基于 SNMP 的网络层拓扑发现

李 琳　李 杰　（中南大学信息科学与工程学院多媒体和网络技术研究所 长沙 410083）

摘要：本文通过对 MIB－II 定义的路由表、地址转换表等信息的分析，描述了一种基于 SNMP 的网络拓扑发现方法，并解决了路由器的多 IP 地址问题，同时给出了详细的数据结构和算法描述。另外，由于不向网络中注入过多的探测数据包，该方法对网络正常流量不会产生较大影响，并且提高了搜索的效率和准确性。

关键词：拓扑发现 网络管理 SNMP MIB

1 引言

网络管理中的拓扑发现是利用网管协议或工具搜集分布在网络各处的原始拓扑数据，通过拓扑生成算法综合出完整的拓扑信息。其主要目的就是获取和维护网络元素的存在性信息以及它们之间的连接关系信息，并在此基础上给出整个网络连接状态的图示，最终实现对它们的有效管理。通过网络拓扑发现，可以帮助人们研究、开发和更好地利用网络。

网络拓扑发现主要涉及三方面的问题[2,3]：针对网络层次的哪一层；是采用被动监测技术还是主动探测技术来采集网络拓扑信息；采用什么方式收集信息。由于网络连接的复杂性和网络协议的多样性，这就决定了发现网络拓扑结构的方法也不是唯一的。

网络拓扑发现可采用多种工具或协议，归纳起来有 ICMP（Ping、Traceroute）、SNMP 和 DNS 等，它们各具特点。但其中以 Ping 和 Traceroute 应用范围最广，以 SNMP 最高效。

（1）ICMP 方法。Ping 和 Traceroute 都是基于 ICMP 协议的 IP 网络的常用工具。这两种工具都采用主动探测手段，适用范围广，但是它需要定期的向网络中的各个节点发出探测数据包，一定程度上加重了网络的负荷，且发现效率不高。通常用于 SNMP 无法顺利应用的场合。

（2）DNS（域名系统）协议。DNS 是提供主机名和 IP 地址间映射的分布式数据库。大多数域名服务器通过“zone transfer”命令返回该域内名字的列表。通常它与广播 Ping 或 Traceroute 工具结合使用。这种技术快速、准确、开销小。但是，发现并不完全准确，因为用 DHCP 获取 IP 地址的主机并没有 DNS 服务，而且一些网管员因安全原因关闭了 DNS 域转换服务。

（3）ARP 协议。ARP 属于网络层，它将上层的 IP 地址与底层的物理地址进行绑定，ARP 协议维护管理了一张地址映射表。ARP 表中的网络设备地址都是最近活动过的有效 IP，而且几乎没有冗余信息。所以采用此方法的发现效率很高，但如果网络过大，ARP 表中的记录可能无法包括网络中实际存在的所有网络设备，并且要求网络设备都支持 ARP 协议。

（4）SNMP。SNMP 是 TCP/ IP 网络中应用最广泛的网络管理协议。它的基本思想是所有的网络设备维护一个 MIB（管理信息库）保存其所有运行进程的相关信息，并对管理工作站的查询进行响应。

SNMP 有两个主要的组成部分：SNMP 代理和 SNMP 管理者。我们想要监视的每个网络设备（节点）都要运行 SNMP 代理。管理终端与代理进程之间进行通信，通过 SNMP 定义的 GetRequest，GetNextRequest，SetRequest，GetResponse，Trap 五种操作对设备进行信息查询和参数设置。在拓扑发现中主要是获取设备的信息，所以主要用到前面两个操作。这种信息是用来对各个网络节点之间相互连接关系的判断依据。

目前主要的网络设备都提供对 SNMP 协议的支持，因此基于 SNMP 协议的网络层拓扑发现技术被广泛采用。本文通过改进数据结构，主要讨论一种基于 SNMP 的主干网络的拓扑发现算法，有效地解决了“多个 IP”路由器的问题。对于子网的拓扑发现则采用基于 ICMP 的方法来实现。

2 MIB－II 规范

管理信息库（MIB）指明了网络元素所维持的变

量，给出了网络中所有可能的被管理对象的集合的数据结构。通过对这些数据项目的存取访问，就可以得到该网关的所有统计内容。MIB－Ⅱ是 MIB 的第二版，是它的扩充。第三层网络拓扑发现主要是判断路由器及子网的连接关系，其信息来源就是 SNMP 定义的 MIB－Ⅱ库。MIB －Ⅱ信息中与路由器级拓扑发现相关的分为 3 组：系统（system）组、接口（interfaces）组和 IP 组[4,5]。分述如下：

（1）系统组。包括 7 个简单变量，其中 sysService 可用于判断设备类型，从其二进制形式最低位到第 7 位，如果某位为 1 则提供 OSI 对应层次服务。第 3 位为 1 则说明该节点提供路由功能，是路由器设备。

（2）接口组。包含关于该实体的物理接口方面的一般信息，包括配置信息和发生在每个接口的事件的统计信息。其中定义了一个表示本设备接口数量的简单变量 ifNumber 和一个接口表（ifTable），每个接口一行。表的索引项是 ifIndex，其取值范围是从 1 到 ifNumber，并为每个接口分配一个唯一的序列号。ifDescr 为接口名称，ifType 表示接口类型，亦即本接口所在子网的类型，常见如以太网（6）、802.5 令牌环（9）、FDDI（15）。可见，访问路由器接口表的 ifType 值可确定与之相连子网的简单拓扑。

（3）IP 组。IP 组定义了许多简单变量，其中 ipForwarding 为 1 时表示该节点具有 IP 转发功能，可作为路由器的判定依据。本组还定义了 3 个十分有用的表格变量：IP 地址表（ipAddrTable）、IP 路由表（ipRouteTable）和 ARP 地址转换表（ipNetToMediaTable），它们是网络层拓扑发现的重要信息来源。

① IP 地址表

节点的每个 IP 地址对应表格中一行，每行有 5 个变量。其中将 ipAdEntAddr 和 ipAdEntNetMask 按位与，便得到该接口所属子网的 IP。可见，访问路由器的 IP 地址表，即可得到连接（N，R），这是路由器级拓扑发现的关键。语法描述如下：

```
SEQUENCE OF IpAddrEntry
IpAddrEntry: : =SEQUENCE{
    ipAdEntAddrIpAddress,//指定设备的 IP 地址信息
    ipAdEntIfIndexINTEGER,//各表目的接口号
    ipAdEntNetMaskIpAddress,//在 ipAddrEntry 对象表中的 IP 子网掩码
    ……}
```

② IP 路由表

通常在路由器节点中该表才有意义，它包含了 13 个对象，发现逻辑拓扑时仅涉及其中的 4 项：ipRouteDest，ipRouteIfIndex，ipRouteNextHop，ipRouteType。其中，当路由类型 ipRouteType 值为 4（indirect）时，表示两者是路由器与路由器直接相连，需经 ipRouteNextHop 路由器地址继续查找；当 ipRouteType 值为 3（direct）时，表示与该路由器直接相连的是目的地址所在子网。

MIB－Ⅱ对路由表结构中各项描述如下：

```
ipRouteTable OBJECT — TYPE
SYNTAX SEQUENCE OF IpRouteEntry
IpRouteEntry: : =SEQUE NCE {
    ipRouteDest IpAddress, //路由器的目的地址
    ipRouteIfIndex INTEGER, //路由的当地接口索引
    ipRouteNextHop IpAddress, //路由器的下一个网关地址
    ipRouteType INTEGER, //路由的类型
    ipRouteMask IpAddress, / 路由目的地的子网掩码
    ……}
```

③ ARP 地址转换表

提供了节点所在子网内设备 IP 地址到物理地址的对应转换。信息来自节点系统上 ARP 高速缓存。访问路由器的地址转换表，可迅速得出子网内设备 IP 地址，便于子网拓扑发现。语法描述如下：

```
SEQUENCE OF IpNetToMediaEntry
ipNetToMediaEntry: : =SEQUENCE{
    ipNetToMediaIfIndex INTEGER,//IpNetToMedia 表中的接口号
    ipNetToPhysAddress PhyAddress,//IpNetToMedia 表中的物理硬件地址
    ipNetToMediaNetAddress IpAddress,//相应于 IpNetToMedia 表中的物理地址的 IP 地址
    ……}
```

3 发现算法实现

一个合理的网络拓扑图应该具有层次性，即包括主拓扑和子拓扑。主拓扑显示网络中子网及网关间的

互连结构;子拓扑则显示子网内部网络设备间的互连关系。下面分别论述。

3.1 主要数据结构

本算法使用的数据结构包括如下五个结构体,其中已访问网关队列,为了提高其成员不重复维护的效率,采用哈希链表的数据结构。

```
Struct SubNetQueueItem{
    IpAddress SubNetAddress(4);//子网地址
    IpAddress SubNetMask(4);//子网掩码
    Struct SubNetQueueItem * next;
} * SubNetQueue;//子网队列
Struct RouterQueueItem{
    IpAddress RouterAddr;//路由器唯一标识 IP 地址
    Struct RouterQueueItem * next;
} * RouterQueue;//路由器队列
Struct VisitedRouterQueueItem{
    IpAddress RouterAddr;//路由器唯一标识 IP 地址
    Int No;//路由器的哈希编号
    Int Bfactor;//平衡因子
    Struct VisitedRouterQueueItem * lchild, * rchild;
};//已访问网关结点
Struct UnVisitedRouterQueueItem{
    IpAddress RouterAddr;//路由器唯一标识 IP 地址
    Int No;//路由器的哈希编号
    Struct UnVisitedRouterQueueItem * next;
} * UnVisitedRouterQueue;//未访问网关队列
Struct ConnectQueueItem{
    IpAddress From;//连接的一方地址
    IpAddress To;//连接的另一方地址
    ConnectQueueItem * next;
} * ConnectQueue;//连接队列
```

3.2 网络层拓扑发现算法

算法描述如下:

```
检测出与网管所在子网直连的路由器;
初始化路由器队列,已访问网关哈希链表,待访问网关队列,子网队列及连接队列;
将检测到的路由器的标识 IP 和哈希编号 No 加入 UnVisitedRouterQueue 中;
while(UnVisitedRouterQueue 非空)
{
    从 UnVisitedRouterQueue 中读取一个网关为 CurrentRouter;
    利用 CurrentRouter 的信息初始化一个已访问网关结点 Router;
    if(Router 加入已访问网关哈希链表中成功)
    {
        读取 CurrentRouter 的 ipRouteTable;
        while(ipRouteTable 非空)
        {
            if(ipRouteType=4)
            {
                初始化一个新的待访问网关,加入其标识 IP 和哈希编号 No;
                将该网关不重复的加入到 UnVisitedRouterQueue;
                将 R-R 连接关系加入到 ConnectQueue;
            }
            if(ipRouteType=3)
            {
                由该网关的 ipRouteTable 和 ipAddrTable 匹配得到该网关的子网;
                将 ipRouteDest 和 ipRouteMask 不重复的加入到 SubNetQueue 中;
                把 CurrentRouter 与子网的连接关系不重复的加入到 ConnectQueue 中;
            }
        }
        将 CurrentRouter 加入 RouterQueue 中,并从 UnVisitedRouterQueue 中删除;
    }
    将 CurrentRouter 从 UnVisitedRouterQueue 中删除;
}
将路由器队列、子网队列和连接队列记录到文件中;
```

3.3 关键技术问题及解决方案

3.3.1 与网管所在子网直连的路由器检测

根据本机 IP 地址和子网掩码 MASK 进行“与”操作获得子网地址 Net 和本网可能包括的所有主机数目 N[6]。以子网地址 + 1 为第 1 个主机地址，依次加 1，增加到 N 台机器为止，这样得到了 N 个主机地址 IP。

检查该 IP 是否在子网内，如果 IP&MA SK = NET，则该地址可能是子网中 1 台主机的地址。对这些可能是子网中主机的 IP 地址用 Ping 工具来测试其存活性，若该设备处于存活状态，便向其 161 号端口发送 SNMP 报文，如果有回应，则表示该设备运行了 SNMP 代理。对于已经检测出的 SNMP 主机，检查其 MIB 值。通过访问其 MIB 库中的 ifNumber 和 ipForwarding 变量确定网络设备的类型。当 ipForwarding 为 1 时表示该设备具有向前转发 ip 数据包的功能，这是判断设备为路由器必须要保证的条件，但是只有这一个不够充分，路由器都有多个接口，所以当 ipForwarding 等于 1 且 ifNumber 值大于 1 时为路由器。如果对路由器发现准确度要求非常高，也可以通过进一步判断 sysServices 变量来确定。如果某台 SNMP 主机的 ipForwarding =1 且 ifNumber >1，则证明它就是我们要找的路由器。

3.3.2 避免同一路由器被重复访问

路由器一般拥有多个接口，每个接口都有相应的 IP 地址，通过其中的任意一个 IP 地址通常都可以访问该路由器。这样就存在同一个路由器被多次访问的问题，并且容易被混淆成为有多个路由器。为了解决这个问题，本文提出了一种方法，就是为每个路由器增加一个标识 IP 和一个哈希编号 No 变量。参照 OSPF 协议将路由器接口表中 ipAdEntAddr 最小的 IP 地址作为该路由器的标记，即为它的标识 IP。哈希编号 No 我们取的是标识 IP 的低两个字节的十进制数的拼接，如：标识 IP 为 202.117.98.1，那么哈希编号 No 就为 981。

在判断该路由器是否被访问过时，我们对已访问网关队列采用哈希链表的存储方式。根据实际发现网络拓扑规模的情况取一个素数 n，构造的哈希函数为 H(No) =No% n，处理冲突时，我们采用的是平衡二叉树的链表结构。当把一个网关加入到已访问网关队列中时，首先将该网关的哈希编号 No 模 n 取余，定位到相应的哈希表的位置。然后，在处理冲突的平衡二叉树链表中，利用标识 IP 进行匹配查找，若查找到了相同的标识 IP，则说明该网关已被访问过，返回该网关加入已访问网关队列失败；若在冲突链表中未找到该标识 IP，则说明该网关还未被访问过，将该网关加入冲突链表中，并返回该网关加入已访问网关队列成功。

在实际的网络拓扑发现中，通常会有成千上万个路由器设备，在这种情况下，对已访问网关队列采用哈希链表的存储方式是较为合理的，利用哈希函数可以将不同的标识 IP 快速的散列开，进行初步的定位。同时，通过哈希列表建立了多棵平衡二叉树，这样就减小了单棵树的深度，从而就可以缩短查找单个标识 IP 的时间，进而提高拓扑发现的速度。

3.3.3 获得网络设备间的连接关系

路由表位于 SNMP 管理信息库中的 IP 组，它是网络设备转发数据包的依据，每台网络设备对数据包做出转发到下一个网关、丢弃或传送到本地网络的操作。根据路由表中的信息可以分析出网络设备的连接情况，对于每条记录，接口索引(ipRouteIfIndex)对应路由器的相应接口。若路由类型(ipRouteType)为 direct，说明该接口直连一个子网，对于只使用地址表来确定子网的算法不够准确，有些环境下路由接口在地址表中也有 IP，但悬空时这样的方法就会得到多余的错误直连子网。将路由表中的 ipRouteDest 和 ipRouteMask 按位与得到子网地址，逐一的到地址表中匹配(地址表中的 ipAdEntAddr 和 ipAdEntNetMask 按位与得到接口所在子网的地址)，如果两者相同才表明路由表中这个目标地址是与路由直连的子网，这样发现的子网信息才最准确，同时记录子网及网关与子网的连接信息。若路由类型为 indirect，说明该接口与一路由器直连，且路由表中的下一跳地址(ipRouteNextHop)是其某个接口的 IP 地址，同时记录路由器及路由器与路由器的连接信息。

3.4 子网拓扑搜索算法

本文主要使用 ICMP 协议来实现子拓扑的搜索。首先，根据子网队列中子网的子网地址与子网掩码，可以算出子网内各主机的 IP 地址范围。利用 Ping 功能依次检测各个 IP 地址，确定子网内主机的存活状态。然后，访问主机的 MIB 库变量，如不能访问，可确定为一般主机，根据 sysServices 的值也可判定该网络设备的类型(如 sysServices =2，说明工作在数据链路层，为桥接器)。最后，访问设备的地址表，得到与该设备接口的各个设备的网络地址及接口索引，根据接口索引，查询设备的接口表，就可得到该设备与其他网络设备的接口类型(如 ifType =15，表示接口为 FDDI 类型)。这样就实现了子网拓扑的发现。

(下转第 47 页)

(上接第 42 页)

4 小结

本文论述了基于 SNMP 拓扑发现的改进方法,通过合理的利用哈希链表,减少了重复访问路由器次数,有效解决了"多个 IP"路由器问题。由该方法所构造的网络拓扑能够较真实地反映网络拓扑情况。但是,上述方法不适合属于链路层的拓扑结构的发现,所以,这将成为我们今后努力的方向。

参考文献

1 STALLINGS W. SNMP 网络管理[M],北京:中国电力出版社,2001.

2 张勇、张德运等,网络拓扑发现的主动探测技术的研究和实现[J],小型微型计算机系统,2000.8.

3 Siamwall R ,Sharma R ,Kershav S. Discovering internet topology[R] . Computer Science Department Cornell University ,1998.

4 McCloghrie. Management information base for network management of TCP/ IP - based internets : MIB - II[S] . RFC1213 ,1991.

5 白英彩著,计算机网络管理系统设计与应用,北京:清华大学出版社, 1998.

6 李佳、石冰心,基于 ICMP 和 SNMP 网络拓扑发现算法研究及实现[J] ,微型机与应用, 1998, (1) : 33 - 35.