

Bulk data over GRASP

**draft-carpenter-anima-grasp-
bulk-00**

**Brian Carpenter
Sheng Jiang
Bing Liu**

**IETF 100
November 2017**

Topics

- Motivation
- Proposal
- Discussion, next steps

Motivation

- GRASP was designed for individual technical objectives, potentially over UDP, so its message size is limited.
- However, an autonomic network may need to distribute larger data objects.
- There are many ways to do this. This draft investigates using GRASP itself, to limit the number of required software components.

Basic proposal

- Use GRASP negotiation as a client/server protocol to transfer bulk data one block at a time
- The communication model is slightly asymmetric compared to normal GRASP negotiation between peers
- Goal is simplicity, not performance

Outline of procedure

1. Client discovers server (M_DISCOVERY / M_RESPONSE)
2. Client requests file (M_REQ_NEG with objective value = file name)
3. Server sends data block (M_NEGOTIATE with value = data)
4. Client acknowledges data (M_NEGOTIATE with value = "ACK")
5. Repeat steps 3 and 4 until done.

Details

- Data block size is chosen to fit GRASP message size
- End of transmission is signaled with M_END, O_ACCEPT
- Any error is signaled with M_END, O_DECLINE
 - relies on TCP or UDP checksum
 - missing blocks are detected by timeout
 - after any failure, start again

Does it work?

- Yes. A simple implementation exists, with an objective named "411:mvFile". It has been used routinely to move files between Windows and Linux.
- Client + server = 215 lines of Python code.

Open questions

- Upload instead of download?
 - Just needs a slight inversion of the logic.
- Add position or block # indicator?
 - Needs an extra parameter in the objective value. Not difficult, but since GRASP is a handshake protocol, is it needed?
- Resume failed transfer instead of restart?
 - Could be done, with block #. Simplicity vs efficiency tradeoff.

Discussion + next steps

- Comments? Questions?
- Should the WG consider this topic?

