Research Article

Mathematical Performance Evaluation Model for Mobile Network Firewall Based on Queuing

Shichang Xuan ¹, Dapeng Man ¹, Jiangchuan Zhang ¹, Wu Yang ¹ and Miao Yu ²

¹Information Security Research Center, Harbin Engineering University, No. 145 Nantong Street, Harbin, Heilongjiang 150001, China

²Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Correspondence should be addressed to Shichang Xuan; xuanshichang@hrbeu.edu.cn and Wu Yang; yangwu@hrbeu.edu.cn

Received 21 January 2018; Accepted 21 March 2018; Published 30 April 2018

Academic Editor: Wei Wang

Copyright © 2018 Shichang Xuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

While mobile networks provide many opportunities for people, they face security problems huge enough that a firewall is essential. The firewall in mobile networks offers a secure intranet through which all traffic is handled and processed. Furthermore, due to the limited resources in mobile networks, the firewall execution can impact the quality of communication between the intranet and the Internet. In this paper, a performance evaluation mathematical model for firewall system of mobile networks is developed using queuing theory for a multihierarchy firewall with multiple concurrent services. In addition, the throughput and the package loss rate are employed as performance evaluation indicators, and discrete-event simulated experiments are conducted for further verification. Lastly, experimental results are compared to theoretically obtained values to identify a resource allocation scheme that provides optimal firewall performance and can offer a better quality of service (QoS) in mobile networks.

1. Introduction

Due to the rapid development of mobile Internet technologies and the propagation of networks, the mobile Internet is progressively becoming significant part of everyone's personal and professional lives. Consequently, the security issues of mobile networks tend to attract more attention. Packet-based filtering in traditional firewalls fails to satisfy the security requirements of the end users. Being an essential component of the entire trusted intranet, a traditional firewall suffers from the disadvantages of simple packet filtering and lower-layer processing. Compared to traditional firewalls, new firewalls are capable of comprehensive data analysis at the application layer, which defines higher-level processing and more comprehensive protection.

Although new firewalls are able to provide a trusted intranet with better security, they consume large amounts of time for analyzing and processing the network packets because they require higher-level assessments and analysis. Furthermore, heavier traffic may reduce the processing efficiency. It makes the firewall prone to becoming a performance bottleneck for the communication between the

intranet and the Internet, consequently reducing the quality of service (QoS) of mobile networks. To overcome this problem, limited system resources (such as number of threads and CPU) must be rationally assigned when developing a firewall. Modifying the assignment of resources may directly affect the overall system performance. In general, when developing systems such as firewall, the corresponding firewall performance test needs to be performed. The throughput of the current system needs to be tested and performance optimization be conducted by software engineers. Further tests should be carried out until the optimal resource allocation scheme is obtained. This method for testing and performance tuning is very time-consuming and complex. However, if a mathematical model is constructed to test the common performance indicators of the firewall, such as throughput and packet loss rate, considerable savings in developer labor costs may be achieved, and the optimal resource allocation system may be derived directly.

In this paper, a performance evaluation model for the firewall system of mobile networks is proposed using the queuing theory. Generally, this model can be divided into two phases. Phase one involves handling the traffic at the lower

layers, including examinations at the network layer and the transmission layer. Phase two involves handling the upper-layer applications, consisting of HTTP, IMAP, and DNS. Each phase involves concurrent processing with multiservice stations. A discrete-event simulated experiment is conducted to validate the model.

The rest of the paper is organized as follows: Section 2 reviews the current state of related research and designates the innovations proposed in this paper. Section 3 discusses the mathematical modeling of a firewall system for mobile networks. Section 4 describes the simulated experiments performed using the developed model and analyzes the results of conducted experiments. Finally, Section 5 summarizes the results and conclusions.

2. Related Work

With the increasing complexity of mobile network security issues, many researchers have been studying and analyzing firewall equipment. As the methods and means of network attacks continue to increase [1], the functions of firewall systems are being enhanced and extended gradually, for example, intrusion detection, DDoS attack detection, and user-behavior detection [2, 3]. Moreover, a number of researchers have conducted network security analysis. Peng et al. mainly performed analysis with respect to user behaviors [4–6]. Chen et al. provided a survey of security problems in mobile networks [7]. A resource optimization approach in mobile networks was proposed in [8] under imperfect prediction.

The queuing theory, which has significant potential in terms of its application to the mathematical modeling of network equipment, may be useful for analyzing the performance of such equipment [9]. Some researchers have conducted modeling and analyzing of systems such as firewalls based on this theory, and they have made certain relatively advanced achievements in the study of network equipment performance. Herrmann constructed a vacation model for network service analysis, where batch arrival is used rather than the common Poisson flow [10]. Furthermore, in some studies, large-scale cloud services [11] have been modeled into a complex processing system with finite queues [12–14]. Salah et al. made considerable achievements by proposing a service system with a finite queue and single window, where packets arrive in a Poisson flow pattern and the service time follows a negative exponential distribution [15, 16]. Salah [17] found that the differences between the applicability of the Erlang distribution modeling and that of a Markov chain modeling have been compared. Zapechnikov et al. conducted an in-depth study based on Salah's work. They proposed a two-phase service system, where Phase 1 was based on a Markov chain model and Phase 2 employed a hyper-Erlang distribution model [18]. Some researchers have also investigated cloud-service-based firewall systems [19, 20].

Several recent studies have only compared modeling methods, while some have merely discussed the handling process of a single-service window or conducted simple hierarchical modeling [21]. However, neither can describe the firewall workflow nor depict its handling process in detail. With the advances in science and technology, the widespread

use of multicore processors and the increased availability of resources in terms of equipment, it is becoming difficult for the above-mentioned methods to fully exploit hardware performance. In this paper, an in-depth examination is conducted with respect to these issues. Further, based on firewall handling and processing characteristics, a two-phase multiservice station [22] and multiprotocol firewall model with multiple concurrent applications is proposed, which is further employed to analyze the overall system performance [23].

3. Model Analysis

Using an Erlang queuing model, this paper proposes a two-phase multiservice station and multiprotocol firewall model with multiple concurrent applications. This model may be hierarchically divided into two phases. When a packet enters the system, it will first be processed on the basis of the rules at the network layer and the transmission layer and then handled by the DNS, FTP, and HTTP protocols. The process flow varies slightly with the characteristics of various protocols. The hierarchy of the firewall model is shown in Figure 1.

The related parameters with respect to the model are defined as follows.

λ denotes the arrival rate of packets that arrive at the firewall. In Phase 1, K_L represents the phase 1 buffer, N_L denotes the number of service windows for Phase 1, r_L is the number of rules for Phase 1, and μ_L is the processing rate of each rule. Phase 2 is divided into three main parts, namely, DNS, FTP, and HTTP. Here, q_D , q_F , and q_H denote the proportions of DNS, FTP, and HTTP traffic, respectively. In the DNS handling part, K_D denotes the buffer, N_D represents the number of service windows, r_D represents the number of rules, and μ_D is the handling rate of each rule. The FTP handling component consists of FTP command handling operations and FTP data transmission handling operations. In terms of the FTP command handling part, K_{F1} denotes its buffer, N_{F1} represents its number of service windows, r_{F1} is the number of rules, and μ_{F1} represents the handling rate of each rule. The FTP data transmission handling part may be divided into multiple transmissions. For example, n concurrent FTP data transmission handling operations are shown in Figure 2, where the traffic proportion is q_{F-i} . Further, K_{F2-i} represents the buffer for data transmission handling, N_{F2-i} denotes the number of service windows for data transmission handling, r_{F2-i} is the number of rules for data transmission handling, and μ_{F2-i} denotes the handling rate of each rule. The HTTP part consists of HTTP-request line handling, HTTP-header handling, and HTTP-body handling. Here, q_{H-i} denotes the traffic proportion of each HTTP application; K_{H1} , K_{H2} , and K_{H3-i} denote the buffers for HTTP line handling, header handling, and body handling for each application, respectively; N_{H1} , N_{H2} , and N_{H3-i} represent the number of service windows for HTTP-request line handling, header handling, and body handling for each application, respectively; r_{H1} , r_{H2} , and r_{H3-i} denote the number of rules for HTTP line handling, header handling, and body handling for each application, respectively; and μ_{H1} , μ_{H2} , and μ_{H3-i}

FIGURE 1: Hierarchy of the firewall model.

FIGURE 2: Submodule structure.

denote the handling rate of each rule for HTTP line handling, header handling, and body handling for each application, respectively.

The firewall model proposed in this paper refines a number of essential upper-layer handling processes in which the traffic flow proportion or number of applications may be easily modified. In addition, this model comprises the commonly used handling processes for DNS, FTP, and HTTP, each of which may be extended as an individual template for other upper-layer application protocols.

3.1. Low-Level Analysis. As shown in Figure 1, the entire model consists of multiple single-buffer and multiconcurrent Erlang processes. Hence, the computation of the entire model may be based on cumulative computations of individual submodules. Packets enter the system sequentially, following a Poisson flow pattern; then, they wait in the buffer to be further handled by the concurrent service windows. Each service window comprises multiple subservice stations that correspond to each rule. The handling efficiency of the service windows follows an Erlang distribution. The structure of the submodules is shown in Figure 2.

The following key formula may be derived on the basis of the results of the previous work.

The distribution density function of the service time may be denoted by

$$f(t) = \frac{N\mu (N\mu t)^{r-1}}{(r-1)!} e^{-N\mu t} \quad (t \geq 0). \quad (1)$$

Further, ∂_k denotes the probability that k packets arrive at the system during the service time of a packet:

$$\partial_k = \int_0^\infty \frac{(\lambda x)^k}{k!} e^{-\lambda x} f(x) dx. \quad (2)$$

Hence, the following formula can be obtained:

$$\begin{aligned} \partial_k &= \frac{\lambda^k (N\mu)^r}{k! (r-1)!} \cdot \frac{\Gamma(k+r)}{(\lambda + N\mu)^{k+r}} \\ &= \frac{\lambda^k (N\mu)^r}{k! (r-1)!} \cdot \frac{(k+r-1)!}{(\lambda + N\mu)^{k+r}}. \end{aligned} \quad (3)$$

In addition, P_{jk} represents the state transition probability, that is, the probability that the number of network packets

in the system changes from j to k at any arbitrary moment. The transition of states is correlated with the number of packet arrivals within the service time; hence, the relationship between P_{jk} and α_k may be obtained as follows:

$$\begin{aligned} p_{0k} &= \begin{cases} \partial_k & 0 \leq k \leq K+N-2 \\ \sum_{m=K+N-1}^{\infty} \partial_m & k = K+N-1 \end{cases} \quad j=0 \\ p_{jk} &= \begin{cases} \partial_{k-j+1} & j-1 \leq k \leq K+N-2 \\ \sum_{m=K+N-1}^{\infty} \partial_m & k = K+N-1 \end{cases} \quad (4) \\ &1 \leq j \leq K+N-1. \end{aligned}$$

In terms of the steady-state probability of the system following the instantaneous exit of the packets π_k ($0 \leq k \leq K+N-1$), the following relationship exists amongst various states:

$$\pi_k = \sum_{j=0}^{K-1} \pi_j p_{jk} \quad 0 \leq k \leq K+N-1. \quad (5)$$

The following may be derived:

$$\pi_k = \pi_0 \partial_k + \sum_{j=1}^{k+1} \pi_j \partial_{k-j+1} \quad 0 \leq k \leq K+N-2. \quad (6)$$

According to the regularity, we can derived

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K-1} (\pi_k / \pi_0)}. \quad (7)$$

The load offered at the network layer is given by

$$\rho = \lambda \cdot \frac{r}{N\mu}. \quad (8)$$

The packet loss rate is denoted by

$$p_{\text{loss}} = p_K = 1 - \frac{1 - p_0}{\rho} = \frac{p_0 + \rho - 1}{\rho}. \quad (9)$$

And the throughput is expressed by

$$\gamma = \lambda (1 - p_{\text{loss}}). \quad (10)$$

3.2. Phase 1 Analysis. As shown in Figure 1, it can be learned that Phase 1 hierarchy is consistent with the subhierarchy described above, where the arrival rate of the packets is λ , the buffer size is denoted by K_L , the number of service windows is represented by N_L , the number of rules is r_L , and the service rate of the rules is indicated by μ_L .

By applying the conclusion drawn in the previous section, it can be learned that the probability that k packets arrive at a queue within the service time of a packet may be expressed by

$$\partial_k = \frac{\lambda^k (N_L \mu_L)^{r_L}}{k! (r_L - 1)!} \cdot \frac{(k + r_L - 1)!}{(\lambda + N_L \mu_L)^{k+r_L}}. \quad (11)$$

The state transition probability for Phase 1 is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K_L + N_L - 2 \\ \sum_{m=K_L+N_L-1}^{\infty} \partial_m & k = K_L + N_L - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \leq k \leq K_L + N_L - 2 \\ \sum_{m=K_L+N_L-1}^{\infty} \partial_m & k = K_L + N_L - 1 \end{cases} \quad (12)$$

$$1 \leq j \leq K_L + N_L - 1.$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_L+N_L-1} \pi_j p_{jk} \quad 0 \leq k \leq K_L + N_L - 1. \quad (13)$$

By performing ratio computation with π_0 , we obtain

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_L+N_L-1} (\pi_k/\pi_0)}. \quad (14)$$

Phase 1 packet loss rate is given by

$$p_{\text{loss-L}} = 1 - \frac{1}{\pi_0 + \lambda \cdot r_L / N_L \mu_L}. \quad (15)$$

Phase 1 throughput is given by

$$\gamma_L = \lambda (1 - p_{\text{loss-L}}). \quad (16)$$

3.3. Phase 2 Analysis. Phase 2 analysis is comparatively complex. It involves handling operations implemented by DNS, FTP, and HTTP. Phase 2 throughputs, that is, the overall throughput of the system, are jointly comprised of the throughputs from DNS, FTP, and HTTP.

3.3.1. DNS Handling Analysis. As shown in Figure 1, the DNS handling process is based on a single-phase concurrent Erlang model. If applications with similar handling processes exist in the system, the DNS handling process may be used as a template. During the DNS handling operations, the arrival rate of packets equals Phase 1 throughput $\gamma_L * q_D$, the buffer size is denoted by K_D , the number of service windows is N_D , the number of rules is represented by r_D , and the service rate of the rules is indicated by μ_D .

As with Phase 1 handling process, the probability that k packets arrive at the queue during the service time of a packet may be expressed by

$$\partial_k = \frac{\lambda^k (N_D \mu_D)^{r_D}}{k! (r_D - 1)!} \cdot \frac{(k + r_D - 1)!}{(\lambda + N_D \mu_D)^{k+r_D}}. \quad (17)$$

The state transition probability for DNS handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K_D + N_D - 2 \\ \sum_{m=K_D+N_D-1}^{\infty} \partial_m & k = K_D + N_D - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \leq k \leq K_D + N_D - 2 \\ \sum_{m=K_D+N_D-1}^{\infty} \partial_m & k = K_D + N_D - 1 \end{cases} \quad (18)$$

$$1 \leq j \leq K_D + N_D - 1.$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_D+N_D-1} \pi_j p_{jk} \quad 0 \leq k \leq K_D + N_D - 1. \quad (19)$$

By performing ratio computation with π_0 , the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_D+N_D-1} (\pi_k/\pi_0)}. \quad (20)$$

The packet loss rate for DNS handling operations is given by

$$p_{\text{loss-D}} = 1 - \frac{1}{\pi_0 + \lambda \cdot q_D \cdot r_D / N_D \mu_D}. \quad (21)$$

The throughput for DNS handling operations is expressed by

$$\gamma_D = \lambda \cdot q_D (1 - p_{\text{loss-D}}). \quad (22)$$

3.3.2. FTP Handling Analysis. As shown in Figure 1, the FTP handling process is based on a two-phase concurrent Erlang model. If applications with similar handling processes exist in the system, the FTP handling process may be used as a template. The FTP handling operations may be divided into two parts. When a packet arrives at the FTP handling process, it will first be processed by the FTP command handling module, followed by the filtering and analysis of the transmitted data.

(1) FTP Command Handling Operations. During the FTP command handling operations, the arrival rate of the packets equals Phase 1 throughput $\gamma_L * q_F$, the buffer size is denoted by K_{F1} , the number of service windows is N_{F1} , the number of rules is represented by r_{F1} , and the service rate of the rules is μ_{F1} .

The probability that k packets arrive at a queue within the service time of a packet may be expressed by

$$\partial_k = \frac{\lambda^k (N_{F1} \mu_{F1})^{r_{F1}}}{k! (r_{F1} - 1)!} \cdot \frac{(k + r_{F1} - 1)!}{(\lambda + N_{F1} \mu_{F1})^{k+r_{F1}}}. \quad (23)$$

The state transition probability for FTP command handling operations may be expressed by

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K_{F1} + N_{F1} - 2 \\ \sum_{m=K_{F1}+N_{F1}-1}^{\infty} \partial_m & k = K_{F1} + N_{F1} - 1 \end{cases} \quad j = 0 \quad (24)$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \leq k \leq K_{F1} + N_{F1} - 2 \\ \sum_{m=K_{F1}+N_{F1}-1}^{\infty} \partial_m & k = K_{F1} + N_{F1} - 1 \end{cases} \quad 1 \leq j \leq K_{F1} + N_{F1} - 1.$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{F1}+N_{F1}-1} \pi_j p_{jk} \quad 0 \leq k \leq K_{F1} + N_{F1} - 1. \quad (25)$$

By performing ratio computation with π_0 , the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{F1}+N_{F1}-1} (\pi_k / \pi_0)}. \quad (26)$$

The packet loss rate for FTP command handling operations is given by

$$p_{\text{loss-F1}} = 1 - \frac{1}{\pi_0 + \lambda \cdot q_F \cdot r_{F1} / N_{F1} \mu_{F1}}. \quad (27)$$

The throughput for FTP command handling operations is denoted by

$$\gamma_{F1} = \lambda \cdot q_F (1 - p_{\text{loss-F1}}). \quad (28)$$

(2) *FTP Data Transmission Handling Operations.* In terms of FTP data transmission handling operations, assume that there are n' concurrent data transmission handling modules, and let i denote the i th data transmission handling module. The packet flow proportion for each data transmission handling module is denoted by q_{F-i} . The arrival rate of packets equals the throughput for FTP command handling operations, $\gamma_{F1} \cdot q_{F-i}$, the buffer size is K_{F2-i} , the number of service windows is expressed by N_{F2-i} , the number of rules is r_{F2-i} , and the service rate of the rules is represented by μ_{F2-i} .

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k (N_{F2-i} \mu_{F2-i})^{r_{F2-i}}}{k! (r_{F2-i} - 1)!} \cdot \frac{(k + r_{F2-i} - 1)!}{(\lambda + N_{F2-i} \mu_{F2-i})^{k+r_{F2-i}}}. \quad (29)$$

The state transition probability for FTP data transmission handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K_{F2-i} + N_{F2-i} - 2 \\ \sum_{m=K_{F2-i}+N_{F2-i}-1}^{\infty} \partial_m & k = K_{F2-i} + N_{F2-i} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \leq k \leq K_{F2-i} + N_{F2-i} - 2 \\ \sum_{m=K_{F2-i}+N_{F2-i}-1}^{\infty} \partial_m & k = K_{F2-i} + N_{F2-i} - 1 \end{cases} \quad 1 \leq j \leq K_{F2-i} + N_{F2-i} - 1. \quad (30)$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{F2-i}+N_{F2-i}-1} \pi_j p_{jk} \quad 0 \leq k \leq K_{F2-i} + N_{F2-i} - 1. \quad (31)$$

By performing ratio computation with π_0 , the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{F2-i}+N_{F2-i}-1} (\pi_k / \pi_0)}. \quad (32)$$

The packet loss rate for FTP data transmission handling operations is given by

$$p_{\text{loss-F2-i}} = 1 - \frac{1}{\pi_0 + \gamma_{F1} \cdot q_{F-i} \cdot r_{F2-i} / N_{F2-i} \mu_{F2-i}}. \quad (33)$$

The throughput for FTP data transmission handling operations is denoted by

$$\gamma_{F2-i} = \gamma_{F1} \cdot q_{F-i} (1 - p_{\text{loss-F2-i}}). \quad (34)$$

The total throughput for FTP handling operations is expressed by

$$\gamma_F = \sum_{i=1}^n \gamma_{F2-i}. \quad (35)$$

3.3.3. *HTTP Handling Analysis.* As shown in Figure 1, the HTTP handling process is based on a three-phase concurrent Erlang model. If applications with similar handling processes exist in the system, the HTTP handling process may be used as a template. The HTTP handling operations may be divided into three parts. When a packet arrives at the HTTP handling process, it will first be handled by HTTP-request line handling module; then, it will be processed by HTTP-header handling module, and finally, it will be handled by the HTTP-body handling module and delivered to upper-layer applications.

(1) *HTTP-Request Line Handling Operations.* In terms of HTTP-request line handling operations, the arrival rate of

packets equals the throughput for Phase 1, $\gamma_L * q_H$, the buffer size is K_{H1} , the number of service windows is denoted by N_{H1} , the number of rules is r_{H1} , and the service rate of the rules is expressed by μ_{H1} .

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k (N_{H1}\mu_{H1})^{r_{H1}}}{k! (r_{H1} - 1)!} \cdot \frac{(k + r_{H1} - 1)!}{(\lambda + N_{H1}\mu_{H1})^{k+r_{H1}}}. \quad (36)$$

The state transition probability for HTTP-request line handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K_{H1} + N_{H1} - 2 \\ \sum_{m=K_{H1}+N_{H1}-1}^{\infty} \partial_m & k = K_{H1} + N_{H1} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j - 1 \leq k \leq K_{H1} + N_{H1} - 2 \\ \sum_{m=K_{H1}+N_{H1}-1}^{\infty} \partial_m & k = K_{H1} + N_{H1} - 1 \end{cases} \quad 1 \leq j \leq K_{H1} + N_{H1} - 1. \quad (37)$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{H1}+N_{H1}-1} \pi_j p_{jk} \quad 0 \leq k \leq K_{H1} + N_{H1} - 1. \quad (38)$$

By performing ratio computation with π_0 , the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{H1}+N_{H1}-1} (\pi_k/\pi_0)}. \quad (39)$$

The packet loss rate for HTTP-request line handling operations is given by

$$p_{\text{loss-H1}} = 1 - \frac{1}{\pi_0 + \lambda \cdot q_H \cdot r_{H1}/N_{H1}\mu_{H1}}. \quad (40)$$

The throughput for HTTP-request line handling operations is denoted by

$$\gamma_{H1} = \lambda \cdot q_H (1 - p_{\text{loss-H1}}). \quad (41)$$

(2) *HTTP-Header Handling Operations.* With respect to HTTP-header handling operations, the arrival rate of packets equals the throughput for the HTTP-request line handling operations, γ_{H2} , the buffer size is K_{H2} , the number of service windows is represented by N_{H2} , the number of rules is r_{H2} , and the service rate of the rules is μ_{H2} .

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k (N_{H2}\mu_{H2})^{r_{H2}}}{k! (r_{H2} - 1)!} \cdot \frac{(k + r_{H2} - 1)!}{(\lambda + N_{H2}\mu_{H2})^{k+r_{H2}}}. \quad (42)$$

The state transition probability for HTTP-header handling operations may be expressed by

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K_{H2} + N_{H2} - 2 \\ \sum_{m=K_{H2}+N_{H2}-1}^{\infty} \partial_m & k = K_{H2} + N_{H2} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j - 1 \leq k \leq K_{H2} + N_{H2} - 2 \\ \sum_{m=K_{H2}+N_{H2}-1}^{\infty} \partial_m & k = K_{H2} + N_{H2} - 1 \end{cases} \quad 1 \leq j \leq K_{H2} + N_{H2} - 1. \quad (43)$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{H2}+N_{H2}-1} \pi_j p_{jk} \quad 0 \leq k \leq K_{H2} + N_{H2} - 1. \quad (44)$$

By performing ratio computation with π_0 , the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{H2}+N_{H2}-1} (\pi_k/\pi_0)}. \quad (45)$$

The packet loss rate for HTTP-header handling operations may be expressed by

$$p_{\text{loss-H2}} = 1 - \frac{1}{\pi_0 + \gamma_{H1} \cdot r_{H2}/N_{H2}\mu_{H2}}. \quad (46)$$

The throughput for the HTTP-header handling operations may be expressed by

$$\gamma_{H2} = \gamma_{H1} (1 - p_{\text{loss-H2}}). \quad (47)$$

(3) *HTTP-Body Handling Operations.* In terms of HTTP-body handling operations, assume that there are n' concurrent HTTP-body handling modules and let i denote the i th HTTP-body handling module. The packet flow proportion for each HTTP-body handling module is denoted by q_{H3-i} . The arrival rate of packets equals the throughput for the HTTP-header handling operations, $\gamma_{H2} * q_{H3-i}$, the buffer size is K_{H3-i} , the number of service windows is expressed by N_{H3-i} , the number of rules is r_{H3-i} , and the service rate of the rules is represented by μ_{H3-i} .

The probability that k packets arrive at a queue within the service time of a packet may be denoted by

$$\partial_k = \frac{\lambda^k (N_{H3-i}\mu_{H3-i})^{r_{H3-i}}}{k! (r_{H3-i} - 1)!} \cdot \frac{(k + r_{H3-i} - 1)!}{(\lambda + N_{H3-i}\mu_{H3-i})^{k+r_{H3-i}}}. \quad (48)$$

The state transition probability for HTTP-body handling operations is given by

$$p_{0k} = \begin{cases} \partial_k & 0 \leq k \leq K_{H3-i} + N_{H3-i} - 2 \\ \sum_{m=K_{H3-i}+N_{H3-i}-1}^{\infty} \partial_m & k = K_{H3-i} + N_{H3-i} - 1 \end{cases} \quad j = 0$$

$$p_{jk} = \begin{cases} \partial_{k-j+1} & j-1 \leq k \leq K_{H3-i} + N_{H3-i} - 2 \\ \sum_{m=K_{H3-i}+N_{H3-i}-1}^{\infty} \partial_m & k = K_{H3-i} + N_{H3-i} - 1 \end{cases} \quad 1 \leq j \leq K_{H3-i} + N_{H3-i} - 1. \quad (49)$$

The following formula may be obtained:

$$\pi_k = \sum_{j=0}^{K_{H3-i}+N_{H3-i}-1} \pi_j p_{jk} \quad 0 \leq k \leq K_{H3-i} + N_{H3-i} - 1. \quad (50)$$

By performing ratio computation with π_0 , the following formula is obtained:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K_{H3-i}+N_{H3-i}-1} (\pi_k / \pi_0)}. \quad (51)$$

The packet loss rate for HTTP-body handling operations may be denoted by

$$p_{\text{loss-H3-i}} = 1 - \frac{1}{\pi_0 + \gamma_{H2} \cdot q_{H-i} \cdot r_{H3-i} / N_{H3-i} \mu_{H3-i}}. \quad (52)$$

The throughput for HTTP-body handling operations may be expressed by

$$\gamma_{H3-i} = \gamma_{H2} \cdot q_{H-i} (1 - p_{\text{loss-H3-i}}). \quad (53)$$

The total throughput for HTTP-body handling operations may be expressed by

$$\gamma_H = \sum_{i=1}^{n'} \gamma_{H3-i}. \quad (54)$$

The overall traffic handled by the firewall should be the sum of the traffic handled by DNS, FTP, and HTTP. The overall packet loss rate of the firewall is given by

$$p_{\text{loss}} = p_{\text{loss-L}} + q_D \cdot (1 - p_{\text{loss-L}}) p_{\text{loss-D}} + q_F \cdot (1 - p_{\text{loss-L}}) p_{\text{loss-F1}} + \sum_{i=1}^n q_{Fi} \cdot [1 - q_F \cdot (1 - p_{\text{loss-L}}) p_{\text{loss-F1}}] \cdot p_{\text{loss-F2-i}} + q_H \cdot (1 - p_{\text{loss-L}}) p_{\text{loss-H1}} + [1 - q_H \cdot (1 - p_{\text{loss-L}}) p_{\text{loss-H1}}] \cdot p_{\text{loss-H2}} + \sum_{i=1}^{n'} q_{Hi}$$

$$\cdot \{1 - [1 - q_H \cdot (1 - p_{\text{loss-L}}) p_{\text{loss-H1}}] \cdot p_{\text{loss-H2}}\} \cdot p_{\text{loss-H3-i}}. \quad (55)$$

4. Experiment Evaluation

This section describes a discrete-event simulated experiment conducted with respect to the firewall model described above. The arrival time of packets and processing time of service windows are stochastically generated, where the arrival process is a Poisson process and the service process follows an Erlang distribution.

Let the number of total resources be 12, enumerate the combinations of the allocated resources, compute the values of the throughput and packet loss rate under each combination using the theoretical formulae and simulation programs, respectively, and calculate the error between the theoretical and simulated values.

The total number of service stations is 12, and the arrival rate of data packets $\lambda = 80$ kpps (kilo packets per second). In Phase 1, the buffer size $K_L = 100$, the number of rules $r_L = 5$, and the handling rate $\mu_L = 250$ kpps. In Phase 2, the DNS traffic accounts for 20%; FTP traffic, 20%; and HTTP traffic, 60%. In terms of the DNS handling part, the buffer size $K_D = 100$, the number of rules $r_D = 5$, and the handling rate $\mu_D = 200$ kpps. In the FTP handling part, the FTP command handling buffer size $K_{F1} = 100$, the number of rules $r_{F1} = 5$, and the handling rate $\mu_{F1} = 350$ kpps. The FTP data transmission handling part may be divided into two data transmission processes, whose traffic proportions are 40% and 60%, respectively. For the first data transmission process, the buffer size $K_{F2-1} = 80$, the number of rules $r_{F2-1} = 5$, and the handling rate $\mu_{F2-1} = 150$ kpps. For the second transmission process, the buffer size $K_{F2-2} = 90$, the number of rules $r_{F2-2} = 5$, and the handling rate $\mu_{F2-2} = 180$ kpps. In the HTTP handling part, the buffer size for HTTP-request line handling $K_{H1} = 100$, the number of rules $r_{H1} = 5$, and the handling rate $\mu_{H1} = 500$ kpps. The buffer size for HTTP-header handling $K_{H2} = 150$, the number of rules $r_{H2} = 5$, and the handling rate $\mu_{H2} = 200$ kpps. The HTTP-body handling operations consist of two HTTP applications, whose traffic proportions are 40% and 60%, respectively. The buffer sizes are $K_{H3-1} = 100$ and $K_{H3-2} = 100$, respectively, the number of rules is $r_{H3-1} = 5$ and $r_{H3-2} = 5$, respectively, and the handling rates are $\mu_{H3-1} = 80$ kpps and $\mu_{H3-2} = 100$ kpps, respectively. The results are presented in Table 1.

As determined from the experimental results, when the resources are allocated as $N_L = 2$, $N_D = 1$, $N_{F1} = 1$, $N_{F2-1} = 1$, $N_{F2-2} = 1$, $N_{H1} = 1$, $N_{H2} = 2$, $N_{H3-1} = 1$, and $N_{H3-2} = 2$, the maximum throughput reaches up to 76.69 kpps with the lowest packet loss rate. Further, the error between the theoretical computation results and the simulated experiment results remains within 3.3%; the mean error is 0.848%.

5. Conclusion

A two-phase multiservice station and multiprotocol firewall model with multiple concurrent applications was proposed

TABLE 1: Relationship of resource allocation versus throughput and packet loss rate.

N_L	N_D	N_{F1}	N_{F2-1}	N_{F2-2}	N_{H1}	N_{H2}	N_{H3-1}	N_{H3-2}	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
1	1	1	1	1	1	1	1	4	50.252	0.372	49.525	0.381	0.015
1	1	1	1	1	1	1	2	3	50.777	0.365	49.545	0.381	0.025
1	1	1	1	1	1	1	3	2	50.322	0.371	49.504	0.381	0.017
1	1	1	1	1	1	1	4	1	50.166	0.373	49.297	0.384	0.018
1	1	1	1	1	1	2	1	3	50.273	0.372	49.545	0.381	0.015
1	1	1	1	1	1	2	2	2	50.181	0.373	49.647	0.379	0.011
1	1	1	1	1	1	2	3	1	51.012	0.362	49.421	0.382	0.032
1	1	1	1	1	1	3	1	2	50.456	0.369	49.507	0.381	0.019
1	1	1	1	1	1	3	2	1	50.257	0.372	49.380	0.383	0.018
1	1	1	1	1	1	4	1	1	49.927	0.376	49.496	0.381	0.009
1	1	1	1	1	2	1	1	3	50.483	0.369	49.405	0.382	0.022
1	1	1	1	1	2	1	2	2	50.720	0.366	49.518	0.381	0.024
1	1	1	1	1	2	1	3	1	50.525	0.368	49.289	0.384	0.025
1	1	1	1	1	2	2	1	2	50.519	0.369	49.496	0.381	0.021
1	1	1	1	1	2	2	2	1	50.690	0.366	49.453	0.382	0.025
1	1	1	1	1	2	3	1	1	50.775	0.365	49.425	0.382	0.027
1	1	1	1	1	3	1	1	2	50.291	0.371	49.455	0.382	0.017
1	1	1	1	1	3	1	2	1	50.226	0.372	49.471	0.382	0.015
1	1	1	1	1	3	2	1	1	50.228	0.372	49.112	0.386	0.023
1	1	1	1	1	4	1	1	1	50.665	0.367	49.351	0.383	0.027
1	1	1	1	2	1	1	1	3	50.308	0.371	49.676	0.379	0.013
1	1	1	1	2	1	1	2	2	50.315	0.371	49.469	0.382	0.017
1	1	1	1	2	1	1	3	1	50.211	0.372	49.351	0.383	0.017
1	1	1	1	2	1	2	1	2	50.310	0.371	49.505	0.381	0.016
1	1	1	1	2	1	2	2	1	50.627	0.367	49.433	0.382	0.024
1	1	1	1	2	1	3	1	1	50.212	0.372	49.396	0.383	0.017
1	1	1	1	2	2	1	1	2	50.889	0.364	49.567	0.380	0.027
1	1	1	1	2	2	1	2	1	50.514	0.369	49.484	0.381	0.021
1	1	1	1	2	2	2	1	1	50.446	0.369	49.150	0.386	0.026
1	1	1	1	2	3	1	1	1	50.177	0.373	49.393	0.383	0.016
1	1	1	1	3	1	1	1	2	50.569	0.368	49.578	0.380	0.020
1	1	1	1	3	1	1	2	1	50.724	0.366	49.376	0.383	0.027
1	1	1	1	3	1	2	1	1	50.240	0.372	49.408	0.382	0.017
1	1	1	1	3	2	1	1	1	50.818	0.365	49.380	0.383	0.029
1	1	1	1	4	1	1	1	1	50.548	0.368	49.206	0.385	0.027
1	1	1	1	2	1	1	1	3	50.689	0.366	49.436	0.382	0.025
1	1	1	1	2	1	1	1	2	50.800	0.365	49.524	0.381	0.026
1	1	1	1	2	1	1	1	3	50.616	0.367	49.140	0.386	0.030
1	1	1	1	2	1	1	2	1	50.589	0.368	49.485	0.381	0.022
1	1	1	1	2	1	1	2	2	50.552	0.368	49.057	0.387	0.030
1	1	1	1	2	1	1	3	1	50.585	0.368	49.365	0.383	0.025
1	1	1	1	2	1	2	1	1	50.558	0.368	49.503	0.381	0.021
1	1	1	1	2	1	2	1	2	50.264	0.372	49.157	0.386	0.023
1	1	1	1	2	1	2	2	1	50.307	0.371	49.194	0.385	0.023
1	1	1	1	2	1	3	1	1	49.989	0.375	49.261	0.384	0.015
1	1	1	1	2	2	1	1	1	50.404	0.370	49.533	0.381	0.018
1	1	1	1	2	2	1	1	2	50.710	0.366	49.593	0.380	0.023

TABLE I: Continued.

N_L	N_D	N_{F1}	N_{F2-1}	N_{F2-2}	N_{H1}	N_{H2}	N_{H3-1}	N_{H3-2}	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
1	1	1	2	2	1	2	1	1	50.271	0.372	49.426	0.382	0.017
1	1	1	2	2	2	1	1	1	50.344	0.371	49.333	0.383	0.020
1	1	1	2	3	1	1	1	1	50.468	0.369	49.524	0.381	0.019
1	1	1	3	1	1	1	1	2	50.982	0.363	49.477	0.382	0.030
1	1	1	3	1	1	1	2	1	50.504	0.369	49.180	0.385	0.027
1	1	1	3	1	1	2	1	1	50.608	0.367	49.106	0.386	0.031
1	1	1	3	1	2	1	1	1	50.326	0.371	49.294	0.384	0.021
1	1	1	3	2	1	1	1	1	50.096	0.374	49.408	0.382	0.014
1	1	1	4	1	1	1	1	1	50.104	0.374	49.302	0.384	0.016
1	1	2	1	1	1	1	1	3	50.105	0.374	49.656	0.379	0.009
1	1	2	1	1	1	1	2	2	50.371	0.370	49.459	0.382	0.018
1	1	2	1	1	1	1	3	1	50.209	0.372	49.377	0.383	0.017
1	1	2	1	1	1	2	1	2	50.027	0.375	49.517	0.381	0.010
1	1	2	1	1	1	2	2	1	50.769	0.365	49.471	0.382	0.026
1	1	2	1	1	1	3	1	1	50.644	0.367	49.533	0.381	0.022
1	1	2	1	1	2	1	1	2	50.417	0.370	49.507	0.381	0.018
1	1	2	1	1	2	1	2	1	49.918	0.376	49.445	0.382	0.010
1	1	2	1	1	2	2	1	1	50.770	0.365	49.469	0.382	0.026
1	1	2	1	1	3	1	1	1	50.103	0.361	49.309	0.384	0.016
1	1	2	1	2	1	1	1	2	50.238	0.372	49.503	0.381	0.015
1	1	2	1	2	1	1	2	1	50.409	0.370	49.476	0.382	0.019
1	1	2	1	2	1	2	1	1	50.696	0.366	49.397	0.383	0.026
1	1	2	1	2	2	1	1	1	50.456	0.369	49.504	0.381	0.019
1	1	2	1	3	1	1	1	1	50.461	0.369	49.256	0.384	0.024
1	1	2	2	1	1	1	1	2	50.134	0.373	49.695	0.379	0.009
1	1	2	2	1	1	1	2	1	50.997	0.363	49.517	0.381	0.030
1	1	2	2	1	1	2	1	1	50.661	0.367	49.401	0.382	0.026
1	1	2	2	1	2	1	1	1	50.414	0.370	49.497	0.381	0.019
1	1	2	2	2	1	1	1	1	50.497	0.369	49.395	0.383	0.022
1	1	2	3	1	1	1	1	1	50.559	0.368	49.366	0.383	0.024
1	1	3	1	1	1	1	1	2	50.271	0.372	49.617	0.380	0.013
1	1	3	1	1	1	1	2	1	50.593	0.368	49.325	0.383	0.026
1	1	3	1	1	1	2	1	1	50.775	0.365	49.488	0.381	0.026
1	1	3	1	1	2	1	1	1	50.413	0.370	49.475	0.382	0.019
1	1	3	1	2	1	1	1	1	50.368	0.370	49.496	0.381	0.018
1	1	3	2	1	1	1	1	1	50.521	0.368	49.376	0.383	0.023
1	1	4	1	1	1	1	1	1	50.204	0.372	49.476	0.382	0.015
1	2	1	1	1	1	1	1	3	50.709	0.366	49.569	0.380	0.023
1	2	1	1	1	1	1	2	2	50.261	0.372	49.497	0.381	0.015
1	2	1	1	1	1	1	3	1	50.264	0.372	49.469	0.382	0.016
1	2	1	1	1	1	2	1	2	50.617	0.367	49.596	0.380	0.021
1	2	1	1	1	1	2	2	1	51.092	0.361	49.449	0.382	0.033
1	2	1	1	1	1	3	1	1	50.560	0.368	49.199	0.385	0.028
1	2	1	1	1	2	1	1	2	50.335	0.371	49.425	0.382	0.018
1	2	1	1	1	2	1	2	1	50.678	0.367	49.385	0.383	0.026
1	2	1	1	1	2	2	1	1	50.582	0.368	49.533	0.381	0.021
1	2	1	1	1	3	1	1	1	50.046	0.374	49.250	0.384	0.016
1	2	1	1	2	1	1	1	2	50.790	0.365	49.413	0.382	0.028

TABLE I: Continued.

N_L	N_D	N_{F1}	N_{F2-1}	N_{F2-2}	N_{H1}	N_{H2}	N_{H3-1}	N_{H3-2}	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
1	2	1	1	2	1	1	2	1	50.258	0.372	49.320	0.384	0.019
1	2	1	1	2	1	2	1	1	50.477	0.369	49.399	0.383	0.022
1	2	1	1	2	2	1	1	1	50.587	0.368	49.272	0.384	0.027
1	2	1	1	3	1	1	1	1	50.573	0.368	49.371	0.383	0.024
1	2	1	2	1	1	1	1	2	50.867	0.364	49.499	0.381	0.028
1	2	1	2	1	1	1	2	1	50.508	0.369	49.524	0.381	0.020
1	2	1	2	1	1	2	1	1	50.313	0.371	49.481	0.381	0.017
1	2	1	2	1	2	1	1	1	50.419	0.370	49.468	0.382	0.019
1	2	1	2	2	1	1	1	1	50.919	0.364	49.391	0.383	0.031
1	2	1	3	1	1	1	1	1	50.633	0.367	49.496	0.381	0.023
1	2	2	1	1	1	1	1	2	50.756	0.366	49.582	0.380	0.024
1	2	2	1	1	1	1	2	1	50.920	0.363	49.349	0.383	0.032
1	2	2	1	1	1	2	1	1	50.259	0.372	49.465	0.382	0.016
1	2	2	1	1	2	1	1	1	50.410	0.370	49.364	0.383	0.021
1	2	2	1	2	1	1	1	1	50.116	0.374	49.503	0.381	0.012
1	2	2	2	1	1	1	1	1	50.377	0.370	49.360	0.383	0.021
1	2	3	1	1	1	1	1	1	50.517	0.369	49.444	0.382	0.022
1	3	1	1	1	1	1	1	2	50.352	0.371	49.692	0.379	0.013
1	3	1	1	1	1	1	2	1	50.652	0.367	49.458	0.382	0.024
1	3	1	1	1	1	2	1	1	50.375	0.370	49.480	0.381	0.018
1	3	1	1	1	2	1	1	1	50.294	0.371	49.354	0.383	0.019
1	3	1	1	2	1	1	1	1	50.999	0.363	49.476	0.382	0.031
1	3	1	2	1	1	1	1	1	51.162	0.360	49.525	0.381	0.033
1	3	2	1	1	1	1	1	1	50.366	0.370	49.316	0.384	0.021
1	4	1	1	1	1	1	1	1	50.761	0.365	49.410	0.382	0.027
2	1	1	1	1	1	1	1	3	72.170	0.098	71.534	0.106	0.009
2	1	1	1	1	1	1	2	2	72.857	0.089	71.463	0.107	0.019
2	1	1	1	1	1	1	3	1	68.360	0.146	67.650	0.154	0.010
2	1	1	1	1	1	2	1	2	77.768	0.028	76.693	0.041	0.014
2	1	1	1	1	1	2	2	1	72.132	0.098	70.706	0.116	0.020
2	1	1	1	1	1	3	1	1	68.340	0.146	67.672	0.154	0.010
2	1	1	1	1	2	1	1	2	72.507	0.094	71.572	0.105	0.013
2	1	1	1	1	2	1	2	1	68.342	0.146	67.676	0.154	0.010
2	1	1	1	1	2	2	1	1	68.404	0.145	67.688	0.154	0.011
2	1	1	1	1	3	1	1	1	68.321	0.146	67.742	0.153	0.009
2	1	1	1	2	1	1	1	2	72.072	0.099	71.473	0.107	0.008
2	1	1	1	2	1	1	2	1	68.403	0.145	67.604	0.155	0.012
2	1	1	1	2	1	2	1	1	68.357	0.146	67.684	0.154	0.010
2	1	1	1	2	2	1	1	1	68.236	0.147	67.687	0.154	0.008
2	1	1	1	3	1	1	1	1	68.010	0.150	67.678	0.154	0.005
2	1	1	2	1	1	1	1	2	72.688	0.091	71.439	0.107	0.017
2	1	1	2	1	1	1	2	1	68.301	0.146	67.607	0.155	0.010
2	1	1	2	1	1	2	1	1	68.221	0.147	67.736	0.153	0.007
2	1	1	2	1	2	1	1	1	68.200	0.148	67.688	0.154	0.008
2	1	1	2	2	1	1	1	1	67.916	0.151	67.650	0.154	0.004
2	1	1	3	1	1	1	1	1	68.191	0.148	67.682	0.154	0.008
2	1	2	1	1	1	1	1	2	72.500	0.094	71.590	0.105	0.013
2	1	2	1	1	1	1	2	1	68.189	0.148	67.531	0.156	0.010
2	1	2	1	1	1	2	1	1	68.121	0.148	67.753	0.153	0.005

TABLE I: Continued.

N_L	N_D	N_{F1}	N_{F2-1}	N_{F2-2}	N_{H1}	N_{H2}	N_{H3-1}	N_{H3-2}	Theoretical throughput	Theoretical packet loss rate	Simulated throughput	Simulated packet loss rate	Err
2	1	2	1	1	2	1	1	1	68.334	0.146	67.689	0.154	0.010
2	1	2	1	2	1	1	1	1	68.261	0.147	67.690	0.154	0.008
2	1	2	2	1	1	1	1	1	67.930	0.151	67.670	0.154	0.004
2	1	3	1	1	1	1	1	1	68.234	0.147	67.619	0.155	0.009
2	2	1	1	1	1	1	1	2	72.719	0.091	71.508	0.106	0.017
2	2	1	1	1	1	1	2	1	68.576	0.143	67.641	0.154	0.014
2	2	1	1	1	1	2	1	1	68.479	0.144	67.734	0.153	0.011
2	2	1	1	1	2	1	1	1	68.157	0.148	67.639	0.155	0.008
2	2	1	1	2	1	1	1	1	68.602	0.142	67.681	0.154	0.014
2	2	1	2	1	1	1	1	1	68.034	0.150	67.622	0.155	0.006
2	2	2	1	1	1	1	1	1	68.375	0.145	67.679	0.154	0.010
2	3	1	1	1	1	1	1	1	68.229	0.147	67.611	0.155	0.009
3	1	1	1	1	1	1	1	2	72.655	0.092	71.542	0.106	0.016
3	1	1	1	1	1	1	2	1	68.046	0.149	67.601	0.155	0.007
3	1	1	1	1	1	2	1	1	68.268	0.147	67.725	0.153	0.008
3	1	1	1	1	2	1	1	1	68.143	0.148	67.674	0.154	0.007
3	1	1	1	2	1	1	1	1	68.494	0.144	67.685	0.154	0.012
3	1	1	2	1	1	1	1	1	68.394	0.145	67.707	0.154	0.010
3	1	2	1	1	1	1	1	1	68.241	0.147	67.700	0.154	0.008
3	2	1	1	1	1	1	1	1	68.154	0.148	67.638	0.155	0.008
4	1	1	1	1	1	1	1	1	67.796	0.153	67.642	0.154	0.002

in this paper. Based on different phases, protocols, and applications, the values of performance indicators such as system throughput and packet loss rate were obtained. An optimal solution was derived after the results of simulated experiments and theoretical computation were compared, and the combinations of resource allocations were enumerated using a total of 12 resources. Furthermore, by comparing the error between the simulated experiment values and the theoretical computation values, it was found that this model may precisely represent the handling process of the firewall. Therefore, it may save a considerable amount of time in development and testing from the viewpoint of utilizing mobile networks, thus exhibiting significant potential for practical application. In the future, we plan to continue with our in-depth research by emphasizing the discussion of DDoS detection and user-behavior analysis.

Disclosure

The funding agency had no role in the study design, the collection, analysis, or interpretation of data, the writing of the report, or the decision to submit the article for publication.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program (2016YFB0801503).

References

- [1] T. Wang, Z. Su, Y. Xia, B. Qin, and M. Hamdi, "Towards cost-effective and low latency data center network architecture," *Computer Communications*, vol. 82, pp. 1–12, 2016.
- [2] K.-K. R. Choo, "The cyber threat landscape: challenges and future research directions," *Computers and Security*, vol. 30, no. 8, pp. 719–731, 2011.
- [3] V. Prokhorenko, K.-K. R. Choo, and H. Ashman, "Web application protection techniques: a taxonomy," *Journal of Network and Computer Applications*, vol. 60, pp. 95–112, 2016.
- [4] J. Peng, K.-K. R. Choo, and H. Ashman, "Bit-level n-gram based forensic authorship analysis on social media: identifying individuals from linguistic profiles," *Journal of Network and Computer Applications*, vol. 70, pp. 171–182, 2016.
- [5] J. Peng, K. K. R. Choo, and H. Ashman, "User profiling in intrusion detection: a review," *Journal of Network and Computer Applications*, vol. 72, pp. 14–27, 2016.
- [6] J. Peng, R. K.-K. Choo, and H. Ashman, "Astroturfing detection in social media: using binary n-gram analysis for authorship attribution," in *Proceedings of the IEEE Trust-com/BigDataSE/ISPA*, 2016.
- [7] M. Chen, Y. Qian, S. Mao, W. Tang, and X. Yang, "Software-defined mobile networks security," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 729–743, 2016.

- [8] N. Bui and J. Widmer, "Mobile network resource optimization under imperfect prediction," in *Proceedings of the 16th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM '15*, 9, 1 pages, USA, 2015.
- [9] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, and Z. M. Mao, "Mobilyzer: an open platform for controllable mobile network measurements," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '15*, pp. 389–404, 2015.
- [10] C. Herrmann, "Complete analysis of the discrete time finite DBMAP/G/1/N queue," *Performance Evaluation*, vol. 43, no. 2–3, pp. 95–121, 2001.
- [11] T. Wang and M. Hamdi, "Presto: towards efficient online virtual network embedding in virtualized cloud data centers," *Computer Networks*, vol. 106, pp. 196–208, 2016.
- [12] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 936–943, 2012.
- [13] H. Khazaei, J. Mišić, V. B. Mišić, and S. Rashwand, "Analysis of a pool management scheme for cloud computing centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 849–861, 2013.
- [14] H. Khazaei, J. Misic, and V. B. Misic, "A fine-grained performance model of cloud computing centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2138–2147, 2013.
- [15] K. Salah, "Queuing analysis of network firewalls," in *Proceedings of the 53rd IEEE Global Communications Conference, GLOBECOM '10*, 2010.
- [16] K. Salah, K. Elbadawi, and R. Boutaba, "Performance modeling and analysis of network firewalls," *IEEE Transactions on Network and Service Management*, vol. 9, no. 1, pp. 12–21, 2012.
- [17] K. Salah, "Analysis of Erlangian network services," *AEÜ - International Journal of Electronics and Communications*, vol. 68, no. 7, pp. 623–630, 2014.
- [18] S. Zapechnikov, N. Miloslavskaya, and A. Tolstoy, "Modeling of next-generation firewalls as queueing services," in *Proceedings of the 8th International Conference on Security of Information and Networks, SIN '15*, ACM, 2015.
- [19] R. Ghosh, F. Longo, V. K. Naik, and K. S. Trivedi, "Modeling and performance analysis of large scale IaaS clouds," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1216–1234, 2013.
- [20] K. Salah, J. M. A. Calero, S. Zeadally, S. Al-Mulla, and M. Alzabi, "Using cloud computing to implement a security overlay network," *IEEE Security and Privacy*, vol. 11, no. 1, pp. 44–53, 2013.
- [21] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile Micro-Task Allocation in spatial crowdsourcing," in *Proceedings of the 32nd IEEE International Conference on Data Engineering, ICDE '16*, pp. 49–60, IEEE, Helsinki, Finland, 2016.
- [22] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "SLADE: a smart large-scale task decomposer in crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2018.
- [23] S. Xuan, W. Yang, H. Dong, and J. Zhang, "Performance evaluation model for application layer firewalls," *PLoS ONE*, vol. 11, no. 11, Article ID e0167280, 2016.