

Przemysław GŁOMB, Krzysztof GROCHLA
Instytut Informatyki Teoretycznej i Stosowanej PAN

MODELOWANIE STACJI ROBOCZYCH W SIECI ATM PRZY UŻYCIU SYMULATORA OMNeT++

Streszczenie. Praca prezentuje opis symulacji stacji roboczej w sieci ATM. Przedstawiono środowisko symulacji OMNeT++, badania przeprowadzone w celu uzyskania charakterystyk protokołów oraz konstrukcję samej symulacji.

AN ATM WORKSTATION MODELLING USING OMNeT++ SYMULATOR

Summary. This work presents a ATM workstation simulation. A overview of the OMNeT++ framework is presented, then the actual simulation, with explanation of the structure and methods for acquiring protocols' characteristics.

1. Wstęp

W badaniach nad złożonymi sieciami komputerowymi, w szczególności nad kształtowaniem ruchu i parametrami jakości usług (ang. *Quality of Service*), często trzeba sięgnąć po modele symulacyjne [7]. Ma to znaczenie, szczególnie gdy sytuacja wyklucza zastosowanie rzeczywistych obiektów badawczych (sieci tworzonych wyłącznie na potrzeby analizy ruchu), a metody analityczne są mniej efektywne.

Częścią prac naukowo - badawczych w Laboratorium Systemów Multimedialnych IITIS PAN jest zagadnienie zapewnienia odpowiedniej jakości usług w heterogenicznych sieciach komputerowych. Badania te są prowadzone częściowo w warunkach rzeczywistych, a częściowo na symulacjach konfiguracji alternatywnych.

Do stworzenia takiej symulacji niezbędne jest posiadanie bazy obiektów odpowiadających rzeczywistym komponentom sieci komputerowych. Szczególnie ważne jest

posiadanie symulacyjnych modeli stacji roboczych, zbliżonych do swoich fizycznych odpowiedników, przy których użytkownicy generują większość ruchu w sieci. Zgodne z rzeczywistością źródła ruchu umożliwiają dokładne przetestowanie różnych algorytmów kształtowania ruchu oraz reakcji sieci na przeciążenia i sytuacje wyjątkowe.

Niniejsza praca zawiera opis symulacji stacji roboczej PC w sieci ATM, wykonanej na potrzeby prac badawczych w ITiS PAN. Następne rozdziały przedstawiają kolejno: opis środowiska modelowania, sposób stworzenia generatorów ruchu dla odpowiednich protokołów, budowę samej symulacji oraz wnioski.

2. Opis środowiska symulacji OMNeT++

OMNeT++ jest środowiskiem tworzenia obiektowych symulacji zdarzeń dyskretnych. Opracowany został przez Andreasa Vargę na Politechnice Budapeszteńskiej [1], z myślą o budowaniu symulacji takich układów jak np. sieci komputerowe (w dowolnej skali), modele protokołów.

Symulator składa się z dwóch części: zbioru bibliotek, plików nagłówkowych i prekompilatora, przystosowanych do współpracy z kompilatorem C++, oraz programów wspomagających. Dostępny jest w wersjach dla systemów Windows i Unix.

2.1. Koncepcje modelowania w OMNeT++

Symulowany system składa się z połączonych ze sobą modułów. Moduły, posiadające swoje wejścia i wyjścia, odpowiadają obiektom symulowanym (komponentom), a połączenia między nimi określają trasy przesyłu komunikatów między nimi (np. pakietów).

Moduły dzielą się na proste i złożone. Te pierwsze konstruowane są w języku C++ w formie oddzielnych klas, dziedziczonych po klasie bazowej (*cSimpleModule*). W każdej z nich użytkownik modyfikuje jedną z dwóch metod (*activity()* lub *handleMessage()*), określając zachowanie tej części symulacji (odpowiednio - aktywnej cały czas lub tylko w momencie przybycia na któreś z wejść wiadomości). Elementami zachowania są np.: odczytywanie wiadomości z wejścia, wysyłanie wiadomości do określonego wyjścia, uaktualnienie statystyk, zmiany stanu.

Moduły złożone składają się z dowolnej liczby prostych i złożonych. Tworzone są w języku opisu symulacji NED; ich określenie to definicja wejść i wyjść, deklaracji elementów składowych, opis sposobu ich połączeń między sobą oraz parametrów początkowych, jakie im należy nadać. Połączenia między modułami można zdefiniować jako kanały komunikacyjne, przypisując im parametry (przepustowość, stopa błędów, opóźnienie).

Tak stworzoną symulację kompiluje się następnie przy użyciu dowolnego kompilatora C++ (np. Visual C++). Najpierw pliki '.ned' zamieniane są na '.cpp' przez prekompilator języka NED, a następnie całość tłumaczona jest na plik wykonywalny '.exe'.

Moduł na najwyższym poziomie hierarchii nazywany jest siecią; każdy wykonywany proces symulacji (uruchomienie) dotyczy jednej konkretnej sieci. Podczas tworzenia symulacji można zdefiniować ich dowolną ilość (odpowiadających np. różnej topologii sieci [5]), a potem przy uruchomieniu wybrać dowolną z nich. Podobnie możemy dowolnie przy każdym uruchomieniu ustalić wartości parametrów poszczególnych modułów. Parametry uruchomienia określone są w pliku tekstowym '.ini'.

Wielką zaletą tego środowiska jest możliwość automatycznego stworzenia wizualizacji symulacji (stąd pochodzi schemat modelu stacji roboczej na rysunku 3).

OMNeT++ dostarcza dwa narzędzia wspomagające projektowanie symulacji: program do graficznego projektowania modułów złożonych GNED oraz program do prezentacji wyników PLOVE (bazujący na programie *gnuplot*).

Wybór tego środowiska uzasadniony był następującymi czynnikami: umożliwiał tworzenie modułowych symulacji, których komponenty mogą być ze sobą dowolnie wymieniane; symulator i jego kod źródłowy są publicznie dostępne, a sam program jest dynamicznie rozwijany i ma dużą liczbę aktywnych użytkowników, którzy wnoszą wkład w jego rozwój; jego możliwości są porównywalne z komercyjnymi symulatorami.

3. Struktura ruchu generowanego przez stację roboczą w sieci ATM

Głównym zadaniem tworzonoego modelu stacji roboczej jest generowanie ruchu jak najbardziej zbliżonego do rzeczywistego. Z tego powodu przyjęto, że model będzie się składał z części, odwzorowujących poszczególne programowe komponenty systemu operacyjnego. Każda z tych części będzie generowała ruch odpowiadający określonym protokołom. Ruch generowany przez stację jest sumą strumieni pakietów odpowiednich protokołów generowanych przez określone części.

W celu uzyskania dokładnych parametrów ruchu przeprowadzono badania w podsięci ATM sieci lokalnej ITiS PAN. Badany obszar obejmował 5 stacji roboczych PC (systemy operacyjne Windows 2000 i NT). Sieć ułożona jest na okablowaniu UTP kat 5, przepustowość wynosi 155 Mbit/s. Obsługuje ją przełącznik ATM ForeRunner LE 155. W sieci uruchomiona jest emulacja sieci lokalnej LANE Ethernet (wersja 1.0). Komputery współpracują z serwerem domeny (Windows 2000 Server) oraz serwerem plików w systemie Novell (wersja 5.0).

3.1. Przebieg badań

Badania przeprowadzono w celu identyfikacji występujących w sieci protokołów oraz ustalenia ich charakterystyk. Ich celem było uzyskanie informacji o dwóch rodzajach ruchu:

- Protokołach odpowiedzialnych za znaczną część ruchu, generowanych bezpośrednio przez aplikacje użytkownika (np. HTTP, POP3);
- Protokołach generowanych automatycznie, przez system operacyjny (np. KERBEROS, DNS), albo przez oprogramowanie karty sieciowej (np. ILMI, LANE Control), najczęściej o periodycznym wzorcu ruchu.

Głównym narzędziem pracy był analizator sieciowy Network General ATM Sniffer, podłączany między stacją roboczą a przełącznik ATM. W części pomiarów wykorzystano dodatkowo programowy analizator ruchu (Distinct Network Monitor), ze względu na bardziej rozbudowane opcje filtrowania i prezentacji zawartości pakietów.

Badania prowadzono w pięcio- i dziesięciominutowych cyklach, część z nich w czasie normalnej pracy sieci (użytkownicy zalogowani i generujący ruch sieciowy) oraz „po godzinach” (włączone komputery bez logowania, tylko ruch automatyczny).

Ze względu na to, że tworzony był model pojedynczej stacji roboczej, zbierano statystyki wyłącznie ruchu wychodzącego ze stacji (dotyczy to również prezentowanych tutaj wyników).

3.2. Wyniki badań

Mimo pozornej złożoności ruch generowany przez stację roboczą wykazuje dużą regularność. Na najwyższym poziomie (według modelu warstwowego ISO/OSI) znajdują się aplikacje użytkownika. Są one głównym źródłem ruchu. W wyniku badań stwierdzono, iż najczęściej powtarzające się źródła należą do jednej z trzech grup: przeglądanie sieci zdalnej (ruch internetowy - HTTP, FTP, HTTPS) lub lokalnej (np. pakiety NetBT), poczta elektroniczna (SMTP, POP3, IMAP) oraz praca zdalna (Telnet, XWindows). Często taki ruch jest ciągiem zapytań / poleceń wysyłanych w łatwy do odwzorowania sposób (np. pakiety HTTP GET - rysunek 2, charakterystyczne „piki” w momencie ładowania nowej strony).

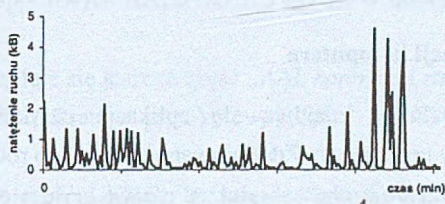
Do tego dochodzi ruch generowany automatycznie: protokół zarządzania siecią SNMP, wyszukiwania nazw DNS, dynamicznej konfiguracji DHCP, wymiany informacji o zasobach w sieci Windows SMB, informacji kontrolnych ICMP, określania adresu fizycznego ARP, zabezpieczenia Kerberos.

Wszystkie te protokoły na poziomie warstwy czwartej są enkapsulowane w datagramy TCP/IP, a następnie (przez oprogramowanie emulacji LAN) w ramki ethernetowe. Na poziomie warstwy adaptacyjnej ATM ramki zawierane są dodatkowo w pakiety AAL5 PDU. Generowany jest również ruch charakterystyczny dla tej sieci - protokoły zarządzania ILMI,

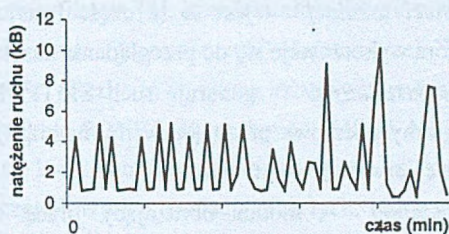
sterowania LANE Control oraz sygnalizacja. Na poziomie ATM ten ruch zamieniany jest na komórki, uzupełniani dodatkowo komórkami zarządzającymi OAM i RM.

Zaobserwowano, że większość ruchu automatycznego generowanego jest periodycznie, niezależnie od działań użytkownika. Wyjątkiem jest część sygnalizacji ATM, DNS i ARP, jednak nawet te protokoły można bez strat dokładności przybliżyć regularnym schematem. Przykład sumarycznego ruchu automatycznego jest na rysunku 1.

Wyniki badań posłużyły do opracowania generatorów ruchu dla poszczególnych protokołów. Całkowity ruch generowany przez stację jest superpozycją tych ruchów, których generowanie nie wymaga skomplikowanych obliczeń. Dzięki temu model jest efektywny czasowo i pamięciowo.



Rys.1. Przykładowe natężenie ruchu generowanego automatycznie; pomiar co 1 s
Fig. 1. Sample intensity of automatically generated traffic at 1 second intervals



Rys.2. Przykładowe natężenie ruchu HTTP; pomiar co 1 s
Fig. 2. Sample HTTP traffic intensity at 1 second intervals

4. Budowa symulacji

Model stacji roboczej wzorowany jest na strukturze części sieciowej systemu operacyjnego Windows (wersja 2000). Poszczególne części głównego modułu - „stacja robocza” - są ułożone hierarchicznie i odpowiadają funkcjonalnie aplikacjom, fragmentom systemu operacyjnego, elementom programowym i sprzętowym karty sieciowej. Jest on

przeznaczony do wykorzystania w większych projektach - na przykład symulacji sieci lokalnej ATM.

Większość z części modułu zawiera generatory pakietów, odpowiadające odpowiednim protokołom (np. *"klient poczty elektronicznej"* - SMTP, POP3 i IMAP). Charakterystyki ruchu tych protokołów zostały stworzone na podstawie badań opisanych w poprzednim rozdziale. Niewielkim nakładem pracy można wprowadzić dodatkowe źródło ruchu (odpowiadające np. uruchomieniu aplikacji o znanej charakterystyce ruchu) - wymaga to jedynie zdefiniowania go jako modułu prostego i przyłączenia do struktury.

Model stacji roboczej przedstawiony został na rysunku 3. Składa się on z dwóch części: elementów tworzących system operacyjny oraz tych składających się na sprzęt i oprogramowanie karty sieciowej.

4.1. Budowa symulacji komputera

Na najwyższym poziomie znajdują się aplikacje użytkownika, obrazujące ruch bezpośrednio generowany przez niego. Zdefiniowano cztery tego rodzaju moduły:

- „*przeglądarka sieci rozległej*” – pod tą nazwą kryje się zazwyczaj przeglądarka intersieci oraz klient ftp. Generowany przez ten moduł ruch obejmuje protokoły internetowe: HTTP (sporadycznie także HTTPS) oraz FTP; ich charakterystyki wyliczane są na bazie zebranych danych;
- „*przeglądarka sieci lokalnej*” – jest to ta część interfejsu użytkownika systemu operacyjnego, którą wykorzystuje się do przeglądania zasobów sieci lokalnej;
- „*klient poczty elektronicznej*” – generuje ruch SMTP, POP3 oraz IMAP. Często spotyka się programy pocztowe pracujące w tle, nawiązujące co kilka - kilkanaście minut połączenie z serwerem pocztowym;
- „*klient pracy zdalnej*” – moduł obrazujący pracę na zdalnym komputerze, wykonywaną przy pomocy programów, takich jak Telnet czy XWindows.

Niższy poziom zajmują trzy fragmenty systemu operacyjnego:

- „*klient sieci Microsoft Network*” – przekazuje dalej zgłoszenia przeglądania sieci lokalnej oraz generuje periodyczny ruch systemowy (związany np. z udostępnianiem folderów i drukarek);
- „*klient sieci Novell*” – wysyła zgłoszenia do serwera Novell, zarówno na żądanie użytkownika (np. przeglądanie zasobów serwera), jak i periodycznie (podtrzymanie łączności);
- „*system operacyjny*” – przede wszystkim zajmuje się obsługą protokołu TCP/IP, pakując w datagramy otrzymane (od wyżej opisanych modułów) pakiety danych.

Zgodnie z charakterystykami produkuje również pakiety niektórych protokołów (konkretnie SNMP, DHCP, Kerberos, DNS, ARP).

Pakiety, wychodzące z części „system operacyjny”, przekazywane są do „karty sieciowej”.

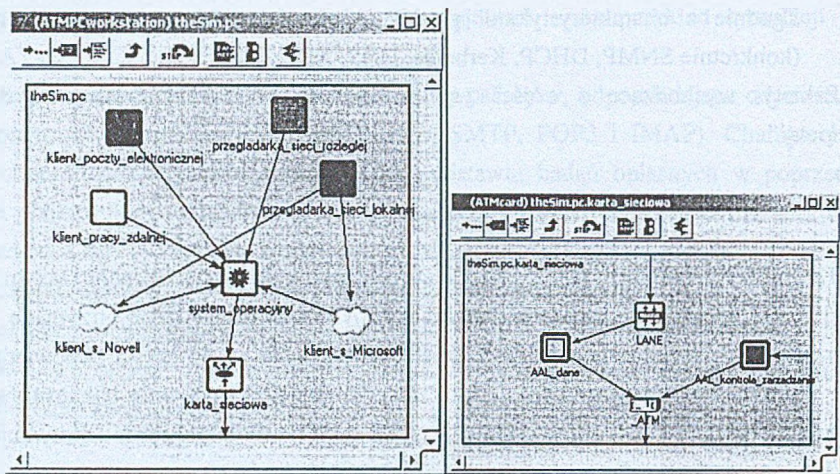
4.2. Budowa symulacji karty sieciowej

Ten moduł zbudowany jest według hierarchii protokołu ATM. Na najwyższym poziomie znajduje się część emulująca sieć lokalną („LANE”), pakująca dane otrzymane z warstw wyższych w ramki ethernetowe. Te z kolei przekazywane są do części odpowiedzialnej za dane warstwy adaptacyjnej ATM („AAL dane”), gdzie są ponownie enkapsulowane, tym razem do jednostek transportowych AAL5 (AAL 5 PDU). W takiej formie otrzymuje je część „ATM”.

Na tym poziomie znajduje się jeszcze część „AAL kontrola i zarządzanie”. Odpowiada za okresowe generowanie pakietów protokołów ILMI i LANE control oraz sygnalizacji ATM.

Najniżej znajduje się część „ATM”. Przychodzące tam pakiety dzielone są na komórki ATM i umieszczane w buforze. Równolegle zawartość z bufora jest pobierana i transmitowana w sieć (kanał wyjściowy z modułu, zdefiniowany w OMNeT zgodnie ze specyfikacją transmisji 155 Mbit/s po skrętce nieekranowanej). W strumień komórek wplatanie są, zgodnie ze specyfikacją [4], komórki zarządzania ruchem RM (ang. *Resource Management*) oraz OAM (ang. *Operation and Maintenance*). Komórki generowane są synchronicznie co 2,56 μ s [2] (liczba ta wynika z rzeczywistej dostępnej przepustowości w sieci ATM opartej na protokole SONET, która wynosi 149.76 Mbit/s), w razie opróżnienia bufora strumień dopełnia się komórkami pustymi (ang. *blank cells*).

Dla potrzeb mniej szczegółowych symulacji oraz tych, które mają objąć większy okres czasu, stworzono moduł „prosta karta sieciowa”, który nie generuje komórek, a jedynie przekazuje do wyjścia otrzymane z góry pakiety. Nie zawsze jest potrzeba dokładnego odwzorowania ruchu na niskim poziomie, a konieczność generowania zdarzenia co kilka mikrosekund czasu symulacji może w niepożądany sposób opóźnić badania.



Rys.3. Model stacji roboczej - komputer i karta sieciowa

Fig. 3. Workstation's model - the computer and the network card

4.3. Działanie symulacji

Po uruchomieniu, poszczególne części modułów przystępują do generowania pakietów zgodnie z zadanymi charakterystykami. W miarę poruszania się w dół hierarchii protokołów, są one odpowiednio enkapsulowane; poszczególne strumienie pakietów splatane są ze sobą, tworząc wynikowy ruch generowany przez stację.

W dowolnym miejscu modelu mogą zostać umieszczone funkcje zbierające lub analizujące dane, których wyniki po zakończeniu symulacji można zaprezentować korzystając z programu PLOVE.

5. Wnioski

W niniejszej pracy przedstawiono symulator stacji roboczej pracującej w sieci ATM. Jest on przeznaczony do wykorzystania w symulacjach sieci lokalnych opartych na tym protokole, chociaż może też zostać użyty do generowania charakterystyk obciążenia.

Dzięki szczegółowym obserwacjom protokołów, występujących w sieci lokalnej, możliwe było wierne oddanie całościowego obciążenia generowanego przez stację przy zachowaniu niskiej złożoności czasowej i pamięciowej. Złożony ruch bardzo często jest sumą prostych składowych - pamiętając o tym można łatwo wygenerować nawet skomplikowane jego charakterystyki.

Wykorzystanie środowiska OMNeT znacznie ułatwia tworzenie modułowych symulacji dzięki dobrym parametrom wykonania i automatycznie tworzonej wizualizacji.

Dalsze prace będą obejmowały rozszerzenie możliwości symulacji o Classical IP, którego w momencie oddawania artykułu do druku nie można było zbadać z przyczyn technicznych.

LITERATURA

1. Varga A.: OMNeT++ Discrete Event Simulation System User Manual, <http://www.hit.bme.hu/phd/vargaa/omnetpp.htm>, Budapest 2001.
2. Adams J.: ATM for Service Providers. Chapman & Hall, London 1997.
3. Sackett G., Metz Ch.: ATM and Multiprotocol Networking. McGraw-Hill, New York 1997.
4. Dokumentacje ATM Forum, <http://www.atmforum.org>.
5. Lai J., Wu E., Varga A., Sekercioglu A., Egan G.: A Simulation Suite for Accurate Modeling of IPv6 Protocols. w: Proc 2nd International OMNET++ Workshop, Berlin 2002.
6. Bless R.: Using Realistic Internet Topology Data for Large Scale Network Simulations in OMNET++. w: Proc 2nd International OMNET++ Workshop, Berlin 2002.
7. McDysan D.: QoS & Traffic Management in IP & ATM Networks. McGraw-Hill 2000.

Recenzent: Dr inż. Mirosław Skrzewski

Wpłynęło do Redakcji 17 kwietnia 2002 r.

Abstract

This paper presents a ATM workstation simulation, constructed using OMNeT++ suite. Chapter two presents a brief introduction into OMNeT++ simulation framework. Chapter three discusses method for obtaining characteristics of client - side traffic of different protocols. A brief summary of results is also presented. Chapter four details simulation structure, presenting individual components that make up the model. Chapter five contains some general conclusions.