SAVBench: A Benchmarking Framework for Source Address Validation Based on SAV Open Playground (SAVOP)

draft-chen-bmwg-savnet-sav-benchmarking-02

Libin Liu

Beijing Zhongguancun Laboratory

BMWG Meeting, IETF 122

March 17, 2025

Background

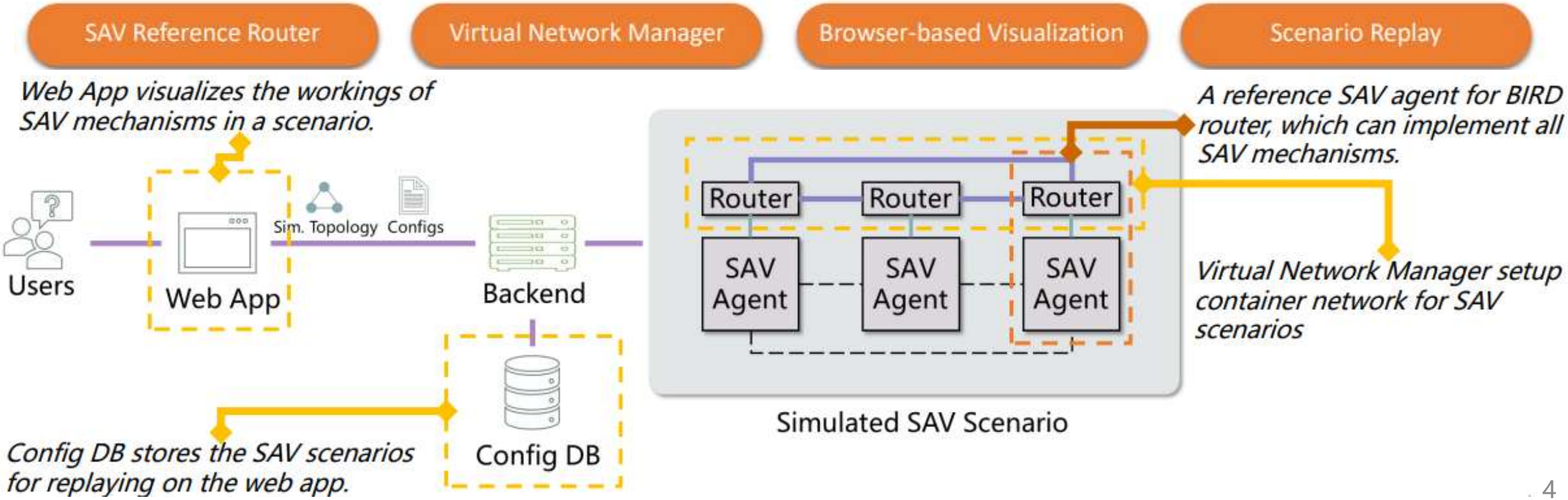
- This document defines the methodologies for benchmarking the performance of source address validation (SAV) mechanisms
  - Taking the approach of considering a SAV device to be a black box and defining the methodology in a manner that is agnostic to the mechanisms.
  - Providing a method for measuring the performance of existing and new SAV implementations.
- Historical versions
  - **draft-chen-bmwg-savnet-sav-benchmarking-00, IETF 120 SAVNET WG and BMWG**
  - draft-chen-bmwg-savnet-sav-benchmarking-01, August 7, 2024
  - **draft-chen-bmwg-savnet-sav-benchmarking-02, IETF 121 SAVNET WG and BMWG**
  - **draft-chen-bmwg-savnet-sav-benchmarking-03, IETF 122 BMWG**

SAVBench

- A framework for evaluating the performance of existing SAV mechanisms using the benchmarking methodology proposed in the document [draft-chen-bmwg-savnet-sav-benchmarking].
  - For operators, it replicates production-grade network environments for SAV mechanism evaluation, enabling operators to identify deployment-optimized solutions through scenario-based performance benchmarking.
  - For vendors, it facilitates large-scale topology construction with hybrid software/hardware components, allowing comprehensive validation of SAV implementations.

SAVOP

- SAVOP is an open platform designed for the implementation and emulation of various SAV mechanisms.
- SAVOP is open-source: https://github.com/SAV-Open-Playground
- Four core components:



Emulation Setups

Testbed

- Using a x86 server machine with two 2.2GHz 26-core Intel Xeon Gold 5320 CPUs, 256GB DDR4 RAM, 2 1TB SSDs, and 1 12TB SAS HDDs
- Running Ubuntu 22.04.2 LTS with kernel version 5.15.0
- Using Docker 24.0.2 with the image ubuntu:22.04 for each container to emulate an AS
- Running BIRD 2.0.12 as the AS border router and using iptables 1.8.7 to filter packets

Methodology

- Evaluating the performance of these mechanisms in terms of **validation accuracy, control plane performance, data plane performance, and scalability**
- Using the network topology with 50 ASes
  - Except for the scalability experiments
- Varying the deployment ratios of the SAV mechanisms from 10% to 100%

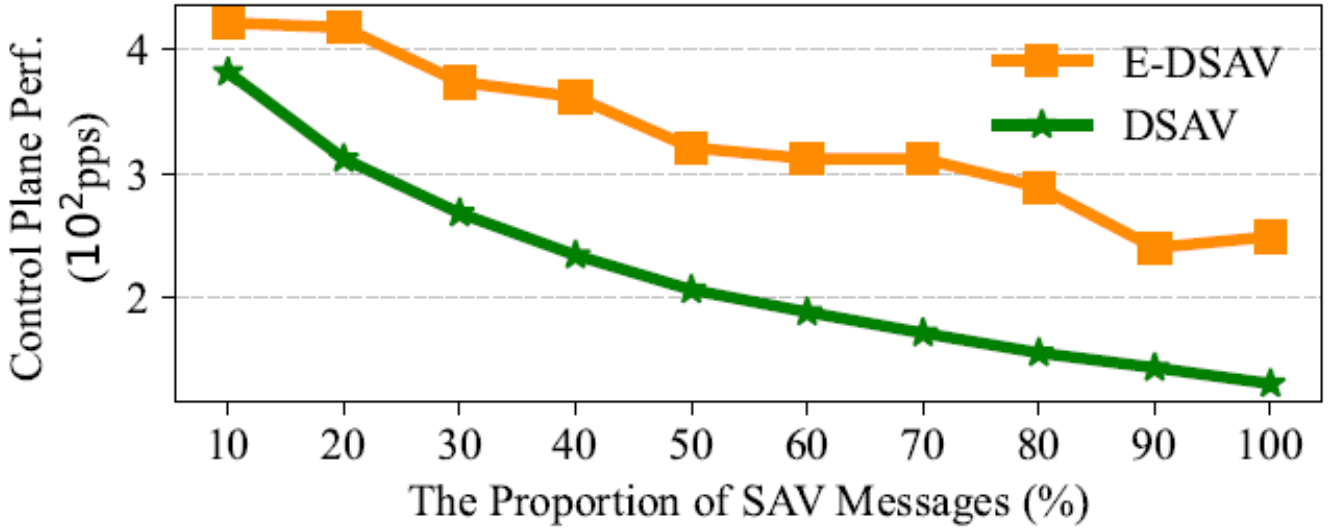
SAV Accuracy

Scenarios	Loose uRPF	Strict uRPF	FP-uRPF	EFP-uRPF-A	EFP-uRPF-B	BAR-SAV	Passport	DSAV	E-DSAV
Symmetric Routing	IP	√	√	√	IP	√	√	√	√
NO-EXPORT	IP	IB	IB	IB	IP	IB	√	√	√
DSR	IP	IB	IB	IB	IP & IB	IB	√	√	√

The SAV accuracy of different SAV mechanisms implemented on top of SAVOP in the scenarios including symmetric routing, NO-EXPORT, and Direct Server Return (DSR) (√: Accurate Validation, IP: Improper Permit, IB: Improper Block).

- The table shows SAV accuracy results of different SAV mechanisms in the three scenarios.
  - In symmetric routing scenario, both Loose uRPF and EFP-uRPF with algorithm B may improperly permit spoofing traffic.
  - In NO-EXPORT and DSR scenarios, both Loose uRPF and EFP-uRPF with algorithm B may improperly permit spoofing traffic; Strict uRPF, FP-uRPF, EFP-uRPF with algorithm A and B, and BAR-SAV may improperly block legitimate traffic.
  - The results are the same as the theoretical analysis in [draft-ietf-inter-domain-problem-statement].

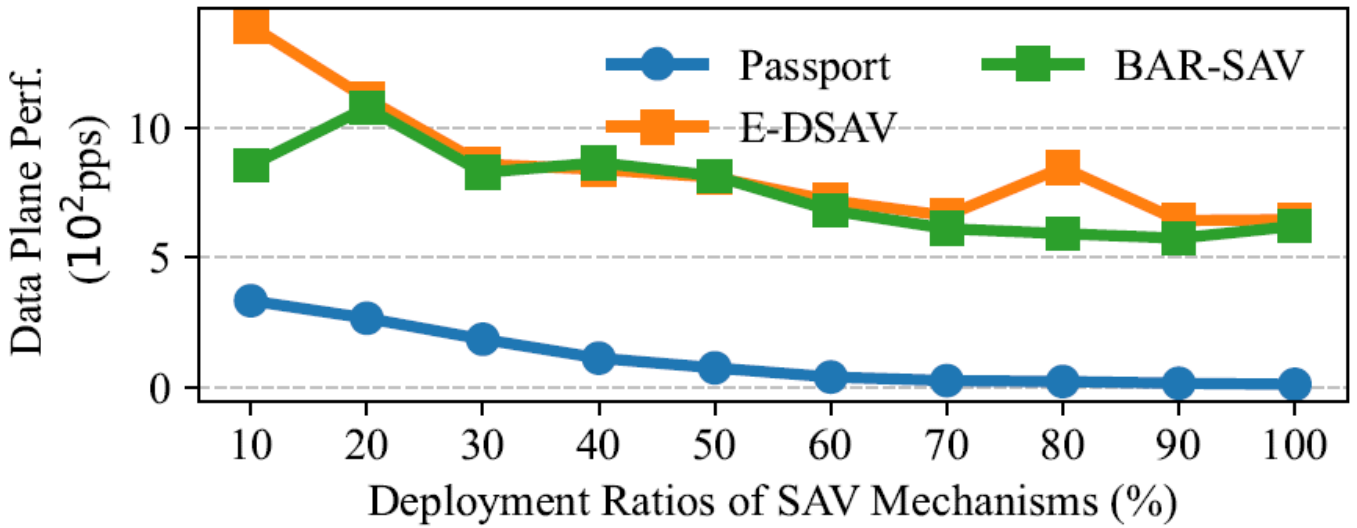
Control Plane Performance



The control plane performance for processing pure BGP messages in terms of packets per second with varying proportions of SAV messages. The proportions of SAV messages are calculated by the number of SAV messages over the total number of messages for SAV and pure BGP.

- The figure shows the control plane performance when processing pure BGP messages under varying proportions of SAV messages.
  - Both DSAV and E-DSAV impact the efficiency of the control plane in dealing with pure BGP messages. However, it is essential to note that the underlying reasons for these effects differ between the two approaches.
  - For E-DSAV, the limitations arise from resource constraints within each container.
  - Instead, DSAV not only needs to communicates more messages but also necessitates additional resources for parsing the delivered SAV messages.

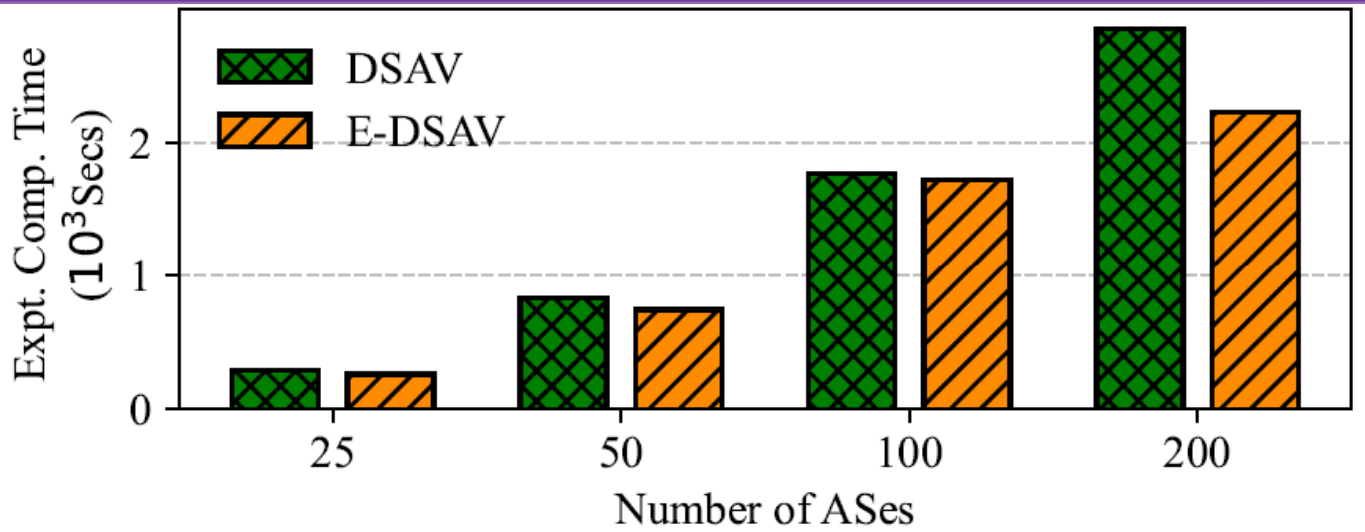
Data Plane Performance



The data plane forwarding performance of the SAV mechanisms with varying deployment ratios. Here, we employ iptables to execute SAV within the data plane. We implement a traffic generation tool to generate packets with fixed 1.5KB to evaluate the data plane forwarding performance in terms of packets per second.

- The figure shows the results of BAR-SAV, Passport, and E-DSAV.
  - Passport always has a more significant impact on data plane forwarding performance than other SAV mechanisms. It is because cryptographical SAV requires the router to perform cryptographic computation on each packet, which increases the processing overhead.
  - The data plane forwarding performance of each SAV mechanism decreases as the deployment ratio increases. This is because the size of the SAV table within each AS increases with the increase of deployment ratio, larger SAV table results in longer query time for each incoming packet.

Scalability of SAVOP



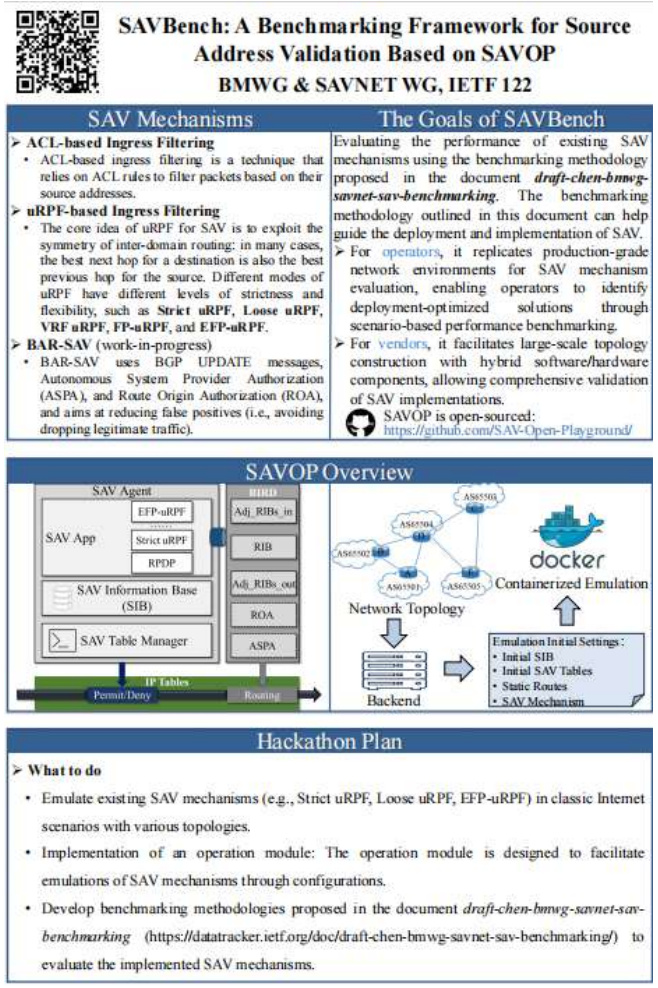
The experiment completion time of SAVOP across different network scales. We vary the network scales by increasing the number of ASes for the testbed experiments, and then calculate the experiment completion time. The experiment completion time is the longest time elapsed from launching the Docker environment to generating complete SAV Table among all ASes.

- The figure shows the total experiment time of SAVOP with AS numbers from 25 to 200, by taking DSAV and E-DSAV as examples.
  - SAVOP can run E-DSAV on a 200-AS network topology and complete the experiments within ~47 minutes.
  - Both the experiment completion times for DSAV and E-DSAV increase along with the increase of AS numbers.
  - Compared with DSAV, E-DSAV shows a slower growth trend with the increase of network size. This is because E-DSAV converges faster than DSAV.

Next Step

- Seek feedback and comments.
- Collaborations are welcome.

Thanks! ☺



IETF 122 Hackathon Poster

SAVOP is open-source:
https://github.com/SAV-Open-Playground

Benchmarking Topologies and Scenarios

- Real-world AS-level network topology
  - Using real BGP data from public route collectors provided by RouteViews[1] and RIPE RIS[2]
  - Parsing and extracting *AS path* attribute from the BGP data and obtaining neighboring relation between ASes
  - Creating links for the neighboring ASes to build the AS-level Internet topology
  - Obtaining the business relationship between ASes according to the data from CAIDA[3]
- Sub-graphs generated based on the full topology
  - A connected component of the full topology
  - Assigning routing policies based on the business relationship and the valley-free principle
- Three classic scenarios
  - Symmetric routing, NO_EXPORT, direct server return (DSR)

[1]http://www.routeviews.org/routeviews/
[2]https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data
[3]https://catalog.caida.org/dataset/as_relationships_serial_1