文章编号:1001 - 9081(2005)10 - 2283 - 03

基于网络处理器的内容过滤的实现

陈洪涛,陈德人,顾学飞
(浙江大学 计算机科学与技术学院,浙江 杭州 310027)
(cht_zju@163.com)

摘 要:内容过滤是网络安全领域中的一个重要组成部分。它对应用层内容协议中传输的信息进行分析,并根据过滤条件控制信息的转发。网络处理器是新一代用来执行数据处理和转发的高速可编程处理器。它以其在网络数据处理方面的明显优势,成为高速网络设备中支持业务管理、安全与网络监控等网络功能必不可少的元件。根据网络处理器的优势,给出了一个基于 Intel IXP2400 网络处理器内容过滤系统的实现。

关键词:内容过滤;IXP2400;网络处理器;网络安全
中图分类号: TP393.08 **文献标识码**:A

Implementation of network processor-based content filtering

CHEN Hong-tao, CHEN De-ren, GU Xue-fei
(*College of Computer Science and Technology, Zhejiang University, Hangzhou Zhejiang* 310027, *China*)

Abstract: Content filtering is an indispensable important part in the network security field. It analyses the information transported in application layer content protocol and controls the forwarding of the information based on the filtering rules. Network Processor is a new generation programmable processor of high speed for carrying out data processing and transmitting. With its obvious advantages in network data processing, the network processor becomes the essential component in high-speed network equipment which supports network functions, such as operation management、security and network monitoring, etc. Based on the advantage of network processor, this paper presented an implementation of an Intel IXP2400 network processor-based content filtering system.

Key words: content filtering; IXP2400; network processor; network security

1 内容过滤

内容过滤系统对应用层内容协议(如 HTTP、SMTP、POP3、IMAP 等)中传输的信息内容进行分析,并根据预先设置的过滤条件,控制信息的下一步传送方向。

内容过滤主要有两种实现形式:包含过滤(inclusion filtering)和排除过滤(exclusion filtering)。包含过滤通常也称为“白名单”(white list),只有在此名单中的信息才能被访问。这样做的好处是安全性高,缺点就是“白名单”可能会很长,维护的代价高。目前比较常用的是排除过滤,通常也称为“黑名单”(black list),它的做法是将影响到网络安全的信息加入黑名单,使其不能被其他网络用户访问,对于那些不在黑名单中的信息都可以被访问到。很明显,这个黑名单将小得多,不过它的缺点就是它并不是十分安全的,需要对黑名单不断更新以保证其安全性。

目前,对网络上的内容进行过滤主要有如下五种方法:

1) 关键字过滤

关键字过滤就是对信息的内容进行关键字匹配,通常用黑名单来实现。只要站点包含有与关键字相匹配的信息,它就会被禁止访问。它的优点是实现比较方便,缺点就是只能检查文字信息,不能检查图像,而且不能联系语境分析,会误判好的信息。

2) 包过滤

网络上的内容信息是以包(packet)进行传送的。每个包都有一个源 IP 地址和一个目的 IP 地址,包过滤可以检查包的 IP 地址来过滤信息内容。包过滤都是在路由器上完成的,它的主要缺点在于精度过粗,IP 地址和内容并不是一一对应关系,因此,包过滤往往会对合法内容提供商造成误判。

3) URL 过滤

这是目前最常见的一种过滤方式,因为 URL 对应的是具体的网页而不是网页所在的服务器,克服了前一种方法的缺点,大大提高了过滤的准确性。一个不良网站可能包含成千上万的不良网页,如果采用完整的 URL 精确到每个网页,势必会导致名单过大,严重影响设备性能。由于 URL 采取从左到右的层次化结构,内容过滤系统可以根据 URL 的一部分进行过滤。

4) 模板过滤

根据源内容的特征对其进行分析,将安全的内容信息呈现给用户,对于那些不安全的信息则阻挡。模板过滤通常跟 URL 过滤结合起来使用。

5) 图像分析过滤

这是最近才出现的一种过滤方法,利用对图像进行分析和精确的计算,来区分色情和艺术,将好的内容信息呈现给用户。它也经常和 URL 过滤一起使用。

收稿日期:2005 - 04 - 13

作者简介:陈洪涛(1982 -),男,湖北沙市人,硕士研究生,主要研究方向:嵌入式应用、网络安全; 陈德人(1951 -),男,浙江杭州人,教授,博士生导师,主要研究方向:电子商务、计算机图形学、人工智能软件; 顾学飞(1980 -),男,江苏南京人,硕士研究生,主要研究方向:嵌入式应用、网络安全.

目前模板过滤和图像分析过滤还在研究之中,应用还比较少,大多数的内容过滤产品采用关键字过滤和 URL 过滤技术。然而,不管采用何种过滤技术,它对于不良内容的反应必须迅速,这也是目前内容过滤的瓶颈。为了解决这个问题,我们采用了网络处理器。

2 网络处理器

网络处理器(Networks Processor,NP)是为网络应用领域设计的专用指令处理器,具有自己的结构特征和专门的电路设计以适于网络分组处理。它具有软件可编程能力,是对分组处理流程的优化,以满足线速处理要求,它可以接管很多原来主 CPU 完成的控制与管理功能。

Intel IXP2400 网络处理器是 Intel 的第二代网络处理器系列中的一种,它具有完全可编程的特性,能灵活的应用于多种网络处理功能,和第一代网络处理器相比,它能更好地满足网络处理的高速化、复杂化的需要。IXP2400 的硬件结构如图 1 所示。

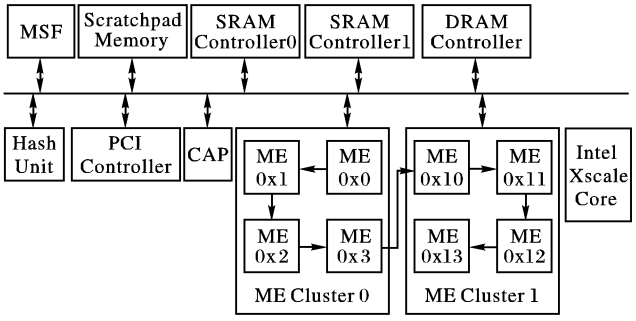

图 1 IXP2400 硬件结构

1) Intel Xscale Core:它是数据和指令高速缓存的嵌入式处理器,遵从 ARM 标准,时钟频率为 600MHz,主要负责处理网络处理器中控制通道任务,执行芯片的初始化配置、系统控制和管理、运行路由协议栈、更新路由表等操作。另外,Intel Xscale Core 还负责对异常数据包进行处理。

2) ME(Microengine):微引擎,ME 是 IXP2400 的核心部件,也是 IXP2400 取得线速处理性能的关键所在,负责绝大部分的数据包处理任务。ME 能访问 IXP2400 中所有共享资源。IXP2400 中有 8 个 ME,分为两组:ME Cluster0 和 ME Cluster1,每个 ME 有 8 个硬件线程。

3) SRAM 控制器:用于接口 SRAM 存储设备,控制、管理 IXP2400 中其他功能单元对 SRAM 存储设备的访问、操作。

4) DRAM 控制器:用于接口 DRAM 存储设备,控制、管理 IXP2400 中其他功能单元对 DRAM 存储设备的访问、操作。DRAM 存储设备的最大存储空间为 2GB,用于存储数据包、路由表等大型的数据结构。

5) MSF 接口:介质和交换结构接口,是 IXP2400 与外部物理层设备、交换结构的接口单元,也是 IXP2400 接收、发送数据包的窗口。

6) PCI 控制器:符合 PCIv2.2 规范,接口总线宽度为 64bit,时钟频率为 66MHz。允许 IXP2400 和一台主机或者使用 PCI 接口的设备通信。

7) SHaC 单元:包括 Scratchpad Memory、Hash Unit 和 CAP(控制状态寄存器访问代理)三部分。其中,Scratchpad Memory 容量为 16KB,用于 ME 之间的通信以及重要数据的内部缓存;Hash Unit 支持 48bit、64bit、128bit 的哈希运算;CAP 用于对 IXP2400 中的控制、状态寄存器进行访问、操作,用于设定 IXP2400 的工作模式和采集运行状态。

由于采用了完全可编程的微引擎进行网络处理,使得 IXP2400 具有灵活的处理功能;由于采用了多微引擎并行处理、分布式处理、多线程等硬件结构和专门的软件技术,使得 IXP2400 具有强大的处理性能。因此,我们采用 IXP2400 来开发高性能的内容过滤系统。

3 基于 IXP2400 的内容过滤设计

3.1 硬件结构设计

系统硬件结构如图 2 所示。核心是一个 IXP2400 的芯片,负责 Ingress、Egress 两个方向的数据包处理。QDR SRAM 是 IXP2400 外接的存储设备,通过 SRAM 控制器与 IXP2400 连接,提供超过 12Gbps 的读吞吐量和 12Gbps 的写吞吐量,主要用于存储数据包处理过程中用到的发送队列、查找表等数据结构。DDR DRAM 通过 DRAM 控制器与 IXP2400,它也是外接的存储设备,可以寻址 2GB 空间,具有 19.2Gbps 的吞吐量,主要用于存储数据包。IXP2400 通过 PCI 控制器与主机相连,主机给 IXP2400 提供电源。系统还有两个千兆光纤端口通过 MSF 与 IXP2400 相连。

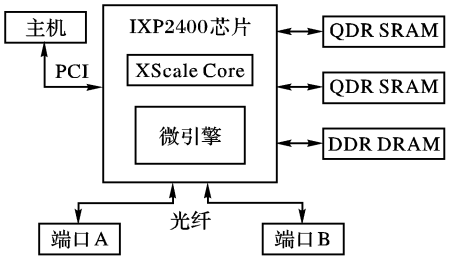

图 2 基于 IXP2400 的内容过滤硬件结构

3.2 软件结构设计

基于 IXP2400 网络处理器的内容过滤系统软件结构包括数据通道和控制通道两部分。其中,数据通道又称 fast path,运行在微引擎中上,有若干个 microblock 组成,是数据转发的直接通道,它具有极高的处理效率。大部分的数据包处理都是在此通道上完成的。控制通道又称为 slow path,运行在 Intel Xscale Core 上,由若干个 core component 组成,一般情况下,一个 microblock 对应一个 core component,是业务控制信息处理及其他无实时性要求的数据通道,它的功能复杂。但处理速度相对较慢,不适合实时业务的处理。

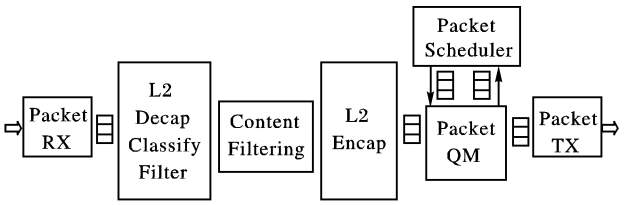

图 3 基于 IXP2400 的内容过滤软件模块

图 3 列出了内容过滤系统的主要软件模块。Packet RX,Packet Scheduler,Packet QM,Packet TX 各自占据一个单独的微引擎,在这些微引擎上不需要使用 dispacth loop。Dispacth loop 用于将多个 microblock 有机地组合成为一个 microblock group 运行在一个微引擎上面,实现特定地数据包处理功能。

主要的功能单元映射到剩下的四个微引擎上,每个微引擎执行相同的代码。每个微引擎内部主要执行数据包解封装、数据包分类,内容过滤,数据包封装操作,此时需要使用 dispatch loop 来控制数据包在这几个 microblock 间的执行顺序。

下面详细介绍每个模块的功能:

1) Packet Rx 模块。负责接收来自 Ingress 的数据包分片(mpacket),并将 mpacket 重组得到完整的数据包(packet)。重组时,Packet Rx 模块通过将 mpacket 顺序存储在 DRAM Buffer 中得到完整的 Packet。同时,创建 Buffer Handler(包含 packet 在 DRAM 中的存放地址)以及 metadata(包含 packet 的描述信息)存储在 SRAM 中。最后,Packet Rx 模块将 Buffer Handler、metadata 通过 Scratch Ring 传递给下一级的数据包处理模块。

2) L2 Decap/Classification 模块。通过修改 packet metadata 的 offset 和 size 字段将 2 层数据包头去掉,然后 classification 子模块执行分类操作,将数据包分为 IPv4、IPv6、MPLS 等类型,然后传递给下一个模块进行处理。

3) Content Filtering 模块。本系统采用排除过滤方式,在 Content Filtering Core Component 中定义了一些过滤规则,这些规则可以是病毒的特征码等,并实现了相应的匹配算法。Content Filtering 模块分析包的内容,调用匹配算法与定制的规则一条条进行匹配,一旦匹配成功,则将这个数据包设为 IX_DROP,然后处理下一个数据包。这些被标记为 IX_DROP 的数据包将由 Packet QM 模块处理。

4) L2 Encap 模块。给数据包加上 Ethernet 头部信息。它首先在 L2 表里面查找,如果找不到该数据包的 2 层信息,则该数据包交给 Core Component 处理。

5) Packet QM(Queue Manager)模块。利用 SRAM 控制器中的 Q-Array 硬件结构对发送队列执行 Enqueue、Dequeue 操作。它通过 Scratch Ring 从 L2 Encap 模块接收 Enqueue 请求;通过另一个 Scratch Ring 从 Packet Scheduler 模块接收 Dequeue 请求。Enqueue、Dequeue 操作以 Packet 为单位。QM 通过 Next Neighbor Ring 将 Queue 的状态信息发送给 Packet Scheduler 模块。

6) Packet Scheduler 模块。负责将 Ethernet 数据包调度到发送队列。在不同的端口之间,Scheduler 采用 WRR(Weighed Round Robin)调度算法,在同一个端口的不同发送队列间,Scheduler 采用 DRR(Deficit Round Robin)调度算法。

7) Packet Tx 模块。负责将 Packet 发送到千兆以太网接口,它接收来自 QM 模块并经过 Dequeue 操作取出的一个 Packet,然后将其拆分为多个 mpacket,然后将 mapcket 置入 TBUF 中发送出去。

3.3 Content Filtering 模块设计

Content Filtering 模块对流进来的数据包进行规则匹配,一旦匹配成功,就将数据包标记为 IX_DROP,交给 QM 处理。在此过程中,如果遇到异常数据包,Content Filtering 模块就把数据包交给 Content Filtering Core Component 处理。同时,Content Filtering Core Component 还提供了对 Content Filtering 模块规则的编辑、解析功能,维护存在 SRAM 中的规则匹配表。

本系统目前只支持 IPv4,因此对于其他类型的包,如 IPv6、MPLS,都直接转发;对于 icmp 和 arp 包,则标记为异常包,给 core component 处理。实现了最常见的 URL 过滤,其他过滤方法留待以后扩展。

模块以流水线形式实现,运行在 4 个微引擎上,微引擎上的每个线程每次处理一个 packet,数据包的顺序由 dl_source 模块和 dl_sink 模块来保证。Content Filtering 模块与其他各模块的关系如图 4。

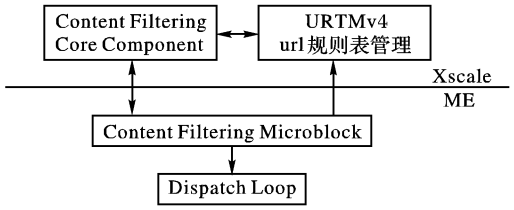

图 4　Content Filtering 模块与其他模块的关系

Dispatch loop 模块将 dl_source, dl_sink, l2_decap, l2_encap, content filtering 等模块连接起来,共同完成内容过滤的功能。Core Component 对 content filtering 模块进行配置,处理异常及统计信息,URTMv4 维护 url 规则表。

URL 规则表在 SRAM 中顺序存储,每条规则代表一个 URL 地址,被分解为两部分:32 位的 urlHost 和 32 位的 urlCatlog。其数据结构如下:

```
struct ix_cc_content_filtering_rule_url_table
{
 //url 地址代表的主机名
 ix_uint32 urlHost;
 //url 地址的扩展
 ix_uint32 urlCatlog;
};
```

我们定义了两个符号:CONTENT_FILTERING_RULE_URL_TABLE_SRAM_BASE,代表 URL 规则表在 SRAM 中的基地址;CONTENT_FILTERING_RULE_URL_TABLE_SIZE,代表 URL 规则表的长度,在初始化的时候,core component 要将这两个符号的值指派(Patch)给相应的微引擎。

我们还定义了一些计数器,用来统计数据包的一些信息。在 SRAM 中采用层次结构来存储所有的计数器,Xscale 负责维护 64 位的计数器,ME 负责维护 32 位的计数器。

Content Filtering 模块的处理流程:进入该模块,先判断 dl_next_block 是不是为 BID_CFURL(url 内容过滤模块),不是该模块就执行结束,否则就将接收数据包计数器加 1,然后判断该包是否是 IPv4 数据包,不是则交给 core component 处理,是就执行匹配,匹配成功则将匹配包计数器加 1,然后把 dl_next_block 设为 IX_DROP,等到和所有规则都匹配完之后,就将不匹配计数器加 1,在将 dl_next_block 设为 BID_NEXT_BLOCK,整个模块就执行完毕。

4　结语

根据网络处理器的优势,给出了一个基于 Intel IXP2400 网络处理器内容过滤系统的实现。这个系统在技术上还可以有继续完善的地方,比如如何进一步地提高匹配效率,扩展其他匹配方法等。

参考文献:

[1] JOHNSON E, KUNZE A. IXP2400/2800 Programming [M]. USA: Intel Press, 2003.

[2] CARLSON B. Intel ® Internet Exchange Architecture and Applications [M]. USA: Intel Press, 2003.

[3] 张宏科, 苏伟, 武勇. 网络处理器原理与技术[M]. 北京: 北京邮电大学出版社, 2004.

[4] GREENFIELD P, MCCREA P, RAN S. Access Prevention Techniques for Internet Content Filtering [R]. Australia: the Commonwealth Science and Industry Research Organization (CSIRO), 1999.

[5] 朱骏, 陈刚. 一种高效的智能内容过滤模型[J]. 计算机工程 2003, 29(21): 146 - 148.