

Introduction to SNMP



— 陳伯寧 —
交大電信工程學系教授

1



Internet NM RFCs

- **RFC 1052 (1988/4)**
 - ▣ IAB Recommendations for the Development of Internet Network Management Standards
- **RFC 1067 (1988/8), RFC 1098 (1989/4), RFC 1157 (1990/5)**
 - ▣ A simple Network Management Protocol
- **RFC 1065 (1988/8), RFC 1155 (1990/5)**
 - ▣ Structure and Identification of Management Information for TCP/IP-based Internets

2



Internet NM RFCs

- **RFC 1066 (1988/8), RFC 1156 (1990/5)**
 - ▣ Management Information Base for Network Management of TCP/IP-based Internet
- **RFC 1212 (1991/3)**
 - ▣ Concise MIB Definitions
- **RFC 1213 (1991/3)**
 - ▣ Management Information Base for Network Management of TCP/IP-based Internets: MIB-II

3



SNMP Architecture

- **Sub-Contents**
 - ▣ Manager, Agent and Managed Objects

 - ▣ Identification of Object Instances
 - ❖ Object Identifier

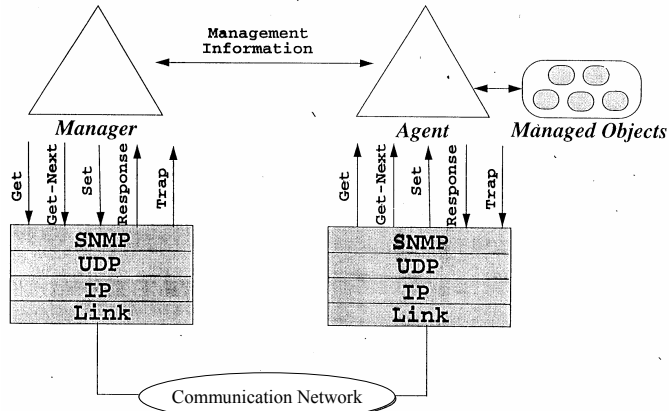
 - ▣ PDUs of SNMP

4



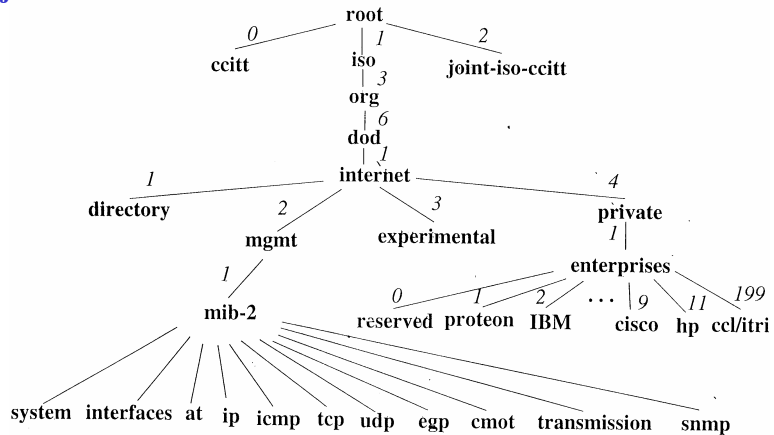
SNMP Architecture

□ Manager, Agent and Managed Objects



SNMP Architecture

□ Object Identifier



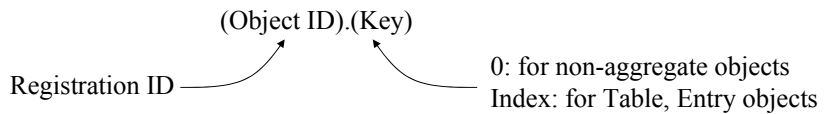


□ Example of Object Identifier.

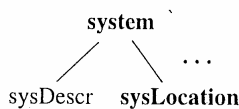


SNMP Architecture

□ Identification of Object Instances



📖 Example 1

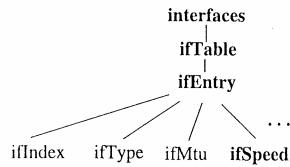


Identifiessystem.sysLocation.0



SNMP Architecture

Example 2



Identifies ...*ifTable.ifEntry.ifSpeed.2*

9



SNMP Architecture

PDU of SNMP

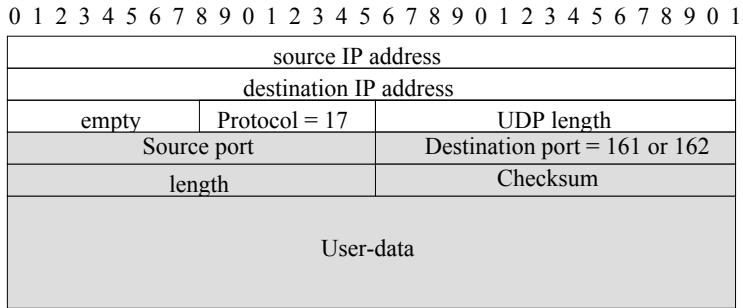
- ☞ Receive messages at UDP port 161
- ☞ Receive traps (notification) at UDP port 162
- ☞ Messages length is no larger than 484 octets
- ☞ PDU types
 - ❖ GetRequest
 - ❖ GetNextRequest
 - ❖ SetRequest
 - ❖ GetResponse
 - ❖ Trap

10



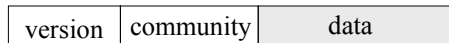
SNMP Architecture

□ User Datagram Protocol(UDP) over Internet Protocol (IP)



SNMP Architecture

☞ General format:



```
Messages ::= SEQUENCE {
    version INTEGER { version-1 (0) },
    community OCTET STRING,
    data ANY
}
```

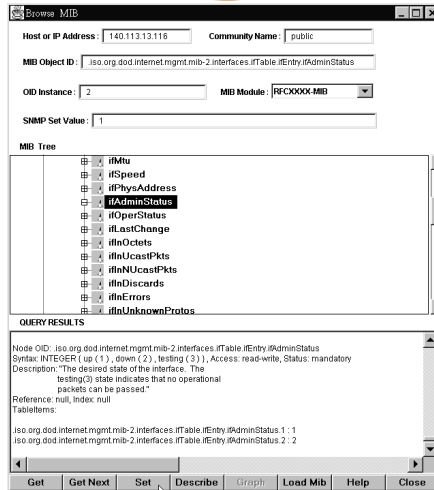
INTEGER = 32 bits using two's-complement arithmetic



SNMP Architecture

Example.

Community name

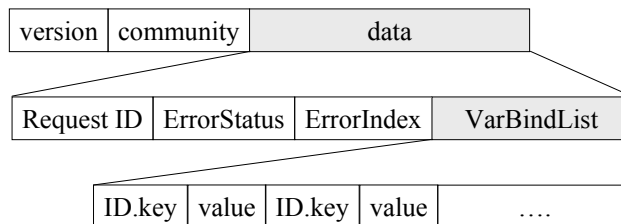


13



SNMP Architecture

For GetRequest, GetNextRequest, SetRequest, and GetResponse



14



SNMP Architecture

- For GetRequest, GetNextRequest, SetRequest, and GetResponse (continue)

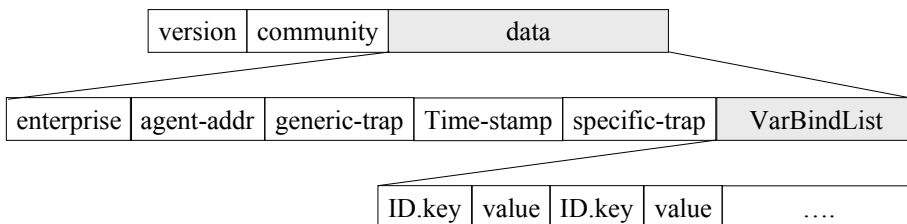
```
RequestID ::= INTEGER
ErrorStatus ::= INTEGER {
    noError(0),
    tooBig(1),
    noSuchName(2),
    badValue(3),
    readOnly(4),
    getErr(5)
}
ErrorIndex ::= INTEGER
```

15



SNMP Architecture

- for Trap :



16



SNMP Architecture

□ for Trap :

```
enterprise      OBJECT IDENTIFIER,
agent-addr      NetworkAddress
generic-trap    INTEGER {
                  coldStart(0),
                  warmStart(1),
                  linkDown(2),
                  linkUp(3),
                  authenticationFailure(4),
                  egpNeighborLoss(5),
                  enterpriseSpecific(6)
                },
specific-trap   INTEGER,
time-stamp      TimeTicks
```

17



MIB Examples

□ IEEE 802.11 Management Information Base

```
IEEE802dot11-MIB DEFINITION ::= BEGIN
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    NOTIFICATION-TYPE, ObjectName, Integer32
FROM SNMPv2-SMI
    DisplayString
FROM SNMPv2-TC
    RowStatus
FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF;
```

18



Examples

□ Importing Macro Definitions

module	macros	Object syntaxes	
SNMPv2-SMI	MODULE-IDENTITY OBJECT-IDENTITY OBJECT-TYPE NOTIFICATION-TYPE	IpAddress Counter32 Gauge32 TimeTicks	Opaque NsnAddress Counter64 UInteger32, etc.
SNMPv2-TC	TEXTUAL-CONVENTION	DisplayString PhysAddress MacAddress TruthValue TestAndIncr	AutonomousType InstancePointer RowStatus TimeStamp TimeInterval DateAndTime, etc
SNMPv2-CONF	OBJECT-GROUP MODULE-COMPLIANCE AGENT-CAPABILITIES		

19



Definition of Object-type Macro

```

OBJECT-TYPE MACRO ::=
BEGIN
  TYPE NOTATION ::=
    "SYNTAX" type(Syntax)
    UnitsPart
    "MAX-ACCESS" Access
    "STATUS" Status
    "DESCRIPTION" Text
    ReferPart
    IndexPart
    DefValPart
  VALUE NOTATION ::=
    value(VALUE ObjectName)
  
```

```

UnitsPart ::= "UNITS" Text | empty
Access ::= "non-accessible"
          | "read-only"
          | "read-write"
          | "read-create"
Status ::= "current" | "deprecated"
          | "obsolete"
ReferPart ::= "REFERENCE" Text
            | empty
IndexPart ::= "INDEX" "{" IndexTypes "}"
            | "AUGMENTS" "{a" Entry "}"
            | empty
IndexTypes ::= IndexType
            | IndexTypes "," IndexType
  
```

20



Definition of Object-type Macro

```
IndexType ::= "IMPLIED" Index | Index
Index ::= value(Indexobject ObjectName)
        -- use the SYNTAX value of the
        -- correspondent OBJECT-TYPE invocation
Entry ::= value(Entryobject ObjectName)
        -- use the INDEX value of the
        -- correspondent OBJECT-TYPE invocation
DefValPart ::= "DEFVAL" "{a" value(Defval Syntax) "}" | empty
Text ::= "" string "" ""
END
```

21



Examples

```
snmpStatePackets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of packets received by the
        SNMPv2 entity from the transport service."
    REFERENCE
        "Derived from RFC1213-MIB.snmpInPkts."
    ::= {snmpStats 1}
```

22



Examples

```
ieee802dot11 MODULE-IDENTITY
LAST-UPDATED "9703050000Z" ←
ORGANIZATION "IEEE 802.11"
CONTACT-INFO "XXXXXXXX"
DESCRIPTION
```

Timestamp defined in the
ASN.1 standard.

```
    "The MIB module for IEEE 802.11 entities.
```

```
    Iso(1).member-body(2).us(840).ieee802dot11(10036)"
```

```
 ::= {1 2 840 10036}
```

```
smt OBJECT IDENTIFIER ::= {ieee802dot11 1}
```

```
mac OBJECT IDENTIFIER ::= {ieee802dot11 2}
```

23



Examples

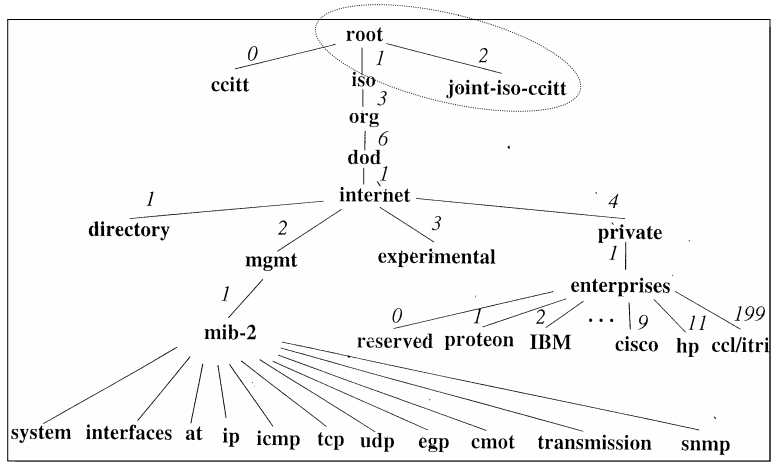
```
res OBJECT IDENTIFIER ::= {ieee802dot11 3}
resAttribute OBJECT IDENTIFIER ::= {res 8}
phy OBJECT IDENTIFIER ::= {ieee802dot11 4}
agStationIDTable OBJECT IDENTIFIER ::= {ieee802dot11 5}
```

```
⋮
⋮
⋮
⋮
```

24



Examples

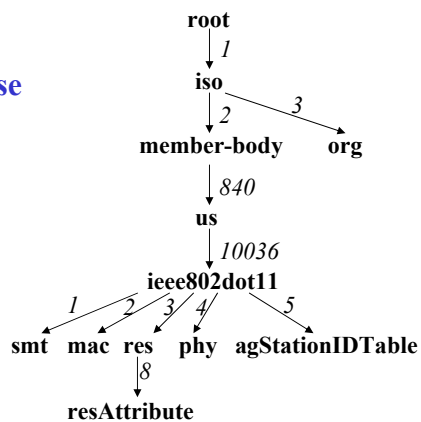


25



Examples

□ IEEE 802.11 Management Information Base



26



Examples

Table

```

agStationConfigGroup OBJECT-TYPE
    SYNTAX SEQUENCE OF TypeagStationConfigGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Station Configuration attributes. In tabular form to
        allow for multiple instances on an agent"
 ::= {smt 1}

```

27



Examples

```

agStationConfigGroupEntry OBJECT-TYPE
    SYNTAX TypeagStationConfigGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the agStationConfigGroup table."
    INDEX {atStationTableIndex} ← Index
 ::= {agStationConfigGroup 1}

```

28



Examples

```
Type agStationConfigGroupEntry ::=
    SEQUENCE {
        aStationTableIndex      Integer32,
        aStationID               MacAddress,
        aMediumOccupancyLimit   Integer32,
        aReceiveDTIMs           INTEGER,
        aCFPollable              INTEGER,
        aAuthenticationType     INTEGER,
        agStationConfigGroupRowStatus RowStatus,
        aCFPPeriod, aCFPMaxDuration,
        RowStatus aAuthenticationResponseTimeOut Integer32
    }
```

29

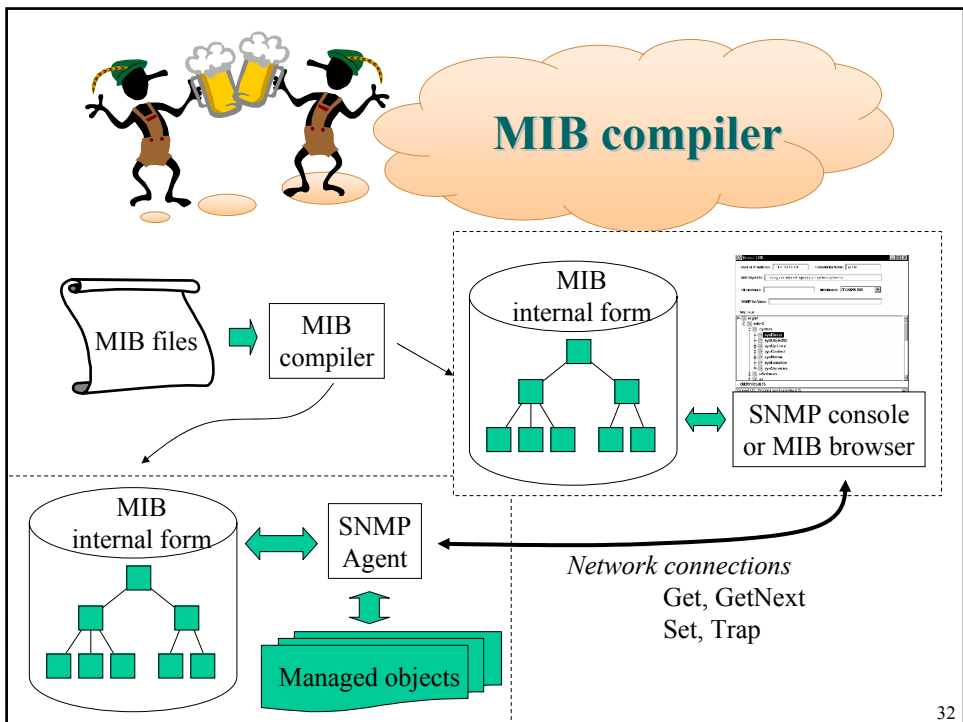
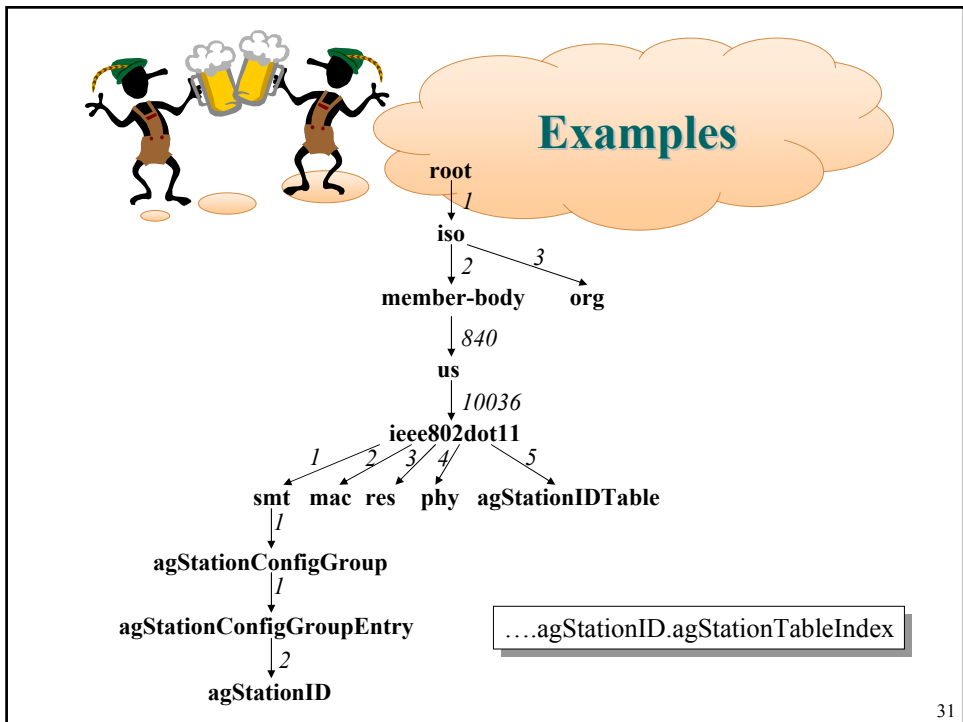


Examples

```

    :
    :
aStationID OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The purpose of aStationID is to allow a manager to rename
        a station for its own purposes. This attribute provides for
        eventuality while keeping the true MAC address
        independent. Its syntax is MAC address and default
        value is the station's assigned unique MAC address."
    ::= { agSttionConfigGroupEntry 2 }
```

30





Limitation of SNMP

□ Limitation of SNMP v1

- ❑ Paradigm not good to the monitoring of large network
- ❑ Inefficient & slow for retrieving large amount data
- ❑ Trap (event) are unacknowledged and unreliable
- ❑ Trivial authentication is not useful for monitoring
- ❑ No manager to manager communication

33



SNMP V2

□ Features of SNMP V2

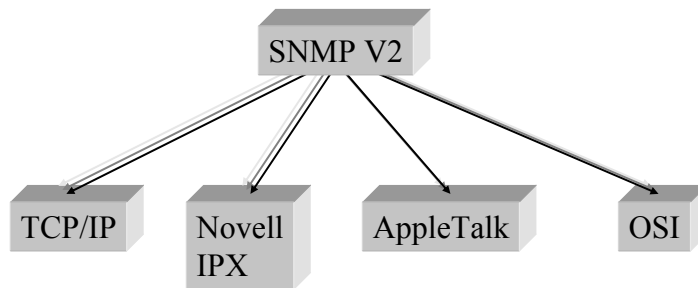
- ❑ Bulk Retrieval Mechanism
- ❑ Manager-to-Manager Communication
- ❑ Multiple Transport Mappings Support
- ❑ Security

34



SNMP V2

- ❑ **Multiple Transport Mappings Support**



35



SNMP V2

- ❑ **SNMP V2 -REF Specifications**
- ❑ **RFC 1441 - Protocol Overview**
- ❑ **RFC 1442 - MIB Structures**
- ❑ **RFC 1443 - Textual Conventions**
- ❑ **RFC 1444 - Conformance Statements**
- ❑ **RFC 1445 - Administrative Model**
- ❑ **RFC 1446 - Security Protocols**
- ❑ **RFC 1447 - Party MIB**
- ❑ **RFC 1448 - Protocol Operation**

36



SNMP V2

- ❑ RFC 1449 - Transport Mappings
- ❑ RFC 1450 - MIB Definitions
- ❑ RFC 1451 - management Station MIB
- ❑ RFC 1452 - Coexistence with SNMP Version 1