DTN Research Group
Internet-Draft
Expires: August 13, 2010

G. Clark
G. Campbell
H. Kruse
S. Ostermann
Ohio University IRG

February 9, 2010

DING Protocol -- A Protocol For Network Management
draft-irtf-dtnrg-ding-network-management-02

Abstract

This memo describes the Diagnostic Interplanetary Network Gateway protocol (DING). DING is a subscription-based network management protocol designed specifically for use in situations where SNMP's request / response model does not perform well, e.g. in heavily delayed and / or disrupted networks.

Status of this Memo

This Internet-Draft will expire on August 13, 2010.

Copyright Notice

Table of Contents

1. Introduction

Over the past 20 years, SNMP has firmly established itself as the de facto standard protocol for network management. There are SNMP agents available for everything from network switches to supercomputers. Further, the protocol works extremely well on most (if not all) terrestrial networks.

In delay tolerant networks (DTNs), however, SNMP tends to perform poorly. The request / response mechanism SNMP uses does not seem practical in environments where transmission delays can range from minutes to days; by the time a query has made it to a remote host and back again, the information contained in the response is often too old to be relevant to the device's status at the current point in time.

Furthermore, in many delay-tolerant networks, there are often severe restrictions in place on the amount of bandwidth available to a given device. While SNMP is not an inefficient protocol, there are areas where existing knowledge about a device could be utilized to drastically reduce the size of SNMP requests and replies.

The Diagnostic Interplanetary Network Gateway (DING) protocol described in this document has been designed to address these issues. Rather than use the request / reply paradigm employed by SNMP, DING relies on a subscription model.

When a host wishes to receive information about a remote node, the host (called a "subscriber") sends that node (called a "provider") a "subscription request". This request consists of two parts: a "schema" and a "schedule".

The "schema" is a static data model definition that defines exactly which data the subscriber wishes to receive from the provider. This schema consists solely of a list of object identifiers (OIDs). The provider, upon examining the schema, parses this list of OIDs and compares each OID to a list of known OIDs it has. If the OID matches a piece of information it knows how to provide, the provider will transmit that information as instructed by the schedule included with the subscription request. If the provider does not know how to translate a given OID into data, the provider will assign 0 to any field where that OID is requested.

The "schedule" contains two things: a time interval and a set of conditions. The conditions determine when a provider sends a scheduled frame: the provider will only send a scheduled frame if all conditions are met (the expression provided in the schedule frame evaluates to true). If no conditions are specified, then the

provider assumes that there are no conditions to meet and will transmit scheduled frames as determined by the time interval.

The time interval itself (specified in seconds) determines how often to transmit "scheduled frames" (sets of data encoded using a given schema). As soon as the conditions in a schedule are met, a single scheduled frame is transmitted, and then an additional scheduled frame is transmitted every time_interval seconds. If the time interval is set to be zero, then the scheduled frame is sent only when the specified conditions are met and is never repeated.

When a provider sends a scheduled frame to a subscriber as described by a schema, the scheduled frame includes a "Schema ID" which can be used by the subscriber to uniquely identify the schema associated with that scheduled frame. It also includes a "Schedule ID" which can be used to identify the schedule that triggered the frame. Assuming Schema IDs are used and agreed upon by both the provider and the subscriber, data can be encoded far more efficiently than it would be if encoded in the standard BER and DER formats used by SNMP. If further compression is desired, DING supports specifying that data has been compressed; see sections 4.3 and 4.4 of this document for details.

DING is designed to operate in conjunction with a gateway. This gateway is designed to act as a kind of SNMP proxy for DTN nodes. To do this, the gateway subscribes to a number of DING providers running on a DTN and caches scheduled frames received from each provider. The gateway also runs an SNMP agent which uses this cached information to respond to SNMP GET messages and generate SNMP TRAP messages. The motivation for and operation of this gateway will be explored further in a separate draft (currently a work in progress).

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Service Definitions

Frame - A collection of consecutive bytes organized into a single logical unit.

Provider - A machine that provides a service which processes subscription requests from and sends scheduled frames to subscribers.

Registration - The act of sending a subscription request to a provider.

Removal Request - Request to remove a certain schema from a provider (so that the provider will quit using that schema along with any associated schedules to send scheduled frames to a subscriber) and / or remove a certain schedule from a provider.

Schedule - Links a schema with a time interval and a set of conditions. Assuming that all conditions described in a schedule are met, the provider is expected to send a scheduled frame every [time interval] seconds.

Scheduled Frame - A frame sent by a provider to a subscriber as directed by the "schedule" portion of a subscription request. The content of this data is described by the "schema" portion of a subscription request. A scheduled frame may be associated with a corresponding schema and / or subscription request through the use of a schema ID.

Scheduled Frame Container - A container meant to transport one or more scheduled frames from a provider to a subscriber.

Scheduled Frame Entry - A single scheduled frame contained within a "Scheduled Frame Container".

Schema - A static data model which is included in a subscription request and defines exactly which pieces of data the subscriber is requesting from the provider.

Schema ID - A sequence of bytes which can be used to uniquely identify a given schema. This documents suggests computing an SHA-1 hash of the schema to compute this value, although the particular method is not as important as is A) ensuring that both subscriber and provider agree on an ID and B) ensuring ID is as unique as possible.

Self-Delimiting Numeric Value (SDNV) - Encoding for a positive integer. Supports numbers of any size through the use of a scheme similar to that used within BER to encode OIDs into numeric values. See RFC 5050 for a complete explanation of SDNVs and their usage.

Subscriber - A machine which has sent a subscription request to a provider.

Unsubscribe - Sending a "removal request" to a provider so that provider will quit sending certain scheduled frames to a given subscriber.

4. Subscription Requests

4.1. Subscription Request Overview

A subscription request is composed of the following fields:

- Type -- Four bit field consisting of the value '0110'.

- Options -- Set of options applying to every field contained within this subscription request. 4-bit field. The following options are supported:

```
 0 1 2 3
+-+-+-+-+
|C|H|S|E|
|M|S|C|X|
|P|H|A|T|
+-+-+-+-+
```

Figure 1: Options

  - CMP -- Compression bit. If set, the schema is compressed. The format is encoded as a one-byte number in the options data. The following options must be supported:

    - 0 -- No compression.

    - 1 -- GZIP format.

    Additional options may be supported assuming both subscriber and provider agree on a common association between number and compression scheme.

  - HSH -- Hash bit. If set, the provider should run a specified hash algorithm on each included schema and verify that the resulting hash matches the specified schema's ID. This is designed to be used as a simple data integrity check, and could perhaps be used as a part of a simple authentication scheme as well. The algorithm to use is specified by the first byte of the options data section (HSHDATA). The following options must be supported:

    - 0 -- No hash; IDs of all included schemas are assumed to be 0. Using this option is not recommended, but may be necessary for some devices depending on hardware specifications and / or bandwidth requirements.

      + 1 -- CRC; 16 bit. Using this option is not recommended, but may be necessary for some devices depending on hardware specifications and / or bandwidth requirements.

      + 2 -- MD5; 128 bit digest.

      + 3 -- SHA-1; 160 bit digest.

      When the hash option is specified, the size of the schema ID field must be the same size as the size of the digest produced. When the hash bit is not specified, the schema ID field must have a size of 20 bytes.

   * SCA -- Scaling bit. If this bit is set, NSC and NSM will both be multiplied by an SDNV provided in the option data.

   * EXT -- Extension bit. If this bit is set, then the option data contains an SDNV integer of value X (after the hash byte and compression byte, if either and / or both are present), followed by X bytes of data. This is to allow the inclusion of additional options, should the need arise.

o Option Data -- Data necessary to support any option bits set in the Options field. Up to three bytes in length unless EXT bit is set, at which point the size of this field is unknown.

o Number of Schemas (NSM) -- 4 bits encoding number of schemas included with this request. Up to 16 * SCA.

o Number of Schedules (NSC) -- 4 bits encoding number of schedules included with this request. Up to 16 * SCA.

o Schema X (SMX) -- schema number X in a sequence of NSM consecutive schema fields. Unknown total length, but there can be no more than NSchma schemas included with a single subscription request.

o Schedule X (SCX) -- schedule number X in a sequence of NSC consecutive schedule fields. Unknown total length, but there can be no more than NSched schedules included with a single subscription request.

Example schedule request header format (assuming CMP = 1, HSH = 1, SCA = 0, EXT = 0):

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  TYPE | OPTN  |    HSHDATA    |    CMPDATA    |  NSM  |  NSC  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               SMX [NSM variable-length schemas]               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                  ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              SCX [NSC variable-length schedules               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Subscription Request Overview

4.2. Schema Section

4.2.1. Schema Format

A schema is composed of an X-bit ID unique ID, followed by a list of SDNV / OID pairs. Each SDNV specifies the depth of a particular OID from the root of the OID tree (.1 = 1, .3.2.4 = 3, .6.1.4.7.24 = 5, etc), and each OID is encoded in standard DER format. The last entry in a schema must consist of a single "OID_DEPTH" field with a value of 0.

```
+-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+ ~ +-+-+-+-+-+-+-+-+ ~ +-+-+-+ ~ +-
+-+
|  SCHEMA_ID  |   |  OID_DEPTH  |   |  OID  |   |  OID_DEPTH  |   | OID |   | 0
|
+-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+ ~ +-+-+-+-+-+-+-+-+ ~ +-+-+-+ ~ +-
+-+
```

Figure 3: Schema Format

Given the nature of DTN, it is a distinct possibility that a request to remove a schema from a system may arrive before the schema itself. Thus, requests for the removal of a schema that does not yet exist should be held for some time, where the exact definition of "some time" is left up to the system implementing this protocol.

If a schema is received with a Schema ID that matches the Schema ID of a schema already stored, the newest schema may be either stored (while waiting for a removal request) or silently discarded. Note that if a schema arrives with a Schema ID and length that matches any removal request currently queued by the system, that schema should be immediately discarded.

4.2.2. Schema / SNMP Compatibility Note

In an effort to provide some compatibility between DING and SNMP, a schema may be stored locally on a host as a MIB encoded in standard ASN.1/SMIng format. To convert a MIB into a schema, a provider or subscriber should extract each leaf OID defined within the MIB. Once the provider or subscriber has done this, they may simply use this list of OIDs as a schema. Note that the order these OIDs appear in the schema must match the order in which they were defined in the corresponding MIB.

4.2.3. OID Compression

[Need to do literature review here; found draft on SNMP Payload Compression by Schoenwaelder et. al.]

[How do OID compression algorithms compare to "standard" compression algorithms e.g. GZIP / BZIP2?]

4.3. Schedule Section

4.3.1. Schedule Overview

A schedule is composed of an X-bit schema ID, an SDNV-encoded schedule ID, an SDNV-encoded interval value, and a set of conditions (an expression).

```
      +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+
      | SCHEMA_ID |   |  SCHED_ID |    | INTERVAL  |    | EXPR  |
      +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+
```

Figure 4: Schedule Format

The SCHEMA_ID must refer to an existing schema (which may be defined by the current subscription request or exist previously); if this field refers to an unknown schema, the provider may choose to A) ignore the schedule request or B) hold the schedule request for an unknown time in the hopes of receiving a relevant schema.

The SCHED_ID is an SDNV which can be used, in conjunction with the schema ID, to uniquely reference this schedule.

The INTERVAL is a time interval, specified in seconds and encoded as an SDNV. Assuming all conditions are met, this value specifies the amount of time the provider should wait between each scheduled frame it transmits.

EXPR is an expression (a set of conditions) that determine whether or not a scheduled frame should be transmitted. Each condition is composed of a single operator (<, >, <=, >=, ==, !=, &&, ||) and two operands, where an operand can be an OID, a constant, or a condition.

- If an operand is an OID, the value of any sensors associated with that OID should be inserted into the expression when the expression is evaluated. If a provider does not have a sensor associated with a given OID, the OID should be treated as the integer '0'.

- An expression must evaluate to be either TRUE or FALSE.

- If the value of an expression changes from FALSE to TRUE, a scheduled frame should be immediately sent unless a scheduled frame has already been sent within the last INTERVAL seconds.

- Example: .3.2.7.4 <= 4 && .3.2.7.5 > .3.2.7.4

- [*** TODO *** : Specify how EXPR is encoded in schedule data (e.g. NOT plaintext). Specify BNF. Perhaps move conditions to its own section?]

4.4. Example Subscription Request

NOTE: This is a contrived example; the OIDs used here are not valid, and are used for the sole purpose of saving space. Also, the schema ID here is not necessarily valid.

Assume a satellite (the provider) knows that OID .92.2.3.1 corresponds to a thermal sensor and that OID .92.2.3.2 corresponds to a battery charge sensor. A subscriber sends a subscription request to this satellite, with the following OIDs listed in the schema portion of the subscription request:

```
        +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+ ~ +-
+-+
        | 0xf73492aaecb |   |  4  |  .92.2.3.1  |   |  4  |  .92.2.3.2  |   | 0
|
        +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+ ~ +-
+-+
```

Figure 5: Example Schema Section

This schema says every scheduled frame that the satellite sends with a Schema ID corresponding to that of this schema (0xf73492aaecb) will contain two data values: the data from the thermal sensor and the data from the battery charge sensor.

The schedule part of the subscription request looks like:

```
           +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           | 0xf73492aaecb |   |   108000    |  .92.2.3.1 > 120  |
           +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Example Schedule Section

The schedule says that scheduled frames with a format corresponding to this schema with ID 0xf73492aaecb will be sent every 30 minutes (108000 seconds) ONLY IF the value corresponding to .92.2.3.1 (in this case, the reading from the thermal sensor) is greater than 120.

5. Scheduled Frames

5.1. Scheduled Frame Container Overview

A scheduled frame container is composed of the following fields:

- Type -- Four bit field consisting of the value '0111'.

- Options -- Set of options applying to every field contained within this scheduled frame. 4 bit field. The following options are supported:

```
 0 1 2 3
+-+-+-+-+
|C|U|S|E|
|M|N|C|X|
|P|U|A|T|
+-+-+-+-+
```

Figure 7: Options

  - CMP -- Compression bit. If set, all data within the scheduled frame is compressed. The format is encoded as a one-byte number in the options data. The following options must be supported:

    - 0 -- No compression.

    - 1 -- GZIP format.

    Additional options may be supported assuming both subscriber and provider agree on a common association between number and compression scheme.

  - UNU -- Unused bit. The value of this bit must be ignored.

  - SCA -- Scaling bit. If this bit is set, NSF will be multiplied by an SDNV provided in the option data.

  - EXT -- Extension bit. If this bit is set, then the option data contains an SDNV integer of value X (after the hash byte and compression bytes, if either and / or both are present), followed by X bytes of data. This is to allow the inclusion of additional options, should the need arise.

- o Option Data -- Data necessary to support any option bits set in the Options field. Up to two bytes in length unless EXT bit is set, at which point the size of this field is unknown.

- o Number of Scheduled Frames (NSF) -- 8 bits encoding number of entries included within this scheduled frame. Up to 256.

- o Scheduled Frame X (SFX) -- scheduled frame number X in a sequence of NSF consecutive scheduled frame fields.

Example scheduled frame container header (assuming CMP = 0, SCA = 0, EXT = 0):

```
                    0                   1
                    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |  TYPE | OPTN  |      NSF      |
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |       SFX [NSF entries]       |
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Scheduled Frame Container Overview

5.2. Scheduled Frame Entries

Each individual scheduled frame entry within a scheduled frame container consists of the following fields:

Schema ID -- ID of the schema that should be used to parse this data. Variable length field.

Sched ID -- ID of the schedule that generated this scheduled frame.

Data -- Data which corresponds to a schema. Each 'Data' field in the schema corresponds to the value mapped from a single OID.

```
        +-+-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-
+-+-+
        |   SCHEMA_ID   |   | SCHED_ID  |   |    DATA   |   |    DATA   |
| ... |
        +-+-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+ ~ +-
+-+-+
```

Figure 9: Scheduled Frame Entry Format

5.3. Scheduled Frame Entry Example

Assuming a subscriber has registered with the provider described in the "Subscription Request Example" section, then an entry in the scheduled frame container originating from the provider (the satellite, in this case) might look like:

```
                +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                | 0xf73492aaecb |   |      170      |       252       |
                +-+-+-+-+-+-+-+-+ ~ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Satellite's Scheduled Frame Entry

assuming that the thermal and charge sensors reported values of 170 and 252, respectively.

6. Removal Requests

Sent by a subscriber to unsubscribe (quit receiving scheduled frames from a provider) or to remove a schedule associated with a given schema (referenced by schema ID and schedule ID). *** TODO ***

7. Status Requests

Authors' Addresses

Gilbert Clark
Ohio University Internetworking Research Group
301 Stocker Center
Ohio University
Athens, OH 45701
US

Email: gc355804@ohio.edu
URI: http://irg.cs.ohiou.edu/

Greg Campbell
Ohio University Internetworking Research Group
301 Stocker Center
Ohio University
Athens, OH 45701
US

URI: http://irg.cs.ohiou.edu/

Hans Kruse
Ohio University Internetworking Research Group
292 Lindley Hall
Ohio University
Athens, OH 45701
US

URI: http://irg.cs.ohiou.edu/

Shawn Ostermann
Ohio University Internetworking Research Group
322b Stocker Center
Ohio University
Athens, OH 45701
US

URI: http://irg.cs.ohiou.edu/