Chapter 9
Network Management

A note on the use of these ppt slides:
We’re making these slides freely available to all (faculty, students, readers). They’re in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we’d like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

© All material copyright 1996-2012 J.F Kurose and K.W. Ross, All Rights Reserved
Chapter 9: Network Management

*Chapter goals:*

- introduction to network management
  - motivation
  - major components
- Internet network management framework
  - MIB: management information base
  - SMI: data definition language
  - SNMP: protocol for network management
  - security and administration
- presentation services: ASN.1
Chapter 9 outline

- What is network management?
- Internet-standard management framework
  - Structure of Management Information: SMI
  - Management Information Base: MIB
  - SNMP Protocol Operations and Transport Mappings
  - Security and Administration
- ASN.1
What is network management?

- autonomous systems (aka “network”): 1000s of interacting hardware/software components
- other complex systems requiring monitoring, control:
  - jet airplane
  - nuclear power plant
  - others?

"Network management includes the deployment, integration and coordination of the hardware, software, and human elements to monitor, test, poll, configure, analyze, evaluate, and control the network and element resources to meet the real-time, operational performance, and Quality of Service requirements at a reasonable cost."
Infrastructure for network management

definitions:

managed devices contain managed objects whose data is gathered into a Management Information Base (MIB)

Network Management
Network management standards

OSI CMIP
- Common Management Information Protocol
- designed 1980’s: the unifying net management standard
- too slowly standardized

SNMP: Simple Network Management Protocol
- Internet roots (SGMP)
- started simple
- deployed, adopted rapidly
- growth: size, complexity
- currently: SNMP V3
- de facto network management standard
Chapter 9 outline

- What is network management?
- Internet-standard management framework
  - Structure of Management Information: SMI
  - Management Information Base: MIB
  - SNMP Protocol Operations and Transport Mappings
  - Security and Administration
- ASN.1
SNMP overview: 4 key parts

- **Management information base (MIB):**
  - distributed information store of network management data

- **Structure of Management Information (SMI):**
  - data definition language for MIB objects

- **SNMP protocol**
  - convey manager<->managed object info, commands

- **security, administration capabilities**
  - major addition in SNMPv3
SMI: data definition language

**Purpose:** syntax, semantics of management data well-defined, unambiguous

- base data types:
  - straightforward, boring
- **OBJECT-TYPE**
  - data type, status, semantics of managed object
- **MODULE-IDENTITY**
  - groups related objects into MIB module

**Basic Data Types**

- INTEGER
  - Integer32
- Unsigned32
- OCTET STRING
- OBJECT IDENTIFIED
  - IPAddress
  - Counter32
  - Counter64
  - Guage32
  - Time Ticks
  - Opaque
SNMP MIB

MIB module specified via SMI

MODULE-IDENTITY

(100 standardized MIBs, more vendor-specific)

MODULE

OBJECT TYPE: objects specified via SMI

OBJECT-TYPE construct
OBJECT-TYPE: ipInDelivers

ipInDelivers OBJECT TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
“The total number of input datagrams successfully delivered to IP user-protocols (including ICMP)”
 ::= { ip 9}

MODULE-IDENTITY: ipMIB

ipMIB MODULE-IDENTITY
LAST-UPDATED “941101000Z”
ORGANIZATION “IETF SNPv2 Working Group”
CONTACT-INFO
“Keith McCloghrie
……”
DESCRIPTION
“The MIB module for managing IP and ICMP implementations, but excluding their management of IP routes.”
REVISION “019331000Z”

::= {mib-2 48}
# MIB example: UDP module

<table>
<thead>
<tr>
<th>Object ID</th>
<th>Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.6.1.2.1.7.1</td>
<td>UDPInDatagrams</td>
<td>Counter32</td>
<td>total # datagrams delivered at this node</td>
</tr>
<tr>
<td>1.3.6.1.2.1.7.2</td>
<td>UDPNoPorts</td>
<td>Counter32</td>
<td># underdeliverable datagrams: no application at port</td>
</tr>
<tr>
<td>1.3.6.1.2.1.7.3</td>
<td>UDInErrors</td>
<td>Counter32</td>
<td># undeliverable datagrams: all other reasons</td>
</tr>
<tr>
<td>1.3.6.1.2.1.7.4</td>
<td>UDPOutDatagrams</td>
<td>Counter32</td>
<td># datagrams sent</td>
</tr>
<tr>
<td>1.3.6.1.2.1.7.5</td>
<td>udpTable</td>
<td>SEQUENCE</td>
<td>one entry for each port in use by app, gives port # and IP address</td>
</tr>
</tbody>
</table>
**SNMP naming**

**question:** how to name every possible standard object (protocol, data, more..) in every possible network standard??

**answer:** *ISO Object Identifier tree:*

- hierarchical naming of all objects
- each branchpoint has name, number

![Diagram of ISO Object Identifier tree]

```
1.3.6.1.2.1.7.1
ISO
US DoD
Internet
udpInDatagrams
UDP
MIB2
management
```
OSI Object Identifier Tree
SNMP protocol

Two ways to convey MIB info, commands:

request/response mode

trap mode
<table>
<thead>
<tr>
<th>Message type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRequest</td>
<td>Mgr-to-agent: “get me data”</td>
</tr>
<tr>
<td>GetNextRequest</td>
<td>(instance,next in list, block)</td>
</tr>
<tr>
<td>GetBulkRequest</td>
<td></td>
</tr>
<tr>
<td>InformRequest</td>
<td>Mgr-to-Mgr: here’s MIB value</td>
</tr>
<tr>
<td>SetRequest</td>
<td>Mgr-to-agent: set MIB value</td>
</tr>
<tr>
<td>Response</td>
<td>Agent-to-mgr: value, response to Request</td>
</tr>
<tr>
<td>Trap</td>
<td>Agent-to-mgr: inform manager of exceptional event</td>
</tr>
</tbody>
</table>
# SNMP protocol: message formats

<table>
<thead>
<tr>
<th>PDU type (0-3)</th>
<th>Request ID</th>
<th>Error Status (0-5)</th>
<th>Error Index</th>
<th>Name</th>
<th>Value</th>
<th>Name</th>
<th>Value</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDU type 4</td>
<td>Enterprise</td>
<td>Agent Addr</td>
<td>Trap Type (0-7)</td>
<td>Specific code</td>
<td>Time stamp</td>
<td>Name</td>
<td>Value</td>
<td>…</td>
</tr>
</tbody>
</table>

**SNMP PDU**

**Get/set header**

**Variables to get/set**

**Trap header**

**Trap info**
SNMP security and administration

- **encryption**: DES-encrypt SNMP message
- **authentication**: compute, send $\text{MIC}(m, k)$: compute hash (MIC) over message $(m)$, secret shared key $(k)$
- **protection against playback**: use nonce
- **view-based access control**:
  - SNMP entity maintains database of access rights, policies for various users
  - database itself accessible as managed object!
Chapter 9 outline

- What is network management?
- Internet-standard management framework
  - Structure of Management Information: SMI
  - Management Information Base: MIB
  - SNMP Protocol Operations and Transport Mappings
  - Security and Administration
- The presentation problem: ASN.1
The presentation problem

Q: does perfect memory-to-memory copy solve “the communication problem”?

A: not always!

```
struct {
    char code;
    int x;
} test;

test.x = 256;
test.code = ‘a’
```

```
<table>
<thead>
<tr>
<th>test.code</th>
<th>test.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>00000001</td>
</tr>
<tr>
<td></td>
<td>00000011</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>test.code</th>
<th>test.x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000001</td>
</tr>
<tr>
<td></td>
<td>00000001</td>
</tr>
</tbody>
</table>
```

**Problem:** different data format, storage conventions
A real-life presentation problem: 

- Grandma
- Aging 60’s hippie
- 2012 teenager

Groovy!
Presentation problem: potential solutions

1. Sender learns receiver’s format. Sender translates into receiver’s format. Sender sends.
   - real-world analogy?
   - pros and cons?

2. Sender sends. Receiver learns sender’s format. Receiver translates into receiver-local format
   - real-world analogy
   - pros and cons?

   - real-world analogy?
   - pros and cons?
Solving the presentation problem

1. Translate local-host format to host-independent format
2. Transmit data in host-independent format
3. Translate host-independent format to remote-host format
ASN.1: Abstract Syntax Notation 1

- **ISO standard X.680**
  - used extensively in Internet
  - like eating vegetables, knowing this “good for you”!
- **defined data types, object constructors**
  - like SMI
- **BER: Basic Encoding Rules**
  - specify how ASN.1-defined data objects to be transmitted
  - each transmitted object has Type, Length, Value (TLV) encoding
TLV Encoding

Idea: transmitted data is self-identifying

- **T**: data type, one of ASN.1-defined types
- **L**: length of data in bytes
- **V**: value of data, encoded according to ASN.1 standard

<table>
<thead>
<tr>
<th>Tag Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Boolean</td>
</tr>
<tr>
<td>2</td>
<td>Integer</td>
</tr>
<tr>
<td>3</td>
<td>Bitstring</td>
</tr>
<tr>
<td>4</td>
<td>Octet string</td>
</tr>
<tr>
<td>5</td>
<td>Null</td>
</tr>
<tr>
<td>6</td>
<td>Object Identifier</td>
</tr>
<tr>
<td>9</td>
<td>Real</td>
</tr>
</tbody>
</table>
**TLV encoding: example**

module of data type declarations written in ASN.1

instances of data type specified in module

Basic Encoding Rules (BER)

Value, 259

Length, 2 bytes

Type=2, integer

Value, 5 octets (chars)

transmitted byte stream

Length, 5 bytes

Type=4, octet string

lastname ::= OCTET STRING
weight ::= INTEGER

{weight, 259}
{lastname, “smith”}
Network management: summary

- Network management
  - Extremely important: 80% of network “cost”
  - ASN.1 for data description
  - SNMP protocol as a tool for conveying information
- Network management: more art than science
  - What to measure/monitor
  - How to respond to failures?
  - Alarm correlation/filtering?