

Design and Implementation of NOBS Test Platform

YU Zhenghao^{1,2}, WANG Gang^{1,2}, DING Wei^{1,2}

1. Department of Compute Science and Engineering, Southeast University, Nanjing 210096, China

2. Jiangsu Provincial Key Laboratory of Computer Network Technology, Southeast University, Nanjing, 210096. China

1.E-mail: zhhyu@njnet.edu.cn, 2.E-mail: gwang@njnet.edu.cn, 3.E-mail: wding@njnet.edu.cn

Abstract—NBOS static test platform support the function that users can customize their scripts to generate specific traffic and mix with background traffic play and test the correctness of NBOS function module. This paper introduced the process of NBOS design and realization and the realization scheme and testing process of key modules.

Keywords- system test; network management; NBOS

NBOS 静态测试平台的设计与实现

余政浩^{1,2}, 王岗^{1,2}, 丁伟^{1,2}

1. 东南大学 计算机科学与工程学院, 江苏 南京, 210096

2. 江苏省计算机网络技术重点实验室, 江苏 南京, 210096

3. 1.E-mail:zhhyu@njnet.edu.cn, 2.E-mail:gwang@njnet.edu.cn, 3.E-mail:wding@njnet.edu.cn

【摘要】 NBOS 静态测试平台支持用户通过定制脚本随机产生特定流量并将其混入背景流量播放的方法, 对 NBOS 功能模块进行正确性测试。文章详细介绍了该测试平台的设计和实现过程, 以及关键模块的实现方案和测试过程。

【关键字】 系统测试; 网络管理; NBOS

1 引言

NBOS (Network Behavior Observation System) [1]是基于路由器设备提供的 Netflow 数据监控和管理 CERNET2 网络服务质量和网络安全状态的新型管理系统。作为国家科技支撑计划课题“新一代可信互联网安全和网络服务”(2008BAH37B04)的组成部分, 该系统在于设计和实现对网络流量的分析、异常发现和应急控制, 限制网络异常流量对正常网络服务的影响, 为用户提供网络服务质量监测和评估功能。NBOS 系统的开发环境位于东南大学的 CERNET 华东北地区网络中心, 基于 JSERNET 和 CERNET 主干网互连节点的流量数据实现对网络服务质量和安全状态的监测, 该系统的运行环境将位于 CERNET/CERNET2 的国家网络中心, 以及东南大学、北京邮电大学、西安交通大学、重庆大学和复旦大学主节点, 并实现这些系统之间的管理数据共享和协同工作。

NBOS 系统以五分钟 Netflow[2]数据作为一个处理单位, 在将其处理成中间数据格式 NBOS 流记录(格式见附录 1)后, 包括延迟、丢包、异常分析、热点分析、聚类分析和公平性分析在内的十余个功能进程并发完成各自的计算, 其结果体现系

统的功能。

为了能够对 NBOS 各个功能模块进行正确性测试和压力测试, 我们设计并实现了一个静态测试平台, 该平台的特点包括:

- I 采用实际流量作为背景流量;
- I 支持基于脚本随机产生测试流量;
- I 支持以混合、压力和互斥等方式播放流量。

播放出去的流量数据将被 NBOS 功能程序读取, 通过判定其运行后的结果是否与脚本描述预期一致达到对程序的正确性进行验证的目的。

2 工作流程的设计



图 1 系统工作流程图

NBOS 静态测试平台的工作流程图如图 1 所示, 系统根据用户的需求生成脚本文件, 脚本文件按照约定的格式去保存用户提交的信息; 然后系统根据脚本文件生成符合用户需求的 NBOS 格式流记录文件; 为了能够控制生成流的播放时间, 我们需要把

流记录文件按流终止时间去划分成每 5 分钟为一个粒度的临时流记录文件，系统将处理后的临时流记录文件与背景流混合播放。

3 系统结构的设计

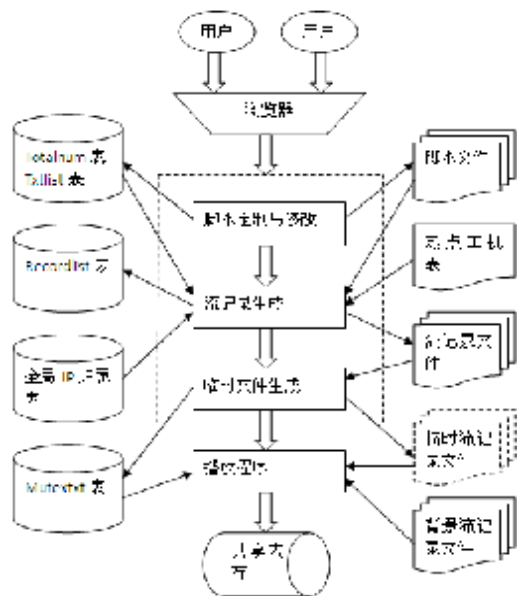


图 2 系统结构图

系统结构图如图 2 所示，系统主要包括 4 个模块：脚本定制与修改，流记录生成，临时文件生成，播放程序。具体说明如下：

①系统采用 B/S 架构，用户通过浏览器去提交自己的请求，后台服务程序根据用户的请求去做出相应的响应。

②脚本定制与修改主要支持用户描述自己所需流的特征，如果用户的描述符合约束条件，此模块则将用户所描述的流的信息保存到脚本文件中，并将脚本信息保存到数据库中的 txtlist 表中。用户可以描述的流特征信息见附录 2。

③流记录生成模块用于生成流记录文件，读入用户定制的脚本，并通过检索数据库以确定本次流记录文件的文件名，由流记录文件生成算法产生流记录文件，同时将其作为一条新记录添加至数据库中的 recordlist 表。④NBOS 以 5 分钟的流数据作为处理单位，而根据用户定制的脚本信息生成的流量没有按照 5 分钟一个单位去划分。临时文件生成做的主要工作就是根据流的终止时间去划分粒度，把终止时间在同一个 5 分钟之内的流划分到同一个粒度文件中去，并根据用户设定的流文件播放的时间把划

分过后的粒度文件放到相应的临时文件夹中供播放模块读取。⑤播放程序以实际流量作为背景流量去播放，在播放之前首先需要检查是否存在相同粒度临时流记录文件，如果存在，则把背景流与生成流按一定的混合方式共同播放。

系统用到数据库中 4 个表：totalnum，txtlist，recordlist，mutextxt。各表的功能如表 2 所示：

表 1 表的功能

表名	功能
totalnum	表 totalnum 用来保存系统中脚本的总个数，由脚本定制与修改模块去更新表中各个字段的值
txtlist	表 txtlist 中包括用户定制脚本名称、根据此脚本生成流记录文件的个数、流持续时间等字段，由脚本定制与修改模块和流生成模块共同维护
recordlist	表 recordlist 记录生成流的文件名和与其对应的流记录文件持续粒度数
mutextxt	表 mutextxt 主要为互斥播放类型服务，为播放程序提供互斥脚本，由临时文件生成模块维护

4 脚本结构

NBOS 静态测试平台使得用户可以按照自己的需求，在 web 端填写(包括 IP、端口、流记录数量等等)，提交后生成对应的脚本，流记录生成程序则根据用户的脚本生成满足用户需求的流记录，脚本在生成时按照事先约定的格式，因此流记录生成程序可以按规则读取并以此产生流记录。

我们约定的格式中：每一行代表一类确定的意义，最终确定每个脚本 25 行，它可以描述用户任意形式的一次定制需求，具体脚本固定格式见附录 3。

5 测试流量生成

用户在定制脚本时，有三种定制形式：只定制单向流、只定制双向流、同时定制单双向流；标志三种定制形式的量是“流记录数量”，若“流记录数量”不为默认值，表示该种流记录是用户已定制，否则不予生成该种流记录，也即：如果用户定制单/双向流记录，“流记录数量”应该是必填项，因此生成程序以“流记录数量”为标志判断是否生成对应流记录。

生成程序的输入：用户定制的脚本，输出：脚

本对应的流记录，根据脚本生成流记录的算法流程图如图 3 所示：

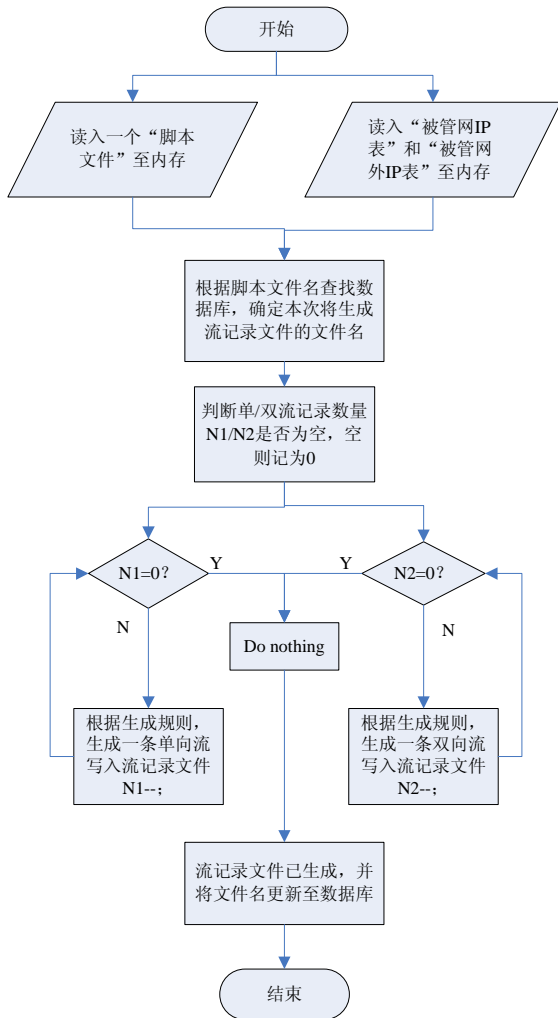


图 3 流记录文件生成算法图

6 播放程序

播放程序读取的背景流量是在 2009 年 10 月 1 日到 2009 年 10 月 7 日期间采集的实际流量，根据流终止时间在同一个 5 分钟之内把这些背景流量划分成 2016 个粒度文件，播放程序每隔 5 分钟读取一份粒度文件到共享内存中，循环播放背景流量。背景流量可以随时根据需要进行替换。为了实现播放程序能够在任意时刻读取生成流，系统做了如下设计：设计和背景粒度相匹配的 2016 个文件夹，把需要在某个粒度播放的临时文件存放到与粒度号相对应的文件夹中，播放程序每次读取背景流量时先检查与背景粒度号匹配的文件中是否有生成流记录文件，若有则按指定的混合方式把生成流与背景

流混合播放。

NBOS 静态测试平台提供的混合播放方式如下表所示：

表 2 混合播放方式

混合方式	简要说明
前插入	先播放完产生流记录，再播放背景流记录
中插入	在背景流播放一半时播放产生流记录
后插入	播放完背景流后再播放产生流
规律插入	按一定的规律去轮流播放背景流和产生流
随机插入	随机的选择播放产生流或者是背景流
互斥插入	按指定的方式过滤掉背景流中符合要求的流
单独播放	只播放产生流，不播放背景流

背景流量中会存在和生成流某些字段相同的流记录，这样会导致 NBOS 功能程序运行后得出的结果与预期结果不一致，为此我们需要根据用户的需求过滤掉背景流中与生成流相匹配的流量。当用户需要互斥播放时，需要提供互斥脚本，播放程序从互斥脚本中获得源、宿 IP，源、宿端口，协议号的信息，这些信息构成互斥五元组。文中把根据此五元组称为常规互斥条件，其判断判定方式如表 5 所示：

表 3 常规互斥条件判定

背景流	混入流	相等或者属于	判断结果编号
源 IP	源 IP	是	A1
		否	A2
宿 IP	宿 IP	是	B1
		否	B2
源端口	源端口	是	C1
		否	C2
宿端口	宿端口	是	D1
		否	D2
协议号	协议号	是	E1
		否	E2

①当背景流与混入流的判断结果编号为：A1、B1、C1、D1、E1 的组合时，背景流将会被过滤，不予播放。

②除了①中以外的任何组合，背景流不被过滤，将会正常播放

为了满足 NBOS 的测试要求，NBOS 静态测试

平台还提供其它的互斥选项：POP 互斥、ID 互斥、源 IP 互斥、宿 IP 互斥、源端口互斥、宿端口互斥、协议互斥。

7 测试实例

NBOS 系统需要在 NBOS 静态测试平台上要做包括丢包率测试、RTT 测试、带宽统计测试、流计算测试、热点主机和活跃地址测试、聚合分析测试、僵尸网络监测在内的十余个功能性测试。

丢包率测度程序实现对 NBOS 流记录的丢包率测度计算。丢包率算法模型针对报文往返比在 (1,2) 区间的双向 TCP 流。并以流的往返报文数为参数来计算单个 NBOS 流记录的丢包。完成被管单位与非被管单位之间丢包率计算，按照非被管单位的不同粒度将丢包率进行聚类写入数据库。丢包率测度计算模型为：

$$LossRate = \frac{3r^2}{4r + 2 + \sqrt{-11r^2 + 16r + 4}}$$

其中 $r = \frac{2N - M}{M}$ ，M>N 且 M, N 为往返

报文数。程序流成图如图 4 所示：

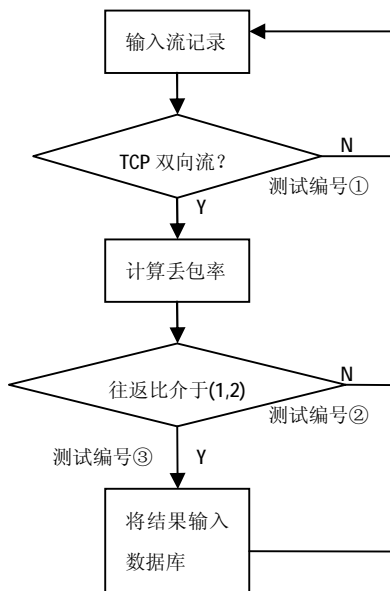


图 4 丢包率测试程序流成图

针对测试计算程序流程图三个主要功能分支，在 NBOS 静态测试平台上分别设计三个对应测试脚

本对程序进行功能性测试。测试所用 IP 列表如表 6 所示：

表 4 IP 列表

IP	POP 归属
58.192.112.111	NJ (东南大学)
211.68.69.254	BJ (北京邮电大学)
202.113.15.83	TJN (天津大学)
202.120.2.102	SH (上海交通大学)
202.117.112.10	XA (西安电子科技大学)

所选取的三个测试脚本见附录 4。

测试步骤如下：

- I 将三个测试脚本生成流记录文件
- I 对三个流记录文件进行 POP 互斥播放，启动程序 NBOS_packet_loss_rate 以及 NBOS_matrix_calculate
- I 进入页面 [http://202.112.25.131/NBOS HTML/second unit_QOS.php#tab4](http://202.112.25.131/NBOS_HTML/second_unit_QOS.php#tab4) 查看测试结果

我们得到的测试结果如表 10 所示：

表 5 测试结果

测试脚本名	预期结果 (单独播放)	测试结果	是否与预期结果相符 (是/否)
240_loss_134_1.txt	图表中横坐标不出现 POP 节点 BJ 以及 TJN 对应的丢包值	不出现 POP 节点 BJ 以及 TJN 对应的丢包值	是
241_loss_134_2.txt	图表中横坐标不出现 POP 节点 SH 对应的丢包值	不出现 POP 节点 SH 对应的丢包值	是
242_loss_134_3.txt	图表中横坐标出现 POP 节点经 XA 对应的丢包值，且值为 0.00768447	0.76 (%)	是

NBOS 系统各项功能性测试均在 NBOS 静态测试平台上顺利通过，测试结果均与预期结果相一致。

8 结束语

文中设计并实现了 NBOS 静态测试平台，此平

台主要为 NBOS 功能性测试和压力测试提供测试数据。它的显著特点是能够根据用户的需求产生具有特定特征的流数据，并把这些生成流与背景流按一定的方式混合播放。为了不让背景流干扰 NBOS 各功能模块的运行结果，测试平台还提供了互斥播放这种混合播放方式，它能够很有效的过滤掉背景流中的干扰流量。目前该测试平台已经成功应用于 NBOS 的测试工作。

参考文献

1. 张艳，网络行为观测系统 NBOS 的并行化设计和实现[D]，南京，东南大学计算机科学与工程学院，2010
2. NetFlow Overview.Cisco 技术手册

附录 1：NBOS 流记录格式

(1) 单向流数据格式

序号	字段	说明	类型
1	srcaddr	源 IP 地址	u_int32
2	dstaddr	宿 IP 地址	u_int32
3	srcport	源端口	u_int16
4	dstport	宿端口	u_int16
5	src_as	源地址自治域号	u_int16
6	dst_as	宿地址自治域号	u_int16
7	src_id	源 IP 地址的归属网络 id	u_int32
8	dst_id	宿 IP 地址的归属网络 id	u_int32
9	src_mask	mask length of source address	u_int8
10	dst_mask	mask length of destination address	u_int8
11	prot	IP 协议号，6=tcp，17=udp	u_int8
12	toa	应用标签	u_int8
13	first_secs	current seconds at first of flow since 1900	u_int32
14	first_msecs	residual milliseconds at first of flow since 1900	u_int32
15	last_secs	current time in seconds of last packet of flow since 1900	u_int32
16	last_msecs	residual milliseconds of last packet of flow since 1900	u_int32
17	dPkts	Packets sent in Duration	u_int32
18	dOctets	Octets sent in Duration	u_int32
19	input	Input interface index	u_int16
20	output	Output interface index	u_int16
21	tos	IP Type-of-Service	u_int8
22	Tcp_flags	OR of TCP header bits	u_int8

23	engine_type	Type of flow switching engine (RP,VIP,etc.)	u_int8
24	pad	备注。	u_int8

(2) 双向流数据格式

序号	字段	说明	类型
1	srcaddr	源 IP 地址	u_int32
2	dstaddr	宿 IP 地址	u_int32
3	srcport	源端口	u_int16
4	dstport	宿端口	u_int16
5	src_as	源地址自治域号	u_int16
6	dst_as	宿地址自治域号	u_int16
7	src_id	源 IP 地址的归属网络 id	u_int32
8	dst_id	宿 IP 地址的归属网络 id	u_int32
9	src_mask	mask length of source address	u_int8
10	dst_mask	mask length of destination address	u_int8
11	Prot	IP 协议号, 6=tcp, 17=udp	u_int8
12	Toa	应用标签	u_int8
13	Tos	IP Type-of-Service	u_int8
14	tcp_flags	OR of TCP header bits	u_int8
15	Pad	备注。	u_int16
16	first_secs	current seconds at first of flow since 1900	u_int32
17	first_msecs	residual milliseconds at first of flow since 1900	u_int32
18	last_secs	current time in seconds of last packet of flow since 1900	u_int32
19	last_msecs	residual milliseconds of last packet of flow since 1900	u_int32
20	dPkts	Packets sent in Duration	u_int32
21	dOctets	Octets sent in Duration	u_int32
22	input	Input interface index	u_int16
23	output	Output interface index	u_int16
24	engine_type	Type of flow switching engine (RP,VIP,etc.)	u_int8
25	tos_rv	IP Type-of-Service (反向流)	u_int8
26	tcp_flags_rv	OR of TCP header bits (反向流)	u_int8
27	engine_type_rv	Type of flow switching engine (RP,VIP,etc.) (反向流)	u_int8
28	first_secs_rv	Current seconds at start of flow since 1900 (反向流)	u_int32
29	first_msecs_rv	Residual mlliseconds at start of flow since 1900 (反向流)	u_int32
30	last_secs_rv	Current time in millisecs of last flow since 1900 (反向流)	u_int32
31	last_msecs_rv	Residual milliseconds of last flow since 1900 (反向流)	u_int32

		流)	
32	dPkts_rv	Packets sent in Duration (反向流)	u_int32
33	dOctets_rv	Octets sent in Duration (反向流)	u_int32
34	input_rv	Input interface index (反向流)	u_int16
35	output_rv	Output interface index (反向流)	u_int16

附录 2: 用户可以定制的特征流信息

单向流	双向流	
	正向流	反向流
源 IP 地址		
宿 IP 地址		
源端口		
宿端口		
协议号		
报文数		
字节数		
		正反向流终止时间间隔
流记录数量		
生成流量持续粒度数		
流记录平均持续时间		
流记录最小持续时间		
流记录最大持续时间		
流持续时间服从分布		
定制人 ID		

附录 3: 脚本格式表

行序号	意义	描述
1	(单) 源 IP 地址	四种类型: (1) 0-IP: 表示固定 IP (2) 1-IP1-IP2: 表示 IP 范围(在此范围内随机生成) (3) 2-IP1-IP2: 表示 IP 范围(在此范围内顺序生成) (4) * : 默认生成
2	(单) 宿 IP 地址	同“(单)源 IP 地址”描述
3	(单) 源端口	四种类型: (1) 0-port: 表示固定端口 (2) 1-port1-port2: 表示端口范围(在此范围内随机生成)

		(3) 2-port1-port2: 表示端口范围(在此范围内顺序生成) (4) * : 默认生成
4	(单) 宿端口	同“(单)源端口”描述
5	(单) 报文数	(1) 整数: 表示每个单向流记录中的报文数量 (2) * : 表示取默认值, 比如 1
6	(单) 字节数	(1) 整数: 表示每个报文的字节数 (2) * : 表示默认生成
7	(单) 流记录数量	整数, 表示单向流记录条数
8	(双) 源 IP 地址	同“(单)源 IP 地址”描述
9	(双) 宿 IP 地址	同“(单)源 IP 地址”描述
10	(双) 源端口	同“(单)源端口”描述
11	(双) 宿端口	同“(单)源端口”描述
12	(双) 报文数 1	(1) 整数: 表示双向流中, 每个正向流记录中的报文数量; (2) *: 表示取默认值, 比如 1
13	(双) 字节数 1	(1) 整数, 表示双向流中, 正向流记录报文的字节数 (2) * : 表示取默认值, 比如 46
14	(双) 报文数 2	(1) 整数: 表示双向流中, 每个反向流记录中的报文数量; (2) *: 表示取默认值, 比如 1
15	(双) 字节数 2	(1) 整数, 表示双向流中, 反向流记录报文的字节数 (2) * : 表示取默认值, 比如 46
16	(双) 正、反向流时间 (last) 间隔	以 s 为单位的整数
17	(双) 正反终止时间在后比例	形如 R1:R2, 其中 R1、R2 是小于 1 的一位小数, 和为 1
18	(双) 流记录数量	整数, 表示双向流记录条数
19	流记录文件持续时间粒度数	整数, 该脚本生成的所有流记录持续的粒度数
20	流记录平均持续时间	整数, 以 s 为单位的秒数
21	流记录最大持续时间	整数, 以 s 为单位的秒数
22	流记录最小持续时间	整数, 以 s 为单位的秒数
23	流持续时间分布方式	整数, 0: 表示正太分布, 1: 表示均匀分布, 2: 表示指数分布
24	(单) 单向流协议	格式 (6 个整数): N1-N2-N3-N4-N5-N6 分别表示: UDP 协议百分比、TCP 协议百分比、其他协议 1 协议号、其他协议 1 百分比、其他协议 2 协议号、其他协议 2 百分比

25	(双) 双向流协议	同“(单) 单向流协议”描述
----	-----------	----------------

附录 4：测试脚本

(3) 测试脚本 1

单向流		
所设参数	值	目的
源 IP 地址	211.68.69.254	产生被管子网到 CERNET 非被管网络的流量
宿 IP 地址	58.192.112.111	
协议	TCP	加入“单向 TCP 流”，来测试算法过滤功能正确性
双向流		
所设参数	值	目的
源 IP 地址	202.113.15.83	产生被管子网到 CERNET 非被管网络的流量
宿 IP 地址	58.192.112.111	
协议	UDP	加入报文往返比在 (1,2) 区间内的双向 UDP 流，来测试算法过滤功能正确性
正向流报文数	100	
反向流报文数	150	

(4) 测试脚本 2

双向流		
所设参数	值	目的
源 IP 地址	202.120.2.102	产生被管子网到 CERNET 非被管网络的流量
宿 IP 地址	58.192.112.111	
协议	TCP	加入报文往返比在 (1,2) 外的双向 TCP 流来测试程序功能的正确性
正向流报文数	100	
反向流报文数	400	

(5) 测试脚本 3

双向流		
所设参数	值	目的
源 IP 地址	202.117.112.10	产生被管子网到

宿 IP 地址	58.192.112.111	CERNET 非被管网络的流量
协议	TCP	加入报文往返比在(1,2)之内的双向 TCP 流来测试程序功能的正确性（模型 2-1）
正向流报文数	100	
反向流报文数	180	