ראה סיכום
הערות
בסוף הדוח

נבדק ללא ציון
נא לתקן הערות במסגרת
הגשת דוח התקדמות

Improving Segment Routing on SDN Networks

Pre-Report

Prof Michael Segal

Bella Khizgiyaev
Sagi Zukerman

30.11.2018

חסר מספר הפרוייקט
חסר מספרי תעודות זהות
חסר ציון שם המחלקה והפקולטה
חסר לוגו המחלקה/אוניברסיטה

Student signature: _______________ Prof signature: _______________

חסר מיספור פרקים ותתי פרקים

TABLE OF CONTACT

לדעתי היה יכול להכנס ביחד עם העמוד הקודם. אין
צורך בעמוד נפרד

TABLE OF FIGURE

יפה ומרשים

INTRODUCTION

BACKGROUND AND MOTIVATION

SDN (Software Define Network) - using SDN we can manage and control low level component in our network using abstraction. In this way, the component that works on a specific layer don't have to know the operation mode of components in lower levels, only the outcome of the operation to work properly. Using this approach gives us the ability to manage and control these networks efficiently and wisely. We can achieve these goals by controlling the network with SDN network controller which have the knowledge and control of all the components in the network.

Segment Routing – provides full control regarding the forwarding package path using simple instructions. Using segment routing don't require adding additional routing protocol which allows us to abstract the network routing protocols.

Machine Learning – a category of algorithms that can by input values and statistically analysis predict the possible outputs.

PROBLEM INTRODUCTION

The optimal path is not always the traditional shortest path we use as our main route on the network. Sometimes the shortest path is congested and sending the packet through it may create delays in the network due to congested queues and even to packet loss. Sending the packet wisely through different paths can balance network load.
However, switching paths while sending a flow of data may cause reordering problem due to different latencies in the paths.

PROJECT GOAL

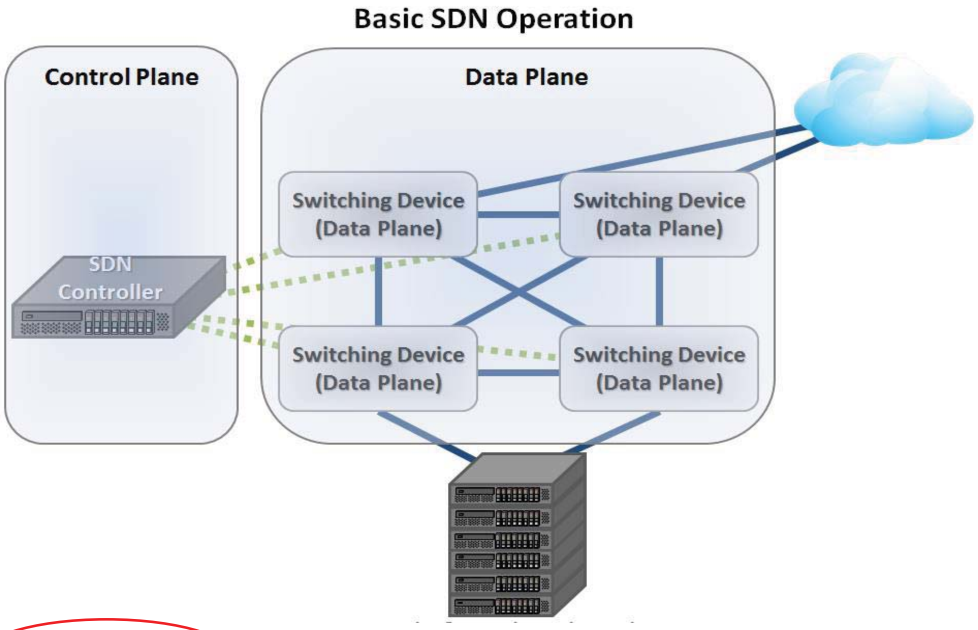
The goal will be predicting the best path wisely and by doing so try to overcome the network congestion and packet lost. By applying wise path selection and more global approach to the network loads, we would be able to improve the segment routing and the network overall performance. This kind of path selection will minimize the interference between pairs of source and destination in the network, exchanging data with each other. Moreover, wise path selection can balance the network traffic between all the links and overcome the packet loss and queuing congestion.
By saving each link 'load history' and machine learning that will learn the network behavior, the algorithm will be able to predict the loads on each link in advance and make a better routing decision.

THEORETICAL BACKGROUND

SDN NETWORKS

The main goal of SDN is to enable quick response to network changes via centralized control console. SDN encompasses a variety of network technologies in order to make the network more flexible and agile and support modern data centers. The SDN main approach is to design, build and manage networks that separate between the network control and the data forwarding in the network. This approach gives the ability to directly control and program the network infrastructure to be abstract to applications and network services.

SDN Controller – the controller is essentially the "brain" of the network. It's allowing to manage the whole network in a centralized way and give to the network administrator the ability to dictate to the network components like switches and routers the routing policy and data forwarding and basically controls the network traffic. [1]


Basic SDN Operation
Control Plane
Data Plane
SDN Controller
Switching Device (Data Plane)
Switching Device (Data Plane)
Switching Device (Data Plane)
Switching Device (Data Plane)

האם זה שרטוט מקורי
שלכם אם כן כל הכבוד.
אם לא נה לציין את המקור

Figure1 : SDN Network

כותרות שרטוטים
ממוספרים תמיד מתחת
ובמרכז השרטוט

SEGMENT ROUTING (SR)

Segment routing it' s technology that add benefits to IP and MPLS networks. These networks will be simpler to operate and more scalable. In SDN network this kind of routing will provide us a quicker interaction with the application.
Segment routing based on label switching but don’t require any extra protocols on top of it (unlike MPLS), only an extension of IGP (ISIS/OSFP). [2]
SR may be used to steer traffic along arbitrary path in the network. This allows us to choose different paths than the normal forwarding paths. [3]

SR provides automatic traffic protection without any topological restrictions. The network protects traffic against failures (link and nodes) without requiring additional mechanism in the network which also simplifies deployment and management. In addition, traffic protection does not impose any additional signaling requirements. Segment routing can provide strict network performance guarantees, efficient use of network resources and very high scalability for application-based transactions while using minimal state information to meet these requirements. [4]

SR based on MPLS to forward the packets, but the labels are carried by an IGP. In SR, every node has a unique identifier called the node SID. This identifier is globally unique (such as Ip or MAC) and would normally be based on a loopback on the device. [5]

לא רואה הסבר והתייחסות
לשרטוט

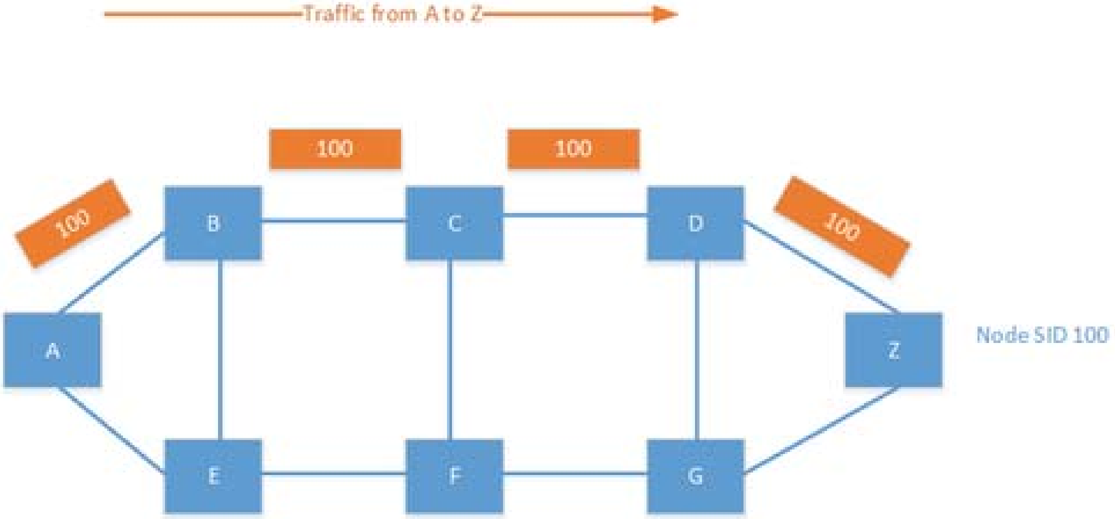
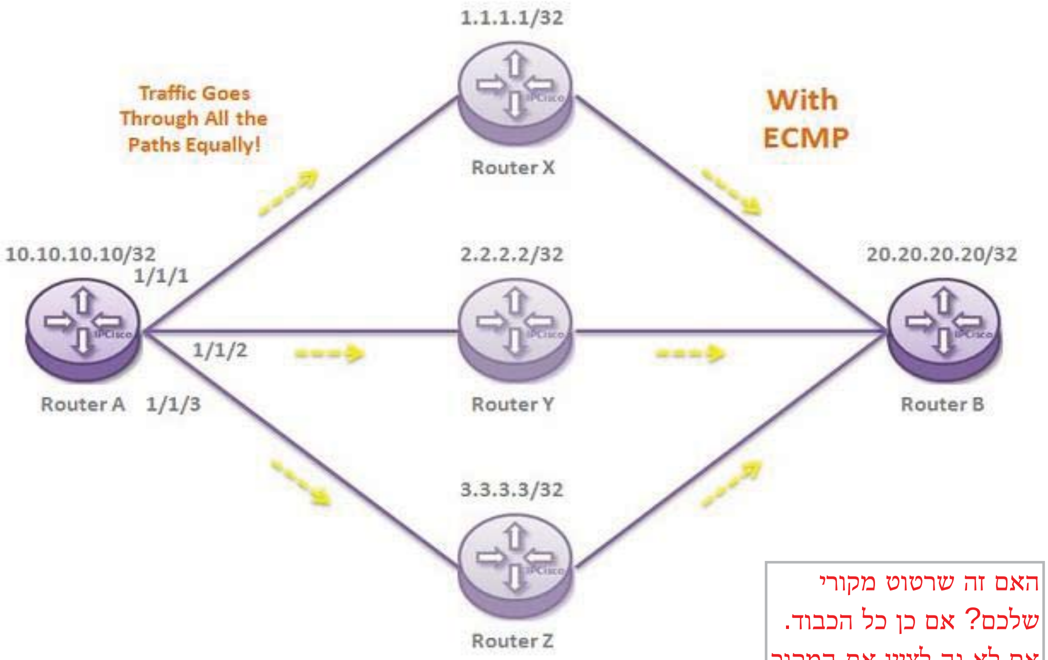


Figure2 : SR example

ECMP

Equal cost multi-path (ECMP) is a routing protocol when the forwarding to a single destination may take place over multiple best routes that equals by their cost (metric). Most routing protocols (like OSPF for example) can be used conjunction with ECMP because it's a per hop decision limited to a single router. The ECMP can increase bandwidth by load balancing traffic over multiple routes in the network. This is the load balancing algorithm that is the base for the “standard algorithm” that we will compare against our improved algorithm, since distribution with this algorithm is random, the improved algorithm is sometimes able to correct “mistakes” made by this algorithm, when big flows are randomly assigned the same path, and are not very well distributed. [6]



האם זה שרטוט מקורי
שלכם? אם כן כל הכבוד.
אם לא נה לציין את המקור

Figure3 : ECMP example

MACHINE LEARNING

Machine learning (ML) is an algorithm category that gives software applications the tools to become more accurate with all regarding predicting possible outcomes without being explicitly programmed. The basic fundament of machine learning is to build algorithms that receive input data and use statistical analysis to predict an possible output. This process similar to data mining and predictive modeling. They basically search the historical data that they have and search for similar patterns and respond accordingly.
we can classify the algorithms into two categories, supervised or unsupervised:

- Supervised – required a data analyst that provide both input and desired output and gives feedback to the algorithm accuracy, this is the training phase. Once the algorithm finished the training it will apply what he learned on new data
- Unsupervised – not like the above kind, these algorithms don’t need to be trained, and uses an iterative approach called deep learning. These networks work by combing together large amount of examples of training data and automatically identifying logical and correct correlations between many variables. Once these algorithms trained, it can use its bank of associations to analyst new data. [7]

THE ELEPHANT AND MICE PHENOMENON

TCP flows interact, traffic at the level of network prefixes and WWW traffic investigation shown that a minor percentage of the described above flows carries the major part of the information. The described behavior may be referring as "The elephant and the mice phenomenon".
Due to this discover network traffic app's such as re-routing or load balancing can take advantage of this property and treating elephant flows differently. The main challenge will be to determine which flows are elephant using the flow's bandwidths and data collection across all link flows. [8]

TOPOLOGIES

In the project we will inspect several kinds of topologies for performance, Dragonfly, Hypercube and Torus.

DRAGONFLY

The Dragonfly topology provides low-diameter connectivity for high-performance computing when all the router connected to each other at the inter-group level (clique). The traffic matrix of this topology that been investigated by several scientific applications shows consistent mismatch between the imbalanced clique to clique traffic and the uniform global bandwidth allocation of Dragonfly. [9]

With its high-connectivity, Dragonfly can greatly reduce the network diameter over several topologies (Fat tree or multidimension path). A Dragonfly can connect any two routers within distance of 3 in a clique to clique network. A price for the high-connectivity, however, is diluted per-link bandwidth. In particular, the bandwidth of inter-group (global) links, carrying all the traffic between two large sets of routers, becomes the scarcest resource and can become bottleneck for the entire network. Many scientific applications on a Dragonfly platform, unfortunately, tend to concentrate traffic on only a few of these links. Neighbor-group based communication pattern, for example, is prevalent for many applications. Moreover, inter-group traffic matrix is often sparse, leading to bandwidth allocated for idle pairs.

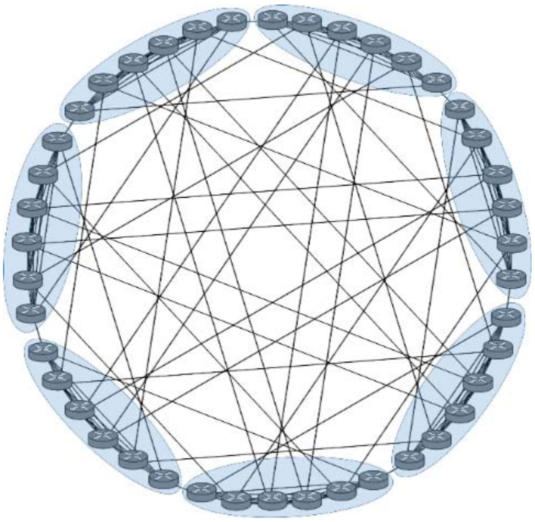


Figure4 : Dragonfly Topology, 7 cliques. each clique consists of 6 routers, each router connected to two other groups.

HYPERCUBE

A hypercube structure can be defined as a graph that it's node set consists of the 2n vectors with values 1 or 0 of dimension n. Two nodes will be neighbors if the difference between them is exactly one coordinate in the vector. We can also Hypercube may be considered as a generalization of a cube of dimensions 3. The mathematical properties of the hypercube are easily represented proving that the number of neighbors a node has and the diameter (maximum distance between any pair of nodes) are both equal to the dimension n. [10]

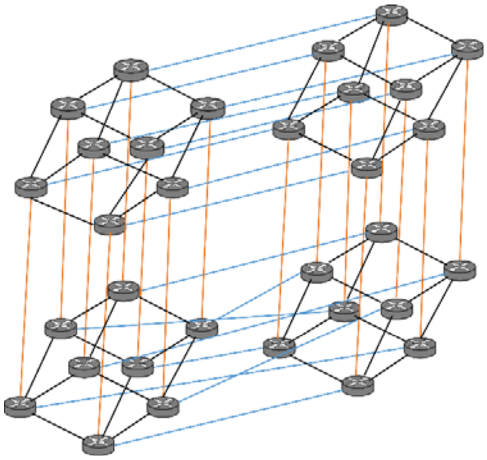
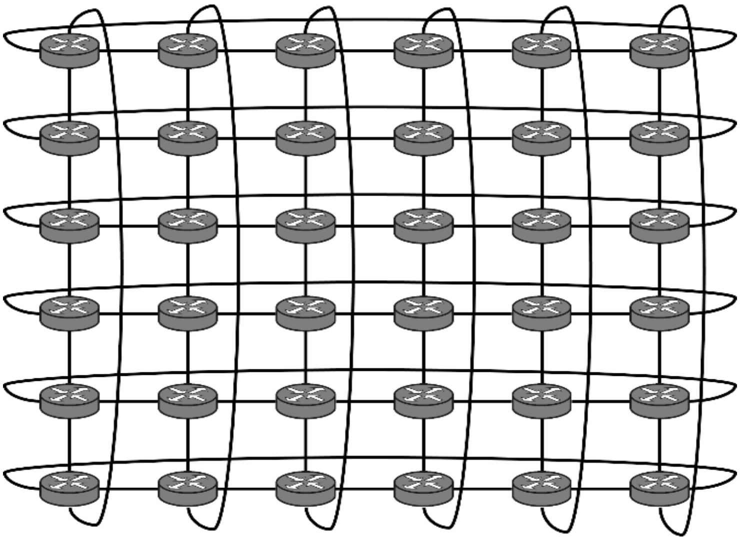


Figure5 : Hyper Cube Topology, 5 dimensions

TORUS

A torus interconnect can be visualized as a mesh with all nodes on the edges having an added connection to the corresponding nodes on the opposite edge. Due to these wraparound links, the diameter is reduced by almost a half compared to the mesh. The node degree (number of neighbors of a specific node) is constant as the network scales up giving the torus an advantage when it comes to implementation costs. However, this may indicate a potentially lower performance level having fewer possible routes than the hypercube. [10]



האם זה שרטוט שלכם?
אם לא נא לציין את המקור

Figure6 : Torus Topology, m=6, n=6

ALGORITHMS AND WORK

ALGORITHM DESCRIPTION

The main idea of our algorithm is making routing decisions that are based upon taking the path with the lowest delay while taking into account other parameters such as the size of the flow and future loads of the network.
The decisions will be made efficiently and quickly by the SDN controller that has a real time view of the entire network, its topology and loads on links.

When a packet arrives to a network node, the node will send it through the default shortest path route, the algorithm goal is to find the best new route by wise prediction in the following way:

- Identify all the flows in the network, they will be categorized into two categories, small flows ("Mice flow") and big flows ("Elephant flow"), the decision will be made by reaching a defined threshold.
- If the flow is "Mice" flow – do nothing.
- Else, ths SDN controller will find better route (if exist). The controller will predict the future load for each link in the network (using the improved predict algorithm) and mark it as the link 'weight'. After all the predict weights are calculated we will use the Dijkstra algorithm on the network graph to find the 'cheapest' path.
- If moving the load to the cheapest path found won't create larger load from the one that exist on the current link, we will move it to the new path. Else, we'll leave the flow on the current path.
- Once the flow had been moved one time, we won't move it again and let the flow finish on the current link.

TOOLS AND PROGRAMS

For simulating our project, we will use Omnet++ with INET library.

Additional Programs and Codes:

1. Scalable Topology Constructor.
2. XML - Configurations Builder. (C)
3. Map-Reduce Traffic Builder. (C)
4. ECMP Capability in Segment Routing. (C++)
5. Simulations and Feasibility. (Omnet++)
6. Embedding our algorithms. (C++)
7. Edge list. (Mathematica)

PROJECT PHASES

The project will be accomplished by several main steps as described below, each stage is important milestone and will bringing us a step closer to finishing the project.

The steps will be:

1. Write the prediction algorithm, the algorithm will predict the future load given the' load history' that will be saved for each link in the network.
2. Implement the improved algorithm for each of the topologies (Dragonfly, Hyper Cube and Torus), Including saving the history in correct format and calculating according the future load, based on the matching we will find.
3. Simulate and gather result for each topology, produce graphs and make relevant result analysis.
   Compare the result to the last year algorithms and to the ECMP protocol and see if the improve protocol gives better routing and load result.

DEADLINES

The deadlines of the project milestones will be:

1. Submit the improved algorithm until the 10.12.18.
2. Submit the implementation of the new algorithm in the Dragonfly topology, gather all the relevant data and making the graphs until 10.2.19.
3. Submit the implementation of the new algorithm in the Hyper Cube topology, gather all the relevant data and making the graphs until 10.4.19.
4. Submit the implementation of the new algorithm in the Torus topology, gather all the relevant data and making the graphs until 10.6.19.

יש להכין טבלת משימות עם תאריכים.
קבוצת משימות מאוגדת יחדיו ומובילה לאבן דרך. לאבן דרך יש תאריך יעד ותוצר שניתן למדידה/לכימות
כלומר יש לייצר שתי טבלאות: טבלת משימות וטבלת אבני דרך

BIBLIOGRAPHY

[1] "*What is Software Defined Networking (SDN)? Definition*," SDx central, [Online]. Available: https://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/.

[2] Y. E. Fathi, "*Introduction to Segment Routing,*" [Online]. Available: https://packetpushers.net/introduction-to-segment-routing/.

[3] "*Segment Routing,*" Segment-Routing, [Online]. Available: http://www.segment-routing.net/.

[4] "*Segment Routing Configuration Guide,*" Cisco, (2018). [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/seg_routing/configuration/xe-3s/segrt-xe-3s-book/intro-seg-routing.pdf.

[5] D. Dib, "*Segment Routing: Potential For Far-Reaching Network Effects*," (2015). [Online]. Available: https://www.networkcomputing.com/networking/segment-routing-potential-far-reaching-network-effects/516090694.

[6] Wikipedia, "*ECMP*," [Online]. Available: https://en.wikipedia.org/wiki/Equal-cost_multi-path_routing.

[7] M. Rouse, "*machine learning (ML)*," (2018). [Online]. Available: https://searchenterpriseai.techtarget.com/definition/machine-learning-ML.

[8] A. W. a. S. W. D. Cai, "*Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*," Sydney, Australia, (2014), pp. 1-6.

[9] P. S. S. R. C. P. C. Y. S. M. B. K. B. J. W. Ke Wen, "*Flexfly: Enabling a Reconfigurable Dragonfly through Silicon Photonic*," (2016). [Online]. Available: https://ieeexplore.ieee.org/document/7877093.

[10] J. V. O. Bogdan Andrus, "*SDN data center performance evaluation of torus and hypercube interconnecting schemes*," (2015). [Online]. Available: https://ieeexplore.ieee.org/document/7365731.

לסכום
דוח מושקע וערוך ברמה טובה.
אך דוח זה היה נפסל ונכשל עקב העתקה ללא ציון שם המקור שממנו הצטטה.
יש לציין מקורות של שרטוטים שנלקחו ממקורות זרים.
אין מספור לכותרות
להשלים טבלת אבני דרך כולל תאריכים ותוצרים
חסר טבלת קיצורים ומושגים
חבל שלא הצלחת למצוא נוסחאה או פיתוח מתמטי שאפשר להוסיף לפרוייקט
אין התייחסות לסיכונים, ומה תעשו אם הם יתרחשו
כדאי להוסיף תאור בדוח של מה יהיה התוצר הסופי של הפרוייקט
all pages should include headers and footer
אין רשימת מילות מפתח
נא לצרף שרטוט גאנט (שרטוט המתאר משימות ולוחות זמנים בצורה גראפית, חובה
לציין נקודות קריטיות- משימה שתלויה במשימה אחרת)