UNIVERSITAS INDONESIA

SIMULASI UNTUK MEMBANDINGKAN KINERJA PPTP DAN L2TP UNTUK BERBAGAI KELAS TRAFIK

SKRIPSI

ASRUL BUDIADJI
07 06 19 9142

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER, 2009

UNIVERSITAS INDONESIA

SIMULASI UNTUK MEMBANDINGKAN KINERJA PPTP DAN L2TP UNTUK BERBAGAI KELAS TRAFIK

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

ASRUL BUDIADJI
07 06 19 9142

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER, 2009

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri,
Dan semua sumber baik yang dikutip maupun dirujuk
Telah saya nyatakan dengan benar.

Nama	**: Asrul Budiadji**
NPM	**: 0706199142**
Tanda Tangan	**:**
Tanggal	**: 15 Desember 2009**

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama	: Asrul Budiadji
NPM	: 0706199142
Program Studi	: Strata 1 Ekstensi
Judul Tugas Akhir	: Simulasi Untuk Membandingkan Kinerja PPTP Dan L2TP Untuk Berbagai Kelas Trafik

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian pernyataan untuk memperoleh gelar Sarjana Teknik pada program studi strata 1 ekstensi, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing	: Prima Dewi Purnamasari ST, M.Sc	(.....)
Penguji	: Prof. Dr. Ir. Bagio Budiardjo M.Sc	(.....)
Penguji	: Muhammad Salman ST., MIT	(.....)

Ditetapkan di : Universitas Indonesia, Depok

Tanggal : 29 Desember 2009

KATA PENGANTAR / UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia.. Oleh karena itu, saya mengucapkan terima kasih kepada:

(1) Prima Dewi Purnamasari ST, M.Sc selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan ide untuk mengarahkan saya dalam penyusunan Tugas Akhir ini, Prof. Dr. Ir. Bagio Budiardjo M.Sc dan Muhammad Salman ST., MIT yang telah menyediakan waktu, tenaga dan pikiran untuk menguji dan mengarahkan Tugas Akhir ini.

(2) Fadry teman dan rekan kerja yang juga merasakan pahit manisnya tugas akhir ini, Mas Irdan dan Mas Yono (P.T. Adiyasa Wicaksana), Burhan dan Ardi (Ast Lab Jarkom), Mba Eka (Lab Cisco).

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 30 Desember 2009

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

==========

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini :

Nama	: Asrul Budiadji
NPM	: 0706199142
Program Studi	: S1 – Ekstensi
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis karya	: Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

Simulasi Untuk Membandingkan Kinerja PPTP dan L2TP Untuk Berbagai Kelas Trafik

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media / formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 30 Desember 2009
Yang menyatakan

(Asrul Budiadji)

ABSTRAK

Nama : Asrul Budiadji
Program Studi : S1 - Ekstensi
Judul : Simulasi Untuk Membandingkan Kinerja PPTP dan L2TP Untuk Berbagai Kelas Trafik

Pada penelitian ini akan dilakukan simulasi untuk membandingkan kinerja protokol *tunneling* PPTP (*Point to Point Tunneling Protocol*) dan L2TP (*Layer 2 Tunneling Protocol*) dengan melakukan transfer data dari *network* satu ke *network* yang lain yang dilewatkan melalui *tunnel*. Trafik yang dilewatkan dalam simulasi ini adalah UDP (*User Datagram Protocol*), TCP (*Transmission Control Protocol*) dan RDP (*Remote Desktop Protocol*). Perbedaan karaketeristik dari trafik-trafik yang dilewatkan dapat membedakan konsumsi *bandwidth* yang mempengaruhi *transfer rate* dari masing-masing trafik, oleh karena itu trafik-trafik tersebut dibedakan tingkat prioritasnya dan dibagi menjadi kelas-kelas tertentu. Sebelum dilewatkan dalam *tunnel*, data akan mengalami proses enkapsulasi yang mengakibatkan bertambahnya paket *header* yang akan mengurangi *byte payload* dari data yang akan dikirimkan sehingga proses pengiriman data melalui *tunnel* akan memakan waktu lebih lama. Hasil simulasi menunjukkan bahwa nilai *throughput* pada PPTP lebih besar daripada L2TP, *throughput* PPTP sebesar 90.68 % dan L2TP sebesar 83.01 % dari *bandwidth* yang tersedia.

Kata kunci:
PPTP, L2TP, TCP, UDP, Enkapsulasi, *Throughput*

ABSTRACT

Name : Asrul Budiadji
Study Program: S1 - Ekstensi
Title : Simulation to Compare PPTP and L2TP Performance For Different Class of Service.

In this study, simulation would be conducted to compare PPTP (Point to Point Tunneling Protocol) and L2TP (layer 2 Tunneling Protocol) performance by carrying data transfer from one network to another passed through the tunnel. The passed traffics in the simulation are UDP (User Datagram Protocol), TCP (Transmission Control Protocol) and RDP (Remote Desktop Protocol). The characters difference from the passed traffic could differ bandwidth consumption that affects transfer rate from each traffics, therefore those traffics are differentiated its priority level and divided into certain classes. Before the data being passed through the tunnel, the data would experience encapsulation process which results in increased packet header that of course would diminish payload bytes from the sent data so that the process of transfering data through the tunnel would take longer time. The simulations show that PPTP's throughput greater than L2TP's, PPTP gets 90.68 % whereas L2TP gets 83.01 % of available bandwidth.

Keywords:
L2TP, PPTP, TCP, UDP, Encapsulation, Throughput

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1. Latar Belakang

Tunneling merupakan proses mengirim paket ke komputer pada jaringan *private* dengan menggunakan jaringan lain seperti internet. *Network routers* yang berada diluar jaringan *private* tidak dapat mengakses komputer yang berada di jaringan *private*. Bagaimanapun, *tunneling* memungkinkan *routing network* untuk mengalirkan data ke komputer perantara, seperti *server* yang dikoneksikan ke *routing network* dan jaringan *pivate*. Keduanya, klien dan *server*, menggunakan *tunneling* untuk mengamankan paket ke komputer pada jaringan *private* dengan menggunakan *routers* yang hanya mengetahui alamat dari jaringan *private server* perantara.

Untuk membangun sebuah *tunnel*, diperlukan sebuah protokol pengaturnya sehingga *tunnel* secara logika ini dapat berjalan dengan baik bagaikan koneksi *point-to-point* sungguhan. Saat ini, tersedia banyak sekali protokol pembuat *tunnel* yang bisa digunakan seperti GRE, PPTP dan L2TP

Generic Routing Encapsulation (GRE) Protokol *tunneling* yang memiliki kemampuan membawa lebih dari satu jenis protokol pengalamatan komunikasi. Bukan hanya paket beralamat IP saja yang dapat dibawanya, melainkan banyak paket protokol lain seperti CNLP, IPX, dan banyak lagi.[3]. Namun, semua itu dibungkus atau dienkapsulasi menjadi sebuah paket yang bersistem pengalamatan IP. Kemudian paket tersebut didistribusikan melalui sistem *tunnel* yang juga bekerja di atas protokol komunikasi IP.

Point-to-Point Tunneling Protocol (PPTP) adalah sebuah protokol yang mengizinkan hubungan *Point-to Point Protocol* (PPP) melewati jaringan IP, dengan membuat *Virtual Private Network* (VPN). *Microsoft* mengimplementasikan protokol dan algoritmanya untuk mendukung PPTP. Implementasi dari PPTP ini disebut *Microsoft* PPTP, yang digunakan untuk mendukung perluasan dalam mengkomersilkan produk tunneling atau VPN khususnya yang telah terdapat pada *Microsoft Windows* 95, 98 dan NT.

Layer Two Tunneling Protocol (L2TP) adalah sebuah *tunneling protocol* yang memadukan dan mengkombinasikan dua buah *tunneling* protokol yang bersifat proprietary, yaitu L2F (*Layer 2 Forwarding*) milik *Cisco Systems* dengan PPTP (*Point-to-Point Tunneling Protocol*) milik *Microsoft.*[1]

1.2 Tujuan Penulisan

Tujuan penulisan tugas akhir ini adalah untuk mengetahui dan memahami konsep *tunneling*, proses enkapsulasi dan besar throughput tunnel PPTP dan L2TP

1.3 Batasan Masalah

Masalah yang dibahas dalam tugas akhir ini adalah analisis kinerja PPTP dan L2TP dalam hal proses enkapsulasi dan besar throughput. Yang akan dilakukan dengan membuat simulasi (*Event Simulation*) dengan melewatkan berbagai trafik (UDP, TCP dan RDP) dalam jaringan *tunneling* PPTP dan L2TP.

1.4 Sistematika Penelitian

Sistematika penelitian pada tugas akhir ini adalah :

Bab 1 Pendahuluan

Bagian pendahuluan terdiri atas latar belakang, tujuan penulisan, batasan masalah, dan sistematika penelitian.

Bab 2 Dasar Teori

Bagian ini akan membahas teori dasar yang digunakan pada penelitian yaitu mengenai trafik-trafik yang akan dilewatkan dalam *tunnel* (TCP, UDP dan RDP), pembagian kelas trafik (*Class of Service*), teknologi *tunneling* dan protokol-protokol *tunneling* (PPTP dan L2TP)

Bab 3 Perancangan Network

Bagian awal dari bab ini membahas mengenai perlengkapan yang dibutuhkan untuk membangun simulasi (*Hardware* dan *Software*),

model perancangan simulasi yang akan dilakukan, dan metode pengambilan data yang akan di analisa (*sniffing*).

Bab 4 Analisa Data

Bagian ini berisi tentang data-data hasil simulasi dan analisisnya. Hasil analisis merupakan dasar pembentukan kesimpulan pada penelitian ini.

Bab 5 Kesimpulan

Bab ini berisi kesimpulan yang diperoleh dari keseluruhan kegiatan penelitian yang telah dilakukan.

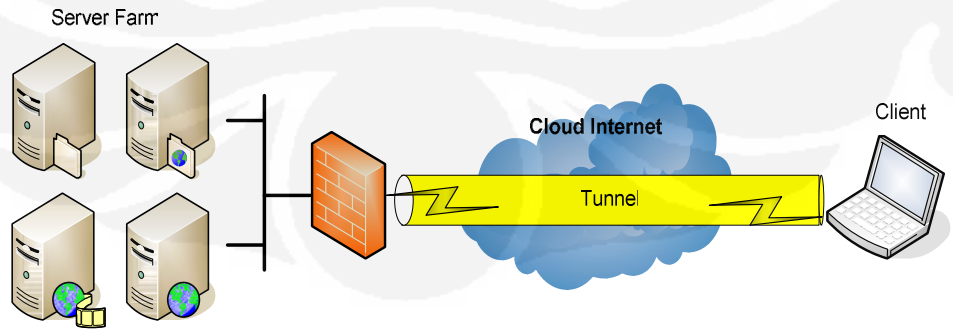
BAB II
TUNNELING, PPTP & L2TP, PROTOKOL TCP & UDP SERTA CLASS OF SERVICE

2.1 Teknologi *Tunneling*

Teknologi *tunneling* merupakan teknologi yang bertugas untuk manangani dan menyediakan koneksi *point-to-point* dari sumber ke tujuannya. Disebut *tunnel* karena koneksi *point-to-point* tersebut sebenarnya terbentuk dengan melintasi jaringan umum, namun koneksi tersebut tidak mempedulikan paket-paket data milik orang lain yang sama-sama melintasi jaringan umum tersebut, tetapi koneksi tersebut hanya melayani transportasi data dari pembuatnya. Koneksi *point-to-point* ini sesungguhnya tidak benar-benar ada, namun data yang dihantarkannya terlihat seperti benar-benar melewati koneksi pribadi yang bersifat *point-to-point*.[3]

Teknologi ini dapat dibuat di atas jaringan dengan pengaturan *IP Addressing* dan *IP Routing* yang sudah matang. Maksudnya, antara sumber *tunnel* dengan tujuan *tunnel* telah dapat saling berkomunikasi melalui jaringan dengan pengalamatan IP. Apabila komunikasi antara sumber dan tujuan dari *tunnel* tidak dapat berjalan dengan baik, maka *tunnel* tersebut tidak akan terbentuk.

Untuk membuat sebuah *tunnel*, diperlukan sebuah protokol pengaturnya sehingga *tunnel* secara logika ini dapat berjalan dengan baik bagaikan koneksi *point-to-point* sungguhan. Saat ini, tersedia banyak sekali protokol pembuat tunnel yang bisa digunakan seperti GRE, PPTP dan L2TP



Gambar 2.1 Teknologi *Tunneling*

2.2 *Point to Point Tunneling Protocol* (PPTP)

PPTP merupakan protokol jaringan yang memungkinkan pengamanan transfer data dari *remote client* ke *server* pribadi perusahaan dengan membuat sebuah VPN melalui TCP/IP.

Teknologi jaringan PPTP merupakan pengembangan dari *remote access Point-to-Point protocol* yang dikeluarkan oleh *Internet Engineering Task Force* (IETF). PPTP merupakan protokol jaringan yang merubah paket PPP menjadi IP *datagrams* agar dapat ditransmisikan melalui intenet. PPTP juga dapat digunakan pada jaringan *private* LAN-to-LAN.[3]

PPTP terdapat sejak dalam sistem operasi Windows NT *server* dan Windows NT *Workstation* versi 4.0. Komputer yang berjalan dengan sistem operasi tersebut dapat menggunakan protokol PPTP dengan aman untuk terhubung dengan *private network* sebagai klien dengan *remote access* melalui internet. PPTP juga dapat digunakan oleh komputer yang terhubung dengan LAN untuk membuat VPN melalui LAN.

Fasilitas utama dari penggunaan PPTP adalah dapat digunakannya *public-switched telephone network* (PSTNs) untuk membangun VPN. Pembangunan PPTP yang mudah dan berbiaya murah untuk digunakan secara luas, menjadi solusi untuk *remote users* dan *mobile users* karena PPTP memberikan keamanan dan enkripsi komunikasi melalui PSTN ataupun internet.

PPTP membuat enkapsulasi frame pada IP dalam sebuah *Generic Routing Encapsulation* (GRE), kemudian GRE dibungkus dalam sebuah paket IP untuk membuat *tunne*l.

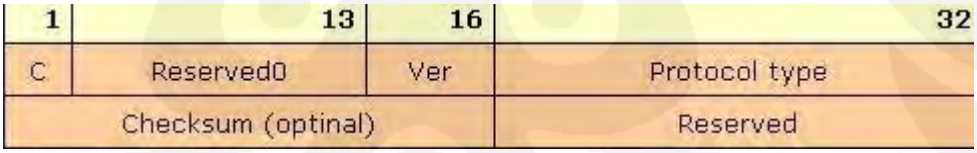
2.2.1 *Generic Routing Encapsulation* (GRE)

Generic Routing Encapsulation adalah *tunneling protocol* yang di dikembangkan oleh *cisco*, protokol ini dapat melakukan enkapsulasi berbagai macam jenis paket dalam lapisan *network protocol* dalam *tunnel*nya, dengan cara membuat virtual komunikasi *point to point* dari *router* asal ke *router* tujuan dengan menggunakan IP pada komunikasi internetwork.[2]

Generic Routing Encapsulation (GRE) Protokol *tunneling* yang memiliki kemampuan membawa lebih dari satu jenis protokol pengalamatan komunikasi.

Bukan hanya paket beralamat IP saja yang dapat dibawanya, melainkan banyak paket protokol lain seperti CNLP, IPX, dan banyak lagi.[3] Namun, semua itu dibungkus atau dienkapsulasi menjadi sebuah paket yang bersistem pengalamatan IP. Kemudian paket tersebut didistribusikan melalui sistem *tunnel* yang juga bekerja di atas protokol komunikasi IP. Dengan menggunakan *tunneling* GRE, *router* yang ada pada ujung-ujung *tunnel* melakukan enkapsulasi paket-paket protokol lain di dalam *header* dari protokol IP. Hal ini akan membuat paket-paket tadi dapat dibawa ke manapun dengan cara dan metode yang terdapat pada teknologi IP. Dengan adanya kemampuan ini, maka protokol-protokol yang dibawa oleh paket IP tersebut dapat lebih bebas bergerak ke manapun lokasi yang dituju, asalkan terjangkau secara pengalamatan IP.

Aplikasi yang cukup banyak menggunakan bantuan protokol *tunneling* ini adalah menggabungkan jaringan-jaringan lokal yang terpisah secara jarak kembali dapat berkomunikasi. Atau dengan kata lain, GRE banyak digunakan untuk memperpanjang dan mengekspansi jaringan lokal yang dimiliki si penggunanya. Meski cukup banyak digunakan, GRE juga tidak menyediakan sistem enkripsi data yang lalu-lalang di *tunnel*-nya, sehingga semua aktivitas datanya dapat dimonitor menggunakan *protocol analyzer* biasa.



Gambar 2.2 Format Header GRE [4]

Keterangan:

- C : Checksum Present
- Reserved 0&1 : Disediakan untuk digunakan kemudian
- Ver : Version number; harus = 0.
- Protocol Type : Berisi *protocol type* dari *payload packet*.
- Checksum : Berisi IP checksum dari *header* GRE dan *payload packet*.

2.2.2 Arsitektur PPTP

Komunikasi yang aman dibuat dengan menggunakan protokol PPTP melewati tiga proses, dimana setiap proses tersebut membutuhkan selesainya proses yang sebelumnya. Ketiga proses tersebut berjalan dengan cara sebagai berikut.

- ***PPTP Connection and Communication***. Klien PPTP menggunakan PPP untuk terhubung ke ISP. Koneksi tersebut menggunakan protokol PPP untuk membangun koneksi dan enkripsi paket data.
- ***PPTP Control Connection***. Menggunakan koneksi ke internet yang telah dibangun oleh protokol PPP, protokol PPTP membuat sebuah *control connection* dari klien PPTP ke *server* PPTP di internet. Koneksi tersebut menggunakan TCP untuk membagun koneksi dan ini disebut dengan PPTP *tunnel*.
- ***PPTP Data Tunneling***. Akhirnya protokol PPTP membuat IP *datagrams* yang di dalamnya terdapat enkripsi paket PPP yang kemudian dikirim melalui PPTP *tunnel* ke *server* PPTP. *Server* PPTP membongkar IP datagram dan mendekripsi paket PPP dan kemudian merutekan paket yang telah didekripsi ke jaringan *private*.

PPTP *Control Connection*

Protokol PPTP menspesifikasikan seri pengiriman dari *control message* antara PPTP-*enabled client* dan *server* PPTP. *Control message* membangun, memelihara dan mengakhiri PPTP *tunnel*. Berikut ini merupakan daftar yang dibuat oleh *control message* dasar yang digunakan untuk membuat dan memelihara PPTP *tunnel* [5] :

- PPTP_START_SESSION_REQUEST: Permintaan untuk memulai *Session*
- PPTP_START_SESSION_REPLY : Untuk menjawab *start session*
- PPTP_ECHO_REQUEST : *Maintain session*
- PPTP_ECHO_REPLY : Untuk menjawab *Maintain session*
- PPTP_WAN_ERROR_NOTIFY : Laporan *error* pada koneksi PPP

- PPTP_SET_LINK_INFO : Merubah setting koneksi antara klien dan *server* PPTP
- PPTP_STOP_SESSION_REQUEST : Mengakhiri *session*
- PPTP_STOP_SESSION_REPLY : Untuk menjawab *stop session*

Control message ditransmisikan pada paket kontrol pada TCP *datagram*. Satu koneksi TCP dibangun antara klien PPTP dan *server* PPTP. Koneksi tersebut digunakan untuk menukar *control message*. *Control messages* dikirim dengan TCP *datagram*. Penukaran *message* antara klien PPTP dan server PPTP melalui koneksi TCP digunakan untuk membuat dan memelihara PPTP *tunnel*.

2.2.3 Format *Header* PPTP

16	32 bit
Length	PPTP message type
Magic cookie	
Control message type	Reserved 0
Protocol Version	Reserved 1
Framing capability	
Bearing capability	
Maximum channels	Firmware revision
Host name (64 Octets)	
Vendor string (64 Octets)	

Gambar 2.3 Format *Header* PPTP [5]

Keterangan:

- Length : panjang total paket PPTP dalam octet termasuk *header* PPTPnya.
- PPTP message type : tipe *message*; 1 *control message*, 2 *management message*
- Magic cookie : *magic cookie* selalu terkirim 0x1A2B3C4D. untuk mengizinkan *receiver* menjamin sinkronisasi dengan TCP data *stream*.
- Control MessageType :

  - Control Connection Management – 1 Start-Control-Connection-Request; 2 Start-Control-Connection-Reply; 3 Stop-Control-Connection-Request; 4 Stop-Control-Connection-Reply; 5 Echo-Request; 6 Echo-Reply.
  - Call Management – 7 Outgoing-Call-Request; 8 Outgoing-Call-Reply; 9 Incoming-Call-Request; 10 Incoming-Call-Reply; 11 Incoming-Call-Connected; 12 Call-Clear-Request; 13 Call-Disconnect-Notify
  - Error Reporting – 14 WAN-Error-Notify
  - PPP Session Control – 15 Set-Link-Info
- Reserved 0 & 1 : Harus = 0
- Protocol version : PPTP *version number*
- Framing Capabilities : Mengindikasikan tipe *framing* yang dapat dilakukan oleh pengirim: 1 - *Asynchronous Framing supported*; 2 - *Synchronous Framing supported*
- Bearer Capabilities : Mengindikasikan kemampuan *bearer* yang dapat dilakukan oleh pengirim: 1 - *Analog access supported*; 2 - *Digital access supported*
- Maximum Channels : jumlah total *session* PPP yang dapat didukung PAC.
- Firmware Revision : berisi jumlah *firmware revision* dari PAC jika dikeluarkan oleh PAC atau versi dari PNS PPTP jika dikeluarkan oleh PNS.
- Host Name : berisi nama DNS dari PAC atau PNS
- Vendor Name : berisi *string* vendor tertentu menjelaskan tipe PAC yang digunakan, atau tipe *software* PNS yang digunakan jika *request* dikeluarkan oleh PNS.

2.3 *Layer* 2 *Tunneling Protocol* (L2TP)

L2TP adalah sebuah *tunneling protocol* yang memadukan dan mengombinasikan dua buah *tunneling protocol* yaitu L2F (*Layer 2 Forwarding*)

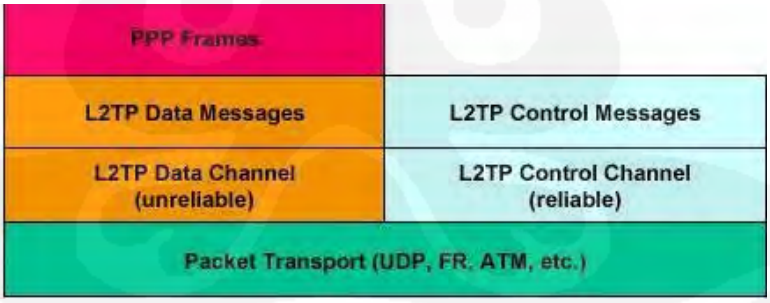
milik *Cisco Systems* dengan PPTP (*Point-to-Point Tunneling Protocol*) milik *Microsoft*.[1]

Pada awalnya, semua produk *Cisco* menggunakan L2F untuk mengurus *tunneling*-nya, sedangkan *operating system Microsoft* yang terdahulu hanya menggunakan PPTP untuk melayani penggunanya yang ingin bermain dengan *tunnel.* Namun saat ini, *Microsoft Windows* NT/2000 telah dapat menggunakan PPTP atau L2TP dalam teknologi VPN-nya.

L2TP biasanya digunakan dalam membuat *Virtual Private Dial Network* (VPDN) yang dapat bekerja membawa semua jenis protokol komunikasi didalamnya. Selain itu, L2TP juga bersifat media independen karena dapat bekerja di atas media apapun. L2TP memungkinkan penggunanya untuk tetap dapat terkoneksi dengan jaringan lokal milik mereka dengan *policy* keamanan yang sama dan dari manapun mereka berada, melalui koneksi VPN atau VPDN.

Protokol L2TP sering juga disebut sebagai protokol dial-up virtual, karena L2TP memperluas suatu *session* PPP (*Point-to-Point Protocol*) dial-up melalui jaringan publik internet, atau sering juga digambarkan seperti koneksi virtual PPP.

2.3.1 Arsitektur L2TP



Gambar 2.4 Struktur Protokol L2TP [5]

Frame PPP dienkapsulasi oleh *header* L2TP dan paket *transport* UDP, kemudian dilewatkan melalui *data channel* yang *unreliable*. *Control messages* dikirimkan melalui suatu *control channel* L2TP yang juga mentransmisikan paket *in-band* melalui paket transport yang sama. *Sequence number* diperlukan pada semua *control message* dan digunakan untuk menyediakan pengiriman yang

handal dalam *control channel*. Data *message* juga harus menggunakan *sequence number* untuk menyusun kembali dan mendeteksi paket yang hilang.

Ada 2 jenis *messages* yang digunakan L2TP: *control messages* dan *data messages*.[6]

1. *Control Messages*
   - Digunakan untuk: -*Establishment* (Pembentukan)
     -*Maintenance* (Pemeliharaan)
     -Pemutusan *tunnel* L2TP dan interkoneksi
   - Menggunakan suatu *contol channel* yang *reliable* didalam L2TP untuk menjamin kepastian paket yang terkirim
2. Data *Messages*
   - Digunakan untuk mengenkapsulasi frame PPP yang akan dibawa melalui *tunnel*
   - Jika *loss packet* terjadi, data *messages* tidak akan dikirim kembali (*not reliable*).

Tipe *Control Message*

Didalam protokol *tunnel*, *control message* dipertukarkan secara *inband* antara *client* dan *server*. Kontrol koneksi bertanggung jawab untuk pembentukan, pemutusan, dan *maintenance session*, yang dibawa didalam *tunnel* dan *tunnel* itu sendiri.

Tipe *control message* adalah sebagai berikut:

- *Control Connection Management*:
  - 0 = (reserved)
  - 1 = (SCCRQ) Start-Control-Connection-Request
  - 2 = (SCCRP) Start-Control-Connection-Reply
  - 3 = (SCCCN) Start-Control-Connection-Connected
  - 4 = (StopCCN) Start-Control-Connection-Notification
  - 5 = (reserved)
  - 6 = (HELLO) Hello
- *Call Management*
  - 7 = (OCRQ) Outgoing-Call-Request

  - 8 = (OCRO) Outgoing-Call-Reply
  - 9 = (OCCN) Outgoing-Call-Connected
  - 10 = (ICRQ) Incoming-Call-Request
  - 11 = (ICRP) Incoming-Call-Reply
  - 12 = (ICCN) Incoming-Call-Connected
  - 13 = (reserved)
  - 14 = (CDN) Call-Disconnect-Notify
- *Error Reporting*
  - 15 = (WEN) WAN-Error-Notify
- PPP *Session Control*
  - 16 = (SLI) Set-Link-Info

Definisi *control messages* diatas adalah sebagai berikut [6]:

- SCCRQ – *control messages* yang digunakan untuk menginisialisasi *tunnel* antara *server* dan *client*. Dikirim oleh client dan *server* untuk proses pembentukan *tunnel.*
- SCCRP – *control messages* yang digunakan untuk mengindikasikan bahwa SCCRQ telah diterima dan pembentukan *tunnel* harus dilanjutkan. Dikirim sebagai jawaban dari SCCRP.
- StopCCN – *control messages* yang dikirim oleh *client* dan *server* untuk menginformasikan *peer* bahwa *tunnel* sedang diputus dan hubungan kontrol harus diputus. Lebih lanjut lagi seluruh koneksi akan terputus (tanpa mengirim *explicit call control message*).
- OCRQ – *control message* yang dikirim oleh server ke *client* untuk mengindikasikan bahwa *outbound call* dari *client* terbentuk. Merupakan *message* pertama dalam pertukaran message yang digunakan untuk membentuk *session* dalam *tunnel* L2TP.
- OCRP – *control message* yang dikirim oleh *client* kepada *server* sebagai respon OCRQ yang dikirim. Merupakan *message* kedua yang bertukar pada pembentukan *session* dalam *tunnel* L2TP

- OCCN – *control message* yang dikirimkan *client* ke *server* mengikuti OCRP setelah *outgoing call* terbentuk. Merupakan message terakhir yang bertukar untuk pembentukan *session* dalam *tunnel* L2TP. OCCN digunakan juga untuk mengindikasikan hasil dari permintaan *outgoing call* yang berhasil dan memberikan informasi pada *server* mengenai parameter yang diperoleh setelah panggilan terbentuk seperti tipe *message*, (TX*) connection speed*, dan tipe *framing*.

2.3.2 Format *Header* L2TP

Gambar 2.5 Format *Header* L2TP [6]

Keterangan:

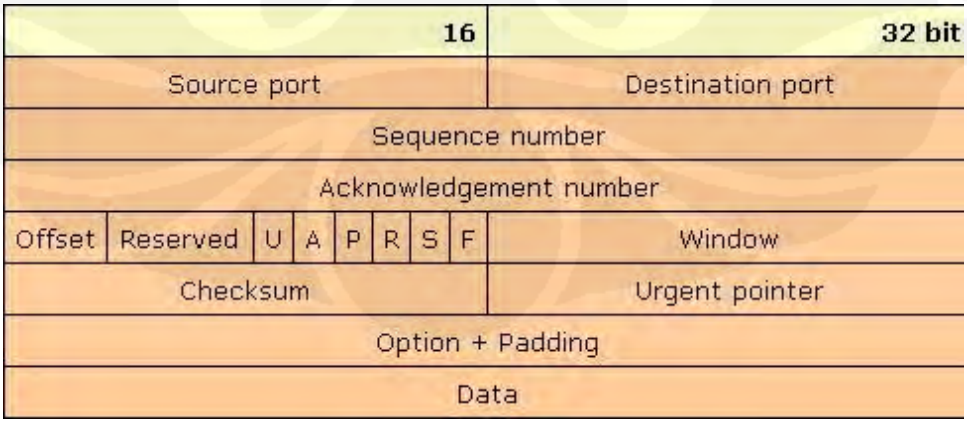
- Type (T) bit : - Tipe dari *message*
  - 0 = *data message*, 1 = *control message*
- Length (L) bit : - L = 1, berarti *field length* terisi
  - Untuk *control message* harus di-set = 1
- X bit : - Disediakan untuk digunakan kemudian
  - Semua bit yang dipesan HARUS di-set 0 pada *outgoing messages* dan pada *incoming messages* diabaikan.
- Sequence (S) bit : - S =1, berarti *field* Ns dan Nr terisi
  - Untuk *control message* harus di set = 1
- Offset (O) bit : - O = 1, *field Size Offset* terisi
  - Untuk *control messages* di-set = 0 (nol)
- Priority (P) bit : - P = 1, mendapatkan perlakuan yang istimewa khususnya dalam *data message*

- Untuk semua *control messages* di-set = 0

- Ver : - Ver = 2, versi header *data message* L2TP
  - Jika *unknown Ver*, paket tersebut harus dibuang.
- Length field : - Panjang total dari *message* (byte).
- Tunnel ID : - *Identifier* untuk *control connection*
  - *Significant* lokal saja
- Session ID : - *Identifier* untuk suatu *session* di dalam suatu *tunnel*.
  - *Significant* lokal saja
- Ns Sequence Number : - *Sequence number* untuk *control message*
- Nr Sequence Number : - *Sequence number control message* berikutnya yang diterima.
- Offset Field : - *Start* dari *payload* data.

2.4 *Transmission Control Protocol* (TCP)

TCP adalah protokol yang memungkinkan program-program aplikasi untuk mengakses/menggunakan layanan komunikasi bersifat *connection-oriented*. TCP mampu memberikan jasa pengiriman yang dapat diandalkan (*reliable*) sekaligus bersifat *flow-controlled*. Sifat *flow-controlled* ini memungkinkan peralatan-peralatan jaringan yang berkecepatan rendah (*slower-speed network devices*) dapat berhubungan dengan peralatan-peralatan jaringan yang berkecepatan tinggi (*higher-speed network devices*).[8]

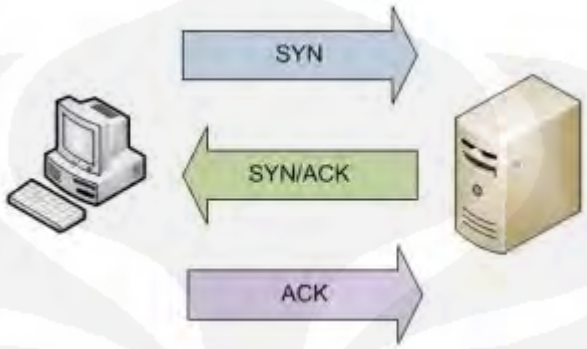


Gambar 2.6 Format *Header* TCP [8]

Keterangan:

- Source port -- Mengindikasikan sumber protokol lapisan aplikasi yang mengirimkan segmen TCP.
- Destination port -- Mengindikasikan tujuan protokol lapisan aplikasi yang menerima segmen TCP
- Sequence number -- Mengindikasikan nomor urut dari oktet pertama dari data di dalam sebuah segmen TCP yang hendak dikirimkan. Dalam fase *connection establishment field* ini juga dapat digunakan untuk mengidentifikasi sebuah initial *sequence number* yang akan digunakan pada transmisi berikutnya.
- Acknowledgment number – Mengindikasikan nomor urut dari oktet selanjutnya dalam aliran byte yang diharapkan diterima oleh pengirim dari si penerima pada pengiriman selanjutnya.
- Data offset – 4 bits.Mengindikasikan di mana data dalam segmen TCP dimulai. *Field* ini juga dapat berarti ukuran dari *header* TCP, *field* ini merupakan angka dari word 32-bit dalam *header* TCP.
- Reserved -- 6 bits. Direservasikan untuk digunakan pada masa depan. Pengirim akan mengeset bit-bit ini ke dalam nilai 0.
- Control bits (Flags) -- 6 bits. Mengindikasikan *flag-flag* TCP yang terdiri atas: URG (*Urgent*), ACK (*Acknowledgment*), PSH (*Push*), RST (*Reset*), SYN (*Synchronize*), dan FIN (*Finish).*
- Window -- 16 bits. Mengindikasikan jumlah byte yang tersedia yang dimiliki oleh *buffer host* penerima segmen yang bersangkutan. *Buffer* ini disebut sebagai *Receive Buffer*, digunakan untuk menyimpan byte *stream* yang datang. Dengan mengimbuhkan ukuran window ke setiap segmen, penerima segmen TCP memberitahukan kepada pengirim segmen berapa banyak data yang dapat dikirimkan dan disangga dengan sukses.
- Checksum -- 16 bits. melakukan pengecekan integritas segmen TCP (*header*-nya dan *payload*-nya).
- *Urgent Pointer* -- 16 bits. Menandakan lokasi data yang dianggap "*urgent*" dalam segmen

- Option + Paddling - Berfungsi sebagai penampung beberapa opsi tambahan TCP
- Data – berisi informasi dari layer diatasnya.



Gambar 2.7 TCP 3-*Way Handshake*

Untuk membuat koneksi, TCP menggunakan *three-way handshake* [7] :

- Klien (yang ingin membuat koneksi) akan mengirimkan sebuah segmen TCP dengan *flag* SYN diaktifkan kepada *server* (yang hendak diajak untuk berkomunikasi).
- *Server* akan meresponsnya dengan mengirimkan segmen dengan *acknowledgment* dan juga SYN kepada klien.
- Klien selanjutnya akan mulai saling bertukar data dengan *server*.

Karakteristik Protokol TCP

1. *Connection-oriented*

Suatu arsitektur/mekanisme komunikasi data di mana dua perangkat yang akan saling berkomunikasi diharuskan untuk membuat sebuah sesi (*session*) terlebih dahulu. Ketika komunikasi telah selesai, *session* tersebut akan berakhir. Hal inilah yang terjadi dalam komunikasi menggunakan telepon, sebuah koneksi harus tersedia dan terjadi terlebih dahulu sebelum telepon yang dituju dapat digunakan untuk berkomunikasi dengan telepon yang digunakan untuk memanggil.

2. *Reliable* (keandalan)

Keandalan yang dimiliki oleh protokol ini disebabkan karena beberapa mekanisme. Berikut mekanisme tersebut:

- Checksum: semua segmen TCP membawa *checksum* yang akan digunakan oleh si penerima (*receiver device*) untuk mengecek adanya *error* baik itu *error* pada data atau pada *header* milik TCP itu sendiri.
- Duplicate Data Detection: kemampuan TCP untuk menjaga setiap byte yang diterima agar byte-byte tersebut tidak mengalami penggandaan (baca: diterima lebih dari satu kali).
- Retransmisson: kemampuan TCP untuk mengimplementasikan skema pengiriman ulang untuk data kiriman yang rusak atau hilang.
- Sequencing: kemampuan TCP untuk menyusun segmen-segmen data yang telah diterimanya. Hal ini akan membuat TCP mampu mengirimkan kembali data tersebut kepada suatu aplikasi dengan susunan yang benar.
- Timers: TCP menggunakan dua *timer* sekaligus dalam pengiriman data. Dua *timer* tersebut yakni *timer* statik dan *timer* dinamis. Protokol yang menjadi pengirim akan menunggu si penerima dalam periode waktu tertentu untuk sebuah "*acknowledgement*". Jika timer telah habis masa periodenya, si pengirim dapat mengirim kembali (*retransmit*) segment yang akan dikirim.

3. *Stream* data transfer

TCP akan mengelompokkan byte-byte yang sebelumnya tidak terstruktur ke dalam bentuk segmen untuk kemudian dikirimkan ke IP. Layanan ini memberikan keuntungan bagi aplikasi-aplikasi karena mereka tidak perlu lagi membuat blok-blok data.

4. Efficient flow control

Ketika mengirim ulang *acknowledgement* ke alamat asal, proses TCP yang menerima mengindikasikan nomor urutan yang bisa diterimanya tanpa harus meng-*over flow buffer* internal miliknya.

5. Full-duplex operation

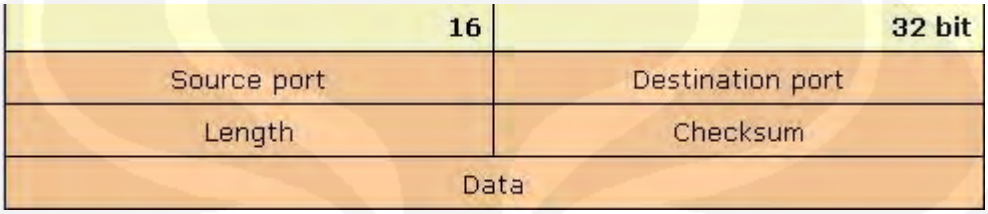
TCP bisa mengirim dan menerima dalam waktu yang bersamaan

6. Multiplexing

Komunikasi antar *upper-layer* yang terjadi secara simultan bisa dimultiplexikan melalui satu koneksi tunggal

2.5 *User Datagram Protocol* (UDP)

Protokol pada lapisan *transport* (layer 4 OSI layer) yang bersifat *connectionless* (tidak ada *handshaking* antara pengirim dan penerima), sederhana (antara penerima dan pengirim tidak perlu menjaga session), *unreliable*/tidak andal (tidak ada nomor urut dan pesan *acknowledgement*) karenanya lapisan protokol aplikasinya harus menyediakan penanganan kesalahan tersendiri. [8]



Gambar 2.8 Format *Header* UDP [8]

Keterangan:

- Source port - Digunakan untuk mengidentifikasikan sumber protokol lapisan aplikasi yang mengirimkan pesan UDP yang bersangkutan. Penggunaan *field* ini adalah opsional, dan jika tidak digunakan, akan diset ke angka 0.
- Destination port - Digunakan untuk mengidentifikasikan tujuan protokol lapisan aplikasi yang menjadi tujuan pesan UDP yang bersangkutan.
- Length - Digunakan untuk mengindikasikan panjang pesan UDP (pesan UDP ditambah dengan *header* UDP) dalam satuan byte.
- Checksum -- Berisi informasi pengecekan integritas dari pesan UDP yang dikirimkan (*header* UDP dan pesan UDP). Penggunaan field ini adalah opsional. Jika tidak digunakan, field ini akan bernilai 0.
- Data – berisi informasi dari layer diatasnya.

2.6 *Remote Desktop Protocol* (RDP)

Remote Desktop Protocol (sering disingkat menjadi RDP) adalah sebuah protokol jaringan yang digunakan oleh Microsoft Windows Terminal Services dan *Remote Desktop*. RDP dibuat berdasarkan protokol T.120 yang spesifikasinya

diumumkan oleh International Telecommunication Union (ITU), yang juga merupakan protokol yang digunakan di dalam perangkat lunak konferensi jarak jauh Microsoft NetMeeting.

Klien-klien yang mendukungnya bervariasi, mulai dari sebagian besar sistem operasi Windows 32-bit (termasuk Windows CE dan PocketPC), hingga sistem operasi lainnya, seperti Linux, FreeBSD, UNIX Solaris, dan Apple Mac OS X. Secara *default*, *server* yang membuka protokol ini, akan membuka *port* **TCP 3389**. [7]

Protokol ini memiliki layanan seperti dukungan terhadap kedalaman warna 32-bit, enkripsi 128-bit (dengan menggunakan algoritma enkripsi RC4), dukungan terhadap protokol *Transport Layer Security* (TLS), redireksi suara (suara yang sebenarnya keluar di *server* bisa didengarkan pada klien local), redireksi sistem berkas, (sistem berkas lokal yang menyimpan berkas-berkas pengguna dapat digunakan di dalam sebuah sesi terminal *server*), redireksi *port* (para pengguna dapat mengakses *port* serial dan paralel lokal secara langsung).

2.7 *Class of Service*

CoS adalah bentuk *queuing* (antrian) berdasarkan prioritas yang umum digunakan dalam sejumlah trafik dan protokol jaringan. Merupakan cara memprioritaskan paket-paket berdasarkan jenis aplikasi (*video streaming*, transfer *file*), jenis pengguna (CEO, sekretaris, staff) atau pengaturan lain. CoS mengklasifikasikan paket-paket dengan memeriksa parameter atau CoS *marking* dan menempatkan paket-paket tersebut dalam antrian prioritas yang berbeda berdasarkan kriteria yang telah ditentukan. [9]

Misalnya label dengan prioritas "*first class*" digunakan untuk aplikasi yang memerlukan penyelesaian cepat seperti *video* atau *voice call*, sedangkan label dengan prioritas yang lebih rendah digunakan untuk aplikasi yang kurang sensitif terhadap waktu seperti *email*, ftp, dan *web surfing*.

BAB III
PERANCANGAN NETWORK

Pada bab ini dibahas mengenai perancangan *system network* melalui *tunnel*, baik itu menggunakan *tunnel* L2TP dan PPTP.

Dalam perancangan *network* ini tiap-tiap *network* terbagi menjadi tiga bagian, yaitu *network* A, *network* B dan *network* ISP. *Network* B bertindak sebagai *network client* yang akan mengakses *network* A yang merupakan *server farm network* yang terdiri dari FTP *server* dan *streaming server* melalui *network* ISP yang bertindak sebagai penyedia layanan internet.

3.1 Perancangan *Network* Secara Global

PPTP *Tunnel*

Protokol yang mengizinkan hubungan *Point to Point Protocol* (PPP) melewati jaringan IP, membuat enkapsulasi frame pada IP dalam sebuah *Generic Routing Encapsulation* (GRE), kemudian GRE dibungkus dalam sebuah paket IP untuk membuat *tunne*l. PPTP client menggunakan TCP pada *port* 1723 untuk membentuk PPTP *control connection* pada *tunnel*.[5] Perancangan *network* PPTP secara global dijelaskan pada Gambar 3.1

Ada beberapa pertimbangan sebuah corporate memilih PPTP sebagai tunnelnya:

1. Kebutuhan 2 buah *network* yang terpisah dalam hal ini LAN (*Local Area Network*) yang dibatasi oleh NAT (*Network Address Translator*).
2. Data yang dilewatkan pada *tunnel* memerlukan enkripsi data oleh MPPE (*Microsoft Point-to-Point Encryption*).

L2TP *Tunnel*

L2TP adalah salah satu protokol yang menggunakan Layer 5 pada *seven OSI layer*. Namun protokol L2TP memiliki fitur layaknya protokol pada *data link* layer (layer 2 OSI) dimana L2TP seolah-olah membentuk satu interkoneksi Layer 2.

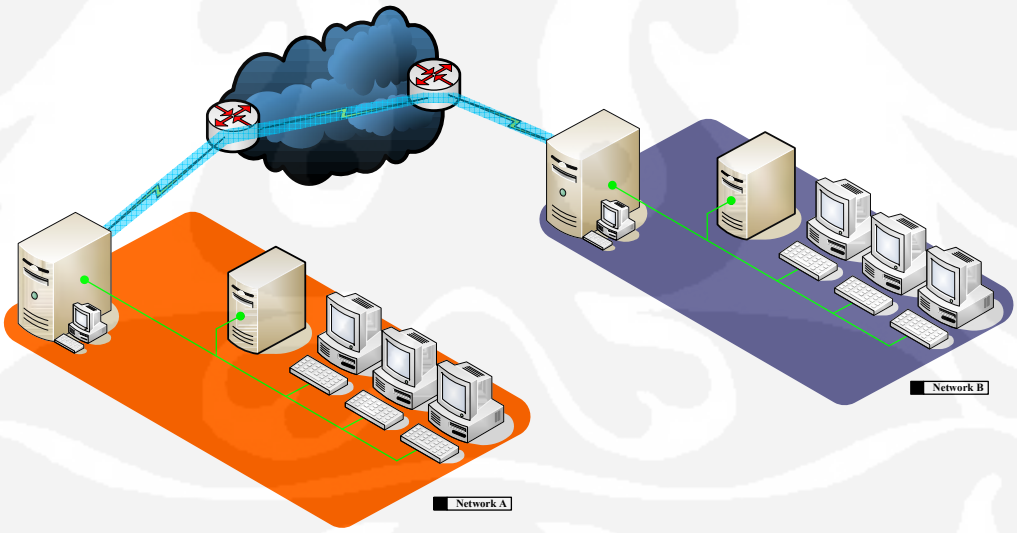
L2TP menggunakan *transport layer* UDP pada *port* 1701. Seluruh paket L2TP di transmisikan dengan mengikuti aturan UDP tersebut. Besar paket L2TP

(*Payload* dan *Header*) adalah sebesar packet UDP datagram, dan kemudian dilanjutkan dengan pengalamat IP.[6]

Pada umumnya *network* L2TP terdiri atas 2 buah *router* yang merupakan L2TP *Server* dan L2TP *Client* yang mana kedua buah *router* ini yang terhubung melalui *cloud provider* baik itu *cloud internet*, MPLS, *frame relay* dan lain-lain. Untuk lebih jelasnya perancangan *network* L2TP tidak jauh berbeda dengan perancangan *network* PPTP yang dijelaskan pada Gambar 3.1

Ada beberapa pertimbangan sebuah *corporate* memilih L2TP sebagai *tunnel*nya:

1. Kebutuhan 2 buah *network* yang terpisah dalam hal ini LAN (*Locak Area Network*) yang dibatasi oleh NAT *(Network Address Translator*).
2. Data yang dilalui pada *tunnel* memerlukan enkripsi data (IPSec)
3. Data yang dilewatkan pada *tunnel* harus dapat melakukan *broadcast* dan *multicast address*. Contohnya: applikasi *Remote Trading*.



Gambar 3.1 PPTP dan L2TP *Network Topology*

3.2. Simulasi Perancangan *Network*

Secara umum model jaringan yang akan disimulasikan terbagi atas dua model, yang masing-masing model memiliki topologi yang hampir sama. Topologi ini terdiri atas 3 buah *network* yaitu *network* A, *network* B, dan *network* ISP. *Network* B merupakan *network client* yang akan mengakses *network* A yang berisikian *server farm* melalui *network* ISP yang merupakan penyedia jasa layanan internet.

3.2.1 Peralatan Yang Digunakan

Peralatan yang digunakan dalam perancangan ini terdiri dari perangkat keras dan perangkat lunak. Daftar peralatan yang digunakan adalah :

Perangkat Keras

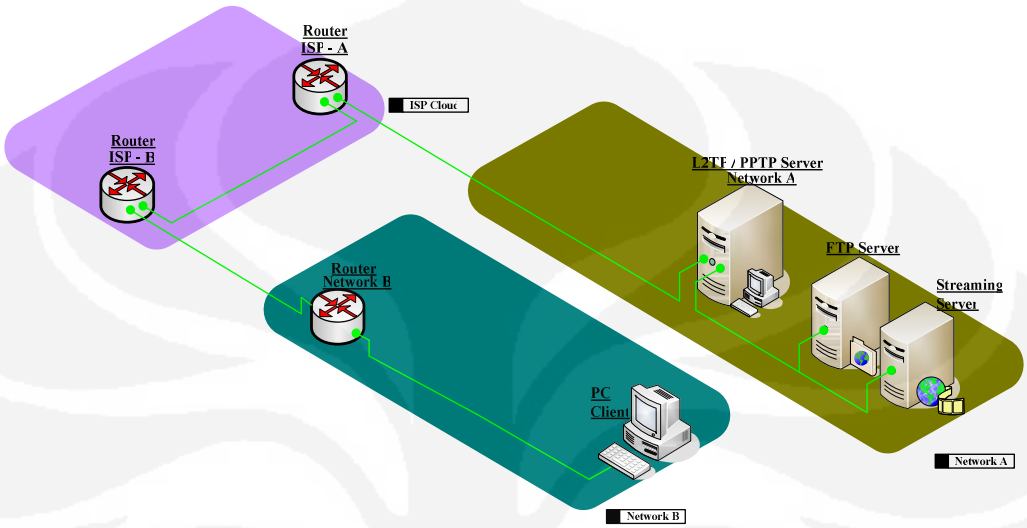
1. Cisco router 2801; 3 buah yang digunakan untuk:
   - 1 buah sebagai router ISP A
   - 1 buah sebagai router ISP B
   - 1 buah sebagai router Network Client
2. Personal Computer; 3 buah yang digunakan untuk:
   - 1 buah sebagai PC Server
   - 1 buah sebagai PC Client
   - 1 buah PC sebagai router mikrotik untuk L2TP/PPTP Server

Perangkat Lunak

1. Cisco IOS C2801-advipservicesk9-mz.124-11.T
2. Mikrotik v.2.9.26
3. FileZilla Server 0.9.31 (FTP Server)
4. FileZilla Client 3.2.4.1 (FTP Client)
5. VLC Streaming Server
6. VLC Streaming Client
7. Winbox 2.13
8. Wireshark Network Protocol Analyzer v.1.0.8
9. Windows XP Professional Edition Service Pack 2

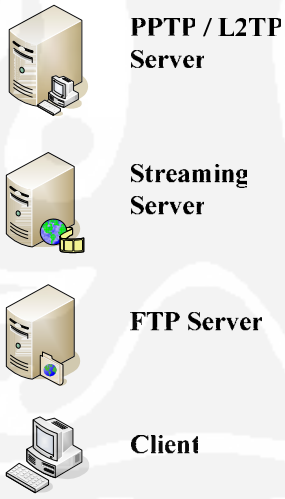
3.2.2. *Setting* Topologi

Untuk penelitian ini pertama-tama yaitu membuat perancangan *network* PPTP dan L2TP dengan topologi seperti ditunjukkan pada Gambar 3.2



Gambar 3.2 Perancangan Simulasi *Network* PPTP dan L2TP

Keterangan:



- Topologi terdiri dari 3 *Network*: *Network* A (*Server*). *Network* B (*Client*) dan *Network* ISP
- *Network* B (*Client*) akan mengakses *Network* A (*Server*) melalui *Network* ISP dengan menggunakan PPTP dan L2TP *tunnel*.

- IP *address* yang digunakan adalah IP *address* kelas C yaitu 192.168.1.0/24 yang disubnet menjadi 32 subnet.

Tabel 3.1 Penggunaan IP pada simulasi network

Device	IP address	Interface	Connected To
FTP *server*	172.16.1.123	FastEthernet 0	*Network* A
RDP *server*	172.16.1.25	FastEthernet 0	*Network* A
Stream *server*	172.16.1.103	FastEthernet 0	*Network* A
PPTP/L2TP *server*	172.16.1.190	FastEthernet 0	*Network* A
PPTP/L2TP *server*	192.168.1.9	FastEthernet 1	*Network* A to ISP A
PPTP/L2TP *Server*	20.20.20.1 / 10.10.10.1	*Tunnel* 0	*Network* A to *Network* B
Router ISP A	192.168.1.10	FastEthernet 0/1	ISP A to *Network* A
Router ISP A	192.168.1.17	FastEthernet 0/0	ISP A to ISP B
Router ISP B	192.168.1.18	FastEthernet 0/1	ISP B to ISP A
Router ISP B	192.168.1.25	FastEthernet 0/0	ISP B to *Network* B
Router *Network* B	192.168.1.26	FastEthernet 0/1	*Network* B to ISP B
Router *Network* B	192.168.1.32	FastEthernet 0/0	*Network* B
PC *Client*	20.20.20.2 / 10.10.10.2	Dial-up VPN	*Network* B to *Network* A
PC *Client*	192.168.1.33	FastEthernet 0	*Network* B

- *Bandwidth* pada *network* ISP di set sebesar 1 Mbps (ISP A dan ISP B).
- *Set-up* PPTP *Server* pada mikrotik
- *Set-up* L2TP *Server* pada mikrotik
- *Set-up* VPN *Client:*
    - *Set-up* PPTP *Client*
    - *Set-up* L2TP *Client*

3.3 Pembagian *Class of Service*

Trafik-trafik yang akan dilewatkan dalam tunnel pada simulasi ini adalah UDP, TCP dan RDP. Trafik-trafik tersebut dibedakan atas *priority, Commited Information Rate* (CIR) dan *Maximum Information Rate* (MIR). Pembagian *class*

of service dilakukan dengan membagi alokasi *bandwidth* per masing-masing jenis trafik seperti pada Tabel 3.2

Tabel 3.2 Alokasi Bandwidth tiap Trafik

Trafik	Aplikasi	CIR (limit-at)	MIR (max-limit)	Priority
UDP	Video Streaming	512 Kbps	1024 Kbps	1
RDP	Remote Desktop	192 Kbps	1024 Kbps	3
TCP	FTP	256 Kbps	1024 Kbps	5

*Keterangan: semakin kecil angka priority semakin tinggi prioritasnya.

3.4 Pengujian *Netwok*

Setelah merancang sesuai dengan topologi, pengujian *network* dilakukan dengan langkah-langkah sebagai berikut:

PPTP/L2TP *client* (*network* B) melakukan dial VPN ke PPTP/L2TP *Server* (*network* A), setelah *client* terhubung ke *server* (*tunnel* up) maka proses transfer data sudah bisa dilakukan. Pada simulasi ini data yang akan dilewatkan adalah aplikasi-aplikasi yang bekerja dengan protokol TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*) dan RDP (*Remote Desktop Protocol*) .

1. Simulasi transfer data TCP dengan aplikasi FTP (*File Transfer Protocol*) dari *network* A ke *network* B menggunakan *software* FileZilla *Server*/*Client*.

2. Simulasi transfer data UDP dengan aplikasi *video streaming* dari *network* A ke *network* B menggunakan *software* VLC *Streaming Server*/*Client*.

3. Simulasi transfer data UDP (Video *Streaming*) dan TCP (FTP) bersamaan dengan aplikasi *Remote Desktop* (RDP) dari *network* A ke *network* B.

4. Simulasi transfer data UDP (*Video Streaming*) dan TCP (FTP) bersamaan dengan aplikasi *Remote Desktop* (RDP) dari *network* A ke *network* B dengan menggunakan *Class of Service*.

3.5 Pengambilan Data

Pengambilan data dilkakukan dengan metode *sniffing* menggunakan *software* wireshark. *Sniffing* data dilakukan di *Ethernet client* pada saat:

1. PPTP/L2TP *client* di *Network* B melakukan *dial* VPN ke PPTP/L2TP *Server* di *Network* A
2. PPTP/L2TP *client* di *Network* B melakukan transfer FTP dari *Network* A
3. PPTP/L2TP *client* di *Network* B melakukan *streaming video* dari *Network* A
4. PPTP/L2TP *client* di *Network* B melakukan transfer FTP dan *streaming video* dari *Network* A bersamaan dengan aplikasi *Remote Desktop* dari *client* ke *Network* A
5. PPTP/L2TP *client* di *Network* B melakukan transfer FTP dan *streaming video* dari *Network* A bersamaan dengan aplikasi *Remote Desktop* dari *client* ke *Network* A dengan diberlakukannya *Class of Service* pada masing-masing trafik.

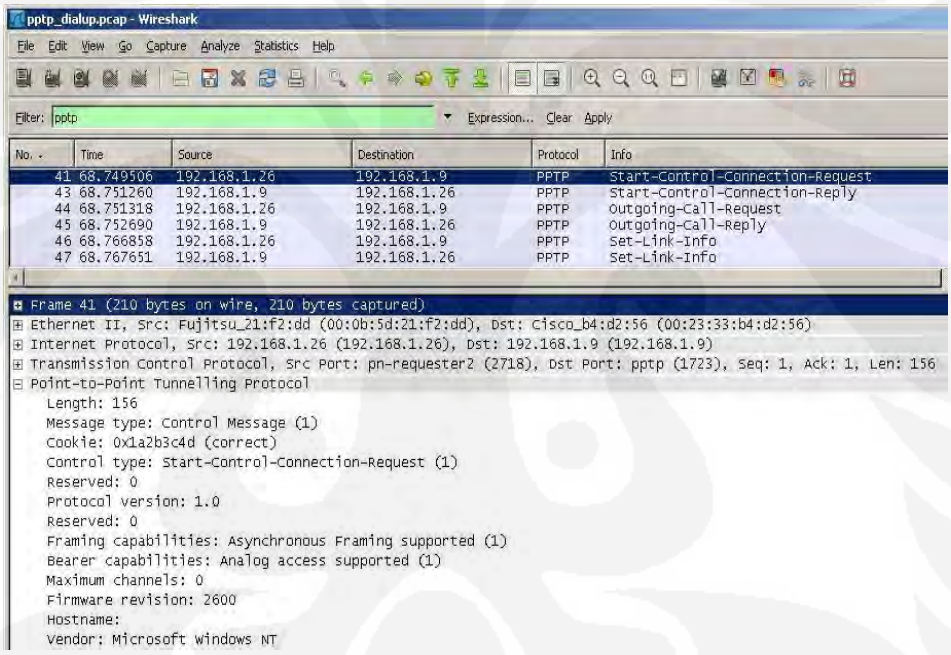
BAB IV
ANALISA DATA

4.1 *Dial* VPN *Connection*

Pengambilan data dilakukan pada saat PPTP/L2TP *client* melakukan *dial* VPN ke PPTP/L2TP *server*, yang ingin di analisa dari pengambilan data ini adalah bagaimana cara protokol PPTP dan L2TP membangun *tunnel* tersebut sebelum data dapat dilewatkan melalui *tunnel*.

4.1.1 *Dial* VPN PPTP

Sniffing data pada saat PPTP *client* membuat koneksi ke PPTP *server*:



Gambar 4.1 *Dial* VPN PPTP

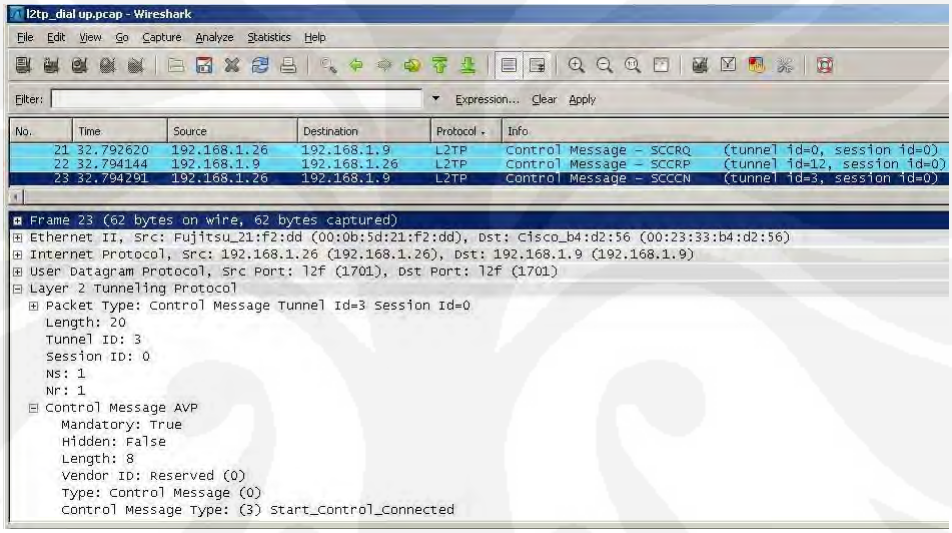
Terjadi pertukaran *message* antara PPTP *client* dan PPTP *server* melalui koneksi TCP untuk membuat *tunnel*, yaitu:

- PPTP *Client* mengirim Start-Control-Connection-Request kepada PPTP *Server*; permintaan untuk memulai *session*.
- PPTP *Server* mengirim Start-Control-Connection-Reply kepada PPTP *Client*; untuk menjawab *start session.*

- PPTP *Client* mengirim Outgoing-Call-Request kepada PPTP *Server*; permintaan untuk melakukan *outgoing call*.
- PPTP *Server* mengirim Outgoing-Call-Reply kepada PPTP *Client*; respon dari *server* telah menerima Outgoing-Call-Request.
- PPTP *Client* mengirim Set-Link-Info kepada PPTP *Server*; permintaan untuk merubah *setting* koneksi antara *client* dan *server*.

4.1.2 *Dial* VPN L2TP

Sniffing data pada saat L2TP *client* membuat koneksi ke L2TP *server*:



Gambar 4.2 *Dial* VPN L2TP

Ada tiga *message* yang dipertukarkan untuk membangun koneksi:

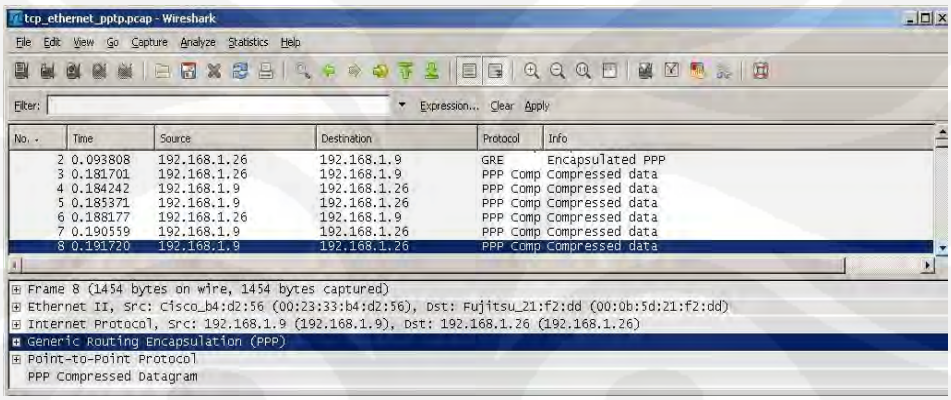
- L2TP *Client* mengirim SCCRQ (Start-Control-Connection-Request) ke L2TP *Server*; untuk menginisialisi *tunnel* antara *server* dan *client*, untuk proses pembentukan *tunnel*.
- L2TP *Server* mengirim SCCRP (Start-Control-Connection-Reply) ke L2TP *Client*; untuk mengindikasikan bahwa SCCRQ telah diterima dan pembentukan *tunnel* harus dilanjutkan. Dikirim sebagai balasan dari message SCCRQ yang dikirim oleh L2TP *Client*.
- L2TP *Client* mengirim SCCCN (Start-Control-Connection-Connected) ke L2TP *Server*; dikirim sebagai balasan dari *message* SCCRP yang dikirim oleh L2TP *Server* mengindikasikan proses pembentukan *tunnel* telah selesai.

4.2 Enkapsulasi pada Protokol *Tunneling*

Pengambilan data dilakukan pada saat PPTP/L2TP *client* melakukan transfer data dari *Network* A, yang akan di analisa dari pengambilan data ini adalah bagaimana cara protokol PPTP dan L2TP mengenkapsulasi data yang dilewatkan dalam *tunnel*.

4.2.1 Enkapsulasi data pada PPTP

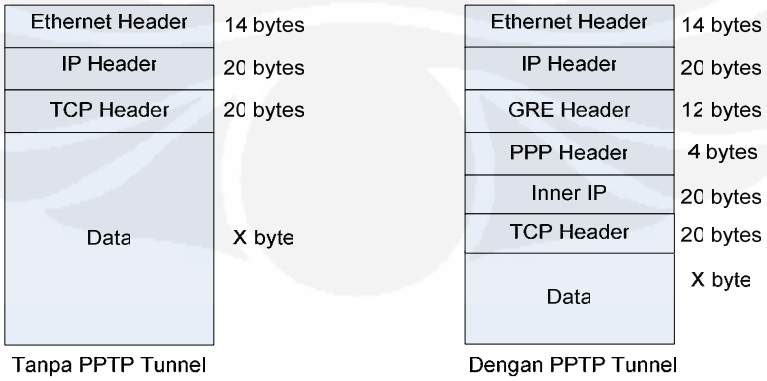
Sniffing data pada saat PPTP *client* melakukan transfer paket TCP menggunakan aplikasi FTP dari *Network* A:



Gambar 4.3 *Sniffing* paket TCP pada PPTP *Tunnel*

Data yang dilewatkan antara *Server* dan *Client* ditransmisikan pada IP *datagram* yang memiliki paket PPP. GRE (*Generic Routing Encapsulation*) melakukan enkapsulasi paket IP yang berisi paket PPP menjadi paket GRE, kemudian paket GRE tersebut dibungkus dalam sebuah paket IP untuk dilewatkan dalam *tunnel*.

Paket TCP yang dilewatkan melalui PPTP *Tunnel* akan berbeda dengan paket TCP yang dilewatkan melalui pengalamatan IP biasa.

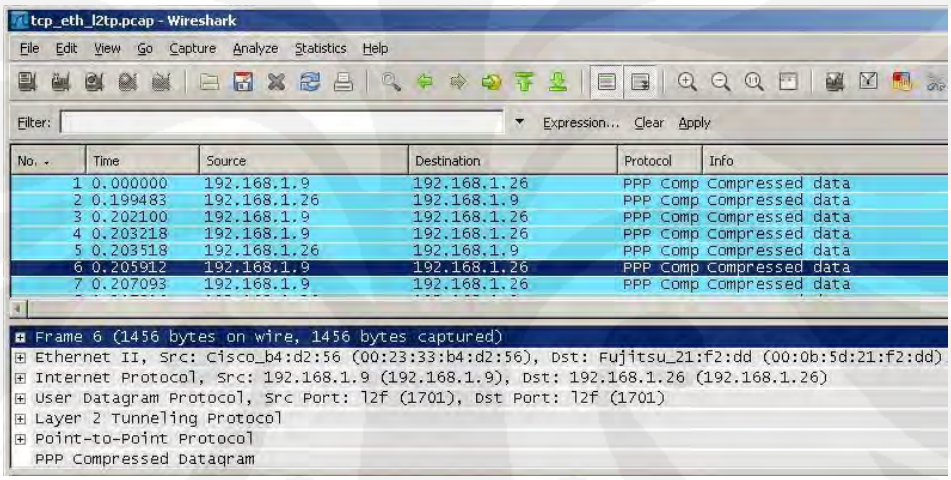


Gambar 4.4 Perbandingan Paket TCP Dengan PPTP dan Tanpa PPTP

Paket PPP dibuat oleh PPTP *server* merupakan paket data yang telah terenksripsi, GRE *header* meringkas paket PPP tersebut menjadi IP *Datagram*, kemudian IP *Datagram* dibungkus oleh IP *Delivery Header* yang membawa informasi penting untuk *datagram* untuk melintasi internet. IP *Datagram* tersebut dirutekan melalui internet hingga mencapai PPTP *client* yang terhubung ke internet.

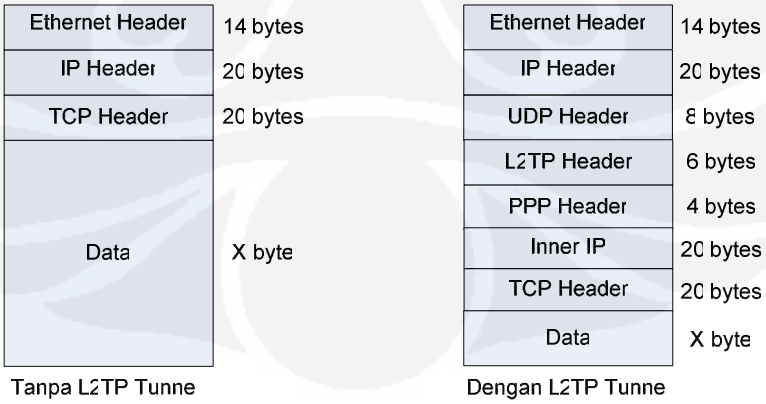
4.2.2 Enkapsulasi Data Pada L2TP

Sniffing data pada saat L2TP *client* melakukan transfer paket TCP menggunakan aplikasi FTP dari *Network* A:



Gambar 4.5 *Sniffing* paket TCP pada L2TP *Tunnel*

Paket TCP yang dilewatkan melalui L2TP *Tunnel* akan berbeda dengan paket TCP yang dilewatkan melalui pengalamatan IP biasa.



Gambar 4.6 Perbandingan Paket TCP Dengan L2TP dan Tanpa L2TP

Paket PPP dienkapsulasi oleh *header* L2TP dan paket transport UDP, kemudian paket di tambahkan IP *Header* untuk dilewatkan melalui *tunnel* sampai ke alamat tujuan.

Dengan menambahkan *tunnel* PPTP atau L2TP berarti akan mengurangi besar byte *payload* pada data yang akan dikirim. Dengan berkurangnya jumlah byte payload pada data yang akan dikirim dalam satuan waktu, maka pengiriman data melalui *tunnel* akan memakan lebih lama.

4.3 *Throughput* Pada Jaringan PPTP dan L2TP

Pengambilan data dilakukan pada saat PPTP/L2TP *client* melakukan *streaming video* (paket UDP) dari *Network* A, yang akan di analisa dari pengambilan data ini adalah besarnya *throughput* pada masing-masing jaringan.

Dari data hasil *sniffing* pada simulasi ini di ambil sampel data kedua model dengan transfer time selama 67 s

Traffic	Captured
Packets	5164
Between first and last packet	67.012 sec
Avg. packets/sec	77.061
Avg. packet size	1378.658 bytes
Bytes	7119388
Avg. bytes/sec	106240.899
Avg. MBit/sec	0.850

Throughput Pada L2TP Tunnel

Traffic	Captured
Packets	6046
Between first and last packet	67.006 sec
Avg. packets/sec	90.231
Avg. packet size	1286.276 bytes
Bytes	7776826
Avg. bytes/sec	116062.071
Avg. MBit/sec	0.928

Throughput Pada PPTP Tunnel

Gambar 4.7 Perbandingan besar *throughput* pada PPTP dan L2TP

Dengan *bandwidth* sebesar 1024 Kbps, dalam waktu 67 s jaringan PPTP dapat mengirim data sebesar 7776826 bytes, maka *throughput* pada jaringan PPTP dapat dihitung:

Data Received = 7776826 bytes

Simulation Time = 67 s

Throughput PPTP [7] $= \frac{Data\ \mathrm{Re}ceived}{TransferTime} * \left(\frac{8}{1000}\right) Kbps$

$= \frac{7776826}{67} * \left(\frac{8}{1000}\right) Kbps$

***Throughput* PPTP = 928. 5762 Kbps**

Sedangkan jaringan L2TP dalam waktu 67s dapat mengirim data sebesar 7119388 bytes, maka *throughput* pada jaringan L2TP dapat dihitung:

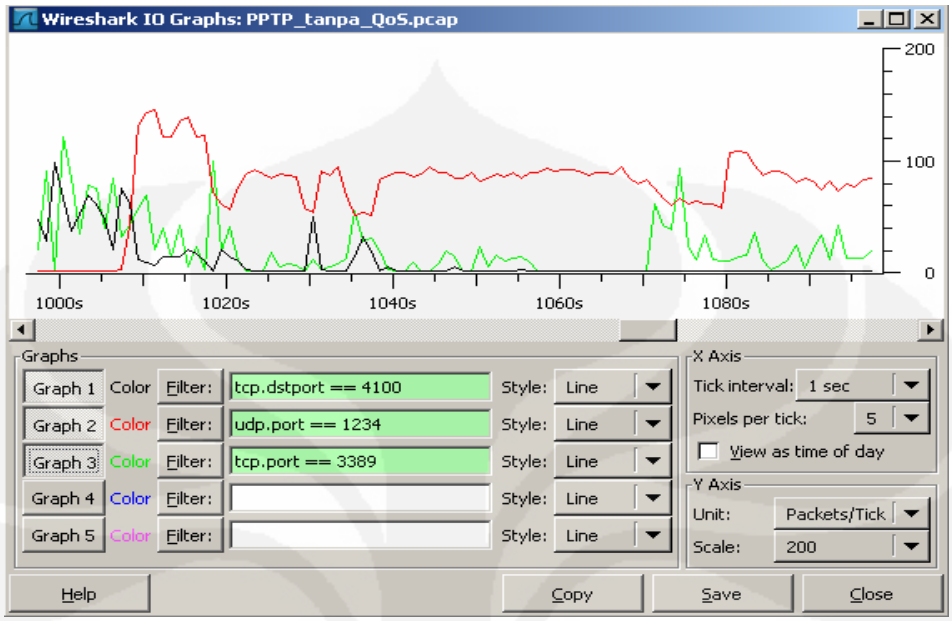
Data Received	= 7119388 bytes
Simulation Time	= 67 s
Throughput L2TP [7]	$= \frac{Data\,\mathrm{Re}\,ceived}{TransferTime} * \left(\frac{8}{1000}\right) Kbps$
	$= \frac{7119388}{67} * \left(\frac{8}{1000}\right) Kbps$
***Throughput* L2TP**	**= 850. 0761 Kbps**

Dari simulasi ini terlihat dalam waktu yang sama PPTP dapat mentransmisikan data lebih besar dari L2TP, hal ini dikarenakan penambahan byte *header* pada proses enkapsulasi PPTP tidak sebanyak pada L2TP, sehingga besar byte payload data yang dapat dikirimkan tiap waktu transmisi pada PPTP lebih besar dari L2TP. Jika MTU (*Maximum Transmission Unit*) dari pengalamatan IP biasa adalah 1500 bytes, maka *payload* data yang dapat dikirim tiap waktu transmisi adalah 1460 bytes (+ *header* TCP 20 bytes + *header* IP 20 bytes), sedangkan pada *tunneling* protokol setelah data mengalami enkapsulasi seperti pengalamatan IP biasa, data mengalami enkapsulasi yang terkait oleh proses pengiriman data tersebut melalui *tunnel*. Pada PPTP maka *payload* data yang dapat dikirimkan tiap waktu transmisi adalah 1444 bytes (+ 4 bytes *header* PPP + 4 bytes *header* GRE), dan pada L2TP sebesar 1442 bytes (+ 4 bytes *header* PPP + 6 bytes *header* L2TP + 8 bytes *header* UDP).

4.4 Trafik TCP, UDP dan RDP Pada PPTP dan L2TP

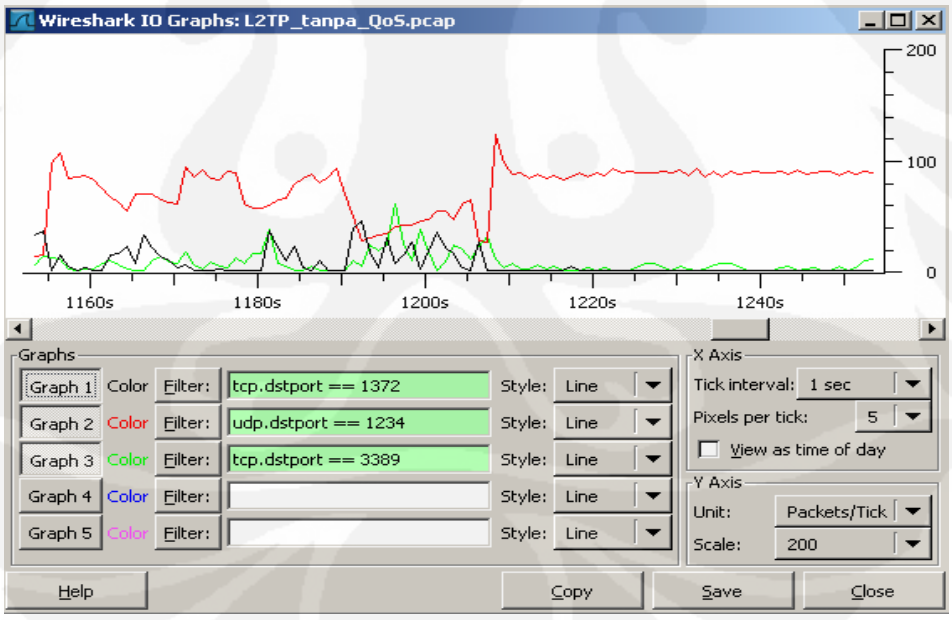
Pengambilan data dilakukan pada saat *client* melakukan transfer data FTP, *streaming video* dan melakukan *remote desktop* ke *server*, yang akan di analisa dari pengambilan data ini adalah bagaimana karakter TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*) dan RDP (*Remote Desktop Protocol*) di dalam *tunnel* jika ketiga trafik tersebut dilewatkan secara bersamaan.

Sniffing data pada saat PPTP *client* melakukan transfer data FTP, *streaming video* dan melakukan *remote desktop* ke *server*:



Gambar 4.8 Grafik *throughput* TCP, UDP dan RDP pada PPTP *tunnel*

Sniffing data pada saat L2TP *client* melakukan transfer data FTP, *streaming video* dan melakukan *remote desktop* ke *server*:



Gambar 4.9 Grafik *throughput* TCP, UDP dan RDP pada L2TP *tunnel*

Dari kedua grafik diatas terlihat protokol UDP memiliki *throughput* yang paling besar dibandingkan dengan protokol TCP dan RDP (TCP *port* 3389), dikarenakan karakter dari UDP itu sendiri merupakan protokol yang bersifat

connectionless (tidak diperlukan *handshaking* terlebih dahulu sebelum mengirimkan data), tidak ada mekanisme *congestion control* dan memperbaiki kesalahan, protokol UDP hanya menekankan kecepatan kirim. Sedangkan TCP bersifat *connection-oriented* (akan melakukan *handshaking* terlebih dahulu sebelum mengirimkan data), ada mekanisme *sequencing* sehingga data yang dikirim berurutan sesuai dengan susunan yang benar, mengecek adanya kesalahan (*checksum*) dan akan melakukan pengiriman ulang terhadap data kiriman yang hilang dan rusak, protokol TCP menekankan pada keandalan dan kepastian data terkirim. Secara teknis protokol UDP juga memiliki *header* yang lebih kecil 8 bytes dibandingkan dengan protokol TCP 20 bytes. Berdasarkan karakterisitik masing-masing protokol inilah yang mengakibatkan protokol UDP akan menggunakan hampir seluruh alokasi *bandwidth* yang tersedia, sedangkan protokol TCP hampir tidak mendapatkan *bandwidth* sama sekali, yang bahkan mengakibatkan koneksi TCP terputus.

Untuk mengatasi hal tersebut diperlukan adanya pembagian *Class of Service* untuk membedakan *Type of Service* dari masing-masing protokol. Dengan menggunakan konsep *Hierarchical Token Bucket* (HTB) *queue* menjadi lebih terstruktur, dengan melakukan pengelompokan-pengelompokan bertingkat. Trafik-trafik yang akan dilewatkan dibedakan atas *priority, Commited Information Rate* (CIR) dan *Maximum Information Rate* (MIR).

- Trafik UDP : CIR = 512 Kbps; MIR = 1024 Kbps; Priority = 1
- Trafik RDP : CIR = 192 Kbps; MIR = 1024 Kbps; Priority = 3
- Trafik TCP : CIR = 256 Kbps; MIR = 1024 Kbps; Priority = 5

CoS di *attach* pada PPTP/L2TP *Server*.

Pertama yang dilakukan adalah *marking* paket:

```
[admin@MikroTik] > /ip firewall mangle add chain=prerouting \
\... protocol=udp dst-port=1234 action=mark-connection new-connection-mark=streaming
passthrough=yes
[admin@MikroTik] > /ip firewall mangle add chain=prerouting \
\... connection-mark=streaming action=mark-packet new-packet-mark=packet_mark1
passthrough=no
[admin@MikroTik] > /ip firewall mangle add chain=prerouting \
\... protocol=tcp dst-port=3389 action=mark-connection new-connection-mark=rdp_conn
passthrough=yes
[admin@MikroTik] > /ip firewall mangle add chain=prerouting \
\... connection-mark=rdp_conn action=mark-packet new-packet-mark=packet_mark2
passthrough=no
```

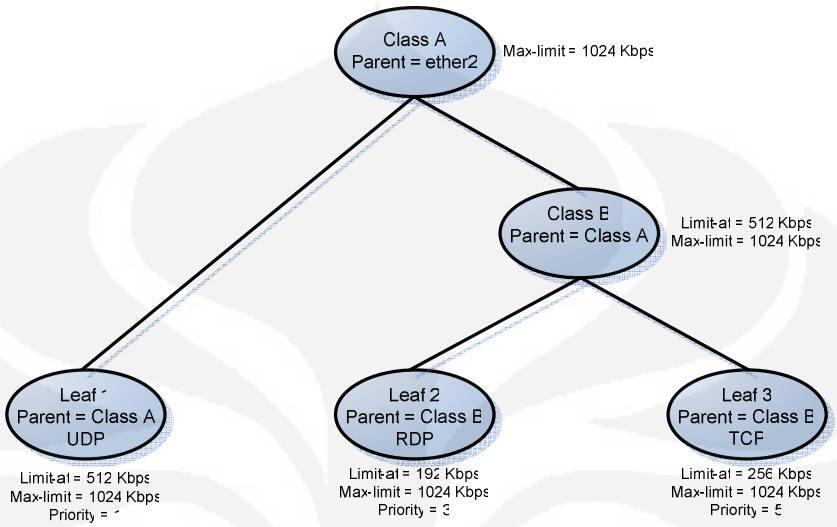
```
[admin@MikroTik] > /ip firewall mangle add chain=prerouting \
\... protocol=tcp dst-port=21 action=mark-connection new-connection-mark=ftp_conn
passthrough=yes
[admin@MikroTik] > /ip firewall mangle add chain=prerouting \
\... connection-mark=ftp_conn action=mark-packet new-packet-mark=packet_mark3
passthrough=no


[admin@MikroTik] ip firewall mangle> print
Flags: X - disabled, I - invalid, D - dynamic
 0   chain=prerouting protocol=udp dst-port=1234 action=mark-connection
     new-connection-mark=streaming passthrough=yes

 1   chain=prerouting connection-mark=streaming action=mark-packet
     new-packet-mark=packet_mark1 passthrough=no

 2   chain=prerouting protocol=tcp dst-port=3389 action=mark-connection
     new-connection-mark=rdp_conn passthrough=yes

 3   chain=prerouting connection-mark=rdp_conn action=mark-packet
     new-packet-mark=packet_mark2 passthrough=no

 4   chain=prerouting protocol=tcp dst-port=21 action=mark-connection
     new-connection-mark=ftp_conn passthrough=yes

 5   chain=prerouting connection-mark=ftp_conn action=mark-packet
     new-packet-mark=packet_mark3 passthrough=no
```

Setelah *marking* packet, membuat *queue tree* pada *interface* LAN:

queue tree:

```
[admin@MikroTik] queue tree> add name=ClassA parent=ether2 max-limit=1024000
[admin@MikroTik] queue tree> add name=ClassB parent=ClassA max-limit=1024000
\... limit-at=512000
[admin@MikroTik] queue tree> add name=Leaf1 parent=ClassA max-limit=1024000 \
\... limit-at=512000 packet-mark=packet_mark1 priority=1
[admin@MikroTik] queue tree> add name=Leaf2 parent=ClassB max-limit=1024000 \
\... limit-at=192000 packet-mark=packet_mark2 priority=3
[admin@MikroTik] queue tree> add name=Leaf3 parent=ClassB max-limit=1024000 \
\... limit-at=256000 packet-mark=packet_mark3 priority=5


[admin@MikroTik] > queue tree
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid
 0   name="classA" parent=ether2 packet-mark="" limit-at=0 queue=default
     priority=8 max-limit=1024000 burst-limit=0 burst-threshold=0 burst-time=0s

 1   name="classB" parent=classA packet-mark="" limit-at=512000 queue=default
     priority=8 max-limit=1024000 burst-limit=0 burst-threshold=0 burst-time=0s

 2   name="leaf1" parent=classA packet-mark=packet_mark1 limit-at=512000
     queue=default priority=1 max-limit=1024000 burst-limit=0
      burst-threshold=0 burst-time=0s

 3   name="leaf2" parent=classB packet-mark=packet_mark2 limit-at=192000
     queue=default priority=3 max-limit=1024000 burst-limit=0
     burst-threshold=0 burst-time=0s

 4   name="leaf3" parent=classB packet-mark=packet_mark3 limit-at=256000
     queue=default priority=5 max-limit=1024000 burst-limit=0
     burst-threshold=0 burst-time=0s
```

Setelah memasukkan CoS pada PPTP/L2TP *server*, maka struktur *queue* dan alokasi *bandwidth*-nya akan terlihat seperti gambar dibawah ini:



Gambar 4.10 Struktur *queue* dan alokasi *bandwidth* menggunakan HTB

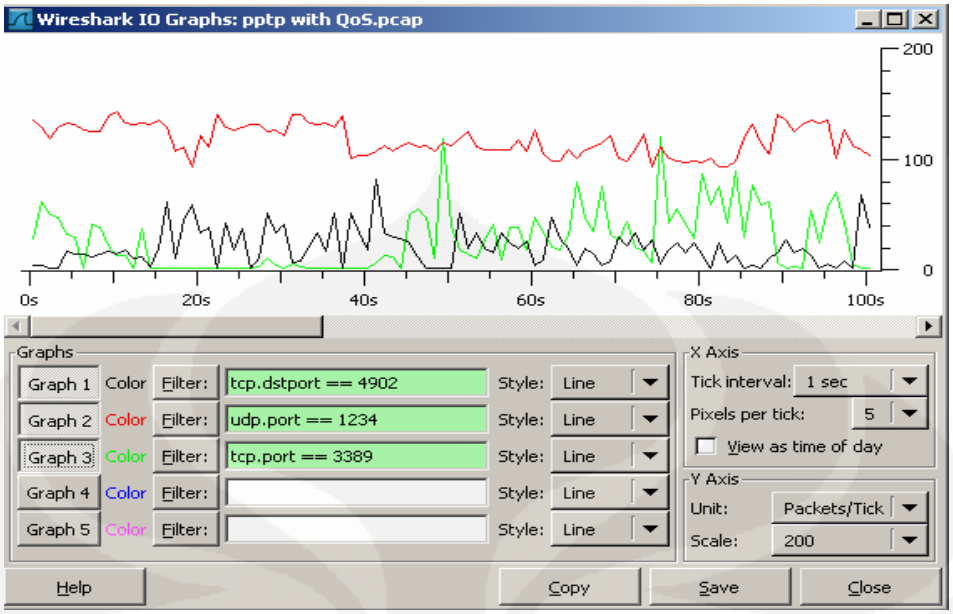
Dengan konfigurasi HTB seperti diatas maka:

- UDP akan mendapatkan *bandwidth* sebesar 576 Kbps.
- RDP akan mendapatkan *bandwidth* sebesar 192 Kbps.
- TCP akan mendapatkan *bandwidth* sebesar 256 Kbps.

Karena setelah menjamin semua *queue* pada ”limit-at”/CIR (*Commited Information Rate*) HTB akan memberikan *throughput* pada *queue* yang memiliki prioritas paling tinggi.

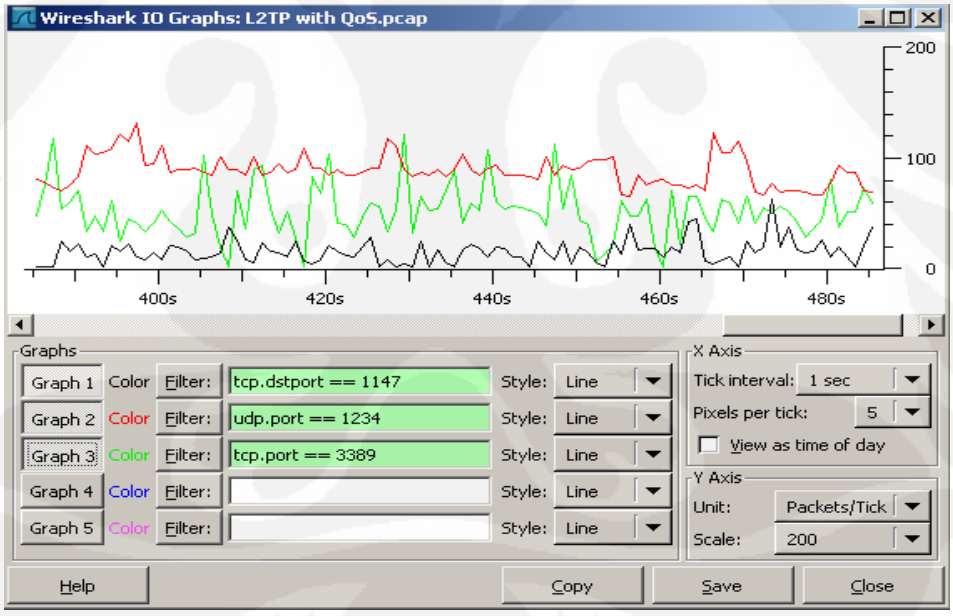
Setelah CoS di *attach*, *sniffing* data kembali dilakukan pada saat jaringan dilewatkan trafik-trafik dengan protokol UDP, TCP dan RDP pada saat yang bersamaan.

Sniffing data pada saat jaringan PPTP dilewatkan protokol UDP, TCP dan RDP (dengan CoS):



Gambar 4.11 Grafik *throughput* TCP, UDP dan RDP pada PPTP *tunnel* (Dengan CoS)

Sniffing data pada saat jaringan L2TP dilewatkan protokol UDP, TCP dan RDP (dengan CoS):



Gambar 4.12 Grafik *throughput* TCP, UDP dan RDP pada L2TP *tunnel* (Dengan CoS)

Dari kedua grafik diatas terlihat bahwa konfigurasi dari CoS bekerja, setiap protokol mendapatkan alokasi *bandwidth* sesuai dengan besar *bandwidth* yang ditentukan, dengan demikian protokol UDP tidak akan mendominasi seluruh *bandwidth* yang tersedia di *network*, sehingga aplikasi yang berjalan diatas TCP

(FTP dan *Remote Desktop*) tetap bisa berjalan pada alokasi *bandwidth* yang telah ditentukan.

4.5 Analisa Keseluruhan

Setelah melakukan perbandingan dari hasil pengujian untuk kedua konfigurasi, dapat disimpulkan bahwa PPTP dan L2TP menawarkan fungsi yang berbeda. L2TP dapat digunakan pada jaringan *non-IP based*, dan protokol membangun *tunnel* untuk *maintenance* dan *control* menggunakan menggunakan *format message* dan protokol yang sama. Sedangkan PPTP hanya bekerja pada jaringan *IP-based* dan menggunakan TCP *Control Connection* yang terpisah untuk *tunnel maintenance*.

Throughput pada PPTP lebih besar dari L2TP karena pada L2TP proses enkapsulasi mengakibatkan penambahan byte header yang lebih banyak dibandingkan pada PPTP. Meskipun L2TP memiliki nilai *throughput* yang kecil, L2TP dapat dikombinasikan dengan enkripsi IpSec (Layer 3) yang jauh lebih aman daripada enkripsi pada PPTP yaitu MPPE (*Microsoft Point to Point Encryption*). L2TP dengan IpSec menyediakan keamanan berlapis yang jika digunakan dengan benar dapat menjamin keamanan data yang dilewatkan didalamnya. Microsoft juga telah membuat *setup* L2TP semudah *setup* PPTP, kemudahan yang hanya membutuhkan beberapa klik *mouse* saja dan keamanan yang dijanjikan membuat L2TP dengan IpSec menjadi pilihan protokol VPN.

BAB V
KESIMPULAN

Setelah melakukan perancangan simulasi *network* PPTP dan L2TP, melakukan pengambilan data dan menganalisa data hasil simulasi tersebut, maka ada beberapa kesimpulan yang dapat diambil:

1. Penggunaan *tunnel* dalam *network* akan menambah byte *header* yang terkait dengan proses enkapsulasi *tunne*l itu sendiri, yang akan mengurangi byte *payload* dari data yang akan dikirim. Sehingga pengiriman data melalui *tunnel* akan memakan waktu lebih lama. Besar penambahan byte *header* pada proses enkapsulasi PPTP lebih kecil daripada L2TP. PPTP mengurangi byte *payload* sebesar 16 bytes (4 bytes *header* PPP + 12 bytes *header* GRE), sedangkan L2TP mengurangi byte *payload* sebesar 18 bytes (4 bytes *header* PPP + 6 bytes *header* L2TP + 8 bytes *header* UDP)

2. Nilai *throughput* pada PPTP lebih besar daripada nilai *throughput* L2TP, dengan besar *bandwidth* tersedia sebesar 1024 Kbps, didapatkan PPTP mendapatkan nilai *throughput* 928. 5762 Kbps atau sebesar 90.68 % dari *bandwidth* tersedia sedangkan L2TP mendapatkan nilai *throughput* sebesar 850. 0761 Kbps atau sebesar 83.01 % dari besar *bandwidth* yang tersedia.

3. Pada saat trafik dengan protokol TCP dan UDP dilewatkan secara bersamaan pada PPTP dan L2TP *tunnel* didapati protokol UDP akan menggunakan hampir seluruh alokasi *bandwidth* yang tersedia, sedangkan protokol TCP hampir tidak mendapatkan *bandwidth* sama sekali, hal ini dikarenakan karakteristik dari masing-masing protokol TCP dan UDP itu sendiri. Oleh karena itu penggunaan *Class of Service* diperlukan, agar dapat memprioritaskan protokol tertentu mendapatkan alokasi *bandwidth* yang lebih besar dibandingkan dengan protokol yang lain.

4. Dari kedua simulasi diatas dapat disimpulkan jika network administrator menginginkan keadaan jaringan dengan nilai throughput yang besar dengan mengesampingkan point keamanan maka tunneling dengan protokol PPTP dapat digunakan, tetapi jika yang diprioritaskan adalah point keamanan maka tunneling dengan L2TP merupakan pilihan yang tepat karena L2TP dapat dikombinasikan dengan enkripsi IPSec (Layer 3) yang jauh lebih aman daripada enkripsi pada PPTP yaitu MPPE (Microsoft Point to Point Encryption). L2TP dengan IPSec menyediakan keamanan berlapis yang dapat menjamin keamanan data yang dilewatkan didalamnya. Kemudahan dalam setup dan keamanan yang dijanjikan membuat L2TP dengan IPSec merupakan pilihan untuk protokol VPN.

DAFTAR ACUAN

[1] "*Feature Summary Layer 2 Tunnel Protocol*", www.cisco.com, diakses Maret 2009. http://www.cisco.com/en/US/docs/ios/12_0t/12_0t1/feature/guide/l2tpT.html

[2] "*Introduction of Generic Routing Encapsulation*", www.cisco.com diakses Maret 2009. http://www.cisco.com/en/US/tech/tk827/tk369/tk287/tsd_technology_support_sub-protocol_home.html

[3] Training Module "*IP Tunneling and VPNs*", Cisco System, Copyright 2001

[4] Network Working Group, "*Generic Routing Encapsulation*", Request for Comments: 1701, diakses April 2009 http://www.faqs.org/rfcs/rfc1701.html

[5] Network Working Group, "*Point to Point Tunneling Protocol*", Request for Comments: 2637, diakses April 2009 http://www.faqs.org/rfcs/rfc2637.html

[6] Network Working Group, "*Layer Two Tunneling Protocol*", Request for Comments: 2661,diakses April 2009 http://www.faqs.org/rfcs/rfc2661.html

[7] Nassar. Daniel, *Network Performance Baselining* (MACMILLAN, 2000)

[8] Lewis. Chris, *Cisco TCP/IP Routing Professional Reference* (McGraw-Hill : 1999)

[9] Sheldon. Tom, *Encyclopedia of Networking and Telecommunications* (McGraw-Hill : 2001

- *Bandwidth* pada *network* ISP di set sebesar 1 Mbps (ISP A dan ISP B).
  - *Router* ISP A

```
interface FastEthernet0/0
 description connected to router ISP B
 bandwidth 1024
 ip address 192.168.1.17 255.255.255.248
 rate-limit input 1024000 128000 128000 conform-action transmit exceed-action
drop
 rate-limit output 1024000 128000 128000 conform-action transmit exceed-action
drop
 speed auto
 full-duplex
!
interface FastEthernet0/1
 description connected to router network A
 bandwidth 1024
 ip address 192.168.1.10 255.255.255.248
 rate-limit input 1024000 128000 128000 conform-action transmit exceed-action
drop
 rate-limit output 1024000 128000 128000 conform-action transmit exceed-action
drop
 speed 10
 full-duplex
```

Lampiran 2

- *Router* ISP B

```
interface FastEthernet0/0
 bandwidth 1024
 ip address 192.168.1.25 255.255.255.248
 rate-limit input 1024000 128000 128000 conform-action transmit exceed-action
drop
 rate-limit output 1024000 128000 128000 conform-action transmit exceed-action
drop
 duplex auto
 speed auto
!
interface FastEthernet0/1
 bandwidth 1024
 ip address 192.168.1.18 255.255.255.248
 rate-limit input 1024000 128000 128000 conform-action transmit exceed
action drop
 rate-limit output 1024000 128000 128000 conform-action transmit exceed-action
drop
 duplex auto
 speed auto
```

- ***Set-up* PPTP *Server* pada mikrotik:**

1. *Set-up client* pada PPTP *Server*:

```
[admin@MikroTik] > ppp secret
[admin@MikroTik] ppp secret> add name=user1 service=pptp
password=user1local-address=20.20.20.1 remote-address=20.20.20.2
[admin@MikroTik] ppp secret> print
Flags: X - disabled
 # NAME  SERVICE  CALLER-ID  PASSWORD  PROFILE   REMOTE-A..
 0  user1  pptp               user1     default   20.20.20.2
```

2.. *Add user* pada PPTP *Server*-list:

```
[admin@MikroTik] > interface pptp-server
[admin@MikroTik] interface pptp-server> add user=user1
[admin@MikroTik] interface pptp-server> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME      USER      MTU      CLIENT-AD... UPTIME   ENCODING
 0    pptp-in1  user1
```

Lampiran 3

3. *Enabling* PPTP *Server*:

```
[admin@MikroTik] interface pptp-server> server
[admin@MikroTik] interface pptp-server server> set enabled=yes
[admin@MikroTik] interface pptp-server server> print
 enabled: yes
 max-mtu: 1500
 max-mru: 1500
 authentication: mschap1,mschap2
 keepalive-timeout: 30
 default-profile: default-encryption
```

- ***Set-up* L2TP *Server* pada mikrotik:**
  1. *Set-up client* pada L2TP *Server*:

```
[admin@MikroTik] > ppp secret
[admin@MikroTik] ppp secret> add name=acul service=l2tp password=maracul
local-address=10.10.10.1 remote-address=10.10.10.2
[admin@MikroTik] ppp secret> print
Flags: X - disabled
 #   NAME  SERVICE  CALLER-ID  PASSWORD PROFILE   REMOTE-A...
 0   acul          l2tp                          maracul         default        10.10.10.2
```

  2. *Add user* pada L2TP *Server*-list:

```
[admin@MikroTik] > interface l2tp-server
[admin@MikroTik] interface l2tp-server> add user=acul
[admin@MikroTik] interface l2tp-server> print
Flags: X - disabled, D - dynamic, R - running
#     NAME      USER         MTU        CLIENT-AD... UPTIME   ENCODING
0     l2tp-in1      acul
```

  3. Enabling L2TP *Server*:

```
[admin@MikroTik] interface l2tp-server> server
[admin@MikroTik] interface l2tp-server server> set enabled=yes
[admin@MikroTik] interface l2tp-server server> print
 enabled: yes
 max-mtu: 1500
 max-mru: 1500
 authentication: pap,chap,mschap1,mschap2
 default-profile: default-encryption
```

Lampiran 4

- **Set-up VPN *Client*:**

  Contol Panel – Network Connection – Create a New Connection

Gambar Tampilan awal Create New Connection

Klik *Next*, lalu pilih seperti gambar dibawah, klik *next*

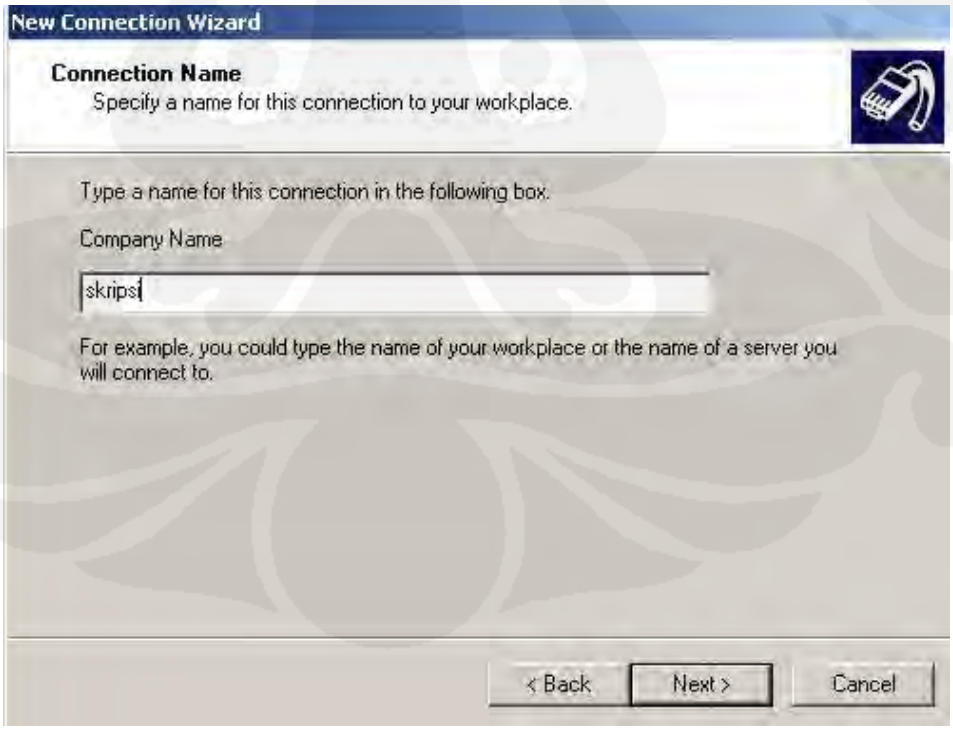
Gambar Tampilan Memilih Network Connection Type

Lampiran 5

Lalu pilih menu *connect* to the *network using* VPN, klik *next*

Gambar Tampilan Memilih VPN Connection

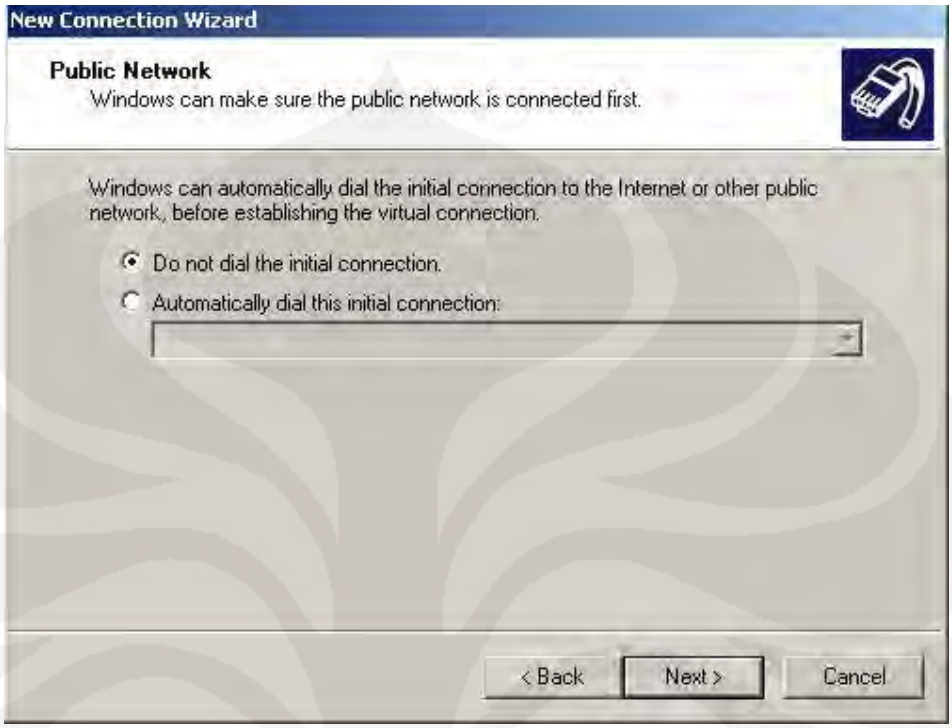
Masukan nama untuk koneksi ini, lalu klik *next*



Gambar Tampilan Memasukan Nama Koneksi

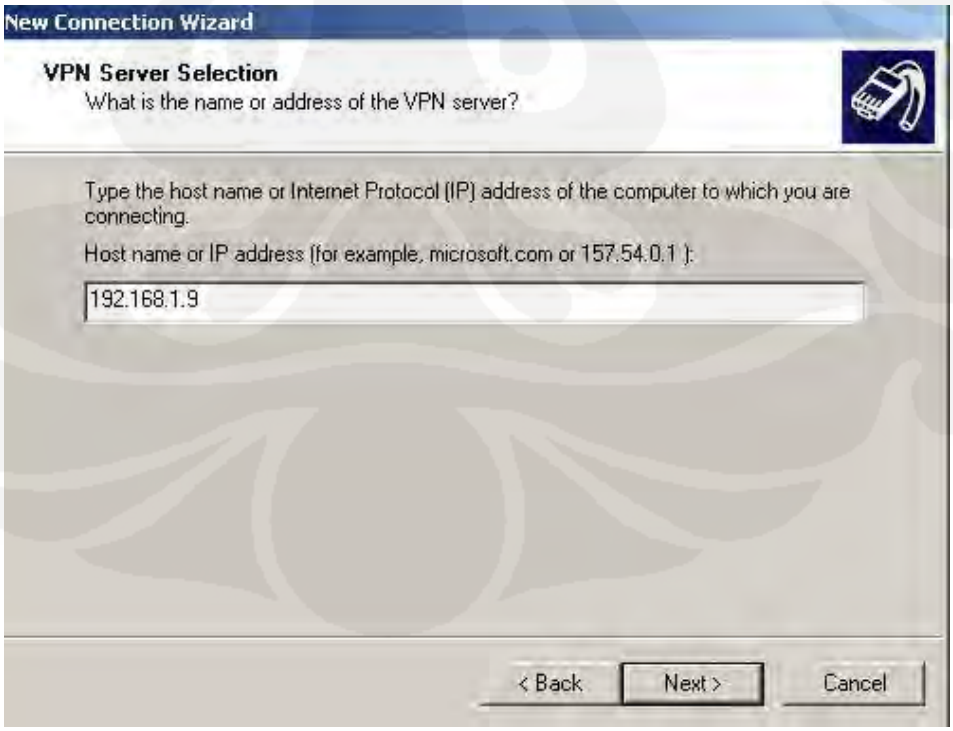
Lampiran 6

Kemudian pilih menu seperti gambar dibawah, klik *next*:



Gambar Tampilan Melakukan Inisialisasi Koneksi

Masukkan alamat IP computer yang akan dituju (IP mikrotik), klik *next*:



Gambar Tampilan Memasukkan alamat IP VPN Server

Lampiran 7

Create New Connection Selesai:

Gambar Tampilan Completing New Connection

- **Set-up PPTP *Client***

  Pada *connection dialog box* (skripsi2) klik *Properties – Networking*, Kemudian pilih PPTP VPN.

Gambar Tampilan Dialog-Box untuk Memilih PPTP VPN

Lampiran 8

Dial PPTP VPN dilakukan dengan memasukan *user* dan *password:*

Gambar Tampilan Melakukan Dial PPTP VPN

- ***Set-up* L2TP *Client*:**

  Untuk melakukan koneksi L2TP (tanpa IP Sec) nilai dari ProhibitIpSec harus diganti menjadi "1"

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rasman\Parameters]
"ProhibitIpSec"=dword:00000001
```

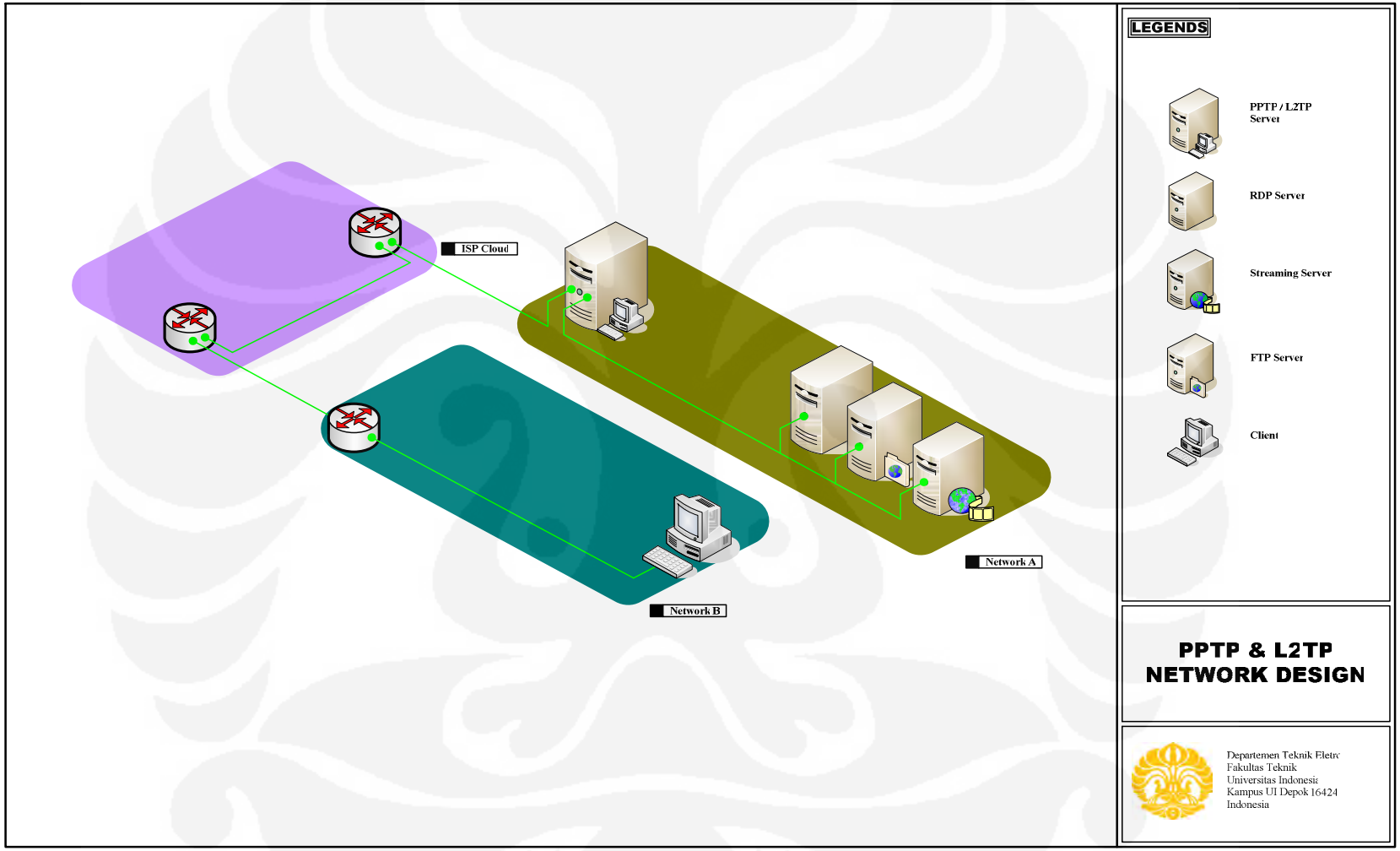
Lampiran 9

Pada *connection dialog box* (skripsi2) klik *Properties – Networking,* Kemudian pilih L2TP IPSec VPN.

Gambar Tampilan Dialog-Box untuk Memilih L2TP VPN

Dial L2TP VPN dilakukan dengan memasukan *user* dan *password*:

Gambar Tampilan Melakukan Dial L2TP VPN


ISP Cloud
Network A
Network B
LEGENDS
PPTP / L2TP Server
RDP Server
Streaming Server
FTP Server
Client
PPTP & L2TP NETWORK DESIGN
Departemen Teknik Eletro
Fakultas Teknik
Universitas Indonesia
Kampus UI Depok 16424
Indonesia