A Fast Failure Detection and Failover Scheme for SIP High Availability Networks[1]

Wei-Ming Wu, Kuochen Wang, Rong-Hong Jan, Chia-Yuan Huang*
Department of Computer Science, National Chiao Tung University
*Computer & Communications Research Lab., Industrial Technology Research Institute**
kwang@cs.nctu.edu.tw

Abstract

The SIP proxy server, which is an important building block of a SIP network, can be in a form of a cluster to prevent service unavailability caused by software or hardware failures. The design of a SIP proxy server cluster with a single dispatcher in a SIP network faces the single-point-of-failure problem. The use of dual SIP dispatcher architecture was proposed and VRRP (Virtual Router Redundancy Protocol) was used as an IP failover mechanism. In this paper, we have designed and implemented a dependable SIP network that includes n + k SIP dispatchers (n active dispatchers + k backup dispatchers) to control and monitor a cluster of m SIP proxy servers for VoIP and video conferencing applications. A fast failure detection and failover (FFF) scheme has also been proposed. FFF uses OpenAIS as a high availability middleware to perform health check and failover of dispatchers and proxy servers. Experimental results have shown that the FFF scheme reduces the dispatcher failover time by 80% compared to the VRRP mechanism and also shorten the proxy failover time by 85% compared to the mechanism provided by OpenSER itself.

1. Introduction

A SIP proxy server is a call-control device that provides many services such as routing of SIP messages between SIP user agents. Since it is directly accessed by SIP user agents, the SIP proxy server is considered as an important device in the SIP networks. The traditional design of SIP high availability uses limited (one or two) dispatchers and multiple SIP proxy servers. Increasing the number of SIP proxy servers will increase the total availability of entire service, but still faces the bottleneck of a limited number of SIP dispatchers.

High availability middleware can provide business continuity, especially the companies that need to fulfill 99.99% or even five nines availability of services. OpenAIS (open application interface specification) [1] is an implementation of Service Availability Forums API Specification. The main feature of OpenAIS is cluster management in order to have high availability. The most important component of OpenAIS is Availability Management Framework (AMF) which is in charge of the health check and the redundancy model.

2. Related work

We classify different design choices to achieve high availability of SIP networks from system architecture point of view.

2.1. Client or DNS based failover

SIP user agents can set a single or multiple backup SIP proxy server entries to deliver SIP requests if the primary SIP proxy server fails to respond. By using DNS SRV records, clients no longer need to record multiple IP addresses or domain names. A DNS SRV record is commonly configured to point to SIP proxy servers and associate the domain name of each service with an IP address. This design can provide failover between multiple SIP proxy servers.

2.2. Single controller for proxy server cluster

F5's BIG-IP [2] system is an application traffic management solution and it provides high availability and reliability of SIP networks. Fig. 1 illustrates a high availability SIP network architecture using F5's BIG-

[1] This work was supported by the National Science Council under Grants NSC94-2219-E-009-023 and NSC96-2219-E-009-012.

0-7695-3054-0/07 $25.00 © 2007 IEEE
DOI 10.1109/PRDC.2007.11

IP solution. The BIG-IP does the advanced health check by sending SIP OPTIONS requests to SIP proxy servers. The BIG-IP product plays an important part of the SIP network architecture because it acts as a SIP server failover controller and a service entry point of SIP user agents. It prevents the system from becoming unavailable when a single SIP media or proxy server fails. However, the system will still become unavailable when BIG-IP itself fails. That is, the single-point-of-failure problem still exists in this architecture.

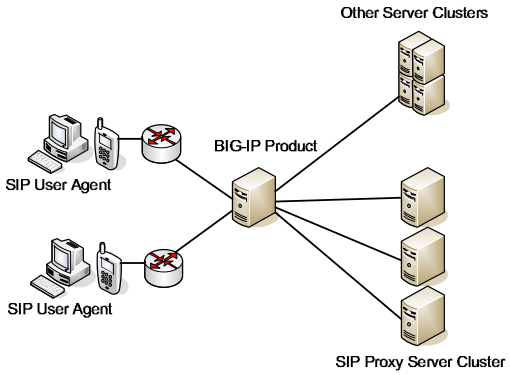

Fig. 1. F5's BIG-IP infrastructure.

2.3. Intelligence in the redundant hop

The design of Cisco IOS SIP High Availability Application [3] is using intelligence at the server side. It uses a redundant hop on SIP servers which can be the SIP proxy servers or SIP load balancers (or called dispatchers). The Virtual Router Redundancy Protocol (VRRP) [4] was used for the redundant control on both sides (active and standby) of redundant nodes. VRRP defines a standard procedure that enables multiple redundant servers on a LAN to negotiate ownership of a single virtual IP address. When the master server becomes unavailable, VRRP selects the highest priority server of the other servers to take over the virtual IP and continue serving the incoming request.

In this design, each hop of SIP servers in the SIP networks used VRRP to set as 1 + 1 redundancy. Fig. 2 shows the architecture of 1 + 1 redundant SIP load balancers. If SIP proxy servers adopt $n + k$ instead of 1 + 1 redundancy model, using 1 + 1 redundant SIP load balancers not only enables the load balancing of SIP proxy servers but also can avoid the single-point-of-failure problem when a SIP load balancer fails. The redundant control of SIP load balancers is based on VRRP.

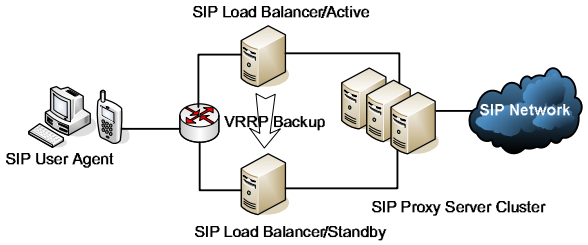

Fig. 2. A high availability SIP network with intelligence placed in the SIP load balancer.

3. System design approach

A fast failure detection and failover (FFF) scheme is proposed. The cluster of SIP proxy servers is controlled by centralized job controllers, which are SIP dispatchers. We use an $n + k$ redundancy model for SIP dispatchers. The reason of using a clustered dispatchers design is not only protecting the system from crashing caused by the single-point-of-failure problem but also improving the overall system availability. Fig. 3 shows the design architecture of FFF for SIP high availability networks. An SAP (service access point) is a virtual IP that is associated with a SIP dispatcher, which is active and is responsible for handling incoming SIP messages. With an SAP, SIP user clients can access the service correctly without knowing the high availability design at the server side. Clients at different regions (for example, regions A and B in Fig. 3) can access respective SAP for enhanced scalability.

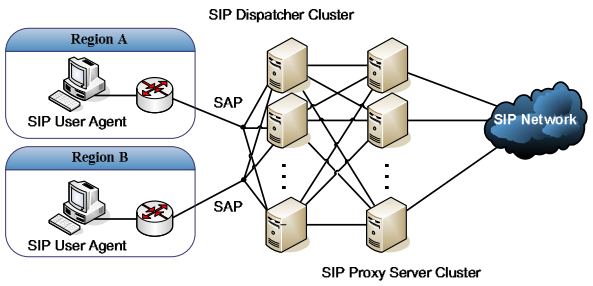

Fig. 3 . Design architecture of the proposed FFF scheme for SIP high availability networks.

The proposed redundancy model of SIP dispatchers is $n + k$ where n is the number of active SIP dispatchers (or SAPs), since each active SIP dispatcher has one virtual IP address (SAP). The operation of SIP proxy servers is an all-active redundancy model because they all can be accessed by any SIP dispatcher.

The mechanism of how a backup SIP dispatcher takes over the SAP is maintained by OpenAIS AMF service. Each SIP dispatcher has its own real IP

address. As to which SIP dispatcher is given the virtual IP (SAP) to become active is assigned by OpenAIS AMF service. AMF services running on each SIP dispatcher would keep sending control messages as heartbeats to the group members of AMF setting with a high frequency (100 *ms*). When the "active" SIP dispatcher fails, the AMF would choose one of the backup (standby) SIP dispatchers to be the "active" SIP dispatcher and inform it to claim that it owns the virtual IP (SAP).

In order to monitor multiple SIP proxy servers and to achieve load balancing, the OpenAIS checkpoint service is used. Each SIP proxy server uses OpenAIS checkpoint service to keep sending a counting number as a heartbeat to SIP dispatchers. The SIP dispatchers would check the number every 100 *ms*. If the number did not increase as expected, the SIP dispatchers would assume that the associated SIP proxy server is failed and stop forwarding SIP messages to it.

The "token" parameter in the AMF can be adjusted to reduce the failover time in our design. It does not cause apparent increase of control messages because it is a timer to decide whether the monitored server is dead if no heartbeat has been received within the "token" period. Adjusting the token parameter value will not change the generating frequency of heartbeats.

4. Evaluation and discussion

SIPp [5] was used as a SIP traffic generator and analyzer in the performance evaluation. The usages of SIPp include generating SIP requests and responses, checking the operation with implemented scenarios, and reporting testing results. The default UAC (user agent client) and UAS (user agent server) scenarios, which have already been implemented in SIPp, were used to test the proposed SIP high availability network architecture. Fig. 4 shows the evaluation environment for FFF. It contains three SIP dispatchers with 2 + 1 redundancy model and three SIP proxy servers with 3 + 0 (all active) redundancy model. SIPp UAC and UAS were placed at different sides of the proposed SIP network to test the functions and failover operations of dispatchers and proxy servers. All SIP servers (dispatchers and proxy servers) were implemented by OpenSER [6] on Linux operating systems.

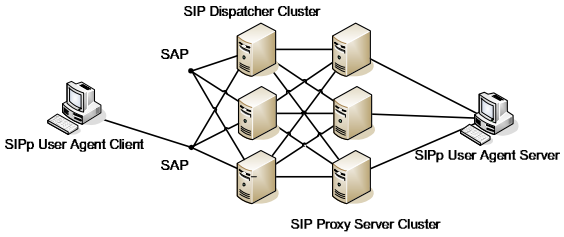

Fig. 4 . Evaluation environment for FFF.

All SIP servers are running on the virtual machines of one PC. The SIPp UAC and UAS were running on the other machine. The test environment parameter settings are shown in Table 1. The default setting of OpenAIS AMF "token" value (1000 *ms*) and a smaller value (300 *ms*) were used to test for failover time. 300 *ms* is the smallest value allowed in the OpenAIS configuration.

Table 1. Test environment parameter settings.

Environment parameter	**value**
CPU (for virtual machines)	Pentium 4 3.0 G
Memory (for virtual machines)	3 Gigabytes
CPU (for SIPp)	Pentium 4 m 1.8 G
Memory (for SIPp)	512 Megabytes
Call rate (test for failover time) in SIPp	10 calls/sec (default in SIPp)
Call limit	10000 calls
Token (OpenAIS AMF)	300, 1000 (default setting in AMF) *ms*

In the test of failover time, we induced a service failure by power down the active SIP dispatcher. We compared the SIP dispatcher failover time of ours with that of VRRP, because VRRP is a common used failover mechanism by IP takeover. The VRRP was setup using an open source demon VRRPd [7] with default settings, and the same test environment as shown in Fig. 4 was adopted.

The effect of CPU loading on failure time is first evaluated. Low CPU loading is defined as no extra loading of other traffic. Running a heavy CPU program in the background is to simulate the high CPU loading condition. Fig. 5 shows the effect of CPU loading on failover time. The FFF scheme had much shorter failover time than VRRPd and using a smaller value (300 *ms*) of the token parameter could reduce the failover time by half compared to using the default value (1000 *ms*). Under the high CPU loading condition, the failover time of the proposed FFF was not affected too much (0.651 vs. 0.769) by the increase of CPU loading compare to that of VRRP. The results also show that the FFF scheme reduces the dispatcher

failover time by 80% (0.651 vs. 3.163) compared to VRRP.

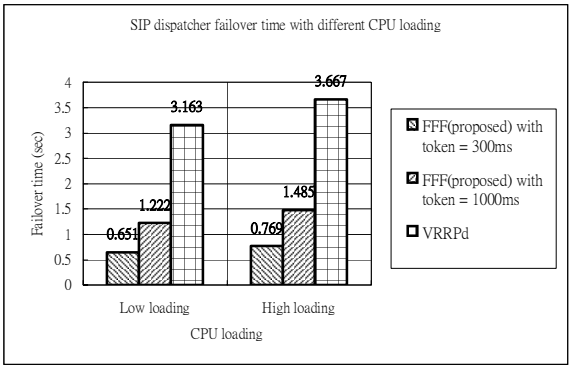

Fig. 5. Effect of CPU loading on failover time.

We turned off SIPp's retransmission mechanism at the UAC side in order to compute the number of failed calls caused by the SIP dispatcher failover under different call rates. Fig. 6 shows failed calls caused by a dispatcher failure under different call rates. This result shows that the FFF scheme would have fewer failed calls compared to the VRRPd scheme when a SIP dispatcher failed, and the number of failed calls is related to the failover time.

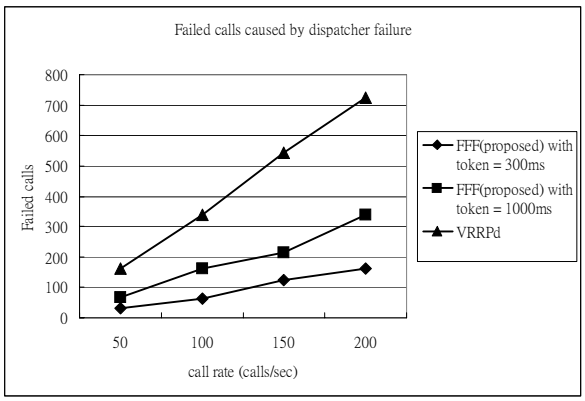

Fig. 6. Failed calls caused by dispatcher failure under different call rates.

The OpenSER with default settings was used to compare with the FFF scheme because the OpenSER itself has a SIP proxy failover scheme. Fig. 7 shows the SIP proxy server failover time comparison. It shows the proposed FFF scheme can detect a failure and react much faster than OpenSER when a SIP proxy server failed. That is, using the FFF scheme, the SIP client will encounter a smaller probability of service errors during the SIP proxy server failover time. The results show that the FFF scheme shortens the proxy failover time by 85% (3.89 vs. 24.79) compared to the mechanism provided by OpenSER itself.

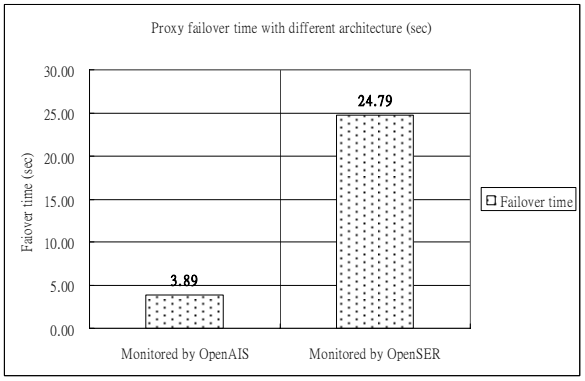

Fig. 7. SIP proxy server failover time comparison.

5. Conclusion

In this paper, we have presented a fast failure detection and failover (FFF) scheme that uses multiple SIP dispatchers and multiple SIP proxy servers to achieve high availability. The OpenAIS is used as a high availability middleware to handle the failure detection and failover among SIP dispatchers and SIP proxy servers. Experimental results have shown that the FFF scheme reduces the SIP dispatcher failover time by 80% compared to the VRRP solution and also shortens the SIP proxy server failover time by 85% compared to the mechanism provided by OpenSER itself.

6. References

[1] OpenAIS at Open Source Development Labs, Beaverton, OR, USA. [Online]. Available: http://developer.osdl.org/dev/openais/.

[2] F5 Networks Inc., "A New Paradigm for SIP High Availability and Reliability," [Online]. Available: http://www.f5.com/solutions/technology/pdfs/sip_wp.pdf.

[3] Cisco Inc., "Overview of High Availability in SIP-based Voice Networks," [Online]. Available: http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/callc_c/sip_c/sipha_c/hachap1.htm.

[4] S. Knight, D. Weaver, D. Whipple, R. Hinden, D. Mitzel, P. Hunt, P. Higginson, M. Shand, and A. Lindem. RFC 2338: Virtual Router Redundancy Protocol, April 1998.

[5] "SIPp - Test Tool/Traffic Generator for the SIP Protocol," March 2006, [Online]. Available: http://sipp.sourceforge.net.

[6]"OpenSER," [Online]. Available: http://www.openser.org/.

[7] "VRRPD," [Online]. Available: http://off.net/~jme/vrrpd/.