

EAP-based Authentication Service for CoAP

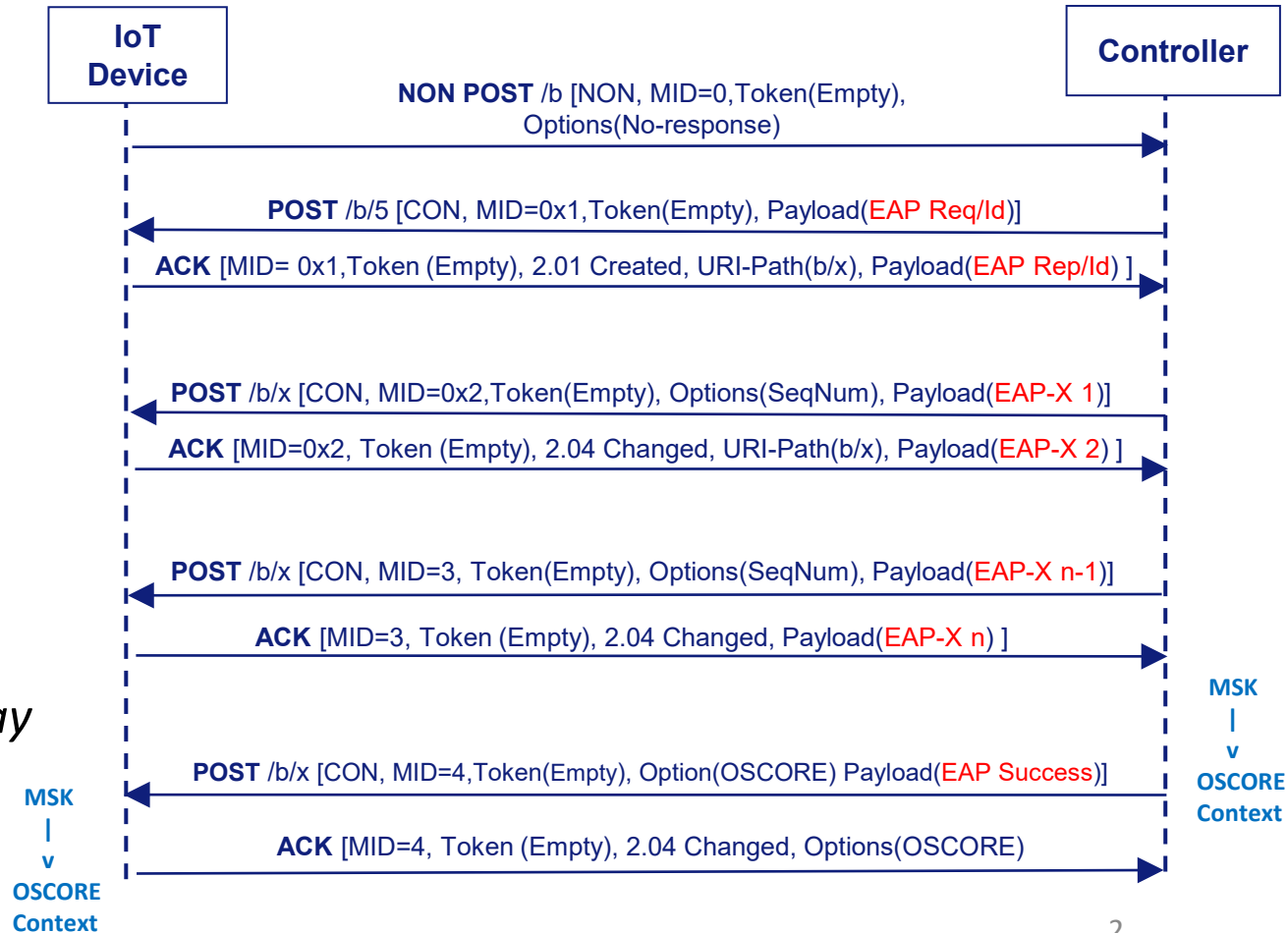
Work in progress for:
draft-ietf-ace-wg-coap-eap-01

Rafael Marín-López, University of Murcia
Dan García-Carrillo, University of Oviedo

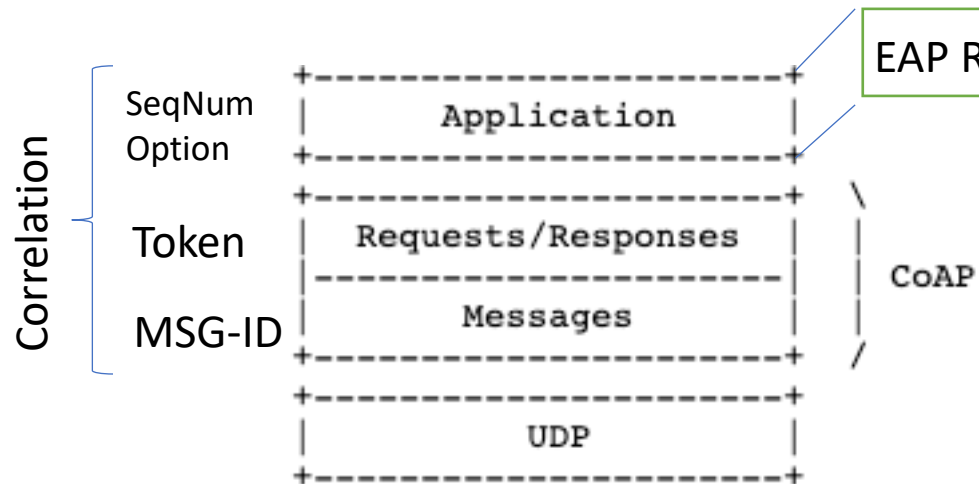
ACE Interim Meeting, April 13th, 2021

CoAP-EAP – EAP Lower layer requirements/assumptions

- ...
- Non-duplication is not mandatory
- **Ordering guarantees -> Relies on EAP lower layer ordering guarantees**
 - If we want an implementation completely independent of the CoAP engine we need a sequence number option for this to work.
 - *NUM-ID & TOKEN can be random and may not be accessible by the application?*



CoAP-EAP - Ordering guarantees



*In -00 versión, this is done with SeqNum Option
A new option defined to keep the ordering guarantee*

Can this be OPTIMIZED or done differently?

Brainstorming

EAP request transported in CoAP Request
EAP response transported in CoAP Response

... we could use a Token for this?

Can we go even further?

Use the same Token for the whole exchange?

Could we use an empty Token to save bytes?

Since CoAP-EAP is a lock-step protocol

Can we use the MSG-ID for the same purpose as Token?

Alternatives to define a new Option (SeqNum) to provide Ordering Guarantee

We can use existing solutions to help provide ordered delivery:

- URI-Query
 - Specify a resource with a subsequent value that indicates the step we are currently in (e.g., /b/x/n) x -> Bootstrapping State; n-> state
- Location-Path and Location-Query
 - Adding the Location-Path and Location-Query Options to indicate what is expected in the server
 - **ISSUE**: Specified with 2.01 response code, not 2.04 Changed . Can we use this?
- Echo Option
 - The server would state a number that has to be replied in another Echo Option in the next POST request from the server

Alternatives to define a new Option (SeqNum) to provide Ordering Guarantee

Echo Option

- Seems a clean solution, using something already defined to solve the problem.
- The approach is similar to the one proposed to by using the SeqNum Option.

Any Thoughts?

THANK YOU