第31卷 第24期 计 算 机 工 程 2005年12月
Vol.31 № 24 Computer Engineering December 2005
·网络与通信· 文章编号：1000—3428(2005)24—0123—03 文献标识码：A 中图分类号：TP393

基于统计的网络流量异常检测模型

梁 昇，肖宗水，许艳美
（山东大学计算机科学与技术学院，济南 250061）

摘 要：提出了一个基于统计的流量异常检测模型。根据网络流量的可测度集，描绘了一个正常网络流量的基线。参照该正常流量基线，使用假设检验理论进行异常检测。采用一个基于滑动窗口的流量更新策略，使异常检测能够更加高效。论述了在高速网络情况下提高检测性能的方法。

关键词：异常检测；网络基线；滑动窗口

Anomaly Detection Model of Network Traffic Based on Statistics

LIANG Sheng, XIAO Zongshui, XU Yanmei
(School of Computer Science and Technology, Shandong University, Jinan 250061)

【Abstract】The paper presents a traffic anomaly detection model based on statistics. According to the measurable aggregate of network traffic, a normal network baseline is built. Compared to the normal network baseline, the theory of hypothesis test is used to execute the anomaly detection. In order to make the anomaly detection more efficiently, the model adopts a traffic update policy based on glide window. The method of improving detection capability in the high-speed network is also discussed.

【Key words】Anomaly detection; Network baseline; Glide window

网络流量异常包括主干流量异常和某台网络主机的流量异常，事实上网络主机的流量异常造成了主干网络流量的异常，因此研究异常流量的监测，定位造成异常的主机，进而对异常主机进行网络隔离，对于避免网络拥塞、保证网络性能、避免网络资源的滥用，具有重要意义。

流量异常检测的核心问题是实现流量正常行为的描述，并且能够实时、快速地对异常进行处理。检测方法可以归结为以下4类：(1)阈值检测方法；(2)统计检测方法；(3)基于小波的检测方法；(4)面向网络安全的检测方法[1]。本文采用了一种基于统计的流量异常检测方法，首先确定正常的网络流量基线，然后根据此基线利用正态分布假设检验实现对当前流量的异常检测。

1 总体设计

网络流量异常检测模型的总体设计思路是：从局域网的总出口采集数据，对每个数据包进行分类，将它的统计值传到相应的存储空间，然后对这些数据包进行流量分析。系统的基本组成见图1。

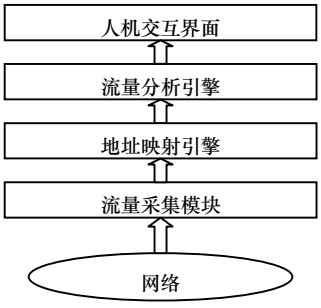

图1 模型总体框架

(1)流量采集模块。在网络核心设备上设置流量采集点，将负责采集流量的服务器网卡设为混杂模式，能够把所有内网主机的流量都采集到。流量采集有两种情况：对共享网络进行采集和对交换式网络进行采集。前者直接在共享设备通过一个端口连接sniffer即可；后者需要设备支持，通过交换机的SPAN（Switched Port Analyzer）端口把流量镜像到sniffer。SPAN在使用中非常灵活，可以监视交换机的单个端口，也可以监视多个端口，还可以对VLAN进行监视。利用libpcap和linnet开发高效的协议分析引擎，对镜像出来的数据包进行报文的监听与发送，从而便于流量统计和流量异常的检测。

(2)地址映射引擎。由于我们要对内网中每台主机的流量情况进行检测，以便定位异常的原因和造成异常的主机，因此对于内网中的每个IP地址，都需要建立一个存储结构，用来存储需要测量的值。对捕获的每个数据包，区分它是流入的还是流出的数据。如果是流入的数据，则把相应的测量值累加到目的地址所对应的存储结构中上去；如果是流出的数据，则把相应的测量值累加到源地址所对应的存储结构中。如果局域网比较大，查找IP地址时会花费很多时间，流量检测的实时性就下降了，检测效率也会下降。因此设立一个地址映射引擎，可以快速查找地址。一般的局域网都是由几块连续的IP地址组成，我们采用数组加链表的形式组成查找引擎。连续的地址可以存放在一个数组中，不连续的数组之间用链表连接，这样可以提高查找效率，降低系统开销。

(3)流量分析引擎。对采集到的数据进行分析处理。通过建立正常网络流量模型，按照一定的规则进行流量异常检测。同时根据滑动窗口的更新策略，能够自动学习最近的流量情况，从而调整检测的准确度。该部分在后文详述。

作者简介：梁 昇(1977—)，男，硕士生，主研方向：计算机网络；肖宗水，副教授；许艳美，硕士生

收稿日期：2004-10-17 **E-mail：**liangsheng@sdu.edu.cn

(4)人机交互界面。实现一个基于 Web 的用户管理界面，可以通过该界面实现查看信息、设定检测规则、设定阈值、设定报警方式以及处理报警等功能。系统应该对于检测出的异常主机进行标记，标记异常的类型、统计量、阈值指标、消息以及异常发生的时间等情况，给系统管理员报告一个异常信息。这个报告可以写入日志，也可以发出声音或者给管理员发送消息。

2 建立正常网络流量模型

要进行流量异常检测，必须首先建立正常的网络流量模型，然后对比正常模型能够识别异常。本文使用基于统计的方法来实现异常检测，利用网络流量的历史行为检测当前的异常活动和网络性能的下降。因此正常流量模型的建立需要把反映网络流量的各项指标都体现出来，使其能够准确反映网络活动。

定义 1 可测度集 设F为一组属性值集合，包含所有审计数据或连接记录中可能出现的属性值，例如源/目的地址、连接标识等，称之为可测度集。设V={$field_1$,$field_2$,…,$field_n$}为F的子集，称之为记录属性[2]。

可测度集表明一个数据包或一条连接的属性集合，例如网络连接的可测度集为(timestamp, service, s_host, d_host, s_port, d_port, flag)，分别表示连接的时间戳、服务类型、源主机、目的主机、源端口、目的端口、标志等属性。我们将需要检测的信息作为记录属性。

传统的流量异常检测只监视少量的对象，例如源 IP、目的 IP、源端口、目的端口等。这样对某些特定的流量异常能够检测出来。但是通过对已有的流量异常分析后发现，仅有这些对象是远远不够的。比如有些主机持续发送大量特别短的包，这可能是在进行网络扫描，也是属于异常流量，这时统计包的长度就非常重要了。还有一些流量异常需要统计各种数据包的比例和数量。因此，选取的可测度集包括目的 IP 地址、源 IP 地址、目的端口、源端口、时间、数据包类型、TCP 头的 syn 和 ack 标志位、包长度、ICMP 协议类型以及 UDP 的相关统计。根据可测度集的内容，我们可以把正常的网络基线从以下几个方面描绘出来:

(1)广播、单播和组播数据包的比例和数量;

(2)对于多个典型的目标，ping 的响应时间;

(3)对于应用层的性能，用 ftp 传输的时间来衡量;

(4)正确的网络拓扑，包括二层拓扑和三层拓扑。二层拓扑可以通过 CDP 或 NDP 协议得到，三层拓扑可以通过路由表得到，使用 snmp 协议;

(5)正常的 Spanning Tree;

(6)正常的路由路经，通过 tracert 得到;

(7)各种协议报文的统计;

(8)各种服务端口报文的统计;

(9)TCP 中 syn 与 syn+ack 的比率,UDP 端口的入出比率，ICMP 的 request/reply 比率等。

在积累了一定数量的测度统计内容后，就形成了历史行为可测度集。利用这些流量历史行为，我们建立过去一段时间内的正常网络基线。在系统运行时，统计当前流量行为可测度集，并同正常的网络基线相比较，如果当前流量行为与正常网络基线出现明显的偏离时，即认为出现了异常行为，并可进一步检测分析；如果两种行为没有明显偏差，则流量正常，更新正常网络流量模型。

3 流量异常检测

Denning 提出了用于异常检测的 5 种统计模型: (1)操作模型: 该模型假设异常可通过测量结果和指标的比较而得到，指标可以根据经验或一段时间的统计平均得到。(2)方差: 计算参数的方差，设定其置信区间，当测量值超出了置信区间的范围时表明可能存在异常。(3)多元模型: 操作模型的扩展，通过同时分析多个参数实现检测。(4)马尔可夫过程模型: 将每种类型事件定义为系统状态，用状态转移矩阵来表示状态的变化，若对应于发生事件的状态转移矩阵概率较小，则该事件可能是异常事件。(5)时间序列模型: 将测度按时间排序，如一新事件在该时间发生的概率较低，则该事件可能是异常事件。

本文采用第 2 种统计模型，根据中心极限定理，如果所研究的随机变量 X 可以表示成很多个独立的随机变量 X_1，X_2，…，X_n之和，只要每个 X_i (i = 1,2,…,n) 对 X 只起微小的作用，不管这些 X_i服从什么分布，在 n 比较大的情况下，就可以认为 X 服从正态分布。由于网络流量测度都是独立的随机变量，因此这些测度可以使用该定理进行估计。比如平均数据包长度，假设每个数据包长度为 T_i，前 n 个数据包总长度为 X_n，平均长度为 $\overline{X_n}$。当有一个新包过来时，数据包总长度和平均长度会更新如下:

$$X_{n+1} = X_n + T_i,\quad \overline{X}_{n+1} = \frac{X_{n+1}}{n+1}$$

这样就得到了样本均值。对于 n 个数据，样本方差计算如下:

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2 = \frac{1}{n-1}(\sum_{i=1}^{n} X_i^2 - n\overline{X}_n^2)$$

样本均值和标准差能为流量特性的总体均值构造一个置信区间，利用这个置信区间可以判定异常。样本标准差的定义为

$$S_n = \sqrt{\frac{1}{n-1}(\sum_{i=1}^{n} X_i^2 - n\overline{X_n}^2)}$$

根据正态总体均值的分布情况可知，样本均值 $\overline{X_n}$ 的标准差为 $\frac{S_n}{\sqrt{n}}$。因此总体均值置信度为 $(1-\alpha)$ 的置信区间为

$$(\overline{X_n} - Z_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}},\ \overline{X_n} + Z_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}})。$$

其中 $Z_{\frac{\alpha}{2}}$ 可以根据正态分布表查得。这样只需在系统中维护 3 个值: 样本数 n、样本的平方和 $\sum_{i=1}^{n} X_i^2$ 以及样本均值的平方 $\overline{X_n}^2$，就可以算出样本均值的标准差，进而得出总体均值的置信区间。如果某个时间范围内的测度在此置信区间内，则认为流量正常；否则认为出现异常，进行异常处理。

4 正常流量模型的调整

网络行为是不断变化的,即使对于比较稳定的网络环境，也会随着用户行为的变化而变化。因此，正常流量模型必须是可调整的，要能够随着网络行为的改变而调整自己的历史行为网络基线。调整时，有些参数可以手动设置，比如广播和单播数据包的比例，ICMP 中请求和应答的比例；有些参数需要通过学习动态进行调整，比如对上面提到的平均包长度的调整。

手动调整时，需要根据历史经验或者网络流量的实际情况，为相关参数设置特定的值。此项内容可以在交互界面直

接手工输入即可。

动态调整测度需要设置一个滑动窗口，利用这个窗口取得新样本，去掉旧样本，这样可以保证窗口内的测度值为最近最新的历史行为。为了维护一个具有固定大小的滑动窗口队列，需要在窗口队列的头部抛弃旧数据，在队列尾增加新到的数据。因为对窗口数据的抛弃和增加是由时间顺序来决定的，所以本文考虑使用基于单位时间尺度的滑动窗口模型。

设滑动窗口的时间长度为 n 个单位时间，在每个单位时间过后需要向前滑动一个单位时间。设窗口队列的头指针为 h，尾指针为 t，则做如下操作：$h \leftarrow h+1$，$t \leftarrow t+1$。其它需要统计的可测度值都要随着更新，把掉出窗口时间段的相应值去掉，再加上进入窗口时间段的相应值。如果检测出异常，则不更新窗口，使用窗口数据继续进行检测。这样虽然会损失一部分数据，但是没有把异常数据加到正常模型中，能够使正常模型更准确。

滑动窗口大小的设置也会对系统有影响。滑动窗口越大，可容纳的信息就越多，从而统计的结果也越准确。但是，大的滑动窗口保存的数据太多，导致消耗的系统资源也多。单位时间长度越小，窗口更新越快，窗口流量也越接近当前流量情况。但是更新频繁也会导致资源的浪费。因此，可以根据实际情况设置单位时间的长度和滑动窗口的大小。

5 检测性能优化

传统的网络流量检测中，都要求对网络中的数据进行直接和高效的实时访问。软件工具在对网络流量的处理中常常是首选，因为它们比硬件的方法价格低而且使用方便。像有名的 libpcap 和 WinPcap，它们可以在许多操作系统中使用，允许应用在没有其它层的干扰下直接与网络相互作用。然而，随着千兆以太网的出现，这些工具在高速网络中被认为可能存在性能问题，在实时流量分析时会产生丢包现象。这就使得解决高速网络问题时必须使用硬件方法。硬件的解决办法通常比较昂贵，又难以部署（例如，硬件不能被复制和轻易移动），而且它们比软件的解决办法灵活度要差。

当前世界上有很多团队在研究如何提高网络分析工具的整体性能，但是效果不太明显。这些研究工作的最大问题在于它们专注于一些流量分析的特殊构件（例如包过滤装置）并且提出提高这些构件性能的解决办法。因为用户感兴趣的是整个流量分析系统的性能而不是单独的一个构件的性能，所以结果表明这些方法是低效的。

数据包从网卡进入并到达用户应用的路径见图 2。

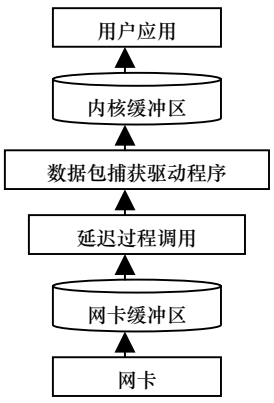

图 2 数据包从网卡到用户应用的路径

当网卡接收到一个有效数据包后，就向总线控制器产生一个总线控制数据传输请求。此时，网卡控制总线传输数据包给网卡缓冲区（见图 2），释放总线，并向高级可编程中断控制器（APIC）芯片产生一个硬件中断。这个芯片唤醒操作系统中断处理例程，它触发网卡驱动程序的中断服务例程（ISR）。接着，ISR 调度延迟过程调用（DPC）。数据包捕获驱动程序把接收到的数据包和那些可能有用的物理层信息，例如长度和接收时间戳联合在一起。然后这些数据包被拷贝到内核缓冲区，它们在缓冲区中等待着被传送到用户应用。用户应用通过一个类似于 read()的系统调用从内核缓冲区中检索数据包。

由数据包经过的路径可知，该过程的开销分为两部分：外部开销和数据包处理程序本身的开销。处理一个数据包涉及到几个部分，像网卡驱动程序和操作系统。虽然它们不是严格地作为数据包捕获体系结构的一部分，但是它们的开销也对数据包处理有一定影响。处理程序本身的开销包括从网卡接收数据包到发送至用户应用所经历的一切开销。根据涉及到的部分执行时所需要的 CPU 时钟周期来测量它们的开销，显示了大多数重要的瓶颈位于隐藏的地方。对数据包捕获驱动程序的过滤过程进行优化，用一个 Just In Time(JIT)引擎将所使用处理软件（如 libpcap）的过滤器翻译成 80x86 二进制代码；数据包从网卡缓冲区到内核缓冲区的拷贝开销很大，可以使用一个标准的 C 库函数来处理这个过程；附加一个简单的基于硬件的时间戳能显著改善数据包的处理。通过使用上述技术对这些地方进行优化，可以把数据包处理软件的性能几乎提升一倍[5]。

6 总结

本文给出了一个基于统计的流量异常检测模型，提出了网络流量的可测度集，描绘了一个正常网络流量的基线，为异常检测提供了参照。在异常检测时，我们使用了方差模型进行统计，把可测度集中的相关参数设定置信区间，然后对网络流量进行实时检测。落入置信区间的值为正常流量，置信区间之外的值为异常流量。由于网络行为是经常变化的，因此不能使用所有的历史流量与当前流量进行比较，应该使用最近的流量来代表网络行为。本文提供了一个滑动窗口进行流量更新，使异常检测能够更加准确。本文还论述了如何在高速网络情况下不使用硬件而提高检测性能。通过实验表明，该模型在高速网络环境中应用时，比传统异常检测模型减少了丢包率，并且在一定程度上提高了检测的识别准确率。

参考文献

1 邹柏贤. 一种网络异常实时检测方法[J]. 计算机学报, 2003, 26(8): 940-947

2 凌　军, 曹　阳, 尹建华等. 基于时态知识模型的网络入侵检测方法研究[J]. 计算机学报, 2003, 26(11): 1591-1597

3 程　光, 龚　俭, 丁　伟. 基于抽样测量的高速网络实时异常检测模型[J]. 软件学报, 2003, 14(3): 594-599

4 辛　颖, 徐敬东, 肖建华. 基于统计的异常检测引擎分析[J]. 计算机应用, 2002,22 (10): 48-50

5 Degioanni L, Baldi M, Risso F, et al. Profiling and Optimization of Software-based Network-analysis Applications[C]. 15th Symposium on Computer Architecture and High Performance Computing, 2003: 226-34