

ABSTRAK

Tujuan penelitian ini adalah untuk mengetahui unjuk kerja *congestion control* TCP SACK pada jaringan kabel dengan menggunakan antrian *DropTail* dan *Random Early Detection*. Latar belakang penelitian ini adalah pengiriman data melalui jaringan kabel masih mengalami *congestion* sehingga data yang sudah dikirim perlu dikirim ulang dan membuat *throughputnya* rendah. Dengan TCP SACK, pengiriman ulang beberapa paket yang hilang dapat dilakukan karena TCP SACK memberitahukan informasi tentang semua paket yang sudah berhasil diterima, sehingga pengirim hanya tinggal mengirimkan beberapa paket yang hilang saja tanpa mengirim paket yang tidak perlu.

Simulasi menggunakan Network Simulator 2 (ns2) dengan TCP Reno sebagai pembanding *congestion control* dan antrian *DropTail* dan *Random Early Detection* sebagai landasan uji.

Hasil penelitian menunjukkan TCP SACK memiliki jumlah *paket drop* yang lebih banyak daripada TCP Reno karena *congestion window* TCP SACK lebih stabil sehingga paket yang dikirimkan relatif lebih banyak sehingga menyebabkan antrian menjadi lebih sering penuh dan menyebabkan banyak paket yang dibuang daripada TCP Reno, TCP SACK memiliki rata-rata *throughput* yang lebih besar dari TCP Reno karena jumlah paket yang diterima lebih banyak dari TCP Reno. Namun hasil kinerja TCP SACK pada antrian *DropTail* dan *RED* berbeda, TCP SACK dapat bekerja dengan lancar pada antrian *RED* karena paket dibuang secara random, sementara algoritma TCP SACK *congestion control* gagal pada antrian *DropTail* karena sering terjadi *cumulative drop*. Dari hasil penelitian tersebut, dapat disimpulkan bahwa unjuk kerja *congestion control* TCP SACK lebih baik daripada TCP Reno, dan TCP SACK tidak mampu mengatasi *cumulative paket drop*.

Kata Kunci : TCP SACK, TCP Reno, *Random Early Detection*, *DropTail*, *packet drop*, *congestion control*, *congestion window*, *throughput*.

ABSTRACT

The purpose of this research is to discover TCP SACK Congestion Control performance on wired network using DropTail and Random Early Detection queue. The background research due sending data on wired network still experiencing congestion so the data that has been sent must be resent and that make the throughput much lower. With SACK, retransmission of the missing packets can be done because the Receiving TCP sends back SACK packets to the sender informing the sender of data that has been received, so the sender can then retransmit only the missing data segments without retransmit unnecessary packets.

This simulation using Network Simulator 2 (ns2) and congestion control of TCP Reno for performance comparison.

The result showed that TCP SACK posses more packets drop than TCP Reno because of stable congestion window of TCP SACK so that packet can be sent more and causing queue become more frequently full and make so many packets are discarded rather than TCP Reno, TCP SACK has an average throughput more than TCP Reno because of the number of received packets. However, TCP SACK performance result differently on DropTail and RED queue. TCP SACK can work smoothly on RED because the packet discared randomly, while the algorithm of TCP SACK congestion control failed in DropTail queue because of cumulative drop. From the result, in can be inferred that TCP SACK has a better performance of congestion control than TCP Reno, and TCP SACK is not able to outcome the cumulative packet drop.

Keyword : : TCP SACK, TCP Reno, Random Early Detection, DropTail, packet drop, congestion control, congestion window, throughput.