Including Overload Control in Existing IMS Compliant Networks by Using Traffic Shapers

Luca Monacelli
Italtel S.p.A. – OSS division
viale dei Caduti nella Guerra di
Liberazione 116, Rome, Italy
luca.monacelli@italtel.it

ABSTRACT

Overload Management in the IMS core of NGNs is a critical task for service providers (SP) in order to achieve a high level of reliability for their communication services. This paper shows a practical approach to protect an existing IMS compliant network from overload by using traffic shapers. An important advantage for SPs is that this solution does not require significant changes in existent Network Elements (NE) functions.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network Management

General Terms

Management, Measurement, Design, Algorithms

Keywords

Overload Control, NGN, IMS, SIP, Network Management

1. INTRODUCTION

Overload happens when an element, such as a SIP server, has insufficient resources to successfully process all of the incoming traffic. Resources include all of the capabilities of the element used to process a request, including CPU processing, memory, or disk resources. It can also include external resources such as a database. Under overload, the throughput of a SIP server can drop significantly and can reach zero or a small fraction of the server's original capacity. In this case, the server enters into a congestion collapse [2].

Recently, Overload Control has been a focus of many research activities, especially for SIP (Session Initiation Protocol) based networks, because of the rapid affirmation of SIP as the main signaling protocol for establishing, modifying and terminating multimedia sessions for many of today's Internet-based communication services including VVoIP, instant messaging and presence. Moreover the SIP protocol has also been adopted by 3GPP and ETSI-TISPAN as the basis protocol for IMS architecture and many service providers are starting to use or are in the process of deploying SIP-based networks. Anyway, despite its widely growing deployment, the SIP protocol provides a poor mechanism for overload control and this is becoming a major problem as the number of users increases.

Without overload control, messages that cannot be processed by the server are simply dropped thus causing the corresponding SIP timers to expire, and further amplifying the overload condition (due to retransmissions). SIP provides a native basic mechanism through its *503 Service Unavailable* response message that may be used to reject a session request and inhibits further retransmissions, but it has proven to be ineffective in practise and often cannot prevent SIP servers overload [2][3].

In [3] Rosenberg analyses the general problem of SIP overload, reviews the current SIP mechanisms for dealing with overload and their problems, and provides a set of requirements for fixing them. In [2], [4] and [5] authors analyse local overload approaches; [4] and [5] propose advanced queuing structures that overperform the standard FIFO queues. [2] shows an adaptive throttling algorithm based on server CPU load, combined with the use of 503 response message without *retry after*. Anyway, local overload controls cannot solve the problem at all; actually rejecting messages has a computational cost (even if it is smaller than fully processing them), therefore in case of severe overload, once the number of incoming requests exceeds the server capacity to reject them, the server collapse. [2],[7] and [8] propose distributed approaches (hop-by-hop or end-to-end) where the receiving (target) entity notifies upstream entities through feedback control loops the maximum amount of messages they are allowed to send in order to avoid congestion. Distributed techniques fix local overload controls problem by moving the throttling task outside the target node. [6] provides a detailed review of the most important distributed feedback control algorithms and compares their performance.

Most of the mentioned solutions require deep modifications to pre-existing, non overload protected, IMS-compliant networks. Unfortunately this is a common situation for many top-operators, that already have an overload-vulnerable SIP based network infrastructure, mostly used to deliver VoIP and advanced telephony services, and they would like to protect it by minimizing costs and change-risks on working NEs. Achieving high service availability, maintaining stable QoS levels [13] and assuring charging reliability [14] are considered by operators fundamental requirements to massively migrate customers (included emergency-numbers) toward full-IP communication solutions, and overload control represents an important step in this direction.

The stabilization approach proposed in this paper cannot be regarded as neither optimal nor particularly "elegant" by a theoretical point of view, but it may be a useful model to approach the overload management problem when various constraints do not permit the adoption of more sophisticated, elegant and

Mobimedia'09, September 7–9, 2009, London, UK.

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7555
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7555

performing techniques (actually it has been tailored to a top-operator needs). At the same time, this model should provide compatibility with novel emerging distributed techniques, as NOCA [8], which seems to be the future of IMS overload management, almost for new network deployments. Lastly, this model should not manage all the possible instability (overload) scenarios, but some documented ones. This modular approach enables satisfying the specific service provider needs. For instance our model does not manage overload caused by resource faults, but it could be extended to manage this scenario as well.

2. NETWORK ARCHITECTURE

In this paper we refer to the ETSI IMS core of a Next Generation Network (NGN), the architecture of which is shown in Figure 1. The interested reader can find a detailed description of the elements of this architecture in [12]. NEs analyzed in our network, group some IMS functions as shown in dashed boxes. These entities are a Session Director (SD) or Session Border Control (SBC), a SIP Server (SS), and a Home Subscriber Server (HSS) or Universal Subscriber Database (UDB). This choice is due to the importance of these elements, which are the core of an IMS-based NGN and are anyway involved in all types of services. They are sufficient to allow users accessing plain wireline VoIP services, even if for accessing other services, such as video-communications, interworking with PSTN/PLMN networks, and other Supplementary Telephone Services (STS), also the other entities shown in Figure 1 are necessary. A more detailed description follows:

2.1 Session Director (SD) or SBC (Session Border Control)

The SD includes the Proxy-Call/Session Control Function (P-CSCF) and the (Access-Border Gateway Function) A-BGF. It is the interface, in the signaling plane, between the IMS terminals and the IMS network. From the SIP point of view the SD acts as an outbound/inbound SIP proxy server; this means that all requests initiated by an IMS terminal or destined for the IMS terminal traverse the SD.

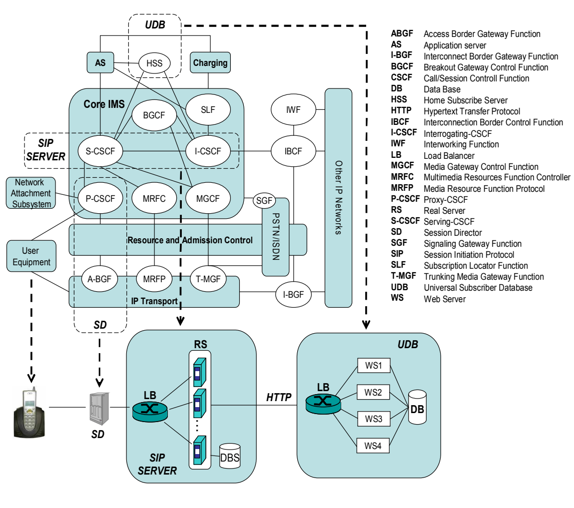

Figure 1. IMS mapping.

2.2 SIP SERVER (SS)

The SS, that is, from an IMS point of view, a Serving (S)-CSCF/ Interrogating (I)-CSCF, is made up of three functional sub-systems:

- Load Balancer - this element has to distribute the SIP requests, sent to the SS, among Real Servers (RSs);
- Real Servers - they are a cluster of N+1 servers that process the SIP messages coming from LB by using, a set of specific Java-based threads. According to the message type, they can issue queries to a centralized database (HSS) for obtaining and updating localization, authentication and authorization information. They are also in charge, in conjunction with the DB Server illustrated below, of creating Call Detail Records (CDR) for accounting purposes.
- DB Server – It is the SS internal database.

2.3 HSS (UDB)

The HSS (Home Subscriber Server) is the central repository of user-related information. An HSS contains all user subscription data required to handle multimedia sessions. These data include, among other items, location information, security information, and user profile information (including the services that the user is subscribed to).

Figure 2 (left) shows the topology scheme of the signaling network analyzed; SDs are shown on the left hand side, SSs in the middle, and UDBs in the right hand side. The additional nodes shown, called VS (Virtual Servers), correspond to the SLF in the ETSI IMS specifications shown in Figure 1. On the right hand side of Figure 2, a geographical distribution of these nodes in Italy is shown.

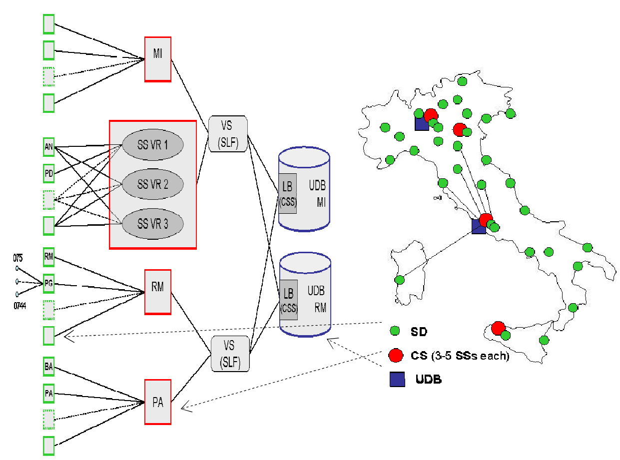

Figure 2. (left) topology scheme of the network analyzed; (right) geographical distribution of NEs analysed.

3. STABILIZATION FRAMEWORK

3.1 Overview

This stabilization (overload control) system, referred to as STBZ which stands for "STaBiliZer", aims to protect SIP servers from overload. In the IMS-like network scenario described above, the target node to be protected is the SS (S-CSCF in the IMS mapping). This NE is a central node in the IMS architecture, that

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7555
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7555

processes a lot of traffic, performs complicated tasks (e.g. respect to P-SCSF for instance) and therefore, it is subject to overload problems. On the other side, because of its centrality, its congestion might produce a large amount of negative side effects through the network. Anyway, apart some specific considerations, this framework may be adopted to protect from overload other non-SIP servers, as for instance, the HSS, which has an http (or Diameter) interface with the S-CSCF (standard *cx* interface). The proposed solution consists of the following three macro-elements (see Figure 3), which are compliant with the "system model for overload control" proposed in [7]: passive probes to get measurements from the network (to be integrated with other measurements provided by Element Manager –EM- layer), a software application (also known as "STBZ core") that collects these measurements, processes them by using specific stabilization rules and controls actuators (i.e. traffic shapers) that execute intelligent traffic shaping, necessary to avoid or solve NE(s) congestion. These shapers must be inserted in the SIP signaling flow, usually as previous-hop or, more generally, upstream the node to be protected, and they should provide deep packet inspection capabilities in order to analyze SIP messages header and eventually reject them according to specific policies. They should also be able to distinguish messages relevant to the same session: actually if the first request message is forwarded, it has no sense to reject subsequent messages expected by the protocol (e.g. 200OK or BYE). This model does not require important modifications of the pre-existing NEs functions (backward compatibility feature) and it is also compliant with requirement expressed in [7][8], related to move throttle functionality outside the target node to solve the well-known local overload control problems.

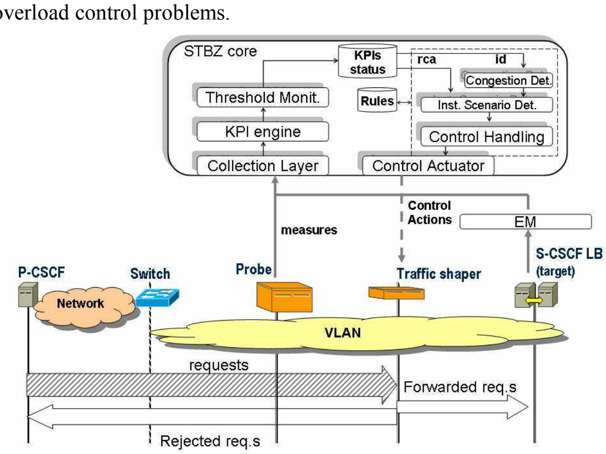

Figure 3. STBZ system overview.

In this paper we are not going to deal with technological issues (detailed protocols and interfaces descriptions, application and hardware architectures, software products etc.); actually these issues are not easily generalized, they are tightly project and environment related and, most important, they are not strictly necessary to understand the proposed stabilization framework.

3.2 Measures Collection

NEs protected by STBZ should be monitored by collecting:

- application traffic counters (e.g. number of SIP Invite per minute);
- response times (i.e. average value and 95th percentile);
- state parameters of protected NEs, which provide useful information about internal resources utilization (this information is not available through probes, but through EMs or NE direct access).

Measurements are usually collected through SNMP southbound interface by a "collection layer" software stratus; the collection period should be a trade off between system responsiveness and computational constraints: a reasonable lower bound is 30s.

3.3 KPI calculation and threshold monitoring

By using the collected measures STBZ calculates key performance indicators (KPI). It is worth to emphasize the difference between measurements and KPIs: a basic measurement as the number of incoming SIP 200OK per unit of time may not be useful by itself if not compared (divided by) to the total amount of SIP INVITE received in the same period (this ratio is a KPI). Each KPI has one or more related thresholds; events are fired when a KPIs values exceed these thresholds.

In this paper we consider three kinds of thresholds:

- L_1 (level 1) threshold
- L_2 (level 2) threshold
- L_R threshold

L_1 and L_2 thresholds have a "high" crossing logic: event is fired when KPI value exceeds the threshold. Sometimes it may be desired to fire this event not at the first time, but after K consecutive crossing repetitions: let's name K as "trigger value".

Each KPI has a discrete *status* whose value (e.g. normal, warning or critical) depends on its threshold crossing events and its former status as in a FSM (finite state machine). Moreover each KPI has a third threshold, named L_R, having a "low" crossing logic. When its value goes below this threshold (lower than L_1 one), KPI status goes back to "Normal"; this creates a hysteresis, which helps the control system to reduce possible oscillatory behaviours (see section.3.6.3).

KPIs can be organised in two logical groups, as explained below, *instability detection* KPIs and *root cause analysis* KPIs.

3.4 Node Overload Detection

A network can be considered overloaded if almost one of his managed NEs (e.g. an S-CSCF) is being overloaded. In this case STBZ should execute his stabilization tasks. Therefore overload handling requires the capability for STBZ to identify overloaded servers. Unfortunately the quantification of congestion for a server is still an open research topic. [9] provides a review of classical server (system) congestion detection approaches, based, for instance, on service time or throughput, and also proposes an interesting service-aware congestion metric.

STBZ should be able to distinguish the three classical server operating zones, also referred to as congestion states:

- "Normal Load" status (congestion state = 'Normal'): the throughput increases linearly with the system load and the service time remains practically constant;
- "Overload" status (congestion state = 'Warning'): resources saturation produces an intensive using of system buffering

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7555
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7555

capacity, which result in an increase of the service time; throughput is more or less constant (or slightly decreasing) in relation to the offered load. Resource utilization is maximized, at the coast of significant increase of service time.
- ”Collapse” status (congestion state = ‘Critical’): throughput decreases approaching to zero and service time can increase up to extremely high values.

The main goal of STBZ is to prevent a server (or more generally, a NE) to reach collapse status during overload situations, or, at least, minimize its permanence in it. Achieving the first objective is more desirable, even if less efficient from a resource utilization point of view: actually some recent researches [2] show that once a SIP server is congested, it requires a quite long time to be decongested, despite throttling input traffic to the “normal” load level.

Table 1 provides an example of SIP server (in particular an RS) congestion due to the increase of incoming RPS (Registrations per second); offered CPS (Calls per second) are kept constant to 2 during the whole test. It should be noted that during a serious congestion, the number of SIP Register messages per Registration increases due to retransmissions.

Table 1. Performance degradation of a RS during overload

Offered RPS	**0**	**0**	**10**	**20**	**38**	**76**	**100**	**128**	**256**
Processed RPS	**0**	**0**	**10**	**20**	**38**	**48**	**43.8**	**33.8**	**14.6**
Offered CPS	**0**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Processed CPS	**0**	**2**	**2**	**2**	**2**	**1.3**	**0.9**	**0.7**	**0.01**
Offered SIP Register/sec	**0**	**0**	**20**	**40**	**76**	**165**	**210**	**202**	**676**
Processed SIP Register/sec	**0**	**0**	**10**	**20**	**38**	**48**	**43.8**	**33.8**	**14.6**

KPI categories that can be used for overload detection purposes are the following:

- server internal parameters (e.g. percentage of CPU, RAM);
- offered load (e.g. number of SIP Register per minute);
- throughput;
- success rate;
- service time.

Given a target NE to be protected and its M instability detection KPIs, each KPI has an associated discrete status S_K={N,W,C} depending on its actual value respect to the related thresholds. A *node congestion rule* (see Table 2) may be defined as $\psi: S_K^M \rightarrow S_N$, where S_N={Normal,Warning,Critical}represents the set of possible node congestion states listed above. Through these rules, STBZ identifies overloaded NEs.

Table 2. An example of Node Congestion rule

State KPI_1	State KPI_2	State KPI_N	Node Congestion State
N	N	N	NORMAL
N	N	W	NORMAL
N	W	W	WARNING
W	C	C	CRITICAL
C	C	C	CRITICAL

It is worth to say that in order to identify the congestion status of an NE one should not use all available instability detection KPIs, which could be a lot; the goal is, on the contrary, to select a minimum subset of instability detection KPIs that allow identifying server congestion with a desirable level of reliability.

Moreover it is important to highlight the difference between an overloaded node and another (non overloaded) node that is experiencing malfunctions due to the first one congestion (because of their inter-dependences). Node congestion rule must distinguish these situations, because STBZ have to directly face the root cause problem (the congested one) and not all the symptoms. Solving overload will also fix all other related malfunctions (symptoms) in the network.

3.5 Instability Scenarios

Once an overloaded node has been detected, it is necessary to analyze the causes of congestion to successfully recover it. Section 4 lists some examples of *Instability Scenarios*, i.e. category of overload causes that should be identified and managed by STBZ. To perform this task, another set of KPIs, called *root cause analysis (RCA)* KPIs have to be defined and monitored. Thanks to this KPI set, STBZ should (1) identify the instability scenario among those listed in section 4 and at the same time (2) identify specific traffic “flows” (e.g. SIP REGISTER from P-CSCFi to S-CSCFj, or SIP INVITE toward +3906123123) that cause the NE congestion because of their anomalous volume. Usually STBZ, through traffic shapers, will try to solve NE congestion by throttling these flows, but this is not the only solution as it will be explained below. RCA KPIs are typically traffic counters provided to STBZ by passive probes and they are not necessary related to the target node (differently to instability detection KPIs).

To recap, the logical entities involved are:

- A target NE to be protected (the SS / S-CSCF in our example).
- Some traffic flows (labelled with A in Figure 4) that cause target NE overload because of their anomalous volume.
- Some traffic flows to be throttled (T); usually, but not necessary, they match the previous ones, i.e. A≡T
- Other thresholds-compliant flows (N) that usually should be impacted as less as possible by stabilization activities.
- A traffic shaper along the path from T traffic sources to target NE; usually the traffic shaper is the previous hop in the SIP signaling path, but as a general guideline, it should be as close as possible to the source of T traffic (hop-by-hop vs. end-to-end overload control).

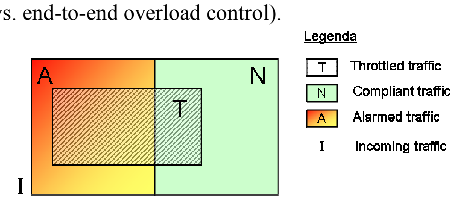

Figure 4. Relation among STBZ traffic flows categories.

We show some examples of mismatching between A and T flows. First, suppose that S-CSCF is overloaded due to some SIP INVITE flows from P-CSCFs exceeding their usual volume. In such a situation, a solution may be to reduce REGISTER flows from P-CSCFs by a certain percentage: the reason is that throttling a SIP REGISTER message (about 70% of the traffic in

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7555
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7555

our network) does not surely imply user disservice and revenue loss for operator (unlike INVITE shaping). A second example is S-CSCF congestion caused by a high traffic volume coming from PSTN/PLMN (from outside IMS network); SP may prefer to protect this kind of traffic (usually more remunerative than VoIP traffic) and throttle SIP REGISTER/INVITE coming from P-CSCFs (within IMS network). The last example is relevant to emergency numbers: despite they could contribute to server overload, especially during disasters, they should not be throttled, but on the contrary, prioritized.

It is important to highlight that STBZ will not perform any action (traffic throttling) if the previous instability detection phase does not detect any overloaded node. This avoids unnecessary traffic shaping (thus related operator revenue losses and user disservices) as shown below. Consider n P-CSCFs forwarding SIP messages to a single S-CSCF (target NE). Let L be the maximum capacity of this node in terms of input traffic (this is clearly an instability detection KPI with L as level 2 threshold value). We may consider the traffic forwarded from each P-CSCF toward target S-CSCF counters as RCA KPIs and set to L/n the related threshold (L2). If one or some RCA KPIs values exceed L/n but the total amount of input traffic is less than L (thus the S-CSCF is not congested), STBZ should not execute any action (e.g. throttling alarmed flows) because the server can still process those requests.

3.6 Control Handling

Control Handling is a functional area of STBZ that handles the lifecycle of each *Control*, from start to stop. A *Control* can be considered as a standard template to face a particular Instability Scenario.

- There is a one-to-many relationship between Instability Scenario and Control. In other words, the same Scenario can be managed by using different Controls that, for instance, may be scheduled from the less important to the most important. After activating the first control, the system holds it for *K* cycles and, if it was ineffective, i.e. it does not solve target node congestion, the next scheduled one is activated, and so on.
- A Control is composed by one or more *Control-Actions* (CA).

Table 3 reports the relations between Control and Control-Actions lifecycles.

Table 3. Control and Control Actions

Control	Control-Action
Control Activation	One or more Control Actions activation
Control Update	New Control Action(s) activation Some Control Actions deactivation Control Actions re-modulation
Control Deactivation	All Control Actions deactivation

3.6.1 Control Activation

Control Actions belonging to the same Control often have similar features in term of shaping technique used and action "granularity". Every Control Action will be activated if relevant RCA KPIs status matches a specific rule.

3.6.2 Control Update

A previously activated Control is updated (after processing new collected measures bulk) in one of the following cases:

- all the Control Actions previously active are confirmed (except, in case, for intensity re-modulation);
- one or more new Control Actions are activated;
- one or more active Control Actions are deactivated (but not all of them).

Actually, even inside the same scenario (thus the same Control), A-flows (and T-flows) set shown in Figure 4 may be time-variant.

3.6.3 Control Deactivation

An active Control deactivation depends on two factors: obviously node congestion status, but also the persistence/end of the congestion causes, as detailed in Table 4.

Table 4. Control deactivation details

NE cong. status	Causes	Description
N	ceased	The simplest case: target node is not more overloaded, all congestion causes ended. The Control should be deactivated.
N	persist	a) Control Deactivation: the causes that leaded the node to congestion hold over, therefore node may tend to congest again and STBZ will activate the control again. (and so on) The system shows an oscillatory behaviour (oscillation frequency is low thanks to the hysteresis in evaluating instability detection KPIs status). After m oscillations cycles it is advisable to switch to (b) option. b) Control Update: Control is kept active, but Control Actions should be re-modulated in order to regulate the node congestion status at the boundary between N and W region, in order to maximize the server processing capacity and avoid oscillating behaviours. Note that without this re-modulation, Control would regulate node congestion status deep inside the N region (inefficient behaviour).
W, C	ceased	This case is less common. Anyway the congestion persistence, despite previous overload causes ceased, may depend on instability scenario change. In this case, the control should be deactivated and the new one (relative to the new detected instability scenario) should be activated. If a new scenario is not detected, the control may be kept active.

Moreover an active Control should be deactivated when one of the following events occurs:

- control timeout (max number of iteration) expires. A new Control (usually more intense), if scheduled, should be activated;
- missing control update information (e.g. measurements, thus KPIs needed to update that Control not available for STBZ southbound interface problems).

To recap, after detecting an overloaded node through instability detection KPIs analysis, STBZ correlates RCA KPIs to detect both instability scenario and anomalous flows (that caused congestion). Then a first Control is activated, consisting of one or more Control Actions, singularly activated by specific rules matching (based on RCA KPIs status). If after K iterations this Control has not fixed the node congestion, STBZ (control handling function) switches to the second scheduled Control,

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7555
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7555

usually more intense than the previous one, and so on, until node congestion has been solved.

3.7 Control Actions

In this paragraph Control Actions, that represents the atomic stabilization element in our framework, will be analysed in more details.

3.7.1 Control Action basic parameters

We show the minimum set of information related to a Control Action. The self-explanatory XML-like format showed in Figure 5 aims to suggest a suitable configuration/control southbound interface between STBZ and traffic shapers (actuators).

```
<control>
   <name>C5</name>
   <shaper>SHAPER_1</shaper>

   <controlAction>
      <id>001</id>
      <action>shaping</action>
      <method>rate_based</method>
      <flow>
            <source>P-CSCF Perugia</source>
            <destination>S-CSCF Rome</destination>
            <protocol>SIP</protocol>
            <traffic>Register</traffic>
      </flow>
      <intensity>120</intensity>
      ...
       <time>10.06.2008|10.32.11.95</time>
   </controlAction>

   <controlAction>
   ...
   </controlAction>
</control>
```

Figure 5. XML-like traffic shapers configuration I/F.

3.7.2 Throttle schemas

This paragraph refers to methods used to limit the amount of traffic forwarded to the protected NE. One or more of these methods should be implemented by shapers in our solution.

- **Rate-based**

The key idea is to limit the maximum request rate from a sender to a receiver.

  - *Request Gapping* (Crowford Gapping)

  After transmitting a request to a downstream element, a server waits for $1/\lambda max$ seconds before transmitting the next request to the same neighbour. Requests that arrive during the waiting period are not forwarded, but may be rejected (as in our model), redirected or buffered. This method asymptotically (i.e. for high input request rates) reaches desired output (forwarded) request-rate λmax.

  - *Leaky Bucket*

  This well-known technique limits the output rate without Crowford Gapping drawback.

- **Percentage-throttling**

The upstream server (shaper) reject X percent of the traffic that is destined to that server. A simple formula for calculating $X_{\%}$ (throttle percentage) is:

$X_{\%}$=(actualValue-targetValue)/ actualValue *100

The main drawback of percentage throttling is that the throttle percentage needs to be adjusted to the current number of request received by the server/shaper: This is important if high traffic fluctuations may exist between two consecutive sampling instants (30s). So, leaky-bucket rate-based technique seems to be the best throttling solution. Anyway also percentage throttling should be implemented by shapers.

3.7.3 Shaper response messages

When the shaper blocks a SIP request, in order to avoid retransmissions from the sending entity after standard SIP timeout periods, it should generate a proper reject response. SIP [1] provides a message *503 Service Unavailable*, which as said in the introduction section, has proven to be ineffective in many situations. In [10] authors suggest the introduction and use of a new SIP message, *507 Server Overload*, which differs from the 503 Service Unavailable message by the following detail: "*A client (proxy or UAC) receiving a 507 (Server Overload) response SHOULD NOT attempt to forward the request to an alternate server*". At the best of our knowledge, there are not detailed quantitative studies on the benefit of this approach.

3.7.4 SIP Register "Retry-After" randomization

Suppose that at time T, STBZ commands the shaper to throttle a specific SIP Register flow, and sets to *t* the retry-after field in the reject response header. At time T + *t* senders retransmissions may produce a new Register peak toward the protected node. In order to fix this problem, the shaper should set the retry-after field according to an optimal probability density function. This issue has to be further investigated.

3.7.5 Control Actions co-operation & conflicts

Until now we have implicitly considered only one congested NE in the whole managed network. But if two or more NEs are overloaded and therefore different Controls are active at the same time, some conflicts among Control Actions may happen, depending on specific rules implementation. To face this problem, some guidelines follow:

- if two or more Control Actions act on non overlapping flows, they should be parallel and independent each other;
- if two or more Control Actions act on overlapping and same granularity flows, the most conservative one (per flow) wins;
- if two (or more) Control Actions act on overlapping flows with different granularity, one sub-flow of another (e.g. Register from a P-CSCF and Register from a single UA belonging to that P-CSCF), shapers should attempt first to execute the lightest ones only. If this not fixes the congestion(s), shapers will execute both.

4. INSTABILITY SCENARIO EXAMPLES

This section provides in Table 5 a list of common S-CSCF instability scenarios [3].

Table 5. Instability Scenarios

Network scenario	**Real-word scenario**
SIP Register peak from single P-CSCF	P-CSCF Restart, blackout
SIP Register peak from many P-CSCF	HSS restart (Figure 6)
SIP Invite peak from single P-CSCF	Disaster, mass event
SIP Invite peak from many P-CSCF	Festivity
SIP Invite peak toward single/few users	Media stimulated event
SIP Register peak from few users	Malfunctioning terminals
SIP Invite peak toward single P-CSCF	Disaster, mass event
SIP invite peak toward few "gold" users	Disaster (emergency numb.)

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7555
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7555

5. RELATION BETWEEN STBZ and OSSs

Outlining the STBZ role in the service provider OSSs (Operation and Support Systems) scenario is a critical task, even if it might appear of little importance. From a vendor/supplier point of view for outline potential boundaries of software components and highlight the required functions, input and output to be supported by the system; for SP/operators to clearly understand system functions, process flows involved, other OSSs relations (thus related impacts) and, moreover, to define management responsibilities for STBZ lifecycle. eTOM (enhanced Telecom Operations Map) Business Process Framework may be considered the reference framework used to categorizing all the business activities that a service provider will use; within this framework [11], STBZ should be involved in (1) the "Assurance" vertical process group which is "*responsible for the execution of proactive and reactive maintenance activities to ensure that services provided to customers are continuously available and to SLA/QoS performance level*"; (2) the Resource Management & Operations horizontal group, which is "*responsible for managing all the resources.(e.g. networks, IT systems, servers, routers, etc.) utilized to deliver and support services required by or proposed to customers*". Above in this paper we have dealt with STBZ's southbound interfaces: to NEs, EMs and probes to collect measures and to shapers to control them. Anyway STBZ may require also many northbound interfaces to a GUI and to other OSSs, in particular Fault and Trouble Ticketing Management, Identity Management, Logging, Inventory, Performance Management Systems.

6. CONCLUSIONS and FUTURE WORKS

In this paper we have shown a practical model to manage overload in the IMS core of wireline NGN networks. Despite a lot of interesting techniques have been recently proposed, their adoption would often require deep (and expansive) modifications in existing SPs' networks, especially in working NEs functions. This is the reason why they better suit to new network deployments than to updating (to overload protection) existing ones. Our model is composed by three elements, compliant with the "system model for overload control" proposed in [7]: passive probes, a rule-based control function (STBZ core) and traffic shapers; it identifies some logical steps in the overload control process, and in particular, it distinguishes congestion detection and root cause analysis phases. Inside each phase a lot of work has still to be done and this work provides some useful tips and examples. As next step, this model should be extended in order to manage further instability scenarios, for instance network capacity reduction caused by one or more resource faults. Furthermore it should be investigated the use of alternative control techniques, not only traffic shaping, but also, for instance, dynamic load balancing. Advanced load balancing may re-route traffic directed to an overloaded NE to another equivalent NE having available processing capacity. This would clearly optimize (locally or globally, depending on implementations) the use of server processing capacity before shaping requests.

7. ACKNOWLEDGMENTS

I am grateful to my colleagues F.Mistretta, S.Iacocagni, S.Vanschoubroeck and P.Cicero for their precious contribution.

8. REFERENCES

[1] J.Rosenberg. "SIP: Session Initiation Protocol", IETF RFC3261, http://www.ietf.org/rfc/rfc3261.txt, June 2002.

[2] V.Hilt, I.Widjaja. "Controlling Overload in Networks of SIP Servers", IEEE International Conference of Network Protocols (ICNP08), October 2008.

[3] J.Rosenberg. "Requirements for Management of Overload in the Session Initiation Protocol", IETF RFC 5390, December 2008.

[4] M. Ohta. "Overload Protection in a SIP signaling network", International Conference on Internet Surveillance and Protection (ICISP '06), August 2006.

[5] S.Montagna, M.Pignolo. "Performance evaluation of Load Control Techniques in SIP Signaling Servers", Third International Conference on Systems (ICONS'08), April 2008.

[6] C.Shen, H.Schulzrinne, E.Nahum. "Session Initiation Protocol (SIP) Server Overload Control: Design and Evaluation", Systems and Applications of IP Telecommunications (IPTComm'08), July 2008.

[7] V.Hilt, I. Widjaja, D.Malas, H.Schulzrinne. "Session Initiation Protocol (SIP) Overload Control", IETF Internet draft, February 2008, work in progress.

[8] ETSI TR 182 015, "TISAPAN; Next Generation Networks; Architecture for Control of Processing Overload", v1.1.1, 2006.

[9] E.Monteiro, G.Quadros, F.Boavida. "A Scheme for Quantification of Congestion in Communication Services and Systems", Third International Workshop on Services in Distributed and Networked Environments, June 1996.

[10] V.Hilt, I.Widjaja. "Essential Correction to the Session Initiation Protocol (SIP) 503 (Service Unavailable) Response", IETF Internet Draft, July 2007.

[11] TeleManagement Forum, "eTOM: The Business Process Framework", GB921V3.6, November 2003.

[12] ETSI ES 282 007, "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IP Multimedia Subsystem (IMS); Functional architecture", V2.0.0, 2008-05.

[13] M.Gerla, C.Cassetti, S.S. Lee, G.Reali. "Resource Allocation and Admission Control Styles in QoS DiffServ", Lecture notes in computer science, volume 1989/2001, Quality of Service in Multiservice IP Networks, Springer, pp.113-128.

[14] D.Di Sorte, M.Femminella, G.Reali. "A QoS Index for IP Services to Effectively Support Usage-based Charging", IEEE Communications Letters, November 2004, pp.686-688.

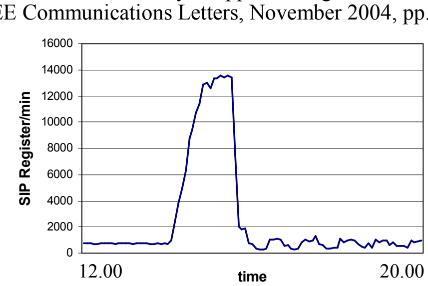

Figure 6. Peak of SIP Register/min (mostly retransmissions) arriving to a SS (S-CSCF) during an HSS fault (October '07).

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7555
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7555