ISSN No. 0976-5697

Volume 6, No. 2, March-April 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

RTO randomization for Low rate DOS attack on a Feedback controlled system

Jayanthi S.
M.Tech student, Dept of Computer Science and Engineering
SRM University, Chennai

Arun Kumar S.
Asst. Professor,Dept of Computer Science and Engineering
SRM University, Chennai

Abstract: Low rate Denial of Service attacks (LRDoS) send high intensity attack pulses in pulsed ON/Pulsed OFF pattern to degrade the Victims performance and could not be easily detected by the conventional DOS attack detection systems. In the growing need for internet applications, feedback control is a critical element of many internet services in order to ensure the agreed Quality of Service. Recent studies have revealed the vulnerability of feedback control based internet services to LRDoS and the behaviour of the systems under attack. In this paper, we analyse the LRDoS attack on a TCP flow in a feedback controlled system and suggest a practical solution to mitigate the performance degradation of the system due to the attack. Varied reviews about Shrew attacks on TCP flows have inferred that the throughput of the victim approaches a near Zero level when attacked by a pulse with a period tuned in the scale of the RTO. This presents a feasible solution to recover from throughput degradation by randomizing the RTO range meticulously. In this paper the benefits of feedback control and RTO randomization are combined to address the effects of TCP targeted LRDoS attack. This is achieved with our proposal to adjust the RTO range based on the Throughput received as feedback from system process output.

Keywords: TCP, low rate DOS, Shrew, RTO, throughput, feedback

I. INTRODUCTION

Low rate denial of service attacks (LRDoS) attacks pose a great threat to the security of cyber space. The traditional dos attacks consume network bandwidth or CPU cycles by flooding the network with packets resulting in denial of quality of service to legitimate users [1] [2]. These attacks generally take advantage of the flaws in network design and drawbacks of application architecture and target various protocols through their HTTP, UDP and SYN flood attacks. This paper highlights TCP based LRDoS attack and ways to overcome their effects with the help of a well-designed feedback control system which is discussed in the following sections.

II.BACKGROUND AND MOTIVATION

Low Rate DOS attacks send intermittent attack pulses at a considerable low rate. They try to force the target system to deviate from its actual state and deteriorate the overall performance. Due to their typical pulsed nature (ON and OFF) [3] and non-periodic, behavior they cannot be easily detected by the traditional DOS detection mechanisms [4].

TCP protocol is widely used in network applications like File transfers and E-commerce because of their reliability of data delivery [5]. Early studies on LRDoS [6] have shown that ‘Shrew Attack’ and ‘Reduction Of Quality’ (ROQ) attack exploits the Retransmission Time Out (RTO) feature of the congestion control mechanism of TCP. Experimental analysis has proved that throughput degrades to zero in accordance with the fixed RTO in the presence of intelligent attack traffic. Researches have tried to solve the problem caused due to RTO by randomizing it. Guang, Mario and Sanadidi in their analysis [7] suggested three different RTO ranges for choosing the minimum and maximum RTO values. These ranges have also been proved to defend against zero throughputs. These RTO ranges have to be manually chosen and the system had to be tuned accordingly. However an automated approach for this solution has not been experimentally validated yet.

Detailed research on the world wide network systems to scan their vulnerability to LRDoS confirmed that feedback controlled systems are one among them. Further analysis were carried out in [8] to study the behavior of feedback controlled system under attack and what impact such an impulse would create on a specifically designed web server. The experimental analysis considered a more generic parameter for a webservice like admission rate, which was shown to adjust itself based on the system output which is the measured Utilization level. The experiment was confined to the availability of webservice and how many user requests are attended based on the admission rate in the presence of a periodic attack pulse. This study gives us an extended scope for analyzing a saturated feedback controlled system’s behavior to a non-periodic attack pulse, since the attack pulses are not steady and periodic at many instances.

A simulated model for LRDoS attacks was proposed in [9] where the combined impact of attack pattern and network environment on attack effect was evaluated mathematically. This study also quoted that [6] ignored TCP’s congestion window adaption and thus the impact of network state on TCP throughput. Further to this the mathematical analysis helped to derive the expression for many key parameters like: number of legitimate packets transmitted successfully, Sender’s congestion window, length of bottleneck queue, inter burst period T which causes the outage etc., This model also revealed key configurations of the three important parameters of Shrew attack – the minimum inter burst period for a given burst length, global minimum burst length *L*. These parameter configurations helped us to arrive at appropriate ranges of the attack period and the burst length for launching a successful attack pulse. This analysis also revealed the way of reconfiguring the network resources (such as the bottleneck buffer) to mitigate shrew DoS with given attack pattern. However, the mathematical implementations were not extended to deploy the identified parameter refinements in a feedback based environment.

Research works on TCP flows under Shrew attack and the experimental results achieved with a feedback controlled system paves way for an interesting proposition to analyze the combination of solutions presented in both these scenarios. This fundamental theory outlined in this paper suggests overcoming an external attack that exploits TCP's Congestion avoidance mechanism with the help of the feedback control logic.

III. RELATED WORK

A quick summary of all the research work conducted in the area of Low Rate DOS Attacks is as below:

a. Analyze how Shrew attacks exploit the RTO mechanism and deny bandwidth to TCP flows [6]
b. Effect of shrew attacks on TCP throughput [6]
c. Priority-tagging filtering mechanism called SAP (Shrew Attack Protection), that protects well-behaved TCP flows [10]
d. Randomized RTO as a solution to counter LRDoS attacks and choice of RTO range which proposed the need for a variable RTO rather than a constant RTO in order to improve the performance and avoid throughput degradations [7]
e. Vulnerabilities of feedback controlled systems to LRDoS attacks which explains how the feedback system's self-adjusting behavior is hampered with the attack pulse[8]
f. Behavior of a feedback controlled system under LRDoS attack that illustrates how the feedback mechanism strives to regularize the system performance in the presence of an attack pulse. [8]
g. Study of vulnerability to LRDoS attacks in mono-process or mono threaded servers [11]
h. Study on how LRDoS attack impacts application servers [12]
i. Mathematical model for LRDoS attack on Application servers [13]
j. Testing and analysis of DOS attacks on real time network systems [14] [15]
k. Defensive approach to DOS attacks in a distributed approach which frees the victim's resources and at the same time detects legitimate traffic within suspected data stream and ensures successful data delivery.

IV. TCP TARGETED LRDOS AND FEED BACK CONTROLLED SYSTEM

A. TCP Timeout mechanism:

TCP congestion control operates on two timescales. On smaller timescales of round trip times (RTT), typically 10's to 100's of milli seconds, TCP performs additive-increase multiplicative-decrease (AIMD) control. During severe congestion, TCP operates on longer timescales of Retransmission Time Out (RTO). As a part of Congestion avoidance mechanism, TCP Reno detects loss via either the timeout when the ACKs are not received or when three or more duplicate ACKs arrive. If a loss is detected and less than three duplicate ACKs are received, TCP waits for a period of Retransmission Timeout (RTO) to expire post which the congestion window is reduced to one packet and the packet is retransmitted. Time period between the instances when the packet is dispatched and the moment timeout occurs is called the RTO.

Base RTO is computed as below [16]:

$$Max\ \{\ SRTT + 4 * RTT_{VAR}\ ,\ minRTO\}.$$

Where RTT is the *Round-Trip Time*
SRTT is the *Smoothed RTT*
$\mathrm{RTT_{VAR}}$ is the *Variance of the measured RTT*

The minimum RTO, $minRTO$, is recommended as 1sec for the purpose of achieving almost maximal throughput [10]. In addition, TCP uses Karn's clamped retransmit backoff algorithm [17] in case of recurring consecutive timeouts. Each successive RTO is twice the value of the previous RTO until it reaches 64 times the base. This is stated as below :

If $SRTT + 4 * RTT_{VAR} < minRTO = 1sec,$
then RTO can be 1, 2, 4, 8, . . . , 64 sec.

B. Shrew Attack:

Shrew attack is a denial-of-service attack on TCP which uses short synchronized bursts of traffic to disrupt TCP connections on the same link, by exploiting a weakness in TCP's retransmission timeout mechanism. To analyze the effect of such a Shrew attack on TCP, we consider a square wave periodic attack pulse with a carefully chosen period.

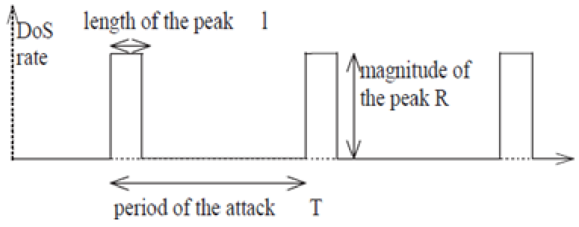

Fig 1: Shrew Attack burst

Consider a single TCP flow and a Shrew attack consisting of periodic "on-off" bursts. These bursts have a magnitude of 0 or 1 in with a period 'T'. The bursts will have sufficiently high magnitude with long "ON" time to induce enough packet losses. The inter-burst period 'T' is carefully chosen to be equal to $minRTO$value of the TCP flow. As shown in *Fig 3*, the TCP sender will wait for a retransmission timer of 1 sec to expire and will then double its RTO. After RTO seconds the sender will attempt to retransmit the lost packets. At this time a new Shrew attack burst is fed pushing the sender to enter timeout again. If the attacker creates a second outage between time 1 and 1+2RTT, it will force TCP to wait another 2 sec. By creating similar outages at times 3, 7, 15 etc., an attacker can deny service to the TCP flow while transmitting at extremely low average rate. If all flows have a fixed $minRTO$parameter, the attacker can create periodic outages and cause the TCP flow to continuously timeout.

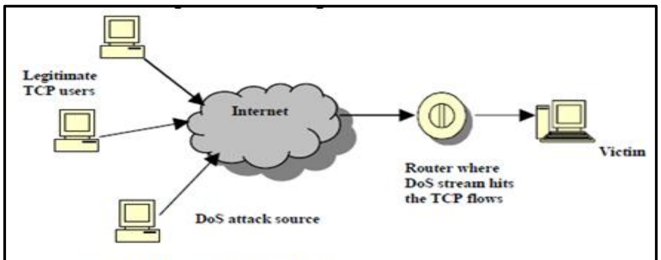

Fig 2: A system under attack

The rationale behind Shrew attack is that when the initial attack burst causes packet drops for a TCP flow, the sender will wait for the retransmission timer to expire before beginning to retransmit. As RTO is an integral multiple of the $minRTO$, all the subsequent retransmissions would face further attack pulses and hence get dropped repeatedly. This is because the attack interval is synchronized with the retransmission timeout value. As a result, the TCP flow fails to exit the timeout phase and experiences near-zero throughput or the TCP session is closed failing to serve the user requests.

C. *TCP flow throughput and LRDOS attack pulse:*

Fig. 3 shows a single Shrew attack pulse which is a square waveform with attack period T, burst length L, and a peak rate R. Kuzmanovic and Knightly [6] showed that such an attack can reduce the throughput of TCP flows to near zero throughputs or cause session resets if the attack has the following characteristics:

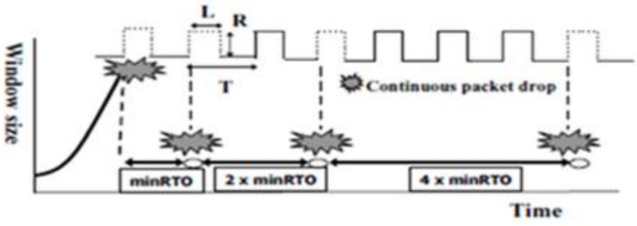

Fig 3: Shrew Attack Vs TCP Timeout Mechanism

a. R is large enough to induce victim's packet loss.
b. L is long enough to induce timeout (e.g., typically not less than the round-trip time), but sufficiently short to escape detection.
c. T is chosen in accordance with the minRTO value such that when flows attempt to exit timeout, they will face continuous drop (i.e., T is scaled in accordance to the minRTO).

Kuzmanovic and Knightly [1] also showed that the normalized TCP throughput under a Shrew attack is:

$$\boldsymbol{\mu_{norm}(T)} = \frac{\left\lceil \frac{minRTO}{T} \right\rceil T - minRTO}{T} \quad \textbf{(1)}$$

where, T is the attack period

Expression (1) also shows that if the Shrew Attack has sufficient peak rate and $T = minRTO$, Throughput Tapproaches Zero.

D. *Vulnerabilities of feedback controlled system to LRDoS:*

Feedback control is a fundamental building block for many network protocols, and Internet services which are designed to handle dynamic service demands. Such a system self-adjusts its configuration based on the feedback it has received on the state of the system [18][19][20][21]. If the current state of the system deviates from the desired state thereby hindering the performance or Quality of Service guaranteed, the control would try to autocorrect and direct the system towards the preferred output or response. A best Internet example for Feedback controlled system would be the TCP congestion control dynamics with an active queue management (AQM) scheme at a router.

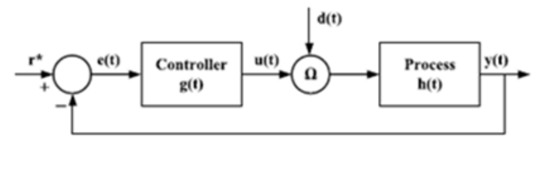

Fig 4: A general Feedback controlled system

Recently there has been more focus on the effect of LRDoS attacks on feedback controlled system to study how the system responds to the attack and how sooner it recovers to a desirable stable state. When the system encounters an attack pulse, it will be temporarily overloaded [22]. The consequences to this overloading are:

a. New requests will be refused during the attack, because of resource degradation by illegitimate requests
b. The system takes more time to recover to the normal state using the feedback controller because of the false feedback signals induced by the attack pulses which forces the victim to operate in a low-throughput region.

Fig. 4 shows a typical feedback controlled system [8] which comprises two major components: a process h(t) and a controller g(t). h(t) represents any Internet service (e.g., web service, video streaming, etc.) while g(t) generates a control signal (i.e., u(t)) to regulate h(t) [23]. The input to the controller is a control error e(t), which is the difference between output y(t) and the expected value r∗ . y(t) can be any measurable metric, such as system utilization or queue length and four our analysis in the below sections, it will be the TCP Throughput. r∗ is usually selected for the system to achieve the best performance by the system designer, and the controller drives y(t) towards r ∗ based on e(t).

When subjected to an attack, a feedback controlled system moves through three different stages before returning to the steady state [8][24]. The stages are Saturation, Recovery I, and Recovery II stages and considered to be behaving as below :

a) Saturation stage - Overall system output is less than the desired value and so the system enters the next two recovery stages
b) Recovery I - System output is still less than the desired value and as per the design logic of the feedback control, the control signal increases trying to push up the system output.
c) Recovery II - In continuation to Recovery I, this stage ends when the system output manages to reach the desired value.

V. RTO RANDOMIZATION TO COUNTER LRDOS

Randomizing RTO is a proposed possible solution against low-rate TCP-targeted DoS attacks [4]. When RTO is not a fixed value, attacker cannot easily predict the next TCP timeout interval and plan a timed periodic attack burst. By meticulously choosing the RTO ranges, the undesirable effects of the attacks can be avoided. Instead of using a deterministic value of 2^k for the k^{th} successive timeout (TCP does not double the timeout value after k > 6), we choose a random value uniformly between 2^k and $2^k + 1$.

By expanding RTO across a range of values, different TCP connections produce different RTOs after an attack burst and so the attacker is not able to synchronize the next round of timeout. This choice of RTO range also adheres to the conventional approach such that it always produces a RTO greater than or equal to the value chosen in legacy TCP. Statistically the average of the randomized RTO is 0.5 sec larger. There are other options also that widen the range to both sides of 2k.

Experimentally from [7], for the k^{th} timeout, these ranges are considered as below :

a. $\{2^k\text{-}0.5 \text{ to } 2^k\text{+}0.5\}$ ➔ represented as [t – 0.5; t + 0.5]

b. $\{0.5*2^k \text{ to } 1.5*2^k\}$➔ represented as [0.5t; 1.5t]

VI. PROBLEM STATEMENT

Analyzing the various possible solution offered to overcome the effects of LRDoS in the internet space, it has been identified that the self-normalizing feature of a typical feedback control system has not been used in conjunction with any of the proposed theories and problem solving techniques.

Hence, this paper recommends a feasible solution to TCP targeted attacks on a feedback controlled system. This is achieved by continuous monitoring of the TCP throughput by the feedback process. On the arrival of an attack pulse, when TCP throughput deviates from its desired value and approaches zero, the control signal generated by the regulate the throughput back to a desired limit. Our study exploits the benefits of a feedback control mechanism and RTO randomization to counteract the adverse effects of LRDoS attacks on a TCP flow.

For this we review the RTO randomization approach proposed in [6],[7] and investigate the solution in parallel with vulnerability of feedback based systems to DOS attacks [8]. The results arrived give us a new insight to proceed with future research on TCP based LRDoS attacks.

VII. ARCHITECTURE

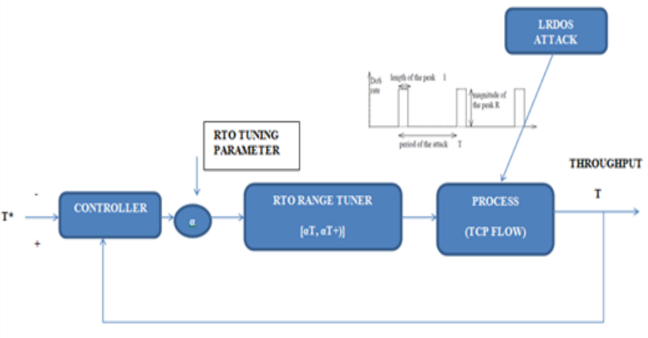

Fig 5: RTO Randomization as a feedback control

The design proposed in this paper works this way – the TCP throughput is constantly monitored by the feedback controlled system and when there is a deviation from the desired level, the control attempts to modify or tune the RTO. So a different RTO value is picked up from the predefined range such that the throughput returns to the normal state. RTO range is also randomized with every attempt. In this manner, the attack pulse period would not be in sync with the RTO and thus the attack would be mitigated.

In this section we propose a novel approach where RTO can be randomized with a tuning parameter α which would be fed as control input to an existing feedback controlled system. This RRFC model (RTO Randomization with Feedback Control) combines the advantage of a feedback controlled system and RTO randomization are exploited in order to direct the system towards a stabilized throughput, when it is trying to deviate from its desired performance levels. Fig 5 shows how RTO Randomization operates as a feedback control.

As shown in Section IV statement (1), when the burst period is meticulously chosen by the attacker to be equal to the min RTO, the throughput approaches a value Zero. Refer Section V, where RTO is suggested to be selected within a range [t – 0.5; t + 0.5) and [0.5t; 1.5t] for a randomized value. Instead of choosing a whole value as stated here, the RTO can be further tuned more intricately with a parameter α. So, the chosen RTO range would be [αt, αt+1]. The choice of this α is another challenge in order to ensure that the actual benefits of RTO randomization are not lost.

The proposed system works as follows:

a. The designed feedback controlled system continuously monitors the Throughput *T* of the TCP flow as per normal methodology used for Throughput measurement in any network device.

b. When the TCP flow is subjected to a shrew attack pulse as explained in Section IV, the system tends to show a degraded throughput and a continuous Shrew attack will render the throughput to reach a zero value.

c. The throughput T is the feedback signal fed to the controller which is positioned to initiate control signal for stabilizing the system. When the throughput T is zero or near Zero, the system validates the input traffic to confirm the presence of LRDOS traffic. This can be accomplished using techniques like Distributed detection, dynamic time warping (DTW)etc., [5]

d. Once the attack pulse is confirmed of its existence, the error signal initiates the RTO Tuning Parameter system to supply a suitable parameter value **α**. The tuning parameter **α** would be based on the throughout measured as the system output.

e. Upon choosing the appropriate parameter value α, the RTO range is selected and fed to the TCP flow which totally alters the RTO range of the TCP flow.

f. As the RTO range is now different and the min RTO is not same as the burst period, the Shrew attack pulse will no longer be able to use the Re-

Transmission Timeout feature of TCP congestion control mechanism to force an attack or loss of packets.

g. If the attacker identify the new range and target an attack, the process would repeat recursively and the RTO range will be shifted to another new range followed by choice of an alternate RTO value from the new range.

The flowchart in Fig 6, shows the functional methodology of RRFC model.

A. Client – Server – TCP Session:

Our analysis starts with an established TCP session between a client and server, which could be any service request/response in a real time network. There is a sequential flow of packets between the client and server.

B. Attack:

An attacker sends in square wave Shrew attack pulses to interfere with the data transfer between client and server, which tries to degrade the throughput as explained in [6]

C. Feedback Control:

The feedback control section of our RRFC model comprises of the below key sections which functions recursively:

a. Throughput measurement

b. LRDoS detection

c. RTO tuner

D. Throughput measurement:

The throughput of the established session is measured using any preferred method continuously. If a zero throughput is measured, the control is transferred to the LRDoS detection system

E. LRDoS Detection:

This section detects if the session stream between client and server contains LRDoS attack packets. This can be accomplished through available methodologies like Random Early detection [3], Distributed detection, DTW [5], Information metrics explained in [25].

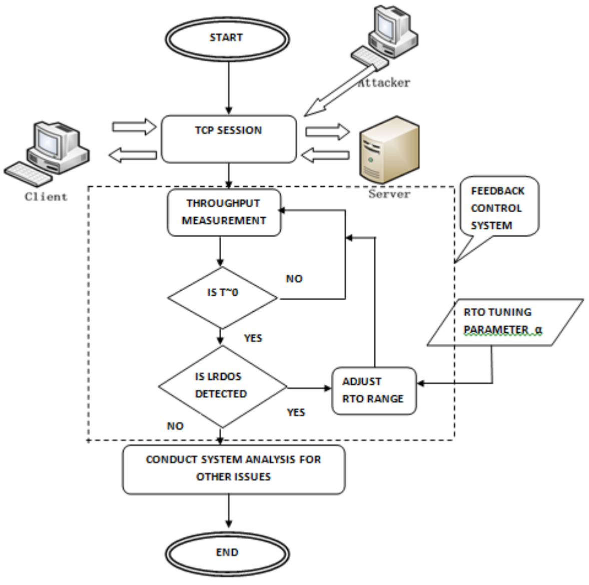

Fig 6: Architecture Diagram with feedback control

In case the throughput issues are not due to the LRDoS threat, then the network system would need to be diagnosed for other network issues which are out of the scope of our analysis here.

F. RTO Tuner:

Once LRDoS attack is detected in the transmitted packets, the control signal is turned ON to generate a tuning parameter **α** which is shift the RTO range there y changing the $\mathrm{RTO}_{\mathrm{lower}}$and $\mathrm{RTO}_{\mathrm{Upper}}$values for the next cycle.

G. Throughput Normalization:

With the new RTO range, the packets are transmitted as per the new RTO values chosen from the new range and hence the attack pulse will not be coherent with the $minRTO$ value.

When the attacker changes the attack pulse period, the throughput measured will again cause a change in the RTO range and this process is repeated continuously in a recursive loop.

VIII. THROUGHPUT ANALYSIS WITH RANDOMISED RTO

With the help of statistical derivations obtained in [7], we shall deploy Consider a single TCP connection under attack. After the attack starts, all TCP packets in the current window are dropped; the connection is forced to timeout with RTO as p_{out}. When the TCP connection comes out from timeout and transmits the first packet p_{out}later, it resorts to one of the following cases :

1. $p_{out} - \lfloor \frac{p_{out}}{T} \rfloor . T \geq l + RTT_{max}$

In this case :

a. Bottleneck queue is empty with no attack

b. TCP can use up to the full bandwidth until the next attack burst

c. Utilization is given by the expression

$$\text{Utilization}=\frac{(\lfloor \frac{p_{out}}{T} \rfloor . T \; - p_{out})}{(\lfloor \frac{p_{out}}{T} \rfloor . T} \qquad (2)$$

2. $l \leq \; p_{out} - \lfloor \frac{p_{out}}{T} \rfloor . T < l + RTT_{max}$

In this case :

a. Bottleneck queue has some attacking packets

b. There is no attack from the burst underway

c. TCP packets must wait until all attacking packets are cleared before they are transmitted.

d. This wait time is estimated as RTT_{max} from the end of last attack burst.

$$\text{Utilization} = \; (T - \; l - \; RTT_{max}) \,/\, (\lfloor \frac{p_{out}}{T} \rfloor . T \quad (3)$$

3. $p_{out} - \lfloor \frac{p_{out}}{T} \rfloor . T < l$

a. Connection finds the attack

b. Packet is dropped and connection timeouts again

c. There is no successful transmission of packets in the period p_{out}

d. Next cycle begins at the moment the new attack burst arrives.

e. Average throughput is given using the renewal theory as below :

$$\frac{\textit{Average number of transmissions in a cycle}}{\textit{Average cycle time}}$$

A function *f*(*p*) is given by

$$f(p) = \begin{cases} 1 & \text{if } p - \lfloor \frac{p}{T} \rfloor .T \geq l + - RTT_{max} \\ 0 & \textit{elsewhere} \end{cases}$$

Functions *g*(*p*) and *h*(*p*) are defined similarly.

$$g(p) = \begin{cases} 1 & \textit{when case 2 is true} \\ 0 & \textit{elsewhere;} \end{cases}$$

$$h(p) = \begin{cases} 1, & \textit{when case 3 is true} \\ 0 & \textit{elsewhere;} \end{cases}$$

From the above expressions we derive the average transmission time, average cycle length and steady-state TCP throughput.

TrT_{avg} ➔ *Average Transmission time in a cycle*

$L_{cycleavg}$ ➔ *Cycle Length*

μ_{Steady} ➔ *Steady state Throughput*

$$TrT_{avg} = \frac{\int_{p_{0,min}}^{p_{0,max}} \left([f(p) \left(\lfloor \frac{p}{T} \rfloor .T - p \right) + g(p)(T - l - RTT_{max})] \, dp \right.}{p_{0,max} - p_{0,min}} \quad (4)$$

$$L_{cycleavg} = \frac{\int_{p_{0,min}}^{p_{0,max}} \left([f(p) \left(\lfloor \frac{p}{T} \rfloor .T \right) + g(p) \left(\frac{p}{T} \rfloor .T \right) + h(p)p] \, dp \right.}{p_{0,max} - p_{0,min}} \quad (5)$$

Steady state throughput, $\mu_{Steady} = \frac{TrT_{avg}}{L_{cycleavg}}$ (6)

In order to analyze the performance of the system under LRDoS attack, we shall begin by comparing the throughput in the case of conventional TCP system without randomization and the arrived throughput with a RRFC system that is proposed.

Let us consider the simulation set up for one-hop scenario illustrated in [1]

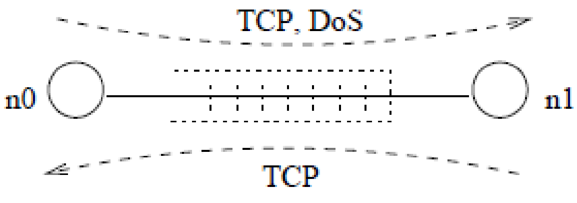

Fig 7: One-hop scenario

Bottleneck capacity is 1.5 Mbps with 6 msec one-way propagation delay. Queue buffer size is 25 packets such that RTT varies between 12 and 132 msec. Since the buffer size is relatively large, the attack rate is set as twice the bottleneck capacity (i.e. 3 Mbps) so as to fill the queue fast. The attack burst length is 80 msec, comparable to RTT. One TCP connection and the attack flow start at *n*0 and end at *n*1. We also set up another TCP connection going in the reverse direction as background traffic. Each simulation runs 300 seconds.

Fig. 8 shows normalized TCP throughput with or without RTO randomization. Without randomization, TCP throughputdrops to near zero when inter-burst period is 0.5 or 1 sec, asmentioned in [6]. With any of the randomization ranges, TCPutilizes at least 15% of the bandwidth resource when inter-burst period is 0.5 sec, more than 30% in all other cases, noticeably around 45% when inter-burst period is 1 sec.

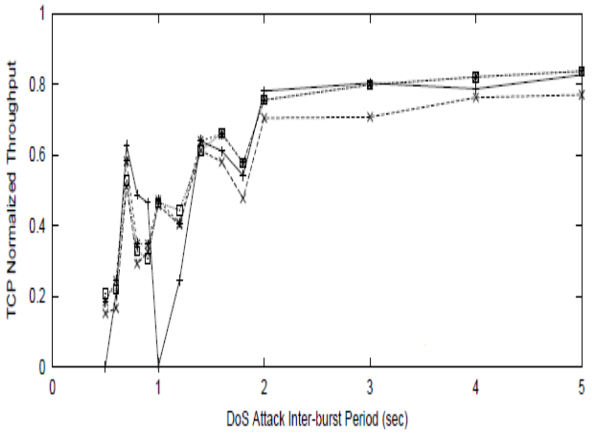

Figure 8: Throughput with and without Randomization. Refer [4]

We can also see that when the inter-burst period is much longer than 1 sec, RTO randomization does not undergo much performance degradation. All three randomization ranges tried in [4] showed similar results in terms of TCP throughput.

A simple analysis of the various values that RTO can take during a TCP timeout is shown in Table 1. First, we consider a typical RTO value that increase as an integral power of 2 for every timeout. This is followed by the concept explained in [4] considering a random RTO range of {2^k-0.5 to 2^k+0.5} and {0.5 * 2^k to 1.5 * 2^k} for the kth timeout. The last eight columns give a sample of lower and Upper bounds of RTO range $RTO_{lower}$and RTO_{Upper} based on the tuning parameter α selected as the control signal in the feedback control system. Based on how minute the value of **α** is chosen, the number of RTO ranges can be more helping us study the behavior of the system more deeply.

Table 1: RTO ranges Conventional Vs Randomized

Time out	Conventional approach	Randomized RTO range				Tuned Randomized RTO based on α							
						α=0.2		α=0.4		α=0.6		α=0.8	
k	2^k	CASE 1 : {2^k-0.5 to 2^k+0.5}		CASE 2: {0.5 * 2k to 1.5 * 2k}		$α2^k$	$α2^k+1$	$α2^k$	$α2^k+1$	$α2^k$	$α2^k+1$	$α2^k$	$α2^k+1$
1	2	1.5	2.5	1	3	0.4	1.4	0.8	1.8	1.2	2.2	1.6	2.6
2	4	3.5	4.5	2	6	0.8	1.8	1.6	2.6	2.4	3.4	3.2	4.2
3	8	7.5	8.5	4	12	1.6	2.6	3.2	4.2	4.8	5.8	6.4	7.4
4	16	15.5	16.5	8	24	3.2	4.2	6.4	7.4	9.6	10.6	12.8	13.8
5	32	31.5	32.5	16	48	6.4	7.4	12.8	13.8	19.2	20.2	25.6	26.6
6	64	63.5	64.5	32	96	12.8	13.8	25.6	26.6	38.4	39.4	51.2	52.2

IX. THROUGHPUT ANALYSIS OF FEEDBACK CONTROLLED SYSTEM POST RTO RANDOMIZATION

Normalized throughput of the designed feedback controlled system under LRDoS attack is studied by simulating the results with the possible inputs of Attack burst period and Randomized RTO. Randomized RTO can be chosen as any value such that the range is 1 sec.

Based on the simulation, the throughput values achieved are consolidated in the graphs below. Fig.9, Fig.10 and Fig.11 depict the achieved throughput values when the attack burst period was chosen as 0.5 sec, 1 sec and 2 sec respectively.

For each case of the attack bust, four different ranges of RTO are used for our analysis to assess the throughput performance:

a. Throughput with attack and no RTO randomization
b. Throughput with attack and RTO randomized as per [1]
c. Throughput with attack and RTO randomized as per RRFC solution covered in section VI.

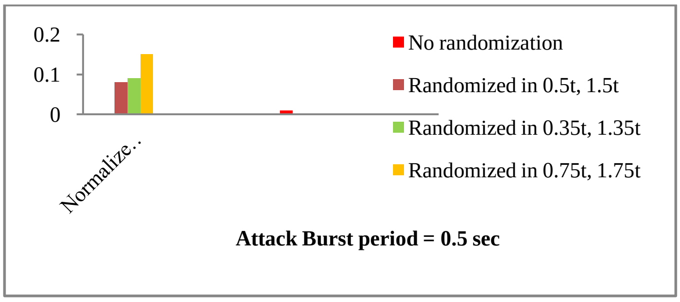

Figure 9: Normalized Throughput Vs RTO | Burst period – 0.5 sec

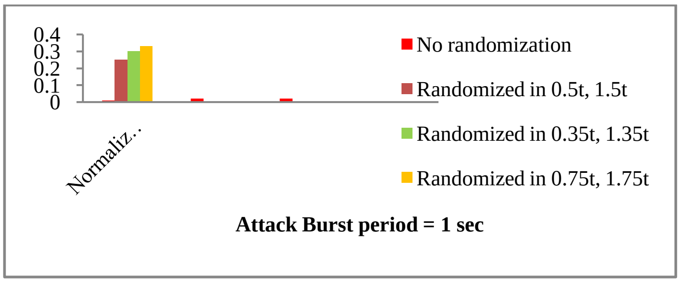

Figure 10: Normalized Throughput Vs RTO | Burst period – 1.0 sec

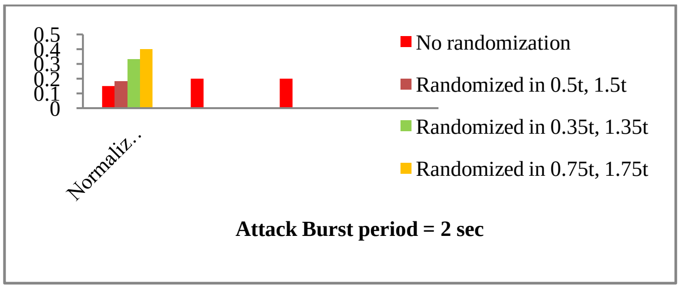

Figure 11: Normalized Throughput Vs RTO | Burst period – 2 sec

X. THREAT ANALYSIS

a. Numerous theories have been proposed in order to detect and overcome the effects of LRDoS attacks on a network topology. Most of the theories highlighted on external mechanisms to defend the attacker but did not concentrate on manipulating the internal protocol parameters in order to build a defensive mechanism. RRFC model explained in our paper focuses only on the built-in RTO feature of TCP protocol to arrive at a self0defensive approach.
b. While some of the studies that discussed on RTO randomization by showing mathematical models, they showed only manually tuning the parameter from the victims end and not by enhancing the prevailing system to auto-correct to the new attack situations.
c. The feedback control mechanism which is a building block of all network systems has not been involved to a greater extent on LRDoS attack detection and avoidance.
d. In order to override all the above issues, RRFC model gives a more viable option to randomize RTO with the help of Feedback mechanism there by improving the Throughput of the system.

XI. CONCLUSION

This paper presents a novel approach to address LRDoS attacks on a feedback controlled system by exploiting the advantageous features of RTO randomization and feedback control. Shrew attacks that target the TCP based systems make use of the TCP timeout mechanism designed for Congestion control in the system, to align its attack burst period with the RTO of the system. Ongoing studies in the area of TCP based LRDoS attacks propose to randomize RTO as a feasible solution to recover the throughput degraded due to the attack burst. In addition to this, we have suggested a new approach to randomize the RTO by using a feedback system where a RTO tuning parameter functions as the control signal. The feedback system is continuously monitoring the throughput of the system and when the output attains a zero level due to the effect of Shrew attack pulse, the control signal triggers a parameter to tune the RTO and shifts it to a different range.

This study can be further enhanced with intensive research to generate the RTO tuning parameter to achieve the desired throughput levels. This would pave way to improve the overall system performance while trying to defend against the Low Rate DoS attacks. RRFC model proposed here can also be combined with other defense mechanisms for enhanced security features against LRDoS attacks. Experimental analysis of the proposed solution using a variety of feedback controlled systems is also a logical approach for further research.

XII. REFERENCES

[1]. A. Shevtekar, K. Anantharam, and N. Ansari, "Low rate TCP Denial of-service attack detection at edge routers" IEEE Commun. Lett., vol. 9,no. 4, pp. 363–365, Apr. 2005.

[2]. GeorgiosLoukas and GulayOke, "Protection against Denial of Service Attacks: A Survey", Intelligent Systems and Networks Group, Imperial College London

[3]. Prashant Viradiya, DivyarajsinhVaghela and DharmeshDhangar, "Study of Low Rate Denial of

Service (LDoS) attacks on Random Early Detection (RED)”

[4]. AmeyShevtekar and Nirwan Ansari **“**A Proactive Test Based Differentiation Technique to Mitigate Low Rate DoS Attacks”, New Jersey Institute of technology, Newark, NJ 07102, USA

[5]. Haibin Sun John C.S. Lui, “Distributed Mechanism in Detecting and Defending Against the Low-rate TCP Attack”, International Conference of Network Protocols (ICNP) 2004

[6]. A. Kuzmanovic and E. Knightly, “Low-rate TCP-targeted Denial-of service attacks: The shrew vs. the mice and elephants”, in Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun., Aug. 2003

[7]. G. Yang, M. Gerla, and M. Y. Sanadidi, “Randomization: Defense against Low-Rate TCP-targeted Denial-of-Service Attacks” in Proc. IEEE Symposium on Computers and Communications, July 2004

[8]. Yajuan Tang, XiapuLuo, Qing Hui, and Rocky K. C. Chang, “Modeling the Vulnerability of Feedback-Control Based Internet Services to Low-Rate DoS Attacks”, IEEE Ttransactions on information forensics and security, vol. 9, no. 3, March 2014

[9]. JingtangLuo, Xiaolong Yang, Jin Wang, Jie Xu, Jian Sun and Keping Long, “On a Mathematical Model forLow-Rate Shrew DDoS”, IEEE Transactions on information Forensics and Security, Vol. 9,, No. 7, July 2014

[10]. Chia-Wei Chang, Seungjoon Lee, Bill Lin and Jia Wang “The Taming of The Shrew: Mitigating Low-Rate TCP-Targeted Attack”, 29th IEEE International Conference on Distributed Computing Systems (ICDCS 2009), 22-26 June 2009

[11]. Gabriel Maci´a-Fern´andez, Jes´us E. D´ıaz-Verdejo, and Pedro Garc´ıa-Teodoro, “Low Rate DoS Attack to Monoprocess Servers”, Security in Pervasive Computing, Third International Conference, SPC 2006, York, UK, April 18-21, 2006, Proceedings

[12]. Ying Zhang, Z. Morley Mao, Jia Wang, “Low-Rate TCP-Targeted DoS Attack Disrupts Internet Routing” National Science Foundation,2010.

[13]. Gabriel Maci´a-Fern´andez, Jes´us E. D´ıaz-Verdejo, and Pedro Garc´ıa-Teodoro, “Model for Low-Rate DoSAttacksAgainst Application Servers”, IEEE Transactions on Information Forensics and Security, Vol. 4, No. 3, September 2009

[14]. Jin-Seok Yang, Min-Woo Park, Tai-Myoung Chung, “A Study on Low-rate DDoS Attacks in Real Networks”

[15]. Jin-Seok Yang, Hyoung-Chun Kim and Tai-Myoung Chung, “A DDoS Attack Test, Analysis and Mitigation Method in Real Networks” The KIPS Transactions on Computer and Communication Systems, Vol.2, No.3, pp.125-132, 2013.

[16]. V.Paxson and M. Allman, “Computing TCP’s Retransmission Timer,” Internet RFC 2988, Nov. 2000.

[17]. P. Karn and C. Patridge, “Improving Round-Trip Time Estimates in Reliable Transport Protocols” In Proceedings of ACM SIGCOMM 1987, Aug.

[18]. T. Abdelzaher, K. Shin, and N. Bhatti, “Performance guarantees for web server end-systems: A control-theoretical approach,” IEEE Trans.ParallelDistrib. Syst., vol. 13, no. 1, pp. 80–96, Jan. 2002.

[19]. J. Hellerstein, Y. Diao, S. Parekh, and D. Tilbury, “Feedback Control of Computing Systems”, Hoboken, NJ, USA: Wiley, 2004.

[20]. M. Welsh and D. Culler, “Adaptive overload control for busy internet servers,” in Proc. USENIX Symp. Internet Technol. Syst., 2003, pp.

[21]. Y. Lu, T. Abdelzaher, C. Lu, L. Sha, and X. Liu, “Feedback control with queuing-theoretic prediction for relative delay guarantees in web servers,” in Proc. 19th IEEE RTAS, May 2003, pp. 208–217.

[22]. Yajuan Tang, XiapuLuo and Rocky Chang, “Degrading Internet Services by intermittent false feedbacks and the counter measures”, 17th International Workshop on Quality of Service, IWQoS 2009, Charleston, South Carolina, USA, 13-15 July 2009

[23]. J. Hellerstein, Y. Diao, S. Parekh, and D. Tilbury, “Feedback Control of Computing Systems” Hoboken, NJ, USA: Wiley, 2004.

[24]. YannLabit, YassineAriba and Fr´ed´ericGouaisbaut, “On Designing Lyapunov-Krasovskii Based AQM for Routers Supporting TCP Flows”, Decision and Control, 2007 46th IEEE Conference on; Jan 2008

[25]. Yang Xiang, Ke Li, Wanlei Zhou, “Low-Rate DDOS Attacks Detection and Traceback by using New Information Metrics”, IEEE Transactions on Information Forensics and Security Vol., 6, No 2, June 2011.