三层交换机基于 vlan 的组播实现①

王建文 王政锋 (河北工程技术高等专科学校计算机系 河北 沧州 061001)

摘 要: 文章介绍了 BCM5695 交换芯片对组播和 VLAN(虚拟局域网)的支持，分析了在局域网中使用三层交换机相对于路由器的优点，描述和解决了 VLAN 之间通过硬件地址表进行组播路由的难点，并介绍了其软件的具体实现。

关键词: 交换机;组播;路由;VLAN

Implemention of Multicast Based on Vlan in Lay3 Switch

WANG Jian-Wen, WANG Zheng-Feng

(Dept. of Computer Science, Hebei Engineering and Technical College, Cangzhou 061001, China)

Abstract: This paper describes the support for multicast and vlan(virtual LAN) on bcm5695 switch chip , analyzes the advantage of lay3 switch comparing router on LAN, discusses and resolves the difficulty of routing on vlan through hardware address table, and presents the implemention of software.

Keywords: switch; multicast; route; vlan

1 引言

三层交换机是宽带高速网络中的重要设备，它的主要特点是利用 ASIC(专用集成电路)技术实现通过硬件进行路由。在局域网上，为避免广播风暴而将整个网络划分为多个 VLAN，而 VLAN 之间的数据通信需要路由器，但局域网内的通信流量很大，随着通信流量的不断增大路由器将会成为网络的瓶颈。而三层交换机既有路由的功能，又具备普通交换机的转发速度，所以在局域网的设计中大多采用三层交换机进行 VLAN 间的通信[1]。

我们使用了二层交换机将用户划分在不同的 VLAN 中，再将二层交换机与一个三层交换机相连接，实现了不同 VLAN 间的通信[2]。本文将介绍三层交换机上组播路由功能需要解决得技术难点以及它的实现和与之相关的软件设计。

2 三层交换机的软件结构

2.1 芯片的特点与数据包在硬件中的流程

我们设计的交换机 STT100 使用的是 BROADCOM 公司的 BCM5695 芯片。它具有 12 个千兆自协商端口，其中 8 个光口，2 个电口以及 2 个光电可切换端口[3]。BCM5695 芯片支持基于端口的 VLAN 实现，每个端口可以属于不同的 VLAN，几个端口也可以属于同一个 VLAN；并同时支持 tagged VLAN 与 untagged VLAN，在具体实现时 untagged VLAN 通常用在与主机直接相连接，因为主机网卡可能不支持 802.1Q 协议[4]，当数据包从主机进入端口时根据为数据包加上端口 VLAN 号，并在数据包离开端口进入主机时去除 VLAN 号，这样一个具有 untagged VLAN 的端口只能属于一个 VLAN；tagged VLAN 用于支持 VLAN 的交换机之间的连接，只有数据包所带的 VLAN 号包含于它要进入的端口所属的 VLAN 号时才被转发[5]。为了使 VLAN 的实现与路由转发相一致，我们将局域网中的每一个子网对应一个 VLAN，并且每个 VLAN 只和一个路由接口相对应。即端口号与 VLAN 号是多对多的关系，VLAN 号与路由接口是一对一的关系。

数据包通过交换芯片的流程如图 1 所示：当数据包通过端口进入交换机后，提交给输入逻辑，每个端口都有一个属于自己的输入逻辑，但是其中有些表是

① 收稿时间:2009-10-19;收到修改稿时间:2009-11-20

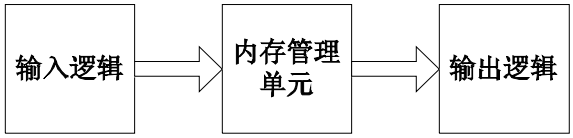

图 1　交换芯片中数据包的流程图

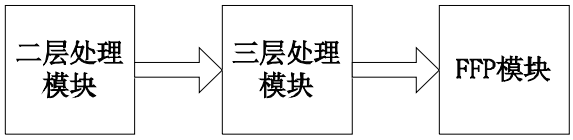

图 2　输入逻辑中的流程图

所有端口共享的。输入逻辑给出所有数据包的转发决策，并将所有转发信息伴随数据包一起交给内存管理单元进行缓存和调度。内存管理单元主要负责缓存和调度数据包。它接收来自输入逻辑的数据包，并缓存它们；调度待转发的数据包，并交给对应转发端口的输出逻辑。输出逻辑主要负责缓存和调度数据包。它接收来自输入逻辑的数据包，并缓存它们；调度待转发的数据包，并交给对应转发端口的输出逻辑。如图 2 所示输入逻辑又分为几个过程：数据包首先进入二层转发模块，通过检查数据包的 VLAN 号和 MAC 地址确定将向那个端口转发，如果发现数据包要经过三层转发，则进入三层转发模块。数据包进入三层模块，判断 IP 地址是组播还是单播，如果是组播则对芯片中的地址表项进行查找，得到下一跳的路由接口。如果没有发现与数据包匹配的表项，这将数据包交给 CPU 处理，通过路由软件寻找下一跳的路由接口；对于单播，数据包的流程也类似。然后数据包进入 FFP(快速过滤处理器)模块，FFP 对与设定的字段相匹配的数据包进行处理，例如通过改变 COS 域，将数据包丢弃，将数据包转发给 CPU 等[5]。

在本文中我们主要关注三层转发模块中对组播包的处理过程，它的详细过程后面还会提到。通过对芯片的逻辑过程和硬件地址表的特点进行软件设计与实现。

2.2 交换机中数据包在软件中的流程

交换机 STT100 使用的操作系统是 Linux，Linux 的内核中实现了 IP 协议以及 TCP，UDP 协议。交换机需要的其他协议则由用户程序实现。如图 3 所示，在交换芯片 BCM5695 上运行驱动程序，其功能主要包括读写芯片中的寄存器中的控制表；对数据包进行处理，完成网络协议栈中二层协议的功能；以及中断处理。二层应用程序主要功能是通过驱动程序对控制表动态读写。三层应用程序主要是路由协议和组播协议。

如前面所述，当数据包进入到输入逻辑中的三层处理模块时，如果没有找到与之相匹配的表项时将数据包转发给 CPU。这时数据包进入到了驱动程序，驱动程序对数据包进行处理后将它转发给内核，通过内核维护的路由表得到数据包下一跳要通过的路由接口和相关信息，然后数据包又进入驱动程序，通过驱动程序将数据包重新交给交换芯片，然后数据包和直接通过输入逻辑中的三层处理模块的数据包一样进入了输入逻辑的 FFT 模块。

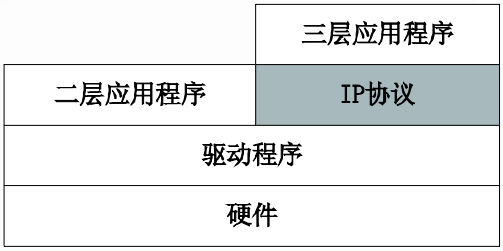

图 3　交换机软件结构简图

3　交换机中所用的组播协议

组播协议主要分为组播管理协议和组播路由协议。组播管理协议运行在与局域网相连的路由器上，通信双方是路由器和主机，用来维护主机加入和退出组播组的信息。组播路由协议的通信双方都是路由器，用来维护组播的路由转发信息。在 STT100 交换机上运行的组播管理协议和组播路由协议分别是 IGMPv2 和 PIM-SM，文献[6]中有对组播协议的详细介绍，这里仅根据后面描述的需要对这两个协议做简要的介绍。

3.1 IGMPv2 简介

IGMPv2 中路由器定时向局域网中的主机发送查询消息，如果有主机要加入某一组播组 G，则在接收消息后向路由器发送这个组播组的地址信息，路由器接收后维护这一地址。当主机要退出 G 时，向路由器发送退出这一组播组的消息，路由器收到消息后再向所有路由器发送查询组播组 G 的消息，如果没有主机响应，则表示局域网内已没有加入组播组 G 的主机了，所以路由器将地址 G 删除，否则继续保留

这个地址。

3.2 PIM-SM 简介

PIM-SM 是运行在路由器之间的协议，它通过单播路由表来建立组播路由的转发信息，与单播路由协议本身不相关。协议中有两个树的概念 SPT(源分布树)和 RPT(共享树)。RPT 对于建立组播路由有重要意义。先来看一下 RPT 的建立过程，先在所有的支持 PIM-SM 的路由器中选举出 RP(汇合点)，并使所有路由器知道 RP 的 IP 地址。如果主机 1 加入组播组 G，则它通过 IGMPv2 向路由器 C 报告，如果 C 原来没有保持这个组播地址的信息则建立了一个新的表项(*，G)，并向 RP 发送一个 PIM(*，G)加入消息。RP 收到(*，G)加入消息后，将它到 C 的链路添加到输出接口列表，这样当 RP 收到组播地址为 G 的组播包时将它转发到 C。同样，当 E 建立一个新的表项(*，G)时，向 RP 发送一个 PIM(*，G)加入消息，当消息到达 C 时由于 C 已有了表项(*，G)，所以不再向 RP 转发，而是将 C 到 E 的链路添加到输出接口列表，这样 C 接收到组播地址为 G 的组播包时就可以将它转发到 E。如果 E 向 RP 发送(*，G)加入消息时 C 还没有(*，G)表项，那么在 C 上建立表项(*，G)，并将 C 到 E 的链路添加到输出接口列表，然后向 RP 发送(*，G)加入消息，这样 RP 在受到地址为 G 的组播地址后可以将它转发到 E。上面的两个例子都最终建立了以 RP 为根的共享树。这样只要使组播源将组播包发送到 RP 就可以将组播包传送到所有加入组播组的主机。如果主机 S 要发送组播为 G 的数据，首先它的下一跳路由器 A 向 RP 发送源注册消息，当 RP 收到消息后，向组播源发送(S，G)加入消息，这个过程与 E 向 RP 发送(*，G)加入消息的过程类似，这样在 B 中也建立了(S，G)表项，这样 A 发送的组播包通过 B 到达 RP，再由 RP 将组播包传到与接收点相连的路由器上。这样建立的组播树有两点明显的不足：首先，组播接收点到源点的路径不是最短的，因为必须通过 RP；而且由于每个组播组的组播树都要包含 RP 增加了 RP 的负载。这时，各组播接收点已经知道了组播源点的 IP 地址，所以可以像建立共享树一样建立起 SPT。

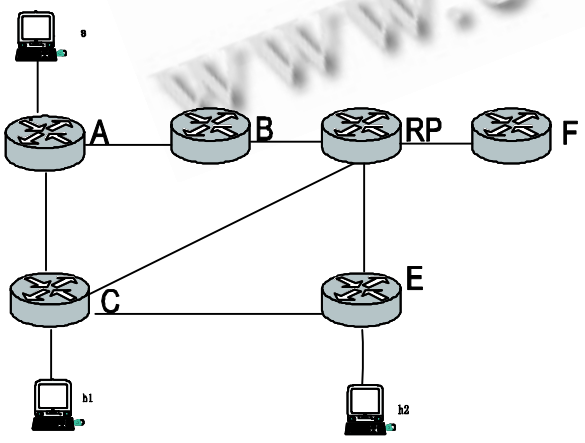

图 4 组播网络拓扑图

4 交换机中组播路由的硬件实现

在交换机 STT100 中通过一次路由，多次转发的思想实现了直接通过交换芯片 BCM5695 对组播包进行转发。对于这项技术的实现需要考虑几个问题：首先，硬件表项是怎样得到路由信息的。第二，路由器关心的只是组播地址和路由接口的对应关系，而交换机要知道组播包向哪些 VLAN 转发和向哪些端口转发。第三，在组播转发的过程中，总是有主机加入组播组和离开组播组，这些信息在软件的路由程序中都能得到，但怎样使它们到达硬件中的地址表，即怎样使软硬件的组播转发表保持一致。第四，由于一个交换机上一个端口可以属于不同的 VLAN，如果在同属于一个端口的不同 VLAN 中都有主机要接收同一个组播地址的组播包，怎样通过这一个端口向不同的 VLAN 进行组播转发。第五，由于硬件地址表的容量是一定的(BCM5695 中是 8192 个，并且是和单播地址表共用)，当地址表满时对要加入的地址项怎样处理。下面通过程序的具体实现对这些问题进行解答。

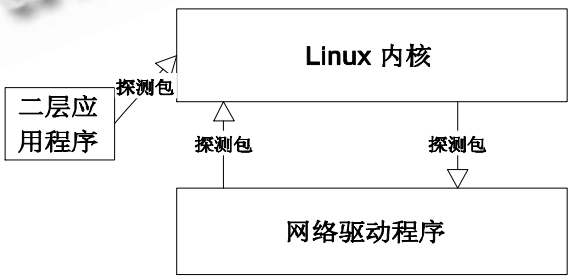

图 5 三层交换机系统结构图

软件层的路由表是由 Linux 内核维护的，如果通过修改内核直接将路由表的信息写入硬件的地址表中，一方面风险大，另一方面建立了软件与内核的相关性。而组播协议的数据包通过内核后得到了组播路由的信息，所以可以通过数据包的路由信息来修改硬件地址

表的信息，由上节的叙述可知在组播路由信息建立的过程中，路由器之间有许多类型的数据包相互通信，如果想从这些数据包中获得路由信息，一方面程序实现比较复杂，另一方面，需要在网络驱动程序中获取 IP 层的信息，降低了驱动程序的通用性。所以我们只关心组播源发送的组播包，当组播包通过软件路由后进入驱动程序时已经获得了组播路由的转发信息，这时可以通过驱动程序将路由信息写入硬件地址表中。具体实现是这样的，当组播包在硬件的地址表中没有找到相应的地址表项时进入软件，首先进入驱动程序，驱动程序根据此组播包构造一个探测包，它与组播包有相同的源地址与组播地址，将探测包发送给内核，内核通过路由表为组播输出列表中的每一个路由接口产生一个组播包，这些组播包进入驱动程序后，通过每个路由接口的信息得到 VLAN 号，并根据 VLAN 号得到属于这个 VLAN 的端口号。芯片 BCM5695 的组播地址表中的记录包括组播源所在的 VLAN 号，源 IP 地址，目的 IP 地址，出口路由的 MAC 地址，需要转发组播包的端口号(包括需要二层转发的端口和需要三层转发的端口)，这些信息都可以从探测包产生的组播包中得到，在驱动程序中将这些信息写入硬件地址表，以后具有相同的 IP 源地址和 IP 组播地址的数据包到达时可以通过查找硬件地址表直接向表中记录的各个端口转发数据包。由于每个路由接口对应一个 VLAN，而且向 VLAN 的所有端口都转发数据包，所以当一个端口只属于一个 VLAN 时，如果一个 VLAN 中有组播组的接收点，则组播包就可以到达。之所以不是通过组播数据包获得组播路由信息而是要通过构造探测包是因为增加了对数据包的处理也就增大了数据包的网络延迟。

前面的叙述回答了前两个问题，再来看一下问题三，解决的方法同样是通过探测包，不过这次不再是通过驱动程序产生，而是通过一个二层的用户进程，每隔一段时间产生一个探测包，直接发送到内核，从内核进入驱动程序后具有了当前的组播路由信息，这样可以每隔一定时间更新硬件的地址表使之与软件的路由表相一致。

对于第四个问题，由上面的叙述可知，交换机通过硬件进行组播转发时只关心向哪些端口转发数据包，但是当一个端口属于不同的 VLAN 时，如果这两个 VLAN 都有主机要接收组播包(如图 6 所示)，就需要从这个端口发送两个组播包。幸好 BCM5695 有一个表专门记录每个组播地址中每个端口对应 VLAN 的情况，实现了当多个 VLAN 需要通过同一个端口获得数据包时交换机自动复制。

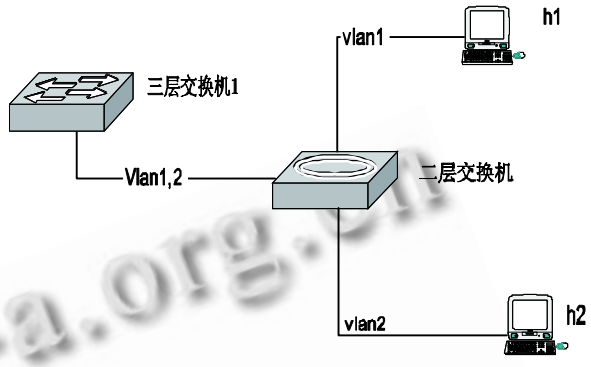

图 6　同一三层交换机上不同 vlan 的网络拓扑图

对于第五个问题，BCM5695 芯片本身具有地址项的老化机制，即如果一段时间内如果地址项没有被查找命中，地址项将自动被设为无效。但是这样一来会出现这样的问题，如果将老化的时间设置过短，当组播组的组播源和接收点仍 然存在，只是有一段时间没有发送组播包时，相应的地址表项便会老化，这样当新的组播包到来时又要重新经过软件路由，增加了转发时间；如果将老化时间设置过长，一方面会使地址表被写满的概率增大，另一方面当已不存在组播接收点时，如果有组播包发送过来，交换机还是会转发，从而浪费了带宽。所以为解决第五个问题，我们没有对硬件地址表进行老化设置，而是在二层应用程序中维护一个地址链表，包括二层地址表中的地址项，每当有新的组播路由地址表项被插入硬件，则将地址项加入地址链表的尾部中，如果地址链表的长度达到了地址表项的上限，则通过驱动程序将相应的硬件地址表项删除；与此同时对根据链表中的地址构造探测包，向内核发送，如果发现一个地址项所对应的地址表已经不存在则直接将它对应的硬件地址表项删除。这样既使一些过期的地址及时删除，有不会使有效的地址在硬件地址表中失效。这里构造探测包的过程实际上与解决问题三时构造探测包是一个过程。

这样实现了交换机直接通过硬件地址表转发组播包，从而提高了组播路由转发的速度，这对于经常进行视频点播的局域网来说是十分重要的。

(下转第 202 页)

(上接第 198 页)

5 下一步优化的方案

在上一节中我们可以看到，组播路由的转发信息是针对路由接口的，由于路由接口与 VLAN 一一对应，所以也可以说是针对 VLAN 的，但一个 VLAN 可以包含多个端口，即只要有一个端口需要转发组播包就要向所有的端口转发数据包。在 BCM5695 芯片中有一个地址表包含二层组播地址及端口映射，由于每一个 IP 组播地址与 MAC 组播地址一一对应，所以将要转发的端口号映射到与 IP 组播地址对应的二层组播地址表项中就可以解决这个问题。可以捕捉下层路由器的 PIM 加入消息，在驱动程序中提取消息中的组播地址，并得到数据包进入交换机的端口号，将这些信息写入二层组播地址表中。这样就可以在组播转发时只向需要得到组播包的端口转发。

参考文献

1 cww.三层交换机的原理和设计.[2006-11-10] http://www.cww.net.cn/Technique/2006/11/51975.htm

2 Computer Networks. 第四版.潘爱民.北京.清华大学出版社. 2004.281－283.

3 Broadcom. Programmer's reference guider BCM 5465.19CCE.

4 IEEE Std 802. 1Q—2003. IEEE Standards for Local and MetropolitanArea Networks: Virtual Bridged Local Area Networks.

5 Broadcom. BCM5690 the theory of operation. 171G7. Alton Parkway. 2004.

6 Beau Williamson. 顾金星等.北京:电子工业出版社. 2000.142－167.