

Ben-Gurion University of the Negev

Faculty of Engineering Science

School of Electrical and Computer Engineering

Dept. of Communication Systems Engineering

Fourth Year Engineering Project

PDR

Distributed Caching-based Acceleration Mechanisms in Datacenter Networks

**Project Number:** p-2022-093

**Students:** Shir Granit 205531445, Anna Axalrod 324682475

**Supervisors:** Prof. Chen Avin, Dr. Gabriel Scalosuv

**Submitting Date:** 24/10/2021

**Table of Contents:**

[Abstract 3](#_Toc54193468)

[Abstract (English) 3](#_Toc54193469)

[Abstract (Hebrew) 4](#_Toc54193469)

[Research Proposal 5](#_Toc54193471)

The [Essence of the Project 5](#_Toc54193472)

[Technologies and Methodologies that Will be Used in the Project 5](#_Toc54193473)

[Work Plan 6](#_Toc54193475)

[Application for our Project 6](#_Toc54193476)

[System Performance Spec 6](#_Toc54193477)

Project General Scheme [6](#_Toc54193477)

[Design Approach 7](#_Toc54193479)

[Different Approaches for the Problem Presented 7](#_Toc54193480)

[Our Approach 7](#_Toc54193481)

[Bibliography 9](#_Toc54193482)

[Appendix: PDR Evaluation 10](#_Toc54193482)

1. **Abstract**
   1. **Abstract (English):**

**Distributed Caching-based Acceleration Mechanisms in Datacenter Networks**

Students Names: Axalrod Anna, Granit Shir

[axalrod@post.bgu.ac.il](mailto:axalrod@post.bgu.ac.il)

Due to high data flow through a data center network, switches in the network face a problem storing an enormous amount of traffic rules which are necessary for correctly transferring packets in the network. Those rules are usually stored in an external device to which the access slows the network’s performance. The purpose of the project is to find a solution that prevents multiple access to the external device, leading to a significant speedup of the routing process in the network.

In our project, we will develop a distributed solution for the problem presented which will be based on a previous solution that uses cache.

Our proposed method includes creating a network with several switches which send queries to each other to get forwarding rules. In case of missing information, we will request the forwarding rule from the external device (the controller), which keeps all the forwarding rules and holds the switch's cache memory.

We expect a decrease in the number of requests to the external device for rules when sharing the cache between the network's switches, as well as faster network performance.

Keywords – Network Topologies, Distributed Algorithms, P4, Cache, Software-Defined Networking, LPM, Rule, Mininet, Pipeline, OpenFlow.

* 1. **Abstract (Hebrew):**

**מנגנוני האצה מבוזרים מבוססי מטמון ברשתות מרכזי נתונים**

שמות הסטודנטים: אקסלרוד אנה, גרניט שיר

[axalrod@post.bgu.ac.il](mailto:axalrod@post.bgu.ac.il)

בעקבות זרימת נתונים גבוהה למרכז נתונים, בפני המתגים ברשת, ניצבת בעיה לאחסן כמות עצומה של חוקי תעבורה אשר נדרשים לביצוע העברה תקינה של החבילות ברשת. בדרך כלל אחסון החוקים קורה במכשיר חיצוני שהגישה אליו מאטה משמעותית את ביצועי הרשת. מטרת הפרויקט היא למצוא פתרון אשר ימנע את הגישה המרובה למכשיר החיצוני ובכך יאיץ משמעותית את תהליך הניתוב ברשת.

בפרויקט הגמר שלנו, נפתח פתרון מבוזר לבעיה זו המתבסס על פתרון קודם שעשה שימוש במטמון. בעזרת מתגים ברי תכנות אנו נדמה רשת בה לכל מתג יהיה זיכרון מטמון משלו אשר ינוהל ע"י פרוטוקול רשת מבוזר.

שיטת העבודה שלנו תכלול הקמת רשת עם מס' מתגים אשר ישלחו שאילתות אחד לשני כדי לקבל חוקי ניתוב. במידה שהחוק לא נמצא, נפנה לבקר מרכזי ששומר את כל חוקי הניתוב לקבלת חוק הניתוב המבוקש ושמירתו במטמון המתג.

בתוצאות הפרויקט, אנו מצפות כי כמות הפניות לבקר המרכזי, כאשר יהיה שיתוף מטמון בין מתגי הרשת, תקטן באופן משמעותי וביצועי הרשת יואצו.

מילות מפתח: טופולוגיות רשת, אלגוריתמים מבוזרים, P4 (שפת תכנות לרשתות תקשורת), זיכרון מטמון, SDN (רשת מוגדרת תוכנה), LPM (התאמת הקידומת הארוכה ביותר(, חוקים, Mininet (תוכנת אמולציה לרשתות), צינור תעבורה, OpenFlow (פרוטוקול תקשורת).

1. **Research Proposal**

Distributed Caching-based Acceleration Mechanisms in Datacenter Networks

* 1. **The Essence of the Project:**

The goal of the project is to implement and evaluate a P4 switch-based distributed caching mechanism for action rules designed to accelerate datacenter networks. We plan to create an emulated distributed datacenter environment, which will consist of switches, hosts, and an SDN network controller with in-network cache management capabilities, in several topologies. Then, we will use this environment to evaluate the performance of a given caching algorithm and measure its efficiency and accuracy.

* 1. **Technologies and Methodologies that Will be used in the Project:**
     + OpenFlow protocol – A protocol used in SDNs allowing remote administration of packet forwarding rules in the network's switches [1].
     + Programmable switches (P4) – P4 is a programming language used to control the packet forwarding plane in programmable network devices, like routers and switches [2].
     + Emulation software (Mininet) – Mininet is a network emulator widely used to emulate virtual hosts, routers, network controllers, and switches supporting the OpenFlow protocol.
     + SDN controller (Written in Python) – The controller is the decision-making mechanism that supports the cache-based switches.
     + Traffic generators (Written in C, in a Linux OS script format) – Traffic will be generated for tests and evaluations of the simulation we will create [3].
     + Longest Prefix Matching (LPM) – A method to perform packet forwarding by matching the packet destination subnet to the closest matched network in the rules table.
     + Caching algorithms – To maximize the number of Cache-hits, we will implement different existing Cache Replacement Policies (FIFO, LRU, MRU, etc.) and select the best suited for this project [4].
     + Communication protocol - To be able to transfer packets between the switches, we need a communication protocol for the fast and efficient transfer of information.
  2. **Work Plan:**
     + Use a Mininet environment created last year and adjust it for our purposes. It will include a small datacenter network with switches, hosts, an SDN controller, and a traffic generator, in several different topologies.
     + Design the packet pipeline inside the switches using the P4 programming language and implement a cache inside the switches to store the forwarding rules.
     + Create (or use a given) caching algorithm and deploy it into the switches.
     + Configure a network controller using Python and OpenFlow to manage the datacenter network and handle the Cache-misses.
     + Design a traffic generator, made from different distributions [5].
     + Evaluate the performance of different topologies.
  3. **Applications for our Project:**

Our purpose is to accelerate traffic forwarding in large networks, allowing networks in datacenters to increase their performance, reduce their costs on expensive hardware and facilitate their support for larger infrastructures with relatively few hardware changes.

* 1. **System Performance Spec:**

We expect our project implementations to accelerate current datacenter networks performance. The caching mechanism should be easily deployable in all current programmable switches running in an SDN network. Our input is packets corresponding to several flows, and we expect to see a reduction in the network's packet processing time.

The equipment needed for this research is a PC equipped with emulation software, and enough memory to emulate a significant number of switches and hosts.

* 1. **General Scheme:**
     + The current state of the system built last year:

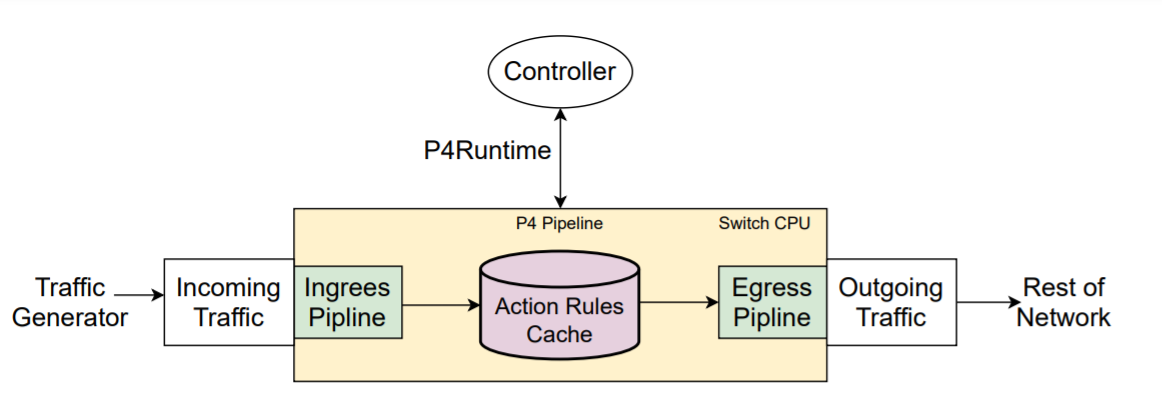
****

Figure 1: General flow of the system that was built last year.

* + - The desired topology to actualize:

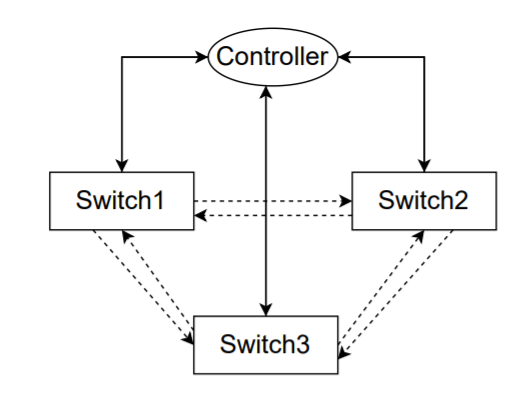
****

Figure 2: The desired topology of the network.

As described in Fig. 1, the existing model consists only of a single switch with a single controller. In our project we plan to create the topology described in Fig. 2. This topology will be composed in a distributed way and includes three switches and a single controller. Furthermore, all components will relate to at least one port between one another. In a further stage, we plan to try and create more complex topologies, in case the topology described in Fig. 2 will yield the expected results.

1. **Design Approach**
   1. **Different Approaches for the Problem Presented**

* One approach is to use switches that can store the entire policy, usually using cloud computing, enabling them to provide line-rate responses to table lookups. That means that all switches are required to hold the forwarding policies of the entire datacenter. This solution faces scalability issues [6].
* A second approach is to use cheaper switches and offload the entire cloud-scale policy to some external device, to store the forwarding rules. Each switch will be connected to such a device, and upon packet arrival, it will ask the device for the correct forwarding rule for that certain packet. In a datacenter that will create bottlenecks in the network.
  1. **Our Approach:**

The current project is an extension of last year’s project.

First, we will adjust the given emulation environment in Mininet, based on Fig. 1.

Creating different topologies and adjusting the current algorithms. After implementation, we will generate differently distributed traffic, and evaluate the performance of our implementation under different workloads.

The goal is to find the attributes that maximize the throughput of the network being tested, in more complex topologies than those that were tested last year.

* **Project Limitations**
  + Mininet-based networks environment cannot (currently) exceed the CPU or bandwidth available on a single server [7].
  + Mininet environment cannot (currently) run non-Linux-compatible OpenFlow switches or applications [7].
  + There is not currently a way to examine our system in a real-life datacenter network since we do not have access to one.
* **Project Assumptions**
  + Our main assumption is that the system is described in Fig. 1. is working correctly, and the simulation environment is working with a single switch connected to a single controller, given traffic in the network.
  + We assume the availability of an SDN controller.
  + We assume BMv2 P4 implementation in Mininet is stable.

1. **Bibliography**

|  |  |
| --- | --- |
| [1] | B. Salisbury, "TCAMs and OpenFlow: What Every SDN Practitioner Must Know" 2 July 2012. |
| [2] | P4 Language Consortium. P4 switch. https://github.com/p4lang. |
| [3] | S. U. A. F. R. S. a. X. H. N. Sarrar, “Leveraging Zipf’s law for traffic offloading”, SIGCOMM Comput. Commun. Rev., 2012. |
| [4] | S. Jiang, "Efficient Caching Algorithms for Memory Management in Computer Systems", College of William & Mary - Arts & Sciences, 2004. |
| [5] | A. D. A. P. Alessio Botta, "A tool for the generation of realistic network workload for emerging networking scenarios", 19 February 2019. |
| [6] | V. Ravikumar and R. Mahapatra, "TCAM architecture for IP lookup using prefix properties,", EEE Micro, vol. 24, no. 2, pp. 60-69, March-April 2004. |
| [7] | Mininet Overview. http://mininet.org/overview. |

1. **Appendix: PDR Evaluation**

**הערכה לשיחת סקר תכנון ראשוני (PDR)**

אם יש צורך, לכל סטודנט/ית בנפרד

מספר הפרויקט: p-2022-093

שם הפרויקט: מנגנוני האצה מבוזרים מבוססי מטמון ברשתות מרכזי נתונים.

שם המנחים מהמחלקה: פרופסור חן אבין, ד"ר גבריאל סקלוסוב.

שם הסטודנטית: אנה אקסלרוד ת.ז.: 324682475

שם הסטודנטית: שיר גרניט ת.ז.: 205531445

|  |  |  |  |
| --- | --- | --- | --- |
| % | מהות | ציון | הערות |
| 67 | שיחה + דו"ח - הבנת הנושא ומהות העשייה, הבנת הצורך, סביבת היישום, הגדרת מדדים, מקורות ועבודות דומות.  הצגת התקצירים, מפרט טכני/הצעת מחקר והצעת תכנון מפורטים. |  |  |
|  |  |  |  |
| 33 | בונוס על מילוי דף הערכה בכנס. והשתתפות בסדנה להכנה לפרויקט. |  |  |
|  | ציון סופי |  |  |

הערות: