Insights into TLS performance: Evaluating OpenSSL 1.1.1 through 3.4 in Firewall Deployments

William Bellingrath
Software Engineer Staff

JUNIPER NETWORKS

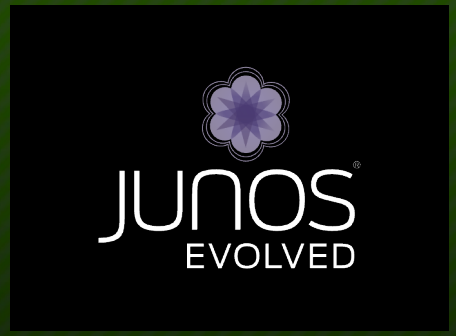
Agenda

1. Introduction
   - Project & team background
   - Firewall platform overview
   - Upgrade OS complexity
2. Upgrade & Performance Insights
   - Evaluation of 3.0, 3.1 & 3.4
   - Performance & mitigation strategies
3. Future Work
   - Next steps
   - Acknowledgements

Introduction

Juniper Networks

- Products run on customized operating systems with distinct cryptographic and FIPS strategies
- Junos OS Evolved
  - Linux based OS
  - Internal OpenSSL 3.5 upgrade underway, no issues
  - Uses the modular FIPS validation approach
  - Narrower product base: no SRX Firewall, few routers and switches



- Junos OS
  - FreeBSD-based OS running on a broader product base, including the SRX
  - Supports both hardware SRX and vSRX (cloud-based) deployments
  - Currently shipping with OpenSSL 1.1.1 w/ Support Contract
  - FIPS validations cover chassis level boundary, adding complexity

Who we are

- Cybersecurity R&D team
  - Juniper key management, Secure Development Lifecycle, developers
  - Platform hardening across Routers, Switches and Firewalls
  - Responsible for centralized security services, OpenSSL, OpenSSH, FIPS validations, etc.
  - Implements TPMs, Device ID, IMA, Secure Boot, all to enhance system security
- Security Platform team
  - Develops and maintains SRX Firewall platforms
  - Implements Data Plane SSL-Proxy and other inline cryptographic acceleration features
  - Relies on integration of OpenSSL for performance and FIPS compliance
  - Optimized for edge, cloud and datacenter deployments with high throughput demands

Why does this matter?

- SRX customers are performance sensitive

- SRX firewall product priorities
  - SRX customers demand high performance in real-world applications
  - SSL Inspection, SSL Proxy
    - Scaling, TLS throughput
  - FIPS compliance

- OpenSSL's role
  - SSL Forward Proxy relies on customized OpenSSL libcrypto/libssl
  - Optimizing these impacts performance and competitiveness
  - Central in FIPS/CC validations

OS and platform architecture

- SRX Platforms & OpenSSL Usage
  - All SRX platforms run Junos OS, built on a FreeBSD-based kernel
  - Applications link to OpenSSL dynamically or statically
  - Modifications made to OpenSSL for performance or features
- Cross-Platform OpenSSL support
  - Our team supports OpenSSL across multiple OS environments
  - Covering a wide range of architectures
- Chassis FIPS boundary adds additional complexity
- Changes of this magnitude are tested by everyone and everything

SRX lineup

- SRX380
  - Branch SRX
  - FreeBSD 12 based Junos OS; Octeon64 Architecture
  - Firewall performance 20 Gbps
  - IPS performance 2Gbps

- SRX2300
  - Mid-range SRX; Campus, Data Center
  - FreeBSD 15 based Junos OS; amd64 Architecture
  - Firewall performance 39 Gbps
  - IPS performance 35 Gbps

SRX lineup

- SRX4300
  - High Performance SRX w/ Hyper Threading; Campus, Data Center
  - FreeBSD 15 based Junos OS; amd64 Architecture
  - Firewall performance 90 Gbps
  - IPS performance 45 Gbps

- SRX5800
  - Datacenter SRX
  - FreeBSD 15 based Junos OS; amd64 Architecture
  - Firewall performance 3.36 Tbps
  - IPS performance 638 Gbps

SRX lineup

- vSRX
  - Virtual SRX, Large – 16CPU-32G
  - FreeBSD 15 based Junos OS; amd64 Architecture
  - Firewall performance 200 Gbps
  - IPS performance 29 Gbps

SRX baseline comparison

- Scaling
  - Measure the maximum SSL sessions supported on the platform using the SSL Proxy
  - HTTPS initiation and teardown rate over SSL forward proxy (CPS)
- Throughput Performance
  - Transparent proxy – SSL encryption and decryption between client and server

Platform	Firewall performance (max)	IPS performance	Max SSL Inspection-FP Sessions (in K sessions) (25.2R1)		SSLFP – As a Service (CPS in K sessions) (25.2R1)	
			TLS 1.2	TLS 1.3	TLS 1.2	TLS 1.3
SRX380	20 Gbps	2 Gbps	**2**	**1.1**	**0.19**	**0.09**
SRX2300	39 Gbps	35 Gbps	**N/A**	**N/A**	**10.5**	**6.45**
SRX4300	90 Gbps	45 Gbps	**250**	**190**	**14.47**	**7.25**
SRX5800	3.36 Tbps	638 Gbps	**130**	**130**	**22**	**13.5**
vSRX	29 Gbps	29 Gbps	**64**	**5**	**6**	**4.4**

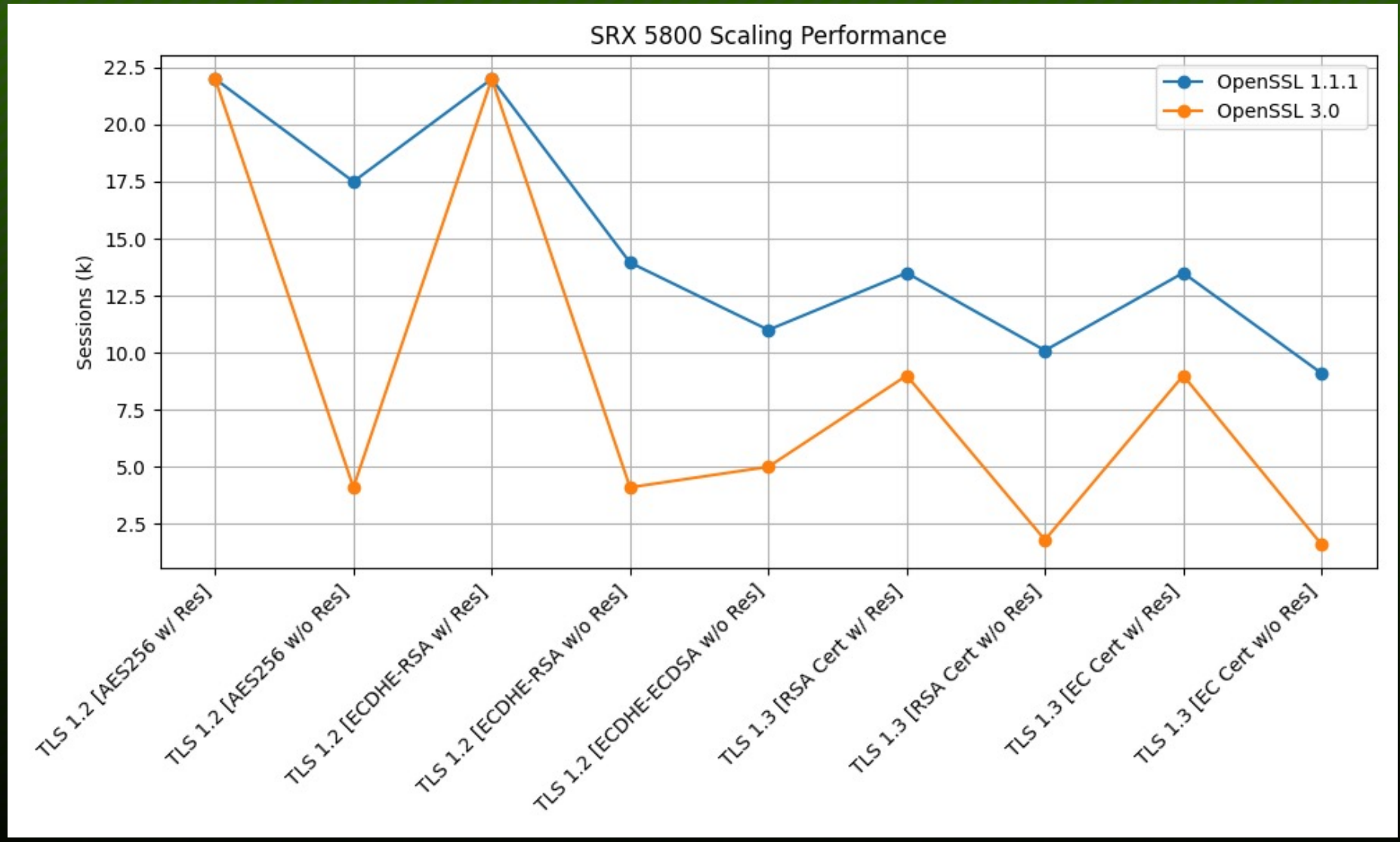
Upgrade Journey

OpenSSL 3.0 testing

- Configurations used in testing
  - TLS 1.3
    - RSA/EC Certificates
    - TLS-AES128-GCM-SHA-256
  - TLS 1.2
    - ECDHE-RSA-AES128-GCM-SHA-256
    - ECDHE-ECDSA-AES128-GCM-SHA-256
    - AES256-GCM-SHA-384
  - Session resumption enabled and disabled
  - Calls-per-second (CPS) / Throughput-per-second (TPS) & Scaling
  - Test cases mirror defaults or common configurations
- Tested on SRX2300 and SRX5800
- OpenSSL 1.1.1u vs. OpenSSL 3.0.8

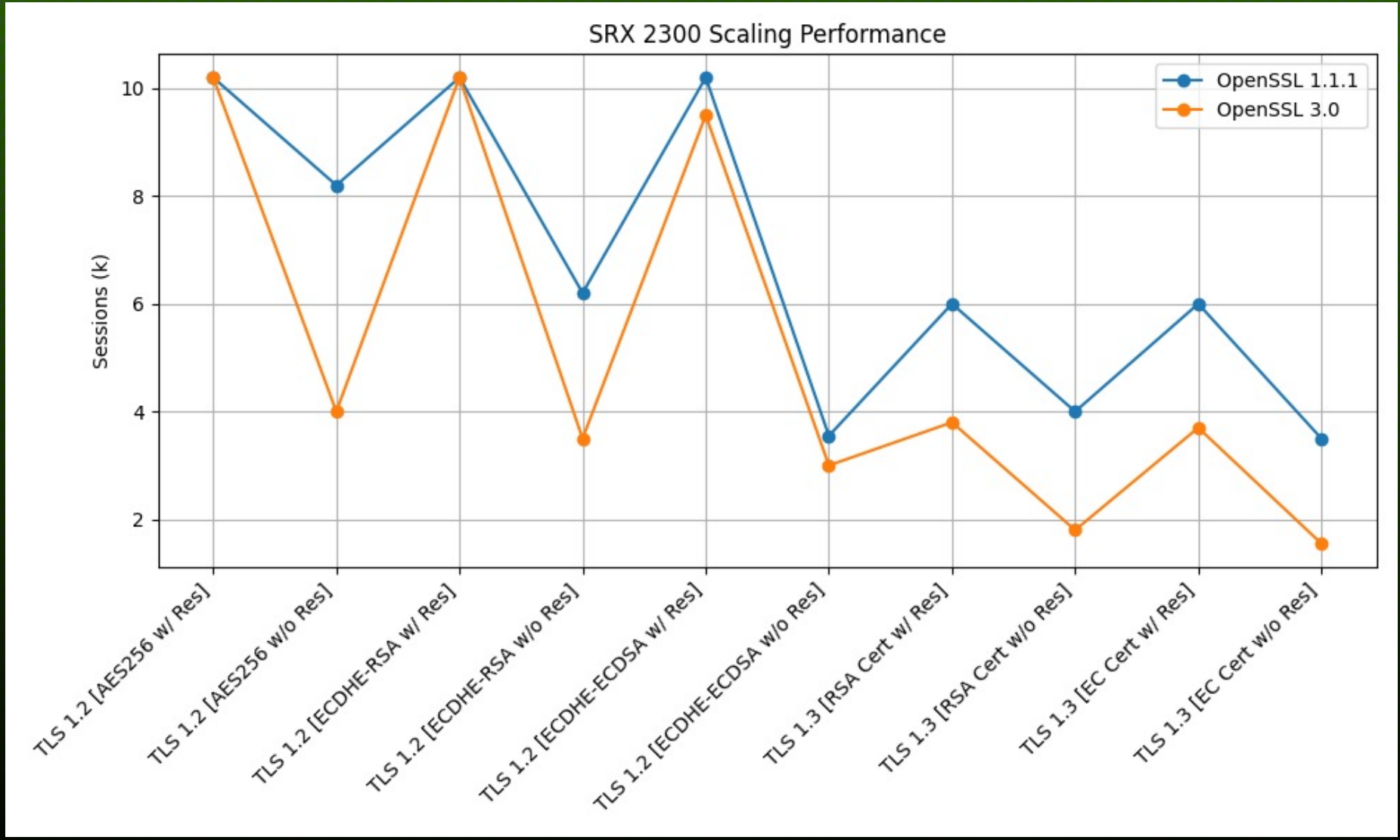
OpenSSL 3.0 findings

- TLS 1.2
  - Cases with session resumption performed on par
  - Without session resumption shows ~50-70% degradation
- TLS 1.3
  - All cases show ~30-80% degradation

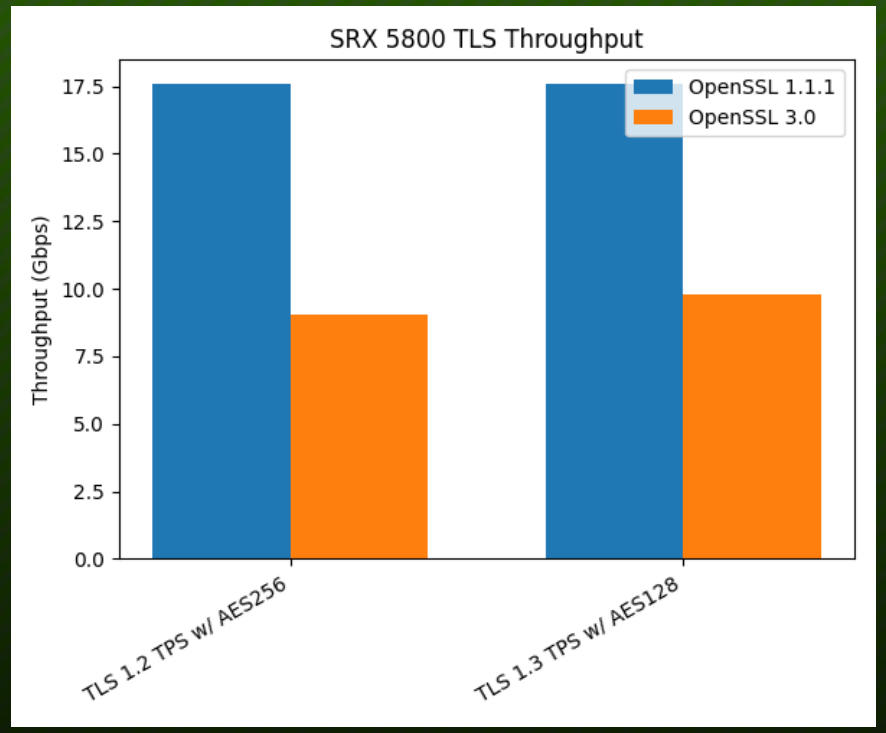


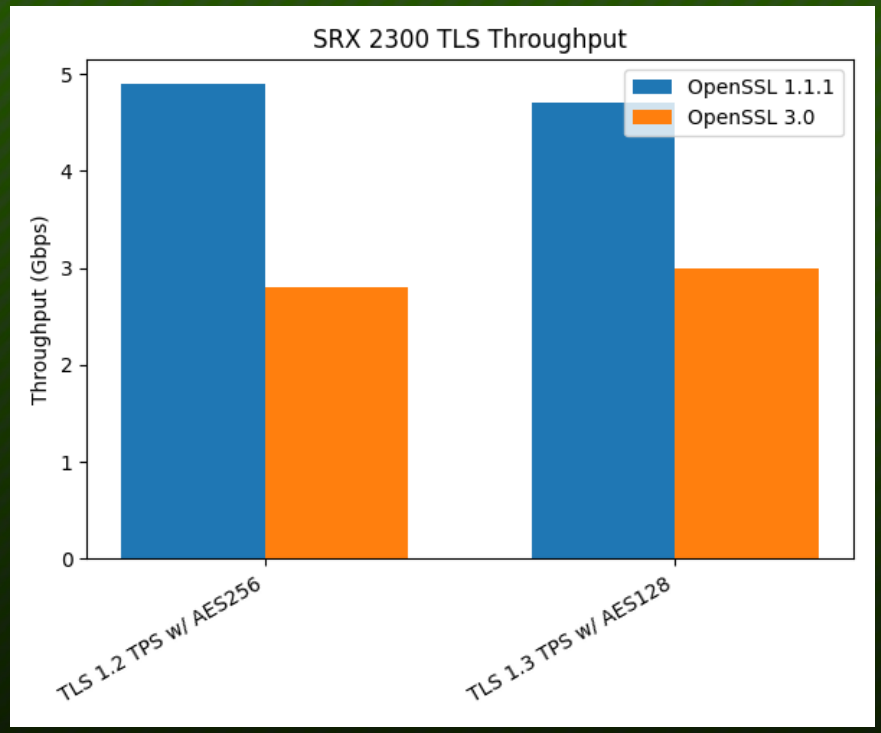
OpenSSL 3.0 findings

- TLS 1.2
  - Cases with session resumption performed on par
  - Without session resumption shows ~50-70% degradation
- TLS 1.3
  - All cases show ~30-80% degradation



OpenSSL 3.0 findings





- SRX 5800 observed ~50% degradation
- SRX 2300 observed ~30-40% degradation

OpenSSL 3.0 findings, mitigations & insights

- Performance Impact
  - Reliably saw a 35-50% drop in throughput
  - Scaling saw a mix of impacts with 0-75% degradation
  - Determined issues stemmed from locks introduced in OpenSSL 3.0
  - Resulting in increase in CPU-bounded operations

- Decision made to defer upgrade and wait for future releases

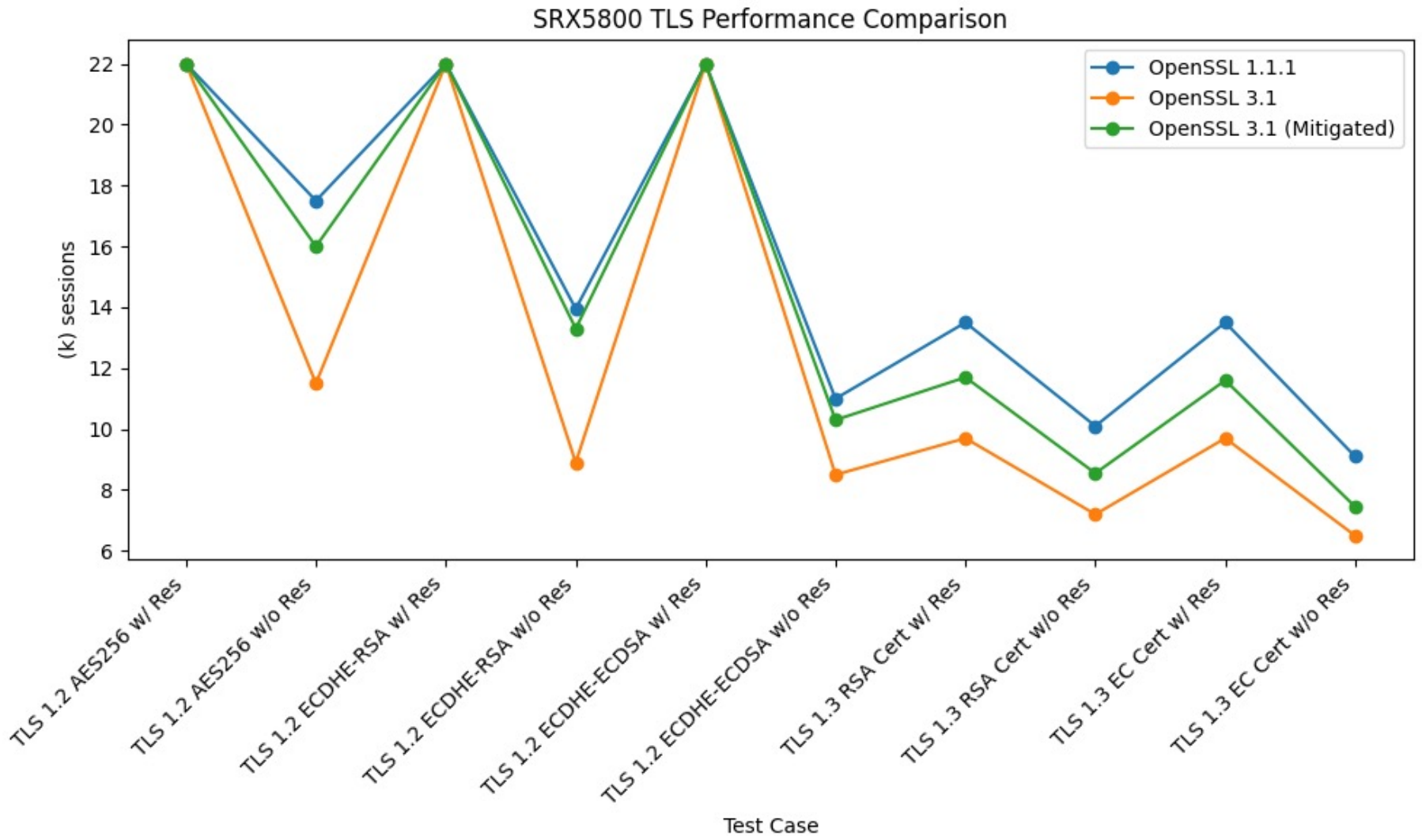
- Explore potential mitigations

OpenSSL 3.1 testing

- Same test suite as OpenSSL 3.0 testing
  - Now performed on wider range of SRX platforms
- OpenSSL 1.1.1za vs. OpenSSL 3.1.2
- Performed mitigations and optimizations
  - Deeper evaluation of our configuration and compilation methods
  - Disabled EVP_PKEY_public_check() during session handshake
  - vSRX experiment in disabling AVX2 paths

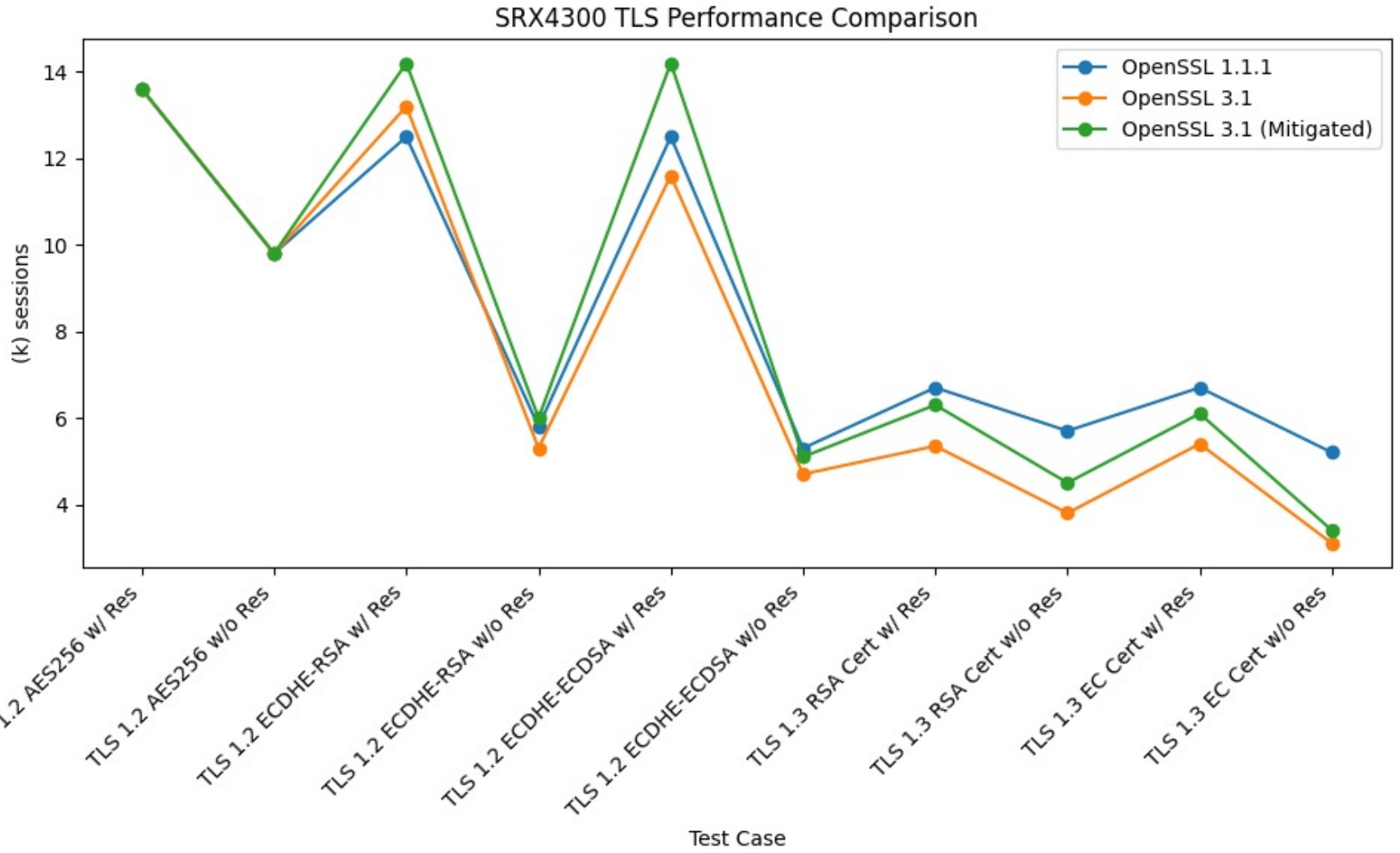
OpenSSL 3.1 findings - Scaling

- TLS 1.2
  - Session resumption performed on par
  - Without session resumption shows ~20-30% degradation
  - With mitigations <10%
- TLS 1.3
  - All cases show only ~30% degradation
  - With mitigations ~15%



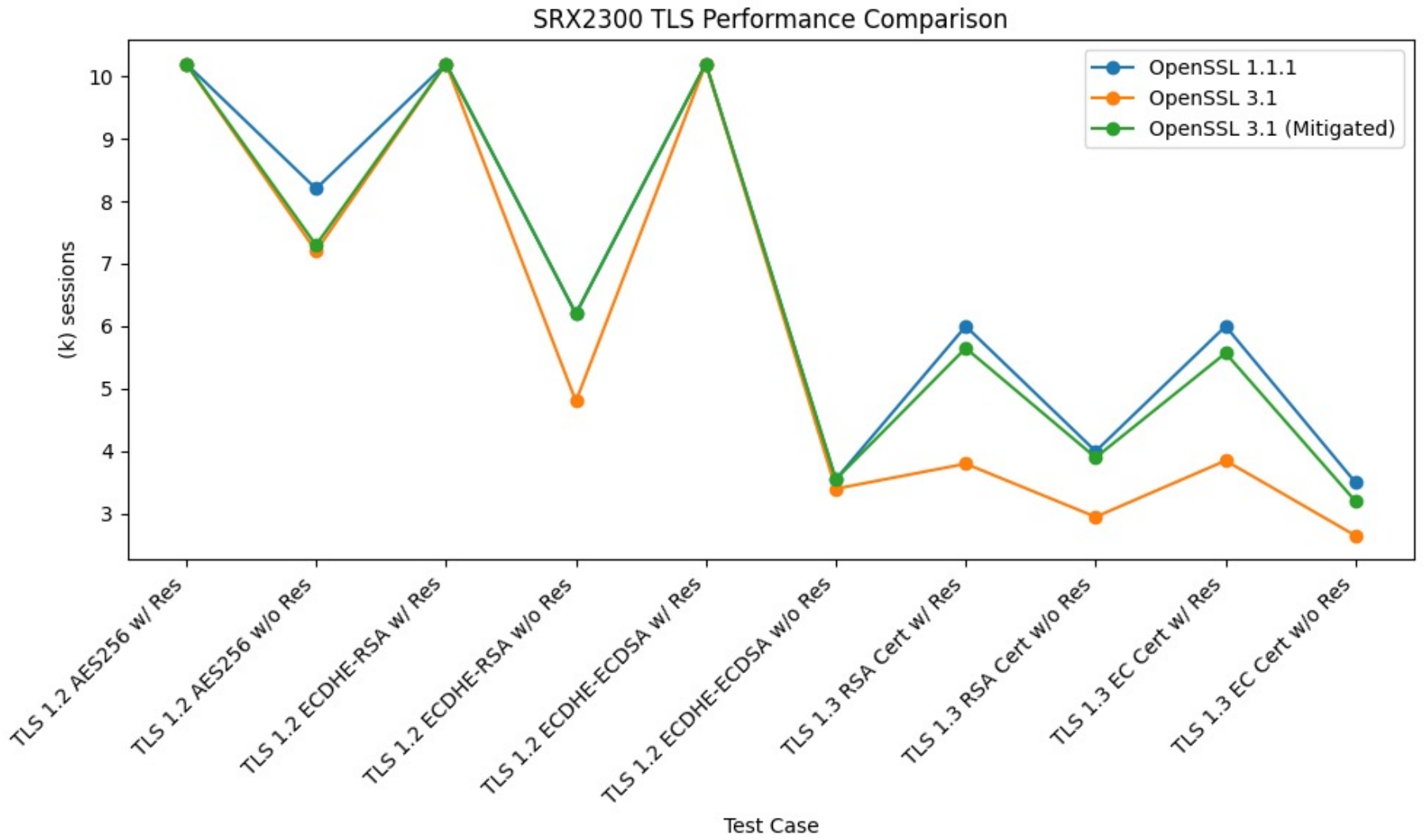
OpenSSL 3.1 findings - Scaling

- TLS 1.2
  - Session resumption performed on par
  - Without session resumption shows ~10% degradation
  - Mitigations improve against 1.1.1
- TLS 1.3
  - All cases show ~20-40% degradation
  - With mitigations 10-30% degradation still seen



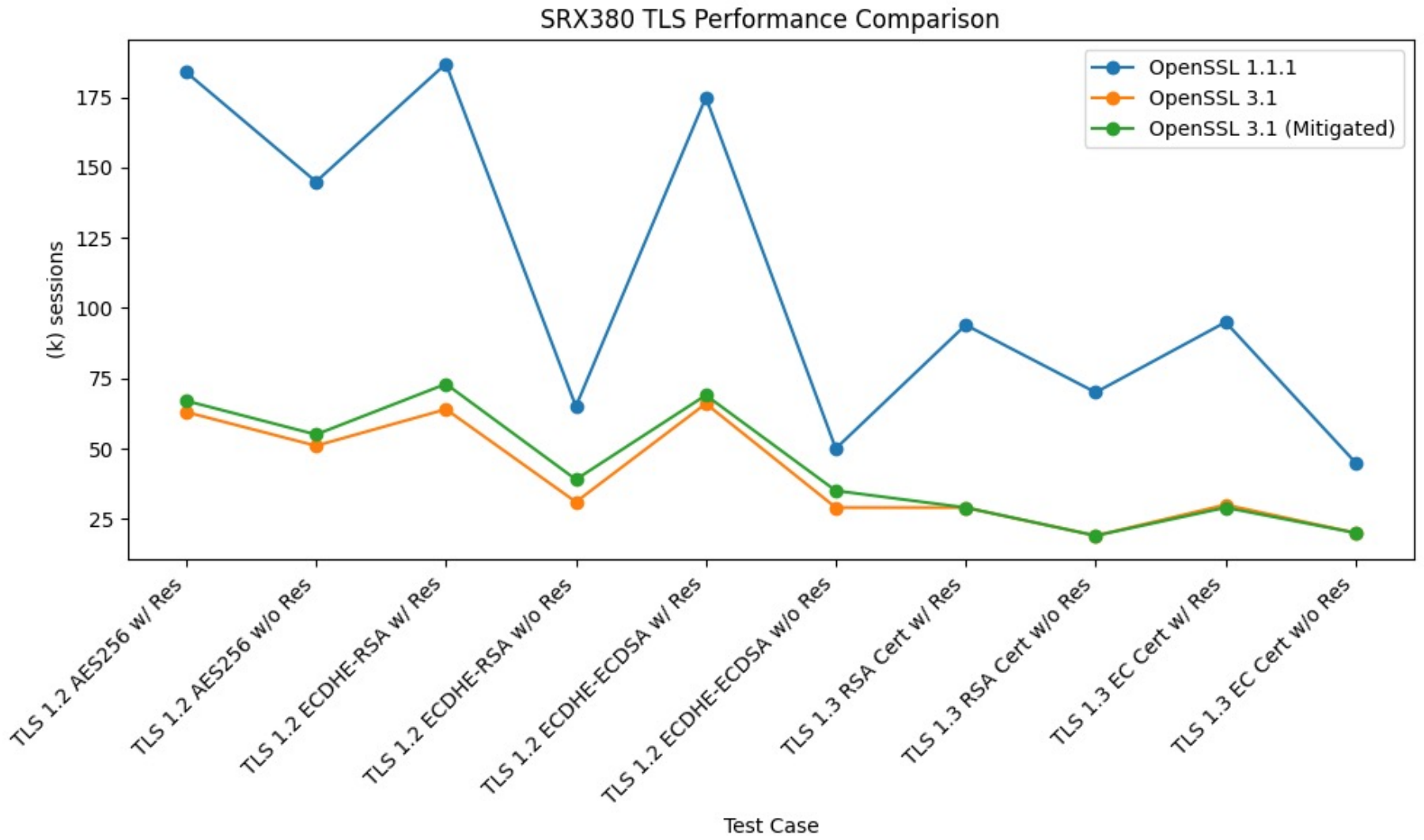
OpenSSL 3.1 findings - Scaling

- TLS 1.2
  - Session resumption performed on par
  - Without session resumption shows ~10-20% degradation
- TLS 1.3
  - All cases show only ~20-30% degradation
  - With mitigations <10%



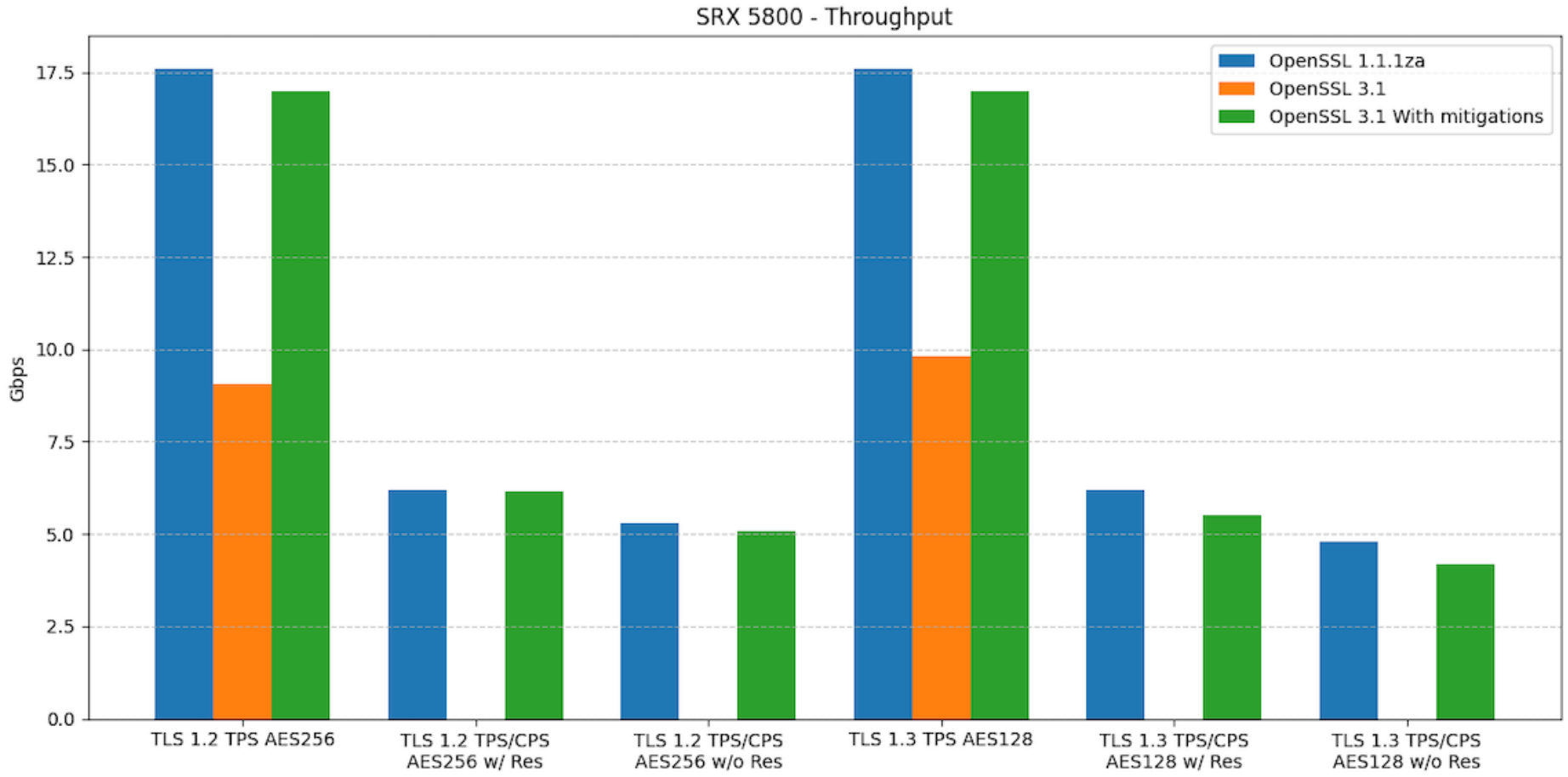
OpenSSL 3.1 findings - Scaling

- TLS 1.2
  - All cases shows ~40-66% degradation
  - Mitigations bring to ~30-60%
- TLS 1.3
  - All cases show only ~50-70% degradation
  - Mitigations had no impact



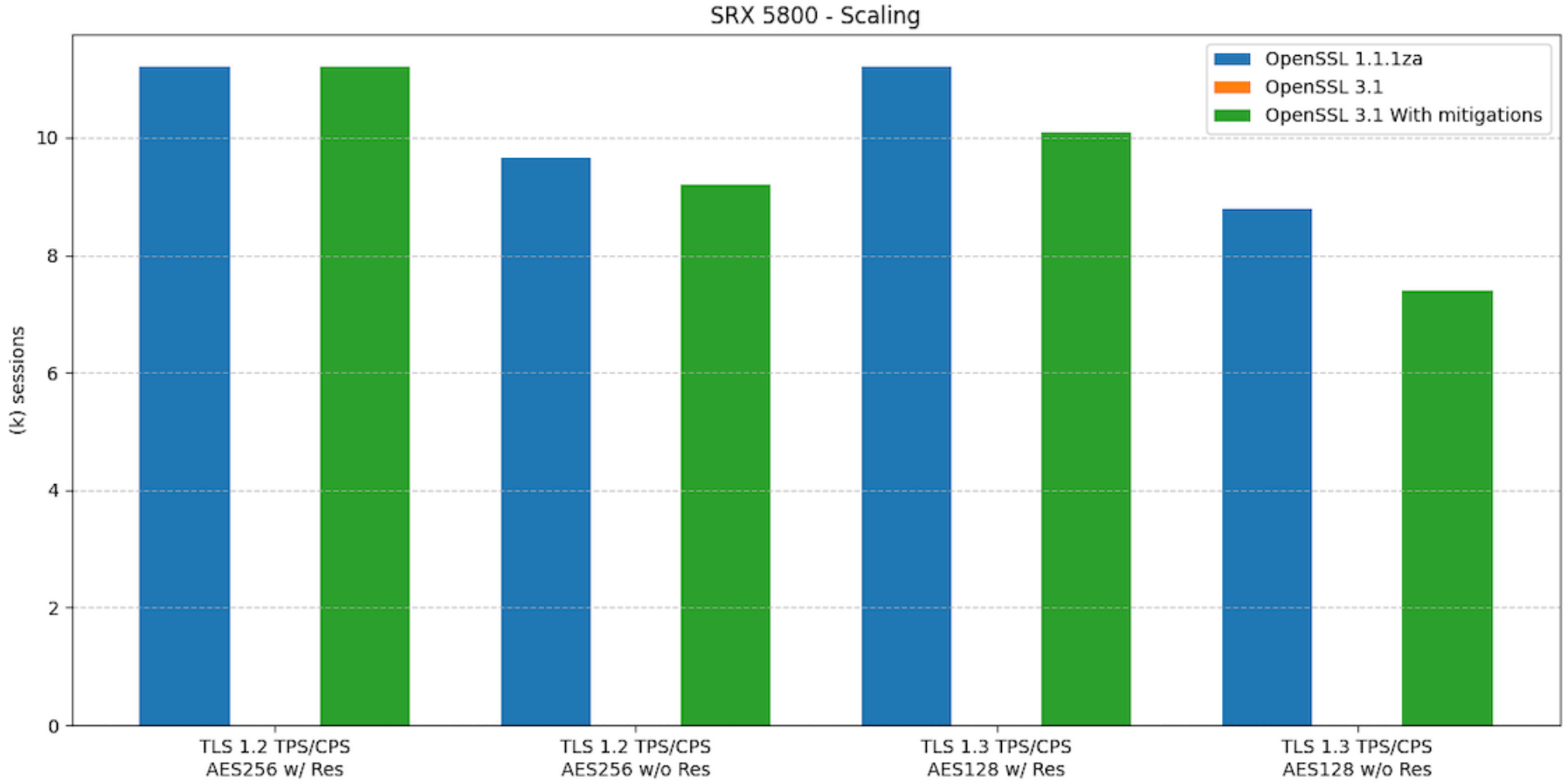
OpenSSL 3.1 findings – Balancing Throughput and Scaling

- Throughput much closer to 1.1.1 after mitigations
- Seeing ~3-4% to 11% degradation now
- Similar story for SRX4300 and SRX2300



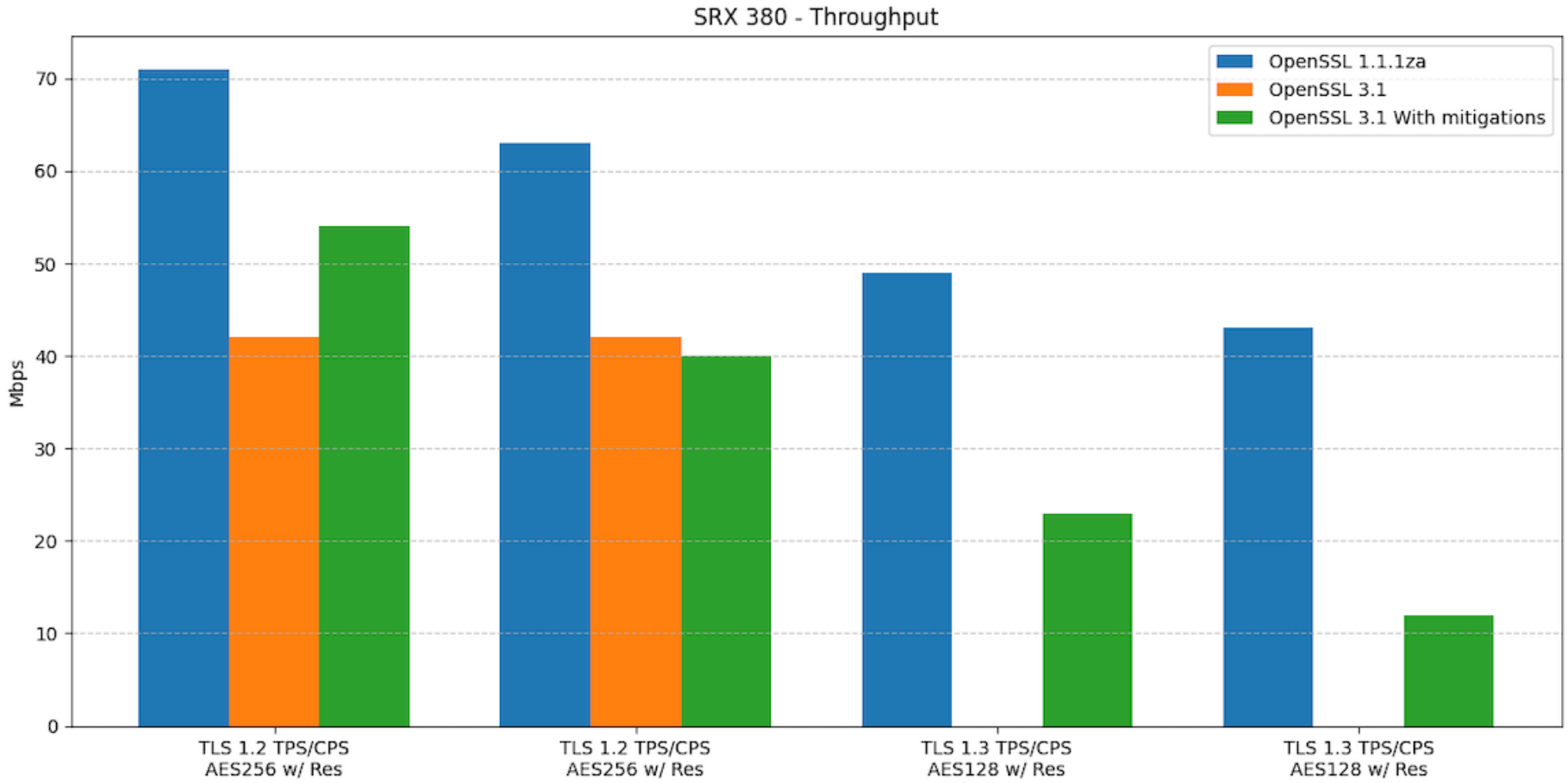
OpenSSL 3.1 findings – Balancing Throughput and Scaling

- Balance of throughput and scaling much closer to 1.1.1 after mitigations
- Seeing 5-15% degradation now
- Similar story for SRX4300 and SRX2300



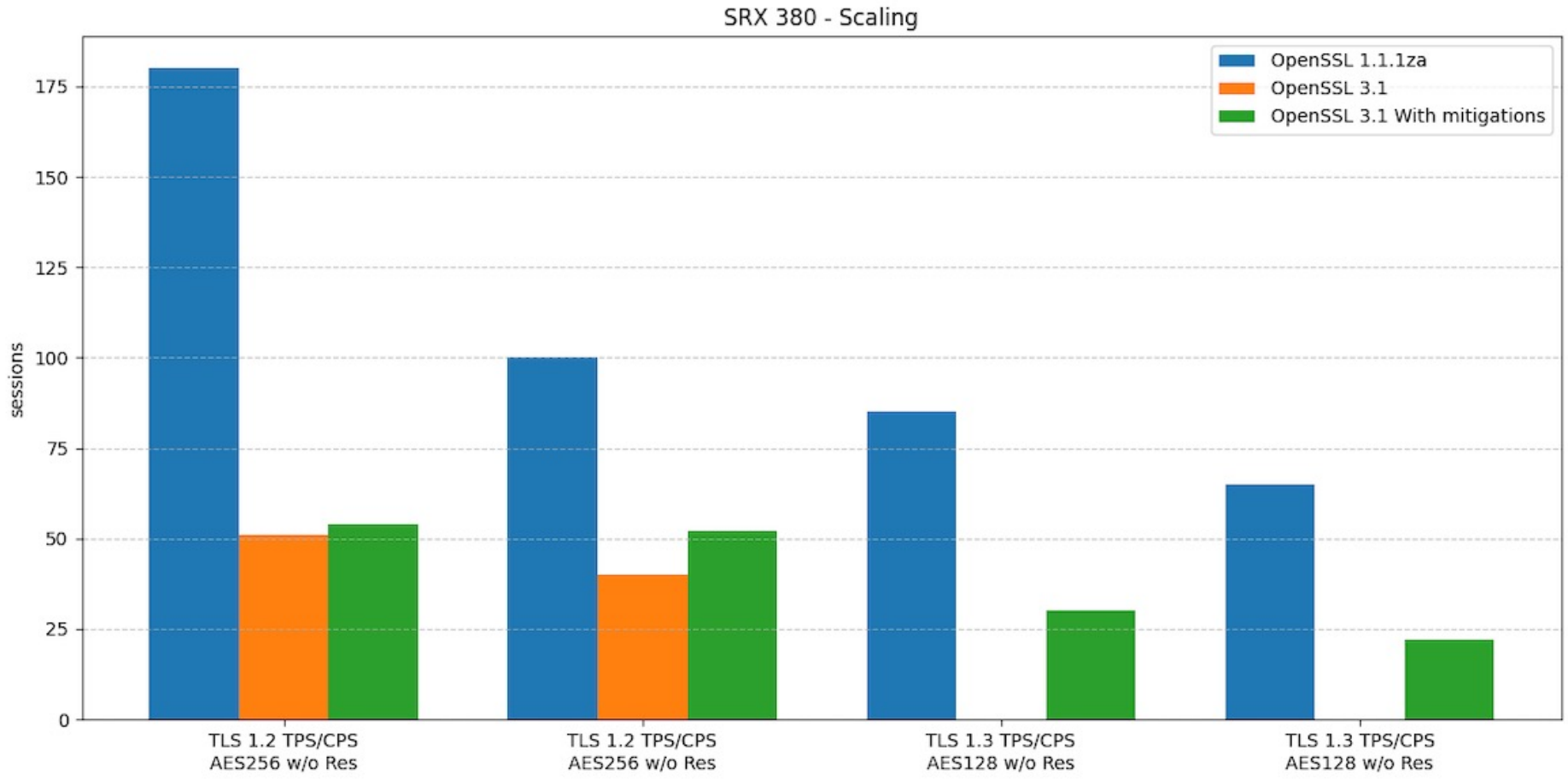
OpenSSL 3.1 findings – Balancing Throughput and Scaling

- SRX380 still has issues
- Even with mitigations, still seeing ~20-70% degradation



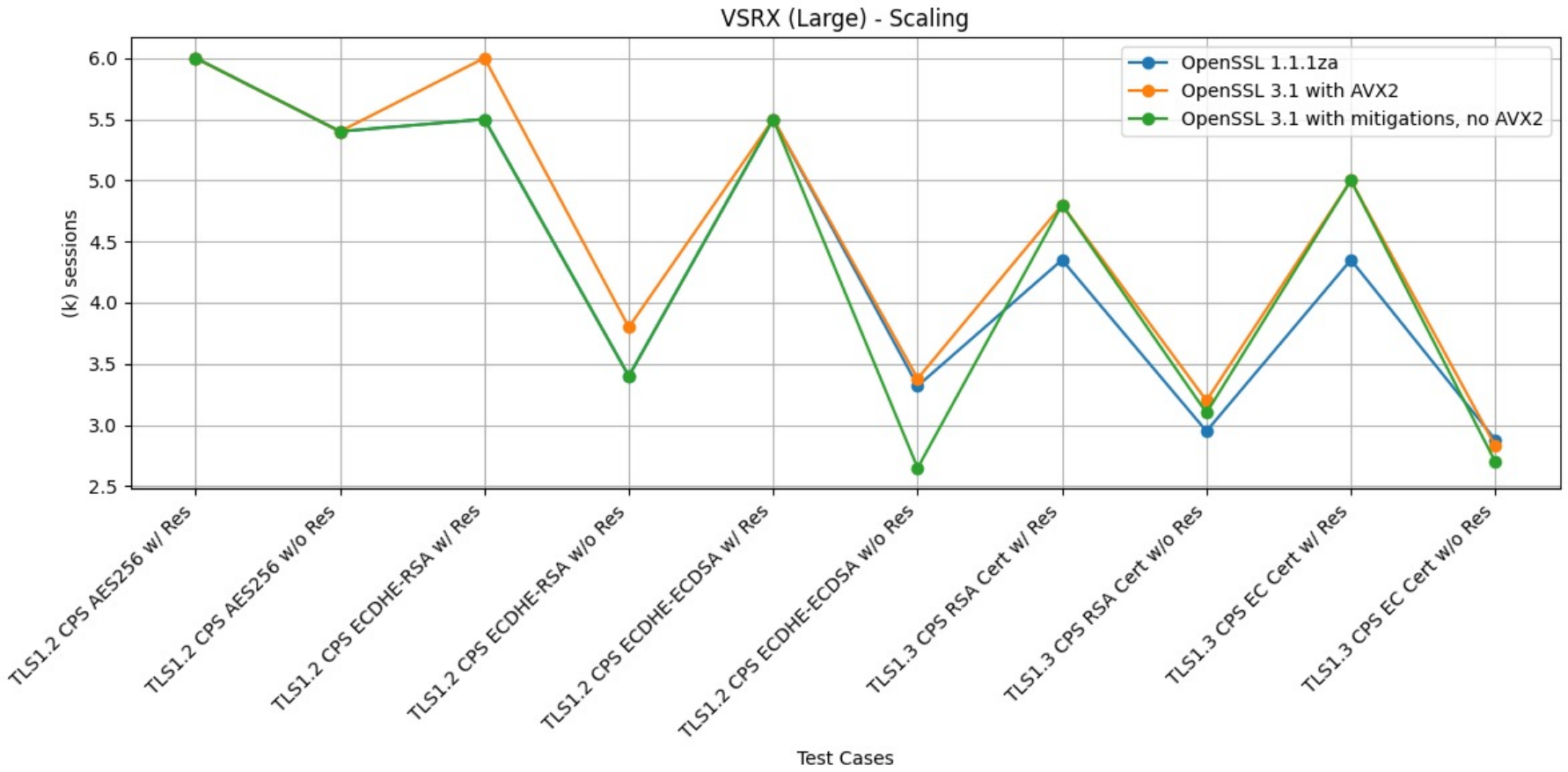
OpenSSL 3.1 findings – Balancing Throughput and Scaling

- SRX380 still has issues
- Even with mitigations, still seeing ~50-70%
- Marginal gains with mitigations



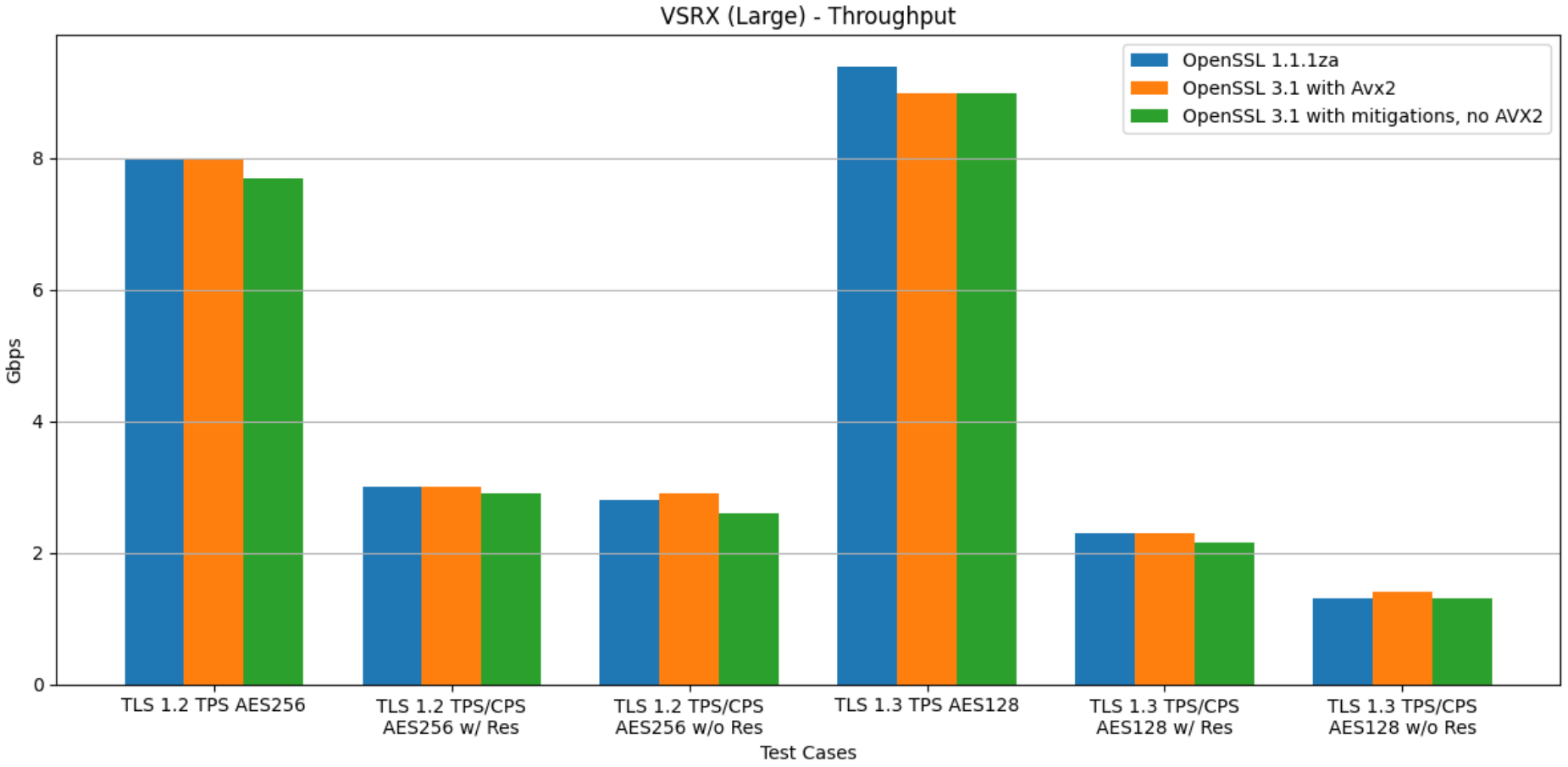
OpenSSL 3.1 findings – vSRX Scaling

- Scaling is inline or improved compared to 1.1.1
- Disabling AVX2 showed no impact or moderate decrease



OpenSSL 3.1 findings – vSRX Balancing Throughput/Scaling

- Balancing throughput and scaling shows ~4% gain or loss – very inline
- Disabling AVX2 drops us to only a 3-7% degradation



OpenSSL 3.1 findings – vSRX Balancing Throughput/Scaling

- Balancing throughput and scaling shows ~5% gain or loss – very inline
- Disabling AVX2 drops us to only a 5-10% degradation

OpenSSL 3.1 findings, mitigations & insights

- Performance improvement over 3.0
  - Most scaling and throughput improved
  - Seeing more inline with 1.1.1, but a few outliers still prevent upgrade

- CPU profiling revealed spike in bn_mul_mont() calls
  - Computationally expensive modular multiplication
  - EVP_PKEY_public_check() was the root cause

- Mitigations
  - Disabled public key validation
    - Showed improvement but still lagging 1.1.1 performance
  - Disabled AVX2-specific paths
    - Suspected this would cause issues but instead AVX2 paths performed nearly the same or only slightly worse

OpenSSL 3.4 testing

- Same SRX lineup minus the SRX380
- Configurations used in testing
  - TLS 1.2 per session memory consumption
  - TLS 1.3
    - CPS with RSA certs toggling session resumption
    - Scaling with ECDSA and RSA
  - TLS 1.2
    - CPS with AES256-GCM-SHA-384 toggling session resumption
    - TP & CPS with AES256-GCM-SHA-384 toggling session resumption
    - Scaling with ECDSA and RSA
  - Test Cases run with optimized memory pooling
  - Minimal concerns with throughput, more interested in scaling
- OpenSSL 1.1.1zb vs. OpenSSL 3.4.0

OpenSSL 3.4 findings – Memory Consumption

OpenSSL 1.1.1za		
No of objs allocated	Obj Size	Bytes allocated
28	32	896
203	64	12992
46	128	5888
25	256	6400
5	328	1640
2	544	1088
12	1296	15552
2	8400	16800
	Total bytes	61256
	Total kb	61.26 Kb

OpenSSL 3.4 (Initial)		
No of objs allocated	Obj Size	Bytes allocated
29	32	928
207	64	13248
37	128	4736
32	256	8192
7	328	2296
2	544	1088
10	1296	12960
6	8400	50400
	Total bytes	93848
	Total kb	93.85 Kb

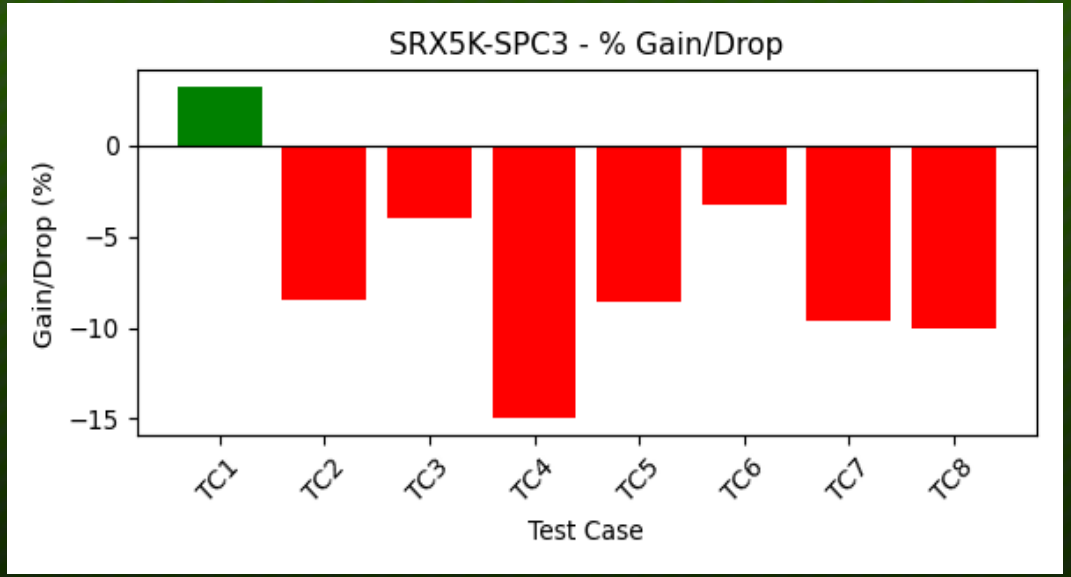
Openssl 3.4 with new pools (4456, 5464, 608 , 800, 1024)		
No of objs allocated	Obj Size	Bytes allocated
29	32	928
208	64	13312
37	128	4736
32	256	8192
8	328	2624
3	544	1632
4	800	3200
6	1024	6144
4	4456	17824
2	5464	10928
	Total bytes	69520
	Total kb	69.52 Kb

OpenSSL 3.4 findings – Memory Consumption

- Investigations into scaling throttling pointed towards memory consumption
- TLS 1.2 per session memory consumption was recorded
  - ~50% initial increase using
  - SRX memory pool optimization and tweaking for performance gains
    - At a cost to other subsystems
- OpenSSL 1.1.1za recorded at 61.2 Kb
- OpenSSL 3.4.0 recorded at 93.85 Kb
  - Using existing memory pool optimization
- Post memory pool optimization, OpenSSL 3.4 recorded with 69.5 Kb

OpenSSL 3.4 findings – SRX5800

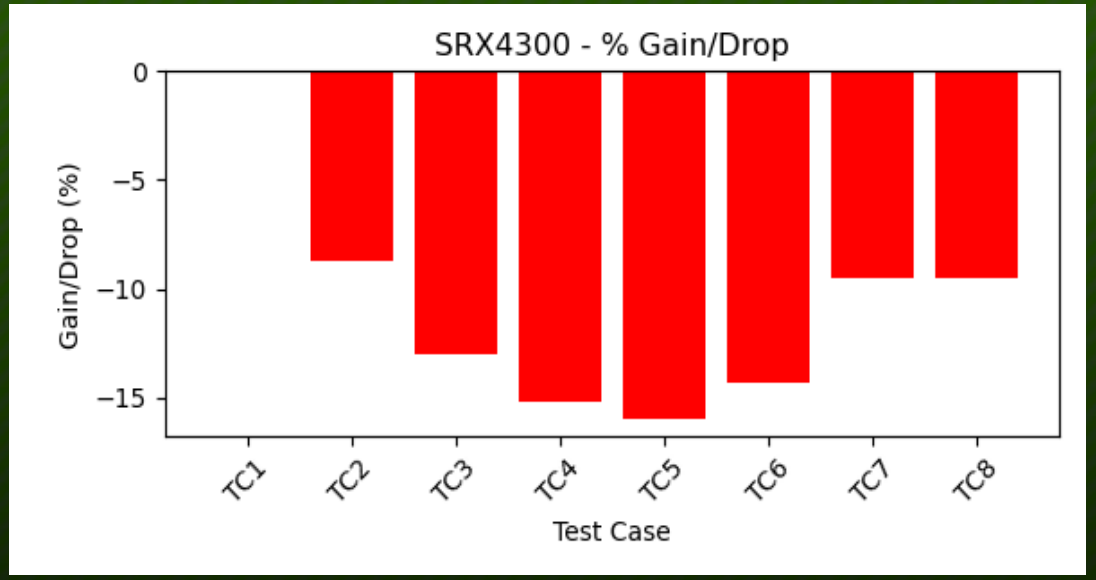
- Mid and high-range SRX hit harder
- Scaling not performing as well as 3.1



SRX5800 Performance Comparison (in sessions)				
TC#	Test Case	OpenSSL 1.1.1	OpenSSL 3.4	% Change
TC1	TLS1.2 CPS with Resumption(AES256-GCM-SHA-384)	25K	25.8K	3.2
TC2	TLS1.2 CPS without Resumption (AES256-GCM-SHA-384)	20K	18.3K	-8.5
TC3	TLS1.3 CPS with Resumption (RSA Cert)	12.5K	12K	-4
TC4	TLS1.3 CPS without Resumption (RSA Cert)	11.8K	9.8K	-15
TC5	TLS1.2 Scaling (RSA)	350K	320K	-8.6
TC6	TLS1.2 Scaling (ECDSA)	300K	290K	-3.3
TC7	TLS 1.3 Scaling (RSA)	310K	280K	-9.6
TC8	TLS 1.3 Scaling (ECDSA)	300K	270K	-10

OpenSSL 3.4 findings – SRX4300

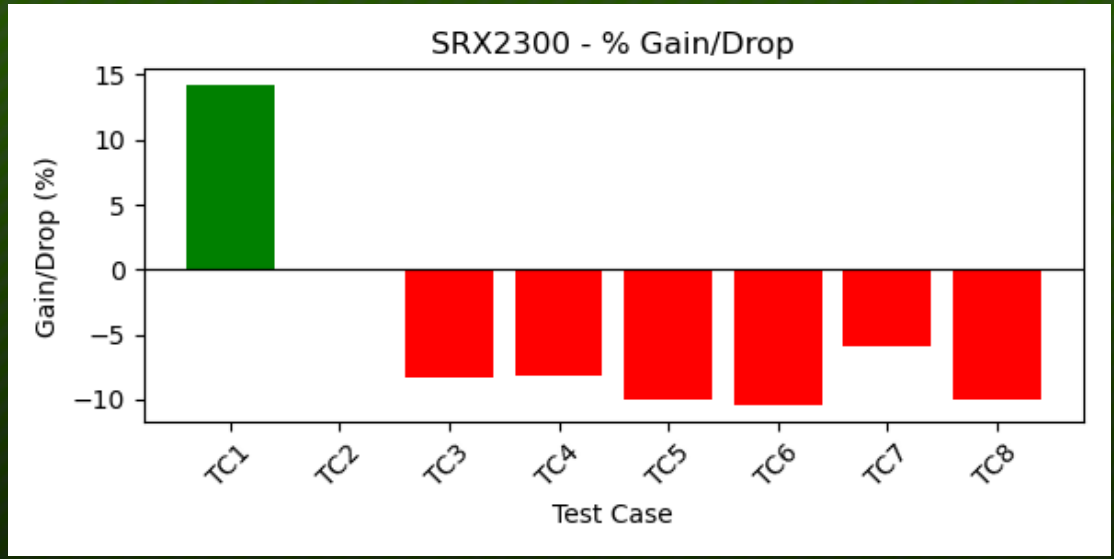
- Mid and high-range SRX hit harder
- Scaling not performing as well as 3.1



SRX4300 Performance Comparison (in sessions)				
TC#	Test Case	OpenSSL 1.1.1	OpenSSL 3.4	% Change
TC1	TLS1.2 CPS with Resumption(AES256-GCM-SHA-384)	14.3K	14.4K	0
TC2	TLS1.2 CPS without Resumption (AES256-GCM-SHA-384)	10.3K	9.4K	-8.7
TC3	TLS1.3 CPS with Resumption (RSA Cert)	6.9K	6.0K	-13
TC4	TLS1.3 CPS without Resumption (RSA Cert)	5.9K	5.0K	-15.2
TC5	TLS1.2 Scaling (RSA)	250K	210K	-16
TC6	TLS1.2 Scaling (ECDSA)	245K	210K	-14.3
TC7	TLS 1.3 Scaling (RSA)	210K	190K	-9.5
TC8	TLS 1.3 Scaling (ECDSA)	210K	190K	-9.5

OpenSSL 3.4 findings – SRX2300

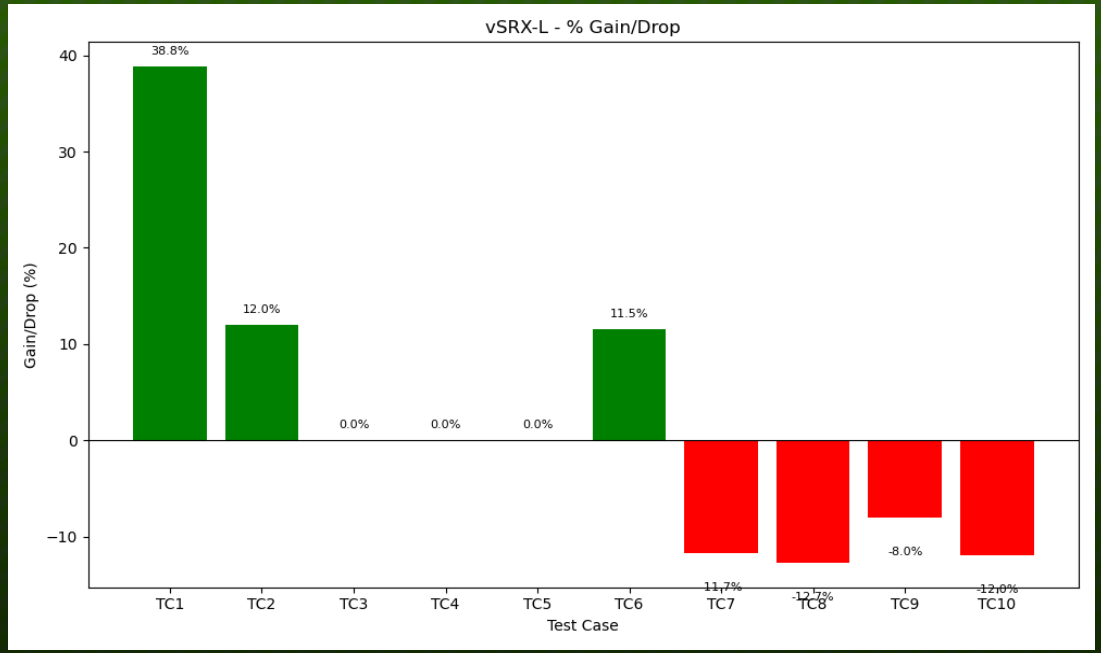
- Consistently seeing 10% or less degradation
- Scaling still the bottleneck



SRX2300 Performance Comparison (in sessions)				
TC#	Test Case	OpenSSL 1.1.1	OpenSSL 3.4	% Change
TC1	TLS1.2 CPS with Resumption(AES256-GCM-SHA-384)	10.5K	12K	14.2
TC2	TLS1.2 CPS without Resumption (AES256-GCM-SHA-384)	7.8K	7.8K	0
TC3	TLS1.3 CPS with Resumption (RSA Cert)	6K	5.5K	-8.3
TC4	TLS1.3 CPS without Resumption (RSA Cert)	4.9K	4.5K	-8.1
TC5	TLS1.2 Scaling (RSA)	105K	95K	-10
TC6	TLS1.2 Scaling (ECDSA)	95K	85K	-10.5
TC7	TLS 1.3 Scaling (RSA)	85K	80K	-5.9
TC8	TLS 1.3 Scaling (ECDSA)	80K	72K	-10

OpenSSL 3.4 findings – vSRX

- vSRX still has mixed results
  - TLS 1.2 scaling has improved dramatically
  - TLS 1.3 scaling is still lagging
  - Balanced test sees improvement as well



vSRX-Large Performance Comparison (in sessions / Gbps)				
TC#	Test Case	OpenSSL 1.1.1	OpenSSL 3.4	% Change
TC1	TLS1.2 CPS with Resumption(AES256-GCM-SHA-384)	8K	11.1k	38.8
TC2	TLS1.2 CPS without Resumption (AES256-GCM-SHA-384)	5.8K	6.5k	12
TC3	TLS1.3 CPS with Resumption (RSA Cert)	4.4K	4.4k	0
TC4	TLS1.3 CPS without Resumption (RSA Cert)	3.4K	3.4k	0
TC5	TLS1.2 TP with CPS (AES256-GCM-SHA-384)(with resumption)	4.0Gbps	4.0Gbps	0
TC6	TLS1.2 TP with CPS (AES256-GCM-SHA-384)(without resumption)	2.6Gbps	2.9Gbps	11.5
TC7	TLS1.2 Scaling (RSA)	60k	53k	-11.7
TC8	TLS1.2 Scaling (ECDSA)	55k	48k	-12.7
TC9	TLS 1.3 Scaling (RSA)	50k	46k	-8
TC10	TLS 1.3 Scaling (ECDSA)	50k	44k	-12

OpenSSL 3.4 findings, mitigations & insights

- We observed considerable improvements on raw throughput performance
  - Some test cases performed faster than 1.1.1
- EVP_PKEY_public_check() issue not seen in 3.4
  - Performance numbers are with no mitigations, using raw 3.4
- Instrumented memory pool to match OpenSSL 3.4 memory allocation
  - Per session memory consumption increased due to architectural changes
  - Our ability to scale concurrent sessions dropped by ~15-20% on average
- CPU overhead reduced -> memory consumption is new concern
  - Scaling still degraded by 3% - 16%

Testing conclusions

- vSRX performed better on average with OpenSSL 3.4
- Mid and High-end SRX impacted more in general, scaling impacts all
- TLS session scaling is constrained by memory consumption
- OpenSSL 3.0: Immediate 40–50% drop in throughput performance
  - CPU-bound operations from architectural changes. Upgrade deferred
- OpenSSL 3.1: Marginal improvement
  - bn_mul_mont and EVP_PKEY_public_check() introduced new CPU overheads
  - Mitigations and compromises helped but didn’t match 1.1.1
- OpenSSL 3.4: Raw performance improved
  - EVP_PKEY_public_check() CPU usage not seen in baseline
  - Raw performance, with no mitigations, closest to 1.1.1
  - per-session memory usage increased, scaling now down ~15–20%

Future Work

Roadmap

- Government, Company mandates for PQC
  - Forcing function for OpenSSL 3.5
- Continue to investigate memory consumption in 3.4
  - Full feature testing for 3.5 not complete yet
- Chassis based FIPS boundary means no FIPS provider
  - Can’t take full advantage of FIPS provider in Junos
  - Modify for FIPS validations

Acknowledgements

- OpenSSL support
  - OpenSSL support has been helpful and communicative
  - Planning to share data and continue to collaborate

- OpenSSL community
  - Public patches related to memory consumption
  - Seeing others with similar issues, results

- Juniper to increase our engagement
  - Joining Large Business calls
  - More communicative back and forth with Open Source
  - Welcome feedback, discussion, ideas

Questions?

- Rakesh Sharma rakeshks@juniper.net
- Subrahmanya M subrahmanyam@juniper.net

- David O’Brien deo@juniper.net
- Kamlesh Kumar kam@juniper.net
- Nivethitha Chandrasekaran nive@juniper.net
- William Bellingrath wbellingrath@juniper.net

Thank you

Juniper Networks | Driven by Experience™