AQM&DoS Simulation Platform

This Active Queue Management and Denial-of-Service (AQM&DoS) Simulation Platform was established for the Robust Random Early Detection (RRED) algorithm [1]. If you use any part of this platform in your research, you have the responsibility to cite this platform as:

> The experiments (or simulations) are conducted on the AQM&DoS Simulation Platform that was created for the Robust Random Early Detection (RRED) algorithm [1].
>
> 1. 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, pp. 489-491, 2010.

Platform Homepage: http://sites.google.com/site/cwzhangres/home/posts/aqmdossimulationplatform

Table of Contents

1. The function of the platform

The Active Queue Management and Denial-of-Service (AQM&DoS) Simulation Platform is based on the Network Simulation 2. It is able to simulate a variety of experimental scenarios related to Distributed Denial-of-Service (DDoS) attacks and Active Queue Management (AQM) algorithms.

The platform can simulate numbers of Denial-of-Service attacks:

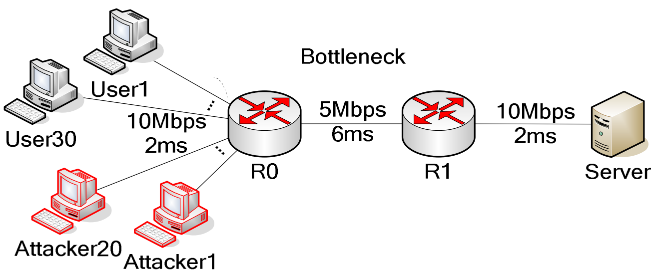
- Denial-of-Service (DoS) attacks
- Distributed Deinal-of-Service (DDoS) attacks
- Spoofing DoS or DDoS attacks
- Low-rate Denial-of-Service (LDoS) attacks
- Distributed Low-rate Denial-of-Service (DLDoS) attacks
- Spoofing LDoS or DLDoS attacks

And a variety of Active Queue Management (AQM) algorithms:

- 1 DropTail; 2 RED; 3 RED/PD; 4 Blue; 5 SFB;
- 6 CBQ; 7 FQ; 8 SFQ; 9 DRR; 10 PI;
- 11 Vq; 12 REM; 13 GK; 14 SRR;
- 15 RED/Robust (RRED) 16 SFB/Robust (RSFB);

To analyse the impact of DoS attacks on normal TCP flows and AQM algorithms, this platform also provides mechanisms to automatically calculate and record the average throughput of normal TCP flows before and after DoS attacks.

The experimental network has a dumbbell topology as the network experimented in the RRED algorithm [1].



2. The installation of the platform

AQM&DoS Simulation Platform is mainly tested on ns-2.33, but it is expected to be compatible with higher versions of ns. If you are using a different version of ns, please replace "2.33" with the version number of your ns in all the following instructions.

To experiment on the AQM&DoS Simulation Platform, you should fellow these steps:

1. Unzip the package of the AQM&DoS Simulation Platform in your Linux system (the subdirectory "result" is necesary to output the simulation result, you should keep it) and run the following command in the directory "aqm-dos-sim-plat".

chmod +x leodos.sh

2. Install the ns-allinone-2.33 simulation software in your operation system (Debian 4.0 Linux is recommend).

NS-2.33: http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.33/ns-allinone-2.33.tar.gz/download

Note1: AQM&DoS Simulation Platform is tested on ns-2.33, but it is expected to be compatible with higher versions of ns.

Note2: AWK is also required to run the platform. But most users do not need to manually install it as it is already included in most Linux distributions. If it is not included in your Linux system, you can refer to the following link to install it.

AWK: http://www.gnu.org/software/gawk/

Or if you are using Debian or Ubuntu Linux, you can use the following two commands to install AWK:

1. sudo apt-get install gawk

2. cd /usr/bin/ && sudo ln -s gawk awk

3. Integrate RRED into your NS2 distribution.

Please follow the instruction in "ns2-integration\integration-of-rred.txt"

4. Modify simulation settings in "leodos.sh" to conduct your specified experiments.

You need to modify the parameters in the "leodos.sh" to conduct a variety of simulations.

4.1 The following line of code means to conduct a single simulation using the parameters specified in the head of "leodos.sh":

dosim 0;

4.2 The following line of code means to conduct a batch of simulations on a specified AQM algorithm x:

"task_aqm_ldos x;"

x is the number of the AQM algorithm. The mapping of x to AQM algorithms is:

1 DropTail; 2 RED; 3 RED/PD; 4 Blue; 5 SFB 6 CBQ 7 FQ; 8 SFQ; 9 DRR; 10 PI; 11 Vq; 12 REM; 13 GK; 14 SRR 15 RED/Robust 16 SFB/Robust;

If you want to experiment on a specific AQM algorithm, please remove the # before its line.

The original setting of the "leodos.sh" is to only conduct a single simulation (dosim 0;).

You might need to understand and modify the logic of the function "task_aqm_ldos" to conduct your specified batch of simulations.

5. Run the simulations using the following command in the directory "aqm-dos-sim-plat".

./leodos.sh

The experimental results are located in the sub-directory "result", including:

1. The overall trace file "leodos.tr"

2. The TCP trace file "leodos_tcp.tr"

3. The queue monitor trace file "leodos_queue_monitor.tr"

4. The bottleneck queue trace file "leodos_queue.tr"

5. The nam trace file "leodos.nam" (To get the nam trace file, you need to change the value of "ns_of" from 2 to 3 in "leodosh.sh")

6. The log files "leodos.log" and "leodos_sh.log". "leodos.log" records the parameters of each simulation and its statistical results in a format shown in Section 4. If you run a batch of simulations using "task_aqm_ldos", these log files will be located in a sub-directory named "AQM_x" (x is the number of the AQM algorithm) under "result".

Optional steps:

o1. Integrate the ip spoofing function into your NS2 distribution (Do this step only if you need to simulate spoofing DDoS attacks).

Please follow the instruction in "ns2-integration\integration-of-ip-spoofing.txt"

o2. Integrate SFB/blue into your NS2 distribution (Do this step only if you need to simulate SFB).

Please follow the instruction (README) in "ns2-integration\ns2-blue.tar.gz" - the code and instruction of SFB/blue.

o3. Integrate RSFB (Resilient Stochastic Fair Blue) into your NS2 distribution (Do this step only if you need to simulate RSFB and have finished the step o2).

Please follow the instruction in "ns2-integration\integration-of-rsfb.txt"

3. The parameters of the platform

Name	Description	Unit
bn_bw	Bottleneck bandwidth	Mbps
bn_dl	Bottleneck link delay	ms
bn_qs	Bottleneck queue size	packets
bn_qm	The AQM algorithm used in the bottleneck link. The mapping of bn_qm to AQM algorithms is: 1: DropTail 2: RED 3: RED/PD 4: Blue 5: SFB 6: CBQ 7: FQ 8: SFQ	

	9: DRR 10: PI 11: Vq 12: REM 13: GK 14: SRR 15: RED/Robust (RRED) 16: SFB/Robust (RSFB)	
nt_bw	Network bandwidth (except the bottleneck link).	Mbps
nt_dl	Network delay (except the bottleneck link).	ms
hp_n	It is not used in this version of the platform.	
ur_n	The number of normal users	
ur_cr	The number of new connections per second. It is used to simulate http traffic.	
ur_ps	User flows packages size	Byte
ur_st	The start time of normal user flows	Second
ur_sp	The stop time of normal user flows	Second
ur_rs	Whether normal user flows randomly start in the period between **ur_st** and **ur_sp**. 0: normal user flows will all start at **ur_st** 1: normal user flows will randomly start in the period between **ur_st** and **ur_sp**	
ur_pt	Normal user flows' type. 1: TCP	
ur_app	The application of normal user traffic. 0: FTP; 1: Telnet; 2:PackMimeHTTP; 3:PackMimeHTTP_DelayBox	
ak_n	The number of attackers	
ak_ng	The number of attacker groups that attackers are organised into. Most of the time, you do not need to change the default value of this parameter "1".	
ak_tg	The start time difference between attacker groups. Most of the time, you do not need to change the default value of this parameter "0".	
ak_st	The start time of attack flows	Second
ak_sp	The stop time of attack flows	Second
ak_rs	The option currently only support Low-rate DoS attacks (when **ak_bp** < **ak_ap**). For most of time, you should set this to be 0. Whether attack flows randomly start in a Low-rate DoS attack period between **0** and **ak_ap**.	

	0: attacker flows will all start at **ak_st** 1: attacker flows will randomly start in every attack period between **0** and **ak_ap**.	
ak_ps	Attacker flows' packages size	Byte
ak_ap	Attacker flows' attack period For a DoS or DDoS attack, it should be configured as: **ak_ap** = (**ak_sp - ak_st)*1000** For a LDoS or DLDoS attack, it is normally configured as: **ak_ap** < (**ak_sp - ak_st)*1000**	ms
ak_bp	Attacker flows' burst period. For a DoS or DDoS attack, it should be configured as: **ak_bp** = **ak_ap** = (**ak_sp - ak_st)*1000** For a LDoS or DLDoS attack, it is normally configured as: **ak_bp < ak_ap**	ms
ak_pr	Attacker flows' packages rate	Mbps
ak_tp	The attack type. Most of the time, this value should be 1. 1:represents period attack 2:represents follow tcp cwnd attack	
ak_mw	Not used in this version of the simulation platform	
ak_cp	Not used in this version of the simulation platform	
ak_spf_lv	Whether attackers use spoofing IP address. 0: attackers use real IP addresses 1: attackers use spoofing IP addresses **ak_spf_mn** and **ak_spf_mn** are two integers. The spoofing address range is from **ak_spf_mn** to **ak_spf_mn.**	
ak_spf_mn	The minimum spoofing address.	
ak_spf_mx	The maximum spoofing address.	
tm_fi	The simulation finishing time.	Second
ns_db	Whether output the debug information. 0: do not output debug information 1: output debug information	
ns_of	The output files of the simulation platform. When **ns_of**: >=3 output leodos.nam (used for nsnam to figure the simulation topology) >=2 output leodos.tr leodos_tcp.tr leodos_queue_monitor.tr >=1 output leodos_queue.tr (The data trace of the bottleneck queue. It is the primary	

analysis data source for this simulation platform)

4. The output of the platform

An example of the platform output is:

```
ak_spf_mx      60000
nt_dl     2
ur_sp     240
ak_st     120
ur_cr     100
ur_n      30
li        1
ns_of     1
ak_spf_lv      0
ak_bp     200
ak_pr     0.25
bn_qm     15
ur_st     20
ak_spf_mn      100
ak_ps     50
tm_fi     240
ak_tp     1
ur_app    0
ak_rs     0
ak_ng     20
bn_bw     5
ak_n      20
ur_ps     1000
ur_pt     1
nt_bw     10
ur_rs     0
bn_qs     50
ak_ap     200
hp_n      25
ns_db     0
ak_cp     10
ak_sp     220
ak_tg     0
ak_mw     1
bn_dl     6
bn_qms  RED/Robust
leodos_queue_awk:  dt=0.010000  s_l=0.000000  t_st=0.000000  t_sp=240.000000  ur_n=30  ak_n=20
ur_st=20.000000 ur_sp=240.000000 ak_st=120.000000 ak_sp=220.000000 p_ct=pktcount
rate_f1_normal  600.680000    rate_f1_attack  597.220000    nth_f1  0.994240
rate_f2_attack  17.940000
```

The lines from "ak_spf_mx" to line "bn_qms" are detailed parameters of this simulation (please refer to Section 3 for the meaning of these parameters). The followed lines are statistical results:

- "*rate_f1_normal*" depicts the average throughput rate (packets/s) of normal TCP traffic through the bottleneck link when there is no DoS/LDoS attack.

- "*rate_f1_attack*" depicts the average throughput rate (packets/s) of normal TCP traffic through the bottleneck link when a DoS/LDoS attack is attacking (from *ak_st* to *ak_sp*).

- "*nth_f1*" represents the preserved ratio of normal TCP traffic throughput under a DoS/LDoS attack, which equals to *rate_f1_attack*/*rate_f1_normal*.

- "*rate_f2_attack*" depicts the average throughput rate (packets/s) of attack traffic through the bottleneck link when a DoS/LDoS attack is attacking (from ak_st to ak_sp).

Note1: To convert the unit of a throughput rate from packets/s to Mbps (millions of bits per second or megabits per second), you need to multiply the original value by 8*ur_ps/1024/1024 (for *rate_f1_normal* and *rate_f1_attack*) or 8*ak_ps/1024/1024 (for *rate_f2_attack*). *ur_ps* is user flows' package size and *ak_ps* is the packet size of attack flows. Their units are both Byte (not Bit. 1 Byte = 8 Bits). The original rate value multiplied by 8*packet-size (ur_ps or ak_ps) changes its unit from (packets/s) to (bits/s), then divided by 1024 changes its unit from (bits/s) to Kbps (Kilobits per second), and finally divided by 1024 changes its units from Kbps (Kilobits per second) to Mbps (Megabits per second).

Note2: Mbps (Megabits per second) is different from MB/s that stands for MegaBytes per second. 1 MB/s = 8 Mbps.

References

[1] 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, pp. 489-491, May 2010.