文章编号：2096-1618(2020)05-0542-05

基于改进 RPS 技术的 IPSEC VPN 网关设计

廖 煊， 刘 苗， 李金珂
(电信科学技术第五研究所，四川 成都 610062)

摘要：基于 strongswan 和多核 IPSEC 并行技术，设计并实现一种多核 IPSEC VPN 网关。针对网卡收发数据时 sk_buff 结构体分配(释放)消耗系统资源问题，提出了一种 sk_buff 重用队列，提高 sk_buff 结构体利用率和报文处理效率；针对 host-to-host 模式下，单队列网卡将数据流分配一个 CPU 核处理而导致其负载过高和处理性能低下问题，改进 RPS 技术中 hash 算法，使其根据报文序列号而不是原有四元组进行分流，充分利用 FT1500A/16 平台多核优势，实现了指定 4 个 CPU 核间负载均衡，使 IPSEC 报文处理速率在千兆网络下最高达到 958Mbps。

关 键 词：信息安全；安全网关；RPS/RFS；负载均衡；FT1500A/16 平台；IPSEC VPN
中图分类号：TN918 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2020.05.011

0 引言

随着物联网技术的发展，日常生活中每件物品都可能成为终端，万物互联使通讯变得极为方便，但同时伴随而来的是终端与终端之间的信息安全问题。IPSEC 协议作为一个成熟的协议，是针对报文安全在网络层的一种解决方案。IPSEC 协议族主要由密钥交换协议、封装载荷协议和认证头协议组成，对 IP 报文提供了机密性保护、完整性校验等服务，其重要性日益突出。同时，随着数据信息爆炸式增长，网络设备对数据的处理能力又会成为一个瓶颈。因此，如何采用灵活的方式提高网络设备的数据处理能力成为了一个新课题。基于 FT1500A/16 平台，采取软件方式提高了平台性能。

1 相关研究

针对提高 IPSEC VPN 网关性能，许多研究者做了大量工作。基于空间复杂性和时间复杂性，Elkeelany 等[1]比较分析了在开源软件 strongswan 和 openswan 上 ESP 和 AH 协议中算法组合的性能差异。Zaharuddin 等[2]比较了不同对称加密算法和 hash 算法组合的性能，发现在 AES256-MD5 组合下性能最好。Agrawal 等[3]用软件方法实现加密模块，并且与硬件实现分析比较。Zhao 等[4]实现了基于 openswan 的安全网关，并研究了其性能瓶颈以及吞吐和加密模块之间的关系。在网络功能虚拟化技术下，Lackovic 等[5]在虚拟服务器上实现了商用 IPSEC 方案，并研究了其拓展性。同样地，基于 strongswan 和开源控制平面 Quagga，Redzovic 等[6]在软件路由器上实现了 IPSEC 加密模块并评估了其性能。Iatrou 等[7]研究了 MTU 以及网卡中断的影响，研究显示，在不改变算法组合的基础上，好的系统配置能够提升系统的性能。以上即是目前从加密模块、算法组合等方面探究其对性能的影响以及提高性能的方法。

Strongswan[8]是一种开源 IPSEC VPN 解决方案，主要在通信实体间建立共享安全策略，包括加密算法和 hash 算法，即 IKE 协商。现有 IKEv1 和 IKEv2 两种版本[9]，文中采用 IKEv2 版本协商。

2 瓶颈分析

2.1 分流

现在主流网卡大多是多队列网卡，在硬件层面采用 RSS 实现了报文分流，除非通过厂商定制否则无法轻易更改。而稍早出现的单队列网卡以及 RPS/RFS 技术是基于软件实现的分流技术，基本原理同 RSS 一致，且其代码开源可以更改。拟采用软件的方法实现报文分流。

传统单队列网卡采用 RPS/RFS[10]技术进行报文分流，见图 1。因此，当一端发送的报文经安全隧道到达另一端时，根据其源地址、目的地址、源端口和目的端口通过 hash 函数计算出一个 hash 值，HASH 函数采用 Toeplitz 哈希算法[11]。然后，根据计算出的 hash 值从设置的 CPU 核中选择一个 CPU 核处理数据。RPS 可以将同一连接的数据流分发给同一个 CPU 核处理，从而达到分流效果。但存在内核态中处理该数据流的软中断所在的 CPU 核和处理该数据流的应用程序的 CPU 核不是同一个问题，从而产生 cache miss 现象，影

收稿日期：2020-06-14

响 CPU 性能。RFS 是对 RPS 的改进,利用两个与报文相关信息的 CPU 映射 hash 表,不仅要解决把同一流报文给 CPU 核处理,还要分发给其“期望”的 CPU 核处理。

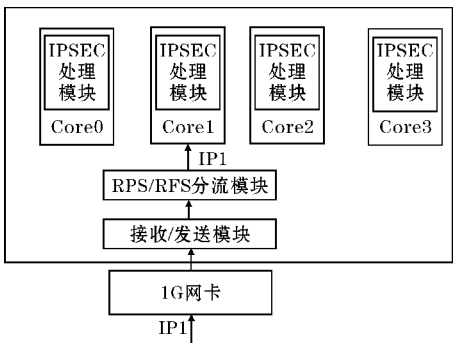

图 1　传统 RPF/RFS 分流

在 end-to-end 场景下[12],接收端收到 IP1 流中数据包的四元组相同,因此根据四元组计算出的 hash 值也相同。所以,即便接收端是多核平台,仍然只会根据 hash 值将数据包交给其中一个 CPU 核处理。因此,当 IP 1 流中数据包激增,若依然采用单核处理该流的数据包,就会使单个 CPU 处理能力成为性能瓶颈,进而浪费多核平台多核优势。

2.2　单个 CPU 核的处理性能

报文处理包含了转发、策略查找、加解密等操作,其中对 CPU 消耗最大的是加解密模块。IPSEC 处理中 ESP 处理模块的主要作用是根据协商的安全策略,包括加解密算法、hash 算法等[13],对报文进行加解密。此模块运算量大,极大地占用了 CPU 资源。所以,在软件层面,想要提高 IPSEC 整体性能,应首先提高解密模块的性能。有两种办法提高加解密模块的性能,其一是提高 CPU 主频,但是当 CPU 主频达到3 GHz 时,会接近物理极限,很难再有大的提升。因此,多核服务器给这种问题的解决提供了一种思路[14]。利用多核服务器多核优势进行并行处理,能极大地提高服务器的处理效率。

2.3　数据收/发

在 linux 中,sk_buff 是缓存数据的基本结构,几乎所有关于数据的处理都是围绕 sk_buff 结构体。当网络数据包经网卡到达内存前,系统会给 sk_buff 结构分配一个 buffer 存放数据包,即将收到的数据填充到 sk_buff数据结构中,以便进一步处理。当数据被发送后,sk_buff结构所占内存被释放,以便存放下一个数据包。可以看出,单个sk_buff结构体利用率很低。当数据量激增时,sk_buff结构的分配和释放也会在一定程度上消耗 CPU,降低整体性能。方便起见,称此模块为数据收/发模块。

3　设计方案

3.1　总体方案

设计方案中,在内存收/发模块,增加了一个重用队列,提高 sk_buff 结构体利用率,减少 sk_buff 结构体的释放对 CPU 的消耗。在 RPS/RFS 分流模块,改变 hash 值计算方式,以 IP 1 数据流中数据包的序列号作为 hash 值的计算参数之一,将同一发送端的数据包分配给 4 个 CPU 核,即 4 个核各处理数据流 IP1 的一部分,见图 2。

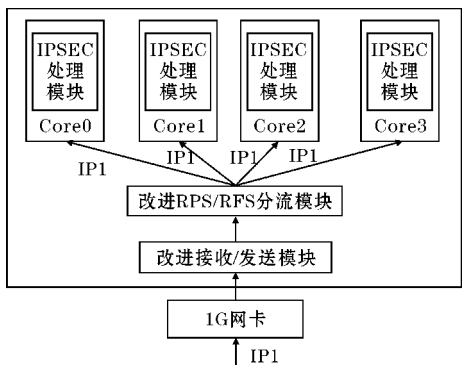

图 2　方案总体框架

IPSEC 处理模块是设计方案中实现报文机密性、完整性的重要组成部分。数据包被分配给对应 CPU 核后,需按照协商策略进行 IPSEC 处理。IPSEC 处理模块设计如图 3 所示。

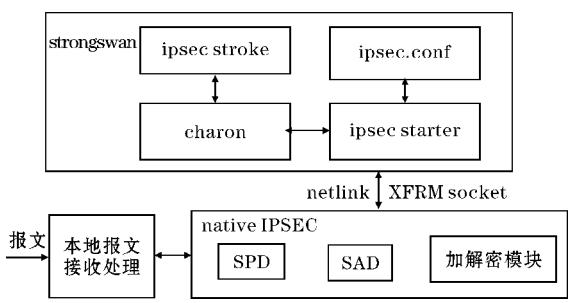

图 3　IPSEC 处理模块框架

在 IPSEC 处理之前,首先应协商加解密算法、hash 算法等安全策略建立安全隧道。采用开源软件 strongswan 进行安全策略自动协商。在 ipsec. conf 文件中配置相应安全策略,利用守护进程 charon 驱动 ipsec starter 启动策略协商并建立安全隧道。当数据包进入时,应首先根据 SPI、源地址、目的地址、端口号在内核的 SAD 和 SPD 库中查对应安全策略,决定是否丢弃、绕过或进行 IPSEC 处理。若进行 IPSEC 处理,再利用协商的加解密算法进行解密,然后再转发或处理。

3.2　改进收/发模块

针对 sk_buff 结构利用率低下问题,设计了一个重

用队列。重用队列的关键数据结构如下：

```
Reuse_queue
{
  struzongct list_head list;
  u16 flags;//状态位
  struct sk_buff *header;//队列头
  struct sk_buff *next; //下一条数据
  struct sk_buff *prev; //下一条数据
  int queue_max;///最大数据量
  u16 addr;//地址
  u16 len;//消息长度
  ……
};
```

当数据发送完毕，将 sk_buff 结构体放到重用队列之中，并将其 flags 位置 0。当新的数据到来时，首先沿着重用队列链表检查每个数据结构的 flags 位。若 flags 位为 0，则表示该结构体是空闲的，可以用来存放数据。如果检查完整个队列依然没发现可用的 sk_buff 结构，则重新分配。伪代码如下：

```
function:packet_rx(dev)
len=get_rev_len();
skb=dev_alloc_skb(len+2);
skb_reserve(skb,2);
insw(addr,skb_put,lenth)
value=check_in_queue(skb);
if(1==value)
    fill_in_queue(skb)
  else
    eth_type_trans(skb,dev)
  netif(skb)
end
```

reuse_queue 结构体中的 queue_max 变量限定了重用队列的最大长度，最大长度以内的 sk_buff 结构不会被释放。当队列上空闲 sk_buff 结构体的数量已达最大长度 queue_max 时，则不再容纳新的 sk_buff，从而将队列所占内存控制在一定范围内。

3.3 改进 RPS/RFS 分流模块

由于 CPU 的选择所依据的 hash 值是根据数据报文的源地址、目的地址、源端口号和目的端口号作为输入参数，通过特定 hash 函数计算，因此针对同一连接的数据流，需要找出数据报文的差异参数。通过研究数据报文结构，发现同一连接中 TCP 报文的序列号不同且递增。因此，选择利用序列号作为分流的输入参数之一。并且，实验设定用前 4 个 CPU 核来实现数据的处理和转发，CPU0、CPU1、CPU2 和 CPU3 被指定为对普通数据报文的处理。

当同一连接的数据经过网卡收/发模块到内存后，根据唯一差异参数序列号进行分流。为了在指定的 4 个 CPU 核进行处理，应对序列号用新的方式进行计算，伪代码如下：

```
function:get_rps_cpu(dev,skb,rflowp)
skb_get_rx_queue(skb)
  if skb->rxhash
    goto get_hsah
  else
    seq_no = get_hsah_seq_no(skb)
    skb->rxhash=jhash_3words(addr1, addr2, ports,
seq_no)
cpu_online(tcpu)
if cpu ! = smp_processor_id()
  rcpus =__get_cpu_var()
end

function:get_hash_seq_no(skb)
  if pskb_may_pull(skb, ihl)
    seq_no = skb->data
```

3.4 IKE 协商模块

在建立安全隧道之前，通信两端需要进行密钥交换以协商安全策略，包括加密算法、hash 算法等。采用软件 strongswan 进行密钥自动协商，结构框架如图 4 所示。

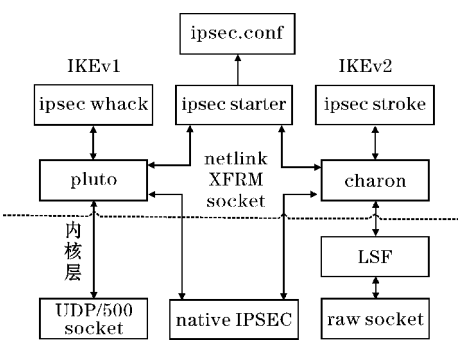

图 4 strongswan 整体框架

ipsec starter 控制密钥协商守护进程 charon 并从 ipsec. conf 中读取策略配置，同对端总共交换 6 条信息。而与内核通过 XFRM socket 进行交互。

3.5 其他模块

(1)加解密模块。Linux 内核协议栈实现了完整的 IPSEC 协议，因此，实验的加解密模块采用协议栈的 IPSEC 处理，虽然这带来性能影响。对于进入报文，根据报文 SPI 进行安全策略查找，然后按照相应策略对数据包进行解密。外出报文也是根据查找的安全

策略进行加密后再进行转发。

(2)SPD 和 SAD 模块。同加解密模块一样,依然采用 linux 内核实现的安全联盟库和安全策略库。

3.6　测试

实验测试的拓扑如图 5 所示,实验配置如表 1 所示。

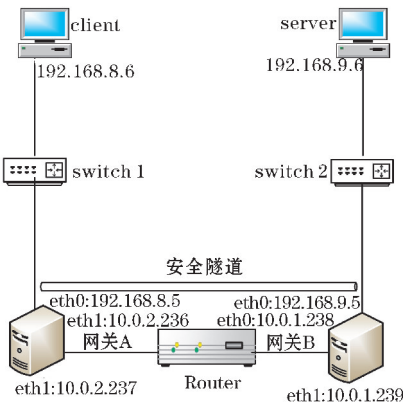

图 5　实验拓扑

表 1　实验配置

名称	型号
CPU	FT1500A/16
内核版本	2.6.36
系统	ubuntu14.04
strongswan	5.8.0
算法组合	AES256CBC-SHA1
网卡	千兆
发包工具	iperf

将设计方案应用在网关 A 和网关 B 上,用 iperf 在 client 端向 server 端发包,包大小为 1510 Bytes,测试其速率。

为比较文中方案的性能,参考 strongswan 官网应用场景,搭建并测试了相同配置下的 IPSEC 性能,比较其性能差异,测试结果如图 6。

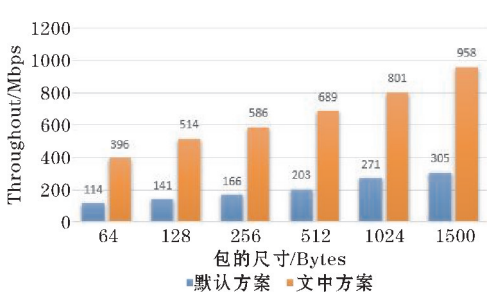

图 6　吞吐量对比图

实验发现,随着包尺寸变大,默认配置和文中方案的吞吐量都在增加,但文中方案的吞吐量增幅明显大于默认配置的增幅。在包尺寸相同的情况下,文中方案的性能明显高于默认配置方案,大约是其吞吐量的 3 倍多,且最高吞吐量达到958 Mbps,基本达到了千兆线速。

4　总结

文中首先分析了基于软件的 IPSEC VPN 网关的性能瓶颈,包括分流模块的 hash 计算方式让一个 CPU 核处理同一连接的数据流,使其负载过大、性能过低;接收发送模块中 sk_buff 结构体的利用率过低,导致结构的分配和释放会在一定上消耗 CPU;另外,单个 CPU 的主频有物理极限,因此处理能力也有限。针对上述问题,提出一种基于多核平台的 IPSEC VPN 网关设计方案。在网卡驱动程序中增加重用队列,增加 sk_buff 结构体的重复使用率,减少该结构体重新分配和释放的次数;改进 RPS 算法中的 hash 计算方式,使得报文序列号成为 hash 计算的差异性参数,让同一数据流的报文能被指定的 4 个 CPU 核处理,极大地利用了多核服务器的多核优势。实验结果表明,相对默认方案,提出的方案能将性能提高 3 倍多,在报文为 1500 Bytes时,最高吞吐量达到了958 Mbps,基本达到了千兆线速。

参考文献:

[1] Elkeelany O, Matalgah M, Sheikh K, et al. Performance analysis of IPSEC protocol: encryption and authentication[J]. Computer Applications and Industrial Electronics, 2002, 12(3): 116-168.

[2] Zaharuddin M, Rahman R, Kassim M. Technical comparison analysis of encryption algorithm on site-to-site IPSEC VPN[J]. Computer Applications and Industrial Electronics, 2018: 641-645.

[3] Agrawal H, Dutta Y, Malik S. Performance analysis of offloading IPSEC processing to hardware based accelerators[J]. Electronic System Design, 2012: 291-294.

[4] Y Z Da, X J Yi, L Chuang, et al. Implementation and performance evaluation of IPSEC VPN based on netfilter[J]. Telecommunications Forum, 2015, 10(1): 98-102.

[5] Lacković D, Tomić M. Performance analysis of virtualized IPSEC VPN endpoints [J]. Information and Communication Technology, Electronics and Microelectronics, 2017: 466-471.

[6] Redžović H, Smiljanić A, Savić B. Performance e-

valuation of software routers with VPN features [J]. Telecommunications Forum,2016(24):1-4.
[7] Iatrou M G,Voyiatzis A G,Serpanos D N. Optimizations for high performance IPSEC execution[J]. International Conference on E-Business and Telecommunications,2009:199-211.
[8] Park J,Jung W,Lee I,et al. Practical ipsec gateway on embed-ded apus[C]. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016: 1255 - 1267.
[9] 李辉.移动VPN技术综述[J].中国电子科学研究院学报,2019,14(9):903-912.
[10] 张娜.基于IPSEC的VPN网络安全的实现[J].中国新通信,2016,18(19):83-86.
[11] Lopez D,Lopez E,Dunbar L. Framework for Interface to Network Security Functions[J]. Computer Applications and Industrial Electronics, 2018,10(6):53-56.
[12] Driessen B,Güneysu T,Kavun E B,et al. Ipsecco:a lightweight and reconfigurable IPSEC core [J]. Reconfigurable Computing and FPGAs, 2012:1-7.
[13] Vajaranta M,Kannisto J,Harju J. IPSEC and IKE as functions in SDN controlled network[J]. Network and System Security,2017:521-530.
[14] Kourtis M A.,Xilouris G,Riccobene V,et al. Enhancing VNF performance by exploiting SR-IOV and DPDK packet processing acceleration[J]. Network Function Virtualization & Software Defined Network,2016:4-7.

Design of IPSEC VPN Gateway based on Improved RPS Technology

LIAO Xuan, LIU Miao, LI Jinke
(The Fifth Research Institute of Telecommunications Technology,Chengdu 610062,China)

Abstract:Based on StrongSwan and multi-core IPSEC parallel technology, a multi-core IPSEC VPN gateway was designed and implemented. For the problem of sk buff struct distributes and consumes the system resource while network card sends and receives data, An Sk_buff reuse queue is proposed to increase the utilization rate of sk_buff and the processing efficiency of packet. For host-to-host mode, single queue network card will distribute data flow a CPU core that it will lead to high load and poor process performance,we improve the technology of RPS hash algorithm, make it conducts diversion according to the serial number of the message instead of the original quad shunt, we make full use of multi-core power FT1500A / 16 platform, realize the specified load balance between four CPU cores, make the IPSEC packet processing rates under the gigabit network up to 958 Mbps.

Keywords:information security;security gateway;RPS/RFS;load balancing;FT1500A / 16 platform;IPSEC VPN