DIME WG
IETF 86

The Diameter Overload Control Application (DOCA)

Wednesday, March 13th, 2013

Jouni Korhonen, Hannes Tschofenig

What has changed from -00 to -01 ?

- See the original presentation from IETF85.
- The “big” changes are marked with red.

What Diameter Overload Control Application is about?

- A simple/minimal (size wise) application for exchanging load information concerning applications and/or Diameter nodes
- Used between two “DOCA Peers”:
  - One peer can represent a pool of other nodes (e.g. a MME pool).
  - Intermediate Diameter nodes (proxies) can add their own load information (similar to Router-Record behavior) but only when allowed by the DOCA request/answer senders.
- Multiple scopes for information:
  - Diameter node specific or Realm wide specific.
  - Node wide load & overload level or application specific overload level.
  - Or any combination of the above.
- Allows explicit negotiation of exchanged load information.
- “Start”, “stop” or implicit “stop” of overload condition.

Why an Application?

- The support for Diameter overload control capability between Diameter peers is explicit (i.e., a new application-id is advertised).
- The support for Diameter overload control capability between Diameter client and server is explicit.
- The peer selection based on standards; including RFC6408.
- Is able to traverse and also propagate overload control information through realms that deploy relay agents without Diameter overload control support.
- The propagation does not depend on a modified behavior of already specified CCF.
- The use of the application concept allows established mechanisms for filtering and Diameter traffic engineering, since it behaves like any other Diameter application.
- …
- The use of the dedicated application allows to isolate (even physically) the overload signaling into a dedicated transport that is not affected by other Diameter messages and network traffic.

Modes of operation

- ~~State maintaining~~
  - ~~Session state established and bi-directional “understanding” of overload information delivery.~~
  - ~~No need to repeat negotiated parameters.~~
  - ~~Provides means to negotiate the “overload information set of interest” across administrative domains.~~
- Stateless
  - Behaves similarly to S6a. No need to maintain session state with any DOCA peer but less control.
  - May lead to more verbal communication than state maintaining.
  - Every message exchange is separate -> no negotiation.
  - Less control what the other peer sends.
- Commands currently proxiable

Messaging details

- A request-reply message exchange:
  - One command: DOCA-Report-Request/Answer.
  - ~~In state maintaining mode used also to agree on the common set of overload information exchange -> after the first message exchange several attributes can be left out.~~
  - In stateless mode every message exchange is standalone.
- No predefined client or server roles:
  - The node that initiates the conversation is a client.
  - Or the role can be “mandated” by configuration.

Message content details

- Skipping the detailed message CCFs, AVPs and scope material.. Read from the draft.

About message exchange

- ~~IF and WHEN establishing session state the command level AVPs:~~
  - ~~OC-Scope, OC-Algorithm, OC-Action and OC-Applications are used to agree on common set for subsequent message exchanges. Can be renegotiated during the session lifetime.~~
  - ~~OC-Sending-Rate and OC-Toci are used by both ends to express their accepted rate & timer values. Can be renegotiated during the session lifetime.~~
  - ~~OC-Information content cannot be greater what sender advertises in its~~
  - ~~OC-Scope and OC-Applications.~~
  - ~~Negotiation: sender proposes a set of values and responder sends back those values out of the proposed value set it agrees on.~~

- Intermediate nodes can:
  - Add their OC-Information AVP if allowed by the OC-Scope setting.
  - Intended use case is to allow DOCA peers to get better understanding what happens on path and implicitly help e.g. DRAs to select next hops based on overload information.

Additional concerns

- A DOCA peer can represent a pool:
  - How information dissemination is arranged within the pool and its “representative” is implementation specific.
- Overload condition “actions” are node wide:
  - How DOCA commands transports & applications is implementation specific.
- A DOCA peer talks to a number of selected peers:
  - A design decision due the selection of application level solution. There is no unconditional information flooding.

Issues under consideration

- Proxiable vs. direct peer approach ?
  - Sender can already enforce this by dropping the Destination-Realm..
- ~~Remove state maintaining mode ?~~
- No transport specific handling i.e., current load information concerns node and applications only.
- Do we need more scope ? Like sessions or groups ?
- Prioritization within a transport connection ?

Other changes

- Some editorials.
- Added simple example of a case where we have:
  - Multiple servers
  - Aggregating proxy
  - Intermediate agents
  - And a client