STUDIA INFORMATICA 2001
Volume 22 Number 2 (44)

Krzysztof GROCHLA
Instytut Informatyki Teoretycznej i Stosowanej PAN

GENERATOR PAKIETÓW W ŚRODOWISKU SIECI HETEROGENICZNYCH

Streszczenie. Artykuł zawiera opis programowego generatora pakietów w sieciach komputerowych, działającego w środowisku heterogenicznym, w połączeniu z programem odbiorczym umożliwiającym pomiar statystyk jakości usług sieciowych. Zaprezentowano także porównanie z konkurencyjnymi rozwiązaniami oraz możliwe zastosowania powstałego narzędzia.

PACKET GENERATOR IN HETEREGENOUS NETWORK ENVIRONMENT

Summary. The Article describes a software packet generator for computer networks. It is designed to work in heterogeneus network environment (both TCP/IP and ATM), and includes a recevier measuring traffic and QoS statistics. A short comparison to existing tools is presented, and possible uses are described.

1. Wprowadzenie

Sieci komputerowe znajdują obecnie coraz szersze zastosowanie. Ich ciągły rozwój prowadzi do wykorzystywania istniejącej infrastruktury do integracji różnych usług komunikacji w ramach pojedynczej sieci. Sieć komputerowa oferująca wiele różnych usług musi sprostać różnym wymaganiom, często wzajemnie się wykluczającym.

Często stajemy przed koniecznością zmierzenia parametrów istniejącej instalacji sieciowej, w celu odpowiedzenia na pytanie, czy spełni ona nasze wymagania co do jakości usług oraz określenia możliwości jej dalszej rozbudowy. Pomiar parametrów pracy działającej sieci staje się często sporym problemem ze względu na brak odpowiednich narzędzi. Na rynku dostępne są wyspecjalizowane analizatory sieciowe składające się z części

programowej oraz sprzętowej, jednak są to rozwiązania drogie i umożliwiające tylko pomiar już istniejącego ruchu. W pewnych wypadkach konieczne staje się wygenerowanie ruchu o określonej charakterystyce, co pozwala np. na sprawdzenie zachowania się sieci podczas określonego typu transmisji lub przetestowanie możliwości dalszej rozbudowy istniejącej instalacji sieciowej. Oprogramowanie może zostać także wykorzystane do weryfikacji wyników przeprowadzonych symulacji [7].

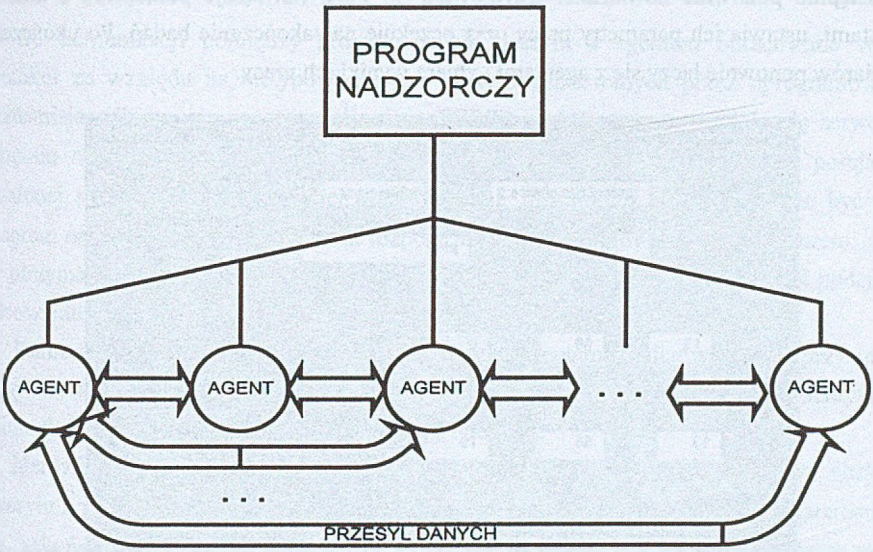
Powstałe w ramach pracy oprogramowanie ma za zadanie umożliwić sprawdzenie, czy istniejąca sieć komputerowa umożliwia przesyłanie ruchu o określonej przez użytkownika charakterystyce oraz pomiar jakości transmisji. W tym celu są realizowane dwie funkcje: generowanie transmisji oraz jej odbiór. Generacja pakietów odbywa się według parametrów ustalonych przez użytkownika. Dzięki wykorzystaniu protokołów IP oraz ATM możliwe jest zastosowanie narzędzia zarówno w sieciach lokalnych, jak i sieci Internet. Narzędzie umożliwia równoczesną generację ruchu w wielu kierunkach (ich liczba jest ustalana na etapie kompilacji) oraz pomiędzy dowolną liczbą punktów. Kierowanie wykonywaniem pomiarów jest możliwe z jednego miejsca za pomocą programu nadzorczego kontrolującego pracę programów agentów generujących lub odbierających ruch.

W oprogramowaniu wykorzystano 2 protokoły: ATM oraz UDP. Zadecydowała o tym ich powszechność oraz dostępność platformy sprzętowej w IITiS PAN. Obydwa protokoły są zawodne, tzn. nie retransmitują utraconych danych, dzięki czemu możliwy jest pomiar liczby utraconych pakietów. Wykorzystanie protokółów niezawodnych, np. TCP, uniemożliwiłoby ten pomiar i wprowadziłoby przekłamania wyników związane z retransmisją. Implementacja obsługi protokołu ATM została oparta na bibliotece sygnalizacyjnej opracowanej przez A. Sochana [1,6].

2. Opis powstałego oprogramowania

Narzędzie składa się z 2 modułów: programu nadzorczego umożliwiającego ustalenie parametrów transmisji i prezentującego wyniki pomiarów oraz programu agenta realizującego transmisje, mierzącego wyniki i przekazującego je do nadzorcy. Program agenta działa w systemie Solaris bądź innym opartym na UNIX'ie, program nadzorczy zaś pod kontrolą Windows NT/9x. Schematyczny podział zadań prezentuje rysunek 1.

Program nadzorczy jest uruchamiany tylko raz dla pojedynczej serii pomiarów. Jego działanie zostanie dokładnie omówione w dalszej części pracy. Program agenta oczekuje na parametry pracy przesłane przez program nadzorczy i następnie zgodnie z nimi generuje bądź odbiera dane. Każde wysłanie lub odebranie porcji danych jest rejestrowane w pliku z wynikami, który następnie może zostać przesłany do programu nadzorczego w celu dalszej



Rys. 1. Schemat struktury generatora
Fig. 1. Schematic of generator structure

analizy wyników. Na podstawie różnic pomiędzy czasem wysłania i odbioru oprogramowanie umożliwia obliczenie czasów opóźnień i liczbę utraconych danych.

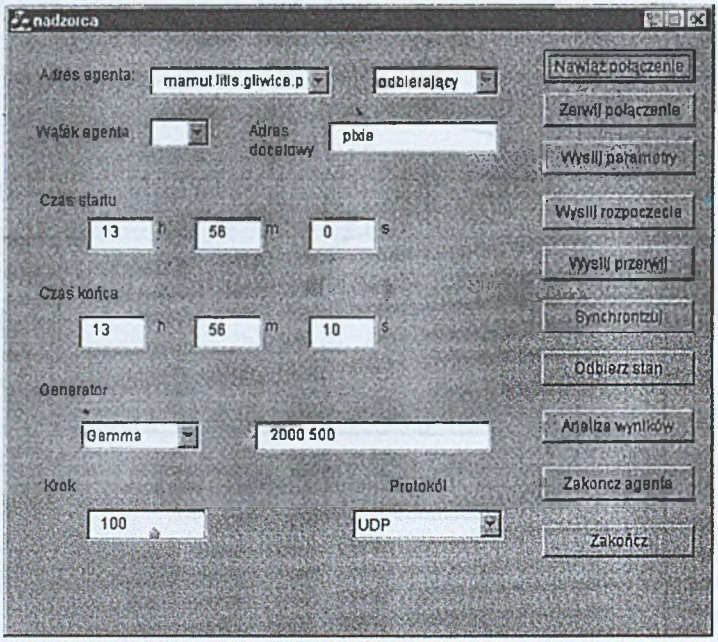
2.1. Program nadzorczy

Program nadzorczy działa w środowisku Windows NT/9x i został napisany przy użyciu pakietu Microsoft Visual C 6.0. Do jego zadań należy:

1) Nawiązanie połączenia z agentem
2) Przesłanie parametrów pracy – w tym
   a) tryb pracy (nadawca/odbiorca)
   b) czas startu, końca
   c) typ i parametry generatora
3) Zbieranie i wstępna analiza wyników
4) Zakończenie pracy agenta

Po rozpoczęciu pracy program oczekuje na wpisanie adresu agenta. Kilka najczęściej używanych adresów można wybrać z listy rozwijanej. Następnie po naciśnięciu przycisku „Nawiąż połączenie" z agentem zestawiane jest połączenie TCP. Po wpisaniu parametrów pracy są one przesyłane do agenta. Program umożliwia również zakończenie pracy agenta i przerwanie nadawania lub odbioru w dowolnym momencie. Połączenie może zostać zerwane,

a następnie ponownie nawiązane. Użytkownik po kolei nawiązuje połączenia z kilkoma agentami, ustawia ich parametry pracy oraz oczekuje na zakończenie badań. Po ukończeniu pomiarów ponownie łączy się z agentami i zbiera wyniki ich pracy.



Rys. 2. Wygląd głównego okna programu nadzorczego
Fig. 2. Main program window

Jeden z przycisków otwiera nowe okno dialogowe umożliwiające odbiór, wstępną analizę i eksport wyników. Można tutaj pobrać wyniki badań od agenta, z którym nawiązano połączenie, zapisać je do pliku oraz wyeksportować do pliku CSV. Jeżeli zostaną odebrane wyniki z obydwu stron transmisji, program umożliwia obliczenie minimalnego, średniego i maksymalnego opóźnienia. Wyniki zostają skorygowane o różnice czasów pomiędzy programem nadzorczym i agentem, dzięki czemu wszystkie pomiary zostają przeliczone na czas lokalny komputera, na którym pracuje program nadzorczy. Mechanizm synchronizacji czasu zostanie dokładniej omówiony w dalszej części pracy. Eksport wyników umożliwia przeprowadzenie dalszej analizy wyników w programach statystycznych, przyjęty format CSV jest odczytywany m.in. przez Statistic'e oraz MS Excel. Pozwala to na dowolne przetwarzanie wyników przez prowadzącego badania.

2.2. Komunikacja

Do komunikacji pomiędzy programem nadzorczym a agentem opracowano własny protokół ze względu na nietypowy zestaw funkcji realizowanych przez oprogramowanie. Transmisja odbywa się poprzez połączenie TCP/IP, w którym agent pełni rolę serwera, a program nadzorczy klienta. Wykorzystano port 6556. Dane są przesyłane w porcjach o ustalonej strukturze, zwanych dalej komunikatami. Inicjatorem transmisji może być tylko program nadzorczy. Każdy komunikat rozpoczyna się jednobajtowym identyfikatorem. Agent po otrzymaniu komunikatu rozpoznaje identyfikator i na podstawie jego wartości podejmuje odpowiednie działanie, np. rozpoczyna transmisję.

Komunikacja pomiędzy programem nadzorcy a agentami realizowana jest przy użyciu klasy *komunikat* oraz klas dziedziczących po niej. Klasa *komunikat* zawiera identyfikator komunikatu oraz dwie wirtualne metody *wyslij* i *odbierz*.

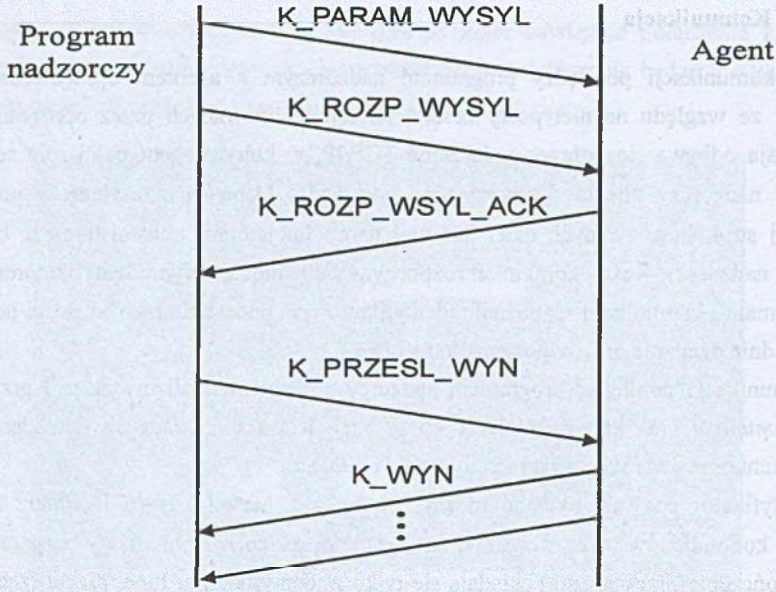
Identyfikator pozwala określić rodzaj komunikatu. Metody *wyslij* i *odbierz* służą do przesyłu komunikatów przez wcześniej nawiązane połączenie. Komunikaty bezparametrowe (np. zakończenie pracy agenta) składają się tylko z identyfikatora. Inne, przenoszące więcej informacji (np. parametry wysyłu), zostały zrealizowane jako osobne klasy dziedziczące po klasie *komunikat* rozszerzone o pola zawierające przesyłane parametry. W każdym wypadku metody *wyslij* i *odbierz* pozwalają na transmisje całego komunikatu, w klasach pochodnych są przeciążane.

Wysłanie komunikatu polega tylko na utworzeniu obiektu odpowiedniej klasy i wywołaniu metody *wyslij*. Do stworzenia wiadomości bezparametrowej, np. rozkaz zakończenia pracy agenta, używa się konstruktora *komunikat(BYTE)*, jako parametr podając identyfikator. Wszystkie klasy pochodne posiadają konstruktor tworzący obiekt z listy parametrów, które mają zostać wysłane.

Podczas odbioru informacji najpierw jest odbierany identyfikator, rozpoznawana na jego podstawie klasa komunikatu, następnie tworzony obiekt odpowiedniej klasy i wywoływana metoda *odbierz* tego obiektu. Metody *wyslij* i *odbierz* realizują konwersje porządku bajtów z sieciowego na lokalny i odwrotnie, dzięki czemu struktury te można wykorzystać do przesyłu danych w sieciach heterogenicznych

Wszystkie adresy są przekazywane w postaci tekstu i zostają przetłumaczone na adres właściwy dla danego protokołu dopiero przed rozpoczęciem samej transmisji. Pozwala to na przesyłanie adresów w tej samej formie dla różnych protokołów i ułatwia dalszą rozbudowę systemu o nowe protokoły transmisji.

Komunikaty o błędach są tworzone poprzez ustawienie identyfikatora jako sumy K_BLAD oraz numeru błędu. Nie zawierają dodatkowych parametrów, więc są przesyłane z wykorzystaniem klasy *komunikat*.



Rys. 3. Typowy przebieg transmisji komunikatów
Fig. 3. Typical message communication

2.3. Program agenta – proces sterujący

Program agenta składa się z jednego procesu sterującego oraz jednego lub więcej procesów „pracowników" tworzonych na czas pomiarów na podstawie parametrów przesłanych przez nadzorcę. Komunikacja pomiędzy procesami jest realizowana przez pamięć dzieloną oraz semafory. Do obsługi pamięci dzielonej wykorzystano funkcje systemu V: *shmget*, *shmat*, *shmdet*, *shmctl*. Procedury operujące na semaforach są oparte na standardzie POSIX: *sem_open*, *sem_close* i *sem_unlink*. Wykorzystano jeden ciągły blok pamięci dzielonej przydzielany na początku działania programu oraz po 2 semafory dla każdego procesu nadawczo/odbiorczego (*sem_stan* oraz *sem_koniec*). Ustawienie pierwszego z semaforów sygnalizuje możliwą zmianę zmiennej stanu, drugiego – zakończenie procesu.

Po uruchomieniu agenta tworzony jest tylko proces sterujący. Program rozpoczyna działanie od inicjalizacji zmiennych statycznych i alokacji pamięci na zmienne dynamiczne. Następnie następuje przydzielenie identyfikatora pamięci dzielonej za pomocą procedury *shmat*. Jej rozmiar określa maksymalną liczbę procesów nadawczo/odbiorczych. Domyślnie istnieje możliwość uruchomienia maksymalnie 10 procesów, lecz można to zmienić modyfikując na etapie kompilacji stałą MAX_PRACOWNIKOW. Następnie proces sterujący

czeka na połączenie z programem nadzorcy. Po jego nawiązaniu w pętli oczekuje na komunikaty sterujące, odbiera je oraz wykonuje rozkazy w nich zawarte. Jeżeli nastąpi zerwanie połączenia przez program nadzorcy następuje powrót do oczekiwania na połączenie.

2.4. Program agenta – proces nadawczo / odbiorczy

Proces nadawczo / odbiorczy rozpoczyna swoje działanie, gdy wszystkie parametry pracy są znane. Proces ten może pracować w 2 trybach – jako nadawca lub odbiorca. Do rozpoznania trybu pracy wykorzystano zmienną stanu.

Po uruchomieniu procesu na podstawie parametrów czasowych jest tworzony zegar (ang. *timer*), wyzwalany pierwszy raz w momencie rozpoczęcia pracy procesu, następnie co krok określony w parametrach pracy. Procedura obsługi zegara ustawia semafor sem_stan. Jeżeli program pracuje w trybie nadawcy, główna pętla procesu oczekuje na ustawienie semafora, po zmianie jego wartości wysyła porcje danych w sieć. Rozmiar paczki danych pobierany jest z klasy generator, opisanej w dalszej części tego opracowania. W trybie odbioru program oczekuje na odbiór danych. Zegar jest wyzwalany co 1 sekundę, aby przerwać procedurę odczytu z gniazda i sprawdzić warunek końca czasu badań. Czas każdego zdarzenia jest odnotowywany w buforze na wyniki. Po jego zapełnieniu pomiary zapisywane są z pliku, a następnie przez proces sterujący przesyłane do programu nadzorcy.

2.5. Pomiar czasu i rejestrowanie wyników

Aby ułatwić pomiar czasu, stworzono klasę *czas*, która przechowuje chwilową wartość czasu z dokładnością do milisekund. Jest wykorzystywana zarówno przez program nadzorczy, jak i agenta. Przeciążone operatory logiczne oraz arytmetyczne umożliwiają porównywanie wartości obiektów klasy. Aby ułatwić transmisje danych przez sieć, zaimplementowano metody *tohost* i *tonetwork* zamieniające porządek bajtów z sieciowego na lokalny i odwrotnie. Zdefiniowano operatory przekształcenia typu na i z typu *int*, przeliczające czas na milisekundy.

Do przechowywania wyników pomiarów stworzono klasę *wynik*. Zawiera jedno pole typu *czas* przechowujące chwilę czasu, w której nastąpiło wysłanie lub odebranie pakietu, jednobajtowe pole *licznik* i dwa pola typu WORD: *liczba* oraz *przekłamania*. W każdym pakiecie znajduje się licznik pozwalający wychwycić pakiety odbierane w innej kolejności, niż zostały wysłane. Dzięki temu polu program nadzorczy analizując wyniki jest w stanie obliczyć prawidłowy czas transmisji każdej porcji danych. Zmienna liczba przechowuje rozmiar pakietu. Pole przekłamania zawiera liczbę przekłamanych bajtów w pakiecie, a jego wartość ma znaczenie tylko po stronie odbierającej.

2.6. Synchronizacja czasów

Wykonywanie pomiarów czasu przesyłu danych pomiędzy dwoma różnymi komputerami w sieci wymaga synchronizacji ich zegarów. Zaprzestanie tego kroku wprowadzałoby znaczny błąd, ponieważ wszystkie czasy transmisji zostałyby zwiększone o różnice wskazań zegarów stacji roboczych wykorzystanych do przeprowadzenia pomiarów. Do synchronizacji można wykorzystać protokół *ntp*, jednak wymaga to uprawnień administratora oraz uruchomienia w systemie kolejnego demona. Aby ominąć te ograniczenia, zdecydowano się wprowadzić własną metodę synchronizacji czasu, która nie modyfikuje zegarów lokalnych żadnego z komputerów biorących udział w pomiarach, a tylko odnotowuje różnice czasów lokalnych i odpowiednio koryguje wyniki. Synchronizacja odbywa się pomiędzy programem nadzorczym a każdym z agentów osobno i jest inicjowana przez użytkownika. Jest dokonywana tylko raz przed wykonaniem pomiarów. Założono, że w trakcie badań zegary nie wymagają synchronizacji, co nie wprowadziło zauważalnych przekłamań wyników w przeprowadzonych testach.

Synchronizacja zegarów odbywa się po naciśnięciu przez użytkownika odpowiedniego przycisku w programie nadzorczym. Program wysyła komunikat K_PRZESL_CZAS oraz odnotowuje czas wysłania. Agent odpowiada przesłaniem komunikatu K_CZAS zawierającego jego czas lokalny. Program nadzorczy odnotowuje wartość zegara lokalnego podczas odbioru tego komunikatu. Następnie oblicza się sumaryczny czas transmisji w dwie strony, dzieląc go przez dwa można obliczyć przypuszczalne opóźnienie wprowadzone przez transmisje. Procedura ta jest powtarzana wiele razy (liczba powtórzeń jest zapisana w stałej, domyślnie przyjmującej wartość 100), aby uśrednić wartość opóźnienia. Założono, że czas transmisji w obydwie strony jest taki sam, co jest prawdą w większości instalacji sieciowych. Różnica wskazań zegarów jest obliczona poprzez uśrednienie różnic wskazanych przez kolejne pomiary, skorygowanych o średnie opóźnienie.

Po zakończeniu badań i odebraniu wyników program nadzorczy przelicza je na czas lokalny. Wyniki każdego z agentów przeliczane są niezależnie, a dalsze obliczenia wyników są wykonywane na podstawie skorygowanych pomiarów.

2.7. Generacja rozmiarów pakietów

Wysyłanie danych w sieć odbywa się cyklicznie co ustalony przez użytkownika czas. Rozmiar pojedynczej paczki danych może być zmienny w czasie. Program nadzorcy pozwala na wybór jednego z kilku generatorów oraz ustalenie jego parametrów. Zdefiniowano wirtualną klasę generator posiadającą tylko jedną metodę iter() zwracającą rozmiar kolejnej ramki w bajtach oraz konstruktor tworzący obiekt z wskaźnika na tekst. Z każdym

stosowanym generatorem jest związana jedna klasa pochodna, przechowująca jego parametry oraz przeciążająca metodę iter. Konstruktor umożliwia stworzenie obiektu o parametrach wpisanych przez użytkownika w postaci ciągu znaków, co ułatwia rozbudowę systemu o dowolny rodzaj generatorów. Dla każdego procesu nadawczo/odbiorczego jest tworzony osobny generator, co pozwala z jednego komputera wysyłać dane o różnej charakterystyce.

W istniejącej aplikacji jest możliwy wybór jednego z poniższych generatorów: gamma, Erlanga, normalny, lognormalny, beta, wykładniczy, Cauchy'ego, Pareto, dwupunktowy deterministyczny, geometryczny, plikowy. Ostatni z powyższych odczytuje kolejne rozmiary pakietów z pliku, co pozwala na generowanie ruchu o dowolnej charakterystyce.

2.8. Prezentacja wyników pomiarów

Wyniki pomiarów są zbierane do plików CSV zawierających identyfikator pakietu, czas z dokładnością do milisekund, rozmiar pakietu oraz liczbę przekłamań. Program nadzorczy pozwala na zapisanie osobnych plików z wynikami dla każdego procesu nadawczo/odbiorczego. Umożliwia także stworzenie plików różnicowych zawierających czas przesłania pakietu (różnicę pomiędzy czasem nadania i odbioru) oraz obliczenie średniego, maksymalnego i minimalnego czasu przesyłu.

Oprogramowanie nie umożliwia graficznej prezentacji wyników. Dzięki zastosowaniu popularnego formatu CSV wyniki mogą zostać przetworzone w oprogramowaniu statystycznym lub arkuszu kalkulacyjnym (np. Excel).

3. Porównanie z konkurencyjnymi rozwiązaniami

Na rynku jest dostępnych stosunkowo dużo narzędzi służących pomiarowi parametrów transmisji [3,4,5]. Jednak absolutna większość pozwala tylko na pomiar istniejącego ruchu, nie umożliwiając generacji pakietów, w związku z tym w pewnych zastosowaniach nie może być wykorzystana.

Najbardziej zbliżonym produktem jest Lantrafic V2 opracowany przez francuską firmę ZTI. Narzędzie to pozwala generować ruch w protokołach TCP i UDP, nie zaimplementowano natomiast ATM. Składa się z dwu części: nadawcy (ang. *transmitter*) i odbiorcy (ang. *receiver*). Moduł nadawcy może jednocześnie generować do 16 transmisji. Parametry transmisji muszą zostać ustalone osobno dla każdego nadawcy – nie przewidziano zdalnego zarządzania i przesyłania wyników, co utrudnia przeprowadzenie badań rozległych sieci. Dane są przesyłane przez odbiorcę z powrotem do nadawcy, który zbiera wyniki. Dzięki temu nie jest konieczna synchronizacja zegarów, jednak wszystkie pomiary dotyczą

sumarycznego czasu transmisji pakietu w dwie strony, przez co narzędzie nie nadaje się do pomiarów kanałów niesymetrycznych. Lantrafic działa w systemie Window 9x lub NT. Wykorzystano tylko 3 generatory rozmiarów pakietów: stały, wykładniczy i Pareto, lecz istnieje także możliwość generowania pakietów w zmiennych odstępach czasu. Program umożliwia wyświetlenie statystyk oraz wykresów wyników pomiarów. Porównanie cech systemu Lantrafic oraz oprogramowania powstałego w ramach niniejszej pracy prezentuje tabela 1.

Tabela 1

Porównanie właściwości narzędzia opracowanego w IITiS PAN i systemu Lantraffic

Cecha	Lantrafic	Narzędzie opracowane w IITiS PAN
zarządzanie transmisją	zdecentralizowane	zcentralizowane
wykorzystywane protokoły	UDP, TCP	UDP, ATM
maksymalna liczba jednoczesnych połączeń	16	10 (możliwa zmiana przez rekompilacje pakietu)
liczba generatorów rozmiarów pakietów	3	11
możliwość wysył. pakietów w zmiennych odst. czasu	tak	nie
metoda pomiaru czasu	jednopunktowa	dwupunktowa

4. Podsumowanie i wnioski

W niniejszej pracy omówiono narzędzie umożliwiające generowanie transmisji o zadanych parametrach oraz pomiar opóźnień tej transmisji. Narzędzie to pozwala na efektywny i tani pomiar parametrów sieci komputerowych. Dzięki funkcji generacji pakietów można sprawdzić wydajność sieci w konkretnych zastosowaniach oraz przetestować jej skalowalność. Oprogramowanie może zostać wykorzystane do weryfikacji wyników przeprowadzonych symulacji. Szeroki zakres generatorów rozmiarów pakietów pozwala na generacje ruchu o bardzo różnej charakterystyce.

W obecnej wersji oprogramowanie agenta może działać tylko w systemach zgodnych z UNIX, co ogranicza jego zastosowanie. W przyszłości planowane jest rozszerzenie narzędzia o moduł agenta pracujący w środowisku Windows. Planowana jest także rozbudowa o dalsze klasy generatorów rozmiarów pakietów m. in. multifraktalne oraz oparte na łańcuchach Markowa [7] opracowane w IITiS PAN.

LITERATURA

1. Sochan A.: Biblioteka sygnalizacyjna dla programowego zestawiania połączeń w heterogenicznym środowisku sieci ATM, Studia Informatica, vol 21, nr 1 (39), pp 451-467, 2000
2. Feit S.: TCP/IP Architecture, Protocols and Implementation, McGraw Hill, 1993.
3. Caceres R.: Measurements of Wide Area Internet Traffic, Report UCB/CSD 89/550, Computer Sciences Division, University of California, Berkeley, 1989.
4. Heegaard P.: GenSyn - a generator of synthetic Internet traffic used in QoS experiments
5. Squillante M., Yao D., Zhang L.: Internet Traffic: Periodicity, Tail Behavior and Performance Implications, System Perormance Evaulation, CRC Press 2000
6. Grochla K., Sochan A.: Sieci prywatne oparte na technologii ATM, Studia Informatica, vol 21, nr 1 (39), pp 205-216, 2000
7. Jędruś S.: Porównanie i ocena dokładności metod modelowania natężenia ruchu w sieciach komputerowych, Studia Informatica, vol 21, nr 1 (39), pp 523-540, 2000

Recenzent: Dr inż. Andrzej Kwiecień

Wpłynęło do Redakcji 27 marca 2001 r.

Abstract

This paper briefly presents a generator of network traffic. It is able to simulate different types of transmissions, and to measure their parameters. It can be used for checking existing network, testing different posibilities of its enlargement, or for verifying simulations done using other programs. The program architecture and behavior is described, along with conclusions, based on the implementation. The software consists of two parts: a network agent and a management program. This separation allows the measurement to be remotely managed via the TCP/IP.

Also, there is presented a short comparison to the existing commercial software of this kind.