Maciej TWARDY [2], **Grzegorz SUŁKOWSKI** [2], **Kazimierz WIATR** [1,2]
[1] AKADEMIA GÓRNICZO-HUTNICZA
[2] ACK CYFRONET AGH

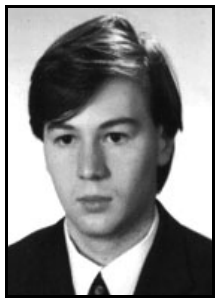
Szybka filtracja portów sieciowych w sprzętowym systemie bezpieczeństwa typu Firewall

Mgr inż. Maciej TWARDY

Ukończył studia na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Od 2005 roku kieruje Działem Archiwizacji i Bezpieczeństwa Danych w Akademickim Centrum Komputerowym CYFRONET AGH. Jego zainteresowania związane są z szeroko pojętą informatyką oraz projektowaniem układów cyfrowych w oparciu o układy reprogramowalne.

e-mail: Maciej.Twardy@cyfronet.pl

Mgr inż. Grzegorz SUŁKOWSKI



Ukończył studia na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Od 2006 roku jest Konstruktorem Systemów Obliczeniowych w dziale Archiwizacji i Bezpieczeństwa Danych w Akademickim Centrum Komputerowym CYFRONET AGH. Jego zainteresowania skupiają się wokół algorytmów obliczeniowych oraz ich realizacji w układach reprogramowalnych.

e-mail: Grzegorz.Sulkowski@cyfronet.pl

Streszczenie

W niniejszym artykule autorzy przedstawiają wyniki prac badawczych związanych z budową sprzętowego klasyfikatora portów sieciowych. Opracowana koncepcja filtru portów opiera się na wykorzystaniu elementarnych pamięci RAM16X1D dostępnych w układach FPGA z rodziny Virtex firmy Xilinx. Uzyskana wydajność przetwarzania danych, przekraczająca 160 milionów pakietów na sekundę oraz pozytywnie rezultaty wstępnych testów praktycznych, stwarzają możliwości zastosowania rozwiązania we współczesnych sieciach teleinformatycznych o dużych przepustowościach.

Słowa kluczowe: systemy bezpieczeństwa informatycznego, układy programowalne, języki opisu sprzętu, Ethernet, firewall.

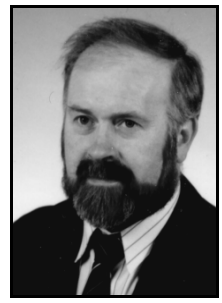
High-speed network port filtering in a hardware Firewall security system

Abstract

The paper presents the results of practical realization of the network ports classifier based on cascades of RAM16X1D memory available in Xilinx Virtex FPGA chips. The first section introduces a packet classification subject. The second one describes the packet classifier internal structure, characterizing in details each of the elements included in the classifier, according to the block diagram of Fig. 1. The network port filter architecture (shown in Fig. 2) assumed by the authors is discussed in the section 3. The section 4 contains details concerning the basic filtering element functionality and implementation method. The last section summarizes the results obtained. The new architecture of the ports classifier based on RAM16X1D storage elements adopted by the authors allows achieving the high speed data processing. The estimated maximum operating frequency for the ports filter is 160 MHz. It means that the module can analyze about 160 million packets per second. The research work is in line with the rapidly developing trend towards using reprogrammable logic for securing data transfer in information technology networks.

Keywords: IT Security Systems, Programmable Logic, Hardware Description Language, Ethernet, Firewall.

Prof. dr hab. inż. Kazimierz WIATR



Studia AGH Kraków (1980), dr nauk technicznych (1987), dr habilitowany (1999) i profesor (2002). Profesor zwyczajny na AGH w Krakowie oraz Dyrektor Akademickiego Centrum Komputerowego Cyfronet AGH. Prowadzone prace badawcze dotyczą systemów wizyjnych, systemów wieloprocesorowych, rekonfigurowanych systemów obliczeniowych i sprzętowych metod akceleracji obliczeń. Jest autorem trzech monografii, w tym najnowsza Akceleracja obliczeń w systemach wizyjnych wydana przez WNT w roku 2003.

e-mail: wiatr@agh.edu.pl

1. Wstęp

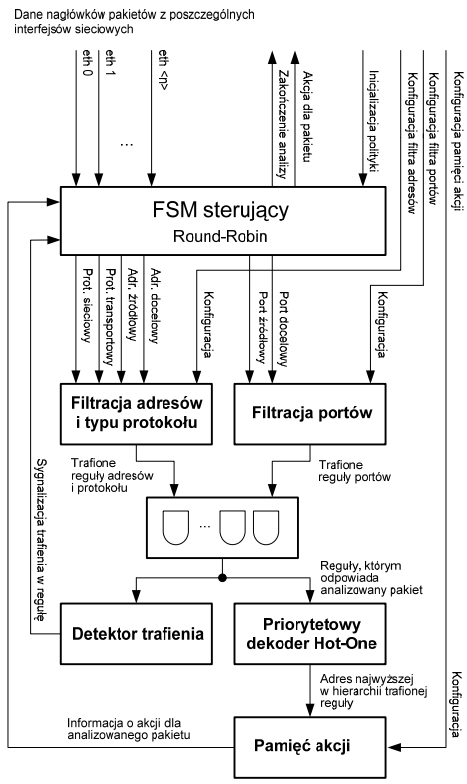
Zadaniem klasyfikatora pakietów w sprzętowym systemie bezpieczeństwa typu Firewall jest weryfikacja zgodności przetwarzanych danych z obowiązującym schematem polityki bezpieczeństwa. Dokonuje się ona poprzez przyporządkowanie analizowanych pakietów do zestawu odpowiadających im reguł bezpieczeństwa na podstawie informacji zawartych w nagłówkach protokołu IP (ang. *Internet Protocol*). Ze względu na uwarunkowania implementacyjne proces klasyfikacji podzielony jest na dwa odrębne obszary: adresację sieciową wraz z informacją o typie protokołu oraz porty sieciowe. Do analizowania adresów sieciowych świetnie nadają się pamięci trójwartościowe TCAM (ang. *Ternary Content-Addressable Memory*) [1]. Pomimo wielu zalet nie są one optymalnym rozwiązaniem dla celów klasyfikacji portów, w szczególności dla reguł zawierających ich zakresy. Niestety, takie przypadki są powszechne w produkcyjnych systemach Firewall, gdzie dla uzyskania zamierzonego kształtu polityki bezpieczeństwa administratorzy wielokrotnie posługują się definicjami zawierającymi szerokie przedziały portów. Aktualnie wiele zespołów badawczych na całym świecie koncentruje się na poszukiwaniu szybkich oraz efektywnych od strony zapotrzebowania na zasoby pamięciowe alternatywnych rozwiązań analizowania zakresów portów.

2. Klasyfikator pakietów

Poglądowy schemat blokowy modułu klasyfikatora pakietów został przedstawiony na rys. 1. Niezbędne do przeprowadzenia weryfikacji przetwarzanych pakietów informacje wejściowe przekazywane są do modułu ze specjalnych bloków pamięci ramkowej, szczegółowo opisanych w pozycji [2]. Opracowane przez autorów rozwiązanie pozwala na równoczesną analizę danych pochodzących z wielu interfejsów sieciowych przy wykorzystaniu algorytmu karuzelowego (ang. *Round-Robin*), sprawdzającego cyklicznie dostępność nowych deskryptorów Jeżeli dla aktualnie monitorowanego wejścia zgłoszony zostanie sygnał obecności deskryptora, główny automat sterujący rozpoczyna procedurę weryfikacji pakietu, blokując na czas niezbędny do jej przeprowadzenia proces kolejkowania. Odczytywane deskryptory bezpieczeństwa składają się z następujących pól nagłówków przetwarzanych pakietów:

a) typu protokołu sieciowego (16 bitów),
b) typu protokołu transportowego (8 bitów),
c) adresu źródłowego (32 bity),
d) adresu docelowego (32 bity),
e) numeru portu źródłowego (16 bitów),
f) numeru portu docelowego (16 bitów).

Pierwsze cztery pola o łącznej długości 88 bitów trafiają do modułu filtrującego adresy oraz typ protokołu. Dwa ostatnie, o łącznej długości 32 bitów, przekierowywane są do modułu filtrującego porty. Ponieważ wynikowa informacja o trafionych regułach generowana jest na podstawie iloczynu wektorów pochodzących z dwóch niezależnych bloków filtrujących, niezbędne jest, aby każdy z nich dostarczał informację wyjściową w formie binarnej niekodowanej (poszczególnym regułom odpowiadają dedykowane wyjścia sygnałowe). Wynik iloczynu logicznego jest następnie zamieniany w priorytetowym dekoderze „gorącej jedynki" (ang. *hot one*) na adres binarny znajdującej się najwyższej w hierarchii trafionej reguły. Na jego podstawie z pamięci akcji odczytywana jest informacja o dalszym postępowaniu z analizowanym pakietem. W obecnej implementacji możliwe są dwa scenariusze: odrzucenie bądź akceptacja i w jej efekcie retransmisja pakietu. Równocześnie w bloku detektora trafienia generowany jest sygnał potwierdzający wystąpienie przynajmniej jednej reguły, której odpowiada analizowany pakiet. W wypadku gdyby taka nie istniała, pakiet domyślnie ulega odrzuceniu. Ostatecznie informacja o zakończeniu analizy wraz z potwierdzeniem akcji trafia z modułu klasyfikatora do bloku pamięci ramkowej. Dla pakietów zaakceptowanych rozpoczyna się wówczas procedura przesyłania danych z pamięci do interfejsu nadawczego, zaś w przypadku odrzucenia pakietu, odpowiadająca mu strona pamięci zostaje zwolniona [2].



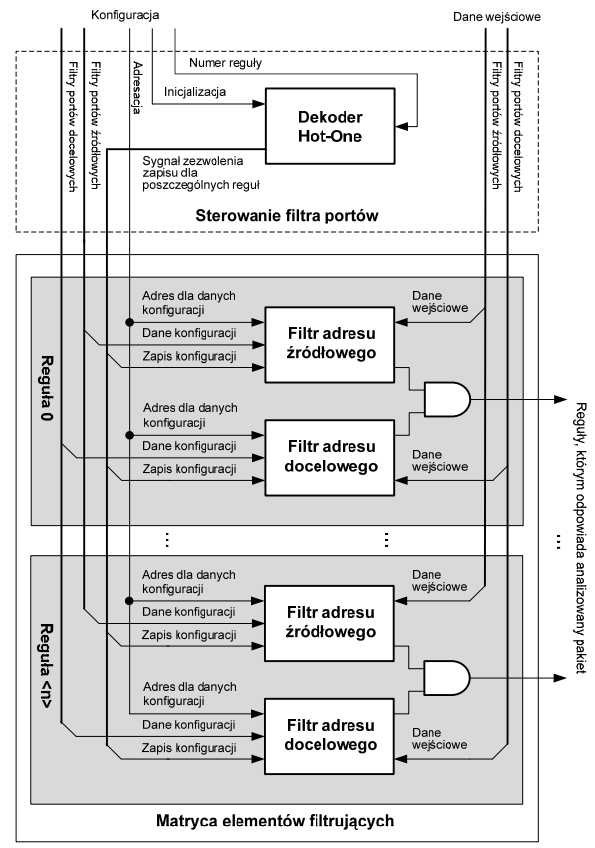
Rys. 1. Schemat blokowy modułu klasyfikatora pakietów
Fig. 1. Block diagram of the packet classifier

3. Filtr portów sieciowych

Koncepcja realizacji szybkiego i skalowalnego filtra portów opiera się na zastosowaniu równoległego przetwarzania tabeli definicji zasad bezpieczeństwa. Kluczowym elementem opracowanego rozwiązania, przedstawionego schematycznie na rys. 2, jest matryca elementarnych bloków filtrujących. Każdy wiersz odpowiadający pojedynczej regule zawiera po dwa takie elementy: jeden weryfikujący porty źródłowe, drugi zaś porty docelowe. Końcowy rezultat klasyfikacji jest wynikiem iloczynu logicznego wyjść bloków filtrujących.

Jeżeli sygnał inicjalizacji polityki bezpieczeństwa jest w niskim stanie logicznym, filtr portów pracuje w trybie odczytu. Wówczas część deskryptora bezpieczeństwa o długości 32 bitów, zawierająca informację o portach źródłowych i docelowych, zostaje podana na wejście matrycy elementów filtrujących. Na jej wyjściu pojawia się informacja o regułach, którym odpowiada weryfikowany pakiet. Czas trwania procesu odczytu wynosi jeden cykl zegara. Jest on niezależny od liczby reguł, jak również od ilości i szerokości zakresów portów w nich zdefiniowanych.

Dla aktywnego sygnału inicjalizacji filtr przechodzi do trybu zapisu konfiguracji wewnętrznej. Wtenczas na wejścia matrycy elementów filtrujących podawane są z bloku sterującego adresy inkrementowane w zakresie od 0 do 15. Służą one wpisaniu do poszczególnych komórek pamięci filtra danych konfiguracji wynikającej z obowiązującego schematu polityki bezpieczeństwa. Na podstawie numeru reguły podawanego na wejście bloku sterującego, dekoder „gorącej jedynki" generuje sygnał zezwolenia na zapis odpowiedniego wiersza. Proces zapisu definicji pojedynczej reguły zajmuje 16 cykli zegarowych.



Rys. 2. Schemat blokowy filtra portów sieciowych
Fig. 2. Block diagram of the network port filter

4. Budowa elementu filtrującego

Element filtrujący, którego schemat wewnętrzny przedstawiono na rys. 3, wykorzystuje ideę łańcucha komparatorów, zaprezentowaną w [3]. Wariant opracowany przez autorów nie bazuje jednak na zmodyfikowanej pamięci TCAM, lecz opiera się na zastosowaniu kaskad dwuportowych pamięci RAM16X1D, wchodzących w skład zasobów sprzętowych układów reprogramowalnych FPGA produkcji firmy Xilinx. Poszczególne pamięci pracują jako elementarne komparatory zakresów cząstkowych.

Działanie czterobitowego komparatora 0, złożonego z pary pamięci RAM_A_0 oraz RAM_B_0 (rys. 3), można przedstawić za pomocą następującej zależności:

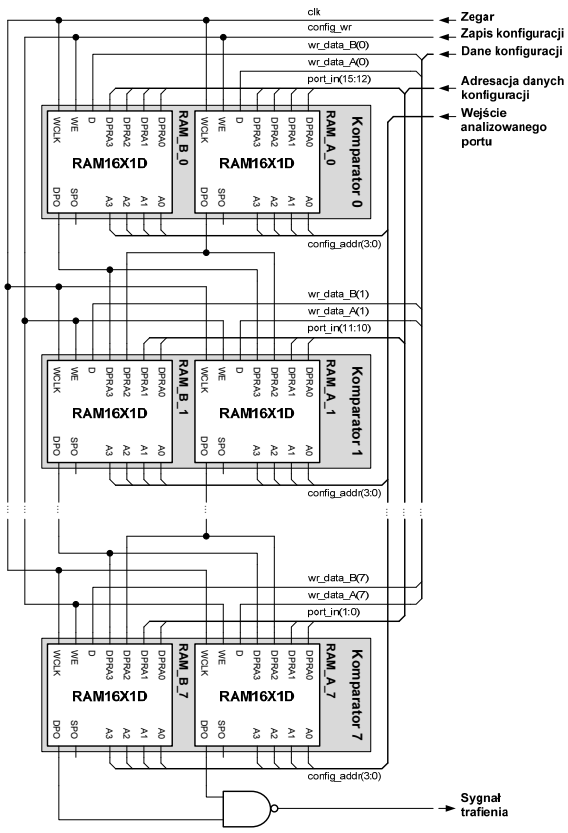
$$Q_{BA0} = f_0(D_0, L_0, H_0), \tag{1}$$

gdzie:

Q_{BA0} – stan wyjścia komparatora 0,

$$f_0(D_0,L_0,H_0)=\begin{cases}00, & (D_0=L_0\wedge H_0>L_0)\vee(D_0=L_0=H_0)\\ 01, & L_0<D_0<H_0\\ 10, & D_0=H_0\wedge H_0>L_0\\ 11, & D_0<L_0\vee D_0>H_0\end{cases}, \quad (2)$$

D_0 – dane wejściowe komparatora 0,
L_0 – dolna granica zakresu cząstkowego komparatora 0,
H_0 – górna granica zakresu cząstkowego komparatora 0.



Rys. 3. Schemat struktury wewnętrznej pojedynczego elementu filtrującego
Fig. 3. Diagram of the single filtering element internal structure

Szerokość analizowanego zakresu dla komparatora 0, oznaczona jako W_0, wynosi:

$$W_0=\begin{cases}H_0-L_0, & (H_0-L_0)\le 2\\ 2, & (H_0-L_0)>2\end{cases}, \quad (3)$$

Pozostałe komparatory, oprócz dwubitowych danych wejściowych, wykorzystują również informację o stanie wyjść komparatora poprzedzającego $Q_{BA\ i-1}$ oraz o szerokości poprzedzającego zakresu cząstkowego W_{i-1}. Do opisu sposobu ich działania zdefiniujmy następujące funkcje:

$$f_1(D_i,L_i,H_i)=\begin{cases}00, & (D_i=L_i\wedge H_i>L_i)\vee(D_i=L_i=H_i)\\ 01, & L_i<D_i<H_i\\ 10, & D_i=H_i\wedge H_i>L_i\\ 11, & D_i<L_i\vee D_i>H_i\end{cases}, \quad (4)$$

$$f_2(D_i,L_i,H_i)=\begin{cases}00, D_i=L_i\\ 01, D_i>L_i\\ 11, D_i<L_i\end{cases}, \quad f_3(D_i,L_i,H_i)=\begin{cases}01, D_i<H_i\\ 10, D_i=H_i\\ 11, D_i>H_i\end{cases}, \quad (5)\ (6)$$

$$W_i=\begin{cases}H_i-L_i, & L_i<H_i\wedge(H_i-L_i)\le 2\wedge(H_i-L_i)\ge W_{i-1}\\ W_{i-1}, & L_i<H_i\wedge(H_i-L_i)\le 2\wedge W_{i-1}>(H_i-L_i)\\ 2, & L_i>H_i\vee(H_i-L_i)>2\end{cases}, \quad (7)$$

gdzie:

W_i – szerokość przedziału cząstkowego dla i-tego komparatora,
D_i – dane wejściowe,
L_i – dolna granica zakresu cząstkowego,
H_i – górna granica zakresu cząstkowego.

Możemy wyróżnić trzy przypadki funkcjonowania komparatora $Q_{BA\,i}$ dla $i>0$ w zależności od szerokości przedziału poprzedzającego:

a) dla $W_{i-1}=0$

$$Q_{BA\,i}=\begin{cases}f_1(D_i,L_i,H_i), & Q_{BA\,i-1}=00\\ 11, & Q_{BA\,i-1}=11\end{cases}, \quad (8)$$

b) dla $W_{i-1}=1$

$$Q_{BA\,i}=\begin{cases}f_2(D_i,L_i,H_i), & Q_{BA\,i-1}=00\\ f_3(D_i,L_i,H_i), & Q_{BA\,i-1}=10\\ 11, & Q_{BA\,i-1}=11\end{cases}, \quad (9)$$

c) dla $W_{i-1}=2$

$$Q_{BA\,i}=\begin{cases}f_2(D_i,L_i,H_i), & Q_{BA\,i-1}=00\\ 01, & Q_{BA\,i-1}=01\\ f_3(D_i,L_i,H_i), & Q_{BA\,i-1}=10\\ 11, & Q_{BA\,i-1}=11\end{cases}. \quad (10)$$

Analizowany port odpowiada danej regule, jeżeli stan wyjść ostatniego komparatora jest różny od wartości „11”. Sygnał trafienia dla takiego przypadku generuje dodatkowa bramka NAND.

Ponieważ pojedynczy port sieciowy ma szerokość 16 bitów, stąd też element filtrujący składa się z 7 komparatorów (1 czterobitowy oraz 6 dwubitowych). Każdy z nich z kolei wykorzystuje po dwie pamięci RAM16X1D, co daje sumaryczną liczbę 56 pamięci na jedną regułę opisującą dwa porty: źródłowy i docelowy. Pamięci dwuportowe (litera D w nazwie komponentu) wybrano ze względu na łatwiejszą realizację procesu zapisu konfiguracji.

5. Wyniki implementacji

Opisane rozwiązanie zostało zaimplementowane i przetestowane praktycznie przy pomocy płyty uruchomieniowej Digilent XUP z układem Virtex II Pro XC2VP30. Badany moduł o pojemności 32 reguł alokował 1833 bloki Slice, co stanowi 13% wszystkich dostępnych zasobów sprzętowych układu XC2VP30. Maksymalna częstotliwość jego pracy wynosi 162MHz. Ponieważ proces weryfikacji zajmuje jeden cykl zegara, teoretyczna przepustowość opracowanego filtra portów to około 160 milionów pakietów na sekundę. W sieci 10Gb Ethernet największa dopuszczalna liczba ramek przesyłanych w ciągu sekundy to 14 880 952,38 fps (ang. *frame per second*). Wynika stąd, że omawiany moduł filtrujący bez problemu i ze znacznym zapasem wydajności jest w stanie znaleźć zastosowanie do weryfikacji danych we współczesnych sieciach teleinformatycznych o dużych przepustowościach.

Praca naukowa finansowana ze środków na naukę w latach 2006-2008 jako projekt badawczy.

6. Literatura

[1] P. Gupta and N. McKeown: Algorithms for packet classification, IEEE Network, vol. 15, no. 2, pp. 24–32, Mar./Apr. 2001.
[2] G. Sułkowski, M. Twardy, K. Wiatr: Wielościeżkowe równoległe przetwarzanie danych w sprzętowym systemie bezpieczeństwa klasy Firewall, Konferencja KNWS’08, Miesięcznik Przegląd Telekomunikacyjny nr 6, Wydawnictwo SIGMA NOT, Warszawa 2008, s. 726-728.
[3] E. Spitznagel, D. Taylor, J. Turner: Packet Classification Using Extended TCAMs, Proceedings of the 11th IEEE International Conference on Network Protocols, p.120, November 04-07, 2003.

Artykuł recenzowany