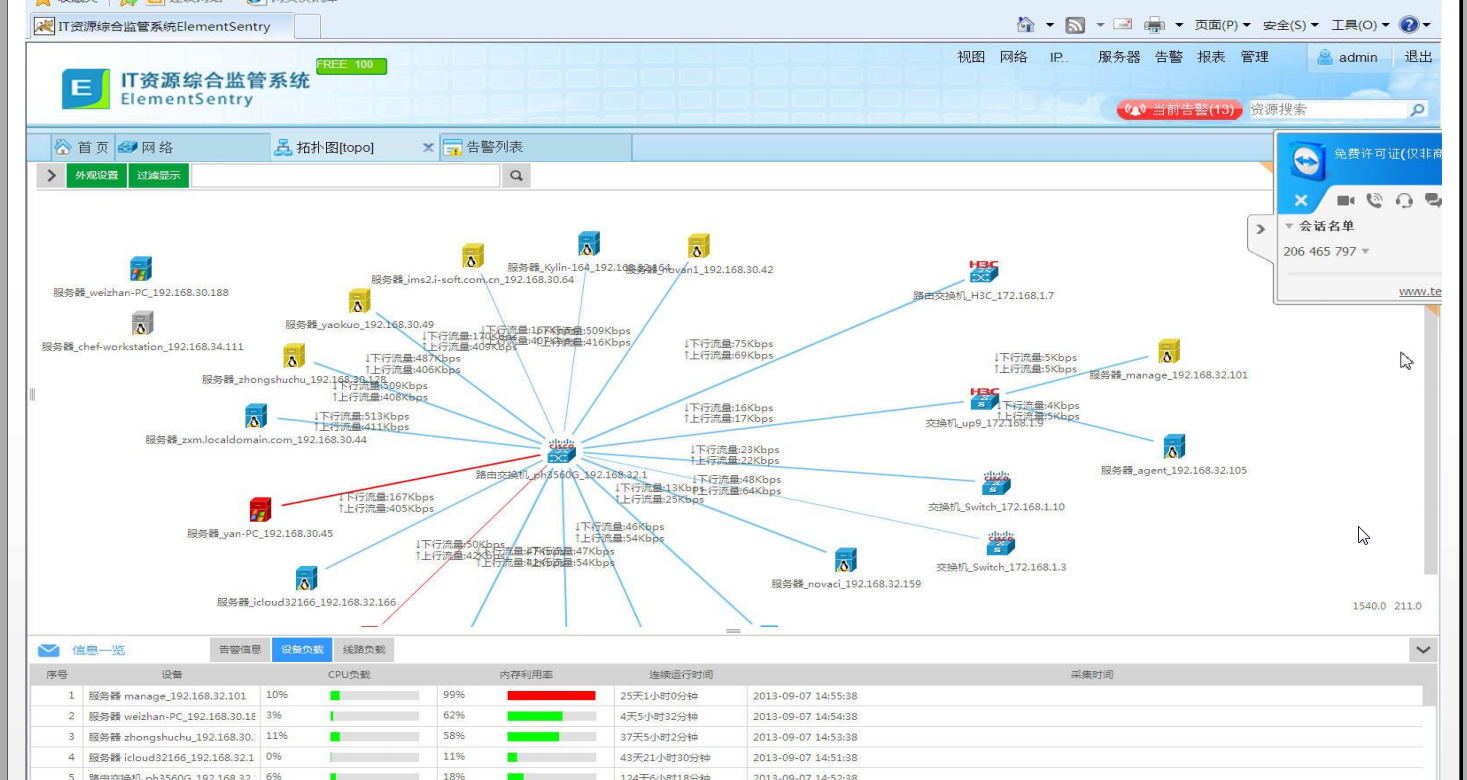
# 速方ElementSentry拓扑发现









# LLDP

[链路层](http://baike.baidu.com/view/4874044.htm)发现[协议](http://baike.baidu.com/view/36190.htm)（[LLDP](http://baike.baidu.com/view/1930049.htm)）是一个厂商无关的二层协议，它允许[网络设备](http://baike.baidu.com/view/1158081.htm)在本地[子网](http://baike.baidu.com/view/65511.htm)中通告自己的设备标识和性能。

## 目录

1[产生背景](http://baike.baidu.com/view/1930049.htm" \l "1)

2[简介](http://baike.baidu.com/view/1930049.htm#2)

3[工作模式](http://baike.baidu.com/view/1930049.htm#3)

4[LLDP报文](http://baike.baidu.com/view/1930049.htm#4)

▪ [概念](http://baike.baidu.com/view/1930049.htm#4_1)

▪ [发送机制](http://baike.baidu.com/view/1930049.htm#4_2)

▪ [接收机制](http://baike.baidu.com/view/1930049.htm#4_3)

## 1产生背景[编辑](http://baike.baidu.com/view/1930049.htm)

目前，[网络设备](http://baike.baidu.com/view/1158081.htm)的种类日益繁多且各自的配置错综复杂，为了使不同厂商的设备能够在网络中相互发现并交互各自的系统及配置信息，需要有一个标准的信息交流平台。

LLDP（Link Layer Discovery Protocol，[链路层](http://baike.baidu.com/view/4874044.htm)发现[协议](http://baike.baidu.com/view/36190.htm)）就是在这样的背景下产生的，它提供了一种标准的[链路层](http://baike.baidu.com/view/4874044.htm)发现方式，可以将本端设备的的主要能力、管理地址、设备标识、接口标识等信息组织成不同的[TLV](http://baike.baidu.com/view/3575978.htm)（Type/Length/Value，类型/长度/值），并封装在LLDPDU（Link Layer Discovery Protocol Data Unit，链路层发现[协议数据单元](http://baike.baidu.com/view/451435.htm)）中发布给与自己直连的邻居，邻居收到这些信息后将其以标准MIB（Management Information Base，管理信息库）的形式保存起来，以供[网络管理系统](http://baike.baidu.com/view/2523209.htm)查询及判断链路的通信状况。

## 2简介[编辑](http://baike.baidu.com/view/1930049.htm)

链路层发现[协议](http://baike.baidu.com/view/36190.htm)（Link Layer Discovery Protocol）-- IEEE802.1ab

2005年5月，此[协议](http://baike.baidu.com/view/36190.htm)已经被认可为IEEE802.1AB-2005标准。它代替了供应商私有的[协议](http://baike.baidu.com/view/36190.htm)，如Cisco公司的思科发现协议（Cisco Discovery Protocol）、Extreme Networks的EDP协议（Extreme Discovery Protocol）、Enterasys Networks的CDP协议（Cabletron Discovery Protocol）以及Nortel Networks的NDP协议（Nortel Discovery Protocol）等。

The Link Layer Discovery Protocol or LLDP is a vendor-neutral Layer 2 protocol that allows a network device to advertise its identity and capabilities on the local network. The protocol was formally ratified as IEEE standard 802.1AB-2005 in May 2005. It supersedes proprietary protocols like Cisco Discovery Protocol, Extreme Discovery Protocol and Nortel Discovery Protocol (also known as SONMP).

简单说来，LLDP是一种邻近发现[协议](http://baike.baidu.com/view/36190.htm)。它为[以太网](http://baike.baidu.com/view/848.htm)[网络设备](http://baike.baidu.com/view/1158081.htm)，如[交换机](http://baike.baidu.com/view/1077.htm)、[路由器](http://baike.baidu.com/view/1360.htm)和[无线局域网](http://baike.baidu.com/view/5470.htm)接入点定义了一种标准的方法，使其可以向网络中其他节点公告自身的存在，并保存各个邻近设备的发现信息。例如设备配置和设备识别等详细信息都可以用该[协议](http://baike.baidu.com/view/36190.htm)进行公告。

具体来说，LLDP定义了一个通用公告信息集、一个传输公告的[协议](http://baike.baidu.com/view/36190.htm)和一种用来存储所收到的公告信息的方法。要公告自身信息的设备可以将多条公告信息放在一个局域网数据包内传输，传输的形式为类型长度值（TLV）域。

## 3工作模式[编辑](http://baike.baidu.com/view/1930049.htm)

TxRx：既发送也接收LLDP[报文](http://baike.baidu.com/view/175122.htm)。

Tx：只发送不接收LLDP[报文](http://baike.baidu.com/view/175122.htm)。

Rx：只接收不发送LLDP[报文](http://baike.baidu.com/view/175122.htm)。

Disable：既不发送也不接收LLDP[报文](http://baike.baidu.com/view/175122.htm)。

当端口的LLDP工作模式发生变化时，端口将对[协议](http://baike.baidu.com/view/36190.htm)[状态机](http://baike.baidu.com/view/1906565.htm)进行初始化操作。为了避免端口工作模式频繁改变而导致端口不断执行初始化操作，可配置端口初始化延迟时间，当端口工作模式改变时延迟一段时间再执行初始化操作。

## 4LLDP报文[编辑](http://baike.baidu.com/view/1930049.htm)

### 概念

封装有LLDPDU的[报文](http://baike.baidu.com/view/175122.htm)称为LLDP报文，其[封装格式](http://baike.baidu.com/view/1942911.htm)有两种：Ethernet II和SNAP（Subnetwork Access Protocol，[子网](http://baike.baidu.com/view/65511.htm)访问[协议](http://baike.baidu.com/view/36190.htm)）。

### 发送机制

当端口工作在TxRx或Tx模式时，设备会周期性地向邻居设备发送LLDP报文。如果设备的本地配置发生变化则立即发送LLDP[报文](http://baike.baidu.com/view/175122.htm)，以将本地信息的变化情况尽快通知给邻居设备。但为了防止本地信息的频繁变化而引起LLDP[报文](http://baike.baidu.com/view/175122.htm)的大量发送，每发送一个LLDP报文后都需延迟一段时间后再继续发送下一个报文。

当设备的工作模式由Disable/Rx切换为TxRx/Tx，或者发现了新的邻居设备（即收到一个新的LLDP[报文](http://baike.baidu.com/view/175122.htm)且本地尚未保存发送该报文设备的信息）时，该设备将自动启用快速发送机制，即将LLDP报文的发送周期缩短为1秒，并连续发送指定数量的LLDP报文后再恢复为正常的发送周期。

### 接收机制

当端口工作在TxRx或Rx模式时，设备会对收到的LLDP[报文](http://baike.baidu.com/view/175122.htm)及其携带的TLV进行有效性检查，通过检查后再将邻居信息保存到本地，并根据TTL（Time To Live，生存时间） TLV中TTL的值来设置邻居信息在本地设备上的老化时间，若该值为零，则立刻老化该邻居信息。

# FDB Forwarding DataBase

[交换机](http://baike.baidu.com/view/1077.htm)从它的所有端口接收Media Access Control (MAC)地址信息，形成MAC地址表并维护它。当[交换机](http://baike.baidu.com/view/1077.htm)收到一帧数据时，它将根据自己的MAC地址表来决定是将这帧数据进行过滤还是转发。此时，维护的这张MAC表就是FDB地址表。

## 目录

1[MAC地址转发表](http://baike.baidu.com/view/2138578.htm#1)

2[富滇银行](http://baike.baidu.com/view/2138578.htm#2)

3[固定数据应答器](http://baike.baidu.com/view/2138578.htm#3)

## 1MAC地址转发表[编辑](http://baike.baidu.com/view/2138578.htm)

FDB——Forwarding DataBase

**一、概述**

如果收到[数据帧](http://baike.baidu.com/view/1366167.htm)的目的MAC地址不在FDB地址表中，那么该数据将被发送给除[源端口](http://baike.baidu.com/view/4728590.htm)外该[数据包](http://baike.baidu.com/view/25880.htm)所属VLAN的其他所有端口。

FDB地址表中的地址表项可以通过以下两个途径被加入：

[交换机](http://baike.baidu.com/view/1077.htm)自学习：交换机可以根据收到的[数据包](http://baike.baidu.com/view/25880.htm)的源MAC地址、端口、VLANID，来自动更新FDB地址表。 手工增加：可以通过命令行接口手工增加地址表项到FDB地址表中。 FDB地址表数目由产品决定。每一个FDB地址表项由MAC地址和VLANID唯一标识。每个FDB地址表项都包含以下内容：

MAC地址 、MAC地址关联的[端口号](http://baike.baidu.com/view/642103.htm)（Port）、MAC地址关联的VLAN的名称(VLAN name)、FDB地址表项的标志(Flags)。

FDB 地址表项标志的含义有如下：

System ：系统（[交换机](http://baike.baidu.com/view/1077.htm)）自动产生的第三层静态FDB地址表项。 Permenant：该FDB地址表项是一个[静态地址表](http://baike.baidu.com/view/1135598.htm)项。 Dynamic ：该FDB地址表项是一个[动态地址](http://baike.baidu.com/view/1449967.htm)表项。 L3 ：该FDB地址表项是一个用于三层转发的地址表项。 [FDB](http://baike.baidu.com/view/2138578.htm)地址表共有三种地址表项：

**动态地址表项**

最开始的时候，[交换机](http://baike.baidu.com/view/1077.htm)FDB地址表中的所有地址表项都是动态的。如果经过一段时间（老化时间Agingtime）之后，设备没有数据传输，那么该地址表项就会被删除。这样能防止地址表项变得过于庞大，当确信某个设备从网络中去除后，就把该设备的地址表项删除掉。当[交换机](http://baike.baidu.com/view/1077.htm)关机重启动或者reset时，所有的[动态地址](http://baike.baidu.com/view/1449967.htm)表项都将被删除。

**固定地址表项**

如果老化时间（Agingtime）被设为0，那么该地址表项将存储在MAC地址表中而不会被动态删除，直到交换机关机或者重启。

**永久地址表项**

永久地址表项将一直保存在MAC地址表中，即使[交换机](http://baike.baidu.com/view/1077.htm)关机或者重启。永久地址表项必须由系统管理员手工设定。一个永久地址表项可以是一个[单播](http://baike.baidu.com/view/492261.htm)地址，也可以是一个[组播地址](http://baike.baidu.com/view/1871353.htm)（本系统暂时不支持组播地址）。所有由命令行输入的[静态地址表](http://baike.baidu.com/view/1135598.htm)项都将被[存储](http://baike.baidu.com/view/87682.htm)为永久地址表项。永久地址表项一经建立，不会老化，但会随[交换机](http://baike.baidu.com/view/1077.htm)的配置变化而变化。

以下事件的发生会引起永久地址表项被删除：

删除一个与FDB静态表项关联的VLAN； 修改一个与FDB静态表项关联的VLAN的tag值；从VLAN中删除与FDB静态表项关联的一个端口；以下事件的发生不会引起永久地址表项的变化：

一个端口被关闭（disable） 一个端口被堵塞（block） 一个端口down掉（link down）

# STP（生成树协议）

## 目录

1[简介介绍](http://baike.baidu.com/subview/28816/5074615.htm#1)

2[技术原理](http://baike.baidu.com/subview/28816/5074615.htm#2)

3[功能介绍](http://baike.baidu.com/subview/28816/5074615.htm#3)

4[VLAN影响](http://baike.baidu.com/subview/28816/5074615.htm#4)

5[协议不足](http://baike.baidu.com/subview/28816/5074615.htm#5)

6[运行过程](http://baike.baidu.com/subview/28816/5074615.htm#6)

7[根端口依据](http://baike.baidu.com/subview/28816/5074615.htm#7)

8[指定端口依据](http://baike.baidu.com/subview/28816/5074615.htm#8)

9[端口状态](http://baike.baidu.com/subview/28816/5074615.htm#9)

10[STP潜在故障](http://baike.baidu.com/subview/28816/5074615.htm#10)

## 1简介介绍[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

STP（Spanning Tree Protocol）是[生成树协议](http://baike.baidu.com/view/72369.htm)的英文缩写。该协议可应用于在网络中建立树形拓扑，消除网络中的环路，并且可以通过一定的方法实现路径[冗余](http://baike.baidu.com/view/104445.htm)，但不是一定可以实现路径冗余。[生成树协议](http://baike.baidu.com/view/72369.htm)适合所有厂商的[网络设备](http://baike.baidu.com/view/1158081.htm)，在配置上和体现功能强度上有所差别，但是在原理和应用效果是一致的。

## 2技术原理[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

STP的基本原理是，通过在[交换机](http://baike.baidu.com/view/1077.htm)之间传递一种特殊的协议[报文](http://baike.baidu.com/view/175122.htm)，[网桥协议数据单元](http://baike.baidu.com/view/1624724.htm)（Bridge Protocol Data Unit，简称BPDU），来确定网络的[拓扑结构](http://baike.baidu.com/view/82343.htm)。BPDU有两种，配置BPDU（Configuration BPDU）和[TCN](http://baike.baidu.com/view/2651159.htm) BPDU。前者是用于计算无环的[生成树](http://baike.baidu.com/view/1340945.htm)的，后者则是用于在二层网络拓扑发生变化时产生用来缩短CAM表项的刷新时间的（由默认的300s缩短为15s）。

Spanning Tree Protocol(STP)在IEEE802.1D文档中定义。该协议的原理是按照树的结构来构造网络拓扑，消除网络中的环路，避免由于环路的存在而造成[广播风暴](http://baike.baidu.com/view/197497.htm)问题。

Spanning Tree Protocol(STP)的基本思想就是按照"树"的结构构造网络的[拓扑结构](http://baike.baidu.com/view/82343.htm)，树的根是一个称为根桥的桥设备，根桥的确立是由[交换机](http://baike.baidu.com/view/1077.htm)或网桥的BID（Bridge ID）确定的，BID最小的设备成为二层网络中的根桥。BID又是由网桥优先级和MAC地址构成，不同厂商的设备的网桥优先级的字节个数可能不同。由根桥开始，逐级形成一棵树，根桥定时发送配置BPDU，非根桥接收配置BPDU，刷新最佳BPDU并转发。这里的最佳BPDU指的是当前根桥所发送的BPDU。如果接收到了下级BPDU（新接入的设备会发送BPDU，但该设备的BID比当前根桥大），接收到该下级BPDU的设备将会向新接入的设备发送自己存储的最佳BPDU，以告知其当前网络中根桥；如果接收到的BPDU更优，将会重新计算[生成树](http://baike.baidu.com/view/1340945.htm)拓扑。当非根桥在离上一次接收到最佳BPDU最长寿命（Max Age，默认20s）后还没有接收到最佳BPDU的时候，该端口将进入监听状态，该设备将产生TCN BPDU，并从根端口转发出去，从指定端口接收到TCN BPDU的上级设备将发送确认，然后再向上级设备发送TCN BPDU，此过程持续到根桥为止，然后根桥在其后发送的配置BPDU中将携带标记表明拓扑已发生变化，网络中的所有设备接收到后将CAM表项的刷新时间从300s缩短为15s。整个收敛的时间为50s左右。

## 3功能介绍[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

[生成树协议](http://baike.baidu.com/view/72369.htm)最主要的应用是为了避免局域网中的单点故障、网络回环，解决成环[以太网](http://baike.baidu.com/view/848.htm)网络的“[广播风暴](http://baike.baidu.com/view/197497.htm)”问题，从某种意义上说是一种网络保护技术，可以消除由于失误或者意外带来的循环连接。STP也提供了为网络提供[备份](http://baike.baidu.com/view/235162.htm)连接的可能，可与SDH保护配合构成以太环网的双重保护。新型以太单板支持符合IEEE 802.1d标准的[生成树协议](http://baike.baidu.com/view/72369.htm)STP及IEEE 802.1w规定的[快速生成树](http://baike.baidu.com/view/1577932.htm)协议RSTP，收敛速度可达到1s。

但是，由于协议机制本身的局限，STP保护速度慢（即使是1s的收敛速度也无法满足电信级的要求），如果在[城域网](http://baike.baidu.com/view/61730.htm)内部运用STP技术，用户网络的动荡会引起运营商网络的动荡。目前在MSTP 组成环网中，由于SDH保护倒换时间比STP协议[收敛时间](http://baike.baidu.com/view/3077397.htm)快的多，系统采用依然是SDH MS-SPRING或[SNCP](http://baike.baidu.com/view/1740927.htm)，一般倒换时间在50ms以内。但测试时部分[以太网](http://baike.baidu.com/view/848.htm)业务的倒换时间为0或小于几个毫秒，原因是内部具有较大缓存。SDH保护倒换动作对MAC层是不可见的。这两个层次的保护可以协调工作，设置一定的“拖延时间”（hold-off），一般不会出现多次倒换问题。

## 4VLAN影响[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

L3，L4交换已经非常成熟。Internet中也越来越广泛地应用了交换技术，全交换网络已经非常普遍。在这些网络中，VLAN的使用是必不可少的。

VLAN是一个根据作用、计划组、应用等进行逻辑划分的交换式网络。与用户的物理位置没有关系。举个例子来说，几个[终端](http://baike.baidu.com/view/105503.htm)可能被组成一个部分，可能包括工程师或财务人员。当[终端](http://baike.baidu.com/view/105503.htm)的实际物理位置比较相近，可以组成一个局域网（LAN）。如果他们在不同的建筑物中，就可以通过VLAN将他们聚合在一起。同一个VLAN中的端口可以接受VLAN中的[广播包](http://baike.baidu.com/view/3113391.htm)。但别的VLAN中的端口却接受不到。

1、网络容错能力不强;

2、报文在环路网络中容易增生和无限循环；

3、不利在VLAN中实现流量[负载均衡](http://baike.baidu.com/view/51184.htm)

[生成树协议](http://baike.baidu.com/view/72369.htm)运行生成树算法(STA).[生成树](http://baike.baidu.com/view/1340945.htm)算法很复杂，但是其过程可以归纳为以下3个步骤：

（1）选择根网桥

（2）选择根端口

（3）选择指定端口

关于选择根网桥：选择根网桥的依据是网桥ID，网桥ID由网桥优先级和网桥MAC地址组成。网桥的默认优先级是32768.使用show mac-address-table时，显示在最前面的MAC地址就是计算时所使用的MAC地址。网桥ID值小的为根网桥，当优先级相同时，MAC地址小的为根网桥。

关于选择根端口：每个非根[交换机](http://baike.baidu.com/view/1077.htm)选择一个根端口。选择顺序为：到根网桥最低的根路径成本→发送BPDU的网桥ID较小→端口ID较小的。端口ID由端口优先级与端口编号组成。默认的端口优先级为128。

关于选择指定端口：每个网段上选择一个指定端口。选择顺序为：根路径成本较低→发送BPDU的[交换机](http://baike.baidu.com/view/1077.htm)的网桥ID值较小→本端口的ID值较小。另外，根网桥的接口皆为指定端口，因为根网桥上端口的根路径成本为0。

## 5协议不足[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

1、拓扑收敛慢，当网络拓扑发生改变的时候，生成树协议需要50-52秒的时间才能完成拓扑收敛。

2、不能提供[负载均衡](http://baike.baidu.com/view/51184.htm)的功能。当网络中出现环路的时候，[生成树协议](http://baike.baidu.com/view/72369.htm)简单的将环路进行Block，这样该链路就不能进行[数据包](http://baike.baidu.com/view/25880.htm)的转发，浪费网络资源。

## 6运行过程[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

生成树协议运行生成树算法（STA）。生成树算法很复杂，但是其过程可以归纳为以下三个部分。

（1）选择根网桥

（2）选择根端口

（3）选择指定端口（也有书籍称为转发端口）

选择根网桥的依据是[交换机](http://baike.baidu.com/view/1077.htm)的网桥优先级，网桥优先级是用来衡量网桥在[生成树](http://baike.baidu.com/view/1340945.htm)算法中优先级的十进制数，取值范围是0～65535.默认值是32768，网桥ID=网桥优先级+网桥MAC地址组成的，共有8个字节。由于[交换机](http://baike.baidu.com/view/1077.htm)的网桥优先级都是默认，所以在根网桥的选举中比较的一般是网卡MAC地址的大小，选取MAC地址小的为根网桥。

## 7根端口依据[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

（1）到根网桥[路径开销](http://baike.baidu.com/view/1624808.htm)最低。

根[路径开销](http://baike.baidu.com/view/1624808.htm)是两个网桥间的路径上所有链路的开销之和，也就是某个桥网到达根网桥的中间所有链路的路径开销之和。

附IEEE标准[路径开销](http://baike.baidu.com/view/1624808.htm)表：

|  |  |  |
| --- | --- | --- |
| 链路速度 | 开销（最新修订） | 开销（以前） |
| 10Gbps | 2 | 1 |
| 1Gbps | 4 | 1 |
| 100Mbps | 19 | 10 |
| 10Mbps | 100 | 100 |

（2）最低的发送方网桥ID。

（3）最低的端口ID。

由于端口的ID是由优先级和[端口号](http://baike.baidu.com/view/642103.htm)组成，保证了根端口的唯一性。

## 8指定端口依据[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

（1）根[路径开销](http://baike.baidu.com/view/1624808.htm)较低

（2）所在的交换机网桥ID值最小

（3）端口号最小

## 9端口状态[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

**Blocking（阻塞状态）：**此时，二层端口为非指定端口，也不会参与数据帧的转发。该端口通过接收BPDU来判断根交换机的位置和根ID，以及在STP拓扑收敛结束之后，各交换机端口应该处于什么状态，在默认情况下，端口会在这种状态下停留20秒钟时间。

**Listening（侦听状态）：**生成树此时已经根据交换机所接收到的BPDU而判断出了这个端口应该参与数据帧的转发。于是交换机端口就将不再满足于接收BPDU，而同时也开始发送自己的BPDU，并以此通告邻接的交换机该端口会在活动拓扑中参与转发数据帧的工作。在默认情况下，该端口会在这种状态下停留15秒钟的时间。

**Learning(学习状态)：**这个二层端口准备参与数据帧的转发，并开始填写MAC表。在默认情况下，端口会在这种状态下停留15秒钟时间。

**Forwarding（转发状态）：**这个二层端口已经成为了活动拓扑的一个组成部分，它会转发数据帧，并同时收发BPDU。

**Disabled（禁用状态）：**这个二层端口不会参与生成树，也不会转发数据帧。

## 10STP潜在故障[编辑](http://baike.baidu.com/subview/28816/5074615.htm)

**双工不匹配：**在点到点链路上，双工不匹配是一种常见的配置错误。当链路的一段采用手工的方式配置为了全双工模式，而另一侧却使用自动协商的默认配置的时候，那么就可能发生双工不匹配的情况。

**单向链路失效：**单向链路是产生桥接环路的一个非常常见的原因。如果光纤链路存在没有检测出来的故障或收发器故障，通常导致单向链路。在启用STP来提供网络冗余的情况下，对于两个链路伙伴之间所连接的物理链路，如果由于某种原因导致这条链路工作在单向通信的状态下，那么就可能导致桥接环路或路由选择黑洞，进而对维护网络稳定非常有害。

**帧破坏：**帧破坏是导致STP故障的另外一种原因。如果接口正在经受高速的物理错误，其结果有可能就会导致BPDU丢失，而这会使处于阻塞状态的接口过渡到转发状态。虽然如此，但因为STP默认参数是非常保守的，所以很少会发生上述情况。

**资源错误：**即使在通过专门的ASIC硬件执行大部分交换功能的高端交换机中，STP仍然由CPU来执行。这就意味着：如果处于某种原因而过度使用了网桥的CPU，那么就可能导致CPU没有足够的资源来发出BPDU。通常情况下，STP不是一种密集调用处理器的应用，而且STP的优先级高于其他进程。因此，出现资源问题的可能性并不大。

**PortFast配置错误：**如果管理员在端口上启用了PortFast特性，那么当链路启动的时候，端口就会绕过STP的监听和学习状态，并且直接过渡到转发状态。但是如果在错误的端口上配置了PortFast特性，那么这种快速过渡就有可能导致桥接环路。

# 网络拓扑发现

网络层拓扑发现的原理是结合ICMP和ARP以及SNMP，对指定的网络进行活动设备的检查，得到所有的活动设备，然后通过SNMP取得设备的基本信息，根据基本信息确定设备的类型，再根据设备的类型取得相应设备的详细信息。

## 目录

1[概述](http://baike.baidu.com/view/1667777.htm#1)

2[拓扑发现原理](http://baike.baidu.com/view/1667777.htm#2)

3[网络层的拓扑发现](http://baike.baidu.com/view/1667777.htm#3)

4[链路层的拓扑发现](http://baike.baidu.com/view/1667777.htm#4)

5[路由层的拓扑发现](http://baike.baidu.com/view/1667777.htm#5)

6[目前Java技术实现的拓扑发现产品](http://baike.baidu.com/view/1667777.htm#6)

## 1概述[编辑](http://baike.baidu.com/view/1667777.htm)

随着信息时代的到来，对计算机网络的依赖使得计算机网络本身运行的可靠性变得至关重要，对[网络管理](http://baike.baidu.com/view/325702.htm)也就有了更高的要求。

按照OSI的定义，[网络管理](http://baike.baidu.com/view/325702.htm)主要包括五个功能域：[故障](http://baike.baidu.com/view/553202.htm)管理、[配置管理](http://baike.baidu.com/view/569797.htm)、性能管理、安全管理和计费管理。在五大功能域中，配置管理是基础，它的主要功能包括发现网络的[拓扑结构](http://baike.baidu.com/view/82343.htm)、监视和管理[网络设备](http://baike.baidu.com/view/1158081.htm)的配置情况。其它的各项功能都以已知网络的[拓扑结构](http://baike.baidu.com/view/82343.htm)为基础。

网络拓扑发现的主要目的是获取和维护[网络节点](http://baike.baidu.com/view/1266538.htm)的存在信息和它们之间的连接关系信息，并在此基础上绘制出整个[网络拓扑图](http://baike.baidu.com/view/116613.htm)。网络管理人员在拓扑图的基础上对故障[节点](http://baike.baidu.com/view/47398.htm)进行快速定位。

## 2拓扑发现原理[编辑](http://baike.baidu.com/view/1667777.htm)

网络拓扑自动发现的方法很多，但归结起来主要有以下三种：**基于SNMP的网络拓扑发现方法；基于通用协议的网络拓扑发现方法；基于**[**路由协议**](http://baike.baidu.com/view/7031.htm)**的网络拓扑发现方法。**

## 3网络层的拓扑发现[编辑](http://baike.baidu.com/view/1667777.htm)

网络层拓扑发现的步骤如下：

**首先**通过[默认网关](http://baike.baidu.com/view/119302.htm)[路由器](http://baike.baidu.com/view/1360.htm)获取存在的[子网](http://baike.baidu.com/view/65511.htm)列表，**然后**通过ICMP Ping或[路由](http://baike.baidu.com/view/18655.htm)器中的ARP信息遍历指定子网中所有的活动设备，并用系统团体名库去找到设备的团体名，如果找到则用SNMP协议获取设备的基本信息，并判断出设备的类型（[路由](http://baike.baidu.com/view/18655.htm)器/[交换机](http://baike.baidu.com/view/1077.htm)/[防火墙](http://baike.baidu.com/view/3067.htm)/UPS/[主机](http://baike.baidu.com/view/23880.htm)等），并在此基础上获取相应设备的详细信息。如果没有找到设备的团体名，则默认此设备为[主机](http://baike.baidu.com/view/23880.htm)。

如何找到默认的路由[网关](http://baike.baidu.com/view/807.htm)呢？查找拓扑发现程序所在计算机的SNMP MIBII中的ipRouteTable，如果发现ipRouteDest值为0.0.0.0的记录，则说明程序所在的计算机设置了[默认网关](http://baike.baidu.com/view/119302.htm)，ipRouteNextHop值即为默认网关的地址。然后检查[默认网关](http://baike.baidu.com/view/119302.htm)的ipForwarding值。如果为1，则表明该[默认网关](http://baike.baidu.com/view/119302.htm)确实是路由设备，否则不是。

如何取得存在的[子网](http://baike.baidu.com/view/65511.htm)列表呢？遍历[路由器](http://baike.baidu.com/view/1360.htm)MIBII的IP管理组中管理对象ipRouteDest下的所有对象，以每个路由目的网络号为索引，查询ipRouteType字段的值。若该值为3（direct）表明为直接路由，若该值为4（indirect）则为间接路由。间接路由表明要通往目的网络或目的主机还要经过其它[路由器](http://baike.baidu.com/view/1360.htm)，而直接路由表明与目的网络或目的主机直接相连，这样就可以得到与[路由器](http://baike.baidu.com/view/1360.htm)直接相连的网络号。再根据网络号中的每条记录查询其路由掩码（ipRouteMask）。根据取得路由掩码，就可以确定每一个存在的网络[子网](http://baike.baidu.com/view/65511.htm)的IP地址范围。

如何发现其它的[路由](http://baike.baidu.com/view/18655.htm)设备？查找[默认路由](http://baike.baidu.com/view/1240261.htm)[网关](http://baike.baidu.com/view/807.htm)MIBII的IP管理组路由表中类型为间接路由的路由表项，得到路由的[下一跳](http://baike.baidu.com/view/1686120.htm)地址（ipRouteNextHop）。遍历[下一跳](http://baike.baidu.com/view/1686120.htm)地址给出的[路由](http://baike.baidu.com/view/18655.htm)设备，就可以得到更大的网络拓扑。

如何发现[网络层](http://baike.baidu.com/view/239600.htm)设备的连接关系？[子网](http://baike.baidu.com/view/65511.htm)与路由器的连接关系遍历每个路由器下包含的子网来确定，[主机](http://baike.baidu.com/view/23880.htm)与子网的关系可以通过主机IP与[子网掩码](http://baike.baidu.com/view/878.htm)来确定。

## 4链路层的拓扑发现[编辑](http://baike.baidu.com/view/1667777.htm)

链路层拓扑发现的原理是根据[交换机](http://baike.baidu.com/view/1077.htm)的cdp[邻居表](http://baike.baidu.com/view/4235114.htm)、端口ifIndex和port对应表、自学习表判断出各交换机的连接关系，具体的拓扑发现步骤如下：

A、获取所有[路由器](http://baike.baidu.com/view/1360.htm)的arp表信息，对于[主机](http://baike.baidu.com/view/23880.htm)则通过SendARP或netbios来获取arp信息。

B、获取所有[交换机](http://baike.baidu.com/view/1077.htm)的cdp[邻居表](http://baike.baidu.com/view/4235114.htm)、端口的ifIndex和port对应表、自学习表。

C、对[交换机](http://baike.baidu.com/view/1077.htm)的互连关系进行判断，如果某个交换机的某个端口学习到了[路由器](http://baike.baidu.com/view/1360.htm)的mac地址，则认为此端口为[上行端口](http://baike.baidu.com/view/4494198.htm)，如果交换机只有一个端口有自学习表项，也认为此端口是[上行端口](http://baike.baidu.com/view/4494198.htm)。如果一个[交换机](http://baike.baidu.com/view/1077.htm)的上行端口中没有接其它的交换机则此交换机为根交换机。

D、对于每个[交换机](http://baike.baidu.com/view/1077.htm)，如果它的某个端口和其它根上的多个端口冲突，通过cdp信息修正2者之间的连接关系，

用①表示邻居的根[交换机](http://baike.baidu.com/view/1077.htm)，用②表示自己的根交换机，

如果①=②，表示它和它的邻居已经在同一棵树上，不处理；

如果是[上行端口](http://baike.baidu.com/view/4494198.htm)上的邻居，将①的层次和父亲改成self的层次和父亲；将①下的所有接点的层次加（self的层次-1）那么多；将self的父亲改成邻居的端口，并将self的level增加（邻居的层次+1）那么多；将self下所有接点的level增加（邻居的层次+1）那么多。

如果是其他端口上的邻居，将②的层次和父亲改成邻居的层次和父亲；将②下的所有接点的层次加（邻居的层次-1）那么多；将邻居的父亲改成self的端口，并将邻居的level增加（self的层次+1）那么多；将邻居下所有接点的level增加（self的层次+1）那么多。

F、对每个[交换机](http://baike.baidu.com/view/1077.htm)，如果不是根交换机，则在其上加一个节点表示[上行端口](http://baike.baidu.com/view/4494198.htm)，并查找该交换机下的主机和UPS。

## 5路由层的拓扑发现[编辑](http://baike.baidu.com/view/1667777.htm)

路由层的拓扑发现原理是TraceRoute所有设备节点，根据返回的路由路径即可得到相关设备的路由拓扑关系，拓扑发现的步骤如下：

首先找到根[节点](http://baike.baidu.com/view/47398.htm)，如果没有则生成一个；然后对每一个非[路由器](http://baike.baidu.com/view/1360.htm)设备调用TraceRoute，将device id 为0而又没有子节点的treenode删掉，将device type是主机，但有子节点的设备改成路由器。

对于多址[路由器](http://baike.baidu.com/view/1360.htm)的问题，可以通过查看路由器的osid是否一致来确定是否属于同一个路由器的。

## 6目前Java技术实现的拓扑发现产品[编辑](http://baike.baidu.com/view/1667777.htm)

ObjectSNMP是采用Java技术和SNMP技术实现的网络拓扑发现产品，目前的功能如下：

全网设备发现：可以按网络号、IP范围、多个网络范围内，自动搜索发现设备，获取设备的基本信息、[设备类型](http://baike.baidu.com/view/167943.htm)（交换、[路由](http://baike.baidu.com/view/18655.htm)、路由交换、[终端设备](http://baike.baidu.com/view/34654.htm)、厂商特有类型等）、MAC地址、ARP表、[交换机](http://baike.baidu.com/view/1077.htm)端口、[路由器](http://baike.baidu.com/view/1360.htm)接口、[路由表](http://baike.baidu.com/view/149989.htm)、交换机[转发表](http://baike.baidu.com/view/8172217.htm)、主机IP地址等信息。

网络漫游发现：给定少数几个已知的网络号、IP范围，按用户指定的漫游深度和漫游广度，进行全网漫游发现。

网络拓扑自动发现：可以发现[交换机](http://baike.baidu.com/view/1077.htm)与交换机、交换机与PC机、交换机与[终端设备](http://baike.baidu.com/view/34654.htm)、交换机与[路由器](http://baike.baidu.com/view/1360.htm)、路由交换机与路由交换机之间 的连接关系。连接关系可以定位到具体的设备端口、设备接口上。支持在任意指定的设备之间发现它们的所有连接，在全网范围内发现连接关系。

ObjectSNMP的物理拓扑自动发现采用了全新的技术：即支持单一Cisco、华为网络，也支持各种厂商设备混合网络。支持模糊连接定位，在数据不全或设备缺失的情况下，尽可能发现连接关系。可在任意的网络环境中工作，不需要用户对网络做任何假设（如[路由器](http://baike.baidu.com/view/1360.htm)假设、根[交换机](http://baike.baidu.com/view/1077.htm)假设、上/下行端口假设、[边缘设备](http://baike.baidu.com/view/101626.htm)假设等）。

# ntdp

**NTDP**

（Neighbor Topology Discovery Protocol）邻居拓扑发现协议

NTDP是用来收集网络拓扑信息的协议。NTDP 为集群管理提供可加入集群的设备信息，收集指定跳数内的[交换机](http://baike.baidu.com/view/1077.htm)的拓扑信息。NDP 为NTDP 提供[邻接表](http://baike.baidu.com/view/549594.htm)信息，NTDP 根据邻接信息发送和转发NTDP 拓扑收集请求，收集一定网络范围内每个设备的NDP 信息和它与所有邻居的连接信息。收集完这些信息后，管理设备或者[网管](http://baike.baidu.com/view/6076.htm)可以根据需要使用这些信息，完成所需的功能。当成员设备上的NDP 发现邻居有变化时，通过握手报文将邻居改变的消息通知管理设备，管理设备可以启动NTDP 进行指定拓扑收集，从而使NTDP 能够及时反映网络拓扑的变化。

# CDP

CDP：思科发现协议（CDP：Cisco Discovery Protocol） 思科发现协议 CDP 基本上是用来获取相邻设备的[协议地址](http://baike.baidu.com/view/1252611.htm)以及发现这些设备的平台。CDP 也可为[路由器](http://baike.baidu.com/view/1360.htm)的使用提供相关接口信息。CDP 是一种独立媒体协议，运行在所有思科本身制造的设备上，包括[路由器](http://baike.baidu.com/view/1360.htm)、网桥、接入服务器和[交换机](http://baike.baidu.com/view/1077.htm)。需要注意的是，CDP是工作在 Layer 2 的协议，默认情况下，每60秒以 01-00-0c-cc-cc-cc 为目的地址发送一次[组播](http://baike.baidu.com/view/492256.htm)通告，当达到180秒的holdtime上限后仍未获得邻居设备的通告时，将清除邻居设备信息。

思科协议

SNMP 中结合使用 CDP 管理信息基础 MIB，能使网络管理应用获知设备类型和相邻设备的 SNMP 代理地址，并向这些设备发送 SNMP 查询请求。Cisco 发现协议支持CISCO-CDP-MIB。

CDP 运行在所有的媒体上，从而支持子网访问协议 SNAP，包括局域网、帧中继和异步传输模式 ATM物理媒体。CDP 只运行于数据链路层，因此，支持不同网络层协议的两个系统彼此相互了解。

CDP 配置的每台设备发送周期性信息，如我们所知的广告到组播地址。每台设备至少广告一个地址，在该地址下，它可以接收 SNMP 信息。广告包括生存期，或保持时间等信息，这些信息指出了在取消之前接收设备应该保持 CDP 信息的时间长短。此外每台设备还要注意其它设备发出的周期性 CDP 信息，从中了解相邻设备信息并决定那些设备的媒体接口什么时候增长或降低。

CDP 版本2，是目前该协议使用最普遍的版本，它具有更高的智能设备跟踪等性能。支持该性能的报告机制，提供快速差错跟踪功能，有利于缩短停机时间（Downtime）。报告差错信息可以发送到控制台或日志服务器（Logging Server），这些差错信息包括连接端口上不匹配（Unmatching）的本地 VLAN IDs（IEEE 802.1Q）以及连接设备间不匹配的端口双向状态。

配置命令：cdp run(CDP协议在路由器全局启用的命令）[2]

# Object SNMP

## 目录

1[简介](http://baike.baidu.com/view/7166512.htm#1)

2[与传统SNMP开发的比较](http://baike.baidu.com/view/7166512.htm#2)

3[常用数据采集服务](http://baike.baidu.com/view/7166512.htm#3)

▪ [系统类MIB描述](http://baike.baidu.com/view/7166512.htm#3_1)

▪ [IP网络类MIB描述](http://baike.baidu.com/view/7166512.htm#3_2)

▪ [交换机MIB描述](http://baike.baidu.com/view/7166512.htm#3_3)

▪ [硬件设施类MIB描述](http://baike.baidu.com/view/7166512.htm#3_4)

▪ [软件类MIB描述](http://baike.baidu.com/view/7166512.htm#3_5)

▪ [Java类MIB描述](http://baike.baidu.com/view/7166512.htm#3_6)

Object SNMP(Object Simple Network Management Protocol)是一个面向对象的SNMP开发组件，与关系数据库流行的O-R Mapping技术类似，实现了Object到SNMP MIB的操作映射（简称O-M Mapping），系统提供了Java/SNMP网关、RMI/SNMP网关 、Web Service/SNMP网关服务。开发者无需理解SNMP协议和MIB细节，只需要定义普通数据对象，然后将数据对象按添加、删除、修改、查询等模式提交给网关即可。Object SNMP网关会自动将数据对象按开发者的期望发送到SNMP设备，并返回用户需要的数据对象。

## 1简介[编辑](http://baike.baidu.com/view/7166512.htm)

Object SNMP提供诸如分布式代理SNMP网关、 网络拓扑发现、物理拓扑发现、网络自动搜索发现 、设备连接关系发现、设备类型发现、网络漫游搜索、以及定义好的可管理各种设备的MIB数据等高级功能。

## 2与传统SNMP开发的比较[编辑](http://baike.baidu.com/view/7166512.htm)

|  |  |  |
| --- | --- | --- |
| 比较项目 | 基于传统SNMP协议栈开发 （如各种开源或商业SNMP协议栈） | 使用Object SNMP开发 |
| 软件架构 | 实现对SNMP协议的原始封装。 | 底层实现对SNMP协议的封装，上层提供基于对象的网关服务和各种高级服务。 |
| 开发技能要求 | 需要对Socket、PDU/UDP、SNMP Get/SNMP Set/GetNext/GetBulk、MIB语法、OID规范、SNMP数据类型等有开发级的掌握。 | 仅需要对SNMP有概念了解就可开发。对开发人员完全屏蔽SNMP技术细节。并且有对象代码的自动生成工具，进一步简化开发。 |
| 开发效率 | 大量的SNMP协议层操作代码 | 对象级的上层抽象操作。代码量仅有传统模式的10%。 |
| 开发时间 | 一般情况下，软件生命周期内需要专人负责SNMP模块的开发和后期维护。 | 使用ObjectSNMP组件后，开发人员在SNMP模块开发和设备联调上花费的时间仅有传统模式的20%。 |
| 软件性能 | 需要大量的调优、测试、验证来保证。 | ObjectSNMP网关通道已通过基准性能验证，新定义的数据对象直接在网关通道上传递。 |
| 软件模式 | 无，需要开发人员重新建立。 | 有统一的对象模式和网关模式，与整体软件的对象风格保持一致。 |
| 网关功能 | 无 | 同时支持Java、RMI、Web Service、Proxy等多种网关。 |
| 高级服务 | 无 | 全新技术的网络拓扑自动发现（全网发现、子网发现、IP发现、漫游搜索、物理连接拓扑发现） 和网络资源发现（发现网络端口、硬件设施、磁盘、CPU、软件服务） |
| 预定义MIB对象 | 无 | 积累了多种MIB数据采集服务，如网络接口、流量、数据包、IP、CPU、磁盘、内存、IP-MAC、802.1X、AAA、路由信息、交换机MAC表、WiFi MIB等MIB数据，可以直接提供给用户使用。 |

## 3常用数据采集服务[编辑](http://baike.baidu.com/view/7166512.htm)

ObjectSNMP提供如下常用数据采集服务，使用者可直接使用：

### 系统类MIB描述

(1). SNMP Agent系统信息(MibSystem)

**描述**：SNMP设备的基本描述信息，如系统名称、启动时间、描述等信息。所有支持SNMP的设备都应该支持此信息。

**Mib类型**：Mib组

**支持的设备**：[路由器](http://baike.baidu.com/view/1360.htm)、[交换机](http://baike.baidu.com/view/1077.htm)、[防火墙](http://baike.baidu.com/view/3067.htm)、服务器、计算机等所有支持SNMP的设备。

**MIB获取方法**：MibSystem mibResult=SNMPAPI. getMibObject(new MibSystem(),target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibSystem

**Java对象描述**：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 中文名 | 是否可写 | 描述 |
| sysDescr | 系统描述 | 只读 | 系统的描述信息。描述信息可作为识别设备类型的补充手段。 |
| sysObjectID | 系统的私有OID | 只读 | 如果系统申请了企业私有OID，显示系统私有OID。私有OID可以作为识别设备类型的方法。 |
| sysUpTime | 启动时间 | 只读 | 系统SNMP Agent模块运行的时间，一般可认为是系统的运行时间。单位：百分之一秒 |
| sysContact | 系统联系人 | 可写 | 系统的管理员或联系人 |
| sysName | 系统名称 | 可写 | 系统的名称 |
| sysLocation | 系统位置 | 可写 | 系统的物理位置 |
| sysServices | 服务类型 | 只读 | 系统提供的服务类型。OSI划分的服务类型。目前很少用到。 |

(2). 系统IP地址信息(MibIPAddrEntry)

**描述**：设备的IP地址、[掩码](http://baike.baidu.com/view/68.htm)、网关等

**Mib类型**：Mib表。设备有多个IP地址。

**支持的设备**：路由器、交换机、防火墙、服务器等支持IP协议的设备

**MIB获取方法**：MibIPAddrEntry mibResult=SNMPAPI. getMibObject(new MibIPAddrEntry(),target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibIPAddrEntry

(3). SNMP Agent统计信息(MibSNMP)

**描述**：SNMP Agent模块统计的SNMP 请求、应答、错误等统计数据

**Mib类型**：Mib组

**支持的设备**：大部分支持SNMP的设备，都提供此信息。

**MIB获取方法**：MibSNMP mibResult=SNMPAPI. getMibObject(new MibSNMP(),target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibSNMP

### IP网络类MIB描述

(1). 网络接口/端口信息(MibIfEntry)

**描述**：[网络设备](http://baike.baidu.com/view/1158081.htm)、计算机均是多网口、多网络接口设备。通过此信息可以了解设备上有多少个网卡、[交换机](http://baike.baidu.com/view/1077.htm)接口、[路由器](http://baike.baidu.com/view/1360.htm)端口、网络接口；并进一步知道每个接口/端口的详细信息。

**Mib类型**：Mib表。设备有多个网络接口。

**支持的设备**：路由器、交换机、防火墙、服务器等端口/接口设备

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibIfEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibIfEntry

(2).网络接口流量和数据包信息(MibIfEntry)

**描述**：网络设备、计算机均是多网口、多网络接口设备。比如[交换机](http://baike.baidu.com/view/1077.htm)有多个接口，每个接口有独立的流量和数据出入统计信息。

**Mib类型**：Mib表

**支持的设备**：路由器、交换机、防火墙、服务器等端口/接口设备

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibIfEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibIfEntry

(3).IP层流量和输入输出统计信息(MibIP)

**描述**：通过整个设备系统的的流量信息、整个设备系统的输入统计信息和输出统计信息。

**Mib类型**：Mib组

**支持的设备**：路由器、交换机、防火墙、服务器等支持IP协议的设备。

**MIB获取方法**：MibIP mibResult=SNMPAPI. getMibObject(new MibIP(),target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibIP

(4).路由表信息(MibIPRouterEntry)

**描述**：IP设备的路由表信息。

**Mib类型**：Mib表。多项路由表信息。

**支持的设备**：路由器、计算机、[防火墙](http://baike.baidu.com/view/3067.htm)、服务器等

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibIPRouterEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibIPRouterEntry

(5). TCP连接资源和统计信息(MibTCPConnEntry)

**描述**：系统中所有TCP网络连接的信息，包含本地TCP监听和远程TCP连接。

**Mib类型**：Mib表。有多条TCP连接。

**支持的设备**：路由器、交换机、防火墙、服务器等

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibTCPConnEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibTCPConnEntry

(6). UDP监听信息(MibUDPEntry)

**描述**：系统中所有UDP监听信息。

**Mib类型**：Mib表。

**支持的设备**：路由器、交换机、防火墙、服务器等

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibUDPEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibUDPEntry

(7). IP-MAC关联表(MibMacIP)

**描述**：IP地址和MAC地址的学习表。设备学习到的MAC地址和IP地址成对显示出来。

**Mib类型**：Mib表。多个IP-MAC地址对。

**支持的设备**：路由器、交换机、计算机等

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibIpAdEntAddr.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. IpAdEntAddr

### 交换机MIB描述

(1). 交换机端口基本信息(Dot1dBasePortEntry)

**描述**：查看[交换机](http://baike.baidu.com/view/1077.htm)所有的物理端口，物理端口与网络接口之间的对应关系，端口的基本

信息

**Mib类型**：Mib表

**支持的设备**：二层/[三层交换机](http://baike.baidu.com/view/44586.htm)

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (Dot1dBasePortEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. Dot1dBasePortEntry

(2). [交换机](http://baike.baidu.com/view/1077.htm)转发MAC地址和状态(Dot1dTpFdbEntry)

**描述**：[交换机](http://baike.baidu.com/view/1077.htm)每个端口学习到的物理地址和地址状态。可用于搜索和发现[交换机](http://baike.baidu.com/view/1077.htm)下设备。

**Mib类型**：Mib表

**支持的设备**：二层/三层交换机

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (Dot1dTpFdbEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. Dot1dTpFdbEntry

### 硬件设施类MIB描述

(1). 硬件设施和工作状态(MibDeviceEntry)

**描述**：计算机中的[硬件设施](http://baike.baidu.com/view/2394256.htm)列表，以及设备的状态。如键盘、CPU、[网卡](http://baike.baidu.com/view/4230.htm)、鼠标等

**Mib类型**：Mib表。系统有多种设施。

**支持的设备**：计算机、服务器、操作系统

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibDeviceEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibDeviceEntry

(2). 存储系统状态(MibDiskAndMemoEntry)

**描述**：[物理磁盘](http://baike.baidu.com/view/2815078.htm)、逻辑磁盘、内存、[虚拟内存](http://baike.baidu.com/view/976.htm)的使用情况。

**Mib类型**：Mib表。

**支持的设备**：计算机、服务器、操作系统

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibDisskAndMemoEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib.MibDiskAndMemoEntry

(3). CPU负载信息(MibProcessorEntry)

**描述**：系统中CPU的负载情况。

**Mib类型**：Mib表。

**支持的设备**：计算机、服务器、操作系统、高端[网络设备](http://baike.baidu.com/view/1158081.htm)

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibProcessorEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibProcessorEntry

### 软件类MIB描述

(1). 软件进程状态信息(MibSoftwareRunEntry)

**描述**：系统中正在运行的进程信息，如：进程名、进程路径、运行状态等。

**Mib类型**：Mib表

**支持的设备**：计算机、服务器、操作系统等

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibSoftwareRunEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib.MibSoftwareRunEntry

(2). 已安装的软件信息(MibSoftwareInstallEntry)

**描述**：系统中已安装的软件信息。

**Mib类型**：Mib表

**支持的设备**：计算机、服务器、操作系统等

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibSoftwareInstallEntry.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibSoftwareInstallEntry

### Java类MIB描述

(1). Java服务器基本信息(MibJvmOS)

**描述**：Java系统的基础操作系统信息。

**Mib类型**：Mib组

**支持的设备**：使用SUN JDK/JRE的Java/J2EE软件、服务器。如：Tomcat、JBoss、SUN ONE

Server、Oracle OC4J Server等。

**MIB获取方法**：MibJvmOS mibResult=SNMPAPI. getMibObject(new MibJvmOS(),target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibJvmOS

(2). JVM基本信息(MibJvmRuntime)

**描述**：Java[虚拟机](http://baike.baidu.com/view/1132.htm)的全局基本信息。

**Mib类型**：Mib组

**支持的设备**：使用SUN JDK/JRE的Java/J2EE软件、服务器。如：Tomcat、JBoss、SUN ONE

Server、Oracle OC4J Server等。

**MIB获取方法**：MibJvmRuntime mibResult=SNMPAPI. getMibObject(new MibJvmRuntime(),target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibJvmRuntime

(3). Java内存和[堆栈](http://baike.baidu.com/view/93201.htm)状态(MibJvmMemory)

**描述**：JVM的[垃圾回收](http://baike.baidu.com/view/159846.htm)、内存、堆栈情况。

**Mib类型**：Mib组

**支持的设备**：使用SUN JDK/JRE的Java/J2EE软件、服务器。如：Tomcat、JBoss、SUN ONE

Server、Oracle OC4J Server等。

**MIB获取方法**：MibJvmMemory mibResult=SNMPAPI. getMibObject(new MibJvmMemory(),target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibJvmMemory

(4). Java线程和资源消耗情况(MibJvmThreading)

**描述**：Java系统中全部线程的信息，包含线程的名称、运行状态、CPU时间等。

**Mib类型**：Mib表

**支持的设备**：使用SUN JDK/JRE的Java/J2EE软件、服务器。如：Tomcat、JBoss、SUN ONE

Server、Oracle OC4J Server等。

**MIB获取方法**：List mibResultList=SNMPAPI. getAllTableData (MibJvmThread.class,target);

**对应的Java对象**：com.zhtelecom.common.snmp.mib. MibJvmThreading