Assignment Report PA1

Programming Assigmnet 1. Congestion Control Contest

RERE GILANG NURI AULADI [ 1301160801 ], MUHAMMAD DERIAN AKBAR [ 1301164722 ],

RIZKI NURHALIZA HARAHAP [ 1301164734 ], RAFANZHANI ELFARIZY [ 1301160792 ]

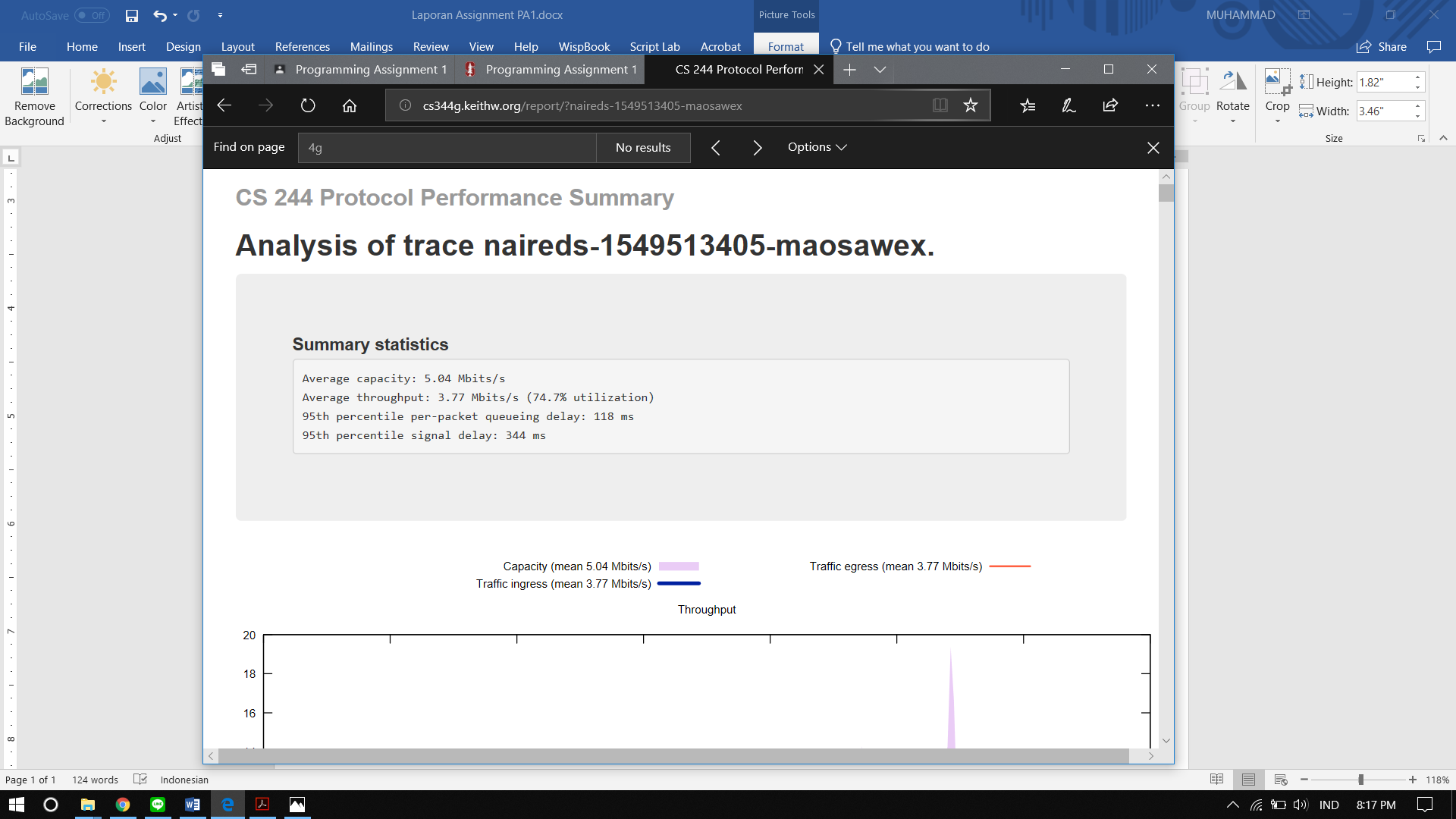
**Keyword *: TCP, window-size, AIMD, congestion-window, throughput, delay***

Pengujian simulasi *congestion-control* 4G LTE dalam Linux environment dan menggunakan bantuan mininet. Dengan codingan yang telah dibuat oleh situs dari *stanford education*. Codingan tersebut berbahasa C++ dimana didialamnya terdapat simulasi pengiriman packet yang terdefinsi dan juga window size yang terdefinisi, code yang kami ubah adalah pada *controller.cc.* Dengan memodifikasi code dan juga berimbas dengan perlakuan dari *congestion-controller* nya, dalam code tersebut bisa mengkontrol bagaimana sender menentukan kapan untuk mengirimkan data-nya. Semakin banyak data yang dikirimkan, semakin besar pula throughput akan terjadi. Dalam pengimplementasiannya, resiko yang kami dapati adalah delay dan queue yang terjadi.

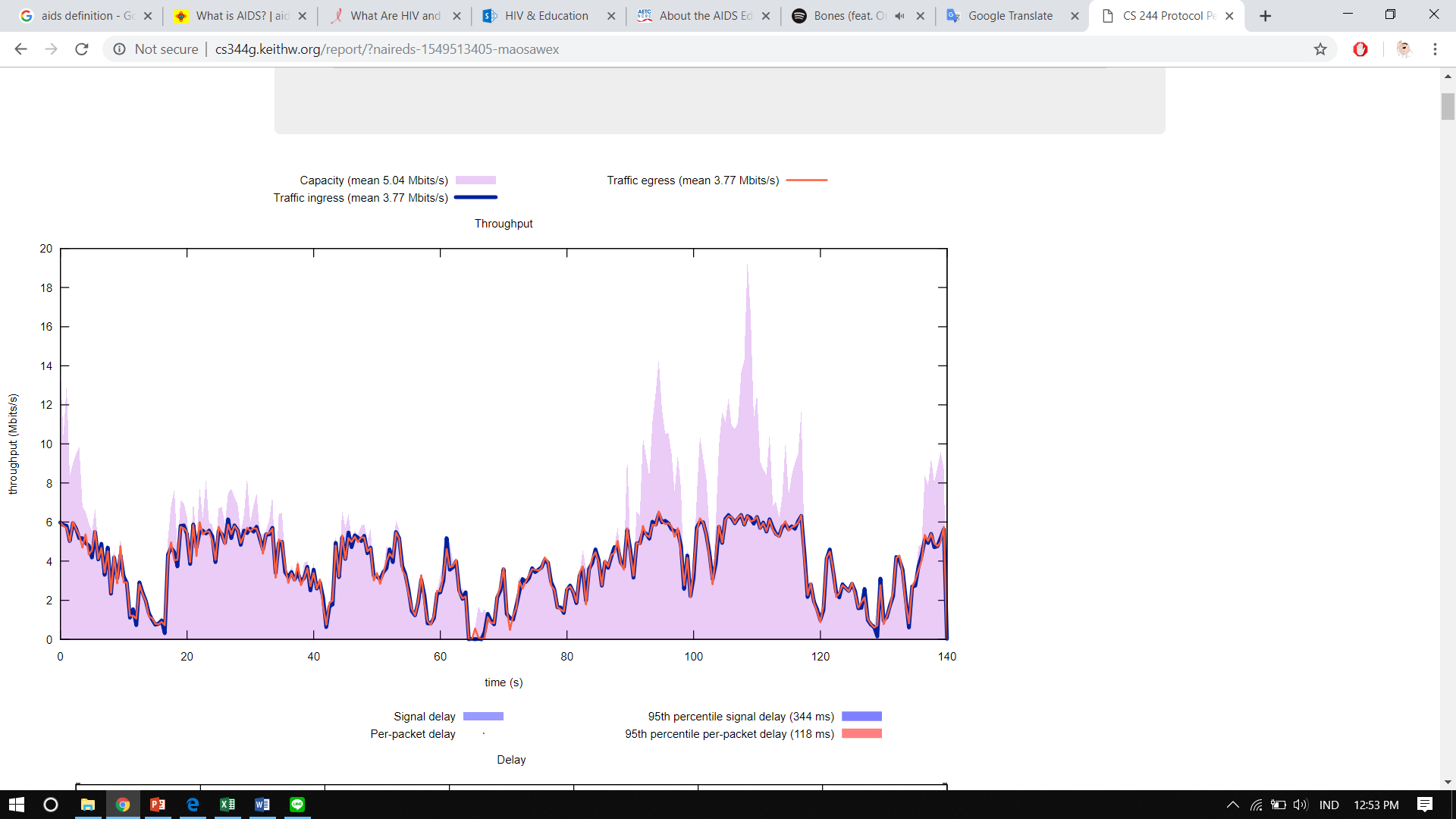
Exercise 1 : editing the ***controller.cc***

Pada permulaan percobaan adalah pengubahan beberapa vairabel yang terdapat pada file *controller.cc.* Variable tersebut merupakan besarnya *window-size* yang didefinisikan oleh simuasi. *Window-size* berpengaruh terhadap laju pengiriman paket, menentukan seberapa besar/banyak paket dapat muat dalam satu window.

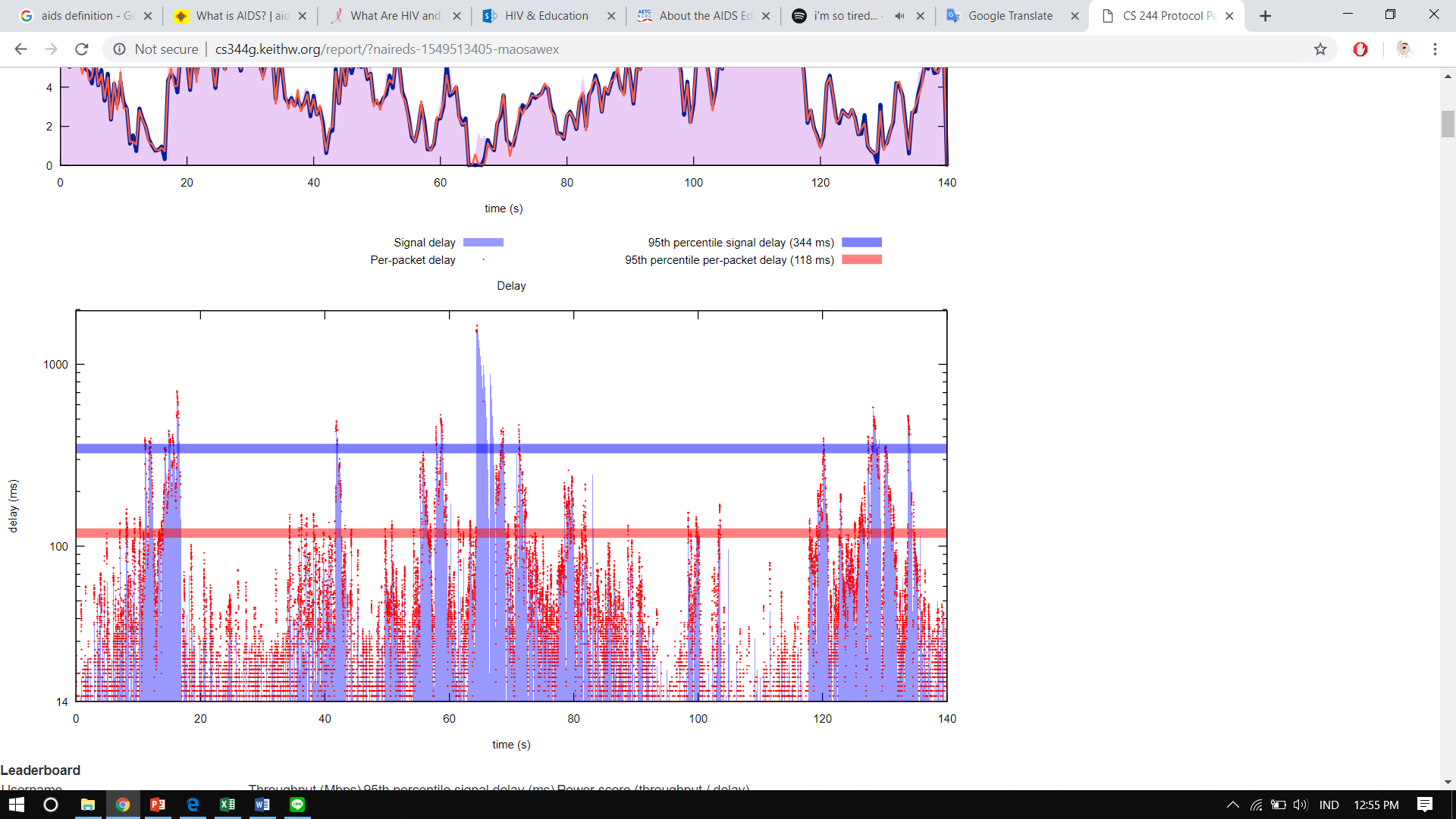
Hasil dari simulasi kami adalah sebagai berikut, dengan username *naireds* kami mencoba runningan code-nya secara DEFAULT(belum terjadi perubahan). Dan inilah hasilnya, [*http://cs344g.keithw.org/report/?naireds-1549513405-maosawex*](http://cs344g.keithw.org/report/?naireds-1549513405-maosawex)



***Gambar 1.0*** *Hasil dari run code controller.cc default*



***Gambar 1.1*** *Graph throughput controller.cc default*



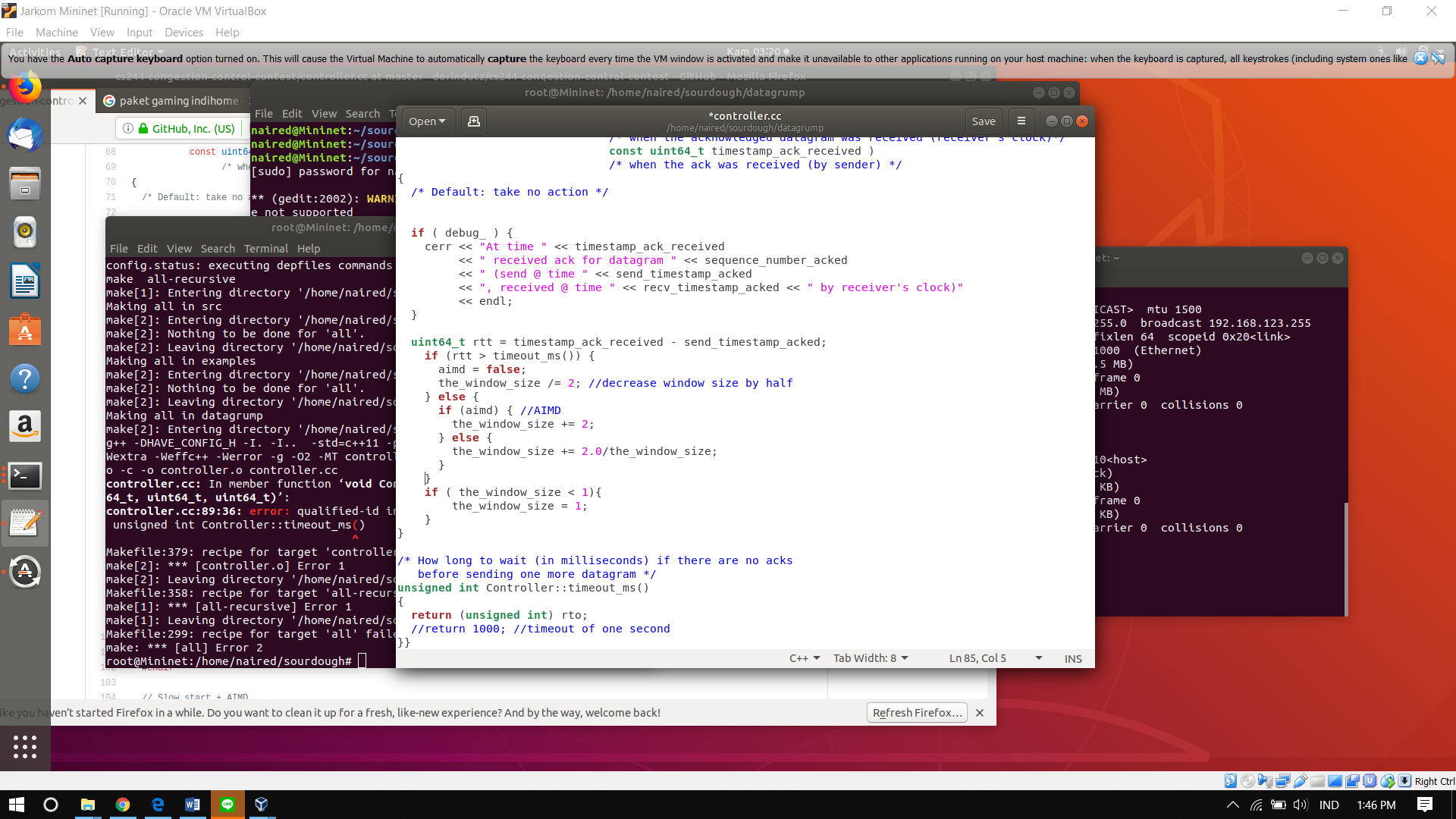
***Gambar 1.2*** *Graph 95-percentile signal delay controller.cc default*

Pengujian diatas merupakan hasil runningan DEFAULT yang belum dilakukan perubahan apapun. Disini kami hanya akan mencoba apakah code program dapat dirun dengan swajarnya dan juga menjadi batasan / acuan dalam menemukan throughput dan delay yang seoptimal mungkin.

Exercise 2 : Implement AIMD

Setelah melakukan belasan perobaan dan juga melihat paper serta referensi orang yang telah melakukan ini, kami mengerti bahwa yang kami butuh kan adalah mengatur *congestion-control-*nya. Agar mendapatkan nilai throughput yang stabil dan tinggi dan juga delay yang efisien, mengubah perilaku *controller.cc* merupakan hal yang tepat.

AIMD merupakan algoritma yang memungkinkan untuk menangani congestion dengan cara mengatur window size. Congestion disebabkan oleh adanya timeout. Window size naik secara linier, ini yang disebut dengan adiptive incresment. Sedangkan ketika terjadi packet loss maka window size akan bernilai setengah dari nilai sebelumnya, ini yang disebut dengan multiplicative decrease. Pada kasus window size kurang dari 1, window size akan bernilai 1.



***Gambar 2.0*** *tambahan code AIMD pada controller.cc*

Dengan menambah beberapa baris code mengenai AIMD, hasil menunjukan sangat baik. Delay yang sekain ratus sekarang menjadi puluhan saja. Dengan kata lain dengan adanya *congestion-control* ini, sangat membantu pengiriman paket data dengan sangat memerhatikan nilai packet-loss serta delaynya.

Exercise 3 : Join **The Contest**

Setelah mengetahui fungsi serta mengimplementasikan AIMD, kami mulai bergabung untuk mencoba lagi contestnya. Pada uji coba kali ini, kami mengubah sedikit parameter yang ada berupa window size yang bernilai 15, dengan kenaikan window size bertambah 2, request timeout bernilai 80 ms. Dari parameter tersebut didapatkan hasil throughput 2.65 mbps dengan delay 93 ms. Sehingga score yang didapat berdasarkan perhitungan throughput dibagi dengan delay percentile.

[*http://cs344g.keithw.org/report?rapanSel-1550082959-zoheebae*](http://cs344g.keithw.org/report?rapanSel-1550082959-zoheebae)

A screenshot of a social media post

Description generated with very high confidence

***Gambar 3.1*** *Hasil dari controller.cc terbaik*

A screenshot of a social media post

Description generated with very high confidence

***Gambar 3.2*** *Throughput controller.cc terbaik*

A screenshot of a social media post

Description generated with very high confidence

***Gambar 3.3*** *Delay controller.cc terbaik*

Kami berasumsi bahwa, Request Time Out berpengaruh ketika semakin kecil request time out maka semakin besar congestion yang akan terjadi dan akan semakin banyak packet yang diretransmit. Percobaan dalam mendapatkan nilai terbaik kami lakukan dalam beberapa data yang telah kami rangkum dalam tabel. Dalam tabel dibawah ini semua menggunakan implementasi AIMD, dengan atribut yang diubah adalah RTO , Window-Size dan Increment pada congestion-controlnya.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Percobaan | Throughput | Delay Percentile | **POWER SCORE** | Request Time Out | Window Size | Increment |
| #1 | 0.43 | 66 | **6.515151515** | 20 | 14 | 2 |
| #2 | 2.16 | 84 | **25.71428571** | 80 | 14 | 1 |
| #3 | 2.71 | 96 | **28.22916667** | 80 | 14 | 2 |
| #4 | 2.65 | 93 | **28.49462366** | 80 | 15 | 2 |
| #5 | 2.65 | 94 | **28.19148936** | 80 | 30 | 2 |
| #6 | 2.18 | 87 | **25.05747126** | 80 | 60 | 1 |
| #7 | 2.65 | 95 | **27.89473684** | 80 | 60 | 2 |
| #8 | 3.21 | 120 | **26.75** | 100 | 60 | 2 |
| #9 | 3.5 | 127 | **27.55905512** | 110 | 50 | 2 |

***Tabel 3.0*** *Percobaan perubahan atribut pada controller.cc*

Kesimpulan :

Algoritma AIMD membantu untuk congestion control, perbedaan yang cukup signifikan saat AIMD di implementasikan dan saat tidak di implementasikan. Karena algoritma AIMD memperkecil delay yang terjadi dengan memangkas CNWD. CNWD dibagi 2 karena terjadi packet loss, dan akan mecoba untuk ke nomor *congestion-window* setelah dibagi oleh 2. Delay akan terjadi saat pemberlakuan AI dan MD ini, seberapa lama delay terjadi yang diperngaruhi juga oleh *window-size* yang terdefinisi sebelumnya.

*Github* :

[*https://github.com/muhammadderian/programmingAssignment1-jarlan*](https://github.com/muhammadderian/programmingAssignment1-jarlan)