6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: September 30, 2014

R. Sudhaakar, Ed.
Cisco
P. Zand
University of Twente
March 29, 2014

6TiSCH Resource Management and Interaction using CoAP
draft-sudhaakar-6tisch-coap-01

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, we need to define how to interact with 6top, control and modify schedules, monitor parameters etc. 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. The goal of the memo is to define 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.wang-6tisch-6top-interface] based on CoAP and CBOR.

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 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 30, 2014.

Table of Contents

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.wang-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.

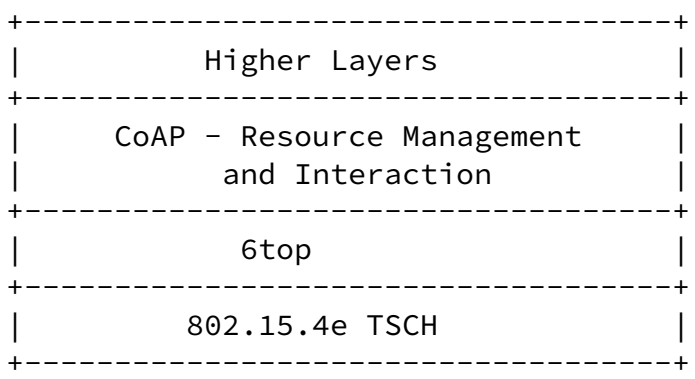


Figure 1: Logical positioning of layers

In order to have an wide impact we need to be able to interoperate with any protocol that may be used by the network layer. 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 adn 6top nodes using COAP. We use the generic YANG data model defined in [I-D.wang-6tisch-6top-interface] 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.

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. We use the basic URI naming conventions and terminology specified in [RFC3986]. 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 '6t'

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 [I-D.ietf-core-coap]. We have 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.bormann-cbor].

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

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. 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	6t/Neighbor
slotframe List	CREATE/READ/ DELETE/UPDATE	6t/slotframe
Cell List	CREATE/READ/ DELETE/UPDATE	6t/Cell
Time Source	CREATE/READ/ DELETE/UPDATE	6t/TimeSource

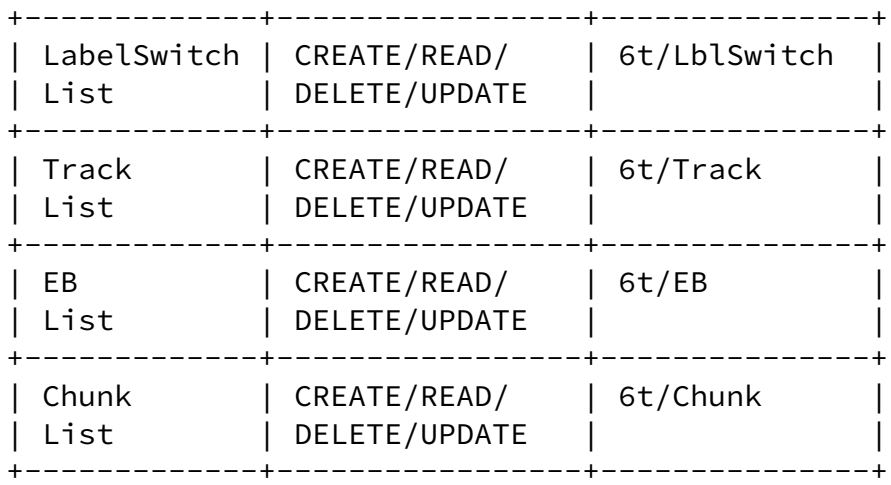


Figure 3: List of Management Resources

In the following table, we provide an example about how Neighbor List components (leafs in the YANG model) can be addressed.

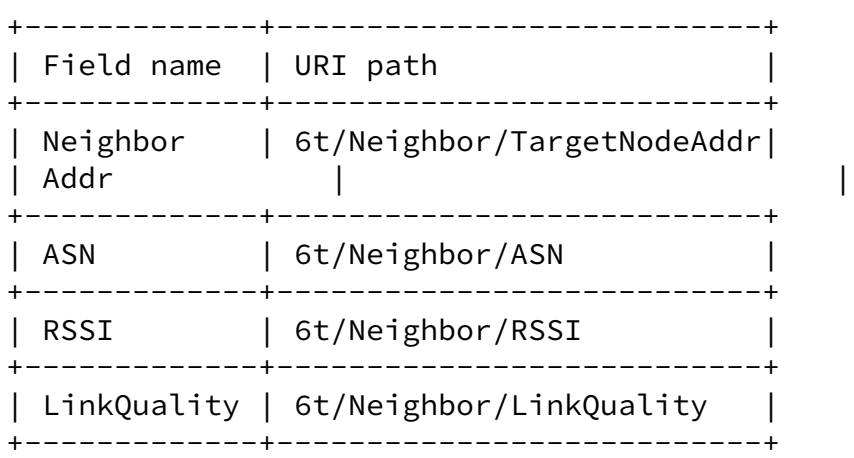
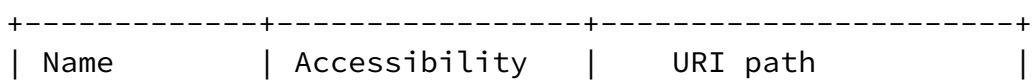


Figure 4: Neighbor Table

4.3.2. 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.



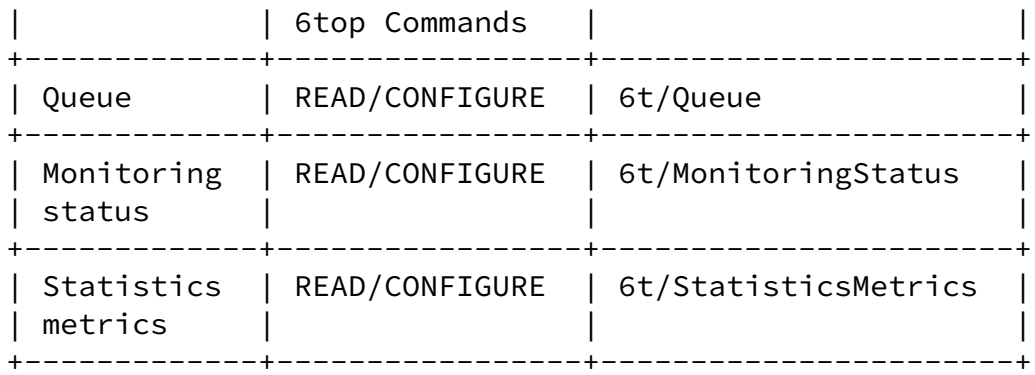


Figure 5: List of Informational Resources

4.3.3. Message Formats

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 table is:

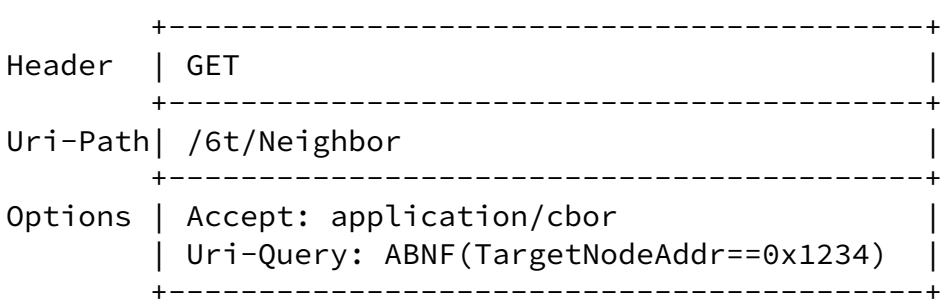


Figure 6: Example GET message

Since this resources points to the entire neighbor table the response returns all the rows (the list of neighbors of that node) and all fields in each row (i.e. entry for a neighbor) of the table in CBOR format. A request with a Uri-Query option may be used to retrieve only specific rows in the table. The value of Uri-Query MUST be in the ABNF format as described in [RFC5234].

Resources that point to collection within a table, such as '/6t/Neighbor/TargetNodeAddr', returns only the values in the

TargetNodeAddr column of the Neighbor table. 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 [I-D.ietf-core-coap]. 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.

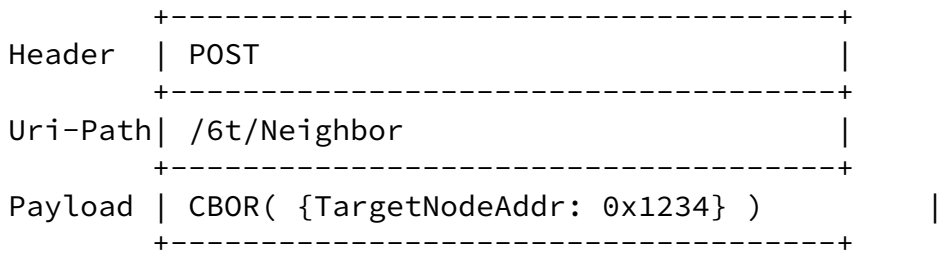


Figure 7: Example POST message

The POST method may not be used on resources that are collection within a table, such as '/6t/Neighbor/TargetNodeAddr'.

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

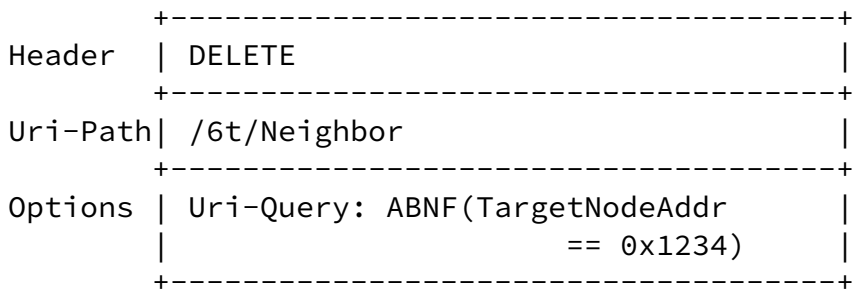


Figure 8: Example DELETE message

A DELETE message SHOULD always contain a Uri-Query option in order to clearly specify which row(s) within the table must be deleted. Ideally, the CoAP client SHOULD make one call per row that must be deleted. An implementation may decide whether or not a DELETE method on '/6t/Neighbor' 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 /6t/Neighbor CBOR({TargetNodeAddr: 1234})	Create.neighbor (address,stats)	Adds a neighbor	2.01 Created
GET /6t/Neighbor	Read.all. neighbor()	Reads all neighbors	2.05 Content CBOR(Neigh- bor Table)
GET /6t/Neighbor Uri-Query - TargetNodeAddr: 1234})	Read.neighbor (address)	Reads neighbor information	2.05 Content CBOR(Neigh- bor Table)
POST /6t/Neighbor CBOR({TargetNodeAddr: 1234})	Update.neighbor (address,stats)	Updates an entry	2.04 Changed
DELETE /6t/Neighbor Uri-Query - TargetNodeAddr == 1234})	Delete.neighbor (address)	Removes the neighbor	2.02 Deleted

Figure 9: CoAP methods and resulting invocation 6top commands

4.3.4. 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.

4.3.4.1. Defining new resources

An extensible resource path MUST always start with '/6t/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 '/6t/custom' with the following payload encoded using CBOR.

Field Name	Type
Resource Name	String
Event Definition	String

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 table 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 table with appropriate values and returns a confirm message. The CoAP endpoint in turn send out an appropriate CoAP response to indicate success. In situation where the addition of the neighbor failed, a failure

message will be returned.

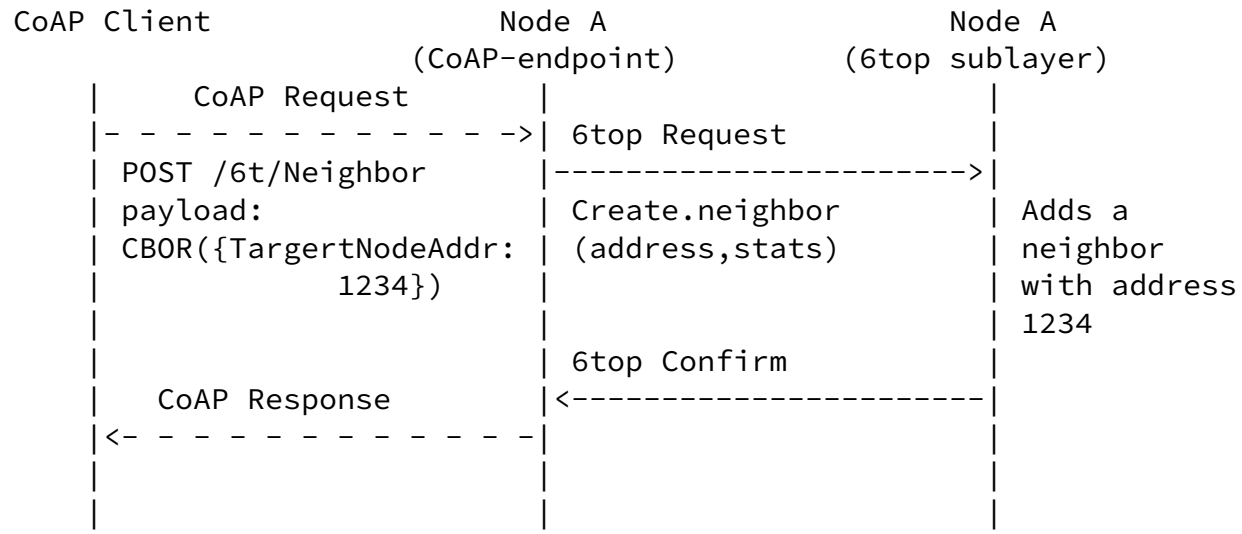
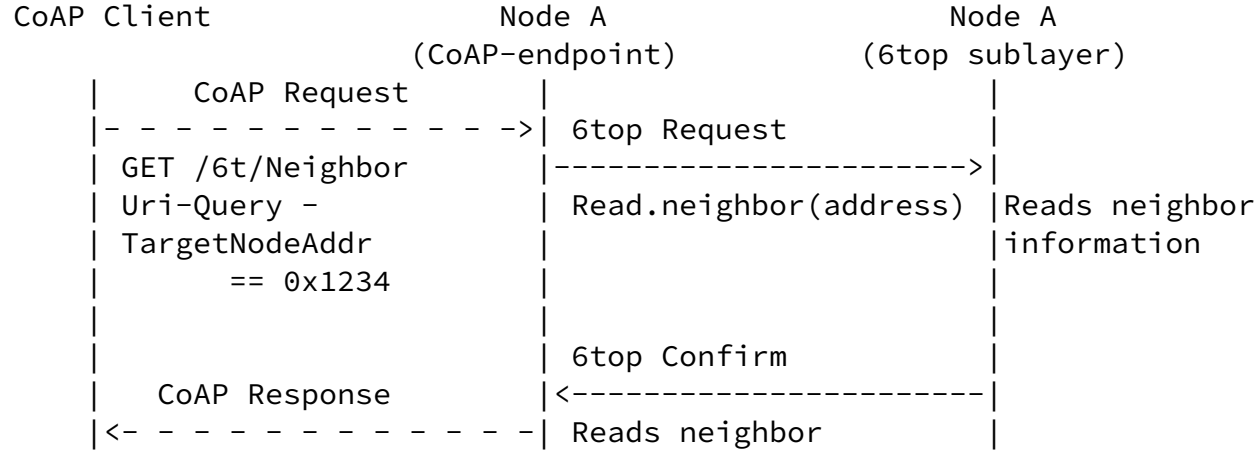


Figure 11: Example of adding a neighbor

In Figure 12, a CoAP client reads a neighbor entry from node A. This neighbor has a target node address 0x1234.



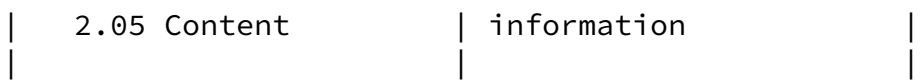
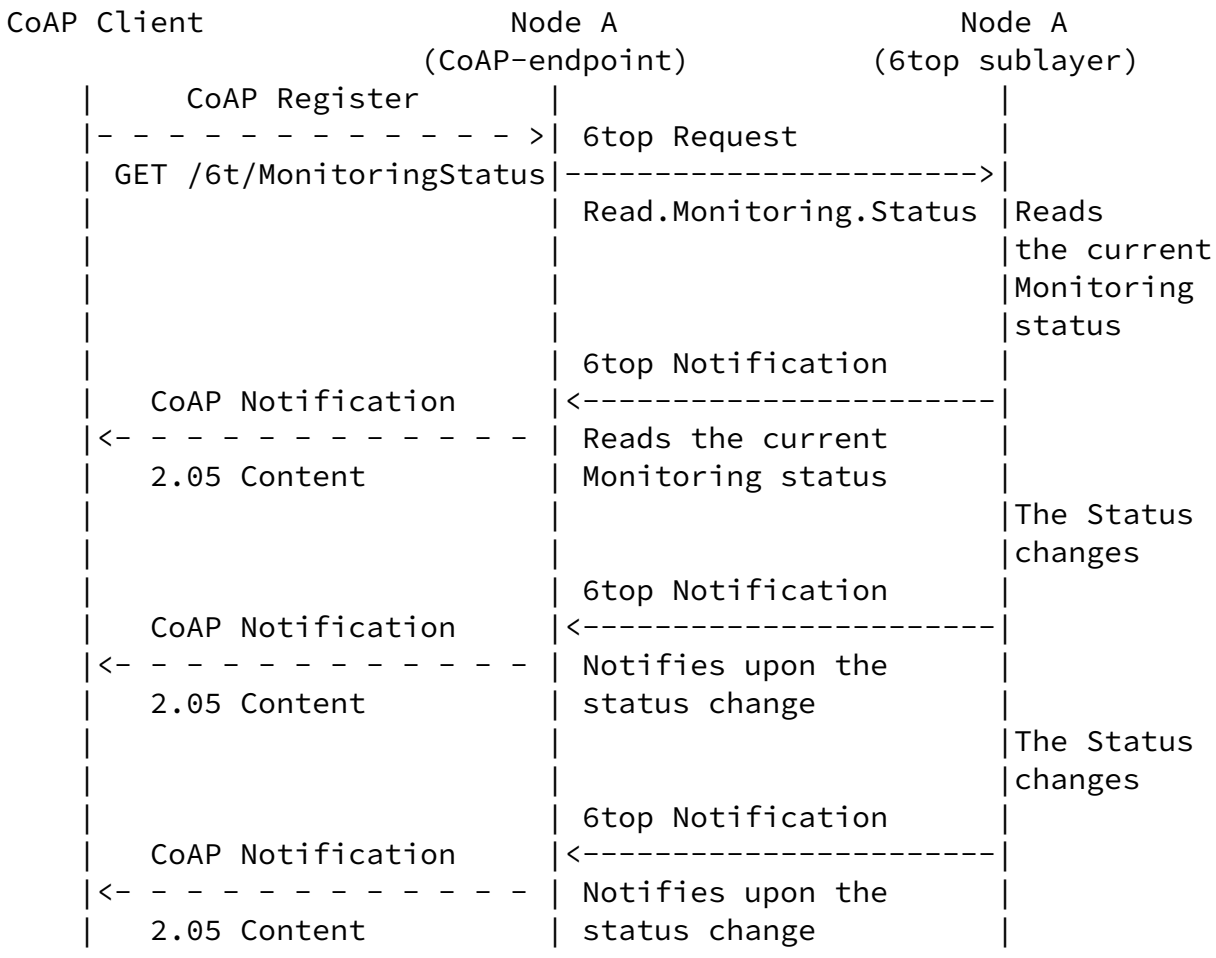


Figure 12: Example of reading a neighbor

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.bormann-cbor]
Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", draft-bormann-cbor-09 (work in progress), September 2013.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-11 (work in progress), October 2013.

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

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