

SCTP (Stream Control Transmission Protocol)

Erick Irawadi Alwi, 10/308994/PTK/07037- Magister Teknologi Informasi
Ibrahim Syawie, 10/308860/PTK/07020- Magister Teknologi Informasi
Jurusan Pascasarjana Teknik Elektro
Fakultas Teknik UGM
Yogyakarta

1.1 SCTP

Stream Control Transmission Protocol (SCTP) adalah suatu hal baru yang dapat diandalkan, *message-oriented* protokol transport layer. 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. Kita coba membandingkan UDP, TCP, dan SCTP:

• UDP

UDP adalah protokol *message-oriented*. Sebuah proses memberikan pesan kepada UDP, yang dienkapsulasi dalam datagram pengguna dan dikirim melalui jaringan. UDP menghemat batas pesan; setiap pesan tidak tergantung pada setiap pesan lainnya. UDP dibutuhkan ketika kita berhadapan dengan aplikasi seperti telepon IP dan transmisi data real-time, seperti yang akan kita lihat nanti dalam teks. Namun, UDP tidak dapat diandalkan karena pengirim tidak dapat mengetahui keadaan pesan terkirim. Sebuah pesan bisa hilang, diduplikasi, atau diterima dalam keadaan rusak. UDP juga tidak memiliki beberapa fitur lainnya, seperti kontrol kongesti dan *flow control*, diperlukan untuk sebuah protokol transport layer yang *friendly*.

• TCP

TCP merupakan protokol berorientasi *byte*, yaitu menerima sebuah pesan dari suatu proses, pengiriman pesan itu sebagai aliran *byte*, dan mengirimkannya dalam segmen. Tidak ada penjagaan batas-batas pesan. Namun, TCP merupakan protokol yang handal. penduplikatan segmen terdeteksi, segmen yang hilang akan dikirim ulang, dan *byte* yang dikirim ke akhir proses secara berurutan. TCP juga memiliki mekanisme kontrol kongesti dan *flow control*.

• SCTP

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.

1.2 LAYANAN SCTP

Layanan yang ditawarkan oleh SCTP dengan proses aplikasi lapisan terbagi atas 3, yaitu :

1.2.1 Process-to-Process Communication

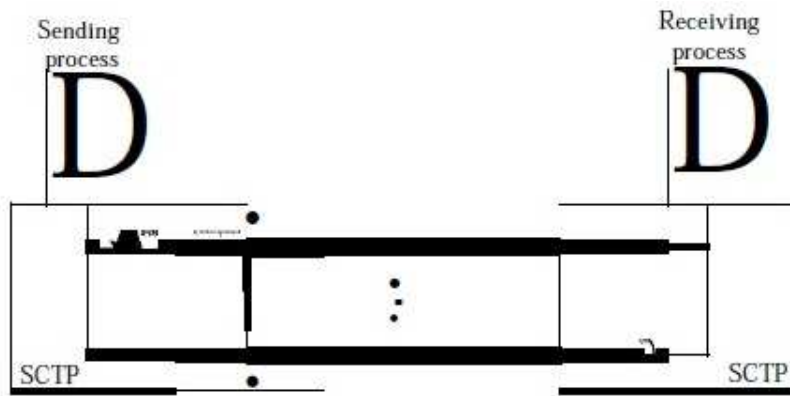
SCTP menggunakan semua port yang terkenal di ruang TCP. Tabel 23.4 daftar beberapa port tambahan nomor yang digunakan oleh SCTP.

Tabel 23.4 Beberapa aplikasi SCTP

<i>Protocol</i>	<i>Port Number</i>	<i>Description</i>
IVA	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

1.2.2 Multiple Streams

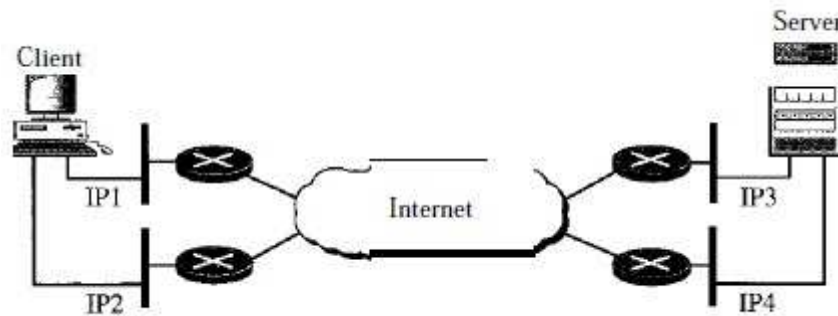
Di TCP setiap koneksi antara TCP klien dan TCP server melibatkan satu aliran. Masalahnya dengan pendekatan ini adalah kerugian pada setiap titik di blok *stream* pengiriman sisa data. Hal ini bisa diterima ketika kita sedang memindahkan teks; bukan seperti pada saat kita mengirim data *real-time* seperti 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 . Hal ini di ibaratkan seperti beberapa jalur di jalan raya. Setiap jalur dapat digunakan untuk berbagai jenis lintasan. Misalnya, satu jalur bisa digunakan untuk lintasan biasa, satu lagi untuk lintasan khusus mobil. Jika lintasan yang diblokir untuk kendaraan biasa, maka lintasan khusus kendaraan mobil masih bisa mencapai tujuan mereka.



Gambar 23.27 konsep Multiple-stream

1.2.3 Multihoming

Sebuah koneksi TCP melibatkan salah satu sumber dan satu tujuan alamat IP. Hal ini berarti jika pengirim atau penerima merupakan *host multihomed* (terhubung ke lebih dari satu alamat fisik dengan beberapa alamat IP), hanya satu dari alamat-alamat IP untuk setiap akhir dapat digunakan selama masih terhubung. Sebuah asosiasi SCTP, di sisi lain, mendukung *multihoming* layanan. *Host* mengirim dan menerima 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 23.28 konsep Multihoming

Dalam Gambar *Multihoming* konsep, klien terhubung ke dua jaringan lokal dengan dua alamat IP. *Server* ini juga terhubung ke dua jaringan dengan dua alamat IP. Klien 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.

1.2.4 Full-Duplex Communication

Seperti TCP, 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.

1.2.5 Connection-Oriented Service

Seperti TCP, 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

1.2.6 Reliable Service

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

1.3 Fitur SCTP

Fitur umum SCTP dan membandingkannya dengan TCP.

1.3.1 Transmission Sequence Number

Transfer data dalam TCP dikendalikan oleh penomoran *byte* dengan menggunakan nomor urut. Di sisi lain, unit data dalam SCTP adalah serangkaian data yang banyak dan mungkin tidak memiliki hubungan satu sama lain dengan pesan yang berasal dari proses karena fragmentasi, Transfer data dalam SCTP dikendalikan oleh penomoran potongan-potongan data. SCTP menggunakan nomor urutan transmisi (TSN) ke nomor potongan data tersebut. Dengan kata lain, TSN dalam SCTP memainkan peran analog ke nomor urutan TCP. TSN mempunyai panjang 32 bit dan diinisialisasi secara acak antara 0 dan 232 - 1. Setiap potongan data harus membawa TSN terkait dalam header-nya.

1.3.2 Stream Identifier

Di TCP, hanya ada satu aliran di masing-masing sambungan. Dalam SCTP, mungkin ada beberapa aliran dalam setiap kelompok. Setiap aliran dalam SCTP perlu diidentifikasi dengan menggunakan *stream identifier* (SI). Setiap potongan data harus membawa SI dalam *header* sehingga ketika tiba di tempat tujuan, dapat dipasang dengan benar pada alirannya. Angka 51 adalah angka 16-bit mulai dari 0.

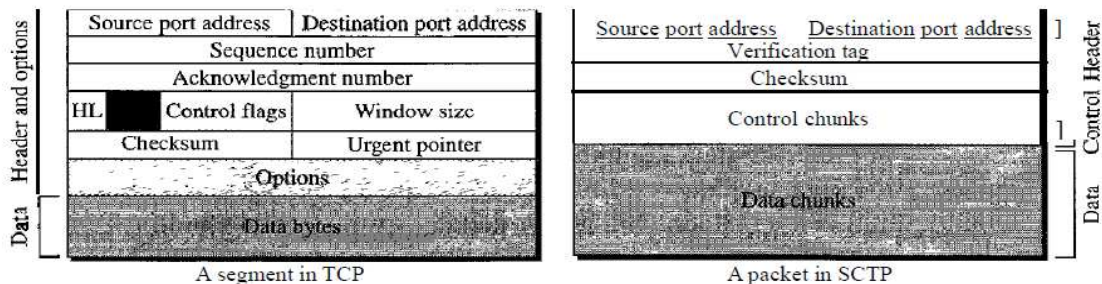
1.3.3 Stream Sequence Number

Ketika sepotong data yang tiba di tempat tujuan SCTP, potongan data itu disampaikan ke aliran yang tepat dan dalam urutan yang tepat. Ini berarti bahwa, di samping sebuah SI, SCTP mendefinisikan setiap potongan data di setiap aliran data dengan nomor urutan aliran (SSN).

1.3.4 Paket

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 informasi dicatat sebagai potongan 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 dibawah ini membandingkan segmen dalam TCP dan paket dalam SCTP. Daftar singkat perbedaan antara sebuah paket SCTP dan segmen TCP:

1. Informasi kontrol dalam TCP merupakan bagian dari *header*, informasi kontrol dalam SCTP dimasukkan dalam potongan kontrol. Ada beberapa jenis potongan kontrol, masing-masing digunakan untuk tujuan yang berbeda dalam data segmen



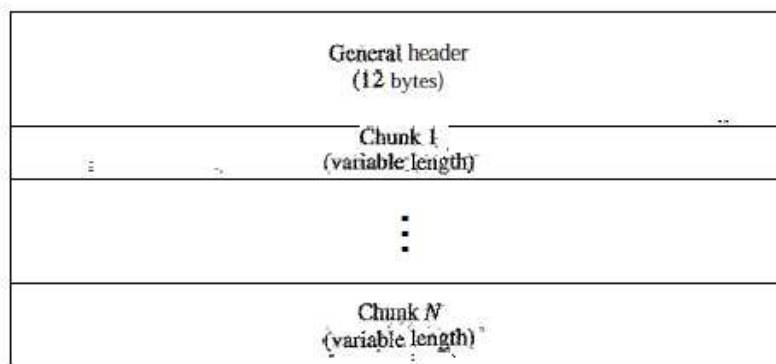
2. TCP diperlakukan sebagai satu kesatuan, sebuah paket SCTP dapat membawa beberapa potongan data; masing-masing dapat berasal dari aliran yang berbeda.
3. Bagian pilihan, yang dapat menjadi bagian dari segmen TCP, tidak ada dalam sebuah paket SCTP. Pilihan pada SCTP ditangani dengan mendefinisikan jenis potongan baru.
4. Bagian wajib dari *header* TCP adalah 20 *byte*, sedangkan *header* umum dalam SCTP adalah hanya 12 *byte*. *Header* SCTP lebih pendek karena sebagai berikut:
 - a. Sebuah urutan nomor SCTP (TSN) memiliki setiap potongan data dan terletak di potongan *header*.
 - b. Jumlah pengakuan dan ukuran jendela adalah bagian dari setiap potongan kontrol.
 - c. Tidak ada kebutuhan untuk panjang *field header* (ditampilkan sebagai HL di segmen TCP) karena tidak ada pilihan untuk membuat panjang variabel *header*, panjang *header* SCTP adalah tetap (12 *byte*).
 - d. Tidak ada kebutuhan untuk *pointer* mendesak di SCTP.

5. *Checksum* pada TCP adalah 16 bit, dalam SCTP, itu adalah 32 bit.
6. *Tag* verifikasi dalam SCTP adalah asosiasi sebuah identifier, yang tidak ada di TCP. Dalam TCP, kombinasi alamat IP dan *port* mendefinisikan koneksi, dalam SCTP mungkin telah *multihorning* menggunakan alamat IP yang berbeda. Sebuah verifikasi unik *tag* diperlukan untuk mendefinisikan setiap asosiasi.
7. TCP termasuk nomor satu urutan di *header* yang mendefinisikan jumlah *byte* pertama di bagian data. Sebuah paket SCTP dapat mencakup beberapa potongan data yang berbeda. TSN, SIS, dan SSN mendefinisikan setiap potongan data.
8. Beberapa segmen TCP yang membawa informasi kontrol (seperti SYN dan FIN) perlu menggunakan satu nomor urutan; kontrol potongan dalam SCTP tidak pernah menggunakan TSN, SI, atau SSN. Ketiga pengidentifikasi hanya memiliki potongan data, bukan untuk seluruh paket.

1.4 Format paket

Pada bagian ini, kita menunjukkan format paket dan berbagai jenis potongan. Sebagian besar informasi yang disajikan dalam bagian ini akan menjadi jelas kemudian, bagian ini dapat dilewati dalam bacaan pertama atau digunakan hanya sebagai referensi. Sebuah paket SCTP memiliki wajib header umum dan satu set blok disebut potongan. Ada dua jenis potongan: kontrol potongan dan potongan data. Sebuah potongan kontrol mengendalikan dan memelihara asosiasi; data bongkahan membawa data pengguna. Dalam sebuah paket, potongan kontrol *corne* sebelum potongan data. Gambar 23.31 menunjukkan format umum dari sebuah paket SCTP.

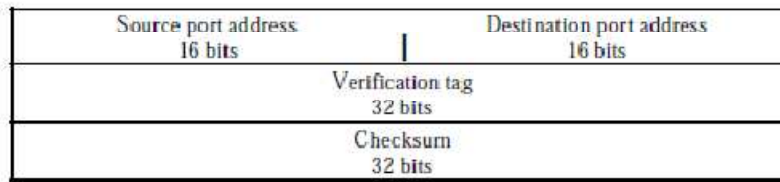
Figure 23.31 *SCTP packet format*



1.4.1 Header Umum

Sebuah paket SCTP wajib memiliki *header* umum dan satu set blok disebut potongan. Ada dua jenis potongan: potongan kontrol dan potongan data. Sebuah potongan kontrol mengendalikan dan memelihara asosiasi; sepotong data yang membawa data pengguna. Dalam sebuah paket, potongan kontrol *corne* sebelum potongan data.

Figure 23.32 *General header*



Ada empat bidang di *header* umum:

- Sumber alamat *port*. Ini adalah bidang 16-bit yang mendefinisikan nomor *port* dari proses mengirim paket.
- alamat *port* tujuan. Ini adalah bidang 16-bit yang mendefinisikan nomor *port* dari proses penerimaan paket.
- Verifikasi *tag* adalah angka yang cocok dengan paket ke asosiasi. Verifikasi *tag* mencegah paket dari sebuah asosiasi sebelumnya dari yang salah sebagai paket dalam asosiasi. Hal tersebut berfungsi sebagai pengidentifikasi untuk asosiasi, hal ini diulang dalam setiap paket selama asosiasi. Ada verifikasi terpisah digunakan untuk setiap arah dalam asosiasi.
- *Checksum*. *Field* ini berisi 32-bit CRC-32 *checksum*. Perhatikan bahwa ukuran *checksum* meningkat dari 16 (di UDP, TCP, dan IP) untuk 32 bit untuk memungkinkan penggunaan CRC *checksum*-32.

1.4.2 Potongan Data

Control information atau pengguna data dibawa dalam potongan. Tiga bidang pertama yang umum untuk semua potongan, bidang informasi tergantung pada jenis potongan. Poin penting untuk diingat adalah bahwa SCTP memerlukan bagian informasi yang akan menjadi potongan kelipatan dari 4 byte, jika tidak, *padding byte* (delapan) ditambahkan di akhir bagian

1.5 Sebuah Asosiasi SCTP

Seperti TCP, SCTP merupakan protokol *connection-oriented*. Namun, sambungan dalam SCTP disebut asosiasi untuk menekankan *multihoming*.

1.5.1 Pembentukan Asosiasi

Pembentukan asosiasi dalam SCTP membutuhkan empat arah lintasan yang saling terhubung. Dalam prosedur ini, suatu proses, biasanya klien, ingin mendirikan asosiasi dengan proses lain, biasanya *server* menggunakan SCTP sebagai protokol lapisan *transport*. Serupa dengan TCP, SCTP *server* perlu dipersiapkan untuk menerima asosiasi (terbuka pasif). Asosiasi pendirian dibangun oleh klien (terbuka aktif).

1.5.2 Transfer Data

Tujuan utama asosiasi adalah untuk mentransfer data antara dua bagian. Setelah asosiasi ini dibangun, maka dua arah transfer data dapat terjadi. Klien dan *server* dapat kedua mengirim data. Seperti TCP, SCTP mendukung penyusupan data.

1.5.3 Asosiasi Termination

Dalam SCTP, seperti TCP, salah satu dari dua pihak yang terlibat dalam pertukaran data (client atau server) dapat menutup koneksi. Namun, tidak seperti TCP, SCTP tidak mengizinkan situasi setengah menutup. Jika salah satu ujung menutup asosiasi, ujung yang lain harus menghentikan pengiriman data baru. Jika ada data yang tersisa dalam antrian penerima permintaan penghentian, mereka akan dikirim dan asosiasi ditutup. Asosiasi pemutusan menggunakan tiga paket, seperti yang ditunjukkan pada Gambar 23,35. Perhatikan bahwa meskipun angka tersebut menunjukkan kasus di mana pemutusan dimulai oleh klien, juga dapat dimulai oleh server. Perhatikan bahwa tidak bisa beberapa skenario pemutusan hubungan asosiasi

1.6 Flow Control

Flow control dalam SCTP adalah mirip dengan yang di TCP. Di TCP, kita perlu berurusan dengan hanya satu unit data, byte. Dalam SCTP, kita perlu menangani dua unit dari data, dan byte chunk tersebut. Nilai *rwnd* dan *cwnd* disajikan dalam byte, nilai-nilai TSN dan pengakuan disajikan dalam potongan. Untuk menampilkan konsep tersebut, kami membuat beberapa asumsi yang tidak realistis. Kami berasumsi bahwa tidak pernah ada kemacetan di jaringan dan bahwa jaringan kesalahan-bebas. Dengan kata lain, kita berasumsi bahwa *cwnd* tidak terbatas dan tidak ada paket hilang atau tertunda atau tiba rusak. Kami juga menganggap bahwa transfer data satu arah. Kami benar asumsi yang tidak realistis kami dalam bagian berikutnya. Lancar SCTP implementasi masih menggunakan jendela byte yang berorientasi untuk kontrol aliran. Kami, bagaimanapun, menunjukkan buffer dihal potongan untuk membuat konsep lebih mudah dimengerti.

1.6.1 Penerima Situs

Penerima memiliki satu penyangga (antrian) dan tiga variabel. Antrian memegang potongan data yang diterima yang belum dibaca oleh proses. Variabel pertama memegang TSN terakhir diterima, *cumTSN*. Variabel kedua memegang ukuran buffer yang tersedia, *winsize*. Variabel ketiga memegang pengakuan akumulatif terakhir, *lastACK*. Gambar 23,36 menunjukkan antrian dan variabel di lokasi penerima.

1.6.2 Situs Pengirim

Pengirim memiliki satu penyangga (antrian) dan tiga variabel: *curTSN*, *rwnd*, dan *inTransit*, seperti yang ditunjukkan pada Gambar 23.37. Kita berasumsi setiap bingkah 100 bytes panjang. buffer yang memegang potongan yang dihasilkan oleh proses yang baik telah dikirim atau siap untuk dikirim. Variabel pertama, *curTSN*, mengacu pada chunk berikutnya yang akan dikirim. Semua potongan dalam antrian dengan TSN kurang dari nilai ini telah dikirim, namun tidak diakui, mereka beredar. Variabel kedua, *rwnd*,

memegang nilai terakhir diiklankan oleh penerima (dalam byte). Variabel ketiga, *inTransit*, memegang jumlah byte dalam perjalanan, byte yang dikirim tetapi belum diakui

1.6.3 Skenario A

Mari kita memberikan skenario sederhana seperti yang ditunjukkan pada Gambar 23.38. Pada awal nilai *rwnd* di lokasi pengirim dan nilai *winSize* di lokasi penerima adalah 2000 (diiklankan selama pendirian asosiasi). Awalnya, ada empat pesan dalam antrian pengirim. Pengirim mengirimkan salah satu potongan data dan menambah jumlah byte (1000) terhadap variabel *inTransit*. Setelah beberapa waktu, pengirim akan memeriksa perbedaan antara *rwnd* dan *inTransit*, yang adalah 1000 byte, sehingga dapat mengirimkan kembali potongan data. Sekarang perbedaan antara kedua variabel adalah 0 dan tidak ada potongan lebih banyak data dapat dikirim. Setelah beberapa saat, sebuah SACK tiba yang mengakui potongan data 1 dan 2. Kedua potongan tersebut dihapus dari antrian. Nilai *inTransit* sekarang 0. SACK, bagaimanapun, diiklankan jendela penerima nilai 0, yang membuat *sender.update rwnd* untuk 0. Sekarang pengirim diblokir, tidak bisa mengirim potongan data (dengan satu pengecualian dijelaskan nanti).

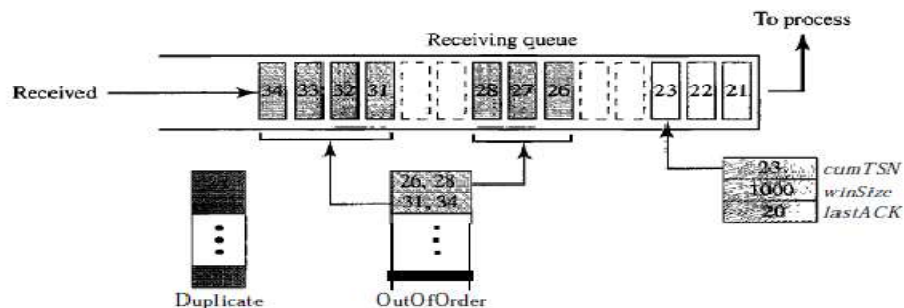
1.7 Error Control

SCTP, seperti TCP, merupakan protokol transport layer yang handal. Menggunakan sepotong SACK untuk melaporkan keadaan buffer penerima ke pengirim. Setiap implementasi yang berbeda menggunakan entitas dan timer untuk situs penerima dan pengirim. Kami menggunakan desain yang sangat sederhana untuk menyampaikan konsep kepada pembaca.

1.7.1 Penerima Situs

Dalam desain kami, toko-toko penerima semua potongan yang telah tiba dalam antrian yang termasuk yang out-of-order. Namun, ia meninggalkan ruang untuk setiap potongan yang hilang. Ini membuang duplikat pesan, tetapi melacak mereka untuk laporan kepada pengirim. Gambar 23.39 menunjukkan khas desain untuk situs penerima dan keadaan antrian menerima pada titik waktu tertentu.

Figure 23.39 *Error control, receiver site*

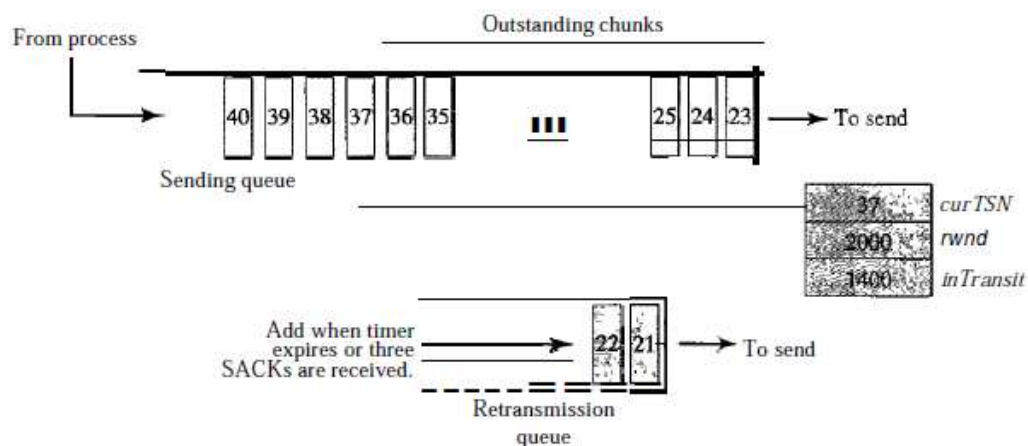


Pengakuan terakhir yang dikirim adalah untuk potongan data 20. Ukuran jendela yang tersedia adalah 1000 byte. Potongan 21 hingga 23 telah diterima secara berurutan. Yang pertama out-of-order blok berisi potongan 26 sampai dengan 28. Kedua out-of-order blok berisi potongan 31-34. Variabel memegang nilai cumTSN. Sebuah array variabel melacak awal dan akhir setiap blok yang rusak. Sebuah array dari variabel memegang duplikat potongan diterima. Perhatikan bahwa tidak ada kebutuhan untuk menyimpan duplikat potongan dalam antrian, mereka akan dibuang. Angka ini juga menunjukkan chunk SACK yang akan dikirim untuk melaporkan keadaan penerima ke pengirim. Para TSN angka untuk out-of-potongan-order yang relatif (offset) ke TSN kumulatif.

1.7.2 Situs Pengirim

Pada situs pengirim, desain kami menuntut dua buffer (antrian): antrian pengiriman dan antrian retransmission. Kami juga menggunakan tiga variabel *rwnd*, *inTransit*, dan *curTSN* seperti yang dijelaskan dalam bagian sebelumnya. Gambar 23,40 menunjukkan desain khas.

Figure 23.40 *Error control, sender site*



Antrian pengiriman memegang potongan 23-40. Para potongan 23-36 telah dikirim, namun tidak diakui, mereka adalah potongan beredar. *curTSN* poin untuk chunk berikutnya untuk dikirim (37). Kami berasumsi bahwa setiap cuil adalah 100 byte, yang berarti 1400 byte data (potongan 23-36) adalah dalam perjalanan. Pengirim saat ini memiliki antrian pengiriman ulang. Ketika sebuah paket dikirim, timer retransmission mulai untuk itu paket (potongan semua data dalam paket itu). Beberapa implementasi menggunakan satu timer tunggal untuk seluruh asosiasi, namun kami terus dengan tradisi kami satu timer untuk setiap paket untuk penyederhanaan. Ketika timer retransmission untuk paket berakhir, atau empat duplikat SACK tiba yang menyatakan paket hilang (retransmission cepat telah dibahas pada Bab 12), bagian dalam paket yang akan dipindah

ke antrian retransmisi dikirimkan kembali. Potongan ini dianggap hilang, bukan yang luar biasa. Ini potongan dalam antrian retransmisi memiliki prioritas. Dengan kata lain, pada saat pengirim mengirimkan sepotong, akan bongkahan 21 dari antrian pengiriman ulang.

1.7.3 Mengirim Potongan Data

Sebuah akhir dapat mengirim data paket apabila terdapat data potongan dalam antrian pengiriman dengan TSN lebih besar dari atau sama dengan curTSN atau jika ada data potongan pada antrian pengiriman ulang. Antrian retransmisi memiliki prioritas. Namun, ukuran total chunk data atau potongan termasuk dalam paket tidak boleh melebihi $rwnd - inTransit$, dan ukuran total dari frame tidak boleh melebihi ukuran MTU seperti yang kita bahas pada bagian sebelumnya.

Retransmission Untuk mengendalikan potongan hilang atau dibuang, SCTP, seperti TCP, mempekerjakan dua strategi: menggunakan *timer retransmission* dan menerima empat SACK dengan potongan data yang hilang sama.

1.7.4 Menghasilkan Potongan SACK

Permasalahan lain dalam kontrol kesalahan adalah generasi potongan SACK. Aturan untuk menghasilkan potongan SACK SCTP serupa dengan aturan yang digunakan untuk pengakuan dengan bendera ACK TCP.

REFERENCES

- [1] Forouzan A. Behrouz, "Data Communication and Networking Fourth Edition" McGraw-Hill, 2007.
- [2] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].