文章编号:1001 - 2486(2005)05 - 0030 - 05

高性能路由器并行转发技术*

胡晓峰,孙志刚
(国防科技大学 计算机学院,湖南 长沙 410073)

摘 要:随着线路传输速率的快速提高,报文线速转发面临极大挑战。基于并行处理技术,提出分布式并行转发引擎结构,实现高速报文转发。针对并行转发引擎负载分配问题,设计 AHDA(Adaptive Hashing Dispatch Algorithm)算法,该算法为综合考虑负载均衡和报文保序提供支持。模拟结果表明,AHDA 算法均匀分配负载,保证很低的报文乱序率,对网络处理器规模具有良好的可扩展性。

关键词:路由器;转发引擎;并行转发
中图分类号:TP393 文献标识码:A

Parallel Forwarding in High Performance Router

HU Xiao-feng, SUN Zhi-gang
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: It is a great challenge to forward packets at line rate as the line rate grows rapidly. This paper proposes a distributed parallel forwarding engine to solve this problem based on parallel processing. Adaptive Hashing dispatch algorithm(AHDA) is proposed to dispatch the packets received, which compromises between load balance and packet ordering. Simulation results show that AHDA algorithm dispatches load in a balanced way, reorders packets at low rate, and scales well with the number of network processors.

Key words: router; forwarding engine; parallel forwarding

互联网流量的快速增长对网络容量提出很高要求,光纤通信技术的发展解决了传输带宽问题,而路由器受限于微电子技术的发展水平,成为互联网发展的瓶颈[1]。

转发引擎是路由器的核心部件,主要负责 IP 路由查表和报文分类。由于当前主要的路由查找模式是最长前缀匹配,匹配的不精确性导致大量的查找开销,设计适应大规模路由表和高速线路的 IP 路由查找算法面临极大挑战[2]。报文分类是一个多维匹配问题,匹配模式的任意性要求支持大的分类器集合和任意范围的匹配,它的设计难度更甚于路由查表[3]。因此报文转发技术是制约路由器性能提高的一个关键因素。

解决报文转发性能瓶颈的一种方法是引入并行处理技术,使用多个较低速转发模块实现高速并行转发引擎[4,5]。负载分配是并行转发引擎的关键问题,需要综合考虑负载均衡和报文保序,在提高吞吐率的同时降低对端系统的影响。

Kencl[4] 基于最大随机权值算法[6],引入自适应负载调整机制,避免网络流量的非一致性造成负载分配不均衡。作者证明,改变报文分配方式影响的报文数最少,从而保证最低的报文乱序率。由于算法的复杂度为 $O(m)$(m 为转发模块数),如何适应高速网络环境是需要进一步解决的问题。

Cao[7] 提出的基于表格的哈希算法特别适合应用于并行转发引擎,该算法根据哈希函数确定表项索引,然后向表项定义的转发模块分配报文。基于表格的哈希算法的特点是,采用不同策略修改表项内容可以实现不同的负载分配方式。一种比较直观的策略是根据阈值修改表项,当某个处理模块的负载大于阈值时,将其部分负载分配到较空闲的模块处理。这种工作策略的缺点是,在流量局部性较强的情况

* 收稿日期:2005 - 05 - 20
基金项目:国家自然科学基金项目(90104001);国家重点基础研究发展计划项目(2003CB314802)
作者简介:胡晓峰(1975—),男,助理研究员,博士。

下,会导致模块负载的急剧变化,引起负载的振荡性调整。为此,Wang[5]提出基于概率的表项修改策略EHDA(Enhanced Hash-based Distributed Algorithm),当负载大于阈值时,依概率调整负载分配方式,从而有效解决各模块负载剧烈变化的问题。EHDA针对TCP协议对乱序敏感的特点,提出应该优先保证TCP报文顺序。调整概率的计算是该策略的核心问题,既需要考虑负载均衡和报文保序,还要求配置灵活,适应不同特性的网络流量。但是,Wang仅列举了具体的概率设置实例,没有讨论一般化的计算方法。针对该问题,本文设计负载分配算法AHDA(Adaptive Hashing Dispatch Algorithm),算法参数含义明确,配置简单,能够根据转发模块的负载状况和网络流量灵活地提供报文保序和负载均衡支持。

文献[4,5]讨论的并行转发引擎将转发模块组织成共享资源,为所有输入端口提供服务,请求和应答两次经过交换开关,造成较大的通信开销。其次,为了保证负载均衡,输入端口需要及时获取转发模块的负载状况,而状态信息的采样通常存在滞后和更新不同步等问题,影响负载分配的均衡性。分布式并行转发引擎结构改变转发模块的集中组织方式,将报文转发功能分布到各个线卡实现,解决了集中式并行转发引擎存在的问题。因此,本文针对分布式并行转发引擎结构进行研究。

1 分布式并行转发引擎

1.1 结 构

分布式并行转发引擎由负载分配器DMUX、报文整合器MUX和网络处理器(即转发模块)组成,与外部线路和交换系统接口,如图1所示。各线卡均包含一个转发引擎。

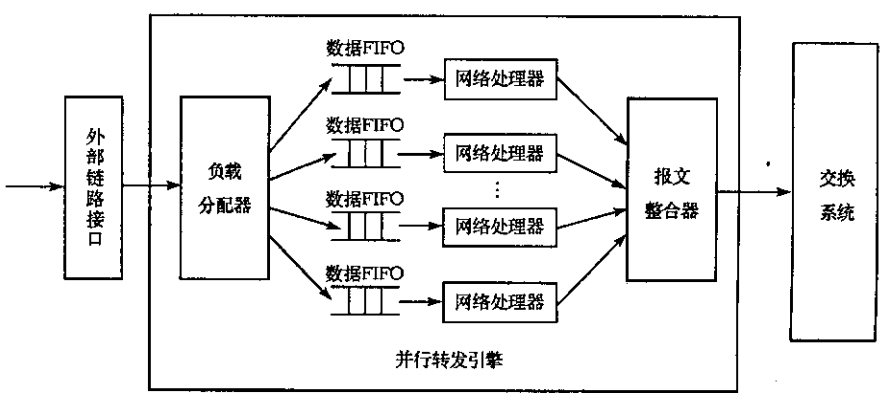

图1 分布式并行转发引擎模型

Fig.1 The model of distributed parallel forwarding engine

接收报文后,负载分配器将报文分配到网络处理器,各网络处理器并行处理报文,报文整合器将处理完的报文发送到交换系统。负载分配算法控制负载分配器的工作方式,它是保证并行转发引擎高效运行的关键。下面着重介绍我们提出的负载分配算法AHDA。

1.2 负载分配算法AHDA

假设并行转发引擎由N个网络处理器组成,分别记作$N_i(0\leqslant i\leqslant N)$,报文的流标识为$Packet$。负载分配算法的任务是寻求函数$f$,建立报文与网络处理器之间的映射关系,即$f(Packet)=N_i$。AHDA算法采用基于哈希表的映射方式,报文的流标识需要经过两级映射才能确定网络处理器,即$f(Packet)=f_2[f_1(Packet)]=N_i$,$f_1$的值域为哈希表项的索引集合,哈希表实现二级映射函数f_2,表项内容为网络处理器标识。间接映射的工作方式如图2所示。修改哈希表项可以改变报文与网络处理器的映射关系,调整网络处理器的负载,为实现负载均衡和报文保序提供有力支持。

AHDA算法包括计算哈希表索引和调整负载分配方式两部分功能。分析结果表明,报文目的IP地址的低8位具有较强的随机性[7],将它作为哈希值既满足均匀分配负载的要求,还具有简单高效的特点。

负载分配的自适应调整机制是分布式并行转发引擎设计的难点。当网络处理器的负载较重时,

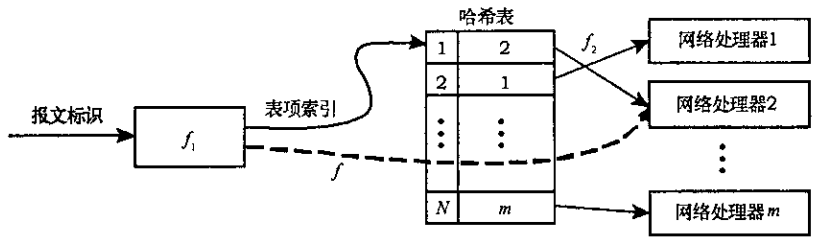

图2 报文—网络处理器的间接映射方式

Fig.2 Mapping of packets to network processors

AHDA算法依概率选择其它网络处理器处理报文,实现负载分流,同时避免网络处理器负载剧烈变化,降低报文乱序率。队列长度反映网络处理器的忙闲程度,为了避免流量抖动引起映射关系的频繁变化,根据(1)式计算平均队列长度 Avg_Queue_Len,将它作为网络处理器负载的衡量参数。式中 w 为影响因子,描述瞬时队列长度 $Queue\_Len$ 对平均队列长度的贡献。

$$Avg\_Queue\_Len = Avg\_Queue\_Len * (1 - w) + Queue\_Len * w \tag{1}$$

报文乱序对不同协议的影响不同,因此需要实现不同的报文保序和负载均衡目标。为此,AHDA算法在哈希表中为TCP和UDP设置不同表项。其次,当网络处理器发生拥塞时,UDP报文与网络处理器映射关系的调整概率大于TCP,以优先保证TCP报文的顺序。TCP和UDP哈希表项的更新概率如式(2)和式(3)所示:

$$P_{tcp} = \begin{cases} 0, & 当\ Avg\_Queue\_Len < Th\_Min_{tcp} \\ \dfrac{Avg\_Queue\_Len - Th\_Min_{tcp}}{Th\_Max_{tcp} - Th\_Min_{tcp}}, & 当\ Th\_Min_{tcp} \leqslant Avg\_Queue\_Len < Th\_Max_{tcp} \\ 1, & 当\ Avg\_Queue\_Len \geqslant Th\_Max_{tcp} \end{cases} \tag{2}$$

$$P_{udp} = \begin{cases} 0, & 当\ Avg\_Queue\_Len < Th\_Min_{udp} \\ \dfrac{Aug\_Queue\_Len - Th\_Min_{udp}}{Th\_Max_{udp} - Th\_Min_{udp}}, & 当\ Th\_Min_{udp} \leqslant Avg\_Queue\_Len < Th\_Max_{udp} \\ 1, & 当\ Avg\_Queue\_Len \geqslant Th\_Max_{udp} \end{cases} \tag{3}$$

式中 Th_Min_{tcp}、Th_Max_{tcp}、Th_Min_{udp}和 Th_Max_{udp}为队列长度阈值,通过设置不同阈值可以平衡负载分配和保序要求。队列阈值对并行转发引擎的性能影响很大,必须选择恰当的队列长度阈值。一般地,有以下取值要求:(1) Th_Min_{tcp}大于 Th_Min_{udp};(2)队列长度阈值上限为下限的2倍;(3)重负载情况时可设置较高的阈值,以获得更好的综合性能。

综合以上,AHDA算法的描述如图3所示。

接收输入报文后:

Step1 将目的IP地址的低8位DstIP[7..0]作为哈希表索引,根据协议类型选择网络处理器

Step2 计算网络处理器的平均队列长度Avg-Queue-Len:

if(队列非空)

Avg-Queue-Len = Avg-Queue-Len * (1 - w) + Queue-Len * w

Step3 根据报文类型,计算映射表项修改概率 P,以概率 P 选择负载最轻的网络处理器 N_i 处理当前报文,并将哈希表项内容改为 N_i

图3 AHDA负载分配算法

Fig.3 AHDA payload dispatching

1.3 负载汇聚算法

报文整合器运行负载汇聚算法,将完成处理的报文发送到交换系统。负载汇聚算法用轮询方式为

网络处理器提供服务,算法描述如图 4 所示,其中 P_{RR} 为轮询指针。

Step1 根据 P_{RR} 选择优先级最高,而且输出队列非空的网络处理器 N_j

Step2 :if(存在这样的 N_j)

发送 N_j 队列头报文

$P_{RR} = (P_{RR} + 1) \bmod N$

图 4 负载汇聚算法

Fig.4 Payload concentration algorithm

2 性能分析

通过模拟实验分析了分布式并行转发引擎的性能,模拟实验将 NLANR 提供的实际网络流量作为输入源(http://pma.nlanr.net/DailyTr/)。根据 Lightreading 对代表性路由器的测试结果(http://www.lightreading.com),模拟实验假设报文处理时间均值为 70μs,服务时间服从几何分布。转发引擎包含 4 个网络处理器。比较了 AHDA 与哈希映射算法 HDA 和轮询调度算法 RRDA 的负载分配均衡率、平均延时和 TCP 报文乱序率。

HDA 算法根据报文源目 IP 地址的异或值直接确定网络处理器[7]。RRDA 算法以轮询方式将报文分配到各网络处理器。AHDA 算法的参数配置为 $Th\_Min_{udp} = 2$, $Th\_Max_{udp} = 4$, $Th\_Min_{tcp} = 2$, $Th\_Max_{tcp} = 5$。

图 5 给出 3 种算法的负载分配情况,从图中可以看到,HDA 算法的负载均衡特性最差,这是因为它没有自适应机制,不能动态调整报文分配方式。RRDA 算法的轮询分配方式保证负载均衡。AHDA 算法具有较好的负载均衡特性,各网络处理器的负载接近 25%。图 6 给出固定时间段内各网络处理器分配的报文数随时间的变化情况,该图显示负载表现出较好的均匀性,这说明 AHDA 算法能够及时调整负载分配方式,具有良好的自适应性。

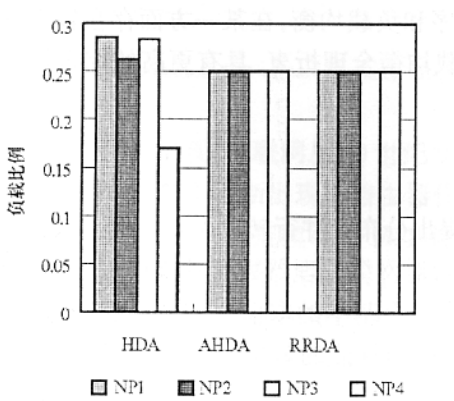

图 5 负载分配比例

Fig.5 Proportion of packets dispatched to NPs

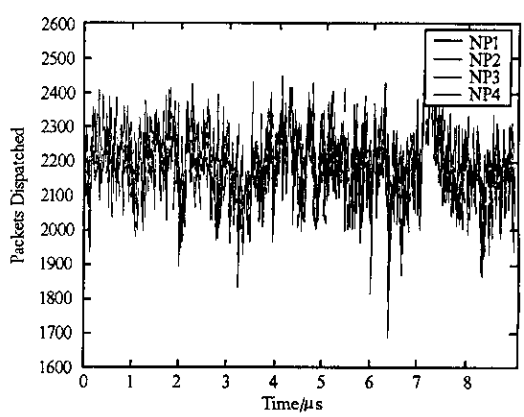

图 6 负载分配随时间变化情况

Fig.6 Evolution of the payload to different NPs

表 1 给出报文乱序率和报文平均延时。HDA 算法没有改变报文的分配方式,它不会造成报文乱序。RRDA 算法强调负载分配的均衡性,因此报文乱序率较高,达到 1.95%。AHDA 算法的报文乱序率为 0.139%,与 HDA 算法相比,它以较低的报文乱序率为代价,取得了均衡的负载分配。

在平均延时方面,RRDA 算法的平均延时最小,而 HDA 算法的平均延时高于 11 534μs,这是因为负载分配的不均衡性极大地增加了报文延时。AHDA 算法的自适应机制保证报文的平均延时与 RRDA 算法基本相当。

表1 报文乱序率和平均延时

Tab.1 Packet reordering rate and average latency

	乱序率	平均延时(μs)
HDA	0	11 534.722
RRDA	0.019547588	404.75178
AHDA	0.001394213	416.91137

从表1还可以看到,由4个网络处理器组成的转发引擎处理能力不足,导致AHDA的平均延时较大。为此,我们将网络处理器增至8个,图7给出各网络处理器的负载比例。该图显示各网络处理器的负载比例接近于12.5%,保证了负载均衡,说明AHDA算法在更多网络处理器的配置情况下,仍具有很好的负载均衡特性。

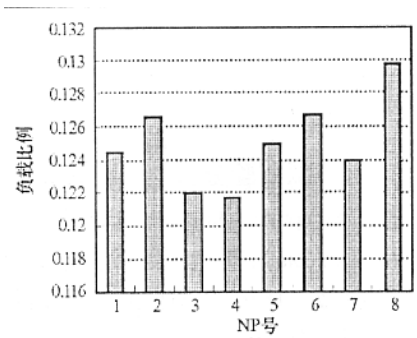

图7 8NP时负载分配比例

Fig.7 Proportion of packets dispatched to 8 NPs

表2给出在不同配置情况下,哈希表的更新次数、报文乱序率和平均延时。增加网络处理器个数提高转发引擎的处理能力,哈希表的更新次数下降两个数量级。相应地,报文乱序率和平均延时也远低于4个网络处理器的配置情况。

表2 哈希表更新次数、报文乱序率和平均延时

Tab.2 Update times to Hash table, packet reordering rate and average latency

	4个NP配置	8个NP配置
哈希表更新次数	824 828	5850
乱序率	0.001394	8.49E-05
平均延时(μs)	416.91137	148.14321

性能分析表明,HDA和RRDA算法不能兼顾报文保序和负载均衡,在某一方面存在较大性能损失。与这两种算法相比较,AHDA算法可以对报文保序和负载均衡合理折衷,具有更高的综合性能,另外算法还具有良好的可扩展性。

3 小 结

针对高速网络环境报文线速转发实现困难的问题,提出分布式并行转发引擎结构,并设计了负载分配算法AHDA,该算法综合考虑报文保序和负载均衡性能,对网络处理器数量具有较好的可扩展性。

队列长度阈值是AHDA算法的重要参数,本文仅给出一个基本的取值原则。今后工作将进一步分析不同参数配置对应的系统吞吐量和乱序率,全面理解参数对性能的影响,为合理选择队列阈值提供理论指导。

参 考 文 献:

[1] McKeown N. Growth in Router Capacity[R]. IPAM Workshop '03, October 2003.

[2] Ruiz-Sanchez M, Biersack E, Dabbous W. Survey and Taxonomy of IP Address Lookup Algorithms[J]. IEEE Network, March 2001.

[3] 喻中超, 吴建平, 徐恪. IP分类技术研究[J]. 电子学报, 2001, 29(2): 260-262.

[4] Kencl L, Boudec J. Adaptive Load Sharing for Network Processors[A]. In Proc. of Infocom '02[C], June 2002.

[5] Wang J, Nahrstedt K. Parallel IP Packet Forwarding for Tomorrow's IP Routers[A]. In Proc. of HPSR '01[C], May 2001.

[6] Thaler D, Ravishankar C. Using Name-based Mappings to Increase Hit Rates [J]. IEEE/ACM Transactions on Networking, 1998, 6(1): 1-14.

[7] Cao Z, Wang Z, Zegura E. Performance of Hashing-based Schemes for Internet Load Balancing[A]. In Proc. of Infocom '00[C], March 2000.