# RED-FT: A Scalable Random Early Detection Scheme with Flow Trust against DoS Attacks

Xianliang Jiang, Jiangang Yang, Guang Jin, and Wei Wei

***Abstract*—In traditional Active Queue Management algorithms, e.g. RED, each flow, defined with the source and destination IP address of packets, fairly contends for the cache of bottleneck queues. However a malicious flow cannot be identified. And it enables potential network-layer attacks, e.g. the flooding Denial-of-Service (DoS) attack and the low-rate DoS attack. In this letter, we propose a new scheme using the flows trust values to defend against DoS attacks. Different from previous schemes, it employs the flow trust to safeguard legitimate flows. A router monitors network flows and calculates flows trust values, which are used for the relevant queue management. Malicious flows would be with lower trust values while legitimate flows would be with higher ones. Simulation results show that the scheme improves the throughput and delay in DoS attacking scenarios comparing with existing queue management algorithms. We consider the scheme is practical and effective to secure networks.**

***Index Terms*—Internet security, DoS attacks, RED, flow, trust.**

## I. Introduction

Currently, the routers' queue management strategies are divided into two categories, Passive Queue Management (PQM) and Active Queue Management (AQM). PQM drops the new arriving packets or packets in the head of the queue when the queue is completely filled. It causes the serious problem of *TCP global synchronous*, which greatly affects the efficiency of networks communication. Different from PQM, AQM emphasizes the initiative of packets dropping and the length of the current queue. When the current queue length meets the threshold defined in advance, the router will drop packets probabilistically or deterministically and maintain the stability of the queue length.

In the past two decades, AQM has been a hot topic in the congestion control. And many researchers focused on the Random Early Detection (RED) algorithm and its variants, i.e. the queue management based on the control theory, the optimization theory, the bionics and the game theory, etc. Considering the severity of Denial-of-Service (DoS) attacks in the network-layer, unlike most precious RED-like schemes, this letter mainly makes up for the insufficiency of the RED suffering from DoS attacks and render it more robust. In this letter, the flow is a virtual channel connecting the source and the destination and its identifier is a hash value defined with the source and the destination IP address of packets.

Floyd et al. [1] proposed the RED algorithm firstly. The RED calculates the dropping probability of packets according to the average queue length and acquires a better queue stability. Furthermore, the adaptive RED [2] improved the weight controlling. Similar to the RED, a Blue algorithm [3] used the packets loss events and the link idle events to handle the link congestion. Recently, Kim et al. [4] proposed a queue management algorithm according to the weighted fairness of flows in wireless networks.

Above analyses show that the existing AQM algorithms sustain the queue stability and could be adapted to normal networks environments without attacks. In management strategies, the mark and probability mechanisms are used to drop packets. All these features make AQM algorithms acquiring better queue stabilities, higher data throughput and lower data transmission delay. However, most AQM algorithms are unsuitable for networks environments under DoS attacks because the impact of attacks were not considered. Recently, a Robust RED scheme [5] can effectively improve the performance of the RED under Low-rate DoS (LDoS) attacks. Inspired by the scheme, this letter presents a RED with Flow Trust (RED-FT) using networks flow characteristics to ensure the legitimate users' communications and the fairness of the queue as much as possible. In other words, we introduce the networks flow trust as an important decision-making factor of AQM and improve the robustness of previous algorithms, e.g. the RED, in complex networks environment under DoS attacks.

## II. Network-Layer Attack and Trust Mechanism

### A. Network-layer Attack

The goals of network-layer DoS attacks include the consumption of networks resources, the destruction of legitimate networks communications, etc. Tao et al. [6] summarized the categories of DoS attacks and analyzed the defense strategies and challenges. Hereinto the network-layer DoS attack has become one of the most popular attacks and cause serious influences to the networks transmission efficiency. Typical network-layer attacks mainly consist of Flooding DoS (FDoS) [7] attack and LDoS [8] attack, etc. FDoS (Fig. 1. (a)) refers to the attacker sending high rate malicious packets to flood and occupy most of the bottleneck queue. And most of legitimate packets will be dropped due to long delay. LDoS (Fig. 1. (b)) mainly targets TCP connections. The attacker periodically

Manuscript received November 26, 2012. The associate editor coordinating the review of this letter and approving it for publication was H. Luo.

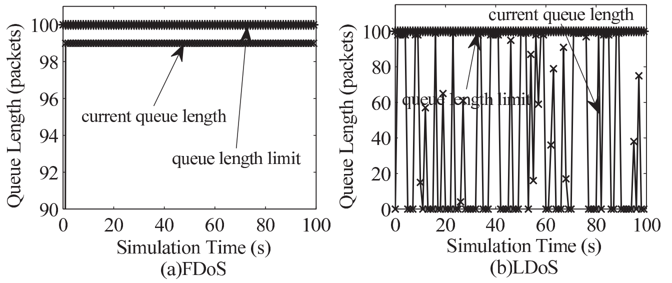
This work was supported in part by the Natural Science Foundation of Zhejiang Province (LY12F02013) of China; Ningbo Natural Science Foundation (2012A610014) of China; Zhejiang Provincial Technology Innovation Team (2010R50009-19/20) of China; Ningbo Municipal Technology Innovation Team (2011B81002) of China; The Graduate Teaching Innovation of Ningbo University(2011004); Key Science and Technology Research Project of Education Department Henan Province (13A413001).

X. Jiang and J. Yang are with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China (e-mail: 11221058@zju.edu.cn).

G. Jin is with the College of Information Science and Engineering, Ningbo University, Ningbo, China (e-mail: jinguang@nbu.edu.cn).

W. Wei is with the Henan University of Technology, Zhengzhou, China.

Digital Object Identifier 10.1109/LCOMM.2013.022713.122652



(a)FDoS (b)LDoS

Fig. 1. FDoS attack flow and LDoS attack flow.

sends burst flows to disconnect TCP connections and reduce its throughput in the long run. This letter only presents the countermeasures against FDoS and LDoS attacks. However, we consider that our scheme can be extended or modified to defend against more other network-layer attacks.

Following the notations of FDoS and LDoS in [7] and [8], we simulate the attacking scenario and analyze the bottleneck queue length. The simulation topology and the parameters of the attacking agent are showed in section IV. And all the queue type is DropTail. The results in Fig. 1 show that FDoS packets always occupy most of the bottleneck queue and LDoS packets gustily fill the entire bottleneck queue.

### B. Trust Mechanism

Recently, the trust is widely studied for network security. It could be used to evaluate the trustworthy among networks entities and improve the security and usability of network services. The typical applications of the trust include the trusted P2P service, the trusted social networks, the trusted routing protocol information, the trusted packets transmission and forwarding, etc. Govindan et al.[9] deeply analyzed the trust computations and trust dynamics in mobile ad hoc networks and highlights comparisons of different approaches. In special applications, the trust is transformed from the evidences of the entities. Equation (1) describes the synthesis of the direct trust and the indirect trust of networks entities. The $T_{ra}$ is the comprehensive trust value. The $T_d$ is the direct trust value and can be calculated by (2). The $T_{id}$ is the indirect trust value and can be calculated by (3). The $\alpha$, $\beta$, $\omega_1$, $\omega_2$, $\ldots$, $\omega_n$, $\eta_1$, $\eta_2$, $\ldots$, $\eta_n$ are the controlling coefficients and weights respectively. The $ed_1$, $ed_2$, $\ldots$, $ed_n$ are the values of direct evidence of entities and the $ei_1$, $ei_2$, $\ldots$, $ei_n$ are the values of indirect evidence of entities. In this letter, we only focus on the values of direct evidence generated by using detection algorithms to monitor the flows' characteristics.

$$T_{ra} = \alpha \cdot T_d + \beta \cdot T_{id} \tag{1}$$

$$T_d = \omega_1 \cdot ed_1 + \omega_2 \cdot ed_2 + \ldots + \omega_n \cdot ed_n \tag{2}$$

$$T_{id} = \eta_1 \cdot ei_1 + \eta_2 \cdot ei_2 + \ldots + \eta_n \cdot ei_n \tag{3}$$

## III. RED WITH FLOW TRUST

### A. RED-FT Steps

Next we will present the structure of the RED-FT scheme, which uses the networks flow trust to identify and drop malicious packets. In previous RED-like algorithms, the stability of the queue was only considered and the networks flows trust was not involved. Thus malicious flows would overflow the routers' queue completely and increase the transmission time of legitimate packets, even make serious packets retransmission. The core steps of the RED-FT are listed as follows.

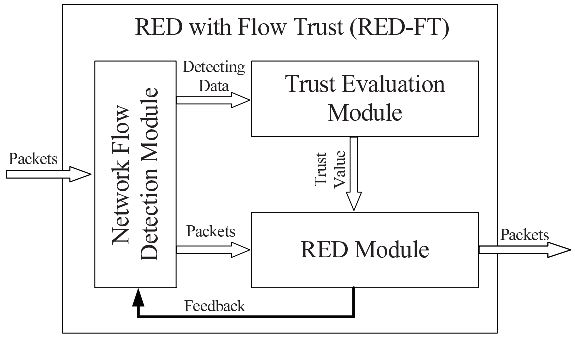


Fig. 2. The framework of RED-FT algorithm.

Firstly, each flow behaviors, such as the data rate and packets loss ratio in routers, are monitored. Secondly, the collecting data should be quantified and the trust values should be calculated. Then different trust values are normalized and fused. Furthermore, three strategies are selected according to the networks flow trust: (*a*) a network flow is trusted and the RED operation is carried out. (*b*) a network flow is distrusted and packets are dropped. (*c*) a network flow is probabilistically trusted and the RED operation is probabilistically carried out. Finally, the execution result of the RED operation is returned to the monitoring module.

### B. Implementation of RED-FT

Fig.2 illustrates the framework of RED-FT. When a router receives packets, it firstly transmits them to the *Network Flow Detection Module* (NFDM) where the flows size and the packets loss ratio etc. are measured. And then the NFDM sends the detected data to the *Trust Evaluation Module* (TEM). Meanwhile, the NFDM also needs to send received packets (as presented in Fig. 2) to the *RED Module* (REDM). The TEM computes and evaluates the flow trust according to the collected data and sends the trust value to the REDM. And the REDM executes standard RED operations according to the current queue length and the trust value.

Here we only pay attention to FDoS and LDoS attacks. So any effective detection methods on FDoS and LDoS could be deployed in the NFDM. The main characteristic of FDoS is the high traffic. According to the characteristic, a sampling timer is adopted every specific time interval, such as [*td, td*+*t**], to sample flows and compute its bandwidth ratio. After obtaining the ratio of a specific flow, we can use them as the evidence of the flows trust evaluation. The main characteristic of LDoS flow is close to a periodic burst. According to the characteristic, a flow *f* is suspected to be malicious if its packet arrives within a short-range [5].

In the trust computing stage, the TEM mainly uses the collected data of the NFDM to compute different direct trust values in (2). For FDoS, it should compare the bandwidth ratio with a threshold. If the bandwidth ratio is less than the threshold, the *fc* (anomaly counter) indicator should be increased. Otherwise, the *fc* indicator should be decreased. On the basis of the *fc* indicator, the flow trust can be calculated by (4) and the *fcp* and *fcf* are the two thresholds of the *fc* indicator. The *pf* (the probability trust value of FDoS flows) in (4) can be calculated by (5). For LDoS, the packet dropping interval should be detected and estimated whether within a short-range, such as [*t, t+t**], or not. If the interval is smaller than the short-range, the *lc* (the anomaly counter of the LDoS flow) indicator should be decreased. Otherwise, the *lc* indicator should be increased. On the basis of the *lc* indicator, the flow trust can be calculated by (6) and the *lcp* and *lcf* are the two thresholds of the *lc* indicator. The *pl* (the probability trust value of LDoS flows) in (6) can be calculated by (7).

$$T_{fdos} = \begin{cases} 1 & fc > fcp \\ pf & \text{other} \\ 0 & fc < fcf \end{cases} \tag{4}$$

$$pf = \frac{fc - fcf}{fcp - fcf} \tag{5}$$

$$T_{ldos} = \begin{cases} 1 & lc > lcp \\ pl & \text{other} \\ 0 & lc < lcf \end{cases} \tag{6}$$

$$pl = \frac{lc - lcf}{lcp - lcf} \tag{7}$$

In the trust fusion stage, we only consider the direct trust and calculate the total trust as (1). In order to enhance the accuracy, a *MIN* strategy should be applied to all trust values of a special flow before the calculation. After getting the minimum trust, the final flow trust could be updated as (8). The trust volatilization is calculated in (9). In (8), $\alpha$ is a reward factor ($\leq$ 0.1) and is a punishment factor ($\geq$0.5). The *nt* is the flow trust while *tu* and *td* are the upper bound and lower bound of *nt* respectively. The *tth* is a threshold. In (9), $\gamma$ is a positive controlling coefficient ($\leq$0.1) and *ts* is the time interval range from the last recorded time to the current update time.

$$nt = \begin{cases} nt + nt \cdot \alpha & MIN(T_{fdos}, T_{ldos}) > tth \& nt < tu \\ nt - nt \cdot \beta & MIN(T_{fdos}, T_{ldos}) < tth \& nt > td \end{cases} \tag{8}$$

$$nt = nt - nt \cdot \gamma \cdot ts \tag{9}$$

The packet dropping in the REDM is based on the flow trust of the TEM. It mainly includes three categories. Firstly, if *nt*$\geq$*tp*, it should execute the RED. Secondly, if *nt*$\leq$*tf*, it should directly drop the packet. Thirdly, if *tf*<*nt*<*tp*, it should probabilistically execute the RED. Here the *tp* and *tf* is the belief bound and disbelief bound of the final trust value.

Fig. 3 shows the pseudo codes of the RED-FT, which consists of two parts, the trust computing and the RED operating. A flow identifier is generated from $FLOW\_HASH(pkt)$ according to $pkt$'s source-destination address pair and a router's secret. The $FDoSDetection(pkt)$ function detects the flooding DoS attack according to current *pkt*'s size and historic records. The $LDoSDetection(pkt)$ function detects the low-rate DoS attack according to *t**.

```
RED-FT-ENQUE (pkt)
fh←FLOW_HASH(pkt)
rt←MIN(FDoSDetection(pkt),LDoSDetection( pkt ))
if (rt > thresh_ and old_trust < trust_max_ ) then
    increase the trust value of the fh flow
else if (rt < thresh_ and old_trust > trust_min_) then
    decrease the trust value of the fh flow
end if
trust_cache[fh].new_trust←UPDATE_TRUST( )
if (trust_cache[fh].new_trust > trust_pass_) then
    execute enque(pkt) and RED
else if (trust_cache[fh].new_trust < trust_fail_) then
    execute drop(pkt) and update ldos_cache
else
    randomly drop packets according to trust and RED
end if
return
```

Fig. 3. The pseudo codes of RED-FT.

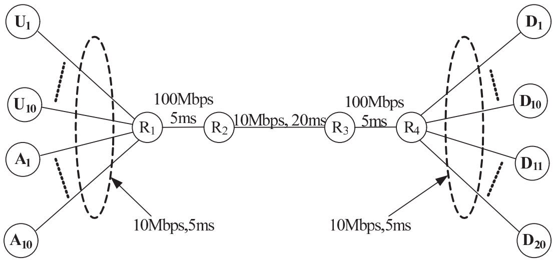


Fig. 4. The topology of simulation experiments.

## IV. Evaluation and Analysis

In order to evaluate and analyze the RED-FT, we use the NS-2 simulator [10] to conduct a set of simulations. Several other AQM algorithms including RED [1], Blue [4], SFB [11] (which is an extension of the Blue and all flows fairly occupy the bottleneck queue), and RRED [5] are used as comparisons. Note that RED-FT is an extensible AQM scheme based on the flow trust, and therefore several more accurate detection mechanisms can be integrated to improve the performance of the RED-FT. Due to the space constraints, only the evaluation on FDoS attacks is provided in detail in this letter.

Fig. 4 shows the simulation topology. The bottleneck link is $R_2$-$R_3$ and its queue size is 100 packets. The evaluated AQM algorithms are implemented on the bottleneck queue, and any other queues use DropTail. A legitimate TCP based FTP flow with packet size of 1000 bytes is generated from $U_1$. HTTP flows with five new connections per second start from $U_2 \sim U_{10}$. And FDoS traffic is generated from $A_1 \sim A_{10}$ by sending UDP packets with packet size of 200 bytes. All flows' destination are randomly selected among $D_1 \sim D_{20}$.

The simulation parameters are listed in Table I. The related trust values are chosen empirically and the false positive of our scheme is very low by using these configurations. Our future work will design a mechanism to determine and analyze the related trust values. The other parameters of the AQM algorithms are all NS-2 default values.

Fig. 5 shows the simulation result of the RED-FT and other selected AQM algorithms on average delay and attacking rate. The result shows that the average delay of the RED-FT is

TABLE I
THE SETTING OF SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Simulation Time | 100s |
| Attacking Rate | 10-50Mbps |
| Attacking Time | 60s |
| Init Trust | 0.5 |
| Min Trust | 0 |
| Max Trust | 1 |
| Pass Trust | 0.8 |
| Fail Trust | 0.2 |

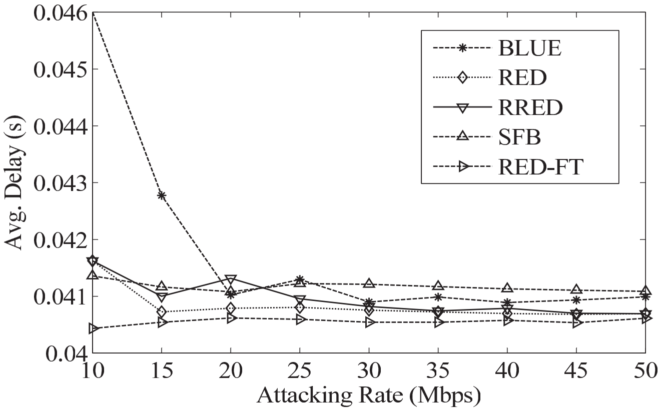


Fig. 5. The average delay under different attacking rates.

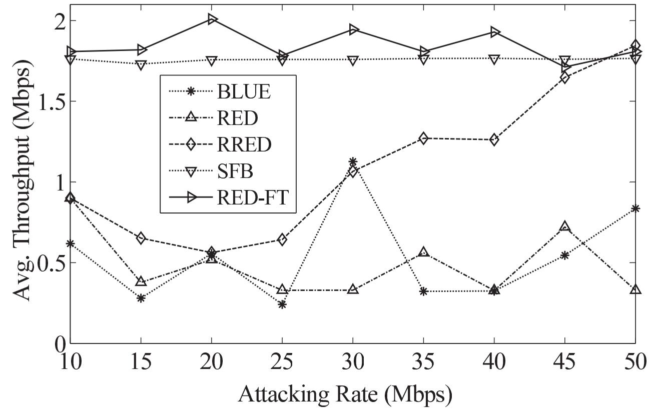


Fig. 6. The average throughput under different attacking rates.

stable and keeps in 0.04 second or so when the attacking rate varies from 10 to 50Mbps. Our scheme performs slightly better than other selected AQM algorithms.

Fig. 6 shows the simulation result on average throughput and attacking rate. The result shows that the average throughput of the RED-FT is stable and keeps in 1.8Mbps or so when the attacking rate varies from 10 to 50Mbps. The proposed scheme also outperforms most other selected AQM algorithms.

We also evaluate the RED-FT in a LDoS attacking scenario and use the *Exponential* procedure in NS-2 to generate LDoS attacking flows. The size of each malicious packet is 200 bytes and the burst interval is 300ms. A legitimate TCP based FTP flow with packet size of 1000 bytes is generated from $U_1$. The simulation parameters are listed in Table I and other parameters of the AQM algorithms are all NS-2 default values. Simulation results also show that the RED-FT is superior to other RED-like algorithms in delay and throughput.

Moreover, the deployment of the RED-FT is incremental. It can be firstly implemented in partial routers and then extended to the others. The resource demand of the RED-FT is relatively limited. For instance, the router only needs to store a few of flow trust information, two structured variables and a trust variable. In our simulations, each structured variable is defined with 8 bytes and each trust variable is defined with 4 bytes. Supposing there are 10000 flows, the demand cache is 0.2MB or so. Furthermore, we consider that the periodical cleaning strategy can help to reduce the resource requirements.

## V. CONCLUSION

In this letter, we have proposed and implemented the RED-FT, a scalable AQM scheme, based on flow trust. It could be used to counter FDoS, LDoS and some other network-layer attacks. The simulations and analyses show that RED-FT performs well in the FDoS and LDoS attacking scenarios. We consider it owns good scalability. It can be extended to combine with other flows detection algorithms to defend against more DoS attacks. Next, potential work on the RED-FT includes the analysis and propagation of flows trust in networks, the enhancement of the detection accuracy and the defense of more network-layer attacks (such as spoofing), etc.

## REFERENCES

[1] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[2] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: an algorithm for increasing the robustness of RED's active queue management," Aug. 2001. Available: http://www.icir.org/floyd/papers/adaptiveRed.pdf.

[3] W. Feng, D. Kandlur, D. Saha, *et al.*, "Blue: a new class of active queue management algorithms," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 513–528, 2002.

[4] D. Kim, J. Kim, H. Yoon, *et al.*, "AQM for weighted fairness in wireless LANs," *IEEE Commun. Lett.*, vol. 15, no. 11, pp. 1199–1201, 2011.

[5] C. Zhang, J. Yin, Z. Cai, *et al.*, "RRED: robust RED algorithm to counter low-rate denial-of-service attacks," *IEEE Commun. Lett.*, vol. 14, no. 5, pp. 489–491, 2010.

[6] P. Tao, L. Christopher, and R. Kotagiri, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Computing Surveys*, vol. 39, no. 1, pp. 1–42, 2007.

[7] R. K. C. Chang, "Defending against flooding-based distributed denial-of-service attacks: a tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42–51, 2002.

[8] 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, 2005.

[9] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile ad hoc networks: a survey," *IEEE Commun. Surveys and Tutorials*, vol. 14, no. 2, pp. 279–298, 2012.

[10] The Network Simulator version 2.
Available: http://www.isi.edu/nsnam/ns/, 2012.

[11] W. Feng, D. D. Kandlur, D. Saha, *et al.*, "Stochastic fair blue: a queue management algorithm for enforcing fairness," in *Proc. 2001 IEEE Infocom.*