

# Observe Notifications as CoAP Multicast Responses

*draft-ietf-core-observe-multicast-notifications-11*

**Marco Tiloca**, RISE  
Rikard Höglund, RISE  
Christian Amsüss  
Francesca Palombini, Ericsson

IETF 122 meeting – Bangkok – March 18<sup>th</sup>, 2025

# Recap

## › Observe notifications as multicast responses

- Many clients observe the same resource on a server (e.g., pub-sub)
- Improved performance due to multicast delivery

## › Clients configured by the server, with a 5.03 error informative response

- Transport-specific information are provided as CRIs (*draft-ietf-core-href*)

## › All clients in a group observation use the same Token value

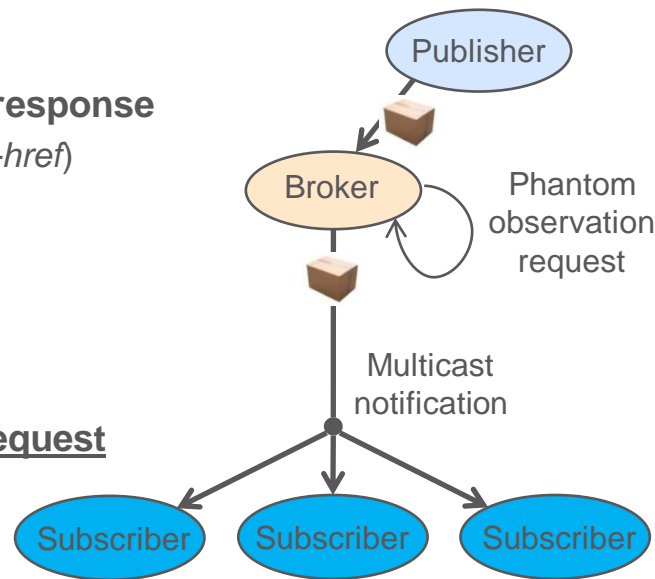
- The Token space belongs to the group (clients)
- The group entrusts the server to manage the token space

## › Multicast notifications are bound to a Phantom Observation Request

- By means of the same Token value for that observation

## › Group OSCORE to protect multicast notifications

- The server aligns all clients of an observation on a same *external\_aad*
- All notifications for a resource are protected using that *external\_aad*



# Recent updates: v -09 → v -10

- › **Overall editorial improvements**
- › **Revised IANA considerations for the new "Informative Response Parameters" registry**
  - › Different ranges and registration policies for the “CBOR Key” column
  - › Defined augmented policy "Standards Action with Expert Review", with possible early registration (RFC7120)

# Recent updates: v -10 → v -11

## More precise text on forbidding redirection of multicast notifications

- › **For convenience, the draft always talks about one server**

- › A client sends a unicast observe registration request to the server
  - › The server rejects the request and replies with an error informative response
  - › The server sends multicast notifications, targeting all the clients that are part of the group observation
- } Repeat for more clients

- › **Multicast notifications must be sent from where error informative responses are sent**

- › **In practice, the old text was overly restrictive**

- › It could not work if the client sent a multicast observe registration request targeting multiple servers [1]

- › **The new text still forbids redirection but also allows multicast observe registration requests**

- › Typically, the multicast notifications come from where observe registration requests are sent to
- › Exception when an observe registration request is sent over multicast

# Recent updates: v -10 → v -11

## New optional parameter 'ending' for the error informative response


```
informative_response_payload = {  
    0 => array, ; 'tp_info' (transport-specific information)  
    ? 1 => bstr, ; 'ph_req' (transport-independent information)  
    ? 2 => bstr, ; 'last_notif' (transport-independent information)  
    ? 3 => uint ; 'next_not_before',  
    ? 4 => uint ; 'ending'  
}
```

- › **Time when the group observation is planned to end and be cancelled by the server**
  - › Number of seconds from 1970-01-01T00:00:00Z UTC until the specified UTC date/time
  - › Encoded as a CBOR unsigned integer

# Recent updates: v -10 → v -11

## Retract published data describing a group observation

### › As in Appendix A:

- › The server can start a group observation and advertise related group observation data as published via other means
  - › A topic in a pub-sub scenario (Appendix A.1) 
  - › Introspection of a Token value at the server (Appendix A.2)
- › Clients obtain the group observation data and get ready to receive multicast notifications

```
Request:
  GET </ps/topics?rt=oic.r.temperature>
  Accept: 65087 (application/coral+cbor)

Response:
  2.05 Content
  Content-Format: 65087 (application/coral+cbor)

  rdf:type [ = <http://example.org/pubsub/topic-list>,
    topic [ = </ps/topics/1234>,
      tp_info_server <coap://[2001:db8::1]>,
      tp_info_client <coap://[ff35:30:2001:db8::123]>,
      tp_info_token "7b"^^xsd:hexBinary,
      ph_req "0160.."^^xsd:hexBinary,
      last_notif "256105.."^^xsd:hexBinary,
      ending "2051251201"^^xsd:unsignedLong,
    ]
  ]
```

Figure 8: Group Observation Discovery in a Pub-Sub Scenario

### › The published data must say when the group observation ends

- › The server commits to keeping the group observation ongoing until then

### › The server might still retract the published data before the group observation ends

# Recent updates: v -10 → v -11

## First description of a scenario with a reverse-proxy and end-to-end security

### › New Appendix H

- › Similar to the setup with a forward-proxy in Appendix G
- › The phantom request is still a deterministic request [2]
- › The server still publishes the group observation data, including the phantom request PH\_REQ (see slide 6)

```
tp_info = [  
    tpi_server: CRI-no-local, ; Addressing information of the server  
    ? tpi_details ; Further information about the request  
]
```

```
tpi_details = (  
    + elements ; Number, format, and encoding of the elements depend  
                ; on the scheme-id of the CRI specified as 'tpi_server'  
)
```

### › In a nutshell:

- › The server knows the address PRX\_ADDR and port number PRX\_PORT that the proxy exposes to clients
- › Within the published group observation data, information pertaining to 'tp\_info':
  - › Includes PRX\_ADDR and PRX\_PORT as 'tpi\_server', i.e., the server-side, transport-specific information
  - › Does NOT include 'tpi\_details' and any client-side, transport-specific information
- › Clients cannot just start listening to multicast notifications; they have to follow-up with the proxy

# Recent updates: v -10 → v -11

## First description of a scenario with a reverse-proxy and end-to-end security

### › Protocol steps

1. Optional initialization procedure, for clients that start from only (PRX\_ADDR, PRX\_PORT)
  - › Interact with the server through the proxy, to obtain missing information
2. The client sends PH\_REQ to the proxy, i.e., to (PRX\_ADDR, PRX\_PORT)
3. Same as in the setup with the forward-proxy in Appendix G
  - a) Cache hit → The proxy returns the latest notification to the client
  - b) No cache hit → The proxy talks to the server, obtains an unprotected informative response, starts listening to multicast notifications, sends the latest notification to the client

### › Unprotected informative response in (3b)

- › It does include client-side, transport-specific information to instruct the proxy



# Next steps

## › Setup with a reverse-proxy

- Add example with message exchange
- In some cases, the ‘tp\_info’ array might be redundant
  - › Consider to make it optional in the payload of the error informative response altogether

## › Align with *draft-ietf-core-transport-indication*

- In ‘tp\_info’, the format of ‘tpi\_details’ depends on the transport to use
- Ongoing work is enabling transport indication also through the URI authority component
- The transport to use cannot be determined solely from the URI scheme in ‘tpi\_server’
  - › In general, the transport to use has to be determined from (URI scheme, URI authority)
  - › This also means simplifying the structure of the new “CoAP Transport Information” IANA registry

## › Comments and reviews are welcome!

```
informative_response_payload = {  
  0 => array, ; 'tp_info' (transport-specific information)  
  ? 1 => bstr, ; 'ph_req' (transport-independent information)  
  ? 2 => bstr, ; 'last_notif' (transport-independent information)  
  ? 3 => uint ; 'next_not_before',  
  ? 4 => uint ; 'ending'  
}  
  
tp_info = [  
  tpi_server: CRI-no-local, ; Addressing information of the server  
  ? tpi_details ; Further information about the request  
]  
  
tpi_details = (  
  + elements ; Number, format, and encoding of the elements depend  
               ; on the scheme-id of the CRI specified as 'tpi_server'  
)
```

Thank you!

Comments/questions?

<https://github.com/core-wg/observe-multicast-notifications>