

ABSTRAK

Multi-Protocol Label Switching (MPLS) VPN (Virtual Private Network) adalah suatu metode *forwarding* data melalui suatu jaringan dengan menggunakan informasi dalam label yang dilekatkan pada paket IP. Dengan fungsi *traffic-engineering* yang efisien, kebutuhan MPLS VPN juga meningkat dengan cepat. Namun permasalahan dapat terjadi apabila jaringan tersebut mengalami kerusakan atau terputus.

Backdoor link merupakan *link* cadangan dengan *bandwidth* rendah yang digunakan ketika *link backbone* terputus. Protokol yang digunakan adalah OSPF (*Open Shortes Path First*). Namun, jika *link backbone* masih tersambung, dan *backdoor link* OSPF menghubungkan dua situs yang berada di daerah OSPF yang sama, maka *backdoor link* OSPF selalu lebih disukai. Karena *backdoor link* OSPF akan di anggap sebagai *link intraarea*. Secara otomatis, OSPF akan memilih *link intraarea*, sehingga *backdoor link* dianggap sebagai jalan aktif. Situasi semacam ini bukan yang kita harapkan karena *backdoor link* ini memiliki *bandwidth* yang rendah dan bukanlah jalur utama. Untuk menghindari hal tersebut, digunakan *Sham Link* (link palsu). *Sham link* merupakan *link* palsu yang di konfigurasi di antara *router* PE sehingga *link backbone* dianggap sebagai *link intraarea*.

Pada tugas akhir ini telah disimulasikan *backdoor link* pada jaringan MPLS VPN dengan menggunakan OSPF *sham link*. Simulasi menggunakan GNS3 sebagai *router* MPLS VPN. Analisis performansi *backdoor link* berdasarkan nilai QoS untuk transfer datanya.

Hasil yang diperoleh adalah *backdoor link* dapat digunakan sebagai *link* cadangan, dan mampu mempertahankan QoS khususnya *throughput*, dengan tetap mengirimkan paket data apabila *backbone* tidak aktif. Apabila terdapat dua *backdoor link*, maka *backdoor link* dengan *bandwidth* yang lebih tinggi akan digunakan untuk jalur pengiriman paket data, hal ini dikarenakan jalur yang memiliki *bandwidth* lebih besar, memiliki nilai *cost* yang lebih kecil.

Kata Kunci : *Backdoor Link, OSPF, Sham link, MPLS VPN, GNS3*