SOC Working Group Eric Noel
Internet-Draft AT&T Labs
Intended status: Standards Track Philip M Williams
Expires: March 5 2012 BT Innovate & Design
Janet Gunn
CSC

September 2, 2011

Session Initiation Protocol (SIP) Rate Control
draft-noel-soc-overload-rate-control-00.txt

Abstract

The prevalent use of Session Initiation Protocol (SIP) [RFC3261] in Next Generation Networks necessitates that SIP networks provide adequate control mechanisms to optimize transaction throughput and prevent congestion collapse during traffic overloads. Already [draft-ietf-soc-overload-control-03] proposes a loss-based solution to remedy known vulnerabilities of the [RFC3261] SIP 503 (service unavailable) overload control mechanism. This document proposes a rate-based control solution to complement the loss-based control defined in [draft-ietf-soc-overload-control-03].

Table of Contents

1. Introduction

The use of SIP in large scale Next Generation Networks requires that SIP based networks provide adequate control mechanisms for handling traffic growth. In particular, SIP networks must be able to handle traffic overloads gracefully, optimizing transaction throughput without causing congestion collapse.

A promising SIP based overload control solution has been proposed in [draft-ietf-soc-overload-control-03]. That solution includes a default loss-based overload control algorithm that makes it possible for a set of clients to limit offered load towards an overloaded server.

However, such loss control algorithm is sensitive to variations in load so that any increase in load would be directly reflected by the clients in the offered load presented to the overloaded servers. In other words, a loss-based control cannot guarantee clients to produce a constant offered load towards an overloaded server.

This document proposes a rate-based control that guarantees clients produce a constant offered load towards an overloaded server. The penalty for such a benefit is in terms of algorithmic complexity, since the overloaded server must estimate a target offered load and allocate a portion to each conversing client.

The proposed rate-based overload control algorithm mitigates congestion in SIP networks while adhering to the overload signaling scheme in [draft-ietf-soc-overload-control-03] and proposing a rate control in addition to the default loss-based control in [draft-ietf-soc-overload-control-03].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The normative statements in this specification as they apply to SIP clients and SIP servers assume that both the SIP clients and SIP servers support this specification. If, for instance, only a SIP client supports this specification and not the SIP server, then follows that the normative statements in this specification pertinent to the behavior of a SIP server do not apply to the server that does not support this specification.

3. Rate-based algorithm scheme

3.1. Overview

The server is what the overload control algorithm defined here protects and the client is what throttles traffic towards the client.

Following the procedures defined in [draft-ietf-soc-overload-control-03], the server and clients signal one another support for rate-based overload control.

Then periodically, the server relies on internal measurements (e.g. CPU utilization, queueing delay...) to evaluate its overload state and estimate a target SIP request rate (as opposed to target percent loss in the case of loss-based control).

When in overload, the server uses [draft-ietf-soc-overload-control-03] via header oc parameters of SIP responses to inform the clients of its overload state and of the target SIP request rate.

Upon receiving the oc parameters with a target SIP request rate, each client throttles new SIP requests towards the overloaded server.

3.2. Client and server rate-control algorithm selection

Per [draft-ietf-soc-overload-control-03], new clients indicate supported overload control algorithms to servers by inserting oc and oc-algo in Via header of SIP requests destined to servers. While servers notify clients of selected overload control algorithm through the oc-algo parameter in the Via header of SIP responses to clients.

Support of rate-based control MUST be indicated by clients and servers by setting oc-algo to “rate”.

3.3. Server operation

The actual algorithm used by the server to determine its overload state and estimate a target SIP request rate is beyond the scope of this document.

However, the server MUST be able to evaluate periodically its overload state and estimate a target SIP request rate beyond which it would become overloaded. The server must allocate a portion of the target SIP request rate to each of its client.

Upon detection of overload, the server MUST follow the specifications in [draft-ietf-soc-overload-control-03] to notify its clients of its overload state and of the allocated target SIP request rate.

The server MUST use [draft-ietf-soc-overload-control-03] oc parameter to send a target SIP request rate to each of its client.

3.4. Client operation (default algorithm)

To throttle new SIP requests at the rate specified in the oc value sent by the server to its clients, the client MAY use the proposed default algorithm for rate-based control or any other equivalent algorithm.

The default Leaky Bucket algorithm presented here is based on [ITU-T Rec. I.371] Appendix A.2.

Conceptually, the Leaky Bucket algorithm relies on a finite capacity bucket to regulate the flow of new SIP requests. If at a new SIP request arrival the content of the bucket is less than or equal to the limit value TAU, then the SIP request is forwarded to the server; otherwise, the SIP request is rejected.

The capacity of the bucket (the upper bound of the counter) is (T + TAU).

At the arrival time of the k-th new SIP request ta(k), the content of the bucket is provisionally updated to the value

X' = X - RATE * ([ta(k) - LCT])

where X is the content of the bucket after arrival of the last forwarded SIP request, RATE is the rate specified by the server in the last received oc parameter and LCT is the time at which the last SIP request was forwarded.

If X' is less than or equal to the limit value TAU, then the new SIP request is forwarded and the bucket content X is set to X' (or to 0 if X' is negative) plus the increment T, and LCT is set to the current time ta(k). If X' is greater than the limit value tau, then the new SIP request is rejected and the values of X and LCT are unchanged.

At the arrival time of the first new SIP request ta(1), the content of the bucket X is set to zero and LCT is set to ta(1).

Note that specification of a value for TAU is beyond the scope of this document.

4. Example

Adapting [draft-ietf-soc-overload-control-03] example in section 6.2 where SIP client P1 sends requests to a downstream server P2:

```
INVITE sips:user@example.com SIP/2.0

Via: SIP/2.0/TLS p1.example.net;

branch=z9hG4bK2d4790.1;received=192.0.2.111;

oc;oc-algo="loss,rate"

...


SIP/2.0 100 Trying

Via: SIP/2.0/TLS p1.example.net;

branch=z9hG4bK2d4790.1;received=192.0.2.111;

oc=0;oc-algo="rate";oc-validity=500;

oc-seq=1282321615.781

 ...
```

In the messages above, the first line is sent by P1 to P2. This line is a SIP request; because P1 supports overload control, it inserts the "oc" parameter in the topmost Via header that it created. P1 supports two overload control algorithms: loss and rate.

The second line --- a SIP response --- shows the topmost Via header amended by P2 according to this specification and sent to P1. Because P2 also supports overload control, it chooses the “rate” based scheme and sends that back to P1 in the “oc-algo” parameter. It also sets the value of “oc” parameter to 0.

At some later time, P2 starts to experience overload. It sends the following SIP message indicating P1 should send SIP requests at a rate no greater than or equal to 150 SIP requests per seconds.

```
SIP/2.0 180 Ringing

Via: SIP/2.0/TLS p1.example.net;

branch=z9hG4bK2d4790.1;received=192.0.2.111;

oc=150;oc-algo="rate";oc-validity=1000;

oc-seq=1282321615.782

 ...
```

5. Syntax

This specification extends the existing definition of the Via header field parameters of [RFC3261] as follows:

```
oc          = "oc" EQUAL oc-value

oc-value    = "NaN" / oc-num

oc-num      = 1*DIGIT
```

6. Security Considerations

None.

7. IANA Considerations

None.

8. References

8.1. Normative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

8.2. Informative References

[draft-ietf-soc-overload-control-03]
Gurbani, V., Hilt, V., Schulzrinne, H., "Session Initiation Protocol (SIP) Overload Control", draft-ietf-soc-overload-control-03.

[ITU-T Rec. I.371]
"Traffic control and congestion control in B-ISDN", ITU-T Recommendation I.371.

Appendix A. Acknowledgments

Many thanks for the contributions, comments and feedback on this document to:

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Eric Noel
AT&T Labs
200s Laurel Avenue
Middletown, NJ, 07747
USA

Philip M Williams
BT Innovate & Design
UK

Janet Gunn
CSC
USA