



# IPv6

## Jak poradzić sobie z identyfikacją coraz większej liczby urządzeń w sieci

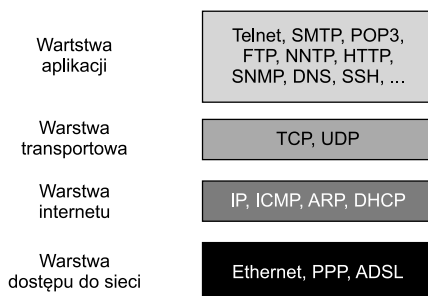
Wzrost popularności Internetu owocuje coraz większą liczbą dołączanych do niego terminali (komputerów), co doprowadzi wkrótce do wyczerpania puli adresów IP. Aby temu zapobiec, opracowano protokół IPv6, który oprócz znacznie zwiększonej przestrzeni adresowej wprowadza kilka usprawnień do protokołu IP w pierwszej wersji. W artykule opisano możliwości protokołu IPv6 oraz jego programowe implementacje dla urządzeń typu *embedded*.

### Krótko o protokołach sieciowych

Stos protokołów TCP/IP jest zbiorem protokołów sieciowych o komunikacji otwartej, co oznacza możliwość komunikacji z dowolnymi rodzajami urządzeń, gdyż w modelu TCP/IP nie jest sprecyzowana warstwa łącza. Określenie stos nawiązuje do warstwowego modelu przepływu danych pomiędzy warstwami. Dane, które są wysyłane przez aplikację użytkownika do innej aplikacji, pracującej na innym urządzeniu, muszą przejść przez kolejne warstwy stosu TCP/IP, zanim zostaną fizycznie przesłane przez sieć. W modelu odniesienia TCP/IP wyróżnia warstwy:

- aplikacji,
- transportową,
- sieciową,
- łącza.

Na rys. 1 przedstawiono graficzną reprezentację modelu TCP/IP wraz z przypisanymi im protokołami. Omawianie zadań każdej z warstw rozpoczniemy od najniższej. Warstwa łącza odpowiada za format pakietów wymienianych między węzłami sieci i ich fizyczne przesyłanie. Przykładem jest protokół Ethernet 802.11 (bezprowadowe sieci lokalne), ADSL czy 10BASE-T.



Rys. 1. Stos protokołów TCP/IP

Warstwa sieciowa Internetu zajmuje się przesyłaniem datagramów (pakietów danych) i ustalaniem ich trasy do komputera docelowego na podstawie adresów IP (źródłowego i docelowego).

W warstwie transportowej jest zapewnione nawiązywanie i utrzymywanie połączeń pomiędzy urządzeniami i aplikacjami. Każdemu połączeniu przypisany jest tzw. port, na podstawie którego otrzymywane datagramy z warstwy niższej są przekazywane do odpowiedniej aplikacji w warstwie aplikacji. W warstwie tej realizowana jest również kontrola przepływu i zapewnienie niezawodności dostarczania danych.

W warstwie aplikacji znajdują się programy użytkownika (aplikacje), które komunikują się z innymi aplikacjami poprzez stos protokołów TCP/IP (mogą to być aplikacje pracujące zarówno na tym samym urządzeniu, jak i na urządzeniu zdalnym). Każda z aplikacji korzysta z własnego protokołu, jak np. HTTP, który jest wykorzystywany przez aplikacje: serwera WWW, przeglądarki stron internetowych i innych aplikacji. Zazwyczaj w urządzeniach typu *embedded* jest stosowa-

**Dodatkowe materiały na CD i FTP:**  
 host: ep.com.pl, user: 12235, pass: 60u61csy  
 • noty katalogowe wybranych stosów protokołów TCP/IPv6

ny własny protokół wymiany danych (często jest stosowany protokół HTTP).

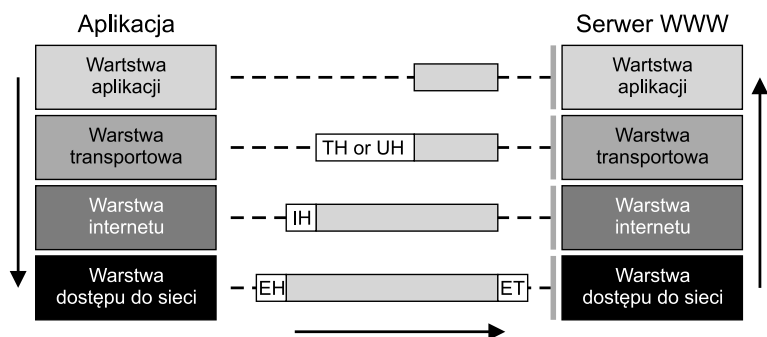
Należy zaznaczyć, że podczas przekazywania danych z warstwy aplikacji do warstwy łącza, każda warstwa pośrednia dołącza do danych z warstwy wyższej własny nagłówek, w którym ustawiane są odpowiednie parametry, jak np. adresy docelowy datagramu. Na rys. 2 przedstawiono kolejne etapy wysyłania zapytania do serwera www przez aplikację sieciową.

### Liczba pralek w Internecie

Jak można zauważyć, każda z warstw modelu TCP/IP jest aktywna przy przysłaniu informacji. Jednak nie wszystkie protokoły są niezbędne, zwłaszcza w urządzeniach specjalizowanych, w których nie są stosowane niektóre protokoły z warstwy aplikacji.

Obecnie praktycznie każdy telefon komórkowy ma wbudowaną przeglądarkę www, a niektóre bardziej zaawansowane usługi sieciowe, jak np. serwer wideo. Nikogo też nie zdziwi lodówka z możliwością

**Ile czasu do końca IPv4?**  
 Według danych statystycznych, na podstawie tempa rezerwacji wolnych adresów, pula publicznych adresów IPv4 (będąca pod kontrolą organizacji IANA) zostanie całkowicie zarezerwowana za 2 do 5 lat. Po tym terminie nadal będzie możliwa rejestracja adresów IP u regionalnych rejestratorów.



Rys. 2. Przepływ wiadomości między aplikacjami uruchomionymi na dwóch różnych urządzeniach

Bity	0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31
0	Wersja	Priorytet	Etykieta przepływu					
32	Długość danych			Następny nagłówek		Limit przeskoków		
64	Adres źródłowy (128 bitów)							
96								
128								
160								
192								
224	Adres docelowy (128 bitów)							
256								
288								

Rys. 3. Budowa nagłówka IPv6

zamawiania brakujących produktów żywnościowych przez Internet.

Projektując warstwę sieciową Internetu i metodę adresowania urządzeń, nie przewidziano tak ekstensywnego jego rozwoju. Identyfikacja urządzenia w sieci następuje poprzez jego adres protokołu IP. Jest to 32-bitowa liczba, zapisywana w postaci czterech liczb dziesiętnych trzycyfrowych, rozdzielonych kropkami (81.219.47.2). Jak łatwo można obliczyć, liczba możliwych adresów (pojemność adresowa) wynosi  $2^{32}$  (4294967296, czyli 256 grup po 16,777,216 adresów), jednak nie wszystkie adresy są używane do adresowania pojedynczych urządzeń. Około 16 milionów adresów jest używanych do adresowania rozgłoszeniowego, a około 18 milionów jest używanych do adresowania w sieciach lokalnych. Po wyłączeniu specjalnych adresów IPv4, do użytku publicznego pozostaje niespełna 220 grup adresów (około 3689544679 adresów).

### IPv6

Protokół IPv6 został zdefiniowany w grudniu 1998 roku przez grupę zajmującą się opracowywaniem standardów protokołów sieciowych IETF (Internet Engineering Task Force) w dokumencie RFC2460. Przy opracowywaniu nowego protokołu główny nacisk położono na zwiększenie puli adresowej, uproszczenie nagłówka (rys. 3) oraz ulepszenie procesu auto-konfiguracji.

W protokole IPv6 adres pamiętany jest na 128 bitach. Notacja tych adresów w postaci przyjaznej dla użytkownika jest zapisywana w postaci rozdzielonych dwukropkami czterocyfrowych liczb zapisanych w postaci szesnastkowej. Przykładem takiego adresu jest: 2001:0db8:0000:0000:0000:0000:3257:9652 przy czym liczby złożone z samych zer mogą być zastąpione pojedynczym zerem lub po-

#### Dlaczego nie IPv5?

Niektórych Czytelników może zainteresować, dlaczego następca protokołu IPv4 nie jest oznaczony numerem wersji 5. Otóż numer wersji protokołu 5 został wcześniej zarezerwowany dla eksperymentalnego protokołu ST służącego do przesyłania danych strumieniowych czasu rzeczywistego (np. dźwięk lub wideo). Miał on być używany obok protokołu IPv4 w warstwie sieciowej. Protokół ten nigdy nie doczekał się implementacji użytkowej.

minięte, ale tylko raz, w przeciwnym przypadku muszą być zapisane wprost: adres 2001:0db8:0:0:0:0:3257:9652 można zamienić na 2001:db8::3257:9652, ale 2001:0:0:0:FFD3:0:0:34ab nie może być zapisany jako 2001::FFD3::34ab, gdyż nie byłby on możliwy do odróżnienia od adresu np.: 2001:0:0:FFD3:0:0:0:34ab.

W implementacjach stosu protokołów TCP typu dual, czyli z jednoczesną obsługą protokołów IPv4 i IPv6, używa się adresów IPv4 mapowanych na adresy IPv6. Adres taki na pierwszych 80 bitach ma same zera, na kolejnych 16 same jedynki, a na ostatnich 32 bitach zapisany jest mapowany adres IPv4. Adres 192.0.2.128 zapisany jest w postaci: ::ffff:c000:280

Adresy IPv4 mapowane w IPv6 mogą być zapisywane przy użyciu notacji mieszanej IPv6 i IPv4: ::ffff:192.0.2.128

Zwiększenie przestrzeni adresowej protokołu IPv6 nie jest jedynym usprawnieniem względem wcześniejszego IPv4. Dla sieci lokalnych zarezerwowano pulę adresów z maską /64. Umożliwia to urządzeniom sieciowym tworzenie unikalnych adresów sieciowych w oparciu o numer MAC interfejsu sieciowego. Adres taki składa się z 64-bitowego prefiksu sieci oraz odpowiednio zapisanego numeru MAC na pozostałych 64 bitach adresu IPv6. Numer MAC złożony jest z 48-bitów. Aby mógł on być użyty w nagłówku, IPv6m jest dzielony na pół (według zapisu heksadecymalnego), a w środek wstawiana jest wartość FFFE, dodatkowo pierwszy bajt adresu MAC jest zwiększany o 2. Dla przykładowego adresu MAC (11:22:33:44:55:66) zostanie stworzony adres: fe80::1322:33ff:fe44:5566 w przypadku sieci lokalnej bez routera. Jeżeli w danej sieci jest router, wysyła on wiadomość rozgłoszeniową, w której zapisany jest 64-bitowy prefiks sieci.

Protokół IPv6 wprowadza również obsługę jakości strumienia danych QoS przy użyciu pola Flow Label w nagłówku. Fragmentacja pakietów może być dla protokołu IPv4 wprowadzana zarówno przez nadającego hosta, jak i przez routery, przez które przechodził pakiet danych. W IPv6 fragmentacja danych jest przeprowadzana jedynie przez host nadający informację (na podstawie najmniejszego możliwego do przesłania pakietu danych na trasie trans-

#### Specjalne adresy

Następujące adresy i grupy adresów mają specjalne, zarezerwowane znaczenie:  
 ::/128 – adres nieokreślony (zawierający same zera)  
 ::1/128 – loopback, adres wskazujący na host lokalny  
 ::/96 – pula zarezerwowana dla zachowania kompatybilności z protokołem IPv4 (pierwszych 96 bitów stanowią 0, pozostają 32 bity na adresy w formacie IPv4)  
 ::ffff:0:0/64 – jw., ale pozwala wykonywać komunikację według protokołu IPv6 w sieci IPv4  
 2001:7ff8::/32 – pula zarezerwowana dla punktów wymiany ruchu, każdy z nich dostaje jedną podsieć /48  
 2001:db8::/32 – adresy do stosowania wyłącznie w przykładach i dokumentacji – adresy z tej puli nie mogą być używane przez urządzenie sieciowe  
 2002::/24 – adresy typu 6to4 – są to adresy wygenerowane na podstawie istniejących, publicznych adresów IPv4  
 3ffe::/16 – adresy testowej sieci 6BONE (adresy zostały wycofane 6 czerwca 2006 w związku z zakończeniem działania 6BONE)  
 fc00::/7 – pula lokalnych unikalnych adresów IPv6 typu unicast, będąca odpowiednikiem adresów prywatnych IPv4, choć, zgodnie z nazwą, powinny być unikalne na świecie  
 fe80::/10 – pula link-local określa adresy w obrębie jednego łącza fizycznego (np. segmentu sieci Ethernet). Pakiety z tej puli nie są przekazywane poza podsieć  
 fec0::/10 – pula site-local określa adresy w obrębie jednej lokalnej organizacji – obecnie nie zaleca się wykorzystywania tej puli, a przyszłe implementacje IPv6 nie będą musiały jej obsługiwać  
 ff00::/8 – pula adresów używana do komunikacji multicast

misji). W celu uproszczenia nagłówka IPv6 pozbyto się sumy kontrolnej, gdyż i tak protokoły warstw wyższych z reguły sprawdzają własną sumę kontrolną (np. protokół TCP). Wszystkie pozostałe dane zostały przeniesione do nagłówków dodatkowych.

Kolejną zmianą dotyczącą bezpieczeństwa na poziomie warstwy sieciowej jest obowiązkowa obsługa protokołu IPsec. Protokół ten służy do szyfrowania danych i autentyfikacji zdalnego hosta.

### Internet embedded

W celu implementacji obsługi komunikacji sieciowej w urządzeniach typu *embedded* należy posłużyć się implementacją stosu protokołów TCP/IP, która spełnia minimalne wymagania wyszczególnione w specyfikacjach RFC. W czasach, gdy nowe produkty pojawiają się każdego dnia, nie warto chyba tracić czasu na opracowywanie własnej implementacji stosu protokołów TCP/IPv6. Na rynku dostępnych jest wiele gotowych rozwiązań zoptymalizowanych pod kątem użycia w urządzeniach typu *embedded*, w których nierzadko mamy do czynienia z ograniczonymi zasobami pamięci RAM i ROM.

Poniżej znajduje się opis kilku wybranych implementacji stosu protokołów TCP, ze zwróceniem uwagi na ich nietypowe cechy.

### Wind River

Wind River Network Stack jest pełnym stosem protokołów TCP/IP typu dual stack IPv4/IPv6 (jest to połączenie technologii firmy

Wind River i technologii zakupionych od Interpeak AB). Zawiera obsługę protokołów UDP, TCP, ICMP oraz IGMP, a także wspiera QoS dzięki implementacji DiffServ. Implementacja stosu TCP/IP firmy Wind River umożliwia jednoczesną obsługę adresowania zarówno IPv4, jak i IPv6. Wind River Network Stack jest jednym z pierwszych implementacji protokołu IPv6, który uzyskał logo IPv6 Ready Phase II.

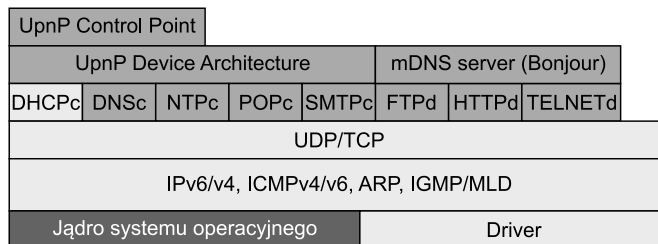
## InterNiche

Kolejną firmą specjalizującą się w dostarczaniu rozwiązań programowych dla urządzeń wbudowanych jest InterNiche Technologies. W swoim portfolio ma stosy protokołów TCP/IP: NicheStack IPv4, NicheLite IPv4, NicheStack IPv6 i NicheStack IPv4/IPv6. Stos protokołów TCP/IP NicheStack IPv6 jest napisany w języku ANSI C. Jest on przeznaczony do urządzeń typu embedded i może być uruchomiony wraz z dowolnym systemem operacyjnym lub bez systemu operacyjnego. NicheStack Dual IPv4/IPv6 jest przeznaczony do aplikacji, w których wymagana jest jednoczesna komunikacja z urządzeniami korzystającymi z jednego z tych dwóch protokołów.

## Mentor Graphics

Mentor Graphics jest producentem systemu operacyjnego typu RTOS Nucleus OS. Ma on moduły programowe do obsługi m.in. USB, bazy danych, pamięci masowych, zaawansowanego graficznie interfejsu użytkownika oraz protokołów sieciowych. Ze strony producenta można pobrać bezpłatną wersję demonstracyjną systemu operacyjnego Nucleus OS. Implementacja protokołu IPv6 w tym systemie operacyjnym uzyskała logo IPv6 ready.

Nucleus Networking obsługuje protokoły po stronie klienta: TCP/UDP, SNMP, DHCP,



Elementy stosu, które można usunąć

Rys. 4. Budowa stosu protokołów Ubiquitous TCP/IP

POP3. Dodatkowo urządzenie z tą implementacją stosu protokołów TCP/IP może pracować zarówno jako klient, jak i serwer w aplikacjach: HTTP, FTP, TFTP i SMTP. Ma on również rozbudowane mechanizmy bezpieczeństwa. Obsługuje algorytmy szyfrowania blokowego (m.in. AES, DES), funkcje skrótu (MD4, MD5, SHA-1, SHA-256) oraz protokoły szyfrowanej transmisji danych: IPsec, SSL 2.0 i 3.0, Secure FTP i Secure Telnet.

## Unicoi Systems

Firma Unicoi Systems oferuje wiele komponentów programowych. Jednym z nich jest stos protokołów TCP/IP, w którym zaimplementowano jednoczesną obsługę protokołów IPv4 i IPv6. Fusion TCP/IPv4/IPv6 oznaczony jest logo IPv6 ready. Może być używany zarówno z systemem operacyjnym, jak i bez niego. Firma oferuje również moduły programowe do obsługi popularnych protokołów, takich jak HTTP, DNS, DHCP, SMTP, Telnet, a także protokoły do bezpiecznej wymiany danych HTTPS/SSL i IPsec oraz pakiet oprogramowania obsługi plików XML dla urządzeń wbudowanych.

## Ubiquitous

Nietypowym rozwiązaniem jest stos UbiquitousTCP/IP. Ma on bowiem zaimplemen-

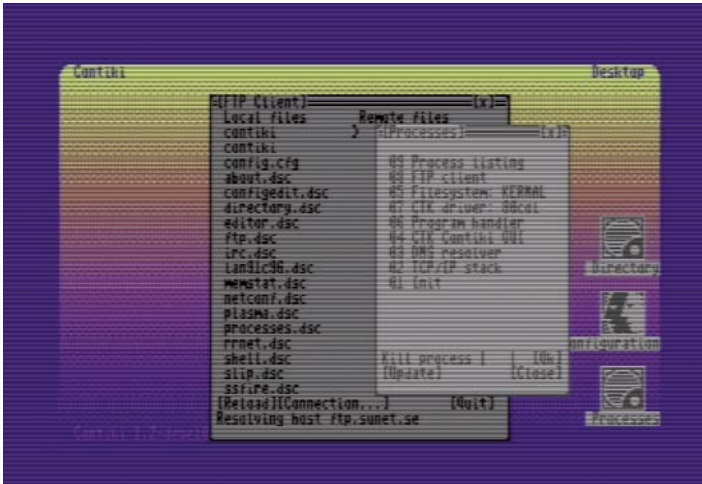
towane jądro systemu operacyjnego (rys. 4), które zarządza działaniem tego stosu. Jak podaje producent, nie jest to system typu RTOS. Odejście od konieczności wyłączenia wątków po zadanym czasie lub przez wątek o wyższym priorytecie ma zmniejszyć narzut systemu operacyjnego i w efekcie zwiększyć wydajność połączeń w systemie typu *embedded*. Przykładowo, dla przesyłania danych poprzez FTP dla rdzenia ARM9EJ-S taktowanego sygnałem zegarowym o częstotliwości 200 MHz można uzyskać przepustowość 240 Mb/s.

## TeamF1

Firma TeamF1 opracowała stos protokołów TCP/IP Hyper V6, który może obsługiwać jednocześnie IPv4 i IPv6. W NetF1 wprowadzono obsługę wirtualnych routerów. Są to routery logiczne, które są uruchamiane jednocześnie w systemie wielowątkowym. Każdy z routerów wirtualnych pracuje niezależnie i z własnym zbiorem obsługiwanych protokołów routingu oraz zestawem danych (np. tablicą routingu). Stos protokołów NetF1 jest kompatybilny ze standardem POSIX i jak podaje producent, może być używany w systemach operacyjnych korzystających z tego standardu bez potrzeby rekompilacji kodu źródłowego.

Tab. 1. Zestawienie omawianych stosów protokołów TCP/IP

Nazwa stosu	Firma	Dual (IPv4 + IPv6)	Praca bez systemu operacyjnego	Praca z systemem operacyjnym	Zaimplementowano w systemie operacyjnym	website	IPv6
eCos TCP/IP	opensource	b.d.		✓	eCos	<a href="http://ecos.sourceforge.org">ecos.sourceforge.org</a>	–
Fusion TCP/IPv4/IPv6	Unicoi Systems	✓	✓	✓		<a href="http://unicoi.com">unicoi.com</a>	II
IPv4/IPv6 Dual Network Stack	EBSnet	✓	✓	✓		<a href="http://www.ebsnetinc.com">www.ebsnetinc.com</a>	I
Kasago IPv6	ZUKEN ELMIC	✓	✓			<a href="http://www.elwsc.co.jp/english">www.elwsc.co.jp/english</a>	II
NetF1	TeamF1	✓				<a href="http://www.teamf1.com">www.teamf1.com</a>	I
NexGenIPv6	NexGen Software	✓				<a href="http://www.nexgen-software.com">www.nexgen-software.com</a>	II
NicheStack IPv4/IPv6	InterNiche	✓	✓	✓		<a href="http://www.iniche.com">www.iniche.com</a>	–
NicheStack IPv6	InterNiche		✓	✓		<a href="http://www.iniche.com">www.iniche.com</a>	–
Nucleus Networking	Mentor Graphics	b.d.		✓	Nucleus	<a href="http://www.mentor.com">www.mentor.com</a>	II
RTXC Quadnet IPv4/IPv6	Quadros	✓		✓	RTXC Quadros RTOS	<a href="http://www.quadros.com">www.quadros.com</a>	–
TargetTCP	Blunk Micro	✓		✓	TargetOS	<a href="http://www.blunkmicro.com">www.blunkmicro.com</a>	I
Treck Embedded IPv4/IPv6	Treck Inc	✓	✓	✓		<a href="http://www.treck.com">www.treck.com</a>	II
UbiquitousTCP/IP	Ubiquitous	✓			✓	<a href="http://www.ubiquitous.co.jp/En">www.ubiquitous.co.jp/En</a>	II
uIPv6	opensource	✓	✓	✓	Contiki	<a href="http://www.sics.se/contiki">www.sics.se/contiki</a>	I
Wind River Network Stack	Wind River	✓		✓	VxWorsk, Linux	<a href="http://www.windriver.com">www.windriver.com</a>	II



Rys. 5. System operacyjny Contiki uruchomiony na komputerze Commodore 64

### Inne implementacje i platformy

Oprócz opisanych w artykule stosów protokołów TCP/IP, jest oferowanych wiele innych, na które warto zwrócić uwagę. Przykładem może być IPv4/IPv6 Dual Network Stack firmy EBSnet, który został wybrany przez Microsoft do implementacji w platformie .NET micro dla mikrokontrolerów 32-bitowych w aplikacjach typu *embedded*. Listę dostępnych rozwiązań stosu TCP/IP z obsługą IPv6 zestawiono w tab. 1.

Wśród implementacji stosów TCP/IP są również takie, które są integralną częścią systemu operacyjnego. Przykładem jest system operacyjny typu RTOS eCos, w którym jest wbudowana obsługa sie i TCP/IP z rozszerzeniem dla IPv6. Kolejnym jest VxWorks – uznany system operacyjny czasu rzeczywistego firmy Wind River.

W systemach operacyjnych typu Linux w większości przypadków jest zaimplementowana obsługa protokołu IPv6. Na rynku dostępnych jest wiele bezpłatnych i płatnych implementacji systemu Linux dla urządzeń wbudowanych. Również  $\mu$ CLinux, czyli Linux dla mikrokontrolerów bez układu MMU, ma zaimplementowaną obsługę IPv6.

Jedną z firm, która oferuje komercyjne rozwiązanie tego systemu operacyjnego, jest Wind River. Wind River Linux 3.0 jest oparty na jądrze 2.6.27 i kompilatorze GCC 4.3.2. Może on być uruchamiany na jednej z pięciu platform sprzętowych (ARM, x86, SPARC, MIPS i Po-

werPC). Wykorzystano w nim w pełni możliwości systemu Linux pod względem obsługi protokołów IPv4/IPv6, w tym MIPv6.


Wśród rozwiązań bezpłatnych typu open source znajduje się uIPv6. Jest on zaprojektowany przez programistów zrzeszonych wokół projektu Contiki. Jest to system operacyjny przeznaczony dla urządzeń typu *embedded*, które pracują dołączone do sieci komputerowej (rys. 5). Podstawowe skonfigurowanie tego systemu operacyjnego wymaga zaledwie 2 kB pamięci RAM i 10 kB ROM. Stos uIPv6 może być uruchamiany zarówno w systemie operacyjnym (dowolnym), jak i bez niego.

Wśród specjalizowanych systemów operacyjnych dla urządzeń przenośnych i telefonów komórkowych nie ma praktycznie systemu operacyjnego, który by nie wspierał IPv6. System operacyjny Symbian OS od wersji 7.0, znany użytkownikom telefonów firmy Nokia, też obsługuje IPv6. Podobnie inny system operacyjny przewidziany dla mobilnych urządzeń multimedialnych Android opracowany przez Google wspiera adresowanie IPv4 i IPv6.


Wśród systemów operacyjnych wspierających IPv6 dla urządzeń *embedded*

**Program IPv6 Ready Logo**

Logo IPv6 jest przyznawane urządzeniom i oprogramowaniu, które spełnia określone wytyczne. Program IPv6 Ready Logo służy do weryfikacji implementacji protokołu IPv6 oraz jego interoperacyjności. Oferuje bezpłatne narzędzia do testowania protokołu IPv6 oraz opracowywane przez międzynarodowe laboratoria testujące. Przyznawaniem logo IPv6 Ready zarządza organizacja IPv6 Forum. Urządzenia i oprogramowanie oznaczone srebrnym IPv6 Ready Logo Phase I spełniają minimalne wymagania odnośnie do głównych funkcjonalności IPv6. Złote logo IPv6 Ready Phase II jest przyznawane, jeżeli urządzenie lub oprogramowanie spełnia wymagania określone przez dokumenty standardyzacyjne RFC wydane przez IETF z klauzulami „MUST” i „SHOULD”. Oprócz podstawowych wymagań, w czasie testów Phase II są badane protokoły: Ipsec, IKEv2, MIPv6, NEMO, DHCPv6, SIP, Management (SNMP-MIBs).



**IPv6 Ready Logo Phase I**



**IPv6 Ready Logo Phase II**

warto wspomnieć również Windows CE 6.0 R3 Microsoftu. Jest to system operacyjny przystosowany urządzeń o niewielkiej pojemności pamięci (300 kB dla 700 standardowych komponentów programowych), z mikrokontrolerami o architekturze ARM, MIPS, x86. W systemie operacyjnym dla telefonów komórkowych Windows Mobile jest także zaimplementowana obsługa IPv6.

### Podsumowanie

Jak wspomniano, protokół IPv6 jest niewątpliwie przyszłościowy i wcześniej czy później zostanie wdrożony do ogólnego użytku, zastępując protokół IPv4. Z tego względu warto już teraz mieć na uwadze obsługę IPv6 w urządzeniach typu *embedded*.

**Maciej Gołaszewski, EP**  
[maciej.golaszewski@ep.com.pl](mailto:maciej.golaszewski@ep.com.pl)

R E K L A M M A

**RK-SYSTEM**  
[www.rk-system.com.pl](http://www.rk-system.com.pl)

**Profesjonalne narzędzia dla elektroników i programistów**

- uniwersalne programatory układów scalonych
- analizatory stanów logicznych
- oscyloskopy cyfrowe
- systemy do wyważania i pomiaru drgań
- oprogramowanie CAD, CAM, CAE
- emulatory, symulatory, debugery dla różnych rodzin procesorów
- kompilatory C/C++ dla różnych rodzin procesorów
- szkolenia w zakresie FPGA, VHDL
- narzędzia na procesory sygnałowe DSP
- projektujemy, produkujemy, szkolimy, dystrybuujemy

05-825 Grodzisk-Maz, ul. Chałmońskiego 30, tel. (022) 724 30 38, 792 05 18, fax: (022) 724 30 37

RAISONANCE Innovative Development Tools | IAR SYSTEMS | SPECTRUM DIGITAL