

Marek KACPRZAK

METODA OCENY CHARAKTERYSTYK CZASOWYCH ADAPTERÓW KOMUNIKACYJNYCH DLA SIECI KOMPUTEROWYCH

Streszczenie. Przedstawiono zastosowanie języka formalnej specyfikacji Estelle (ISO 9074) do budowy modelu symulacyjnego adapterów komunikacyjnych do oceny charakterystyk czasowych.

THE METHOD FOR PERFORMANCE EVALUATION OF COMMUNICATION CONTROLLERS FOR COMPUTER NETWORKS

Summary. The application of formal description technique Estelle (ISO 9074) for generation of simulation model for communication controllers' performance evaluation is presented.

МЕТОД ОЦЕНКИ ВРЕМЕННЫХ ХАРАКТЕРИСТИК КОММУНИКАЦИОННЫХ АДАПТЕРОВ ДЛЯ СЕТЕЙ ЭВМ

Резюме. В работе представлено применение языка формального описания протоколов Estelle (ISO 9074) для создания имитационной модели для оценки пропускной способности коммуникационных адаптеров.

1. Wstęp

Nowoczesne rozłożone systemy informatyczne są budowane na bazie lokalnych, miejskich lub rozległych otwartych sieci komputerowych, tzn. zgodnych z warstwowym modelem odniesienia OSI (ISO 7493). Dla kompletności i precyzji, protokoły komunikacyjne poszczególnych warstw są opisywane w językach formalnej specyfikacji, takich jak: Estelle (ISO 9074), LOTOS (ISO 8807) lub SDL (CCITT Z.100).

Dolączenie komputera dowolnego typu do otwartej sieci komputerowej wymaga opracowania adaptera komunikacyjnego — urządzenia pośredniczącego w wymianie informacji między komputerem a podszytem komunikacyjnym sieci komputerowej. Nowoczesne adaptery komunikacyjne realizują funkcje warstw 1–6 i części warstwy 7 modelu odniesienia ISO–OSI [1]. Umożliwia to zdecydowane odciążenie komputera od zadań komunikacyjnych, pod warunkiem właściwego zaprojektowania adaptera — realizacji protokołów komunikacyjnych zgodnie z ich specyfikacją, w możliwie najkrótszym czasie. Ocena na etapie projektowania adaptera jego przewidywanych charakterystyk czasowych, ze względu na złożoność zagadnienia, wymaga zastosowania metod symulacji komputerowej. Znane są w literaturze [2] metody generowania oprogramowania komunikacyjnego i testów do weryfikacji implementacji bezpośrednio z formalnej specyfikacji protokołów. Podejście takie może być również stosowane do tworzenia modeli symulacyjnych do oceny charakterystyk czasowych projektowanego adaptera. Spośród wymienionych języków formalnej specyfikacji najdogodniejszym do tych celów jest język Estelle, jako najbardziej zbliżony do implementacji [3].

W pracy przedstawiono metodę budowy programu symulacyjnego adaptera komunikacyjnego, realizującego określone protokoły komunikacyjne, na podstawie opisu tych protokołów w języku Estelle.

2. Użycie języka Estelle do implementacji i symulacji

System komunikacyjny jest opisywany w języku Estelle za pomocą modułów i kanałów [4]. Moduły reprezentują rozszerzone automaty o skończonej liczbie stanów. Moduł zawiera, między innymi, deklaracje zgodne ze składnią języka Pascal oraz opis przejścia (zmiany stanu automatu) w postaci programu w języku Pascal, ze specyficznymi dla Estelle rozszerzeniami i ograniczeniami. Moduły komunikują się ze sobą przesyłając interak-

cje przez zewnętrzne punkty dostępu. Punkty dostępu są połączone kanałami. Każdy kanał tworzą dwie kolejki z regulaminem FIFO, z nieskończoną liczbą miejsc. Specyfikacje w Estelle celowo pomijają aspekty implementacji systemów komunikacyjnych. Jest to zgodne z ogólną zasadą przyjętą w pracach normalizacyjnych w zakresie współdziałania systemów otwartych — głównym obszarem zainteresowania jest zgodność (ang. compatibility) systemów, a nie zagadnienia implementacyjne. Cechami charakterystycznymi Estelle są: deklaracja stałej określonego typu bez podania jej wartości („any”), deklaracja zmiennej bez podania jej typu („...”), deklaracja funkcji lub procedury standardowej bez podania jej ciała („primitive”), deklaracja modułu bez podania jego ciała („external”), deklaracja podmodułów modułu-ojca, których kolejność wykonania nie jest określona (atrybut activity). Konstrukcje takie wystarczają do opisu, ale dla celów implementacji i symulacji muszą być uzupełnione, odpowiednio, podaniem wartości stałej, deklaracją typu, podaniem ciała dla funkcji i procedur standardowych, podaniem ciała modułu, określeniem kolejności wykonywania modułów. Wszystkie wymienione uzupełnienia stanowią szczególnie implementacji systemu komunikacyjnego i decydują o jego charakterystykach czasowych. Pojęcie czasu w specyfikacjach Estelle występuje jedynie w klauzulach „delayed(t1,t2)”, oznaczając warunki czasowe wykonania przejścia spontanicznego — po upływie t1 jednostek czasu od chwili spełnienia warunków przejścia, ale przed upływem t2 jednostek czasu. Dla celów budowy programu symulacyjnego konieczne jest przypisanie czasów przejścia dla każdej zmiany stanu modułów.

Implementacja adaptera komunikacyjnego wymaga podjęcia różnych decyzji projektowych, np. wyboru realizacji jednoprocessorowej lub wieloprocessorowej, doboru pamięci, ustalenia sposobu współpracy z siecią i z komputerem głównym, przyjęcia struktury oprogramowania (sposób tworzenia i wykrywania zdarzeń, komunikacja międzywarstwowa, zarządzanie pamięcią) itp. Czasy przejścia, zależne od szczegółów implementacji sprzętowo-programowej adaptera, mogą być oszacowane już na etapie projektowania adaptera — w przypadku realizacji sprzętowej przez bezpośrednie pomiary (np. analizatorem stanów), a w przypadku realizacji programowej przez wykonanie programu (np. za pomocą programu uruchomieniowego — debuggera). Program symulacyjny ma umożliwić ocenę wpływu decyzji projektowych na charakterystyki czasowe projektowanego adaptera. Jako podstawowe kryterium miary efektywności implementacji przyjmuje się przepustowość adaptera. Przy tworzeniu modelu symulacyjnego możliwe są dwa podejścia:

- użycie specjalizowanych programów symulacyjnych (np. pakietu symulatora rozszerzonych sieci kolejkowych QNAP [5]),

- napisanie programu symulacyjnego w języku symulacyjnym lub uniwersalnym języku wysokiego poziomu.

W pierwszym podejściu dostępne jest gotowe i przetestowane narzędzie, które jednak zazwyczaj posiada pewne ograniczenia utrudniające jego użycie w projektowaniu adaptera. Drugie podejście — omawiane w pracy — wiąże się z koniecznością samodzielnego opracowania i przetestowania narzędzia do symulacji, ale już dostosowanego do konkretnej specyfiki zadania projektowego. Wybrano język TopSpeed Modula-2 [6], mając na uwadze: podobieństwa składni i semantyki języków Estelle i Modula-2 (prekursorem obu języków był Pascal), możliwość programowania obiektowego, efektywny kompilator.

W rozdziale 3 przedstawiono na prostym przykładzie protokołu przesuwnego okna SWP (Sliding Window Protocol) sposób budowy programu symulacyjnego w języku Modula-2 na podstawie specyfikacji protokołu w Estelle.

3. Program symulacyjny dla protokołu SWP

3.1. Protokół SWP

Protokół SWP umożliwia jednokierunkowe przesyłanie danych z pozytywnym potwierdzeniem po każdym przesłaniu, z użyciem okien dla sterowania przepływem. Nadawca przesyła do odbiorcy ciąg wiadomości, które muszą być dostarczone bezbłędnie, z zachowaniem kolejności. Przesyłanie wiadomości odbywa się przy użyciu adapterów komunikacyjnych, pełniących funkcję nadajnika i odbiornika. Adaptery są połączone medium komunikacyjnym, które może powodować zniekształcenie, utratę, powtórzenie lub zmianę kolejności przesyłanych ramek. Zniekształcenia ramek są wykrywane przez adaptery w sposób niezawodny. Nadajnik wysyła do odbiornika ramkę informacyjną, zawierającą jedną wiadomość i numer sekwencyjny. Numer sekwencyjny jest zwiększany o jeden dla każdej nowej ramki. Numery sekwencyjne nie są ograniczone. Po wysłaniu ramki jest odliczany czas przerwy retransmisji (time-out). Odbiornik wysyła do nadajnika ramkę potwierdzającą po odebraniu każdej ramki informacyjnej. Ramka potwierdzająca zawiera numer sekwencyjny ostatniej ramki poprawnie odebranej i przekazanej do odbiorcy. Nadajnik utrzymuje okno nadawcze numerów sekwencyjnych, równe liczbie buforów nadawczych, w których wysłane wiadomości są przechowywane do czasu odebrania ich potwierdzenia przez odbiornik. Odbiornik utrzymuje okno odbiorcze numerów sekwencyjnych, równe liczbie buforów odbiorczych, w których są przechowywane wiadomości przed wysłaniem

ich do odbiorcy. Wartości czasu przerwy oraz wielkości okna nadawczego i odbiorczego zależą od implementacji. Specyfikacja protokołu SWP w Estelle jest podana w [7]. W specyfikacji przyjęto, że system komunikacyjny składa się z następujących oddzielnych, pracujących równolegle i asynchronicznie modułów: nadawcy, nadajnika, medium komunikacyjnego, odbiornika i odbiorcy.

3.2. Zasady implementacji protokołu SWP

W dalszych rozważaniach przyjęto następujące zasady implementacji protokołu SWP:

- adaptery działają w sposób sekwencyjny — nie ma przetwarzania równoległego (szczegóło realizacji sprzętowo-programowej pomijamy),
- adaptery mają identyczną strukturę oprogramowania — program sterujący (scheduler) na podstawie analiz kolejek wejściowych powołuje odpowiednie procedury, realizujące protokół komunikacyjny.

Na podstawie specyfikacji systemu komunikacyjnego w Estelle, uwzględniając powyższe zasady implementacji, należy zbudować program symulacyjny w języku Modula-2, umożliwiający ocenę przepustowości systemu, mierzonej liczbą bajtów przesłanych w jednostce czasu od nadawcy do odbiorcy. Wymagana jest możliwość porównania różnych algorytmów analiz kolejek przez programy sterujące adapterów.

3.3. Zasady przejścia od specyfikacji do programu symulacyjnego

Uwzględniając specyfikę Estelle, jest rzeczą naturalną użycie techniki programowania obiektowego. W programie symulacyjnym deklarowane są dwie klasy bazowe: EstelleChannel i EstelleModule. Deklaracja klasy jest deklaracją typu rekordowego wraz z procedurami, zwanymi metodami. Deklaracjom kanałów w specyfikacji Estelle odpowiadają deklaracje klas pochodnych klasy bazowej EstelleChannel. Kanał jest reprezentowany przez dwie kolejki o długościach dobieranych eksperymentalnie podczas badań symulacyjnych tak, aby nie wystąpiło przepelnienie kolejek. Do obsługi każdej z kolejek służą procedury, umożliwiające: podanie stanu określonego miejsca kolejki, dodanie interakcji do kolejki i usunięcie interakcji z kolejki. Deklaracjom modułów w specyfikacji Estelle odpowiadają deklaracje klas pochodnych klasy bazowej EstelleModule. Punkty dostępu modułów są zmiennymi wskaźnikowymi do odpowiednich kolejek kanałów. Utworzeniu reprezentacji (ang. instance) modułu w Estelle odpowiada utworzenie obiektów odpowiednich klas pochodnych. Jako metodę symulacji wygodnie jest wybrać planowanie zdarzeń

[8]. Ze względu na potrzebę symulacji wielu równoległe działających obiektów, konieczne jest przyjęcie rozszerzonego stanu modułów, obejmującego zmienne podane w specyfikacji oraz dodatkowo nazwy przejścia i liczniki czasów przejścia. Część główna programu symulacyjnego traktuje każdy z symulowanych obiektów jako izolowaną część programu, jedynymi dostępnymi zmiennymi są liczniki czasu przejścia dla obiektów klas pochodnych klasy EstelleModule. Szczegółowe rozważania dotyczące przejścia od specyfikacji systemu komunikacyjnego w Estelle do programu w języku Modula-2, symulującego implementację tego systemu oraz pełne wyniki badań symulacyjnych, są podane w pracy [9].

3.4. Testowanie programu symulacyjnego

Testowana była poprawność działania programu symulacyjnego. Przyjęta strukturyzacja programu umożliwiła dekompozycję zadania — oddzielne testowanie każdego z modułów razem z kanałami, z którymi jest połączony. Ze względu na mały wymiar przestrzeni stanów, było możliwe pełne przetestowanie każdego z modułów przy użyciu wszystkich możliwych sekwencji testujących dla wszystkich możliwych stanów modułów. Dzięki takiej metodzie testowania udało się wykryć kilka błędów w specyfikacji podanej w [7], powodujących zakleszczenie systemu komunikacyjnego.

3.5. Badania przepustowości

Badania przeprowadzono przy założeniu następujących czasów przejść (wartości przykładowe w ms dla wiadomości wielkości 1 kB):

- przesłanie wiadomości od nadawcy do bufora nadawczego, wysłanie ramki informacyjnej, odebranie ramki informacyjnej i zapis do bufora odbiorczego, przekazanie wiadomości do odbiorcy — 1,
- pozostałe akcje oraz działanie programów sterujących adapterów — 0,1. Badany był wpływ wartości czasu przerwy, wielkości okna nadawczego, okna odbiorczego i algorytmów działania programów sterujących na przepustowość systemu komunikacyjnego.

Maksymalna przepustowość dla bezbłędnego ośrodka transmisji wynosi 500 kB/s. Badania przeprowadzono dla ośrodka transmisji z prawdopodobieństwem błędnego przesłania ramki informacyjnej i potwierdzającej równym 0,15 (wartość przykładowa). Dla algorytmów programów sterujących wynikających bezpośrednio ze specyfikacji maksymalna przepustowość wyniosła 269 kB/s. Dla algorytmów adaptacyjnych (pełna analiza stanu modułu o kolejek) maksymalna przepustowość wyniosła 332 kB/s; dla algorytmów adaptacyjnych przyjęto zwiększony dwukrotnie czas działania programów sterujących.

Program symulacyjny był również użyty do optymalizacji protokołu SWP. Stwierdzono, że zmiana w specyfikacji protokołu SWP, polegająca na ograniczeniu retransmisji po upływie czasu przerwy jedynie do ramki przeterminowanej (bez retransmisji ramek wysłanych później), umożliwia zwiększenie maksymalnej przepustowości do 319 kB/s.

4. Wnioski, kierunki dalszych prac

Stosowanie narzędzi symulacji komputerowej na etapie projektowania adapterów umożliwia podejmowanie odpowiednich decyzji implementacyjnych, koniecznych do uzyskania właściwych charakterystyk czasowych systemów komunikacyjnych. Szczególnie ważna jest możliwość badania wpływu algorytmów programów sterujących adapterów na przepustowość w sytuacji realizacji przez adapter wielu warstw komunikacyjnych modelu OSI, wobec znanych uwag krytycznych w odniesieniu do efektywności implementacji sieci otwartych. Wymaganie czasooptymalnej realizacji protokołów komunikacyjnych przez adapter jest kluczowym zagadnieniem, szczególnie dla szybkich sieci lokalnych HSLAN (ang. High Speed LAN), które są obecnie przedmiotem intensywnych prac badawczych i normalizacyjnych [10]. Wobec dużej szybkości transmisji w sieciach HSLAN (rzędu 1 Gb/s), adaptory komunikacyjne mogą stanowić „wąskie gardła”. Powiększenie przepustowości adapterów jest możliwe do uzyskania, między innymi, dzięki realizacjom wieloprocesorowym (przetwarzanie równoległe).

Metoda zaprezentowana w pracy może stanowić podstawę do opracowania programów tłumaczących specyfikację systemu komunikacyjnego w Estelle na program symulacji pracy systemu.

LITERATURA

- [1] Dean S.M., et al.: CONE: A software environment for network protocols, Hewlett-Packard Journal 90(2), 18-28, 1990.
- [2] Sidhu D.P., Blumer T.P.: Semi-automatic implementation of OSI Protocols, Comput. Networks ISDN Systems 18(3), 221-238, 1990.
- [3] Bochmann G., Protocol specification for OSI, Comput. Networks ISDN Systems 18(3), 167-184, 1990.

- [4] ISO 9074, Estelle: A formal description technique based on an extended transition model, 1989.
- [5] Dembiński P.: Queuing network model for Estelle, FORTE'92 — Fifth international conference on formal description techniques, 83–97, Lannion, France.
- [6] Jensen & Partners International: TopSpeed Modula-2, 1991.
- [7] ISO/IEC TR 10167, Guidelines for the application of Estelle, LOTOS and SDL, 1991.
- [8] Tyszer J.: Symulacja cyfrowa, WNT, Warszawa 1990.
- [9] Kacprzak M.: Rozprawa doktorska (w przygotowaniu). Politechnika Śląska.
- [10] Spaniol O., Danthine A. (ed.): High Speed Local Area Networks II, North-Holland, 1990.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do redakcji 23 września 1993r.

Abstract

As an example, the proposed method for generation of controller's simulation model is applied for performance evaluation of simple Sliding Window Protocol implementation. Several inconsistencies of SWP specification were reported while testing the simulation model. Some computational results of predicted controller's performance are given. Possible improvement of SWP for better performance is discussed.