

OpenDaylight Performance Stress Test Report

v1.1: Lithium SR3

Intracom Telecom SDN/NFV Lab

www.intracom-telecom.com | [intracom-telecom-sdn.github.com](https://github.com/intracom-telecom-sdn) | sdn-nfv@intracom-telecom.com



OpenDaylight Performance Stress Tests Report

v1.1: Lithium SR3

Executive Summary

In this report we investigate several performance aspects of the OpenDaylight Lithium RC0 controller and compare them against the Helium SR3 release. The investigation targets on stability and scalability tests. Stability tests explore how controller throughput behaves in a large time window with a fixed topology connected to it, the goal of which, is to detect performance fluctuations over time. Scalability tests measure controller performance as the switch topology scales, giving hint on the controller's upper bound.

Contents

1	Introduction	3
2	NSTAT Toolkit	3
3	Experimental setup and configurations	4
4	Switch scalability stress tests	5
4.1	Active MT-Cbench switches	5
4.1.1	Test configuration, "RPC" mode	5
4.1.2	Test configuration, "RPC" mode, "Lithium-design" OpenFlow plugin	5
4.1.3	Test configuration, "DataStore" mode	6
4.1.4	Test configuration, "DataStore" mode, "Lithium-design" OpenFlow plugin	6
4.1.5	Native vs virtualized execution	6
4.2	Idle Mininet switches	7
4.2.1	Disconnected and Linear topologies	7
4.2.2	Mesh topology	7
4.3	Idle MT-Cbench switches	8
4.3.1	RPC mode	8
4.4	Conclusions	8
5	Stability tests	9
5.1	Active MT-Cbench switches	9
5.1.1	"DataStore" mode, 12 hours running time	9
5.1.2	"RPC" mode, 12 hours running time	10
5.1.3	"RPC" mode, "Lithium Design", OpenFlow plugin 12 hours running time	10
5.2	Conclusions	10
6	Flow scalability tests	10
6.1	Idle multinet switches	11
6.1.1	10 switches	11
6.1.2	10 switches, "Lithium-design" OpenFlow plugin	11
6.1.3	100 switches	11
6.2	Conclusions	11
7	Contributors	12
	References	12

1. Introduction

In this report we investigate several performance and scalability aspects of the OpenDaylight Lithium SR3 controller. The investigation targets the following objectives:

- controller throughput
- switch scalability
- controller stability (sustained throughput)
- flow scalability and provisioning time

For our evaluation we have used NSTAT [1], an open source environment written in Python for easily writing SDN controller stress tests and executing them in a fully automated and end-to-end manner.

For Southbound (SB) trafficgeneration NSTAT uses both MT-Cbench [2] and Mininet [3]. MT-Cbench is a direct extension of the Cbench [4] emulator which used threading to generate OpenFlow trafficfrom multiple streams in parallel. The motivation behind this extension was to be able to boot-up and operate network topologies with OpenDaylight much larger than those with the original Cbench, by gradually adding switches in groups. We note here that, as in the original Cbench, the MT-Cbench switches implement only a minimal subset of the OpenFlow 1.0 protocol, and therefore the results presented here are expected to vary in real-world deployments.

This gap is largely filled using Mininet in NSTAT as an additional SB trafficgenerator, as it uses OVS virtual switches, that accurately emulate the OpenFlow 1.3 protocol. NSTAT provide custom handling logic [5] for Mininet for better control over the lifecycle management of the topology. Specifically, as with MT-Cbench, the main goal was to be able to control the boot-up phase of switches by gradually adding them in groups.

For Northbound (NB) trafficgeneration NSTAT uses custom scripts, originally developed by the OpenDaylight community [5], for creating and writing flows to the controller configuration datastore.

2. NSTAT Toolkit

The general architecture of NSTAT is depicted in Fig.1. The NSTAT node lies at the heart of the environment and controls all others. The test nodes --controller node, SB node(s), NB node-- are mapped on one or more physical or virtual interconnected machines. Unless otherwise stated, in all our experiments every node is mapped on a separate virtual machine.

Table 1: Stress tests experimental setup.

Host operating system	Centos 7, kernel 3.10.0
Server platform	Dell R720
Processor model	Intel Xeon CPU E5-2640 v2 @ 2.00GHz
Total system CPUs	32
CPUs configuration	2 sockets × 8 cores/socket × 2 HW-threads/core @ 2.00GHz
Main memory	256 GB, 1.6 GHz RDIMM
Controller distribution	OpenDaylight Lithium (SR3)
Controller JVM options	-Xmx8G, -Xms8G, -XX:+UseG1GC, -XX:MaxPermSize=8G

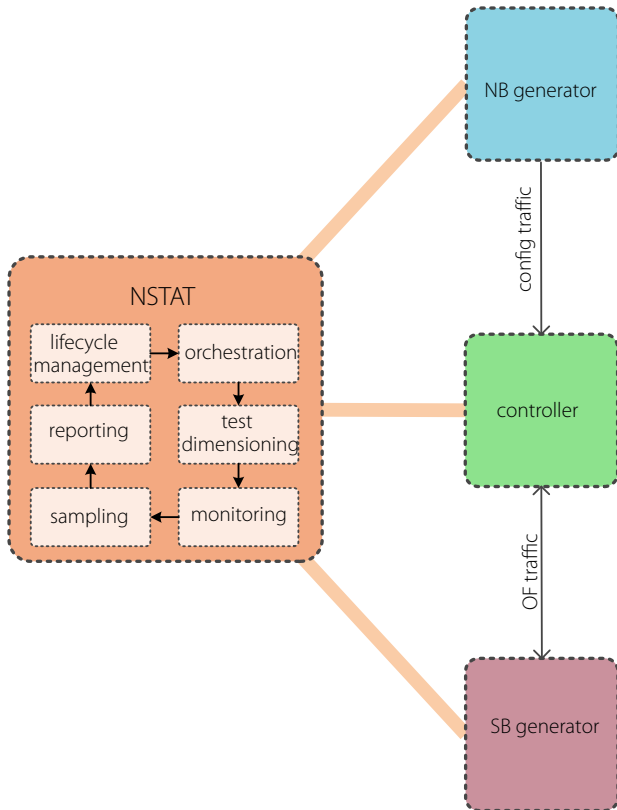
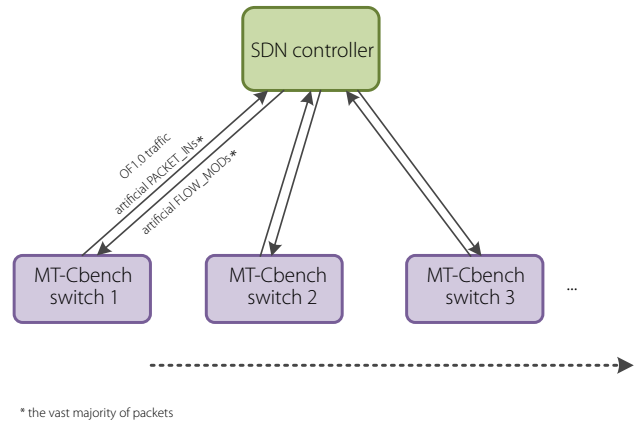


Fig. 1: NSTAT architecture

NSTAT is responsible for automating and sequencing every step of the nodes, initiating and scaling SB and NB traffic, monitoring controller operation in terms of correctness and performance, and producing performance reports along with system and application health state and profiling statistics. Each of these steps is highly configurable using a rich and simple JSON-based configuration system.

Each stress test scenario features a rich set of parameters that determine the interaction of the SB and NB components with the controller, and subsequently trigger varying controller behavior. Such parameters are for example the number of OpenFlow switches, the way they connect to the controller, the number of NB application instances, the rate of NB/SB traffic, and others, and in a sense they define the “dimensions” of a stress test scenario.

One of NSTAT’s key features is that it allows the user to specify



* the vast majority of packets

Fig. 2: Representation of switch scalability stress test with active MT-Cbench switches. The switches send artificial OF1.0 PACKET_IN messages to the controller, which replies with also artificial OF1.0 FLOW_MOD messages.

multiple values for one or more such dimensions at the same time, and the tool itself takes care to repeat the test over all their combinations in a single session. This kind of exhaustive exploration of the multi-dimensional experimental space offers a comprehensive overview of the controller’s behavior on a wide range of conditions, making it possible to easily discover scaling trends, performance bounds and pathological cases.

3. Experimental setup and configurations

The setup used in all experiments included in this report is summarized in the Table 1.

In our stress tests we have experimented with emulated switches operating in two modes: switches in **idle mode** do not initiate any trafficto the controller, but rather respond to messages sent by it. Switchs in **active mode** consistently initiate trafficto the controller, in the form of Packet-In messages. MT-Cbench switches were tested in both modes, while Mininet was operating only in idle mode.

In MT-Cbench tests the controller should be configured to start with the “drop-test” feature installed. The emulated switches are arranged in a disconnected topology, meaning they do not have any interconnection between them. As we have already mentioned, this feature, along with the limited protocol support, constitute MT-Cbench a special-purpose OpenFlow

generator and not a full-fledged, realistic OpenFlow switch emulator.

Two different configurations of the controller are evaluated with MT-Cbench active switches: in **RPC mode** configuration, the controller is configured to directly reply to Packet-In's sent by the switches with a predefined message at the OpenFlow plugin level. In **DataStore mode** configuration, the controller additionally performs updates in its data store. In all cases, MT-Cbench is configured to operate in "Latency mode", meaning that each switch sends a Packet-In message only after it has received a reply for the previous one.

In some stress tests we also evaluate the two different implementations of the OpenFlow plugin found in the Lithium release: the baseline implementation, codenamed "**Helium-design**", and a new, alternative implementation codenamed "**Lithium-design**", [6, 7]. Unless otherwise stated, the "**Helium-design**" implementation is used by default.

4. Switch scalability stress tests

Switch scalability tests aim at exploring the maximum number of switches the controller can sustain, when switches are being gradually added either in idle or active mode. Apart from the maximum number of switches the controller can successfully see, a few more additional metrics are reported:

- in idle mode tests, NSTAT also reports the topology boot-up time for different policies of connecting switches to the controller. From these results we can deduce how to *optimally boot-up a certain-sized topology* and connect it to the controller, so that the latter can successfully discover it at the minimum time.
- in active mode tests, NSTAT also reports the controller throughput, i.e. the rate at which the controller replies back to the switch-initiated messages. Therefore, in this case, we also get an overview of how controller throughput scales as the topology size scales.

4.1 Active MT-Cbench switches

This is a switch scalability test with switches in active mode emulated using MT-Cbench. Its target is to explore the maximum number of switches the controller can sustain while they consistently initiate traffic to it (active), and how the controller servicing throughput scales as more switches are being added. MT-Cbench switches send artificial OF1.0 PACKET_IN messages to the controller, which replies with also artificial OF1.0 FLOW_MOD messages; these message types dominate the traffic exchanged between switches and the controller. In order to push the controller performance to its limits, all test nodes (controller, MT-Cbench) were executed on bare metal. To isolate the nodes from each other, the CPU shares feature of NSTAT was used [8].

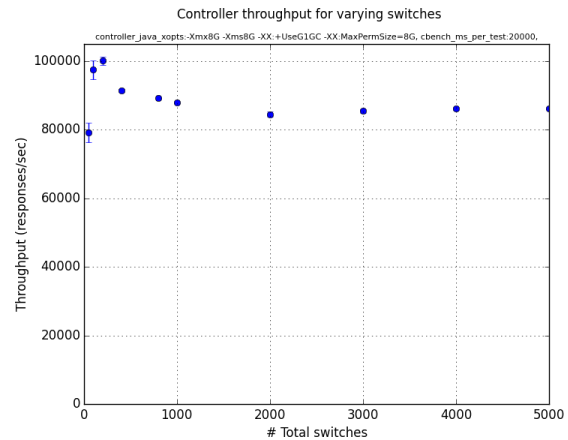


Fig. 3: Switch scalability with active MT-Cbench switches (RPC mode)

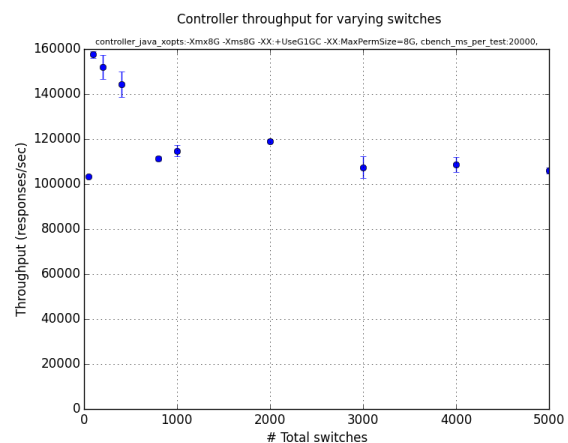


Fig. 4: Switch scalability with active MT-Cbench switches (RPC mode, "Lithium-design" OpenFlow plugin)

4.1.1 Test configuration, "RPC" mode

- controller: "RPC" mode
- generator: MT-Cbench, latency mode
- number of MT-Cbench threads: 1, 2, 4, 8, 16, 20, 40, 60, 80, 100 threads,
- topology size per MT-Cbench thread: 50 switches
- traffic generation interval: 20s

In this case, the total topology size is equal to 50, 100, 200, 400, 800, 1000, 2000, 3000, 4000, 5000.

4.1.2 Test configuration, "RPC" mode, "Lithium-design" OpenFlow plugin

- controller: "RPC" mode, "Lithium-design" OpenFlow plugin
- generator: MT-Cbench, latency mode
- number of MT-Cbench threads: 1, 2, 4, 8, 16, 20, 40, 60, 80, 100 threads,
- topology size per MT-Cbench thread: 50 switches
- traffic generation interval: 20s

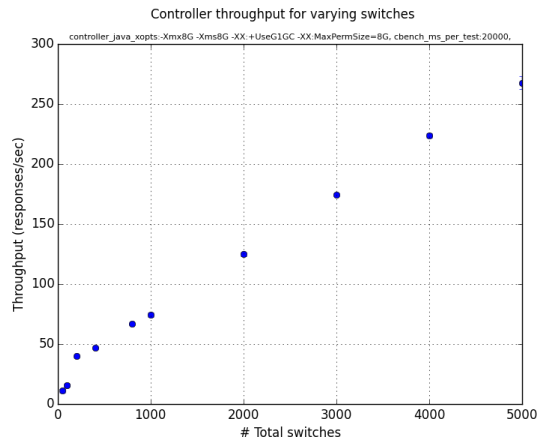


Fig. 5: Switch scalability with active MT-Cbench switches (DataStore mode)

In this case, the total topology size is equal to 50, 100, 200, 400, 800, 1000, 2000, 3000, 4000, 5000.

4.1.3 Test configuration, "DataStore" mode

- controller: "DataStore" mode
- generator: MT-Cbench, latency mode,
- number of MT-Cbench threads: 1, 2, 4, 8, 16, 20, 40, 60, 80, 100 threads,
- topology size per MT-Cbench thread: 50 switches
- group delay: 15s
- trafficgeneration interval: 20s

In this case, the total topology size is equal to 50, 100, 200, 300, 400, 800, 1000, 2000, 3000, 4000, 5000.

4.1.4 Test configuration, "DataStore" mode, "Lithium-design" OpenFlow plugin

- controller: "DataStore" mode, "Lithium-design" OpenFlow plugin
- generator: MT-Cbench, latency mode,
- number of MT-Cbench threads: 1, 2, 4, 8, 16, 20, 40, 60, 80, 100 threads,
- topology size per MT-Cbench thread: 50 switches
- group delay: 15s
- trafficgeneration interval: 20s

In this case, the total topology size is equal to 50, 100, 200, 300, 400, 800, 1000, 2000, 3000, 4000, 5000.

4.1.5 Native vs virtualized execution

In this section we investigate the impact of virtualization on the performance observed for the switch scalability test case with active MT-Cbench. Two different scenarios are evaluated:

- all components (controller, MT-Cbench, NSTAT) running directly on bare metal

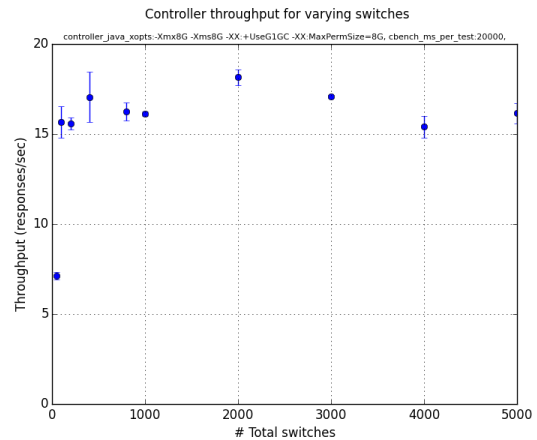


Fig. 6: Switch scalability with active MT-Cbench switches (DataStore mode, "Lithium-design" OpenFlow plugin)

- all components running within an virtual machine

The configuration of the virtual machine was the following:

- Hypervisor: VirtualBox 5.0.10
- Guest OS: Ubuntu 14.04 LTS, kernel 3.13.0
- VM resources 16 vCPUs, 32GB RAM

On both scenarios, the same JVM parameters were used¹. Also, in native execution all components were confined on 16 physical CPUs in order to have the same resources with the virtualized scenario.

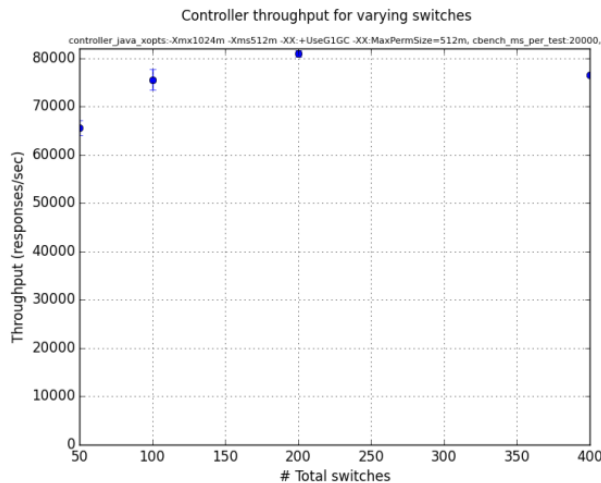
NSTAT configuration

- controller: "RPC" mode, "Lithium-design" OpenFlow plugin
- generator: MT-Cbench, latency mode
- number of MT-Cbench threads: 1, 2, 4, 8 threads,
- topology size per MT-Cbench thread: 50 switches
- trafficgeneration interval: 20s
- group delay: 15s

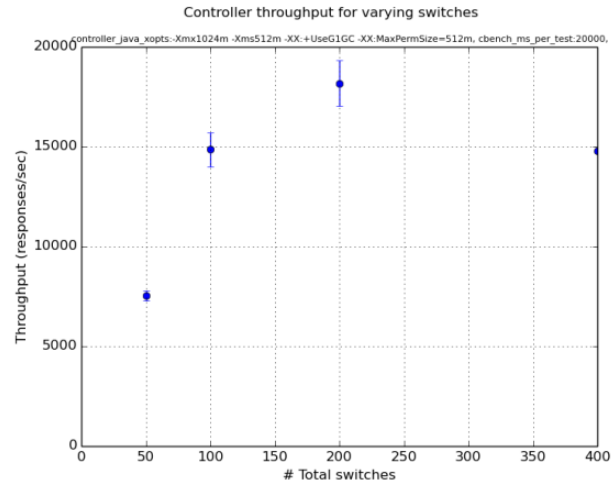
As we can see, controller suffers a significant performance degradation in the virtualized case. In an effort to bridge this performance discrepancy we performed additional tests, assigning more vCPUs and memory to the VM, but without notable improvements. As the virtualized execution involves all components running within the same VM, without needing to perform any kind of inter-VM communication, we speculate that this overhead could be attributed to pure compute virtualization overheads.

In future releases of this report we intend to perform extensive experiments to assess the impact of virtualization, by evaluating additional hypervisors, different I/O virtualization tech-

¹JVM parameters: [-Xmx1024m, -Xms512m, -XX:+UseG1GC, -XX:MaxPermSize=512m]



(a) Native execution



(b) Virtualized execution

Fig. 7: Throughput value.

nologies, and different containment options (e.g controller-only virtualization, generator-only virtualization, etc.). The value of this investigation will be twofold: first, we will better understand the sensitivities of each test scenario with respect to virtualization overheads, and second, we will have a comprehensive comparison of different virtualization technologies and options as regards performance.

4.2 Idle Mininet switches

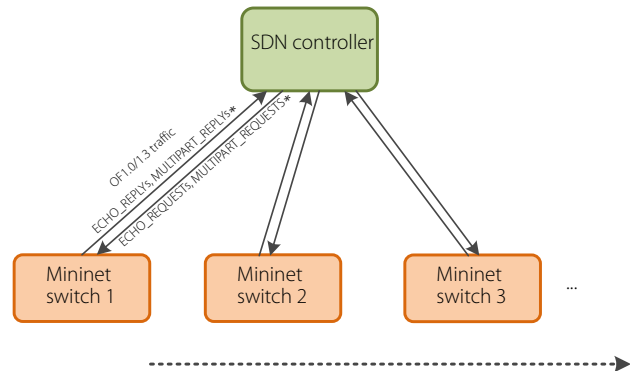
This is a switch scalability test with idle switches emulated using Mininet. The objectives of this test are: first, to find the largest number of idle switches the controller can accept and maintain. Second, to find the combination of boot-up-related configuration keys that leads to the fastest successful network boot-up. Specifically, these keys are the group size at which switches are being connected, and the delay between each group. We consider a boot-up as successful when all network switches have become visible in the operational data store of the controller. NSTAT reports the number of switches finally discovered by the controller, and the discovery time.

During main test execution Mininet switches respond to **ECHO** and **MULTIPART** messages sent by the controller at regular intervals. These types of messages dominate the total traffic volume during execution.

Mininet switches are being manipulated via custom handling logic. The topology types currently supported are

- Disconnected
- Linear
- Mesh

In order to push the controller performance to its limits, the controller node was executed on bare metal. Mininet was ex-



* the vast majority of packets

Fig. 8: Representation of switch scalability stress test with idle Mininet switches.

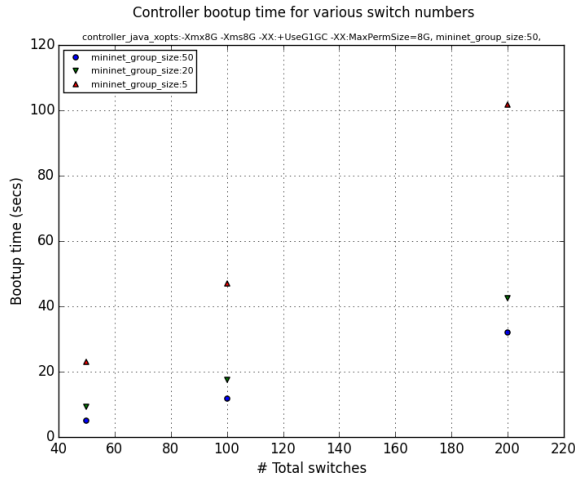
ecuted within a virtual machine.

4.2.1 Disconnected and Linear topologies

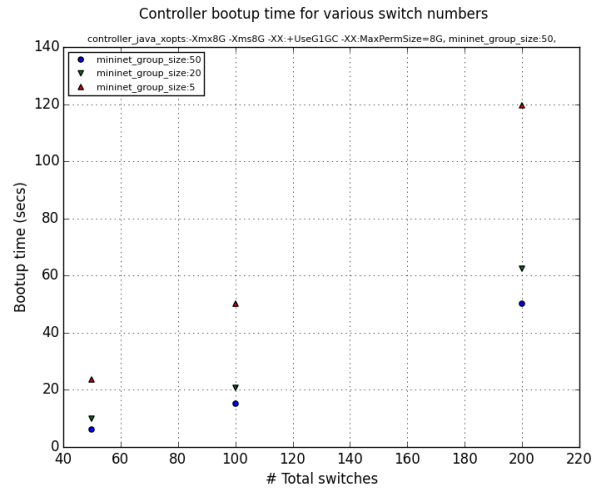
- topology size (number of topology switches), [50, 100, 200]
- Group size: [50, 20, 5]
- Group delay: [100, 500, 2000] msec

4.2.2 Mesh topology

- topology size (number of topology switches), [10, 15, 20, 25]
- Group size: [1]
- Group delay: [100, 500] msec



(a) Disconnected topology



(b) Linear topology

Fig. 9: Switch scalability test with idle Mininet switches. Boot-up time Vs number of network switches

4.3 Idle MT-Cbench switches

This is a switch scalability test with idle switches emulated using MT-Cbench, Fig.10. As in the Multinet case, the goal is to explore the maximum number of idle switches the controller can sustain, and how a certain-sized topology should be optimally connected to it.

In contrast to idle Multinet switches that exchange ECHO and MULTIPART messages with the controller, MT-Cbench switches typically sit idle during main operation, without sending or receiving any kind of messages. Due to this fact, the ability of the controller to accept and maintain MT-Cbench switches is expected to be much larger than the case of using realistic OpenFlow switches.

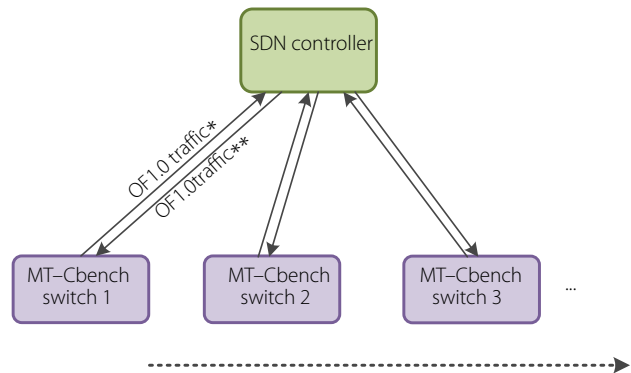
In order to push the controller performance to its limits, all test nodes (controller, MT-Cbench) were executed on bare metal. To isolate the nodes from each other, the CPU shares feature of NSTAT was used, [8].

4.3.1 RPC mode

- Controller: RPC mode
- MT-Cbench generator: Latency mode, 50 switches per thread, [1, 2, 4, 8, 16, 20, 30, 40, 50, 60, 70, 80, 90, 100] threads, inter-thread creation delay [500, 1000, 2000, 4000, 8000, 16000] ms.

4.4 Conclusions

In active MT-Cbench switch scalability tests the throughput in "RPC mode" is saturated at about 85k-90k reponses/s for which counts larger than 500. The same trend is observed with the "Lithium-design" OF plugin as well, but now the throughput sees a notable improvement, ranging between 85k-90k



*, ** a few messages during initialization, idle during main execution.

Fig. 10: Representation of switch scalability stress test with active MT-Cbench switches.

reponses/s.

In "Datastore mode" tests, the controller can achieve the same switch counts without problem, but a significantly lower throughput which scales linearly with the number of switches. This is expected since the controller datastore is being involved in this test case, which adds to the critical path of the processing pipeline of a packet. In contrast with the enhanced behaviour observed in "RPC mode", the "Lithium-design" OF plugin in "Datastore mode" does not exhibit better or scalable throughput, restricting it below 20 responses/sec for every case.

In idle Mininet switch scalability tests the main differentiation is among the Disconnected-Linear and Mesh topologies. In the first two topology types NSTAT can successfully boot switches up to 200 without problems. The discovery times vary significantly with the switch group size, yet the controller succeeds in discovering all switches for every boot-up parameter. This implies that it could perform even better, that is, with

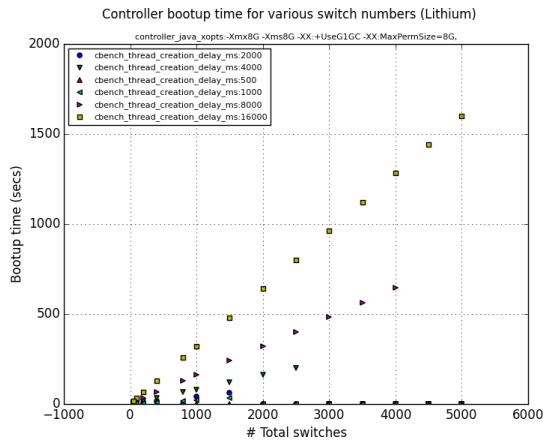


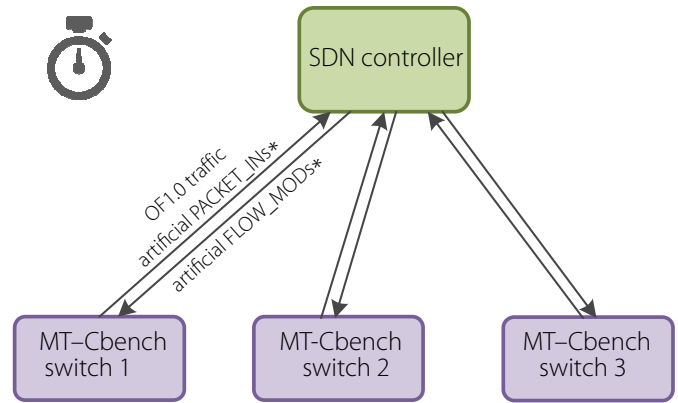
Fig. 11: Switch scalability with idle MT-Cbench (RPC mode).

switches being added at a faster pace. Booting larger topologies using Mininet was impractical. For this purpose, NSTAT is currently in the process of integrating Multinet [9] as an effective alternative for booting large scale Mininet topologies in a fast and resource efficient manner. This will give us the opportunity to perform tests with switch counts in the order of thousands. These results will be presented in a future release of this report. In mesh topologies NSTAT was able to run scalability tests, for much lower counts. Indicatively, we present results for up to 25 switches. For larger counts, booting the Mininet topology was impractical due to the exponential link growth.

In idle MT-Cbench switch scalability tests we can observe more clearly the impact of boot-up related parameters to switch scalability. Specifically, as the delay between switch addition grows ("cbench_thread_creation_delay_ms"), the controller is able to discover more switches, presumably because it is given more time to perform the necessary registration actions in its data store. This is a behavior we expect to see, in idle Mininet tests in the future, using large scale network emulation as we mentioned above. In any case, we stress once more the value of these results: not only we can tell the absolute switch limit for a certain case, but we can also derive the group size/group delay parameters that will give the optimal boot-up time.

5. Stability tests

Stability tests explore how controller throughput behaves in a large time window with a fixed topology connected to it. The goal is to detect performance fluctuations over time. The controller accepts a standard rate of incoming traffic and its response throughput is being sampled periodically. NSTAT reports these samples in a time series.



* the vast majority of packets

Fig. 12: Controller stability stress test with active MT-Cbench switches

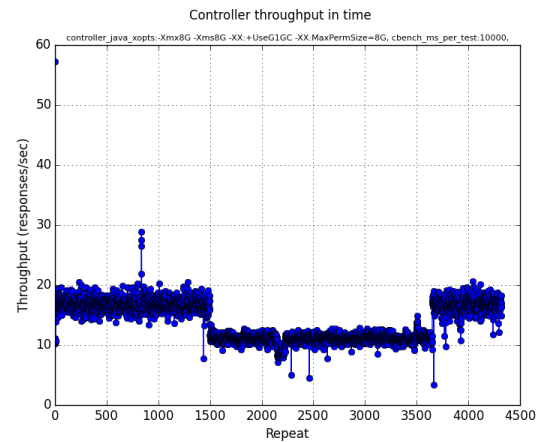


Fig. 13: Controller stability with active MT-Cbench switches (DataStore mode)

5.1 Active MT-Cbench switches

In the series of experiments NSTAT uses a fixed topology of active MT-Cbench switches to generate traffic. The switches send artificial OpenFlow 1.0 Packet-In messages at a fixed rate to the controller, which replies with also artificial OF1.0 Flow-Mod messages. These message types dominate the traffic exchanged between the switches and the controller. We evaluate the controller both in "RPC" and "Datastore" modes.

In order to push the controller performance to its limits, all test nodes (controller, MT-Cbench) were executed on bare metal. To isolate the nodes from each other, the CPU shares feature of NSTAT was used.

5.1.1 "DataStore" mode, 12 hours running time

- controller: "DataStore" mode
- generator: MT-Cbench, latency mode
- number of MT-Cbench threads: 10

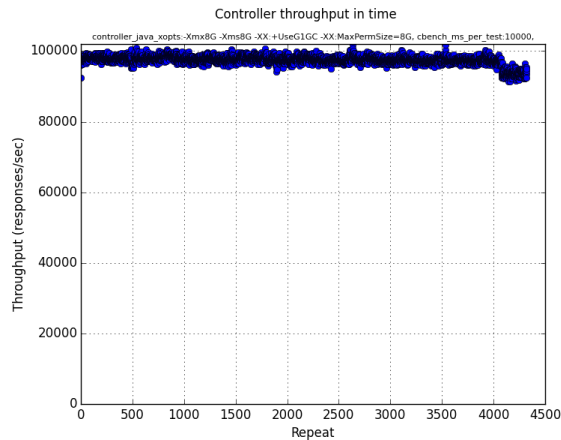


Fig. 14: Controller stability with active MT-Cbench switches (RPC mode)

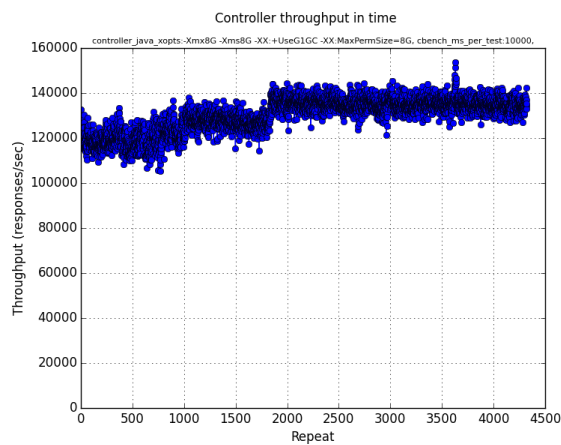


Fig. 15: Controller stability with active MT-Cbench switches (RPC mode, "Lithium-design" OF plugin)

- topology size per MT-Cbench thread: 50 switches
- group delay: 8s
- number of samples: 4320
- period between samples: 10s
- total running time: 12h

In this case, the total topology size is equal to 500 switches.

5.1.2 "RPC" mode, 12 hours running time

- controller: "RPC" mode
- generator: MT-Cbench, latency mode
- number of MT-Cbench threads: 10
- topology size per MT-Cbench thread: 50 switches
- group delay: 8s
- number of samples: 4320
- period between samples: 10s
- total running time: 12h

In this case, the total topology size is equal to 500 switches.

5.1.3 "RPC" mode, "Lithium Design", OpenFlow plugin 12 hours running time

- controller: "RPC" mode, OpenFlow plugin
- generator: MT-Cbench, latency mode
- number of MT-Cbench threads: 10
- topology size per MT-Cbench thread: 50 switches
- group delay: 8s
- number of samples: 4320
- period between samples: 10s
- total running time: 12h

In this case, the total topology size is equal to 500 switches.

5.2 Conclusions

The OpenDaylight Lithium SR3 release exhibits in general stable sustained throughput within a large 12-hour time window. This is a clear improvement over previous releases (e.g Helium) where controller response throughput was steadily degrading, to finally reach zero values. The different performance levels observed in each of the "DataStore mode" and "RPC mode/Lithium-design" cases should be attributed probably to periodic external system loads. The "Lithium-design" OF plugin implementation performs stably and with improved throughput, as we have seen in the switch scalability tests.

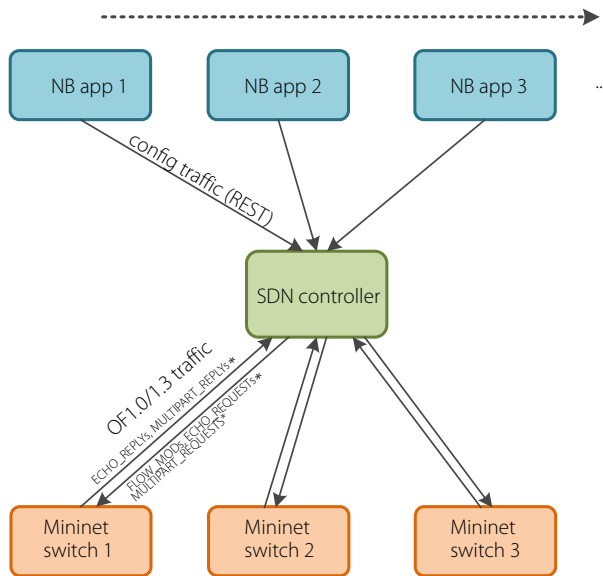
6. Flow scalability tests

In flow scalability stress tests an increasing number of NB clients install flows on the switches of an underlying OpenFlow switch topology. With this test one can investigate both capacity and timing aspects of flows installation via the controller NB (REST-CONF) interface. Specifically the target metrics that can be explored are:

- maximum number of flows the controller can handle from the NB interface
- time required to install a certain number of flows (or alternatively, the flow installation throughput)

This test uses the Northbound flow generator [10] to create flows in a scalable and configurable manner (number of flows, delay between flow creation). To emulate conditions of an ever increasing stressing towards the controller NB interface, the generator issues flows concurrently from multiple worker threads, which mimic real-world NB clients. The number of these workers is also configurable.

The flows are being written to the controller configuration datastore via its NB interface, and then forwarded to an underlying OpenFlow switch topology as flow modifications. The test verifies that the specified number of flows have been successfully installed on the switches by checking the controller's operational datastore.



* the vast majority of packets

Fig. 16: Representation of flow scalability stress test. An increasing number of NB clients (NB app, $j=1,2,\dots$) install flows on the switches of an underlying OpenFlow switch topology.

6.1 Idle multinet switches

This is a flow scalability test with flows being installed on an idle Mininet topology. A varying number of NB clients send configuration traffic as described in the previous section. The mininet switches do not initiate any traffic, but respond with **ECHO** and **MULTIPART** messages sent by the controller at regular intervals. These types of messages dominate the total SB traffic volume during execution. In order to push the controller performance to its limits, the controller and NB generator nodes were executed on bare metal. Mininet was executed within a virtual machine.

6.1.1 10 switches

- Mininet topology: 10 switches, linear, 1 host per switch
- Total flows to be added: [1000, 10000, 100000]
- Flow creation delay: [0,5,10] *ms*
- Flow worker threads: [1,10,20]

6.1.2 10 switches, "Lithium-design" OpenFlow plugin

- Controller: "Lithium-design" OpenFlow plugin
- Mininet topology: 10 switches, linear, 1 host per switch
- Total flows to be added: [1000, 10000, 100000]
- Flow creation delay: [0,5,10] *ms*
- Flow worker threads: [1,10,20]

6.1.3 100 switches

- Mininet topology: 100 switches, linear, 1 host per switch
- Total flows to be added: [1000, 10000, 100000]

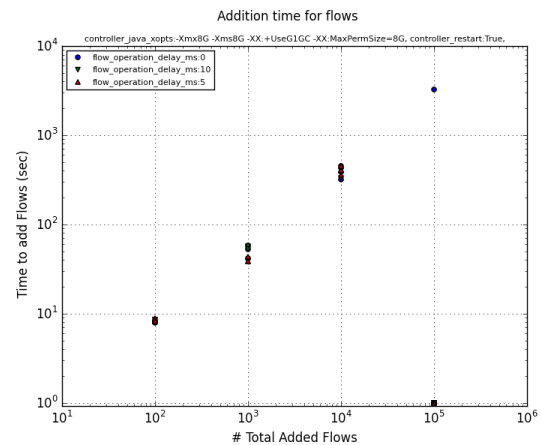


Fig. 17: Flow scalability with idle Multinet switches (10 switches)

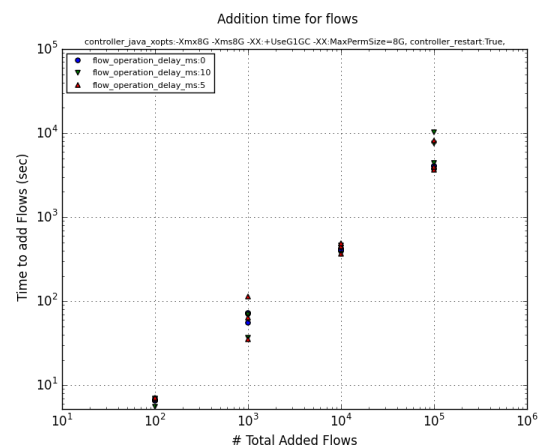


Fig. 18: Flow scalability with idle Multinet switches (10 switches, "Lithium-design" OF plugin)

- Flow creation delay: [0,5,10] *ms*
- Flow worker threads: [1,10,20]

6.2 Conclusions

A first conclusion is that the number of underlying switches clearly affects the flows installation time. In general, the 10 switches topology requires much less time for installation of the same number of flows as compared to the 100 switches topology, which can be even a half of the latter. This is explained by the additional complexity introduced to the controller when it has to manage a larger number of switches. In the future we will perform more detailed tests to investigate how exactly flow installation time scales with the topology size.

A second observation is that with the "Lithium-design" OF plugin there were more successful installation of 100k flows, which witnesses enhanced ability of the new plugin implementation to forward flows to the switches more efficiently.

In both 10 and 100 switches cases the flow installation times

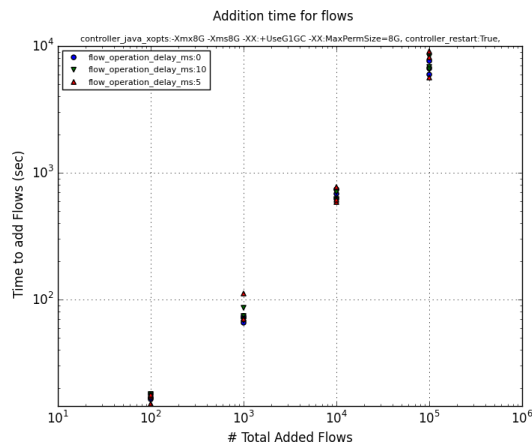


Fig. 19: Flow scalability with idle Multinet switches (10 switches)

generally scale linearly with the number of flows. It does not seem to be a clear trend on how a certain flow count can be optimally installed, yet a rule of thumb derived from most cases is that few workers with small flow creation delay yield the best installation times.

7. Contributors

- Nikos Anastopoulos
- Panagiotis Georgiopoulos
- Konstantinos Papadopoulos

References

- [1] "NSTAT: Network Stress Test Automation Toolkit." <https://github.com/intracom-telecom-sdn/nstat>.
- [2] "MT-Cbench: A multithreaded version of the Cbench OpenFlow traffic generator." <https://github.com/intracom-telecom-sdn/mtcbench>.
- [3] "Mininet. An instant virtual network on your Laptop." <http://mininet.org/>.
- [4] "Cbench: OpenFlow traffic generator." <https://github.com/andi-bigswitch/oflops/tree/master/cbench>.
- [5] "Mininet custom topologies." <https://github.com/intracom-telecom-sdn/nstat/wiki/Mininet-custom-topologies>.
- [6] "OpenDaylight OpenFlow Plugin:Lithium Design Proposal." https://wiki.opendaylight.org/view/OpenDaylight_OpenFlow_Plugin:Lithium_Design_Proposal.
- [7] "OpenDaylight OpenFlow Plugin:He vs Li comparison." https://wiki.opendaylight.org/view/OpenDaylight_OpenFlow_Plugin:He_vs_Li_comparison.

- [8] "Capping controller and emulator CPU resources in collocated tests." <https://github.com/intracom-telecom-sdn/nstat/wiki/Capping-controller-and-emulator-CPU-resources-in-collocated-tests>.
- [9] "Multinet: Large-scale SDN emulator based on Mininet." <https://github.com/intracom-telecom-sdn/multinet>.
- [10] "NSTAT: NorthBound Flow Generator." <https://github.com/intracom-telecom-sdn/nstat-nb-generator>.