OpenDaylight Performance Stress Test Report

v1.2: Lithium vs Beryllium

Intracom Telecom SDN/NFV Lab

www.intracom-telecom.com | intracom-telecom-sdn.github.com | sdn-nfv@intracom-telecom.com

OpenDaylight Performance Stress Tests Report v1.2: Lithium vs Beryllium

Executive Summary

In this report we investigate several performance and scalability aspects of the OpenDaylight Beryllium controller and compare them against the Lithium release. Beryllium outperforms Lithium in idle switch scalability tests both with Multinet (realistic OF1.3) and MT–Cbench (artificial OF1.0) emulators. A single Beryllium instance can boot "Linear" and "Disconnected" idle Multinet topologies of up to 6400 switches, as opposed to 4800 with Lithium. In general, Beryllium manages to successfully boot topologies that Lithium fails to, or it can boot them faster. In MT–Cbench tests the superiority of Beryllium is even more evident, being able to successfully discover topologies that expose themselves to the controller at a faster rate as compared to Lithium. With regard to controller stability, both releases exhibit similarly stable behavior. In addition, Multinet–based tests reveal traffic optimizations in Beryllium, witnessed by the reduced volume related to MULTIPART messages. Finally, in NorthBound performance tests Beryllium seems to underperform in accepting flow installation requests via RESTCONF, but as far as the end–to–end flow installation is concerned it is the clear winner. In many extreme cases of large topologies (e.g. 5000 switches) or large flow counts (1M), Beryllium manages to successfully install flows while Lithium fails.

Contents

1. Introduction

In this report we investigate several performance and scalability aspects of the OpenDaylight Beryllium controller and compare them against the Lithium release. The investigation focuses on both the SouthBound (SB) and NorthBound (NB) controller's interface and targets the following objectives:

- controller throughput,
- switch scalability,
- controller stability (sustained throughput),
- flow scalability and flow 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. NSTAT's architecture is exhibited in Fig.1 and its key components are the

- SDN controller,
- SouthBound OpenFlow traffic generator,
- NorthBound flow generator (RESTCONF traffic).

The SDN controller used in this series of experiments is OpenDaylight. We perform and demonstrate results from the Lithium SR3 and Beryllium RC2 releases.

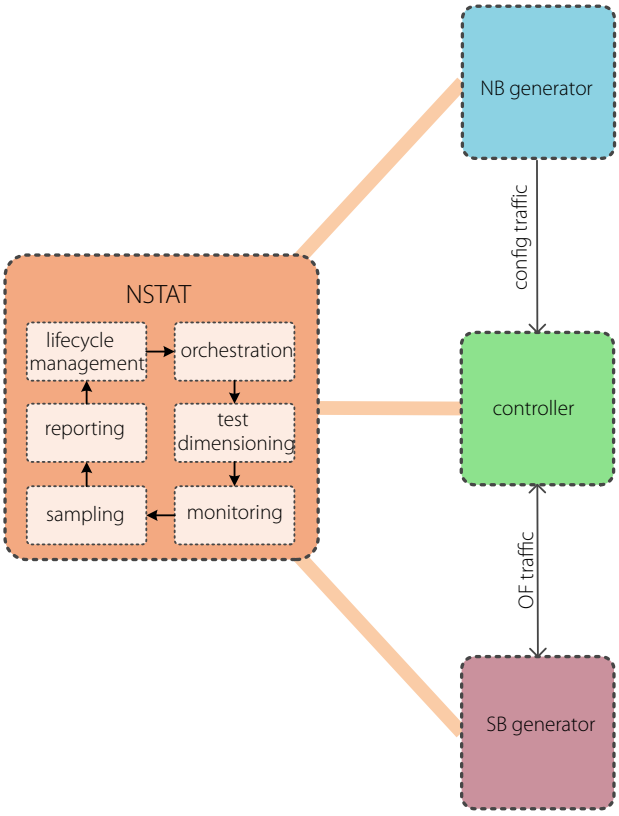


Fig. 1: NSTAT architecture

For SouthBound (SB) traffic generation we have used both MT–Cbench [2] and Multinet [3]. For NorthBound (NB) traffic generation NSTAT uses custom scripts [4], originally developed by the OpenDaylight community [5], for creating and writing flows to the controller configuration data store.

MT–Cbench is a direct extension of the Cbench [6] emulator which uses threading to generate OpenFlow traffic from multiple streams in parallel. The motivation behind this extension was to be able to boot–up and operate network topologies 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 OF1.0 protocol, and therefore the results presented for that generator are expected to vary in real–world deployments.

This gap is largely filled using Multinet [7] as an additional SB traffic generator, as it uses OVS virtual switches that accurately emulate the OF1.3 protocol. The goal of Multinet is to provide a fast, controlled and resource efficient way to boot large–scale SDN topologies. This is achieved by booting in parallel multiple isolated Mininet topologies, each launched on a separate virtual machine, and all connected to the same controller[1]. Finally, by providing control on the way that switches are being connected to the SDN controller, one can easily test various switch group size/delay combinations and discover configurations that yield optimal boot–up times for a large–scale topology.

In our stress tests we have experimented with emulated switches operating in two modes, idle and active mode: switches in **idle mode** do not initiate any traffic to the controller, but rather respond to messages sent by it. Switches in **active mode** consistently initiate traffic to the controller, in the form of PACKET_IN messages. In most stress tests, MT–Cbench and Multinet switches operate both in active and idle modes.

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.

NSTAT is responsible for automating and sequencing every step of the testing procedure, including building and deploying components on their corresponding 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 etc., and in a sense these define the "dimensions" of a stress test scenario.

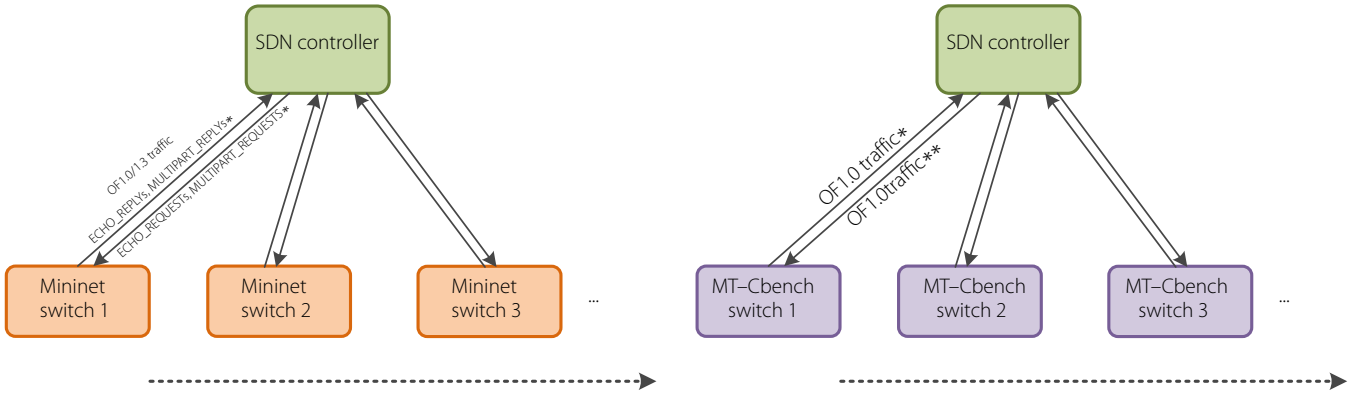
One of NSTAT's key features is that it allows the user to specify multiple values for one or more such dimensions at the same time, and the tool itself takes care to repeat the test over all their combinations in a single session. This kind of exhaustively exploring the multi–dimensional experimental 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

Details of the experimental setup are presented in Table 1.

As mentioned in section 1, in our stress tests we have experimented with emulated switches operating in idle and active

[1] As an example, Multinet can create more than 3000 Openvswitch OF1.3 switches over moderate virtual resources (10 vCPUs, <40GB memory) in less than 10 minutes.



* the vast majority of packets

(a) Multinet switches sit idle during main operation. ECHO and MULTIPART messages dominate the traffic being exchanged between the controller and the switches.

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

(b) MT–Cbench switches sit idle during main operation, without sending or receiving any kind of messages.

Fig. 2: Representation of a switch scalability test with idle Multinet and MT-Cbench switches.

Table 1: Stress tests experimental setup.

Host operating system	Centos 7, kernel 3.10.0
Hypervisor	QEMU v.2.0.0
Guest operating system	Ubuntu 14.04
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
SDN Controllers under test	OpenDaylight Beryllium (RC2), OpenDaylight Lithium (SR3)
OpenDaylight configuration	OF plugin implementation: "Helium design"
Controller JVM options	-Xmx8G, -Xms8G, -XX:+UseG1GC, -XX:MaxPermSize=8G
Multinet OVS version	2.0.2

mode using both MT–Cbench and Multinet for emulating the network topology.

In MT–Cbench tests the controller is 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.

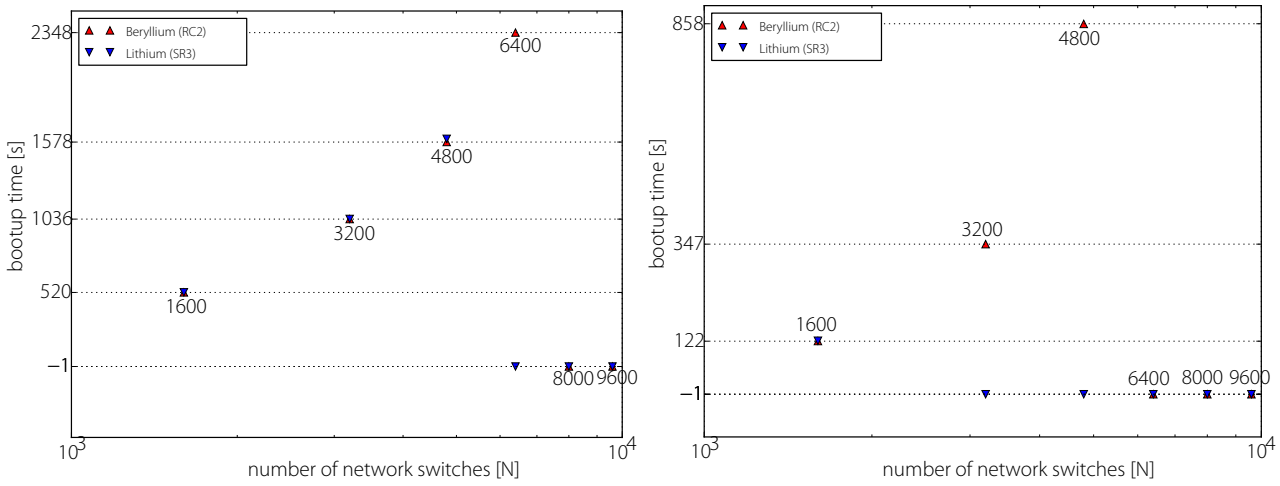
Two different configurations of the controller are evaluated with MT–Cbench active switches: in "RPC" mode, the controller is configured to directly reply to PACKET_IN's sent by the switches with a predefined message at the OpenFlow plugin level. In "DataStore" mode, the controller additionally performs 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 all stress tests the "Helium–design" implementation of the OpenFlow plugin was used. In future versions of this report we will evaluate the new implementation, codenamed "Lithium–design".

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:

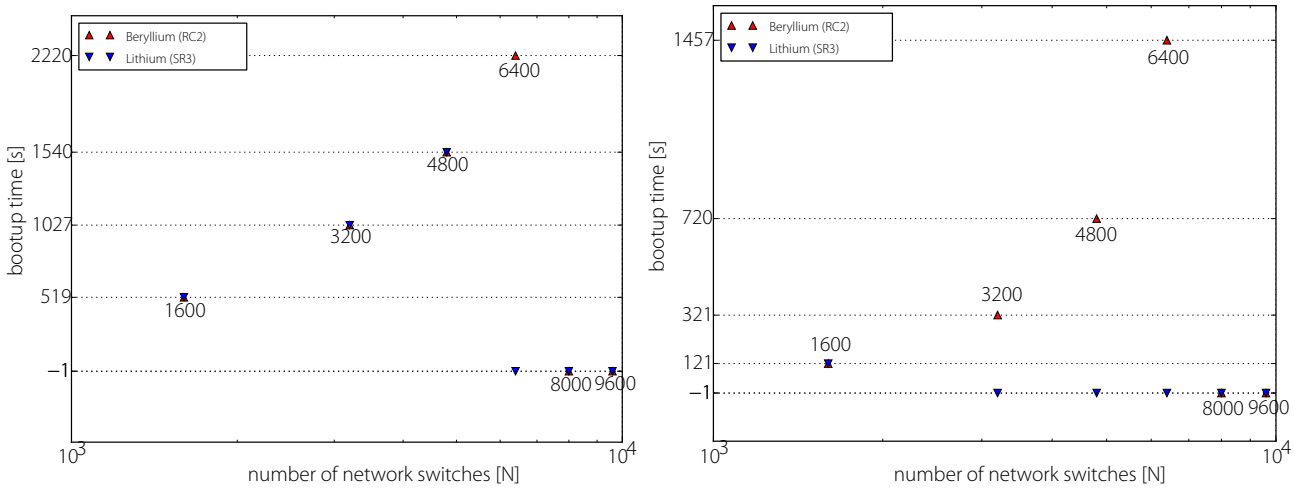
- in idle mode tests, NSTAT also reports the topology boot–up time for different policies of connecting switches to the controller. From these results we can deduce how to optimally boot–up a certain–sized topology and connect it to the controller, so that the latter can successfully discover it at the 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.



(a) Group size: 1, topology type: Linear

(b) Group size: 5, topology type: Linear

Fig. 3: Switch scalability stress test results for idle Multinet switches. Network topology size from 1600→9600 switches. Topology type: Linear. Boot–up time is forced to -1 when switch discovery fails.



(a) Group size: 1, topology type: Disconnected.

(b) Group size: 5, topology type: Disconnected.

Fig. 4: Switch scalability stress test results for idle Multinet switches. Network size scales from 1600→9600 switches. Topology type: Disconnected. Boot–up time is forced to -1 when switch discovery fails.

4.1 Idle Multinet switches

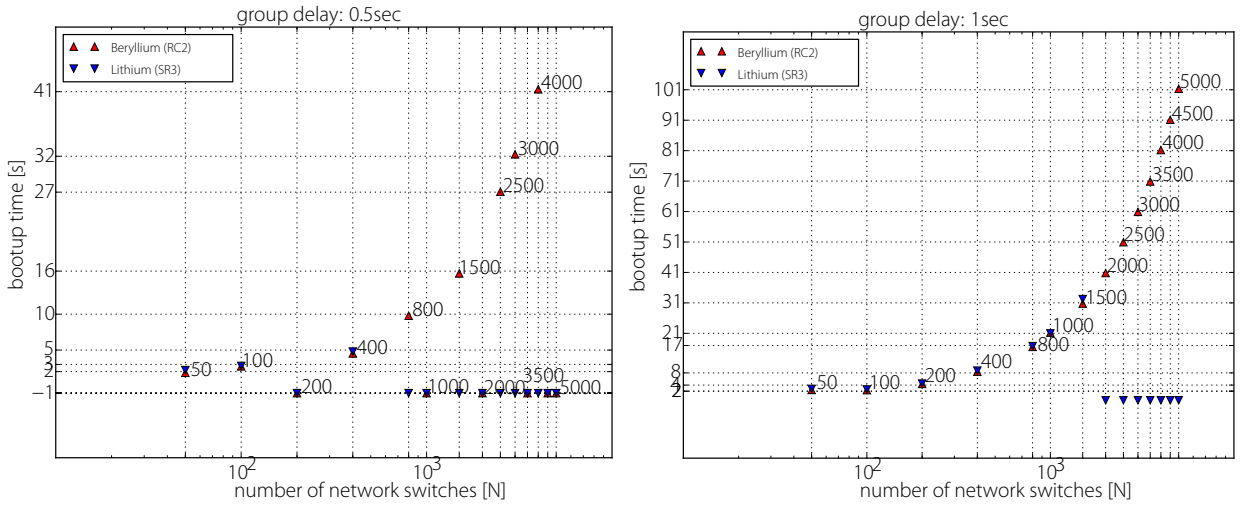
This is a switch scalability test with idle switches emulated using Multinet, Fig.2(a). The objective of this test is twofold:

- to find the largest number of idle switches the controller can accept and maintain.
- to find the combination of boot–up–related configuration options that leads to the fastest successful network boot–up.

Specifically, these options 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. If not all switches have been reflected in the data store within a certain deadline after the last update, we declare the boot–up as failed. NSTAT reports **the number of switches** finally discovered by the controller, and the **discovery time**.

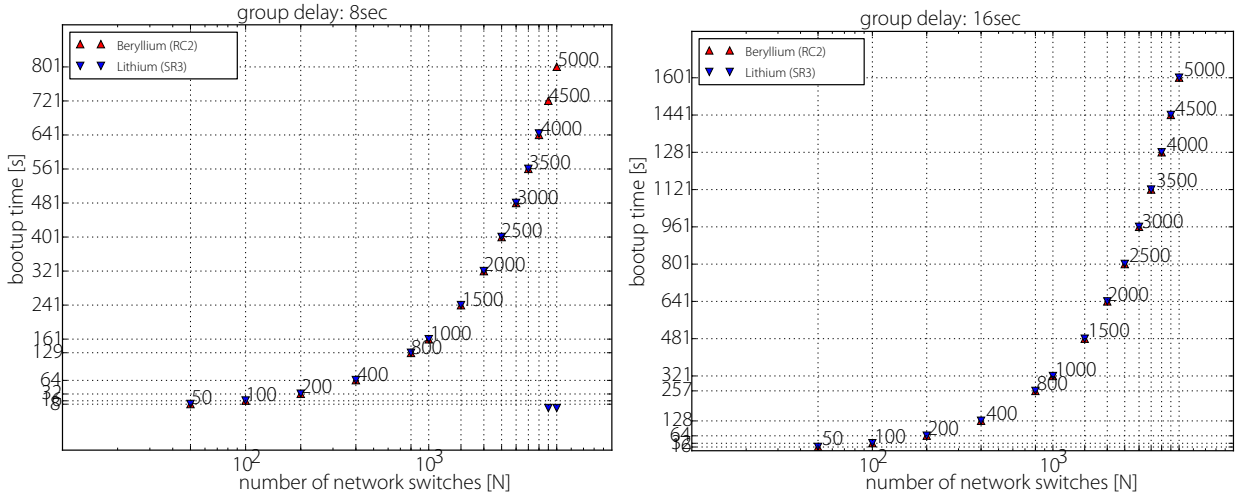
During main test execution Multinet switches respond to ECHO and MULTIPART messages sent by the controller at regular intervals. These types of messages dominate the total traffic volume during execution. We evaluate two different topology types, "Linear" and "Disconnected".

In order to push the controller performance to its limits, the controller node is executed on bare metal. Multinet is executed within a set of virtual machines, see master/worker functionality of Multinet, [8].



(a) group delay: 0.5s. (b) group delay: 1s.

Fig. 5: Switch scalability stress test results with idle MT–Cbench switches. Topology size scales from 50 → 5000 switches.



(a) group delay: 8s. (b) group delay: 16s.

Fig. 6: Switch scalability stress test results with idle MT–Cbench switches. Topology size scales from 50 → 5000 switches.

4.1.1 Test configuration

- topology types: "Linear", "Disconnected"
- topology size per worker node: 100, 200, 300, 400, 500, 600
- number of worker nodes: 16
- group size: 1, 5
- group delay: 5000ms
- persistence: enabled

In this case, the total topology size is equal to 1600, 3200, 4800, 6400, 8000, 9600.

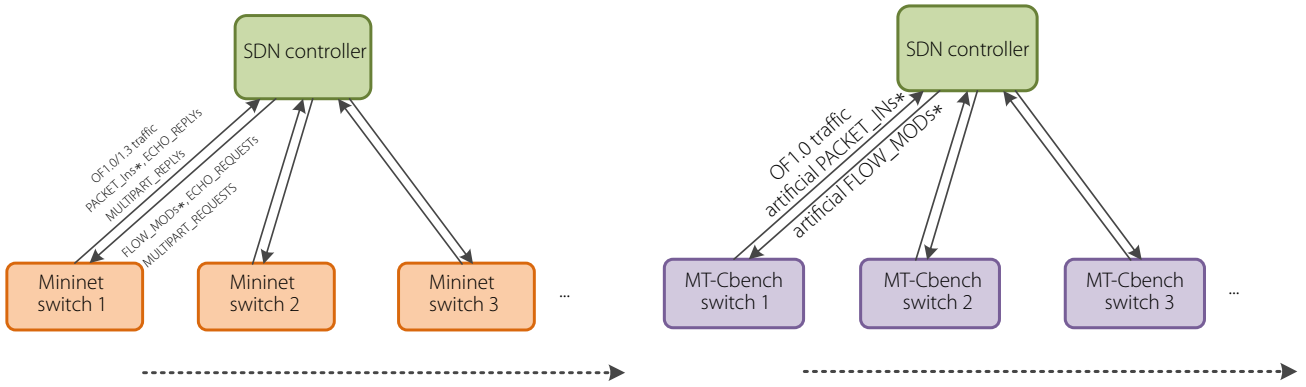
4.1.2 Results

Results from this series of tests are presented in Fig.3(a), 3(b) for a "Linear" topology and Fig.4(a), 4(b) for a "Disconnected" topology respectively.

4.2 Idle MT–Cbench switches

This is a switch scalability test with idle switches emulated using MT–Cbench, Fig.2(b). 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.



* the vast majority of packets

(a) The switches send OF1.3 PACKET_IN messages to the controller, which replies with OF1.3 FLOW_MODs.

* the vast majority of packets

(b) The switches send artificial OF1.0 PACKET_IN messages to the controller, which replies with also artificial OF1.0 FLOW_MOD messages.

Fig. 7: Representation of switch scalability stress test with active (a) Multinet and (b) MT–Cbench switches.

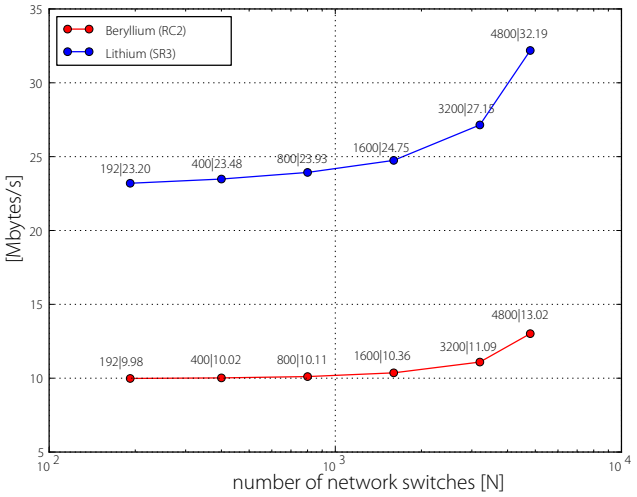


Fig. 8: Switch scalability stress test results with active Multinet switches. Comparison analysis of controller throughput variation Mbytes/s vs number of network switches N. Topology size scales from 192 → 4800 switches.

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, [9].

4.2.1 Test configuration

- controller: "RPC" mode
- generator: MT–Cbench, latency mode
- number of MT–Cbench threads: 1, 2, 4, 8, 16, 20, 30, 40, 50, 60, 70 , 80, 90, 100 threads.
- topology size per MT–Cbench thread: 50 switches
- group delay: 500, 1000, 2000, 4000, 8000, 16000.
- persistence: enabled

In this case switch topology size is equal to: 50, 100, 200, 300, 400, 800, 1000, 1500 2000, 2500, 3000, 3500, 4000, 4500, 5000.

4.2.2 Results

Results of this test are presented in Fig.5(a), 5(b), 6(a), 6(b).

4.3 Active Multinet switches

This is a switch scalability test with switches in active mode emulated using Multinet, Fig.7(a).

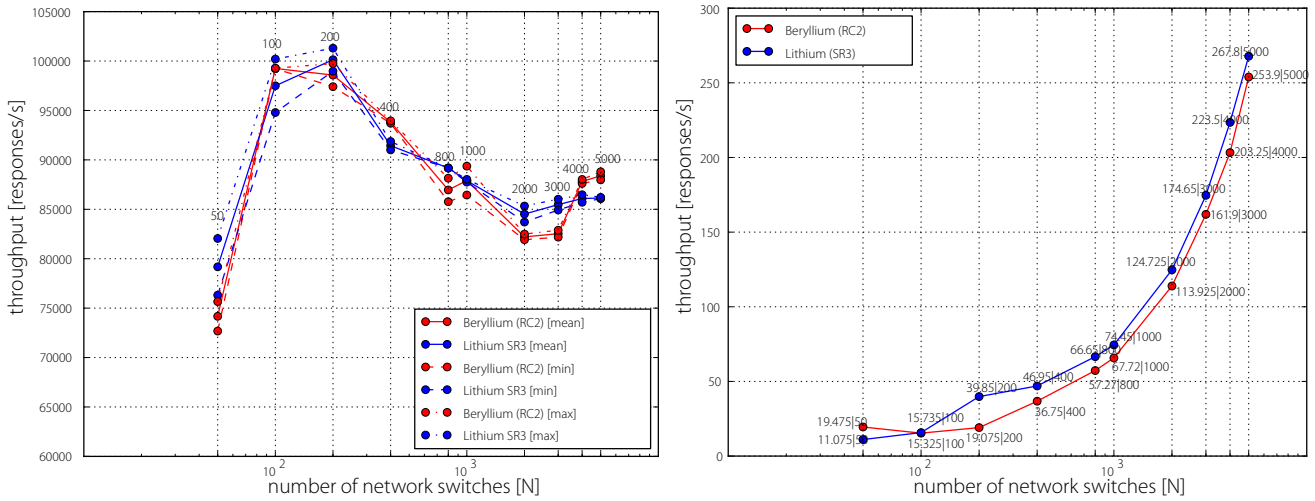
Its target is to explore the maximum number of switches the controller can sustain while they consistently initiate traffic to it (active), and how the controller servicing throughput scales as more switches are being added.

The switches start sending messages after the entire topology has been connected to the controller. They trigger OF1.3 PACKET_IN messages for a certain interval and at a configurable rate, and the controller replies with OF1.3 FLOW_MODs. These message types dominate the traffic exchanged between the switches and the controller. Our target metric is the **throughput** of the controller's outgoing OpenFlow traffic. NSTAT uses the oftraf [10] tool to measure it.

In order to push the controller performance to its limits, the controller is executed on the bare metal and Multinet on a set of interconnected virtual machines.

4.3.1 Test configuration

- controller: with l2switch plugin installed, configured to respond with mac–to–mac FLOW_MODs to PACKET_IN messages with ARP payload [11]
- topology size per worker node: 12, 25, 50, 100, 200, 300
- number of worker nodes: 16
- group size: 1
- group delay: 3000ms



(a) OpenDaylight running in RPC mode.

(b) OpenDaylight running in DS mode.

Fig. 9: Switch scalability stress test results with active MT–Cbench switches. Comparison analysis of controller throughput variation [responses/s] vs number of network switches N with OpenDaylight running both on RPC and DataStore mode.

- topology type: "Linear"
- hosts per switch: 2
- traffic generation interval: 60000ms
- PACKET_IN transmission delay: 500ms
- persistence: disabled

Switch topology size scales as follows: 192, 400, 800, 1600, 3200, 4800.

4.3.2 Results

Results of this test are presented in Fig.8.

4.4 Active MT–Cbench switches

This test is equivalent to the active Multinet switch scalability test described in section 4.3

The switches start sending messages after the entire topology has been connected to the controller. MT–Cbench switches send artificial OF1.0 PACKET_IN messages to the controller, which replies with also artificial OF1.0 FLOW_MOD messages; these are the dominant message types during execution. 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 [9] of NSTAT was used.

4.4.1 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
- group delay: 15s
- trafficgeneration interval: 20s
- persistence: enabled

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

4.4.2 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
- persistence: enabled

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

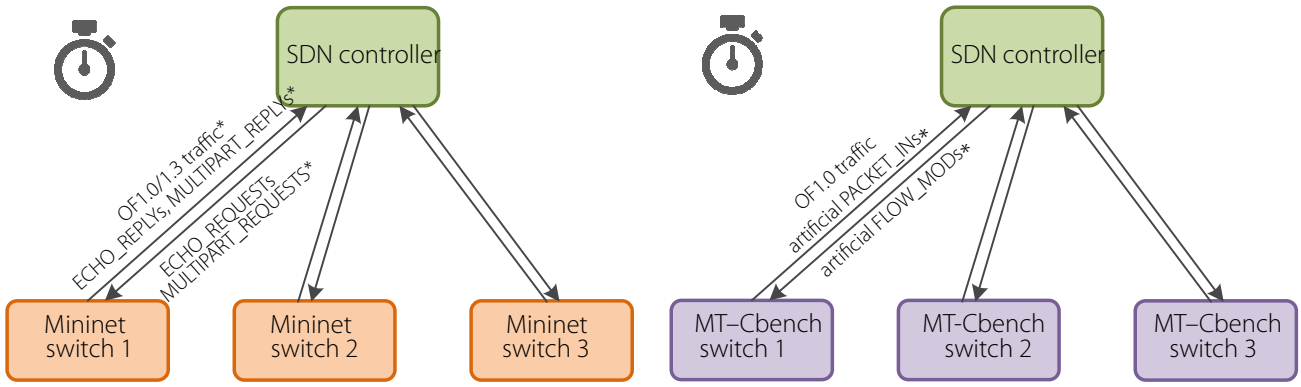
4.4.3 Results

Results for test configurations defined in sections 4.4.1, 4.4.2 are presented in Figs.9(a), 9(b) respectively.

4.5 Conclusions

4.5.1 Idle scalability tests

In idle Multinet switch scalability tests we can see that Beryllium controller has a significantly improved performance as it manages to handle larger switch topologies than Lithium. Beryllium can successfully boot "Linear" topologies of up to 6400 switches in groups of 1, and 4800 switches in groups



(a) Representation of switch stability stress test with idle Multinet switches.

(b) Representation of switch stability stress test with active MT–Cbench switches.

Fig. 10: Representation of switch stability stress test with idle Multinet (a) and active MT–Cbench switches (b). The controller accepts a standard rate of incoming traffic and its response throughput is being sampled periodically.

of 5, while Lithium can boot such topologies for up to 4800 switches regardless of the group size, Fig.3. For "Disconnected" topologies Beryllium manages to boot topologies of 6400 switches for both combinations of group size, Fig.4.

Thinking of the best boot–up times for a certain sized topology and for both topology types, we conclude that Beryllium can either successfully boot a topology that Lithium cannot, or it can boot it faster.

We note here that the boot–up times demonstrated in our tests are not necessarily the best possible that can be achieved for a certain topology size /type; testing with larger group sizes and/or smaller group delays could reveal further enhancements.

In idle MT–Cbench switch scalability tests the superiority of Beryllium is even clearer. As we can see, Beryllium is much more successful in discovering switches that expose themselves to the controller at a faster rate (lower group delay, Figs.5(a) and 5(b)). This performance difference disappears as the switches are being added less aggressively (Figs.6(a) and 6(b)). The Beryllium release can successfully discover almost the max number of switches even with a group delay of 0.5 seconds. In contrast, Lithium can achieve the same numbers with a delay of at least 8 seconds. Beryllium can generally boot large topologies much faster than Lithium.

4.5.2 Active scalability

In Multinet–based tests Lithium exhibits significantly higher throughput levels than Beryllium for every switch count. Nevertheless, this does not necessarily imply inferior performance, since, as we will show in section 5.3.1, traffic optimizations might have led to reduced volume overall. In any case, this scenario needs further analysis to better understand the difference, and will be a matter of future research.

In MT–Cbench based tests the performance gap between the two releases tends to close. In "RPC" mode the throughput is almost equivalent both for Lithium and Beryllium. In "DataStore" mode there is also equivalence, with a slight superiority of Lithium. In this case the throughput values are much lower, but this is expected as the controller data store is being intensively hit with the persistence module enabled. This adds to the critical path of the packet processing pipeline.

5. Stability tests

Stability tests explore how controller throughput behaves in a large time window with a fixed topology connected to it, Fig.10(a), 10(b). 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.

5.1 Idle Multinet switches

The purpose of this test is to investigate the stability of the controller to serve standard traffic requirements of a large scale Multinet topology of idle switches.

During main test execution Multinet 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.

NSTAT uses the oftraf [10] to measure the **outgoing traffic of the controller**. The metrics presented for this case are the

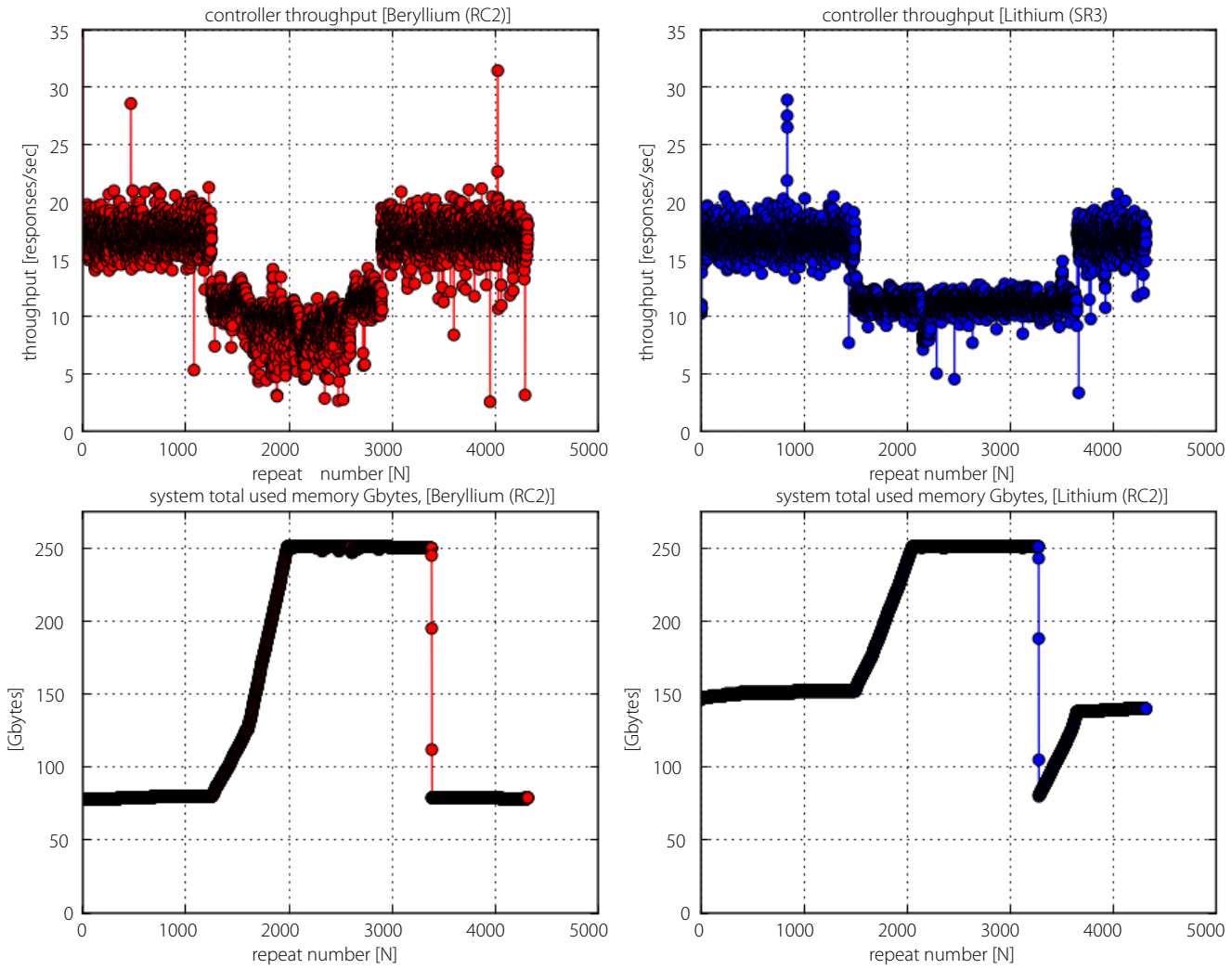


Fig. 11: Controller stability stress test with idle MT–Cbench switches. Throughput and system total used memory comparison analysis between OpenDaylight Lithium (SR3) and Beryllium (RC2) versions. OpenDaylight running in "DataStore" mode.

OpenFlow packets and bytes collected by NSTAT every second, Fig.13.

In order to push the controller performance to its limits, the controller is executed on the bare metal and Multinet on a set of interconnected virtual machines

5.1.1 Test configuration

- topology size per worker node: 200
- number of worker nodes: 16
- group size: 1
- group delay: 2000ms
- topology type: "Linear"
- hosts per switch: 1
- period between samples: 10s
- number of samples: 4320
- persistence: enabled
- total running time: 12h

In this case switch topology size is equal to 3200.

5.1.2 Results

The results of this test are presented in Fig.13

5.2 Active MT–Cbench switches

In this series of experiments NSTAT uses a fixed topology of active MT–Cbench switches to generate traffic for a large time window. The switches send artificial OF1.0 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

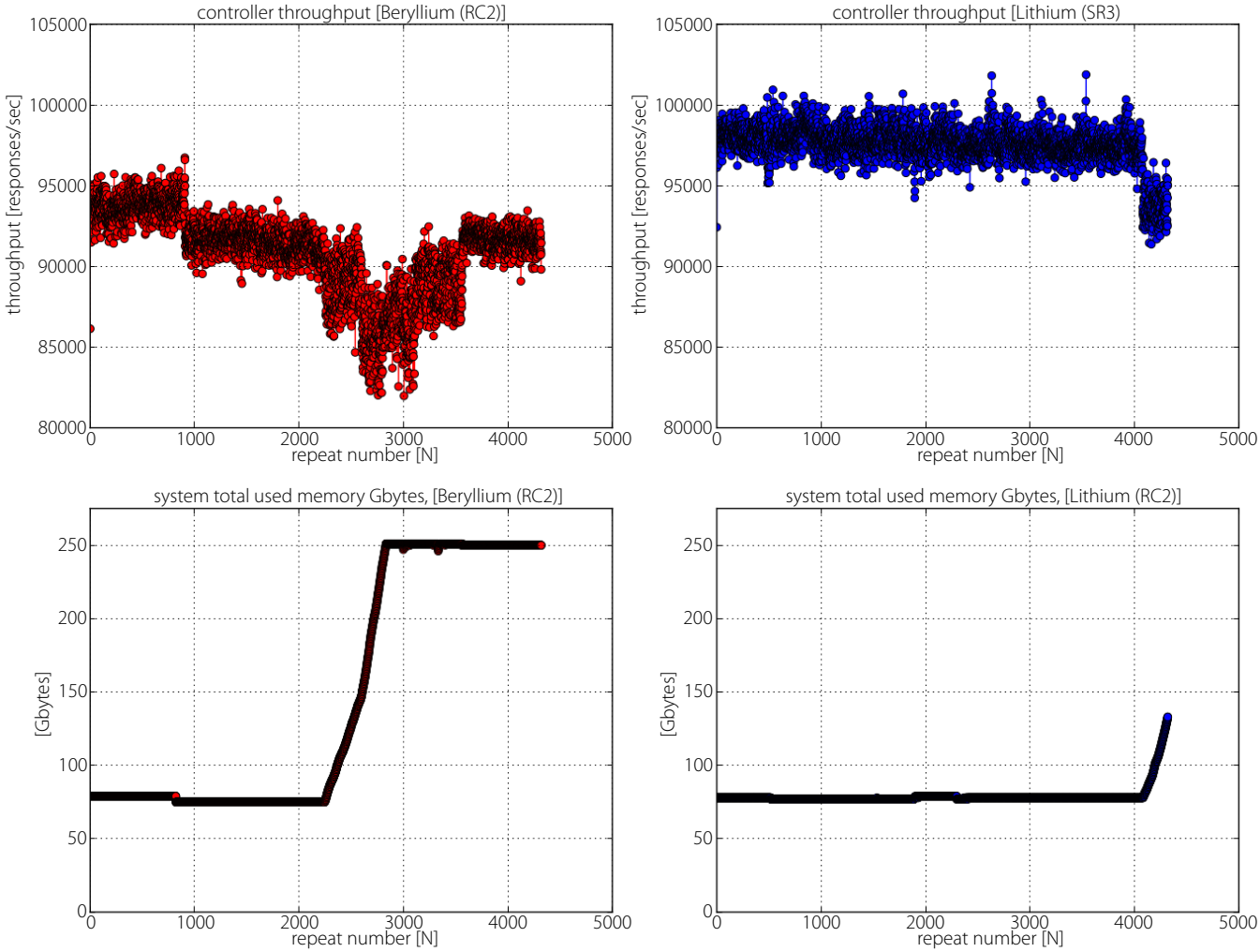
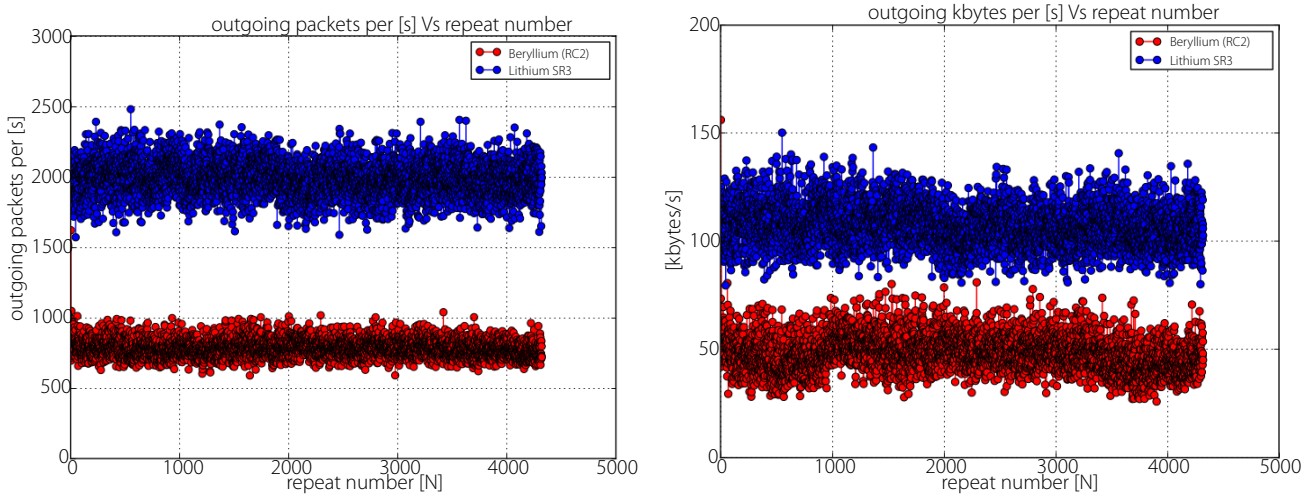


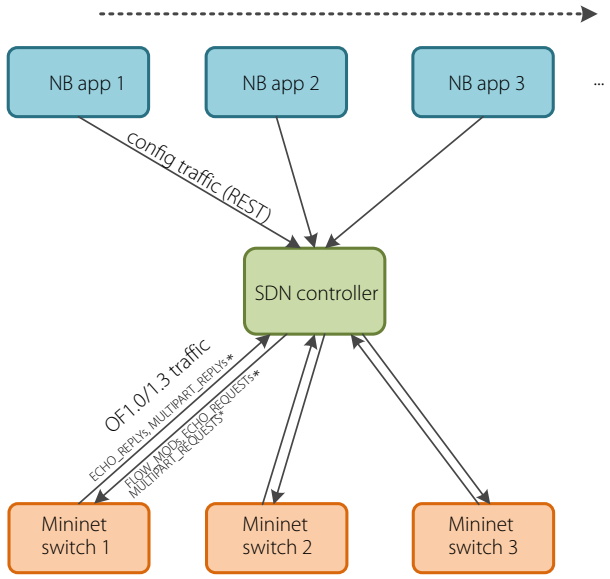
Fig. 12: Controller stability stress test with idle MT–Cbench switches. Throughput and system total used memory comparison analysis between OpenDaylight Lithium (SR3) and Beryllium (RC2) versions. OpenDaylight running in "RPC" mode.



(a) Comparison analysis for outgoing packets/s between OpenDaylight Lithium (SR3) and Beryllium (RC2) releases.

(b) Comparison analysis for outgoing kbytes/s between OpenDaylight Lithium (SR3) and Beryllium (RC2) releases.

Fig. 13: Controller 12–hour stability stress test with idle Multinet switches.



* the vast majority of packets

Fig. 14: Representation of flow scalability stress test. An increasing number of NB clients (NB app$_j$, j=1,2,. . .) install flows on the switches of an underlying OpenFlow switch topology.

of NSTAT was used [9].

5.2.1 Test configuration, "DataStore" mode

- controller: "DataStore" 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
- persistence: enabled
- total running time: 12h

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

5.2.2 Results

The results of this test are presented in Fig.11.

5.2.3 Test configuration, "RPC" mode

- 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
- persistence: enabled
- total running time: 12h

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

5.2.4 Results

The results of this test are presented in Fig.12.

5.3 Conclusions

5.3.1 Idle stability tests

From Fig.13 it is apparent that both releases exhibit stable behavior in a time window of 12 hours.

A first observation is that Beryllium has lower idle traffic overhead compared to Lithium. To investigate the exact reasons for this discrepancy we analyzed the traffic during a sample 1–min stability test with a Linear–switch Multinet topology. Both controllers were configured to request statistics every 5000 ms. The traffic breakdown for both cases is depicted in Table 2. It is made clear that the reduced traffic in Beryllium

Table 2: OpenFlow traffic breakdown for a sample 1–min stability test with a 3200–switch Multinet topology.

	Lithium (SR3)		Beryllium (RC2)	
	Packets	Bytes	Packets	Bytes
Incoming traffic				
OFPT_ECHO_REQUEST	6558	45906	6832	47824
OFPT_PACKET_IN	4723	612772	4957	643468
OFPT_MULTIPART_REPLY	139528	8029791	4787	3402334
Outgoing traffic				
OFPT_PACKET_OUT	5037	643848	5307	678138
OFPT_ECHO_REPLY	9135	63945	8901	62307
OFPT_MULTIPART_REQUEST	80040	4265880	4797	132531

is attributed to the reduction of MULTIPART messages, both incoming and outgoing, possibly as a result of optimizations implemented in Beryllium in this area.

5.3.2 Active stability tests

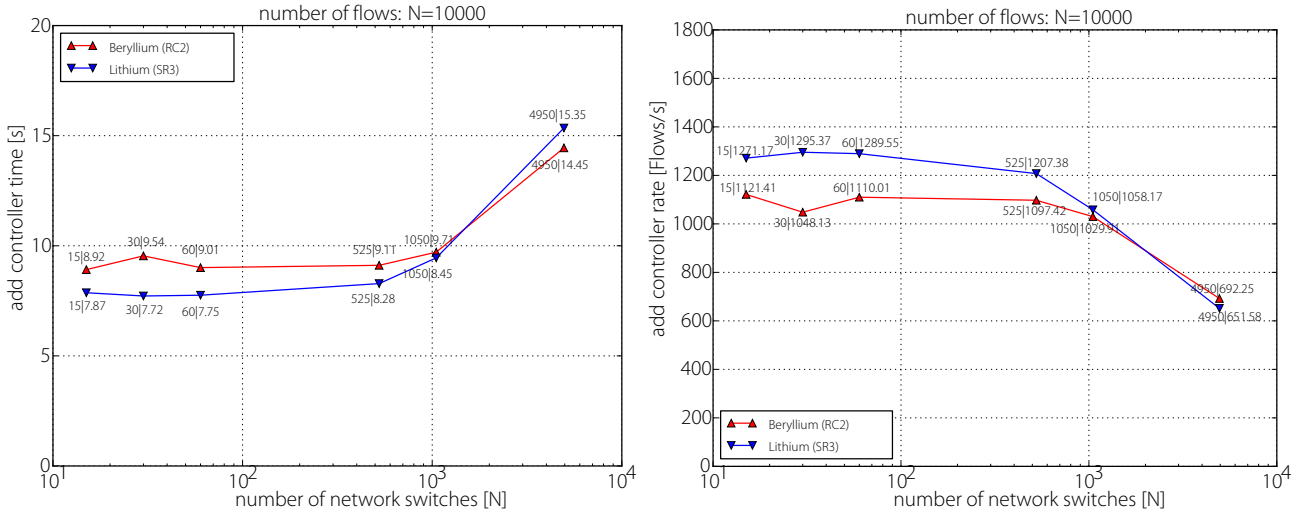
As we can see in Figs.11, 12, both controllers exhibit in general stable performance, at almost the same throughput levels.

Lithium performs slightly better than Beryllium in "RPC" mode. In all four cases, a transient performance degradation is observed for a relatively large period. After analyzing system metrics we concluded that this behavior could be related to an increase in the total used memory of our system (see Figs.11, 12) as a result of possible memory leak.

6. Flow scalability tests

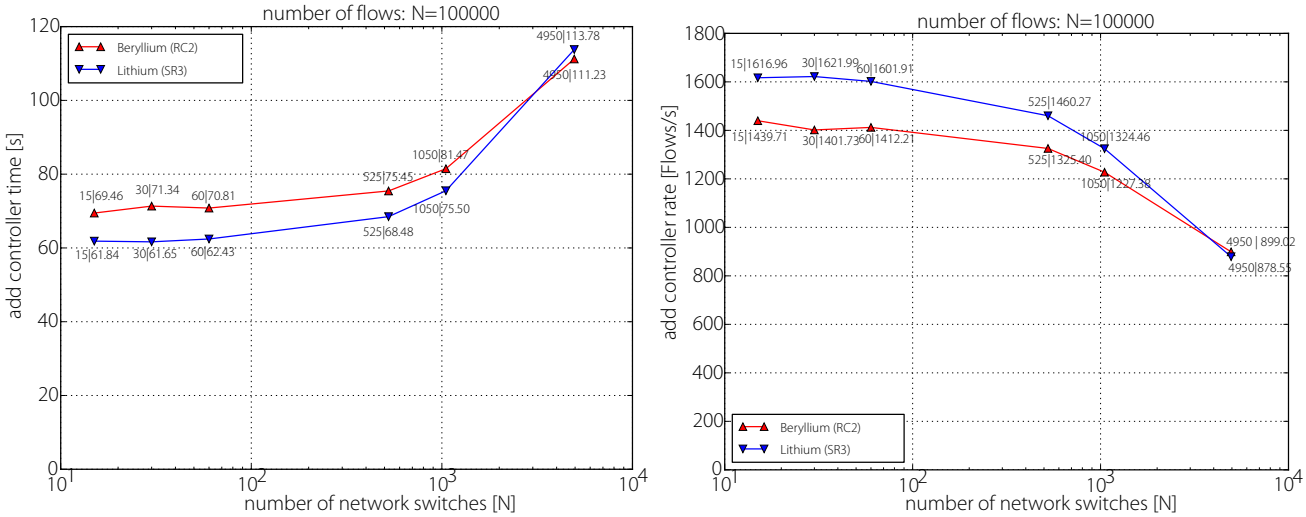
With flow scalability stress tests we try to investigate both capacity and timing aspects of flows installation via the controller NB (RESTCONF) interface, Fig.14.

This test uses the NorthBound flow generator [4] to create flows in a scalable and configurable manner (number of flows,



(a) Flow operations: 10^4. (b) Flow operations: 10^4.

Fig. 15: Flow scalability stress test result. Comparison performance for add controller time/rate vs number of switches for various numbers of flow operations N. Add controller time is forced to -1 when test fails. Add controller time/rate vs number of switches for N=10^4 flow operations.



(a) Flow operations: 10^5. (b) Flow operations: 10^5.

Fig. 16: Flow scalability stress test result. Comparison performance for add controller time/rate vs number of switches for various numbers of flow operations N. Add controller time is forced to -1 when test fails. Add controller time/rate vs number of switches for N=10^5 flow operations.

delay between flows, number of flow writer threads). The flows are being written to the controller configuration data store via its NB interface, and then forwarded to an underlying Multinet topology as flow modifications, where they are distributed into switches in a balanced fashion.

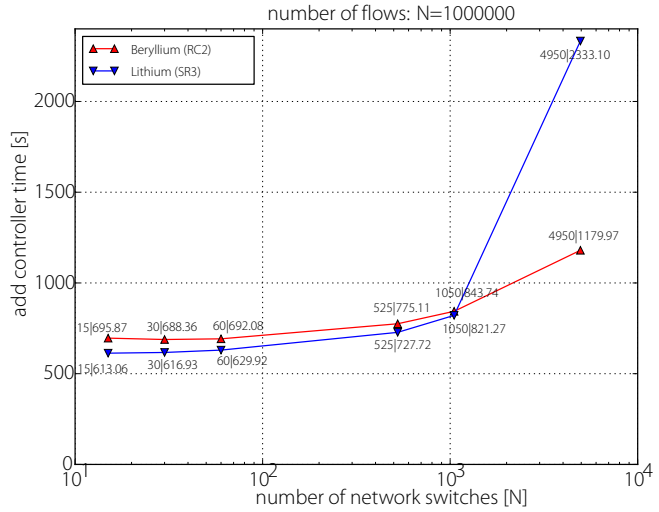
The test verifies the success or failure of flow operations via the controller's operational data store. An experiment is considered successful when all flows have been installed on switches and have been reflected in the operational data store of the controller. If not all of them have become visible in the data store within a certain deadline after the last update, the experiment is considered failed.

Intuitively, this test emulates a scenario where multiple NB applications, each controlling a subset of the topology[2], send simultaneously flows to their switches via the controller's NB interface at a configurable rate.

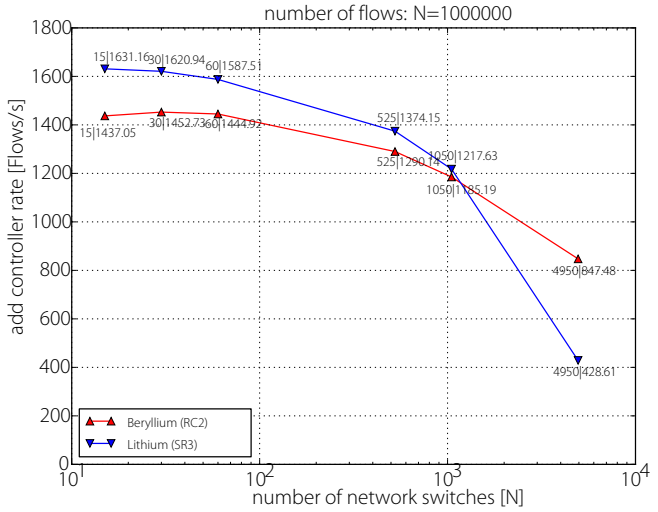
The metrics measured in this test are:

- **Add controller time** (t_{add}): the time needed for all requests to be sent and their response to be received (as in [5]).
- **Add controller rate** (r_{add}): r_{add} = N / t_{add}, where N the aggregate number of flows to be installed by worker threads.
- **End–to–end installation time** (t_{e2e}): the time from the

[2] subsets are non-overlapping and equally sized

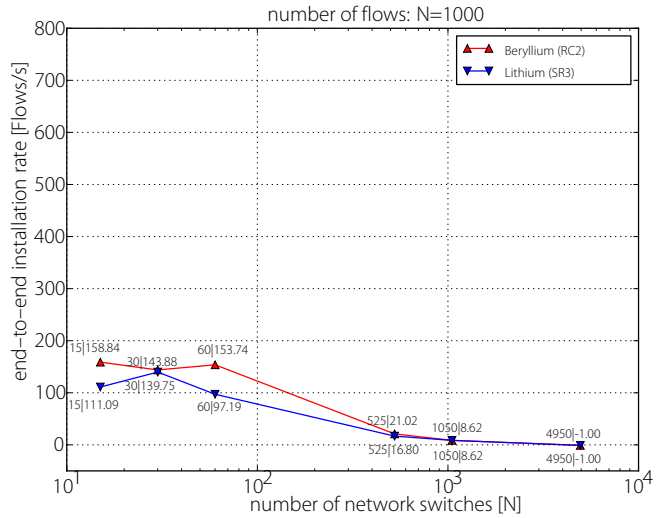


(a) Add controller time vs number of switches for N=10^6 flow operations.

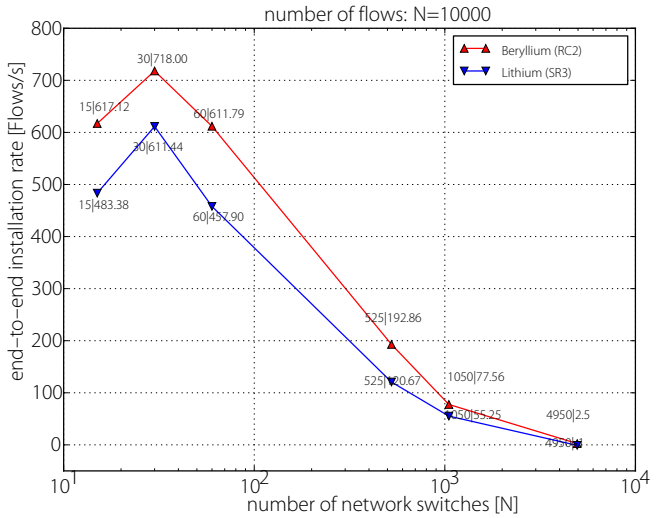


(b) Add controller rate vs number of switches for N=10^6 flow operations.

Fig. 17: Flow scalability stress test result. Comparison performance for add controller time/rate vs number of switches for various numbers of flow operations N. Add controller time is forced to -1 when test fails. Add controller time/rate vs number of switches for N=10^6 flow operations.



(a) Flow operations: 10^3.



(b) Flow operations: 10^4.

Fig. 18: Flow scalability stress test result. Comparison performance for end–to–end installation rate vs number of switches for various numbers of flow operations N. End–to–end installation rate is forced to -1 when test fails.

first flow installation request until all flows have been installed and become visible in the operational data store.

- **End-to-end installation rate** (r_{e2e}): $r_{e2e} = N / t_{e2e}$

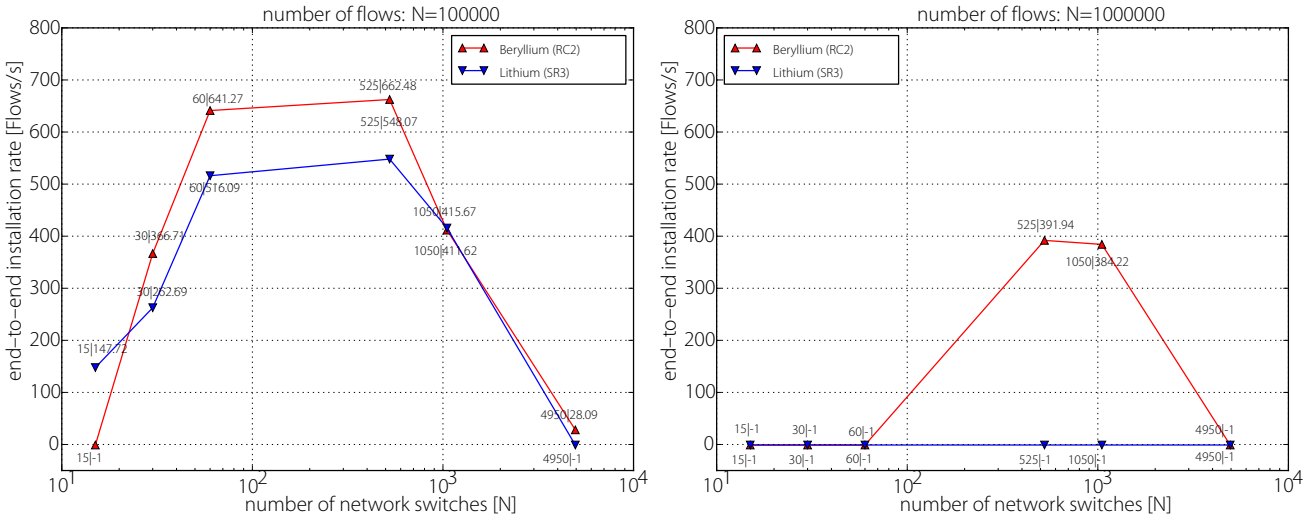
In this test, Multinet switches operate in idle mode, without initiating any traffic apart from the MULTIPART and ECHO messages with which they reply to controller's requests at regular intervals.

6.1 Test configuration

For both Lithium and Beryllium we used the following setup

- topology size per worker node: 1, 2, 4, 35, 70, 330.
- number of worker nodes: 15
- group size: 1
- group delay: 3000ms
- topology type: "Linear"
- hosts per switch: 1
- total flows to be added: 1K, 10K, 100K, 1M
- flow creation delay: 0ms
- flow worker threads: 5
- persistence: disabled

In this case switch topology size is equal to: 15, 30, 60, 525, 1050, 4950.



(a) Flow operations: 10^5. (b) Flow operations: 10^6.

Fig. 19: Flow scalability stress test result. Comparison performance for end–to–end installation rate vs number of switches for various numbers of flow operations N. End–to–end installation rate is forced to -1 when test fails.

6.2 Results

6.2.1 Add controller time/rate

The results of this test are presented in Figs.15, 16, 17.

6.2.2 End–to–end flows installation controller time/rate

The results of this experiment are presented in Figs.18, 19.

6.3 Conclusions

Regarding NB performance, as it is expressed in terms of add controller time/ rate, Lithium performs generally better than Beryllium, but this trend tends to diminish and finally gets reversed for large topologies. Notably, Be outperforms Li by a large factor in the 1M flows/5K switches case.

However, regarding end–to–end performance, Beryllium is the clear winner, outperforming Lithium in all cases of topology sizes and flow counts. In many extreme cases where Lithium fails, Beryllium manages to successfully install flows. This happens for example in the case for 1M flows, or in some 5K switch topologies. This witnesses improvements in the Beryllium release relating to flow installation and discovery.

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] "Multinet: Large–scale SDN emulator based on Mininet." https://github.com/intracom-telecom-sdn/multinet.

[4] "NSTAT: NorthBound Flow Generator." https://github.com/intracom-telecom-sdn/nstat/wiki/NorthBound-flow-generator.

[5] "OpenDaylight Performance: A practical, empirical guide. End–to–End Scenarios for Common Usage in Large Carrier, Enterprise, and Research Networks." https://www.opendaylight.org/sites/www.opendaylight.org/files/odl_wp_perftechreport_031516a.pdf.

[6] "Cbench: OpenFlow traffic generator." https://github.com/andi-bigswitch/oflops/tree/master/cbench.

[7] "Mininet. An instant virtual network on your Laptop." http://mininet.org/.

[8] "Multinet Features." https://github.com/intracom-telecom-sdn/multinet#features.

[9] "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.

[10] "OFTraf: pcap–based, RESTful OpenFlow traffic monitor." https://github.com/intracom-telecom-sdn/oftraf.

[11] "Generate PACKET_IN events with ARP payload." https://github.com/intracom-telecom-sdn/multinet#generate-packet_in-events-with-arp-payload.