# DSL21 Project Group 4: P4Update Flow Update Time

Jiayi Ma, Dingyi Zhou, Weig Johannes


## ABSTRACT

Networks are in a constant state of change. Here change means that network traffic are re-routed through different sets of intermediate points, such as switches or vantage points. Using P4, the classical controller algorithms could be offloaded into data plane, and be realized flexibly by means of P4. But the efficiency is still unknown, so the flow update time is used to evaluate the efficiency of the algorithm. As the first step to validate the superiority of P4Update over state-of-the-art algorithms. The amount of data we got was relatively small, on the other hand the sample size of some data was different, we asked our tutor, this was due to the extreme values of the data being filtered out. However, by analyzing our limited data, we conclude that this algorithm is superior for different topologies in both flow and single flow cases, and its update time is shorter.


## 1 INTRODUCTION/BACKGROUND

In the Scenario we need some basic knowledge of Software-Defined Networking (SDN) and Programmable Data Plane with P4. SDN is used to operate and build large networks. In this way, different devices can be programmed centrally from one server. Before we can do SDN, both the control and data plane have to be coupled within one network. This method brings a lot of benefits and has applied to OpenVSwitch, hardware SDN switches, end host, and so on. The invention of P4 extend the network programmability from controller to data plane and expand protocol-independence, target-independence and flied reconfigurability

The motivation of doing this analysis is to compare the running update times of the three algorithms. Therefor we have three different scenarios the single-flow scenario and the multiple flow scenario. For each scenario, we have three topologies. In each topology, we run experiments for the three algorithms to evaluate flow update time. Each experiment is repeated for 30 times. The analysis helps us to determine the best algorithm for the problem with respect of running time. The Report is structured in Scenario a Data Description, Methods and Results and a Conclusion. In the Scenario and Data Description part, we describe the Algorithm, how the Data is measured and the Topology of the Data. The Methods and Result part is divided in two sections. The first, Mandatory Analysis includes plots like Scatter, Histogram, Boxplot and Violin plot and gives insides into the most important values of the analysis. In the Additional Analysis, which is the second fraction, we go more into detail and show some special plots. In the end of the Report the conclude with the Conclusion.

## 2 SCENARIO AND DATA DESCRIPTION

### 2.1 Algorithm descriptions

The following is a description of our three algorithms.

- Dionysus[1]: This is a centralised algorithm that is deployed in the controller and it is used as a baseline for our data.
- ez-Segway[1]: This is the most advanced algorithm and is deployed on top of each local controller. The required time cost is reduced based on the Dionysus algorithm.
- p4updata[1]: An algorithm for offloading controllers to the data plane and implemented in P4, which relies heavily on each node getting different kinds of update information, update version numbers and distance to the destination. The basic idea is that through traffic state, nodes can verify and coordinate the update process on the data plane. By doing so, the update time is reduced and the risks associated with controller of inconsistencies are avoided.

### 2.2 Data measurement

The data was obtained by the tutor measuring the running times of the above three algorithms in single and multi-flow contexts of three topologies. (Single-flow: adversarial[1],b4[2],i2[3];Multi-flow:b4,fatter,i2).After the measurements, the tutor processed the extreme outliers of "dionysus algorithm in i2 topology in multiple flow, p4update algorithm in adversarial and i2 topology in single flow ", encapsulated the data in csv file and uploaded it on to LRZ-Gitlab.

### 2.3 Topology of Data

Under the folder "100-network-update-measurement" there are two folders "flow-completion-time" and "single-flow-completion-time", which correspond to the multi-flow and single-flow cases respectively. In the "flow-completion-time" folder there are folders "b4, fatter, i2", which correspond to the topology of each of the three multi-flow scenarios. The same is true for the "single-completion-time" folder, which corresponds to the "adversarial,b4,i2" topology. In each of these folders there are three csv files, corresponding to the flow time values measured for the three algorithms mentioned above. For this, we collated the measured values for each topology in each case to obtain a total of 6 sets of csv data.

# 3 METHODS AND RESULTS

In the following analysis, we use the data in flow b4 as an example and include his analysis results in the report because it is representative and we use the same method for all other data and get the same conclusion. The complete results and images have been uploaded to LRZ-Gitlab.

## 3.1 Mandatory Analysis

- According to the statement of "2.2 Data measurement", our data are obtained from experimental measurements made by our instructor for three algorithms in three different topologies in the single and multi-stream cases. First we analyzed the data for the three algorithms of the topology flowb4 and calculated his mean, median, and variance.

p4updata mean: 969 ms
p4updata median: 951 ms
p4updata variance: 6.8 ms
dionysus mean: 1362 ms
dionysus median: 1128 ms
dionysus variance: 24ms
ez segway mean: 1592ms
ez segway median: 1569ms
ez segway variance: 33ms

- The explanatory variable is the thirty experiments for each algorithm under b4 topology, and the response variable is the corresponding update time.

- **Scatterplot:**

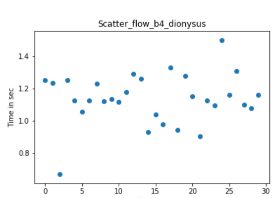


**Figure 1: Scatter of Dionysus algorithm**

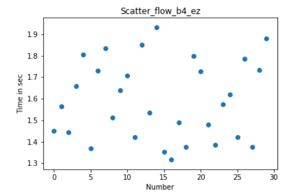


**Figure 2: Scatter of EZ-Segway algorithm**

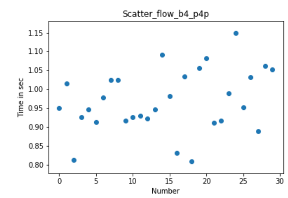


**Figure 3: Scatter of P4-Update algorithm**

- **Dionysus Histogram:**

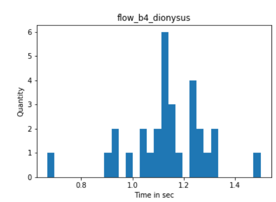


**Figure 4: Histogram of Dionysus algorithm**

The histogram is Unimodal with one prominent peak. There are two small outliners on both sides.

- **ez_segway Histogram:**

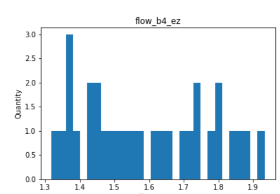


**Figure 5: Histogram of EZ-Segway algorithm**

EZ-Segway: The plot is bimodal with two prominent peaks at 12 und 17. There are almost no out liners.

- **P4update Histogram:**

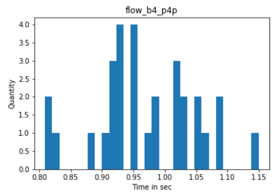


**Figure 6: Histogram of P4-Update algorithm**

P4update: The graph is Unimodal with one prominent peak at 0.95. There are bigger outliners on both sides.

- **Boxplot:**

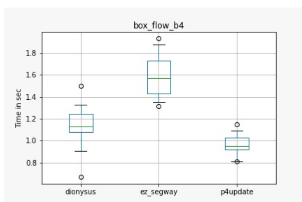


**Figure 7: Boxplot**

- **Violin plot:**

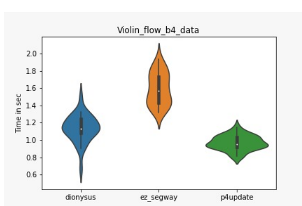


**Figure 8: Violinplot**

Dionysus: The median of the Dionysus is represented with the white dot and is approximately at 1.1. The dataset has one mode. From the shape we can conclude that the distribution is normal.

Ez_segway: The median of the Ezsegway is higher with appox. 1.6. From the shape we could conclude that there are two modes.

P4update: The median of the p4update is with appox. 1.0 lower as compared to the others. The shape shows us that the data is normal and has one mode.

- **Point estimate for a parameter of interest:**

Dionysus: 1362 ms
EZ-Segway: 1592 ms
P4-Update: 969 ms

- **Independence and success failure criterion for a sample and normality check:**

The data is sampled random and therefor the samples obersations are independent. As we have 30 Samples per Dataset and no particularty extreme outliers at the histogram we can assume that the sampling distrubution is nearly normal.

- The mean and variance we have obtained in the above. We have chosen %90,%95,and %99 confidence levels for the study respectively. Their corresponding z-values are 1.64,1.96 and 2.58, respectively. By $\bar{x} \pm Z_{\alpha/2} \cdot \frac{\alpha}{\sqrt{n}}$ We obtained the following confidence intervals.

**confidence interval:flow b4 data**

**confidence interval for dionysus:**

confidence interval of 90% : (0.8802575076469931, 1.392340759019568)
confidence interval of 95% : (0.8312067186250542, 1.441391548041507)
confidence interval of 99% : (0.7353397339286212, 1.53725853273794)

**confidence interval for ezsegway:**

confidence interval of 90% : (1.290866892163497, 1.8944051745032087)
confidence interval of 95% : (1.2330559231963245, 1.9522161434703813)
confidence interval of 99% : (1.120067663725452, 2.0652044029412537)

**confidence interval for p4update:**

confidence interval of 90% : (0.8333384657414116, 1.104662400925529)
confidence interval of 95% : (0.8073492285116327, 1.130651638155308)
confidence interval of 99% : (0.7565547386072031, 1.1814461280597375)

- Because we have more than two groups of data under b4 topology, and the data satisfy the three conditions of Independence, Approximately Normal and Constant Variance, we can perform ANOVA analysis with the help of F-test. $H_0$: three algorithms (dionysus, ez_segway, p4update) have the same average flow update time,$H_A$: the average flow update time of the algorithms varies across three algorithms. For different significance levels, we can reject $H_0$ through python code implementation, So we can conclude that different algorithms do affect the flow update time. To further compare which algorithm is more efficient, we perform the analysis of the difference of two means with the help of T-test. Since we are studying the superiority of P4update algorithm here, we perform T-test on P4update and dionysus,ez_segway. $H_0$: there is no difference in flow update time between dionysus and P4update ($\mu_{p4p} - \mu_d = 0$); $H_A$:there is some difference in flow update between dionysus and P4update ($\mu_{p4p} - \mu_d < 0$). Through Python Code implementation, we can reject $H_0$, That is, the P4update algorithm is more efficient than the dionysus algorithm. The same method is used for the T-test of P4update and ez_segway, and we can also conclude that the flow update time of P4update is shorter compared to ez_segway. Therefore the P4update algorithm is the best algorithm among the three algorithms.

- After discussion with our supervisor, we decided to set the minimum effect size of the algorithm update time to 50 ms and the required significance level to 0.05. By running the code, we obtained that the required sample size for the t-test of P4update and dionysus is 48 and for P4update and ez_segway is 63.

- As with the calculation of sample size, we assume that the minimum effect size is 0.05 and the required significance level is 0.05. By running the code, we obtained that the power for the T-test of P4update and dionysus is 34.3% and for P4update and ez_segway is 27.5%.

- **Linear Regression:**

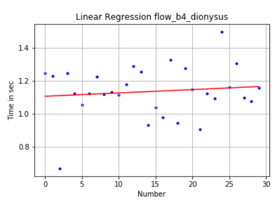


**Figure 9: Linear Regression of Dionysus algorithm**

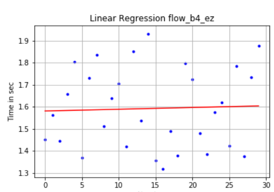


**Figure 10: Linear Regression of EZ-Segway algorithm**

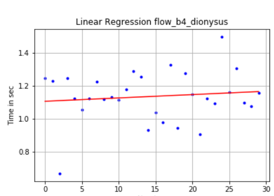


**Figure 11: Linear Regression of P4-Update algorithm**

## 3.2 Additional Analysis

In order to get a clearer picture of the distribution and trend of the points, we additionally performed a strip plot and linear regression on the data.

- We can see from the linear regression plot that the flow update times of different algorithms approximately tend to be a constant straight line, which means that the time of each update falls around this straight line. The flow update time of different algorithms all tend to be a constant value.

- With the strip plot, we can clearly know the distribution of individual points and the update time of points, which is convenient for us to find outliers and analyze them. By observation, we find that most of the points in p4update algorithm are lower than the points in the other two algorithms, which means that the update time of p4update algorithm is shorter and the algorithm is more efficient.

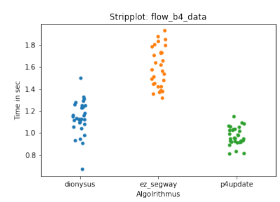


**Figure 12: Stripplot of flow b4**

# 4 CONCLUSION

With the help of the method mentioned above, we perform ANOVA analysis with the help of F-test on the data in each topology under single flow and flow, and conclude that different algorithms do affect the corresponding flow update time. In order to analyze more accurately whether there is a better algorithm, we conducted a two-by-two analysis of the three algorithms with the help of the T-test, and finally came up with that the P4update algorithm takes less time to update on average.

# REFERENCES


[1] Zikai Zhou, Mu He, Wolfgang Kellerer, Andreas Blenk, and Klaus-Tycho Foerster. 2021. P4Update: fast and locally verifiable consistent network updates in the P4 data plane. In Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies. 175-190.

[2] S.Jain and A.Kumar, "B4:experience with a globally-deployed software defined wan", ACM SIGCOMM Computer Communication ReviewVolume 43Issue 4October 2013 pp 3-14 https://doi.org/10.1145/2534169.2486019

[3] "Contact Us - Internet2". Internet2. Retrieved 2013-11-14.