High Availability for the design of stateful networking equipments

P. Neira, Laurent Lefèvre, R. M. Gasca

QUIVI R Research Labs - University of Sevilla, Spain
INRIA RESO - LIP Labo (UMR CNRS, INRIA, ENS, UCB)
Outlines

- Starting point
- Problems & Limitations
- Example
- Our Contribution: Architecture
- Scenarios
- Conclusions
- Future Works
Starting point

- Networking equipments like routers, firewalls, proxies... introduce Single Point of Failures (SPOF) in the network schema

- High Availability (HA) protocols: redundancy + health checking. They ensure continued working service
Starting point: HA basics
Problems & Limitations

- Current HA protocols are static: don’t cover specific aspects of the running services.
- The dynamic nature of some services deployed on critical machines makes this approach insufficient.
Example

- **Linux Connection Tracking System**: subsystem that keeps in memory information about the state of the connections going through the firewall in order to perform stateful filtering
Example

TCP 35887 -> 22 (SSH)
Example (2)

TCP 35887 -> 22 (SSH) ESTABLISHED
Example (3)
Our contribution

- We define: Architecture to ensure the availability of stateful networking equipments
- We provide: Add-on library (SNE) to extend current HA protocols: it covers the dynamic nature of the deployed services (https://svn.netfilter.org/)
Event Notifier: sends events that can be listened by the SNE library
Architecture
Events

- NEW: it contains information about a new connection that has been established.
- UPDATE: any critical information about a connection have changed.
- DESTROY: this event occurs if a connection is closed.
Level of event notification

- strong: notifies every update in current connections
- weak: only notifies crucial messages (new, destroy)
- incremental: transfer the messages periodically
Actions

- Set of actions provided by the SNE library (sockets):
  - Create, update and destroy entries that represents a given operation in the current service
  - Restart the service: resynchronization
  - Listen to events
Scenarios

- We solve a set of problematic scenarios:
  - Primary fails
  - Backup node fails and it comes back to life again
  - Backup node fails and remains dead
  - Former primary node comes back to life
  - Backup lost communication with Primary
Conclusions

- HA protocols ensure continued working services but don’t guarantee the success of current operations.
- We provide an architecture to extend HA protocols and solve the current limitations.
- Experimentation and Evaluation.
Future work

- Integration with intelligent equipments: programmable switch, active nodes
- Synchronization protocol
- Active-active settings: resources are scarce
- Scenarios with multiple failures
Questions

Thanks for your attention