KARAN BAVISHI

kbavishi@cs.wisc.edu | karan.bavishi90@gmail.com | +91-9620-418-397
Karan Bavishi | kbavishi

EDUCATION

M.S. Computer Science | UNIVERSITY OF WISCONSIN-MADISON
2016 - Ongoing | GPA: 0.0/4.0

B.E. (Hons.) Computer Science | BITS - PILANI, PILANI CAMPUS
2008 - 2012 | CGPA : 8.08/10.0

PROGRAMMING SKILLS

C • C++ • Python • Bash
Java • Go

COMPETITIVE PROGRAMMING

Project Euler (PE)

SOLVED: 239/532
INDIA RANK : 11 * | WORLD RANK : 918 *
* PE account is needed to access rank links.

WORK EXPERIENCE

ARISTA NETWORKS | SOFTWARE ENGINEER | JUL 2012 - JUL 2016

◇ High-availability for Licensing in Extensible Operating System (EOS) — Sep 2015 - Jul 2016

Adding mechanisms to support license state replication which ensures high-availability for the Licensing system in EOS. This helps customers to reliably deploy license management

- Added mechanisms using *inotify* and *rsync* to support directory-level replication of the license files across multiple nodes in the datacenter
- Added mechanisms using clock timers and file locks to support directory snapshot replication. This made the design more robust by avoiding partial directory replication, and thus not lose licenses because of inconsistencies

◇ Licensing in Extensible Operating System (EOS) — Nov 2014 - Sep 2015

Software licensing system in Arista's EOS to check that a switch is allowed to use the advanced software features present

- Helped design a complex licensing system to replace the earlier honor system with no license enforcement, where the design had to reliably work for several use-case scenarios - customers with large datacenters, customers with 'dark site' switches, and customers wanting to bypass
- Worked on parts of both the client and server modules of the system. Integrated license awareness in more than 15 features in several different areas, including routing protocols (eg. OSPF, BGP, ISIS), virtualization features (eg. VXLAN, VM Tracer), and SDN features (eg. OpenFlow)
- Developed an extensive testing framework for the license libraries, to simulate license distribution in unit tests and avoid interacting with the production servers

◇ Control plane policing (CoPP) for CloudVision Exchange (CVX) traffic — Jul 2014 - Oct 2014

CVX is Arista's SDN solution which allows network management via a controller. Added policing rules for traffic flowing between a controller and the managed switches, to prevent throttling and starvation

- Added hardware rules in multiple switch platforms for trapping the CVX traffic to the CPU, and classifying it to a dedicated traffic-class, to protect it from aggressive shaping
- Also added dynamic QoS rules for the CVX traffic to protect it from starvation by other non-priority traffic
- This work reduced the convergence time for control-plane services like VXLAN with hundreds of managed devices by nearly 40%

◇ Audio Video Bridging (AVB) — Jul 2013 - Jun 2014

AVB is a set of IEEE standards defined for delivering Audio/Video(AV) traffic across a LAN with guaranteed bandwidth, predictable low latency and correct time synchronization

- Worked on several areas of Multiple Streams Registration Protocol (MSRP), a link layer signaling protocol for end-to-end resource reservation, fault detection, bandwidth reservation using QoS, and interaction with Spanning Tree Protocol (STP) to allow rapid recovery from link failures.
- Single-handedly implemented and tested Multiple VLAN Registration Protocol (MVRP), for dynamically provisioning VLANs in networks
- Added multicast MAC address support for the Arad (7500E) platform, which is critical for transmission of data packets from one source to multiple destinations.
- Added mechanisms for delayed buffer writing to avoid buffer overflows in the sockets used for synchronizing MSRP process state to *SysDB*. This improved the control-plane handling capacity from 700 to about 2000 audio/video streams
- Added mechanisms for delayed state updates immediately on the new active supervisor after a failover. This improved the recovery time for streams to resume transmission after software / hardware failures, from 30 seconds to 1 second
- Developed an AVB-capable endpoint simulation using the open-source Open-AVB code, which eliminated the need for purchasing expensive AVB-capable hardware endpoints

◇ Simple Network Management Protocol (SNMP) Management MIBs — Jan 2013 - Jul 2013

ARISTA-REDUNDANCY-MIB provides information about redundancy information on modular systems with dual supervisor engines. ARISTA-QOS-MIB provides information about QoS configuration and packet counters on Arista devices

- Wrote ARISTA-REDUNDANCY-MIB, which makes redundancy information about modular switches available through SNMP. It also allows for the generation of an SNMP trap for supervisor failovers
- Wrote the ARISTA-QOS-MIB, which makes QoS configuration and packet counters available through SNMP
- Added support for the abovementioned MIBs by exposing relevant state to the SNMP server running on the switch. This allows the server to respond to SNMP queries from the client

---

◇ Redundancy support for Link Layer Discovery Protocol (LLDP) in Stateful Switchover (SSO) mode — Sep 2012 - Dec 2012

Added support for the LLDP process to run on the secondary switch supervisor in 'standby' role, to reduce downtime during failures

- Added support for the LLDP process to be able to run on the secondary supervisor in a standby role, ready to transition to the active role in case of a failure on the primary supervisor.
- Also refactored code to ensure that LLDP state isn't unnecessarily deleted, then re-added. This would have caused unnecessary churn because state is constantly being replicated from the active to standby supervisors.
- This reduced the LLDP downtime from 2 mins to 30 secs, and eliminated the generation of spurious SNMP traps about LLDP neighbor being down

---

◇ Switch reboot cause enhancements — Jul 2012 - Sep 2012

Enhanced the infrastructure for displaying the cause for why a switch was rebooted for several scenarios

- Enhanced the infrastructure to cover several scenarios where no reboot reason was being displayed. This allowed us to distinguish between reboots caused due to potentially serious issues and reboots caused due to user actions.
- Added mechanisms to create a reload reason file, and *sync* to disk before the switch got rebooted so that it wasn't lost

---

ARISTA NETWORKS | SOFTWARE INTERN | JUL 2011 - DEC 2011

◇ Userspace simulation of a kernelspace election module — Oct 2011 - Dec 2011

A userspace simulation of the kernelspace election module, which was responsible for determining the active and standby supervisors in a modular switch

- Wrote a Python userspace simulation of a kernelspace election module, for use in unit tests. This simulation helped reduce the time needed to test functionality, by allowing unit tests instead of tests with an actual modular switch which took several minutes

---

◇ Rate-limiting of syslogs and other improvements — Jul 2011 - Sep 2011

Adding support for rate-limiting of frequently occurring syslogs and other enhancements.

- Implemented support for rate-limiting of syslogs, which helped prevent diskspace exhaustion in */var/log* because of frequently occurring syslogs
- Discovered and fixed some issues with the Python syslog infrastructure. This helped improve performance by about 50%.

ACADEMIC PROJECTS

Peer to peer file synchronization system in Linux | Link

- Users have a shared folder across different machines, with local copies, in which any changes made are synced across all devices
- Linux inotify API used to track changes in the shared folder and rsync used to sync the modifications to ensure efficient transfer
- Multithreading with mutexes used to parallelize syncing and file-monitoring operations

---

Desktop Chat Application

- Developed client and server modules of the application which ran over TCP. It was capable of providing video and text services
- Also designed a GUI in C#, integrating all the services being provided

---

OTHER PROJECTS

Parallelized diff tool using Go

- Implemented a parallelized *diff* tool in Go, to generate faster diffs between two branches than the default tool
- Improved performance by 80 times, bringing down the running time from several minutes to a few seconds
- Won the 'Most Useful Tool' in the Arista Hackathon held in Oct 2015
