一种高速网络流量测量系统开发包

黄鹂声，汪文勇，孙维勇

(电子科技大学 宽带光纤传输与通信网技术教育部重点实验室 成都 610054)

【摘要】针对网络流量测量系统的性能问题，分析了目前流行的Libpcap开发包的机制和性能，并在Libpcap和RTFM流量测量架构的基础上，提出了一种通用内核模式开发包NADDK，该NADDK能够用于各种流量测量系统的设计和开发，功能灵活，性能明显优于基于Libpcap的流量测量系统。

关 键 词 流量测量； 开发包； 流量分类； 流量跟踪

中图分类号 TP393 **文献标识码** A

A DDK for High Performance Network Measurement System

HUANG Li-sheng，WANG Wen-yong，SHUN Wei-yong

(Key Laboratory of Brodband Optical Fiber Transmission and Communication Networks UEST of China, Ministry of Education Chengdu 610054)

Abstract Aiming at the performance problem of network flow measurement system, we analyzed the mechanism and performance of the popular Libpcap flow measurement develop kit, and proposed a general kernel mode network analyze driver development kit based on Libpcap and realtime traffic flow measurement frame. It can be used to design and develop various kinds of flow measurement systems. Systems based on analyze driver development kit have flexible functions, and have the performance obviously better than Libpcap based systems.

Key words flow measurement; development kit; flow classification; flow tracing

随着网络规模的不断扩大，以及语音、视频等各种新的网络应用的普及，对骨干网络流量的测量与分析显得越来越重要。网络流量测量与分析能为网络计费、协议行为统计分析提供技术支持，并能为网络扩展部署提供决策信息，对网络安全威胁进行防范。

目前用于网络测量的产品很多，既有软件系统也有硬件系统。软件系统由于功能灵活、部署成本低廉而获得广泛的应用。IETF相应的工作组定义了多种网络安全测量软件的标准构架，如实时流量测量(Realtime Traffic Flow Measurement，RTFM)等[1]。但RTFM只定义了测量程序的整体架构，没有对其中的关键部件流量测量器(Meter)的实现标准、性能等指标进行详细的设计和定义，难以满足实际的流量分析需求[2]。目前根据RTFM实现的网络测量系统，如NeTraMet，存在功能有限、性能不高的局限性[3]。

本文提出一种通用型流量测量开发包(Network Analyze Driver Development Kit，NADDK)，NADDK基于著名抓包工具Libpcap，能够为RTFM构架中的Meter组件开发提供有力支撑[4]。本文将首先对NADDK开发包的功能进行设计，然后介绍主要技术原理，最后分析性能指标。

1 功能设计

NADDK开发包提供的基本功能包括以下5个方面：

(1) 宏观流量统计，如总流量、平均每秒网络帧、平均包长、流量比重分析等；

收稿日期：2005－09－26

作者简介：黄鹂声(1975－)，男，硕士，主要从事计算机网络方面的研究.

(2) 单数据流跟踪分析，对各个单独的传输控制协议(Transfer Control Protocol，TCP)和用户数据报协议(User Datagram Protocol，UDP)通信进行协议跟踪，统计结果可包括平均延时、协议安全分析、协议流量统计、应用协议分析，以及其他相关的统计分析；

(3) 基于流量分类的统计分析，对流量进行多级别分类，并针对不同的分类进行各种指标分析，统计指标包括宏观流量统计和单数据流跟踪统计所包含的各项指标，时间窗口可调；

(4) 安全测量，对安全事件和安全威胁进行分析和发现；

(5) 性能测量，对网络性能、关键应用服务性能进行统计分析。

2 技术构架

NADDK开发包基于Libpcap开放源代码进行改写和扩展，以实现更强大的统计功能和更高的分析性能。

首先分析Libpcap的工作机制，Libpcap系统构架如图1所示[5]。可以看出，Libpcap通过对网络帧的过滤、缓冲、拷贝，将网络帧传递到各个应用程序缓冲区。Libpcap仅提供宏观上的流量统计，其余分析工作交给应用程序自身完成。

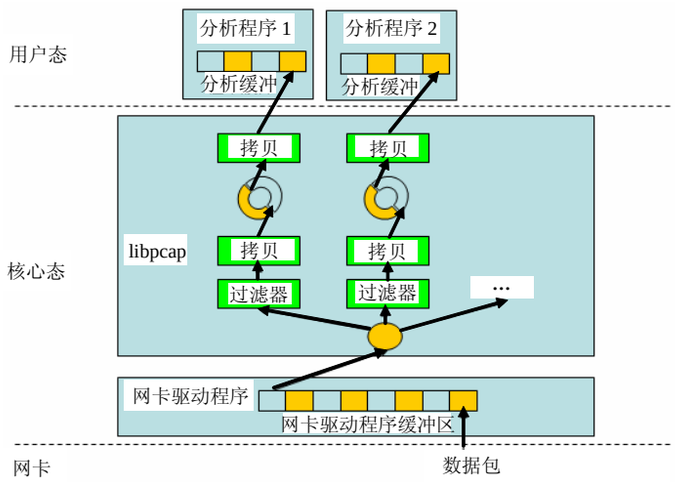

图1 Libpcap技术构架

Libpcap的主要目的在于向应用程序提供通用的“网络帧捕获”功能，在Libpcap技术构架下开发完成的网络流量测量系统必然存在如下性能问题[5]：

(1) 多次数据拷贝造成性能降低，每拷贝一个帧需要消耗259～8 550个CPU时钟周期；

(2) 为在Libpcap和应用之间拷贝数据帧，系统必须不断地在核心态(kernel mode)和用户态(user mode)之间不断切换，每次切换约消耗1 200个CPU时钟周期；

(3) Libpcap自身内置的时间戳timestamp功能，每处理一个帧约消耗270个CPU时钟周期；

(4) 分析程序中包含的多种复杂算法需要实时运行，但在libpcap构架中，分析程序作为普通进程，运行于操作系统用户态，受到其他进程的影响，性能极不稳定，导致大量丢包。

为了避免上述性能问题，NADDK采用了不同的运行模式，将大量的通用流量分析功能(如RTFM的Meter部分)放入操作系统核心态，以内核模式运行，并按照RTFM标准，提供多种配置管理和数据获取接口。开发完成的NADDK构架如图2所示。

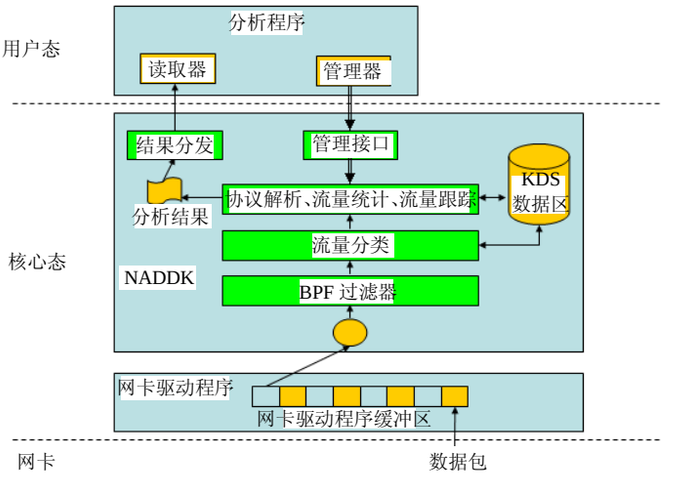

图2 NADDK的技术构架

NADDK能够从多方面提升流量测量系统的性能：由于在内部采用了零拷贝，因此节省了内存拷贝带来的开销；数据帧不再被传递到应用程序，系统不再频繁地在核心态和用户态之间切换；取消timestamp功能可提高抓包效率；更为重要的是，对网络帧的实时分析功能全部在核心态完成，不再受操作系统进程调度的约束，性能非常稳定。

NADDK在Libpcap原有功能的基础上，添加了流量分类、流量统计、协议解析、协议跟踪等多项功能，同时将Libpcap库函数由30多个扩展到160多个，使流量测量系统可以轻松地获取各项流量统计指标。

3 部分实现细节

NADDK的目标是产生一个通用的网络流量分析构架，因此必须在功能上实现最大程度的灵活性，许多关键技术细节需要认真考虑其实现方式。

3.1 多规则集流分类问题

为使NADDK能同时按照多个规则集对流量进行实时分类，必须解决对多个流分类规则集的合并和选择合适的流分类算法两个问题。

为了对流分类规则集进行合并，本文提出并设计了一个规则集合并编译器，其主要原理为假设规则集S_1将网络流量划分为多个域{D_1，D_2，…，D_m}，而规则集S_2采用另外的标准，将流量划分为{E_1，E_2，…，E_n}，通过编译器进行集合相交，对S_1和S_2进行合并，可以得到更为细致的域划分{F_1，F_2，…，F_k},其中每个域F_x可能是D_i的一个子集，也可能是E_j的一个子集，这两个规则集被合并为一个规则集，虽然规则数量增加，但只需为每条新规则的数据结构增加CLASSID链表，分别指向原始规则集的相应规则记录，就可以实现一次查找，多规则集共享，查找次数大大降低，提高了效率。

常见的流分类算法大致可以分为：硬件算法，基本数据结构算法，几何算法，启发式算法(如RFC算法，HiCuts算法，tuple-space search算法等)[6-7]。启发式算法中的HiCuts算法速度快，可以多维分类，而且更新复杂度低，得到广泛应用。HiCuts算法的一个重要特点就是能够调整算法的一些参数，对时间复杂度和空间复杂度进行调整平衡。本文改进了HiCuts算法，在保持Hicuts算法原有优点的基础上，对其规则预处理和CLASSID的获取方式进行改进，实现了一次查找就可获得多个CLASSID的功能。

3.2 流量跟踪与分类统计问题

在测量系统中，为了实现安全分析和协议解析功能，往往需要对单个TCP连接进行状态跟踪，这就需要为每个TCP连接建立跟踪数据结构，为了实现快速查找和定位，目前通常采取的技术手段是哈希链表。

此外，出于统计要求，测量系统应能按照流量分类规则集，对所有网络连接的统计指标进行高速汇总，但由于HASH表的特性，若直接对其进行遍历统计，效率会相当低下。

针对以上问题，本文提出了“回溯式二阶段统计”方法。其主要数据结构如图3所示。

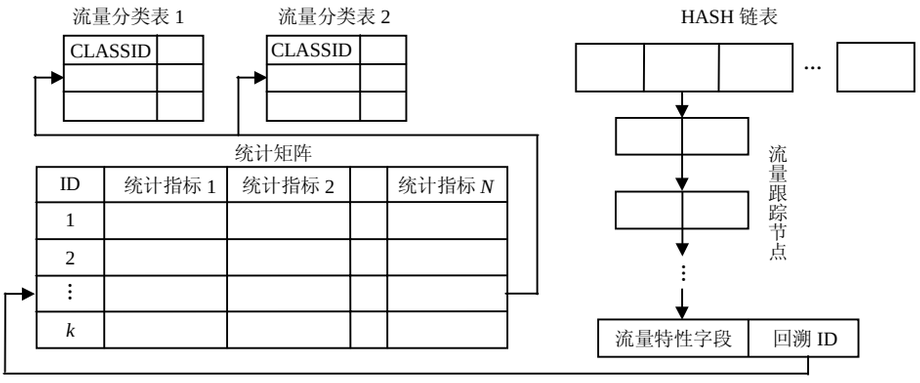

图3　回溯式二阶段统计

所有TCP连接仍以HASH链表的方式建立协议跟踪数据结构，在多规则集合并的基础上，为所有流量分类表建立一个公用的中间阶段统计表T，每个TCP连接都可以通过分类ID，回溯到统计表T中的某个单元，修改其统计数据，这样，对每个数据包，都可以$O(1)$的时间开销来完成初步的流量指标统计。统计表各单元存放的临时统计结果，又可以被回溯到流量分类表中，根据需要被再次统计，从而成为不同流量分类的汇总统计结果。统计过程被分为两个阶段：从原始流量跟踪链表到统计表T和从统计表到结果流量分类表两个阶段。

回溯式二阶段统计方法的空间复杂度仅为$O(M*K)$，其中M为统计指标数量，K为流分类规则集合并后的子域数量；时间复杂度也较低，第一阶段为$O(1)$，第二阶段为$O(M*K)$，而且第二阶段并非实时完成，而是根据可调整的时间窗口，间隔固定时间执行一次。

3.3 内存分配问题

在内核态中的大规模内存分配和使用是一个相当谨慎的问题，NADDK的大部分代码以内核级驱动方式运行，其中必然需要分配大量内存。由于Windows的内存管理机制和C语言编译上的问题，频繁的内存申请和释放会造成大量的内存碎片，而且容易堆积内存垃圾。

因此，本文采取的是预先分配全局内存堆方式，并以链表、数组的方式进行格式化，各个模块在内存堆中申请和释放存储空间，避免直接向操作系统申请分配存储空间。

4 性能测试

为了验证设计性能，将同一套流量分析代码分别加入不同的编译工程，分别以“NADDK+管理器”方式和“Libpcap+应用程序”方式编译，在880 Mb/s峰值的网络流量环境下同时运行并进行对比观测，同时部署流量生成工具，生成测试流量以验证丢包率。用于试验的硬件平台为2.4 GHz CPU的PC服务器，内存为2 GB，操作系统为Windows XP Professional，开发编译工具为VC6.0。经过连续测试，结果如表1所示。

表1 NADDK性能对比测试结果

流量 /Mb·s^{-1}	每秒网络帧 fps/K	CPU开销/(%) (Libpcap+程序)	CPU开销/(%) (NADDK+管理器)	丢包率/(%) (Libpcap+程序)	丢包率/(%) (NADDK+管理器)
500～520	116	65	44	<0.01	0
600～620	140	67	46	0.02	0
700～720	162	70	50	0.1	0
800～820	186	75	55	0.2	0

5 结 论

经过分析和实验，证明基于NADDK开发完成的流量测量系统具有较高的效率，而且性能稳定，在试验过程中，能够对测试流量进行100%捕获和处理，特别适合于对丢包率要求严格的测量场合，具有深入开发研究的价值。当然，NADDK本身在灵活性、算法性能、存储设计等方面还有待进一步研究和改进。

参 考 文 献

[1] Brownlee N. RTFM: Applicability statement [EB/OL]. http://www.ietf.org/rfc/rfc2721.txt, 1999-10-21

[2] Brownlee N, Mills C, Ruth G. Traffic flow measurement: architecture[EB/OL]. http://www.ietf.org/rfc/rfc2722.txt, 1999-10-21

[3] Brownlee N. Traffic flow measurement: experiences with NeTraMet. [EB/OL]. http://www.ietf.org/rfc/rfc2723.txt, 1997-03-23

[4] WinPcap Workgroup. WinPcap documentation[EB/OL]. http://winpcap.polito.it/docs/, 2005-06-01

[5] Degioanni L, Baldi M, Risso F, et al. Profiling and optimization of software-based network-analysis application [EB/OL]. http://www.winpcap.org/docs/WinPcap-SBAC03.pdf, 2003-11-19

[6] Gupta P, McKeown N. Packet classification on multiple fields[C]. ACM SIGCOMM'99, Massachusetts, 1999

[7] Srinivisan V, Suri S, Varghese G. Packet classfication using tuple space search [C]. ACM SIGCOMM Massachusetts, 1999

编 辑 熊思亮