

Dariusz DOLIWA, Wojciech HORZELSKI  
Uniwersytet Łódzki, Wydział Matematyki i Informatyki

## **APLIKACJA WSPOMAGAJĄCA ZARZĄDZANIE ADRESAMI IP DLA DOSTAWCY INTERNETU**

**Streszczenie.** Prezentowana aplikacja bazodanowa ułatwia zarządzanie zasobami adresowymi oraz administrowanie wirtualnymi sieciami lokalnymi. Dzięki zastosowaniu zmodyfikowanego algorytmu BestFit do wyszukiwania wolnych pól adresowych, umożliwia elastyczne dysponowanie posiadanymi zasobami IP. Dodatkowo automatycznie tworzy i aktualizuje pliki konfiguracyjne dla przełączników i routerów.

**Słowa kluczowe:** zarządzanie adresami IP, relacyjna baza danych, VLAN

## **IP ADDRESSES MANAGEMENT SUPPORT APPLICATION FOR ISP**

**Summary** Presented database application supports address resource management and administration of virtual LANs. By using a modified algorithm BestFit to find the address pools allows efficient disposition of IP networks owned by ISP. In addition, automatically creates and updates the configuration files for the switches and routers.

**Keywords:** IP management, relational database, VLAN

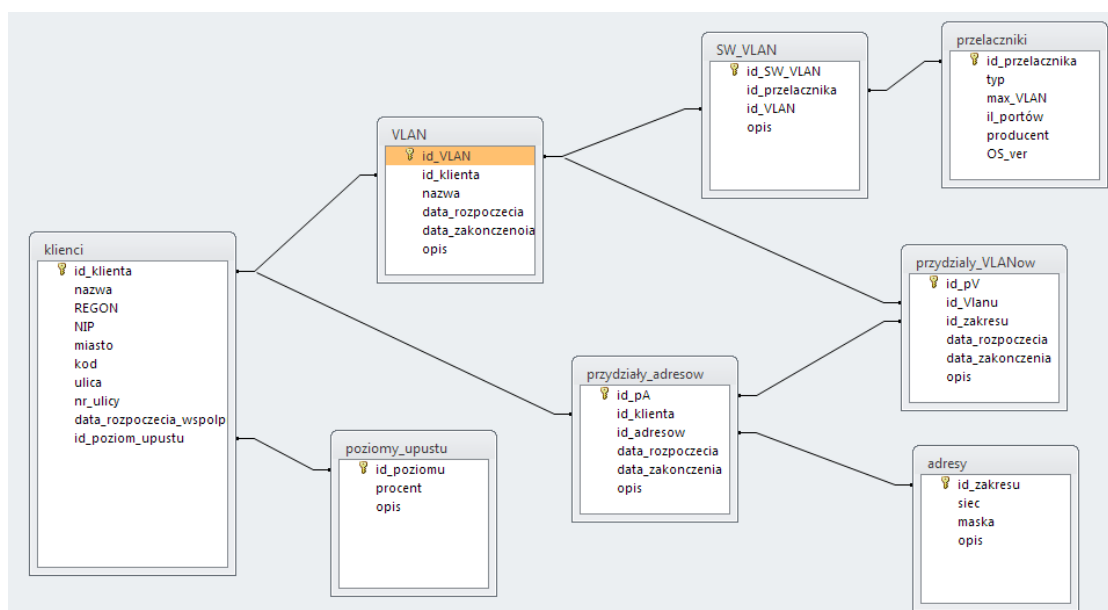
### **1. Funkcjonalność aplikacji**

Protokół Internetu (IP) w obecnie stosowanej wersji (v.4) posiada silnie ograniczone zasoby adresów [1], dlatego dostawcy Internetu dysponują z reguły niewielkimi pulami adresów publicznych. W takiej sytuacji ich efektywne wykorzystywanie stanowi jeden z priorytetów działalności takich firm. Prezentowana tu aplikacja ułatwia zarządzanie adresami przeznaczonymi dla poszczególnych klientów, zarówno instytucjonalnych, jak i prywatnych. Wspomaga też dystrybucję zarządzanych przez ISP (Internet Service Provider) pul adresów pomiędzy wirtualnymi sieciami lokalnymi, zarządzanymi przez tego dostawcę, oraz automatyczne konfigurowanie routerów i przełączników w celu realizacji tych przydziałów w jego

systemie sieciowym. Aplikacja jest wygodnym narzędziem wspomagającym pracę administratora. Dodatkowo ułatwia zarządzanie infrastrukturą sieciową.

## 2. Struktura bazy danych

Baza danych składa się z tabel zawierających dane o klientach, wykorzystywanych pulach adresowych, wirtualnych sieciach lokalnych oraz przełącznikach. Struktura tabel przedstawiona została na rys. 1.



Rys. 1. Struktura bazy danych

Fig. 1. Database structure

Program przechowuje informację o przydzielonych klientom pulach adresowych za pomocą tabel *przydziały\_adresów* i *adresy*. Każdy taki przydział jest powiązany z VLAN-em przez tabelę *przydział\_VLANów*. Dodatkowo możliwe jest przypisanie do klienta sieci VLAN i niewiązanie jej z żadną pulą adresową – służy to rezerwacji wirtualnych sieci do przyszłego wykorzystania. Za pomocą tabeli *SW\_VLAN* przechowywane są informacje o sieciach wirtualnych, które powinny być reprezentowane na konkretnych przełącznikach. Konfiguracje tych ostatnich (razem z pozostałymi istotnymi informacjami) przechowywane są w tabeli *przełączniki*.

## 3. Algorytmy wyszukiwania wolnych zakresów

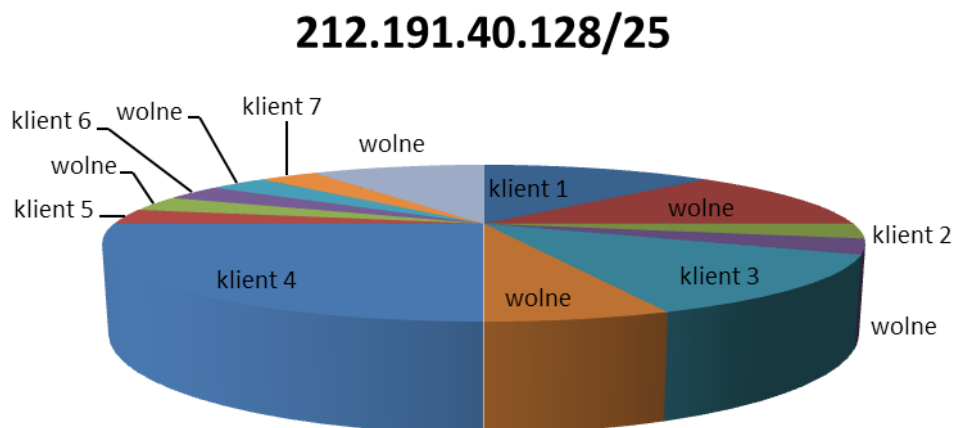
Projektując aplikację, przyjęto założenie, że przyznawana klientowi pula adresów publicznych powinna ulegać zmianie jedynie w sytuacji jej rozszerzania. Założenie takie jest

oczywiste z punktu widzenia klienta (nie ma potrzeby rekonfiguracji usług sieciowych). Powoduje jednak powstawanie niespójności wykorzystywanych adresów, ponieważ kontrahenci mogą rezygnować z usług dostawcy (pozostawiając „dziurę” w przestrzeni adresowej) bądź żądać większej sieci – wtedy może zachodzić konieczność przydzielenia im nowej, innej, większej puli adresów [2]. Dlatego algorytm wyszukujący wolne pule adresów musi być tak skonstruowany, aby z jednej strony jak najlepiej wypełniał wolne przestrzenie adresowe, z drugiej natomiast dawał możliwość powiększania przyznanej puli w przyszłości, czyli w miarę możliwości nie przydzielał kolejnemu klientowi sąsiedniej podsieci [3].

Zastosowany w aplikacji algorytm jest modyfikacją algorytmu BestFit dla zadania Bin-packing [4]. Przestrzeń adresowa podzielona zostaje na podprzestrzenie o rozmiarze 32 (wielkość ta została dobrana na podstawie wymagań klientów i może być modyfikowana w aplikacji). Następnie wykorzystywany jest zmodyfikowany algorytm BF dla tak rozumianych pojemników. Algorytm ten wyszukuje wśród wszystkich dostępnych, tj. już częściowo wykorzystanych, podprzestrzeni adresowych („pojemników”) tę, dla której:

- 1) po zarezerwowaniu tej puli pozostanie największa ilość wolnego miejsca, jeżeli jest ona większa lub równa przydzielanej puli oraz możliwa jest alokacja obszaru o adresie będącym wielokrotnością podwojonej puli;
- 2) po zarezerwowaniu pozostanie najmniejsza ilość wolnego miejsca w pozostałych przypadkach;
- 3) jeśli nie ma takiej podprzestrzeni, rozpoczynamy wykorzystanie następnej – kolejnego „pojemnika”.

W klasycznym BF wybierany jest pojemnik, dla którego zostaje minimalna ilość miejsca. Modyfikacja taka ma na celu tworzenie wolnych przestrzeni adresowych, które mogą być w przyszłości dokładane do przestrzeni adresowych klientów. Rysunek 2 przedstawia przykładowe wykorzystanie przestrzeni adresowej po zastosowaniu tego algorytmu dla siedmiu klientów wymagających kilku różnych rozmiarów sieci.

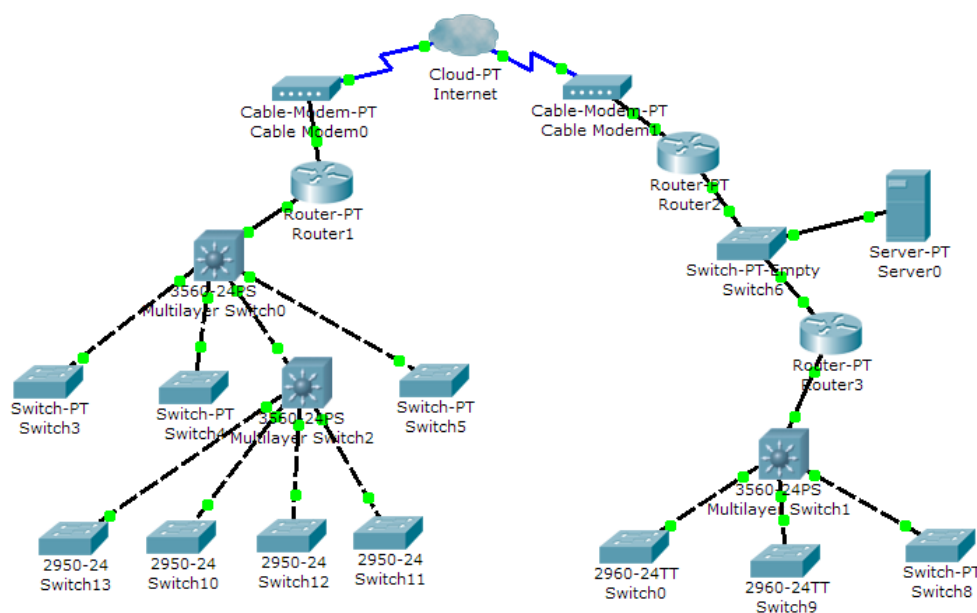


Rys. 2. Wykorzystanie puli adresowej  
Fig. 2. Use of address pool

Aplikacja pozwala również na wyszukiwanie dostępnych przestrzeni adresowych za pomocą klasycznego algorytmu BF oraz samodzielne dodawanie wskazanych przez administratora pul. Może to nastąpić w szczególnych przypadkach, kiedy wymagania klienta są niestandardowe, np. wtedy, kiedy wiadomo, że klient nie będzie w przyszłości potrzebował dodatkowych adresów bądź liczba adresów potrzebnych w przyszłości jest znana wcześniej.

#### 4. Topologia ISP

Aplikacja została zaprojektowana dla dowolnej firmy zajmującej się dostarczaniem Internetu (ISP). Testy przeprowadzono dla topologii przedstawionej na rys. 3. Firma dysponuje routerami dostępowymi w dwóch lokalizacjach. Routery te połączone są tunelem VPN. W każdej lokalizacji połączenia dla klientów udostępniane są przez przełączniki (realizujące zadania warstwy III), wszystkie przełączniki skonfigurowane są za pomocą protokołu VTP. Połączenia do klientów zdalnych realizowane są za pomocą modemów kablowych w centralnej lokalizacji, dla każdego z takich odbiorców dedykuje się jedną sieć wirtualną na przełączniku.



Rys. 3. Topologia sieci

Fig. 3. Network topology

#### 5. Konfiguracja urządzeń sieciowych

W bazie danych dla każdego urządzenia przechowywany jest jego plik konfiguracyjny. Jest on kopią zapasową, pozwalającą zarówno przywrócić poprawne ustawienia w razie awa-

rii sprzętu, jak i ustalić aktualne przypisania portów przełącznika do określonej sieci wirtualnej (VLAN). Aplikacja, przez analizę tego pliku, znajduje niewykorzystywane porty, spośród których wybieramy port (lub porty), który będzie przydzielony klientowi. Decydujemy również o tym, czy porty te mają być przypisane do nowo stworzonej sieci wirtualnej czy klient tylko rozszerza posiadaną sieć o dodatkowe adresy IP w ramach posiadanego już VLAN-u. Po wyborze przez administratora systemu wymaganych opcji, wywoływana jest funkcja aktualizująca plik konfiguracyjny. Parametrami tej funkcji są nazwa klienta, przydzielony klientowi adres sieci, numer sieci wirtualnej, do której będzie należało połączenie klienta, wybrany port lub porty przełącznika oraz opcja ustalająca, czy jest to nowa sieć czy już istniejący VLAN. Dodatkowo musimy określić, do którego routera jest podłączony wybrany przełącznik. Po uruchomieniu funkcja generuje w pliku konfiguracyjnym przełącznika odpowiednie wpisy (część jego konfiguracji). Na przykład, jeżeli dołączamy nowego klienta, dla którego przy użyciu wcześniej opisanych algorytmów wyznaczyliśmy adres sieci 212.191.40.128/29, i przypisujemy go do nowo tworzonej sieci wirtualnej o numerze 100 i portu o nazwie FastEthernet 0/10, to funkcja zmodyfikuje plik konfiguracyjny przełącznika vlan.dat z informacją o nowej sieci wirtualnej, a następnie właściwą konfigurację o następujące komendy:

```
interface FastEthernet0/10
  description Nazwa_klienta
  switchport access vlan 10
```

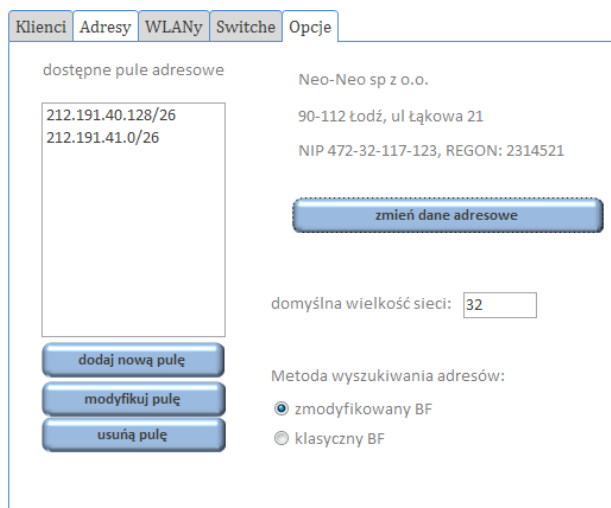
zamiast dotychczasowego opisu interfejsu. Ponadto, na wskazanym routerze w jego pliku konfiguracyjnym zostanie dopisana funkcjonalność serwera DHCP, który będzie przydzielał adresy klientowi, oraz skonfigurowany zostanie podinterfejs, obsługujący nową sieć wirtualną:

```
...
ip dhcp pool nazwa_klienta
  network 212.191.40.128 255.255.255.248
  default-router 212.191.40.129
  dns-server 212.191.40.254
...
interface FastEthernet0/0.100
  encapsulation dot1Q 100
  ip address 212.191.40.129 255.255.255.248
...
```

Komendy te zmodyfikują pliki konfiguracyjne urządzeń oraz dodatkowo zostaną zapisane w osobnych plikach, które po przesłaniu (przez protokół TFTP) do tych zmodyfikują sposób ich działania bez potrzeby ponownego ich uruchamiania (nie spowoduje to przerwy w funkcjonowaniu sieci dla pozostałych klientów).

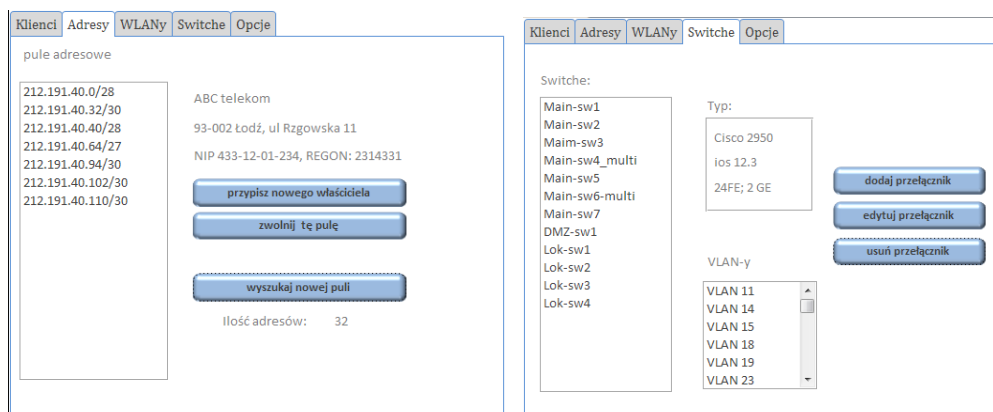
## 6. Interface komunikacyjny

Interface aplikacji zaprojektowano tak, aby obsługa aplikacji była możliwie prosta. Opcje konfiguracyjne programu (dostępne przestrzenie adresowe, dane adresowe właściciela firmy, metody wyszukiwania adresów) zgrupowano w dialogu *opcje*, przedstawionym na rys. 4.



Rys. 4. Dialog konfiguracyjny aplikacji  
Fig. 4. Application configuration dialog

Pozostałe dialogi służą do obsługi poszczególnych grup informacji – są to informacje o klientach (dane adresowe, informacje o rabatach itp.), informacje o przyznanym klientom pulach adresów, wykorzystywanych VLAN-ach oraz o przełącznikach i ich konfiguracjach. Rysunek 5 przedstawia dwa przykładowe dialogi (adresy oraz przełączniki).



Rys. 5. Interface aplikacji  
Fig. 5. Application interface

## 7. Podsumowanie

Zaprezentowana tutaj aplikacja w znacznym stopniu usprawnia funkcjonowanie firmy zajmującej się dostarczaniem dostępu do sieci rozległej. Upraszcza proces administrowania zasobami adresów IP i wirtualnymi sieciami lokalnymi (VLAN-ów), obsługującymi sieci klienckie. Zastosowanie zmodyfikowanego algorytmu BestFit pozwala na szybkie wyszukiwanie wolnych puli adresów, zapewniając jednocześnie skalowalność sieci logicznych poszczególnych klientów. Dodatkowo automatycznie wytwarza ona potrzebne pliki konfiguracyjne, co pozwala na utrzymanie spójności systemu sieciowego.

## BIBLIOGRAFIA

1. Murphy N., Wilson D.: The End of Eternity. IPv4 address Exhaustion and Consequences, The Internet Protocol Journal, Vol. 11 (4), 2008.
2. Sportack M.: IP Addressing Fundamentals. Cisco Press, 2002.
3. White R.: Working with IPAddresses. The Internet Protocol Journal, Vol. 9 (1), 2006.
4. Hoffman E. G., Leuker G. S.: Probabilistic Analysis of Packing and Partitioning Algorithms. John Wiley & Sons, New York 1991.
5. Blum R.: C# Network Programming. Sybex, 2003.
6. Postel J. B.: Internet Protocol (RFC 791), 1981.

Recenzenci: Dr hab. inż. Andrzej Chydziański, prof. Pol. Śląskiej  
Dr inż. Paweł Kasprowski

Wpłynęło do Redakcji 11 stycznia 2011 r.

## Abstract

Application presented in this paper significantly improves the performance of the company providing access to wide area network (ISP). Information about managed resources are represented by a relational database (Fig. 1). The program simplifies the process of administration of the resources of IP addresses and virtual LANs managed by company. The use of a modified BestFit algorithm allows to quickly find free address pool (Fig. 2), while ensuring network scalability for individual clients. In addition, it automatically generates the

necessary configuration files and modifies the configuration of network devices, allowing to maintain the integrity of a ISP network system.

### **Adresy**

Dariusz DOLIWA: Uniwersytet Łódzki, Wydział Matematyki i Informatyki, ul. Banacha 22, 90-238 Łódź, Polska, [doliwa@imul.uni.lodz.pl](mailto:doliwa@imul.uni.lodz.pl).

Wojciech HORZELSKI: Uniwersytet Łódzki, Wydział Matematyki i Informatyki, ul. Banacha 22, 90-238 Łódź, Polska, [horzel@imul.uni.lodz.pl](mailto:horzel@imul.uni.lodz.pl).