High-Performance Multi-Pattern Matching Structure in Hardware Network Firewall

Wang Jie, Ji Zhen-zhou, Hu Ming-zeng
Department of Computer Science and Technology
Harbin Institute of Technology
No.92 Xidazhi Street, Harbin, Heilong
China
{wj, jzz, mzhu}@pact518.hit.edu.cn

Abstract: - Along with development of network techniques and firewall techniques, multi-pattern matching for content filtering, virus detection and other network security measures is emphasized in the field of hardware firewall. Principles of multi-pattern matching on network firewall based on FPGA (Field Programmable Gate Array) are presented: low latency, supporting non-fixed-length pattern and forward matching. And architecture of high-performance matching on data link layer forming pipeline with frame transmission is proposed especially for multi-port scheduling. By academic analysis and experimental validation it can support applications of high-performance hardware network firewall and effectively reduce additional delays by expand logics.

Key-Words: - FPGA, Hardware network firewall, Multi-pattern matching, Servos’ array Aho-Corasick algorithm

1 Introduction

With development of network techniques and the rapid increase in network bandwidth, network security is becoming increasingly severe. Network security and content security are emphasized in Internet applications. Multi-pattern matching algorithms are widely applied to intrusion detection, firewall, routing, content filtering, virus detection, and other network security techniques[1]. With development and promotion of FPGA techniques, FPGA-based network hardware market is more and more broad. With clear declaration of Cisco in 2003 that firewall products should have content filtering, a great number of hardware firewalls have integrated security techniques based on multi-pattern matching. The most important measure is to deduce high-performance hardware architecture and rapid ASIC (Application Specific Integrated Circuit) content filtering.

Single-pattern matching algorithms such as KMP (Knuth-Morria-Pratt) algorithm, BM (Boyer-Moore) algorithm laid the foundation for the development of multi-pattern algorithms. Because multi-pattern matching is more complicated, and applies to more advanced applications, more research focuses multi-pattern matching. AC (Aho-Corasick) algorithm[2-5], TCAM-based search algorithm[6-8] and Bloom Filter approach[9-10] are common hardware multi-matching algorithm. In recent years, along with continuous improvement of FPGA techniques and memory technology, hardware multi-pattern matching algorithms attracts by more and more attention.

With development of network firewall architecture, other network security products currently integrated into the main direction of development request to establish a firewall at the core of the network security system. The argument on the dominant controversy of ASIC has evolved into how to use the advanced technology integration of a comprehensive defense system. Study on pattern-matching of firewall is not only to strengthen firewall function, but also to build a bridge to connect different techniques.

At present many commercial firewalls have contained content filter, but executive efficiency is usually unsatisfied. When content filter turns on, transmit speed will decrease 1-2 factor of the order, and complex and huge hardware resources will be needed. In this paper, architecture of high-performance matching on data link layer forming pipeline with frame transmission is proposed especially for multi-port scheduling to greatly enhance transmit efficiency.

2 Multi-pattern matching of hardware network firewall

Firewall is designed originally to access control. Packet filtering and stateful inspection are designed
based on classification algorithm, and rules set matching. And application-level content filtering, virus detection and other functions needed to deal with a large number of additional logic and memory, which greatly reduce the performance of the firewall. At present, hardware pattern-matching algorithms of intrusion detection system are emphasized, such as TDP-DFA multi-state machine[11] that makes multi-pattern matching performance of intrusion detection system have been greatly improved. However, network firewall does not make a lot of good for transplantation for features its own features.

Multi-pattern-matching of hardware network firewall should meet the following principles:

(1) Low latency
Firewall is a type of network equipment by store and forward, such as a typical three-port firewall with screened subnet. So content filtering, virus detection and so on based on multi-pattern matching should not bring too much delay, otherwise it will severely reduce the network performance. And generally packets should not be transmitted under the out-of-order. In order to reduce the delay in forwarding, multi-pattern matching algorithms and storage strategy are closely related. This is the most critical indicators of availability of multi-pattern matching. But unfortunately, to many of the current hardware firewall products, opening content filtering function will increase the delay time with adding one magnitude.

(2) Support non-fixed-length pattern
Pattern length of the multi-pattern matching should be any length less than the largest text string. Rules set of matching algorithms usually apply to fixed-length matching or non-fixed-length prefix matching of short patterns. It can not suitable for virus characteristics sequence detection and text content filtering, and also makes TCAM-based search algorithm more complicated because it usually needs more accesses.

(3) Forward matching
Reverse matching algorithms require firewall to receive a complete packet at least before matching, and may take on many occasions packet buffer. So it will bring more scheduling difficulties and increase transmitting delay. Forward matching algorithms can start matching from the initial reception and are available for pipeline operations.

This article focuses on the utilization of FPGA on-chip memory resources to implement fast and efficient multi-pattern matching structure for content filtering.

3 Multi-pattern matching of data link layer
For storage-forward network firewall, usually it does not require restoring protocol stack and stitching fragments of IP packets. Matching of data link layer cannot miss any pattern that would appear in packets. It is more rapid and effective means to realize the underlying pattern-matching with using FPGA based dedicated hardware logic.

3.1 Applicability of Aho-Corasick algorithm
AC algorithm is based on the finite state automata. First it builds a tree finite-state automaton (FSA) by preprocesses pattern set. On the basis of the FSA, all potential matching can be found just scan the whole text once. Time complexity of AC algorithm is O(n), and space complexity is O(l), l is the length of all patterns.

AC algorithm supports well non-fixed-length and forward match principle that precedes other pattern-matching algorithms. Due to mach each of the whole text, the worst and the best time complexity is both O(n). It does not seem to follow low latency principle. But in fact, with adoption of reasonable storage management, receiving data frame of MAC (Media Access Control) runs parallel to automata matching by pipeline operations. Features of structure design will be specifically described in the next section.

AC preprocessing generates a conversion chart by software, and its storage structure is two-dimensional sheet directly on the FPGA configuration. Thus preprocessing time can be ignored which greatly simplifies hardware design. We jest only consider storage and access two-dimensional sheet.

Major bottleneck of AC algorithm is expansion of storage space. With development of FPGA technology, capability of on-chip SRAM and off-chip SRAM (such as ZBT SRAM) increase rapidly, which reduces difficulty of storage management and cheaper than TCAM. On the other hand, combined to hash compress and regular matching, storage space can reduce further.

3.2 FPGA on-chip resource planning
To Gigabit Ethernet, since receiving one byte in one cycle, it needs m cycles to e store a frame (m is the frame length), and an illegal frame is discarded immediately. If Aho-Corasick algorithms deals with one byte in each cycle, receiving and matching can carry at the same time, while fortunately Block RAMs in FPGA can also be access in one cycle.
Suppose pattern set \( P = \{ p_1, p_2, ..., p_k \} \), and pattern \( p_i \) is frame segment \( T = T[1...n] \) for a given n-length footage composed of character set \( \Sigma \). Each character of the frame segments is also in set \( \Sigma \). \( p_i \) is the length of \( l_i \), \( S \) is state number of the automata, then

\[
S \leq \sum_{i=1}^{k} \frac{l_i + 1}{2}
\]

(1)

a (Width of state coding) + b (width of input character) is used for address to access two-dimensional table, \( M \) is size of the table, then

\[
a = \left\lfloor \log_2 S \right\rfloor
\]

(2)

\[
M = a * 2^{a+b}
\]

(3)

Since the width of each unit in Block Rams must support

\[
c = 2^j + \left\lfloor 2^j - 3 \right\rfloor, \quad j \in \{1,2,3,4,5\}
\]

(4)

And

\[
c \geq a
\]

(5)

When the size of a single Block Ram that is \( r = 18 \) Kbits (Xilinx Virtex series) which can not support necessary of \( M \) (usually satisfied), more Block Rams in series or in parallel to store the two-dimensional table of the state automata. So with increasing of \( S \) it needs to adjust \( c \), and storage space will be expanding rapidly.

Under this structure, if \( j \) is too small, few states will be supported. So only when \( j \geq 2 \), it can be applied. \( R \) is the required number of Block Rams, then

To address this issue, for example, set \( j = 4 \) and the total length of all patterns is 1KBytes, at this time \( c = 18 \), the followings are considered:

\[
R = 2^{a+b} - \log_2 (r/c)
\]

(6)

To address this issue, for example, set \( j = 4 \) and the total length of all patterns is 1KBytes, at this time \( c = 18 \), the followings are considered:

1. For Fast Ethernet, half a byte (nibble) is received in each cycle, then \( b = 4 \), then according to formula (4) and formulas (5), when \( a = 11 \), it can be calculated that \( R = 16 \).

2. For Gigabit Ethernet, \( b = 8 \), in accordance with the above-mentioned method, when \( a = 10 \), it can be calculated that \( R = 256 \). Only a small number of advanced chips can support it because huge internal resources are required.

3. In this article we mainly discuss network firewall, so design of algorithms should be able to support multi-port applications. Set three-port gigabit network firewall with screened subnet as an example. Each port is independent to receive frames, so multi-pattern matching is an independent state machine and memory resources increase in direct proportion to number of firewall ports. Generally as \( a < c \), each memory unit of Block Rams is idle causing a waste of space. So in order to economize on-chip memory resources, we propose the two improvements:

I. Frequency doubling of state machine

So set \( b = 4 \), then \( a = 11 \), then a nibble is be sent to match and the formula (1) will be modified as

\[
S \leq 2^{k} \sum_{i=1}^{k} \frac{l_i + 1}{2}
\]

(7)

According to the formula (7), one port needs only 32 Block Rams to the same pattern set with lengths of 1Kbytes. But frequency doubling maybe result in instability of hardware logic, and additional units are required for clock management and data latches.

II. Additional off-chip ZBT SRAM

Timing for reading or writing of ZBT SRAM timing is similar to on-chip Block Rams, and capacity of single chip ZBT SRAM is up to 8Mbits at least.

3.3 Hardware logic of multi-pattern matching

(1) Preprocess

According to improved AC algorithm, put all patterns of \( P = \{ p_1, p_2, ..., p_k \} \) into state machine and configure it on FPGA.

(2) Matching

Process of frame receiving and multi-pattern matching is as shown in Fig.1 (not including head processing). Every byte of frames is written into buffer and at the same time the address to two-dimensional table is created with current state from state machine. When the frame is illegal or matched successfully by state machine (assuming strategy for matching successfully is throwing away), the current frame in buffer is rejected and state machine is reset.

![Fig.1 Flow chart of frame receiving and multi-pattern matching](image-url)
As multi-pattern matching and frame receiving are parallel by pipeline, although time complexity of multi-pattern matching is O(n), extra time for matching to the whole firewall system almost can be ignored. Algorithm description of multi-pattern matching part is showed as Fig.2.

3.4 Multi-pattern matching based on reconstruction servos’ array of multi-pattern matching

In order to expand functions, dynamic servos’ array can configure more state machines and all packets carry unified multi-pattern matching. FPGA resources are divided into small k servos. According to size of state machine generated by preprocess, reconstruction of servos is determined to enhance parallelism degree of AC algorithm. The greatest degree is k when every servo is an independent state machine, so k packets can be scheduled to multi-pattern matching; and the minimum degree is 1 when all the servos form one state machine, then only one packet can be scheduled. Though servos’ array can implement reconfigurable hardware, a great lot of resources are still required to large pattern set. And the modified methods in section 3.2 also apply to servos’ array.

Time complexity of servos’ array algorithm is O(n/k) and space complexity is O(kl). Results similar to section 3.2 can be obtained if servos’ array combines with frame receiving, and because other technical means have rejected a lot of frames, operations of look-up table will be significantly reduced.

In short, filtering on data link layer filter specifically can tap multi-pattern matching and data storage in parallel and effectively reduce additional delay costs.

4 Verification and analysis of Hardware multi-pattern matching

It is verified on Xilinx xc4vlx80-12ff1148 for simulation, synthesis, placement and routing. Table 1 shows space of multi-pattern matching on data link layer of three-port Gigabit Ethernet. The minimum pattern length is 3Bytes and the largest is 128Bytes. a is width of state coding, b is width of input character, c is width unit of Block Rams, and R is number of Block Rams.

We can see that this algorithm supports well for small pattern set. When the pattern set is greater than 256Bytes, as a result of restriction on the size of each unit in Block Rams and rapid growth of resources, look-up table of doubling frequency and ZBT SRAMs can be used to make up some drawbacks, and is a necessary means to support big pattern set. Table 2 shows resources report of the first case in table 1.

<table>
<thead>
<tr>
<th>patterns length(Bytes)</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>16*3</td>
</tr>
<tr>
<td>512</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>4*3</td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>4</td>
<td>18</td>
<td>16*3</td>
</tr>
<tr>
<td>2048</td>
<td>11</td>
<td>4</td>
<td>18</td>
<td>32*3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part</th>
<th>xc4vlx80-12</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK-Estimated Frequency</td>
<td>150.4MHz</td>
</tr>
<tr>
<td>CLK-Requested Frequency</td>
<td>125.0MHz</td>
</tr>
<tr>
<td>Block Rams</td>
<td>50 of 200 (25%)</td>
</tr>
<tr>
<td>Total Luts</td>
<td>5214 (7%)</td>
</tr>
</tbody>
</table>

By theoretical prediction and experimental proof, hardware multi-pattern matching based on Aho-Corasick supports well for content filtering on network firewall, accords with principle of multi-pattern matching of firewall, and reduces greatly storage-forward delays. Additional delays of multi-pattern matching on data link layer are almost zero in possession of good feasibility. For firewall
processing packets, time of data access and time of matching are considered together.

However, there are also some shortcomings on efficiency of storage in this strategy mainly due to memory cell structure of Block Rams and ZBT SRAM leading to a low utilization rate. At the same time the same state coding in a large number of memory units with different address also reduces the space utilization. Therefore HASH can compress address space if dealing well with synchronization of frame receiving and state matching.

5 Conclusion
In this paper, principles to design algorithms for multi-pattern matching in network hardware firewall are proposed: low latency, non-fixed-length patterns support and positive match, which are fit for high-performance content filtering. On data link layer frame transmission and multi-pattern matching carry at the same time by pipeline operations specifically to reduce maximum of time cost for matching.

References: