

PROTOKÓŁ IPv6 W MOBILNYCH SYSTEMACH DOSTĘPOWYCH

**Lukasz Czekierda
Dominik Radziszowski
Krzysztof Zieliński *
Sławomir Zieliński**

**Akademia Górniczo-Hutnicza - Katedra Informatyki,
ACK Cyfronet AGH*
ul. Mickiewicza 30
30-059 Kraków**

Streszczenie

Celem niniejszego opracowania jest zaprezentowanie prac prowadzonych w Katedrze Informatyki AGH w ramach 5 Programu Ramowego Unii Europejskiej dotyczących tworzenia i dostosowywania aplikacji do środowiska bezprzewodowego opartego na protokole IPv6 (projekt 6WINIT). Artykuł zawiera krótkie omówienie cech tego protokołu oraz powodów, dla których konieczne jest wprowadzenie nowego środowiska. W dalszej części opisano platformy sprzętowe i systemowe wykorzystywane w projekcie 6WINIT jak również architekturę systemów, nad którymi prowadzone są prace. Przedstawiono także wyniki przeprowadzonych testów.

1. Wstęp

Rozwój sieci Internet jest obecnie uwarunkowany dostępnością publicznych adresów IP koniecznych dla wciąż bardzo szybko rosnącej liczby urządzeń podłączonych do sieci. Jest to szczególnie ważne w kontekście mobilnych urządzeń dostępowych do sieci Internet takich jak urządzenia PDA lub telefony komórkowe nowej generacji, które z roku na rok stają się coraz bardziej popularne, gdyż komunikacja z wykorzystaniem technologii WLAN, GPRS czy UMTS, jaką używają, bądź będą niebawem używać te urządzenia, wiąże się z koniecznością alokacji dla końcowego terminala publicznego adresu IP. Rozwiązanie problemu ograniczonej przestrzeni adresowej oraz usprawnienie procedury alokacji adresów warstwy sieciowej wymaga przejścia od stosowania protokołu IPv4 na eksploatację protokołu IPv6.

Chociaż protokół IPv6 został opracowany przez IETF [10] na początku lat dziewięćdziesiątych, wciąż nie nastąpiło jego upowszechnienie. Pogłębiający się problem braku adresów IPv4, szczególnie w Europie Zachodniej oraz perspektywa wprowadzenia telefonii 3G i 4G, spowodowały pewne przyśpieszenie prac nad wdrażaniem protokołu IPv6. Także wzrost możliwości mobilnych urządzeń dostępowych pozwala obecnie na instalację stosu IP oraz typowych aplikacji internetowych.

W niniejszym artykule przedstawiono prace realizowane w Katedrze Informatyki AGH i Akademickim Centrum Komputerowym Cyfronet w ramach projektu 5 Programu Ramowego Unii Europejskiej pt. „Wireless IPv6 Internet Initiative – 6WINIT” koordynowanego przez UCL London. W projekcie tym bierze udział szereg czołowych firm, takich jak Ericsson, VITT, British Telecom, czy IABG oraz uniwersytetów, jak np. Stuttgart, Bremen oraz Southampton.

Pierwsza część artykułu obejmuje krótkie omówienie własności protokołu IPv6. Następnie zaprezentowano platformy systemowe i oprogramowanie korzystające z IPv6 dostępne obecnie na urządzeniach mobilnych. W kolejnym punkcie przedstawiono architektury aplikacji rozproszonych dla zastosowań mobilnych. Przykłady aplikacji zrealizowanych w Katedrze Informatyki AGH w trakcie pierwszego roku projektu 6WINIT omówiono w ostatnim punkcie prezentacji. Całość artykułu podsumowano wnioskami dotyczącymi perspektyw rozwoju platform tworzenia aplikacji dla protokołu IPv6 i podstawowych trudności w tym zakresie.

2. Zagadnienia związane z IPv6

Protokół IPv6 powstał na bazie powszechnie obecnie stosowanego protokołu IPv4, jego twórcy pragnęli by nowa wersja nie posiadała wielu poważnych ograniczeń protokołu wersji czwartej. Największy nacisk położono na rozwiązanie problemów:

- ograniczonej przestrzeni adresowej. Trwający od kilku lat „kryzys” stał się głównym czynnikiem motywującym do poszukiwań nowego rozwiązania.
- braku odpowiednio silnych mechanizmów bezpieczeństwa. Od nowej wersji protokołu oczekuje się zapewnienia funkcji związanych z zabezpieczeniem sieci udostępnianych dotychczas przez protokoły wyższych warstw.
- braku odpowiednio elastycznych mechanizmów autokonfiguracji. Mimo istnienia w IPv4 protokołów takich, jak DHCP, konfiguracja nadal nie jest realizowana w sposób wystarczająco elastyczny. Zagadnienie to nabiera szczególnego znaczenia w kontekście urządzeń mobilnych, które w czasie pracy mogą przemieszczać się pomiędzy obszarami zasięgu różnych punktów dostępowych.

Stosowanie od wielu lat i na szeroką skalę IPv4 dostarczyło wielu doświadczeń, które można było wykorzystać w procesie tworzenia nowej wersji tego protokołu. Warto tu wspomnieć choćby o znikomej liczbie producentów obsługujących różne rodzaje transmisji (definiowane przez pole TOS (ang. *type of service*))¹, czy też o odmiennej, w stosunku do zamierzeń, obsłudze czasu życia pakietu – zamiast mierzenia ilości sekund, jaką pakiet przebywa w routerach, urządzenia te zmniejszają wartość pola TTL (ang. *time to live*) o jeden.²

Należy także zauważyć, że tablice routingu routerów szkieletowych często mają rozmiary nawet rzędu 100000 wpisów. Skutkiem tego są znaczne opóźnienia w przesyłaniu pakietów. Jak twierdzi P. Loshin, wpływ wzrostu rozmiarów tablic routingu na tempo rozwoju Internetu jest znacznie większy, niż efektu wyczerpywania adresów.³ Jedną z

¹ Brak obsługi różnych typów transmisji w dużym stopniu wynikał z braku sprecyzowania, w jaki sposób ta obsługa miałaby być implementowana.

² Nie stoi to w sprzeczności z zamierzonym zastosowaniem pola; służy ono – tak, jak zamierzano – do usuwania pakietów wpadających w pętlę routingu.

³ P. Loshin, IPv6 Clearly Explained, Academic Press, San Francisco, 1999, s. 43

oczekiwanych własności nowej specyfikacji IP jest możliwość bardziej efektywnej agregacji adresów.

Każda modyfikacja specyfikacji protokołu – zwłaszcza tak ważnego, jak IP, musi pociągnąć za sobą istotne zmiany w oprogramowaniu lub konfiguracji sprzętowej urządzeń obsługujących ten protokół. Zebranie dużej ilości doświadczeń pozwalającej na podjęcie decyzji o usunięciu niektórych – nieużytecznych – właściwości IPv4 stanowi ważny argument na korzyść wymiany protokołu, zamiast dokonywania poprawek, które mogłyby polegać np. wyłącznie na zwiększeniu liczby adresów, czy dodaniu nowej funkcjonalności poprzez zdefiniowanie nowych opcji nagłówka itp.

Wskutek zebranych doświadczeń, IPv6 został skonstruowany tak, aby mógł być obsługiwany przez urządzenia sieciowe sprawniej. Zasadnicze zmiany w stosunku do poprzedniej wersji dotyczyły:

- zwiększonej przestrzeni adresowej. Oprócz powiększenia przestrzeni adresowej (długość adresu wzrosła z 32 do 128 bitów), zapewniono taką jej organizację, aby zmniejszyć rozmiary tablic routingu.⁴ Zlikwidowano możliwość transmisji rozgłoszeniowej (ang. *broadcast*), wprowadzono natomiast nowy rodzaj adresacji, odnoszący się do najbliższego odpowiadającego urządzenia (ang. *anycast*).
- uproszczenia nagłówka pakietu. Nagłówek IPv6 cechuje się – inaczej niż w poprzedniej wersji – stałą długością, co upraszcza jego obsługę. Zrezygnowano także np. z możliwości dokonywania fragmentacji pakietu przez urządzenia pośredniczące w transmisji; w nowej wersji protokołu podzielenia pakietu może dokonać wyłącznie urządzenie źródłowe.
- usprawnienia obsługi rozszerzeń i opcji. Wszelkie opcje zawierane są w osobnych – opcjonalnych – nagłówkach, które nie zmieniają rozmiaru nagłówka pakietu.
- etykietowania strumieni danych. Wprowadzenie oznaczania strumieni pakietów pozwala na łatwiejszą implementację jednakowej obsługi każdego pakietu ze strumienia. Brak konieczności analizy wszystkich nagłówków z osobna umożliwia skrócenie czasu ich przebywania w routerze. Warto zauważyć, że o ile w przypadku zastosowań takich, jak pobieranie pliku ze zdalnego serwera, czy odtwarzanie strumienia wideo opóźnienie nie ma większego znaczenia, o tyle w przypadku nowoczesnych aplikacji interaktywnych (wideokonferencje, telefonia IP) nawet niewielkie opóźnienie ma znaczący wpływ na komfort pracy z aplikacją. Przykładem trudnego do spełnienia wymagania jest zalecenie, aby całkowite opóźnienie transmisji dźwięku między nadawcą a odbiorcą nie było większe niż 100 ms.
- autentykacji i zapewnienia poufności danych. IPv6 korzysta z dwóch mechanizmów związanych z bezpieczeństwem transmisji: obliczania sum kontrolnych⁵ (za pomocą uważanych za obliczeniowo bezpieczne algorytmów, np. MD5) w celu ustalenia, czy nie zostały one zmodyfikowane oraz szyfrowania zawartości pakietu.⁶

⁴ por. RFC 2373

⁵ Nie należy mylić tego rodzaju sum kontrolnych z występującymi w IPv4. Wykorzystanie tego mechanizmu nie jest obligatoryjne, a jedynie umożliwiane przez z opcjonalnych nagłówków (AH, ang. *authentication header*). Definicja tego nagłówka znajduje się w RFC 1826.

⁶ Zaszyfrowanie zawartości pakietu pozwala np. na tworzenie wirtualnych sieci prywatnych (VPN). Za pomocą mechanizmu ESP (ang. *encapsulated security payload*) – opisanego w RFC 1827 – można ukryć nie tylko treść komunikacji, ale także szczegóły topologii sieci korporacyjnej.

Rezultatem wymiany protokołu warstwy sieciowej jest konieczność redefinicji również innych – zależnych, bądź współpracujących z IP – protokołów. Dobrym przykładem mogą być ICMP, stosunkowo prosty protokół służący do podstawowej diagnostyki pracy sieci⁷, oraz IGMP – realizujący zadania związane z organizacją transmisji grupowej.

Zmiana wersji protokołu IP dokonuje się stopniowo. Implikuje to konieczność obsługi dwóch różnych protokołów warstwy sieciowej przez urządzenia podłączone do Internetu, co stanowi niebagatelny problem dla producentów oprogramowania. Maszyny obsługujące dwie wersje protokołu IP muszą zapewniać także komunikację w wyższych warstwach, tzn. obsługiwać dwa stosy protokołów. Występuje również konieczność dublowania usług takich, jak np. DNS.

Sieci IPv6 (obejmujące na razie swoim zasięgiem znacznie mniej maszyn, niż oparte na IPv4) są łączone między sobą za pomocą tzw. tuneli – umieszczania pakietów IPv6 wewnątrz ramek IPv4 transmitowanych pomiędzy urządzeniami nie potrafiącymi komunikować się za pomocą nowej wersji protokołu. Inną techniką, wykorzystywaną do zapewnienia poprawnej współpracy pomiędzy maszynami obsługującymi różne wersje protokołu, jest użycie translatorów zmieniających w sposób niezauważalny dla maszyn końcowych format ramki, co jednakże jest procesem złożonym i mogącym wprowadzać znaczące opóźnienia.

Pewne jest, że jeszcze przez długi czas w użyciu będą obydwie wersje IP, choć czynniki takie, jak brak „wolnych” adresów będą skłaniały zarówno użytkowników, jak i producentów oprogramowania do stosowania nowego protokołu. O tempie przechodzenia użytkowników sieci na IPv6 będzie decydować przede wszystkim ilość aplikacji wykorzystujących ten protokół do komunikacji ze światem zewnętrznym.

3. Platformy sprzętowe i systemowe IPv6 urządzeń mobilnych

W niniejszym punkcie zostaną przedstawione obecnie dostępne rozwiązania sprzętowe przenośnych urządzeń dostępowych typu PDA oraz zarówno systemowe jak i aplikacyjne oprogramowanie, mogące pracować na tych urządzeniach wykorzystując IPv6.

3.1. Platformy sprzętowe

Urządzenia mobilne z rodziny *PDA (Personal Digital Assistant)*⁸ stają się coraz bardziej niezbędnymi narzędziami pracy współczesnych managerów. Ten mały, lekki komputer, składający się z wyświetlacza, kilku przycisków i rysika nie jest już tylko wyposażony w terminarz, listę kontaktów, kalkulator; zawiera dodatkowo arkusz kalkulacyjny, edytor tekstów i przeglądarkę internetową. W ostatnich latach obserwujemy bardzo dynamiczny rozwój tych urządzeń – coraz lepsze wyświetlacze, szybsze procesory powodują, że PDA przejmuje coraz więcej funkcji zwykłego komputera, a parametrami technicznymi nie odbiega od tych, które z dumą zajmowały nasze biurka jeszcze kilka lat temu. Obecnie konkurują ze sobą następujące modele PPC (*Pocket PC*): iPAQ serii H3600 (produkt firmy Compaq), Palm serii V, HP Jornada i Cassiopeia. Są one wyposażane, w zależności od producenta, w system operacyjny PalmOS, WindowsCE, Epoc bądź Symbian.

⁷ por. RFC 2463

⁸ PDA - podręczne komputery, których system operacyjny jest przystosowany do współpracy z elektronicznym pisakiem - słownik informatyczny angielsko-polski.

Większość producentów oferuje obecnie urządzenia wyposażone w kolorowe lub monochromatyczne wyświetlacze o wymiarach 320×240 do 640×480 pikseli, pamięć operacyjną 4÷64MB, Flash ROM 2÷16 MB, port szeregowy i IrDA oraz w zależności od producenta procesor RISC Intel StrongARM 206 MHz (iPAQ), Hitachi SH3 (HP Jornada), NEC VR4122 (Cassiopeia). Przeprowadzone przez autorów w połowie roku 2001 porównania dostępnych modeli pod względem parametrów technicznych oraz dostępnych rozszerzeń zdecydowanie wskazywały na model iPAQ H3660 jako najbardziej dojrzały PPC. Dostępne dziś rozwiązania proponowane przez innych producentów wydają się być już jednak w pełni konkurencyjne.

W swojej podstawowej konfiguracji PDA ma stosunkowo niewielkie możliwości komunikacyjne, dopiero wyposażony w moduły rozszerzeń staje się on pełnowartościowym urządzeniem z możliwością zainstalowania jednej lub dwóch kart PCMCIA, dodatkowej pamięci typu Flash lub czytnika kodów kreskowych. Uwaga autorów skupiła się tu na możliwości podłączenia bezprzewodowych kart sieciowych standardu 802.11b, zestawów GSM/GPRS oraz kart pamięci.

3.2. Linux dla komputerów PPC

Pomimo, iż specyfikacja protokołu IPv6 jest dostępna już od kilku lat, jednak obecnie obsługują go głównie systemy uniksowe (Linux, FreeBSD, Solaris), firma Microsoft stworzyła jedynie dodatki do systemów Windows2000 i NT4.0, obie w wersji beta.

Jak dotąd również żaden z systemów operacyjnych dostarczanych standardowo przez producenta razem z PDA nie obsługuje IPv6. Jedynym sposobem, by protokół ten był dostępny na tej platformie jest instalacja systemu Linux. Podobnie jak w wypadku komputerów PC, istnieje co najmniej kilka dystrybucji systemu Linux dla tych komputerów. Do niedawna najbardziej znaną była stworzona przez firmę Comapaq - *hh* (od handhelds.org). Na jej bazie powstawały kolejne, wraz z najbardziej dziś popularnymi dystrybucjami, *Familiar* i *Imitiate*. Zawierają one między innymi okrojoną wersję pakietu *Python*, *Xfree86* z managerem okienek *BlackBox* oraz nowy system zarządzania pakietami *ipkg*, [7].

W porównaniu z pozostałymi systemami, Linuksa cechuje duża elastyczność w zakresie wyboru instalowanych aplikacji i narzędzi. Można dowolnie wybierać spośród kilku menadżerów okienek (QtEmbedded, BlackBox), przeglądarek internetowych (Konqueror, ViewML, Dillo), edytorów tekstu (Pyditor, Ion). Bardzo ważną dla użytkownika jest prostota obsługi środowiska okienkowego oraz zwiększająca się lawinowo liczba użytecznych aplikacji. Administratorowi oferowane są możliwości porównywalne ze zwykłą stacjonarną instalacją. Całość cechuje stabilność oraz bardzo



Rys. 1 Przeglądarka WWW w środowisku Linux na

przydatna cecha, znana z każdego systemu Unix, zdalnej pracy, usprawniającej administrowanie systemem.

3.3. Tworzenie aplikacji na PDA

Duża liczba dostępnych w kodzie źródłowym aplikacji i przyzwyczajenia programistów preferujących C/C++, są powodem dla którego większość programów, tworzonych bądź adaptowanych na platformę Linux/iPAQ, jest napisana w tym języku.

W celu stworzenia własnej aplikacji w języku C/C++ na urządzeniu PDA niezbędne jest zaadaptowanie posiadanego na stacjonarnej maszynie uniksowej kompilatora dla potrzeb kompilacji do kodu wykonywanego na innym procesorze (ang. crosscompilation). Nie można bowiem, z powodu bardzo ograniczonych zasobów, wyposażyć PDA we własny kompilator. Odpowiednio skompilowany i zlinkowany program jest, po przeniesieniu, gotowy do działania. Należy jednak podkreślić, że tak przygotowany program będzie działał jedynie na PDA z procesorem dla którego został skonfigurowany kompilator, ewentualna instalacja na innych urządzeniach wymaga kolejnej adaptacji kompilatora i ponownej rekompilacji.

Stosowanie języka C/C++ w aplikacjach niskopoziomowych zdaje się być nieuniknione, aplikacje użytkowe dużo wygodniej i efektywniej pisać jednak w języku Java.

Do najważniejszych zalet języka Java, oprócz łatwości pisania oprogramowania, należy jego przenośność pozwalająca uruchomić skompilowany już kod na dowolnym urządzeniu wyposażonym w maszynę wirtualną tego języka. Ta cecha czyni język Java bardzo przydatnym do tworzenia aplikacji projektowanych do pracy na różnych rodzajach urządzeń, a urządzenia mobilne charakteryzują się właśnie dużym zróżnicowaniem platform sprzętowych. Ułatwiona jest także faza budowy oprogramowania, gdyż jego testy można prowadzić na stacjonarnym komputerze, a dopiero końcową wersję przenieść na urządzenie mobilne.

Java, począwszy od wersji 1.4, która w postaci stabilnej została udostępniona przez firmę Sun w lutym 2002 r., pozwala na obsługę protokołu IP w wersji 6. Niestety w chwili pisania tego artykułu nie istniało jeszcze żadne środowisko uruchomieniowe pozwalające na obsługę aplikacji dla urządzeń mobilnych pracujących w sieciach opartych na z IPv6. Jak dotąd na większość platform sprzętowo-programowych opracowano jedynie środowiska uruchomieniowe dające ograniczoną funkcjonalność (Java Micro Edition lub Personal Java) projektowane specjalnie na urządzenia o małej ilości zasobów, chociaż nowsze modele urządzeń klasy PDA posiadają już parametry pozwalające na uruchamianie dużo bardziej rozbudowanych aplikacji. Przykładowo nie powstała jak dotąd wersja zgodna z JDK 1.2 dla platformy Windows CE pracującej m.in. na urządzeniach PDA iPAQ firmy Compaq, natomiast te same urządzenia pracujące pod kontrolą systemu Linux mogą już korzystać z maszyny wirtualnej JDK1.3 kontrolowanej przez koncern Sun firmy Blackdown. Jak zostało zauważone wcześniej, system Windows CE jak dotąd nie obsługuje protokołu IPv6⁹. Wspomniana już cecha przenośności języka daje jednak możliwość rozwijania aplikacji na innej platformie i po pojawieniu się stosownego środowiska na niemal natychmiastowe przeniesienie jej na urządzenie mobilne.

⁹ Trwają prace nad wersją Windows CE 2002, który ma obsługiwać protokół IPv6, jednak nie jest on obecnie dostępny.

4. Architektury aplikacji rozproszonych w zastosowaniach mobilnych

Specyfika urządzeń PDA rodzi wiele problemów – nowych lub zapomnianych już w świecie komputerów stacjonarnych. Wymaga to zarówno tworzenia aplikacji dedykowanych dla konkretnej platformy, a często i modyfikacji architektur systemów już istniejących.

4.1. Wymagania interfejsów użytkownika

Urządzenia PDA – poza swoimi niewątpliwymi zaletami jakimi są małe rozmiary – posiadają wynikające z nich ograniczenia, które mogą być dość dotkliwe zarówno dla użytkowników jak i programistów. Dają się one we znaki szczególnie w interakcji z użytkownikiem. Problemem jest mały wyświetlacz (320x240 pikseli), który musi zaoferować funkcjonalność porównywalną z tą jaką uzyskujemy na laptopach czy komputerach stacjonarnych. W wielu aplikacjach często nie wystarczy proste przeskalowanie interfejsu czy dodanie pasków przewijania (por. Rys 1). Najczęściej programista musi całkowicie reorganizować graficzny interfejs użytkownika. W wypadku aplikacji dostępnych przez WWW dodatkowo problemem są mniejsze możliwości przeglądarek pracujących na PDA. Microsoft Internet Explorer w wersji CE nie obsługuje ramek, dostępny pod Linuksem Konqueror ma problemy z przeskalowaniem napisów i przycisków. Drugi poważny problem to brak klawiatury, a więc trudność z szybkim wprowadzaniem danych. Tu z rozwiązaniem przychodzi zdolność PDA do rozpoznawania znaków rysowanych na wyświetlaczu rysikiem, dzięki której po kilku dniach treningu można już dość sprawnie komunikować się z tym urządzeniem. Alternatywą jest programowa klawiatura, na której pokazujemy znaki jakie chcemy wprowadzić – tu jednak szybkie pisanie jest niemożliwe. Dodatkowo tworzone aplikacje powinny maksymalnie upraszczać swoje interfejsy, stosując pola jedno i wielokrotnego wyboru, listy itp., tak by ograniczyć konieczność wpisywania tekstów.

Dodatkowo należy pamiętać, że PDA ma stosunkowo małą ilość pamięci, nie tworzy pliku wymiany, w związku z czym należy baczniej obserwować problemy z jej alokacją w odpowiednich miejscach kodu. Komputer taki nie posiada również jednostki zmiennopozycyjnej, zatem na etapie projektowania aplikacji należy rozważyć możliwość zdalnego wykonywania zadań wymagających złożonych operacji na maszynie o większych możliwościach obliczeniowych.

4.2. Architektura systemu dostępu do danych

Wspomniane wcześniej ograniczenia związane z urządzeniami mobilnymi implikują konieczność zastosowania zupełnie innej niż tradycyjna techniki tworzenia aplikacji mających funkcjonować na takich urządzeniach w systemie rozproszonym. Tradycyjne podejście oparte było o model dwuwarstwowy, w którym klient komunikujący się z serwerem zajmował się kwestiami komunikacji oraz przetwarzania i wyświetlania otrzymanych wyników. Nowoczesne pod względem architektury systemy rozproszone (oczywiście można i należy stosować je także w tradycyjnych środowiskach) posiadające strukturę trójwarstwową ograniczają zbiór zadań stawianych klientowi. Najniższa warstwa skupia się na przechowywaniu danych, jest nią zazwyczaj baza danych w zależności od potrzeb obiektowa lub relacyjna, zoptymalizowana pod kątem odczytu lub zorientowana na zrównoważone sposoby dostępu. Warstwę drugą, zajmującą się przetwarzaniem danych,

budują najczęściej w nowoczesnych architekturach rozproszonych systemy komponentowe, takie jak Enterprise JavaBeans [12]. Warstwa ta, oprócz kwestii związanych z bezpieczeństwem dostępu do danych oraz zapewnieniem transakcyjności koncentruje się także na dostarczeniu interfejsu warstwie wyższej, a więc w konsekwencji końcowemu użytkownikowi. W kontekście zastosowań mobilnych to zadanie ma szczególne znaczenie, gdyż najczęściej informacja dostarczana do urządzeń przenośnych musi mieć inną postać niż dla urządzeń stacjonarnych. Właściwa konstrukcja warstwy przetwarzania pozwala na bardzo łatwe dodawanie nowych typów interfejsów bez ingerowania w dotychczas istniejące.

Jednym z najpopularniejszych rozwiązań jest obecnie rozdzielenie semantyki informacji od sposobu jej prezentacji. Do opisu struktury danych używany jest język XML, zaś do ich prezentowania można używać bardzo wielu metod, wśród których najważniejsze miejsce zajmuje język HTML używany przez przeglądarki WWW. Strony HTML mogą mieć oczywiście zróżnicowaną postać zależną np. od wielkości wyświetlacza lub jego możliwości graficznych. Inną metodą prezentacji może być sformatowanie danych do postaci akceptowalnej przez telefony komórkowe (a więc WML).

5. Przykładowe aplikacje IPv6

W ramach ciągle trwającego grantu 6WINIT zostało opracowanych kilka aplikacji korzystających z protokołu IPv6, bądź tworzonych od początku lub też dostosowanych do współpracy z tym protokołem. W dalszej części tego punktu zostaną opisane dwie z nich, jedna zapewniająca dostęp do radiologicznej bazy danych, druga służąca do przeprowadzania wideokonferencji.

5.1. NetRAAD

Graficzne wyniki badań medycznych, takich jak zdjęcia radiologiczne czy ultrasonografia, ze względu na swoją specyfikę i zastosowanie nie mogą być przechowywane przy użyciu formatów ogólnego przeznaczenia (takich jak GIF czy JPEG) wymagając istnienia specjalnego formatu.

Standardem powszechnie akceptowanym i szeroko stosowanym w bardzo wielu ośrodkach na całym świecie jest DICOM (Digital Imaging and Communication in Medicine) [9].

Plik zgodny z tym standardem może zawierać obraz statyczny – na przykład zdjęcie tomograficzne – lub animację przedstawiającą dynamiczne procesy pacjenta, jak na przykład badanie hemodynamiczne. Oprócz informacji graficznej plik taki zawiera szereg



Rysunek 2 Dostęp do wyników badań radiologicznych przy użyciu systemu NetRAAD

szczegółowych informacji – identyfikowanych przez specjalne pola (*ang.* tag) – dotyczących pacjenta oraz przeprowadzonego badania. Część tych danych jest wykorzystywana przez przeglądarki w celu właściwego zaprezentowania obrazu, inne są jedynie wyświetlane i mogą być pomocne np. dla lekarza dokonującego diagnozy. Popularność standardu spowodowała, że istnieje bardzo wiele przeglądarek DICOM o bardzo różnym stopniu zaawansowania przeznaczonych na komputery stacjonarne, choć możliwe też jest wyświetlanie plików DICOM na komputerach klasy PDA (np. przy użyciu oprogramowania reViewMD o stosunkowo dużych możliwościach opracowanego przez firmę Digisoft). Należy zwrócić uwagę, że systemy te w zdecydowanej większości obsługują jedynie pliki przechowywane lokalnie. Notowany w ostatnich latach gwałtowny rozwój telemedycyny przyczynił się jednak do tego, że coraz więcej informacji medycznej powinno być w nowoczesny sposób dostępne poprzez sieć komputerową.

System NetRAAD pracujący w szpitalu im. Jana Pawła II w Krakowie opracowany przez firmę Uniwersytecka Opieka Zdrowotna z Lublina pozwala na dostęp do radiologicznej bazy danych szpitala poprzez WWW (patrz rys. 2). (Dzięki dostosowaniu serwera HTTP do obsługi IPv6, akceptuje on żądania w obu wersjach protokołu.) Baza ta oprócz informacji o personaliach pacjenta zawiera wyniki badań przechowywane w standardzie DICOM. Strony WWW generowane przez system NetRAAD dobrze nadają się do wyświetlania na stacjonarnych komputerach, jednakże konstrukcja tych stron nie pozwala na przyjazne użytkownikowi prezentowanie ich zawartości na urządzeniach klasy PDA, które mogą być bardzo przydatne dla lekarzy na przykład w czasie ich pracy poza szpitalem. Zapewnienie požądanej funkcjonalności wymagało zatem stworzenia nowego interfejsu graficznego przy zachowaniu systemu dostępu do danych umieszczonych w bazie.

Aplikacja została zaimplementowana w języku Java. Do samej obsługi plików DICOM wykorzystano specjalistyczne API Java DICOM Toolkit belgijskiej firmy Softlink. Do komunikacji z serwerem aplikacja używa jedynie standardowego portu protokołu HTTP, co pozwala na swobodną komunikację nawet w przypadku wprowadzenia administracyjnych ograniczeń dostępu (które omijają zazwyczaj usługę WWW). Ze względu na charakter aplikacji – korzystanie z bazy danych – zapytania kierowane do serwera używają języka SQL, są one natomiast wysyłane przy użyciu metody POST protokołu HTTP do pełniącego rolę pośrednika *serwletu* pracującego na serwerze Tomcat. Wykorzystując mechanizm JDBC kieruje on w imieniu przeglądarki żądanie do bazy danych, a zwracane wyniki w postaci zserializowanego obiektu języka Java są wysyłane z powrotem do aplikacji. Także wszystkie pliki DICOM oraz mocno skompresowane pomocnicze pliki referencyjne są transferowane jedynie przy użyciu protokołu HTTP.

Projekt aplikacji był bardzo mocno uwarunkowany koniecznością obsługi protokołu IP w wersji 6. Dzięki użyciu języka Java w wersji JDK 1.4 możliwa jest komunikacja przy użyciu protokołu IP w wersji zarówno 4,



Rysunek 3 Dostęp do wyniku badania radiologicznego na urządzeniu iPAQ

jak i 6. W celu umożliwienia obsługi protokołu IPv6 także po stronie serwera, wszystkie zapytania kierowane są bezpośrednio do głównego serwera WWW, którym jest serwer Apache obsługujący IP w wersji 6, stamtąd zaś delegowane lokalnie przez protokół IPv4 przy użyciu specjalnego połączenia między oboma serwerami poprzez protokół AJP (Apache Jserv Protocol). Zaletą takiej konstrukcji jest też utrzymanie przezroczystości dostępu dla użytkowników. Niezależnie od tego, czy korzystają oni z przeglądarki WWW w sposób zaimplementowany przez firmę UOZ czy ze specjalizowanej aplikacji (oraz niezależnie od wykorzystywanego protokołu warstwy sieciowej) odwołują się do tego samego serwera.

Jak wspomniano, urządzenia przenośne zostały wyposażone w osobną aplikację, nie zaś coraz popularniejsze rozwiązanie oparte na wyświetlaniu stron HTML. Zdaniem autorów, specjalizowana aplikacja lepiej wykorzystuje niewielką powierzchnię wyświetlacza oraz oferuje przyjaźniejszy interfejs graficzny. Dzięki wykorzystaniu technologii Jini [11] pozwalającej na zdalny dostęp do usług zarejestrowanych w specjalnym repozytorium (w tym wypadku do aplikacji klienta systemu NetRAAD), zadanie instalacji lub uaktualniania oprogramowania wykonywane jest automatycznie.

5.2. VIC i RAT

Duża wygoda użytkowania urządzeń typu PDA, możliwość stosowania w nich łączności bezprzewodowej standardu 802.11 oraz Bluetooth stwarzają olbrzymie możliwości dla implementacji aplikacji multimedialnych. Istniejące w tej dziedzinie rozwiązania dla komputerów stacjonarnych pracujących pod kontrolą różnych systemów operacyjnych współpracują między sobą najczęściej w oparciu o standard H.323. Przykładami takich programów są: znany z platformy Windows NetMeeting, oraz VIC i RAT działające zarówno w środowisku Unix jak i Windows. Program RAT (ang. Robust Audio Tool) – jest programem służącym do realizowania audiokonferencji, VIC (ang. Video Conferencing) – pozwala na realizację konferencji wideo.

Oba programy można stosować zarówno do zestawiania konferencji audio i wideo z bezpośrednim połączeniem między dwoma komputerami przy użyciu transmisji *unicastowych* (dwustronnych) typu punkt-punkt (ang. point-to-point), jak i konferencji z udziałem wielu uczestników z transmisją *multicastową* (grupową). W poprzednich latach programy te były wykorzystywane przez autorów w ramach projektu Meccano na komputerach stacjonarnych z użyciem protokołu IP w wersji 4. Obecnie oba programy: RAT w wersji 4.2.20 oraz VIC w wersji 2.8, działają w oparciu o protokół IP w wersji 6.

Rys. 4 Aplikacje multimedialne VIC i RAT



Zostały one zaadaptowane do używania na PDA iPAQ pracującego pod kontrolą systemu Linux.

W ramach testów, poza doświadczeniami w sieci lokalnej, wykonano kilka telekonferencji z UCL (University College London) w Anglii. Wszystkie testy wykonano przy konfiguracji unicastowej nad protokołem IPv6. W konferencji z jednej strony brał udział komputer stacjonarny wyposażony w kamerę (Linux i SUN) z drugiej iPAQ z interfejsem sieciowym w postaci karty bezprzewodowej AiroNet firmy Cisco. Niestety obecnie nie ma możliwości podłączenia do PDA kamery w związku z czym komunikacja wideo zachodziła tylko w jedną stronę. Przy połączeniu dwóch urządzeń iPAQ możliwa jest zatem na razie wyłącznie komunikacja głosowa. Zarówno w testach lokalnych jak i międzynarodowych rozmowa odbywała się z dobrą jakością, a obserwowany obraz był płynny. Uzyskana zajętość pasma na poziomie 250kb/s była wystarczająca dla dźwięku i obrazu wideo w rozdzielczości QSIF. Mimo bardzo oszczędnego gospodarowania pasmem przez oba programy trudno dziś sobie wyobrazić realizację konferencji przy wykorzystaniu telefonu komórkowego GSM/GPRS jako urządzenia dostępowego do Internetu z powodu zbyt małych przepustowości oferowanych obecnie przez te technologie.

6. Podsumowanie

Mobilne systemy wykorzystujące IPv6 powoli osiągają status pozwalający na ich uruchamianie i eksploatację. Tworzenie aplikacji nie jest jednak wygodne ze względu na brak dopracowanych pakietów oprogramowania współpracujących z tym protokołem. Poważnym krokiem w rozpowszechnieniu protokołu IPv6 byłoby wbudowanie jego stosu w systemy operacyjne firmy Microsoft, w szczególności Windows CE. Znaczącą przeszkodą jest także brak szerszej dostępności protokołu IPv6 na routerach większości producentów. Jedną z największych na rynku sprzętu sieciowego firm – Cisco Systems – dopiero w ostatnim okresie wprowadziła pełne stabilne implementacje protokołu IPv6 jako część systemu IOS. W kontekście aplikacji mobilnych systemów wykorzystujących IPv6 ważne negatywne znaczenie posiada fakt, iż system GPRS jest dostępny wyłącznie dla protokołu IPv4. Należy się jednak spodziewać, że w najbliższym okresie zostaną znacznie przyspieszone prace nad wdrażaniem protokołu IPv6.

Bibliografia

- 1 E. Amir et al.: An Active Service Framework and its Application to Real-time Multimedia Transcoding, Proc. ACM SIGCOMM'98, Vancouver, Sep. 1998.
- 2 M. Handley, J. Crowcroft, C. Bormann, J. Ott; The Internet Multimedia Conferencing Architecture; Internet Draft draft-ietf-mmusic-confarch-03.txt; work in progress; July 2000
- 3 P. T. Kirstein et al.: Accessing Mbone Sessions over Point-to-Point Connections. Technical report, Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK, 1998.
- 4 Ed Roman: Mastering Enterprise JavaBeans and the Java 2 Enterprise Edition; Wrox Press Ltd, 2001
- 5 Richard Monson-Haefel: Enterprise Java Bean; O'Reilly, 1999
- 6 Robert Orfali, Dan Harkey: Java and Corba; John Wiley&Sons Inc., 1997
- 7 Jacek Kosiński, Dominik Radziszowski: Linux i iPAQ; Linux Plus 11/2001
- 8 Radosław Ruchala, Krzysztof Zieliński: IPv6 in the Context of Mobile Applications; Proceedings of the 8th Conference „Computer Networks", 18-20 June 2001, Krynica, Poland
- 9 National Electrical Manufacturers Association, DICOM – Digital Imaging and Communication in Medicine, <http://medical.nema.org>
- 10 P. Loshin, IPv6 Clearly Explained, Academic Press, San Francisco, 1999, s. 43
- 11 W. K. Edwards, “Core Jini”, The Sun Microsystems Press, 1999
- 12 Rahim Adatia, Faiz Arni et al. “Professional EJB”, Wrox Press 2001