

基于 Oction 多核网络处理器的 IPv6 联动 IPS 研究与设计

杨吉喆, 王玲玲, 陆建德

(苏州大学计算机学院 江苏省计算机信息处理技术重点实验室, 江苏 苏州 215006)

摘 要: 对基于 Oction 多核网络处理器的新一代 IPv6 高速网络联动入侵防御系统进行研究, 设计了新型联动入侵防御原型. 系统基于 Oction 多核的高速处理, 并结合了 IPv6 网络中入侵的新特点. 在基于入侵检测规则库规则匹配技术的基础上, 运用新型的协议分析技术和基于流的检测技术, 在 Oction 多核间分配控制层与数据层的不同执行, 采用命名块机制进行多核间通信, 通过数据层核向控制层核的反馈, 实现了流处理及协议分析模块与控制模块的高速联动. 系统实现了 Gbps 级的高速入侵检测与联动防御处理.

关键词: Oction; 多核; IPv6; IPS; 联动

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2011)12-0079-05

Research and Design of Linked IPS Based on Oction Multi-core Network Processor for IPv6

YANG Ji-zhe, WANG Ling-ling, LU Jian-de

(School of Computer of Soochow University, Jiangsu Province Computer IT Key Lab, Suzhou 215006, China)

Abstract: The paper has made research to the Linked Intrusion Prevention System based on Oction multi-core network processor for new generation high-speed IPv6 network, and designed a new type of prototype. The system design is based on high-speed processing on Oction multi-core, and combines new intrusion characteristics occurred in IPv6 network. On the basis of the technique of matching rules in rule library for intrusion detection, and using the new protocol analysis and flow-based detection techniques, the different executions including control plane and data plane are distributed on multiple cores of Oction. Adopting the mechanism of named blocks to communicate between multiple cores, and by means of the feedbacks from the cores running data plane code to the control plane core, the system has realized the high-speed linking between the flow processing, protocol analysis module and the control module, which is competent for the high-speed intrusion detection and linked prevention at Gbps level.

Key words: Oction; multi-core; IPv6; IPS; linking

1 引言

新一代网络带宽越来越高, 通信具有更小延迟和更大吞吐量, 分布于骨干网上的传统网络入侵检测系统(NIDS)与网络入侵防御系统(NIPS)的处理能力越来越难以跟上现代网络的速度, 单 CPU 平台已成为 IDS 与 IPS 处理的瓶颈. 需要研究在网络多核处理环境下提高 NIDS 与 NIPS 的处理能力.

同时, 新一代网络 IPv6 在安全性方面也面临许

多新威胁, 因此研究高速网络 IPv6 环境^[1]下的网络入侵检测与防御系统具有重要意义.

本文研究基于 Oction 多核网络处理器的新一代 IPv6 高速网络联动入侵防御系统的体系架构设计中的若干关键技术, 研究在 HOST+Core 架构下的 Oction 多核上的并行流处理技术、既协作又分工的并行的协议分析检测技术、IPS 系统的联动体系架构的设计技术.

收稿日期: 2011-03-10; 修回日期: 2011-05-11

基金项目: 国家自然科学基金(61070170); 江苏省高校自然科学基金项目(08KJB520011)

2 系统总体设计

系统的软件设计从逻辑上可以分为 Host 层、Core 层以及 PCI bus 层. 系统分层是为了最大程度的软件重用以及系统的可移植性, 使得系统不仅可以运行在 HOST+Octeon 多核处理器板卡环境中, 还可以运行于独立的 Octeon 多核网络处理器板卡中(Stand Alone 形式). 系统总体架构如图 1 所示.

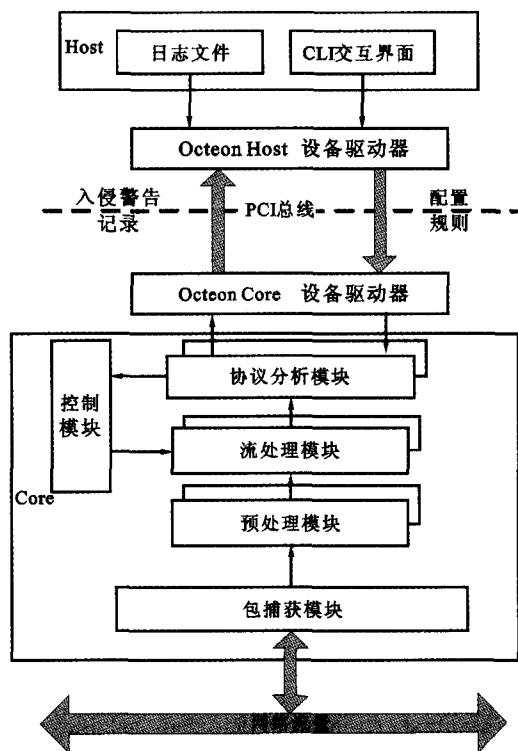


图 1 系统总体架构

Host 层包括日志文件、CLI 交互界面. 日志文件主要记录由协议分析模块发送来的报警信息, 以便对入侵规则库的管理和维护. 系统设计了仿路由器、交换机配置方式的 CLI 用户交互模式^[2], CLI 交互界面根据日志文件中的报警信息通过 PCI 总线对 Core 层的协议分析模块中的规则库进行管理和配置, 按照日志文件中报警信息的等级重新调整规则库中规则的位置, 使得频繁检测到的规则放到规则库的前面, 以便提高匹配效率.

Core 层主要由包捕获模块、预处理模块、流处理模块、协议分析模块和控制模块组成. 各模块分工如下所示.

包捕获模块: 采用多核网络处理器 Octeon 底层硬件数据包捕获接口, 与传统的 Linux 下的 libpcap 相比, 具有硬件加速的功能.

预处理模块: 对捕获到的数据包, 如果是分片包, 首先进行分片重组. 如果数据包长度 (除了 ICMPv6 包和最后一个分片包) 小于 1280 字节, 直接丢弃. 对 IPSec 协议加密过的数据包, 不进行检测^[3-4].

流处理模块: 关于流表的处理, 在 IPv6 网络环境下, 用三元组(源地址、目的地址、流标签)就可以标识一条流. 如果流标签字段不为零, 根据三元组查询流表, 并进行相应的处理; 如果流标签字段为零, 直接进入协议分析模块.

协议分析模块: 对在流表中查找失败的数据包, 根据 IPv6 协议的特点进行协议分析, 通过基本首部中的下一首部值来确定其扩展首部的类型, 直至应用层, 将分解出的数据部分进行规则匹配, 匹配的规则库采用的是优化了的 snort 规则库, 并根据匹配结果, 反馈控制信息写入到前面的流表中.

控制模块: 负责系统的初始化和流表的维护工作. 通过事件驱动循环的方式查询协议分析模块的反馈控制信息, 并相应的修改流表. 并定时查询流表, 删除流表中生存期结束的流表项.

3 系统中的关键技术

3.1 基于流的检测技术

随着 IPv6 新一代网络速度的提高, 传统的基于包检测方式的 NIPS^[2] 变得很困难. 本文提出基于流的检测技术. “流”在 IPv6 中的定义是: 流是从源地址到单播、任播或者组播地址的一系列包, 源地址希望通过流标签来区分流^[5]. 在 IPv4 网络中, 可以用源 IP 地址、目标 IP 地址、协议、源端口和目标端口五元组来区分流, 属于同一个流的数据分组的五个字段的值相同. 通过检查五个字段值才能唯一地确定一个流, 而 IPv4 网络中对流的处理分布在网络层和传输层, 这会使处理效率降低. 在 IPv6 中, 为了支持服务质量, 在数据报头中保留了 20 位流标签(flow label)位. 流标签与源地址和目的地址结合唯一地确定了一个数据流. 因此 IPv6 采用流标签可以更加高效的处理数据分组流. 考虑以上因素, 本系统通过在共享内存中维护一张流表, 根据流表中已有的行为直接对后续的流进行相应的处理, 从而不必再进行后续的协议分析, 提高了检测的效率. 它的数据结构如下:

```
struct ips_flow_table {
    uint128_t src_ip; /* 源 IP 地址 */
    uint128_t dst_ip; /* 目的 IP 地址 */
}
```

```
uint32_t label;20; /* 流标签值 */
uint32_t hash_id;16; /* hash index */
ips_flow_statistics_t statistics; /* 流的统计信息 */
uint8_t l4_stage; /* 四层解析的阶段,用于存放四
层解析的状态值 */
uint8_t l4_index; /* 四层匹配的结果 */
uint8_t l7_stage; /* 七层解析的阶段,用于存放七
层解析的状态值 */
uint8_t l7_index; /* 七层匹配结果 */
uint8_t packet_buf; /* 包存放的位置 */
ips_flow_action_common_t action; /* 流行为 */
uint64_t expire_time; /* 流生存时间 */
};
```

其中 hash_value 字段用来作为流表中的查找关键字,它是源地址、目的地址和流标签三元组的哈希值。Statistics 是用来存储所需统计的服务器的链接数以及每一个连接的传输容量等信息。由于流表不可能无限增长,根据 RFC 3697,一条流的生存时间大于 120 s。系统用 expire_time 字段记录每一条流的生存时间,默认流的生存期为 2 分钟。流表中的 action 包括以下三种状态:(1) Forward:直接转发此包;(2) Drop:直接丢弃;(3) Divert:送到协议分析模块。

由于流表不可能无限增长,根据 RFC3697,一条流的生存时间大于 120 s。系统用 expire_time 字段记录每一条流的生存时间。系统对流表的维护策略是:对于一条流生存时间结束或者在默认时间内老化后,删除此流状态,放弃对其策略的维护。

3.2 协议分析技术

协议分析技术充分利用网络协议的高度有序性,结合高速数据包捕捉、协议分析和命令解析,来快速检测某个攻击特征是否存在。协议分析的最大特点是将捕获的数据包从网络层一直分析到应用层,将真实数据还原出来,然后将还原出来的数据再与规则库进行匹配,因此能通过对数据包进行结构化协议分析来识别入侵企图和行为。协议分析大大减少了计算量,即使在高负载的高速网络上,也能逐个分析所有的数据包。IPv6 基本首部固定为 40 字节^[6],其中的下一首部字段值标识下一个扩展首部的类型。因此 IPv6 协议具有固定的格式,满足了协议分析的要求。协议分析技术通过分析 IPv6 数据包,跳到感兴趣的字段来进行分析,提高了单纯的模式匹配的效率。针对受保护网段内的目的地址提出的连接请求数及流量限制,将协议分析出的流特征与默认的阈值进行比较,可以有效地防止 DoS 攻

击。IPv6 扩展头分析分析过程见图 2。

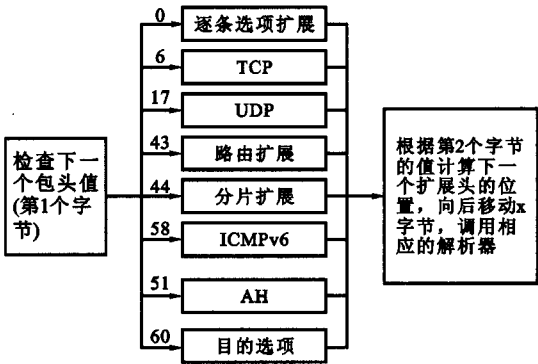


图 2 IPv6 扩展头分析过程

根据 IPv6 协议的特点,对在流表中匹配失效的包,进入到协议分析模块。首先提取基本首部的前四位,检查是否为 6,只处理版本号为 6 的包,然后跳到第 7 字节提取下一报头值,根据该值确定下一个头类型,如果下一个报头值为 6 或者 17,说明该包没有扩展首部,直接进入传输层进行解析,否则跳到包头的第 41 字节对扩展头进行分析。在扩展头中,第 1 字节代表该扩展头的下一报头,第 2 字节代表该扩展头的长度,所以在获得这两个值后,就可以跳过该扩展头,顺利进入下一个头的分析。对于扩展头中未使用的选项字段,为了防止被攻击者利用,系统直接将其置为零。对 TCP 头、UDP 头等解析与 IPv4 中的情况相似;对于扩展头的计算,我们设扩展头长度字段值转换为十进制后为 n ,则移动到下一扩展头的字节数 x 为:(1)当扩展报头为逐跳选项、目的地选项和路由头时, $x=(n+1)*8-2$;(2)当该扩展头为分段头时,不用再检查扩展头长度, $x=8$;(3)当该扩展头为认证头时, $x=n*4+6$ 。这样经过层层解析后,得到该数据包的应用层协议,然后送往相应的 HTTP, FTP, SMTP 等应用层协议处理部分进行解析,,综合运用协议分析技术和模式匹配技术检测数据部分,高效地判断数据包是否有人入侵企图。

3.3 联动技术的实现

为了实现流处理模块、协议分析模块与控制模块的联动,系统采用控制层和数据层配合的方式。系统使用 Octeon 多核网络处理器 CN5650^[7],位于板卡中的多核芯片中有多达 12 个核。系统设计时,分配 Core 0 运行在控制层,运行控制模块,专门负责对流表的维护,接收其它核的反馈信息然后修改流表,并定期对流表进行查询,删除生存时间结束的流表项。其余的核 Core1~Core11 运行在数据层,运行预处理模块、流处理模块和协议分析模块。具体的运

行模式如图 3 所示。

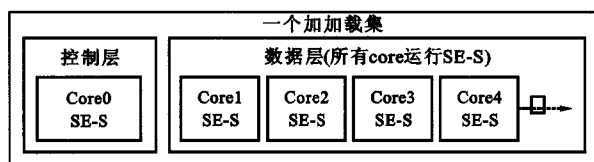


图 3 控制层与数据层模式

Octeon 多核处理器提供了物理内存分配器, 这个分配器被提供来允许应用程序在启动以后分配空闲的物理内存。空闲列表是由启动引导程序创建, 然后传递给应用程序。cvmx_bootmem_alloc_named(uint64_t size, uint64_t alignment, char * name) 函数用来从启动引导程序传递给应用程序的空闲列表中分配一个内存块, 并且在全局命名块表中给这个命名块分配一个名字, 这个被命名的空闲的物理内存块就称为命名块。

系统在设计时由 Core 0 在物理内存中调用 cvmx_bootmem_alloc_named() 函数为流表分配一个命名块, 将流表存放在这个命名的内存块中, 并返回一个指向命名块的指针, 其他核调用 cvmx_bootmem_find_named_block() 函数来接收指向命名块的指针以实现流表的共享。同样, 系统也通过命名块方式来存储控制层和数据层之间的消息。消息格式以数据结构 struct ips_message { ips_msg msg_content, uint32_t msg_tag } 定义; msg_content 存储消息内容, 消息的格式与流表项的结构相符, 以方便 Core 0 对流表的修改; msg_tag 是用来标识消息是否被更新过, 初始值为 0。控制层定期的轮询 msg_tag 的值, 当数据层传来消息后, msg_tag 变为 1; 而控制层通过轮询的方式查看 msg_tag 的变化, 如果 msg_tag 值为 1, 就接收消息并修改流表, 并重置消息的内容和消息的标志位为 0。对于消息的读写操作是互斥的, 当有某个核在修改消息时, 其它核不能够对消息进行读写操作。

4 多核并行处理优势最大化的设计与分析

Octeon 多核网络处理器支持传统流水线模式^[7]、运行到完成模式、改进的流水线模式。

运行到完成模式时, 每一个核运行相同的应用程序。每一个核接收新的 work 并且处理它到完成。一个核的处理不能够拖延整个系统对包的处理, 只涉及到单一的包。运行到完成模式易扩展, 只需要简单的增加核。

传统的流水线模式时, 每一个核处理包的一部分, 包被从一个核传递到下一个核直到处理完成。在

Octeon 中, 这种模式可以通过使每一个核只接受一个 Group 来完成。在每一核完成它自己处理的那一部分后, 它就执行 swtag_desched() 函数改变这个包的 Group, 传递这个包到下一个核。在传统的流水线中, 每一个核只会从一个 Group 接收 work。这种模式使得一个核能更加高效地获得一个包, 对包的处理不是一直到完成, 而是使用了不同的阶段。这种模式预留了额外的周期花费在 switch_desched() 上。

在改进的流水线模式中, 核可以处理包处理过程中的多个阶段。这种模式很容易通过修改每一个核的 Group 值来实现。每一个核运行相同的应用程序, 当 get_work() 操作返回一个 WQE 指针到这个核以后, 这个核能够基于包的 Group 值执行恰当的函数。改进的流水线模式和传统的流水线模式一样, 有花费在 get_work() 和 switch_desched() 的开销的缺点。当增加新功能时, 负载均衡就成了一个问题。而运行到完成模式通常是最容易负载均衡和规模扩展的架构。

综合上述因素, 系统设计采用如图 4 所示的方案: SSO 负责将捕获的网络包进行分流处理, 为了保持数据包的有序性, 将属于同一个流的数据分组放在在同一个核上处理。Core 0 运行在控制层, 负责流表的维护, 其余的核, 每个核执行从运行到完成模式。具体执行流程如下:

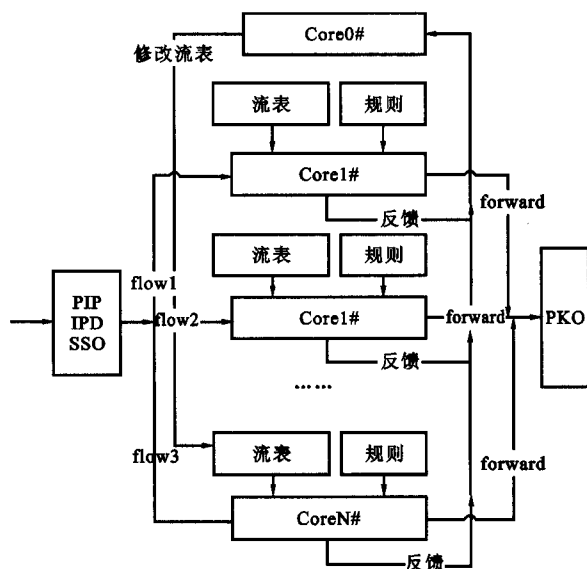


图 4 并行设计方案

(1) 接收端口接收到数据包并检查是否为 IPv6 包后, 通过 I/O 总线将数据包传给 PIP。PIP 执行数据包分析, 包括所有的软件配置检查, 计算元组哈希值, 并产生 IPD 需要的 WQE 域。

(2) IPD 从 FPA 中分配一个 WQE 缓冲区和包数据缓冲区. 将 WQE 域写入 WQE 缓冲区, 将包数据写入 L2/DRAM 包数据缓冲区. 然后执行 `add_work()` 函数将 WQE 指针添加到 SSO 相应的 Group 队列中.

(3) 数据层核执行 `get_work()` 操作, SSO 负责从相应的 Group 队列中获得一个新的 WQE 指针, WQE 包含包数据缓冲区的指针.

(4) 数据层核处理包数据, 首先进入预处理模块, 对数据包进行预处理, 检查数据包的有效性. 对于检查通过的数据包, 交给流处理模块, 进行流表的查询, 若在流表中查找成功, 根据相应的行为进行处理, 否则交由协议分析模块进行协议分析和模式匹配, 对于匹配成功的数据包, 说明发现了攻击行为, 此时发送消息到控制层核, 控制层核负责对流表进行修改, 进而对后续的流进行有效的防御.

(5) 对于未发现攻击的数据包, 处理器发送包数据缓冲指针和数据起始值到 PKO^[4] (The Packet Output Unit) 合适的输出队列. 队列的配置指定输出端口和包优先级. 如果必要, 将会按包进入顺序输出数据包.

(6) 处理器(核)释放 WQE 缓冲返还给 FPA.

(7) PKO 从 L2/DRAM 中的包数据缓冲区中读取数据到其内部存储器.

(8) PKO 选择性地添加 TCP 或 UDP 校验和, 然后通过输出总线进行 DMA, 将包数据从其内部存储器传输到输出端口将其转发. PKO 选择性的通告处理器数据包已发送.

(9) PKO 释放包数据缓冲区返还给 FPA.

5 系统测试及分析

为了验证系统是否能够达到 Gb/s 级的处理速度, 搭建了如图 5 所示的测试环境, Ixia 测试仪的四个千兆端口和 InfiniWay Octeon 多核处理器板卡的四个千兆端口对接. 测试使用的硬件工具是 Ixia, 软件工具是 THC-IPv6^[8] 工具和 Wireshark 软件, THC(The Hacker's Choice)工具用来在 Linux 环境下模拟产生大量的 IPv6 攻击流, 然后用 Wireshark 软件捕获这些攻击流, 以 .cap 文件的形式保存下来, 为了模拟出 Gb/s 级的流量, 再将其导入到 Ixia 中进行重放, Ixia 本身也可以产生大量的 IPv6 的随机数据流, 模拟真实环境的攻击行为. 配置 Ixia 固定发包速率为 2 Gb/s, 随着核数目的增加, CPU 占用率、内存占用率、包丢失率如图 6 所示. 可以看

出, 随着所使用的 Octeon 多核处理器中核的数量的增加, 在 2 Gb/s 的发包速率情况下, CPU 占用率呈线性速率下降, 内存占用率的变化不太明显, 而丢包率并没有按照核数量的增加而成倍的递减, 也就是随着核数量按倍数的增加, 处理速率并没有按倍数增加, 其主要原因是随着核数量的增加, 数据层与控制层之间的通信也不断增加, 进而 CPU 要花费更多的周期在通信的开销上, 而且在每个核处理包的过程中, 都至少要有 2 至 3 个的上下文转换开销. 因此处理速率呈线性增加, 实验结果表明, 当核数量增加至 5 个后, 在 2 Gb/s 的速率下, 系统的丢包率只有 9.57%; 当核数量增加到 9 个时, 系统基本上不丢包, 可见系统达到了 Gb/s 级的处理速度.

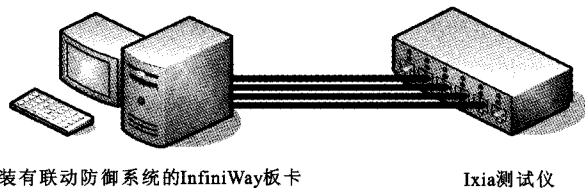


图 5 系统测试环境

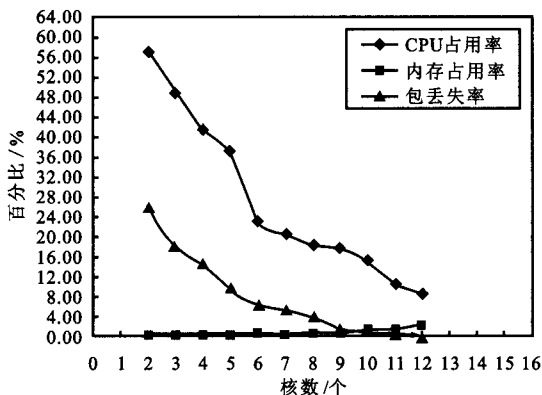


图 6 系统测试结果

参考文献:

- [1] Carlos E Caicedo, James B D Joshi, Summit R Tuladhar. IPv6 security challenges[J]. Computer, 2009, 42(2):36-42.
- [2] 廖海宁. 高效可扩展路由器命令行接口的实现[J]. 计算机工程, 2004, 30(19):173-175.
- [3] 陈晓梅. 入侵检测产品的评价研究[J]. 微电子学与计算机, 2005, 22(9):74-76.
- [4] 贺文华, 陈志刚, 胡玉平. 基于 IPv6 的网络安全与性能分析[J]. 微电子学与计算机, 2007, 24(10):120-122.
- [5] Rajahalme J, Conta A, Carpenter B. IPv6 flow label specification [J]. Request for Comments, 2004, RFC 3697.

(下转第 88 页)

决电路提供了稳定的输入,完成了限幅功能。

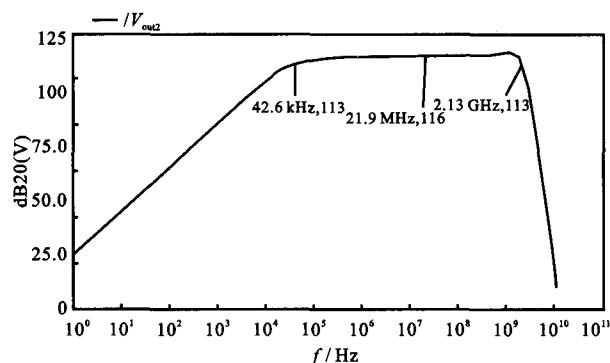
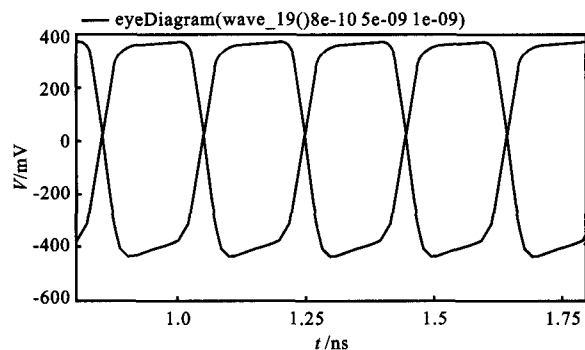
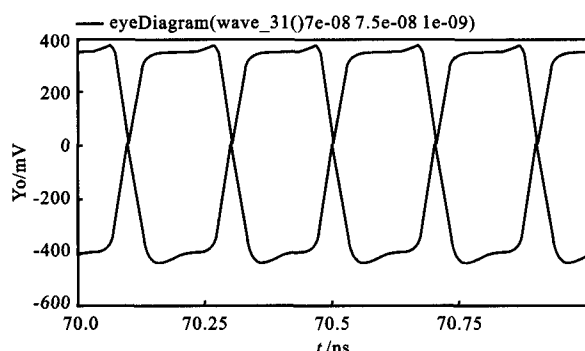


图9 前端放大电路幅频特性曲线



(a) 输入电流为10 μA时的输出眼图



(b) 输入电流为1 mA时的输出眼图

图10 前端放大电路输出眼图

5 结束语

文中基于0.18 μm CMOS工艺实现了2.5 Gb/s

s速率级的光接收机前端放大电路。该放大电路中频增益116 dBΩ, -3 dB带宽2.13 GHz, 当输入电流从10 μA到1 mA变化时, 均能限幅到800 mV, 有较宽的输入动态范围, 符合2.5 Gb/s光纤通信系统设计标准。

参考文献:

- [1] Behzad Razavi. 光纤通信集成电路设计[M]. 北京:人民邮电出版社, 2008:84-85.
- [2] Sung Min Park, Hoi Jun Y. 1.25 Gb/s Regulated cascode CMOS trans-impedance amplifier for gigabit ethernet applications[J]. IEEE Journal of Solid-State Circuits, 2004, 39(1):112-121.
- [3] 刘全, 冯军. 光通信用宽动态范围10 Gb/s CMOS跨阻前置放大器[J]. 半导体光电, 2009, 30(2):264-267.
- [4] 韩良, 尹帅. 基于CMOS工艺高速光纤通信系统的限幅放大器[J]. 微电子学与计算机, 2009, 26(7):76-79.
- [5] Sherif Galal, Behzad Razavi. 10-Gb/s limiting amplifier and laser/modulator driver in 0.18-μm CMOS technology[J]. IEEE Journal of Solid-State Circuits, 2003, 38(12):2138-2146.
- [6] Eduard Säking, Wilhelm C Fischer. A 3 GHz 32 dB CMOS limiting amplifier for SONET OC-48 receivers [J]. IEEE Journal of Solid-State Circuits, 2000, 35(12):1884-1888.
- [7] Eduard Säking. Broadband circuits for optical fiber communication[M]. New York: John Wiley Sons Inc, 2005:190-193.

作者简介:

韩良男, (1969-), 副教授. 研究方向为电流源、电压源以及光纤通信集成电路等。

孙骏毅男, (1987-), 硕士研究生. 研究方向为RF、跨阻放大器、限幅放大器等。

张颖女, (1985-), 硕士研究生. 研究方向为跨阻放大器、限幅放大器等。

(上接第83页)

thc-ipv6-1.4.tar.gz/.

[6] Qing Li, Tatuya Jinmei, Keiichi Shima. IPv6 详解[M]. 北京:人民邮电出版社, 2009:131-142.

[7] OCTEON Software Overview [R]. Cavium Networks, 2008.

[8] The Hacker's Choice (THC). [EB/OL]. [2011-03-10]. <http://www.thc.org/download.php?t=r&f=>

作者简介:

杨吉喆女, (1984-), 硕士研究生. 研究方向为计算机网络与信息安全。

王玲玲 硕士研究生. 研究方向为网络与信息安全。

陆建德 教授. 研究方向为计算机网络与信息安全。