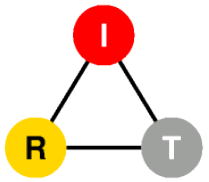


A Session Initiation Protocol (SIP) Load Control Event Package

draft-ietf-soc-load-control-event-package-00.txt

Charles Shen† and Henning Schulzrinne, Columbia University
Arata Koike, NTT

IETF 80, Prague, Czech Republic
March 2011

† Now with AT&T

Changes after the Interim Meeting in December 2010

- Addressed comments from the Interim and the list
  - Filter distribution
  - RPH/message priority treatment during overload
  - How this work meets the requirements of RFC 5390

- Becomes WG item after the consensus call following the Interim

Item #1: Bruno's comments about filter distribution

- Clarified that “The network entry point for load filtering control may be the entity to be protected or another SIP entity it is connected to.” Added an additional example to illustrate the latter case, where an Application Server hosting an 800-number translation service may be the filtering entry point and request filtering of calls to a specific 800 number.

- Clarified that in the earthquake example case, the core proxy may either be *configured* with filter policies or *receive dynamic filters* from edge proxies it connects to.

Item #2 RPH/message priority treatment during overload

Updated description in Section 4.4 and added references to the existing WG documents on related treatment.

" In addition, whatever the actual policy is, SIP servers SHOULD honor the Resource-Priority Header (RPH) [RFC4412] when processing messages. The RPH contents may indicate high priority requests that should be preserved as much as possible, or low priority requests that could be dropped during overload. The request rejection and message prioritization at an overloaded server are also discussed in Section 5.1 of

[I-D.ietf-soc-overload-control] and Section 12 of [I-D.ietf-soc-overload-design]."

Item #3 Janet and other's comments on meeting RFC5390

REQ 1: The overload mechanism shall strive to maintain the overall useful throughput (taking into consideration the quality-of-service needs of the using applications) of a SIP server at reasonable levels, even when the incoming load on the network is far in excess of its capacity. The overall throughput under load is the ultimate measure of the value of an overload control mechanism.

P/A. The goal of the load filtering control is to prevent overload or maintain overall goodput during the time of overload, but it is dependent on the advance predictions of the load. If the predictions are incorrect, in either direction, the mechanism will throttle too much or too little.

REQ 2

REQ 2: When a single network element fails, goes into overload, or suffers from reduced processing capacity, the mechanism should strive to limit the impact of this on other elements in the

N/A if filter values are installed in advance and do not change during the potential overload period. P/A if filter values are dynamically adjusted due to the specific filter computation algorithm. The dynamic filter computation algorithm is outside the scope of this document, while the distribution of the updated filters uses the event package mechanism of this document.

REQ 3

REQ 3: The mechanism should seek to minimize the amount of configuration required in order to work. For example, it is better to avoid needing to configure a server with its SIP message throughput, as these kinds of quantities are hard to determine.

No. This mechanism is entirely dependent on advance configuration, based on advance knowledge. In order to satisfy Req 3, it should be used in conjunction with other mechanisms which are not based on advance configuration.

REQ 4

REQ 4: The mechanism must be capable of dealing with elements that do not support it, so that a network can consist of a mix of elements that do and don't support it. In other words, the mechanism should not work only in environments where all elements support it. It is reasonable to assume that it works better in such environments, of course. Ideally, there should be incremental improvements in overall network throughput as increasing numbers of elements in the network support the mechanism.

No. This mechanism is entirely dependent on the participation of all possible neighbors. In order to satisfy Req 4, it should be used in conjunction with other mechanisms, some of which are described in Section 4.4.

REQ 5

REQ 5: The mechanism should not assume that it will only be deployed in environments with completely trusted elements. It should seek to operate as effectively as possible in environments where other elements are malicious; this includes preventing malicious elements from obtaining more than a fair share of service.

No. This mechanism is entirely dependent on the non-malicious participation of all possible neighbors. In order to satisfy Req 5, it should be used in conjunction with other mechanisms, some of which are described in Section 4.4.

REQ 6

REQ 6: When overload is signaled by means of a specific message, the message must clearly indicate that it is being sent because of overload, as opposed to other, non overload-based failure conditions. This requirement is meant to avoid some of the problems that have arisen from the reuse of the 503 response code for multiple purposes. Of course, overload is also signaled by lack of response to requests. This requirement applies only to explicit overload signals.

N/A. This mechanism signals anticipated overload, not actual overload. However the signals in this mechanism are not used for any other purpose.

REQ 7

REQ 7: The mechanism shall provide a way for an element to throttle the amount of traffic it receives from an upstream element. This throttling shall be graded so that it is not all-or-nothing as with the current 503 mechanism. This recognizes the fact that "overload" is not a binary state and that there are degrees of overload.

Yes. This event package allows rate/loss/windows-based overload control options as discussed in Section 6.4.

REQ 8

REQ 8: The mechanism shall ensure that, when a request was not processed successfully due to overload (or failure) of a downstream element, the request will not be retried on another element that is also overloaded or whose status is unknown. This requirement derives from REQ 1.

N/A to the load control event package itself.

REQ 9

REQ 9: That a request has been rejected from an overloaded element shall not unduly restrict the ability of that request to be submitted to and processed by an element that is not overloaded. This requirement derives from REQ 1.

Yes. For example, the filter format [Section 4.1] allows the inclusion of alternative forwarding destinations for rejected requests.

REQ 10

REQ 10: The mechanism should support servers that receive requests from a large number of different upstream elements, where the set of upstream elements is not enumerable.

No. Because this mechanism requires advance configuration of specific identified neighbors, it does not support environments where the number and identity of the upstream neighbors are not known in advance. In order to satisfy Req 10, it should be used in conjunction with other mechanisms.

REQ 11

REQ 11: The mechanism should support servers that receive requests from a finite set of upstream elements, where the set of upstream elements is enumerable.

Yes. See also answer to REQ 10.

REQ 12

REQ 12: The mechanism should work between servers in different domains.

Yes. The load control event package is not limited by domain boundaries.

REQ 13

REQ 13: The mechanism must not dictate a specific algorithm for prioritizing the processing of work within a proxy during times of overload. It must permit a proxy to prioritize requests based on any local policy, so that certain ones (such as a call for emergency services or a call with a specific value of the Resource-Priority header field [RFC4412]) are given preferential treatment, such as not being dropped, being given additional retransmission, or being processed ahead of others.

P/A. This mechanism does not specifically address the prioritizing of work during times of overload. But it does not preclude any particular local policy.

REQ 14

REQ 14: The mechanism should provide unambiguous directions to clients on when they should retry a request and when they should not. This especially applies to TCP connection establishment and SIP registrations, in order to mitigate against avalanche restart.

N/A to the load control event package itself.

REQ 15

REQ 15: In cases where a network element fails, is so overloaded that it cannot process messages, or cannot communicate due to a network failure or network partition, it will not be able to provide explicit indications of the nature of the failure or its levels of congestion. The mechanism must properly function in these cases.

P/A. Because the filters are provisioned in advance, they are not affected by the overload or failure of other nodes. But, on the other hand, they may not, in those cases, be able to protect the overloaded node (see Req 1).

REQ 16

REQ 16: The mechanism should attempt to minimize the overhead of the overload control messaging.

Yes. The standardized SIP event package mechanism RFC3265 [RFC3265] is used.

REQ 17

REQ 17: The overload mechanism must not provide an avenue for malicious attack, including DoS and DDoS attacks.

P/A. This mechanism does provide a potential avenue for malicious attacks. Therefore the security mechanisms for SIP event packages in general [RFC3265] and of section 10 of this document SHOULD be used.

REQ 18

REQ 18: The overload mechanism should be unambiguous about whether a load indication applies to a specific IP address, host, or URI, so that an upstream element can determine the load of the entity to which a request is to be sent.

Yes. The identity of load indication is covered in the filter format definition in Section 4.1.

REQ 19

REQ 19: The specification for the overload mechanism should give guidance on which message types might be desirable to process over others during times of overload, based on SIP-specific considerations. For example, it may be more beneficial to process a SUBSCRIBE refresh with Expires of zero than a SUBSCRIBE refresh with a non-zero expiration (since the former reduces the overall amount of load on the element), or to process re-INVITEs over new INVITEs.

N/A to the load control event package itself.

REQ 20

REQ 20: In a mixed environment of elements that do and do not implement the overload mechanism, no disproportionate benefit shall accrue to the users or operators of the elements that do not implement the mechanism.

No. This mechanism is entirely dependent on the participation of all possible neighbors. In order to satisfy Req 20, it should be used in conjunction with other mechanisms, some of which are described in Section 4.4.

REQ 21

REQ 21: The overload mechanism should ensure that the system remains stable. When the offered load drops from above the overall capacity of the network to below the overall capacity, the throughput should stabilize and become equal to the offered load.

N/A to the load control event package itself.

REQ 22

REQ 22: It must be possible to disable the reporting of load information towards upstream targets based on the identity of those targets. This allows a domain administrator who considers the load of their elements to be sensitive information, to restrict access to that information. Of course, in such cases, there is no expectation that the overload mechanism itself will help prevent overload from that upstream target.

N/A to the load control event package itself.

REQ 23

REQ 23: It must be possible for the overload mechanism to work in cases where there is a load balancer in front of a farm of proxies.

Yes. The load control event package does not preclude its use in a scenario with server farms.

Backup slides:
Mechanism overview

Problem Statement

SIP overload feedback control is reactive

- typically affects traffic already admitted & treat it equally

Where applicable, it is desirable to leverage known overload contexts (e.g., time and scope)

- Complement feedback control
- Push control closer to the source
- Specify selected parties to be controlled
- Setting up control in advance

Solution

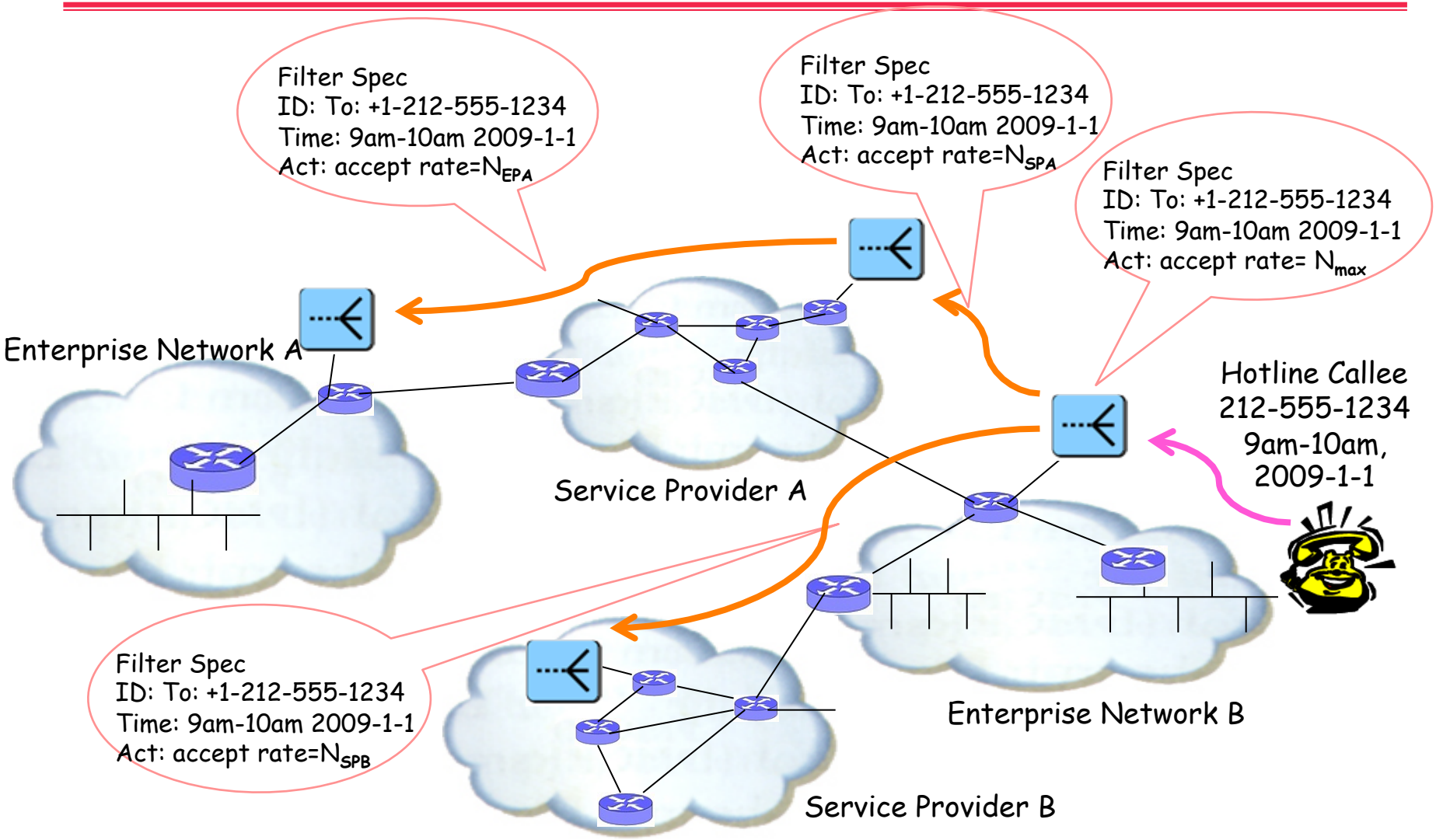
SIP event package for load control

- Subscribe and Notify-based mechanism, instantiation of SIP event framework RFC3265

Definition of load control XML document

- Condition
  - Call Identity: source/destination, SIP or Tel URI(s)
  - Validity: time period to activate control
  - Method: e.g., INVITE
- Actions
  - E.g., accepting a target controlled rate

Example


Filter Spec
ID: To: +1-212-555-1234
Time: 9am-10am 2009-1-1
Act: accept rate=N_{EPA}
Filter Spec
ID: To: +1-212-555-1234
Time: 9am-10am 2009-1-1
Act: accept rate=N_{SPA}
Filter Spec
ID: To: +1-212-555-1234
Time: 9am-10am 2009-1-1
Act: accept rate= N_{max}
Enterprise Network A
Service Provider A
Hotline Callee
212-555-1234
9am-10am,
2009-1-1
Filter Spec
ID: To: +1-212-555-1234
Time: 9am-10am 2009-1-1
Act: accept rate=N_{SPB}
Enterprise Network B
Service Provider B