

一种高性能内容过滤网关的设计与实现

何怀文^{1,2}, 张绮雯^{1,3}, 傅 瑜¹

(1 电子科技大学 中山学院计算机学院, 广东 中山 528402; 2 中山大学 信息科学与技术学院, 广东 广州 510275; 3 江门海事局, 广东 江门 529040)

摘 要: 采用 NetGraph 内核组件, 基于 UNIX 的 V 流思想, 设计实现了一个高性能的内容过滤网关. 网关的数据包捕获框架是由一系列 NetGraph 节点连接而成的有向图, 数据包沿着边可以在不同的节点间进行流动, 通过零拷贝技术完成在系统内核空间的报文捕获. 内容检查模块采用了改进的 AC 多模式匹配算法, 提高了数据包关键词的检查速度和系统的性能.

关键词: NetGraph; 内容过滤; 多模式匹配; 内核空间

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)08-0076-04

Design and Implementation of a High Performance Content Filtering Gateway

HE Huai-wen^{1,2}, ZHENG Qi-wen^{1,3}, FU Yu¹

(1 School of Computer, Zhongshan Institute, University of Electronic Science and Technology of China, Zhongshan 528402, China; 2 School of Information Science and Technology, Sun Yat-sen University, Guangzhou 5102753, China; 3 Jiangmen Maritime Bureau, Jiangmen 529040, China;)

Abstract: By using NetGraph kernel components and based on Unix v-flow design concept, a high-performance content filtering gateway was developed. Gateway packet capture frame is a directed graph which composed a series of NetGraph nodes connected together, packet can flow between different nodes along the graph edge, packet capture was accomplished through zero-copy technology in kernel space. Meanwhile an improved AC multi-pattern matching algorithm is used to accelerate the speed of keyword inspection and system performance in content inspection module.

Key words: Netgraph; content filtering; multi-pattern matching; kernel space

1 引言

如何在高速、海量的网络流量中快速识别、过滤和阻断有害信息的传播, 成为当前网络安全研究的热点问题. 内容级别的过滤技术正得到越来越广泛的应用, 由于内容过滤对实时性要求较高, 对流经的数据包必须在较短时间内做出处理, 否则会造成网络拥塞^[1].

网关性能主要取决于高效报文捕获技术和内容检测算法, 本文提出的容过滤网关基于 NetGraph

框架, 在系统内核级别完成报文捕获, 数据包处理采用零拷贝技术, 不存在复制操作, NetGraph 框架还具有面向对象设计特性, 易于修改和扩展. 同时本文还通过对 AC 多模式匹配算法进行改进, 提高关键词的检查速度, 使系统具有速度快、灵活性高和处理效率高等特点.

2 NetGraph 介绍

NetGraph (Graph Based Kernel Networking Subsystem, 基于图的内核网络子系统) 是一个遵循

收稿日期: 2013-10-23; 修回日期: 2013-12-17

基金项目: 国家自然科学基金项目(61300095); 广东省自然科学基金(S2012040011123)

UNIX 规范的内核子系统, NetGraph 采用 System V 的流思想, 采用“图”概念以及面向对象设计思想, 为网络应用开发提供了强大和灵活的方式^[2]. 其中, 麒麟操作系统中的多网卡 bonding 技术, 利用了 Netgraph 来解决多服务器的通信瓶颈问题^[3].

NetGraph 由一系列的节点和边构成, 数据沿着边在节点之间流动, 构成一个有向图^[1]. 图 1 便是使用 NetGraph 节点来构建的一个网络有向图, 边的集合为 {E1, E2, ..., E8}, 定点的集合为 {Node1, Node2, ..., Node6}. 在图 1 中, 数据流源点为内核模块 A, 首先传递到 NetGraph 的 Node1 节点, 在根据节点和节点之间的定义的传递规则, 在 NetGraph 模块中进行传递, 最后由 Node6 节点传递到内核模块 B.

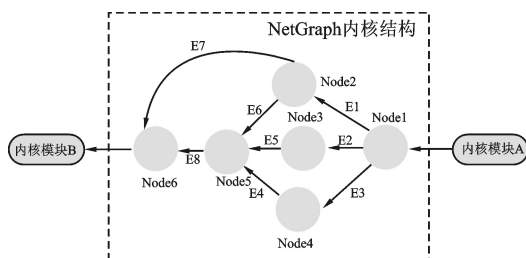


图 1 NetGraph 的内核结构图

NetGraph 的基本要素: 节点边和钩子. 节点是预先定义好的节点类型的实例, 具有面向对象特性. 每个节点有若干个钩子, 如果需要在两个节点之间进行数据通信, 可以将两者的钩子钩起来构成钩子对. 两个节点之间的钩子对形成边. 数据流沿着钩子对进行传递, 在 NetGraph 中根据需要可以创建新的节点, 并在该节点进行数据的捕获和处理.

在大流量的网络环境下, 报文捕获的效率严重影响网关的性能^[4]. 低效率的报文捕获框架容易成为整个系统的瓶颈. 其中主要影响因素有两点: (1) 在核心态多次进行报文的操作, 如头部校验等; (2) 在处理数据时网络进行多次的内存拷贝操作^[4]. NetGraph 内核模块可以直接从网卡驱动程序中获取网络数据, 避免不必要的内存拷贝. 并且, Net-graph 中报文数据在节点之间传递也采用“零拷贝”方式(只传递报文数据的指针, 没有内存拷贝操作)^[2]. 表 1 从五个方面对目前流行的报文捕获框架进行了比较. 从表 1 中可以看出, NetGraph 可以从驱动级别捕获报文, 能对链路层的报文进行处理, 支持零拷贝和实时处理, 非旁路. 因此, 采用 NetGraph 架构的报文捕获引擎, 将在大流量、高并发的网络中, 将比其他的报文捕获平台具有更好的性能和更

底层的数据处理.

表 1 流行的报文捕获技术的比较

	操作系统	获取报文级别	是否零拷贝	是否实时处理	是否旁路
NetGraph	FreeBSD	内核态(驱动级别)	Y	Y	N
NDIS	Windows	内核态(驱动级别)	N	N	Y
NPF	Windows	内核态(驱动级别)	N	N	Y
BPF	UNIX	内核态(驱动级别)	N	N	Y
NetFilter	Linux	内核态(IP 层以上)	N	Y	Y
Squid	不限	用户态(应用层)	N	N	Y

3 内容过滤引擎设计

3.1 系统设计

内容过滤网关由两部分构成: 数据包捕获模块和数据包内容检查模块. 系统架构如图 2 所示. 数据包捕获模块工作在操作系统内核空间, 提高数据包的捕获效率. 内容检查模块对报文的应用层内容进行检查, 可根据不同的需求挂载不同的过滤器, 如 HTTP 过滤器、POP/SMTP 过滤器、FTP 过滤器等. 非法关键词库保存在数据库中, 同时数据库也保存了对报文处理相关规则, 远程进行维护和管理.

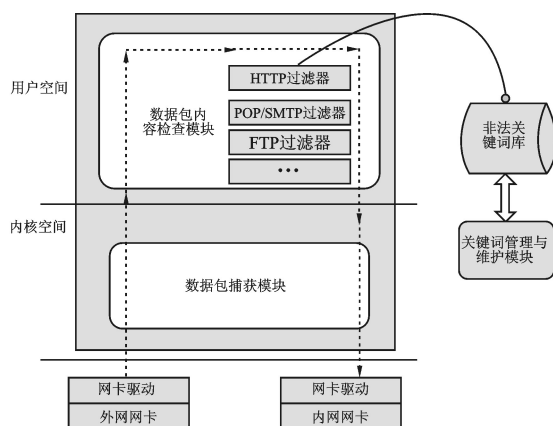


图 2 系统设计架构图

来自外网的数据包经网卡驱动解码后, 沿着协议栈向上传递, 数据包捕获模块可以在链路层进行直接捕获. 捕获模块通过 NetGraph 框架完成, 报文被捕获后直接从内核空间传递到用户空间, 避免数据的多次复制, 提高系统的性能.

3.2 报文捕获模块设计

数据包捕获按照数据流向的不同可分为两类: (1) 对内网访问外网的数据进行检查过滤, 常见的有 URL 地址过滤、IP 地址过滤等; (2) 对外网返回的数据进行内容过滤, 如网页的非法关键词检查、邮件病毒检测等. 本文系统分别设计了两个线程来完成以上的任务, 彼此工作流程类似. 以下将重点介绍如何对外网数据进行报文捕获的线程(简称线程 1).

线程的流程结构如图 3 所示.

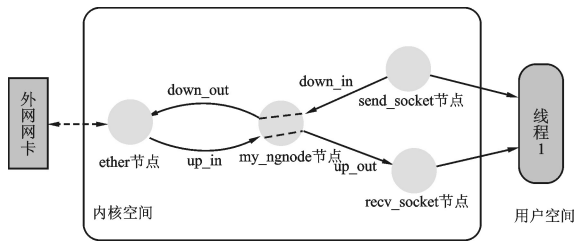


图 3 数据包捕获线程结构

在图 3 中,线程 1 负责处理来自外网发送到外网的数据包.在内核空间 Netgraph 框架中,使用了四个节点.其中 ether 节点为 ng_ether 节点类型,send_socket,recv_socket 节点为 ng_socket 节点类型,皆为边缘节点.ether 节点是内核模块和网卡硬件驱动的桥梁,接收到的 ethernet 数据帧,将封装成 mbuf 结构,放到 ether 节点的输入队列中.ng_socket 节点负责内核节点与用户进程的数据通信,功能和套接字 socket 类似^[2].

my_ngnode 节点是用于捕获数据包的节点,嵌入在 ether 节点和 socket 节点之间.my_ngnode 节点从 ether 节点接收传递过来的报文,对报文进行分析,如果报文符合继续向上转发的条件,则把报文传递到 recv_socket 节点,由 recv_socket 节点将数据包传递到用户进程.同样,my_ngnode 节点从 send_socket 节点接收报文,对报文进行内容检查,符合条件再将它转发给 ether 节点.my_ngnode 节点一共设置了 4 个钩子,如图 3 所示.其中 up_in、up_out 的钩子负责处理向上层传输的报文,up_in 用来从 ether 节点获取报文,up_out 将符合条件的报文发我 recv_socket 节点,再传递到应用程序.而 down_in、down_out 钩子负责处理下底层传递的报文,down_in 用来从 send_socket 节点获取报文,down_out 将符合条件的报文发送 ether 节点.my_ngnode 节点定义如下:

```
struct ng_node {
    char *      name;    // 结点的 ASCII 名字
    void *      private; // 结点的私有数据类型
    struct ng_type * type; //指向结点的类型信息的指针
    int         refs;    // 对该结点的引用数目
    int         numhooks; // 该结点的钩子数目
    struct ng_hook * hooks; //指向该结点的钩子链的指针
}
```

其中 my_ngnode 定义用来保存钩子的结构体 private_data 如下:

```
struct privdata {
    node_p      node;    // 结点
    struct hookinfo up_in; // up_in 钩子
    struct hookinfo up_out; // up_out 钩子
    struct hookinfo down_in; // down_in 钩子
    struct hookinfo down_out; // down_out 钩子
};
```

在 my_ngnode 节点中还定义了 my_rcvdata 函数,用来处理接收的数据,该函数流程见图 4.

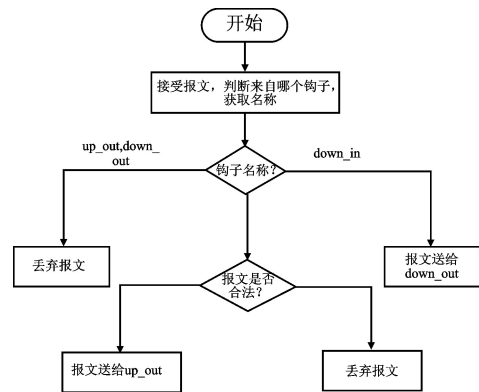


图 4 my_ngnode 节点的报文处理流程

当 my_ngnode 接收到报文之后,先判断该报文里来自于那个钩子,并获取钩子的名称.若是来自于 up_out,down_out 钩子,则丢弃报文;若是来自于 down_in 钩子,表示该报文要发往网卡,则将报文从 down_out 送出;若是来自于 up_in 钩子,则表示该报文来自于外网网卡,则需要对报文进行合法性检查(包括源地址、目的地址、端口等头部信息的检查),若合法,则送往 up_out 钩子,否则丢弃.

4 内容检查模块

4.1 多模式匹配算法

基于字符串匹配的关键词匹配算法是内容过滤系统中的重要分析方法,匹配算法严重影响系统的性能^[5].模式用 p 来表示,模式集数量用 n 来表示,模式集用 $P = \{p_1, p_2, \dots, p_n\}$ 来表示,有待搜索的文本用 T 来表示,最短模式字符串的长度用 $\minlen = \min\{m_k | 1 \leq k \leq q\}$ 来表示.多模式匹配算法的目的就是要在 T 中找到所有出现的模式 p_i .

经典的模式匹配算法有 Quick Search(QS)算法、KMP 算法、BM 算法、Aho-Corasick(AC)算法、Commentz-Walter 算法等.QS、KMP、BM 算法属于单一模式匹配算法;AC 算法利用有限状态机工作

进行查找的多模式匹配算法,但是在搜索文本时无法跳过不需要的比较^[6].本文结合内容过滤网关的特点,对经典 AC 算法进行改进,设计一种新的快速多模式匹配算法.

算法对所有待匹配的模式进行分析和处理,构造一个有限状态模式匹配自动机,并计算出失配下跳转的位置信息,保存在合适的数据结构中.然后利用这个模式匹配自动机对文本进行一次性搜索,并利用预先计算好的信息避免文本回溯,记忆实现跳跃式比较,加快搜索过程.对于固定的模式集,只需要构建一次自动机,便可对不同文本进行搜索.

算法的改进:(1)利用 KMP 的思想,若文本 t 在第 i 字符进行失配,则将有限状态匹配机的当前状态跳转到新的状态 s ,该状态表示的串和文本第 i 个字符之前最长的后缀子串相同,从而避免了文本指针回溯.(2)进利用 Wu-Manber 算法思想提出的跳跃加速.当 $P[1:m]$ 和 $T[i:i+m-1]$ 对齐进行比较,在失配的情况下,通过分析 $T[i+m]$,计算该字符在所有模式中最后出现的位置,然后移动自动机,使该字符与其在模式中最后出现的位置对齐.

例如,待匹配模式集合为 $P=\{\text{'heros'}, \text{'headset'}, \text{'shells'}\}$,可以构成图 5 中的有限匹配状态机.在有向图中,每个节点表示一个状态,表示一个模式前缀或多个模式的共同前缀,边表示读入的字符.图 5 中,状态 2 表示 'heros' 和 'headset' 的共同前缀 'he',状态 14 表示 'shells' 的前缀 'shel' 等^[7].

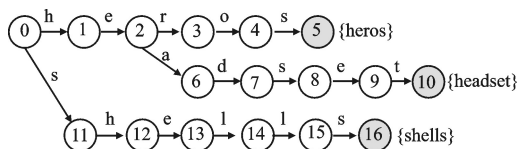


图 5 有限状态匹配机

假设需要匹配的文本为 'hexzshellsb', 模式集为 $P=\{\text{'heros'}, \text{'headset'}, \text{'shells'}\}$. 如图 6 所示,从状态 0 开始读入文本 'h', 进入状态 1, 继续读入字符 'e', 进入状态 2. 此时扫描指针指向第三个字符 'x', 失配情况出现. 根据改进措施, 文本指针无须回溯, 以最短模式为标准向前看一个字符 (第 5 个字符), 将自动机移动到和第 5 个字符对齐, 即与模式 'shells' 进行比较, 如图 所示, 进入状态 11, 继续比较, 直到到达状态 16, 匹配成功.

4.2 算法描述与结果测试

算法的关键是在于预先计算失配后的文本指针位置和有限状态机跳转的位置. 本文使用一个二维表 shift 表来保存相关信息, 表示为 $\text{shift}[\text{当前状}$

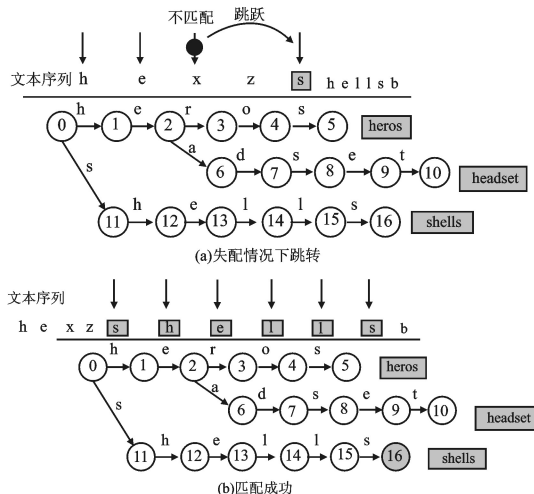


图 6 多模式匹配过程

态][向前看的字符] = {文本跳跃的位置, 新的状态}. 例如, $\text{shift}[3][\text{'s'}] = (2, 8)$, 表示如果在状态 6 失配, 且向前看的字符为 's' 时, 则文本扫描指针向前跳跃 2 个位置, 并从对齐状态 8 开始继续比较.

跳转表的方法如下: 先计算每个字符 c 的最后出现位置 $\text{last}[c]$, 以及在每个状态 s 失配后, 若不考虑向前看字符, 应转入的状态 $\text{next}[s]$, $\text{next}[s]$ 借助队列数据结构来计算, 然后根据 $\text{last}[c]$ 计算 jump , 若 jump 为负数, 表示会出现文本指针回溯情况, 则指针不跳跃, 转入状态 $\text{next}[s]$, 避免回溯.

其中, 文本匹配过程的详细算法请参考文献 [5]. 本文对单模式匹配算法、AC 算法和本文改进的算法进行了测试对比, 结果如图 7 所示.

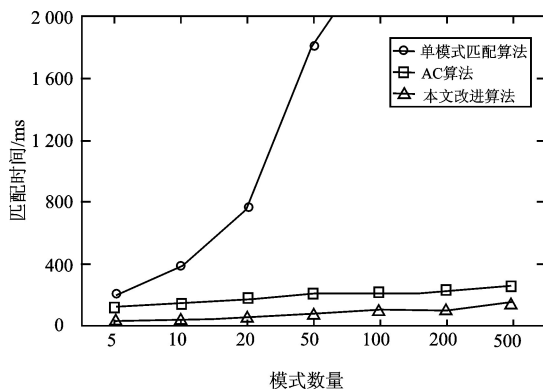


图 7 测试结果图

在图 7 中, 可以看到单模式匹配算法由于要对每个模式进行逐一检查, 匹配时间随着模式数量的增加呈线性递增; 而多模式匹配的 AC 算法和本文算法在模式数量的增加下表现良好, 匹配时间增长平缓. 另外, AC 算法和本文算法都存在预处理时

(下转第 84 页)

有明显差距,MAX 的最低;时延随 T_{PS} 增加近似呈线性增大趋势,可认为取较小的 T_{PS} 取有益于提升吞吐量。

6 结束语

文中提出了面向规则拓扑结构的媒体接入调度方法,分析、比较了对应的开销和理论最大吞吐量,能为高效的部署认知无线传感网提供方法参考。其它值得进一步探讨的问题包括:面对信道之间的特性差异和信道间的干扰,如何为 CU 选择具体的信道;针对具有双工及多工通信能力的节点,如何设计调度方法以进一步提升性能;采用层次化结构的规则拓扑 CRSN 的时隙和信道分配问题。

参考文献:

- [1] De Domenico A, Strinati E C, Di Benedetto M. A survey on MAC strategies for cognitive radio networks [J]. IEEE Communications Surveys and Tutorials, 2012, 14(1): 21-44.
- [2] Vijay G i, Ben Ali Bdira E, Ibnkahla M. Cognition in wireless sensor networks: a perspective [J]. IEEE Sensors Journal, 2011, 11(3): 582-592.
- [3] Yun Z Q, Bai X L, Xuan D. Complete optimal deployment patterns for full-coverage and k-connectivity ($k < 6$) wireless sensor networks [J]. IEEE/ACM Transactions on Networking, 2010, 18(3): 934-947.
- [4] Hong K Z, Hua Y B. Throughput analysis of large

wireless networks with regular topologies [J]. Eurasip Journal on Wireless Communications and Networking, 2007(1):3.

- [5] Wang Y, Dharma P A. Optimizing sensor networks for autonomous unmanned ground vehicles [C] // SPIE. Cardiff, UK: SPIE Press, 2008(711215)1-11.
- [6] CicconeIII C, Gardellin V, Lenzini L, et al. PaMeLA: A joint channel assignment and routing algorithm for multi-radio multi-channel wireless mesh networks with grid topology [C] // Proc. of 6th IEEE International Conference on Mobile Ad hoc and Sensor Systems. China:Macau, 2009: 199-207.
- [7] 阙保强, 范建华, 王建业. 认知无线网络信道接入协议 [J]. 软件学报, 2012, 23(7): 1824-1837.
- [8] Satish C Jha, Mohammad M Rashid, Vijay K Bhargava. Medium access control in distributed cognitive radio networks [J]. IEEE Wireless Communications, 2011, 18(4): 41-51.
- [9] 李云, 王学军. 基于同构型环境的认知无线网络拥塞控制算法 [J]. 微电子学与计算机, 2013, 30(10): 13-16.

作者简介:

李文翔 男,(1979-), 博士, 副教授. 研究方向为认知无线网。

张峰 男,(1990-), 硕士研究生. 研究方向为无线传感网。

马娅婕 女,(1974-), 博士, 教授. 研究方向为无线传感网。

(上接第 79 页)

间,本文算法预处理时间比 AC 算法多,是由于本文改进预处理阶段所做的额外工作造成。由于本文算法对 AC 算法的改进,可看到本文算法的匹配时间明显小于 AC 算法,使系统的性能得到大大提高。

5 结束语

本文从内容过滤网关的架构设计和多模式匹配算法进行分析,采用 Netgraph 内核级的数据包捕获引擎和改进的多模式匹配优化算法相结合,大大提高了网关对报文的处理速度和内容过滤的性能,同时在实际产品中也取得了良好的应用,将具有广阔的前景。

参考文献:

- [1] 沈凤仙, 朱巧明, 刘钊. 基于 Netfilter 的实时网页内容过滤器的设计与实现 [J]. 微电子学与计算机, 2009, 26(7): 198-201.

- [2] netgraph [EB/OL]. [2013-10-23]. <http://www.freebsd.org/cgi/man.cgi>.

- [3] 石磊. 多网卡绑定技术的研究与实现 [D]. 长沙: 国防科学技术大学, 2005.
- [4] 王佰玲, 方滨兴, 云晓春. 零拷贝报文捕获平台的研究与实现 [J]. 计算机学报, 2005, 28(1): 46-52.
- [5] 王若梅, 张绮雯, 周凡. 一种新的多模式快速匹配算法 [J]. 中山大学学报: 自然科学版, 2005(11): 117-120.
- [6] Aho A V, Corasick M J. Efficient string matching: an aid to bibliographic search [J]. Communications of the ACM, 1975, 18(6): 333-340.
- [7] 孙钦东, 黄新波, 王倩. 面向中英文混合环境的多模式匹配算法 [J]. 软件学报, 2008, 19(3): 674-686.

作者简介:

何怀文 男,(1980-), 讲师, 博士研究生. 研究方向为云计算、网络安全、绿色计算。