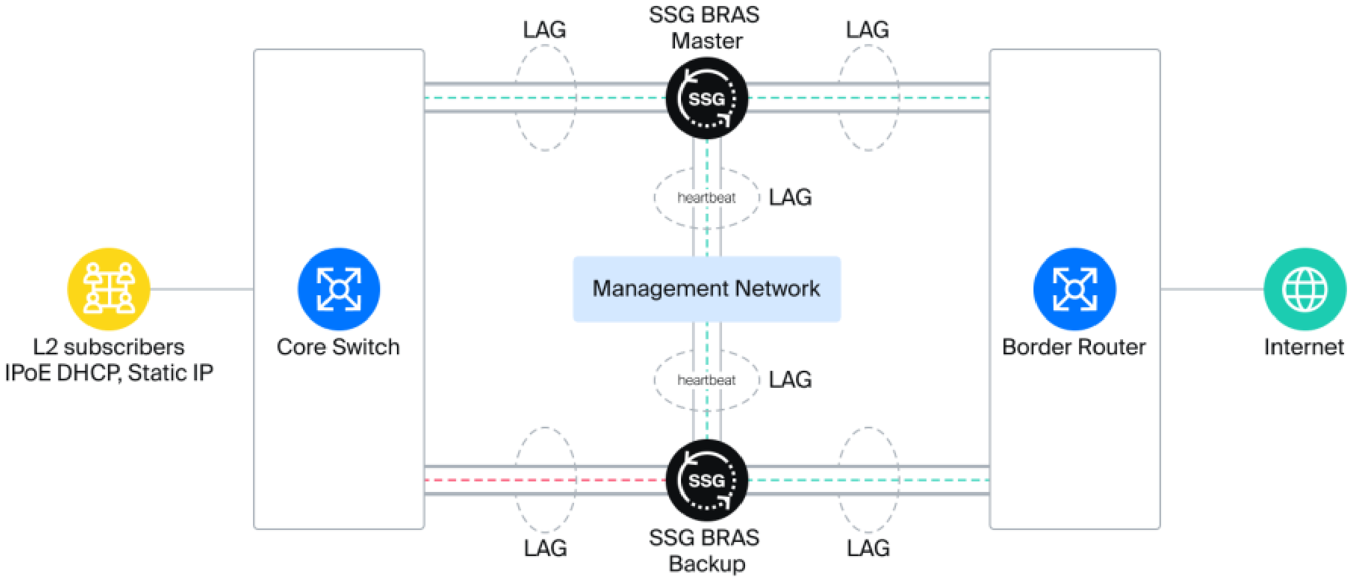
Table of Contents

BRAS Active-Standby (Master-Backup) redundancy

Description of the switching algorithm for BRAS L2 (DHCP, Static IP)

- BRAS L2 redundancy for IPoE (DHCP and Static IP) is recommended using the Active-Standby scheme, which involves connecting two SSG BRAS devices to a single broadcast L2 domain: one in Master mode, the other in Backup mode (hot standby).
- The Master is the active server and processes traffic during normal network operation. The Backup is in standby state and does not pass traffic through itself; the DPDK interfaces towards subscribers (IN ports) are administratively shut down (down).
- The Backup server monitors the operation of the Master server using a script (heartbeat) via dedicated management ports. Upon detecting a failure of the Master server, the Backup server automatically activates (up) the DPDK interfaces towards subscribers (IN ports) and begins processing traffic.
- Single traffic switchover to the Backup server and stopping the Master server is implemented to avoid multiple traffic transfers and network impact. Switching traffic back to the Master is performed by the network administrator manually.
- For correct operation, all service profiles must be identically configured on both the Master and Backup servers; we recommend using a profile synchronization script.
- Note that SSG BRAS supports dynamic (OSPF, BGP) and static routing. In case of dynamic routing, for Static IP subscribers with public IP addresses, the announcement will change automatically when switching to the Backup server; for subscribers with private addresses, a NAT profile will be applied, under the same name, but from a different public address pool configured on the Backup server.



Master server status monitoring script

The script must be installed on the Backup server, where it runs in a continuous loop, monitoring the state of the Master server via SSH.
Four checks are used to confirm the normal operation of the Master server:

1. Server is reachable over the network (pingcheck)
2. The fastDPI process is present
3. The PID of the fastDPI process has not changed (no uncontrolled process restart)
4. The link state on the main fastDPI has not changed (optional check). This check is disabled by default, as it may not be needed in some topologies

Script Installation Process:

1. Download all files from the archive to the target backup server
2. Configure the Master server's IP address in the `SRS.sh` script
3. Create an SSH key pair on the Backup server using the command

   ```
   ssh-keygen -t ed25519
   ```

4. Create a new user with sudo rights on the Master server
5. Copy the private SSH keys from the Backup server to the `authorized_keys` file of the new account on the Master server
6. Add execute permissions to the installation script using the command

   ```
   chmod +x install.sh
   ```

7. Run

   ```
   install.sh
   ```

Service Management:

1. Start the service:

   ```
   systemctl start fastsrs
   ```

2. Check service status:

   ```
   systemctl status fastsrs
   ```

3. Stop the service:

   ```
   systemctl stop fastsrs
   ```

4. Check service logs:

   ```
   journalctl -u fastsrs
   ```

Service profile synchronization script

The script synchronizes service profiles 4 (blacklist filtering), 5 (whitelist and Captive Portal), 18 (session policing and traffic class override) and policing between Master and Backup servers.
The script runs on the Master server; service profiles on the Backup server will be aligned with those on the Master server. Profile transfer is performed using `fdpi_ctrl` commands and remote SSH access.

System Requirements:

- SSH
- Bash
- Jq
- Installed SSG
- Rsync

Script Logic:
The script retrieves the current service profile from the Master server and then sends it to the specified Backup server. Then the script connects to the Backup server and retrieves data for profiles present on the Master server, retrieves the profile data on the Backup server, compares them, and deletes profiles missing on the Master server.

Installation and management

1. Configure certificate authentication: create a certificate on the Master server using `ssh-keygen -t ed25519`; using the root account for authentication is easiest.
2. Download the script to the Master server and place it in the `/usr/local/bin/` directory
3. Add permissions for the script using the command

   ```
   chmod +x /usr/local/bin/profile_sync.sh
   ```

4. Configure the user and IP of the Backup server within the script. The user must have write access to the `/etc/dpi` directory; the simplest option is to use the root user. Another user with appropriate rights can also be configured.
5. Configure cron to run the script at desired intervals **(optional)**:

   ```
   crontab -u root -e
   0 * * * * /bin/bash /usr/local/bin/profile_sync.sh
   ```

6. Add a bash alias to run the script on demand:

   ```
   echo "alias dpi_sync='/bin/bash /usr/local/bin/profile_sync.sh'">>
   ~/.bashrc
   ```

7. Create the directory `/etc/dpi/service18` and save all service 18 files in it.

Script Operation:
The script is run by crontab at specified intervals or manually using the `dpi_sync` command.

Note that if a service profile is applied to a subscriber, it will not be deleted. Also note that any files not saved in the service18 folder will not be transferred to the Backup server, and thus the synchronized service profile 18 will not work. If the alias dpi_sync is absent, the script should be run via sudo bash /usr/local/bin/profile_sync.sh.