

## ABSTRAK

Pada umumnya, internet menggunakan protocol TCP sebagai protokol transport untuk mengirimkan data. Pengiriman data yang besar memerlukan juga mekanisme yang bisa mengakomodir jalannya data tersebut agar tidak ada kemacetan yang timbul. Model antrian yang biasa dipakai adalah antrian droptail.

Pada penelitian ini penulis menguji unjuk kerja protocol TCP Newreno dan TCP Vegas menggunakan NS2. Metrik unjuk kerja yang digunakan adalah *throughput*, *packet drop*, dan *congestion window*. Parameter yang digunakan pada setiap pengujian adalah penambahan kapasitas *buffer*.

Hasil pengujian menunjukkan TCP Vegas dapat mengungguli TCP Newreno ketika jaringan menyebabkan *packet loss* yang besar. Hal ini dikarenakan *packet loss* sangat berpengaruh terhadap unjuk kerja TCP Newreno. Hal tersebut dapat dilihat dari nilai *throughput* yang tinggi. Tetapi TCP Vegas tidak cocok digunakan pada jaringan dengan *buffer size* yang besar dengan trafik padat karena menghasilkan nilai *throughput* yang kecil.

Kata kunci : *Droptail*, *TCP Newreno*, *TCP Vegas*, *packet drop*, *congestion window*, *throughput*.

## ABSTRACT

In general the internet uses the TCP protocol as transport protocol to transmit data. Large data transmission require a mechanism that can accommodate the way the data so as not to arise congestion. The usual queue model is droptail.

In this study the authors test the performance Protocol TCP Newreno and TCP Vegas using NS2. The metric used is the performance throughput, packet drop and congestion window. The Parameters used in each test is the addition of buffer capacity.

The result of the testing indicate TCP Vegas can outperform TCP Newreno when the networks are major causes packet loss. Because it effects the performance of TCP Newreno. It can be seen from the higher value throughput. TCP Vegas is not suitable for se on a network with a larger buffer size with heavy traffic because it produces a small throughput.

Kata kunci : *Droptail, TCP Newreno, TCP Vegas, packet drop, congestion window, throughput.*