ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996

Seria: INFORMATYKA z.30 Nr. kol. 1315

Robert CHODOREK

PEWNE ASPEKTY ZARZĄDZANIA POŁĄCZENIAMI W PROTOKOLE XTP

Streszczenie. W artykule przedstawiono wybrane zagadnienia zarządzania połączeniami w protokole transportowym XTP. Przedstawiono mechanizmy ustanawiania połączenia, podstawy pracy grupowej w protokole XTP oraz zarządzanie przepływem danych.

SOME ASPECTS OF CONNECTION MANAGEMENT IN XTP PROTOCOL

Summary. In this paper some aspects of connection management in XTP protocol are presented. We present a multicast management, rate control and connection setup.

ПРОБЛЕМЫ УПРАВЛЕНИЯ В ПРОТОКОЛЕ XTP

Резюме. В работе представлено проблемы управления в протоколе XTP. Представлено тоже групповый порядок работы, уравление течения и установления связи.

Pracę wykonano w ramach projektu badawczego nr 3 P406 027 06 finansowanego przez Komitet Badań Naukowych w roku 1994/95.

1. Wprowadzenie

Protokół XTP (*Xpress Transport Protocol*) [2] jest odpowiedzią na zapotrzebowanie ze strony aplikacji multimedialnych systemów czasu rzeczywistego oraz systemów militarnych, wymagających specjalnych usług transmisyjnych[1], o odpowiednim poziomie bezpieczeństwa i prędkości transmisji rzędu gigabitów [2][7][9]. Prace nad protokołem 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.

Aż do wersji 3.6 włącznie protokół XTP łączył w sobie funkcje, jakie spełnia zespół protokołów TCP/IP (funkcje warstwy transportowej i części warstwy sieciowej), tworząc w ten sposób jedną wspólną "warstwę transferu" [10]. Nosił on wówczas nazwę *Xpress Transfer Protocol*. W późniejszych wersjach protokołu XTP dokonano szeregu istotnych zmian, eliminując między innymi routing. Najnowsza, datowana na marzec 1995 r., wersja protokołu XTP (4.0) [8] nosi w związku z tym nazwę *Xpress Transport Protocol*.

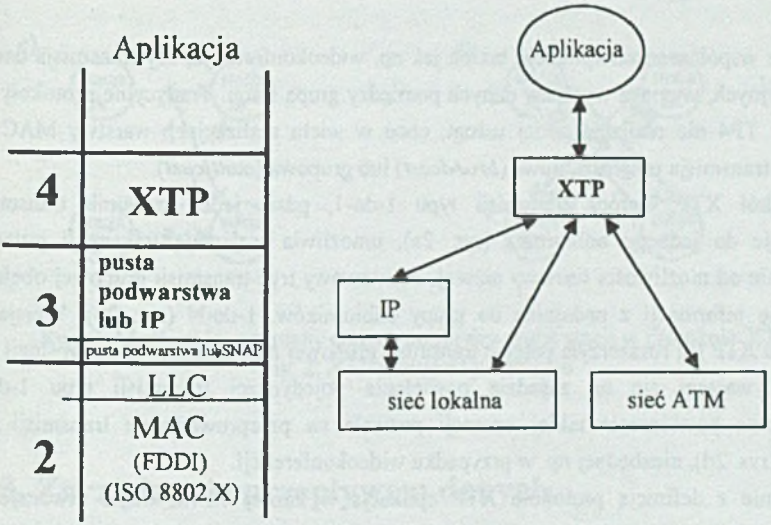
Protokół transportowy XTP został zaprojektowany pod kątem efektywnej realizacji nowych zadań stawianych systemom sieciowym, ze znaczną redukcją operacji przetwarzania protokołowego i łatwą implementacją z użyciem układów VLSI. Ponadto charakteryzują go między innymi: kompatybilność z dotychczasowymi standardami adresowania (schemat adresacji jak w TCP czy OSI/TP4), adresowanie z wykorzystaniem funkcji indeksowej, komunikaty priorytetowe (istotne dla systemów czasu rzeczywistego), mechanizmy wielodostępu, niezależne mechanizmy kontroli przepływu i prędkości, selektywne retransmisje i potwierdzenia, stała długość nagłówka pakietu, zmienny rozmiar ramki z przypisaniem na granicy 64 bitów, możliwość negocjacji parametrów ruchu i jakości świadczonych usług, możliwość włączenia dowolnego spośród mechanizmów kontroli przepływu, prędkości czy korekcji błędów.

Pakiety w protokole XTP składają się z nagłówka o stałej długości 32 bajtów i segmentu informacyjnego o zmiennej długości. Ze względu na informacje zawarte w segmencie informacyjnym pakiety dzielimy na: pakiety kontrolne i pakiety danych. Należą one do jednego z siedmiu typów pakietów zdefiniowanych w specyfikacji XTP: FIRST - pakiet inicjujący połączenie; może także przenosić dane użytkownika, DATA - pakiet przenoszący dane użytkowników, CNTL - pakiet kontrolny, przenoszący informacje o stanie połączenia,

[1] Na przykład połączenie samolotu C-130 jednocześnie z 22 superkomputerami Cray. [9]

ECNTL - pakiet kontrolny, przenoszący informacje o wystąpieniu błędu oraz o stanie połączenia, TCNTL - pakiet kontrolny, służący do negocjacji strumienia ruchu i jakości świadczonych usług oraz przenoszący informacje o stanie połączenia, JOIN - pakiet zbliżony do pakietu FIRST, służący do dołączenia się do trwającej konwersacji grupowej, DIAG - pakiet diagnostyczny. W pakietach CNTL, ECNTL i TCNTL przesyłany jest segment kontrolny, zawierający dane o kontekście. Pakiety FIRST, DATA, DIAG i JOIN zawierają segment informacyjny. Segment informacyjny może zawierać dane wyższych warstw (pakiety FIRST, DATA) lub wiadomości warstwy transportowej (pozostałe pakiety).

Protokół XTP może współpracować bezpośrednio z wieloma standardami niższych warstw sieciowych. W sieciach LAN, bezpośrednio transmitując do warstwy LLC lub MAC, obejmującej Ethernet, FDDI lub protokoły określone w normie ISO 8802.x., może współpracować z protokołem IP (pozwala to wykorzystać protokół XTP w sieci Internet) lub bezpośrednio z ALL 5 ATM [1][6][8] (rys. 1).



Rys. 1. Współpraca protokołu XTP z siecią LAN (opartą na FDDI lub protokołach określonych w normach ISO 8802.X) oraz z siecią ATM
Fig. 1. Interaction of XTP and IP protocols and underlying networks

Duża elastyczność protokołu osiągnięta została dzięki zdefiniowaniu wielu mechanizmów protokołu, mogących pracować w różnych środowiskach sieciowych. Do podstawowych mechanizmów protokołu XTP zaliczamy: mechanizmy zarządzania

kontekstami[2)] i połączeniami, mechanizmy zarządzania przepływem danych, mechanizmy adresowania grupowego.

Dla każdej z powyższych grup zostały określone procedury definiujące operacje realizowane w protokole XTP. Mechanizmy zarządzania kontekstami i połączeniami oparte są na procedurach realizujących: otwarcie i zamknięcie kontekstu oraz negocjację parametrów określających prędkości pracy nadajnika i odbiornika. Procedury zarządzające przepływem danych wykonują operacje: kontroli przepływu, kontroli prędkości, kontroli błędów. Adresowanie grupowe określone jest przez charakterystyczne dla tego trybu: zarządzanie kontekstem w trybie adresowania grupowego, kontrolę przepływu i błędów w trybie adresowania grupowego, włączanie się do trwającej konwersacji.

2. Transmisja grupowa

Wiele współczesnych aplikacji, takich jak np. wideokonferencja, czy transmisja danych nawigacyjnych, wymaga wymiany danych pomiędzy grupą stacji. Tradycyjne protokoły, jak TCP czy TP4 nie realizują takiej usługi, choć w wielu realizacjach warstwy MAC jest możliwa transmisja rozgłoszeniowa (*broadcast*) lub grupowa (*multicast*).

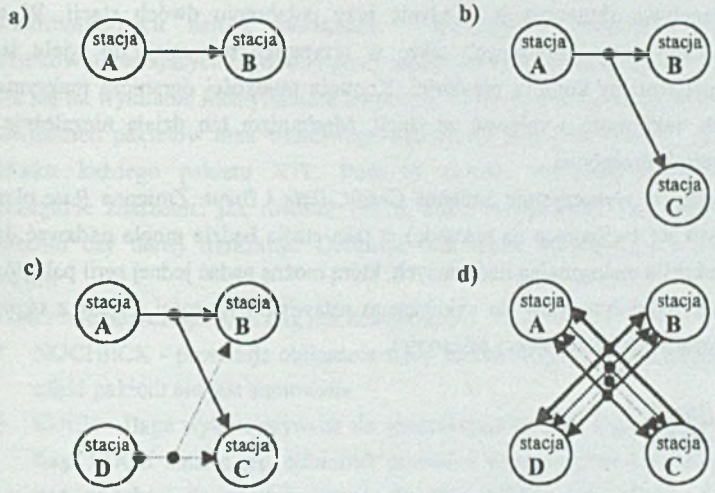
Protokół XTP, oprócz transmisji typu 1-do-1, gdzie jeden nadajnik transmituje informacje do jednego odbiornika (rys. 2a), umożliwia realizację transmisji grupowej niezależnie od możliwości warstwy niższej. Podstawowy tryb transmisji grupowej obejmuje transmisję informacji z nadajnika do grupy odbiorników, 1-do-N (rys. 2b). Wersja 4.0 protokołu XTP [8] rozszerzyła pojęcie transmisji grupowej na transmisję N-do-M stacji (rys. 2c), odbywającej się na zasadzie powielenia pojedynczej transmisji typu 1-do-N. Wielokrotne powtórzenie takiej operacji pozwala na przeprowadzenie transmisji typu N-do-N (rys. 2d), niezbędnej np. w przypadku wideokonferencji.

Zgodnie z definicją protokołu XTP aplikacja w każdej chwili może utworzyć lub dołączyć się do dowolnej liczby grup. Aplikacja transmitująca dane zarządza grupą za pośrednictwem managera grupy (*Multicast Group Manager, MGM*), którego zadanie sprowadza się do kontroli trzech podstawowych zagadnień:

- przyjmowania do grupy nowych stacji,
- usuwania z grupy stacji,
- zapewnienia integralności grupy w trakcie pracy.

[2)]Kontekst - informacja o stanie połączenia, zawierająca zmienne służące do zarządzania wejściowym i wyjściowym strumieniem danych, przechowywana w danym systemie końcowym.

Pomimo iż skład grupy może się zmienić w dowolnej chwili czasu, aplikacja transmitująca dane jest na bieżąco informowana o każdej zmianie. Reakcja protokołu na usunięcie stacji (z powodu błędów transmisji lub dobrowolnego opuszczenia grupy) nie jest określona z góry, lecz definiuje ją użytkownik. W zależności od tego, czy usunięta stacja powinna mieć zapewnioną niezawodną transmisję, czy też nie jest to wymagane, aplikacja zarządzająca grupą może całkowicie przerwać transmisję, kontynuować transmisję lub podjąć dowolne inne działanie.



Rys. 2. Możliwości wymiany danych pomiędzy grupą stacji w protokole XTP
Fig. 2. XTP multicast paradigms

3. Zarządzanie przepływem danych

W szybkich sieciach LAN przesyłanie dużego strumienia danych do odbiornika może spowodować, że nie będzie on w stanie ich odebrać. Taka sytuacja może być przyczyną przepełniania buforów nadawcy lub gubienia ramek. Przeciwdziałać tym zjawiskom mają mechanizmy kontroli przepływu i prędkości.

Strumień danych przesyłanych pomiędzy dwoma stacjami jest sterowany za pomocą mechanizmu sterowania przepływem [3]. Mechanizm ten, na żądanie aplikacji, można wyłączyć. W protokole XTP sterowanie przepływem oparte jest na mechanizmie

przesuwnego okna (*sliding window*). Początkowe parametry okna ustawiane są podczas nawiązywania połączenia.

Aby nadajnik mógł przesyłać dane, musi sukcesywnie otrzymywać nowy kredyt. Dlatego też kluczową sprawą w sterowaniu przepływem jest procedura przydziału nowego kredytu. Nowy kredyt przesyłany jest przez odbiornik w pakietach kontrolnych (CNTL, ECNTL, TCNTL). 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 stacji. W celu zapewnienia mechanizmów kontrolnych także w przypadku transmisji do wielu stacji wprowadzono mechanizmy kontroli prędkości. Kontrola prędkości ogranicza maksymalny strumień danych, jaki może wypływać ze stacji. Mechanizm ten działa niezależnie od mechanizmu kontroli przepływu.

Kontrola prędkości wykorzystuje zmienne *Credit*, *Rate* i *Burst*. Zmienna *Rate* określa maksymalną szybkość (w bajtach na sekundę), z jaką stacja będzie mogła nadawać dane. Zmienna *Burst* określa maksymalną ilość danych, którą można nadać jednej serii pakietów.

RTIMER jest wykorzystywany do cyklicznego ustawiania wartości *Credit* z okresem *RTimer*, wyznaczanym z następującej zależności:

$$RTimer = \frac{Burst}{Rate} \tag{1}$$

Wartość zmiennej *Credit* w chwili początkowej jest równa *Burst*. Każda transmisja powoduje zmniejszanie wartości *Credit* o ilość wysłanych bajtów. Jeżeli zmienna *Credit* osiągnie wartość równą zero lub ujemną, przesyłanie danych zostaje wstrzymane, niezależnie od aktualnego stanu mechanizmu kontroli przepływu. Zmienna *Credit* jest na nowo ustawiana przez wartość *Burst* przed upływem czasu *RTimer*. W C++ procedura ta ma następującą postać:

```
if      (Credit <= 0)
        Credit = Credit + Burst;
else
        Credit = Burst
```

Wartości zmiennych *Rate* i *Burst* ustawiane są po odebraniu pakietów FIRST, JOIN i TCNTL. W tych pakietach przesyłane są bowiem informacje dotyczące 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.

4. Zarządzanie kontekstami i połączeniami

Do mechanizmów służących do zarządzania połączeniami w sieciach teleinformatycznych należą nawiązanie i rozwiązanie połączenia oraz negocjacja parametrów określających prędkości pracy nadajnika i odbiornika. Zarządzanie kontekstami opiera się na wymianie identyfikatora kontekstu. Wszystkie te operacje wymagają wymiany odpowiednich pakietów oraz właściwego ustawienia flag pola *options*, definiowanego w nagłówku każdego pakietu XTP. Pole to określa właściwą reakcję protokołu na poszczególne zdarzenia, jak również ustala, które mechanizmy będą aktywne w danym połączeniu czy danej transmisji. Definiuje ono także mechanizmy i tryby operacyjne protokołu.

Znaczenie poszczególnych flag jest następujące:

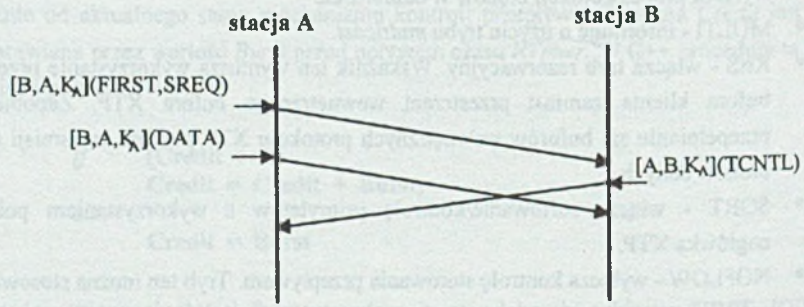
- NOCHECK - powoduje obliczanie sumy kontrolnej jedynie dla nagłówka, pozostała część pakietu nie jest sumowana.
- EDGE - flaga wykorzystywana do generowania pakietu kontrolnego. Jeżeli wartość flagi EDGE zmieni się, odbiornik powinien wysłać pakiet kontrolny. Działanie to jest podobne do występującego dla flagi SREQ lub DREQ, nie jest jednak zabezpieczone mechanizmem czasowym wykorzystującym WTIMER.
- NOERR - informuje odbiornik, że nadajnik nie będzie retransmitował informacji i wyłącza proces korekcji błędów w odbiorniku.
- MULTI - informuje o użyciu trybu *multicast*.
- RES - włącza tryb rezerwacyjny. Wskaźnik ten wymusza wykorzystanie przestrzeni bufora klienta zamiast przestrzeni wewnętrznego bufora XTP. Zapobiega to przepełnianiu się buforów wewnętrznych protokołu XTP podczas transmisji dużych bloków danych.
- SORT - włącza sortowanie/kontrolę priorytetów z wykorzystaniem pola sort nagłówka XTP.
- NOFLOW - wyłącza kontrolę sterowania przepływem. Tryb ten można stosować, gdy aplikacja posiada własne mechanizmy kontroli przepływu lub możliwość pracy w "czasie rzeczywistym" protokołu.
- FASTNAK - włącza mechanizm szybkiego negatywnego potwierdzenia. Jeżeli odbiornik wykryje brak pakietu, musi natychmiast wysłać pakiet ECNTL.

- SREQ - żądanie natychmiastowego wysłania pakietu kontrolnego przez odbiornik.
- DREQ - żądanie wysłania pakietu kontrolnego przez odbiornik po dostarczaniu wszystkich odebranych danych (włączając w to aktualny pakiet) z kolejki wewnętrznej XTP do aplikacji.
- WCLOSE - sygnalizuje zamknięcie nadajnika.
- RCLOSE - sygnalizuje zamknięcie odbiornika.
- EOM - oznacza koniec wiadomości.
- END - sygnalizuje usunięcie danego kontekstu (koniec połączenia).
- BTAG - informuje o przesłaniu danych ważnych dla aplikacji.

4.1. Nawiązanie połączenia w protokole XTP

W protokole XTP nadawanie lub odbieranie pakietów jest możliwe, jeżeli w stacji powstanie aktywny kontekst XTP. Połączenie pomiędzy kontekstami jest tworzone poprzez przesłanie pakietu FIRST od jednego aktywnego kontekstu do innego oczekującego kontekstu. Po odebraniu pakietu FIRST kontekst odbiornika przechodzi ze stanu oczekiwania do stanu aktywnego.

Przykładowy scenariusz nawiązania połączenia pomiędzy kontekstami przedstawiono na rys. 3. Stacja A wysyła do stacji B pakiet FIRST, zawierający żądanie (ustawiona flaga SREQ) natychmiastowego wysłania pakietu kontrolnego przez stację B. Stacja B odpowiada, wysyłając pakiet kontrolny TCNTL. Stacja A nie musi oczekiwać na przybycie pakietu TCNTL - może od razu nadawać dane używając pakietów DATA.

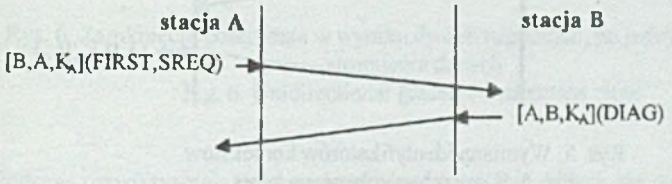


Rys. 3. Nawiązanie połączenia pomiędzy stacjami i przesyłanie danych[3)]
Fig. 3. Connection setup

[3)] Opis mechanizmów protokołu XTP jest zgodny z notacją zamieszczoną w [10].

Pakiet FIRST zawiera także informacje o podstawowych parametrach transmisji: wielkość strumienia wyjściowego i wejściowego oraz informacje umożliwiające lepsze dostosowanie się do danej implementacji warstw niższych poprzez negocjowanie wartości MTU (*Maximum Transmission Unit*). Systemy końcowe i routery deklarują własną wielkość MTU, a nadajnik wybiera najmniejszą z nich. Zapobiega to segmentacji pakietów w niższych warstwach sieci.

Pakiet FIRST przenosi informację o parametrach prędkości strumienia wejściowego i wyjściowego. Odbiornik może te parametry odrzucić, przyjąć lub przyjąć z modyfikacjami. Odrzucenie parametrów sygnalizowane jest przez pakiet DIAG. Sytuację taką przedstawiono na rys. 4. Pakiet FIRST przybywa do stacji B i zostaje odrzucony. Do stacji A zostaje wysłany pakiet DIAG z informacją o przyczynie odrzucenia pakietu FIRST. Odrzucenie pakietu może być spowodowane przez zbyt duże wymagania transmisyjne postawione przez nadawcę, brak kontekstu oczekującego na połączenie, zły adres lub z innych przyczyn.



Rys. 4. Odrzucenie pakietu FIRST przez stację B z wysłaniem pakietu diagnostycznego DIAG
Fig. 4. Reject FIRST packet with a DIAG message

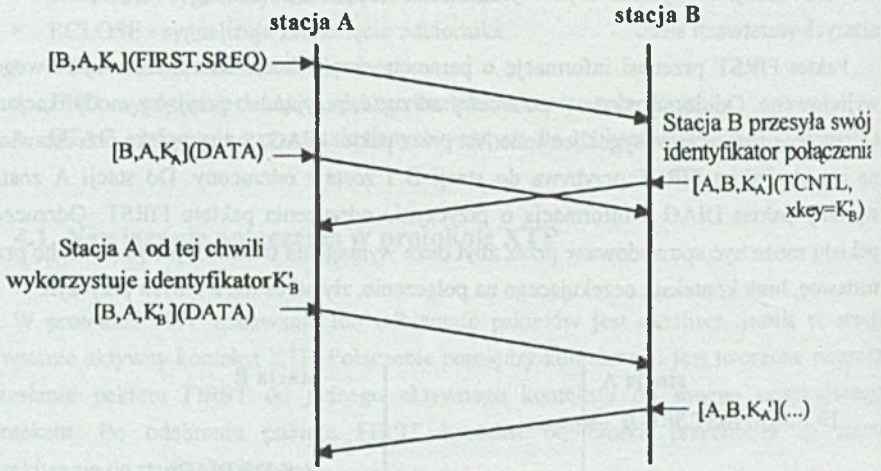
4.2. Wymiana wartości *key*

Wymiana pakietów wymaga zidentyfikowania właściwego kontekstu w danej stacji. W sytuacji przedstawionej na rys. 3 stacja B przesyła do stacji A pakiet [A,B,K$_A$'](TCNTL), gdzie wartość K$_A$' jest identyfikatorem powrotnym kontekstu[4]. Aby w stacji A zidentyfikować kontekst, wystarczy znać wartość K$_A$'. Zwiększa to efektywność znalezienia kontekstu.

W stacji B do identyfikacji kontekstu konieczna jest analiza zmiennych adresu stacji A oraz pola *key*. Jest to operacja wymagająca więcej czasu niż podczas transmisji w drugą stronę. Aby przesyłanie danych mogło odbywać się z wykorzystaniem tych samych mechanizmów,

[4]Identyfikator powrotny - identyfikator z ustawionym najbardziej znaczącym bitem.

pomiędzy stacjami prowadzącymi wymianę danych istnieje możliwość wymiany identyfikatorów kontekstów. Na rys. 5 przedstawiono wymianę identyfikatorów kontekstów. Stacja B przesyłając pakiet TCNTL przesyła do stacji A identyfikator powrotny własnego kontekstu K_B'. Ten identyfikator stacja A wykorzystuje w dalszych transmisjach.

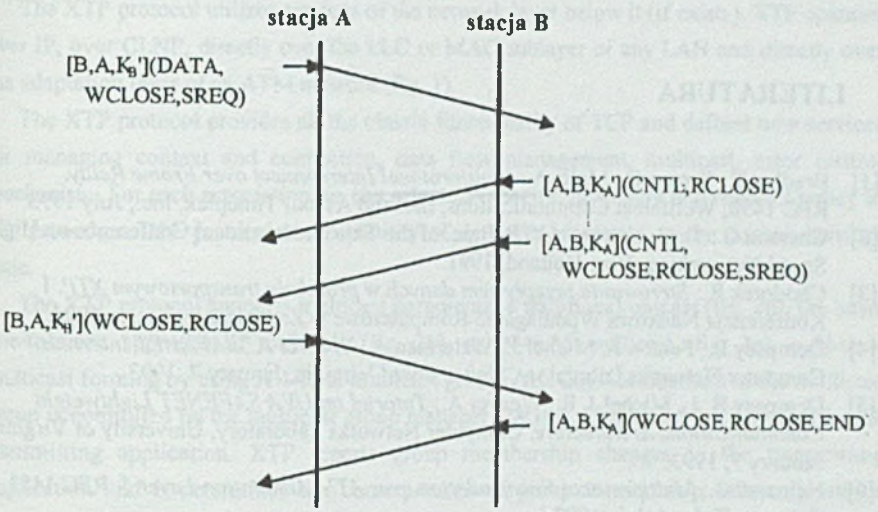


Rys. 5. Wymiana identyfikatorów kontekstów
Fig. 5. Key exchange between hosts

4.3. Zamknięcie połączenia

Protokół XTP dla każdego połączenia tworzy parę jednokierunkowych strumieni danych. Aby połączenie zostało zakończone, zamknięty zostać musi każdy strumień danych. Zamykanie połączenia następuje po całkowitej wymianie informacji z ustawionymi bitami WCLOSE, RCLOSE, END. Rozwiązanie połączenia może być płynne lub wymuszone. Płynne zamknięcie strumienia danych występuje w sytuacji, gdy wszystkie dane zostały poprawnie przekazane i potwierdzone przez odbiornik. Następuje to w wyniku dwóch uzgodnień, po jednym dla każdego strumienia danych (rys. 6).

Wymuszone zamknięcie połączenia występuje w sytuacji nagłego przerwania obu strumieni danych. Tego typu zamknięcie nie gwarantuje dostarczenia wszystkich danych do aplikacji. Wymuszone zamknięcie połączenia może również wystąpić na żądanie użytkownika i niekoniecznie musi oznaczać sytuację awaryjną.



Rys. 6. Zamknięcie połączenia w wyniku dwóch uzgodnień, po jednym dla każdego strumienia danych
Fig. 6. Unidirectional graceful connection close

Podczas rozwiązywania połączenia obydwa strumienie danych nie muszą być zamknięte w ten sam sposób. Jeden z nich może zostać zamknięty łagodnie, a drugi przerwany nagle. Jest to tzw. skrócone łagodne rozwiązanie połączenia. Połączenie można przerwać także ustawiając w pakiecie flagę END, co spowoduje natychmiastowe zamknięcie połączenia.

5. Zakończenie

Protokół XTP może efektywnie współpracować z nowoczesnymi sieciami teleinformatycznymi. Posiada możliwość pracy grupowej, ważnej dla wielu współczesnych aplikacji. Mechanizmy zarządzania kontekstami i połączeniami oraz mechanizmy zarządzania przepływem danych zapewniają dużą elastyczność protokołu, wspomagając jednocześnie paradygmat pracy grupowej.

XTP jest protokołem transportowym popieranym przez grupę dużych firm i uczelni zrzeszonych w XTP Forum. Jako część standardu MIL-STD-2204, jest on zalecany przez armię Stanów Zjednoczonych dla systemów komunikacyjnych o dużym poziomie bezpieczeństwa [4][5]. Jest on poważnym konkurentem protokołu TCP.

LITERATURA

[1] Bradley T., Brown C., Malis A.: *Multiprotocol Interconnect over Frame Relay.* RFC 1490, Wellfeleet Communications, Inc. and Ascom Timeplex, Inc., July 1993.
[2] Chesson G.: *Te Evolution of XTP.* Proc. of the Third International Conference on High Speed Networking, Nort-Holland, 1991.
[3] Chodorek R.: *Sterowanie przepływem danych w protokole transportowym XTP.* I Konferencja Naukowa Wpomaganie Komputerowe w Zarządzaniu, Kielce 1994.
[4] Dempsey B., Fenton J., Michel J., Waterman A., Weaver A.: *SAFENET Internals.* Computer Networks Laboratory, University of Virginia, January 7, 1993.
[5] Dempsey B. J., Michel J. R., Weaver A.: *Tutorial on UVA SAFENET Lightweight Communications Architecture.* Computer Networks Laboratory, University of Virginia, January 7, 1993.
[6] Heinanen J.: *Multiprotocol Encapsulation over ATM Adaptation Layer 5.* RFC 1483, Telecom Finland, July 1993.
[7] Ransom M. Niel, Spears Dan R.: *Applications of Public Gigabit Networks.* IEEE Network Magazine, vol. 6, no. 2, pp. 30-40, March 1992.
[8] Strayer W. T. ed.: *Xpress Transport Protocol 4.0 Specification.* XTP Forum Inc., Santa Barbara, March 1995.
[9] Weaver Alfred C.: *Xpress Transport Protocol Version 4.* XTP Forum, 1995
[10] Protocol Engine: *XTP Protocol Definition. Revision 3.6.* Protocol Engines Incorporated, Santa Barbara, January 1992.

Recenzent: dr inż. Wojciech Mielczarek

Wpłynęło do Redakcji 4 stycznia 1996 r.

Abstract

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. The protocol is currently promoted by the XTP Forum, composed of representatives from industry, academia, military and US government.

The XTP protocol utilizes services of the network layer below it (if exists). XTP operates over IP, over CLNP, directly over the LLC or MAC sublayer of any LAN and directly over the adaptation layer of an ATM network (fig. 1).

The XTP protocol provides all the classic functionality of TCP and defines new services for managing context and connection, data flow management, multicast, error control mechanism. For each association we can select the specific procedure and relevant policy in this procedure. The protocol has capability of dynamic adaptation to the current network state.

The XTP protocol supports multicast paradigms: a traditional unicast (fig. 2a), the basic one-to-many approach of multicast (fig. 2b), an N-to-M multicast (fig. 2c), a N-to-N multicast forming by using N 1-to-N multicast groups (fig. 2d). Membership in the multicast group is controlled by the multicast group manager (MGM). The MGM is controlled by the transmitting application. XTP reports group membership changes to the transmitting application and it determines the consequences of group membership failure (continue transmission, abandon transmission, etc.).

The XTP protocol supports two types of the data flow management: the selectable flow control and the rate and burst control. The flow control mechanism is based on the sliding window scheme. User can also disable flow control entirely by using the *noflow* option; this mode is useful for multimedia applications.

The multicast data flow management is provided by the rate and burst control. The rate control parameter limits the amount of data that can be transmitted per unit time. The burst parameter limits the size of data that can be sent.

Connection management use three main procedures for: opening context (fig. 3, fig. 4), performing a key exchange (fig. 5) and closing context (fig. 6).