Whitepaper

Failover Management of SIP-based Multimedia Communication Sessions: Design Improvements for Call Continuity

Abstract

With increasing digitization and proliferation of IP networks, use of SIP protocol to provide IP telephony services has seen a steady rise. In IP telephony networks, call continuity in the event of a failover is of primary importance.

The paper presents design improvement proposal on top of the patented “Failover Management of SIP-based Multimedia Communication Sessions” [1] work to increase the call-continuity ratio and reduce the time a secondary server currently takes to take over the call for better user experience, setup and call flows with media (RTP) passing through the SIP server.

1. Introduction

With the current patented [1] solution design, we are able to provide call concurrency in failover cases for SIP-based multimedia communication sessions.

The patented [1] solution design provides an approach for call continuity in a deployment where SIP server acts as a Signaling Gateway Application. This may not require endpoints to be REGISTERED and media to be traversed via the SIP server, as may be required in other deployments such as Softswitch, (Hosted) IPPBX.

The patented design also does not take into account the need for starting secondary server’s service beforehand i.e. before the handover takes place in the event of a failover. It takes considerable time for the secondary server to initiate services and this delays the call continuity handover.

Hence, to provide better call continuity during failover in a high-availability setup, we propose the following design improvements to the existing patent:

- Improve call continuity probability by sending “Re-INVITE” for non-registered endpoints also.
- Include use cases where media (RTP) does not pass directly between the endpoints but passes through the SIP server which has media traversal capacity to provide
    - RTP flow among different networks
    - Services like call recording
- Improve call continuity user experience by reducing the time it takes for a secondary SIP server to take over the call by proposing a design change which enables services at the secondary SIP server to start beforehand.

The paper is organized as follows:

- Section 2 presents the current architecture, design and its challenges
- Section 3 introduces our suggestions to optimize the current design, overcome the challenges and provide better user experience for call continuity during high availability
- Section 4 presents conclusions and future considerations

2. Call Continuity Using High Availability - Current Architecture

The section explains the current design as per the patent [1].

Shown below is a high availability deployment diagram to achieve call continuity during failover in a high availability setup. The setup uses VRRP [2] based keepalived [3] service to provide failover discovery and virtual IP movement among the SIP servers.

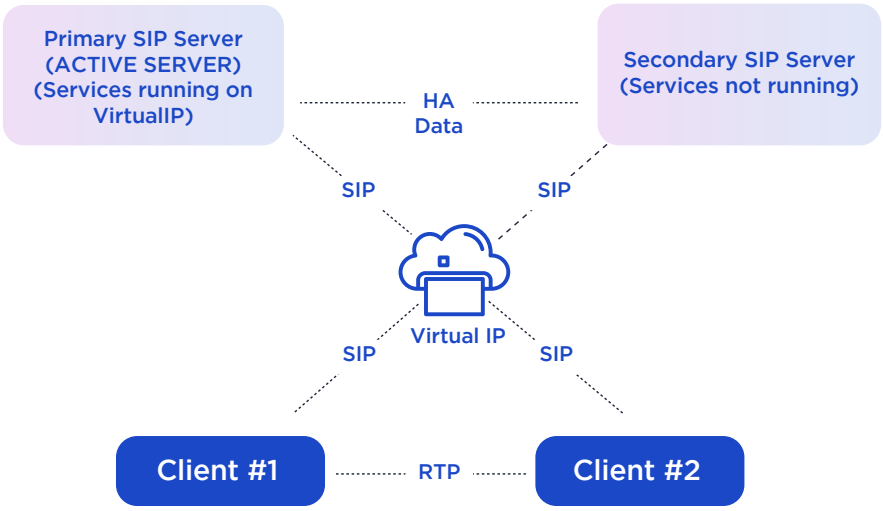

Figure 1: SIP High Availability Deployment Diagram - Before Failover

In the example above, “Client #1” and “Client #2” are connected using a virtual IP, which is currently assigned to “Primary SIP Server” marked as “ACTIVE SERVER”.

Primary Server serves the current SIP sessions while the media (RTP) path is between the clients directly.

Primary server keeps the call information (“HA Data”) at the secondary server using one or more methods. This call information is required at the secondary server to allow “Call Continuity”.

In the event of a failover, i.e. non reachability of primary server, Virtual IP is moved to secondary server (using VRRP [2] based keepalived [3] or similar services) and secondary server now becomes “ACTIVE SERVER” as depicted below:

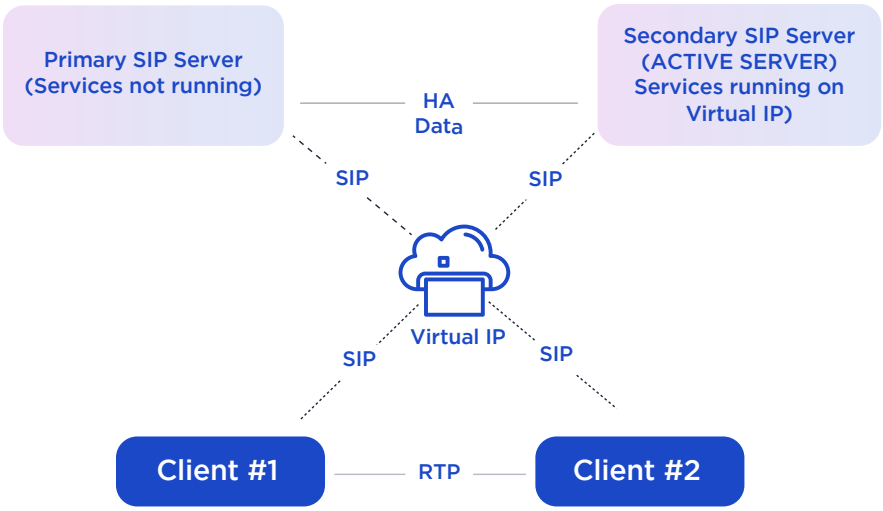

Figure 2: SIP High Availability Deployment Diagram - After Failover

As soon as the secondary server becomes “ACTIVE”, patented design [1] proposes services to start using virtual IP.

In the example, secondary server, uses the “HA Data” of active calls as sent by primary server before the failover, to provide call continuity feature using the design as in “Failover Management of SIP-based Multimedia Communication Sessions” [1].

2.1. Challenges

The solution design in the current proposal does not take into account the below-mentioned use cases:

- **Delayed or No SIP REGISTER by SIP clients**

  Clients may not send “SIP REGISTER” message to the secondary server or there may be considerable delay in sending the “SIP REGISTER” message.

  Issue: Current solution does not send Re-INVITE for the endpoints which are not REGISTERED. In this scenario, ongoing calls for the clients who are not registered at the secondary server, will not be continued.

  Challenge: Low probability of call continuity due to non-registration by SIP clients.

- **Media (RTP) passing through the SIP server.**

  Issue: There are use cases where media (RTP) may need to pass through the SIP server e.g. to support call recording, NAT traversal etc. Current design does not take into consideration the media (RTP) and in these scenarios, call signaling will be continued but media (RTP) may be lost.

  Challenge: Issue of “No Voice” for call continuity cases where media (RTP) is passing through the SIP server.

- **Delay due to application initialization process at the secondary SIP server**

  Issue: Since, with the current design, the application runs on virtual IP rather than physical IP, application at the secondary server will initially be stopped and needs to be started in the event of failover. This is because, virtual IP will be available at the secondary SIP server only in the event of a failover. This takes considerable time, causing delay in secondary server’s processing of “call continuity”. This might also result in no media between the clients during this handover.

  Challenge: Delay in call continuity due to the time it takes to start the application at the secondary server.

3. Improved Design for Call Continuity

We propose a design to improve the current architecture and solve all the challenges listed in section 2.

In the proposed solution, we take into account the following design considerations:

- **Delayed or No SIP REGISTER by SIP clients.**

Improve the design of SIP server to send Re-INVITE as in the patent [1] without waiting for REGISTER request from the SIP clients.

In the event of a failover in high availability setup, Secondary SIP Server will become "ACTIVE". It will use "HA Data" information to send Re-INVITE to the SIP clients to continue calls without waiting for these SIP clients to be REGISTERED. This will ensure continuity of the calls by sending Re-INVITE even for endpoints that are not REGISTERED, thereby improving the probability of call continuity.

- **Media (RTP) passing through the SIP server.**

Current design [1] uses existing media (RTP) parameters in "HA Data" [Figure 1] to be stored at secondary server and secondary server takes these information to construct Re-INVITE SIP method to be sent to both "Client #1" and "Client #2".

With media (RTP) passing through the SIP server, this media information (IP address and port number) may not be valid at the secondary SIP server. Secondary SIP server will use new media information (IP Address and port number) to construct Re-INVITE SIP method to be sent to both "Client #1" and "Client #2"

This ensures that both clients use updated RTP information during call continuity and thus no voice in this particular case, gets resolved.

Media (RTP) passing through the SIP server, may affect network and server performance and sizing.

- **Delay due to application initialization process at the secondary SIP server.**

Use of physical IP for application rather than virtual IP so that we can keep the services "ON" at the secondary server to reduce the time it takes for a secondary server's application to start in the event of a failover

Initialization process will still take the time to process sending Re-invites from the secondary server in high volume scenarios.

Given below is a proposed High Availability Deployment diagram with improved design to achieve call continuity during failover in high availability setup. The setup uses VRRP [2] based keep alived [3] service to provide failover discovery and virtual IP movement among the SIP servers.

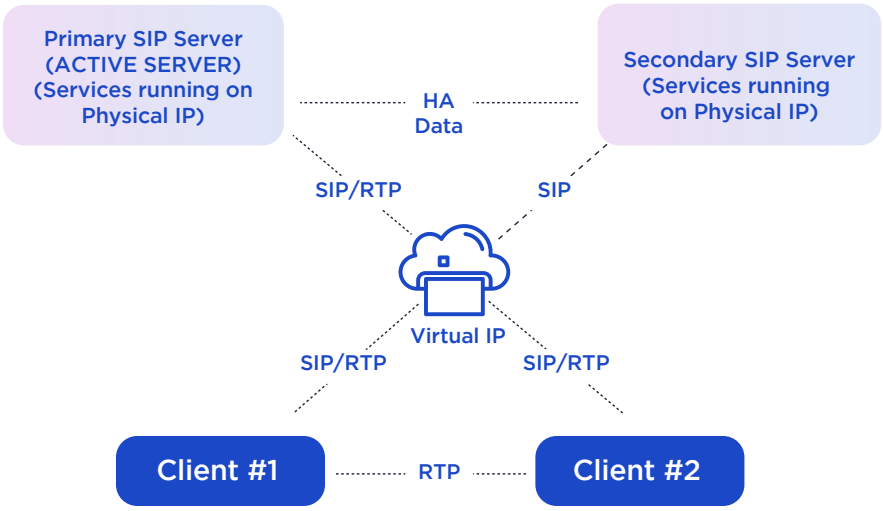

Figure 3: Proposed SIP High Availability Deployment Diagram - Before Failover

Figure-3 depicts the 'before the failover' setup. Both the primary and secondary SIP servers now use physical IP with services running on both the servers. The example also uses media (RTP) passing through the SIP server.

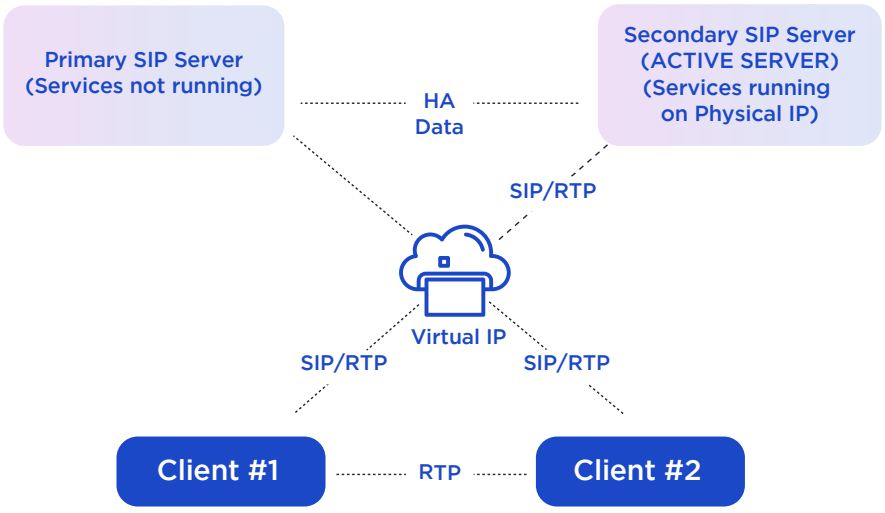

Figure 4: Proposed SIP High Availability Deployment Diagram -After Failover

Figure-4 depicts the 'after the failover' setup. Secondary SIP server is already running using the physical IP. Services on primary.

In the example, the secondary SIP server uses "HA Data" as received from the primary SIP server before the failover and uses updated media (RTP) information to send Re-INVITE as in the patent[1].

3.1. Experiments

Taking the above improvements into consideration, we ran tests for call continuity during failover in high availability cases for setup with clients delay in REGISTRATION, media passing through the SIP server and application running on physical IP rather than virtual IP. Listed below are the results of the experiments that prove the significance of the work in providing better call continuity and user experience in the event of a failover.

- **Delay due to application initialization process at the secondary SIP server.**

  SIP REGISTER message is sent by endpoints / clients to the SIP server to inform or register the location (IP Address / Port) with the SIP server. This helps the SIP server to route incoming calls to these endpoints / clients as per their correct location (IP Address /Port).

  The original patent [1] was for a solution flow supporting calls originating from non-subscribers, which do not need to send SIP REGISTER. Hence there was no issue.

  If we use the same design to propose call continuity in a solution requiring support of subscribers who notify their location using SIP REGISTER message, we need to take care of SIP REGISER message in design.

  With solutions requiring support for subscribers, in the event of a failover and using the current patent design [1], the secondary SIP server can send ReINVITE to only those subscribers who have already sent SIP REGISTER messages. Now, in the event of a failover, the current patent [1], sends ReINVITE, immediately after failover to ensure immediate call continuity. The subscribers' endpoints send REGISER after a certain duration as defined by SIP REGISTER expire [4] SIP header.

  Before:
  Assuming 20 seconds for SIP REGISTER Expire [4], 10 calls during failover and random SIP REGISTER time for all endpoints. We see approximately 30% success rate. The success rate can vary depending on value of Expires [4], SIP REGISTER time from SIP clients.

  After:
  We propose a change of call flow where the secondary SIP server will not check for SIP REGSITER message, and will use the location information of endpoints/clients as received from the primary SIP Server for call continuity.

  With design improvement for sending Re-INVITE without waiting for REGISTRATION from the SIP clients as explained in section 3, for the above use case, we can achieve 100% success in call continuity.

- **Media (RTP) passing through the SIP server**

  In usual SIP call flows, media (RTP) is usually between the endpoints / clients and this media (RTP) does not flow via the SIP Servers.

  To support features like call recording or NAT (where endpoints / clients are in different networks), it is required to pass media (RTP) flow via the SIP Server.

.

The original patent [1] did not consider the media (RTP) passing through the SIP server. For solutions requiring media (RTP) to be traversed through the SIP server, we may need to propose changes in the current patent [1] to support media (RTP) to be part of the call continuity design.

Before:
For setup involving media (RTP) passing through SIP server, after failover, the call is continued i.e. SIP signaling is continued, but there is no media (RTP) between the SIP clients, resulting in “No Voice” issues.

After:
We propose a design change: send updated Media (RTP) information of the secondary SIP server in the ReINVITE message. Using this updated media (RTP) information, we can achieve call continuity. This can be achieved for the call flow too where RTP is passing through the SIP Server.

With design implementation for updating the media (RTP) information as explained in section 3, we have been able to perform call continuity successfully with media (RTP) for the above use case.

RTP passing through the SIP server requires additional processing and may impact concurrency by 60-80% using the same hardware

- **Delay due to application initialization process at the secondary SIP server.**

Before:
Original patent [1] design was to start services of the secondary SIP server after failover occurred. As all services use network IP (virtual IP) which is assigned to the secondary SIP server only in the event of a failover, it results in some delay at secondary server’s services to start and enable call continuity.

Delay of approximately two minutes at the secondary SIP server to start services on virtual IP address resulting in no call continuity for two minutes.

- After:
We propose a design, where the core services will run on “0.0.0.0”. “In the context of servers, 0.0.0.0 can mean "all IPv4 addresses on the local machine". If a host has two IP addresses, 192.168.1.1 and 10.1.2.1, and a server running on the host is configured to listen on 0.0.0.0, it will be reachable at both of those IP addresses” [5]

With this, the secondary server, in order to start services, no longer needs to wait for the failover to occur and virtual IP to be assigned to the secondary SIP server.

With design implementation of using “0.0.0.0” IP for running service s as explained in section 3, services on the secondary SIP server can be started before failover. This results in saving the two minutes of failover at the secondary SIP server.

Secondary server will send these Re-INVITE based on the current CPS support of the SIP server. This may delay sending Re-INVITE messages to endpoints in a high volume scenario. Example: Assuming current CPS support is 50, then per second 50 calls will be initiated. If the number of calls being continued during high availability case are say 500, then the last 50 calls will take 500/50 = 10 seconds to get Re-INVITE.

4. Conclusions

We showed that using enhancements as proposed in the paper, we can increase the user experience of call continuity during failover cases by reducing the failover time in call continuity.

We can also propose the solution to use cases involving functionalities other than SIP gateways, such as Softswitch, (Hosted) IPPBX, where users are REGISTERED and may use SIP gateways for media traversal.

This effectively serves the use cases with RTP flowing through SIP server, increases the probability of call continuity and reduces the time it takes to handover. All these combined, result in better user experience.

It is important to note that one or more approach can also be used in other similar applications to achieve faster handover of ongoing sessions at the secondary server in the event of a failover.

4.1. Future Considerations

In the days ahead, we will need to work on ensuring call continuity from secondary to primary server handover cases to make it more effective and automated.

Proposed design as shown in section 3 updates the media (RTP) information while sending Re-INVITE for call continuity. Some SIP clients may not support this. We may need to improve this design to use earlier shared media (RTP) information at the secondary SIP server to support such SIP clients.

.

About the author

Shiv Premani
Engineering Manager

Shiv Premani is a computer science enthusiast and an innovator with over 16 years of experience in IT Product Development. As an Engineering Manager and Software Architect at STL, Shiv oversees Product Development roadmap, custom projects' requirements and is involved from SRS to the delivery phase of mission-critical projects.

Kalpesh Shukla
Sr Tech Lead

Kalpesh Shukla is an IT professional with extensive experience in architecting and developing complex software with over 13 years of experience. He is a quick learner and have worked on various technologies including Angular JS, PHP, MySQL, Radius and Diameter Protocol. As Software Architect and tech lead at STL, Kalpesh oversees product development and delivery of important roadmap and project requirements.

References

1 Original Patent "Failover management of SIP based multimedia communication sessions". https://patents.google.com/patent/US10367856B2/en?oq=10367856
2 VRRP Protocol https://en.wikipedia.org/wiki/Virtual_Router_Redundancy_Protocol
3 Keepalived: Load balancing & High-Availability https://github.com/acassen/keepalived/blob/master/README.md
4 SIP Expires https://tools.ietf.org/html/rfc3261#section-20.19
5 0.0.0.0 IP Address https://en.wikipedia.org/wiki/0.0.0.0

About STL - Sterlite Technologies Ltd:

STL is an industry-leading integrator of digital networks.
We design and integrate these digital networks for our customers. With core capabilities in Optical Interconnect, Virtualized Access Solutions, Network Software and System Integration, we are the industry's leading end-to-end solutions provider for global digital networks. We partner with global telecom companies, cloud companies, citizen networks and large enterprises to deliver solutions for their fixed and wireless networks for current and future needs.We believe in harnessing technology to create a world with next generation connected experiences that transform everyday living.

With intense focus on end-to-end network solutions development, we conduct fundamental research in next-generation network applications at our Centre of Excellence. STL has a strong global presence with next-gen optical preform, fibre and cable manufacturing facilities in India, Italy, China and Brazil, optical interconnect capabilities in Italy, along with two software-development centres across India and one data centre design facility in the UK

For more on STL, visit: www.stl.tech