

ASSIGNMENT

PERFORMANCE ANALYSIS OF COMPUTER NETWORK

Dosen Pengampu

Dr. Bayu Erfianto



Nama Mahasiswa

Dani Agung Prastiyo 1301154646

Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

2018

ASSIGNMENT LATIHAN

1. Why does the **Segment Sequence Number** remain unchanged (indicated by a horizontal line in the graphs) with every drop in the congestion window?

Jawab :

Berdasarkan pada uji coba yang telah dilakukan, hal itu disebabkan karena congestion dalam jaringan dan paket-paket yang di drop. Congestion Window hanya berkurang ukurannya ketika mendeteksi adanya timeout, yang terjadi sejak drop paket 5% yang ditugaskan ke IP Cloud

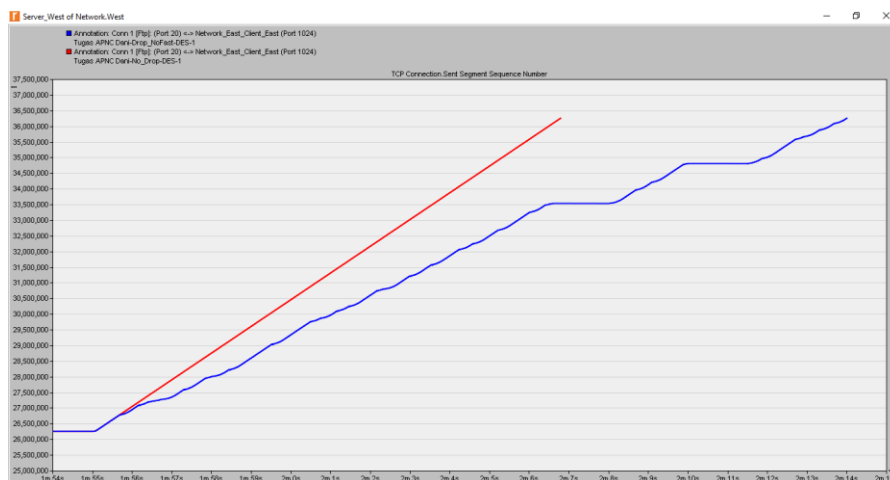
2. Analyze the graph that compares the **Segment Sequence** numbers of the three scenarios. Why does the **Drop_NoFast** scenario have the slowest growth in sequence numbers?

Jawab :

Pada Skenario Pengujian Drop_NoFast memiliki kenaikan paling lambat paket karena ada 5% paket drop dan opsi retransmit dinonaktifkan dalam pengaturan TCP. Sehingga ada waktu jeda di dalam jaringan dalam panjang periode timeout TCP sebelum paket dikirim ulang. Sedangkan Drop_Fast, karena fast retransmit diatur, hanya tiga putaran perjalanan senilai waktu yang lewati paket sebelum paket mengirimkan ulang, sehingga pemulihan dari paket yang turun jauh lebih cepat. Sedangkan pada NoDrop, tidak ada paket yang hilang dan semua data dikirim secara berurutan dengan lancar.

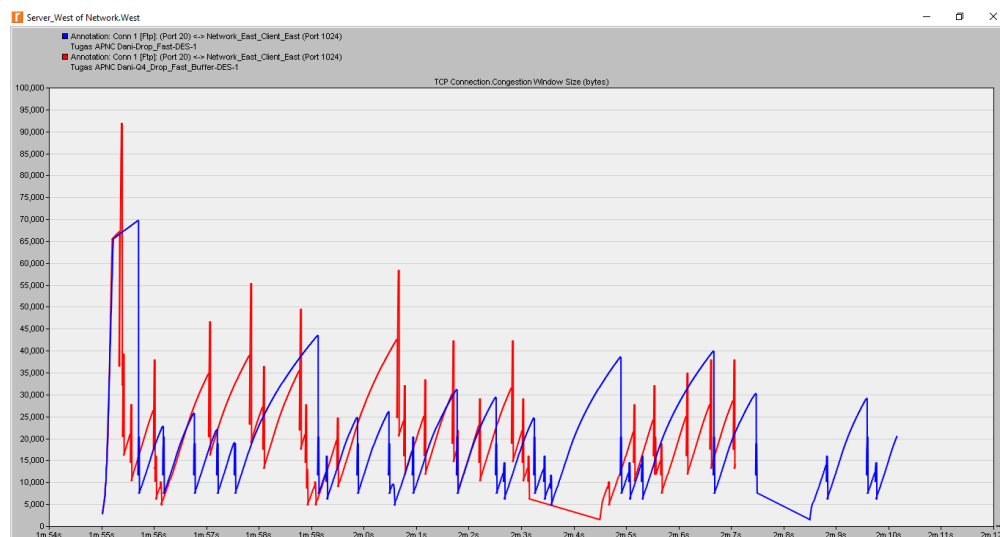
3. In the **Drop_NoFast** scenario, obtain the overlaid graph that compares **Sent Segment Sequence Number** with **Received Segment ACK Number** for **Server_West**. Explain the graph.

Jawab :



Grafik diatas secara visual menunjukkan hilangnya paket dan transmisi ulang paket selama periode timeout dalam jaringan. Dapat dilihat bahwa garis biru (sent segment sequence number) naik di atas garis merah (received segment ack number) dan kemudian garis merah mencapai tingkat yang sama beberapa detik kemudian. Ini berarti bahwa klien tidak acknowledge hingga segmen periode waktu tersebut.

4. Create another scenario as a duplicate of the **Drop_Fast** scenario. Name the new scenario **Q4_Drop_Fast_Buffer**. In the new scenario, edit the attributes of the **Client_East** node and assign 65535 to its **Receiver Buffer (bytes)** attribute (one of the **TCP Parameters**). Generate a graph that shows how the **Congestion Window Size (bytes)** of **Server_West** gets affected by the increase in the receiver buffer (compare the congestion window size graph from the **Drop_Fast** scenario with the corresponding graph from the **Q4_Drop_Fast_Buffer** scenario.)



Garis biru menunjukkan peningkatan receive buffer. Dengan peningkatan buffer, banyaknya byte dapat diterima dalam jumlah waktu yang sama dan hanya beberapa periode paket yang hilang terjadi sebelum file 10 MB penuh ditransmisikan. Sehingga menghasilkan seluruh sesi percobaan dalam sekitar seperempat waktu dibandingkan dengan buffer penerima yang lebih kecil.

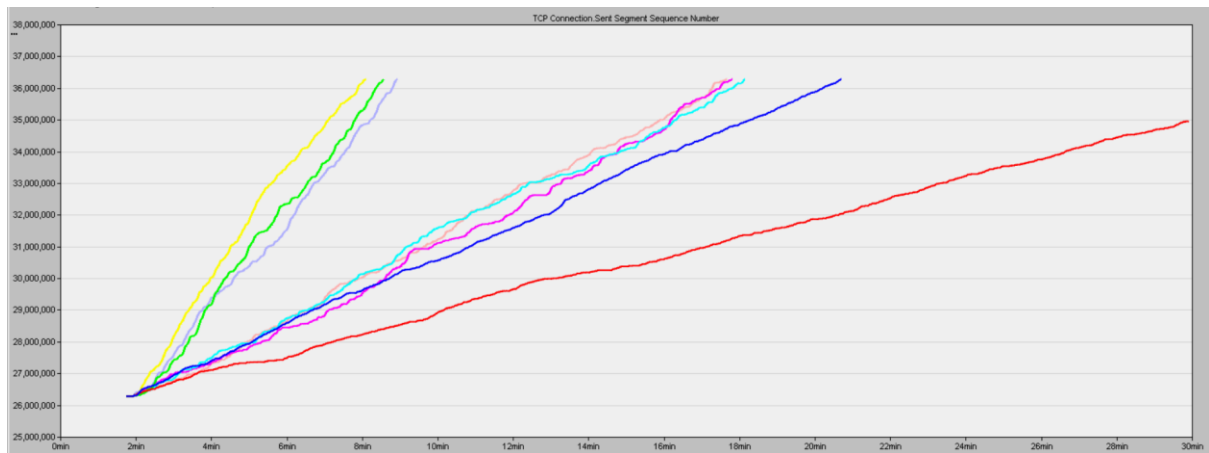
Assignment 1

Tampilkan statistik hasil simulasi anda. Algoritma TCP Congestion Control manakah yang memberikan throughput paling besar serta lebih stabil ketika network mengalami drop rate ratio yang cukup tinggi ? Mengapa?

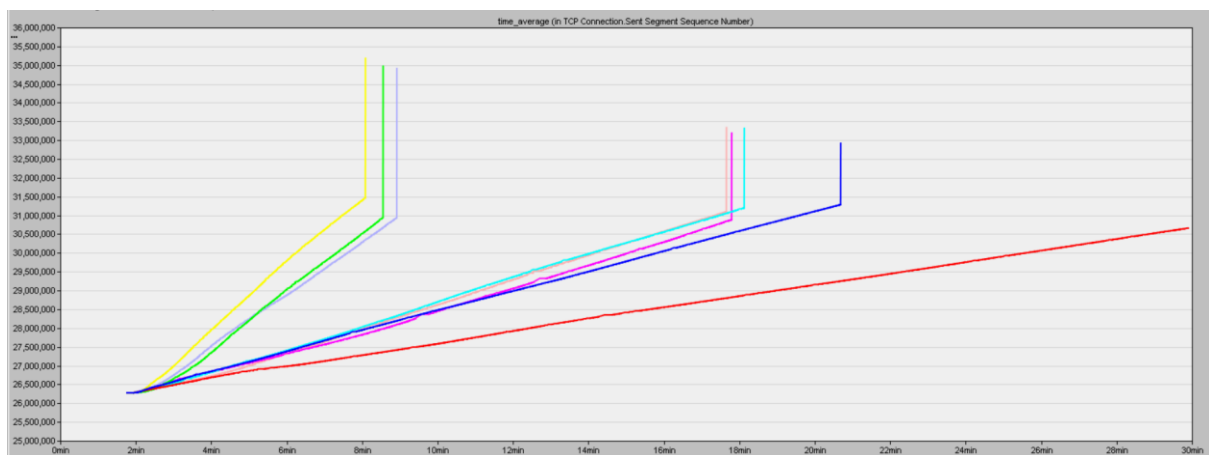
Jawab :

Berdasarkan pada Gambar 1-4 dapat disimpulkan bahwa pada kasus pengujian dengan dengan drop rate paket sebesar 2 % dan 5 % pada IP Cloud Internet dengan menjalankan simulasi selama 30 menit, dihasilkan besarnya nilai throughput rata-rata TCP CUBIC mengungguli TCP lainnya. Selisih nilai 2.5 Mbits/second dari TCP New Reno untuk pengujian 2% drop rate paket, sedangkan pada pengujian packet drop rate sebesar 5 % selisih 4.3 Mbits/seconds nilai dari New Reno. Hal ini disebabkan karena estimasi bandwidth yang dilakukan oleh TCP Cubic pada jaringan cukup efektif untuk menanggulangi packet loss dengan pengiriman FTP sebesar 1 GB yang terjadi pada jaringan. CUBIC tidak langsung menurunkan nilai cwndnya secara multiplikatif, melainkan meletakkan nilai cwnd pada nilai Wmax sampai waktu tertentu. Selain itu perubahan RTT tidak terlalu mempengaruhi pertumbuhan cwnd saat terjadi kongesti pada jaringan.

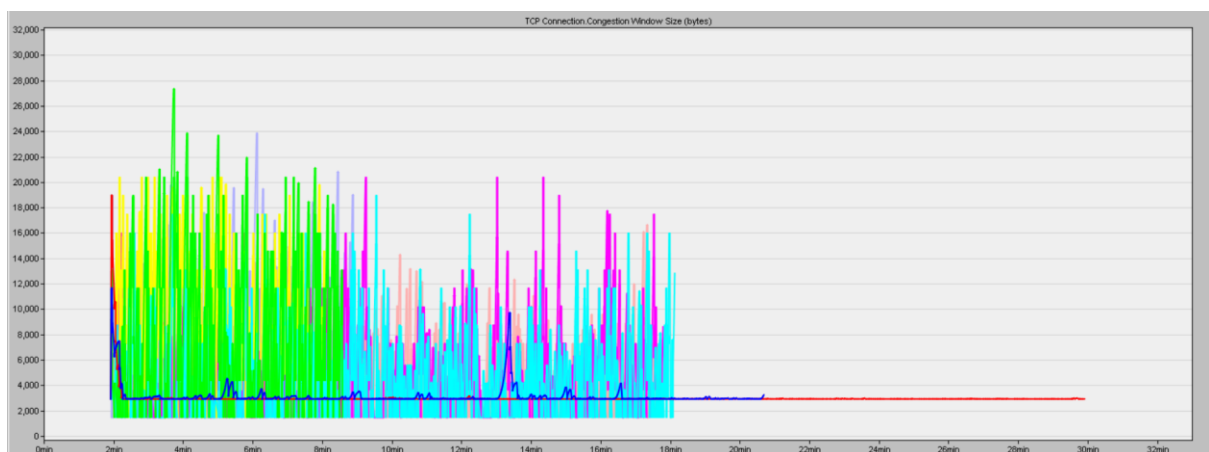
Disamping itu dari pengamatan besarnya segment yang dikirim pada gambar 2, Cubic memiliki jumlah segment transmisi lebih besar sehingga ketika terjadi loss, maka diperlukan retransmisi yang lebih sedikit dan membuat throughput menjadi semakin besar. Performa Cubic untuk delay transmisi data pada jaringan dengan delay besar menunjukkan keunggulan yang terpaut cukup jauh. Sama seperti pada pengukuran nilai throughput, nilai delay yang besar dimiliki oleh TCP lainnya. Hal ini disebabkan karena RTT yang berubah-ubah sehingga menyebabkan estimasi bandwidth seperti TCP New Reno, Reno dan Tahoe menjadi tidak efektif. Dapat dilihat dari nilai deviasi pada gambar dibawah ini, TCP CUBIC lebih besar dan cenderung paling stabil dari TCP lainnya, hal ini menunjukkan bahwa besarnya throughput relatif stabil saat melewati jaringan Loosy.



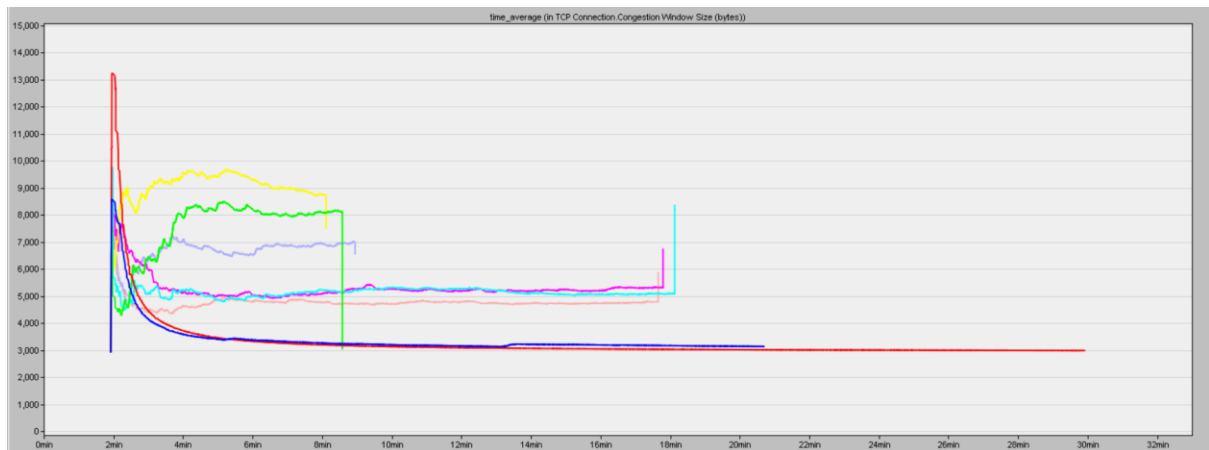
Gambar 1 Perbandingan Sent Segment Sequence Number (As It)



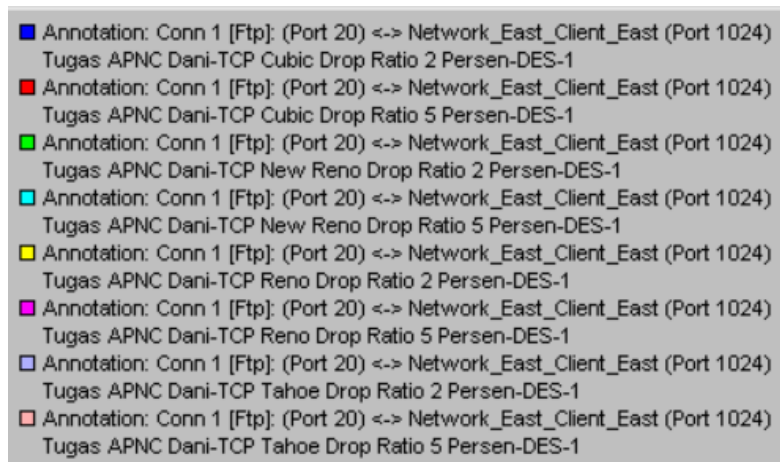
Gambar 2 Perbandingan Sent Segment Sequence Number (Time Average)



Gambar 3 Perbandingan Congestion Windows Size (As It)

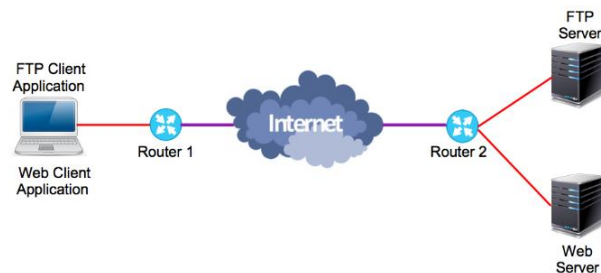


Gambar 4 Perbandingan Congestion Windows Size (Time Average)



Gambar 5 Legend Nomor 1

ASSIGNMENT 2



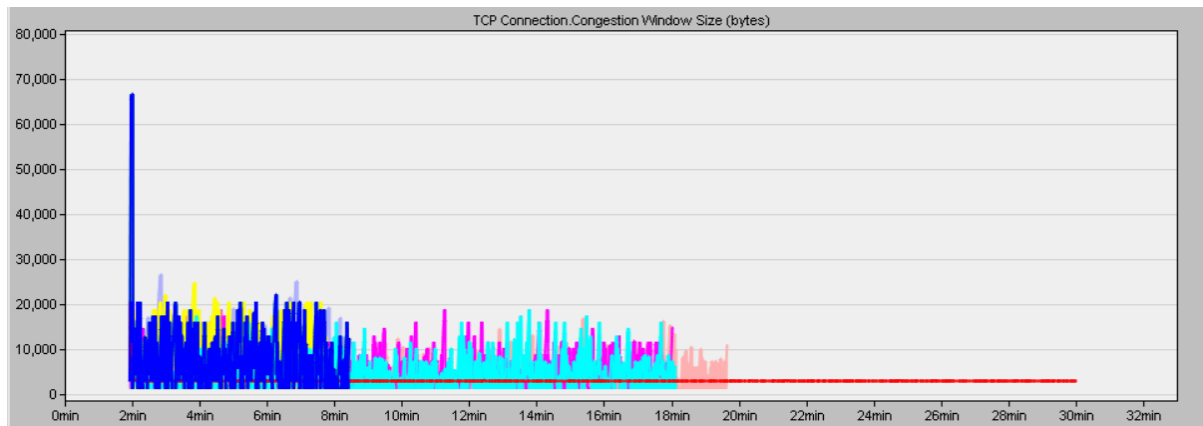
Gambar 6 Topologi Assignment 2

1. Tampilkan statistik hasil simulasi anda (Congestion Windows Statistics). Algoritma TCP Congestion Control manakah yang memberikan throughput paling besar serta lebih stabil ketika terdapat dua aplikasi yang menjalankan TCP Congestion Control serta network mengalami drop rate ratio yang cukup tinggi ? Mengapa?

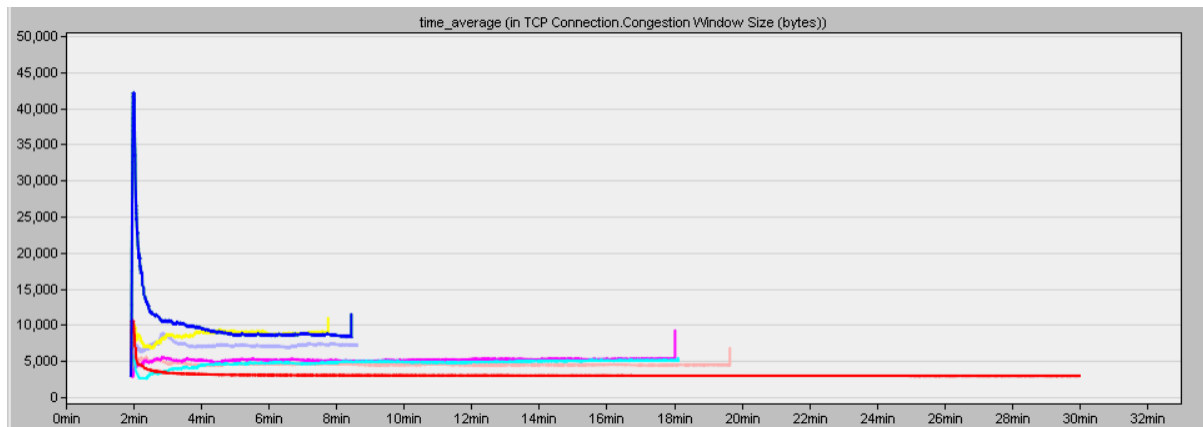
Jawab :

Berdasarkan simulasi skenario yang sudah dilakukan yang ditambahkan pada gambar 7 dan 8 menyimpulkan bahwa Cubic lebih fairness dengan memberikan throughput paling besar dan lebih stabil selama simulasi dilakukan dengan drop rate 2% dan 5 %, Cubic TCP berkinerja cukup baik dibandingkan dengan TCP lainnya. Mekanisme congestion control kompatibel dengan waktu both short dan long round trip. Cubic mampu menyeimbangkan ukuran congestion window size. CUBIC tidak langsung menurunkan nilai cwndnya secara multiplikatif, melainkan meletakkan nilai cwnd pada nilai Wmax sampai waktu tertentu. Selain itu perubahan RTT tidak terlalu mempengaruhi pertumbuhan cwnd saat terjadi kongesti pada jaringan.

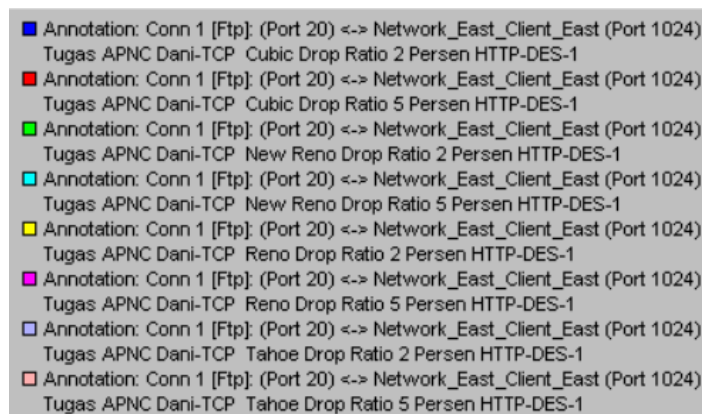
Perbedaan utama lainnya antara CUBIC dan TCP lainnya yang dilakukan dalam pengujian, Cubic tidak bergantung pada penerimaan ACK untuk meningkatkan window size. Window size Cubic hanya bergantung pada peristiwa kemacetan terakhir. Dengan RTT yang sangat singkat akan menerima ACK lebih cepat dan karenanya congestion windows mereka tumbuh lebih cepat daripada algoritma TCP lainnya (Terlihat pada statistik hasil pengujian gambar 7-8). Dengan RTT yang lebih panjang. CUBIC memungkinkan untuk lebih fairness karena pertumbuhan window terjadi secara independen dari RTT.



Gambar 7 Congestion Windows Statistics (As It)



Gambar 8 Congestion Windows Statistics (Time Average)



Gambar 9 Legend

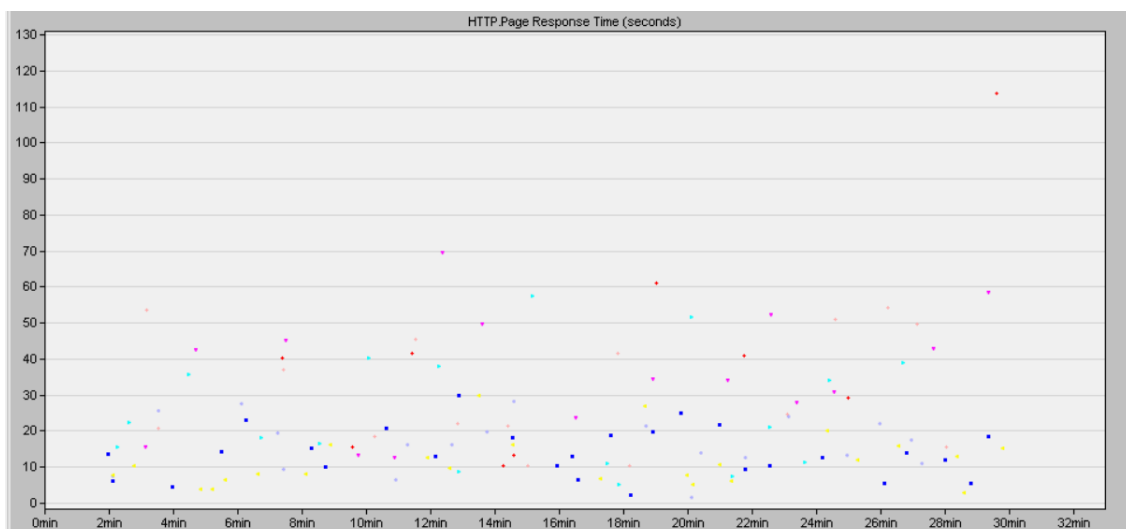
2. Tampilkan statistik hasil simulasi anda (HTTP Page Response Time). Berdasarkan hasil simulasi tersebut, manakah Congestion Control Algorithm yang memberikan HTTP Page Response Time paling singkat? Pada kondisi drop rate ratio berapa ? Mengapa demikian?

Jawab :

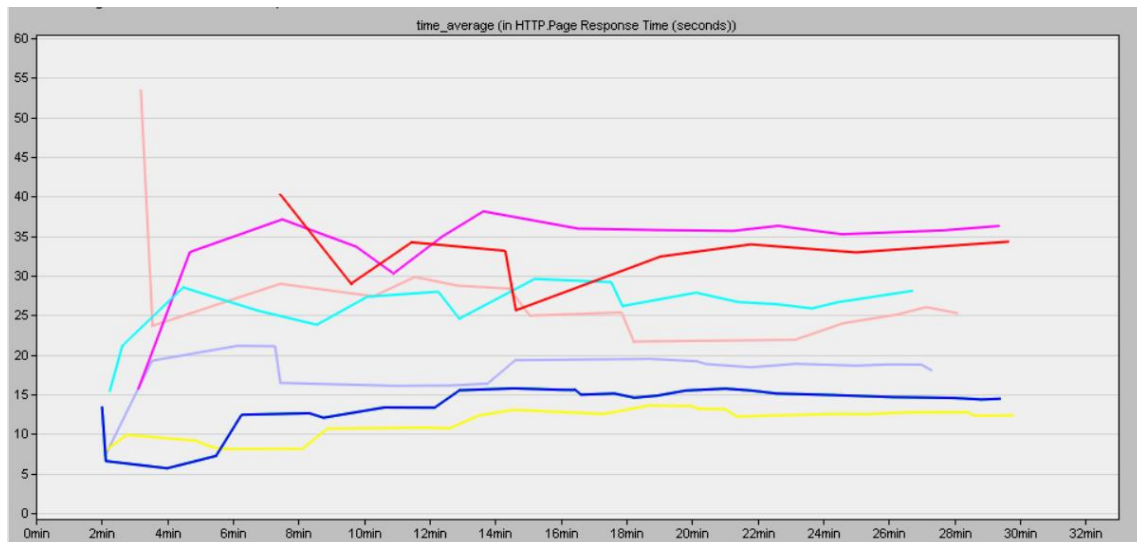
Pada gambar 10 dan 11 menunjukkan hasil simulasi dengan menampilkan HTTP response Time didapatkan kesimpulan bahwa Algoritma congestion windows New reno memiliki Respon time yang paling singkat pada drop ratio 5 persen. waktu respon time reno berhenti di menit ke 26 lebih cepat dibanding algoritma lainnya, Hal ini dikarenakan algoritma TCP New Reno bekerja hanya pada sisi sender, yang bertujuan untuk menangani jalur dengan bandwidth tinggi delay, dengan potensial paket loss karena transmisi atau error lainnya (pipa bocor), dan beban dinamik (dynamic pipe).

Selain pada TCP NewReno proses retransmit lebih jarang terjadi dibanding pada algoritma lainnya walaupun seperti yang diketahui bahwa paket TCP pulih karena adanya proses retransmit. Akan tetapi, proses retransmit tersebut akan menurunkan nilai throughput itu sendiri.

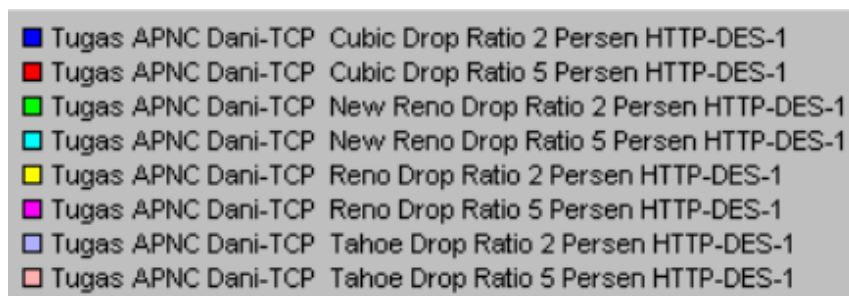
Sedangkan performansi TCP Cubic dalam pengiriman, dikarenakan TCP Cubic seringkali berada di fase slow start, serta membutuhkan waktu yang lama untuk keluar dari fase tersebut yang disebabkan oleh batas threshold slow start yang cukup tinggi dari TCP Cubic



Gambar 10 Perbandingan HTTP Page Response Time (As It)



Gambar 11 Perbandingan HTTP Page Response Time (Time Average)



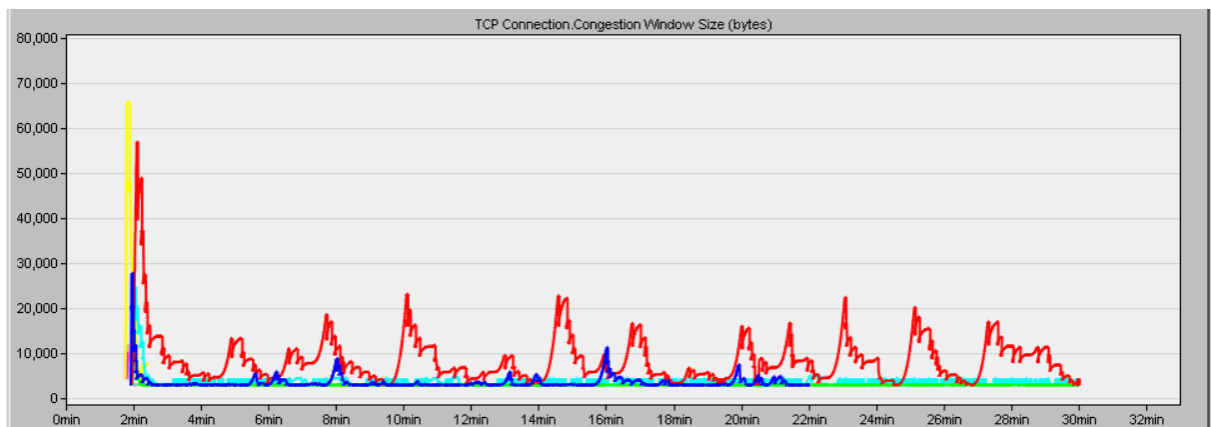
Gambar 12 Legend

ASSIGNMENT 3

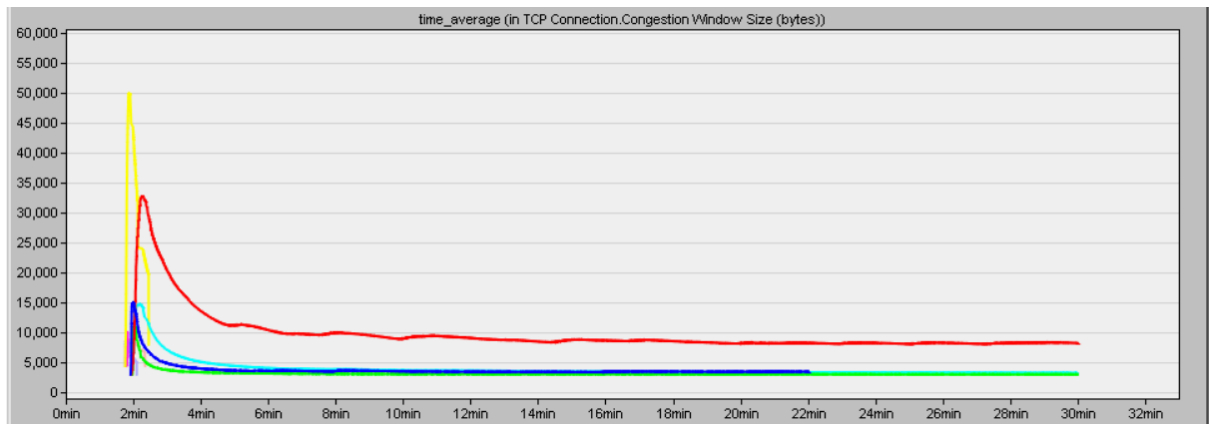
1. Tampilkan statistik hasil simulasi anda (Congestion Windows Statistics). Pada lingkungan manakah TCP CUBIC memberikan throughput paling besar serta lebih stabil ketika terdapat dua aplikasi dijalankan ? Mengapa demikian ?

Jawab :

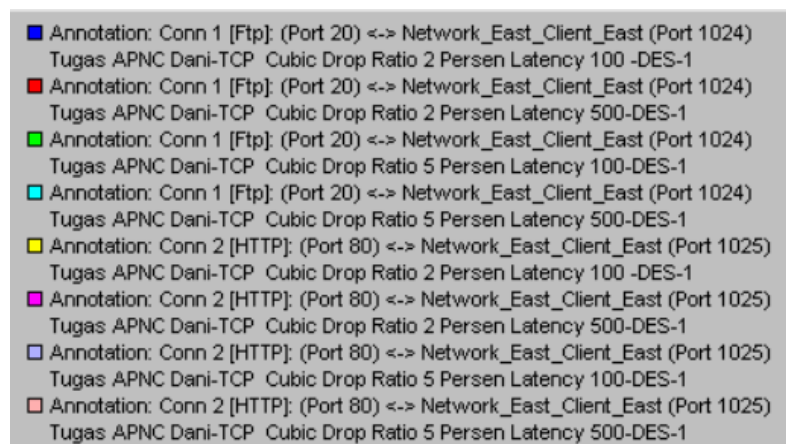
Hasil Pengujian yang tersedia pada gambar dibawah ini ketika terdapat dua aplikasi yang dijalankan bersamaan memberikan kesimpulan bahwa kinerja TCP Cubic dengan latency 100 ms dan menggunakan link internet D3 memberikan throughput paling besar dan lebih stabil. Hal ini karena semakin sedikit latency yang diberikan maka cubic akan bekerja dengan baik, sebaliknya jika semakin besar latency yang diberikan maka akan berpengaruh terhadap pengiriman paket. Selain latency drop rate juga mempengaruhi seperti halnya latency, selain itu jenis bottleneck link bandwidths yang diberikan juga mempengaruhi semakin lebar link yang diberikan kinerja TCP cubic semakin baik. fakta bahwa peningkatan RTT karena packet loss atau kemacetan terus mempercepat pertumbuhan throughput, yang pada gilirannya mengarah ke cwnd yang lebih besar. Sebuah cwnd yang lebih besar mengarah ke lebih banyak paket yang menjangkau jaringan.



Gambar 13 TCP Congestion Window Size (As It)



Gambar 14 TCP Congestion Window Size (Time Average)



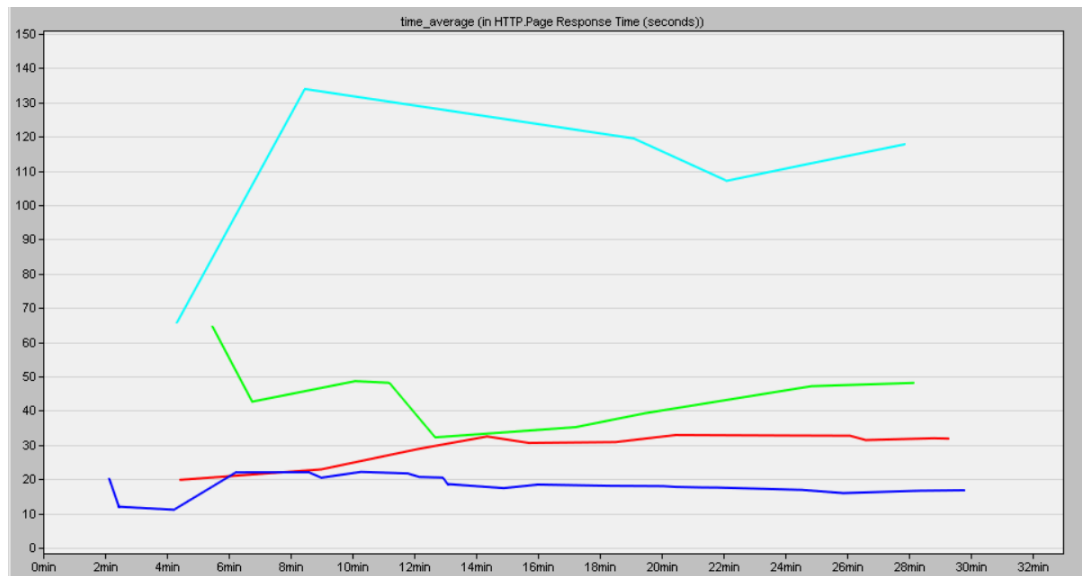
Gambar 15 Legend keterangan gambar diatas

2. Tampilkan statistik hasil simulasi anda (HTTP Page Response Time). Berdasarkan hasil simulasi tersebut, pada lingkungan manakah manakah TCP CUBIC menyebabkan HTTP Page Response Time paling singkat ? Mengapa demikian?

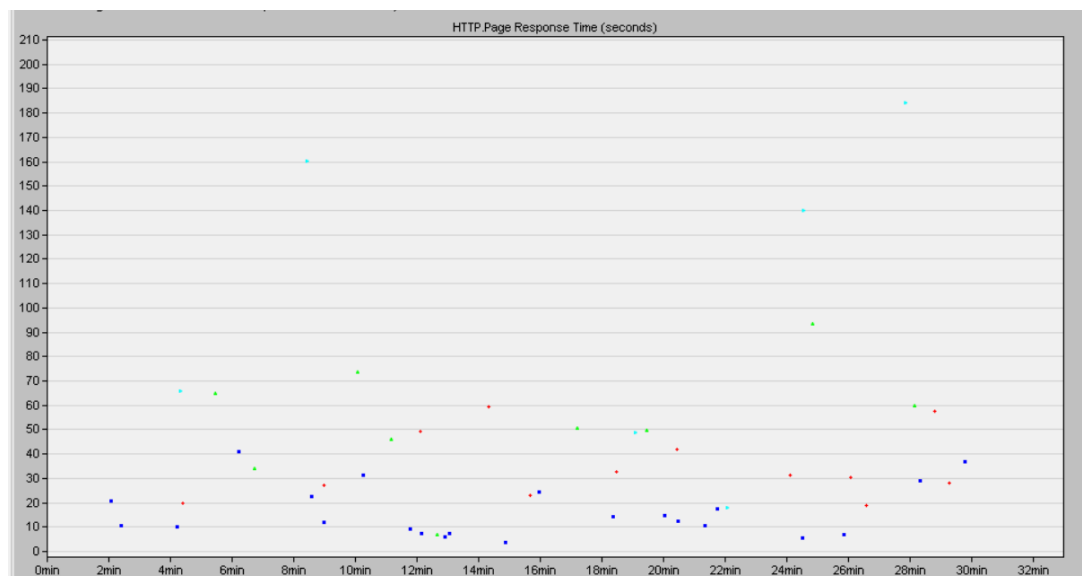
Jawab :

Dari hasil statistik simulasi yang ditunjukkan pada gambar dibawah ini HTTP Page Response Time paling singkat adalah TCP Cubic dengan drop rate 5 % dan latency 100 ms, dengan waktu respon time 26 menit, sedangkan untuk skenario pengujian TCP Cubic lainnya waktu respon time lebih jauh diatas yaitu 30 menit. Drop rate, latency dan bottleneck link bandwidths mempengaruhi transfer rate walaupun TCP yang digunakan sama, Perbedaan signifikan juga terjadi pada drop rate 2% dan 5% respon time dengan drop 5% jauh lebih baik, hal ini karena kinerja TCP Cubic mampu mereduksi losses paket HTTP yang besar karena CUBIC tidak langsung menurunkan nilai cwndnya secara multiplikatif, melainkan meletakkan nilai cwnd pada nilai Wmax sampai waktu tertentu.

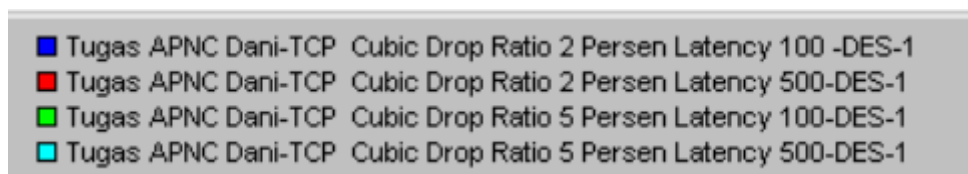
Selain itu perubahan RTT tidak terlalu mempengaruhi pertumbuhan cwnd saat terjadi kongesti pada jaringan.



Gambar 16 HTTP Page Response Time (Time Average)



Gambar 17 HTTP Page response Time (As It)



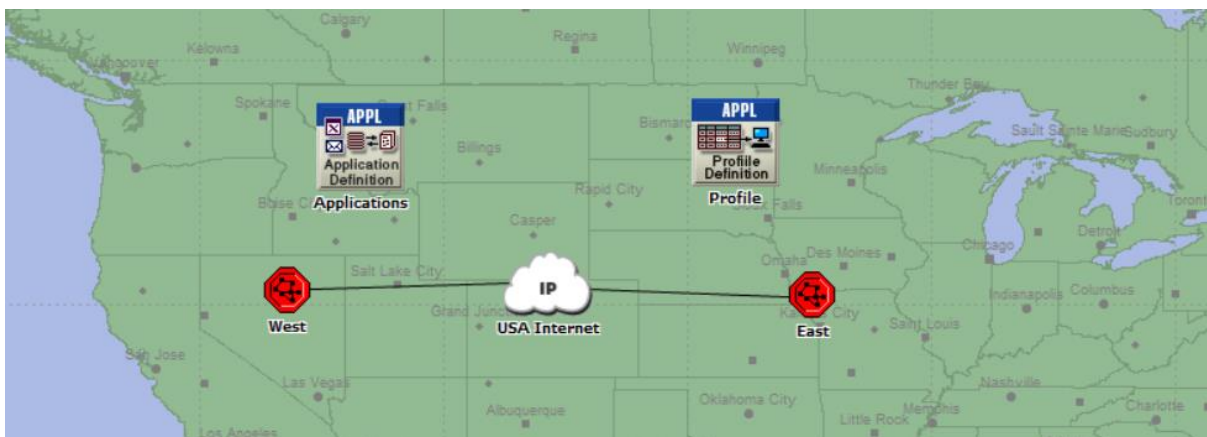
Gambar 18 Legend gambar diatas

KESIMPULAN

Dari simulasi lab yang telah dilakukan dari assignment latihan dan 1-3 membantu memvisualisasikan konsep mekanisme congestion kontrol di TCP. Saya belajar berapa banyak dari beberapa fungsi lain seperti fast retransmit dan ukuran buffer penerima dapat mempengaruhi waktu yang diperlukan untuk mentransfer data. Saya juga memeriksa dengan seksama dan memahami apa yang memengaruhi mekanisme pada segmen sequence number dan segmen acknowledgement.

Pada kesimpulannya Parameter throughput dan fairness pada kondisi jaringan yang berbeda, TCP CUBIC memberikan hasil yang lebih baik dibandingkan dengan TCP lainnya kecuali pada pengujian HTTP Response Page dimana TCP New Reno memberikan nilai yang sedikit lebih baik jika dibandingkan dengan TCP CUBIC dengan selisih nilai yang tidak signifikan waktu respon timenya. TCP CUBIC baik digunakan ketika kondisi jaringan yang memiliki round-Trip-Time berubah-ubah secara acak dan jaringan yang mempunyai losses yang tinggi.

Dokumentasi pengerjaan assignment latihan dan assignment 1-3 saya dokumentasikan di <https://github.com/daniagung/assignment-pacn>



Gambar 19 Gambar Topologi Lab