Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp440-444)

Distribution Strategies for Parallel VPN Servers

Myeonggil Choi, Hyoun Cheul Kim, Youngsun Choi, SANGMUN SHIN
Dept. of Systems Management Engineering
INJE University
607, Obang-dong, Gimhae, Gyeongnam, 621-749
SOUTH KOREA

Abstract: - The load-balancing problem of parallel VPN configuration is formulated and solved in consideration of the characteristics of VPN services in this paper. Herein, four load-balancing strategies are suggested for VPNs. The response-time performances and the load-uniformity performance of the proposed strategies are measured as results of load balancing in VPNs.

Key-Words: - VPN, IPsec, Load-Balancing, Encryption Algorithm, SA, Load-Balancing Strategy

1 Introduction

Many organizations have adopted Virtual Private Network (VPN) for security. The performance of the VPN server is critical to reliable communication between parties [1]. We focus on the IPSec-based VPN in this paper. To increase the processing speed of VPNs, multiple IPSec modules should be adopted in a VPN server or multiple VPN servers should be adopted in a parallel manner by using a load balancer. The first is not effective due to the capacity limit of the CPU controlling IPSec modules and the capacity limit of the network interface. The second has been effectively used to improve the performance of a VPN. In this letter, we formulate the load-balancing problem of VPNs and suggest four practical load-balancing strategies for VPNs.

Load-balancing architectures for web server systems are surveyed by Cardellini et al [2]. Bestavros et al. propose a distributed approach in which all hosts of the distributed system participate in connection routing [3]. Wolf and Yu formulate the load-balancing problem and propose a practical scheme that attempts to optimally balance the load on the servers of a clustered web [4].

However, load-balancing schemes used in web servers are not suitable for VPN servers because the available information and operational process are different between these two servers. VPN servers provide Security Association (SA) services according to the security policies negotiated by end-parties; SA denotes the suit of Encapsulating Security Protocol (ESP) and Authentication Header (AH) [5]. Each SA causes different load according to its ESP-AH suit. Thus the balancer in a VPN should distribute these SAs among VPN servers. The setup of a SA requires expensive bandwidth and time resources.

For this reason, long-lasting SAs are efficient. Consequently, end-parties maintain their SA during a negotiated long period. Moreover, in a VPN, the actual service unit is not a SA but application data packet which is generated by the end-parties. The application data packets belonging to the same SA should be served through the same ESP and AH algorithms. The loads of application data packet vary individually due to their size. Therefore the balancer in a VPN should consider both the required load of the ESP-AH suit for each SA and the load caused by application data packets of each SA.

IPsec is well defined protocol. The method of protecting IP datagram or upper-layer protocols involves the use of one of the IPSec protocols, ESP or AH. AH provides proof-of-data origin on received packets, data integrity, and anti-replay protection. ESP provides all that AH provides in addition to optional data confidentiality and limited traffic flow confidentiality [6].

There are many kinds of load-balancing architectural styles used on the Web. We consider the dispatcher-based architecture for parallel VPN servers. It can be easily built and operated by an organization without complex connection with a public network. We assume that the VPN dispatcher has a policy table that describes to which SA each incoming packet belongs and by which VPN server it should be served. When packets arrive at the VPN dispatcher, the VPN dispatcher parses packets and then forwards the packets to the corresponding VPN server. The dispatcher-based architecture in a VPN is shown Figure 1.

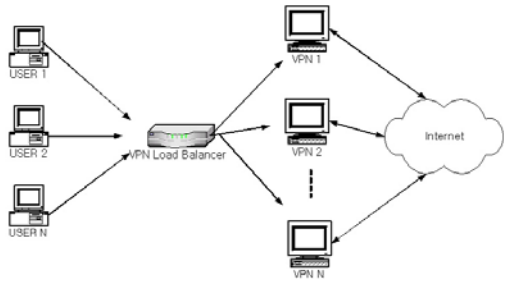


Fig. 1. Considered load-balancing architecture

2 Load-Balancing for VPNs

The goal of our load-balancing is to minimize the average response time for user requests which was served by corresponding SA in a VPN. However, the load in each VPN server must be less than the predefined maximum load because the delays of packets in an over-loaded server may increase rapidly. In addition, a SA should be assigned to exactly one server. We formulate the load-balancing problem as

$$\min \quad \sum_{j=1}^{N} R_j\left(\left(\sum_{k=1}^{M'} c_{kj}\cdot a'_k\right)+\sum_{i=1}^{M} x_{ij}\cdot a_i\right) \quad (1)$$

$$s.t. \quad \sum_{k=1}^{M'} c_{kj}\cdot a'_k+\sum_{i=1}^{M} x_{ij}\cdot a_i \le L_j, \qquad \forall j \quad (2)$$

$$\sum_{j=1}^{N} x_{ij}=1, \qquad \forall i, \quad (3)$$

where the binary decision variable denotes assignment for new SA to VPN server , the binary indicator denotes past assignment for the established SA to VPN server , denotes the predicted load of a new SA , denotes the observed load of the established SA, is the number of new SAs, is the number of established SAs, is the number of VPN servers, and is the expected response time function of the VPN server. Associated with each VPN server , the function measures the expected response time which is increasing and convex under certain very modest conditions.

We can distribute loads among VPN servers by solving the problem periodically for each planning period. The formulation does not reallocate already established SAs due to the excessive cost of reallocation [4]. Nonetheless, the formulation could be extended to the case of reallocation by just taking the as decision variables.

However, because a new SA is set up occasionally in a VPN, there is no request of new SA in many planning period or some requests of new SAs are delayed until next optimization time. For this reason, we propose to solve the problem in one-by-one fashion whenever a new SA needs to be set up. In this case, we need only allocate a new SA to one of the VPN servers. When the VPN dispatcher assigns a new SA to a VPN server, the criteria to select a VPN server is the increase of expected response time for each VPN server. The increase depends on the current load of each VPN server and the unknown load of the new SA; this means that the coefficients and is critical for the solution. Theoretically, these can be obtained by the queuing analysis. But, a simple, practical method to grasp the approximated loads is needed for rapid implementation. We suggest the following four strategies approximating the coefficients and . Then we will solve the load-balancing problem in one-by-one fashion according to the strategies.

Strategy 1 : The number of ongoing SAs in a VPN server can be considered as the current load of the VPN server. In the same context, the load of the new SA can be set to 1. This strategy is based on the assumption that all the loads of SAs are identical. For this strategy, and can be set to 1 in the load-balancing problem. The number of the loads of SAs served by a certain VPN server can be considered as the current load of the VPN server.

Strategy 2 : However, the loads of SAs are not the same. Different SAs adopt different encryption algorithms, i.e., ESP-AH suit, according to the negotiation between the end parties. Assuming that the packet arrival rates and the packet sizes of the SAs are the same, the packet processing time of a SA can be considered as the load of the SA. In the same manner, the summation of the loads of SAs served by a certain VPN server can be considered as the current load of the VPN server. For this strategy, is the packet processing time of a newly arrived SA , and is the packet processing time of the ongoing SA in the load-balancing problem in the given packet arrival rate and packet size.

Strategy 3 : However, the load of a SA depends on its packet arrival rate and average packet size, which are different from SA to SA and could be observed by the VPN dispatcher. For the new SA, the historical data of early established SAs should be used for prediction. The summation of the loads of SAs served by a certain VPN server can be considered as the current load of the VPN server. For this strategy, in the load-balancing problem we can set to multiplication of the packet arrival rate, average packet size and the corresponding packet processing time of SA .

Similarly, we can set to multiplication of the predicted packet arrival rate, the predicted packet size and the corresponding packet processing time of SA . Strategy 4 : Load-balancing can be approached based on queuing information. This is somewhat greedy strategy. The total processing time of queued packets in a VPN server is considered as the load of the VPN server and is the total processing time for its queued packets in the corresponding VPN server. Because we do not know the processing time for a packet of new SA, the average processing time for queued packets of all SAs is used for the load of the new SA. That is, is the average of all s.

3 Simulation Results

For computer simulations, five kinds of security policies were considered; (Table). Each policy was a suit of ESP and AH algorithm negotiated by two parties. SEED is a Korean encryption algorithm standard. Table I shows the throughput (Mbps) of a VPN server for given ESP-AH suit and was obtained by SmartBits equipment. Our simulations were performed based on the numerical values of Table I. Five identical VPN servers were simulated in the manner of Fig. 1 because of the simplicity, although our load-balancing problem was designated for heterogeneous VPN servers. The primary performance measure is the objective function value of the load-balancing (i.e., the total summation of the expected response time). However, the average response time was a user-side performance measure. Thus the standard deviation of servers' busy time was introduced as a secondary performance measure on the side of VPN servers. Our simulation was conducted under two scenarios. First, we varied the SA arrival rate with the fixed average packet arrival rate of one SA. Second, we varied the average packet arrival rate of SAs with the fixed SA arrival rate.

Figure 2 and Figure 3 show the results of the two performance measures under the first scenario. The packet traffic load of one SA was 0.22 Erlang in this scenario. The average response time and standard deviation of Strategy 3 outperformed those of the others, as shown in Fig. 2 and Fig. 3. Figure 4 and Figure 5 show the results under the second scenario. The SA traffic load was fixed at 15 Erlang. Similar to the results of the first scenario, Strategy 3 performed best in both measures. The differences among strategies in the second scenario were less significant than those in the first scenario, revealing that the performance of load-balancing was more sensitive to SA traffic load than packet traffic load.

Table I. The throughput of the simulated VPN equipment according to ESP_AH suit and packet size (Mbps).

Bytes	No IPsec	3DES_NONE	3DES_SHA1	SEED_NONE	SEED_SHA1
64	30.11	14.78	13.84	13.45	8.21
128	48.85	25.02	23.40	20.73	12.90
256	81.90	41.07	38.55	30.33	19.61
512	99.91	58.43	56.27	39.47	27.59
1024	100.00	73.06	71.31	46.25	33.97
1400	100.00	84.62	83.02	48.36	36.08
1518	100.00	65.55	65.25	45.00	33.96

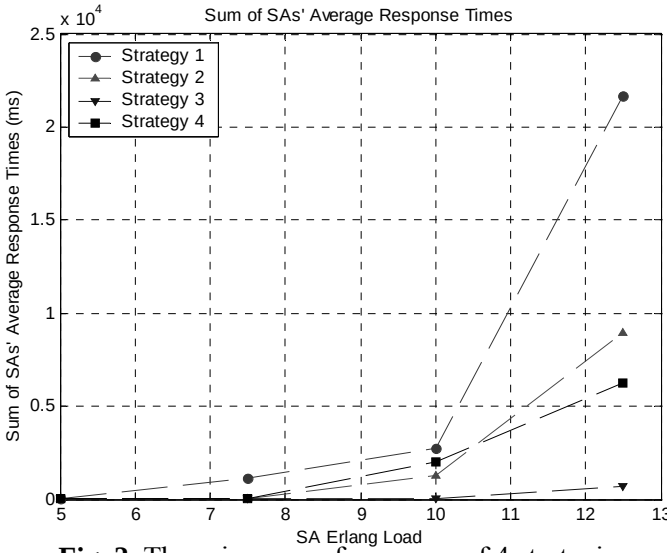


Fig. 2. The primary performances of 4 strategies under first scenario are shown. The packet traffic load of one SA is fixed to 0.22 Erlang.

Fig. 3. The secondary performances of 4 strategies under first scenario are shown. The packet traffic load of one SA is fixed to 0.22 Erlang.

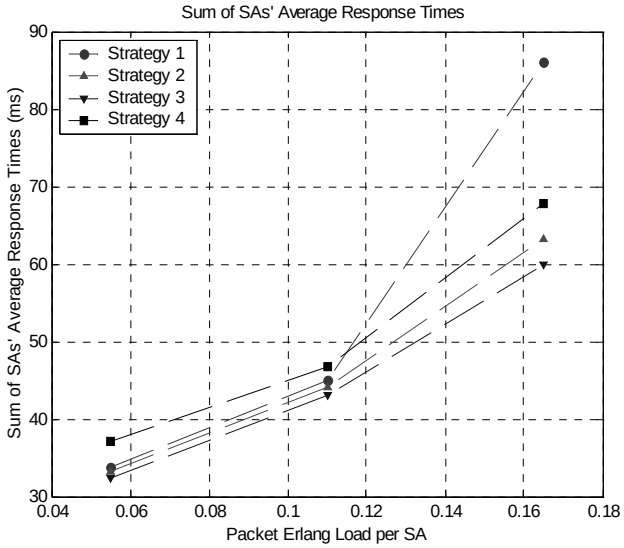


Fig. 4. The primary performances of 4 strategies under second scenario are shown. The SA traffic load is fixed to 15 Erlang.

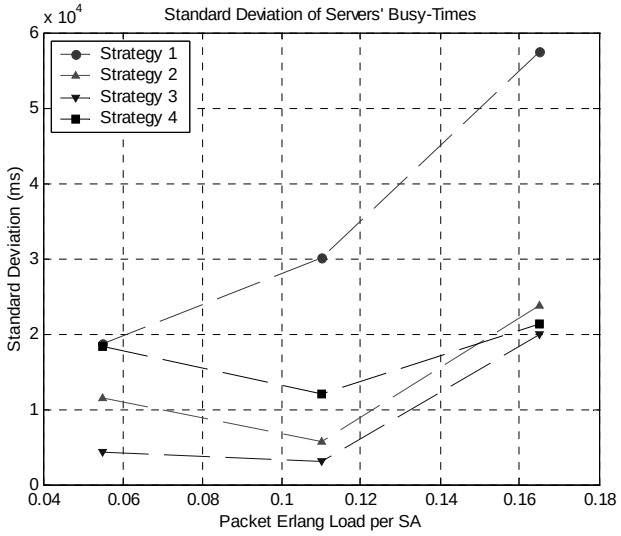
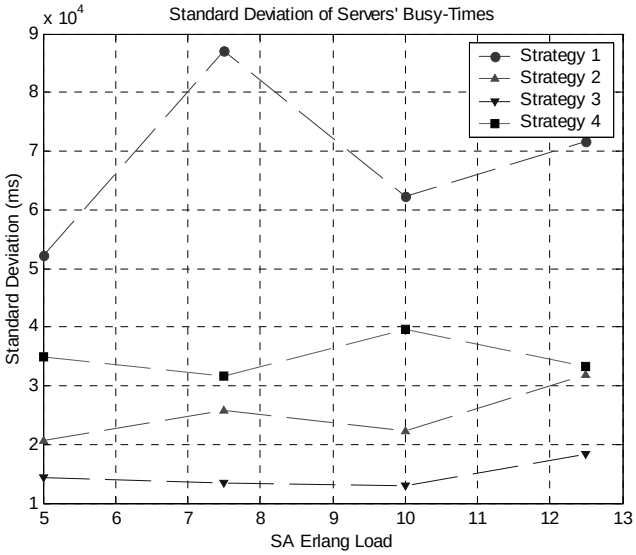


Fig. 5 The secondary performances of 4 strategies under second scenario are shown. The SA traffic load is fixed to 15 Erlang.



4 Conclusion

While load-balancing architectures and specific techniques for web servers have been developed intensively, there has been little work conducted for VPN servers. To achieve high speed, VPN servers should be adopted in a parallel manner using a load balancer, necessitating load-balancing research on the parallel configuration of VPN servers.

In this paper, we formulate the load-balancing problem of VPN and propose four strategies to measure the load of SAs approximately for practical implementation and solve the problem by one-by-one fashion based on the characteristics of VPN and information available in the VPN. In computer simulations, we compare the performances of the strategies in terms of server-side measure as well as user-side measure. For these reasons, we believe that our results are useful for actual operation of VPNs.

References:

[1] K. H. Cheung, J. Mišić, On Virtual Private Network Security Design Issues, *Computer Networks,* Vol.38, No.2, 2002, pp.165-179..

[2] V. Cardellini, M. Colajanni, P. S. Yu, "Dynamic Load Balancing on Web-Server Systems", *IEEE Internet Computing*, May-June, 1999, pp. 28-39.

[3] A. Bestavros, M. E. Crovella, J. Liu, D. Martin, Distributed Packet Rewriting and its Application

to Scalable Server Architectures, *1998 International Conference on Network Protocols*, 1998.

[4] L. J. Wolf, P. S. Yu, On Balancing the Load in a Clustered Web Farm, *ACM Transaction on Internet Technology*, Vol. 1, No.2, 2001, pp. 231-261.

[5] S. Kent, R. Atkinson, Security Architecture for Internet Protocol, IETF RFC 2401,1998.

[6] D. Maughan, M. Schertler, M. Schneider, J. Turner, Internet Security Association and Key Management Protocol (ISAKMP), IETF RFC 2408, 1998