NMRG Report

Improving/Enhancing SNMP and SMI

Jürgen Schönwälder
IRTF Network Management Research Group

http://www.ibr.cs.tu-bs.de/projects/nmrg/

Outline

- o Introduce the Network Management Research Group
- o Report on bulk data transfer improvements for SNMP
- o Report on the enhanced SMI proposal called SMIng

NMRG Background

- NMRG is a forum where people can discuss and develop new technologies for improving the management of the Internet
- Approved by the IAB in March 1999
- NMRG membership requires commitment to active participation
- Mailing list archive, minutes and documents are publically available
- Trying to get people with different technological and organizational backgrounds working together

NMRG Business Model

- Focus on solutions for real-world technical problems
- Prove the feasibility of ideas and concepts through implementation
- Make results openly available to the whole community
- Keep close contacts with IETF working groups in order to coordinate work
- Bring solutions back into the IETF as potential future standardization efforts

IETF Network Management Problems

- The requirements for Internet management technologies have changed during the last 10 years
- Fundamental assumptions must be revisited and potentially revised to better reflect today's realities
- WG members mostly coming from network device vendors
- Solutions sometimes tend to be too device specific or way too detailed for running real-world networks
- Many competing technologies inside and outside of the IETF have led to "network management technology fragmentation"

SNMP Specific Problems

- Work on SNMP security took many many years to finally result in a stable and accepted SNMPv3 specification
- Other urgendly needed improvements were kept on hold during this time
- One such problem is the inefficiency of SNMP for retrieving bulk MIB data
- Extensions to the SMI data definition language needed to simplify, generalize and harmonize data definitions
- NMRG decided to look at these two important issues

Outline

✓ Introduce the Network Management Research Group

o Report on bulk data transfer improvements for SNMP

o Report on the enhanced SMI proposal called SMIng

Review of SNMP Basics

- SNMP protocol is used to access and manipulate variables organized in conceptual tables and groups of scalars
- Each scalar and each cell in a conceptual table is uniquely identified by an OID value (in a given context)
- SNMP operates on an ordered lists of scalars and table cells (varbind list)
- Each element of such a list contains an OID value identifying a scalar or a table cell and its value
- SNMP runs over UDP (stateless, retransmission control)

SNMP Bulk Retrieval Problem

- Problem:
  - SNMP shows poor performance when retrieving several thousands of MIB variables in a single logical transaction

- Reasons:
  - Lack of flow control
  - Bandwidth inefficiency due to OID naming overhead
  - High latency caused by a large number of request/response interactions

Proposal #1: SNMP over TCP

- SNMP over TCP gives flow and congestion control for virtually no costs
- Originator of a request/response transaction chooses the transport for one or more complete transactions
- SNMP engines can close TCP connections at any point in time
- SNMP engines may revert to SNMP over UDP when needed
- SNMP engines must perform packetizing and connection management

Proposal #2: Lossless Payload Compression

- Compress and wrap SNMP PDUs in a CompressedPDU
- Each SNMP message is compressed and decompressed by itself without any relation to other SNMP messages (stateless compression)
- The size of a compressed SNMP message must never exceed the size of the uncompressed message (non-expansion policy)
- Support for multiple compression algorithms
- Negotiation of compression algorithm via MIB objects

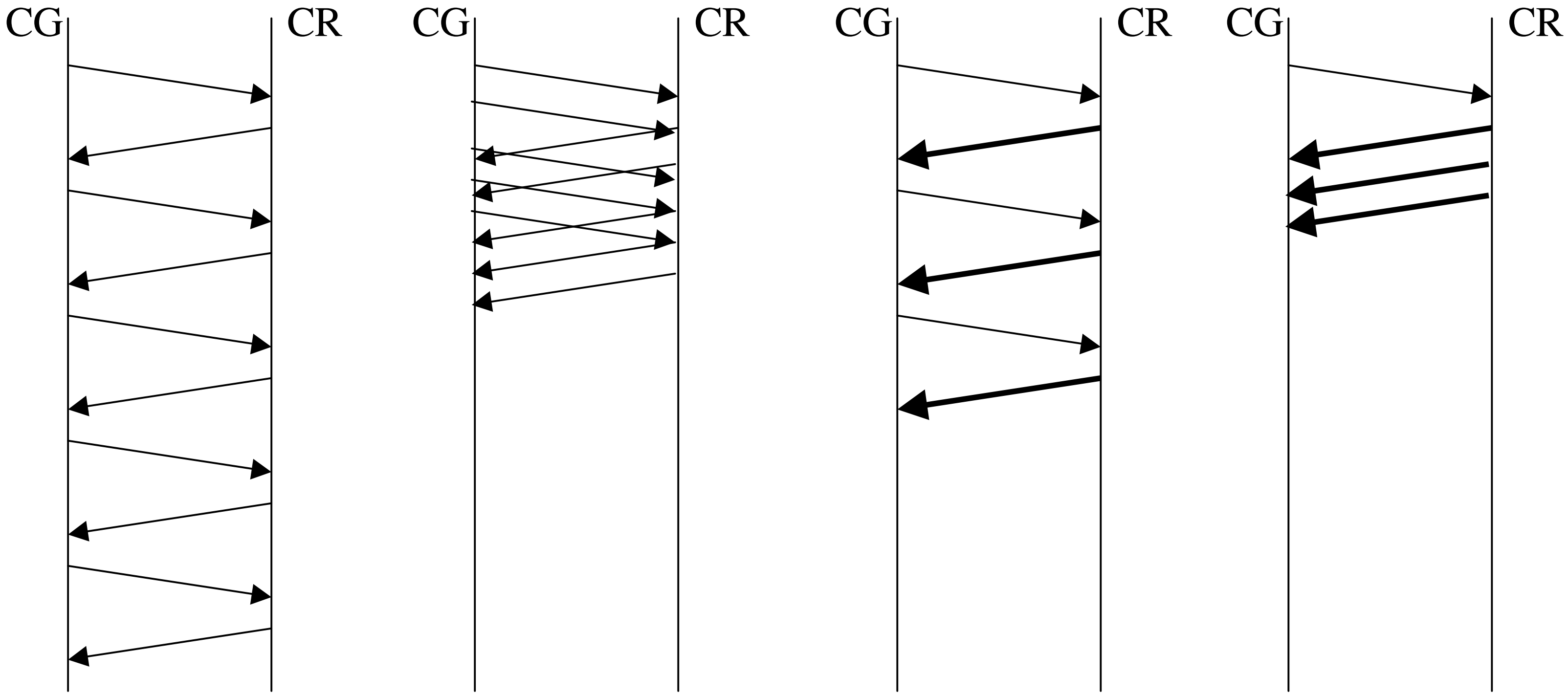
Proposal #2: Lossless Payload Compression

- Deflate:
  - Deflate achieves high compression ratios of 80 % on typical MIB-II data
  - Requires noticeable CPU resources on the sending SNMP engine
  - Interactions with message size constraints make it difficult to build response messages that send as much data as possible

Proposal #2: Lossless Payload Compression

- ODC:
  - OID Delta Compression (ODC) has been invented to reduce only the OID overhead in SNMP PDUs
  - The general idea is to encode an OID as a delta to the previous OID
  - Algorithm can be made to run very fast by integrating it into the BER encoder, even though compression is conceptually a transformation on the encoded PDU
  - Achieves compression ratios of 40 % on typical MIB-II data

Proposal #3: Get-Subtree / Linked Responses



GetNext
(row by row)

GetNext
(rows in parallel)

GetBulk

GetSubTree

Proposal #3: Get-Subtree / Linked Responses

- The format of get-subtree PDUs is similar to the format of the get-bulk PDU, except that there is no max-repetitions parameter
- OIDs in the variable-bindings list identify the roots of the subtrees to be retrieved
- Linked response PDUs are ordinary response PDUs where the error-index contains a sequence number if the error-status is noError
- Measurements over the loopback interface on a Linux box show that get-subtree is 4.5 times faster compared to get-next walks and 2 times faster than get-bulk walks

Outline

✓ Introduce the Network Management Research Group

✓ Report on bulk data transfer improvements for SNMP

o Report on the enhanced SMI proposal called SMIng

SMIng Project Goals

- Design a next generation SMI to enhance the SMIv2
- Work upwards towards a richer data definition language
- SMIng must be compatible with SMIv2 so that automatic translations are feasible
- SMIng language definition should not rely on "external" standards (and thus not use ASN.1)
- SMIng must be extensible so that enhancements can be introduced gradually without destabilizing the whole language
- Simplify the SMI language wherever possible

SMIng Syntax

- SMIng introduces a new syntax (fully defined in ABNF)
- Syntax rules designed to improve readability and to simplify parser implementations
- SMIng files contain sequences of statements
- Statements can have statement blocks as argument (nesting)
- Each statement is terminated by a semicolon
- Case sensitivity of identifiers inherited from SMIv2

Structure of an SMIng Module

```
module IF-MIB {

  import IRTF-NMRG-SMING (mib-2);

  // …

  oid   mib-2.31;
  organization          "IETF Interfaces MIB WG";
  contact               "…";
  revision {
    date                "1996-02-28 21:55";
    description         "Revisions made by the IFMIB WG";
  };
  revision {
     date               "1993-11-08 21:55";
     description        "Initial revision, RFC 1573";
  };
```

Structure of an SMIng Module

```
    // extension statements

    // typedef statements

    // node/scalar/table/ statements

    // notification statements

    // group statements

    // compliance statements
};
```

SMIng Table Definition

```
table ifTable {
  oid          interfaces.2;
  description  "A list of interface entries";

  row ifEntry {
    oid          ifTable.1;
    index        (ifIndex);
    description  "…";

    column ifIndex {
      oid          ifEntry.1;
      type         InterfaceIndex;
      access       readonly;
      description  "…";
    };
    // …
```

SMIng Base and Core Derived Types

- Integer32
- Unsigned32
  - Counter32
  - TimeTicks
  - Gauge32
- Enumeration
- Unsigned32
- Integer64
- Unsigned64
  - Counter64
  - Gauge64
- Float32
- Float64
- Float128
- OctetString
  - IpAddress
  - Opaque
- Bits
- ObjectIdentifier

- SMIng allows to derive types from other derived types
- Some SMIv2 base types are now defined as derived types
- Type definitions can have “units” statements attached to it

Extensions and Annotations

- SMIng makes it possible to define language extensions (new statements) over time
- Parser implementations are required to ignore unknown statements
- An annotation mechanism can be used to invoke statements in the scope associated with a particular definition in a separate module
- These two mechanisms can be used to separate protocol specific definitions from the core data structures and to supply additional semantics in a machine readable form over time

Annotation & Extension Example

```
module IF-MIB {

  notification linkDown {
    // …
  };

};
```

```
module SEVERITY-EXT {

  extension severity {
    // …
  };

};
```

```
module IF-MIB-ANNOTATIONS {

  import IF-MIB (linkDown);
  import SEVERITY-EXT (severity);

  annotation linkDownSeverity {
    target (linkDown);
    severity “critical”;
  };

};
```

SMIng Status

- Proof of concept implementation available in the libsmi package
- The libsmi API has been designed on the basis of SMIng and hides 99 % of the SMI version differences
- Some translations implemented so far:
  - SMIng -> SMIv2 (SMIv2 subset)
  - SMIv2 -> SMIng
  - SMIng -> CORBA IDL (based on JIDM rules)
  - SMIng -> XML
- The XML DTD has been defined as a common exchange format for applications that need to access SMI definitions

Summary

- NMRG is working very well and has produced significant results for improving/enhancing SNMP and the SMI
- Discussion underway about moving IRTF results back into the IETF
- There is much more work waiting to be addressed (e.g. adding COPS-PR support to the SMIng)
- Volunteers who are interested to join the NMRG to work on these or new items are encouraged to contact us

More Information

- Web pages:
  - NMRG http://www.ibr.cs.tu-bs.de/projects/nmrg/
  - SMIng http://www.ibr.cs.tu-bs.de/projects/sming/
  - libsmi http://www.ibr.cs.tu-bs.de/projects/libsmi/

- Documents:
  - SNMP over TCP draft-irtf-nmrg-snmp-tcp-04.txt
  - SMIng draft-irtf-nmrg-sming-01.txt
  - SMI XML DTD draft-ietf-nmrg-smi-xml-00.txt