

# PERANCANGAN FTP (FILE TRANSFER PROTOCOL) MELALUI SCTP (STREAM CONTROL TRANSMISION PROTOCOL) MENGGUNAKAN SOCKET PROGRAMMING

I Wayan Ade Sapura, Cokorda Rai Adi Pramatha, S.T., M.MSI  
Program Studi Teknik Informatika,  
Jurusan Ilmu Komputer,  
Fakultas Matematika Dan Ilmu Pengetahuan Alam,  
Universitas Udayana

Email: [ade.saputra@cs.unud.ac.id](mailto:ade.saputra@cs.unud.ac.id)

## ABSTRAK

*File Transfer Protocol (FTP)*. *File Transfer Protocol* menjadi banyak digunakan karena kemudahan penggunaannya untuk proses pertukaran data. Namun seringkali terjadi permasalahan pada proses *transfer* data yang dilakukan. Upaya untuk mengatasi hal tersebut untuk menyempurnakan proses pengiriman data dengan menggunakan teknologi baru yang lebih mendukung untuk menjamin proses *transfer* data dengan *Quality of Service (QoS)* seperti misalnya *Integrated Service Architecture (ISA)* atau *Diffrentiated Service Architecture (DSA)* namun keduanya sulit untuk dilakukan karena memerlukan perubahan arsitektur jaringan secara mendasar. Salah satu solusi untuk mendukung kebutuhan kualitas layanan yaitu dengan menggunakan suatu protokol baru yang dapat diterapkan tanpa mengubah bentuk arsitektur jaringan.

*Stream Control Transmision Protocol (SCTP)* adalah protokol baru yang dapat diandalkan dan memiliki banyak kelebihan dibanding *Transmission Transfer Protocol (TCP)*. SCTP bersifat *message-oriented* yang handal SCTP juga menyimpan batas-batas pesan dan disaat yang sama mampu mendeteksi kehilangan data, duplikasi data, dan out of order data. SCTP juga memiliki mekanisme kontrol aliran data. Dengan penggunaan protokol SCTP pada FTP diharapkan *delay* dan kerusakan data dapat diminimalisir karena SCTP memiliki kinerja dan kehandalan yang lebih di banding dengan penggunaan TCP.

Kata Kunci : FTP, SCTP, File Transfer, Transfer Protocol.

## ABSTRACT

File Transfer Protocol (FTP). File Transfer Protocol is becoming widely used because of the ease of use for the data exchange process. But often the problem occurs in the process of data transfer is done as the data sent later, damaged not even get to the destination due to an interruption occurs during the process of data transmission.

Attempts to overcome this to improve the data delivery process by using new technology to ensure better support data transfer process with Quality of Service (QoS) such as Integrated Services Architecture (ISA) or Diffrentiated Service Architecture (DSA), but both are difficult to do because requires a fundamental change in the network architecture. One solution to support quality of service requirements by using a new protocol that can be implemented without changing the architecture of the network.

Stream Control Transmission Protocol (SCTP) is a new protocol that is reliable and has a stout advantage over Transmission Transfer Protocol (TCP). SCTP is a reliable message-oriented SCTP also stores the message boundaries and at the same time is able to detect data loss, data duplication, and out of order data. SCTP also has a flow control mechanism. With the use of the FTP protocol SCTP is expected to delay and data loss can be minimized because SCTP has better performance and reliability compared to the use of TCP.

Keywords : FTP, SCTP, File Transfer, Transfer Protocol.

## 1. PENDAHULUAN

Seiring dengan perkembangan *teknologi informasi* saat ini, pertukaran data melalui *internet* bukanlah suatu yang baru lagi. Hal ini tidak terlepas dari peranan media *Internet* yang menyediakan fasilitas untuk melakukan pertukaran data melalui *File Transfer Protocol* (FTP). *File Transfer Protocol* menjadi banyak digunakan karena kemudahan penggunaannya untuk proses pertukaran data.

*File Transfer Protocol* (FTP) tidak hanya digunakan sebagai media pertukaran data antar komputer, tetapi juga dapat digunakan untuk pertukaran data antara *server* dengan *client*. Namun seringkali terjadi permasalahan pada proses *transfer* data yang dilakukan seperti data yang dikirim lambat, rusak bahkan tidak sampai ke tujuan karena terjadi gangguan pada saat proses pengiriman data.

Upaya untuk mengatasi hal tersebut untuk menyempurnakan proses pengiriman data dengan menggunakan teknologi baru yang lebih mendukung untuk menjamin proses *transfer* data dengan *Quality of Service* (QoS) seperti misalnya *Integrated Service Architecture* (ISA) atau *Diffrentiated Service Architecture* (DSA) namun keduanya sulit untuk dilakukan karena memerlukan perubahan arsitektur jaringan secara mendasar. Salah satu solusi untuk mendukung kebutuhan kualitas layanan yaitu dengan menggunakan suatu protokol baru yang dapat diterapkan tanpa mengubah bentuk

arsitektur jaringan. (Budiardjo dan Thiotrisno, 2003)

Dalam *Transfer file, Protocol Transport* merupakan bagian yang sangat penting. *Transport layer* didisain untuk komunikasi terminal diantara aplikasi yang berjalan pada *host-host* yang berbeda (Seo dkk, 2006). *Transmission Control Protocol* (TCP) merupakan protokol yang kebanyakan dipakai di *layer transport*. TCP sangat reliable sehingga sampai saat ini menjadi pilihan utama.(Geoff dan Telstra 2007).

Namun seiring berkembangnya layanan internet seperti multimedia TCP dirasakan masih kurang dalam memenuhi layanan yang ada. TCP memiliki beberapa kelemahan yang sangat mendasar yaitu *packet latency* yang cenderung panjang, *head of line blocking* (HOL), *denial of service* yang disebabkan oleh mekanisme *three-way-handshake* dan pengurutan informasi yang ketat (Budiardjo dan Thiotrisno, 2003). Ketika jaringan padat yang otomatis *kongesti* sangat tinggi menyebabkan *time-out* dan akan mengirimkan *retransmisi* karena sifatnya yang *conection oriented*. (Geoff dan Telstra 2007).

. Hal ini akan menyebabkan *delay* yang tinggi dan berakibat turunnya *throughput*. Maka mulailah dikembangkan protokol lain di layer transport seperti *Stream Control Transmission Protocol* (SCTP) yang lebih dapat diandalkan. *Stream Control Transmission Protocol* (SCTP) adalah protokol baru yang dapat diandalkan dan memiliki banyak kelebihan dibanding

*Transmission Transfer Protocol* (TCP). SCTP bersifat *message-oriented* yang handal SCTP juga menyimpan batas-batas pesan dan disaat yang sama mampu mendeteksi kehilangan data, duplikasi data, dan out of order data. SCTP juga memiliki mekanisme kontrol aliran data (Alwi dan Syawi, 2009).

Dengan penggunaan protokol SCTP pada FTP diharapkan *delay* dan kerusakan data dapat diminimalisir karena SCTP memiliki kinerja dan kehandalan yang lebih di banding dengan penggunaan TCP.

## 2. Tinjauan Pustaka

### 2.1 STREAM CONTROL TRANSMISION PROTOCOL (SCTP)

*Stream Control Transmission Protocol* (SCTP) adalah suatu hal baru yang dapat diandalkan. SCTP kebanyakan dirancang untuk aplikasi *Internet* yang baru-baru ini telah diperkenalkan. Aplikasi baru ini, seperti IUA (ISDN over IP), M2UA dan M3UA (*telephony signaling*), H.248 (*media gateway control*), H.323 (*IP telephony*), and SIP (*IP telephony*), membutuhkan pelayanan yang lebih canggih daripada yang disediakan oleh TCP. SCTP ini memberikan peningkatan kinerja dan kehandalan.

SCTP menggabungkan fitur terbaik dari UDP dan TCP. SCTP adalah protokol *message-oriented* yang handal. SCTP menyimpan batas-batas pesan dan pada saat yang sama mendeteksi kehilangan data, duplikasi data, dan *out-of-order*

data. SCTP juga memiliki kontrol *kongesti* dan mekanisme kontrol aliran.

### 2.2 Process-to-Process Communication

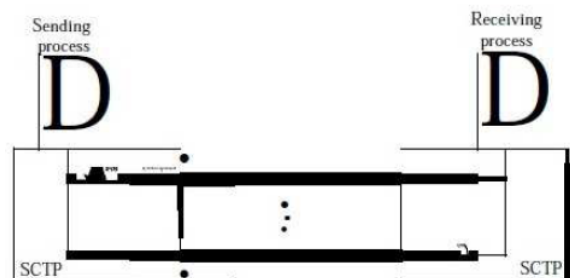
SCTP menggunakan semua *port* yang terkenal di ruang TCP. Pada tabel berikut merupakan daftar beberapa port tambahan nomor yang digunakan oleh SCTP.

Tabel 2.1 port SCTP

<i>Protocol</i>	<i>Port Number</i>	<i>Description</i>
IUA	9990	ISDN overIP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony

### 2.2 Multiple Streams

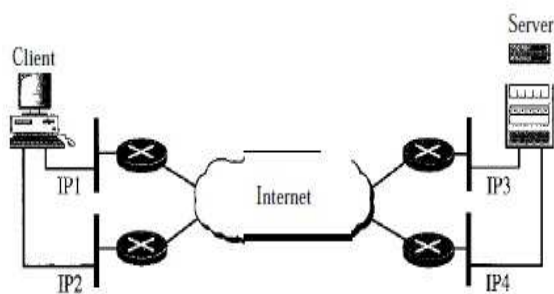
Di TCP setiap koneksi antara TCP *client* dan TCP *server* melibatkan satu aliran. Masalahnya dengan pendekatan ini adalah kerugian pada setiap titik di blok stream pengiriman sisa data. Hal ini bisa diterima ketika kita sedang memindahkan teks, bukan seperti pada saat kita mengirim data *real-time* seperti *audio* atau *video*. SCTP memungkinkan layanan *multistream* di setiap koneksi yang disebut asosiasi dalam terminologi SCTP. Jika salah satu aliran diblokir, maka aliran yang lain masih bisa mengirim datanya .



Gambar 2.1 *Multiple Stream*

### 2.3 Multihoming

Sebuah asosiasi SCTP, di sisi lain, mendukung layanan *multihoming*. *Host* mengirim dan menerima data sekaligus dapat menentukan beberapa alamat IP pada setiap akhir untuk asosiasi. Dalam pendekatan *fault-tolerant*, ketika salah satu jalur gagal, antarmuka lain dapat digunakan untuk pengiriman data tanpa gangguan. Fitur *fault-tolerant* ini sangat membantu ketika kita mengirim dan menerima muatan *real-time* seperti *Internet telephony*



Gambar 2.2 *Multihoming*

Dalam Gambar *Multihoming* konsep, *client* terhubung ke dua jaringan lokal dengan dua alamat IP. Server ini juga terhubung ke dua jaringan dengan dua alamat IP. *Client* dan *server* dapat membuat asosiasi, menggunakan empat pasang alamat IP yang berbeda. Namun, perlu diketahui bahwa dalam implementasi saat ini SCTP hanya sepasang alamat IF dapat dipilih untuk komunikasi normal, alternatif digunakan jika pilihan utama gagal. Dengan kata lain, pada saat ini, SCTP tidak mengizinkan berbagi beban antara jalur yang berbeda.

### 2.4 Full Duplex Communication

SCTP menawarkan layanan *full-duplex*, di mana data dapat mengalir dalam dua arah

pada saat yang sama. Setiap SCTP kemudian memiliki *buffer* mengirim dan menerima, dan paket dikirim di kedua arah.

### 2.5 Connection-Oriented Service

SCTP merupakan protokol berorientasi koneksi. Namun dalam SCTP, koneksi ini disebut asosiasi. Proses ketika suatu proses pada situs A ingin mengirim dan menerima data dari proses lain di situs B.

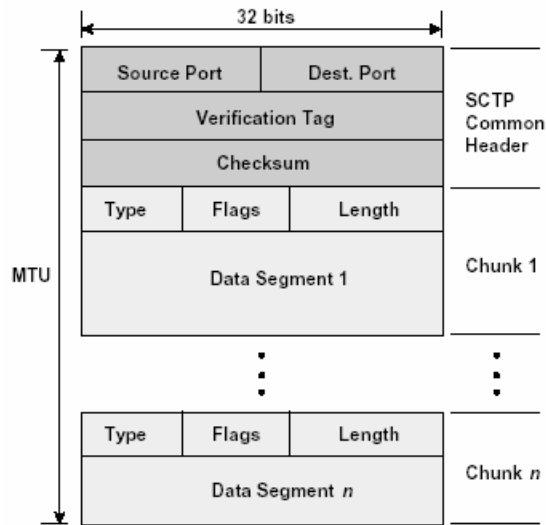
1. Kedua SCTP membangun hubungan antara satu sama lain.
2. Data dipertukarkan dalam kedua arah.
3. Asosiasi dihentikan.

### 2.6 Reliable Service

Seperti halnya TCP, SCTP merupakan protokol *transport* yang handal. SCTP menggunakan mekanisme pengakuan untuk memeriksa data suara itu aman.

### 2.7 Paket data

Dalam TCP, segmen membawa data dan informasi kontrol. Data dicatat sebagai kumpulan *byte*. Kontrol informasi didefinisikan oleh enam bendera kontrol pada *header*. Desain SCTP sama sekali berbeda, data dicatat sebagai potongan data, kontrol. Beberapa potongan kontrol dan potongan data dapat dikemas bersama dalam sebuah paket. Sebuah paket dalam SCTP memainkan peran yang sama sebagai *segmen* dalam TCP.



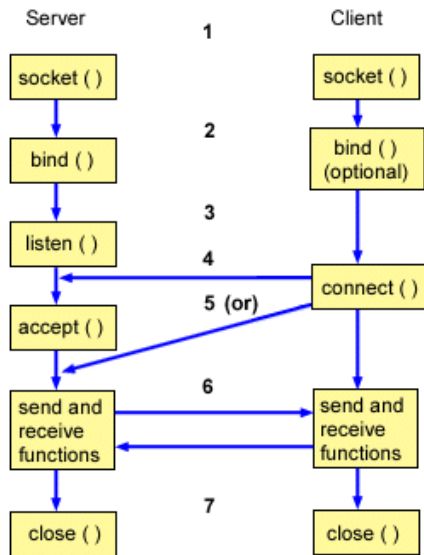
Gambar 2.3 Paket data SCTP

Setiap paket SCTP terdiri dari SCTP common header dan beberapa blok chunk. Ukuran paket SCTP adalah sebesar 1 MTU (Maximum Transmission Unit) sepanjang 1500 bytes. Common header mendukung proses validasi dan koneksi pada SCTP sedangkan chunk merupakan unit informasi pada paket SCTP yang dapat berupa chunk data atau chunk kontrol. Common header SCTP berukuran 12 bytes, yang terdiri dari: alamat port sumber dan port tujuan masing-masing sebesar 16 bit, tag verifikasi (32 bit) dan checksum (32 bit). Alamat port sumber (source) menyatakan identifikasi asal paket, sedangkan alamat port tujuan adalah alamat penerima/tujuan paket SCTP. Setiap endpoint menggunakan tag verifikasi untuk mengidentifikasi asosiasi dimana penerima melakukan validasi terhadap pengirim paket SCTP. Istilah asosiasi menggambarkan koneksi antara dua endpoints. Checksum digunakan untuk menjaga data integrity dan menghindari duplikasi. Setiap chunk SCTP berukuran 8 bytes

yang terdiri dari Field Type, Flags, Length dan User Data. Field Type digunakan untuk membedakan antara tipe chunk data atau chunk kontrol. Terdapat 13 macam chunk yang digunakan sebagai standar chunk SCTP. Field Flag digunakan untuk menentukan bit mana yang akan digunakan dalam koneksi. Field Length digunakan untuk menentukan panjang chunk. Field data segment mengandung payload data setiap chunk. Sebagai ganti dari three-way-handshake pada koneksi TCP, SCTP menggunakan four-way-handshake untuk menginisialisasi suatu asosiasi.

## 2.5 Aliran fungsi Soket

*Socket* memiliki ciri khas aliran, pada model *connection-oriented client-to-server*, *socket* pada server proses menunggu *request* dari *client*. Untuk melakukan hal ini, server pertama-tama menetapkan (*binds*) sebuah alamat yang dapat digunakan oleh *client* untuk dapat terhubung dengan server. Saat alamat telah ditetapkan *server* kemudian menunggu *request-service* dari *client*. pertukaran data dari server ke *client* terjadi saat *client* terhubung dengan server melalui *socket*. Server menanggapi request dari *client* dan mengirimkan balasan kepada *client*.



Gambar 2.4 Aliran fungsi Soket

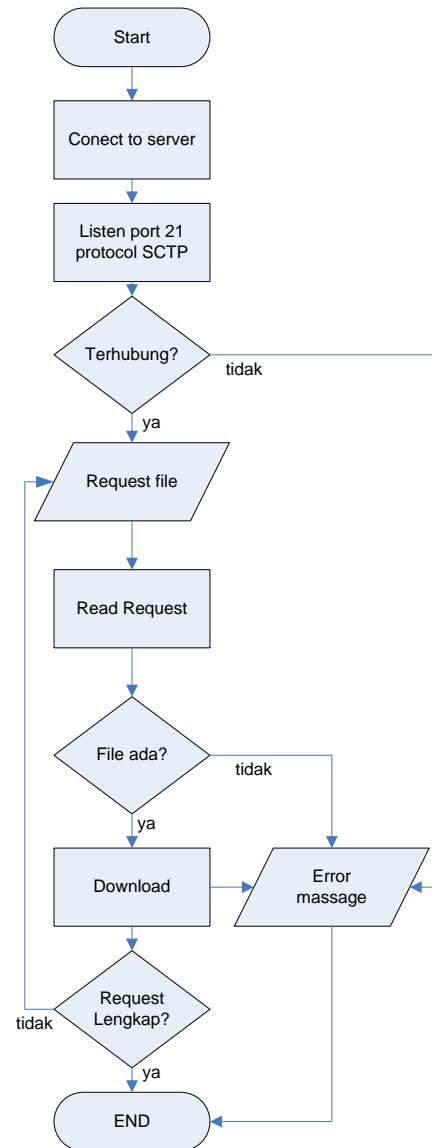
### 3. IMPLEMENTASI SISTEM

#### 3.1 Perancangan sistem

Dalam tahap perancangan system dilakukan pencarian dan pengumpulan informasi yang dibutuhkan untuk perancangan system File Transfer Transmission Protocol. (FTP).

Metode yang digunakan untuk mengumpulkan informasi dan data adalah metode studi literatur, yaitu dengan mempelajari literatur yang terkait dengan penelitian, antara lain mengenai Stream Control Transmission protocol dan penerapannya pada sistem File Transfer Protocol (FTP).

Pengiriman file pada FTP berjalan pada protokol SCTP. Proses transfer data pada ftp akan ditampilkan pada gambar 3.1



Gambar 3.1 Flowchart Download

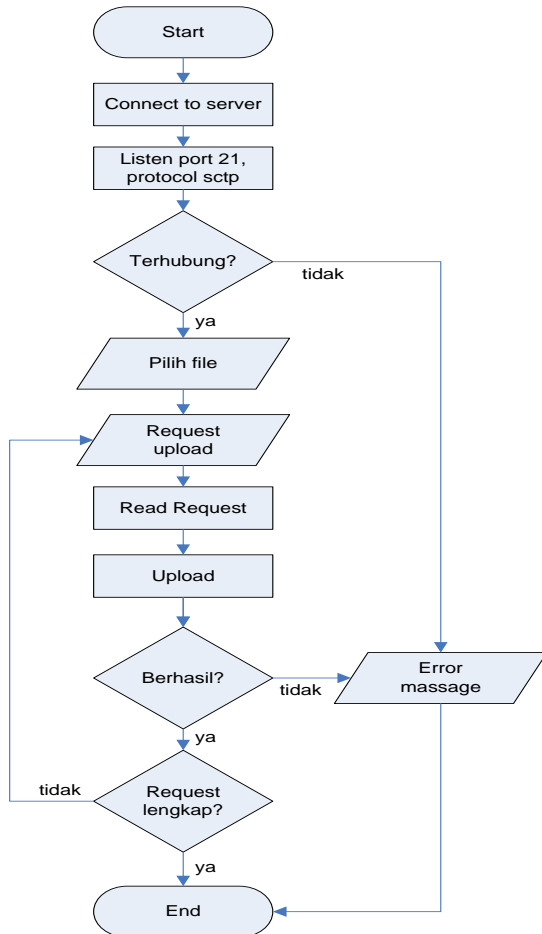
Penjelasan flowchart download :

1. *Client* melakukan *request* ke FTP
2. Kemudian *System* FTP akan menerima dan memeriksa *request* yang ada.
3. Selanjutnya FTP akan mencari file yang diminta berdasarkan alamat yang diberikan oleh *client*.
4. Jika alamat file yang diberikan oleh *client* valid maka proses download akan segera dilakukan. Namun jika file yang diminta tidak ada maka akan di

tampilkan error message bahwa file yang sedang dicari tidak ditemukan.

5. Jika *client* masih ingin meminta *file*, maka proses akan diulangi lagi pada proses *request file*.

Untuk menggambarkan proses upload pada FTP dapat ditunjukkan pada Flowchart Upload



Gambar 3.2 Flowchart Upload

Penjelasan :

- *Client* akan memilih *file* yang akan di *upload* di FTP.
- Kemudian *client* akan melakukan *request* ke FTP.
- setelah request diterima oleh FTP maka file akan langsung di upload.
- Jika upload gagal maka akan ditampilkan *error messages* dan jika berhasil maka proses selesai.

### 3.2 Perancangan Socket

Pada perancangan socket File Transfer Protokol untuk dapat menggunakan Stream Control Transimision Protocol (SCTP) pada File Transfer Transmision Protocol (FTP) maka pada sisi server dan sisi client harus dilakukan konfigurasi proses socket. Diantaranya konfigurasi pengalamatan (binding), konfigurasi port yang akan digunakan oleh system untuk melakukan pengiriman file dan pengaturan jenis protocol yang akan digunakan oleh system.

Berikut potongan code program dari rancangan File Transfer Protocol (FTP) yang berjalan pada Stream Control Transmision Protocol (SCTP).

#### a. Pembuatan Socket

Proses ini meliputi pengsturan jenis socket dan pengaturan protocol yang akan digunakan.

```

if ((sck = socket(AF_INET, SOCK_STREAM, IPPROTO_SCTP)) < 0) {
    perror("socket error()");
    exit(1); }
  
```

#### b. Binding

Dalam proses ini diatur proses pengalamatan dan penentuan port yang akan digunakan oleh sistem.

```

#define SERVICE_PORT 3482

/* pengaturan server address dan port */
memset(&serv_adr, 0, sizeof(serv_adr));

serv_adr.sin_family = AF_INET;

serv_adr.sin_addr.s_addr = htonl
(INADDR_ANY);

serv_adr.sin_port =
htons(SERVICE_PORT);

/* Binding */

if (bind(sck, (struct sockaddr *)&serv_adr,
sizeof(serv_adr)) < 0) {

perror("bind error");

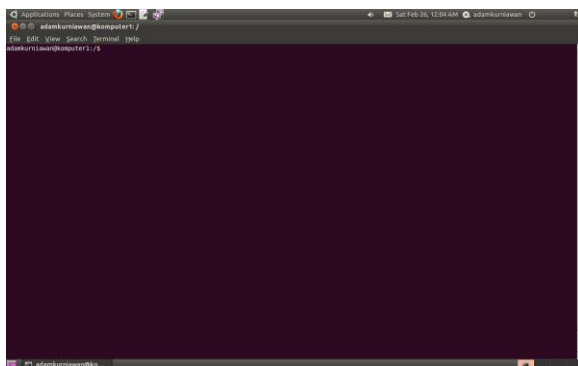
exit(3);

```

#### 4. HASIL DAN PEMBAHASAN

File transfer Protocol (FTP) ini dapat berjalan pada sistem operasi linux dengan antarmuka berupa Command Line Interface (CLI) dengan perintah-perintah umum yang digunakan pada File Transfer Protocol (FTP).

Berikut ini adalah tampilan awal Sistem berikut ini merupakan tampilan dari antarmuka File Transfer Protocol (FTP).



Gambar 4.1 Tampilan CLI

Pengguna hanya perlu mengetikkan perintah diikuti oleh nama file perintah FTP untuk melakukan download atau upload.

Jika terdapat kesalahan dalam pengetikkan perintah atau nama file maka sistem akan menampilkan pesan error.

#### 5. KESIMPULAN

Dari pembahasan yang sudah dilakukan untuk perancangan File Transfer Protocol (FTP) melalui Stream Control Transmission Protocol menggunakan Socket Programming maka dapat ditarik kesimpulan sebagai berikut :

1. Dengan menggunakan Protokol SCTP pada FTP dapat memperbaiki Quality Of Service (QOS) jaringan tanpa merubah struktur fisik jaringan yang ada.
2. Penggunaan SCTP pada File Transfer Protocol (FTP) mampu memaksimalkan penggunaan bandwidth untuk melakukan transfer file.
3. Penggunaan Protokol SCTP pada FTP untuk transfer file membutuhkan waktu yang lebih singkat dibanding penggunaan protokol TCP.



## 6. DAFTAR PUSTAKA

1. Alwi, E.I., Syawie, Ibrahim. 2010. *SCTP (Stream Control Transmission Protocol)*. Yogyakarta.
2. Anonim. 2010. *How sockets work*. [website], <http://publib.boulder.ibm.com/infocente r/series/v5r4/index.jsp?topic=%2Frzab6%2Fhowdosockets.htm> Diakses tanggal 16 April 2012.
3. Amer, Sourabh., Amer, P.D. 2007. *Improving Multiple File Transfers Using SCTP Multistreaming*. Delaware.
4. Amer, Sourabh., Amer, P.D. 2008. *Improving File Transfers in FCS Network*. Delaware
5. Budiardjo, Bagio., Thiotrisno, Michael. 2003. *Protocol (SCTP) dan Transmission Control Protocol (TCP) Pada Jaringan Unicast*. Depok.
6. Davis, Keir., Turner, J.W., Yowcom, Nathan. 2004 *The Definitive Guide to Linux Network Programming*. New York: Appres.
7. Forouzan, A. (2007). *Data Communications And Networking 4th Edition*. New York: McGraw-Hill.
8. Huston, Geoff., Telstra. *Future for TCP*. [http://www.cisco.com/web/about/ac123/ac147/ac174/ac195/about\\_cisco\\_ipj\\_archive\\_article09186a00800c83f8.html](http://www.cisco.com/web/about/ac123/ac147/ac174/ac195/about_cisco_ipj_archive_article09186a00800c83f8.html) Diakses tanggal 28 Februari 2012.
9. Ong, L., Youakum, J. *An Introduction to the Stream Control Transmission Protocol* (SCTP). <http://grenache.tools.ietf.org/html/rfc3286> Diakses 28 Februari 2012.
10. Seu, Jinwook., Ahn, Youngju., No, Minki., Ahn, Seongjin., Chung, Jinwook. 2006 *Implement of FTP application using SCTP*. South Korea

