中国科学 E 辑：信息科学 2008 年 第 38 卷 第 10 期：1652 ~ 1662
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE IN CHINA PRESS

面向下一代互联网的可扩展路由器体系结构——MPFS

孙志刚*，戴艺，龚正虎

国防科技大学计算机学院，长沙 410073
*E-mail: sunzhigang@263.net

收稿日期: 2008-06-03; 接受日期: 2008-08-21
国家重点基础研究发展计划(批准号: 2003CB314802)资助项目

摘要 随着以IPv6为核心的下一代互联网的发展，现有路由器体系结构在性能、复杂性、扩展能力和节能等方面存在许多难以克服的问题，特别是随着网络规模的扩展，如何实现大容量 IPv6 FIB(forwarding information base)线速查找是下一代高性能路由器设计面临的重大挑战．文中提出一种与目前分布转发集中交换结构不同的新型路由器体系结构——MPFS(massive parallel forwarding and switching). MPFS 基于 FIS(forwarding in switching)思想，将网络处理器嵌入到可扩展多级交换网络中，通过流水和分布并行方式实现转发．与 MPP(massive parallel processing)一样, MPFS 通过互连大量简单同构的 FSN(forwarding and switching node)实现可扩展的转发和交换．重点研究了 MPFS 中 IPv6 FIB 查表问题，提出了将 IPv6 FIB 映射到多级 FSN 上的方法．模拟和计算表明基于现有 DRAM 器件和 Tree Bitmap 查找技术，MPFS 可在 40 Gbps 接口上实现包含 1 M 个 IPv6 前缀的 FIB 的线速查找．最后提出了一种基于 MPFS 体系结构的吞吐率为 160 Tbps 的核心路由器实现方案．

关键词
路由器
转发交换
可扩展
体系结构

下一代互联网是以IPv6协议为核心，规模更大，性能更高，可扩展性和应用支撑能力更强的新型网络．路由器是支持网络体系结构在规模和性能等多个维度上扩展的核心设备．随着IPv6协议的部署，特别是互联网规模的不断扩大，现有路由器的分布转发，集中交换体系结构逐渐暴露出许多难以克服的问题．主要包括：①性能问题．大量出现的P2P流媒体应用要求互联网提供更多带宽．虽然单根光纤已实现 12.8 Tbps的传输速率，但由于路由器端口处理速度只能达到 40 Gbps，无法将光网络的原始带宽提供给网络应用，因此路由器已经成为互联网的“电瓶颈”．②复杂性问题．为获取较高的处理性能，网络处理器的设计复杂度越来越高．如

Cisco CRS-1 路由器 1)中SPP(silicon packet processor)网络处理器内部集成多达 188 个 32 位CPU以及复杂的cache和存储器控制逻辑, 不但增加了设计成本, 更加剧了软件编程的复杂性. ③规模扩展问题. 现有路由器结构不支持通过组件堆叠实现规模和性能扩展, 因此设备使用寿命受到影响. 例如运营商现在不得不考虑设备的完全替换以解决FIB极限问题2). ④节能问题. 随着能源和环保问题得到越来越多关注, 网络设备缺少有效功率管理和节能手段的缺点暴露得越来越明显 [1,2], 而网络核心设备缺少有效的节能方案, 难以根据流量变化动态调节系统的能量消耗.

转发和交换是路由器基本的处理功能. 现有路由器可扩展能力主要体现在交换结构可扩展, 即采用多级交换方式, 基于多个规模较小交换开关的分布处理实现较大规模的交换能力. 如Cisco公司的CRS-1 路由器通过可扩展Benes交换网络可支持 1296 个网络接口. 而由于印制板板面、结构设计和成本等因素限制, 路由器转发处理的扩展能力相对较弱, 网络处理器只能通过片内并行处理实现高速分组的转发. 因此当前网络处理器已经成为高性能路由器中设计最复杂、成本最高和功耗最大的芯片, 也是制约路由器性能进一步提高的瓶颈. 更加严重的是IPv6 分组转发需要实现比IPv4 分组转发更加复杂的 64 位最长前缀匹配, 而IPv6 网络中FIB容量的扩展问题更加恶劣 [3], 这进一步加大了下一代网络处理器设计的复杂性和实现难度.

我们认为下一代互联网路由器设计只有突破现有体系结构框架, 才能有效解决性能和复杂性等问题, 才能对具有多维可扩展能力的下一代互联网提供有效支撑. 本文从体系结构创新入手, 通过对分组转发交换流程进行重新划分和映射, 提出一个与现有路由器分布转发, 集中控制体系结构完全不同的新型结构——MPFS. 该结构不但硬件实现简单, 而且可以像 MPP 计算机一样通过部件的堆叠实现性能和规模的扩展.

论文后续组织如下: 第 1 部分对相关研究进行介绍; 第 2 部分提出 FIS 处理机制和基于 FIS 的 MPFS 体系结构, 并对其可扩展能力以及转发操作到交换网络的映射进行分析; 第 3 部分提出 MPFS 结构中 FIB 查表的实现方法, 并对其实现复杂度进行分析; 第 4 部分提出一种基于 MPFS 结构的 160 Tbps 路由器的实现方案; 第 5 部分是全文总结, 并对下一步工作进行介绍.

1 研究现状

目前对路由器体系结构的研究不是很多. 美国GENI计划中提出基于通用硬件传输平台研制meta-router的思想 [4], 主要思路是通过硬件资源的虚拟化和隔离, 为不同的网络体系结构创新实验提供平台. 美国DARPA 2004 年支持的 100 Tbps光分组交换路由器项目 [5], 目的是利用全光器件实现高性能分组交换, 主要思路是利用现有体系结构, 用光器件取代电器件以获得更高的处理性能. IETF提出了路由器转发和控制平面分离的框架 [6], 目的是支持转发平面和控制平面相对独立的研究和实现. 上述研究都是基于分布转发、集中交换体系结构, 没有为解决路由器规模扩展和复杂性等问题提出新的思路.

可扩展多级交换近年来一直是路由器交换技术研究的热点. 多级交换结构分为动态互连网络和静态互连网络. 动态互连网络节点间的连接关系不固定, 通过在连接通路上使用交叉

1) http://www.cisco.com/en/US/prod/collateral/routers/ps5763/prodbrochure0900aecd800f8118.pdf
2) http://www.nanog.org/mtg-0702/presentations/bof-report.pdf

点动态配置输入和输出接口的连接关系, 如Benes网络 [7,8]. 静态互连网络中节点间的连接关系固定, 如 3D-torus网络 [9]. 动态互连网络存在内部阻塞问题, 特别是同时连接多个输入输出端口时, 会引发对内部链路和输出端口的竞争, 因此调度和流控是保证动态网络性能的关键. 静态互连网络硬件代价高, 特别是多路径负载均衡带来的报文乱序问题难以解决.

目前多级交换的研究主要集中于动态互连网络. 并行报文交换(parallel packet switch, PPS)结构 [10]由多个低速的交换模块组成, 各模块并行工作, 为报文提供多个交换通路. PPS降低了系统设计和实现难度, 通过增加低速模块的数量可提高交换的整体性能; 但其缺点是集中式信元分派受限于通讯复杂性而难以实现, 而分布式信元分派则会导致信元乱序和重组复杂. 此外, 由于外部端口数必须等于内部交换平面端口数, PPS交换结构难以支持较高的端口密度. 负载均衡交换(load-balanced switching)结构 [11]采用 2 级交换方式, 第 1 级对到达各输入接口的流量进行负载均衡, 第 2 级将数据交换到相应输出端口. 这种结构信元调度简单, 在不均衡流量下也可获得较高吞吐率, 但缺点是信元顺序控制复杂. 目前T比特路由器产品中Cisco的CRS-1 采用 3 级Benes结构, Juniper的T640 采用 3 级Clos结构 1).

网络处理器是报文转发处理的核心器件, 是影响路由器扩展能力的重要因素. 网络处理复杂性主要来自FIB查表、QoS控制以及信元重组等需要大量访问片外存储的操作, 因此存储访问延时与网络处理性能要求之间的矛盾是网络处理器设计必须解决的根本矛盾 [12]. 目前网络处理器设计的核心思想是采用并行处理技术隐藏外部存储器访问延时, 以取得高效的处理性能. 然而随着并行度的不断增加, 必须解决共享资源的访问瓶颈问题, 因此网络处理器设计又必须引入复杂的资源管理和调度机制, 这进一步增加了其复杂性. 目前对FIB查表、报文调度的研究比较成熟, 如树位图(Tree Bitmap)查表算法 [13]、DRR调度算法 [14]都得到了广泛应用.

流水转发引擎中存储器分配是网络处理器并行处理中的重要问题. Chung等 [15]提出一种两端口共享存储模型, 克服了单端口存储模型访问性能高, 但资源利用率低的缺点. Basu等人 [16]提出用MinMax算法平衡不同流水栈上存储器的使用, 然后通过多种优化来减少、平衡每一流水栈因更新而访问存储器的次数, 确保不会产生瓶颈; 为解决共享存储器多处理单元访存冲突问题, Sherwood等人 [12]提出一种独特的流水化宽字存储器, 通过使用更小的存储器片(tiles)来流水化存储器设计, 增加每次存储器访问宽度, 使得多线程处理器在同一时间可处理更多任务. 大量计数器管理是网络处理器面临的新需求. 目前广泛研究的SRAM与DRAM混合计数器结构 [17, 18], 虽然减少了SRAM的使用, 但计数器管理算法难以高速实现.

目前网络处理器设计复杂度不仅远远高于交换系统, 而且单位功耗获得的计算能力和 IO 带宽也远远高于通用多核 CPU. 根据统计 2), 高性能路由器中网络处理器(含外围存储器件)的功耗占系统总功耗的 61%, 而交换网络只占 11%. 因此简化网络处理器的复杂性是目前高性能路由器体系结构研究的重要目标.

1) http://arl.wustl.edu/~jst/cse/577/readings/juniperTseries.pdf
2) http://www.cisco.com/web/about/ac50/ac207/proceedings/POWER_GEPPS_rev3.ppt

2 MPFS 体系结构

2.1 FIS 处理机制

路由器 N 端口 M 级交换网络模型如图 1(a)所示. 每个交换的报文将依次通过第 1,2,…,M 级的一个交换模块. 现有路由器采用 FBS(forwarding before switching)处理机制, 即报文在进入交换网络前必须完成精确转发以确定输出端口. 而 MPPS 基于 FIS(forwarding in switching)机制, 在每个交换单元中嵌入网络处理器, 将转发操作嵌入到多级交换单元中分多个阶段实现, 图 1(b)和(c)是 2 种处理机制的比较.

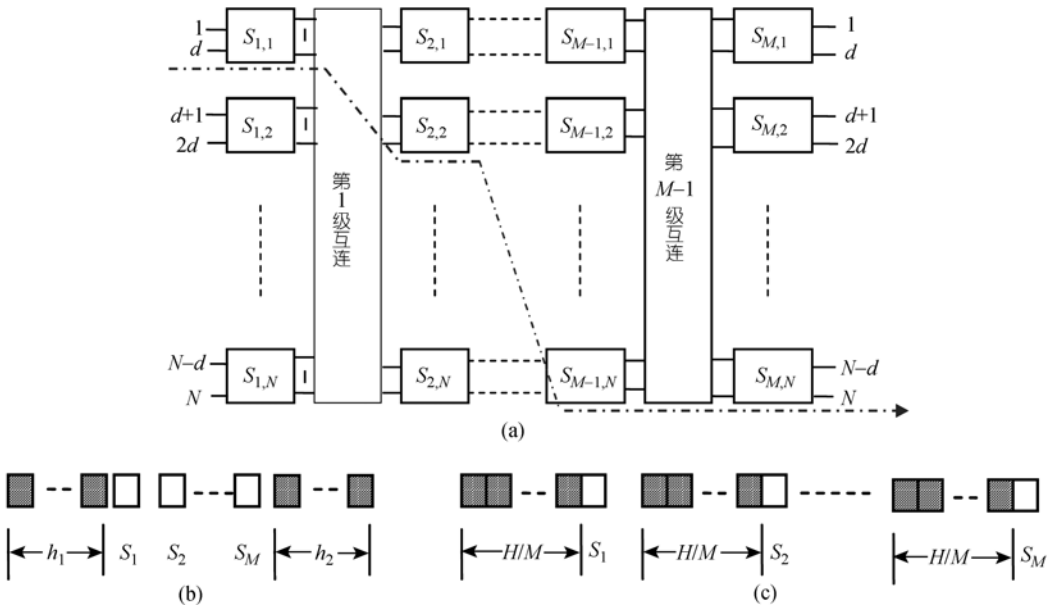

图 1 MPFS 体系结构的 FIS 处理机制

(a) 路由器多极交换处理流程; (b) FBS 处理流程; (c) FIS 处理流程

设报文转发处理共需 H 次访存, h_1 为输入(ingress)处理访存次数, h_2 为输出(egress)处理访存次数, $H=h_1+h_2$. FBS 处理流程分为 M+2 级, 第 1 级和第 M+2 级为网络处理器实现的转发处理, 第 2 级到第 M+1 级为交换处理. 由于网络处理器需要同时支持输入输出处理, 局部必须实现 H 次访存. FIS 机制将转发处理嵌入到交换路径上 M 个交换单元中实现分组转发, 平均每个单元局部只需 H/M 次访存, 因此实现简单. MPFS 体系结构基于 FIS 处理机制, 其中支持转发处理的交换单元称为 FSN.

2.2 MPFS 可扩展性分析

MPFS 将转发操作映射到多级交换网络中实现, 因此其扩展能力与多级交换网络的扩展性密切相关. Benes 网络由 2 个背靠背的蝶形网络组成, 由于内部具有多条等价的共轭路径以及理论上的无阻塞, 十分适合用作路由器的交换结构.

以下分析以 Benes 网络为例. 设交换网络端口数 $N=2^a$, FSN 端口数 $d=2^b$, 则每级交换单元数为 2^{a-b}, 交换级数 $M=(2a-b)/b$. 若将报文转发需要的 H 次访存平均分配到交换路径上的每个 FSN 上, 则每个 FSN 网络处理访存次数为 $Hb/(2a-b)$. 根据前面对网络处理器设计复杂性的分析, 可认为传统网络处理器复杂性为 H, FSN 网络处理复杂性为 $b/(2a-b)$. 若使用交叉点(cross- point)数衡量交换网络复杂性, 则单级 crossbar 复杂性为 $22a$. FSN 交换复杂性为 $22b$, MPFS 交换复杂性为 $2a-b\times(2a-b)/b\times22b=2a+b(2a-b)/b$, 与单级 crossbar 相比复杂性为 $(2a-b)/(b\times2a-b)$.

MPFS 模型转发和交换的扩展能力如图 2 所示. 当 FSN 规模固定时, 路由器规模(端口数 N)越大, 转发和交换的相对复杂性越低, 因此 MPFS 具有良好的可扩展能力. 当路由器规模(端口数 N)固定时, FSN 规模越小, 其网络处理和交换的复杂性越低. 因此基于 MPFS 模型, 可用较小的 FSN 构建大规模的路由器系统.

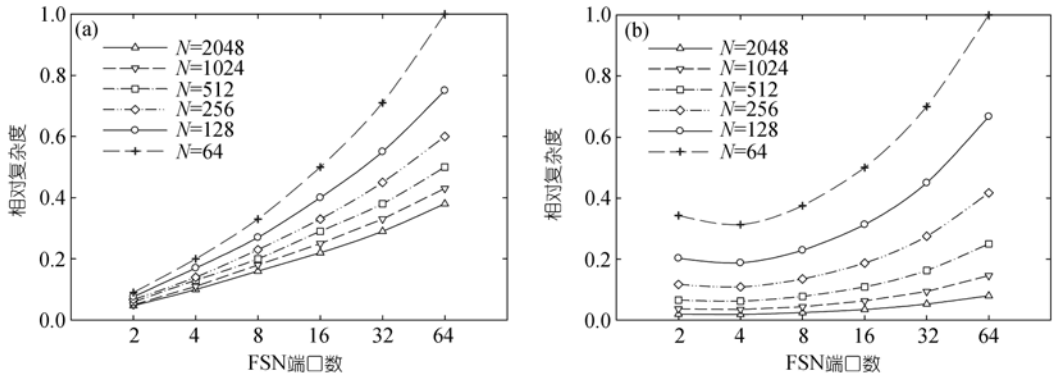

图 2　MPFS 可扩展能力

(a) 转发处理可扩展性; (b) 交换网络可扩展性

2.3　转发功能的映射

MPFS的核心是将各种转发操作映射到多级FSN上实现. 这些操作包括报文分类、FIB查找、QoS调度和计数器管理等. FIB查表是网络处理中最复杂, 需要访存次数最多的操作 [13], 直接影响网络处理器的复杂性和性能. 本文第 3 部分将对FIB查表的映射进行重点研究. 需要注意的是, 交换结构拓扑以及转发操作间的依赖关系对映射具有重要影响. 如输出调度必须映射到最后一级FSN, 而计数器管理必须在FIB查表之后进行等. 而实现FIB查表的FSN必须可达所有输出端口, 以有效支持报文单播和组播. 因此MPFS模型实现必须考虑交换拓扑的特点, 对各种操作的映射进行统筹划分. 对于Benes网络, 由于其前$a/b-1$ 级用于负载均衡, 后a/b级实现报文交换, 可将FIB查表功能映射到前$a/b-1$ 级FSN, 而其他依赖FIB查表结果的操作, 如计数器管理, 可映射到第a/b级到 $2a/b-2$ 级FSN, 而最后一级FSN实现输出端口调度.

随着内嵌缓冲区交换算法的逐渐成熟, MPFS 可采用基于报文的变长交换, 避免在输出接口实现复杂的信元重组. 此外, 虽然 Benes 等无阻塞交换网络内部链路无须加速比, 但为实现较好的服务质量控制, 需要在最后一级 FSN 上实现一定的加速比. 如 CRS-1 路由器在 Benes

交换网络的最后一级通过多个交换单元并行工作获取加速比. MPFS 也可在最后一级配置多个 FSN 并行工作, 在获取交换加速比的同时增加转发处理能力, 减轻输出调度实现的压力.

3 MPFS 中的 FIB 查找的实现

IPv6 分组转发、组播、MPLS 和 VPN 转发处理的核心都可归结为 FIB 查表问题, 因此本节对 MPFS 结构中 FIB 查找的实现方法进行研究.

3.1 MPFS 中 FIB 查表流程

设 FIB 表包含以下 10 个前缀: P_1(110, 1), P_2(0000, 3), P_3(00010, 2), P_4(11100, 4), P_5(11110, 1), P_6(000110, 3), P_7(111111, 3), P_8(0001100, 4), P_9(1110010, 2), P_{10}(1110011, 1). 其中 P_1(110, 1) 表示前缀 P_1 为 110, 对应输出端口 1. 上述前缀构造的 FIB 树如图 3(a)所示. 设 MPFS 结构的 FSN 阵列如图 3(b)所示. 因此将 FIB 树划分为 4 层.

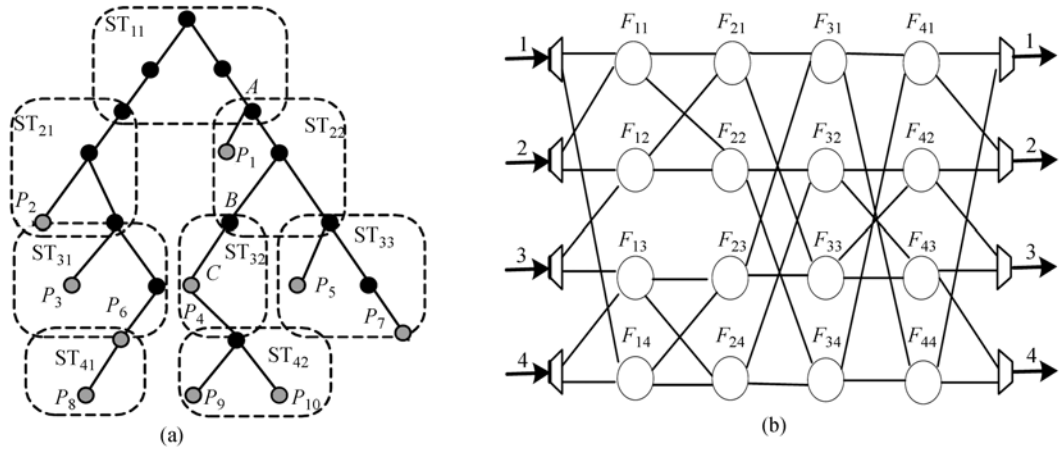

图 3 基于树分层和子树映射的 FIB 查表

(a) FIB 树的分割; (b) FSN 的拓扑结构

设第 i 层的第 j 棵子树标记为 ST_{ij}, ST_{ij} 可达输出接口表示该子树根节点可达的下层结点包含的输出接口的集合, 记为 STO_{ij}. 如 ST_{32} 代表第 3 层的第 2 个子树, STO_{32}={1,2,4}. 设第 i 级的第 j 个 FSN 结点标记为 F_{ij}, 从该 FSN 出发, 通过互连链路以及下级 FSN 可达的输出端口集合标记为 FO_{ij}. 易得 STO_{11}=STO_{22}={1,2,3,4}, STO_{21}={2,3,4}, STO_{31}={2,3,4}, STO_{32}={1,2,4}, STO_{33}={1,3}, STO_{41}={4}, STO_{42}={1,2}, FO_{11}=FO_{12}=FO_{13}=FO_{14}={1,2,3,4}, FO_{21}=FO_{22}=FO_{23}=FO_{24}={1,2,3,4}, FO_{31}=FO_{34}={1,2,4}, FO_{32}=FO_{33}={2,3,4}, FO_{41}={1,2}, FO_{42}={2,3}, FO_{43}={3,4}, FO_{44}={1,4}.

对于第 i 层子数的映射, 若将子树 ST_{ij} 映射(放置)到 F_{ik} 上, 必须满足 $STO_{ij} \subseteq FO_{ik}$. 例如 FO_{32}={2,3,4}, 而 STO_{32}={1,2,4}. 由于 F_{32} 无法将报文交换到输出端口 1, 子树 ST_{32} 不能被映射到 F_{32}. 根据上述规则, 可得子树到 FSN 的映射关系. F_{11}={ST_{11}}, F_{12}={ST_{11}}, F_{13}={ST_{11}}, F_{14}={ST_{11}}, F_{21}={ST_{21}, ST_{22}}, F_{22}={ST_{21}, ST_{22}}, F_{23}={ST_{21}, ST_{22}}, F_{24}={ST_{21}, ST_{22}}, F_{31}={ST_{32}}, F_{32}={ST_{31}}, F_{33}={ST_{31}}, F_{34}={ST_{32}}, F_{41}={ST_{42}}, F_{42}={}, F_{43}={ST_{41}}, F_{44}={ST_{41}}.

假设输入端口 P_1 到达分组的目的地址为 11100011, 该端口将该分组按照某种策略(如负载情况)送 F_{11}. F_{11} 检索子树 ST_{11}, 分组匹配到图 3(a)中 A 点, 指向子树 ST_{22}. 由于 ST_{22} 分布在所有第 2 级 FSN 上, 可将该分组(以及 ST_{22} 的标识)交换到任何一个第 2 级 FSN, 假设 F_{22}. F_{22} 查找子树 ST_{22}, 匹配到图 3(a)中 B 点, 即指向第 3 级子树 ST_{32}, 由于 ST_{32} 在 F_{31} 和 F_{34} 上, 可以选择交换到 F_{31} 或 F_{34} 上, 假设 F_{34}. F_{34} 接收到该分组(以及 ST_{32} 的标识)后, 查找子树 ST_{32} 匹配到图 3(a)中 C 点, 命中输出端口 4. F_{34} 根据本地拓扑信息, 将分组送 F_{44}, 同时携带已经得到的目的端口 4, 根据已经查到的结果, F_{44} 直接将分组送输出端口 4.

根据上述流程可以看出, 针对 FSN 阵列结构进行 FIB 子树分割, 并将其按某种规则映射到多级 FSN 是 MPFS 结构中 FIB 查表实现的关键.

3.2 FIB 子树映射算法

MPFS 中 FIB 查表的实现就是将 FIB 树的遍历过程映射到分组从输入端口到输出端口的交换路径上, 真正实现了在交换中转发. FIB 子树到 FSN 的映射算法包括初始化、子树映射和完备性检查 3 个阶段, 如表 1 所示.

表 1 FIB 子树到 FSN 的映射算法

```
定义
    S: 实现 FIB 查表的 FSN 层数
    M: 每层 FSN 的个数
    P_i: 第 i 层子树个数, 1≤i≤S
    F_i: 第 i 层 FIB 子树的集合, 1≤i≤S
    F_ij: 映射到 F_ij 的子树集合, 1≤i≤S, 1≤j≤M
    ok_i: 第 i 层映射是否完备, 1≤i≤S
初始化
    (1) 根据前缀集合生成 FIB 树, 将 FIB 树划分为 S 层, 计算子树的 STO_ij
    (2) 根据 FSN 的拓扑计算 FO_ij
    (3) for (i=1, i<S, i++)                    //对于每一层 FSN
            for (j=1, j<M, j++)                //对于每一个 FSN
                F_ij=∅, ok_i=0;                //清空所有映射
子树映射
    for (i=1, i<S, i++)                        //对于每一层 FSN
        for (j=1, j<M, j++)                    //对于每一个 FSN
            for (z=1, z<P_i, z++)              //对于每一个子树
                if ( STO_iz ⊆ FO_ij )
                    F_ij = F_ij ∪ {ST_iz};
完备性检查
for (i=1, i<S, i++)                            //对于每一层 FSN
    if ( F_i ⊆ F_i1 ∪ F_i2 ... ∪ F_iM )        //每一颗子树都至少映射到一个 FSN 上
        ok_i = 1
```

某层映射是完备的是指该层每个 FIB 子树都至少映射到一个 FSN 上. 显然, 当某层映射不完备时, 可能会造成转发处理错误. 例如图 3 中, ST_{33} 无法映射到任何一个第 3 层 FSN, 当某

个第 2 层 FSN 接收到目的地址以 1111 开始的分组时, 通过检索 ST_{22} 可得 ST_{33} 的指针, 由于 ST_{33} 不在任何一个第 3 层 FSN 上, 该报文转发失败. 解决映射不完备的有效方法是增加各级 FSN 的连通性. 连通性较差的拓扑还会造成 FSN 的子树集合为空, 如 $F_{42}=\varnothing$. 虽然这种情况不影响转发正确性, 但由于没有任何报文会转发到该 FSN, 造成转发处理资源的浪费.

容易证明, 对于任何 $1\leqslant j\leqslant M$, FO_{ij} 都等于所有输出接口集合时, 一定有 $ok_i=1$ 且 $F_{ij}=F_i$. 本文以下分析均假设 $F_{ij}=F_i$, 即同一层 FSN 保存相同的 FIB 子树信息.

3.3 FSN 上的 FIB 查表算法

经过子树映射, 每个FSN会保存一个或多个FIB子树. 每个到达FSN的报文将携带由上一级FSN写入的本级子树标签. 本级FSN可根据该标签迅速找到该子树入口, 继续报文的FIB查表流程. 理论上任何无回溯的树遍历查找算法都可在FSN FIB子树查找中使用. 本文以下分析基于树位图算法 [13]. 根据树位图算法特点, FIB变化只触发对局部修改的子树进行更新. 因此MPFS路由器控制平面可将变化的FIB子树信息广播到所有FSN, 每个FSN独立的更新本地保存的FIB子树. 由于多级FIB子树的更新是并行的, MPFS中FIB更新性能优于传统网络处理器.

3.4 性能评测

目前IPv6骨干网络中FIB表项规模在1000左右, 因此不足以评估IPv6 FIB查找算法的性能. 性能评估必须采用模拟生成的大容量IPv6 FIB表. 由于随机生成的FIB表不具备代表性 [19], 本文针对IPv6 地址分配策略, 采用Cisco公司对IPv6 前缀长度分布的预测结果 [20](以下简称Cisco分布)及CERNET2 真实IPv6 FIB的前缀分布 1)(以下简称Cernet分布)构造大容量FIB表, 并根据该表对MPFS中FIB的实现性能进行评测. Cisco和Cernet的前缀分布如图 4 所示.

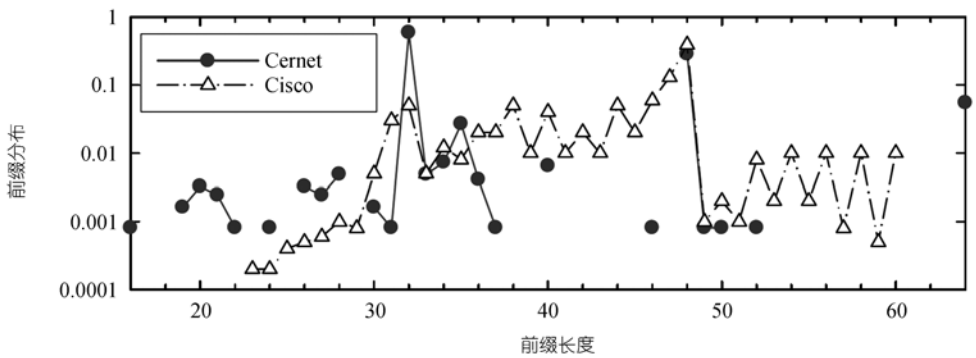

图 4 Cisco 合成 IPv6 前缀长度分布和目前 Cernet 真实 IPv6 前缀长度分布

性能评测将子树映射到 4 级 FSN 上, 每个 FSN 均可达任意输出端口. 根据树位图算法思想, 将 FIB 树 64 位前缀的 1~12 位进行初始阵列优化, 13~64 位以 4 为步长, 划分为 12 层子树. 由于第 8 层(对应前缀 32)和第 12 层(对应前缀 48)子树数目较多, 需要的存储空间较大, 而其他

1) http://bgpview.6test.edu.cn/datav6/

子层子树存储需要的空间则相对较少. 综合存储空间和查表性能 2 方面因素, 本文采用映射方法如表 2 所示.

表 2　子树层到 FSN 的映射关系

FSN 级数	映射子树层	对应前缀长度	最大访存次数
1	初始阵列、第 1~3 层	1~24	7
2	第 4~7 层	25~40	8
3	第 8~9 层	41~48	4
4	第 10~12 层	49~64	8

FIB树构造采用初始阵列、位向量分离、CAM存储和路径压缩 4 种存储优化方法 [13]. 其中只有CAM结点保存前缀的下一跳索引, 其他结点中前缀的下一跳索引保存在result结点中. 因此FSN最大访存次数是查找每级子树都同时命中CBV和PBV时需要的访存次数. 由于每个FSN保存的最后一层子树, 即按表 2 映射的第 3, 7, 9, 12 层子树的下一级子树在下一级FSN上可能不会连续存放, FSN的最后一层子树需保存后续子树在下一级FSN上的入口地址. 设系统支持 2048 个输出端口和每层最多 1 M个子树 1). 根据树位图算法的计算方法, 每个子树节点需要 41 bit存储.

我们采用 Cisco 和 Cernet 两种前缀分布生成了具有不同前缀数目的 IPv6 FIB 表. 根据表 2 映射方法得到的每级 FSN 结点的存储开销如图 5 所示. 其中 Stage1~4 代表每级 FSN 的存储需求, Tree Bitmap 代表传统网络处理器上按树位图方法组织 FIB 表的存储开销. Extra 表示系统为保存 FSN 子树入口地址的额外存储开销.

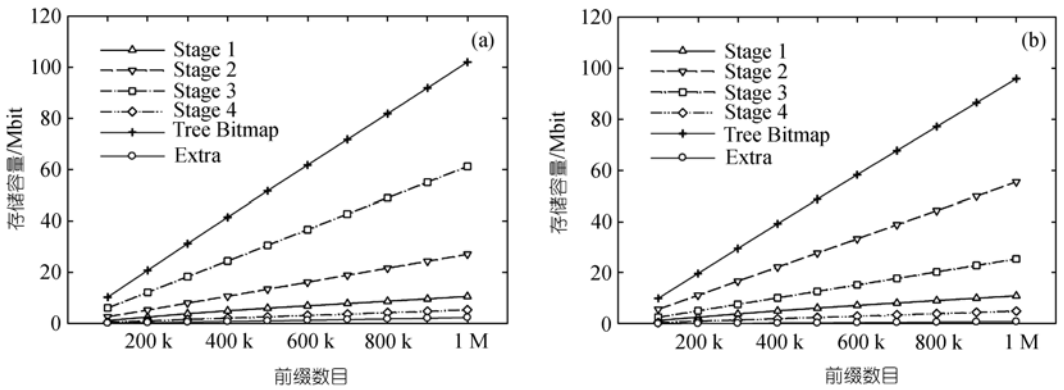

图 5　FSN 结点的存储开销

(a) Cisco 前缀长度分布比例下的存储开销; (b) Cernet 前缀长度分布比例下的存储开销

在空间复杂性方面, 由图 5 可以看出, Cernet 前缀分布下第 2 级 FSN 的存储开销最大, 而在 Cisco 前缀长度分布下第 3 级 FSN 的存储开销最大. 因此 FSN 的存储开销与前缀长度分布特性密切相关. 同时, 传统网络处理器需要的存储开销约为 FSN 最大存储开销的 2~18 倍

1) 当前缀数目为 1M 时, Cisco 分布下第 9 层子树数目最多, 为 580517 个; Cernet 分布中, 第 5 层子树数目最多, 为 532290 个. 因此选择每层支持 1 M 个子树是合理的

(Cisco 分布)或 2~20 倍(Cernet 分布). Extra 为树位图算法在 MPFS 结构中分布实现带来的额外开销, 与树位图算法的整体开销相比, 这部分开销可以忽略.

在时间复杂性方面, FSN 实现 FIB 查找最大访存次数为 8 次, 而传统网络处理器实现至少需要 27 次, 因此 MPFS 中 FIB 实现方法简化了网络处理器设计复杂性.

4 160 T 路由器的实现方案

从工程实现角度, MPFS 路由器有 2 个优点. 一是 FSN 逻辑可由单片 ASIC 实现, 且无须使用 SRAM; 二是可用大量同构 FSN 节点的堆叠实现可扩展的转发交换平台. 目前单片 RLDRAM II 存储器包含 8 个 bank, 每个 bank 最大容量为 72 Mbit, 数据接口宽度 36 位. 当芯片工作在最高频率 533 MHz 时, TRC 为 8 个 TCK, 因此随机访问延时最小为 15 ns. 根据图 6, 若将包含 1 M 个 IPv6 前缀的 FIB 表映射到 4 级 FSN 上, 每个 FSN 需要最大存储空间为 60 Mbit 左右. 若将 FIB 复制 8 份保存在 8 个 bank 中, 且由 2 片 RLDRAM 通过宽度级联组成一个访问通道, 那么每个时钟周期可读出一个子树节点. 即支持的 FIB 查表性能为 533 M/8=66.7 Mpps. 若将 FSN 上 FIB 子树存放到两个独立的 RLDRAM 通道中, 通过流水可获得 133 Mpps 的查表性能, 远大于 40 G 接口报文到达速率. 因此单 ASIC 芯片可实现 4 个 40 Gbps 网络处理引擎和 1 个 320 Gbps 交换核心. 每个网络处理引擎外部接口包括 40 Gbps 双向数据接口(Serdes 速率为 6.25 Gbps), 2 个 36 位 RLDRAM II 通道, 1 个 128 位 DDR3 DRAM 通道, 芯片合计引脚数目小于 1300. 另外, 芯片还可增加一个标准 AMC(advanced mezzanine card)接口, 用来扩充专用协处理器. 如在第 1 级 FSN 上扩充 TCAM 以支持线速报文分类.

160 T路由器包含 2048 个 40 Gbps端口, 每个FSN具有 4 个输入输出接口, 采用Benes多级交换网络连接. 因此共需FSN 5120 个, 分为 10 级, 每级 512 个. 若每块 60 cm×60 cm PCB插件集成 4 个FSN或 8 个 40 Gbps网络接口, 则 5120 个FSN需要 1280 块PCB插件, 2048 个接口需要 256 块PCB插件. 若单机柜组装 16 块插件, 则 80 个FSN机柜和 16 个接口机柜即可实现 160 Tbps的路由器. 显然, 基于MPFS体系结构实现的T比特路由器不但在单位体积的转发交换容量方面远高于目前世界上性能最高、可扩展能力最强的CRS-1 路由器 [1)], 而且转发交换阵列全部由相同的FSN PCB组装, 因此维护简单, 性价比高.

5 结束语

针对以 IPv6 为核心的下一代互联网对网络核心交换设备的要求, 本文提出一种可扩展的新型路由器体系结构——MPFS. 并对 MPFS 基本设计思想、扩展能力、关键转发功能实现进行了分析. 当然, 距离 MPFS 结构的实用化还有很多具体的问题需要进一步研究.

这些问题包括: ①转发平面的控制与管理问题. 特别是分布在不同 FSN 上 FIB 子树更新的同步问题; 分布存储的状态的一致性维护问题等; ②流量控制问题. 基于反压的多级多粒度流控机制是保证多级交换网络交换性能的重要手段, 特别在 MPFS 结构中, 有效的流控机制对减小 FSN 队列管理和调度复杂性具有重要意义. ③体系结构级的节能问题. MPFS 结构在源和目的间存在多条转发交换路径, 原理上可根据流量的大小控制转发交换阵列中活跃和休眠的

1) Cisco CRS-1 路由器使用 80 个机柜实现 92 Tbps 的转发交换能力

FSN 个数, 以在体系结构级实现有效的功率管理.

针对上述问题, 课题组正加紧对 MPFS 体系结构模型进行理论上的分析, 并开始原型验证系统的研发, 希望通过理论上的分析和原型验证系统的研制和相关试验, 对 MPFS 的各项关键技术进行更加深入的研究.

致谢 感谢张子文、王慧、马思瑶和袁宗仪对关键算法模拟所做的工作.

参考文献

1 Gupta M, Singh S. Greening of the Internet. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003. 19—26
2 Christensen K J, Nordman B, George A D. The next frontier for communications networks: power management. Comput Commun, 2004, 27(18): 1758—1770 [DOI]
3 Meyer D, Zhang L, Fall K. Report from the IAB workshop on routing and addressing. IETF RFC4984. September 2007
4 Turner J S, Taylor D E. Diversifying the Internet. In: Proceedings of the Global Telecommunications Conference, 2005. 1110—1123
5 Gripp J, Stiliadis D, Simsarian J E, et al. IRIS optical packet router. J Opt Netw, 2006, 5(8): 589—597 [DOI]
6 Yang L, Dantu R. Forwarding and control element separation (ForCES) framework. IETF RFC3746. April 2004
7 Sapountzis G, Katevenis M. Benes switching fabrics with O(n)-complexity internal backpressure. IEEE Commun Mag, 2005, 43(1): 88—94 [DOI]
8 Turner J, Yamanaka N. Architectural choices in large scale ATM switches. IEICE Trans Commun, 1998, E81-B(2): 120—137
9 Chao H J, Liew S Y, Jing Z G. A dual-level matching algorithm for 3-stage Clos-network packet switches. In: Proceedings of the 11th Symposium on High Performance Interconnects, 2003. 38—43
10 Aslam A, Christensen K J. A parallel packet switch with multiplexors containing virtual input queues. Comput Commun, 2004, 27: 1248—1263 [DOI]
11 Keslassy I, Chuang S T, Yu K, et al. Scaling internet routers using optics. In: Proceedings of the 2003 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003. 189—200
12 Sherwood T, Varghese G, Calder B. A pipelined memory architecture for high throughput network processors. In: Proceedings of the 30th International Symposium on Computer Architecture, 2003. 288—299
13 Eatherton W, Varghese G, Dittia Z. Tree bitmap: hardware/software IP lookups with incremental updates. ACM SIGCOMM Comput Commun Rev, 2004, 34(2): 97—122 [DOI]
14 Shreedhar M, Varghese G. Efficient fair queuing using deficit round-robin. IEEE/ACM Trans Netw, 1996, 4(3): 375—385 [DOI]
15 Chung F, Graham R, Varghese G. Parallelism versus memory allocation in pipelined router forwarding engines. In: SPAA, 2004
16 Basu A, Narlikar G. Fast incremental updates for pipelined forwarding engines. IEEE/ACM Trans Netw, 2005, 13(3): 690—703 [DOI]
17 Shah D, Iyer S, Prabhakar B, et al. Maintaining statistics counters in router line cards. IEEE Micro, 2002, Jan-Feb: 76—81
18 Ramabhadran S, Varghese G. Efficient implementation of a statistics counter architecture. In: SIGMETRICS, 2003
19 Harsha N, Ramesh G, George V. The impact of address allocation and routing on the structure and implementation of routing tables. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003. 125—136
20 Wang M, Deering S, Hain T. Non-random generator for IPv6 tables. In: IEEE Symposium on High Performance Interconnects, 2004. 35—40