SCTP protokol

Lukáš Krejčík

krejcl2@fel.cvut.cz

Obsah

1 Úvod

SCTP je spolehlivý transportní protokol využívající služeb potenciálně nespolehlivé služby jako je např.IP. SCTP nabízí bezchybný, neduplikovaný přenos datagramů (zpráv). Detekce poškození ,ztráty nebo duplikace dat je dosaženo pomocí kontrolních součtů (checksums) a pomocí pořadových čísel (sequence numbers).

Původ protokolu SCTP najdeme v telefonických kruzích - s definicí protokolu přišla IETF skupina SIGTRANS, která se zabývá přenosem telefonní signalizace po IP. Odtud pochází požadavek na několik navzájem nezávislých kanálů, které jsou přepravovány paralelně.

Právě tohle je zřejmě největší odlišností SCTP od stávajících transportních protokolů. Po navázání spojení, kterému se v terminologii SCTP říká asociace, po něm lze přenášet řadu navzájem nezávislých proudů (stream). V rámci každého z nich dokáže SCTP garantovat doručení všech dat ve správném pořadí. Případný výpadek (a pozdější opakování, čili zdržení) v některém z proudů se však nijak netýká proudů ostatních. Jejich komunikace pokračuje bez přerušení.

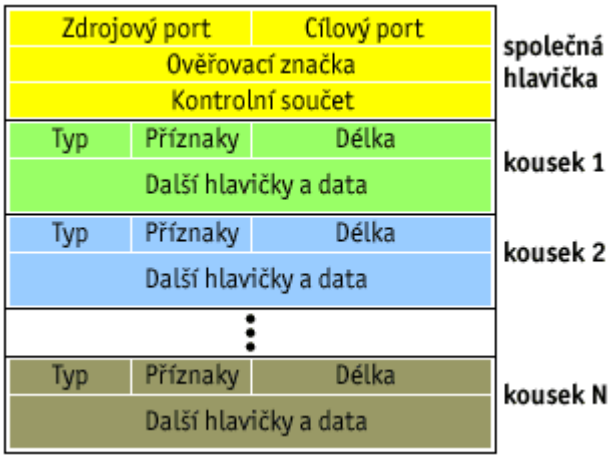
Schopnosti SCTP asociace by se tedy zhruba daly přirovnat ke svazku souběžných TCP spojení. Ovšem SCTP asociace by měla mít menší režii a nabízí také pár vylepšení.

2 Pakety

PDU SCTP (Protocol Data Units) jsou nazývány pakety. Pokud SCTP běží přes IP, SCTP paket tvoří payload IP paketu. Nejprve si řekneme něco k formátu SCTP paketu.

2.1 Hlavička

Formát SCTP paketu je následující. Nejdříve je universální 12ti bytová hlavička, ta obsahuje jen nejzákladnější údaje: čísla portů identifikující aplikace na obou koncích asociace, údaj pro ověření totožnosti odesilatele a kontrolní součet. Každý SCTP paket je chráněn 32-bitovým součtem (na rozdíl od 16-ti bitového kontrolního součtu TCP a UDP). SCTP paket s neplatným kontrolním součtem je zahozen. Hlavička také obsahuje ověřovací značku (verification tag), která je specifická pro každou asociaci. Jsou zde dvě hodnoty značek používané v rámci jedné asociace (více v sekci Stavy SCTP) . Po hlavičce následují tzv. kousky (chunks).



Obrázek – Formát SCTP paketu

2.2 Kousky (chunks)

Každý kousek začíná polem označujícím typ kousku, odlišující datové kousky od různých typů řídicích kousků, dále následují specifické příznaky kousku a velikost kousku, protože jednotlivé kousky mohou mít velikost různou. Hodnota velikosti je aktuální payload kousku.

SCTP rozlišuje 13 druhů kousků definovaných pro běžné použití. Jejich názvy uvedené níže jsou pro jednoduchost okopírované z RFC2960:

```
ID Value    Chunk Type
-----       ----------
0           - Payload Data (DATA)
1           - Initiation (INIT)
2           - Initiation Acknowledgement (INIT ACK)
3           - Selective Acknowledgement (SACK)
4           - Heartbeat Request (HEARTBEAT)
5           - Heartbeat Acknowledgement (HEARTBEAT ACK)
6           - Abort (ABORT)
7           - Shutdown (SHUTDOWN)
8           - Shutdown Acknowledgement (SHUTDOWN ACK)
9           - Operation Error (ERROR)
10          - State Cookie (COOKIE ECHO)
11          - Cookie Acknowledgement (COOKIE ACK)
12          - Reserved for Explicit Congestion Notification Echo (ECNE)
13          - Reserved for Congestion Window Reduced (CWR)
14          - Shutdown Complete (SHUTDOWN COMPLETE)
15 to 62    - reserved by IETF
63          - IETF-defined Chunk Extensions
64 to 126   - reserved by IETF
127         - IETF-defined Chunk Extensions
128 to 190  - reserved by IETF
191         - IETF-defined Chunk Extensions
192 to 254  - reserved by IETF
255         - IETF-defined Chunk Extensions
```

3 Stavy SCTP

Tato část popisuje stavy do kterých instance SCTP protokolu vstupuje při navazování asociace a při jejím ukončování. Inicializace asociace je hotová na obou stranách po výměně čtyř zpráv. Pasivní strana (server) nealokuje žádné prostředky dokud nepřijde třetí z těchto zpráv a je potvrzena. Tím je zajištěno, že požadavek na navázání asociace pochází od zdroje mající vážný zájem o navázání asociace, že se nejedná o útok.

3.1 Navázání asociace

3.1.2 Strana serveru

Server přijme požadavek na navázání asociace (kousek INIT), obvykle ve stavu CLOSED, a analyzuje data v tomto kousku. Z těchto dat generuje všechny potřebné hodnoty pro navázání asociace na jeho straně a zašifruje je bezpečnostním klíčem (obvykle MD5 nebo SHA-1 algoritmus). Tyto hodnoty tvoří tzv. COOKIE, společně s odvozeným ověřovacím kódem zprávy (message authentication code – MAC). Tento COOKIE je odeslán zpět odesílateli INIT kousku jako INIT-ACK kousek. Server zůstává v CLOSED stavu a odstraní se vše ohledně přijatého INIT kousku.

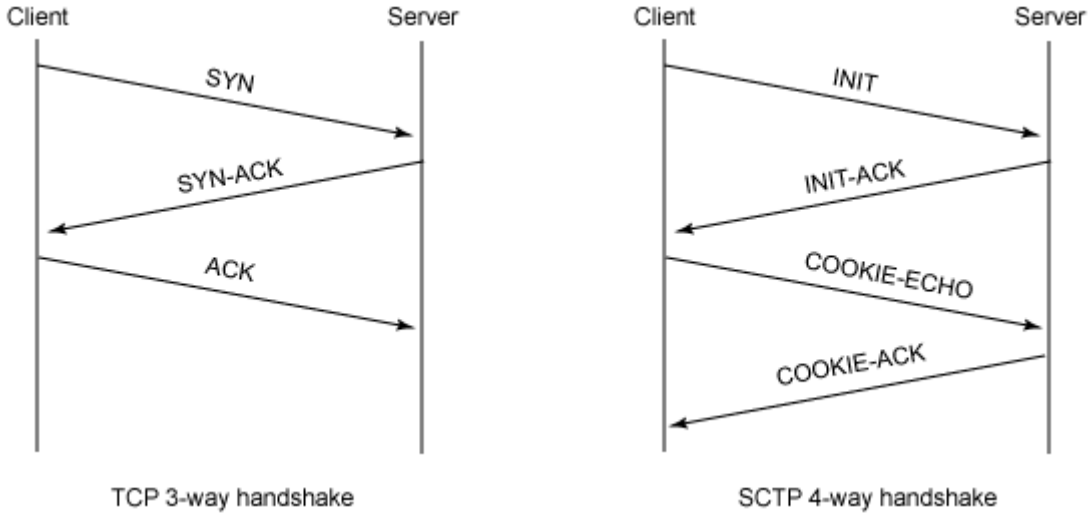
Hned po přijetí COOKIE-ECHO kousku (který obsahuje COOKIE datovou strukturu jako parametr), server rozbalí data obsažená v tomto COOKIE, a použije znovu MAC zde obsaženou k ověření, zda je on původce tohoto COOKIE. Pokud je vše v pořádku, jde o COOKIE vytvořený dříve serverem, a data obsažená v COOKIE jsou použita pro inicializaci SCTP instance. Server pošle COOKIE-ACK klientovi a vstoupí do ESTABLISHED stavu. V tomto stavu je schopen přijímat data nebo posílat samotné datové kousky.

3.1.2 Strana klienta

Při požadavku vyšší vrstvy o navázání asociace (voláním ASSOCIATE primitiva) jsou inicializována všechna potřebná data pro vytvoření INIT kousku. Tento INIT kousek je poslán na jednu transportní adresu (IP-adresa a port) serveru. Při tom je odstartován čítač, který pravidelně spouští opakované posílání INIT kousku při vypršení času pro přijetí INIT-ACK kousku. Pokud při volitelném počtu pokusů není přijat INIT-ACK, je hlášena chyba vyšší vrstvě a server je ohlášen jako nedostupný. Po odeslání INIT kousku vstupuje klient do COOKIE-WAIT stavu.

Když klient přijme INIT-ACK kousek ze strany serveru v COOKIE-WAIT stavu, zastaví se čítač pro opakované odesílání INIT kousku, vytvoří COOKIE-ECHO kousek, vloží se COOKIE z přijatého INIT-ACK kousku do COOKIE-ECHO kousku a pošle zpět serveru. Po odeslání prvního COOKIE-ECHO, vstupuje instance protokolu do COOKIE-ECHOED stavu. Pokud není po volitelném počtu odeslaných COOKIE-ECHO žádná COOKIE-ACK přijata, server je nahlášen jako nedostupný.

Po přijetí COOKIE-ACK kousku ze serveru klient vstupuje do ESTABLISHED stavu. COOKIE-ECHO může být doprovázen několika datovými kousky. Je na serveru, zda budou akceptována nebo zahozena.



Obrázek znázorňuje porovnání TCP a SCTP navazování spojení. SCTP je oproti TCP chráněn proti útokům ze strany klienta.

3.2 Ukončení asociace

Obě strany se mohou rozhodnout pro ukončení SCTP asociace z mnoha důvodů, a mohou tak učinit kdykoliv (za předpokladu že nejsou ve stavu CLOSED). Je zde možnost standardního ukončení , zajištující že žádná data nebudou ztracena, nebo násilného ukončení, kde se dále nestaráme o protějšek.

3.2.1 Standardní ukončení asociace

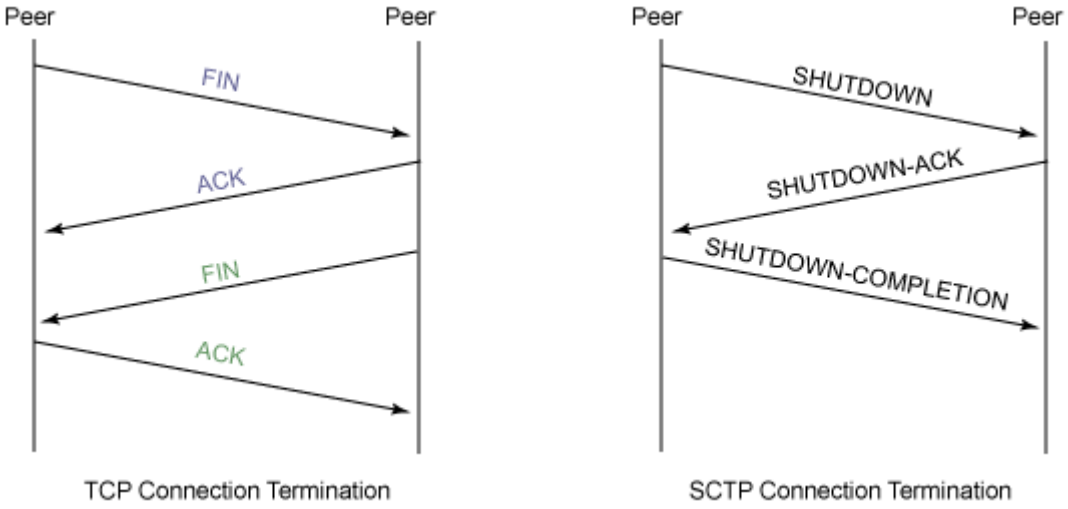
Okamžitě po přijetí SHUTDOWN primitiva z vyšší vrstvy , instance SCTP zastaví přijímání dat a začne posílat SHUTDOWN kousek, hned po potvrzení všech nevyřízených dat. Tento proces je hlídán časovačem, který toto pravidelně opakuje, z důvodu ztráty SHUTDOWN. Pokud druhá strana přijme SHUTDOWN kousek, odpoví SHUTDOWN-ACK kouskem, hned po potvrzení všech jeho dat.

Když první strana (která odstartovala ukončení) obdrží SHUTDOWN-ACK, zastaví časovač, pošle SHUTDOWN-COMPLETE kousek, odstraní se všechny záznamy o asociaci a vstoupí do CLOSED stavu.

Strana, která přijala SHUTDOWN-COMPLETE kousek také odstraní záznamy o asociaci a vstoupí do CLOSED stavu. Pokud je SHUTDOWN-COMPLETE zpráva ztracena, opakovaně se posílá SHUTDOWN-ACK kousek, dokud není druhá strana označena jako nedostupná.

3.2.2 Násilné ukončení asociace

Kterákoliv strana se může rozhodnout násilně ukončit asociaci posláním ABORT kousku, všechna odeslaná ale ještě nepotvrzená data nebudou potvrzena. Odesilatel musí vyplnit ověřovací značku (verification tag) v odesílaném paketu a **nesmí** k ABORT připojit žádná data. Příjemce neodpovídá na ABORT, ale potvrdí kousek a odstraní asociaci, pokud ABORT obsahuje správnou hodnotu značky. Pokud ano, oznámí ukončení vyšší vrstvě.



Obrázek znázorňuje porovnání TCP a SCTP ukončení spojení. SCTP asociace neumožňuje po ukončení z jedné strany touto stranou dále přijímat data, na rozdíl od TCP

Grafické znázornění stavů viz. Příloha.

4 Přenos dat

Implementace SCTP musí mít mechanismus řízení toku a řízení zahlcení s ohledem na RFC 2960, který zajistí že SCTP může být bez problémů uveden v sítích kde je široce používán TCP.

4.1 Obecné pojetí

SCTP rozlišuje několik datových proudů v rámci jedné asociace. SCTP pracuje na dvou úrovních:

- V rámci asociace je bezpečný přenos datagramů zajištěn pomocí kontrolních součtů (checksum), pořadových čísel (sequence number) a selektivním opakovacím mechanismem. Aniž by byla brána v potaz počáteční sekvence, každý správně doručený datový kousek je doručen do druhé, nezávislé úrovně.

- Druhá úroveň realizuje flexibilní doručovací mechanismus založený na představě několika nezávislých proudů datagramů v rámci jedné asociace. Kousky náležící jednomu nebo několika proudům mohou být dány do jednoho paketu a přeneseny jako jeden SCTP paket, pod podmínkou že nejsou delší než současná velikost MTU (maximální přenosové jednotky Maximum Transmission Units).

Detekce ztráty a zdvojení datových kousků je umožněno číslováním všech datových kousků odesílatelem pomocí tzv. transportních pořadových čísel (Transport Sequence Number –TSN). Potvrzení posílaná příjemcem k odesílateli jsou založena na těchto pořadových číslech.

Opakování přenosu je řízeno časovačem. Doba časovače je odvozena z neustálého měření zpětného prodlení (round trip delay). Jakmile časovač vyprší, (a řízení zahlcení umožňuje vysílání), všechny nepotvrzené datové kousky jsou poslány znovu a časovač je spuštěn znovu s dvojnásobnou dobou trvání (obdobně jako TCP).

Jakmile příjemce detekuje jednu nebo více děr v pořadí datových kousků, každý přijatý SCTP paket je potvrzen vysláním Selektivního Potvrzení (Selective Acknowledgement – SACK) který oznamuje všechny trhliny. SACK je obsažen ve specifickém řídicím kousku. Jakmile odesílatel obdrží 4 po sobě jdoucí SACK označující stejný chybějící datový kousek, je tento datový kousek poslán okamžitě (tzv. rychlé opakování – Fast Retransmit). Novější operační systémy mají podobnou podporu v TCP (RFC 2018).

4.2 Identifikátor proudu a Pořadové číslo proudu

Každý datový kousek musí nést platný identifikátor proudu. Pokud je přijat datový kousek s neplatným identifikátorem proudu, potvrdí přijetí datového kousku, okamžitě však odešle ERROR kousek označující „neplatný identifikátor proudu (Invalid Stream Identifier) a zahodí DATA kousek. Pořadové číslo proudu (Stream Sequence Number – SSN) ve všech proudech začíná od nuly když je navázána asociace. Pokud SSN dosáhne čísla 65535, další číslo je nastaveno na nulu.

4.3 Řízení toku

SCTP používá okénkový potvrzovací mechanismus pro řízení toku a zahlcení, podobný jako je v TCP (RFC 2581). Příjemce dat může řídit tempo v kterém odesílatel posílá data specifikací velikosti okénka a posláním této hodnoty zpět se všemi SACK kousky.
Odesílatel vlastní proměnnou zvanou Okno Zahlcení (Congestion Window – CWND) která řídí maximální počet momentálně nepotvrzených bajtů, (tj. bajtů, které mohou být odeslány před tím,než budou potvrzeny). Všechny přijaté datové kousky musejí být potvrzeny, na to odesílatel čeká určitý čas (obvykle 200 ms).

4.4 Selektivní potvrzení

Každé potvrzení sebou nese všechna TSN čísla kousků které byli přijaty. Hodnota zvaná **Cumulative TSN Ack** označuje všechna data která byla úspěšně sestavena na straně příjemce a mohou být doručena vyšší vrstvě. Kromě toho, tak zvaná hodnota **Gap Blocks** označuje segmenty datových kousků, které dorazily, ale mají trhliny v podobě chybějících datových kousků mezi nimi.

Datové kousky, které se mohly ztratit při přenosu budou doručeny po vypršení času pro opakování přenosu, nebo po obdržení čtyř SACK kousků označujících stejná chybějící datové kousky (rychlé opakování) .

V případě opakování přenosu, které značí ztrátu paketů, implementace musí přiměřeně upravit parametry řízení toku a zahlcení

4.5 Řízení toku pro Multihoming

Standardně je všechen přenos směrován na předem zvolenou adresu z množiny cílových adres, která je nazývána Primární Adresa (Primary Address). Opakování přenosu může být po jiné cestě, pokud první je přetížena, tak opakování tuto cestu neovlivní (pokud opakování nejde po cestě přes stejný uzel, kde se ztrácí data). Potvrzení by mělo být odesláno na adresu, z které pocházejí data.

Pokud aktivní cesta vykazuje velký počet chyb a tento počet přesáhne určitou hranici, SCTP implementace oznámí vyšší vrstvě, že cesta se stává nečinnou. Pak by měla být aplikací zvolená nová primární cesta. Více viz. SCTP Multihoming.

4.6 Řízení zahlcení

Řízení zahlcení SCTP implementace s ohledem na RFC2960 může mít vliv tam, kde je požadováno včasné doručení zpráv (např. doručení signalizačních dat). Nicméně, toto zajišťuje správné chování SCTP pokud by byl uveden ve velkém měřítku do existující sítě s přepínáním paketů jako je internet. Mechanismus řízení zahlcení pro SCTP je odvozen z RFC2581 – TCP řízení zahlcení, a bylo upraveno pro multihoming. Pro každou cílovou adresu jsou drženy parametry pro řízení toku a zahlcení, z tohoto pohledu je možné SCTP asociaci s více cestami přirovnat ke stejnému počtu TCP spojení.

Podobně jako TCP, má SCTP dva režimy: pomalý start (*slow start*) a vyloučení zahlcení (*congestion avoidance*). Režim je dán množinou proměnných řízení zahlcení, a jak již bylo zmíněno, jsou příslušné určité cestě. Takže, zatímco přenos po primární cestě může být v režimu vyloučení zahlcení, pro záložní cesty může být použit pomalý start.

V režimu pomalý start je proměnná okno zahlcení (congestion window - CWND) pomalu zvětšována a pokud přesáhne určitou hranici, zvanou práh pomalého startu (Slow Start Threshold – SSTRESH) , režim se změní na vyloučení zahlcení. Pokud dojde k opakování přenosu (z důvodu vypršení času nebo rychlého opakování) , je SSTRESH drasticky snížen a vynulováno CWND (vypršení času způsobí nový pomalý start).

5 Multihoming

Další nezbytnou vlastností SCTP je podpora tzv. Multi-home uzlů, tzn. Uzlů, které mohou být dosaženy pomocí několika IP adres. Pokud uzly SCTP a odpovídající IP síť má topologii takovou, že přenos z jednoho uzlu do jiného jsou po jiné fyzické cestě, pokud je zvolena jiná cílová IP adresa, asociace se stává odolnější proti fyzickým výpadkům sítě a problémům tohoto druhu.

5.1 Navázání asociace, správa adres

Pokud je u klienta možnost multi-home, klient informuje server o všech jeho IP adresách a předává je jako parametr INIT kousku. Klient nemusí znát všechny IP adresy serveru, server ho informuje o všech svých IP adresách v INIT-ACK kousku. IP adresy mohou být Ipv4 nebo IPv6, případně jejich směs. SCTP instance považuje všechny IP adresy jeho protějšku za jednu přenosovou cestu k tomuto bodu.

Pokud není žádná IP adresa obsažena v INIT nebo INIT-ACK kousku, je použita zdrojová IP adresa IP paketu, který nese SCTP datagram. Toho usnadňuje aplikaci SCTP pokud je v síti překladač adres (NAT). Pro další ulehčení, byla v RFC2960 uvedena další vlastnost, která umožňuje použití jmen místo IP adres.

5.2 Sledování cest

Instance SCTP sleduje všechny přenosové cesty k protějšku v rámci asociace. Pokud se cesta momentálně nepoužívá pro přenos datových kousků, jsou zde posílány HEARTBEAT kousky, které musí být potvrzeny HEARTBEAT-ACK kousky.

Každé cestě je přiřazen stav : buď je aktivní, nebo neaktivní. Aktivní je pokud je (nebo nedávno byla) používána pro přenos libovolných SCTP datagramů potvrzených protějškem.

Některé transportní adresy se však mohou z důvodů častých výpadků stát neaktivní. Pokud máme odchozí data a primární cesta se stává neaktivní (např.z důvodu chyb), nebo pokud uživatel vyžaduje přenos na neaktivní cílovou adresu, dříve než je nahlášena chyba vyšší vrstvě, SCTP zkusí poslat data na alternativní cílovou adresu, pokud je nějaká k dispozici.

Počet událostí, kdy HEARTBEAT kousky nebyly potvrzeny v určitém čase, nebo nastalo opakované vysílání, je zaznamenáván a pokud je větší než určitá hranice (volitelná), protějšek je považován za nedostupný a asociace bude ukončena.

5.3 Výběr cesty

Při navazování asociace je jedna z IP adres z obdrženého seznamu zvolena jako primární, tzv primární cesta (*primary path)*. Datové kousky jsou přenášeny po této primární přenosové cestě. Pro opakování přenosu však může být zvolena jiná, neaktivní cesta, pokud je k dispozici. Pro podporu měření zpětného zpoždění by měly SACK kousky být poslány na zdrojovou adresu IP paketu, obsahujícího datový kousek, který zapříčinil SACK.

Uživatel SCTP je informován o stavu přenosové cesty na žádost nebo pokud přenosová cesta změní svůj stav. Pak lze nařídit lokální SCTP instanci použití nové primární cesty.

6 Datové proudy SCTP

SCTP rozlišuje různé proudy (streams) zpráv v rámci jedné SCTP asociace. Každý z těchto proudů je nezávislý, ale přísluší dané asociaci. Každému proudu je přiřazeno číslo, které je zakódované v SCTP paketu. Pod pojmem „proud“ se v SCTP rozumí pořadí uživatelských zpráv, které mají být doručeny vyšší vrstvě v určitém pořadí s ohledem na ostatní zprávy ve stejném proudu.

Uživatel SCTP může určit počet proudů během navazování asociace. Tento počet je pak projednán se vzdáleným koncem. Pak je každá zpráva přiřazena různým proudům (primitiva SEND, RECEIVE). Uvnitř pak SCTP přiřadí pořadové číslo proudu (Stream Sequence Number) uvedené uživatelem každé zprávě. SCTP umožňuje na straně příjemce doručení zpráv v pořadí v rámci daného proudu. Nicméně pokud je jeden proud blokován čekáním na další zprávu v pořadí (z důvodu např. ztráty paketu), doručování uvnitř ostatních proudů pokračuje.

SCTP umožňuje zrušení doručení zpráv v pořadí, pak jsou uživatelské zprávy doručeny uživateli SCTP hned po přijetí.

7 SCTP z hlediska vysoké dostupnosti

SCTP nabízí tedy několik zajímavých mechanismů z hlediska vysoké dostupnosti. Zde si ještě stručně popíšeme jejich přínos s ohledem na vysokou dostupnost. Mezi ty nejzajímavější patří tedy:

Multi-homing
Multi-streaming
Ochrana navazování asociace a ukončení
Volitelné neuspořádané doručeni dat

Multi-homing

Aplikace využívající multihoming poskytují vyšší dostupnost než aplikace používající TCP. Koncový bod umožňující multihoming disponuje více síťovými rozhraními a tím pádem více IP adresami. Je tedy možné při navazování asociace využít více cest pro přenos dat v rámci jedné asociace. Toho lze využít pro zachování spojení aplikace při rozpadu jedné z cest. Uvažujme příklad, máme notebook obsahující bezdrátové rozhraní 802.11 a ethernet rozhraní. Pokud máme notebook v jeho docking station, vysokorychlostní ethernet může být upřednostňován (v SCTP jako primární adresa), ale při ztrátě tohoto připojení (např. vyjmutí z docking station), komunikace pokračuje přes bezdrátové rozhraní bez rozpadu navázaných asociací. Při návratu notebooku do docking station se může komunikace navrátit zpět na ethernet.

Multi-streaming

Jak bylo zmíněno v sekci 6, SCTP umožňuje několik datových proudů v rámci jedné asociace. Toho lze využít pro lepší přístupnost v přenosu dat. Například, HTTP protokol sdílí řízení a data přes stejný soket. Webový prohlížeč požaduje soubor ze serveru, a server ho pošle zpět přes stejné spojení. Server s více datovými proudy je více interaktivní, neboť více požadavků může být vedeno více proudy, zpoždění v jednom proudu neovlivní proudy ostatní. Pak lze paralelizovat odezvy a i když celková rychlost je stejná, máme dojem lepší interaktivity. V TCP, kde řízení a data sdílí stejné spojení, mohou být řídicí pakety zpožděny za datovými. Pokud jsou řídicí a datové pakety rozděleny do nezávislých proudů, mohou být řídicí informace doručeny nezávisle na datech.

Ochrana navazování asociace a lepší ukončení

Problém při navazování spojení TCP nastává, pokud klient útočí na server následujícím způsobem: vytvoří IP paket s falešnou adresou zdroje, a pak zaplavuje server s TPC SYN pakety. Server alokuje prostředky při přijetí každého SYN paketu, až do vyčerpání (tzv. *Denial of Service* (DoS) útok).

SCTP má proti takovému útoku ochranu. Klient mající zájem o vytvoření asociace nejprve pošle serveru INIT paket, server odpoví INIT-ACK paketem, který obsahuje *cookie* (zašifrované údaje identifikující žádanou asociaci). Klient odpoví COOKIE-ECHO paketem, který obsahuje *cookie* odeslané serverem. V tomto bodě server alokuje prostředky pro spojení a posílá klientovi COOKIE-ACK paket.

TCP spojení umožňuje při ukončení spojení z jedné strany stále touto stranou přijímat data (polo-uzavřený stav). Vývojáři SCTP se rozhodli toto odstranit a nahradili čistší

ukončovací sekvencí. Pokud protějšek ukončí spojení, oba konce jsou nuceni ukončit asociaci a žádná data již nemohou být posílana.

Volitelné neuspořádané doručeni dat

TCP garantuje správné doručení dat a doručení v pořadí. UDP negarantuje pořadí ani správnost. Zprávy v SCTP jsou přenášeny spolehlivě, ale ne nutně v pořadí. Toho může být využito v aplikacích, v kterých jsou různé požadavky nezávislé a na jejich pořadí nezáleží.

8 Podpora různých OS

SCTP je implementován do následujících operačních systémů:

- Linux kernel 2.4/2.6
  http://sourceforge.net/projects/lksctp

- Sun Solaris s externím patchem KAME
  http://playground.sun.com/sctp/

- QNX Neutrino OS

- AIX Verze 5

- FreeBSD/NetBSD/OpenBSD

- Microsoft Windows 2000 a XP
  http://www.sctp.be/sctplib/

Více na http://www.sctp.org/implementations.html

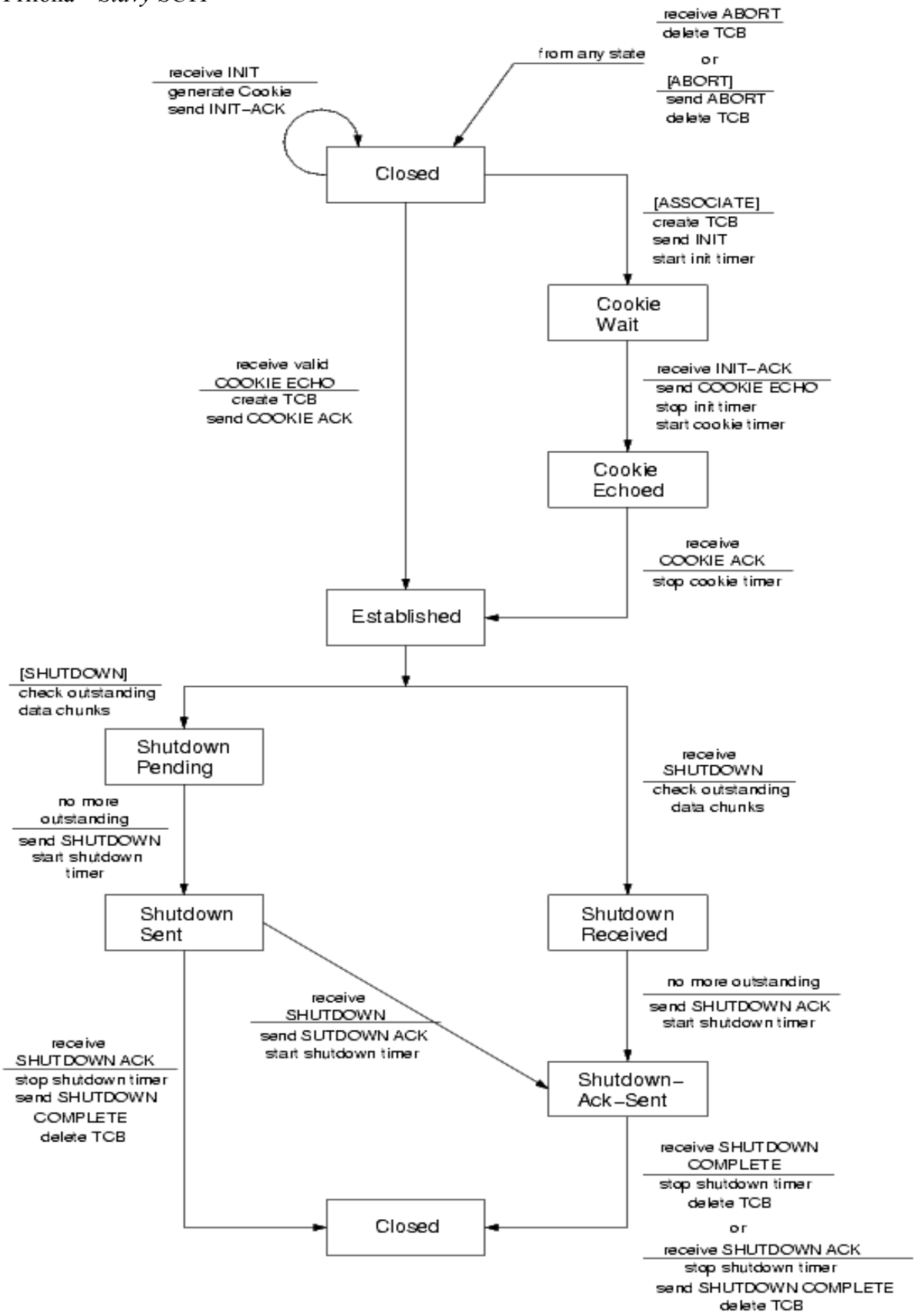
9 RFC související se SCTP

RFC 2960 Stream Control Transmission Protocol.
RFC 3257 Stream Control Transmission Protocol Applicability Statement.
RFC 3286 An Introduction to the Stream Control Transmission Protocol (SCTP).
RFC 3309 Stream Control Transmission Protocol (SCTP) Checksum Change.
RFC 3436 Transport Layer Security over Stream Control Transmission Protocol.
RFC 3554 On the Use of Stream Control Transmission Protocol (SCTP) with IPsec.
RFC 3758 Stream Control Transmission Protocol (SCTP) Partial Reliability Extension.
RFC 3873 Stream Control Transmission Protocol (SCTP) Management Information Base (MIB).
RFC 4460 Stream Control Transmission Protocol (SCTP) Specification Errata and Issues.

10 Použité zdroje

http://rfc.net/rfc2960.html
http://tdrwww.exp-math.uni-essen.de/inhalt/forschung/sctp_fb/
http://www-128.ibm.com/developerworks/linux/library/l-sctp/?ca=dgr-lnxw07SCTP

Příloha – *Stavy SCTP*



Created by Andreas Jungmaier
with XFig