

Towards constrained GRASP

draft-ietf-anima-constrained-grasp-00

CORE-WG, IETF123 Madrid

Toerless Eckert, Futurewei USA (tte@cs.fau.de)

Background

GRASP (RFC8995)

- new protocol to simplify development of in-network automation software
- Not to replace http, coap, .. – but to amend if/where beneficial in new designs
- CBOR encoding. Simple header definition
- Does not specify transport or security
 - „transport and security substrate“ to be added from other specs
 - RFC8994: hop-by-hop TCP, end-to-end TLS, certificates from BRSKI
- Unique features vs. http/coap
 1. Sequence of back & forth messages for negotiation
 - Eg: negotiate some property: try value 1 -> fail, try value 2, fail, try value 4 succeed.
 - Negotiate compound: „config A“ -> ok, „config B“ -> ok, „config C“ -> fail – undoes /never-apply A, B!
 2. Network wide information (service) announcement / discovery
 - RELIABLE because of per-hop reliable propagation. Avoid IP multicast and RMT protocols, high periodic retransmit, ... (30 painful years with information distribution via IP multicast...)

Problem at hand

- ANIMA-WG want to start defining distributed automation agent (ASA)
 - For different network wide automation use-cases
- IMHO:
 - These SHOULD be able to leverage multi-exchange negotiation/synchronization
 - But ANIMA method: use our existing dog food until we show we need something more
 - These certainly WILL use discovery to auto-configure across components
 - Requiring IP multicast or network wide configured use of central servers (DS-SD)not good enough
- Use-cases will likely involve „IoT“ edge problems
 - Have „constrained“ solution for ANIMA security arch (constrained BRSKI)
 - But not for GRASP
- Problem at hand:
 - How to best define GRASP on top of UDP with reliable messages.

Options / Questions to CORE

- Option 1:
 - Use only reliable message mechanisms (not packet format) from RFC7252
 - Extract functionality, given how this is not specified as an obvious reusable sub-layer
 - But certainly do not want to re-define/re-invent reliability/congestion-control parameters when there is with CoAP a well deployment proven solution
- Option 3:
 - Use CoAP and build GRASP as an application – nothing to do for CORE
 - Incurs overhead of URI layer to encode GRASP message types
 - Overhead to build multi-exchange reliable message exchange on top of CoAP reliable messages (duplication of effort)
 - Inefficiencies for message distribution (no buffer sharing)
- Option 2:
 - Define new CoAP message types for GRASP: negotiation, flooding (3..4)
 - Biggest question to CORE... What are the criteria for such an approach to be accepted ?

Last words

- For new designs, if there is a „lightweight / constrained“ protocol option, it's likely preferred even if not urgently needed – because it will easier adopt to future constrained aspects.
 - Solving constrained GRASP would therefore make the approach (IMHO) much more interesting than (TCP) GRASP.
- I like option 2 the best, but it creates most dependencies ANIMA / CORE
- Would make most sense if CORE sees benefits in two new message type functionality even beyond GRASP
 - Aka: maybe specify so that they could be used with any common CoAP URI signalling elements as an option as well
- Thoughts / Questions ?