

CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 31, 2013

C. Ma
P. Hong
K. Xue
USTC
July 30, 2012

Stateful Observation in CoAP
draft-ma-core-stateful-observe-00

Abstract

The Observe Option allows a CoAP client to observe changes in the state of resources and obtain a current representation of the last resource state. To be an observer of an origin server's resources, the client is required to register its interest with the server. A successful registration will make the client added into the server's observation list, while a failed one may drive the client to re-register.

However, even repeated and frequent re-registrations cannot guarantee the client to eventually become an observer of the target server. In the case that the server is unable or unwilling to accept an observer, the time-intensive re-registrations will just bring redundant messages in the constrained network and considerable energy consumption on both the client and the server.

This memo defines a new CoAP option, State, for providing stateful observation on the resources of CoAP servers. By observing the state of the server in terms of the Observe Option, a client can explicitly judge when the server will not actively reject an observation registration, and wisely performs the re-registration. This avoids the potential registration flooding that causes considerable network overhead and energy consumption on the constrained nodes.

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 January 31, 2013.

Copyright Notice

Copyright (c) 2012 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	4
1.1.	Background	4
1.2.	Protocol overview	4
1.3.	Terminology	5
2.	The State Option	5
3.	Using the State Option	6
3.1.	State notifications	7
3.2.	State option in request	7
3.3.	State option in initial response	7
3.4.	State option in state notifications	8
3.5.	Cancellation	8
4.	Intermediaries	8
5.	Examples	9
6.	Security Considerations	14
6.1.	Amplification attacks	14
6.2.	IP address spoofing attacks	14
7.	IANA Considerations	14
8.	Acknowledgements	14
9.	References	15
9.1.	Normative reference	15
9.2.	Informative reference	15
	Authors' Addresses	15

1. Introduction

1.1. Background

CoAP [I-D.ietf-core-coap] is an Application Protocol for Constrained Nodes/Networks. The observe [I-D.ietf-core-observe] specification describes a subject/observer pattern to establish an observation relationship between a CoAP client and a CoAP server. In this way, the CoAP client will be notified whenever the state of the observed resource changes. Furthermore, to provide the CoAP client with tailored resources, the conditional observe [ID.li-core-conditional-observe] specification proposes to implement frequently occurring conditional observations by virtue of a new CoAP Condition Option.

In both of the specifications, the client is required to register its interest with the server. A successful registration will make the client added into the server's observation list, while a failed one may drive the client to re-register. If the registration is frustrated by factors such as network congestion and transmission error, the client still has chances to become an observer by re-registering with the server. In the case that the registration failure results from the target server's incapability or unwillingness, however, time-intensive re-registrations will not work but just cause redundant messages in the constrained network and waste the constrained energy of both the client and the server.

1.2. Protocol overview

This memo provides a mechanism for CoAP clients to explicitly learn when the server will not actively reject an observation registration, and to wisely perform a re-registration. This is accomplished through a new CoAP option "State", which defines three states of a CoAP server in terms of the "Observe" option (observation states) as follows.

STATE 0: The CoAP server can add a CoAP client in the observation list;

STATE 1: The CoAP server cannot add a CoAP client in the observation list, but into the candidate queue, which is maintained by the server for notifying the members about the changed observation state;

STATE 2: The CoAP server is unwilling or unable to accept a CoAP client as an observer or a candidate.

A CoAP client who wishes to observe the resources of a CoAP server can add the State Option along with the Observe Option in the initial registration request. If the server is in STATE 0, the client will

be successfully added into the observation list and will become an observer of the resources of the server. If the server is in STATE 1, the client will become a candidate, and thus can re-register with the server when it is notified that the server returns to STATE 0. This case is a typical stateful observation pattern as shown in Figure 1. If the server is in STATE 2, the client temporarily quits registering with the server.

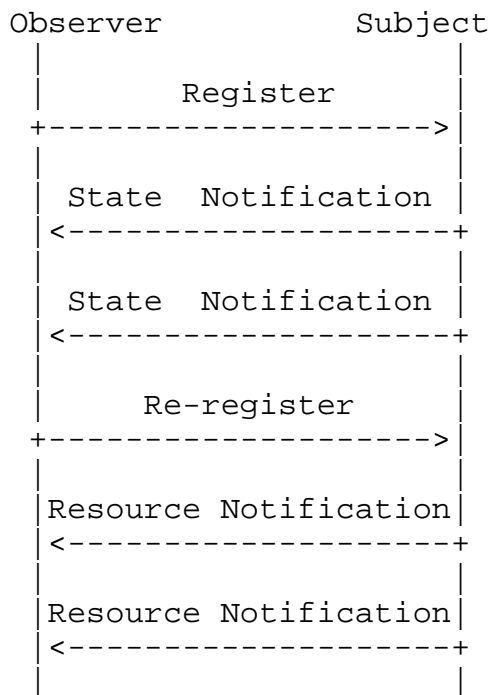


Figure 1: Stateful Observation Design Pattern

1.3. Terminology

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. The State Option

Type	C/E	Name	Data type	Length	Default
30	E	State	uint	1-2B	(none)

Table 1: State Option Format

The State Option can be presented in both request and response messages. In this document, the State Option MUST be used together with the "Observe" Option and MAY be used with the "Max-age" Option.

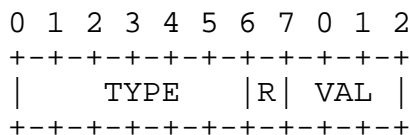


Figure 2: State Option Value

TYPE: The type indicates which option is described by the State Option. In this document, the option number of the described Observe Option is 10, therefore, the type should be 001010. It is set to be a 6-bit integer for option description extension use.

R: The reliability flag indicates whether a state notification needs to be sent in non-confirmable (0) form or in confirmable (1) form.

VAL: The state value that is represented by the VAL flag is used to indicate the state of a CoAP server in terms of an option. For instance, the state values combined with the Observe Option are 0, 1, and 10, referring to the three observation states of an observable CoAP server that are summarized in TABLE 2. The maximal length of VAL is set to be 3 bits for extension use. The VAL part of a State Option in the request message MUST be vacant.

State Type	State Value	Indication
1010	0	Observation list is available
1010	1	Observation list is unavailable Candidate queue is available
1010	10	Both are unavailable

Table 2: State Indication

3. Using the State Option

3.1. State notifications

If a CoAP client is added in the observation list of a CoAP server, it will receive resource notifications as described in [I-D.ietf-core-observe]. If the client is added in the candidate queue, however, it will receive state notifications describing the observation state of the server. A state notification includes an Observe Option, a State Option, a Token Option that matches the token specified by the client in the GET request, and a Max-age Option that indicates the state notification delivery interval, and MAY contain the payload which is in reply to the GET request.

3.2. State option in request

If a CoAP client wants to establish an observation relationship with a CoAP server, it sends a GET request with an Observe Option, a State Option and a Token Option to the server. Specifically, the TYPE of the State Option is 1010, indicating that the client wants to observe the observation state of the server, and the VAL part of the State Option in the request message MUST be vacant.

3.3. State option in initial response

A CoAP server keeps an observation list and a candidate queue. When the CoAP server receives the request, it performs as follows.

It firstly checks the state of its observation list and the candidate queue.

If the server is in STATE 0, it adds the client in the observation list, and responds as the same way described in [I-D.ietf-core-observe], which excludes the State Option.

If the server is in STATE 1, it adds the client in the candidate queue and sends a response containing the payload in reply to the GET request, the Observe Option, the Token Option, the Max-age Option, and the State Option with the TYPE part of 1010 and the VAL part of 1. Note that the Max-age Option is set for indicating the life time of a state notifications, rather than the payload, and its default value is 60s.

If the server is in STATE 2, it sends a response containing the payload in reply to the GET request, the Observe Option, the Token Option, and the State Option with the TYPE part of 1010 and the VAL part of 10. The client will not be added into the candidate queue and will not receive subsequent state notifications.

3.4. State option in state notifications

The server MUST choose an appropriate max-age for the state notification.

The server sends the state notification with the VAL part of 1 to the observation candidates whenever the last state notification has been sent over the max-age, and notifies the candidates when its observation state changes to be State 0.

Upon receiving the state notification with the VAL part of 0, the candidate re-registers with the server by sending a GET request with the Observe Option, the State Option and a new token.

The number (n) of the responsive candidates MAY be larger than that (N) of the observers the server can additionally add in the list. In this case, the server just simply chooses the top N candidates whose responses are firstly received by it. The rest n-N responders still stay in the candidate queue. A former candidate learns whether the former re-registration succeeds through the subsequent notification. Namely, a subsequent rescure notification indicates a successful re-registration, while a subsequent state notification indicates a failed re-registration.

3.5. Cancellation

When a candidate wants to leave the candidate queue, it can reject a confirmable notification with a RST message or perform a GET request without a State Option for a currently observed state. In both of the two cases, the server will remove the client from the candidate queue.

If the server wants to dismiss a candidate, it SHOULD set the VAL part of the state notification to be 10 and choose a large max-age for the state notification. Otherwise, the server SHOULD NOT send the state notification with the VAL part of 10 to the candidates.

4. Intermediaries

It is possible that there is an intermediary between an origin server and the CoAP client that is interested in a resource in the namespace of the server. In this case, the client registers its interest with the intermediary towards the origin server, acting as if it was communicating with the origin server itself as specified in Section 3. The intermediary provides the client with a current representation of the target resource and sends (state) notifications upon changes to the resource (or observation) state, as an origin

server as specified in Section 3. To accomplish this, the intermediary SHOULD make use of the protocol specified in this document, taking the role of the client and registering its own interest in the target resource with the original server.

If there is more than one CoAP-to-CoAP intermediary between the CoAP client and the origin server, the first intermediary towards the origin server is in charge of providing the client with a current representation of the target resource and sends (state) notifications upon changes to the resource (or observation) state, as an origin server as specified in Section 3. The intermediary transmits the GET request from the client hop by hop. If the next hop does not return a response with an Observe Option, the intermediary MAY resort to polling the next hop, or MAY itself return a response without an Observe Option. The last intermediary towards the server transmits the request to the server as if it is the client, or responds to the request with the storage in its cache.

Additionally, the intermediary can also register in several target resources with the original servers that it can directly reach, and keep its own cache up to date. Whenever there is a GET request with an Observe Option and a State Option from the client (or other intermediary), the intermediary responds to the client with the storage in its cache.

5. Examples

This section gives a number of examples to illustrate the use of State Option in a GET request.

The first example (Figure 3) shows that a registration that is carried in the GET request with Observe Option and State Option makes the client added in the observation list of the server. The client receives resource notifications of the current state and upon a state change.

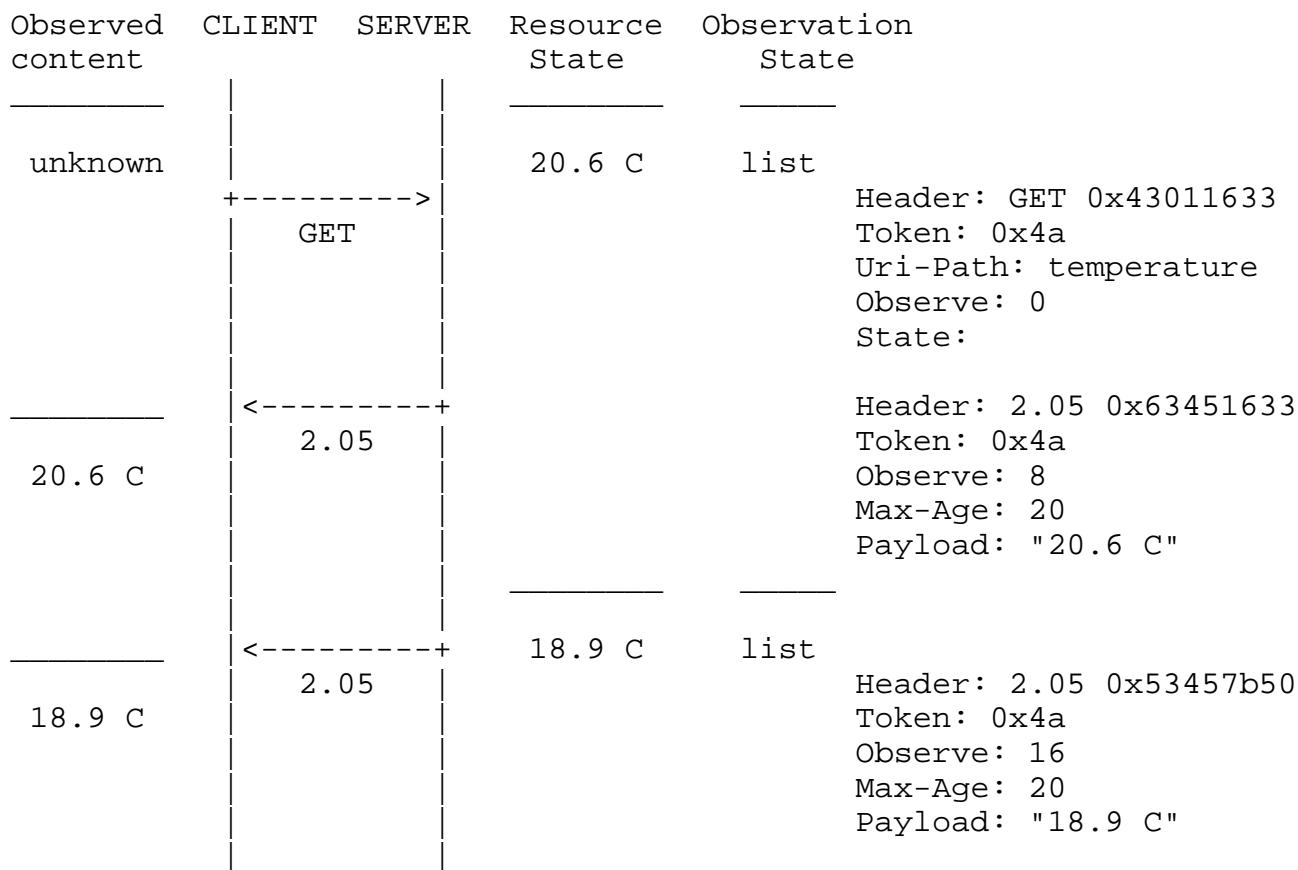


Figure 3: The client is added in the observation list after a registration

The example in Figure 4 shows that a registration that is carried in the GET request with Observe Option and State Option makes the client added into the candidate queue of the server. The client receives a state notification of the current observation state and upon a state change. When the client learns the server is in STATE 0 through the state notification, the client re-registers with the server and then is added into the observation list.

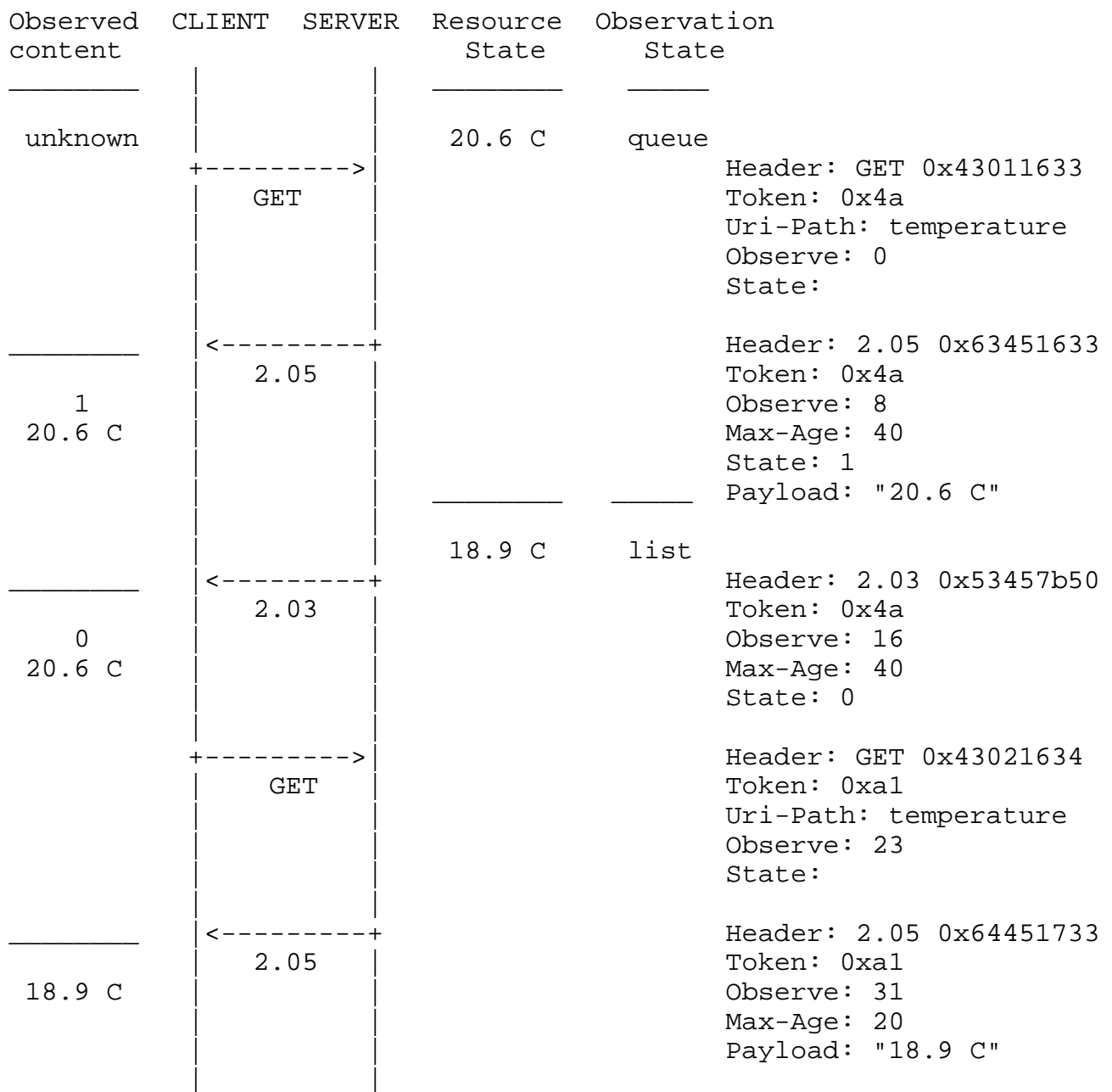


Figure 4: The client is added into the candidate queue after the initial registration and is added into the observation list after the re-registration

This example (Figure 5) shows that the client that is in the candidate queue hasn't received a state notification over a max-age time of 40s, and thus re-registers with the server.

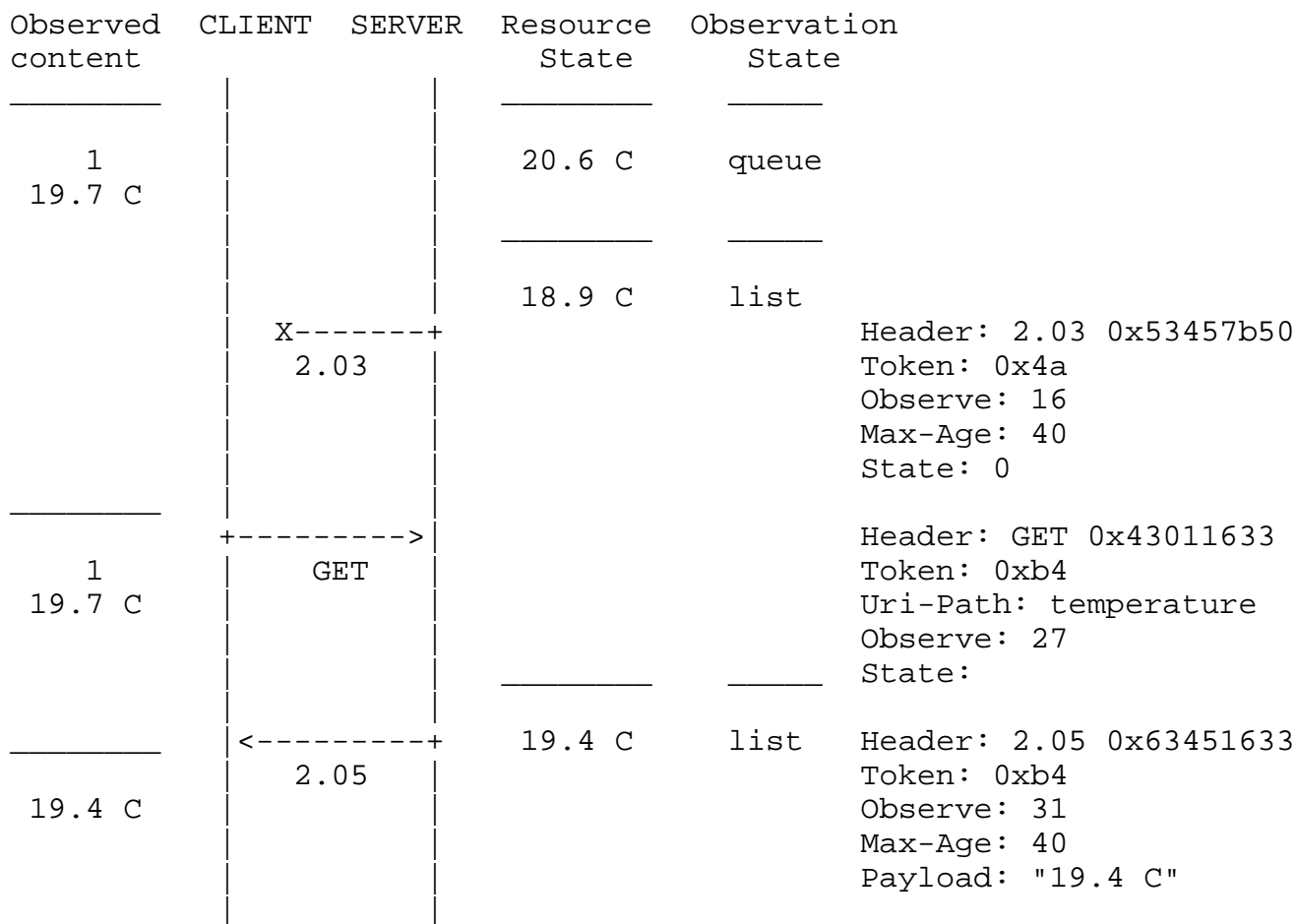


Figure 5: The client re-registers after Max-Age ends

The example in Figure 6 shows that a proxy registers its interest with the server, and receives state notifications to keep its cache up to date. The client gets state notification from the proxy and registers with the server (via the proxy) when the server returns to State 0.

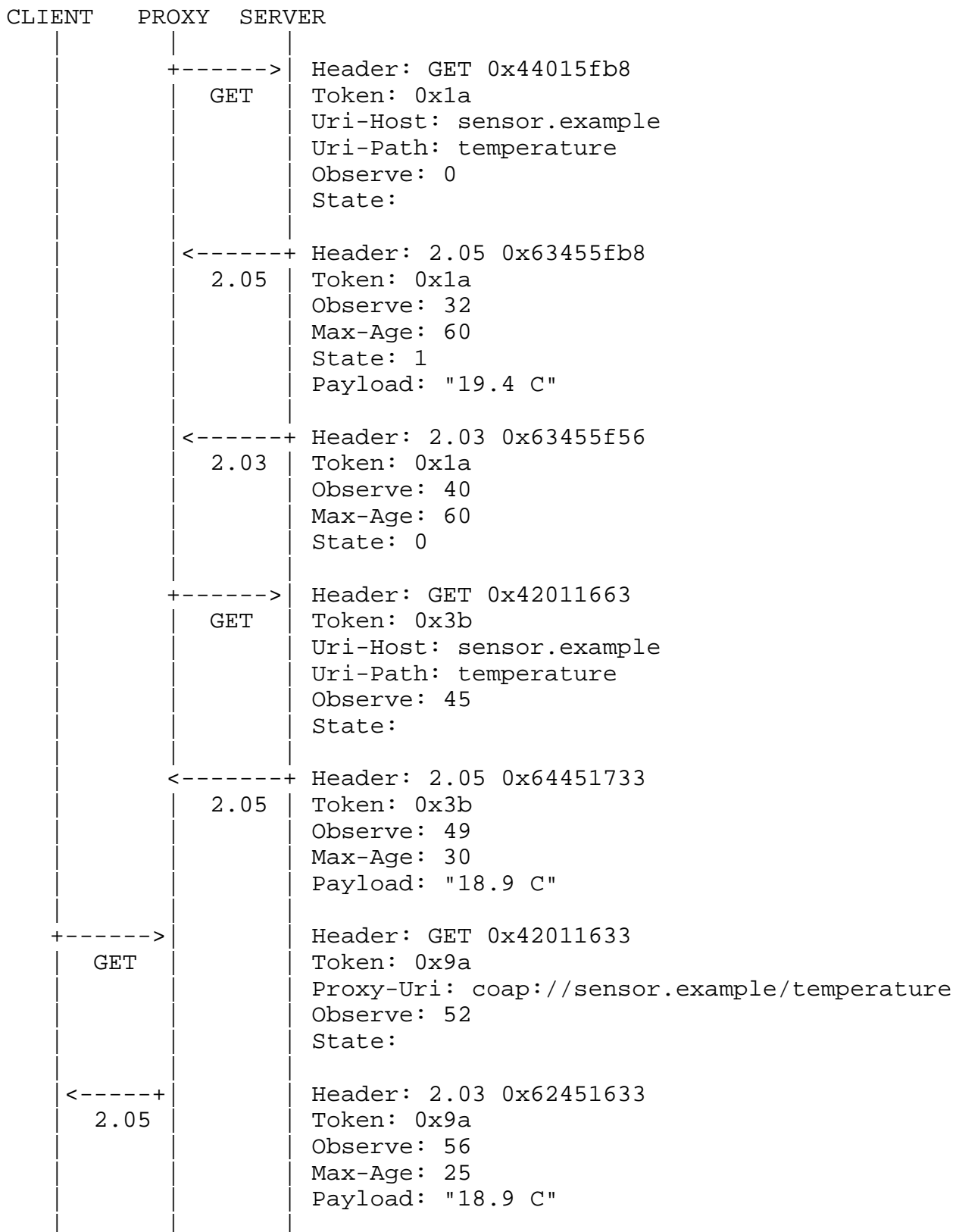


Figure 6: A proxy registers with the server and receives state notification to keep its cache up to date

6. Security Considerations

6.1. Amplification attacks

The amplification attacks towards observing the observation state are more mitigatory compared to observing resources, since the state notification has smaller message size and lower frequency than a resource notification due to the lack of resource representation payload and a larger max-age. Nonetheless, to further reduce the harmful effects caused by the amplification attacks, the state notifications sent in non-confirmable messages are recommended to be interspersed with confirmable messages.

6.2. IP address spoofing attacks

Spoofing (state) notifications or RST in the response to a (state) notification in a CON message, MAY make a client "deaf". For instance, if a state notification with "VAL=0" is tampered by the spoofing attacker to be a state notification with "VAL=10" and a large max-age, the client will be spoofed to be waiting for a long time. This kind of attack can be prevented using security modes other than NoSec.

7. IANA Considerations

The following entries are added to the CoAP Option Numbers registry of [I-D.ietf-core-coap]:

Number	Name	Reference
30	State	[RFCXXXX]

8. Acknowledgements

Thanks to the support from the National Science and Technology Major Project of China under Grant No.2011ZX03005-006.

9. References

9.1. Normative reference

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., Bormann, C., and B. Frank,
"Constrained Application Protocol (CoAP)",
draft-ietf-core-coap-10 (work in progress), June 2012.

[I-D.ietf-core-observe]

Hartke, K., "Observing Resources in CoAP",
draft-ietf-core-observe-05 (work in progress), March 2012.

9.2. Informative reference

[I-D.li-core-conditional-observe]

Li, S., Hoebeke, J., and A. Jara, "Conditional observe in
CoAP", draft-li-core-conditional-observe-02 (work in
progress), June 2012.

Authors' Addresses

Changsha Ma
USTC
Room 307, EEIS Department, USTC West Campus
Shushan District, Hefei, Anhui 230027
P. R. China

Phone: +86-551-3601334
Email: machangs@mail.ustc.edu.cn

Peilin Hong
USTC
Room 305, EEIS Department, USTC West Campus
Shushan District, Hefei, Anhui 230027
P. R. China

Phone: +86-551-3601334
Email: plhong@ustc.edu.cn

Kaiping Xue
USTC
Room 305, EEIS Department, USTC West Campus
Shushan District, Hefei, Anhui 230027
P. R. China

Phone: +86-551-3601334
Email: kpxue@ustc.edu.cn