

6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

R. Sudhaakar, Ed.
Cisco
P. Zand
University of Twente
March 9, 2015

6TiSCH Resource Management and Interaction using CoAP
[draft-ietf-6tisch-coap-03](#)

Abstract

The [[IEEE802154e](#)] standardizes the TSCH mode of operation and defines the mechanisms for layer 2 communication between conforming devices. 6top defines a set of commands to monitor and manage the TSCH schedule. To realize the full functionality of sensor networks and allow their adoption and use in real applications we need additional mechanisms. Specifically, the interaction with 6top, control and modify schedules, monitor parameters etc must be defined. Higher layers monitoring and management entities are then able to use these capabilities to create feedback loops. Although, there have been many custom implementations of such feedback loops between the routing, transport and MAC layers in sensor network deployments, there has been a lack of standards based approaches. This draft defines the messaging between monitoring and management entities and the 6top layer and a mapping to the 6top commands. The document also presents a particular implementation of the generic data model specified in [[I-D.ietf-6tisch-6top-interface](#)] based on CoAP and CBOR.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

Internet-Draft

ietf-6tisch-coap

March 2015

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | | |
|-----------------------------|---|--------------------|
| 1. | Requirements notation | 3 |
| 2. | Introduction | 3 |
| 3. | Scope of the document | 3 |
| 4. | Data Model definition for CoAP | 4 |
| 4.1. | Naming Convention for URI schemes | 4 |
| 4.2. | Convention for accessing URIs | 5 |
| 4.3. | 6TiSCH Resources | 5 |
| 4.3.1. | Versioning | 6 |
| 4.3.2. | Management Resources | 6 |
| 4.3.3. | Informational Resources | 8 |
| 4.3.4. | Message Formats | 9 |
| 4.3.5. | Extensible Resources | 11 |
| 4.4. | Example | 12 |
| 4.4.1. | Request-Response | 12 |
| 4.4.2. | Publish-Subscribe | 13 |
| 5. | References | 14 |
| 5.1. | Normative References | 14 |
| 5.2. | Informative References | 14 |
| 5.3. | External Informative References | 16 |
| Appendix A. | | 16 |
| Authors' Addresses | | 16 |

Internet-Draft

ietf-6tisch-coap

March 2015

1. 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](#)].

2. Introduction

The 6TiSCH Operation Sublayer (6top) [[I-D.ietf-6tisch-6top-interface](#)] describes the main commands provided to higher layers that allow them to build TSCH schedules, make routing decisions, perform TSCH configuration and control procedures and supports centralized and decentralized scheduling policies among other functionalities. However, there is still a need for specifying the methods, including message exchanges and message formats that higher layers use to invoke these command described by 6top.

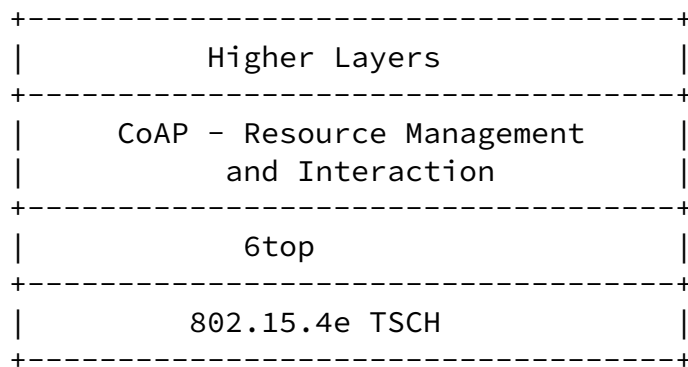


Figure 1: Logical positioning of layers

Interoperation with any protocol that may be used by the network layer is necessary to have a wide impact. This documents aims at defining the message exchanges and the formats of the messages that the network layer uses to interact with the 6top sub-layer. The messaging scheme defined in this document is aimed for use between 6top nodes and higher layer management entities as well as between

6top nodes.

This document also specifies an implementation of this generic message exchange and data model using CoAP as the transport mechanism.

[3.](#) Scope of the document

This draft defines the communication mechanism between PCE and 6top nodes using COAP. The generic YANG data model defined in [[I-D.ietf-6tisch-6top-interface](#)] is used to define the various CoAP messages and payloads. The payload used CBOR for the encoding

format. The document also defines the URIs that used to identify the resources exposed by 6top.

This document also defines how users can install custom resources that allow them to extend the basic resource exposed by 6top.

The CoAP Management Interface (CoMI) [[I-D.vanderstok-core-comi](#)] draft specifies a common constrained device management interface. The conventions used in this draft follow the guidelines in the CoMI draft . This draft expects CoMI to define the access methodologies, discovery mechanisms, resource installation procedures required for the management of constrained devices. This draft presents some examples in [Section 4.3.4](#) on how to use the CoMI specifications to manage the 6top sublayer.

NOTE: CoMI specifications are not finalized at the time of this writing. In case of any discrepancies, CoMI will supersede the message formats in the examples presented in [Section 4.3.4](#).

[4.](#) Data Model definition for CoAP

[4.1.](#) Naming Convention for URI schemes

Universal Resource Identifiers (URIs) help us uniquely identify the various commands and parameters that 6top exposes to the higher layers. The basic URI naming conventions and terminology specified in [[RFC3986](#)] is used. Specifically, the terms, 'scheme', 'authority', 'path', 'query' are used as defined in the [[RFC3986](#)].

The following provides the guidelines that are followed in this draft to name the URIs that identify the resources exposed by 6top.

1. All URIs naming 6top resources MUST use the 'coap' scheme
2. The authority MUST have the username '6top' and the IP address of 6top node
3. The root path MUST always start with '6top'
4. Each component of the path SHOULD be of minimum possible length while being self descriptive.
5. Typographical conventions as described in A SHOULD be followed

These guidelines MUST be followed by users who install extensible resources. It SHOULD be followed for future extensions of the data model in order to provide consistency.

[4.2.](#) Convention for accessing URIs

We use the GET, POST and DELETE methods described by CoAP. These methods MUST be used in accordance with their definition in Sec. 5.8 of [[RFC7252](#)] and as specified in the CoMI draft [[I-D.vanderstok-core-comi](#)]. There is no need for the PUT method as the functionality of the POST method can be used for all situations that need updating or modification of a resource.

The CoAP methods are mapped to 6top commands as shown in the figure below.

| CoAP method | 6top command | Description |
|-------------|--------------------|--------------------------------|
| GET | READ | Retrieves 6top resources |
| POST | CREATE / UPDATE | Creates/Updates a new entry |
| DELETE | DELETE | Deletes an entry |

| | | |
|------|-----------|----------------------|
| POST | CONFIGURE | Configures a setting |
|------|-----------|----------------------|

Figure 2: Mapping between CoAP methods and 6top commands

The GET method may use queries to allow higher layer entities to perform conditional GETs or filter the results of a GET on resource that is a collection.

The POST method is used in all situations where an argument needs to be passed to the 6top layer. The Content-Type option is set to 'application/cbor'. The payload is encoded using CBOR format as described in [[I-D.vanderstok-core-comi](#)] and [[RFC7049](#)].

The DELETE method is used to invoke the 6top DELETE command on a particular resource.

The GET method may use queries to allow higher layer entities to perform conditional GETs or filter the results of a GET on resource that is a collection.

[4.3.](#) 6TiSCH Resources

The 6TiSCH resources presented in this draft offer a comprehensive way to manage 6top nodes based on the requirement known to us as of this writing. These resources are bound to evolve and will be

identified by appropriate version numbers that will be tied to revisions of this draft.

Management resources are classified as resources to which a higher layer entity may create, update or delete. They are typically used to create schedules, identify time sources that TSCH needs. They are the means to close the control loop between TSCH and higher layers.

Informational resources are classified as resources to which a higher layer entity typically has only READ access. They are typically used to monitor operational parameters of TSCH and the values used as input to routing algorithms and other mechanisms.

[4.3.1.](#) Versioning

The version number describes the set of resources that can be accessed on a node that implements the recommendations in this draft.

Each revision of this draft will define a version number which will uniquely identify the set of resources defined in that particular revision of the draft. Specifically, a change to the major version number indicates that resources have been added, deleted, renamed or their message formats have changed. In most cases, this MAY require changes to the implementation. The minor version number indicates changes to options supported by resources or other textual/language changes to the draft. In most cases, this MAY NOT require any changes to the implementation.

The 6TiSCH resource version information is available by executing a GET method on the resource '/6top/version' of a 6top node.

[4.3.2.](#) Management Resources

All the attributes in the management resources have the Read/Write accessibility. The following table lists the 6top management resources and the related URI paths.

| Name | Accessibility 6top Commands | URI path |
|------------------|--------------------------------|----------------|
| Neighbor List | CREATE/READ/ DELETE/UPDATE | 6top/nbrList |
| slotframe | CREATE/READ/ | 6top/slotFrame |

| | | |
|-----------------------|---------------------------------|------------------|
| List | DELETE/UPDATE | |
| +-----+ | +-----+ | +-----+ |
| Cell List | CREATE/READ/ DELETE/UPDATE | 6top/cellList |
| +-----+ | +-----+ | +-----+ |
| Time Source | CREATE/READ/ DELETE/UPDATE | 6top/timeSource |
| +-----+ | +-----+ | +-----+ |
| LabelSwitch List | CREATE/READ/ DELETE/UPDATE | 6top/labelSwitch |
| +-----+ | +-----+ | +-----+ |
| Track List | CREATE/READ/ DELETE/UPDATE | 6top/tracklist |
| +-----+ | +-----+ | +-----+ |
| EB List | CREATE/READ/ DELETE/UPDATE | 6top/ebList |
| +-----+ | +-----+ | +-----+ |
| Chunk List | CREATE/READ/ DELETE/UPDATE | 6top/chunkList |
| +-----+ | +-----+ | +-----+ |

Figure 3: List of Management Resources

The following table provides an example about how Neighbor List components (leafs in the YANG model) can be addressed.

| Field name | URI path |
|----------------------|--------------------|
| Target Neighbor Addr | 6top/nbrList/tna |
| ASN | 6top/nbrList/asn |
| RSSI | 6top/nbrList/rssi |
| LinkQuality | 6top/nbrList/linkQ |

Figure 4: Neighbor Table

4.3.3. Informational Resources

All the attributes in the Informational resources have the Read accessibility. The following table lists the 6top informational resources and the related URI paths.

| Name | Accessibility 6top Commands | URI path |
|--------------------|--------------------------------|------------------|
| Version | READ | 6top/version |
| Queue | READ/CONFIGURE | 6top/queue |
| Monitoring status | READ/CONFIGURE | 6top/monitStatus |
| Statistics metrics | READ/CONFIGURE | 6top/stats |

Figure 5: List of Informational Resources

4.3.3.1. Version

The version resource is a read-only resource that provides information on the methods, resources, message formats that is supported by the node. The version resource does not directly map to a 6top resource defined in [[I-D.ietf-6tisch-6top-interface](#)].

Upon receiving a GET on the '/6top/version' resource, the node MUST respond with a version number that is described by a major and minor

number. It is expressed using 2 bytes - The first and second bytes are 8-bit unsigned integers indicating the major and minor version numbers respectively. The valid values for the major version are 1 through 255 (both inclusive) and for the minor version are 0 through 255 (both inclusive).

A 6top node implementing the recommendations in this draft will respond with the following 2 byte version number - '0x01 0x00', indicating major version = 1 and minor version = 0.

The major and minor versions are separately accessible using the resources '/6top/version/major' and '/6top/version/minor' respectively. The response will be an 8-bit unsigned integer containing the major or minor version number, respectively.

[4.3.3.2.](#) Resource Discovery

As new resources are defined (both native and custom), it is essential for the PCE as well peers to discover the resources. The CoMI draft presents methods by which standard CoAP resource discovery mechanisms are extended to the management of constrained devices. The methods described in Section 4.3 of [[I-D.vanderstok-core-comi](#)] SHALL be used for discovering new resources available at a node.

[4.3.4.](#) Message Formats

NOTE: The message formats presented in this section follow the specifications in the CoMI draft [[I-D.vanderstok-core-comi](#)]. In case of any discrepancies, the CoMI draft will take precedence.

GET messages do not contain any payload. However, they can contain a query option to filter on the resource that is being retrieved. An example query on the neighbor list is:

```

+-----+
Header  | GET                                     |
+-----+
Uri-Path| /6top/nbrList                            |
+-----+
Options | Accept: application/cbor                 |
        | Uri-Query: ABNF(TargetNodeAddr==0x1234) |
+-----+

```

Figure 6: Example GET message

Since this resources points to the entire neighbor list, the response

returns all the entries (the list of neighbors of that node) and all fields in each entry (i.e. entry for a neighbor) of the list in CBOR

format. A request with a Uri-Query option may be used to retrieve only specific entries in the list. The value of Uri-Query MUST be in the ABNF format as described in [[RFC5234](#)].

Resources that point to collection within a list, such as '/6top/nbrList/tna', returns only the values in the TargetNodeAddr entry of the Neighbor list. The usage of the Uri-Query option has the same effect of filtering on the result.

The endpoint MUST appropriately respond with a 2.05 Content or 4.04 Not Found message as defined in [[RFC7252](#)]. If the resource is found then the payload of the response MUST contain a CBOR representation of the data that is referenced by the URI.

To create or update a Neighbor, the CoAP client MUST send a POST message as shown in Figure 7. The payload MUST describe the argument that is passed to 6top in CBOR format.

```
+-----+
Header  | POST                               |
+-----+
Uri-Path| /6top/nbrList                         |
+-----+
Payload | CBOR( {TargetNodeAddr: 0x1234} )    |
+-----+
```

Figure 7: Example POST message

The POST method may not be used on resources that are collection within a list, such as '/6top/nbrList/tna'.

To delete a Neighbor, the CoAP client MUST send a DELETE message as shown in Figure 8.

```
+-----+
Header  | DELETE                               |
+-----+
Uri-Path| /6top/nbrList                         |
+-----+
```

```
Options | Uri-Query: ABNF(TargetNodeAddr |
        |                               == 0x1234) |
+-----+-----+-----+-----+
```

Figure 8: Example DELETE message

A DELETE message SHOULD always contain a Uri-Query option in order to clearly specify which entry(s) within the list must be deleted. Ideally, the CoAP client SHOULD make one call per entry that must be

deleted. An implementation may decide whether or not a DELETE method on '/6top/nbrList' may be allowed.

The endpoint MUST appropriately respond with a 2.02 (Deleted) message.

A sample of mapping between CoAP methods and 6top commands for manipulating the neighbor list is shown in the figure below.

| CoAP method | 6top command | 6top behaviour | CoAP Response |
|---|------------------------------------|-------------------------------|--|
| POST /6top/nbrList CBOR({TargetNodeAddr: 1234}) | Create.neighbor (address,stats) | Adds a neighbor | 2.01 Created |
| GET /6top/nbrList | Read.all. neighbor() | Reads all neighbors | 2.05 Content CBOR(Neigh- bor List) |
| GET /6top/nbrList Uri-Query - TargetNodeAddr: 1234}) | Read.neighbor (address) | Reads neighbor information | 2.05 Content CBOR(Neigh- bor List) |
| POST /6top/nbrList CBOR({TargetNodeAddr: 1234}) | Update.neighbor (address,stats) | Updates an entry | 2.04 Changed |
| DELETE /6top/nbrList | Delete.neighbor | Removes | 2.02 Deleted |

```

| Uri-Query -          | (address)          | the neighbor |          |
| TargetNodeAddr      |                     |              |          |
|           == 1234}) |                     |              |          |
+-----+-----+-----+-----+

```

Figure 9: CoAP methods and resulting invocation 6top commands

4.3.5. Extensible Resources

Extensible resources are to be used when a higher layer entity wants to be notified of an event. An event may be defined as the result of a mathematical operation on a 6top resource. For example, the CoAP client might want to monitor when the DAG rank of a particular node crosses a threshold. Once the extensible resource is installed the CoAP client uses the observe mechanism defined in [\[I-D.ietf-core-observe\]](#) to monitor the resource.

4.3.5.1. Defining new resources

An extensible resource path MUST always start with '/6top/custom' and follow the guideline for URI naming as described in 4.1. The event associated with the extensible resource must be defined using the ABNF notation described in [\[RFC5234\]](#).

An extensible resource may be created by performing POST operation to the resource '/6top/custom' with the following payload encoded using CBOR.

```

+-----+-----+
| Field Name | Type |
+-----+-----+
| Resource   | String |
| Name       |        |
+-----+-----+
| Event      | String |
| Definition |        |
+-----+-----+

```

Figure 10: Payload format for creating an Extensible Resource

4.4. Example

This section gives a number of short examples of how to use the data model and CoAP mapping defined in this document.

4.4.1. Request-Response

Figure 11 shows how a CoAP client adds an entry in the neighbor list of node A. This new neighbor has a target node address 0x1234. The client sends out a POST request containing the CBOR encoding of '{TargetNodeAddr: 1234}'. This message is received and processed by the CoAP endpoint of Node A and in turn, the 6top command, Create.neighbor is invoked with the appropriate parameters. In this case, the address is the 'TargetNodeAddr' parameter passed in the payload of the POST message and the stats argument has the default value. In the response to the invocation of the Create.neighbor command, the 6top sublayer adds an entry to the neighbor list with appropriate values and returns a confirm message. The CoAP endpoint in turn send out an appropriate CoAP response to indicate success. If the addition of the neighbor failed, a failure message will be returned.

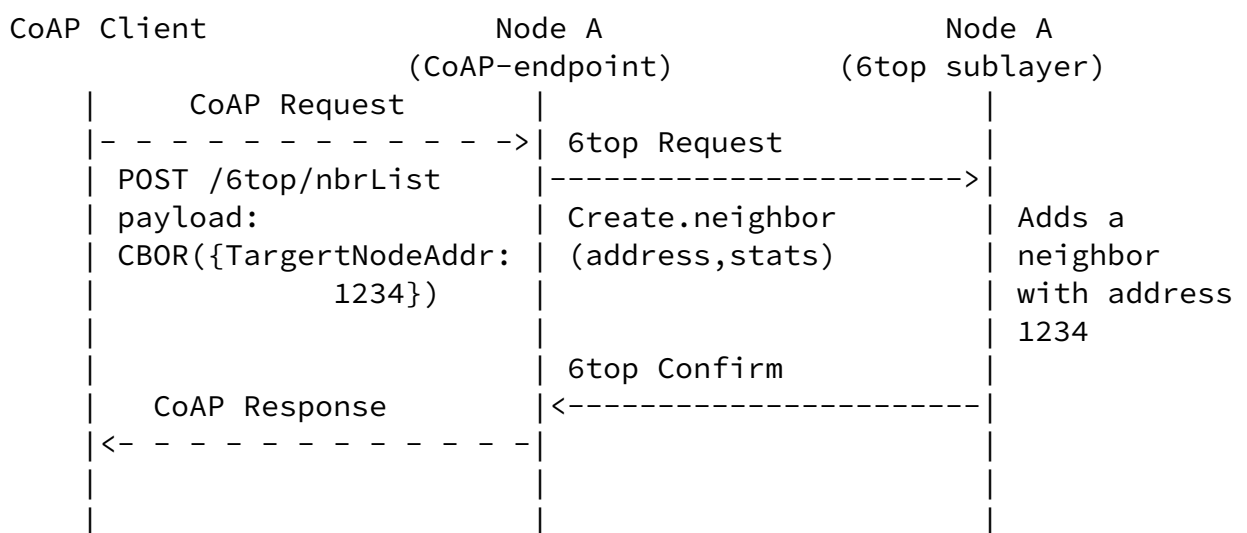


Figure 11: Example of adding a neighbor

In Figure 12, a CoAP client reads a neighbor entry from node A. This

neighbor has a target node address 0x1234.

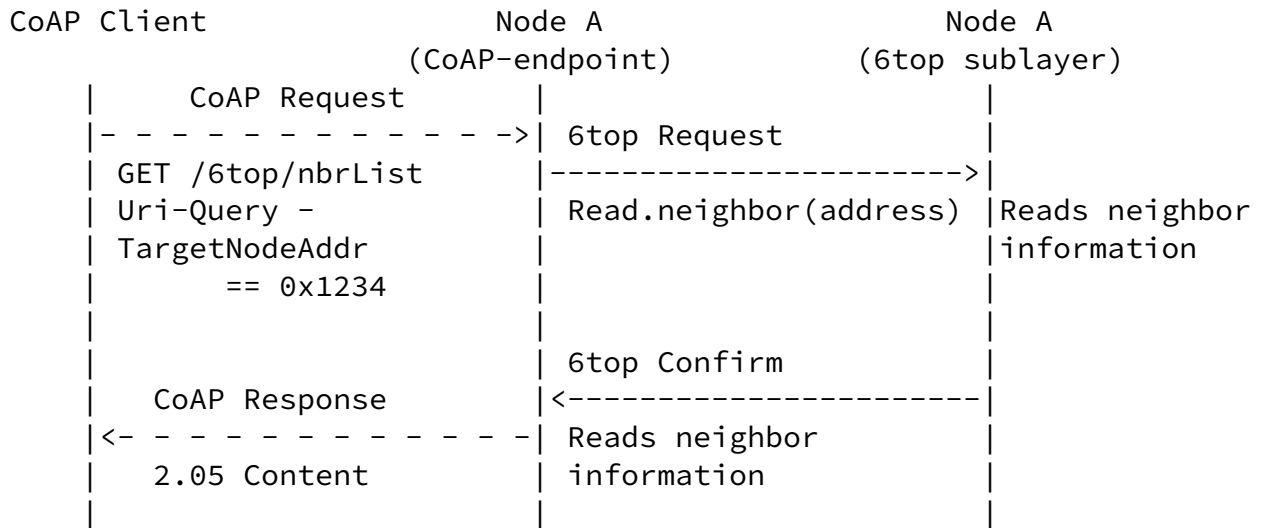
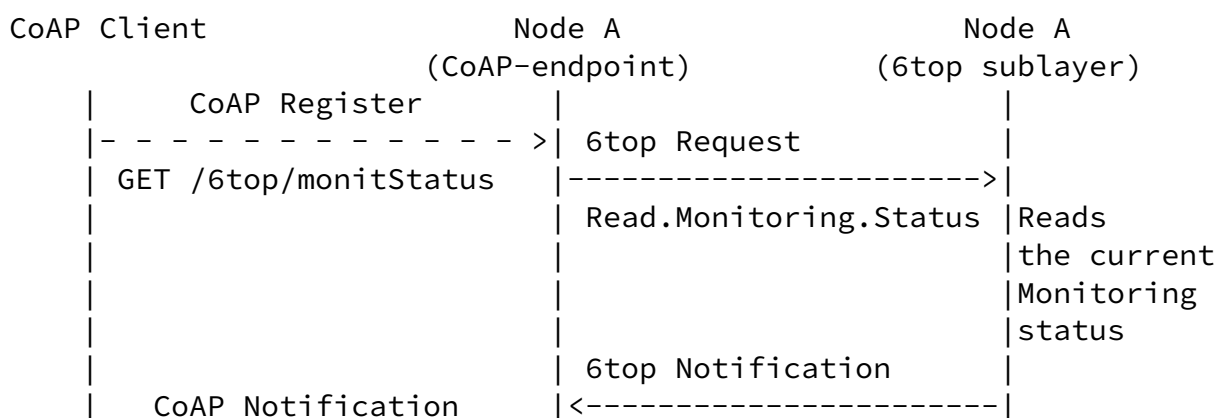


Figure 12: Example of reading a neighbor

4.4.2. Publish-Subscribe

In Figure 13, a CoAP client subscribes to Monitoring Status of node A. The Monitoring status of Node A is constantly monitored by the CoAP client.



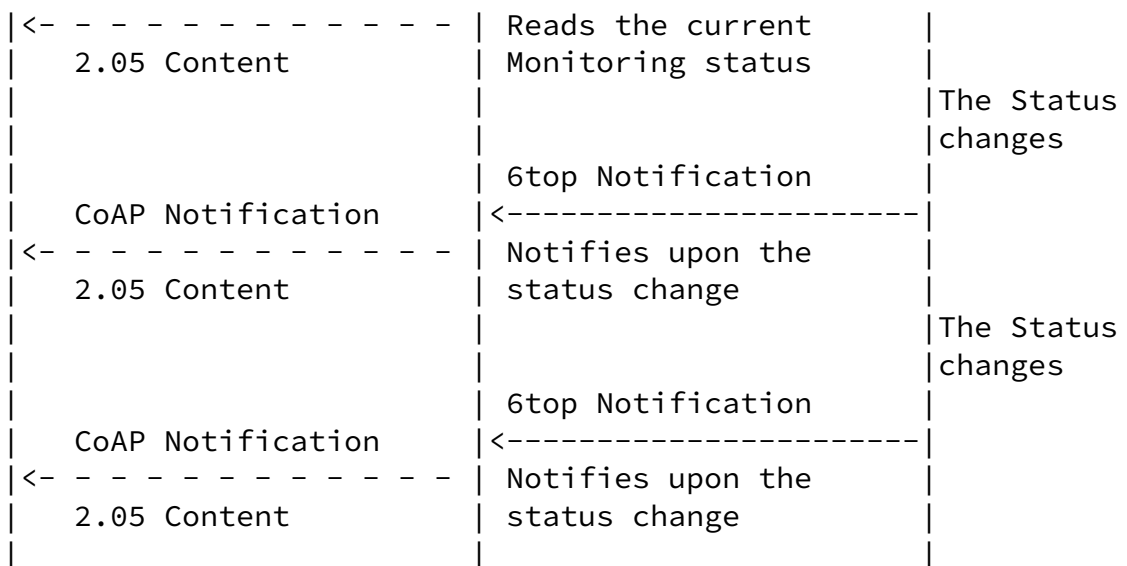


Figure 13: Example of Subscribing to Monitoring Status

5. References

5.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

5.2. Informative References

[I-D.ietf-6tisch-6top-interface]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", [draft-ietf-6tisch-6top-interface-02](#) (work in progress), October 2014.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", [draft-ietf-6tisch-architecture-06](#) (work in progress), March 2015.

[I-D.ietf-6tisch-minimal]
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", [draft-ietf-6tisch-minimal-06](#) (work in progress), March 2015.

- [I-D.ietf-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", [draft-ietf-6tisch-terminology-03](#) (work in progress), January 2015.
- [I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", [draft-ietf-6tisch-tsch-05](#) (work in progress), January 2015.
- [I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", [draft-ietf-core-observe-16](#) (work in progress), December 2014.
- [I-D.vanderstok-core-comi]
Stok, P., Greevenbosch, B., Bierman, A., Schoenwaelder, J., and A. Sehgal, "CoAP Management Interface", [draft-vanderstok-core-comi-06](#) (work in progress), February 2015.
- [I-D.wang-6tisch-6top-sublayer]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", [draft-wang-6tisch-6top-sublayer-01](#) (work in progress), July 2014.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", [RFC 7049](#), October 2013.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), June 2014.

[5.3.](#) External Informative References

[IEEE802154e]

IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.

Appendix A.

Guidelines for constructing URI path names:

1. The first letter of each element of the path SHOULD be capitalized
2. If an element has multiple words, each the first letter of each work SHOULD be capitalized

Authors' Addresses

Raghuram S Sudhaakar (editor)
Cisco Systems, Inc
Building 24
510 McCarthy Blvd
San Jose 95135
USA

Phone: +1 408 853 0844
Email: rsudhaak@cisco.com

Pouria Zand
University of Twente
Department of Computer Science
Zilverling Building
Enschede 7522 NB
The Netherlands

Phone: +31 619040718
Email: p.zand@utwente.nl

