draft-ietf-soc-overload-design-00

Volker Hilt, Eric Noel,

Charles Shen, Ahmed Abdelal

(volker.hilt@alcatel-lucent.com, eric.noel@att.com,

charles@cs.columbia.edu, aabdelal@sonusnet.com)

IETF 78, July 2010

Status

-01 version under preparation

- Will be submitted after IETF ‘78

Few changes since -00 version so far:

- Added references to paper by Ahmed
- Addressed comments from Geoff

Scope of Overload Control

Overload control is used by a SIP server if it is unable to process all SIP requests due to resource constraints.

- SIP server running out of capacity to process SIP messages.

Other mechanisms may be required (e.g., for B2BUAs) that have resource requirements beyond the processing of SIP messages (e.g., DSPs, trunk lines).

- Some error conditions are already covered by existing standards.
- Additional work may be needed to cover this case.
- Out of scope for SOC right now.

Selecting Messages for Throttling

Selecting messages for throttling is important for the performance of an overload control mechanism.

Common guidelines to avoid poor selection decisions.

Messages that can be throttled

- Out-of-dialog messages

Messages that need to be preserved if possible

- Messages with an RPH in a known namespace
- In-dialog messages

Local policies to accommodate specific configurations/devices.

- Example: SIP server only processing subscribe and notify messages.
- Devices can use configuration/device specific rules for throttling traffic if needed.

Feedback Units

Overload control feedback needs to be specified in a unit.

- Example: reduce the number of requests by 20%

Candidates for units:

- Number of sessions/calls
- Number of requests

Feedback-based SIP Overload Control Types

Rate-based Overload Control

Idea: limit the request rate at which an upstream element is allowed to forward requests to the downstream neighbor.

- Server instructs upstream neighbors to send at most X requests/second.

Loss-based Overload Control

Idea: reduce the number of requests an upstream element would normally forward to the downstream neighbor by a percentage X.

- Server instructs each upstream neighbor to reduce load by X%.

Window-based Overload Control

Idea: allow an entity to transmit a certain number of messages before it needs to receive a confirmation for the messages in transit.

- Overload window limits the number of unconfirmed messages.

Signal-based Overload Control

Idea: allow an entity to transmit until no overload indications are received from a downstream server.

- Servers increase/decrease rate until no overload notification is received.

Conclusion

-01 will be submitted soon

- Addresses feedback on the mailing list