Shrew Attack Prevention in RED Queue with Partial Flow Analysis

Lija Mohan
Department of Computer Science
Federal Institute of Science & Technology, Kerala.

Jyothish K. John
Department of Computer Science
Federal Institute of Science & Technology, Kerala.

Bijesh M.G.
Techncal Director
Quest Innovative Solutions Pvt. Ltd.
Kadavanthra, Kerala.

ABSTRACT
Shrew Attacks or Low Rate Denial of Service(LDoS) Attacks are initiated by sending large amount of packets for very short span of time such that the packet sending rate crosses the link capacity resulting in network congestion. Compared to Denial of Service (DoS) Attack, LDoS attack is very difficult to be detected because, the attacker can maintain low average packet sending rate while executing an attack. If the rate and interval of LDoS attack is properly estimated and executed, this attack can cause a severe threat to the retransmission time out adjustment of TCP and hence reduce its throughput to near zero. This paper proposes a lightweight LDoS filter which can be added with Preferential Dropping RED, to detect and prevent LDoS packets before they reach RED dropping policy. The advantage of this method is that only partial flows need to be analyzed to detect an attack. Simulations done in NS2 shows that, our method can effectively mitigate LDoS attack while maintaining fairness in bandwidth and low average queuing delay.

General Terms
Networking, Security, Intrusion Detection System

Keywords
LDoS, RED-PD, RTO, TCP, Network Security

1. INTRODUCTION
Denial of Service (DoS) Attacks are initiated by sending large amount of unwanted packets for long period of time thus creating congestion in the network. Thus for a normal user the service gets denied. Low Rate Denial of Service (LDoS)[1] attack is a variant of DoS attack in which large amount of packets are sent for a short span of time and this process repeats over several intervals. If the packet sending rate is chosen such that it crosses the link capacity, then congestion occurs in network. Also it is very difficult to detect such attack, since the attacker can maintain low average rate even while creating network congestion.

LDoS attacks are initiated at proper time intervals. This interval is chosen such that it coincides with the retransmission time out (RTO) period of TCP. According to Karn's Algorithm [5], initially as a part of its window adjustment mechanism TCP send 1MSS packet by setting its RTO as 1ms. If the packet is delivered within this RTO period, TCP doubles its packet size and halves its RTO time and the process continues. But if the packet is not delivered within that RTO, it doubles its RTO value and tries again to deliver the packet. TCP will set its RTO period as that period in which the packet is successfully delivered. In case of LDoS attack, attacker will initiate an attack at 1ms thus producing congestion; so TCP cannot deliver the packets, it doubles its RTO value and send the packet again. Knowing this, attacker will initiate the attack at 2ms. Again the TCP packets get dropped. Similarly proceeding TCP goes on increasing its RTO value without being able to deliver a single packet. Thus the throughput of TCP is reduced to near zero due to LDoS attack.

Studies [4] conducted on LDoS attack shows that Active Queue Management Schemes like RED, SFQ, etc are adversely affected by this attack.

Preferential Dropping RED (RED-PD) [6] is a variant of RED [7] which monitors the flows which consume bandwidth above a target bandwidth and proportionally drops packets from these monitored flows at times of congestion. Thus RED-PD monitors high bandwidth consuming flows making use of only partial flow analysis (i.e. only the flows that consume more than target bandwidth are monitored). RED-PD is able to prevent DoS attacks but fails to prevent LDoS attacks.

In the proposed scheme, a filter is added with RED-PD to detect LDoS attacks from among the monitored flows and only those packets which survive this filter will reach the normal RED's dropping policy. Among different methods existing to prevent LDoS attack, ours is the first one which can detect attack by analyzing only partial flows. Hence complexity reduces to O(n), where n implies number of monitored flows. Another advantage of integrating LDoS filter with RED-PD is that it provides max-min fairness of bandwidth among the flows due to its fair dropping probability determination.

In the next section we discuss some existing solutions to prevent LDoS attack. Section 3 describes the system architecture and implementation aspects of RRED-PD in detail and Section 4 evaluates it using NS2 Simulation. Finally Section 5 concludes the paper.

2. Previous Work
According to the studies conducted in [1], the only effective method to prevent LDoS attack is to randomize TCP retransmission time out period But Karn's algorithm is proven to be the optimal solution for RTO adjustment under no attack. Other methods for preventing LDoS attacks are pattern matching approaches [2]. The input traffic pattern is compared with an attacking traffic pattern. If these patterns match, then an attack can be suspected. But pattern matching approaches detect attacks with expensive arithmetic operations. [3] studies the effect of router buffer sizes on LDoS attack. According to them by increasing the buffer size of router, effect of attack can be reduced. But this method neither provides a mathematical model for detecting LDoS attack nor cost efficient. [8] proposed Robust RED technique to prevent LDoS attack. Here an LDoS filter is added before normal RED to detect and drop attacking packets. But this method makes use of per flow analysis to detect an attacking flow which increases the execution latency and complexity of detection process. The method adopted in our

paper uses only partial flows to detect an attack. Initially some flows are categorized as "to be monitored" based on the bandwidth they consume and only such flows are later analyzed to detect whether they initiate an LDoS attack.

3. System Architecture Of Preferential Dropping RED with LDoS Attack Prevention

The proposed system (illustrated in fig 1) uses preferential dropping RED (RED-PD) with a filter as in Robust RED to detect and prevent LDoS attack packets. Therefore our work can be considered as a successor to RED-PD and Robust RED (RRED) and hence can be named Robust RED-PD or RRED-PD.

When packets from different flows reach RED-PD, the flows will be categorized as "monitored" if its bandwidth consumption exceeds the target bandwidth of network. Else the flow will pass through normal RED dropping mechanism. A monitored flow initially reaches the pre-filter of RED-PD where the packets are dropped (only if congestion occurs in network) proportional to their bandwidth consumption. Thus a max-min fairness of bandwidth is provided among the flows i.e. flows which consume more bandwidth will suffer more drops at times of congestion.

Next the packets reach LDoS packet detection and filtering mechanism where the LDoS attack packets are detected and filtered out. Anyway the attacking packets will always consume high bandwidth. Such flows will be monitored. So we need to filter out LDoS attacks from those monitored flows thus avoiding per flow scheduling. After filtering attacking packets, it passes through normal RED mechanism.

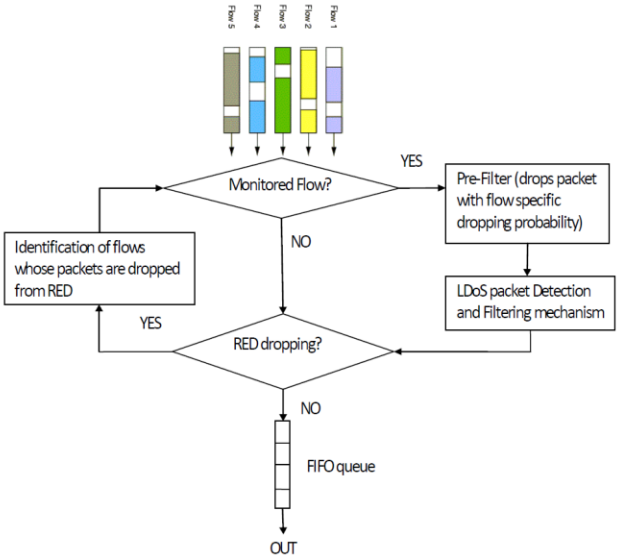


Fig 1: RRED-PD : Preferntial Dropping RED with an LDoS filter

A flow which initially consumed less bandwidth may consume more lately. Such flows are identified using the identification engine of RRED-PD. They will be added to "monitored flows" from next iteration onwards.

3.1 LDOS packet Detection and Filtering Mechanism

LDoS filter examines whether a sender sends more packets during a time slot where RED is undergoing packet drops. Such flows are categorized as attackers and their packets will be dropped before they reach RED dropping mechanism. For that each monitored flow is assigned a local drop indicator and it is decremented each time the flow initiates burst of packets. If the flow is sending packets only in normal rate, drop indicator is incremented. Overall if the drop indicator value is positive, flow can be categorized as normal or else the flow is categorized as attack. So if a flow accidently send burst traffic, its drop indicator is decremented, but since the flow has send only normal rate of packets earlier its drop indicator still remains positive. So the flow will not be categorized as attack. In Robust RED, LDoS filter detects an attack solely based on time interval of packet arrival, but in RRED-PD we have added the rate of packet arrival also as a parameter for LDoS detection to eliminate false positives.

For an LDoS attacking flow, each time it is sending rate of packets greater than link capacity and hence each time its drop probability reduces by 1 and overall it will be having negative drop probability, so the packets will be filtered out. Therefore the LDoS filter will analyze the rate of packet send at each time slot and the continuous evaluation over long interval will help to correctly identify and prevent LDoS attacks.

Based on the architecture description, the steps done to monitor high bandwidth flows and LDoS attack prevention can be summarized as follows:

Step 1: Flow Monitor

3.1.1 Compare flow's bandwidth with Target Bandwidth of network.

3.1.2 *If the bandwidth exceeds target bandwidth the flow is monitored and goto step2 else goto step4*

Step 2: Pre-Filter

3.1.1 Dropping probability of a flow is assigned proportional to that flow's bandwidth consumption.

3.1.2 *Packets are dropped only at times of congestion*

Step 3: LDoS Filter

1) *For each time slot check the rate of packet sending.*
2) *If the rate exceeds target at times of congestion, the local drop indicator associated with that flow is decremented else incremented.*

3.1.3 *If the drop probability is negative, filter out entire packets from that flow*

Step 4: RED dropping

Step 5: If RED drops a packet from a flow that is not monitored, identify them and monitor them later on.

3.2 Algorithm for LDoS attack Detection and Prevention with max-min fairness of bandwidth

Input: Packets from different flow, pkt
Output: LDOS attack detected and filtered

Maximum Rate Threshold of network,
η_G *= bandwidth * delay*

Initialize Monitored_Flows[] with flow id's having bandwidth greater than target bandwidth.

RREDPD – ENQUE(pkt)

1: f ← FID(pkt)
2: if (f in Monitored_Flows[])
*3: Rate of flow, η_L = bandwidth(f) * delay(f)*
4: T_{max}← MAXIMUM(Flow[f].T1, T2)
5: if pkt.arrivaltime > Tmax and

pkt.arrivaltime < (T_{max} + T) or η_L > η_G then*
6: f.Indicator --
7: else
8: f.Indicator++
9: end if
10: if Flow[f].Indicator >= 0 then
11: REDPD-UPDATE-PROBABILITY(Flow[f])
12: RED–ENQUE(pkt)
13: if RED drops pkt then
14: T2← pkt.arrivaltime
15: Add f to Monitored_Flows[]
16: end if
17: else
18: REDPD-INCREASE-PROBABILITY(Flow[f])
19: Flow[f].T1←pkt.arrivaltime
20: drop(pkt)
21: end if
22: else // flow is not monitored
23 : RED-ENQUE(pkt)
24: end if
25: return

RRED-PD algorithm accepts packets from different flows. An array *Monitored_Flows*[] is initialized with flow id's which consume their bandwidth greater than target bandwidth. The target bandwidth is defined as the bandwidth obtained by a *reference TCP flow* with the *target round-trip time, RTT* and the drop rate, *DR* at the output queue. Target bandwidth *Target(RTT,DR)* is given by the TCP response function [39]

$$Target\ (RTT, DR) =\sim \frac{\sqrt{1.5}}{RTT.\sqrt{DR}} \text{packets/second}$$

If the flow is to be monitored, then check for LDoS attack. Packets from a flow is categorized as LDoS attack, if the packets come within an interval of (T_{max},T_{max}+ T*)[36] or the rate of packet sent is greater than the maximum rate threshold of network link capacity. T* is chosen to be 10ms, which is proven to be optimal for detecting LDoS attack under different time intervals. If the packet is from an attacking sender, decrement the flow indicator value by one else increment it by one. If flow indicator is becoming negative, probability of packet drop for that flow is increased and packets from that flow is dropped else the probability of flow is updated so as to provide fairness in bandwidth. Refer [31] to obtain details about drop probability update method. After that it will pass through normal RED-Enque process. If the packet is dropped after RED-Enqueu then that flow is added to *Monitored_Flows*[], since they were not classified into monitored flows initially.

3.3 Complexity Analysis

To compare the relative complexity of RED, RRED, RED-PD and RRED-PD algorithms, we should consider the overhead in computations associated with each. Except RED, every other algorithm should analyze the header of the packet to classify them as attacking or non attacking packets. In that aspect, overhead of RED will be less, but it is not able to detect and prevent LDoS attack. RRED uses per-flow analysis to detect an LDoS attack. Therefore space complexity will be O(N), where N is the total number of flows. Time required for computation will be greater than RED because there is an additional LDoS filter to prevent attacking packets. RED-PD and RRED-PD will be having only O(n) space complexity, where n is the total number flows which consumes high bandwidth which is very much less than N (where N = total number of flows). Time needed for computation will be higher in RRED-PD compared to RED-PD since it has mechanism to filter out LDoS packets. Since hashing is used in our method to monitor high bandwidth flows, computational complexity and run-time execution time is much improved.

4. Performance Evaluation

The proposed scheme has been successfully simulated and tested under various LDoS attack using NS2 simulation platform.

4.1 Simulation Process

A network is created with link capacity 10Mbps, queue limit 50 bytes and 6 senders initially out of which 2 are normal senders which send TCP packets at the rate of 1Mbps, the next 2 are senders which consume high bandwidth and sends at the rate of 10Mbps, and the remaining senders are LDoS attacking flows which sends UDP packets at the rate of 5Mbps but at 200ms delay which is capable of creating a congestion in network whenever initiated. Attacking packets are set as UDP, because the attackers do not wait for any acknowledgement.

Each normal user (User1 to User4) generates packet size of 1000 bytes on a *Newreno* TCP based FTP flow. Attacker's packet size is set as 50 bytes UDP flow.

RED is set to packet count mode and RRED-PD's T* is set to 10ms. Other AQM parameters are left to be NS2 specific. Simulation period is 100s.

Performance of RRED-PD is compared with AQM schemes like RED, Robust RED and RED-PD.

LDoS attack is simulated by setting *P (period of attack)* = 1s since according to the findings of [9] LDoS attacks is prominent with *P = 1s.* W(attack burst width) is set to 100ms and R(rate of attack) is set as 5Mbps so that the aggregate of 2 attackers is equal to the bottleneck bandwidth of the network (10Mbps). With the same signature set for LDoS attack, three scenarios (by varying different parameters) are tested under RRED-PD and other AQM schemes to ensure the robustness of the implemented technique.

TABLE II

EXPERIMENTAL PARAMETERS

Scenarios	P (s)	W (ms)	R (Mbps)
Scenario1	[.5-5]	100	5
Scenario2	1	[0-1000]	5
Scenario3	1	100	[1-10]

Range of values chosen for P, W and R under three scenarios are provided in Table II.

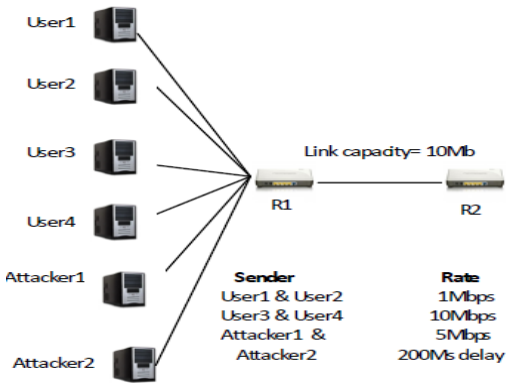


Fig 3: Simulation Scenario

4.2 Result Analysis

4.2.1 **LDoS Attack Detection:** Above mentioned network has been simulated using NS2 and the nam file obtained demonstrates how the LDoS packets are correctly identified and dropped. Here senders 0 and 1 are normal senders, 2 and 3 are high bandwidth consuming senders (colored green) and 4 and 5 indicate attacking senders and are colored red. After passing through LDoS filter, red packets (attack) are dropped, but at the same time blue and green packets pass through network.

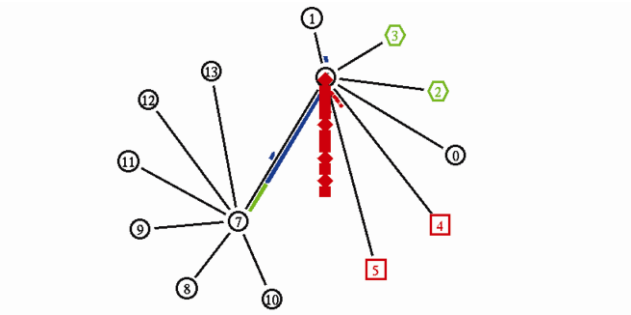


Fig 4: NAM File indicating LDoS packet dropping.

Attacking Packet Drop Ratio for RRED-PD is very high compared to others, where

$$\text{Attacking Packet Drop Ratio, } \gamma = \frac{\text{Total Drop of Attacking Packets}}{\text{Total Packet Drop}}$$

Table II includes the Attacking Packet Drop Ratio associated with each queue type. Here RRED-PD drops 77% of attack packets.

TABLE II

Comparison of *Attacking PacketDrop Ratio,* γ

Queue	γ (scenario1)	γ (scenario2)	γ (scenario3)
RED	.05	.04	.02
RED-PD	.02	.02	.06
RRED	.70	.68	.63
RRED-PD	.77	.73	.76

4.2.2 **TCP Throughput***:* If LDoS attack is not present in a network, then RRED-PD maintains a throughput as equivalent to RED and RED-PD. This is illustrated with figure 5[a] where TCP Throughput is defined as :

$$\text{TCP Throughput} = \frac{\text{Number of bytes correctly received}}{\text{Simulation Duration}}$$

Figure 5[b-d] illustrates the TCP throughput obtained at 3 different scenarios. In each one of this, RRED-PD is able to obtain highest TCP throughput almost as equivalent to link capacity. It is clear from the illustration that, performance of RED and RED-PD under LDoS attack is very poor. Throughput of RED and RED-PD is least when the period of attack is varied from .5 to 5 seconds.

4.2.3 **Fairness in Bandwidth Allocation:** Figure 5a illustrates the bandwidth utilized by each of the flow under no LDoS attack scenario and Figure 5[b-d] demonstrates bandwidth allocation under LDoS attack at 3 different scenarios. If we analyze the bandwidth utilized at different time slots, RED-PD and RRED-PD maintains an average rate of bandwidth utilization and hence obtaining max-min fairness in bandwidth consumption between the flows. Whereas RED and RRED shows a wide variation in bandwidth utilization between the flows since RED allows burst traffic to consume more bandwidth and RRED accidently drops burst traffic misunderstanding them as LDoS packets. Bandwidth allocation of RED and RRED is widely varied when period of attack varies from .5 to 5 seconds.

4.2.4 Average Queuing Delay of packets*:* Average queuing delay remains same for all types of queues compared, but the attacking packets remain in the queue for a very short period of time in the case of RREDPD because the LDoS packets are fast filtered. Since such packets produce congestion in the network, delay is much greater in RED and RED-PD.

$$\text{Average Queuing Delay} = \frac{\sum(\text{Start Time} - \text{End Time of each packet})}{\text{Total no of packets in that flow}}$$

Table III includes the average queuing delay for each type of flow, like normal flow, high bandwidth flow and LDoS attack flow. Normal flows and high bandwidth consuming flows suffer almost similar queuing delay in all AQMs. Since LDoS attack flows are abruptly deleted incase of RRED-PD it has the lowest queuing delay compared to other AQMs.

TABLE III

Comparison of Average Queuing Delay

Queue Vs Average Queuing Delay	**Normal Flows (seconds)**	**High bandwidth Consuming Flows (seconds)**	**LDoS Attack Flow (seconds)**
RED	0.00313	0.00308	0.3205
RED-PD	0.00312	0.00300	0.2507

RRED	0.00309	0.00301	0.0507
RRED-PD	0.00318	0.00310	0.0424

4.2.5. **Packet Delivery Ratio of LDoS Packets:** Table IV gives the Packet Delivery Ratio and Attack Packet Delivery ratio of different AQMs. Number of attacking packets detected in RRED-PD is very high. Packet delivery ratio of attacking packets in RRED-PD is very low compared to others. Hence the overhead in sending attack packets can be reduced. RRED also have low attack packet delivery ratio. But compared to RRED-PD, its packet delivery ratio is also very less, means a good amount of normal packets are also getting discarded in RRED.

Different parameters in table are calculated using the formula:

$$\text{Packet Delivery Ratio} = \frac{\text{Total Number of packets received}}{\text{Total Number of packets sent}}$$

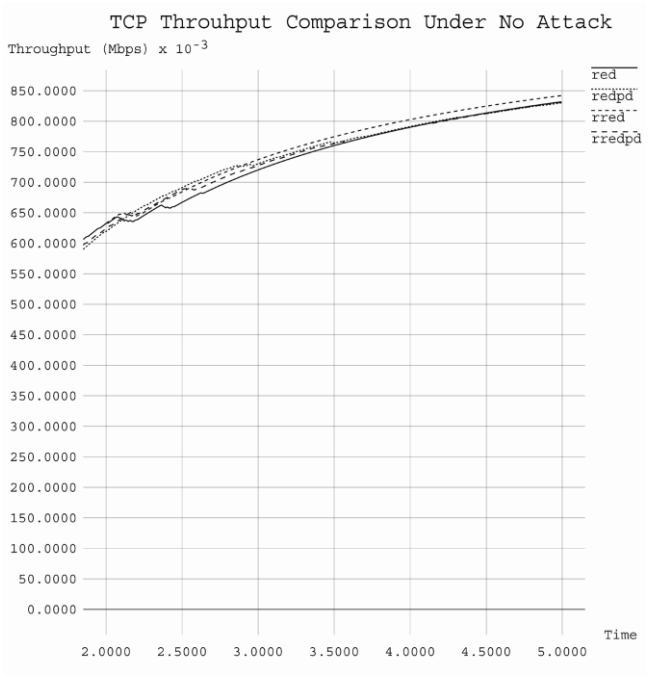
$$\text{Packet Delivery Ratio} = \frac{\text{Total Number of attack}}{\text{Total Number of attack packets sent}}$$

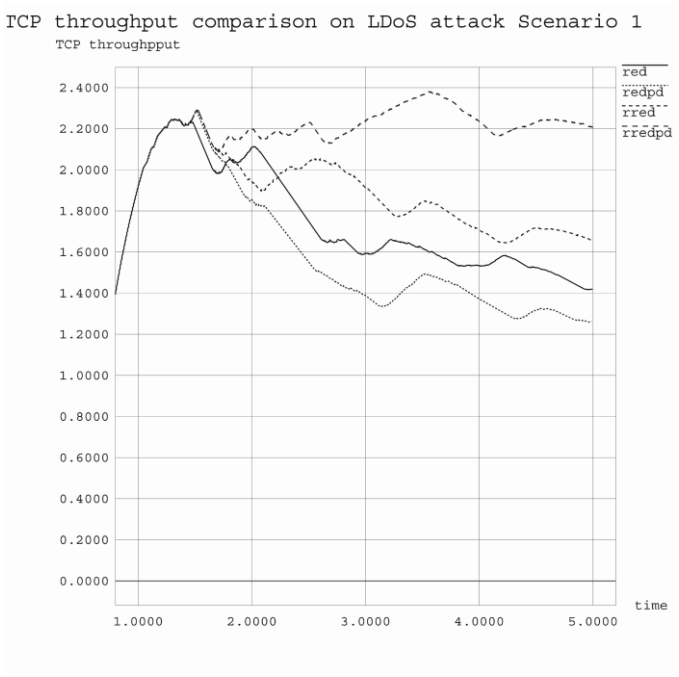
Where, Total Number of Attack Packets received = Total Number of Attack Packets sent – Total Attack Packets discarded.

TABLE IV

Comparison of Packet Delivery Ratio

Properties Vs Queue	RED	RED-PD	RRED	RRED-PD
Total Packets send	1483	1392	1604	1424
Total Packets Received	1258	1248	711	1053
Total Packets Discarded	158	105	866	1110
Total Attacking Packets	1072	1072	1072	1072
Number of Attacking Packets Discarded	48	11	741	825
Packet Delivery Ratio	.848	.8965	0.443	0.739
Packet Delivery Ratio of Attack Packets	.955	0.989	0.308	0.230





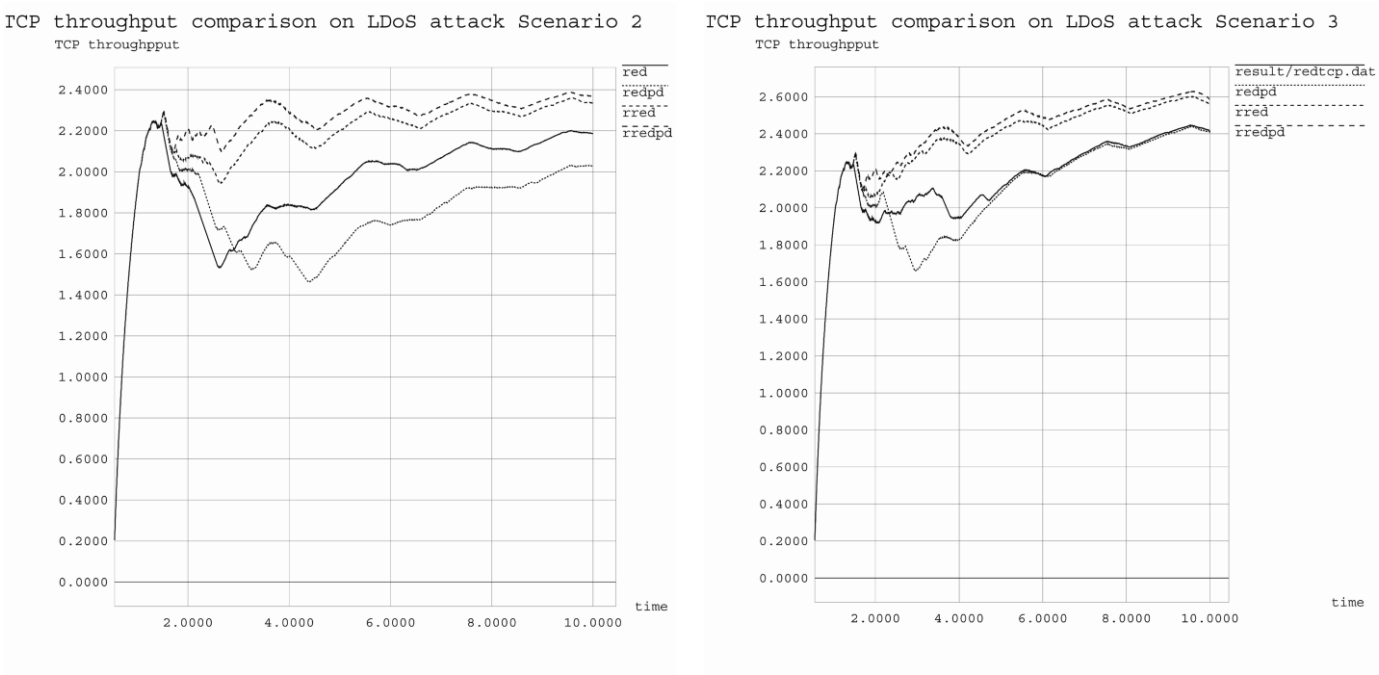
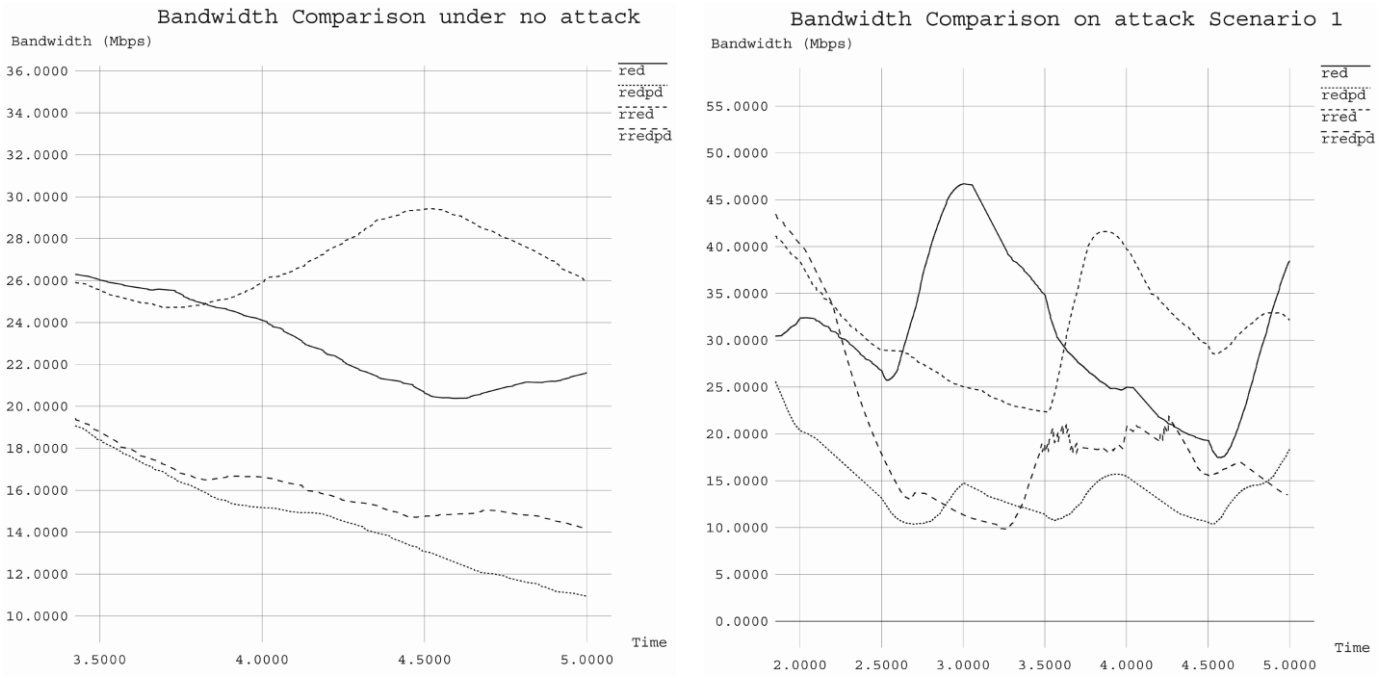


Figure 5. Throughput comparison under no attack and LDoS attack under 3 different scenarios



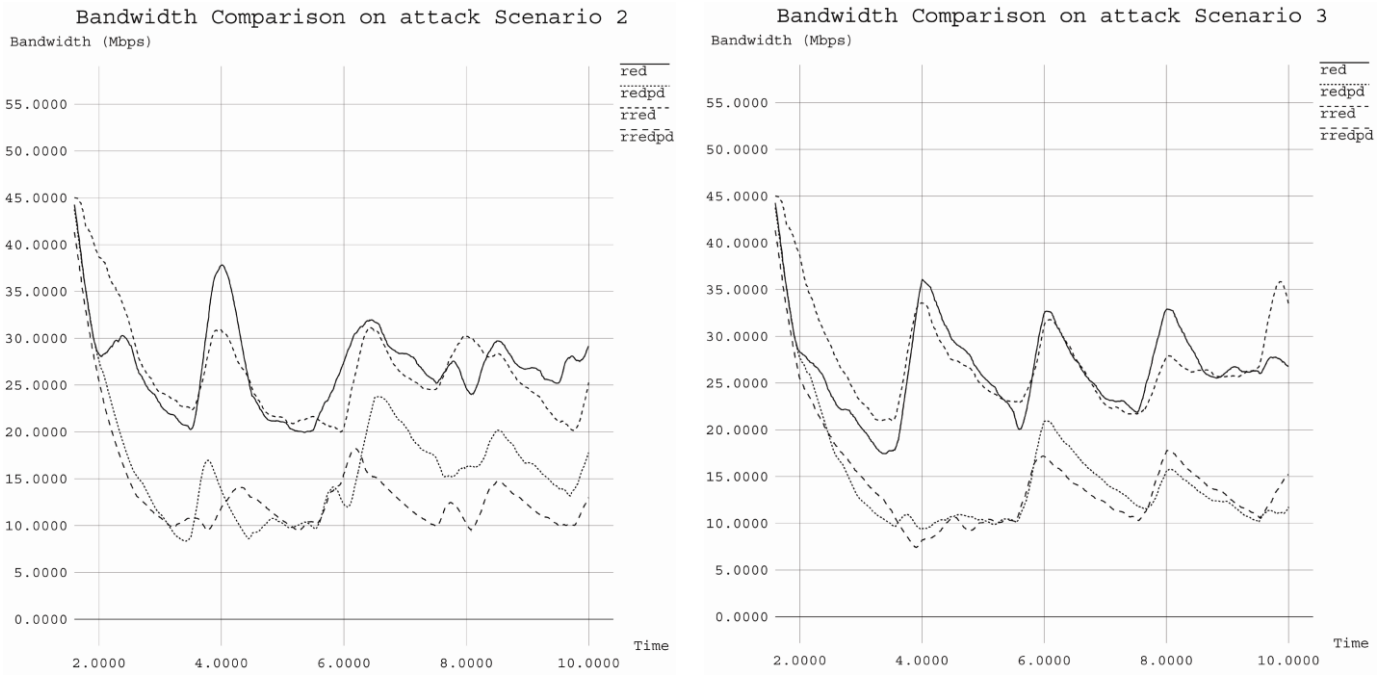


Fig 6. Bandwidth Comparison under no Attack and LDoS attack under 3 different Scenarios

5. Conclusion and Future Work

Low Rate Denial of Service Attack is found to be causing severe threat to TCP's performance. In order to prevent the effect of this attack, we modified RED-PD with a filter to detect and prevent LDoS packets. RED-PD identifies the high bandwidth consuming flows and monitors them. From that monitored flows, filter detects LDoS packets using a continuous evaluation strategy based on the rate of packets sent at times of dropping periods of RED. Since RED-PD is used, a max-min fairness of bandwidth is obtained among different flows, and only the high bandwidth consuming flows need to be monitored thus enabling LDoS attack detection possible with only partial flow analysis. Simulation results using NS2 shows that under no attack conditions, the throughput and average queuing delay of RRED-PD are same as that of RED-PD. At times of attack our system is better if we compare the average queuing delay and bandwidth consumption in network.

RRED-PD uses static value for T*. But by applying soft computing methods, T* can be dynamically changed to suit different kinds of network and attacking scenarios. Also performance of RRED-PD under UDP packet flows are not analyzed since LDoS attack tries to exploit TCP's retransmission time out mechanism.

6. REFERENCES

[1] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks and counter strategies," IEEE/ACM Trans. Netw., vol. 14,no. 4, pp. 683–696, 2006.

[2] Zenghui Liu, Liguo Gua, "Attack simulation and signature extraction of low-rate DoS." 3rd International Symposium on Intelligent Information Technology and Security Informatics IEEE 2010 Computer Society (2010)

[3] Sandeep Sarat and Andreas Terz, "On the Effect of Router Buffer Sizes on Low-Rate Denial of Service Attacks", IEEE Computer Society (2005)

[4] Jing Zhang, Bo Liu, Huaping Hu, Lin Chen, "Simulation and Analysis of LDoS Attacks", International Conference on Multimedia Information Networking and Security (MINES), 2010.

[5] Karn, Phil; Craig Partridge (11-13 August). "Improving Round-Trip Time Estimates in Reliable Transport Protocols" (PS). *ACM SIGCOMM '87*. pp. 2–7. http://www.ka9q.net/papers/rtt.ps.gz.

[6] Ratul Mahajan and Sally Floyd AT&T Center for Internet Research at ICSc I (ACIRI)," Controlling High Bandwidth Flows at the Congested Router", In Proceedings of IEEE ICNP 2001, Riverside,CA, Nov. 2001.

[7] S. Floyd and V. Jacobson,"Random Early Detection gateways for congestion avoidance," IEEE/ACM Trans. Netw., vol. 1, no. 4, pp.397–413,1993

[8] Changwang Zhang, Jianping Yin, Zhiping Cai, and Weifeng Chen, "RRED: Robust RED Algorithm to Counter Low-Rate Denial-of-Service Attacks", IEEE COMMUNICATIONS LETTERS, VOL. 14, NO. 5, MAY 2010.

[9] Chia-Wei Chang, Seungjoon Lee, Bill Lin and Jia Wang, "The Taming of The Shrew: Mitigating Low-Rate TCP-Targeted Attack", AT&T Labs-Research, Florham Park,, 2011.