



Introduction

Obtaining a reliable, fault tolerant, highly available system with ASTERISK® has been difficult and nearly impossible especially from the PRI/TDM standpoint. IP based toolsets have been around for a number of years to allow integrators the ability to provide fault-tolerance on the VoIP side of their implementation. Until today however, providing quick, reliable and robust T1/E1 fail-over has been nearly non-existent. The foneBRIDGE fills this void allowing IT Managers and Systems Integrators to implement Asterisk in the most demanding of environments where downtime is a luxury a business simply can not afford.

High Availability

A number of open and commercial tool sets are available for building high-available fault tolerant Linux clusters. The general objective of deploying HA (High-Availability) in a server environment is to provide reliability, availability, and serviceability (RAS). In the traditional Asterisk implementation model, T1/E1 connectivity is provided by PCI based interface cards on a single server. In this scenario the systems overall integrity and availability is limited by the reliability of that single server. For many businesses, taking this gamble on a mission critical function such as voice is unacceptable.

As an externalized point of T1/E1 termination, the foneBRIDGE is decoupled from the single server limitation and when combined with the proper tool set can be implemented to provide rapid and automatic failover amongst a cluster of servers.

The most common tool set used to provide this functionality when paired with the foneBRIDGE and Asterisk is the 'heartbeat' tool from the Linux-HA project www.linux-ha.org. As the name employs, 'heartbeat' monitors the state of 2 or more nodes (servers) in an Asterisk cluster. If it detects a fault, for example when a primary node stops responding to the heartbeat messages it executes scripts that allow it to quickly failover to a secondary server. All of this is done transparently and without user intervention.

Rapid Recovery

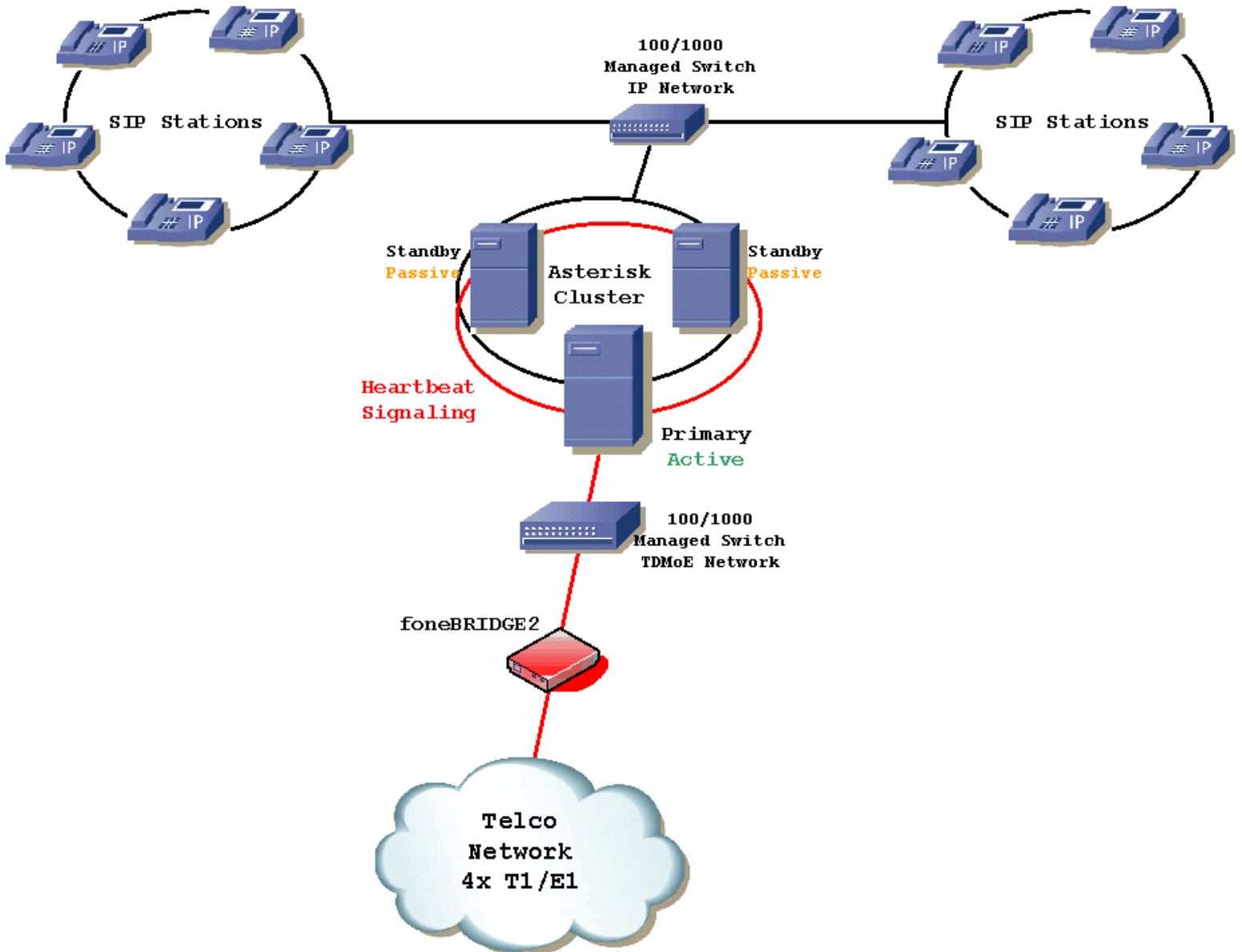
With the foneBRIDGE, rapid failover and system recovery can occur in seconds, not minutes or hours. Through the foneBRIDGE's rapid re-configuration options, we can program it on the fly through high availability tools to begin routing the TDM stream and calls to a secondary standby server in under one second. Add to that the short amount of time it takes the server to start Asterisk and clear circuit alarms and your stand-by server can be operational and handling calls in under 3 seconds, all transparently and without admin intervention.

Uptime During Maintenance and Upgrades

Over are the days when it was necessary to bring down a server for patching or software upgrades. With foneBRIDGE, the server admin can manually failover to a back-up server and maintain the voice system operational while performing upgrades or patching on the primary server. This unique feature also lends itself nicely to the ability to failover to a server where production testing can be done on new code releases or features and in the event of a bug discovery or issue that may jeopardize the integrity of operations the system admin can easily fail back to a primary server with only seconds of downtime.

Scalability

With connectivity externalized, and placed on the Ethernet not tied to any PCI card or single server, adding additional T1/E1 capacity to an Asterisk implementation is simplified and can be realized with little or no downtime to operations.



Sample Asterisk server cluster running Linux High Availability heartbeat with foneBRIDGE2.



HA Best Practices

- Test your dial-plan, PBX functionality, system performance and overall call quality before going into production. Try to emulate the production environment by building a second Asterisk server to act as a T1/E1 telco and use it to generate and receive calls coming from your production machines. Stress the machine as well by trying to determine what its system resource thresholds are. Knowing what your setup is capable of before going into production will help you avoid any “gotchas” as the system grows through the addition of new users or T1/E1 lines.
- Standardize your hardware platform– This goes hand in hand with the point above. Once you’ve gone through the trouble of benchmarking the capabilities and shortcomings of your server and related hardware and have settled on a production quality working environment, document everything that was performed to get to that point. Make note of BIOS settings, kernel boot options, application versions, etc. If you intend on rolling out more of the same system consider forecasting your hardware needs and investing in a small stockpile of the hardware you will need. The server hardware market is a constant moving target. The Dell, HP, IBM, etc. you bought today will most certainly not be the same tomorrow irregardless of the model type, system specs, etc. If you decide to follow the upgrade path of your server manufacturer then it is advisable to do benchmark testing all over.
- Rotate your servers in and out of operation on a daily basis. If working with a simple two system cluster running in an Active/Passive mode force failover each night when convenient to ensure both servers are still operational. This will help you identify any potential hardware issues before actually needing to failover during a hardware fault on the primary. This also allows you to better utilize your investment in your server hardware.
- Monitor your systems pro-actively to preempt potential issues. Cacti, Nagios, Zenoss, monit are all excellent open source monitoring tools. Asterisk-1.4x now includes snmp support to allow for a more granular perspective for monitoring the Asterisk application itself.
- Cluster replication/synchronization tools– DRBD provides comprehensive disk mirroring on a cluster of servers. Rsync, csync2 are easier to manage and more convenient if the goal is to simply mirror a few directories and not entire partitions.



Redfone Communications, LLC
9941 West Jessamine Street
Miami, FL 33157 USA
www.red-fone.com
(786)544-1196