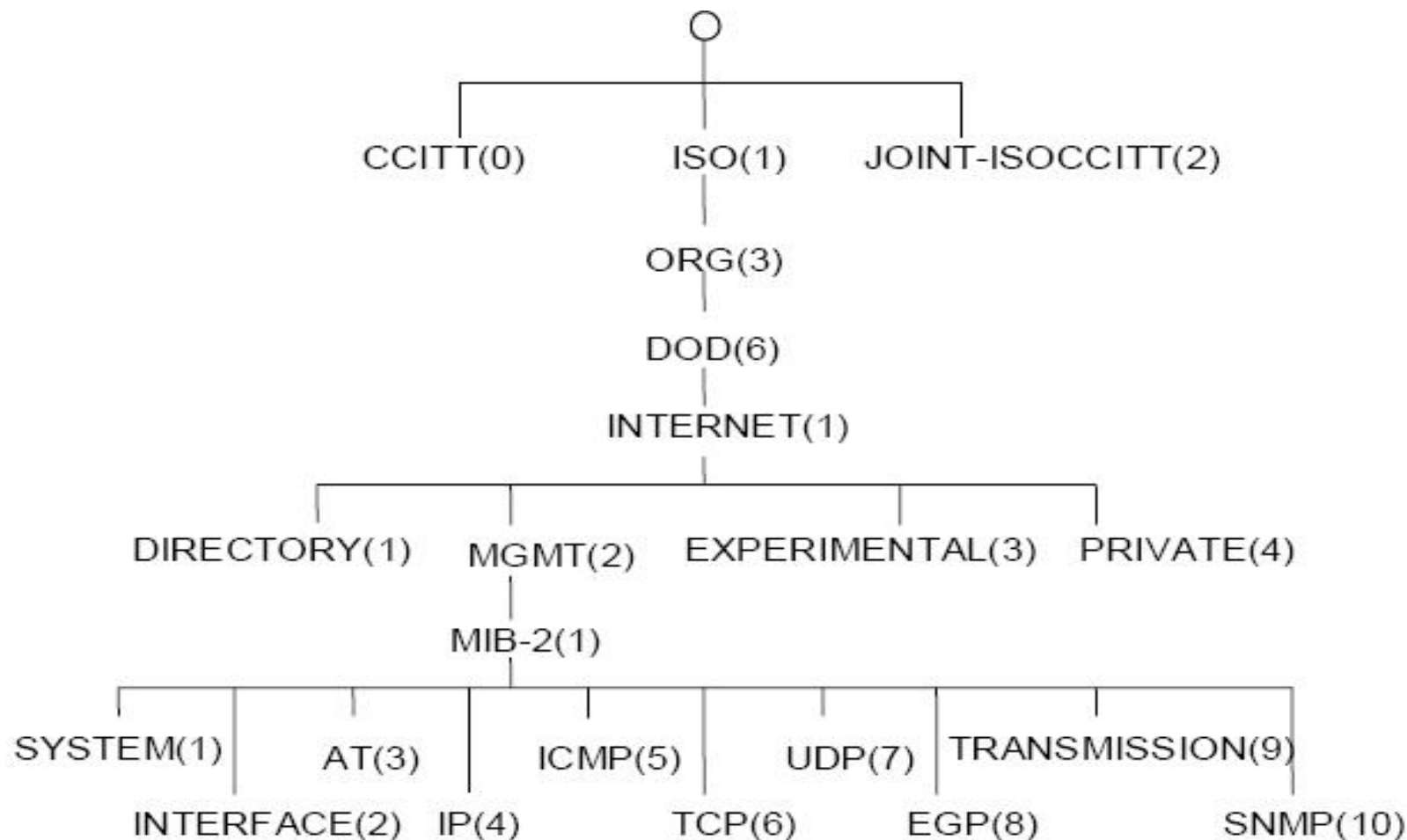


# 1 MIB-2 功能组

- RFC 1213中定义了管理信息库第2版，即MIB-2。B22080228徐基恒  
MIB-2 包含11个功能组，共171个对象。



# MIB-2 功 能 组

功 能 组	对象标识符	主 要 描 述
System	Mib-2 1	系统总体信息，如系统说明和管理信息
Interfaces	Mib-2 2	关于子网的各个接口的信息
At	Mib-2 3	关于IP地址与物理地址的转换
<b>Ip</b>	<b>Mib-2 4</b>	<b>关于IP的信息</b>
<b>Icmp</b>	<b>Mib-2 5</b>	<b>关于ICMP的信息</b>
<b>Tcp</b>	<b>Mib-2 6</b>	<b>关于TCP的信息</b>
Udp	Mib-2 7	关于UDP的信息
Egp	Mib-2 8	关于EGP的信息
<b>Cmot</b>	<b>Mib-2 9</b>	<b>关于CMOT协议保留</b>
Transmission	Mib-2 10	传输介质的管理信息，为传输信息保留
Snmp	Mib-2 11	关于SNMP的信息

# MIB-2 功 能 组

## 1. 系统组(system)

- 系统（System）组是MIB-2中最基本的一个组，提供了系统的一般信息，包含一些最常用的被管对象。
- 网络管理系统一旦发现新的系统被加到网络中，首先需要访问该系统的这个组，来获取该系统的名称、物理地点和联系人等信息，所有系统都必须包含System组。

# MIB-2 功 能 组

## System(mib-2 1)

sysDescr (1)	系统描述
sysObjectID (2)	系统制造商标识
sysUpTime (3)	系统的运行时间
sysContact (4)	系统的管理人员
sysName (5)	系统的名字
sysLocation (6)	系统的放置地点
sysServices (7)	系统提供的协议层服务

系统组(System组)的对象标识符子树

# MIB-2 功 能 组

对象	语法	访问方式	功能描述	用途
sysDescr (1)	DisplayString (SIZE (0..255))	RO	有关硬件和操作系统的描述	配置管理
sysObjectID(2)	OBJECT IDENTIFIER	RO	系统制造商标识	故障管理
sysUpTime (3)	Timeticks	RO	系统运行时间	故障管理
sysContact (4)	DisplayString (SIZE (0..255))	RW	系统管理人员描述	配置管理
sysName (5)	DisplayString (SIZE (0..255))	RW	系统名	配置管理
sysLocation(6)	DisplayString (SIZE (0..255))	RW	系统的物理位置	配置管理
sysServices (7)	INTEGER (0..127)	RO	系统服务	故障管理

系统组的对象

# MIB-2 功 能 组

- 对象sysServices是7位二进制数，每一位对应OSI/RM 7层协议中的一层。如果系统提供某一层服务，则对应的位为1，否则为0。

例如系统提供应用层和传输层服务，则该系统的sysServices对象的值为： $1001000=72_{(10)}$

- 系统启动时间sysUpTime可用于故障管理。比如，管理站可以周期地查询这个值，如果发现当前得到的值比最近一次得到的值小，则可推断出系统已经重启动过。

# MIB-2 功 能 组

## 2. 接口组(interface group)

- 接口组提供了网络实体物理层接口的信息，包括主机接口的配置信息和每个接口上发生事件的统计信息。
- 所有的SNMP代理都要求实现interface组。
- 接口组中有两个对象：
  - 👉 一个标量对象ifNumber：指网络接口数。
  - 👉 一个表对象ifTable，每个接口对应一个表项。  
该表的索引是ifIndex，取值为：1~ifNumber

# MIB-2 功 能 组

ifNumber(1) 网络接口数，记录一个网络设备的所有接口的总数

ifTable(2) 记录接口的信息，一个接口占表的一行记录，表长为ifNumber

ifEntry(1)

ifIndex(1) 接口表惟一的索引项，索引值为1— ifNumber范围

ifDescr(2) 接口的描述，制造商名，产品名和版本等

ifType(3) 接口的类型,用一个整数表示,表示物理层和数据链路层协议确定的接口类型

ifMtu(4) 表示该接口上可以发送或接收的最大协议数据单元大小(位组数)

ifSpeed(5) 指定一个接口的传输速率，单位为“位/秒”（bps）

ifPhysAddress(6) 接口的物理地址

ifAdminStatus(7) 用于配置接口的管理状态，up(1) down(2) testing(3)

ifOperStatus(8) 提供一个接口的当前操作状态， up(1) down(2) testing(3)

ifLastChange(9) 接口进入当前状态的时间

输入 { ifInOctets(10) 接口收到的总字节数

ifInUcastPkts(11) 单点发送到一种高层协议上的包的总数目

ifInNUcastPkts(12) 发往高层协议的非单播包数（广播和多播）

ifInDiscards(13) 接口丢弃的输入包数

ifInErrors(14) 有错的输入包数

ifInUnknownPorotos(15) 由于未知或不支持的协议而被抛弃的输入包数

输出 { ifOutOctets(16) 接口发送的总字节数

ifOutUcastPkts(17) 高层协议请求传输的单播包数

ifOutNUcastPkts(18) 高层协议请求的非单播（广播和多播）包数

ifOutDiscards(19) 由于资源局限而导致丢弃的发出包的总数目

ifOutErrors(20) 由于错误而导致丢弃的发出包的总数目

ifOutQLen(21) 输出包队列中包的总数

ifSpecfic(22) 指向MIB中特定介质的定义



对象	语法	权限	功能描述
ifNumber	INTEGER	RO	<b>网络接口数</b>
ifTable	SEQUENCE OF ifEntry	NA	接口表
ifEntry	SEQUENCE	NA	接口表项
<b>ifIndex</b>	INTEGER	RO	<b>唯一的索引</b>
ifDescr	DisplayString(SIZE (0..255))	RO	接口描述信息，制造商名， 产品名和版本等
ifType	INTEGER	RO	<b>物理层和数据链路层协议确定的接口类型</b>
ifMtu	INTEGER	RO	最大协议数据单元大小
ifSpeed	Gauge	RO	<b>接口数据速率</b>
ifPhysAddress	PhysAddress	RO	<b>接口物理地址</b>
ifAdminStatus	INTEGER	RW	<b>管理状态：</b> up (1) down (2) testing (3)

对象	语法	权限	功能描述
<code>ifOperStatus</code>	INTEGER	R0	<b>操作状态:</b> up (1) down (2) testing (3)
<code>ifLastChange</code>	TimeTicks	R0	接口进入当前状态的时间
<code>ifInOctets</code>	Counter	R0	<b>接口收到的总字节数</b>
<code>ifInDiscards</code>	Counter	R0	丢弃的分组数
<code>ifInErrors</code>	Counter	R0	接收的错误分组数
<code>ifInUnknownPorotos</code>	Counter	R0	未知协议的分组数
<code>ifOutOctets</code>	Counter	R0	<b>通过接口输出的分组数</b>
<code>ifOutDiscards</code>	Counter	R0	丢弃的分组数
<code>ifOutErrors</code>	Counter	R0	输出的错误分组数

# MIB-2 功 能 组

- 接口组中的对象可用于故障管理、性能管理和计费管理。

比如，本组两个关于接口状态的对象：

- 👉 ifAdminStatus表示操作员设定的管理状态，
- 👉 ifOperStatus表示接口的实际工作状态。
- 👉 这两个变量状态组合的含义如下表所示：

ifOperStatus	ifAdminStatus	含义
up(1)	up(1)	正常
down(2)	up(1)	故障
down(2)	down(2)	停机
testing(3)	testing(3)	测试

- 对象 `ifSpeed` 是一个只读的计量器，表示接口的比特速率。例如 `ifSpeed` 取值 100000000，表示 10 Mb/s。有些接口速率可根据参数变化，`ifSpeed` 的值反映了接口当前的数据速率。
- 通过检查进出接口的字节数或队列长度来检测拥挤：**性能管理**
- 可以统计出输入/输出的错误率，即
  - 👉 输入错误率 =  $\text{ifInErrors} / (\text{ifInUcastPkts} + \text{ifInNUcastPkts})$
  - 👉 输出错误率 =  $\text{ifOutErrors} / (\text{ifOutUcastPkts} + \text{ifOutNUcastPkts})$
- 该组提供接口发送的字节数和分组数，这些数据可作为**计费**的依据。
- 接口的利用率：利用率 =  $(\text{每秒的字节数} \times 8) / \text{ifspeed}$

# MIB-2 功 能 组

## 3. 地址转换组 (at)

### ● 地址转换组 (Address Translation)

☞ 仅包含一个表对象，表中的每一行对应系统的一个物理接口，表示：网络地址到接口的物理地址的映射。

☞ 通常，网络地址就是IP地址，而物理地址决定于实际采用的物理子网情况。

例如，如果接口对应的是以太网，则物理地址是MAC地址，如果对应X.25分组交换网，则物理地址可能是一个X.121地址。

# MIB-2 功 能 组

At(mib-2 3)

└─ atTable (1)

└─ atEntry (1)

└─ atIf Index (1)

└─ atPhys Address (2)

└─ atNet Address (3)

MIB-2地址转换组

# MIB-2 功 能 组

## 4. IP组

- IP组提供了与IP协议有关的信息。

☞ 一个网络结点中有关IP实现和操作的信息都包含在IP组

- IP组共23个对象，大致分为4类：

☞ 有关性能和故障监控的**标量对象**。

☞ 关于IP地址**信息表**。

☞ 关于IP路由信息的**路由表**。

☞ 转换IP地址到物理地址的**映射表**。

___ ipForwarding(1)	设备是否被设置成IP网关。1—是；2—否
___ ipDefaultTTL(2)	IP头中的Time To Live字段的值
___ ipInReceives(3)	接收的全部IP数据报的总数目，包含错误的数据报
___ ipInHdrErrors(4)	由于其IP包头中错误而丢弃的输入数据报数目
___ ipInAddrErrors(5)	由于目的IP地址无效而导致丢弃的输入IP数据报数目
___ ipForwDatagrams(6)	在接收的IP数据报中，转发的数据报数目
___ ipInUnknownProtos(7)	接收的数据报中，由于未知协议或不支持协议而丢弃的数据报数目
___ ipInDiscards(8)	由于资源局限（如：缓冲区空间不足）而丢弃的输入数据报数目
___ ipInDelivers(9)	由IP层成功提交给上层的数据报数目
___ ipOutRequests(10)	本地IP的上层协议提供给IP传送的IP数据报数目
___ ipOutDiscards(11)	由于资源局限(如：缓冲区空间不足)而丢弃的输出数据报数目
___ ipOutNoRoutes(12)	因无法找到到达目标地址的路由而丢弃的输出数据报数目
___ ipReasmTimeout(13)	接收到的IP分段在等待重组时要保持的时间值，以秒为单位
___ ipReasmReqds(14)	接收到需要在本地重组的IP分段数目
___ ipReasmOKs(15)	成功重组的IP数据报数目
___ ipReasmFails(16)	IP重组检测到的失败次数
___ ipFragOKs(17)	成功分段的数据报数目
___ ipFragFails(18)	由于需要分段而无法分段而导致丢弃的IP数据报数目
___ ipFragCreates(19)	创建的IP分段数目
___ ipAddrTable(20)	
___ ipRouteTable(21)	
___ ipNetToMediaTable(22)	
___ ipRoutingDiscards(23)	尽管有效仍然被丢弃(如资源局限)的路由表项目数



IP组对象	语法	访问	功能描述
ipForwarding (1)	INTEGER	RW	IP gateway (1), IP host (2)
ipDefaultTTL (2)	INTEGER	RW	IP头中的Time To Live字段的值
ipInReceives (3)	Counter	R0	IP层从下层接收的数据报总数
ipInHdrErrors (4)	Counter	R0	由于IP头出错而丢弃的数据报
ipInAddrErrors (5)	Counter	R0	地址出错的数据报
ipForwDatagrams (6)	Counter	R0	已转发的数据报
ipInUnknownProtos(7)	Counter	R0	不支持数据报的协议，因而被丢弃
ipInDiscards (8)	Counter	R0	因缺乏缓冲资源而丢弃的数据报
ipInDelivers (9)	Counter	R0	由IP层提交给上层的数据报
ipOutRequests (10)	Counter	R0	由IP层交给下层需要发送的数据报
ipOutDiscards (11)	Counter	R0	在输出端因缺乏缓冲资源而丢弃的数据报
ipOutNoRoutes (12)	Counter	R0	没有到达目标的路由而丢弃的数据报
ipReasmTimeout (13)	INTEGER	R0	数据段等待重装配的最长时间（秒）

对象	语法	访问	功能描述
ipReasmReqds (14)	Counter	R0	需要重装配的数据段
ipReasmOKs (15)	Counter	R0	成功重装配的数据段
ipReasmFails (16)	Counter	R0	不能重装配的数据段
ipFragOKs (17)	Counter	R0	分段成功的数据段
ipFragFails (18)	Counter	R0	不能分段的数据段
ipFragCreates (19)	Counter	R0	产生的数据报分段数
ipAddrTable (20)	SEQUENCE OF	NA	<b>IP地址表</b>
ipRouteTable (21)	SEQUENCE OF	NA	<b>IP路由表</b>
ipNetToMediaTable (22)	SEQUENCE OF	NA	<b>IP地址转换表</b>
ipRoutingDiscards (23)	Counter	R0	无效的路由项,

# MIB-2 功 能 组

IP地址表(ipAddrTable)包含与本地IP地址有关的信息。每一行对应一个IP地址，由ipAddrEntIfIndex作为索引项，其值与接口表的ifIndex一致。

ipAddrTable(ip 20)

└─ ipAddrEntry(1)

└─ ipAdEtnAddr(1) 本地主机IP地址

└─ ipAdEtnIfIndex(2) 对应接口表的索引

└─ ipAdEtnNetMask(3) 基本路由度量(整数)

└─ ipAdEtnBcastAddr(4) 广播地址最低位

└─ ipAdEtnResamMaskSize(5) 重装配的最大数

MIB-2 IP组

## ipRouteEntry(1)

## 说明

→	<b>ipRouteDest (1)</b>	<b>目标IP地址 (索引对象)</b>
	<b>ipRouteIfIndex(2)</b>	对应接口表的索引
	<b>ipRouteMetric1(3)</b>	<b>基本的路由度量 (整数)</b>
	<b>ipRouteMetric2(4)</b>	另外一种路由度量
	<b>ipRouteMetric3(5)</b>	另外一种路由度量
	<b>ipRouteMetric4(6)</b>	另外一种路由度量
	<b>ipRouteNextHop(7)</b>	<b>转发的IP地址</b>
	<b>ipRouteType(8)</b>	路由类型:other(1),invalid(2),direct(3),indirect(4)
	<b>ipRouteProto(9)</b>	路由协议:RIP、OSPF、ICMP、BGP 等
	<b>ipRouteAge(10)</b>	自路由上次更新之后的秒数
	<b>ipRouteMask(11)</b>	目标地址的子网掩码
	<b>ipRouteMetric5(12)</b>	另外一种路由度量
	<b>ipRouteInfo(13)</b>	指向MIB中的其他地方定义的路由信息(对象标识符)



# MIB-2 功 能 组

- **ipRouteTable**列出了当前路由设备中的路由信息。
- 表中的一行对应于一个已知的路由，由目标IP地址**ipRouteDest**索引。
- 对于每一个路由，通向下一结点的本地接口由**ipRouteIfIndex**表示。
- 其中大部分对象是可读写的，可以通过Set操作增加新的路由。

# MIB-2 功 能 组

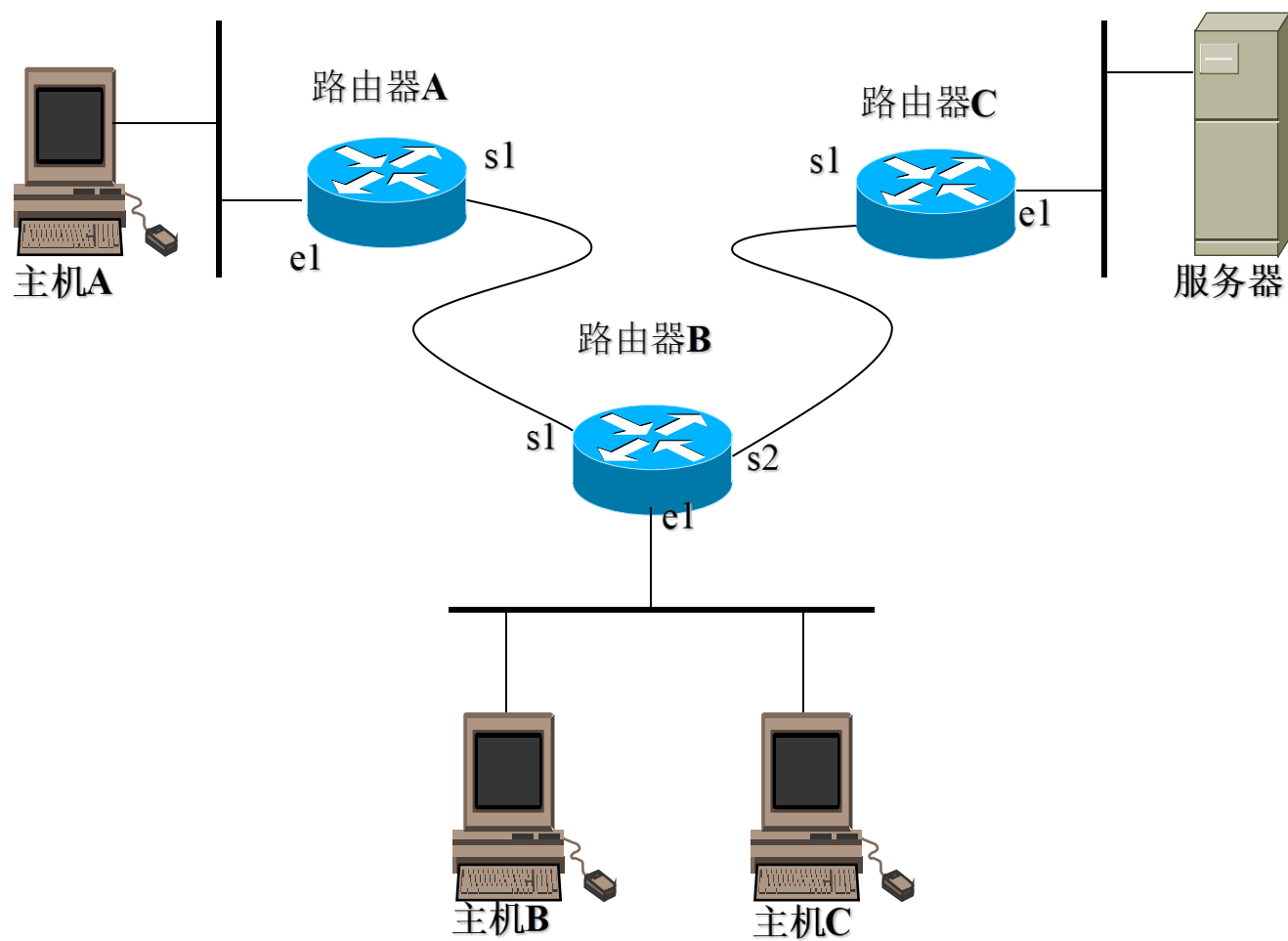
- 路由表中的信息可用于配置管理：

因为这个表中的对象是可读写的，所以可以用SNMP设置路由信息。

- 这个表也可以用于故障管理：

如果用户不能与远程主机建立连接，可检查路由表中的信息是否有错。

**比如，** 一个主机A要访问网络上的一台服务器，但是现在主机A无法连接到服务器。已知服务器、主机A都是正常的，如何查找故障？



## 故障排除步骤：

1. 查找主机A的网络配置，查出它的默认网关是路由器e1接口；
2. 查询路由器A的ipRouteTable表，根据ipRouteDest, ipRouteNextHop, ipRouteIfIndex来确定到达服务器所要经过的下一站。
3. 从查询结果可以得知，主机A的数据到达路由器A后，是通过接口s1传向下一个路由器B。
4. 继续检查路由器B的ipRouteTable表发现路由器B中没有指向服务器的路由器信息。
5. 检查路由器C的路由设置，正常。
6. 确认是路由设置不匹配造成的故障。



# IP地址转换表(ipNetToMediaTable)

- 提供了物理地址和IP地址的对应关系。
- 每个接口对应表中的一项。
- 这个表与地址转换组语义相同

## ipNetToMediaTable(22)

ipNetToMediaEntry(1)		说明
→	ipNetToMediaIfIndex(1)	该记录对应的接口索引号
—	ipNetToMediaPhysAddress(2)	物理地址
—	ipNetToMediaNetSddress(3)	网络地址 即IP地址
—	ipNetToMediaType(4)	IP到物理地址的映射的类型
1—其他； 2—无效； 3—动态； 4—静态		

- RFC 1354 (1992年7月) 提出了代替IP路由表的新标准，叫做IP转发表。原来的MIB-2中的IP路由表只由一项ipRouteDest索引，因此对一个目标只能定义一个路由。RFC 1354定义的转发表可以表示多路由的路由表。
- ipForward中的ipForwardNumber是一个只读的计量器，它记录IP转发表的项数。
- ipForwardTable的大部分对象与ipRouteTable的对象对应，有相同的语法和语义。

# MIB-2 功 能 组

## 5. ICMP (Internet Control Message Protocol) 组

- ICMP是IP的伴随协议。所有实现IP协议的结点都必须实现ICMP协议。它对于网络安全具有极其重要的意义。
- 用于在源IP主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息对于用户数据的传递起着重要的作用。比如，当遇到IP数据无法访问目标、IP路由器无法按当前的传输速率转发数据包等情况时，会自动向源发送ICMP消息。
- 发送的ICMP报文返回到发送原数据的设备。发送设备随后可根据ICMP报文确定发生错误的类型，并确定如何才能更好地重发失败的数据报。
- ICMP的功能是报告问题而不是纠正错误，纠正错误的任务由发送方完成。

# MIB-2 功 能 组

- **ICMP组**中的对象都是对各种类型的ICMP消息的接收和发送的统计结果，都是计数器。
- 在查找可能的路由问题时，ICMP组非常有用。主要用于性能管理和故障管理。

icmp(mi25)

- icmpInMsgs(1)
- icmpInErrors(2)
- icmpInDestUnreachs(3)
- icmpInTimeExcds(4)
- icmpInPramProbe(5)
- icmpInSrcQuenches(6)
- icmpInRedirects(7)
- icmpInEchos(8)
- icmpInEchoReps(9)
- icmpInTimestamps(10)
- icmpInTimestampReps(
- icmpInAddrMasks(12)
- icmpInAddrMaskReps(
- icmpOutMsgs(14)
- icmpOutErrors(15)
- icmpOutDestUnreachs(
- icmpOutTimeExcds(17)
- icmpOutPramProbe(18)
- icmpOutSrcQuenches(19
- icmpOutRedirects(20)
- icmpOutEchos(21)
- icmpOutEchoReps(22)
- icmpOutTimestamps(23
- icmpOutTimestampRep
- icmpOutAddrMasks(25
- icmpOutAddrMaskRep:

**ICMP组**

对象	语法	访问	功能描述
i c m p l n M s g s (1)	Counter	R0	接收的 i c m p 报文总数（以下为输入报文）
i c m p l n E r r o r s (2)	Counter	R0	出错的 i c m p 报文数
i c m p l n D e s t U n r e a c h s (3)	Counter	R0	目标不可送达型 i c m p 报文
i c m p l n T i m e E x c d s (4)	Counter	R0	超时型 i c m p 报文
i c m p l n P r a m P r o b e (5)	Counter	R0	有参数问题型 i c m p 报文
i c m p l n S r c Q u e n c h s (6)	Counter	R0	源抑制型 i c m p 报文
i c m p l n R e d i r e c t s (7)	Counter	R0	重定向型 i c m p 报文
i c m p l n E c h o s (8)	Counter	R0	回声请求型 i c m p 报文
i c m p l n E c h o R e p s (9)	Counter	R0	回声响应型 i c m p 报文
i c m p l n T i m e s t a m p s (10)	Counter	R0	时间戳请求型 i c m p 报文
i c m p l n T i m e s t a m p R e p s (11)	Counter	R0	时间戳响应型 i c m p 报文
i c m p l n A d d r M a s k s (12)	Counter	R0	地址掩码请求型 i c m p 报文
i c m p l n A d d r M a s k R e p s (13)	Counter	R0	地址掩码响应型 i c m p 报文

对象	语法	访问	功能描述
icmpOutMsgs (14)	Counter	R0	输出的icmp报文总数（以下为输出报文）
icmpOutErrors (15)	Counter	R0	出错的 icmp报文数
icmpOutDestUnreachs (16)	Counter	R0	目标不可送达型icmp报文
icmpOutTimeExcds (17)	Counter	R0	超时型icmp报文
icmpOutPramProbe (18)	Counter	R0	有参数问题型icmp报文
icmpOutSrcQuenchs (19)	Counter	R0	源抑制型icmp报文
icmpOutRedirects (20)	Counter	R0	重定向型icmp报文
icmpOutEchos (21)	Counter	R0	回声请求型icmp报文
icmpOutEchoReps (22)	Counter	R0	回声响应型icmp报文
icmpOutTimestamps (23)	Counter	R0	时间戳请求型icmp报文
icmpOutTimestampReps (24)	Counter	R0	时间戳响应型icmp报文
icmpOutAddrMasks (25)	Counter	R0	地址掩码请求型icmp报文
icmpOutAddrMaskReps (26)	Counter	R0	地址掩码响应型icmp报文

# MIB-2 功 能 组

## 6. TCP组

- TCP组包含与面向连接的传输控制协议（TCP）有关的被管对象。TCP组的对象主要用于：对流量控制，确保数据在端到端之间可靠的传送。比如，丢弃重传和网络拥挤等问题的解决。
- TCP组的组成：
  - 14个TCP的系统对象
  - 1个记录当前TCP连接的表



# MIB-2 功能组

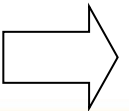
tcp(mib6)

- tcpRtoAlgorithm(1)
- tcpRtoMin(2)
- tcpRtoMax(3)
- tcpMaxConn(4)
- tcpActiveOpens(5)
- tcpPassiveOpens(6)
- tcpAttemptFails(7)
- tcpEstabResets(8)
- tcpCurrEstab(9)
- tcpInSegs(10)
- tcpOutSegs(11)
- tcpRetransSegs(12)
- tcpConnTable(13)
- tcpInErrors(14)
- tcpOutRests(15)

tcpConnTable(tcp 13)

- └─ tcpConnEntry(1)
  - tcpConnState(1)
  - — tcpConnLocalAddress(2)
  - — tcpConnLocalPort(3)
  - — tcpConnRemAddress(4)
  - — tcpConnRemPort(5)

**TCP组**



TCP组 对象	语法	访问	功能描述
tcpRtoAlgor i thm(1)	INTEGER	R0	重传时间算法
tcpRtoMin(2)	INTEGER	R0	重传时间最小值
tcpRtoMax(3)	INTEGER	R0	重传时间最大值
tcpMaxConn(4)	INTEGER	R0	可建立的最大连接数
tcpAct i veOpens(5)	Counter	R0	主动打开的连接数
tcpPass i veOpens(6)	Counter	R0	被动打开的连接数
tcpAttemptFai l s(7)	Counter	R0	连接建立失败数
tcpEstabResets(8)	Counter	R0	连接复位数
tcpCurrEstab(9)	Gauge	R0	状态为established或closeWait的连接数
tcpInSegs(10)	Counter	R0	接收的TCP段总数
tcpOutSegs(11)	Counter	R0	发送的TCP段总数
tcpRetransSegs(12)	Counter	R0	重传的TCP段总数
tcpConnTable(13)	SEQUENCE OF	NA	连接表
tcpInErrors(14)	Counter	R0	接收的出错TCP段数
tcpOutRests(15)	Counter	R0	发出的含RST标志的段数

# MIB-2 功 能 组

- TCP组的前3项与重传有关。
- 一个TCP实体发送数据段后就等待应答并开始计时。如果超时时没有得到应答，就认为数据段丢失了，因此要重新发送。
- 该组对象tcpRtoAlgorithm说明计算重传时间的算法，其值可取：
  - 👉 other (1) : 不属于以下3种类型的其他算法；
  - 👉 constant (2) : 重传超时值为常数；
  - 👉 rsre (3) : 根据通信情况动态地计算超时值，即把估计的周转时间乘一个倍数。这种算法是美国军用TCP标准MIL-STD-1778定义的；
  - 👉 vanj (4) : 这是由杰克布森发明的一种动态算法。

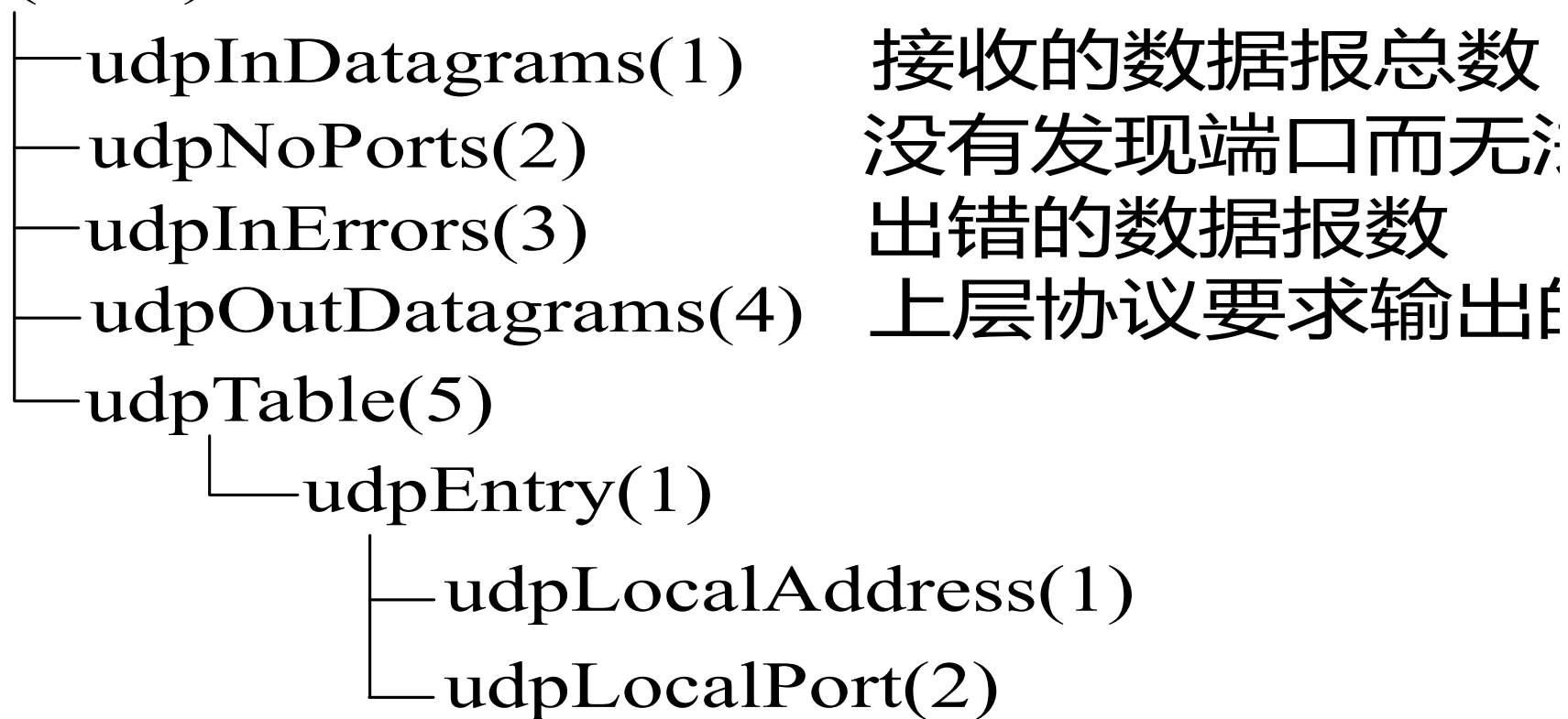
# MIB-2 功 能 组

## 7. UDP组

- UDP组包含与无连接传输协议UDP有关的被管对象。UDP组类似于TCP组，但没有TCP严格。UDP不保证可靠性连接，因此UDP不提供有关连接的信息。
- UDP组提供了关于UDP数据报和本地接收端点的详细信息。UDP表相当简单：只有本地地址和本地端口两项。

# MIB-2 功 能 组

udp(mib7)



UDP组

# MIB-2 功能组

## 8. EGP组(External Gateway Protocol)

- 自治系统 (AS, autonomous system) 是指在一个管理机构管辖下的所有IP网络和路由器的全体。在一个自治系统中的所有路由器必须：(1) 相互连接；(2) 运行相同的路由协议；(3) 同时分配同一个自治系统编号。
- EGP是用来链接两个自治系统的路由协议。在自治系统之间进行EGP通信的设备称为EGP邻居。
- EGP组提供了关于EGP路由器发送和接收的EGP报文的信息，以及关于EGP邻居的详细信息等。

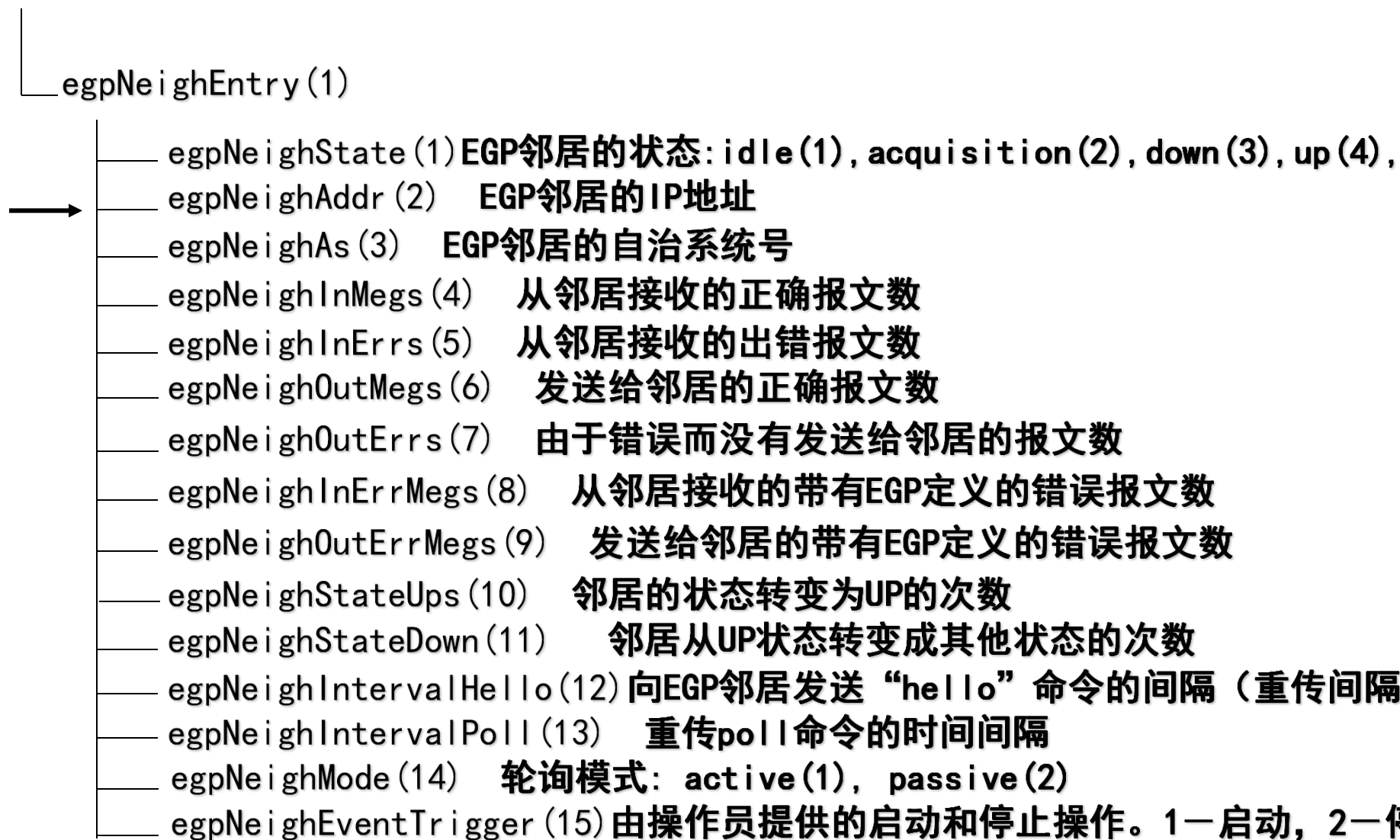
# MIB-2 功能组

## egp (mib-2 8)

egpInMeps (1)	接收的正确的EGP报文总数
egpInErrors (2)	接收的错误的EGP报文总数
egpOutMeps (3)	发送的正确的EGP报文总数
egpOutErrors (4)	因错误未发送的EGP报文总数
egpNeighborTable (5)	邻居表
egpAs (6)	本地自治系统的编号

# MIB-2 功能组

## egpNeighTable (egp 5)





# MIB-2 功 能 组

在EGP邻居表中，邻居状态`egpNeighborState`可取的值有：

- `idle(1)`； ---空闲
- `acquisition(2)`； ---接收
- `down(3)`；
- `up(4)`；
- `cease(5)`。 ---停止、中止

轮询模式`egpNeighborMode`可取的值：`active(1)` `passive(2)`

# MIB-2 功 能 组

## 9. 传输组(Transmission Group)

- 传输组主要是针对各种传输介质提供详细的管理信息。
- 接口组包含各种接口通用的信息，传输组提供与子网类型有关的接口专用信息。

# 本章小节

- 在现代网络管理模型中，管理信息库（MIB）是网络管理系统的核心。
  - 👉 网络管理员在对网络进行监视时就是读取MIB中有关对象的数据
  - 👉 在对网络进行控制时，只需更新MIB中相关对象所对应的数据即可。
- MIB树中的被管对象是利用管理信息结构（SMI）来定义。
  - 👉 管理信息库第2版，即MIB-2包含11个功能组，共171个对象。本章介绍了9个组的功能描述和每个功能组中的被管对象。
  - 👉 利用功能组中的这些对象来实现网络管理的配置管理、故障管理、性能管理、计费管理和安全管理。

# MIB-2的局限性

- MIB-2的简单性有利于网络管理系统的实现，增强互操作性。

MIB只存储**两种简单的数据类型**：标量和二维数组。SNMP协议仅仅能够**检索标量对象实例和一个表中的单个对象实例**。这样，对管理系统功能的实现带来了一定的限制。如：

- ☞ 不支持基于对象值或类型进行复杂查询；
- ☞ 安全机制不够完善
- ☞ 只知道某个TCP实体建立连接的数量，而不知道在TCP连接上的通信量。

- MIB-2的简单性有利于网络管理系统的实现，但是也对管理系统的功能实现带来了一定的限制，所以必须要对MIB-2进行改进和扩充。主要在**RMON规范**中进行实现。