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"): 100s or 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:

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)

(objects specified via SMI **OBJECT-TYPE** construct)
SMI: Object, module examples

**OBJECT-TYPE: ipInDelivers**

Object ID: ipInDelivers

**MODULE-IDENTITY: ipMIB**

```
OBJECT-TYPE: ipInDelivers
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}
```

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># undeliverable datagrams no app at port1</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>UDOuDatagrams</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

```
ISO
US DoD
Internet
```

```
1.3.6.1.2.1.7.1
iso
udpInDatagrams
UDP
MIB2
management
```

**OSI Object Identifier Tree**

```
ITUT (3) ISO (1) Joint ISO/ITUT (2)
Standard (0) ISO member body (2) ISO identified organisation (3)
US DoD (6) ... Open Software Foundation (22) ... NATO identified (157)
Internet (1)
```

```
directory (1) experimental (3) security (5) mail (7)
management (2) private (6) snmp2 (6)
MIB-2 (11)
```

```
system (1) address translation (3) icmp (5) udp (7) icmp (7)
        (3) tran-        (7)              (16) (7)        (16)
        nm (2)         meas (3)        (4) (4)        (10)
```

Check out [www.alvestrand.no/harald/objectid/top.html](http://www.alvestrand.no/harald/objectid/top.html)
**SNMP protocol**

Two ways to convey MIB info, commands:

- **request/response mode**
  - Managing entity requests data from an agent.
  - The agent responds with the requested data.

- **trap mode**
  - The managing entity sends a trap message to inform the agent of an exceptional event.

---

**SNMP protocol: message types**

<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></td>
<td>(instance, next in list, block)</td>
</tr>
<tr>
<td>GetNextRequest</td>
<td></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>
</table>

| PDU Type (4) | Enterprise | Agent Addr | Trap Type (0-7) | Specific Code | Time Stamp | Name | Value | ... |

SNMP PDU

SNMP security and administration

- **encryption**: DES-encrypt SNMP message
- **authentication**: compute, send 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!

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

test.x = 256;
test.code = 'a'
```

```plaintext
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
</tr>
<tr>
<td>00000001</td>
<td></td>
</tr>
<tr>
<td>test.x</td>
<td></td>
</tr>
</tbody>
</table>

```

*host 1 format*

```plaintext
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
</tr>
<tr>
<td>00000011</td>
<td></td>
</tr>
</tbody>
</table>
```

*host 2 format*

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

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

Transmitted byte stream

- Value, 5 octets (chars)
- Length, 5 bytes
- Type=4, octet string

Network Management 9-26
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?