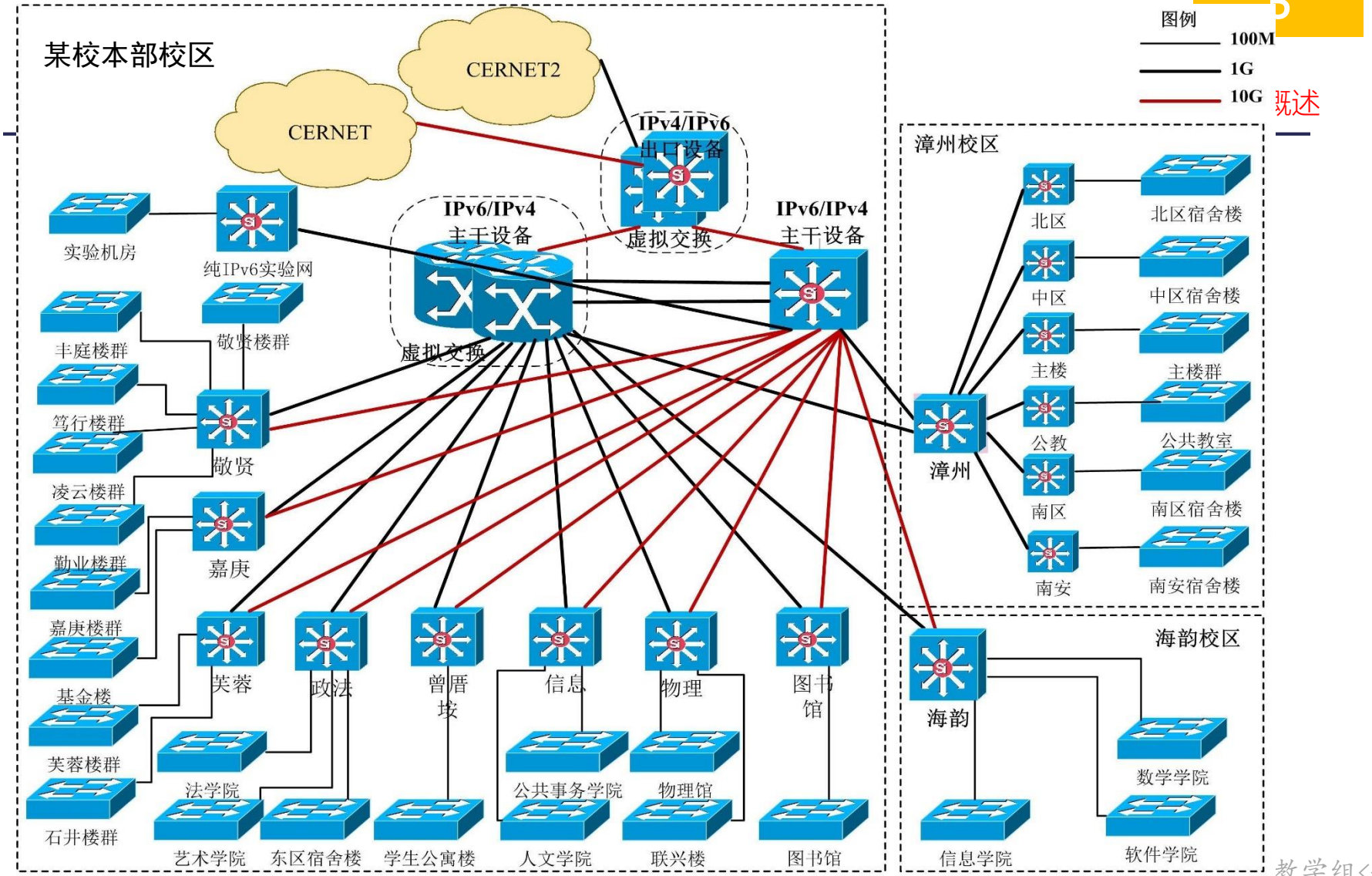
# 网络运维管理

## 第7讲：网络监控管理

河南中医药大学信息技术学院
《网络运维管理》课程教学组

# 本章目录

# 一、认识网络管理


某校本部校区
CERNET2
CERNET
IPv4/IPv6
出口设备
虚拟交换
IPv6/IPv4
主干设备
虚拟交换
IPv6/IPv4
主干设备
图例
100M
1G
10G
实验机房
纯IPv6实验网
丰庭楼群
敬贤楼群
笃行楼群
凌云楼群
敬贤
勤业楼群
嘉庚
嘉庚楼群
基金楼
芙蓉
芙蓉楼群
法学院
石井楼群
艺术学院
东区宿舍楼
政法
曾厝垵
学生公寓楼
信息
公共事务学院
人文学院
物理
物理馆
联兴楼
图书馆
图书馆
漳州校区
北区
北区宿舍楼
中区
中区宿舍楼
主楼
主楼群
公教
公共教室
南区
南区宿舍楼
南安
南安宿舍楼
漳州
海韵校区
海韵
数学学院
信息学院
软件学院

# 1、认识网络管理

- 每一个网络管理和运维人员都希望能够及时准确地掌握网络的运行状况，例如服务器是否宕机、服务器CPU的使用率有多大、网络中各种协议的流量情况等等，从而实现网络安全稳定地运行。
- 早期的计算机网络规模小，结构简单，网络管理活动也相对简单，但随着计算机网络技术的迅速发展，网络规模日益庞大，结构也越来越复杂。简单、粗陋的管理方式已经不再适应现代的计算机网络，网络管理必须向高度集中和高度智能化的方向发展。

# 1、认识网络管理

- 网络管理简称为网管
  - 网络管理包括对硬件、软件和人力的使用、综合与协调，以便对网络资源进行监视、测试、配置、分析、评价和控制，这样就能以合理的价格满足网络的一些需求，如实时运行性能，服务质量等。
  - 网络管理并不是指对网络进行行政上的管理，通常分为五大功能。
    - 故障管理：故障检测、隔离和纠正。
    - 配置管理：初始化网络、并配置网络。
    - 计费管理：记录网络资源的使用。
    - 性能管理：估价系统资源的运行状况及通信效率等。
    - 网络安全管理：对授权机制、访问控制、加密和加密关键字的管理。

# 1、认识网络管理

- 网络管理模型中的主要构件 之一：
  - 管理站：
    - 管理站是整个网络管理系统的核心，由网络管理员直接操作和控制。
    - 所有向被管设备发送的命令都是从管理站发出的，
    - 管理站（硬件）或管理程序（软件）都可称为管理者 (manager)。Manager 不是指人，而是指机器或软件。
    - 大型网络往往实行多级管理，因而有多个管理者，而一个管理者一般只管理本地网络的设备。

# 1、认识网络管理

- 网络管理模型中的主要构件 之二：
  - 被管设备（被管对象）：
    - 在被管网络中有很多的被管设备：主机、路由器、交换机、打印机等；
    - 真正被监管的，是被管对象！每一个被管设备中可能有多个被管对象，可以是被管设备中的某个硬件（例如网卡、CPU、内存等），也可以是一些软件（例如，操作系统、路由协议）的配置参数的集合。
    - 在被管设备中也会有一些不能被管的对象（不是什么都能被管理！）
    - 对象命名树，在树上的，属于可管理对象，不在树上的，不能管理。

# 1、认识网络管理

- 网络管理模型中的主要构件 之三：

  问：管理站和被管设备之间如何通信？

  - 管理程序和代理程序

    - 管理程序：管理站中的关键构件是管理程序，管理程序在运行时就成为管理进程。
    - 代理程序：在每一个被管设备中都要运行一个程序以便和管理站中的管理程序进行通信。这些运行着的程序叫做网络管理代理程序，简称为代理。代理程序在管理程序的命令和控制下在被管设备上采取本地的行动。

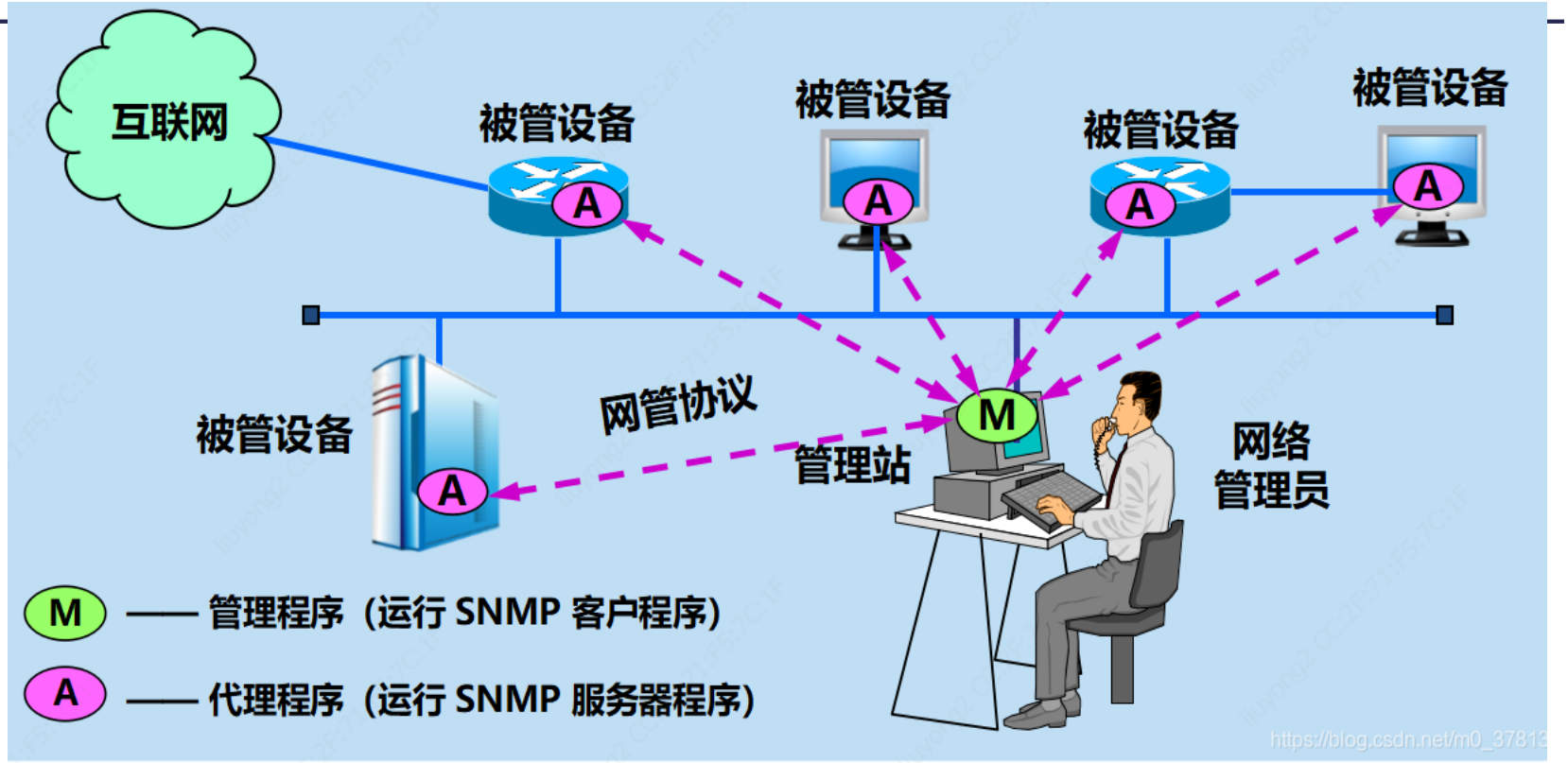
# 1、认识网络管理

- 网络管理模型中的主要构件 之四：

  问：如何使管理站（管理程序）、被管设备（代理程序）之间协调工作？

  - 网络管理协议
    - 网络管理协议简称为网管协议。
    - 网络管理协议是管理程序和代理程序之间进行通信的规则。
    - 网络管理员利用网络管理协议，通过管理站对网络中的被管设备进行管理。
    - 需要注意的是，网管协议本身不管理网络。

# ➢ 总结：网络管理的一般模型

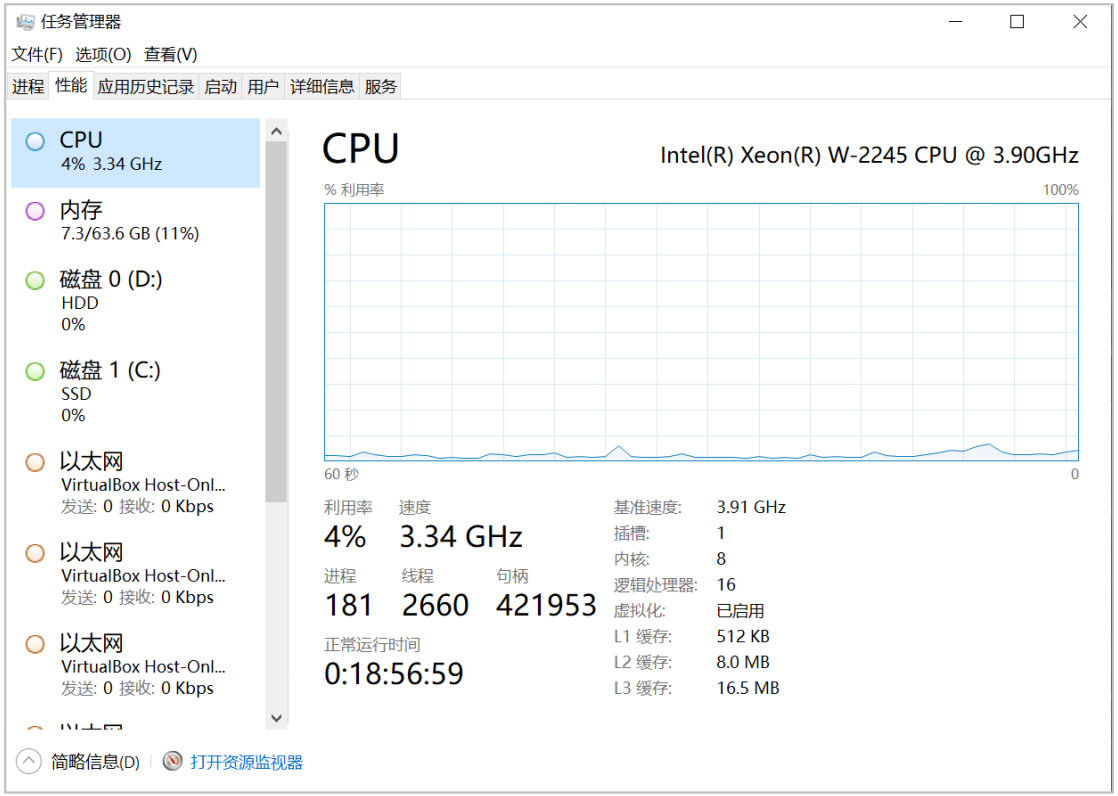



管理站、管理程序、被管设备、代理程序、网管协议

# 二、SNMP概述

# 2、SNMP概述

- 操作：
  - 查看Windows中的任务管理器
    - CPU利用率
    - 内存使用情况
    - 网卡通信情况
    - ……



# 2、SNMP概述

- 思考：
  - 能否设计一种体系，使管理员能够远程获取（监控）到网络中各种设备（服务器、路由器、交换机、防火墙等）的某些状态信息（被管对象）？
  - 例如，获取到网络中某台服务器的CPU的利用率、硬盘的使用量、网络接口入分组流量和出分组流量、网络接口的数目等等，则可以大大提高管理效果、降低管理成本。

# 2、SNMP概述

- 思考：这个体系应包含哪些功能?

① 包含管理程序（功能）和代理程序（功能），从而实现对被管对象的各种参数信息的读取和修改；（完成管理动作）

例：在被管对象上运行代理程序功能，监听来自管理站的请求，一旦发现，就返回管理站所需的信息（例如，将被管对象的CPU的使用率的值返回给管理站）

# 2、SNMP概述

- 思考：这个体系应包含哪些功能?

② 包含一套对被管对象的命名、数据值的范围、长度、编码等定义的规则，从而确保网络管理数据的语法和语义无二义性。（建立数据规则）

例：对于路由器的接口（被管对象），制定其命名规则

# 2、SNMP概述

- 思考：这个体系应包含哪些功能?

③ 包含一套明确的、被管对象各变量的标记说明。（明确变量标记）

例如，Linux操作系统（也是一个被管对象），其在网络管理体系中的标记名称是什么？即变量的具体实例。

# 2、SNMP概述

- SNMP的定义：
  - SNMP（Simple Network Management Protocol），为简单网络管理协议。
  - 从狭义上讲，是一种专门用于网络管理软件和网络设备之间通信的协议；从广义上讲，是一组为实现网络的自动化管理任务而制定的一系列通用标准，包括管理信息的表示与命名、通信协议等内容。
  - 一种应用层协议，使用TCP/IP协议族对互联网上的设备进行管理的框架，它提供一组基本的操作，用来监控和管理网络。

# 2、SNMP概述

- SNMP的定义：
  - SNMP（Simple Network Management Protocol），为简单网络管理协议。
  - 从狭义上讲，是一种专门用于网络管理软件和网络设备之间通信的协议；从广义上讲，是一组为实现网络的自动化管理任务而制定的一系列通用标准，包括管理信息的表示与命名、通信协议等内容。
  - 一种应用层协议，使用TCP/IP协议族对互联网上的设备进行管理的框架，它提供一组基本的操作，用来监控和管理网络。

# 2、SNMP概述



- 关于网络管理有一个基本原理，就是：

若要管理某个对象，就必然会给该对象添加一些软件或硬件，但这种“添加”必须对原有对象影响尽量小些。

- SNMP正是按照这样的基本原理来设计的。

# 2、SNMP概述

- SNMP工作的基本思路：
  - SNMP中的管理程序和代理程序按照C/S方式工作。
  - 管理程序运行SNMP客户程序，而代理程序运行SNMP服务器程序。
  - 在被管对象上运行的SNMP服务器程序不停地监听来自管理站的SNMP客户程序的请求，一旦发现，就立即返回管理站所需的信息，或执行某个动作（例如，把某个参数的设置进行更新）。
  - 在网管系统中往往是一个（或少数几个）客户程序与很多的服务器程序进行交互。

# 2、SNMP概述

- SNMP的网络管理由三个部分组成
  - 管理信息结构SMI (Structure of Management Information)

    定义对象命名和定义对象类型的通用规则，以及把对象和对象的值进行编码的规则。
  - 管理信息库MIB (Management Information Base)

    在被管理的实体中创建命名对象，并按照SMI的规则规定了其类型。
  - SNMP本身

    实现对被管对象信息（变量名及其值）的操作，即负责读取和改变这些数值。

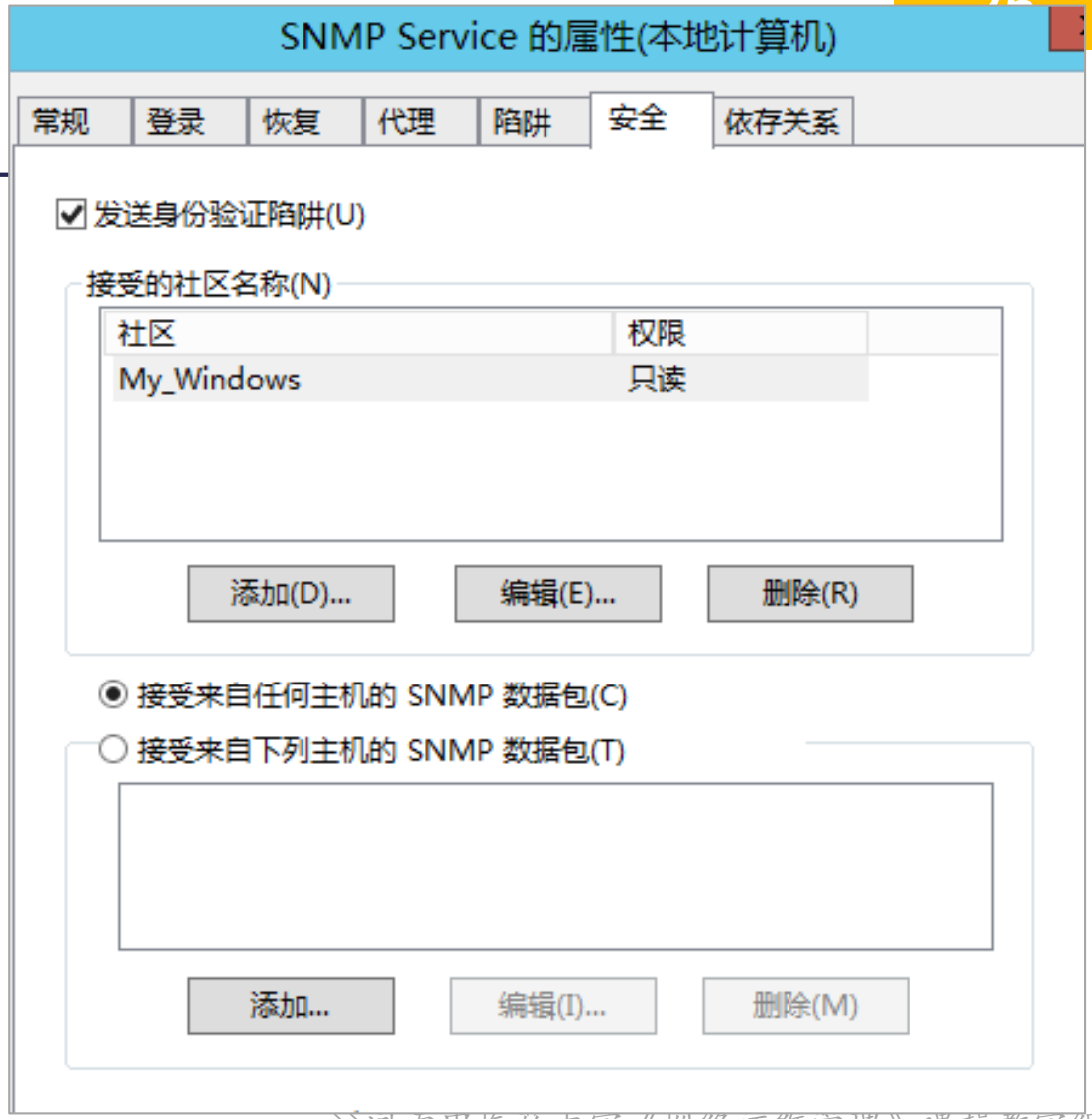
# 2、SNMP概述

- SNMP的版本
  - 目前，SNMP有三个版本，SNMPv1、SNMPv2、SNMPv3。
  - v1和v2没有太大差距，SNMPv2是增强版本，包含了其他协议操作。SNMPv3则包含更多安全和远程配置。
  - SNMPv2由SNMPv1演化而来，v1中的操作（例如get、get next、set等）同样适用于v2，只是v2添加和增强了有关操作。

# 2、SNMP概述

- 设置SNMP共同体名的意义
  - 在进行SNMP配置时，如果使用的是SNMPv1和SNMPv2版本，则需要在代理（被管设备）一侧设置其共同体（community）名称。
  - 共同体名称类似一个字符串，它是一种安全机制，是代理和管理站之间的认证。SNMPv1和V2的安全机制很简单，只是验证共同体名。即，当管理站对被管设备（代理）进行访问时，在相关命令中通常需要输入代理的共同体名，作为一种认证机制。
  - SNMPv3则是采用更复杂的安全机制。

- 在Windows中配置共同体（又称为社区）名称



# 在Linux中配置共同体（community）名称

（1）安装SNMP服务组件

```
# yum –y install net-snmp-libs net-snmp net-snmp-utils net-snmp-devel
net-snmp-perl
```

（2）编辑打开snmpd.conf文件，配置共同体名称

```
# vi /etc/snmp/snmpd.conf
```

```
# First, map the community name "public" into a "security name"

#         sec.name  source          community
com2sec notConfigUser  default       public
```

```
# First, map the community name "public" into a "security name"

#         sec.name  source          community
com2sec notConfigUser  192.168.31.100   My_Cacti
```

# 三、SNMP—— 管理信息结构 SMI

# 3、管理信息结构 SMI

- 何谓“管理信息”
  - 网络管理活动中，管理信息主要是管理工作站感兴趣的、任何与被管理设备有关的信息。即前面所说的被管对象的集合。
  - 这些信息可以是和网络设备运行状态有关的信息，如设备网络接口的工作状态；被管理设备的系统软、硬件资源，如设备CPU利用率、软件版本等。因此，一切对于网络管理有意义的、来自被管理设备的资源、状态信息，都可以是管理信息。
  - 管理信息可以用整数或文字（字符串）表示。例如用整数表示网络接口某一时刻接收的字节总数，或者表示一种设备的运行状态。

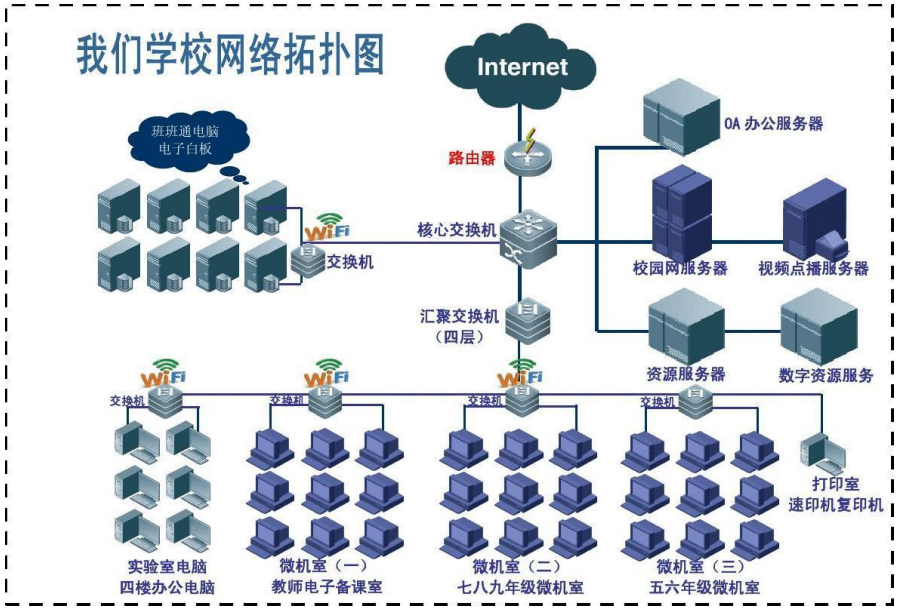
# 3、管理信息结构 SMI

- SMI的功能有三个
  - 规定被管对象应该怎样命名（不是命名结果，而是指命名规则）；
  - 规定用来存储被管对象的数据类型有哪些；
  - 规定在网络上传送的管理数据应如何编码；

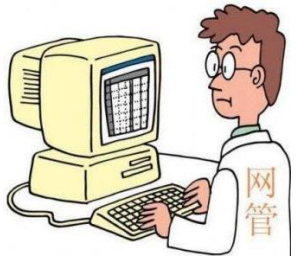
# 3、管理信息结构 SMI

## （1）被管对象的命名

这里的“CPU的使用情况”就是一个被管对象，在SNMP中，该如何去命名它？




我要查看位于网络中心的视频点播服务器的CPU的使用情况



# 3、管理信息结构 SMI

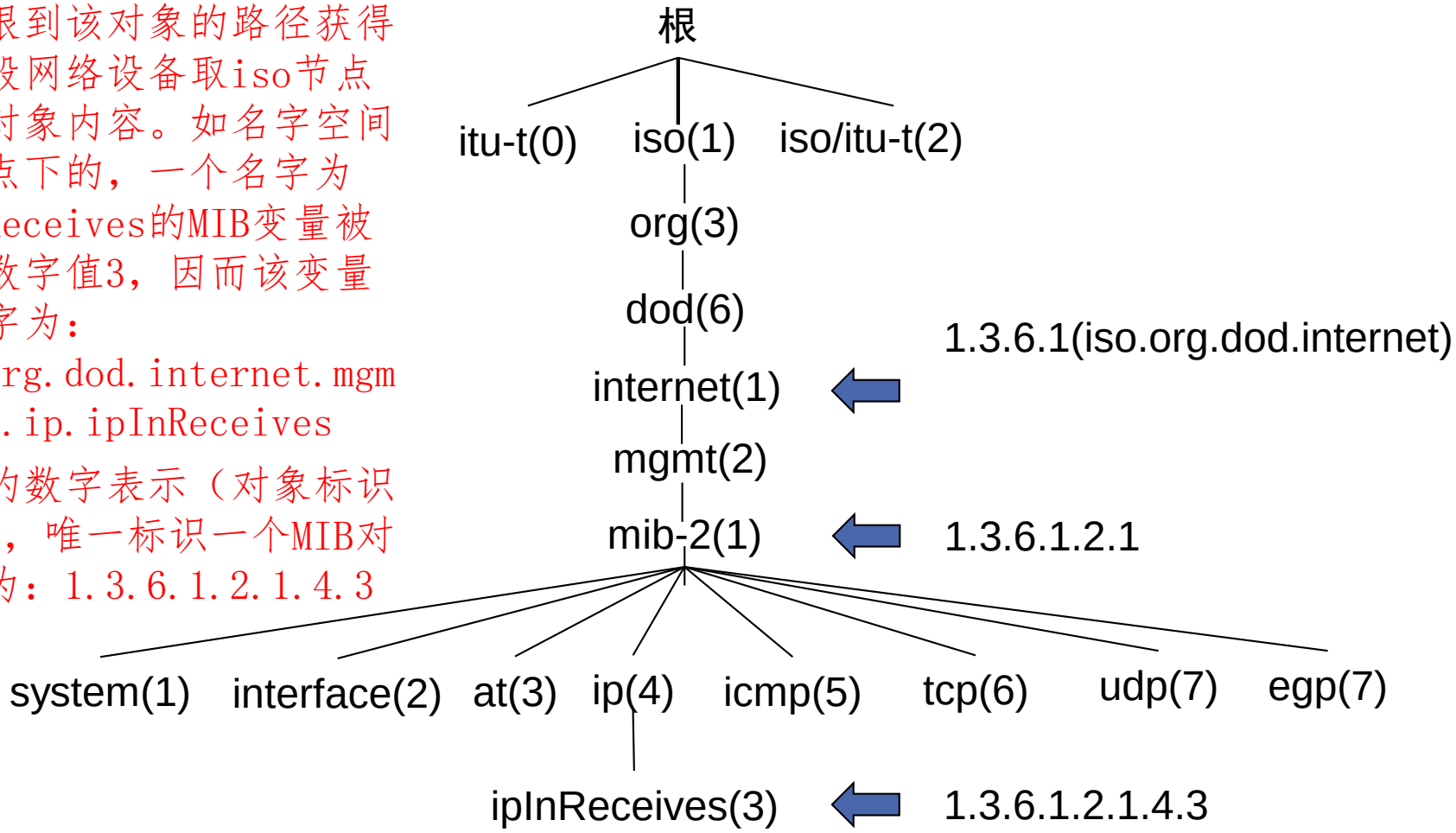
## （1）被管对象的命名

- 根据SMI的规定，被管对象（变量）使用的名字取自ISO和ITU（两个著名的标准制定组织）管理的对象标识符（object identifier，简称OID）名字空间。
- 这个名字空间是一种分级树的结构，而且是一棵“倒置”的树，最上面是树根，根没有名字，它的下面有3个顶级对象，即itu-t、iso、iso/itu-t，它们的标号分别是0,1,2。再下一级的对象ID分别由相关组织分配。
- SMI规定，所有的被管对象都必须处在对象命名树上。

一个特定对象的标识符可通过由根到该对象的路径获得。一般网络设备取iso节点下的对象内容。如名字空间ip结点下的，一个名字为ipInReceives的MIB变量被指派数字值3，因而该变量的名字为：

iso.org.dod.internet.mgmt.mib.ip.ipInReceives

相应的数字表示（对象标识符OID，唯一标识一个MIB对象）为：1.3.6.1.2.1.4.3



# 3、管理信息结构 SMI

## （2）被管对象的数据类型

- 任何数据都具有两种重要的属性：值（value）和类型（type）
  - 值：某个值集合中的一个元素；
  - 类型：值集合的名字
- SMI采用ANS.1（抽象语法标记）来定义数据类型，把数据类型分类两大类：
  - 简单类型
  - 结构化类型

# 3、管理信息结构 SMI

## （2）被管对象的数据类型

几种最主要的简单类型

| 类　型 | 大　小 | 说　明 |
|---|---|---|
| INTEGER | 4 字节 | 在$-2^{31}$到$2^{31}-1$之间的整数 |
| Interger32 | 4 字节 | 和 INTEGER 相同 |
| Unsigned32 | 4 字节 | 在 0 到$2^{32}-1$之间的无符号数 |
| OCTET STRING | 可变 | 不超过 65535 字节长的字节串 |
| OBJECT IDENTIFIER | 可变 | 对象标识符 |
| IPAddress | 4 字节 | 由 4 个整数组成的 IP 地址 |
| Counter32 | 4 字节 | 可从 0 增加到$2^{32}$的整数；当它到达最大值时就返回到 0 |
| TimeTicks | 4 字节 | 记录时间的计数值，以 1/100 秒为单位 |
| BITS | — | 比特串 |
| Opaque | 可变 | 不解释的串 |

# 3、管理信息结构 SMI

## （2）被管对象的数据类型

ASN.1 部分数据类型

| 分类 | 数据类型名称 | 含义 |
| --- | --- | --- |
| 结构化类型 | SEQUENCE | 由多个数据类型按序组成的值 |
| | SEQUENCE OF | 由同一数据类型按序组成的值 |
| | CHOICE | 可以从多个数据类型中选择一个 |
| | ANY | 任何数据类型 |

# 3、管理信息结构 SMI

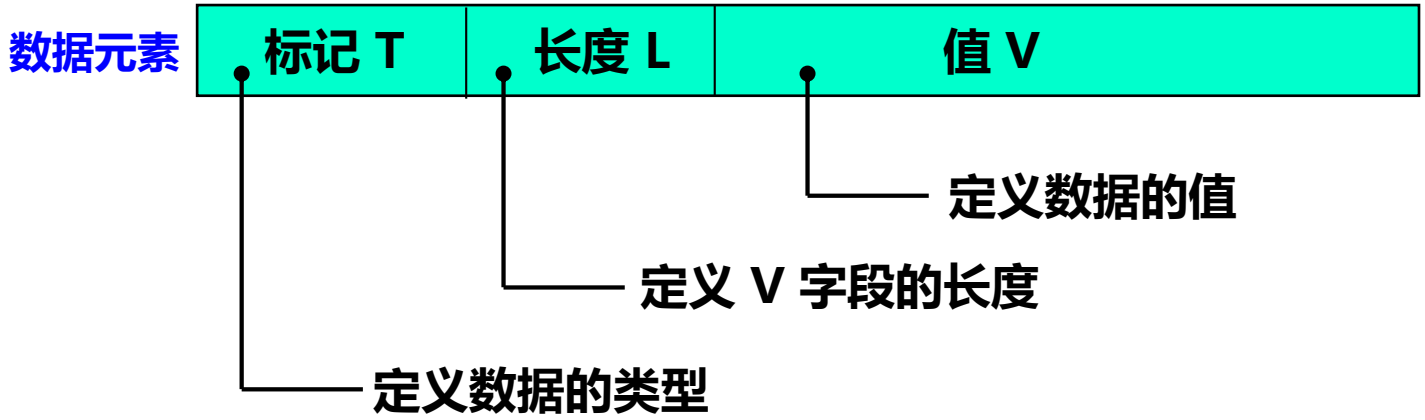
## （3）编码方法（略）

- 基本编码规则 BER
  - ISO 在制订 ASN.1 语言的同时也为它定义了一种标准的编码方案，即基本编码规则 BER (Basic Encoding Rule)。
  - BER 指明了每种数据类型中每个数据的值的表示。
  - 发送端用 BER 编码，可将用 ASN.1 所表述的报文转换成唯一的比特序列。
  - 接收端用 BER 进行解码，得到该比特序列所表示的 ASN.1 报文

# 3、管理信息结构 SMI

## （3）编码方法（略）

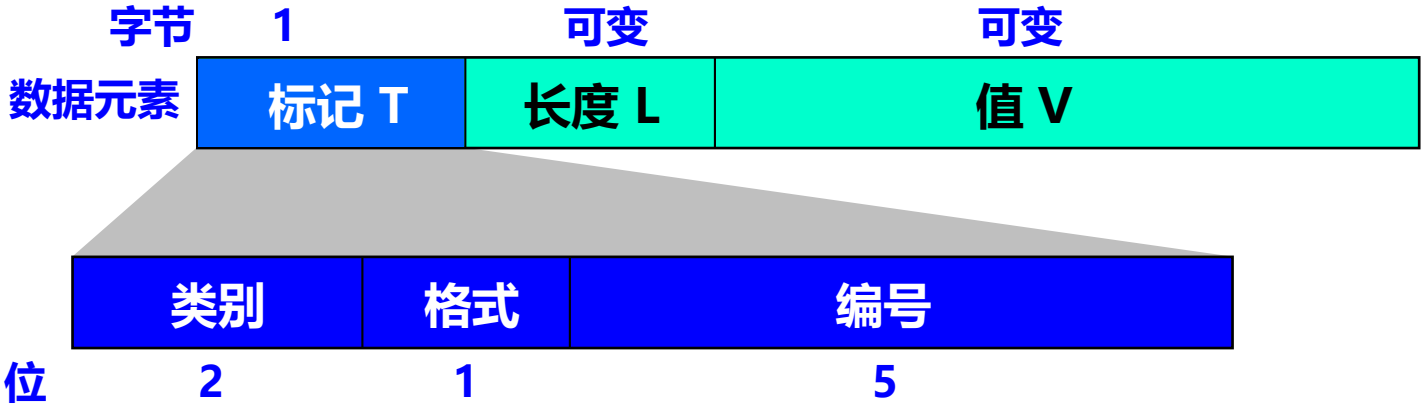
- 基本编码规则 BER
  - 用 TLV 方法进行编码



# 3、管理信息结构 SMI

## （3）编码方法（略）

- 用 TLV 方法进行编码： T 字段定义数据的类型



# 3、管理信息结构 SMI

## （3）编码方法（略）

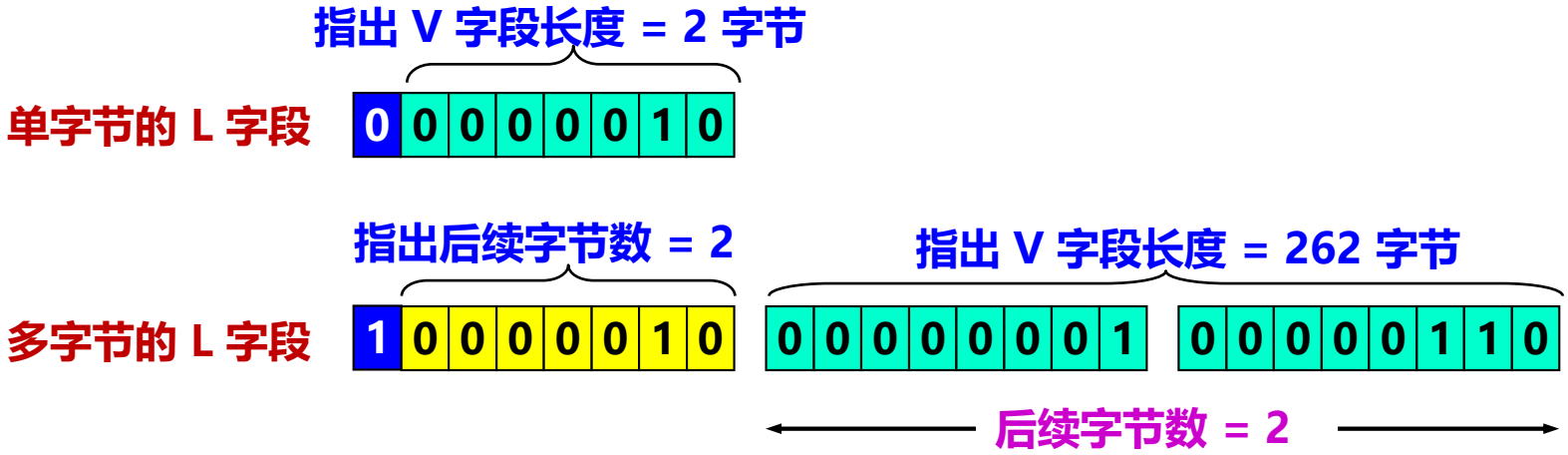
- 用 TLV 方法进行编码： T 字段定义数据的类型

| 数据类型 | 类别 | 格式 | 编号 | T字段（二进制） | T字段（十六进制） | |
|---|---|---|---|---|---|---|
| INTEGER | 00 | 0 | 00010 | 00000010 | 02 | ← 举例 |
| OCTET STRING | 00 | 0 | 00100 | 00000100 | 04 | |
| OBJECT IDENTIFIER | 00 | 0 | 00110 | 00000110 | 06 | |
| NULL | 00 | 0 | 00101 | 00000101 | 05 | |
| Sequence, sequence of | 00 | 1 | 10000 | 00110000 | 30 | |
| IPAddress | 01 | 0 | 00000 | 01000000 | 40 | ← 举例 |
| Counter | 01 | 0 | 00001 | 01000001 | 41 | |
| Gauge | 01 | 0 | 00010 | 01000010 | 42 | |
| TimeTicks | 01 | 0 | 00011 | 01000011 | 43 | |
| Opaque | 01 | 0 | 00100 | 01000100 | 44 | |

# 3、管理信息结构 SMI

## （3）编码方法（略）

- 用 TLV 方法进行编码： L 字段定义 V 字段的长度



# 3、管理信息结构 SMI

## （3）编码方法（略）

- 用 TLV 方法进行编码： V 字段定义数据的值

| 类 型 | 大 小 | 说 明 |
|---|---|---|
| INTEGER | 4 字节 | 在 $-2^{31}$ 到 $2^{31}-1$ 之间的整数 |
| Interger32 | 4 字节 | 和 INTEGER 相同 |
| Unsigned32 | 4 字节 | 在 0 到 $2^{32}-1$ 之间的无符号数 |
| OCTET STRING | 可变 | 不超过 65535 字节长的字节串 |
| OBJECT IDENTIFIER | 可变 | 对象标识符 |
| IPAddress | 4 字节 | 由 4 个整数组成的IP地址 |
| Counter32 | 4 字节 | 可从 0 增加到 $2^{32}$ 的整数；当它到达最大值时就返回到 0 |
| TimeTicks | 4 字节 | 记录时间的计数值，以 1/100 秒为单位 |
| BITS | — | 比特串 |
| Opaque | 可变 | 不解释的串 |

# 3、管理信息结构 SMI

## （3）编码方法（略）

- 用 TLV 方法进行编码： 举例
  - INTEGER 15，其 T 字段是 02， INTEGER 类型要用 4 字节编码

| T | L | V |
|---|---|---|
| 02 | 04 | 00 00 00 0F |

  - IPAddress 192.1.2.3，其 T 字段是 40，V字段需要4字节表示

| T | L | V |
|---|---|---|
| 40 | 04 | C0 01 02 03 |

# 四、SNMP—— 管理信息库 MIB

# 4、管理信息库 MIB

## （1）MIB中的对象才是SNMP所能够管理的

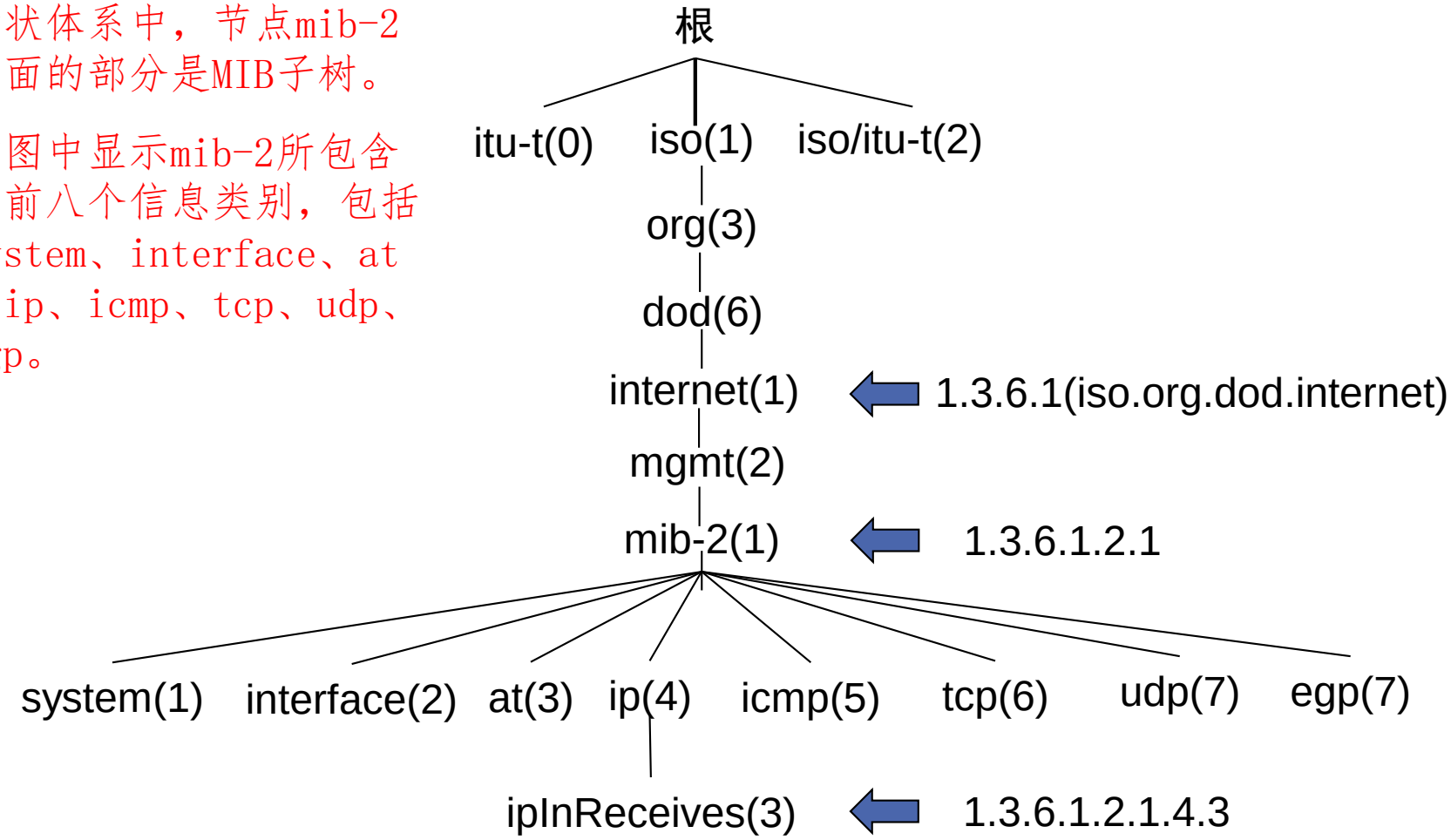
- 任何一个被管理的资源都表示成一个对象，称为被管对象。
- MIB（Management Information Base）是被管理对象的集合。它定义了被管理对象的一系列属性：对象的名称、对象的访问权限和对象的数据类型等。
- 这种定义是依据SMI制定的规则。
- 管理程序使用 MIB 中这些信息的值对网络进行管理（如读取或重新设置这些值）。
- 只有在 MIB 中的对象才是 SNMP 所能够管理的。

# 4、管理信息库 MIB

## (2) MIB中的信息类别和变量

- MIB中所有的被管理对象按照所表示的管理信息的不同，被分为不同组，代表不同的类别。
  - 例如，MIB-II（第2代MIB）中，所有和系统有关对象分在system组中；所有和IP有关对象分在ip组中，每个对象组分配一个OID节点。
- 定义在组中的每个对象，其对象标识符（即OID值）均以组节点的对象标识符（OID值）作为前缀。

- 在对象标识符（OID）的树状体系中，节点mib-2下面的部分是MIB子树。
- 右图中显示mib-2所包含的前八个信息类别，包括system、interface、at、ip、icmp、tcp、udp、egp。



# MIB类别所包含的相关信息

| MIB类别 | 包含的相关信息 | MIB类别 | 包含的相关信息 |
|---|---|---|---|
| system | 被管理对象（如主机、路由器等设备）系统的总体信息 | interface | 各个网络接口的相关信息 |
| at | 地址转换（如：ARP映射）的相关信息 | ip | IP协议的实现和运行相关信息 |
| icmp | ICMP协议的实现和运行相关信息 | tcp | TCP协议的实现和运行相关信息 |
| udp | UDP协议的实现和运行相关信息 | egp | 外部网关协议实现和运行相关信息 |
| snmp | 描述了SNMP协议自身的一些信息 | transmission | 根据网络接口，描述相关的管理信息 |
| dot1Bridge | 网络中网桥的相关管理信息 | host | 主机自身上运行的相关信息 |

# MIB变量及类别、含义对比表

| MIB对象 | 类别 | 含义 |
| --- | --- | --- |
| sysUpTime | system | 系统开启时间 |
| ifNumber | interface | 网络接口数 |
| ifMtu | interface | 某特定接口的MTU值 |
| icmpInEchos | icmp | 接受ICMP发送请求数目 |
| tcpInSegs | tcp | 已收到的TCP报文段数目 |

例如，表示某设备的系统开启时间这个被管对象，其MIB变量的名字是：
Iso.org.dod.internet.mgmt.mib-2.system.sysUpTime

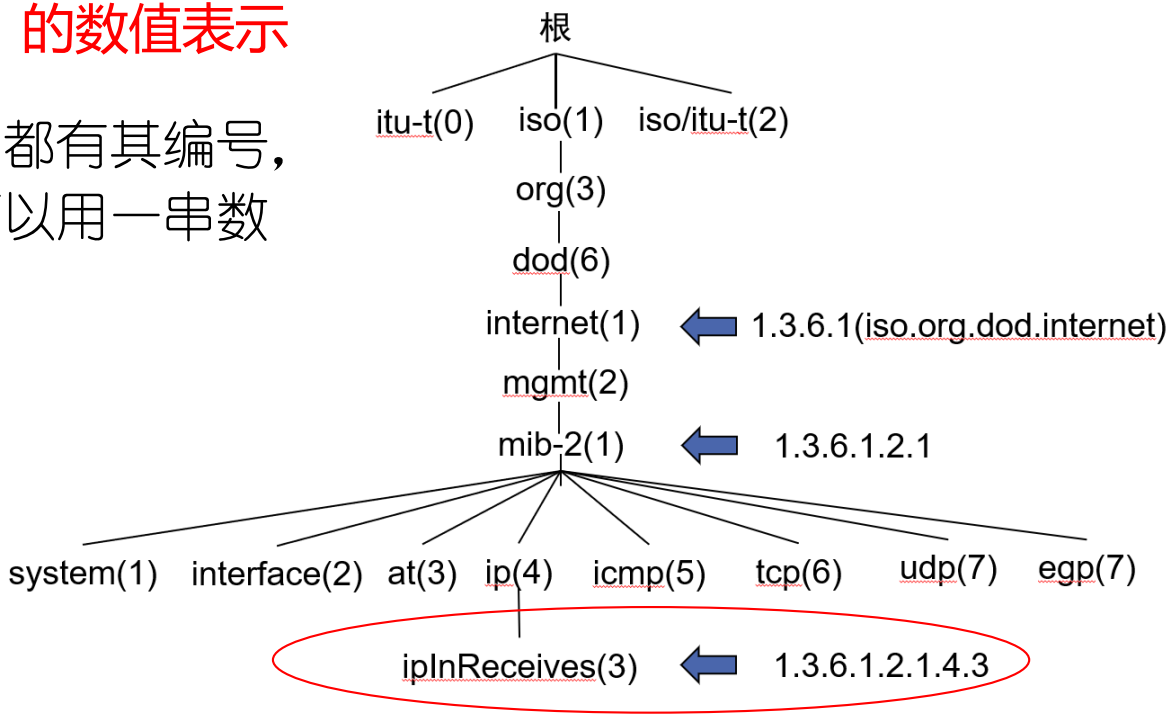
# MIB变量及类别、含义对比表

| MIB 变量 | 所属类别 | 意　义 |
|---|---|---|
| sysUpTime | system | 距上次重启动的时间 |
| ifNumber | interfaces | 网络接口数 |
| ifMtu | interfaces | 特定接口的最大传送单元 MTU |
| ipDefaultTTL | ip | IP 在生存时间字段中使用的值 |
| ipInReceives | ip | 接收到的数据报数目 |
| ipForwDatagrams | ip | 转发的数据报数目 |
| ipOutNoRoutes | ip | 路由选择失败的数目 |
| ipReasmOKs | ip | 重装的数据报数目 |
| ipFragOKs | ip | 分片的数据报数目 |
| ipRoutingTable | ip | IP 路由表 |
| icmpInEchos | icmp | 收到的 ICMP 回送请求数目 |
| tcpRtoMin | tcp | TCP 允许的最小重传时间 |
| tcpMaxConn | tcp | 允许的最大 TCP 连接数目 |
| tcpInSegs | tcp | 已收到的 TCP 报文段数目 |
| udpInDatagrams | udp | 已收到的 UDP 数据报数目 |

# 4、管理信息库 MIB

## （3）OID（对象标识符）的数值表示

- OID树上的每个节点都有其编号，因此一个MIB对象可以用一串数值表示。



# 4、管理信息库 MIB

## （3）OID（对象标识符）的数值表示

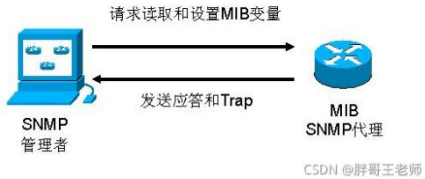
| OID值 | 描述 | 适用操作系统 |
| --- | --- | --- |
| .1.3.6.1.2.1.1.1.0 | 获取系统基本信息 | Linux / Windows |
| .1.3.6.1.2.1.1.3.0 | 监控时间 | Linux / Windows |
| .1.3.6.1.2.1.2.1.0 | 网络接口的数目 | Linux / Windows |
| .1.3.6.1.2.1.2.2.1.3 | 网络接口类型 | Linux / Windows |
| .1.3.6.1.2.1.2.2.1.6 | 接口的物理地址 | Linux / Windows |
| .1.3.6.1.2.1.2.2.1.10 | 接口收到的字节数 | Linux / Windows |
| .1.3.6.1.2.1.25.2.3.1.4 | 硬盘簇的大小 | Linux / Windows |
| .1.3.6.1.2.1.25.2.3.1.5 | 硬盘簇的的数目 | Linux / Windows |
| .1.3.6.1.2.1.25.2.3.1.6 | 硬盘占用率（已使用/总容量） | Linux / Windows |
| .1.3.6.1.4.1.2021.11.10.0 | 系统CPU百分比 | Linux |
| .1.3.6.1.4.1.2021.11.11.0 | 空闲CPU百分比 | Linux |

# 4、管理信息库 MIB

## (4) 专有MIB和通用MIB

- 一些厂商采用专有的管理MIB库，以实现对厂商设备本身的管理，包括可以显示出厂商设备图形化的界面等。如思考的Cisco View和华为的Quidview等；
- 一些通用网络管理软件可提供一个第三方的网管平台，支持对所有SNMP设备的发现和监控，可集成厂商的私有（专有）MIB库，可实现对全网（多厂商）设备进行识别和统一管理。

# 五、SNMP—— SNMP（本身）



# 5、SNMP（本身）

## （1）管理站获取被管设备中的管理信息有两种模式

- SNMP是管理站与代理（被管设备）之间进行数据交互的通信协议。
- 一般来说，管理站获取代理中的管理信息有两种模式：
  - 探询：由管理站主动发起，代理接到请求后做出响应；
  - 事件报告：当事件发生时，代理主动向管理站报告情况；

# 5、SNMP（本身）

## （1）管理站获取被管设备中的管理信息有两种模式

- 探询操作：包含两种基本的管理功能。
  - “读”操作，用 get 报文来检测（获取）各被管对象的状况
  - “写”操作，用 set 报文来改变各被管对象的状况

# 5、SNMP（本身）

## （1）管理站获取被管设备中的管理信息有两种模式

- 事件报告：SNMP 陷阱（SNMP trap）
  - SNMP 不是完全的探询协议，它也允许不经过询问就能发送某些信息。这种信息称为陷阱，表示它能够捕捉“事件”。
  - 当被管对象的代理检测到有事件发生时，就检查其门限值。代理只向管理进程报告达到某些门限值的事件（即过滤）。
  - 过滤的好处是：
    - 仅在严重事件发生时才发送陷阱
    - 陷阱信息很简单且所需字节数很少

# 5、SNMP（本身）

## （1）管理站获取被管设备中的管理信息有两种模式

- 两种模式的总结：
  - 管理站使用探询（至少是周期性地）以维持对网络资源的实时监视。
  - 代理，也可采用陷阱机制，主动报告特殊事件，使得 SNMP 成为一种有效的网络管理协议。

# 5、SNMP（本身）

## （2）SNMP规定的具体操作

- SNMPv1中规定了5种操作（5种消息类型）：

① Get-Request：　　管理站→代理（被管设备）

SNMP 管理站用Get-Request消息从拥有SNMP代理的网络设备中检索信息，即从代理进程处提取一个或多个参数值

② Get-Next-Request：　　管理站→代理（被管设备）

管理站从代理进程处提取一个（或多个）参数的下一个参数值。

③ Set-Request：　　管理站→代理（被管设备）

SNMP管理站用该消息对网络设备进行远程配置（包括设备名、设备属性、删除设备或使某一个设备属性有效/无效等）

# 5、SNMP（本身）

## （2）SNMP规定的具体操作

- SNMPv1中规定了5种操作（5种消息类型）：

④ **Get-Response:** 代理（被管设备）→管理站

SNMP代理则用Get-Response消息响应request操作。

⑤ **Trap:** 代理（被管设备）→管理站

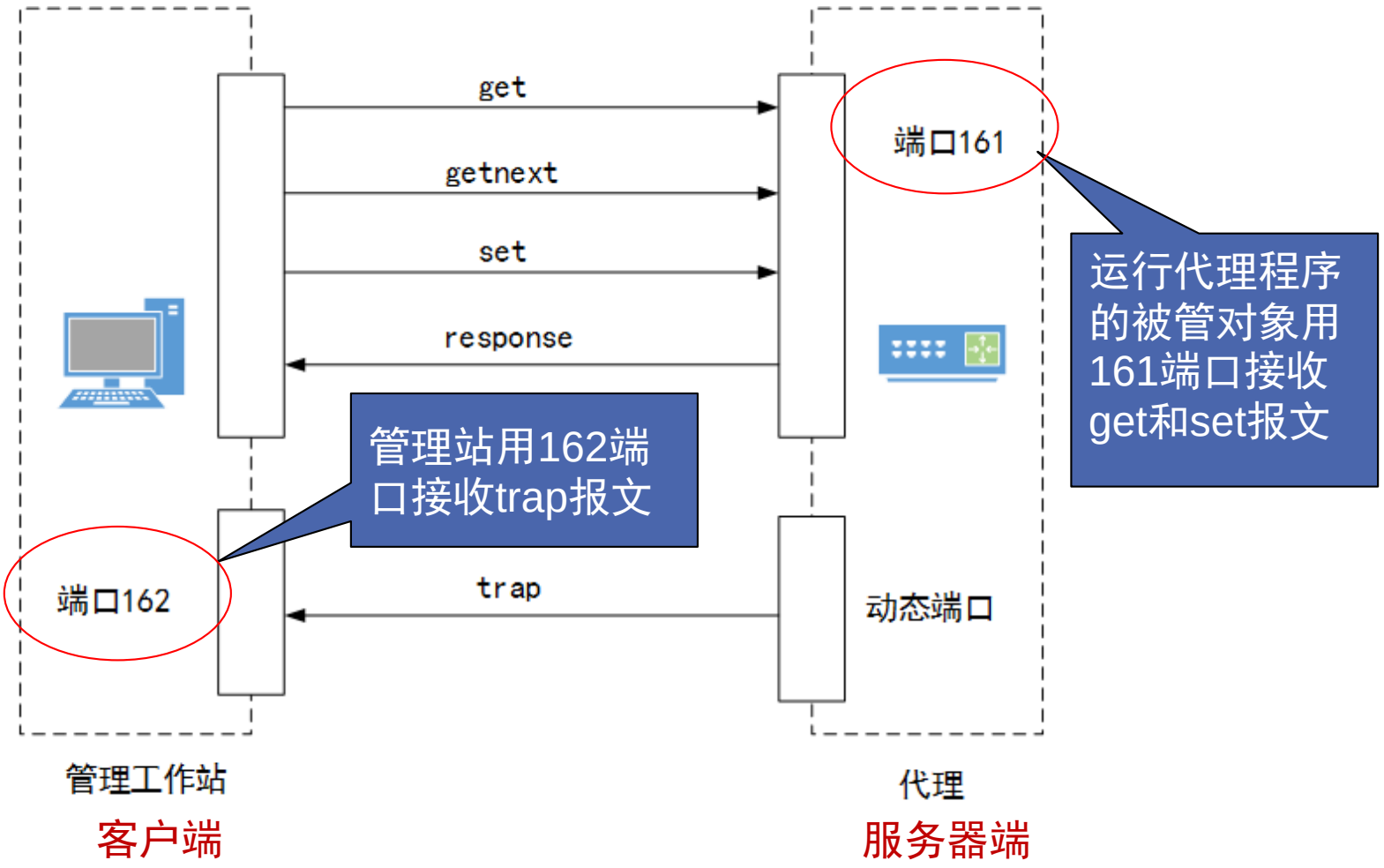
SNMP代理使用Trap向SNMP管理站发送非请求消息，一般用于描述某一事件的发生。Trap是从代理进程发送到管理进程的，而管理进程不需要给代理进程发送确认。

# 5、SNMP（本身）

## （2）SNMP规定的具体操作

- 总结：SNMPv1中规定的5种操作（5种消息类型）：
  - Get-Request　　管理站→代理（被管设备）
  - Get-Next-Request　　管理站→代理（被管设备）
  - Set-Request　　管理站→代理（被管设备）
  - Get-Response　　管理站←代理（被管设备）
  - Trap　　管理站←代理（被管设备）

# SNMP的工作过程图


get
getnext
set
response
trap
端口161
运行代理程序的被管对象用161端口接收get和set报文
管理站用162端口接收trap报文
端口162
动态端口
管理工作站
客户端
代理
服务器端

# 5、SNMP（本身）

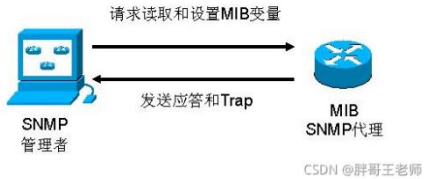
## （2）SNMP规定的具体操作

- SNMPv2中新增了2种操作：

① get-bulk-request 操作：允许SNMPv2管理者请求得到在给定的条件下尽可能大的应答。

② inform操作：与trap操作相同，也是代理主动向管理发送报文。inform操作相当于trap操作的升级版，因为trap报文发出去之后不会收到响应报文，而inform报文在发送之后能收到响应报文。

# 六、SNMP的命令与应用



# 6、SNMP的命令与应用

## （1）get请求的命令有两个

```
snmpget   -v2c  -c 共同体名 IP地址 OID号
snmpwalk  -v2c  -c 共同体名 IP地址 OID号
```

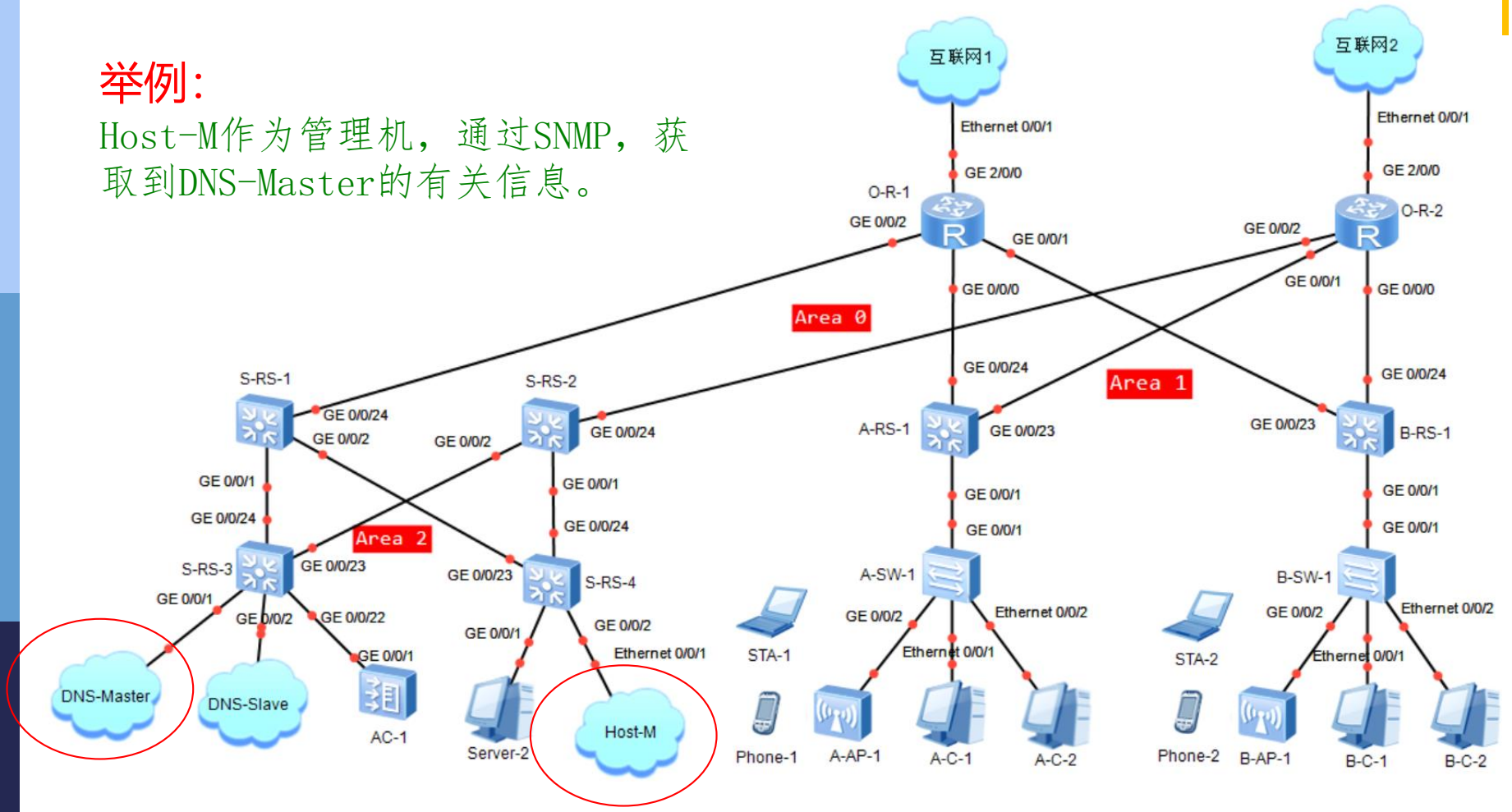
- snmpget和snmpwalk两者主要的区别
  - snmpwalk是对OID值的遍历，比如某个OID值下面有N个子节点，则依次遍历出这N个子节点的值；
  - snmpget是取具体的OID的值，适用于OID值是一个叶子节点的情况

# 回忆（2、SNMP概述）

- *设置SNMP共同体名的意义*
  - *在进行SNMP配置时，如果使用的是SNMPv1和SNMPv2版本，则需要在代理（被管设备）一侧设置其共同体（community）名称。*
  - *共同体名称类似一个字符串，它是一种安全机制，是代理和管理站之间的认证。SNMPv1和V2的安全机制很简单，只是验证共同体名。即，当管理站对被管设备（代理）进行访问时，在相关命令中通常需要输入代理的共同体名，作为一种认证机制。*
  - *SNMPv3则是采用更复杂的安全机制。*

# 举例：

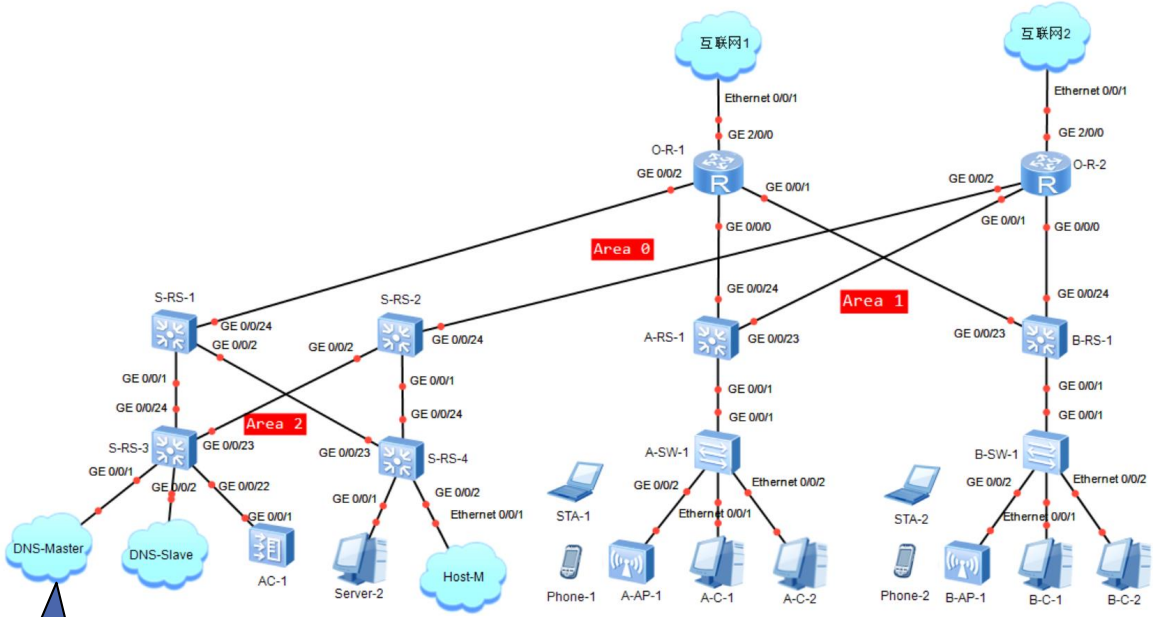
Host-M作为管理机，通过SNMP，获取到DNS-Master的有关信息。



# ➢ get命令举例（1）

① 在服务器DNS-Master中要安装snmp的代理程序（即服务器端程序），并启动snmp服务。

② 此处的虚拟机DNS-Master上部署的是CentOS 8操作系统。

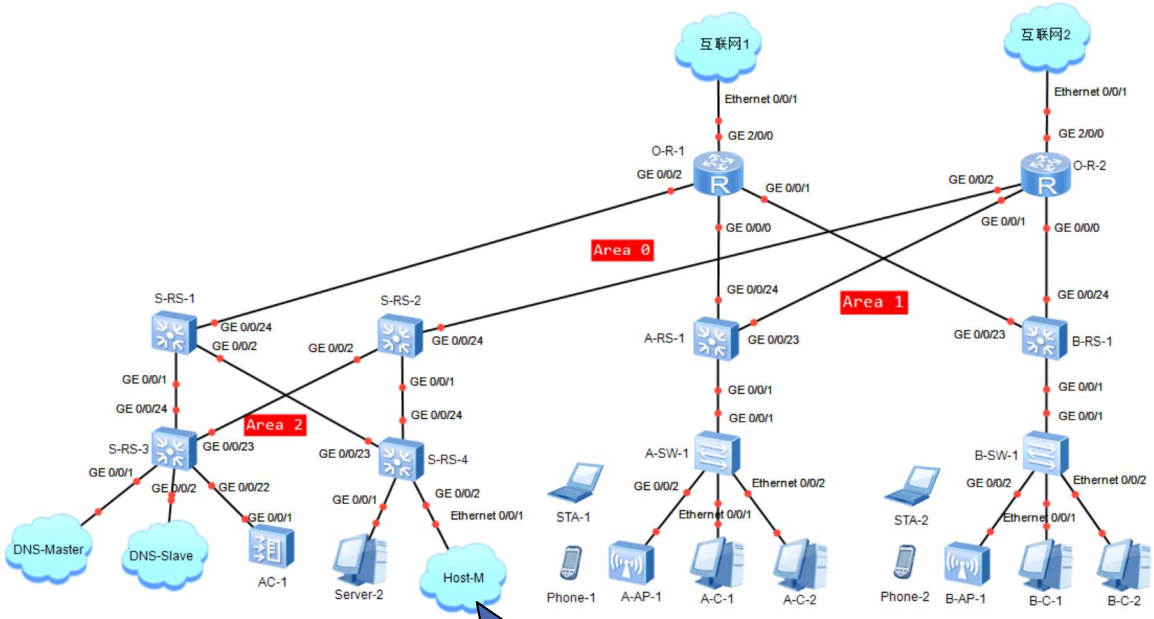



被管设备上要安装snmp

# ➢ get命令举例（2）

③ 在管理站（此处是Host-M）上要安装snmp的管理程序（即客户端程序），此处安装的是net-snmp。

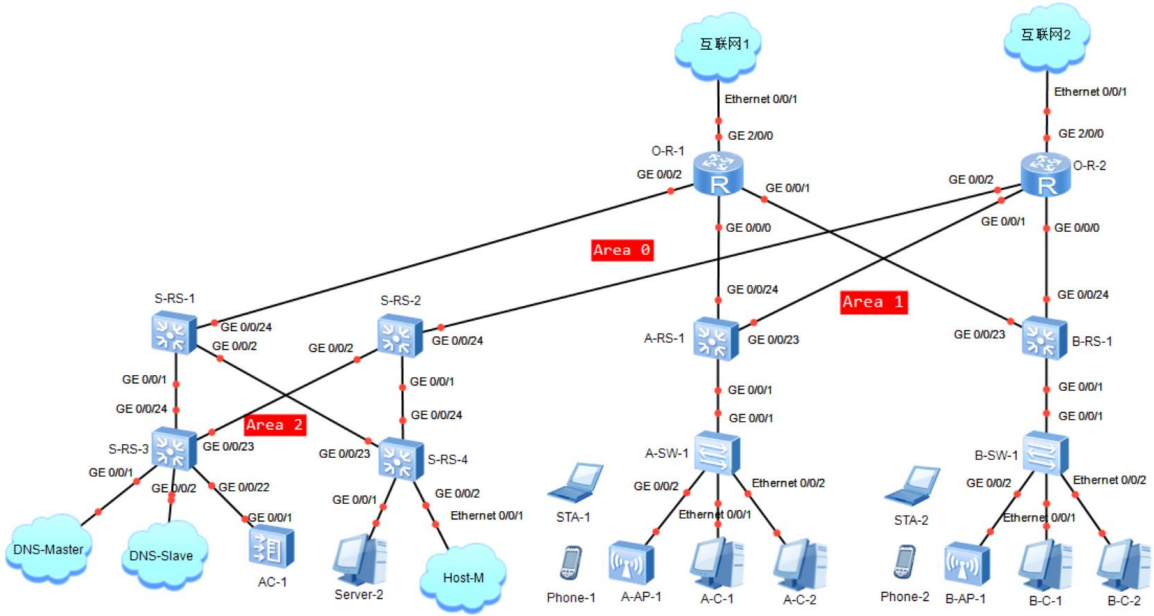
④ 此处的Host-M用的是实体主机，部署的是Windows10操作系统。



# ➢ get命令举例（3）

⑤ 确认管理站（此处是Host-M）与被管设备（DNS-Master）之间是网络互通的。

例如，Host-M能ping通DNS-Master



## ➢ get命令举例（4）—— 获取被管设备的内存总额

- 在Host-M中输入命令

```
snmpget -v2c -c My_Cacti 192.168.31.50  .1.3.6.1.4.1.2021.4.5.0
```

  - 此命令是通过Net-SNMP工具向被监控主机发送了一个SNMP请求，其中，snmpget为命令动词，-v2c表示使用SNMPv2，My_Cacti是被监控主机的共同体名称，192.168.31.50是被监控主机的IP地址，.1.3.6.1.4.1.2021.4.5.0是MIB的值（即OID值，此处表示获取内存总额）。

```
C:\Users>snmpget -v 2c -c My_Cacti 192.168.31.50 .1.3.6.1.4.1.2021.4.5.0
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 501080 kB
```

## ➢ get命令举例（5）——获取被管设备的内存信息

- 在管理站中输入命令

```
snmpwalk -v 2c -c public 192.168.31.50  .1.3.6.1.4.1.2021.4
```

  - 此命令是通过Net-SNMP工具向被监控主机发送了一个SNMP请求，其中，snmpwalk为命令动词，-v 2c表示使用SNMPv2，public是被监控主机的共同体名称，192.168.31.50是被监控主机的IP地址，.1.3.6.1.4.1.2021.4是MIB的值（即OID值，此处表示获取内存相关信息，即遍历.4后面子节点信息）。
  - 截图见下图

## ➢ get命令举例（6）——— 获取被管设备的内存信息

- 在管理站中输入命令

snmpwalk -v 2c -c My_Cacti 192.168.31.50 .1.3.6.1.4.1.2021.4

```
C:\Users>snmpwalk -v 2c -c My_Cacti 192.168.31.50 .1.3.6.1.4.1.2021.4
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 839676 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 839676 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 501080 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 241792 kB
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 1081468 kB
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000 kB
UCD-SNMP-MIB::memShared.0 = INTEGER: 4464 kB
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 764 kB
UCD-SNMP-MIB::memCached.0 = INTEGER: 167456 kB
UCD-SNMP-MIB::memSwapError.0 = INTEGER: noError(0)
UCD-SNMP-MIB::memSwapErrorMsg.0 = STRING:
```

# 6、SNMP的命令与应用

## (2) get-next请求命令:

snmpgetnext -v 2c -c 共同体名 IP地址 OID号

- get-next请求主要是为了得到“命令中OID对象”的下一个对象的值，而“下一个对象”是按照MIB库中的对象实例的顺序来返回对象的值。
- 如使用get-next请求采集OID为“1.3.6.1.2.1.1.3.0”的值，而应答报文中返回是OID为“1.3.6.1.2.1.1.4.0”对象实例的值。

# 6、SNMP的命令与应用

## （3）get-bulk请求命令

snmpbulkget －v2c －c 共同体名 IP地址 OID号

- get-bulk请求是为了使用户一次请求可以得到多个对象的值，从而减少用户的一些操作。
- 如假定发送的get-bulk请求对象实例为1.3.6.1.2.1.1.3.0（sysUpTime,系统开机时间），由于请求报文中non-repeaters的值为0、max-repetitions的值为10，所以在应答报文中将会返回该对象后不重复的10个对象实例的值。

# 6、SNMP的命令与应用

## （4）set请求命令

snmpset－v2c －c 共同体名 IP地址 OID号 命令类型 命令设置的值

- set请求是为了修改MIB库中某个OID对应的值，但不同的OID采集的值是不同类型的，因此在set请求时，应该设置相应值的类型。但设置的set请求命令类型必须和原来采集值的类型一致。
- 例：使用该命令将被管设备的系统名称改为“linux”，系统名为字符串型，具体命令为：

snmpset -v2c -c 共同体名 IP地址 .1.3.6.2.1.1.5.0 s linux

命令类型

# 6、SNMP的命令与应用

set请求命令类型对比表

| 命令类型 | 说明 | 命令类型 | 说明 |
|---|---|---|---|
| i | INTEGER（整数，有符号的 32 位整数，取值范围为－2147483648 到 +2147483648） | u | UNSIGNED（无符号的 32 位整数，值的范围为：0－4294967295） |
| s | STRING（字符串类型，通常限制在 255 个字符内） | x | HEX STRING（十六位整数，一般用于私有的 MIB 中进行设置） |
| b | BITS（比特串，一个无符号的数据类型） | n | NULLOBJ（将对象设置成空对象，从而不能采集该对象的值） |
| t | TIMETICKS(表示代表数据的一个无符号整数，2^32 取模（4294967296），以百分之一秒为单位） | a | IPADDRESS（表示一个 IP 地址） |

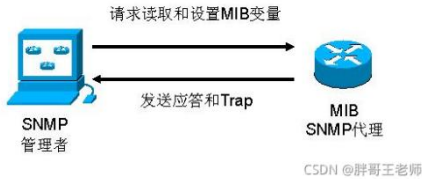
# 6、SNMP的命令与应用

## （5）trap命令SNMP的操作命令（v2版本）

snmptrap -v2c -c 共同体名 管理主机 uptime OID号

- Trap命令是代理主动采集设备中的相关信息，然后向管理站发送这些信息，上述的命令是在Linux系统下完成代理，并向Windows系统下的主机发送。
- trap消息的条件为：代理主机安装并开启了SNMP服务，并且确认能够使用SNMP的trap服务，并且在防火墙上开启相关端口允许SNMP发送消息。

# 七、 构建基于Cacti的园区网监控系统



# 7、构建基于Cacti的园区网监控系统

## 认识Cacti

- Cacti是一套基于PHP、MySQL、SNMP及RRDTool的运维监控图形分析系统。

# 7、构建基于Cacti的园区网监控系统

## 认识Cacti

### RRDTool

RRDTool是一个强大的绘图引擎，它是一套软件，包含：

- 存储：把近期的原始采集数据+统计分类的数据存在rrd文件中；
- 统计：分类统计功能；
- 操作：数据读写（从rra文件中）；
- 绘图：绘图工具。

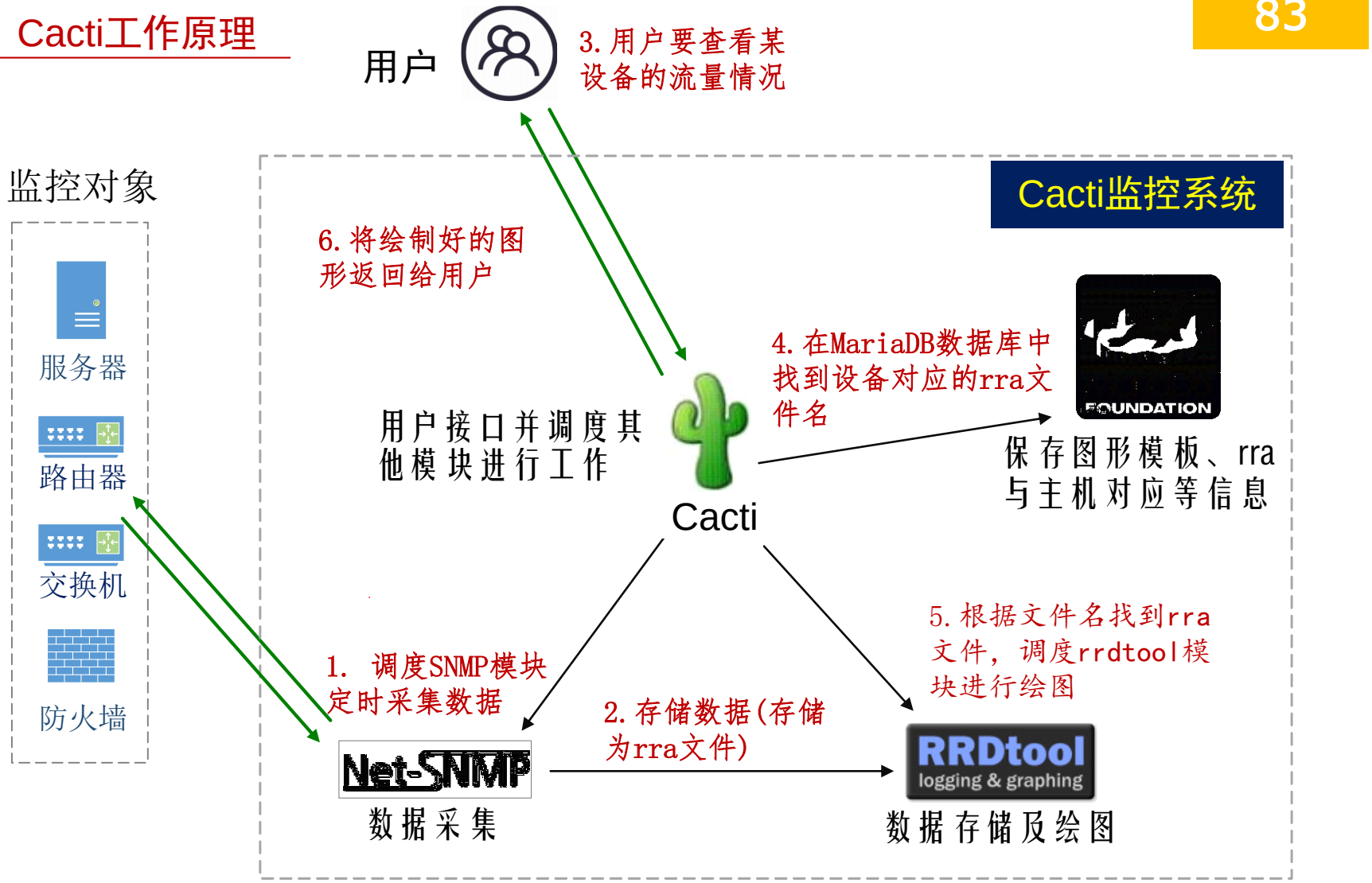
其中，RRD的含义是Round Robin Database，环形数据库。Round Robin是一种存储数据的方式，使用固定大小的空间来存储数据，并有一个指针指向最新数据的位置。RRD可使监控采集的数据循环更新。

# 7、构建基于Cacti的园区网监控系统

## Cacti工作原理

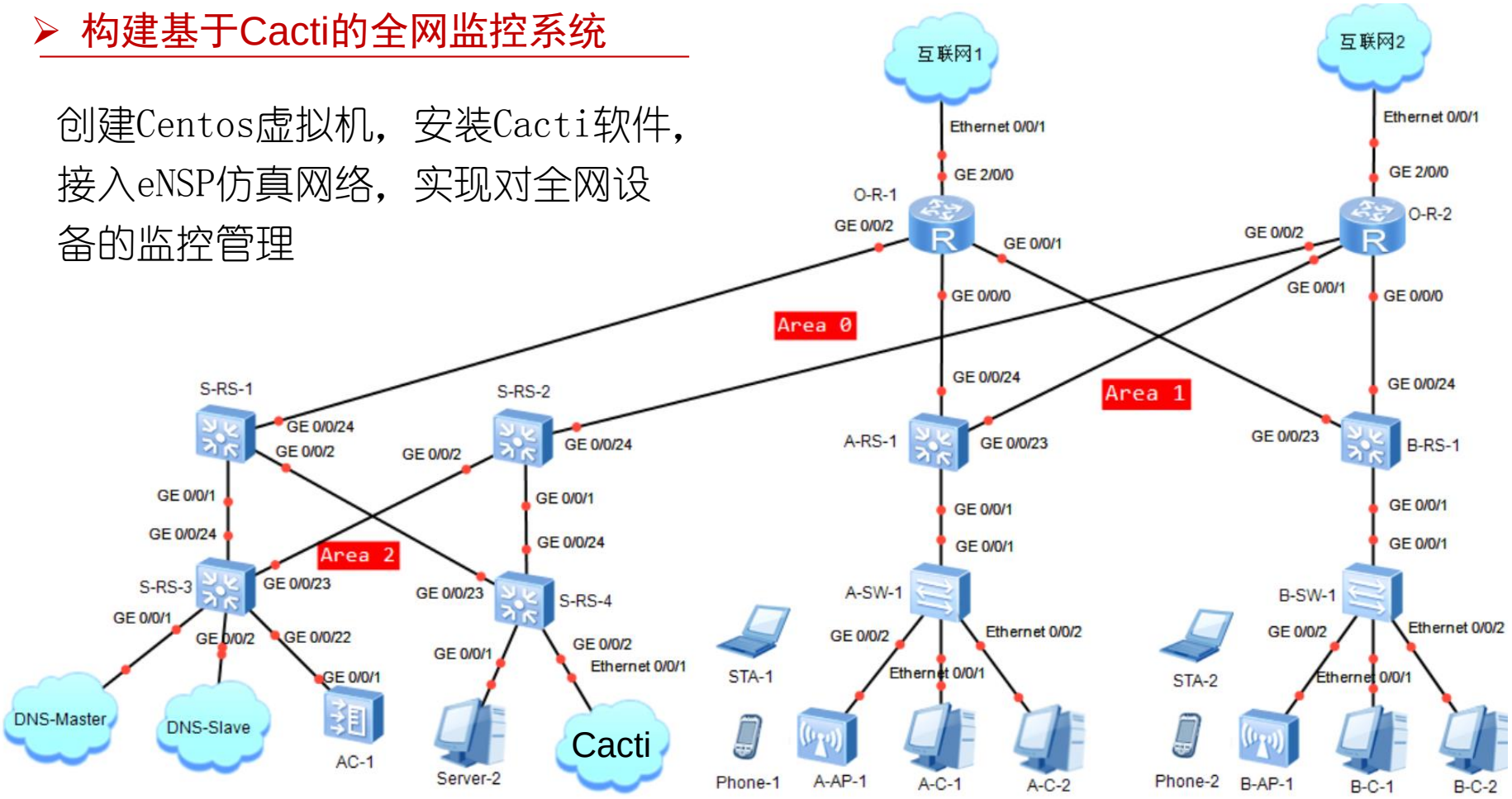
- Cacti使用PHP语言开发，通过snmpget来获取数据，使用RRDTool绘制监控数据图形。
- 在Cacti监控体系中，将监控数据（即监控采集的数据）和系统数据（例如账号密码、受监控设备的基本配置信息等）分开存放。监控数据放置在RRDTool的数据文件中，系统数据放置在MySQL/MariaDB数据库中。
- Cacti可通过配置监控对象模板文件实现自定义监测指标。

# Cacti工作原理


用户
3.用户要查看某设备的流量情况
监控对象
服务器
路由器
交换机
防火墙
Cacti监控系统
6.将绘制好的图形返回给用户
4.在MariaDB数据库中找到设备对应的rra文件名
用户接口并调度其他模块进行工作
Cacti
保存图形模板、rra与主机对应等信息
5.根据文件名找到rra文件，调度rrdtool模块进行绘图
1. 调度SNMP模块定时采集数据
2.存储数据(存储为rra文件)
Net-SNMP
数据采集
RRDtool
logging & graphing
数据存储及绘图

# ➢ 构建基于Cacti的全网监控系统

创建Centos虚拟机，安装Cacti软件，接入eNSP仿真网络，实现对全网设备的监控管理



Thanks.