

ABSTRAK

Transmission Control Protocol (TCP) menyediakan mekanisme pengiriman data yang reliabel bagi paket IP. TCP *congestion control* adalah mekanisme dalam TCP yang menangani terjadinya kongesti. Seiring dengan makin luasnya penggunaan internet, efisiensi dari TCP *congestion control* patut mendapatkan perhatian yang besar. TCP *congestion control* terdiri dari empat fase, yaitu *Slow Start*, *Congestion Avoidance*, *Fast Retransmission*, *Fast Recovery*. Sekarang ini sudah banyak usulan perbaikan TCP *congestion control* di antaranya adalah TCP Vegas, TCP SACK, TCP FACK, TCP Reno, TCP NewReno. Semua usulan perbaikan TCP *congestion control* tersebut memodifikasi bagian akhir dari fase TCP *congestion control* (*Fast Retransmission* dan *Fast Recovery*). Algoritma *Smooth Start* memodifikasi bagian awal dari fase TCP *congestion control*.

Pada Tugas akhir ini dievaluasi performansi dari algoritma *Smooth Start* dengan mensimulasikannya di Network Simulator. Algoritma *Smooth Start* diimplementasikan di TCP Reno, menggantikan algoritma *Slow start*.

Algoritma *Slow Start* hanya berpengaruh signifikan pada koneksi pendek. Hal ini karena algoritma *Smooth Start* hanya memodifikasi fase awal dari TCP *congestion control*. Pada jaringan yang padat, algoritma *Smooth Start* mempunyai *packet loss* yang lebih rendah dan *throughput* yang lebih tinggi dari pada algoritma *Slow Start*. Pada jaringan yang tidak padat, algoritma *Smooth Start* mempunyai *packet loss* yang lebih rendah tetapi dengan *throughput* yang lebih rendah juga dibanding algoritma *Slow Start*.

Kata kunci: kongesti, *congestion control*, *Smooth Start*