学术论文

高性能安全路由器 BW7000 的设计与实现

徐明伟，徐 恪，熊勇强，江 勇，孙晓霞，吴 剑，喻中超
（清华大学计算机科学与技术系，北京 100084）

[摘要] 高性能和安全是计算机网络研究的两个主要问题。路由器在保证转发性能的前提下提供网络安全保护已经成为当前的研究热点。文章介绍了在完成国家“八六三”计划重大课题“高性能安全路由器”的过程中解决的若干关键技术问题。高性能安全路由器 BW7000 基于自主设计的高性能路由器操作系统 HEROS。为保证高性能的路由转发，设计实现了基于 RAM 的高性能路由查找算法；为支持服务质量控制和安全管理，设计实现了基于无冲突 Hash-Trie 树的分组分类算法和基于反馈的分布式分组调度算法；为保证网络安全，提出了基于分布式密钥管理的路由器安全体系结构。

[关键词] 路由器；安全；路由器操作系统；路由查找；分组分类；分组调度
[中图分类号] TP393 [文献标识码] B [文章编号] 1009－1742（2002）03－0054－09

1 引言

路由器是 Internet 的核心，网络带宽超摩尔定律的增长对路由器的性能提出了更高的要求，网络安全问题也日益突出，这已成为当前的研究热点。在骨干网核心路由器上实现安全功能是不现实的，因为安全功能会极大地降低分组转发的性能，导致骨干网络带宽浪费；即使在核心路由器上实现了安全功能，由于这些功能没有明确的应用需求，很难得到充分利用。在骨干网络的边缘路由器上实现安全功能是比较好的选择。骨干网络的边缘路由器不同于网络末端的接入路由器，它一般也需要有高速背板（吉比特以上）和多个高速接口（例如 Gigabit Ethernet 和 OC－3 POS）。如何在这种高性能的边缘路由器上实现完善的安全功能而又不影响分组转发性能是一个亟待解决的问题。

清华大学计算机系承担的国家“八六三”计划重大课题“高性能安全路由器”——BW7000 路由器正是满足以上要求的边缘路由器。BW7000 路由器以自主设计的高性能路由器操作系统（HEROS）为基础，接口类型包括 Gigabit Ethernet，OC－3 POS，Fast Ethernet 等多种高速接口。基于硬件的加密算法使得加解密具有较高的性能。

BW7000 路由器研制过程中解决了一系列关键技术问题，设计实现了高性能路由器操作系统 HEROS、基于 RAM 的高性能路由查找算法、基于无冲突 Hash-Trie 树的分组分类算法和基于反馈的分布式分组调度算法，提出了基于分布式密钥管理的路由器安全体系结构，文章还给出了系统实现和部分测试结果。

2 BW7000 路由器结构

2.1 BW7000 路由器的硬件结构

BW7000 路由器硬件采用了基于共享总线的分布式多处理器结构，如图 1 所示。这种结构包括三级功能模块：中央处理器模块（CPM）完成路由器的操作配置、网络管理和维护路由表等功能；扩展处理器模块（PM）完成 IP 报文的转发和安全过滤等功能，并实现高速路由查找算法；接口单元（IU）主要完成对高速通信接口的驱动。利用随机

[收稿日期] 2001－08－20；修回日期 2001－09－30
[基金项目] 国家“八六三”高技术研究发展计划资助项目（863－306－ZD－07－01）
[作者简介] 徐明伟，男，（1971－），辽宁朝阳市人，博士，清华大学副教授

Petri 网对该结构建立了模型并进行了理论分析[1]，结果表明，这种分布式结构具有很好的可扩展性和较高的性能。

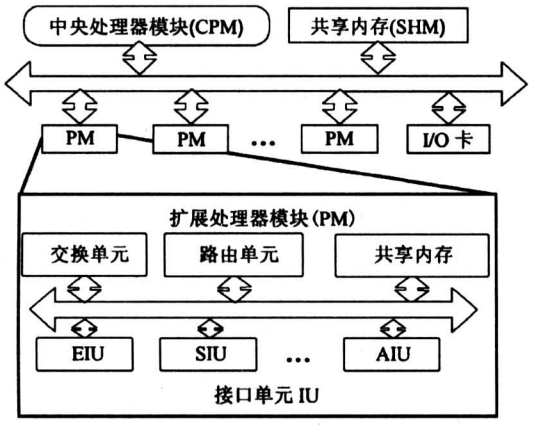

图 1　BW7000 路由器的硬件结构

Fig.1　Hardware architecture of the router BW7000

2.2　BW7000 路由器的软件结构

在分布式硬件结构的基础上，BW7000 路由器的软件结构也是分布式的，如图 2（未画出安全子系统，见第 5 节）。BW7000 路由器支持多种路由协议和网管协议，提供服务质量控制和安全等多种功能。

BW7000 路由器的软件结构由 7 部分组成：

1）BW7000 路由器的操作系统；

2）操作管理子系统；

3）路由协议子系统；

4）支撑子系统；

5）转发子系统；

6）接口单元子系统；

7）安全子系统。

在中央处理器模块上主要运行第 1，2，3，4 子系统和第 5 子系统的一部分，在多个扩展处理器模块上主要运行第 1，5，6 子系统。

3　BW7000 路由器操作系统 HEROS

3.1　HEROS 的结构

HEROS 的结构如图 3 所示。为了适应高性能安全路由器的分布式硬件结构，HEROS 是一种分布式的多任务实时操作系统，整个系统可以在多个具有自治能力的物理节点上运行，每个自治节点都有自己的 CPU、内存和输入输出设备，都运行单处理器的实时多任务 HEROS 内核，内核实现了常用实时操作系统的主要功能。在各个 HEROS 内核上采用一种全局的分布式消息通信机制，可以透明地在各个节点的任务之间传递消息，路由器中的 IP 协议和其他上层协议基于该通信机制实现。

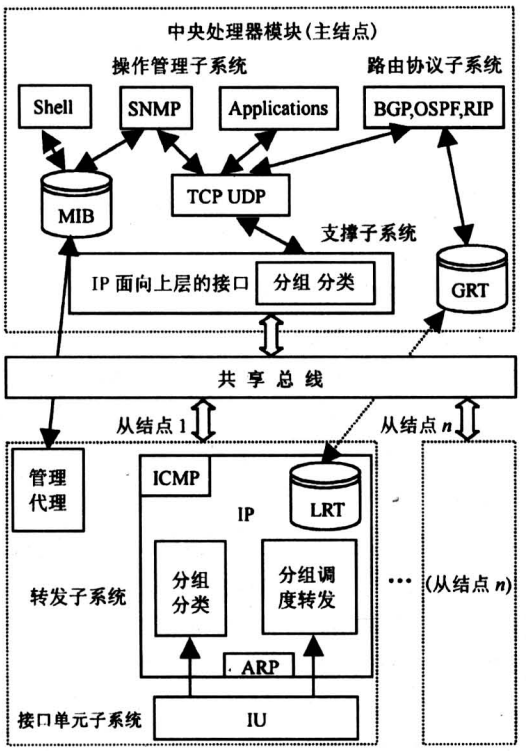

图 2　BW7000 路由器的分布式软件结构

Fig.2　Distributed software architecture of the router BW7000

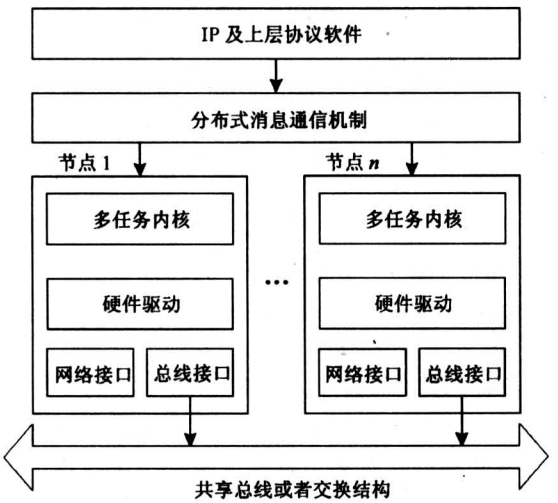

图 3　HEROS 的结构

Fig.3　Architecture of HEROS

3.2 HEROS 中的分布式消息通信机制 DMQ

为了适应分布式处理的需要，对 HEROS 的单处理器内核进行了扩展，实现了分布式消息队列 DMQ（distributed message queue），其特点：

1）DMQ 是一种轻负载的、与传输介质无关的、具有容错能力的分布式消息通信机制，基于 DMQ 可以开发分布式应用；

2）DMQ 可以保证系统中不出现单一故障点，这是在整个多节点系统中通过复制多份系统对象数据库来实现的；

3）DMQ 支持消息的单播和多播传送；

4）DMQ 具有位置透明性，对象可以在系统中移动而不用重新编写应用代码。

DMQ 的组成见图 4，由一组服务和分布式对象数据库组成。DMQ 提供的服务包括：分布式消息队列服务，处理来自远程节点的分布式消息队列消息；分布式名字数据库服务，处理来自远程节点的分布式名字数据库消息；分布式节点组相关服务，处理来自远程节点的分布式消息队列组数据库消息，包括组消息队列服务和组选举协议服务（用于选择单一的组 ID 号）；分布式节点协作服务，处理节点状态维护协议，保证系统中各节点运行状态一致。

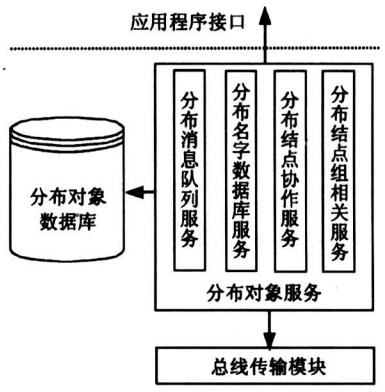

图 4 分布式消息队列服务

Fig.4 Services of distributed message queue

DMQ 维护的分布式对象数据库包括如下几部分：分布式名字数据库，保存系统中全局对象的名称、值和类型；在多节点系统中的每个节点上都有分布式名字数据库，这样可以保证任何节点上的任务都能够找到系统中的全局对象；分布式组数据库，保存分布式消息队列组的信息；分布式节点数据库，维护系统中节点的信息和当前的状态。

总线传输模块负责通过总线收发消息，分为发送模块和接收模块。发送模块提供原语调用，接收模块监听总线，有消息到来时将其放入临时缓冲区等待任务接收。总线传输模块和底层硬件驱动之间是自定义的通用接口，通用接口机制可以使消息通信机制与底层硬件通信机制无关。

4 面向高性能分组转发的若干算法

面向高性能分组转发的算法包括：路由查找算法，分组分类算法和分组调度算法。这些算法共同完成了高性能的分组转发处理过程。

4.1 基于 RAM 的路由查找算法

高性能安全路由器的接口有四类：Fast Ethernet，Gigabit Ethernet，OC－3 POS 和 E1。对于 Gigabit Ethernet 接口，如果按照每个分组长度为 64 B（字节）计算，那么该接口单元至少需要达到 1 000／（64×8）＝ 1.95 Mb/s 的路由查找速率才能以线速转发分组。传统的基于 Hash 链表和 Trie 树的路由查找算法远不能满足如此高速的分组转发要求。提出了一种基于 RAM 的快速路由算法，并在此基础上，结合 Hash 链表和 Trie 树路由查找算法的特点，提出一种可配置的路由查找算法。该算法可以动态配置评价系数，适用于多种网络环境。

基于 Hash 的路由查找方案中最快的 Hash 函数是线性函数 H（address）＝ address，即直接使用 IP 地址作为 Hash 表的索引。使用这种 Hash 函数进行地址查找只需要一次存储器访问操作，查找性能最优。但是由于 IP 地址需要用长度 32 b 表示，这样的 Hash 表表项数目需要 2^{32}项。受硬件条件的限制，目前的路由器还不能满足如此大的内存容量需求，实际应用意义不大。

快速路由查找算法采用了线性 Hash 函数的思想，并借鉴文献［2，3］的算法设计思想。算法的数据结构中包括两种查找表结构，分别称为 Table16 和 TableNext。Table16 保存路由地址前缀≤16 b 的表项，该查找表以 IP 地址的前 16 b 作为索引进行查找，共包含 2^{16}个表项，代表从 0.0 到 255.255 的路由前缀项。TableNext 保存路由地址前缀 ＞16 b 的表项，表项中保存对应路由地址的路由信息。只要 Table16 中某表项对应的路由项中至少包含一个前缀＞16 b 的路由项时，就需要为该 Table16 表项分配一张 TableNext 表，每张 TableNext 表的表项数目均为 2^{16}个。假设在空的查

找表中插入一条新的路由前缀表项 Addr，如果 Addr 的前缀长度 ≤16 b，那么它只需要被保存在 Table16 表中，然后将 Table16 中相对应表项的第一位置成 0，表示表项剩余字段保存的是该表项的路由信息；如果 Addr 的前缀长度＞16 b，那么需要根据 Addr 的前 16 b 地址找到 Table16 中的对应表项，然后将该表项的第一位置成 1，表示表项剩余字段保存的是 TableNext 的指针，最后分配相应的查找表 TableNext，并将该表地址指针保存到 Table16 相应表项中。各表具体的表项结构如图 5 所示。

图 5　查找表表项结构

Fig.5　Architecture of lookup table item

在每条路由表项的插入过程中，有可能需要更改查找表中多条表项的内容。例如插入地址前缀 166/8 时，需要将查找表 Table16 中从 166.0 到 166.255 共 256 个表项的内容修改成路由项 166/8 的路由信息。

当需要决定待查 IP 地址的路由时，将该 IP 地址的前 16 b 作为 Table16 表的索引，通过一次查找表访问找到 Table16 中对应表项，并判断该表项的第一位值：若为 0，表项剩余字段就是对应的路由信息；若为 1，表项剩余字段是 TableNext 表的指针，根据该指针以及 IP 地址中的后 16 b 地址，得到 TableNext 中对应该 IP 地址的路由信息。

对算法实现的测试结果表明，当表项数目达到 100 000 时，该算法仍然能够达到 6.5 Mb/s 的查找速度，完全能够满足吉比特路由器的高速转发要求。

基于 RAM 的路由查找算法查找速度快，但即使经过一定的改进，所需的存储器容量仍然比较大。Hash 链式表查找算法和动态前缀树查找算法的查找速度一般，但是所需的存储器容量较小。因此，把基于 RAM 快速查找算法和 Hash 链式表查找算法、动态前缀树查找算法结合起来，发挥各自算法的优点，以获得查找算法的更好整体性能。

根据三种算法的各自特点，可以定义评价函数为 $A(N, L) = S(N, L) / M(N, L)$，其中 $S(N, L)$ 为算法查找速度（查找过程算法复杂度的倒数），$M(N, L)$ 为算法所需的存储容量，N 为表项的数目，L 为 N 个表项中最长的表项长度。因此，评价函数反映了查找算法的整体性能。我们设计和实现了基于评价函数的可配置路由查找算法[4]。

4.2　无冲突的 Hash-Trie 树分组分类算法

路由器安全功能和服务质量控制都需要对到达的 IP 分组进行分类。目前的 IP 分类算法制约了路由器分组转发性能。为提高性能，在 Grid of Tries 算法[5]的基础上提出了无冲突的 Hash-Trie 树分组分类算法。该算法的时间和空间平均复杂度都比 Grid of Tries 多维分类好，并且消除了 Grid of Tries 对过滤规则的严格限制。

有 7 个域原则上可选为过滤规则的域：源/目的 IP 地址，源/目的传输层端口，TOS，协议域和传输层协议标志。实际上过滤规则并不对所有的域都感兴趣。对一些 ISP 的过滤规则数据库的统计表明，17 %的过滤规则指定了 1 个域，23 %的过滤规则指定了 3 个域，60 %的规则指定了 4 个域。

表 1 是一个具有 6 条过滤规则的数据库。以目的端口为例，0～65 535 中所有可能值对应一个位图，表示一个端口值与哪些过滤规则匹配。例如，端口 21 和 22 的位图都是 100111，表示都与过滤规则 0，3，4，5 匹配。根据不同位图，将 0～65 535 中所有可能的目的端口值划分为不相交的等价类，等价类 a 的位图记为 $b_{mp}(a)$。目的端口等价类总数记为 D，所有目的端口等价类的集合记为 D_{set}。根据表 1 构造的 D_{set} 为 {{80}, [20, 21], {0～65 535 中除 20，21，80 之外的其他值}}。源端口等价类集合和协议号等价类集合分别记为 S_{set} 和 P_{set}，其等价类数目分别记为 S 和 P。

表 1　一个过滤规则数据库示例

Table 1　An example of filter database

CLASSID (分类 ID)	DEST-IP (目的 IP)	SRC-I (源 IP)	DEST-PORT (目的端口)	SRC-PORT (源端口)	PROTOCOL (协议号)
0	10.1.*.*	10.2.*.*	*	*	*
1	10.3.*.*	10.4.*.*	80	*	17
2	10.5.*.*	10.6.*.*	80	*	17
3	10.5.*.*	10.6.*.*	[20, 21]	*	6
4	10.7.*.*	10.7.*.*	*	gt 1023	6
5	*	*	*	*	*

如果 $a \in D_{set}$，$b \in S_{set}$，$c \in P_{set}$，则三元组

(a, b, c) 称为一个交叉组合。将所有的交叉组合的集合进一步划分成不同的等价类，该等价类的集合记为 C_{DSPset}。划分方法如下：两个交叉组合 (a, b, c), (d, e, f)，如果 $b_{mp}(a)$ & $b_{mp}(b)$ & b mp (c) 与 $b_{mp}(d)$ & $b_{mp}(e)$ & $b_{mp}(f)$ 相同，则 (a, b, c) 和 (d, e, f) 属于同一个交叉组合等价类，否则属于不同的等价类。

与 C_{DSPset} 中每个元素对应的有一个目的 - 源 IP 前缀对集合，其元素为与 C_{DSPset} 中某个元素的端口和协议号匹配的过滤规则的目的 IP 和源 IP 前缀对。无冲突的 Hash-Trie 树算法：根据包头 H 中的目的端口、源端口和协议号，通过无冲突的 Hash 查找方法，找到与上述交叉组合元素匹配的目的 - 源 IP 前缀集合的指针；然后在目的 - 源 IP 前缀对中对 H 进行二维的 IP 分类，得到最终的分类结果。属于同一个 C_{DSPset} 元素的交叉组合共享一个目的 - 源 IP 前缀对集合指针。

对一个包头 H $(d_{port}, s_{prot}, p_{roto})$（分别表示目的端口、源端口和协议号）分别以 d_{port}, s_{port}, p_{roto} 为索引查表，得到 f_d (d_{port}), f_s (s_{port}), f_p (p_{roto})，然后以其某个函数 g (f_d, f_s, f_p) 为索引再进行查表，得到 h $(g(f_d, f_s, f_p))$，此值即为目的 - 源 IP 集合的指针。对 D_{set} 中所有的 D 个等价类以 0, 1, 2, …, $D-1$ 依次编号，f_d 定义为端口 d_{port} 对应的等价类号。f_s 和 f_p 用同法定义。上述过程见图 6，其中方框表示查表，g 为 Hash 函数。

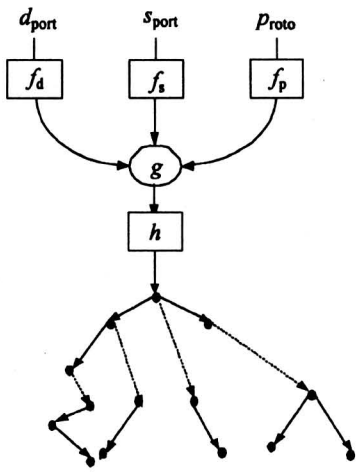

图 6 无冲突的 Hash-Trie 树算法

Fig.6 Algorithm of non-collision Hash-Trie tree

建立上述各表的算法描述见算法 1 和算法 2。

算法 1 f_d 表的构建（f_s 和 f_p 表可用同法构建）：

1) 初始化：为 f_d 表分配 65 536 个表项空间，等价类计数器 c_c 初始化为 0；

2) 对 n 从 0 到 65 535 做第 3 步和第 4 步；

3) 扫描过滤规则的 d_{port} 域对应的列，求 n 的 b_{mp} (n)；

4) 如 n 的 b_{mp} (n) 出现过，则 n 属于 b_{mp} (n) 对应的 D_{set} 等价类，将 n 属于的等价类的 i_d 填入表 f_d 的第 n 个表项；否则 n 属于新的等价类，该等价类的 i_d 值为 c_c 当前值，将其填入表 f_d 的第 n 个表项，同时 c_c+1。

算法 2 h 表的构建：

1) 为 h 表分配 $D \times S \times P$ 个表项空间，同时初始化等价类计数器 $c_c=0$, $n=0$；

2) 对 D_{set}, S_{set} 和 P_{set} 的等价类元素依照 i_d 递增顺序进行交叉组合；对每个 $e_{qd} \in D_{set}$, $e_{qs} \in S_{set}$, $e_{qp} \in P_{set}$，做第 3 步；

3) 求 $b_{mp} = b_{mp}(e_{qd})$ & $b_{mp}(e_{qs})$ & $b_{mp}(e_{qp})$（逐位与），如果 b_{mp} 已经出现过，则 n 属于 b_{mp} (n) 对应 C_{DSPset} 的等价类，将 n 属于的等价类的 i_d 填入 h 的第 n 个表项；否则 n 属于新的等价类，该等价类的 i_d 值为 c_c 当前值，将其填入表 h 的第 n 个表项，同时 c_c+1, $n+1$。

在最坏情况下，得到二维 Trie 树的指针需要经过 4 次查表（串行查找），即查找 f_d, f_s, f_p 和 h。沿二维 Trie 树查找，最坏情况下需要查找的最多节点数目为 $2W$（W 为 IP 地址宽度），因此需要访存 $2W+4$ 次，查找时间基本上与过滤规则的数目无关。

4.3 带反馈的分布式分组调度算法

BW7000 路由器的转发调度机制采取组合输入输出排队体系结构，在输入和输出端口都有基于每流的排队。在输入端口的每流排队，按照输出端口的不同组成多个虚拟输出队列（VOQ），这样可以有效地解决队头阻塞问题。

流的输入控制中，路由器实际分配给流 f 的速率 r_f 等于流 f 要求分配的速率 R_f，r_f 会随反馈信息有所变化。每个流有一个起始和结束时间标记，调度器根据这些标记来进行分组调度。实现的分组公平调度算法是 WF^2Q[6]。

对基于分组公平排队（PFQ）的分组调度器

$P_{FQ}(i, j)$，虚拟时间函数为：

$$V_{i,j}(f) = \max\{v(\tau) + t - \tau, \min_{n \in B(t)} S_{i,j,n}(t)\}, \tag{1}$$

τ 为 t 时刻前一个分组调度输出的时间，$B(t)$ 为 t 时刻有空队列的流集合；

$$S_{i,j,f}(t) = \begin{cases} F_{i,j}(t) & \text{分组 } p_f^k \text{ 服务结束时,} \quad (2.1) \\ \max\{V_{i,j}(t), F_{i,j,f}(t)\} & \text{分组到来时流 } f \text{ 队列为空;} \quad (2.2) \end{cases}$$

$$F_{i,j,f}(t) = S_{i,j,f}(t) + L_f^k / r_f。 \tag{3}$$

$V_{i,j}(t)$ 按式 (1) 计算；$V_{i,j}(0) = 0$，$F_{i,j,f}(0) = 0$；各流的起始和结束时间标记按照式 (2) 和式 (3) 计算。初始 $V_{i,j}(t) = 0$，各流的起始时间标记均为 0。虚拟时间的更新和各流起始时间标记只在下面两种情况下计算：一种情况是，流 f 的分组到来时其队列为空，此时按照式 (2.1) 计算新的 $S_{i,j,f}(t)$，所用的 $V_{i,j}(t)$ 是按照式 (1) 计算的新值，从而可以计算出结束时间标记；另一种情况是，调度流 f 分组 p_f^k 后，按照式 (2.2) 和式 (3) 重新计算该流起始时间标记，若流队列中还有分组，还需计算新的结束时间标记。之后，选择调度有最小结束时间标记的流分组。对输出端的分组调度器 $P_{FQ\,j}$，其虚拟时间函数为 $V_j(t)$，也是采用相同的算法进行调度输出。

设计了一种基于每流定时预测的拥塞反馈机制，以有效避免拥塞发生。反馈机制原理如下：对某一个输出端 i 而言，当输入到这个输出端的流 f 的速率 $r_{fin,i}$ 大于输出的速率 $r_{fout,i}$ 时，输出端队列中的元素呈增长趋势，拥塞有可能发生。可以根据此时的队列输入速率和输出速率估计该流的队列将要达到饱和的时间 $t_{f,full,i}$。如果它小于某个预定的阈值，则拥塞发生概率将很大，需向该流来自的输入端分组调度器发送反馈信息，报告拥塞预测状况。输入端分组调度器根据此信息，适当调整可能拥塞流的实际分配速率 r_f，减缓往那个输出端口调度输出的速率，避免拥塞的发生。

反馈机制的实现如下：

设定时检测的间隔为 T，拥塞时间阈值为 t_{max} 和 t_{min}。

1) *输出端 i 的流 f 的输入速率 $r_{fin,i}$ 估计*　输入速率实际上是 n 个 T 时间的平均输入速率。每个流输出队列记录此前 n 个 T 时间中输入分组的总长度，此值随每个分组的入队及每个 T 间隔的到来而不断更新。

2) *输出端 i 的流 f 的输出速率 $r_{fout,i}$ 估计*　同上类似，输出队列记录此前 n 个 T 时间的输出该流的分组总长度，此值随每个分组的出队以及每个 T 间隔的到来而不断更新。

3) *拥塞预测和反馈信息发送*　$r_{fin,i} > r_{fout,i}$，即有可能发生拥塞时，估计队列将要饱和的时间为

$$t_{f,full,i} = \frac{\text{队列空闲元素数目}}{r_{fin,i} - r_{fout,i}}。$$

$t_{f,full,i} > t_{max}$ 时，拥塞发生概率很小，不发反馈信息；$t_{min} < t_{f,full,i} \leqslant t_{max}$ 时，为一般拥塞，发送一般反馈信息；$t_{full,i} \leqslant t_{min}$ 时，拥塞即将发生，发送紧急反馈信息；需要发送反馈信息时，输出端向流 f 的输入队列所在的输入端口发送反馈信息。

4) *输入端对反馈信息处理*　对一般反馈信息，减小向对应输出端口输出的流 f 的实际分配速率 r_f，$r_f \leftarrow \alpha r_f$，减小因子 $\alpha = r_{fout,i} / r_{fin,i}$，当收到同一个输出端口发来的一般反馈信息超过 m 次时，不再理会。对紧急反馈信息，要大幅度地减小 r_f，$r_f \leftarrow \beta r_f$，减小因子 $\beta = r_{fout,i} / r_{fin,i}$。当 k 个定时间隔 T 内没有收到反馈信息时，实际分配速率应逐渐回升，$r_f \leftarrow \gamma r_f$，回升因子 $\gamma > 1$，直至回升到该流的请求分配速率 R_f 为止。

5) *若干个常数选定*　反馈机制中要用到不少的常数：定时间隔 T，预测拥塞发生的时间阈值 t_{max} 和 t_{min}。紧急反馈信息处理中，对实际分配速率的减小因子 β 中的 >1 的常数 ω。用于输入端响应的有效的一般反馈信息次数的 m，用于输入端分配速率回升判断的 k，以及回升因子 γ。这些常数对反馈机制的性能有很大的影响。对这些常数的取值，先取经验值再用实验辅佐的方法来确定。

5　BW7000 路由器安全体系结构

BW7000 路由器的安全体系结构如图 7 所示。路由器集成了 4 个安全模块，分组过滤模块，协议认证模块；协议安全模块和信息加密模块。

*信息加密模块*是安全体系结构的基础，它除了提供所有的加密算法和对应的加密函数之外，还要提供密钥的生成、管理、传输和加密存储；同时为

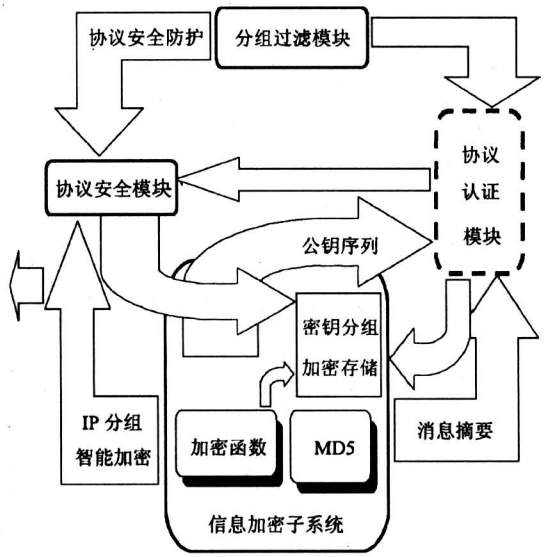

图 7 路由器安全体系结构

Fig.7 Security architecture of router

了满足协议认证的需要，加密模块还实现了消息摘要算法 MD5。为了保证加密的性能，采用 DSP 芯片设计了专用的信息加密卡，实现安全密钥存储，并基于硬件实现了清华大学自主版权的 TUC 加密算法、DES、IDEA 等分组加密算法，以及 RSA 公钥算法和 MD5 消息摘要算法。

分组过滤模块在网络层和传输层提供分组过滤功能。在实现过滤时，路由器逐一查找屏蔽规则直到找到一个匹配并产生一个特定的动作。如果找不到可用规则，就使用默认规则来处理该分组。默认规则一般是将分组丢弃。分组过滤模块同样使用了前面介绍的基于无冲突的 Hash-Trie 树的分组分类算法。

协议认证模块严格说来不是一个单独的模块，它们集成在各自的协议实现中。协议认证是对等协议之间的身份认证，即路由器的身份认证。在 BW7000 路由器中实现了对 OSPFv2 的协议认证，通过检验 OSPF 信息中的数据签名来保证路由器的身份。在协议认证模块中，主要使用加密模块中的公开密钥算法和消息摘要算法 MD5。

协议安全模块主要是针对 IP 协议而言的。由于 IP 处于路由器的核心地位，其安全性至关重要。采取了两种方法来保证这一点：首先是协议自身的安全防护，包括 IP 协议的抗源路由攻击、抗源地址欺骗、抗极小数据段攻击等；其次是协议信息的加密，即根据用户要求实现 IP 数据报的加密。在协议安全模块中，主要使用加密模块中的序列密钥算法和密钥生成算法。为了能够用 BW7000 路由器构造安全的信息网络，在路由器中实现了 IPSec 协议族的主要功能[7]。

6 分布式密钥管理

在大规模的分布式路由系统中，每个实体（主要指路由器）都可能与其他所有实体通信，密钥的分配管理更是个极为复杂的问题。为此，提出了基于 RSA 的分布式密钥生成算法，使用该算法，可以保证所有生成的密钥在整个系统里是唯一的，从而在不降低加密强度的基础上增强系统的扩展性。

6.1 分布式密钥生成

在一个典型的分布式路由系统中，相同管理层次上的所有路由器地位都是平等的，并不存在一个特殊的实体。很自然的，考虑在系统中使用一种分布式的密钥生成方案，对应于某种密钥管理层次（可直接映射为路由管理层次），由分布的地位平等的密钥生成器来生成其管理区域的路由器的密钥，如图 8 所示。

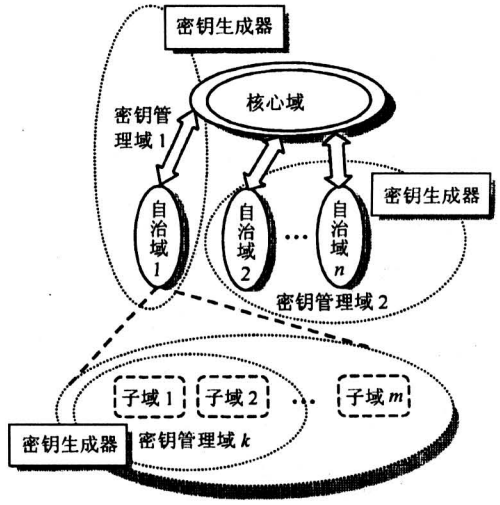

图 8 分布式密钥生成

Fig.8 Distributed key generation

算法思想是，首先，给每一个密钥生成器分配一个全局唯一的密钥种子范围，这个范围包含了其负责区域的全局标识 $[Z', i']$（网络区域 $Z=(z_1, z_2, \cdots, z_n)$ 以及局部标识 i）；其次，保证其产生密钥的某固定部分将落在这个密钥种子范围内。进一步来说，假设第 g 个密钥生成器的密钥

种子范围为 $[L_g, U_g]$，将产生一个密钥，其低 m 位就落在此区间内，表示为

$$L_g \leqslant (n \bmod 2^m) \leqslant U_g,$$

其中 n 是公钥，m 是公钥的固定部分位数（即低 m 位）。在这里，采用了分布式生成方法，而且不需要一个能被所有人信任的特殊的路由器；一个被某路由器信任的密钥生成器并不一定为其他路由器所信任。这样就不再需要一个单独的受信实体，在消除了单一失效点的同时，也减少了产生系统瓶颈（尤其当使用密钥管理中心/密钥服务器生成和分发密钥时）的可能性。

与此同时并没有降低任何安全性，因为在 RSA 加密体制中，加密密钥 e 是依赖于公钥的模数部分 n 的，而 n 是唯一的。这样一来，只有公钥的模数部分 n 需要被生成。下面给出一种基于 RSA 的密钥生成算法：每当需要生成密钥时，可从密钥生成器 g 的密钥种子范围中随机选取一个种子数 $r \in [L_g, U_g]$，再将它作为公钥的模数部分（即 n）的低 m 位，即要求

$$n \equiv r(\bmod 2^m)。$$

为达到此目的，可用如下的算法生成，设公钥的模数部分 n 约为 2 L 位：

1）随机生成 L 位的大素数 p；

2）计算 $q_0 = (rp^{\varphi(2^m)-1}) (\bmod 2^m) + 2^{L-1}$；

3）随机选取整数 i_0，计算 $q = q_0 + 2^m i$，$i = i_0, i_0 + 1, \cdots$，直至 q 为素数；若失败，则返回第 1 步；

4）用上节中计算 RSA 密钥的一般公式计算：$n = pq$；选取 e，d 满足 $(e, \varphi(n)) = 1$；$de \equiv 1 \bmod \varphi(n)$；由算法可知，

$$q = ((rp^{\varphi(2^m)-1}) \bmod 2^m) + 2^{L-1} + 2^m i,$$

所以，素数 q 的位数不低于 L 位，因而 n 的位数约为 2 L 位。

此外，$q \equiv (rp^{\varphi(2^m)-1}) \pmod{2^m}$，$n \equiv pq \pmod{2^m}$，所以 $n \equiv p (rp^{\varphi(2^m)-1}) \pmod{2^m}$。而 $p^{\varphi(2^m)} \equiv 1 \pmod{2^m}$，故 $n \equiv r \pmod{2^m}$，符合系统设计要求。

6.2 密钥的分发

在分布式路由系统中，每个参与路由的实体（R_i）都由信任实体（TE）配置一个证书（C_i），以证实此路由实体的基本信息，如路由的拥有者等。在密钥的分发中，R_i 将证书（C_i）和本实体的公钥（P_i）以及本路由器的基本信息（I_i）用本路由器的私钥（V_i）加密签名，生成密钥证书串 $(P_i, C_i, I_i, S(V_i, P_i, C_i, I_i))$（$S$ 为签名函数），然后向外广播以进行密钥分发。

7 系统实现及测试

基于 Motorola MCP750 开发系统[8]，实现了基于 HEROS 操作系统的高性能安全路由器。其接口类型包括 Gigabit Ethernet，Fast Ethernet，OC-3 POS 和 E1 等。背板是 Compact PCI 总线，板间实现了基于 DMA 机制的 DMQ。系统最多支持 7 个接口板，每个接口板最多支持 5 个 Fast Ethernet 或 2 个 Gigabit Ethernet 和 1 个 Fast Ethernet 接口。

BW7000 路由器是一个复杂系统，需要从多角度进行测试：利用清华大学计算机系研制的协议集成测试系统 PITS[9] 对路由器进行了协议一致性测试和互操作性测试；利用 SmartBits 性能测试仪对路由器做了性能测试，其中板间 IP 分组转发速率的测试结果见表 2。

表 2　性能测试结果

Table 2　Result of performance testing

包长/B	单组板间转发/Mb·s^{-1}	两组板间转发/Mb·s^{-1}
64	450	815
512	750	1 320
1 024	820	1 443

8 结语

介绍了“八六三”重大课题——高性能安全路由器 BW7000 研制过程中解决的一系列关键技术问题。为了彻底掌握路由器的核心技术并保证系统的安全性，设计实现了高性能路由器操作系统 HEROS。为了保证高性能的路由转发，设计实现了基于 RAM 的高性能路由查找算法。为了支持服务质量控制和安全管理，设计实现了无冲突的 Hash-Trie 树分组分类算法和基于反馈的分布式分组调度算法。为了保证网络安全，设计实现了基于分布式密钥管理的路由器安全体系结构。

性能测试表明，BW7000 路由器已经达到了 cisco7000 系列水平。我国自主设计的高性能安全路由器，对保障我国互联网络的安全具有重大意义。使用 BW7000 路由器既可以构建安全的信息网络，又可以在现有网络基础上构建虚拟私有网络，其应用前景相当广泛。目前，此项研究成果正由清

华紫光比威网络技术有限公司进行产品化。

BW7000路由器在不进行分组加密的情况下达到了较高的性能，但是在执行分组加解密时性能将有所下降。虽然采用了硬件实现加解密算法，但是算法的复杂性会影响性能的大幅度提高。在保证不降低安全性的情况下，提高加解密速度是下一阶段进行研究的重点。

参考文献

[1] 范晓勃，林 闯，吴建平，等. 分布式路由器的性能模型与分析 [J]. 计算机学报，1999，(11)：1223～1227

[2] Gupta P, Lin S, Mckeown N. Routing lookups in hardware at memory access speeds [A]. Proceedings of IEEE INFOCOM 98 [C], San Francisco, 1998

[3] Huang Nenfu, Zhao Shiming, Pan Jenyi, et al. A fast Ip routing lookup scheme for gigabit switching routers [A]. Proceedings of IEEE INFOCOM 99 [C], San Francisco, 1999

[4] 吴 剑. 高性能路由器路由管理及RIP协议的研究与实现 [D]. 北京：清华大学，2001

[5] Srinivasn V, Varghese G, Suri S, et al. Fast scalable level four switching [J]. ACM Computer Communication Review, 1998, 28 (4): 191～205

[6] Bennett J C R, Zhang Hui. WF^2Q: Worst-case fair weighted fair queuing [A]. Proceedings of ACM-SIGCOMM 96 [C], Palo Alto: CA, 1996. 143～156

[7] Kent S, Atkinson R. Security architecture for the Internet protocol [M]. RFC2401, 1998

[8] Motorola Inc. MCP750 Series single board computer programmer's reference guide [M]. Computer Group, 1999

[9] 吴建平，陈修环，郝瑞斌，等. 基于形式化技术的协议集成测试系统——PITS [J]. 清华大学学报，1998，38 (S1)：26～29

Design and Implementation of High Performance Security Router BW7000

Xu Mingwei, Xu Ke, Xiong Yongqiang, Jiang Yong, Sun Xiaoxia, Wu Jian, Yu Zhongchao

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

[**Abstract**] High performance and security are hot areas of the research of Internet. How to provide security protection but not decrease the forwarding performance is a hot research topic currently. This paper is based on the research of the high performance security router, a key project of national high technology research and development plan. Operating system (HEROS) of the high performance router BW7000 was developed independently. In order to provide high performance IP packets forwarding, a high performance routing lookup algorithm based on RAM was developed. A novel classification algorithm based on non-collision Hash-Trie-tree and an algorithm based on distributed packet fair queuing with feedback mechanism weve designed and impemented to support QoS control and security management. In order to secure the network, a router security architecture based on distributed key management was proposed.

[**Key words**] router; Security; router operating system; route lookup; packet classification; packet scheduling