

Overload control for session setups

Field of invention

The present invention relates to overload control in a Voice over Internet Protocol (VoIP) and multimedia
5 session network and especially to a method and system for overload control of session establishment in transit networks connecting different service providers.

Background of invention

In next generation VoIP and IP multimedia networks, individual control plane components, (e.g. Session
10 Border Controllers (SBC), Media Gateway Controllers (MGC) and Call Servers (CS)) usually have some inbuilt overload self-protection mechanisms intended to prevent their collapse of processing under overload by discarding messages that request a new session set-up. Most mechanisms have little or no discrimination on the type of calls that are being affected, with emergency calls often receiving the only discrimination in the load shedding process. Whilst protecting itself, this does nothing to maximise the successful
15 establishment of new sessions.

In networks implementing VoIP technology the call control and the media transport are separated into independent media and session control planes. This enables the design of a flat control architecture that allows any node to have a signalling relationship with any other node. With the addition of a centralised routing function this allows for highly scalable and distributed networks but it also leads to scaling issues in
20 large networks when it comes to signalling the appropriate load information of all destination nodes to all the nodes generating the new session set-ups.

For such flat control plane architectures, some schemes have been proposed where an overloaded node responds to requests for new session set-up by signalling that all sending nodes to discard a proportion of
25 its new requests. This scheme has a number of problems.

It either has to hold a configuration of all the sending nodes to which to send the overload control information or respond just to the nodes making the load requests. This means that any new sending nodes will not initially have the restriction information and may start sending at full rate. Also as traffic demand varies, particularly with a very fast rise time, it becomes impractical to match and feedback the proportional
30 discard rate that will produce the optimal traffic demand offered to the overloaded node. If for example a node can receive no more than 100 new session set-ups per second and it anticipates a traffic demand of 200 new session set-ups per second, requesting a 50% discard of traffic will not help if the real traffic rate

risers to 1000 new session set-ups per second before the overloaded node can re-evaluate and signal the change in discard rate. Therefore, for nodes to protect themselves within such schemes, they tend to use overly severe restriction predictions which will reduce customer experience through unsuccessful sessions and reduce network revenue below that which could have been achieved with the available resources.

- 5 Another technique that is used for overload control is for the node under stress to signal to its neighbour nodes the absolute rate of session set-ups it requires instead a proportional discard as described above. For example, send me no more than 100 new session set-ups per second rather than discard 30% of your new call requests. The advantage of this scheme is that the demand sent can be matched to the available capacity of the overloaded node and be can safely protected regardless of how rapidly the traffic rate may
- 10 rise. If the sending nodes are well behaved, the demand sent will not exceed the demand requested. The problem with current implementations of this scheme is that the node in question needs to know how to apportion its optimal total target rate for offered traffic amongst all the nodes that can send it traffic. This means that not only does it has to have preconfigured information of the other nodes so it can send each the restriction information of that specific node, but it must also know something of the relative size of the
- 15 nodes so that some apportionment of fairness can be built in. If 10 source nodes can send traffic to a destination node that decides it needs to limit demand to 100 new session set-ups per second, dividing the traffic limit to 10 new session set-ups per second for each connecting node may suffer from the following problems:

- One or more of the nodes may only be able to generate less than 10 new session set-ups per second in
- 20 total so the total demand at the destination can only be sub-optimal, i.e. the destination node will be underutilised.

- One or more of the nodes may be able to send much more than 10 new session set-ups per second so the disproportional restriction will be 'unfair' on the user on the node in terms of customer experience.

- In mass media events, such as television gaming applications, where 'winners' are selected from the calls
- 25 arriving at the destination; callers from the disproportionately restricted nodes are disadvantaged.

US patent application 20100220587 discloses a method for implementing a two throttle overload control architecture in a node, one throttle on the input and one on the output side of the node. The in-throttle filters the input traffic according to the perceived limits of the target nodes' processing capacities. This allows requests that would be dropped by the target nodes due to overload to be filtered prior to being

30 processed by the node. In addition, filter weights for the in-throttle is set such that the possibility of admitting to-be-dropped requests is minimized. The purpose of the method is thus to protect the node

itself from having to process traffic that will anyway later be dropped. The disadvantage of this method is that each target node has to report its capacity to every possible source node and a source node cannot filter requests to targets it is not in direct contact with.

Patent application EP2445164 describes a method for load balancing between call session control functions (CSCF) in order to overcome the problem of a registration storm when a large number of user terminals try to reconnect for a service. Each CSCF reports its load, or rather the number of users currently registered to it, to a centralised DNS server which then uses the information for load balancing SIP registration requests between available CSCFs. This system is, however, only relevant for the process of SIP registering, that is user equipment must register with a CSCF upon connecting to a network. However, once the user equipment is registered to one of the CSCFs it remains associated with this node. Thus, this method does not provide a solution to the problem of avoiding call setups to overloaded session control edge nodes on a call by call basis.

Summary of invention

According to a first aspect of the present invention there is provided a method for optimising session set-up between session control nodes, which nodes interface to different networks and/or end user terminals in a globally scalable VoIP network, said method comprising at a centralised server system:

- receiving a load status message indicating a load status from each session control node;
- receiving session set-up requests from said session control nodes where each session set-up request identifies a sender node and a destination node and determining from said requests the number of session set-up requests directed to each destination session control node;
- for each received session set-up request selecting one or more session set-up policies to include in a response to the requesting session control node based on said indicated load status and the number of session set-up requests directed to each session control node ; and
- sending the one or more selected session setup policies to the requesting session control node.

By enabling a central server to determine the number of session set-up requests directed to each node as well as load restriction requirements for the session control node, a more efficient load balancing between the different session control nodes can be achieved, thus avoiding or reducing the risk both of over-loading and under-loading the nodes. A further advantage is that each destination node only needs to communicate its load status updates to one location, therefore in a network of n destination nodes, each node would only

need to be aware of one centralised overload control server and would generate only one message for each status update, thus reducing the management complexity and the communication effort required when experiencing overload at the destination node, as well as reducing the signalling traffic on the network.

- 5 Preferably, the load status message comprises a request for a defined restriction on new session set-ups per second or a request for a proportional discard. However, other target load information could be conveyed.

According to a second aspect of the invention there is provided a server system for controlling session setup between session control nodes, which nodes interface to different networks and/or end user terminals in a globally scalable VoIP network, said server system comprising:

- 10 a session setup policy module arranged in operation to select one or more session setup policies in response to session setup requests received from said session control nodes, where each session setup request identifies a sender node and a destination node;
- an overload control module arranged in operation to receive from each session control node a load status
- 15 message indicating a load status;
- the server system being further arranged in operation to determine from said received session setup requests the number of session set-up requests directed to each session control node, and to select one or more session setup policies to include in a response to a requesting session control node based on said received load status and the number of session set-up requests directed to each session control node.

20

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings in which:

Figure 1 shows an exemplary embodiment of a system implementing centralised overload control.

- 25 Figure 2 shows an exemplary embodiment of a centralised server.

Figure 3 shows an exemplary embodiment of an alternative configuration where a centralised overload control server acts as a proxy to a policy server.

Figure 4 shows an example implementation of the invention in an IP Transit network using SIP signalling.

30

Description of preferred embodiments

A system in which a first exemplary embodiment of the present invention is implemented will now be described in relation to figure 1. The voice and multimedia network 20 of figure 1 is a large transit network where many different network operators interconnect. The network 20 comprises a number of session control edge nodes 21A-D, which terminate the interconnection transmission to different operators, and a centralised session set-up policy server 12. The session set-up policy server 12 comprises a processor 32, a session set-up policy store 36, a selection module 40 and a modification module 41, and one or more memories 33. A session set-up policy comprises an address of a destination session control node and preferably associated trunk group identifiers. The modification module 41 is arranged to filter or discard policies selected by the selection module 40 based on a current overload status and demand for a certain destination session control node.

The edge nodes 21 can be for example Session Border Controllers, Call Servers, Soft switches, Application Servers, Application Gateways, Media Gateways Controllers and Call State Control Functions defined in the Internet Protocol Multimedia Subsystem (IMS) architecture.

In addition, there is a logical centralised overload control server 18 incorporated into the network. The centralised overload control server 18 comprises a processor 37, one or more memories 39, a session control node overload status module 38, a store 42 and for certain embodiments also a modification module 41. This centralised server 18 may be logical, being realised as a number of distributed servers acting as one, or may be imbedded into the functionality of the centralised session set-up policy server 12.

The session control edge nodes 21 are configured to send status messages 22 indicating overload to the centralised overload control server 18. The overload server acknowledges the overload messages by sending responses 24 to the session control nodes. The overload status messages can be sent at regular intervals and/or sent when a session control edge node experiences overload or approaches an overload threshold. The status messages may periodically be refreshed and/or have a 'validity timeout' attached to it. The status message comprises a required overload restriction, for example that the session control node 21A-D requires an offered load of X session set-ups per second, or that a percentage of the traffic to the node needs to be dropped. The session control nodes determine their load status by determining the processor load, i.e. CPU occupancy or software message queue lengths. The required overload restriction for each edge node is stored in the store 42.

Each session control edge node can be configured with the address of the centralised overload control server 18, for example as a DNS fully qualified domain name (FQDN). The edge nodes 21A-D comprise a message generating module 23 (only one shown in the figure) arranged in operation to send overload messages 22 at regular intervals, or due to a sudden load increase, to the session control edge node

5 overload status module 38 in the overload control server 18, which stores the address of each edge node in the memory 39 as well as information about any interconnects associated with the edge node.

Alternatively, when a new session control edge node 21 is added to the network 20 its identity, e.g. IP address or SIP URI, is added to the memory 39 of the centralised overload control server 18 which initially probes the node for overload status messages.

10 The overload control server 18 is configured to inform or update 3 the session set-up policy server 12 of any overload restrictions required by the session edge nodes 21A-D via a management interface, not shown in figure 1. The session set-up policy server 12 stores in the store 36 session set-up policies governing which destination edge node 21A-D a source edge node 21A-D should preferably use in order to reach the destination network or end user 16 as well as which alternative edge nodes to use and in which order. The

15 session set-up policy server receives session set-up requests 4 from all source session control edge devices 21 and therefore it can determine the total traffic, i.e. the number of session set-ups, intended for each destination session control node. The session set-up policies are static but based on the overload restriction requirements received from the overload control server 18 and the number of session set-up requests intended for the session control node, the modification module 41 is configured to filter out any session

20 set-up policies received from the selection module 40 which rely on an overloaded session control edge node 21A-D and/or one or more connections to a downstream network connection, controlled by one of the session control edge nodes, and indicated as overloaded in the load status messages from said session control edge node. If a session control edge node has signalled to the centralised overload control server that it is overloaded and requires a maximum load of X session set-ups per second, the modification module

25 41 is configured to set a rate limiter (leaky bucket) that monitors initial session set-up responses from selection module 40 to requesting sources. If the primary session set-up choice is via the node which has requested the maximum load of X session set-ups per second and if directing the source node to establish a session to said node would cause the set-up rate to exceed the set-up rate requested by the node, the modification module 41 will remove it from the session set-up choices sent to the source nodes in a

30 response 6, thus ensuring the destination does not receive set-ups that would exceed that rate. The total traffic offered to that destination will be pegged at X session set-ups per second. The source node 21A then

routes 7 the session set up to the destination session control node 21D, which forwards 8 the session set up to the network or end user 16.

Once a message is received indicating that an edge node is no longer overloaded or have more or less stringent overload requirements, the overload control server informs the policy server so it can adjust any
5 restrictions when that edge node is selected as an option in its routing decisions.

If the session set-up policy server receives a sudden increase of requests directed to a specific end user the modification module will select the session set-up policies to include in each response such that the load is distributed between alternative destination session control nodes, i.e. directing some requesting control nodes to the primary destination session control node and the remainder to other destination session
10 control nodes in accordance with their load status and indicated session set-up restrictions.

If received status messages indicate that a destination session control node is not overloaded and does not require session setup restriction, the modification module will not modify or filter the session set-up policies. If a destination session control node is overloaded and requests via status messages for session set-up restrictions to be applied, the modification module will modify the set of session set-up policies to
15 include in the response so that the number of session set-up requests directed to said overloaded node is restricted by the requested amount.

In an alternative embodiment the functionality of the centralised overload control server and the centralised session set-up policy server are implemented in one, single, logical server 30. Figure 2 shows such a server 30 comprising a processor 32, a memory 34, a store 36 for storing session set-up policies, an
20 overload status module 38, a policy selection module 40 and a modification module 41. The overload status module 38, policy selection module 40 and modification module 41 are preferably implemented as program code which when loaded into the memory 34 and executed by the processor 32 performs the different steps for optimising the session set-up with respect to overload status of edge nodes in the application layer.

25

If the centralised policy server 12 does not have the capability to filter or modify the session set-up policies the centralised overload control server 18 can act as a proxy for the session set-up requests and filter or modify the session set-up policies received from the policy server before forwarding them to the requesting

session edge nodes 21A-D. In this alternative embodiment the modification module 41 is arranged in the overload control server 18.

With reference to Figure 3 an embodiment in which the centralised overload control server 18 acts as a proxy for the session set-up requests is now described.

A voice and multimedia transit network N_1 interconnects a number of other networks N_2 to N_n via connections C_1 to C_n which are controlled by edge session control devices 21 for session admission and establishment. In this example the edge session control devices 21 are session border controllers (SBCs) S_1 to S_n although other devices such as media gateway controllers, soft switches, SIP trunking servers and entities implementing 3GPP IMS Call State Control Functions could also be the edge devices of the network N_1 . Each SBC S_1 - S_n sends overload status messages to the Overload Control Server OL1 regarding how new session set-ups should be restricted to it or restricted to one of the connections C_i connected to it. This could be the session set-up rate, e.g. calls per second, or a proportional discard value e.g. 0% to 100%, indicating that a proportion of the new session requests should be discarded (e.g. 33% = discard every 3rd call; 20% = every 5th call, 50% = every other call, etc.). When a new session set-up is requested by one of the interconnected networks 14, the serving SBC sends session establishment requests to the Overload Control Server which proxies the request to a SIP Redirect Server R1 (corresponding to the policy server 12 in figure 1) which examines the information in the session request, for example a called telephone number on a destination network, and decides from that information to which out of one or more SBCs the session set-up should be directed. The possible destination SBCs and interconnections are sent back to the Overload Control Server as session set-up policies from the SIP Redirect Server R1 in a priority order. The Overload Control Server examines the first priority policy against any overload restriction criteria signalled by the corresponding destination SBC. If the first policy passes the restriction criteria, all the session set-up policies are returned to the requesting SBC unaltered. If the first policy fails the restriction criteria, it is removed from the policy options and the second session set-up policy is tested against the restriction criteria signalled by the corresponding destination SBC. If this test passes, this and the remaining session set-up policies are returned to the requesting SBC else it is also removed from the policies options and next policy is tested. This recursive process continues until one of the session set-up policies passes its associated restriction criteria or the session set-up policy list is exhausted. If policy list is exhausted, the Overload Control Server sends a set-up rejection message to the requesting SBC.

In an extension to this described case, the destinations edge devices do not only communicate the overload status / target session set-up demand for themselves but can also include a report on the status of connections to other networks.

With further reference to Figure 3 an embodiment including this expansion of the general description above will now be described. In this example each connection C_1 to C_n , between the network N1 and the other networks N2 to N_n , is assumed to be identified by a SIP trunk group as defined in RFC 4904 [2].

The example further uses SIP as the signalling protocol as defined by RFC 3261 [1] to demonstrate the principal of the invention. The SIP messages used are illustrative and for brevity, only the relevant fields in these messages are shown to demonstrate the invention rather than that of a full implementation. The parameters "restrictor-value" and "restrictor-type" are illustrative extensions to SIP contact headers that could be used and are specific to this example, reflecting that different restriction schemes can be applied within the invention. In this example restrictor-type=cps indicates that a 'calls per second' target delivery rate for new session set-up is being requested as opposed to another scheme such as restrictor-type=pd to indicate 'percentage discard' of new sessions requests.

15

Example of overload control status message sequences

Step 1. Each SBC S1 to S_n is configured to send overload restriction messages to the Overload Control Server OL1 at periodic intervals and when there is a significant change in overload status. This can be both for the total processing capacity for the totality of its own processing and/or for some defined processing limit for the connections C_1 to C_n it serves to other networks.

20

Step 2. Each SBC S1 to S_n sends a SIP OPTIONS message to the Overload Control Server, to convey the overload status information in line with the design and configuration options in Step 1. For example, SBC S4 would send the following to the Overload Control Server OL1 to indicate that it was capable of receiving a total of 100 calls per second and that interconnects C₆, C₇ and C₈ could receive 50, 60 and 40 calls per second respectively.

25

OPTIONS sip: OL1.N1.net SIP/2.0

To: OL1.N1.net

From: S4.N1.net;tag=34678392;restrictor-type=cps; restrictor-value=100

Contact: sip:S4.N1.net;tgrp=C6;trunk-context=N1.net;

Restrictor-type=cps; Restrictor-value=50

Contact: sip:S4.N1.net;tgrp=C7;trunk-context=N1.net;

Restrictor-type=cps; Restrictor-value=60

5 Contact: sip: S4.N1.net;tgrp=C8;trunk-context=N1.net;

Restrictor-type=cps; Restrictor-value=40

Step 3. Either through static configuration or dynamically in response to the first appearance of an OPTIONS message, the Overload Control Server OL1 instantiates a restrictor mechanism (corresponding to the modification module 41 in figure 1) for each SBC and each trunk group configured within or reported by the SBC. This restrictor mechanism stores the session set-up restrictor values for each SBC and for the trunk groups associated with the connections C_n that the reporting SBC serves. These restrictor values are updated as each overload OPTIONS message is received. The Overload Control Server acknowledges the OPTIONS message by sending a SIP "200 OK" response to each SBC. When the restrictor mechanism for each SBC and trunk group is set up, it also initialises a store (42 in Figure 1) for holding the central view of the dynamic session setup behaviour e.g. sessions per second, being established to that SBC and trunk group. In this example, this would be the calls per second for each SBC in totality and each individual trunk group and would be initially set to zero. In this example it is assumed that SBCs S_1 to S_n have signalled via OPTIONS messages that they have maximum session rate limits of sr_1 to sr_n respectively and in addition SBCs S_1 to S_n have also conveyed that there is a session rate limit associated with their connections to other networks C_1 to C_n of cr_1 to cr_n respectively.

Step 4. New session set-up requests are made via one of more source SBCs, some of which will be destined for the same destination SBC and network interconnect. In this context consider a new session set-up request received from a Source Network (N_2) on interconnect (C_1) at SBC (S_1) as a SIP INVITE request.

INVITE sip:+16305550100@S1.N1.net;user=phone SIP/2.0

25 To: sip:+16305550100@S1.N1.net;user=phone

...

Contact: <sip:source@N2.net; user=phone>

...

Step 5. SBC (S1) sends a modified INVITE request with the originating trunk group identifier of the interconnect (C₁) to the Overload control server using the same SIP URI as that used for the overload

5 OPTIONS requests (i₂).

INVITE sip:+16305550100@OL2.N1.net;user=phone SIP/2.0

To: sip:+16305550100@OL2.N1.net;user=phone

...

Contact: <sip:source@S1.N1.net;tgrp=C1;

10 trunk-context=N1.net;user=phone>

...

Step 6. The Overload Control Server proxies the INVITE request to the SIP Redirect Server (R1) but demands that responses must be returned via itself on interface i₁ by adding a SIP Record-Route header

INVITE sip: +16305550100@R1.N1.net;user=phone SIP/2.0

15 To: sip:+16305550100@R1.N1.net;user=phone

...

Record-Route: <sip:R2.N1.net;lr>

Contact: <sip:source@S1.N1.net;tgrp=C1;

trunk-context=N1.net;user=phone>

20 ...

Step 7. The SIP Redirect Server (R1) determines that the end user address +16305550100 is served by Destination Network (N4) and would prefer to connect directly via interconnects C₃, C₇ or C₅ but if these are unavailable or congested, to reach network N4 via the Transit Network N5 using interconnects C₈ or C₉. The Redirect Server (R1) therefore constructs a SIP 300 (Multiple Choices) response and sends it to the

Route: <sip:R2.N1.net;lr>

Contact: <sip:+16305550100@S4.N1.net;grp=C7;
trunk-context=N1.net;user=phone>

Contact: <sip:+16305550100@S4.N1.net;grp=C8;
trunk-context=N1.net;user=phone>

15 Step 8. The Overload Control Server examines the restriction monitors for the first choice entry in the contacts list for SBC S2 and then the associated trunk group C₃.

It first determines if allowing this session set-up would exceed the restrictor value set for the SBC S2 of sr_2 . If allowing this session set-up choice would cause that restrictor value to be exceeded then this Contact header

is deleted from the list. If the SBC rate value is not exceed the Overload Control Server OL1 examines the restrictor value cr_3 for the connection C_3 , indicated in the trunk group parameter. If this session set-up would cause the value cr_3 to be exceeded then this Contact header

Contact: <sip:+16305550100@S2.N1.net;tgrp=C3;

trunk-context=N1.net;user=phone>

is deleted from the list.

When a Contact header is deleted from the session set-up destinations list the Overload Control server
 5 examines the next Contact header and applies the same logic. In this example if directing the session to be
 set-up to SBC S2 or connection C₃ would exceed the requested restrictor limits then after deleting the
 contact header the next contact header become the first choice and the restrictor sr₄ for SBC 4 is similarly
 tested followed by the restrictor cr₇ for connection C₇ if the sr₄ restrictor test did not fail.

This process is repeated until a Contact header passes its restrictor test or the Contact header list is
 10 exhausted.

No Overload Restrictions are Applied.

Step 9. If in Step 8, the call rate of sending the session set-up to SBC 2 and C₃ does not exceed the
 restrictor rates associated with these two destination entities, the Overload Control Server OL1 updates its
 record of the session setup behaviour, in this example the calls per second for both SBC S2 and for
 15 connection C₃.

Step 10. The Overload Control Server OL1, removes the route header in the response received from the SIP
 Redirect Server R1 and proxies the response to the requesting SBC S1 with the Contact Header list
 unaltered.

300 (Multiple Choices)

20 Contact: <sip:+16305550100@S2.N1.net;tgrp=C3;

trunk-context=N1.net;user=phone>

Contact: <sip:+16305550100@S4.N1.net;tgrp=C7;

trunk-context=N1.net;user=phone>

Contact: <sip:+16305550100@S3.N1.net;tgrp=C5;

trunk-context=N1.net;user=phone>

Contact: <sip:+16305550100@S4.N1.net;tgrp=C8;

trunk-context=N1.net;user=phone>

Contact: <sip:+16305550100@S5.N1.net;tgrp=C9;

5 trunk-context=N1.net;user=phone>

Step 11. On receipt of the 300 (Multiple Choices) message the SBC S1 would use the first Contact header to form an invite to SBC S2 and trunk group C₃.

INVITE sip:+16305550100@S2.N1.net;tgrp=C3;

trunk-context=N1.net;user=phone SIP/2.0

10 To: sip:+16305550100@S2.N1.net;tgrp=C3;

trunk-context=N1.net;user=phone

...

Contact: <sip:souce@S1.N1.net;tgrp=C1;

trunk-context=N1.net;user=phone>

15 Step 12. On receipt of the session set-up request, SBC S2 would pass the request to network N4 over connection C₃ to complete the set-up.

INVITE sip:+16305550100@N4.net;

trunk-context=N1.net;user=phone SIP/2.0

To: sip:+16305550100@N4.net;

20 trunk-context=N1.net;user=phone

...

Contact: <sip:souce@S3.N1.net;

trunk-context=N1.net;user=phone>

Some Overload Restrictions are Applied.

Step 13. Same as Step 4

Step 14. Same as Step 5

Step 15. Same as Step 6

5 Step 16. Same as Step 7

Step 17. Same as Step 8

Step 18. In this example, in Step 17 if the call rate of sending the session set-up to SBC 2 exceeds the associated restrictors then the first contact header is deleted from the session set-up options list and the Overload Control Server OL1 does not update its record of the session setup behaviour, in this example the
10 calls per second for the SBC S2 nor the Connection C₃.

Also in this example, in Step 17 if the call rate of sending the session set-up to connection C₇ exceeds the associated restrictors then the second contact header is deleted from the session set-up options list and the Overload Control Server OL1 does not update its record of the session setup behaviour, in this example the calls per second for SBC S4 nor the Connection C₇.

15 Then, in this example, in Step 17, the call rate of sending the session set-up to the destination entities within the third Contact header returned from SIP Redirect Server R1, i.e. SBC 3 and C₅ does not exceed the restrictor rates associated with each of them, the Overload Control Server OL1 updates its record of the session setup behaviour, in this example the calls per second for both SBC S3 and for connection C₅.

Step 19. The Overload Control Server OL1, removes the route header in the SIP 300 response received from
20 the SIP Redirect Server R1 and proxies the modified response with the deleted Contact headers to the requesting SBC S1.

300 (Multiple Choices)

Contact: <sip:+16305550100@S3.N1.net;tgrp=C5;

trunk-context=N1.net;user=phone>

25 Contact: <sip:+16305550100@S4.N1.net;tgrp=C8;

trunk-context=N1.net;user=phone>

Contact: <sip:+16305550100@S5.N1.net;tgrp=C9;

trunk-context=N1.net;user=phone>

- Step 20. On receipt of the 300 (Multiple Choices) message the SBC S1 would use the first Contact header in
5 the now modified list to form an INVITE to SBC S3 and trunk group C₅.

INVITE sip:+16305550100@S3.N1.net;tgrp=C5;

trunk-context=N1.net;user=phone SIP/2.0

To: sip:+16305550100@S3.N1.net;tgrp=C5;

trunk-context=N1.net;user=phone

10 ...

Contact: <sip:souce@S1.N1.net;tgrp=C1;

trunk-context=N1.net;user=phone>

- Step 21. On receipt of the session set-up request, SBC S3 would pass the request to network N4 over
connection C₅ to complete the set-up.

15 INVITE sip:+16305550100@N4.net;

trunk-context=N1.net;user=phone SIP/2.0

To: sip:+16305550100@N4.net;

trunk-context=N1.net;user=phone

...

20 Contact: <sip:souce@S3.N1.net;

trunk-context=N1.net;user=phone>

Overload Restrictions are Applied to All Destination Choices.

- Step 22. Same as Step 4

Step 23. Same as Step 5

Step 24. Same as Step 6

Step 25. Same as Step 7

Step 26. Same as Step 8

- 5 Step 27. If none of the session set-up choices in the Contact header list pass their respective restrictor tests, the Overload Control Server OL1 returns a SIP response to SBC S1 to reject the session setup request.

503 Service Unavailable

Step 28. SBC S1 passes the reject response to the originating network N1.

503 Service Unavailable

- 10 which in turn signals back to the source and the session establishment is abandoned.

Examples of further applications and extensions

With reference to Figure 4 a further embodiment where other possible destination entities to which a SIP Redirect Server R1 could send calls to are shown. The could be for example:

- 15 a) a call server CS1 to which a number of end users are registered
- b) a Media Gateway Controller MGC1 that signals to a Network N3 via connection C₃ and not using SIP, e.g. Signalling System 7 or
- c) a SIP Trunking Server STS1 providing IP connectivity to VoIP PBXs.

- 20 In this example the overload status message sent from these destination entities via SIP OPTIONS messages could include an extension to the Contact Header to set and change a restrictor in the centralised overload server for the destination entity itself e.g. CS1, MGC1 and SC1, but also for the destination address, e.g. called telephone number, it serves. This could be for single numbers, ranges of numbers and SIP URIs. For example:-

25 OPTIONS sip:OL1.N1.net SIP/2.0

To: OL1.N1.net

From: CS1.N1.net;tag=34678392;restrictor=100;restrictor-type=cps

Contact: sip:CS1.N1.net;Restrictor-type=cps; Restrictor-value=50;

Restrictor-adrs=+16305550100; Restrictor-adrs-type=E164

Contact: sip:CS1.N1.net;Restrictor-type=cps; Restrictor-value=60;

5 Restrictor-adrs=+16307770100--+16307770199;

Restrictor-adrs-type=E164-range

Contact: sip:CS1.N1.net;Restrictor-type=cps; Restrictor-value=40;

Restrictor-adrs=fred.blogs@N1.net;

Restrictor-adrs-type=SIP-URI

10 The restrictors associated with destination addresses are envisaged as being dynamically created in the Overload Control Server on receipt of the first OPTIONS message that referenced it and would be removed after an expiry period when updates via subsequent OPTIONS messages for that destination address were not received by a specific cancellation update.

15 Exemplary embodiments of the invention are realised, at least in part, by executable computer program code which may be embodied in application program data provided by the program modules in the Centralised overload control server 18 and Centralised session set-up policy server 12. When such computer program code is loaded into the memory of each server for execution by the respective processor, it provides a computer program code structure which is capable of performing at least part of the methods in
20 accordance with the above described exemplary embodiments of the invention.

Furthermore, a person skilled in the art will appreciate that the computer program structure referred can correspond to the process flow shown in figures 1, 3 and 4 where each step of the processes can correspond to at least one line of computer program code and that such, in combination with the processor
25 in respective server, provides apparatuses for effecting the described process.

The above embodiments are to be understood as illustrative examples of the invention. Further embodiments of the invention are envisaged. For example, the modules or part of the modules for effecting the described process can be implemented in hardware or a combination of hardware and software.

- 5 Further, the session set-up policies stored in the server system could instead of being excluded by the selection module based on the overload status messages received from the session control edge nodes be modified or amended to avoid session set-up recommendations to overloaded destination session control edge nodes or their interconnects.
- 10 In summary, a centralised server system is arranged to monitor overload restriction requirements of session control nodes within a globally scalable VoIP and Multimedia network in order to handle session set-up requests. An overload status module in the server system receives overload status messages from the session control nodes and based on these messages and the demand on each session control node determined from session set-up requests, the server system selects which session control nodes as well as
- 15 which of their associated interconnects to other networks to include in a session setup response to a session setup request from a source session control node.

References

- 20 [1] RFC 3261 SIP: Session Initiation Protocol, Internet Engineering Task Force
- [2] RFC 4904 Representing Trunk Groups in tel/sip Uniform Resource Identifiers (URIs), Internet Engineering Task Force

Claims

1. A method for optimising session set-up between session control nodes, which nodes interface to different networks and/or end user terminals in a globally scalable VoIP network, said method comprising at a centralised server system:

5 receiving a load status message indicating a load status from each session control node;

receiving session set-up requests from said session control nodes where each session set-up request identifies a sender node and a destination node and determining from said requests the number of session set-up requests directed to each destination session control node;

10 for each received session set-up request selecting one or more session set-up policies to include in a response to the requesting session control node based on said indicated load status and the number of session set-up requests directed to each session control node ; and

sending the one or more selected session setup policies to the requesting session control node.
- 15 2. A method according to claim 1 further comprising receiving in said load status message a request for a defined restriction on new session setups per second.
3. A method according to claim 2 wherein if by setting up a session from a source node to a destination node the setup rate requested by said destination node would be exceed, excluding from

20 the response any session setup policies which direct the source node to said destination session control node.
4. A method according to claim 1 further comprising receiving in said load status message a request for a proportional discard of all new session set-ups.

25
5. A method according to any preceding claim further comprising said session control nodes sending the load status message upon approaching an overload threshold.
6. A method according to any preceding claim wherein said load status message received from each

30 session control node comprises the overload status of the node and the overload status of any associated trunk connections to other networks.

7. A method according to any preceding claim further comprising reinstating a session control node and/or a connection as selectable options in a session setup policy decision upon said session control node indicating no overload in a new load status message.
- 5 8. A method according to any of the preceding claims further comprising storing an overload status table in the server system indicating an overload status and an overload restriction criteria for each session control edge node and the overload status and an overload restriction criteria for their associated trunk connections to other networks.
- 10 9. A server system for controlling session setup between session control nodes, which nodes interface to different networks and/or end user terminals in a globally scalable VoIP network, said server system comprising:
 - 15 a session setup policy module arranged in operation to select one or more session setup policies in response to session setup requests received from said session control nodes, where each session setup request identifies a sender node and a destination node;
 - an overload control module arranged in operation to receive from each session control node a load status message indicating a load status;
 - the server system being further arranged in operation to determine from said received session setup requests the number of session set-up requests directed to each session control node, and to select
 - 20 one or more session setup policies to include in a response to a requesting session control node based on said received load status and the number of session set-up requests directed to each session control node.
- 25 10. A server system according to claim 9 in which the server system is arranged in operation to store an overload status table indicating an overload status and an overload restriction criteria for each session control edge node and the overload status and an overload restriction criteria for their associated trunk connections to other networks.
- 30 11. A server system according to claims 9 to 10 in which the functionality of the server system is distributed in one or more session set-up policy servers and overload status servers communicating with each other via one or more management interfaces.

12. A server system according to claim 11 in which the overload status server is configured as a proxy server for the session set-up policy server.
- 5 13. A server system according to any of claims 9 to 12 in which the session control nodes are Session Border Controllers, Call Servers, Softswitches, Application Gateways , Media Gateways controllers or Call State Control Functions in an IMS network.
- 10 14. A session control node of a first network that interfaces to a different network in a globally scalable application layer VoIP transit network, said session control node comprising an overload status message generating module arranged in operation to send overload status messages and overload restriction criteria for the node as well as for any associated trunk connections to downstream networks, to a centralised server system in the application layer network.
- 15 15. A computer program or suite of computer programs executable by a processor to cause the processor to perform the method of any one of claims 1 to 8.
16. A non-transitory computer readable storage medium storing a computer program or a suite of computer programs according to claim 15.