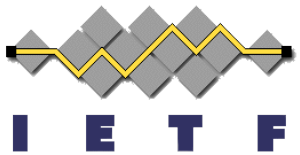
Diameter Overload Control Application (DOCA)

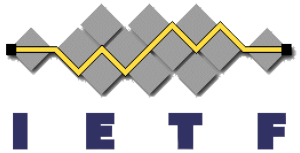
draft-korhonen-dime-ovl-00

Jouni Korhonen

DIME WG – IETF #85



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).
- Explicit capability negotiation between Diameter client and server.
- The peer selection based on standards; including RFC6408.
- Able to traverse through realms that deploy 'vanilla' relay agents without Diameter overload control support.
- The information propagation does not depend on a past, existing or future Diameter commands or their CCF.
- Avoids flooding, especially across administrative domains.
- Applications allow established mechanisms for filtering and Diameter traffic engineering, since it does not differentiate, from a Diameter point of view, from any normal application.
- Uses existing Diameter extensibility mechanisms!

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.. but this is TBD.

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.

CCF for the DOCA-Report-Request/Answer

- The DOCA specific part is the same for request and answer.

```
< Session-Id >
...

{ OC-Scope }
[ OC-Action ]
[ OC-Algorithm ]
[ OC-Sending-Rate ]
[ OC-Tocl ]
[ OC-Applications ]
* [ OC-Information ]

* [ AVP ]
```

Not needed when state is maintained (after the first exchange)

Who can add overload information and what is included in OC-Information AVP. Can e.g. turn off application specific overload monitoring etc.

Overload condition “start” or “stop”. Also used for “interim” updates.

Algorithm to apply during “overload condition”. Currently “drop” and “throttle” defined.

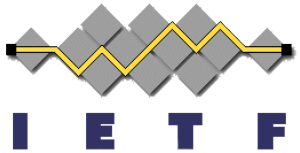
Accepted maximum sending rate the other peer can send messages to the request originator. Applied during “overload condition”.

Accepted interval for periodic overload information “interim” updates.

Applications of interest to monitor. Default is all applications.

Application, node etc specific load information.

About the scope – OC-Scope AVP (up to 32 scopes)



- Host scope (0x00000001) – The OC-Information AVP concerns only a single host within a realm (which internally MAY represent of pool). Note, there can be an array of OC-Information AVPs to cover multiple hosts.
- Realm scope (0x00000002) – The OC-Information AVP concerns a realm. No specific hosts are identified.
- Only origin realm (0x00000004) – The OC-Information AVP can only be included by a Diameter node on the path that has the same Origin-Realm as the DOCA client.
- Application information (0x00010000) – The OC-Information AVP MAY contain application related information (the OC-Applications AVP).
- Node utilization information (0x00020000) – The OC-Information AVP MAY contain node wide load related information (the OC-Utilization AVP).
- Application priorities (0x00040000) – The OC-Information AVP SHOULD priority information (the OC-Priority AVP) so when the overload condition is on, Diameter nodes are able to prioritize between different applications, for example, when dropping or throttling messages.

Overload information contents

```
OC-Information  ::= < AVP Header: TBD3 >
                    { OC-Origin }
                    { OC-Best-Before }
                    [ OC-Level ]
                    [ Vendor-Id ]
                    [ OC-Applications ]
                    [ Product-Name ]
                    [ OC-Utilization
                    [ OC-Priority ]
                  * [ AVP ]
```

Who added this information. Can be host or realm specific.

How long this information is considered valid. Acts also as an implicit “stop” of overload condition when timeouts.

Overload level:
- normal
- raising
- alarming
- panic
- hold
- switch servers

List of applications this information concerns. Must not contain more applications than command level application list.

Utilization of the node.

Priority of this overload information..

- The content of information and what set goes in depends on the command level OC-Scope AVP content.
- Certain combinations of AVPs make no sense such as command level OC-Action=“stop” and OC-Level=“panic” but are possible. The OC-Action still determines whether overload condition is on or off.

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-Tocl 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 ?

Next steps

- WG interested in this overload control approach?