (*Resource ReSerVation Protocol*) jest stosowany, gdy aplikacja żąda określonej jakości przepływu danych lub strumieni w sieci. RSVP jest także używany przy dostarczaniu zgłoszeń QoS (*Quality-of-Service*) do wszystkich routerów w węzłach wzdłuż ścieżek strumieni oraz ustalenia i utrzymania stanu obsługi żądań. RSVP nie jest protokołem routingu, dlatego nie narzuca specyficznej metodyki znajdowania trasy. Protokół RSVP jest konstrukcją do operowania na bieżących i przyszłych protokołach routingu w trybie unicast i multicast. Proces RSVP sprawdza lokalne bazy danych routingu w celu otrzymania trasy.

Protokół ten zgłasza zapotrzebowanie zasobów dla strumieni jednokierunkowych. RSVP współpracuje z IPv4 [5] lub IPv6 [3].

Proces rezerwacji rozpoczyna się z chwilą, gdy źródło danych, na przykład aplikacja serwera, wysyła do potencjalnego odbiorcy komendę PATH. Komenda zawiera identyfikator strumienia, który wskazuje routerom, jak rozpoznać datagramy jako część sesji RSVP. Mechanizm ten wspierany jest przez protokół IPv6 [3], który posiada identyfikator strumienia. Komenda PATH opisuje strumień informacji, który ma nadejść. Każdy router obsługujący komendę PATH zapamiętuje identyfikator przepływu i łącze, przez które przeszła komenda. Pozwala to routerom rozpoznawać drogę przepływu danych i upewnić się, czy każdy router jest przygotowany do zarezerwowania niezbędnych zasobów.

Jeżeli potencjalni odbiorcy chcą przyjąć dane, to po otrzymaniu komendy PATH wysyłają komendę RESV. Komendy RESV, używające tego samego identyfikatora przepływu, przechodzą zwrotną trasę ustaloną przez komunikat PATH i informują o oczekiwanym poziomie QoS. Router, który otrzyma komunikaty RESV od wielu użytkowników, będzie je scalał w pojedynczą komendę.

Jeżeli router może zapewnić wymagany poziom QoS, to zapisze rezerwację do tablicy przepływu. W przeciwnym wypadku router odrzuci rezerwację.



Rys. 1. Rezerwacja zasobów w sieci dostępowej1Mb/s
Fig. 1. Reservation of resources in 1Mbps access network

Na rys. 1 przedstawiono przykładowe wykorzystanie protokołu RSVP do rezerwacji zasobów w sieci dostępowej 1Mb/s. Rezerwacji RSVP podlega jedynie 0,5Mb/s. Pozostała część pasma nie podlega kontroli RSVP. Z analizowanych strumieni danych rezerwacji podlega jedynie strumień 2. Przy dużym ruchu generowanym przez strumienie sieci (w

warunkach limitowanych zasobów) rezerwacja pasma dokonywana dla strumienia 2 wpływa na pozostałe strumienie. Jak widać, strumień 1 i strumień 3 wykorzystują całe dostępne pasmo, transmisyjne – suma wszystkich strumieni ma stałą wartość 1Mb/s. Zaletą stosowania RSVP jest umożliwienie innym strumieniom wykorzystania pozostałego pasma aktualnie nie wykorzystywanego. Dzięki temu transmisje nie mające dużych wymagań QoS mogą wykorzystać wszystkie wolne zasoby nie zakłócając strumieni podlegających rezerwacji.

4. Transmisja multimedialna w sieciach dostępowych o różnych przepustowościach

W przypadku realizowania systemów konferencyjnych istnieje konieczność dostarczania odpowiednich strumieni danych (audio, wideo) do wielu odbiorców. Odbiorcy mogą znajdować się w wielu sieciach jednostkowych i posiadać różnego typu sieci dostępowe. Powoduje to wpływ indywidualnych własności sieci użytkownika na całą transmisję. Aby umożliwić udział w konferencji wszystkim użytkownikom, można dostosować prędkość transmisji do najwolniejszego z nich lub wykorzystać obecne w protokole RTP miksery i translatory.

Translator lub mikser RTP łączy dwie lub więcej domen poziomu transportowego. Translator przetwarza każdy ze strumieni wejściowych na odpowiadający mu strumień wyjściowy. Mikser łączy strumienie wejściowe w jeden nowy strumień.

Translator przesyła pakiety RTP z ich nienaruszonym identyfikatorem źródła (SSRC). Najczęściej translatory zmieniają rodzaj kodowania lub parametry kodowania danego strumienia. Pozwala to na zmianę prędkości bitowej obserwowanej na wyjściu translatora. Czasem stosowane są translatory nie zmieniające transmitowanego strumienia. Z punktu widzenia odbiorcy translator jest niewidoczny.

Mikser odbiera strumienie pakietów danych RTP z jednego lub więcej źródeł (czasami zmieniając format danych), łączy strumienie według ustalonej reguły, a następnie połączone strumienie przesyła dalej. Ponieważ strumienie mogą pochodzić od wielu niezsynchronizowanych źródeł, mikser musi wygenerować nowy sygnał synchronizujący dla połączonego strumienia. Dzięki temu wszystkie pakiety danych przesłane przez mikser będą wyznaczone z własnym identyfikatorem SSRC miksera. Aby umożliwić identyfikację oryginalnych źródeł sygnału składających się na wynikowy strumień, mikser powinien umieszczać ich identyfikatory SSRC w wykazie identyfikatorów CSRC.

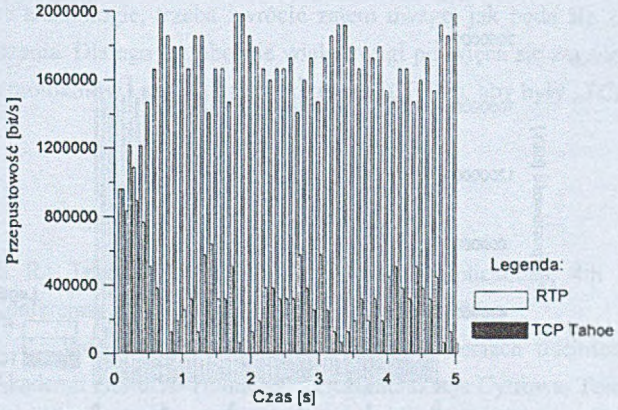
Miksery często stosowane są przy konferencjach audio i wideo. Przykładem może być mikser audio łączący poszczególne strumienie w jeden wynikowy.

Zastosowanie translatorów i mikserów pozwala na skalowanie transmisji multimedialnej do wydajności sieci dostępowych, umożliwiając wszystkim uzyskanie jakości prezentacji stosowanie do możliwości ich sieci.

5. Analiza transmisji wielu strumieni w sieci dostępowej

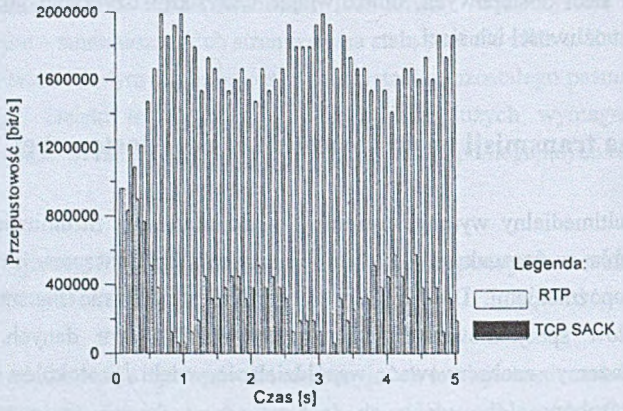
Przekaz multimedialny wymaga transmisji wielu strumieni. Strumienie te mogą mieć różne wymagania w stosunku do sieci transmisyjnej, np. związane z niezawodnością przekazu czy opóźnieniami. Dlatego, aby optymalnie zrealizować transmisję, stosuje się wiele protokołów specjalizowanych do określonego przekazu danych. Stąd w sieci dostępowej możemy zaobserwować współdziałanie wielu protokołów w warunkach limitowanych zasobów.

Protokół TCP posiada wiele mechanizmów zapobiegających przeciążeniom sieci. Mechanizmy te są niezbędne do prawidłowego funkcjonowania sieci. Transmisja danych za pomocą nowych protokołów, zwłaszcza wykorzystujących UDP, do sieci z protokołem TCP powoduje znaczną degradację jakości połączenia realizowanej przez protokół TCP. Zazwyczaj obserwuje się znaczące ograniczenie wydajności protokołu TCP.



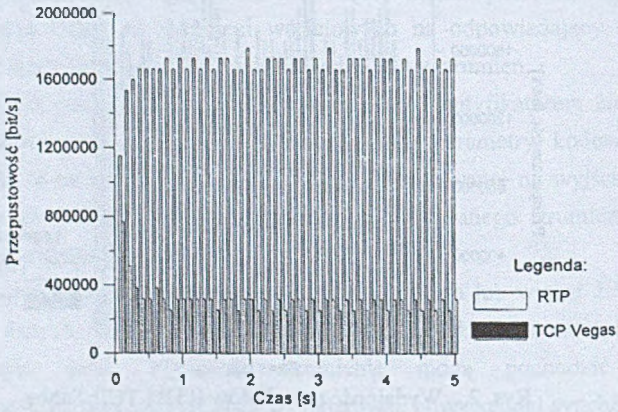
Rys. 2. Wydajność protokołów RTP i TCP Tahoe
Fig. 2. Throughput of RTP and TCP Tahoe protocols

Przedstawione badania zostały przeprowadzone w sieci dostępowej o przepustowości łącza 2Mb/s. W sieci tej transmitowano strumień RTP oraz klasyczny strumień TCP. Na rys. 2 przedstawiono współpracę protokołu TCP Tahoe z protokołem RTP. Widać, że protokół TCP zachowuje się bardzo niestabilnie. Wpływa to także na sumaryczną przepustowość sieci. Sieć nie jest w pełni wykorzystana.



Rys. 3. Wydajność protokołów RTP i TCP SACK
Fig. 3. Throughput of RTP and TCP SACK protocols

Podobnie jest w przypadku wersji TCP SACK (rys. 3) obserwujemy dużą niestabilnos sieci. Osiągana przepustowość jest jednak znacznie większa niż w przypadku TCP Tahoe.



Rys. 4. Chwilowa wydajność protokołów RTP i TCP Vegas
Fig. 4. Throughput of RTP and TCP Vegas protocols

Zupełnie inaczej zachowuje się wersja TCP Vegas (rys. 4). Obserwujemy dużą stabilność sieci. Wynika to m.in. ze sposobu estymacji pasma zastosowanej w tym protokole.

Obserwowany problem ilustruje zjawisko znane przy współpracy protokołu TCP z protokołem UDP. Powyższe zachowanie się protokołu RTP wynika m.in. z korzystania przez protokół RTP z protokołu UDP. Aby zjawisko znacznego stłumienia strumienia TCP nie

występowało, można: dokonać niezbędnych rezerwacji dla strumieni, wbudować mechanizmy kontroli przepływu w aplikacje lub też wykorzystać bardziej „odporne" wersje protokołu TCP, jak chociażby TCP Vegas.

6. Podsumowanie

W artykule przedstawiono, w jaki sposób aplikacje multimedialne współpracują ze współczesnymi sieciami. Przeanalizowano zasady współpracy aplikacji z sieciami dostępowymi o limitowanych zasobach. W sieciach tych, aby zapewnić niezbędną jakość usług, należy wykorzystać mechanizmy protokołu RTP – translatory i miksery. Pozwalają one na dostosowywanie prędkości do parametrów danej sieci dostępowej. W wielu przypadkach trzeba również stosować rezerwację zasobów, np. za pomocą protokołu RSVP. Najlepszy efekt możemy uzyskać łącząc obydwa rozwiązania.

Wprowadzenie transmisji multimedialnej (korzystającej z protokołu UDP lub RTP) objawia się również degradacją jakości połączeń TCP. Strumień multimedialny bez kontroli przepływu w skuteczny sposób może stłumić transmisję TCP (często w o wiele większym stopniu niż wynikałoby to z zapotrzebowania na pasmo przez protokół multimedialny). Wprowadzając nowe technologie, trzeba zwrócić zatem uwagę, jak będą się zachowywały już istniejące rozwiązania. Dlatego też obecnie wiele uwagi poświęca się zagadnieniu dobrej współpracy nowych protokołów i aplikacji z protokołem TCP (tzn. aby były „*TCP friendly*").

LITERATURA

1. Chodorek R. R.: Transport Protocols for Modern Applications, 4th International Distributed Conference, IDC'99, Madryt, 22-23 września 1999.
2. Chodorek R. R.: Protokoły transportowe w szybkich sieciach teleinformatycznych. Kwartalnik Akademii Górniczo-Hutniczej: Telekomunikacja Cyfrowa, Tom 1, zeszyt 4, 1998.
3. Deering S., Hinden R.: Internet Protocol, Version 6 (IPv6) Specification. RFC 2460. December 1998
4. Mathis M., Mahdavi J., Floyd S., Romanow A.: TCP Selective Acknowledgement Options. RFC 2018, October 1996.
5. Postel J.: Internet Protocol. RFC 791, USC/Information Sciences Institute, Sepetember 1981.
6. Postel J.: Transmission Control Protocol. RFC 793, USC/Information Sciences Institute, Sepetember 1981.

7. Postel J.: User Datagram Protocol. RFC 768, USC/Information Sciences Institute Sepetember 1981.
8. Schulzrinne H., Casner S., Frederick R., Jacobson V.: RTP: A Transport Protocol for Real-Time Applications. RFC 1889. January 1996.
9. Stevens W.: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001. January 1997.

Recenzent: Dr inż. Bartłomiej Zieliński

Wpłynęło do Redakcji 12 kwietnia 2000 r.

Abstract

The multimedia network application composes many different types of data (e.g. HTML text, binary files, graphics, images, audio and video). For each type of data we use specific method for compression and coding. Therefore, the transmission is observed as a multiplexed continuous data stream and each of stream has its own QoS characteristics (timing, bandwidth, etc.). During the transfer of different streams of data we can use one, universal transport protocol or we can select a special protocol for stream with specific characteristics.

The modern networks integrate many various applications. Some of them required reliable data transmission; the others provide real time transmission. For reliable transmission in the Internet we often use the multi-purpose transport protocol - TCP, while typical real-time multimedia transfer utilize specialized protocol - RTP.

In the paper we describe problem of transmission multimedia stream with limited capacity of access network. To provide efficient transmission to different access network we must use RTP mixers and translators. The RTP translators provide necessary changes to stream e.g. change the encoding of the data to new quality.

Another problem is reservation of resources for multimedia transmission. We analyze RSVP protocol and its possibility of resources allocation (fig.1).

The problem of transport protocol coexistence is well known (everybody knows, UDP is not familiar to TCP), but is not well identified or well solved. In ours experiments we use two various data stream. The first one is reliable data stream transmitting by TCP protocols. The second one is real-time stream transmitting by RTP protocol. RTP protocol can utilize all of bandwidth used by TCP. We analyze how congestion control mechanism implemented in different versions of TCP protocol influence to better transmission of TCP stream (fig. 2,3,4).