第4章简单网络管理协议

- SNMP的简介
- SNMP的发展
- SNMP相关概念
- SNMPv1
- SNMPv2
- SNMPv3

网络管理的标准化

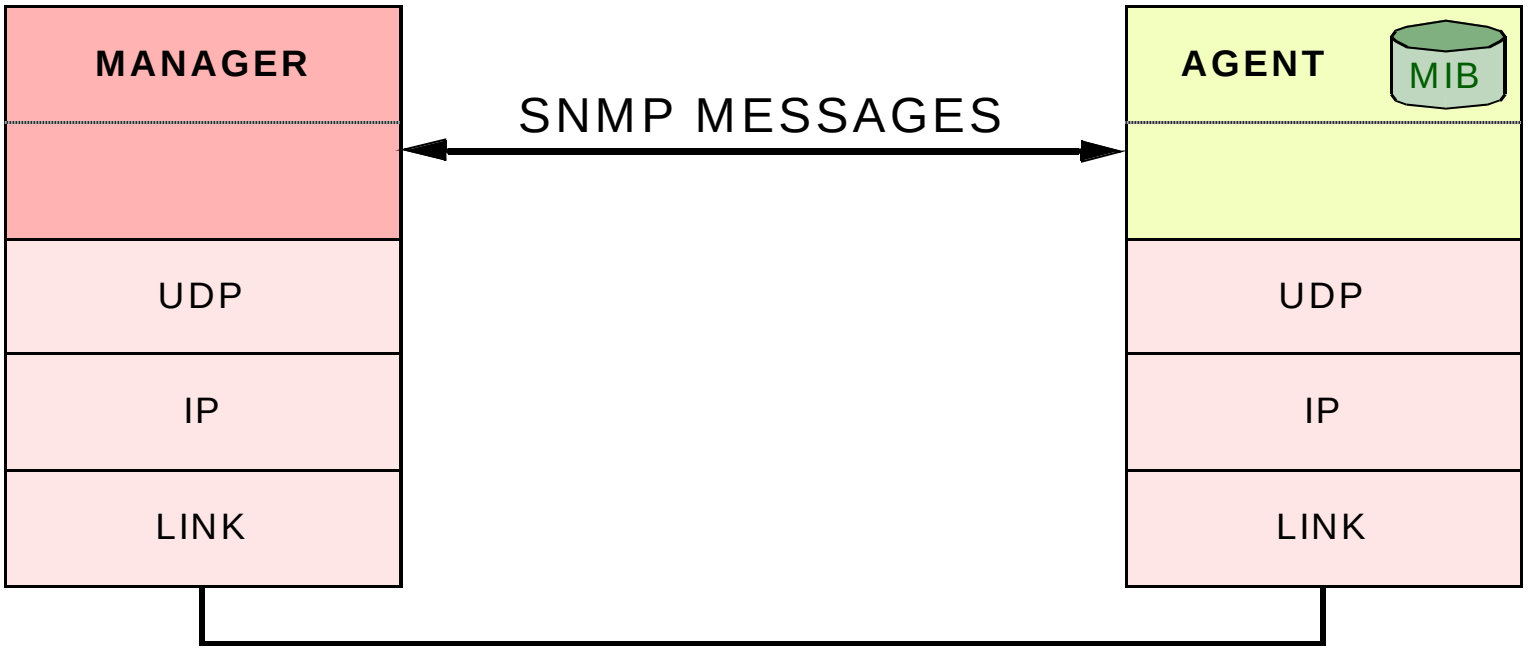
- 如果每个厂商的网络设备都提供一套自己独特的网管方法和界面，网络管理的工作将很难进行。
- 网络管理的标准化
  - 每个的网络设备必须提供一致的网络管理的界面 (亦即相同的网络管理通信协议)。

SNMP简介

- SNMP:
  - Simple Network Management Protocol
  - 是管理TCP/IP 网络 (Internet)的事实上的标准
  - 所有TCP/IP网络设备都应该支持SNMP.

SNMP简介

- SNMP基于UDP, 提供的是一种面向无连接的服务



SNMP简介

- SNMP的网络管理共有三部分
  - 管理信息的结构SMI: 报告对象是如何定义的以及如何表示在MIB中的
  - 管理信息库: 管理信息库MIB描述存放报告对象的管理参数
  - SNMP本身: 提供在网络管理站和被管设备交互信息的方法

复习几个概念

- ASN.1: 用于定义语法的正式语言。
- SMI（Structure of Management Information管理信息结构）：定义了一个ASN.1的子集，规定了SNMP使用到哪些ASN.1符号与元素，通过使用这些ASN.1子集的符号和元素描述SNMP。
- BER(Basic Encoding Rule基本编码规则)是一种编码规格，描述如何将ASN.1类型的值编码成字节串(string of octets)的方法。是ASN.1标准的一部分。SNMP使用BER编码将SNMP的操作请求和应答编码进行传输，并于接收端进行解码。

数据传输

SNMP的发展

- 1987年11月提出简单网关监控协议(SGMP)
- 简单网络管理协议第一版(SNMPv1) 公布在1990年和1991年的几个RFC文件中，即

  RFC 1155(SMI) RFC 1157(SNMP) RFC 1212(MIB) RFC 1213(MIB-2)
- 双轨制策略：
  - SNMP满足当前的网络管理需要，并可平稳过渡到新的网络管理标准。
  - OSI网络管理(CMOT)作为长期解决办法，可提供更全面的管理功能，但需较长开发及接受过程。但OSI MIB采用面向对象模型，开发缓慢，SNMP无法顺利过渡。

SNMP的发展

- SNMP简单易实现，但没有实质性的安全措施，无数据源认证功能，不能防止被偷听。为弥补SNMP的安全缺陷，1992年发布S-SNMP，该协议增强了以下安全方面的功能：用报文摘要算法MD5保证数据完整性和进行数据源认证；用时间戳对报文排序；用DES算法提供数据加密功能。
- 但S-SNMP没有改进SNMP功能和效率方面的缺点。于是又提出SMP协议，该协议在使用范围、复杂程度、速度和效率、安全措施、兼容性等方面对SNMP进行了扩充。1993年发布SNMPv2，它以SMP为基础，放弃了S-SNMP。
- 1996年1月发布SNMPv2C

SNMP的发展

- 1999年4月公布了SNMPv3的新标准草案。增加了安全和高层管理功能，且能和以前的标准（SNMPv1和SNMPv2）兼容，以便于以后扩充新的模块，从而形成了统一的SNMP新标准。

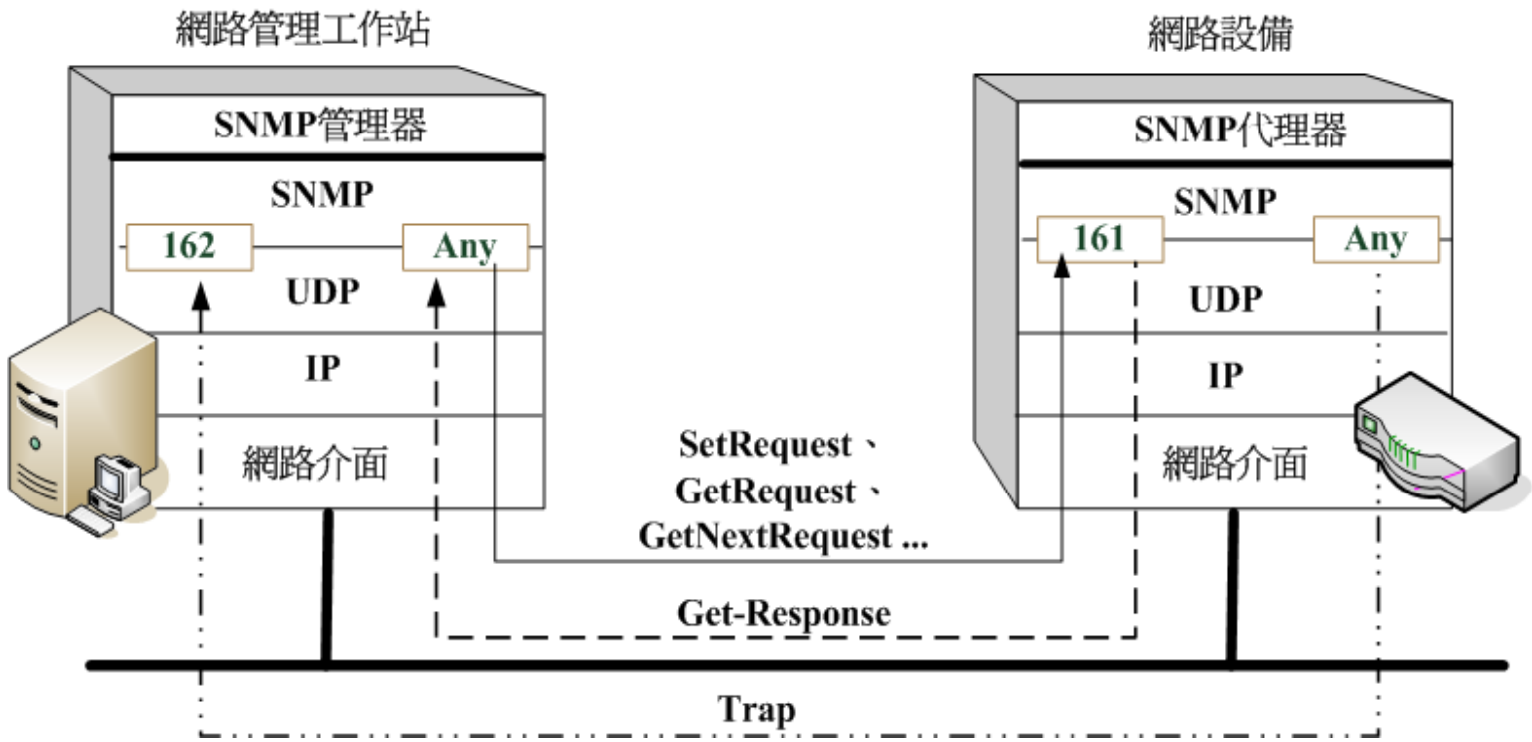
SNMP 基本框架

- 管理者(管理进程)

  管理指令的发出者，监视和控制网络设备
- 网管代理

  执行管理指令，向管理者报告被管对象发生的事件
- 管理信息库(MIB)——核心
  - 可管对象的集合；
  - 被管对象结构化组织的抽象；
  - 概念上的数据库
- SNMP通信协议主要包括以下能力：
  - Get：管理站读取代理者处对象的值。
  - Set：管理站设置代理者处对象的值。
  - Trap：代理者向管理站通报重要事件。

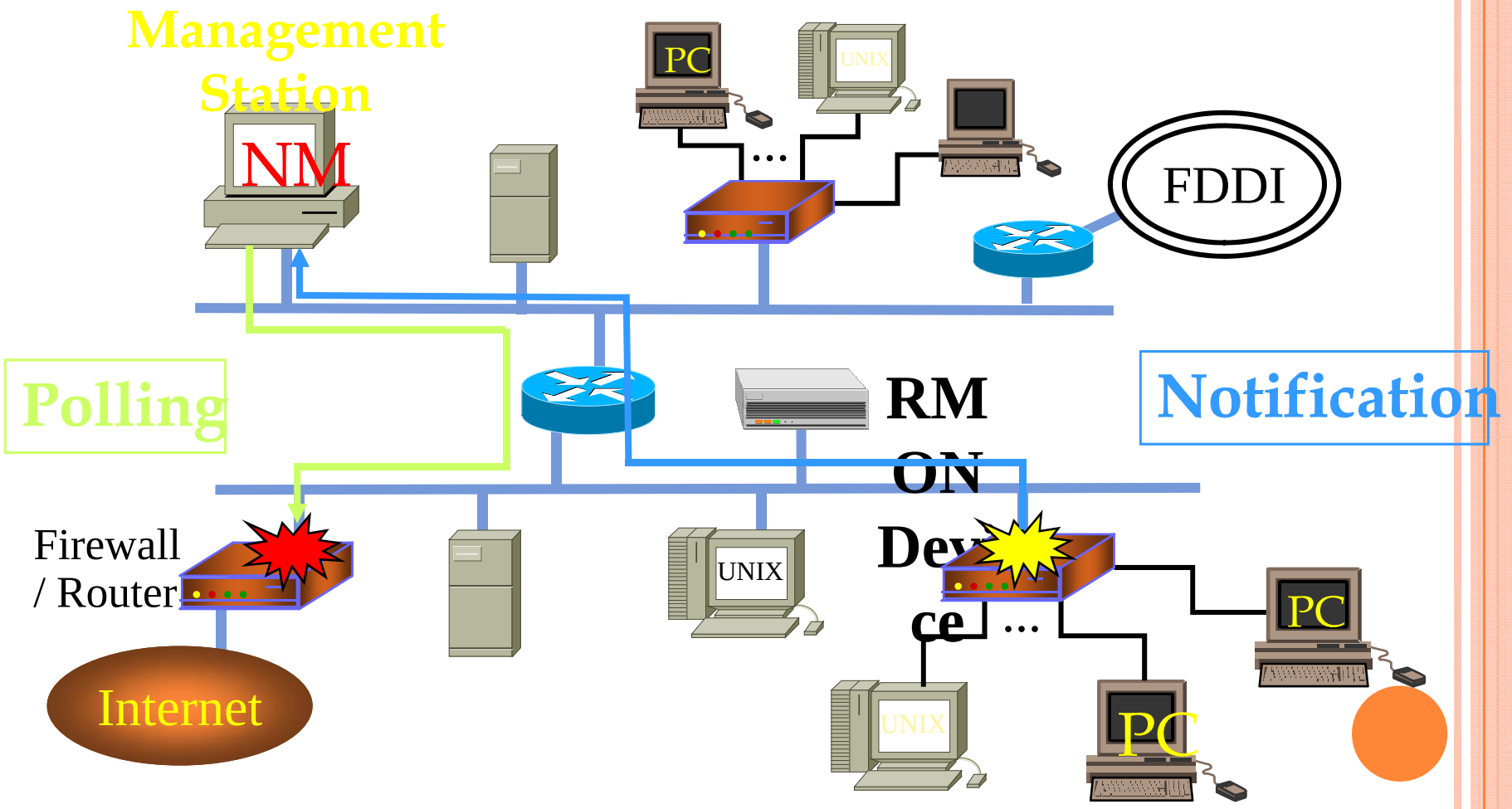
SNMP通讯模式架构


網路管理工作站
SNMP管理器
SNMP
162
Any
UDP
IP
網路介面
網路設備
SNMP代理器
SNMP
161
Any
UDP
IP
網路介面
SetRequest、
GetRequest、
GetNextRequest ...
Get-Response
Trap

SNMP的相关概念

1.SNMP的操作

- get：获取特定的对象的管理信息值
- get-next：遍历MIB树获取对象的值
- set：修改对象的值
- trap：代理发送非请求性通知给网络管理系统，通报有重要事件发生了。


Management Station
NM
PC
UNIX
FDDI
Polling
Notification
RM
ON
Dev
ce
Firewall
/ Router
Internet
UNIX
UNIX
PC
PC

SNMP的相关概念

2. 团体名(community name)

每个团体被赋予一个惟一的名字，管理者只能以代理认可的团体名行使其访问权。

get或get-next操作使用读团体名

set操作使用写团体名

有效范围：在定义它的代理系统中

缺省的 'Get'团体名： public

SNMP的相关概念

3. 变量绑定

指定要收集或修改的管理对象，是一个OBJECT IDENTIFIER值对应的列表。

对于get或get-next请求，将忽略该值部分。

SNMPv1

- SNMPv1是一种简单的请求/响应协议，使用管理者-代理模型，仅支持对管理对象值的检索和修改等简单操作。网络管理系统发出一个请求，管理器则返回一个响应。该过程通过SNMP操作实现。但SNMP不支持管理站改变管理信息库的结构，即不能增加和删除管理信息库中的管理对象实例；管理站只能逐个访问管理信息库中的叶节点，不能一次性访问一个子树，这些限制简化了SNMP的实现，但是也限制了网络管理的功能。

SNMPv1报文格式

variable bindings:

NAME *1*	VALUE *1*	NAME *2*	VALUE *2*	•••	•••	NAME *n*	VALUE *n*

SNMP PDU:

PDU TYPE*	REQUEST ID	ERROR STATUS	ERROR INDEX	VARIABLE BINDINGS

SNMP message:

VERSION	COMMUNITY	SNMP PDU

SNMPv1报文格式

- 版本号：指定SNMP的版本号，写入版本字段的是版本号减1
- 团体名：OCTET STRING类，用于身份认证，作为管理进程和代理进程之间的明文口令
- SNMP PDU：协议数据单元。包含5种类型：
  - GetRequest-PDU
  - GetNextquest-PDU
  - SetRequest-PDU
  - GetResponse-PDU
  - Trap-PDU

SNMPv1报文格式

1. 命令和响应的PDU格式

PDU 类型	请求标识符	差错状态	差错索引	变量绑定表

(1)PDU 类型： GetRequest-PDU， GetNextquest-PDU，SetRequest-PDU，GetResponse-PDU(分别是0xA0、0xA1、0xA3、0xA2)

(2)请求标识符(request ID)字段：赋予每个请求报文惟一的整数，用于区分不同请求。

(3)差错状态(error status)字段：代理处理管理者的请求时可能出现的各种错误。

- 只在GetResponse-PDU中使用，其他类型的PDU中为0。
- 6种差错状态：

SNMPv1报文格式

① noError(0)：成功处理该请求

② tooBig(1)：代理不能把该请求的结果放入到单个SNMP报文中

③ noSuchName(2)：在指定团体名的基础上指定了一个代理不知道的对象

④ badValue(3)：set操作试图把一个对象修改成无效的或不一致的值

⑤ readOnly(4)：指示一个set操作试图修改不能被写入的变量

⑥ genError(5)：任何其他错误

(4) 差错索引字段:当差错状态非0时指向变量绑定表中第一个导致差错的变量

(5)变量绑定列表:变量名和对应值的表,说明要检索或设置的所有变量及其值

名字1　值1　名字2　值2　……　名字n　值n

SNMPv1报文格式

2. TrapPDU格式

PDU 类型	制造商ID	代理地址	通用陷阱	特殊陷阱	时间戳	变量绑定表

(1)PDU类型：Trap

(2)制造商ID：设备

(3)代理地址：产生

(4)通用陷阱：SNM

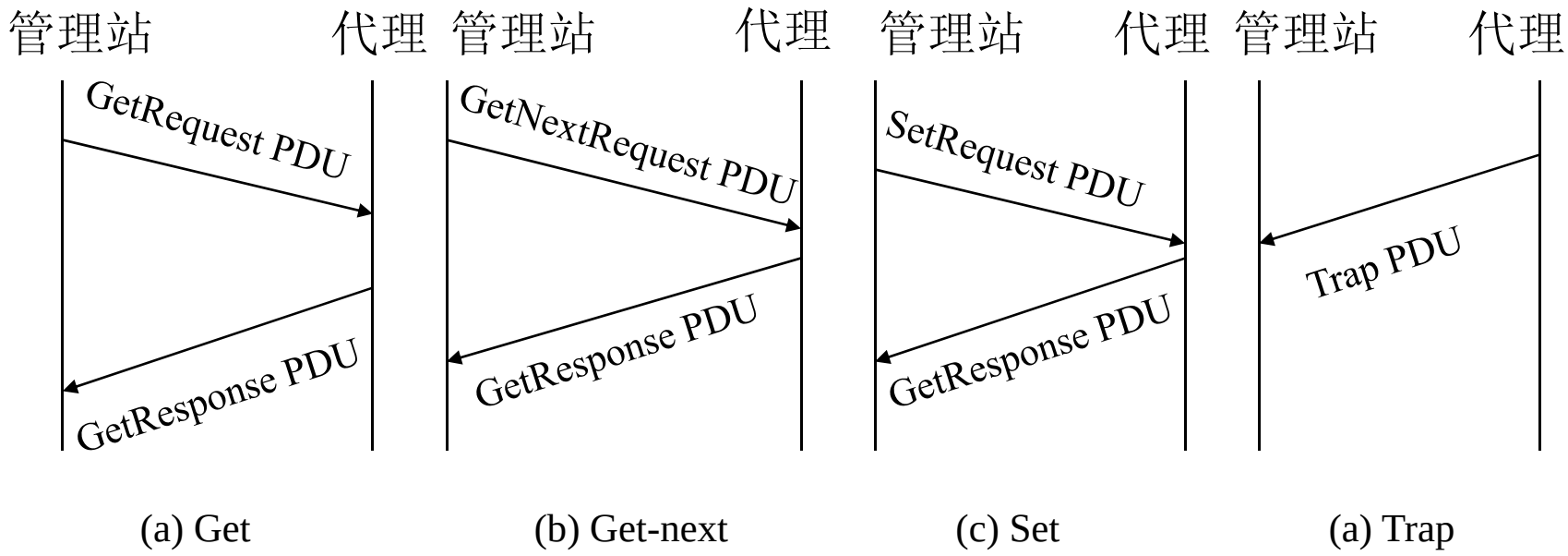
(5)特殊陷阱：与设

(6)时间戳：代理发

(7)变量绑定表：变量名和对应值的表

trap类型	含 义
ColdStart (0)	设备正在重启或初始化，以致于代理和配置也许会改变，通常此消息表示系统崩溃或其他重启状态
WarmStart (1)	设备正在重启或初始化，但代理和配置不会改变，通常表示一个简单刷新或重新启动操作系统
LinkDown (2)	表示设备的一个通信接口连接失败
LinkUP (3)	表示设备的一个通信（接口）链路正在接通或运行
AuthenticationFailure (4)	设备发生了认证失败或其他安全问题，一般情况下是由于一个无效的SNMP团体名引起
EgpNeighorLoss (5)	表示与该设备对等的EGP邻居（external gateway protocol，外部网关协议）宕机或互通关系无效
EnterpriseSpecific (6)	表示一些厂商专用的事件发生了，在特殊陷入（specifc-trap）字段指明具体的陷入类型。

SNMP报文应答序列



(a) Get　(b) Get-next　(c) Set　(a) Trap

- SNMP报文应答序列

报文发送与接收

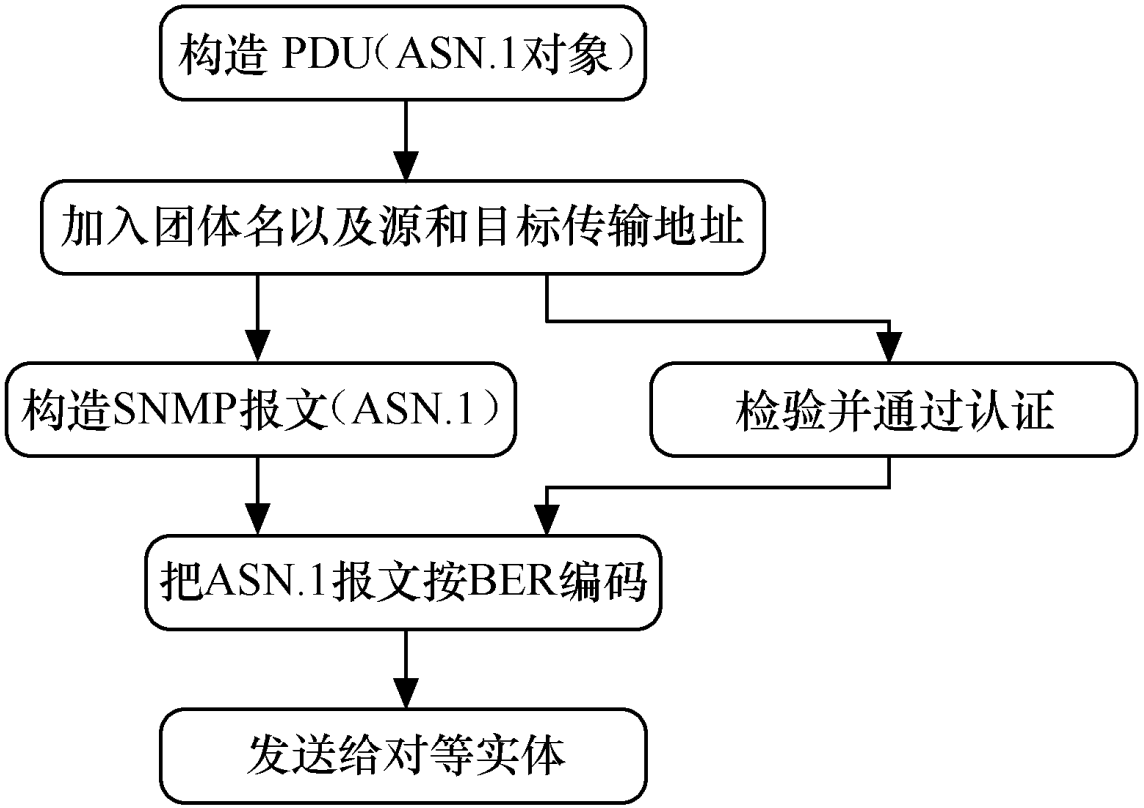
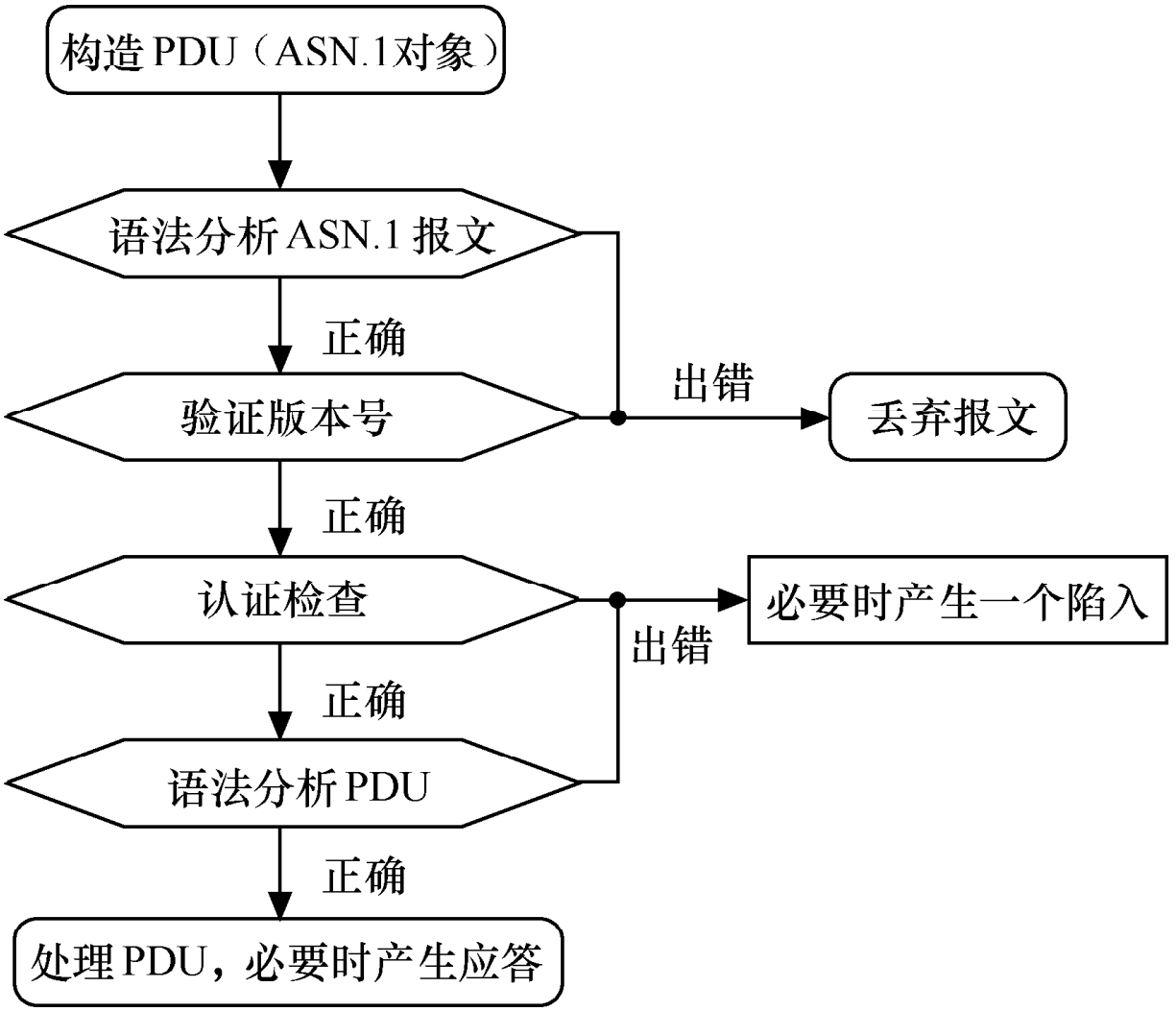


图5-3 生成和发送SNMP报文



接收和处理SNMP报文

响应的格式

0xA2	请求标识符	差错状态	差错索引	变量绑定表

- 0xA2：指示该SNMP是GetResponse。
- 请求标识符(reqid)：与原请求中的值相同。
- 差错状态(es)：指示该代理是否能成功处理该请求。
- 差错索引(ei)：非0则指示第一个变量在错误请求中的位置。
- 变量绑定：一个变量列表，每个变量包含一个对象标识符和一个值。

SNMP报文应答序列

3. 处理get和get-next请求：

请求	差错	GetResponse
get和 get-next	一个对象在指定团体名下不可用(对于get-next，如果下一个对象不存在)	差错状态：noSuchName 差错索引：该对象在变量绑定中的位置
	PDU太大	差错状态：tooBig
	由于其他原因不能获取一个值	差错状态：genErr 差错索引：该对象在变量绑定中的位置

SNMP报文应答序列

3. 处理set：

请求	差错	GetResponse
set	一个对象在指定团体名下不可用	差错状态：noSuchName 差错索引：该对象在变量绑定中的位置
	为该对象指定的值不一致	差错状态：badValue 差错索引：该对象在变量绑定中的位置
	PDU太大	差错状态： tooBig
	由于其他原因不能获取一个值	差错状态： genErr 差错索引：该对象在变量绑定中的位置

SNMP报文应答序列

- 一个SNMP响应信息包大小：至少484字节
- 一条484字节容纳的变量数：取决于对象标识符的长度和数据类型

检索简单对象

- 检索简单的标量对象值可以用Get操作。
- 如果变量绑定表中包含多个变量。则一次还可以检索多个标量对象的值。接收GetRcqucst的SNMP实体以请求标识相同的GetRcsponse响应。
- GetResponse操作具有原子性，即如果所有请求的对象值可以得到，则给予应答；反之，只要有一个对象的值得不到，则可能返回下列错误条件之一。

  （1）noSuchName，错误索引设置为一个数．指明有问题的变量，变量绑定表中不返回任何值。

  （2）tooBig。

  （3）由于其他原因（如代理不支持）响应实体至少不能提供一个对象的值，则返回的PDU中错误状态字段置为genErmr．错误索引置一个数，指明有问题的变量。变量绑定表中不返回任何值。

例4-1

- GetRequest(udpInDatagrams.0, udpNoPorts .0, UdplnErrors.0, udpoutDatagrams.0)

- GetResponse(udpInDatagrarns.0 = 100, udpNoPorts.0 = 1, udplnErrors.0 = 2, udpPoutDatagrams.0 = 200)

GET操作

实例	ifIndex	ifDescr	ifType
1	1	Ethernet	6
2	2	Ethernet	6
3	3	serial	22
4	4	ppp	23
5	5	Ethernet	6
6	6	Ethernet	6

get-request {sysUpTime.0，ifIndex.1，ifDescr.2，ifType.4}

sysUpTime.0 287231 ifIndex.1 1

ifDescr.2 ethernet ifType.4 23

(ifIndex.1，ifDescr.2，ifType.7) ?

noSuchName 差错索引3

(ifTable) ?

noSuchName 差错索引1

GET-NEXT操作

- GetNextRequest(udpInDatagrams，udpNoPorts,udpInErrors,udpOutDatagrams)

- GetResponse (udplnDatagrams.0 = 100, udpNoPorts.0 = 1, udplnErrors.0 = 2, udpOutDatagrams.0 =200)

GET-NEXT操作

实例	ifInOctets	ifInUcastPkts	ifInNUcastPkts
1	200123	5601	912
2	4587213	8876	1790
3	735543	7268	
4	6537722	200211	3388
5	2987653211	101392199	46421
6	783101	53211	4241

get-next {ifInOctets，ifInUcastPkts，ifInNUcastPkts}
ifInOctets.1 200123
ifInUcastPkts.1 5601
ifInNUcastPkts.1 912

GET-NEXT操作

get-next {ifInOctets.1，ifI

ifInOctets.2 4587213

ifInUcastPkts.2 8876

ifInNUcastPkts.2 1790

实例	ifInOctets	ifInUcastPkts	ifInNUcastPkts
1	200123	5601	912
2	4587213	8876	1790
3	735543	7268	
4	6537722	200211	3388
5	2987653211	101392199	46421
6	783101	53211	4241

get-next {ifInOctets.2，ifInUcastPkts.2，ifInNUcastPkts.2}

ifInOctets.3 735543

ifInUcastPkts.3 7268

ifInNUcastPkts.4 3388

实例	ifInOctets	ifInUcastPkts	ifInNUcastPkts
1	200123	5601	912
2	4587213	8876	1790
3	735543	7268	
4	6537722	200211	3388
5	2987653211	101392199	46421
6	783101	53211	4241

GET-NEXT操作

例：查询一个设备所有接口的速率

{iso(1)org(3)dod(6)internet(1)mgmt(2)mib(1)
interfaces(2)ifNumber(1)}

命令：

GetRequest(1.3.6.1.2.1.2.1.0)

GetResponse(2)

GetRequest(1.3.6.1.2.1.2.2.1.5.1)

GetResponse(10000000)

GetRequest(1.3.6.1.2.1.2.2.1.5.2) ----或者用

GetNextRequest(1.3.6.1.2.1.2.2.1.5.1)

GetResponse(56000)

检索未知对象

- GetNext命令检索变量名指示的下一个对象实例，但是并不要求变量名是对象标识符或者是实例标识
- 利用GetNext的检索未知对象的特性可以发现MIB的结构。

- 例4-3 管理站不知道Udp组有哪些变量，先发出命令
- GetNextRequest(udp)
- 得到的响应是

  GetResponse(udpInDatagrams.0=100)
- 继续这样找到其他管理对象。

检索表对象

- GetNextRequest (ipRouteDest, ipRouteMetric1, ipRouteNextHop)

ipRouteDest	ipRouteMetric1	ipRouteNext Hop
9.1.2.3	3	99.0.0.3
10.0.0.51	5	89.1.1.42
10.0.0.99	5	89.1.1.42

- GetResponse(ipRouteDest.9.1.2.3=9.1.2.3,ipRoutcMetric1.9.1.2. 3 =3, ipRouteNextHop.9.L2.3= 99.0.0.3)

检索表对象

- GetNextRequest (ipRouteDest.9. 1.2.3, ipRouteMetric1.9.1.2.3 , ipRouteNextHop.9.1.2.3)
- GetResponse (ipRouteDest. 10.0.0.51 = 10.0.0.51, ipRouteMetric1. 10.0.0.51=5, ipRouteNextHop.10.0.0.51 = 89.1.1.42)
- GetNextRequest (ipRouteDest. 10.0.0.51, ipRouteMetric1. 10.0.0.51, ipRouteNextHop. 10.0.0.51)
- GetResponse (ipRouteDest.1O.0.0.99 =10.0.0.99, ipRouteMetric1.10.0.0.99= 5, ipRouteNextHop. 10.0.0.99=89.1.1.42)

IpRouteDest	ipRouteMetric1	ipRouteNext Hop
9.1.2.3	3	99.0.0.3
10.0.0.51	5	89.1.1.42
10.0.0.99	5	89.1.1.42

检索表对象

- GetNextRequest (ipRouteDest. 10.0.0.99, ipRouteMetric1. 10.0.0.99, ipRoutcNextHop. 10.0.0.99)
- GetResponse (ipRouteDest.9.1.2.3 3, ipRouteMetric1.9. 1 .2.3=99.0.0.3, ipNetToMedialfindex.1.3 = 1)

IpRouteDest	ipRouteMetric1	ipRouteNext Hop
9.1.2.3	3	99.0.0.3
10.0.0.51	5	89.1.1.42
10.0.0.99	5	89.1.1.42

表的更新

- Set命令用于设置或更新变量的值
- Set命令的应答也是GetResponse，同样是原子性的。如果所有的变量都可以设置，则更新所有变量的值，并在应答的GetResponse中确认变量的新值：如果至少有一个变量的值不能设置，则所有变量的值都保持不变，并在错误状态中指明出错的原因。
- Set出错的原因
  - tooBig
  - noSuchName
  - badValue
  - genError

例4-6

- 改变列对象ipRouteMetric1的第一个值
- 可以发出命令
- SetRequest(ipRoutcMetric1.9. 1.2.3 =9)
- 得到的应答是
- GetResponse (ipRouteMetric1.9. 1.2.3=9)
- 其效果是该对象的值由3变成了9。

表的删除

- 如果要删除表中的行，可以把一个对象的值设为invalid.
- 例4-7

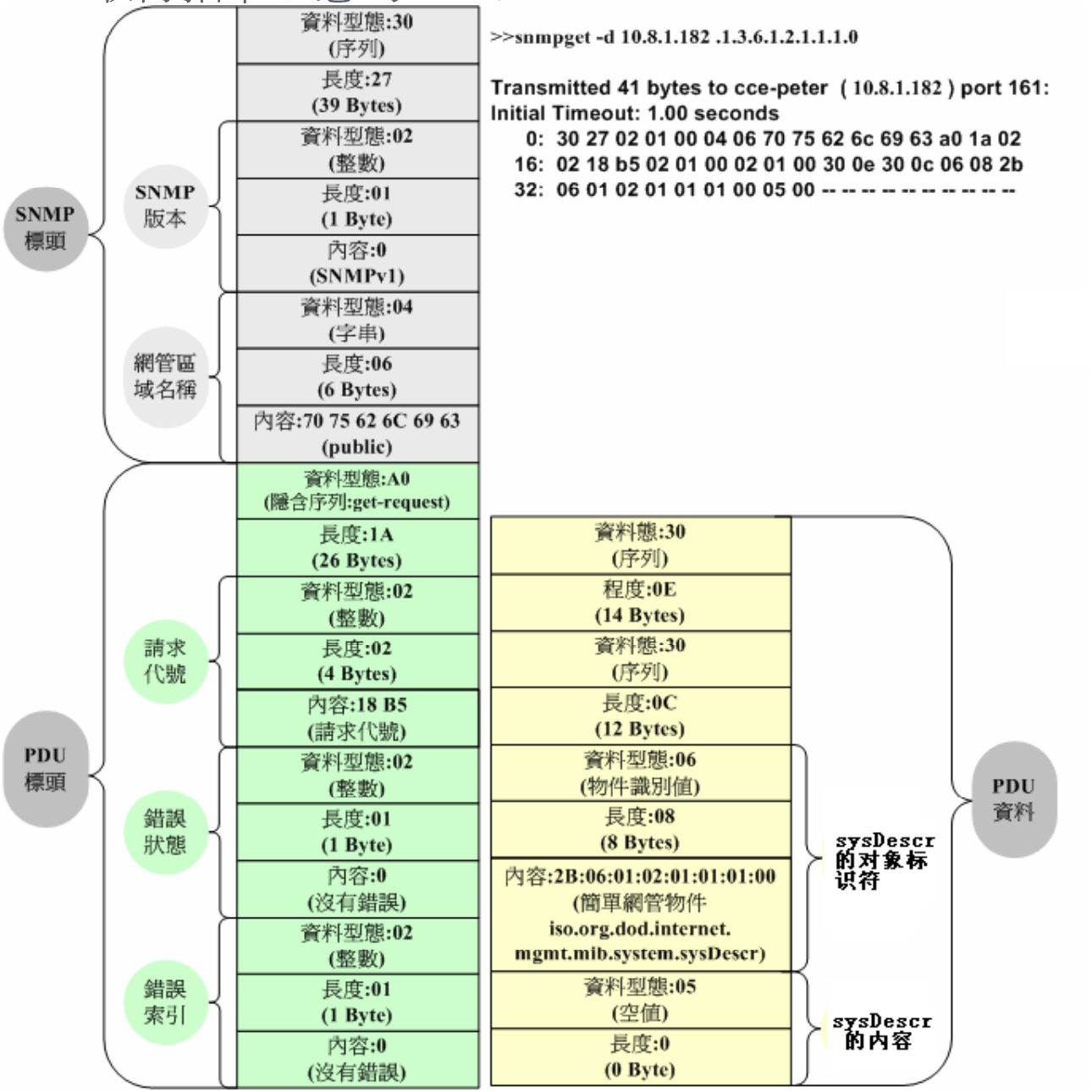
SetRequest(ipRouteType.211.10.10.10=invalid)

Get Response (ipRouteType.211.10.10.10=invalid)

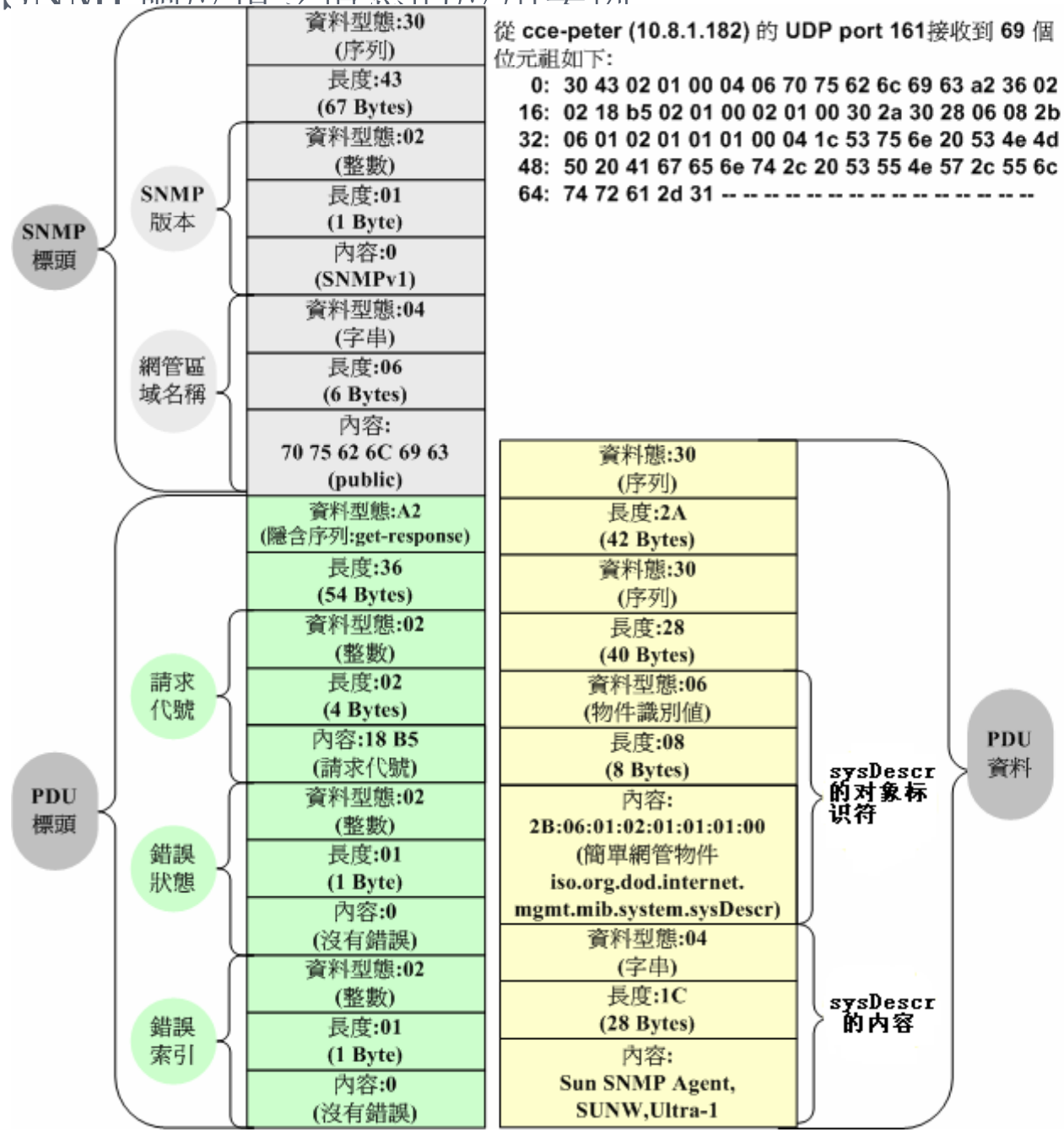
BER编码原则

BER编码代号	类型	BER编码代号	PDU类型
02	INTEGER	A0	GetRequest-PDU
04	OCTET STRING	A1	GetNextRequest-PDU
06	OBJECT IDENTIFIER	A2	GetResponse-PDU
05	NULL	A3	SetRequest-PDU
16	SEQUENCE/ SEQUENCE OF	A4	Trap-PDU

SNMP取得指令信息的应用实例



SNMP响应指令信息的应用实例



SNMPv1缺点

- 由于轮询的性能限制， SNMPv1不适合管理很大的网络。轮询产生的大量管理信息传送可能引起网络响应时间的增加。
- SNMPv1不适合检索大量的数据，例如检索整个表中的数据。
- SNMPv1的陷入报文时没有应答的，管理站是否收到陷入报文，代理不得而知。这样可能丢掉很重要得管理信息。
- SNMPv1只提供简单的团体名认证，这样的安全措施时很不够的。
- SNMPv1并不直接支持向被管理设备发送命令。
- SNMPv1的管理信息库MIB-2支持的管理对象是有限的，不足以完成复杂的管理功能。
- SNMPv1不支持管理站之间的通信，而这一点在分布式网络管理中是很需要的。

SNMPv2

SNMPv2 SMI对SNMPv1 SMI进行了扩充，提供了更严格的规范，规定了新管理对象和MIB文件，是SNMPvl SMI的超集。

SNMPv2相对于SNMPv1的改进：

(1) 加强了数据定义语言，改进了SMI，定义扩充了对象类型宏，增强对象表达能力，扩展了数据类型

(2) 提供了更完善的表操作能力，支持分布式管理

(3) 定义了新MIB功能组，丰富了故障处理能力

(4) 引入两种新PDU，用于大数据块的传送和管理者之间的通信。

SNMPv2 管理对象的语法和语义

```
OBJECT-TYPE MACRO ::=
BEGIN
   TYPE NOTATION ::="SYNTAX" Syntax
            UnitsPart
            "MAX-ACCESS" Access
            "STATUS" Status
            "DESCRIPTION" Text
            ReferPart
            IndexPart
            DefValPart
   VALUE NOTATION ::=value(VALUE ObjectName)
   Syntax ::= type(ObjectSyntax)
   UnitsPart ::= "UNITS" Text | empty
   Access ::="not-accessible"| "accessible-for-notify"| "read-only" |
   "read-write"| "read-create"
   Status ::="current" | "deprecated"| "obsolete"
   ReferPart ::="REFERENCE" Text| empty
   IndexPart ::="INDEX"    "{" IndexTypes "}" | "AUGMENTS" "{" Entry
   "}"| empty
... ...
```

SNMPv2中对象的定义

- SNMPv2的对象定义的变化

(1) 数据类型：

① 增加了两种数据类型Unsigned32和Counter64

② 规定计数器没有已定义的“初始值”

③ 计量器类型达到最大值时保持其最大值，并可随信息的减少而减少，计量器最大值可以设置为小于$2^{32}-1$

SNMPv2中对象的定义

(2) Units Part：增加了UNITS子句

这个子句用文字说明与对象有关的度量单位。比如时间，示例如下：　　UNITS　“seconds”

(3) MAX-ACCESS子句：去掉write-only，增加read-create和accessible-for-notify。

访问级别(由低到高)：

- not-accessible(不可访问)
- accessible-for-notify(通报访问，通报访问对象只有在网络管理器或其他代理进行通告时才有效，直接查询该对象是不允许的，这种访问方式与trap有关)
- read-only(只读)
- read-write(读写)
- read-create(读-创建)

SNMPv2中对象的定义

(4) STATUS子句：指明对象状态

SNMPv2标准去掉了SNMPvl中的optional和mandatory，只有3种可选状态。

① current:表示在当前的标准中是有效的

② obsolete:表示对象已经过时了，不必实现这种对象

③ deprecated:表示对象已经过时了，但是为了兼容旧版本实现互操作，实现时还要支持这种对象。

SNMPv2中表的操作

- 表的类型

(1) 禁止管理者进行行生成和行删除的表

这种表的最高的访问级别是read-write。在很多情况下这种表由代理控制，表中只包含read-ouly型的对象。

(2) 允许管理者进行行生成和行删除的表

这种表开始时可能没有行，由管理站生成和删除行。行数可由管理站或代理改变。

表的操作

1. 概念行的状态列

允许生成和删除行的表必须有一个列对象Rows status

(1) Acitve （可读写）：激活,可以使用概念行

(2) notInservice （可读写）:未被激活,概念行存在但不能使用

(3) notReady （只读）:不能被激活,概念行存在但没有信息不能使用

(4) createAndGo （只写不读）:生成并使用。管理站生成一个概念行实例时先设置成这种状态，生成过程结束时自动变为active，被管理设备就可以使用了。

(5) createAndWait （只写不读）:生成并等待。管理站生成一个概念行实例时先设置成这种状态，但不会自动变成active。

(6) Destroy （只写不读）：删除。管理站需删除所有的概念行实例时设置成这种状态。

表的操作

2. 概念行的挂起和删除

挂起：置状态列为noInservice

删除：置状态列为Destroy

SNMPv2管理信息库

- SNMPv2 MIB扩展和细化了MIB-2中定义的管理对象，又增加了新的管理对象。
- 1．系统组

  SNMPv2的系统组是MIB-2系统组的扩展

  SNMPv2在system中加入新的标量

(1) sysORTable：描述了一个SNMPv2实体对不同MIB模块的支持

- sysORIndex　　用做sysORTable的索引
- sysORID　　该表项的OID，类似sysObjectID对象
- sysORDescr　　对象资源的描述，类似sysDescr对象
- sysORUpTime　　包含该实例（该行）最后更新或实例化时， sysUpTime对象的值

(2) sysORLastChange：记录任何sysORID对象的实例最近一次变化时，sysUpTime对象的值

SNMPv2管理信息库

- 2．Snmp组

  Snmp组是由MIB-2的对应组改造而成的，有些对象被删除，同时又增加了一些新对象，新的Snmp组对象少了，去掉了许多对排错作用不大的变量。

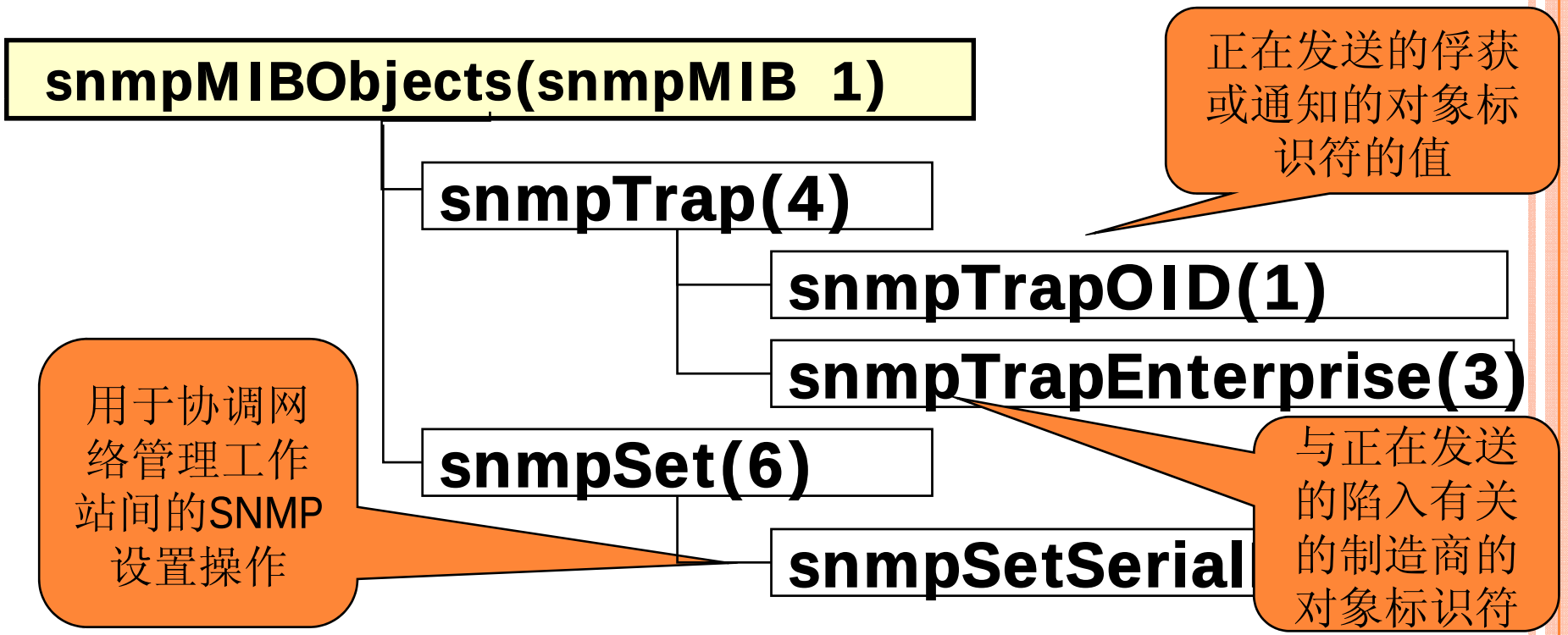
SNMPv2 管理信息库

Snmp(mib-2 11)

- snmpInPks(1)传输层服务提交给 SNMP 实体的报文数
- snmpInBadVersion(3)接收的含有版本错误的报文数
- snmpInBadCommunityNames(4)接收的含有团体名错误的报文数
- snmpInBadCommunityUses(5)含有不支持的团体操作的报文数
- snmpInASNParseErrs(6)含有 ASN 译码错误的报文数
- snmpEnableAuthenTraps(30)认证失效陷入工作（1），认证失效陷入不工作（2）
- snmpSilentDrops(31)由于响应报文太长无法应答而丢弃的请求报文总数
- snmpProxyDrops(32)由于向委托代理传送失败无法应答而丢弃的请求报文数

SNMPv2管理信息库

- 3．Mib对象组
- 　　这个新组包含的对象与管理对象的控制有关，分为两个子组。



- 说明：
- SnmpSerialNo的语法是TestAndIncr（约定为0—2147483647之间的一个整数），假定它的当前值是K，①如果代理收到的set操作SnmpSerialNo的值为K，则这个操作成功，响应PDU中返回K值，这个对象的新值增加为K+1（mod 2^31）；②如果代理收到一个set操作，置这个对象的值不等于K，则这个操作失败，返回错误值inconsistenvalue。

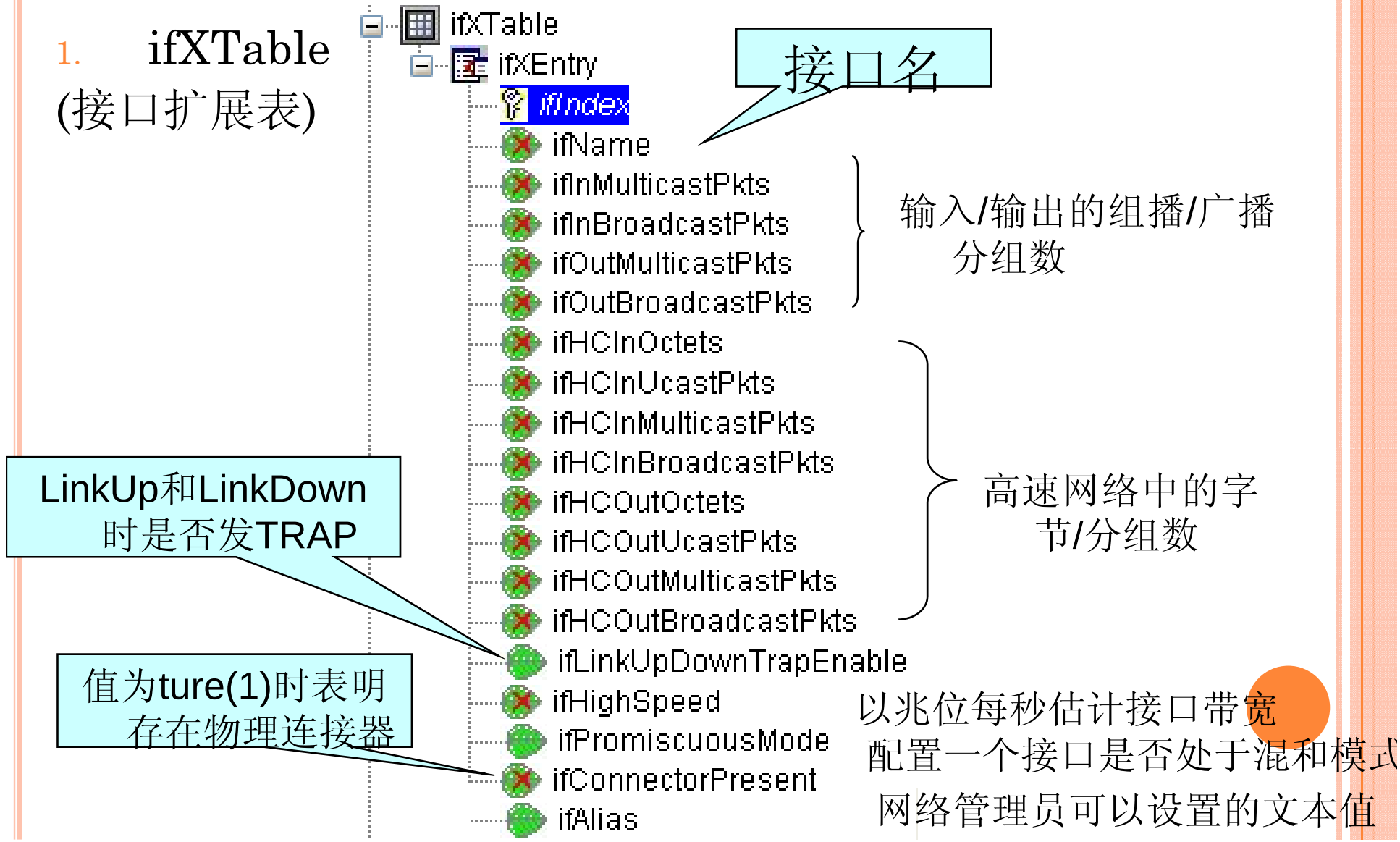
SNMPv2管理信息库

4. 接口组

- 新引入4个表
  - ifXTable
  - ifStackTable
  - ifTestTable
  - ifRcvAddressTable
- 改进的几个地方:
  - 广播包和多点发送包分开计数
  - 使用64位计数器替换32位计数器
  - 接口可以大于2.2G
  - 提供在物理接口之上映射虚拟接口的方法
  - 去掉ifInNucastPkts, ifOutNucastPkts, ifSpecific, ifOutQLen

4. 接口组

1. ifXTable (接口扩展表)



4．接口组

2. ifStackTable(接口堆栈表)

- 显示了网络接口的不同子层之间的关系。
- 包含关于哪些子层运行在其他哪些子层上的信息
- 相关对象

(1) ifStackHigherLayer：相应于关系的高子层的ifIndex值。

(2) ifStackLowerLayer：相应于关系的低子层的ifIndex值。

(3) ifStackStatus：RowStatus对象，可能是active、notInService 或destroy，用于在该表中添加/删除行

例：ifStackStatus.0.133:-->active(1)

4．接口组

3. ifTestTable(接口测试表)

作用：由管理站指示代理系统测试接口的故障。

（1）ifTestId：每个测试的唯一标识符

（2）ifTestStatus：测试是否正在进行，可以是notInUse和inUse

（3）ifTestType：测试类型，如noTest，test-to-run

（4）ifTestResult： none(1), success(2),inProgress(3), notSupported(4), unAbleToRun(5), aborted(6), failed(7)

（5）ifTestCode：测试结果代码

（6）ifTestOwner：管理站标识符

4．接口组

4. ifRcvAddressTable(接口地址表)

包含每个接口对应的各种地址：广播地址、多播地址和单播地址

（1）ifRcvAddressAddress：接口接收分组的地址

（2）ifRcvAddressStatus：RowStatus对象，用于行的删除和修改

（3）ifRcvAddressType：地址的类型，可能是other(1), volatile(2),nonVolatile(3)

SNMPv2的协议操作

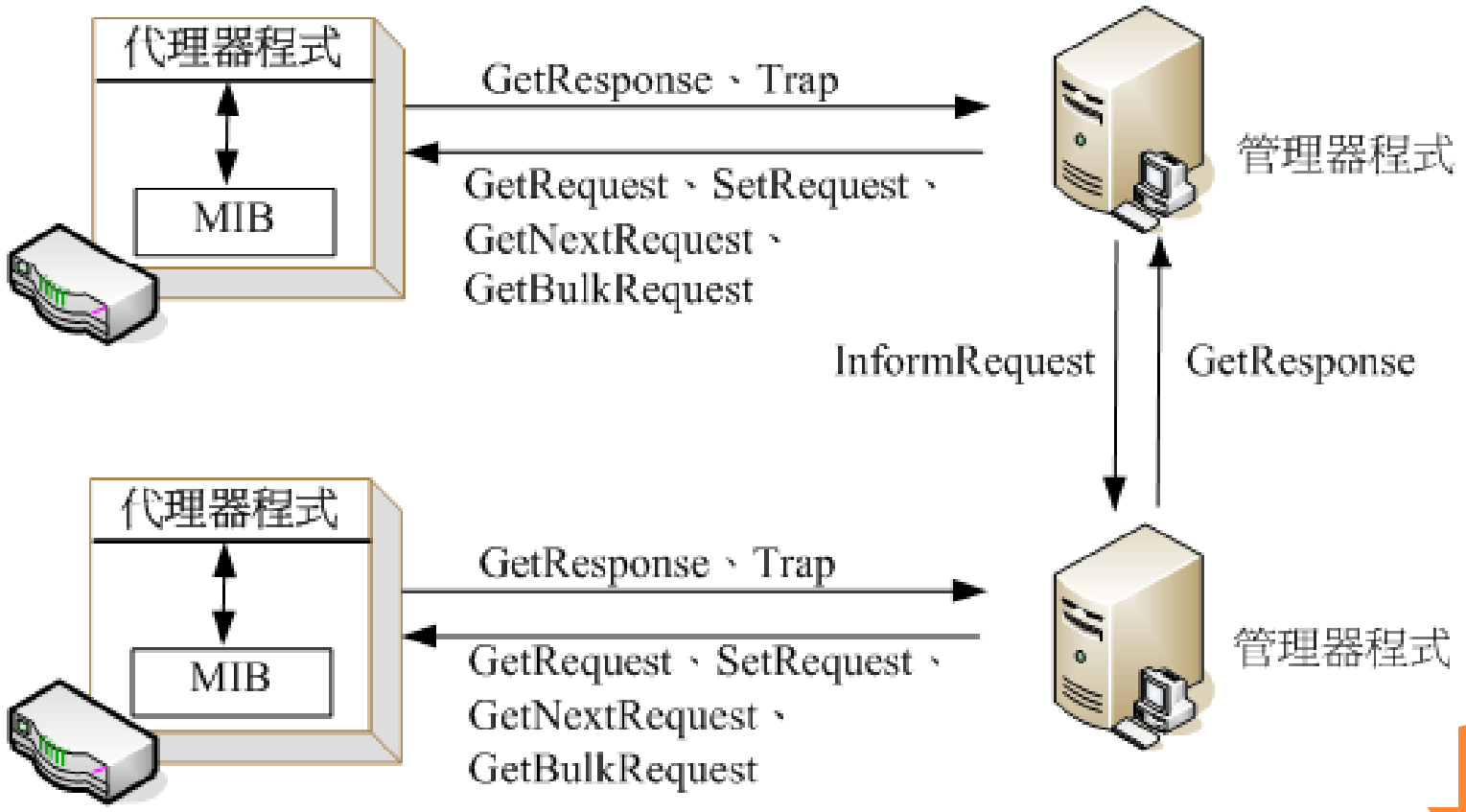
- SNMPv2访问管理信息的方法

(1) 管理者和代理之间请求/响应通信

(2) 管理者之间

(3) 代理向管理者发送陷阱报文

SNMPv2的协议操作



SNMPv2协议数据单元

1．SNMPv2报文

SNMPv2报文的结构分为3部分：版本号、团体名和作为数据传送的PDU。这个格式与SNMPvl一样。版本号取值0代表SNMPvl，取值1代表SNMPv2。团体名提供简单的认证功能，与SNMPvl的用法一样。

SNMPv2协议数据单元

SNMPv2实体发送报文一般要经过下面4个步骤。

（1）根据要实现的协议操作构造PDU。

（2）把PDU、源和目标端口地址以及团体名传送给认证服务，认证服务产生认证码或对数据进行加密，返回结果。

（3）加入版本号和团体名，构造报文。

（4）进行BER 编码，产生0/1比特串，发送出去。

SNMPv2协议数据单元

SNMPv2实体接收到一个报文后要完成下列动作。

（1）对报文进行语法检查，丢弃出错的报文。

（2）把PDU部分、源和目标端口号交给认证服务。如果认证失败，发送一个陷入，丢弃报文。

（3）如果认证通过，则PDU转换成ASN.1 形式。

（4）协议实体对PDU做句法检查，如果通过，根据团体名和适当的访问策略作相应的处理。

SNMPv2协议数据单元

SNMPv2 报文

(a) GetRequest、GetNextRequest、SetRequest、InformRequest 和 Trap

PDU 类型	请求标识	0	0	变量绑定表

(b) ResponsePDU

PDU 类型	请求标识	错误标志	错误索引	变量绑定表

(c) GetBulkRequest PDU

PDU 类型	请求标识	非重复数 N	最大后继数 M	变量绑定表

SNMPv2的协议操作

1. SNMPv2 Traps

0xA7	请求标识符	0	0	变量绑定表

SNMPv2 traps的特点：

(1) 关于trap的信息被嵌入在变量绑定中

(2) SNMPv1定义的通用trap具有与SNMPv2定义的对象标识符相同的值

例：linkDown trap具有对象标识符snmpTraps.3，则linkDown trap的变量绑定为

snmpTrapOID

linkDown

ifIndex

ifDescr

ifType

Object ID: .1.3.6.1.6.3.1.1.4.1.0 Object ID: .1.3.6.1.6.3.1.1.5.3

Object ID: .1.3.6.1.2.1.2.2.1.1.1 INTEGER: 1

Object ID: .1.3.6.1.2.1.2.2.1.2.1 STRING: GigabitEthernet1/1

Object ID: .1.3.6.1.2.1.2.2.1.3.1 INTEGER: 6

Object ID: .1.3.6.1.4.1.9.2.2.1.1.20.1 STRING: administratively down

SNMPv2的协议操作

2. InformRequest

- Inform：仅仅是一个需要得到响应的SNMPv2 Trap。
- InformRequest-PDU格式

0xA6	请求标识符	0	0	变量绑定表

引入inform是为了解决trap丢失的问题，但是它不能完全解决问题，还可能使问题恶化。

SNMPv2的协议操作

3. Get-Bulk

在概念上与通过重复的get-next命令遍历一个表的逻辑类似。

优点：改善了性能，当响应太大而不能在一条消息中容纳时，它将被截断以发送尽可能大的消息。

作用：允许用户获取表的一部分

PDU格式

0xA5	请求标识符	n	m	变量绑定表

n是non-repeaters，m是max-repetitions

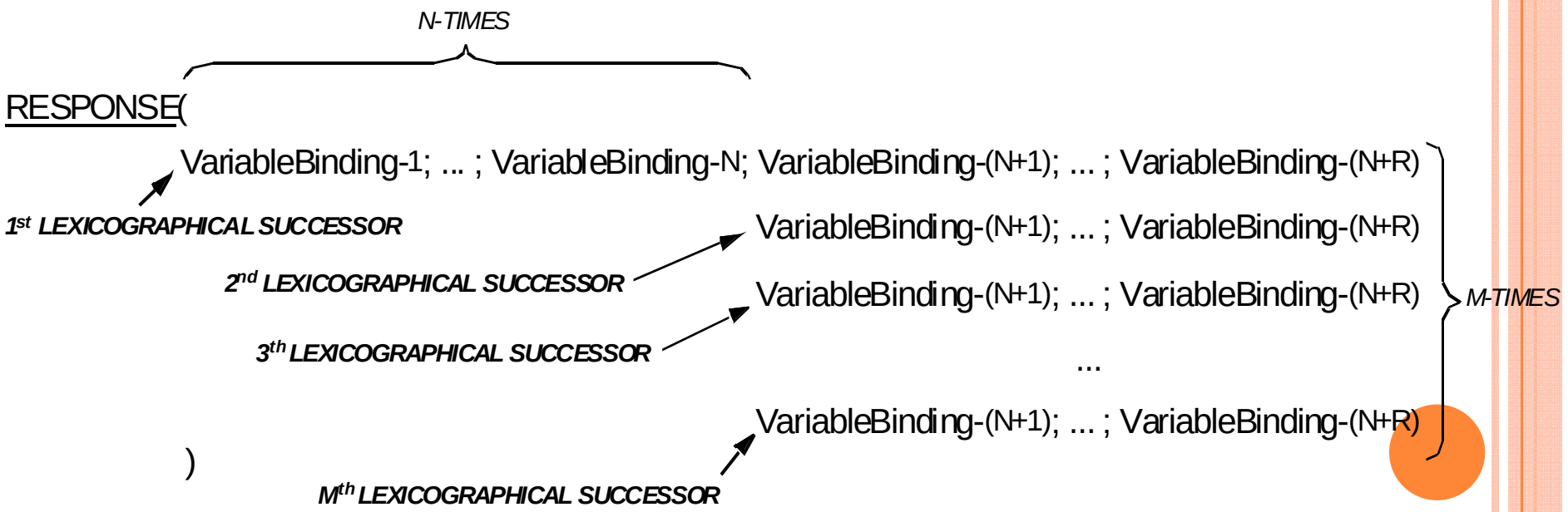
SNMPv2的协议操作

3. Get-Bulk

REQUEST(non-repeaters = N; max-repetitions = M;

VariableBinding-1; ... ; VariableBinding-N; VariableBinding-(N+1); ... ; VariableBinding-(N+R)

)



SNMPv2的协议操作

例：

getBulkRequest{non-repeaters=1,max-repetitions=3,
varbindlist={sysUpTime,ifInOctets,ifInErrors}}

响应：

sysUpTime.0 12432
ifInOctets.1 34543
ifInErrors.1 11
ifInOctets.2 222
ifInErrors.2 33
ifInOctets.3 44346
ifInErrors.3 0

SNMPv2的协议操作

4. 其他差错状态

(1) noAccesss(6)：试图设置一个不可访问的变量

(2) wrongType(7)：试图把一个变量设置成与它所要求类型不一致的值

(3) wrongLength(8)：试图把一个变量设置成与它所要求长度不一致的值

(4) wrongEncoding(9)：试图把一个变量设置成其ASN.1编码与该字段的ASN.1标记不一致的值

(5) wrongValue(10)：试图把一个变量设置成不正确的值

(6) noCreation(11)：试图修改或创建一个不存在并且不能创建的变量

(7) inconsistent Value(12)：试图把一个变量设置成与当前值不一致的值

SNMPv2的协议操作

(8) resourceUnavailable(13)：试图把一个变量设置成要求收集不存在的资源的值

(9) commitFailed(14)：一个set操作失败

(10) undoFailed(15)：一个set操作失败并且有些赋值不能恢复

(11) authorizationError(16)：get、get-next、set或inform请求没有通过认证

(12) notWritable(17)：试图修改一个当前存在但是不能被修改的变量

(13) inconsistentName(18)：试图修改一个当前不存在并且当前不能创建的变量

SNMPv2的协议操作

5. get与get-next的改进

- SNMPv1与SNMPv2的get和get-next的区别：当指定的变量之一无法获取时，响应的处理方式不同。

例：发送对象{ifIndex.5, ifDescr.20, protocolDistStatsPkts.1}的SNMPv2 get请求，该设备没有索引为20的接口或不支持RMONv2

ifIndex.5	5
ifDescr.20	noSuchInstance
protocolDistStatsPkts.1	noSuchObject

SNMPv3

SNMPv3特点：

(1) 适应性强

(2) 扩充性好

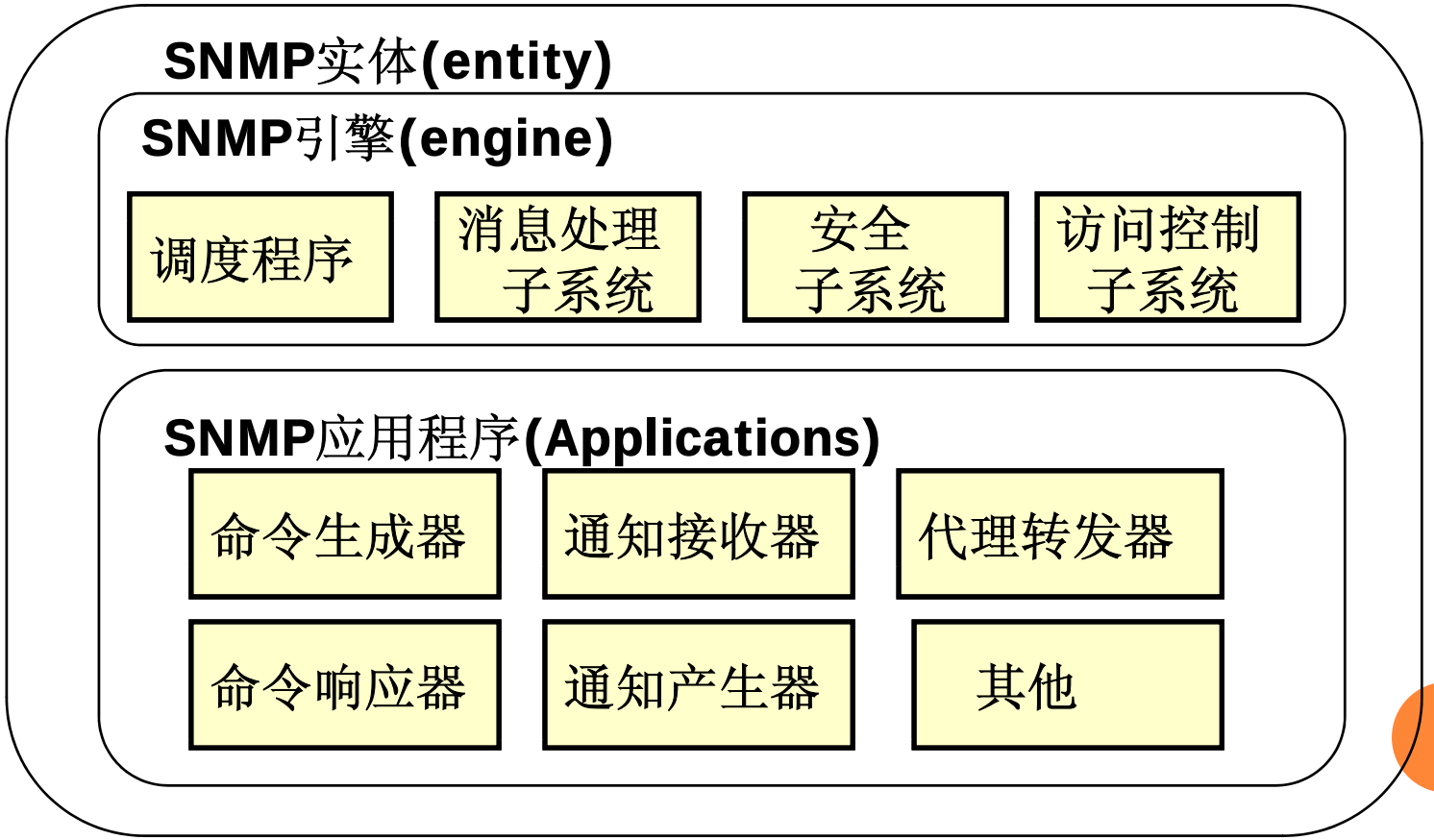
(3) 安全性好

SNMPv3体系结构

- SNMP实体：以前的SNMP代理和SNMP管理者的统称。
- 组成：SNMP引擎和SNMP应用程序

SNMPv3体系结构

- SNMP实体



SNMPv3体系结构

1. SNMP引擎

SNMP引擎提供三项服务：发送和接收报文；认证和加密报文；控制对管理对象的访问。

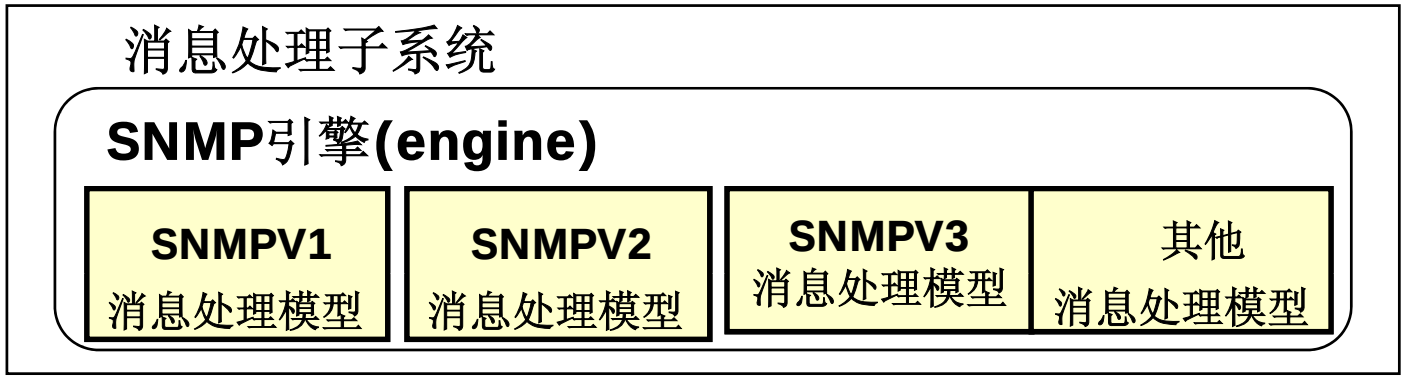
(1) 调度程序

负责发送和接收消息。确定SNMP报文的版本，并交给相应的报文处理模块处理。

(2) 消息处理子系统

由一个或多个消息处理模型组成。

功能：按照预定的格式准备要发送的消息或者从接收到的消息中提取数据。

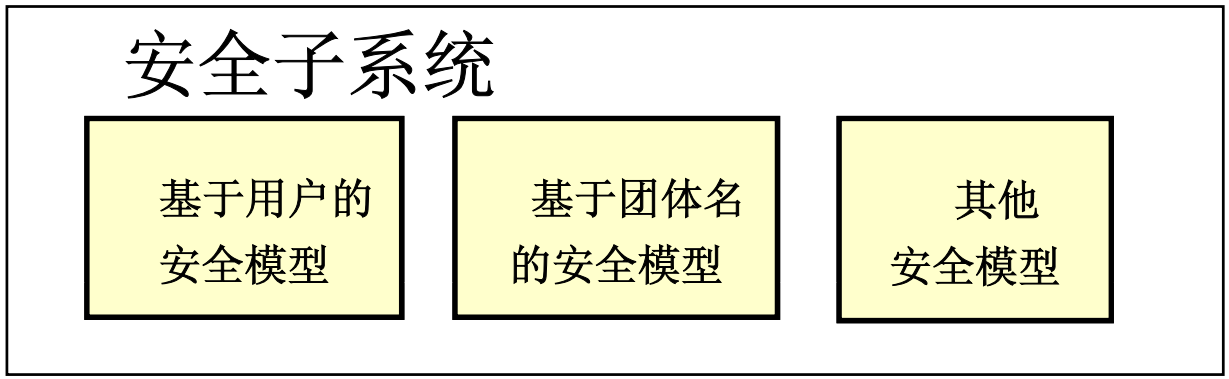


SNMPv3体系结构

1. SNMP引擎

(3) 安全子系统

提供验证消息和加/解密消息的安全服务



① 基于用户的安全模型：提供身份验证和数据保密服务

② 基于团体的安全模型

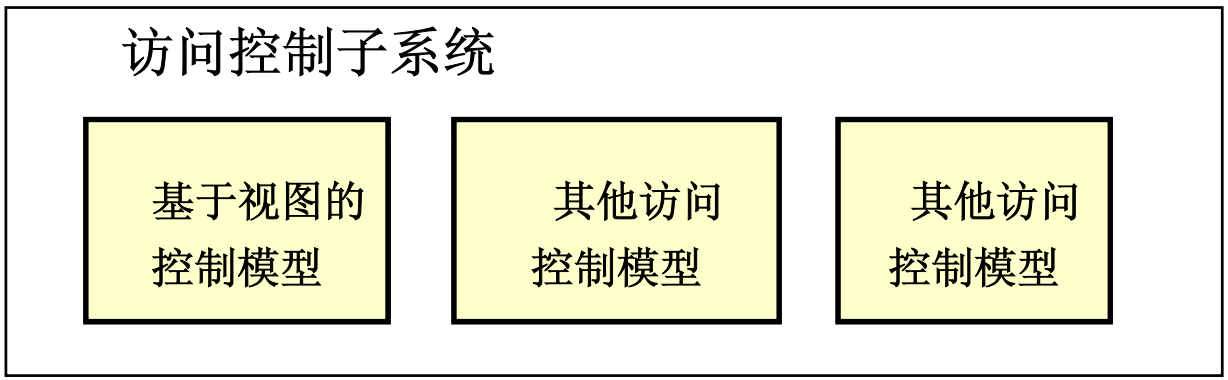
③ 其他安全模型：企业待定的；将来的标准

SNMPv3体系结构

1. SNMP引擎

(4) 访问控制子系统

确定是否允许访问管理对象，或者是否可以对某个管理对象实施特殊的管理操作



SNMPv3体系结构

1. SNMP引擎

(4) 访问控制子系统

用于确定是否允许访问管理对象：

① 当处理一个**SNMPGet，Get-Next，Get-Bulk**或**SetPDU**时调用它以确认在变量绑定中所指定的**MIB**对象允许访问。

② 当生成一个通知时调用它以确保在变量绑定中所指定的**MIB**对象允许访问。

SNMPv3体系结构

2. 应用程序

类型：

(1) 命令生成器:生成收集或设置管理数据的SNMP命令

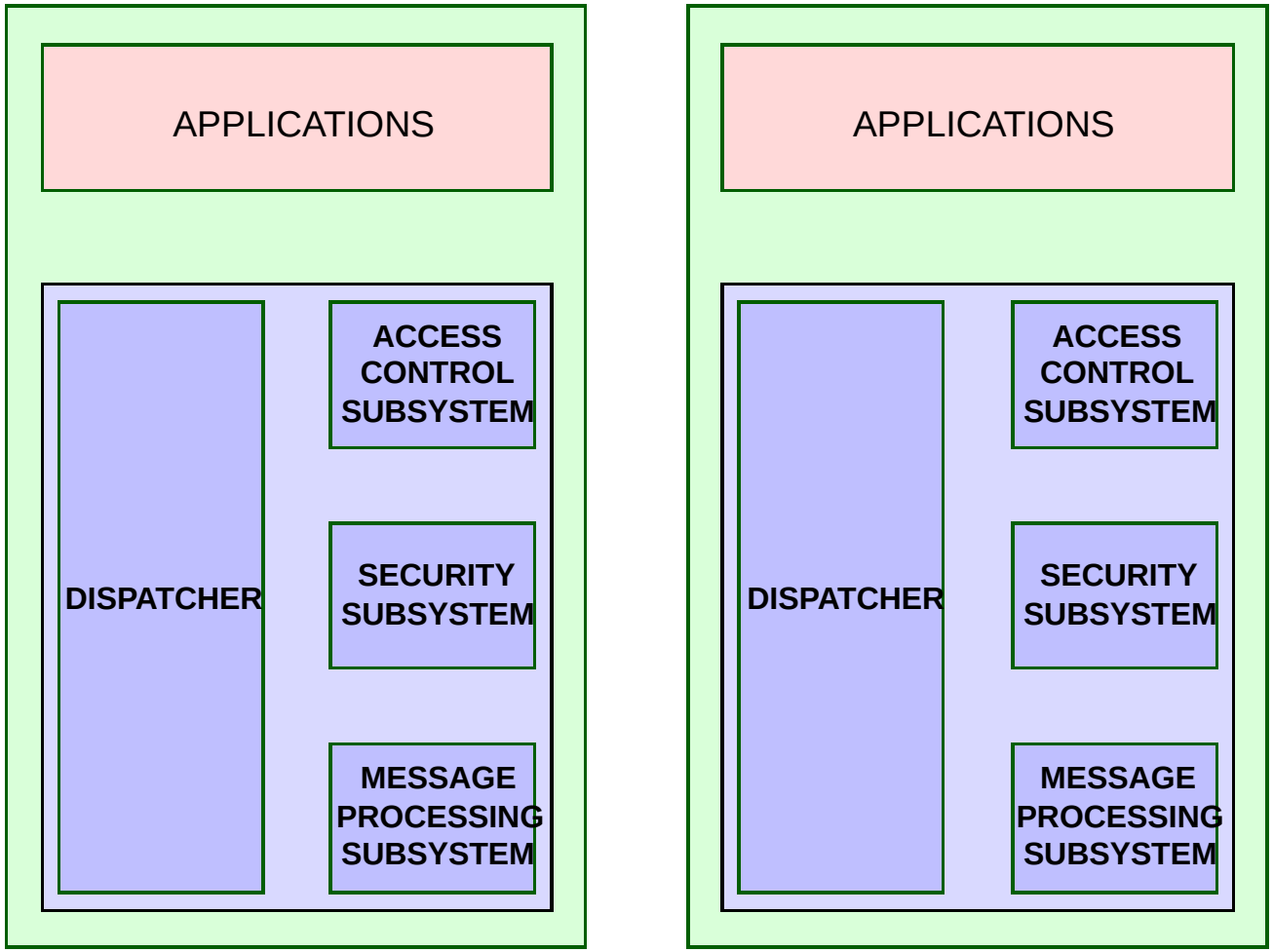
(2) 命令应答器:接收SNMP Read/Write请求，对管理数据进行访问，并按照协议规定的操作产生响应报文，返回给读/写命令的发送者。

(3) 通知产生器:监控系统中出现的特殊事件，产生Trap或Inform消息

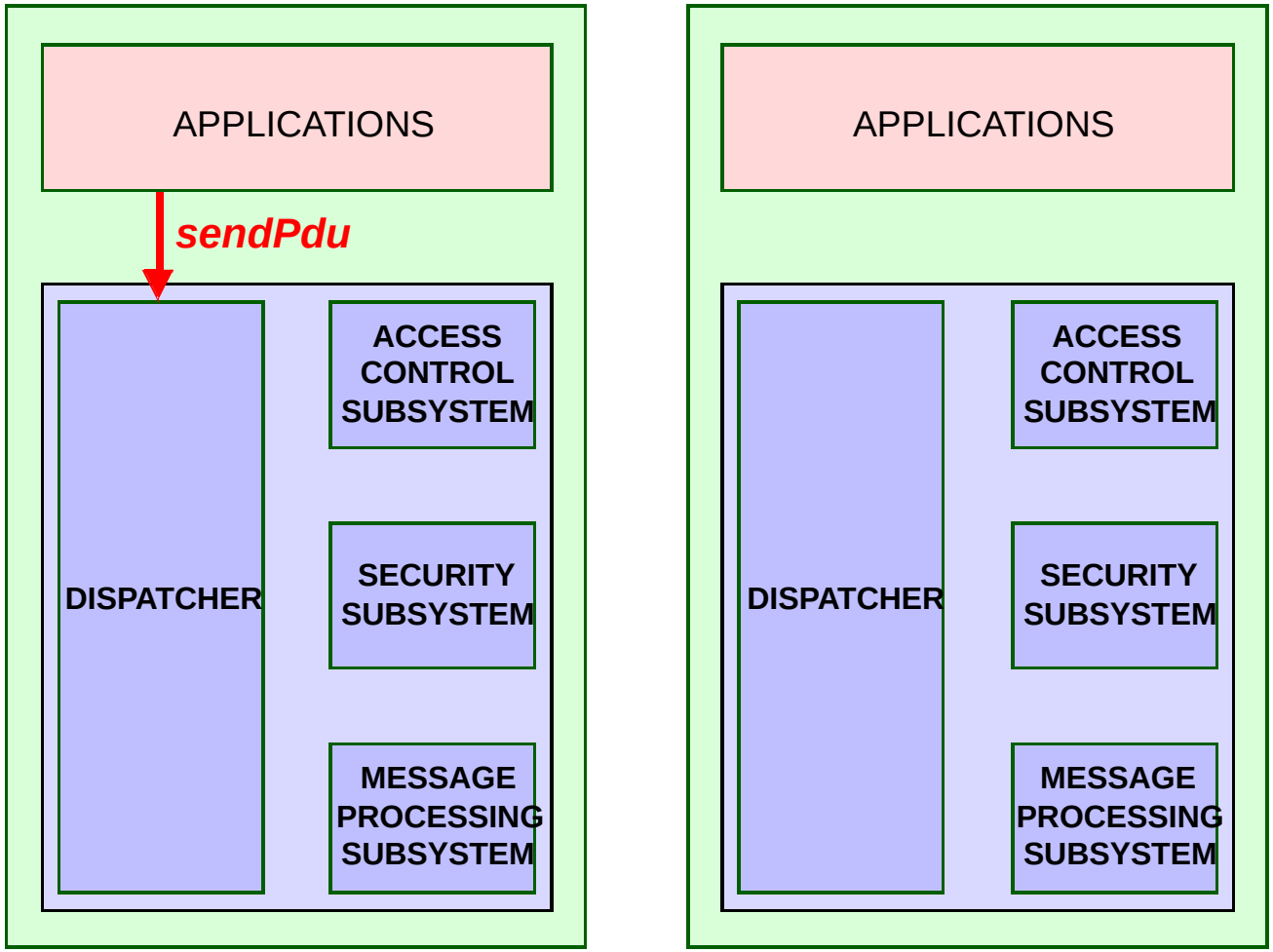
(4) 通知接收器:接收并处理Trap或Inform消息

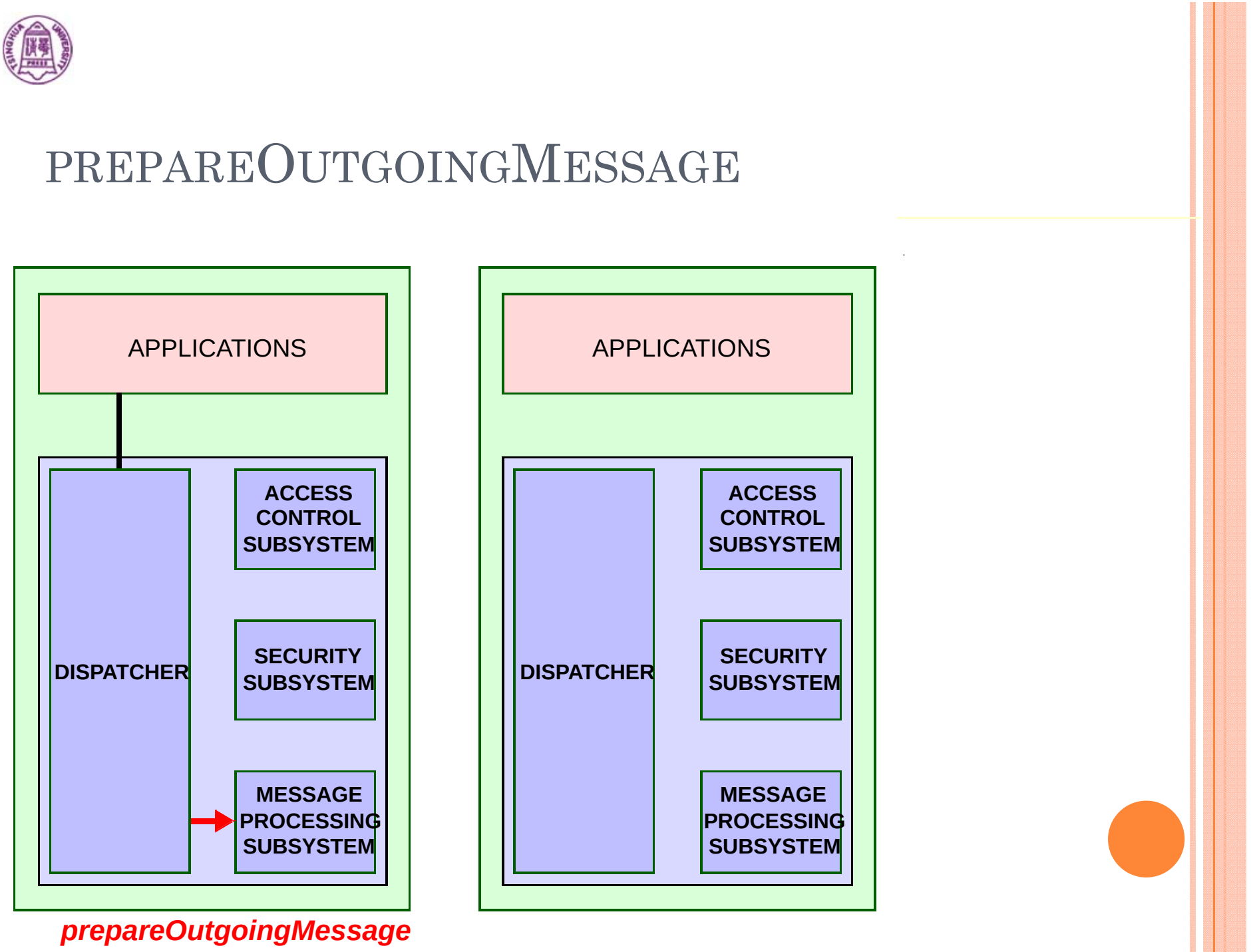
(5) 代理转发器:转发SNMP实体之间的消息

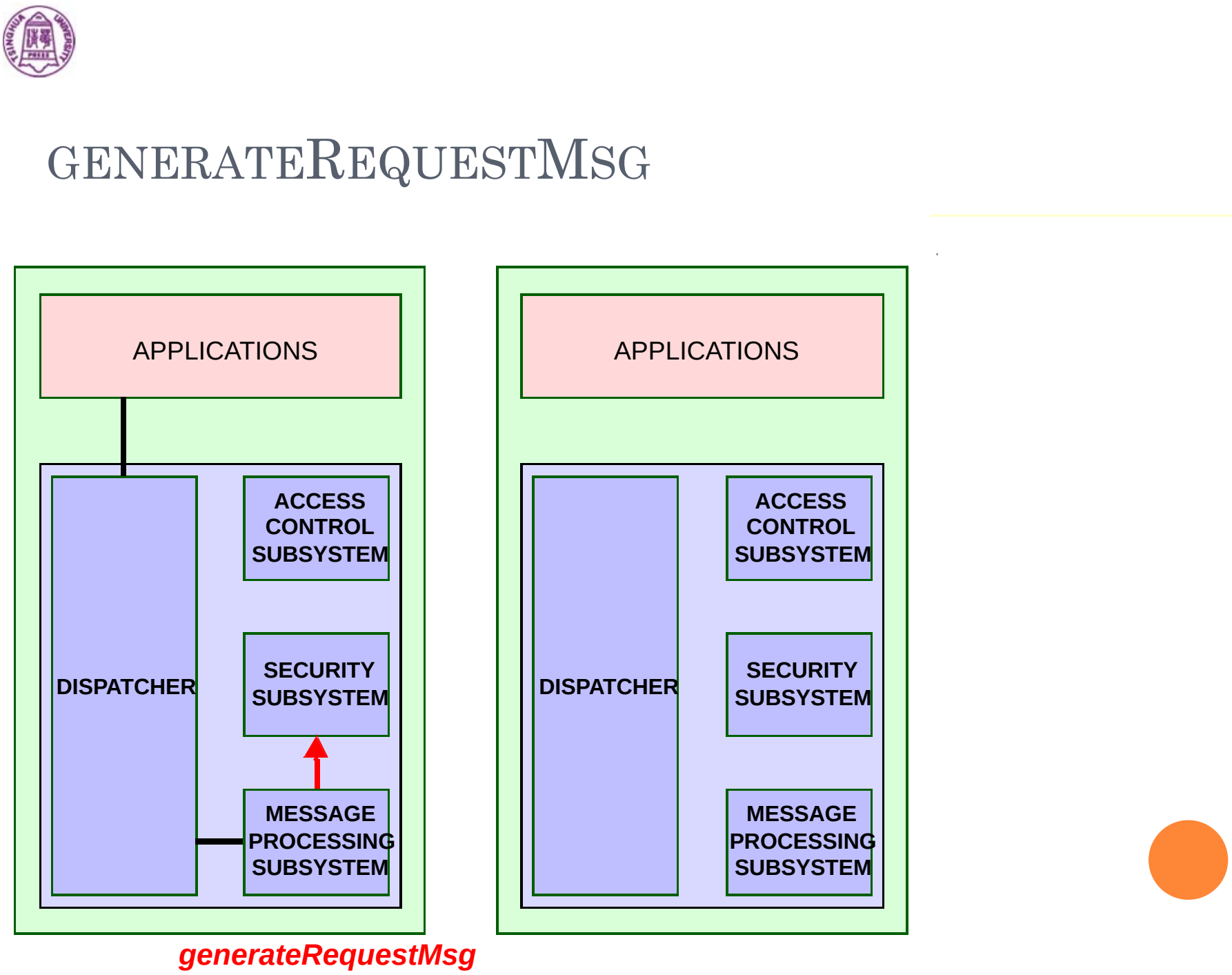
初始状态


APPLICATIONS
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM
APPLICATIONS
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM

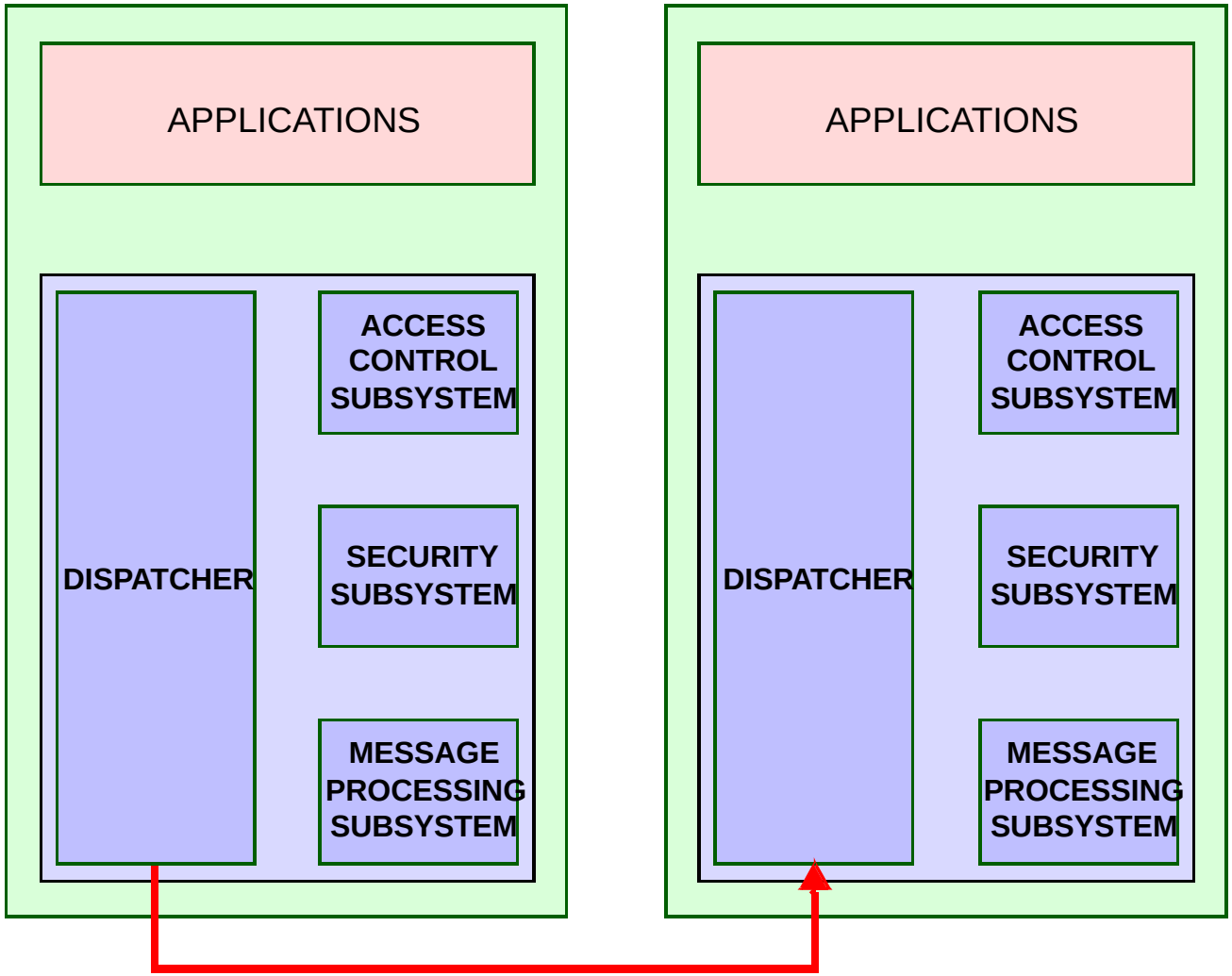
SENDPDU

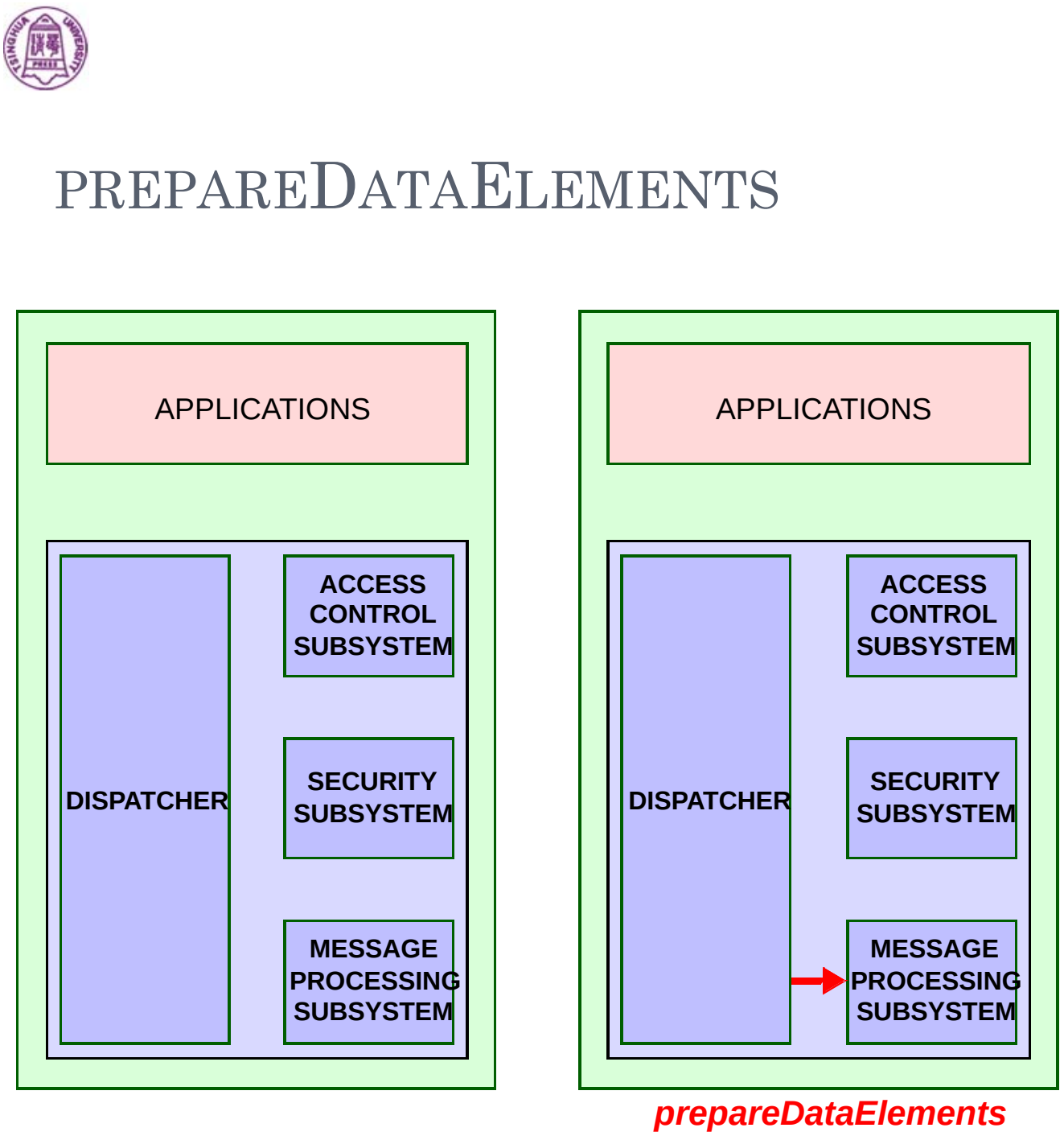

APPLICATIONS
sendPdu
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM
APPLICATIONS
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM


PREPAREOUTGOINGMESSAGE
APPLICATIONS
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM
APPLICATIONS
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM
prepareOutgoingMessage


GENERATEREQUESTMSG
APPLICATIONS
APPLICATIONS
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM
generateRequestMsg

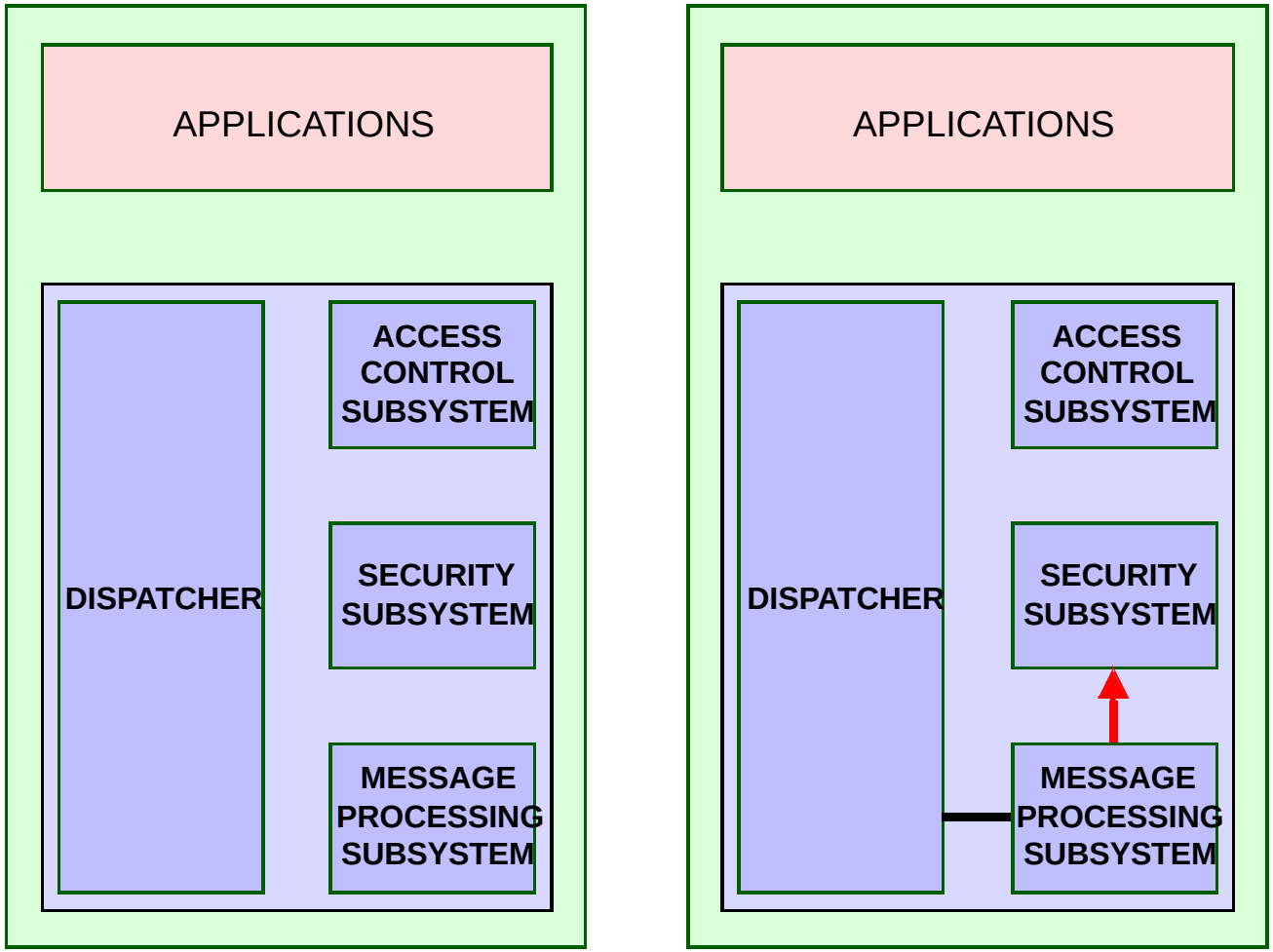
SEND / RECEIVE





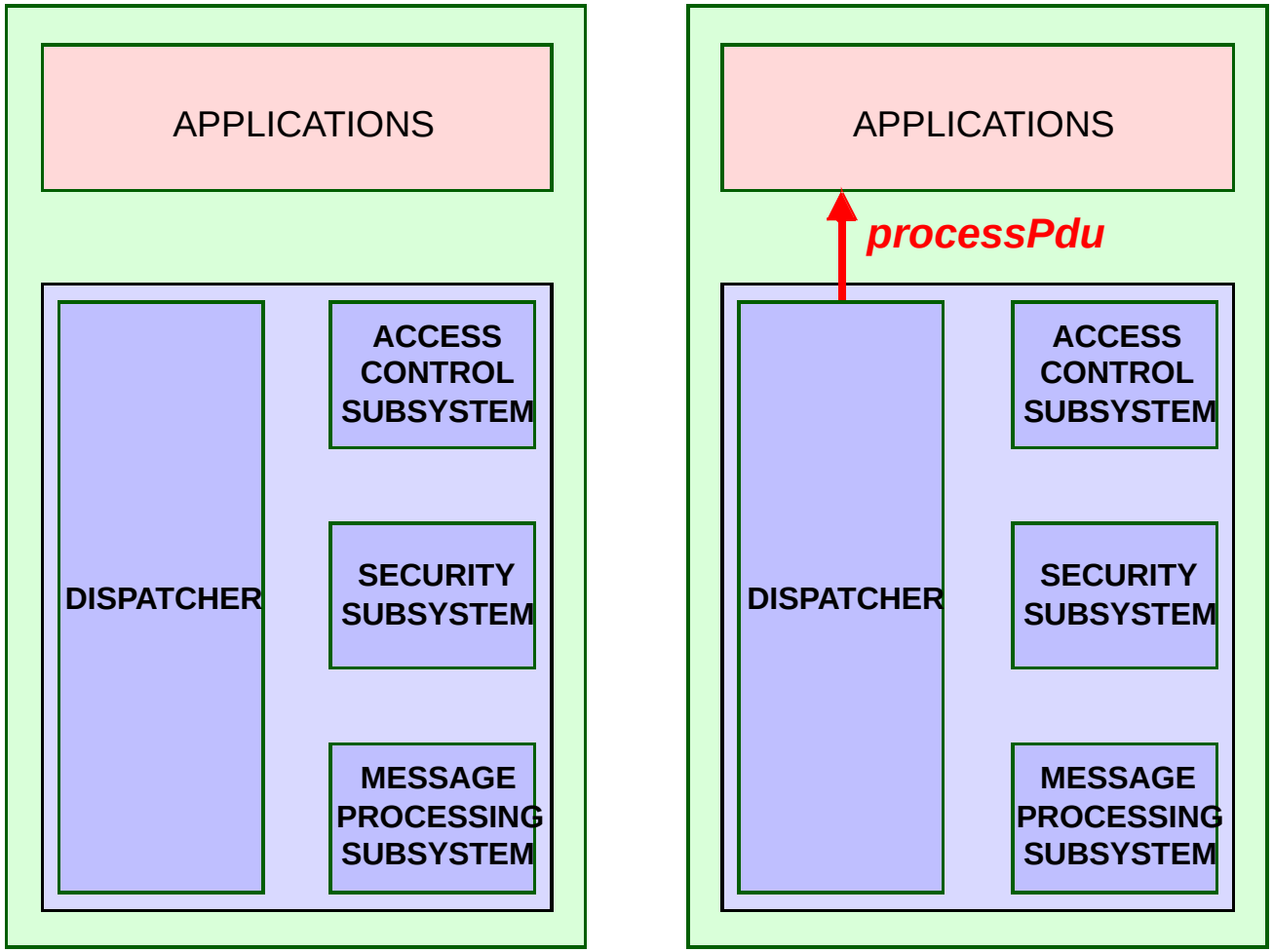
PREPAREDATAELEMENTS

PROCESSINCOMINGMSG

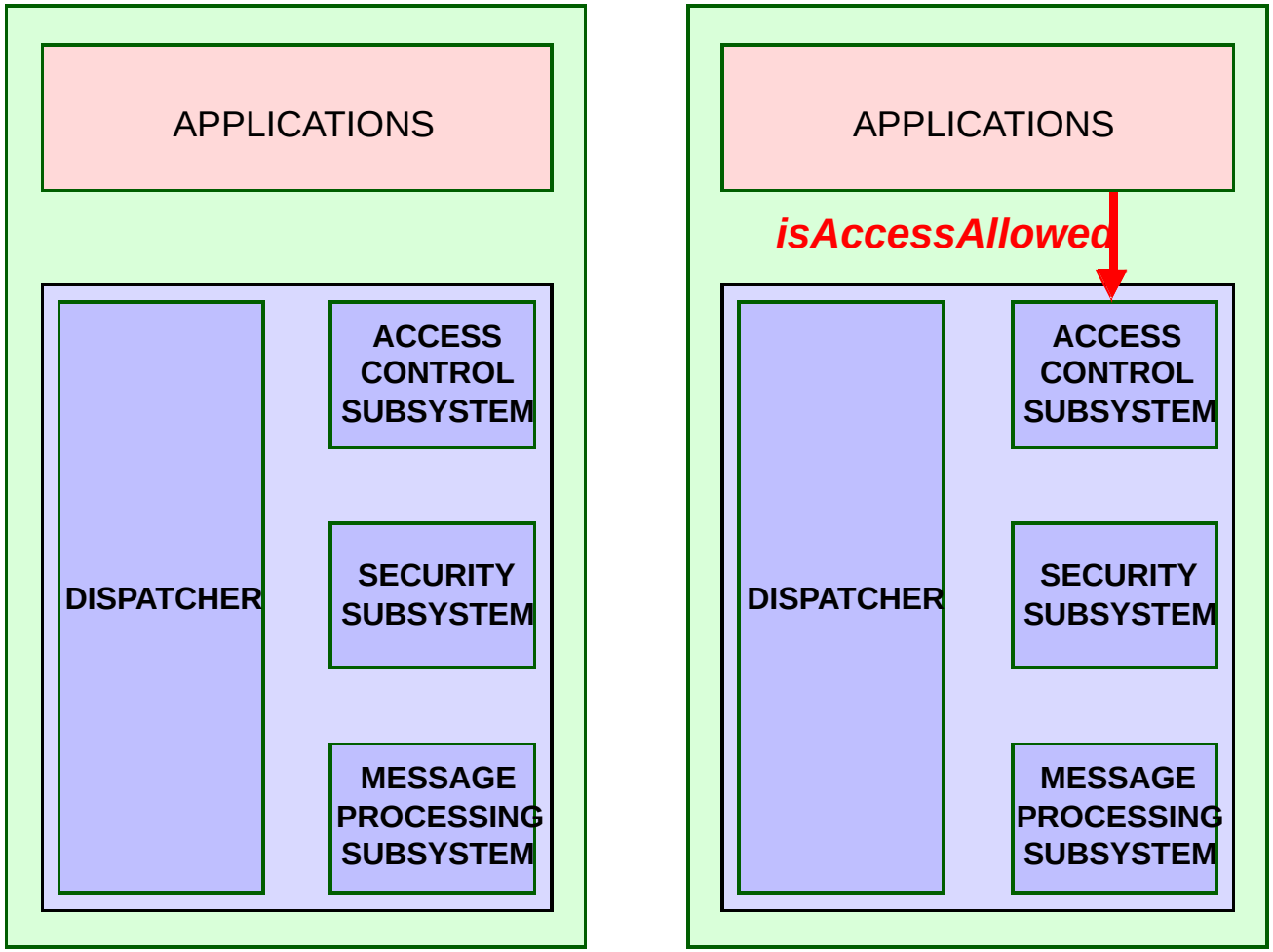


processIncomingMsg

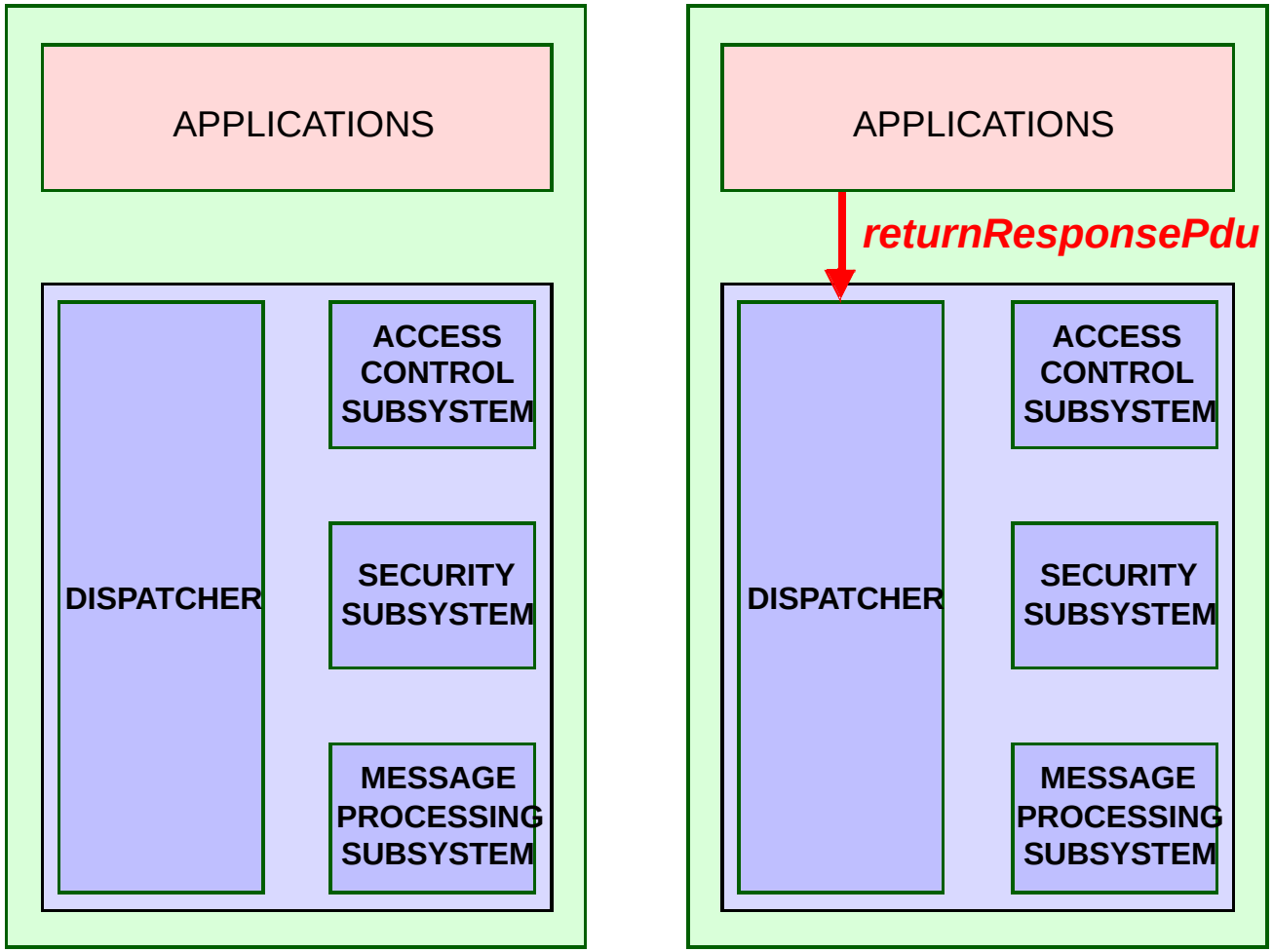
PROCESSPDU


APPLICATIONS
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM
APPLICATIONS
processPdu
DISPATCHER
ACCESS CONTROL SUBSYSTEM
SECURITY SUBSYSTEM
MESSAGE PROCESSING SUBSYSTEM

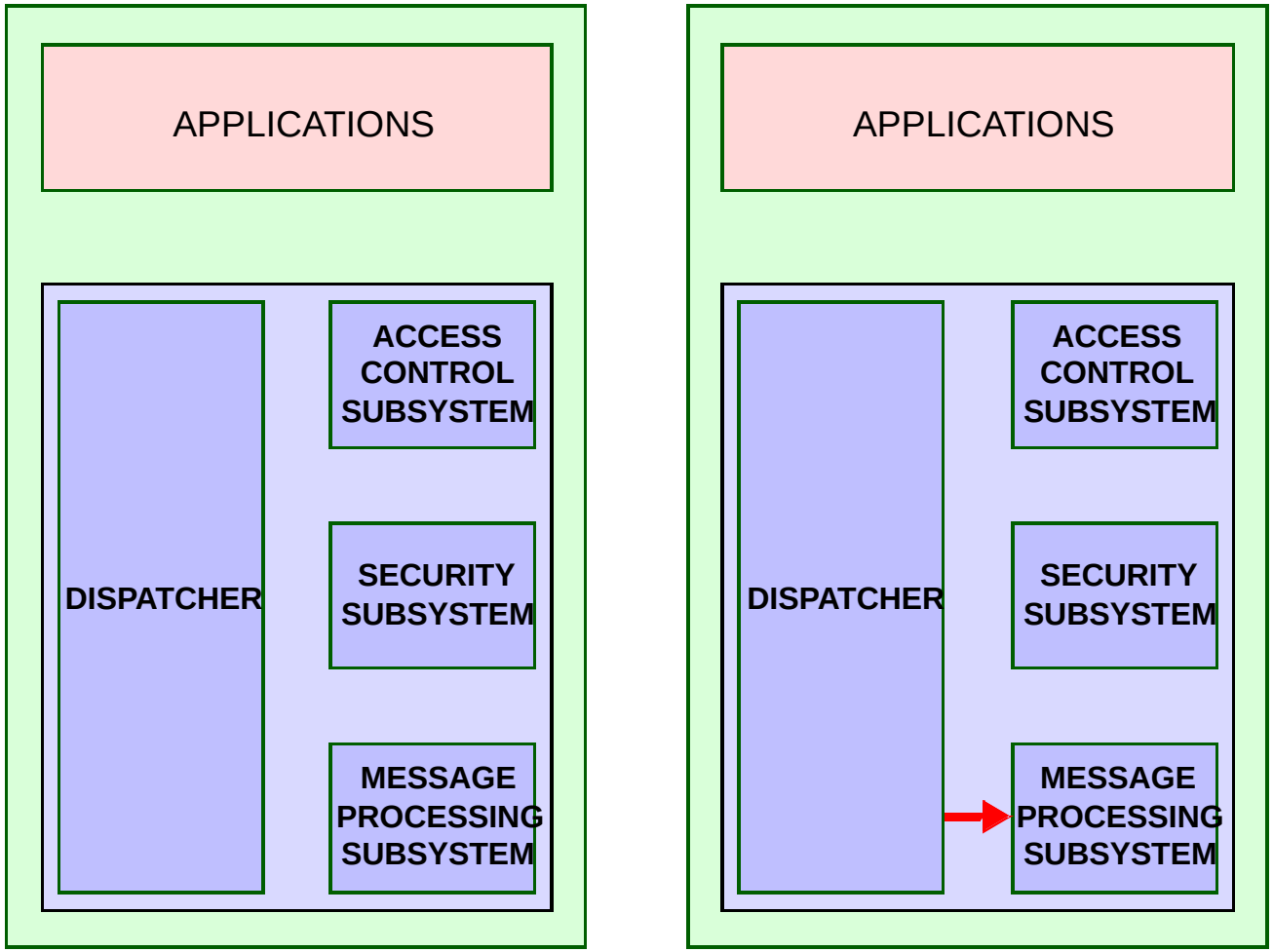
ISACCESSALLOWED

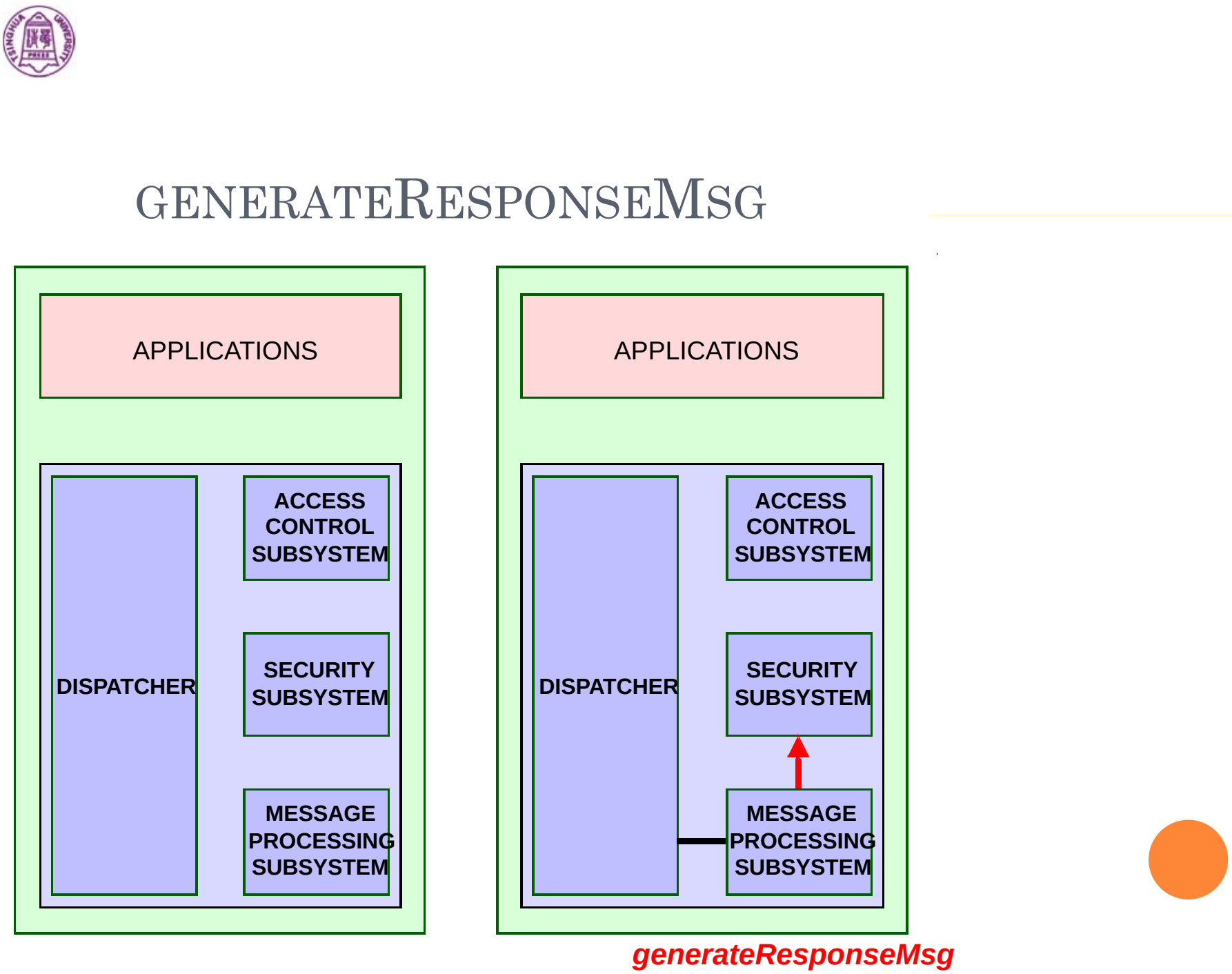


RETURNRESPONSEPDU

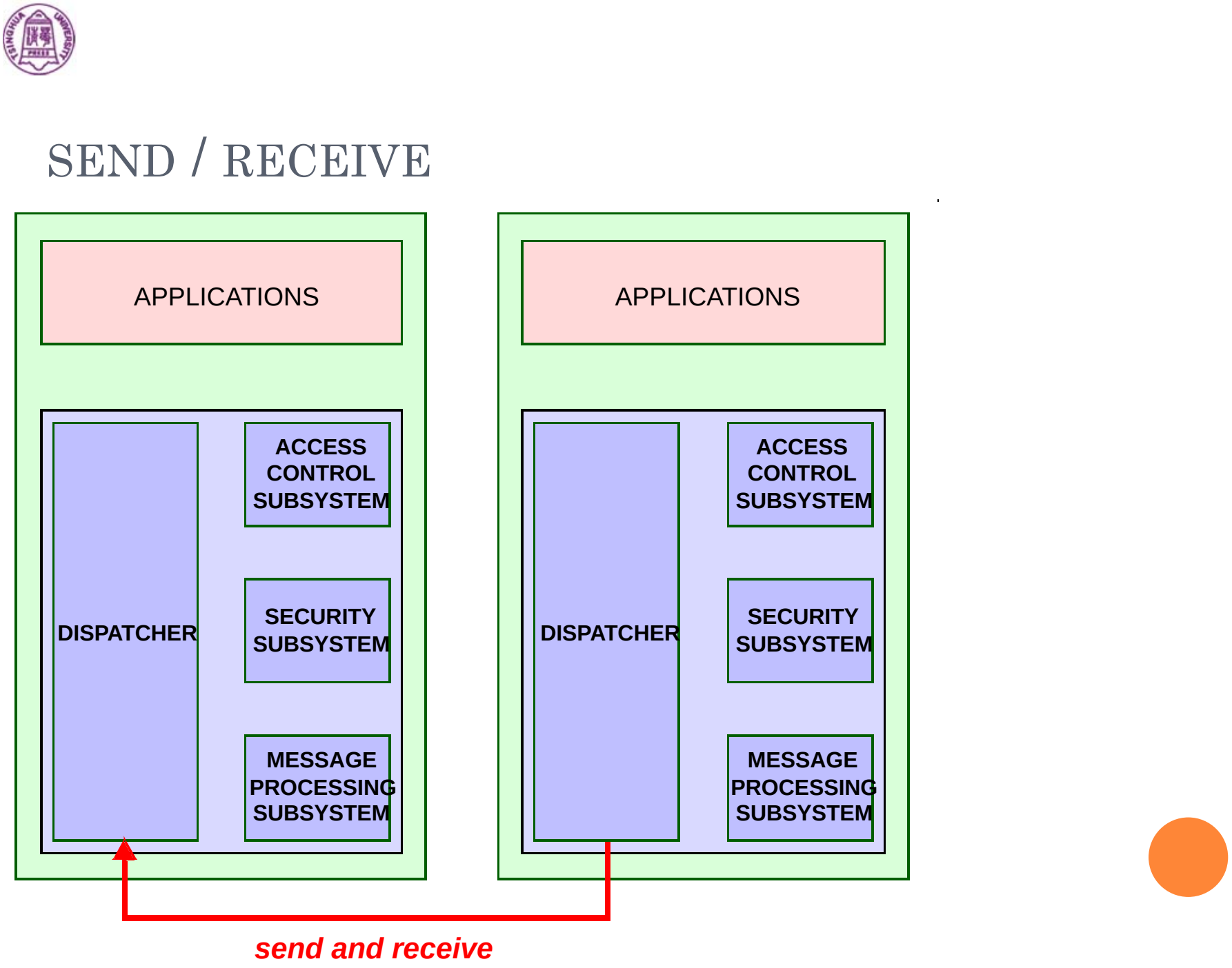


PREPARERESPONSEMESSAGE



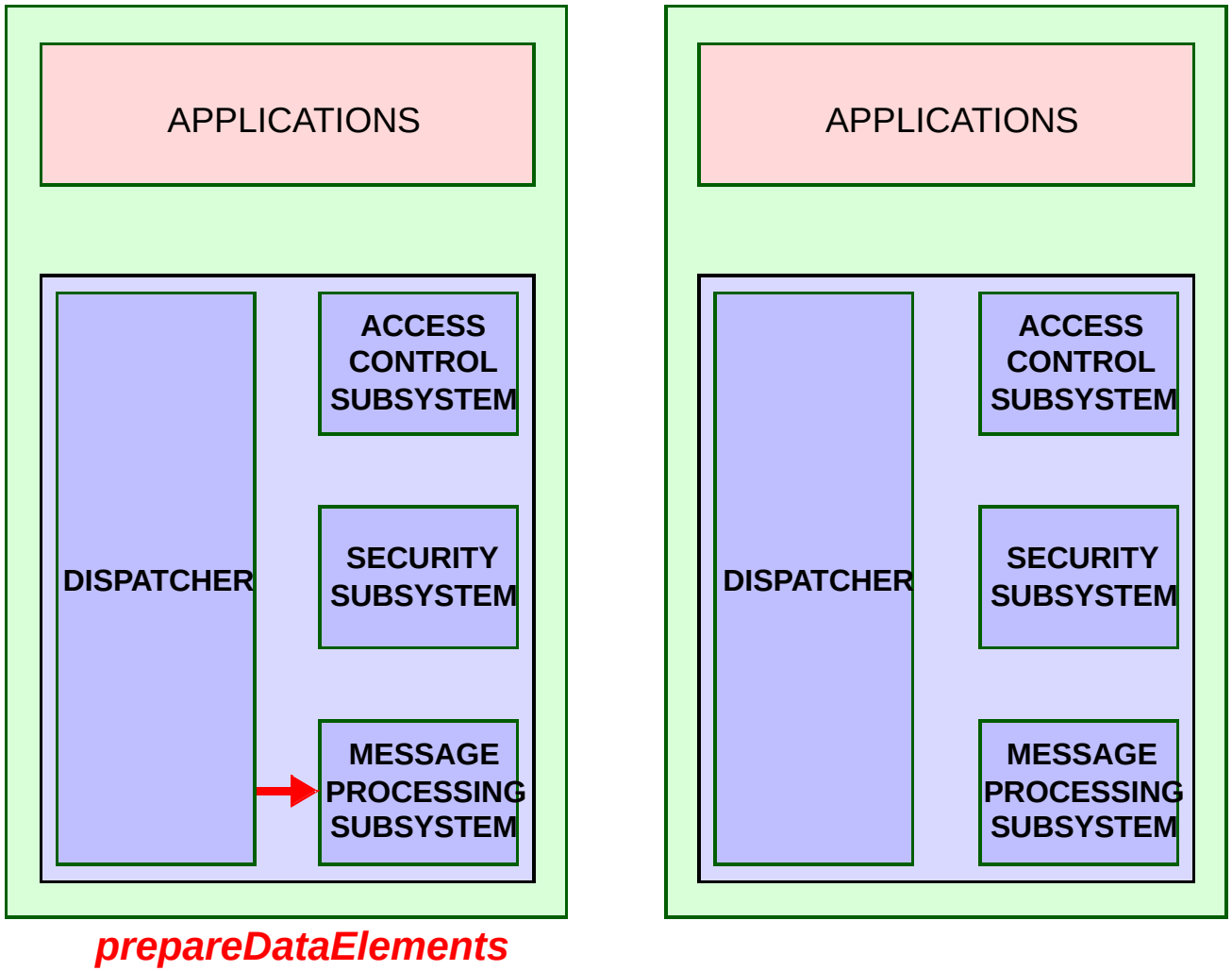


GENERATERESPONSEMSG



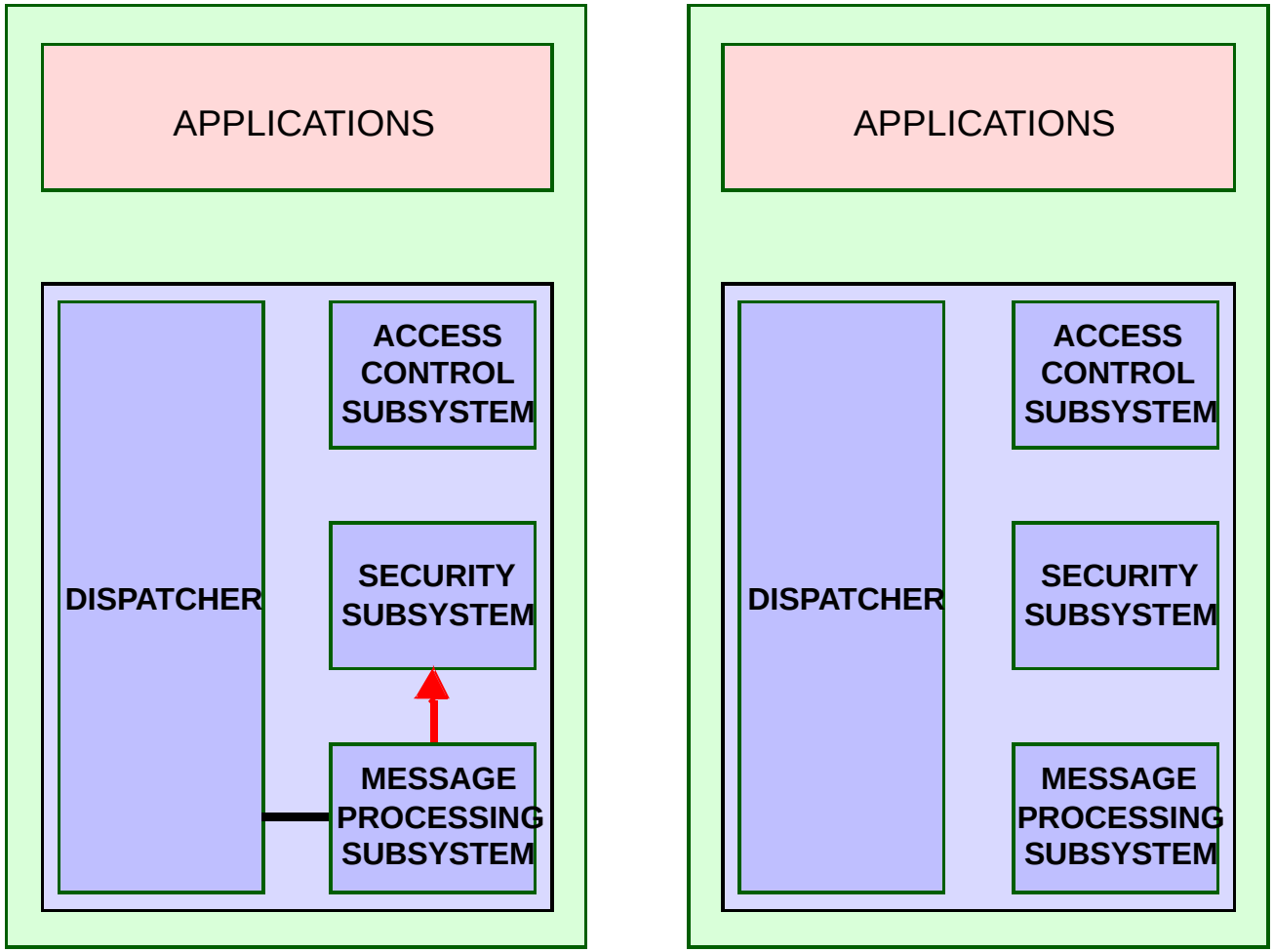
SEND / RECEIVE

PREPAREDATAELEMENTS



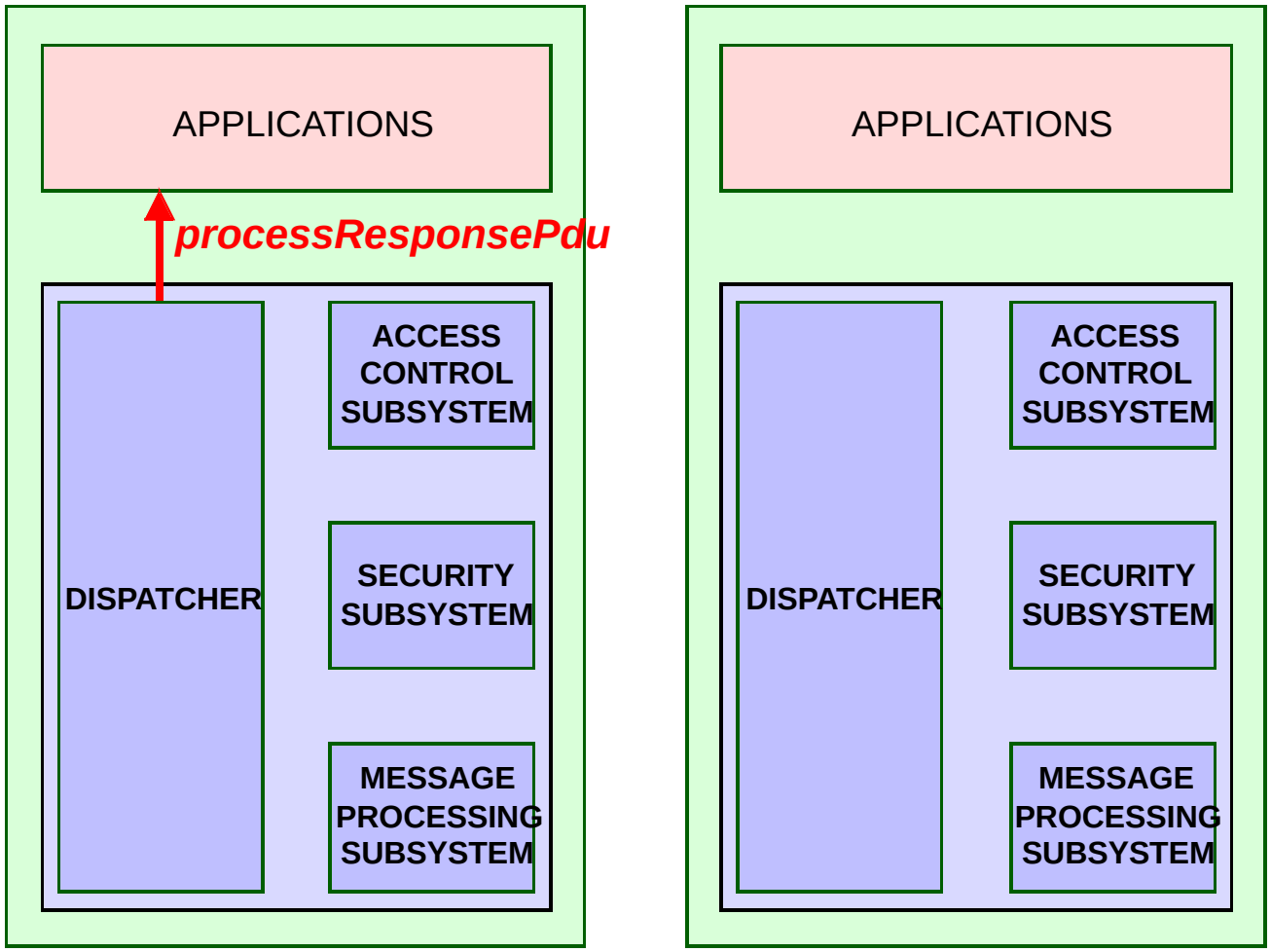
prepareDataElements

PROCESSINCOMINGMSG

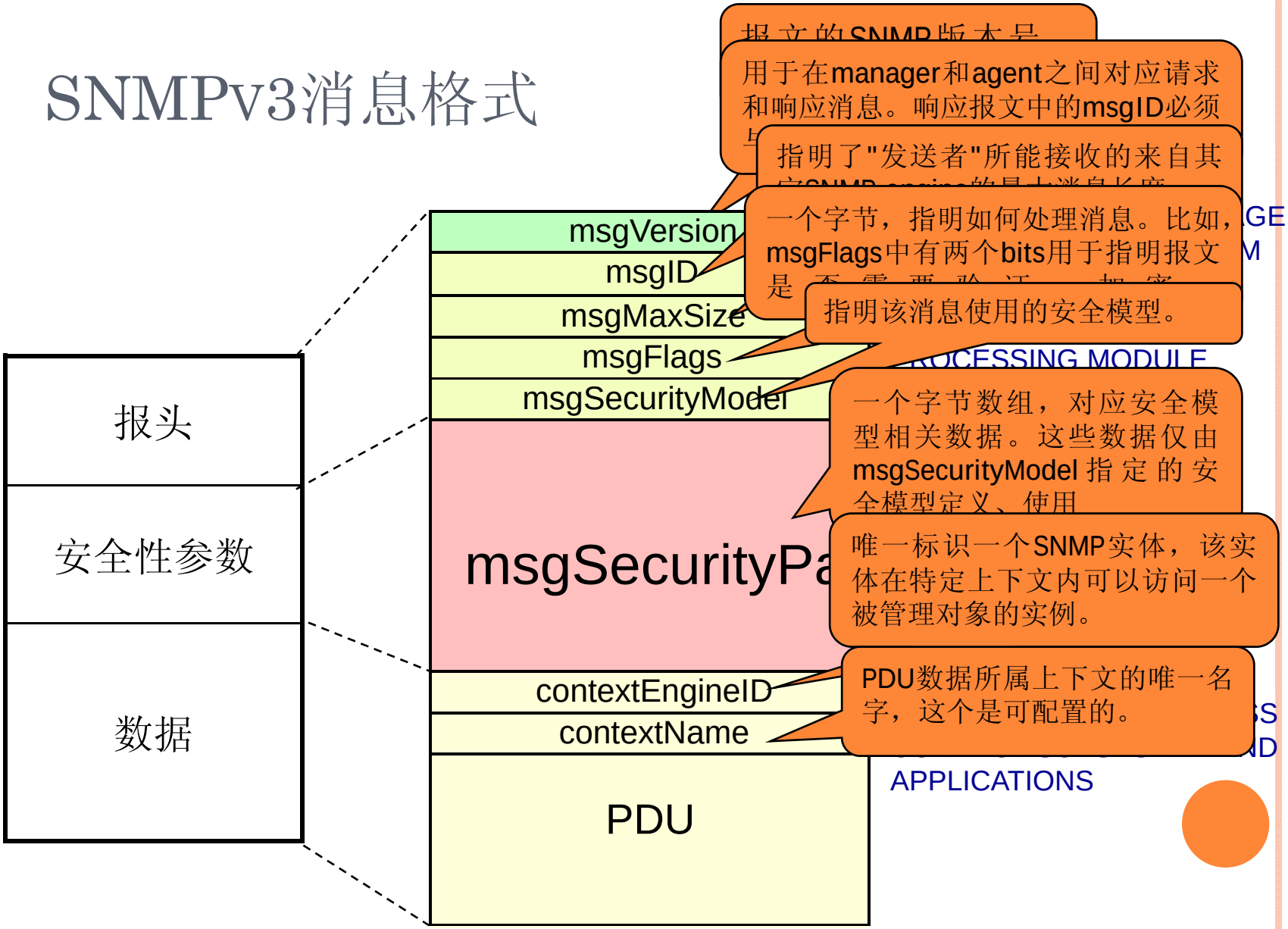


processIncomingMsg

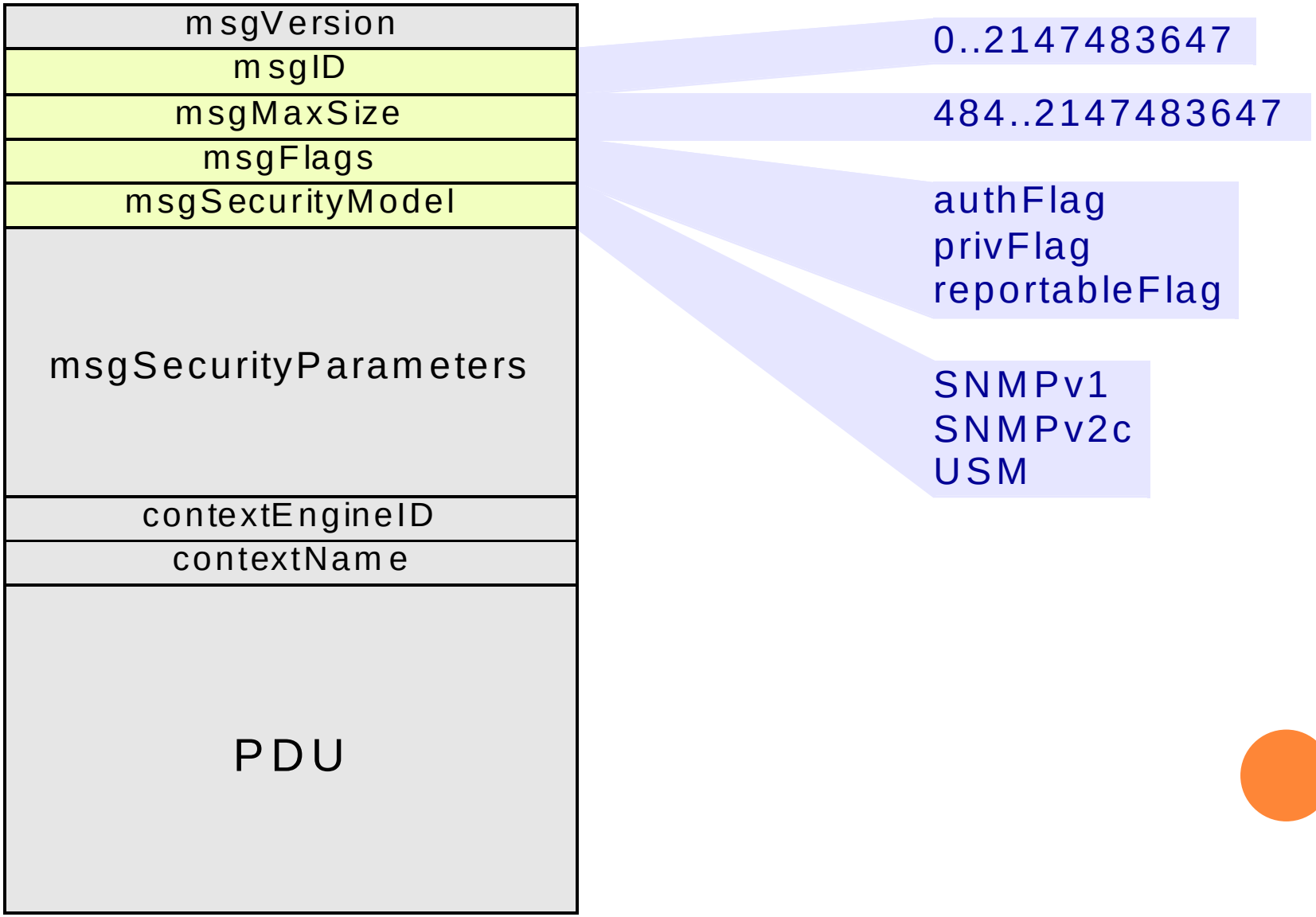
PROCESSRESPONSEPDU



SNMPv3消息格式


报头
安全性参数
数据
msgVersion
msgID
msgMaxSize
msgFlags
msgSecurityModel
msgSecurityPa
contextEngineID
contextName
PDU
用于在manager和agent之间对应请求和响应消息。响应报文中的msgID必须
指明了"发送者"所能接收的来自其
一个字节，指明如何处理消息。比如，msgFlags中有两个bits用于指明报文
指明该消息使用的安全模型。
一个字节数组，对应安全模型相关数据。这些数据仅由msgSecurityModel指定的安全模型定义、使用
唯一标识一个SNMP实体，该实体在特定上下文内可以访问一个被管理对象的实例。
PDU数据所属上下文的唯一名字，这个是可配置的。
APPLICATIONS

SNMPv3 PROCESSING MODULE PARAMETERS



5.5 小结

- SNMP是目前TCP/IP网络中应用最广泛的网络管理协议，使网络管理事实上的标准。它不仅包括网络管理协议本身，而且代表采用SNMP的网络管理框架。
- SNMP经历了从v1到v3的发展历程。

作业

1. 简述SNMP的体系结构。
2. SNMP v1 的报文和SNMP v2的报文由哪些部分组成？
3. SNMP v1 和SNMP v2 分别有哪些操作？
4. 试比较SNMPv2与SNMPv1有什么不同？
5. 简述SNMPv3体系结构的组成，并说明各部分的功能