ทำความรู้จักกับโพรโทคอล Stream Control Transmission Protocol

Introduction to Stream Control Transmission Protocol

พงษ์พิสิฐ วุฒิดิษฐโชติ (Pongpisit Wuttidittachotti)*

บทคัดย่อ

โพรโทคอล SCTP (Stream Control Transmission Protocol) เป็นโพรโทคอลในชั้นทรานสปอร์ตที่พัฒนามาจากพื้นฐานการทำงานของ TCP และ UDP ที่มีการใช้งานมาอย่างยาวนาน โดยมีการเพิ่มความสามารถต่าง ๆ ลงไป เช่น Multi-homing คือการส่งข้อมูลหลาย ๆ อินเตอร์เฟซพร้อม ๆ กัน Multi-streaming คือการส่งแยกข้อมูลที่จะส่งออกเป็นหลาย ๆ สตรีมพร้อม ๆ กัน ฯลฯ ซึ่งทำให้ประสิทธิภาพของ SCTP นั้นสูงกว่า TCP และ UDP โดย SCTP ได้รับความนิยมมีการนำไปใช้งานอย่างกว้างขวางในช่วงหลายปีหลังนี้ ดังนั้นในบทความนี้ได้มีการกล่าวถึงโพรโทคอล SCTP ในด้านต่าง ๆ รวมทั้งมีคุณสมบัติใดบ้างที่เหนือกว่า TCP และ UDP

คำสำคัญ: SCTP มัลติโฮมมิ่ง มัลติสตรีมมิ่ง ชั้นทรานสปอร์ต

Abstract

SCTP (Stream Control Transmission Protocol) is a protocol in transport layer. It is developed based on TCP and UDP that have been used for many decades. Addition features are added to SCTP, for example, multi-homing which enables concurrent transfers of data through multiple interfaces and multi-streaming which divides a single data stream to multiple data streams. Therefore, SCTP has a higher performance than TCP and UDP, and is widely used in various environments in the last several years. This paper presents the overview of SCTP in many aspects and indicates which features of those of SCTP are better than TCP and UDP.

Keywords: SCTP, Multi-homing, Multi-streaming, Transport layer.

1. บทนำ

การใช้งานระบบเครือข่ายนั้นมีมาอย่างยาวนาน และได้รับความนิยมมากขึ้นเรื่อย ๆ ดังจะเห็นได้ว่าในชีวิตประจำวันของเรานั้นมีความเกี่ยวข้องกับการใช้งานระบบเครือข่ายไม่ทางตรงก็ทางอ้อม โดยจุดเริ่มต้นของการสื่อสารที่เป็นที่นิยมในปัจจุบันคือเครือข่ายอินเทอร์เน็ต ที่มีพื้นฐานมาจากการสื่อสารด้วยเทคโนโลยี IP (Internet Protocol) [1] จากนั้นได้มีการนำเทคโนโลยี IP ที่ใช้งานในอินเทอร์เน็ตนั้นไปใช้งานในวงกว้างมากขึ้น เช่น ในวงการโทรคมนาคมนั้นมีการใช้ Voice over IP (VoIP) เข้ามาทดแทนระบบโทรศัพท์แบบดั้งเดิม (Plain old telephone service: POTS) โดย Internet Engineering Task Force (IETF) ได้กำหนดสถาปัตยกรรมสำหรับการขนส่งด้วยระบบสัญญาณ (SIGnaling TRANsport: SIGTRAN) [2] โดยสถาปัตยกรรม SIGTRAN นี้ถูกออกแบบมาเพื่อรองรับการส่งสัญญาณโทรศัพท์ในสภาพแวดล้อมที่เครือข่ายแบบดั้งเดิม (Legacy networks) และเครือข่าย IP (IP-networks) ทำงานร่วมกัน แต่เนื่องด้วยสัญญาณโทรศัพท์มักจะมีความต้องการในการมีให้ใช้งาน (Availability) สูง และเวลาที่ใช้ในการนำส่งสัญญาณเสียงน้อยกว่าทราฟิก (Traffic) ชนิดอื่น ๆ เช่น ทราฟิกขนาดใหญ่ (Bulk traffic) จึงมีความจำเป็นที่จะต้องมีโพรโทคอลชั้นทรานสปอร์ต (Transport protocol) หรือชั้นนำส่งข้อมูลตัวใหม่ขึ้นมา โดยมีการกำหนดรายละเอียดของโพรโทคอลตัวใหม่ในระหว่างการกำหนดมาตรฐานของสถาปัตยกรรม SIGTRAN เช่น ความต้องการในเรื่องความสามารถในการมีให้ใช้งาน (Availability) ของโทรศัพท์นั้นสูงถึง 99.9998% [2] ซึ่งหมายความว่า ในหนึ่งปีระบบสามารถหยุดทำงานได้เพียงสูงสุด 10 นาที นอกจากนี้การส่งสัญญาณโทรศัพท์ยังมีข้อกำหนดที่เข้มงวดในเรื่องของเวลา โพรโทคอลชั้นทรานสปอร์ตในขณะนั้น คือ User Datagram Protocol (UDP) (RFC 768) [3] และ Transmission Control Protocol (TCP) (RFC 793) [4] ซึ่งมีการใช้งานมา

* ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

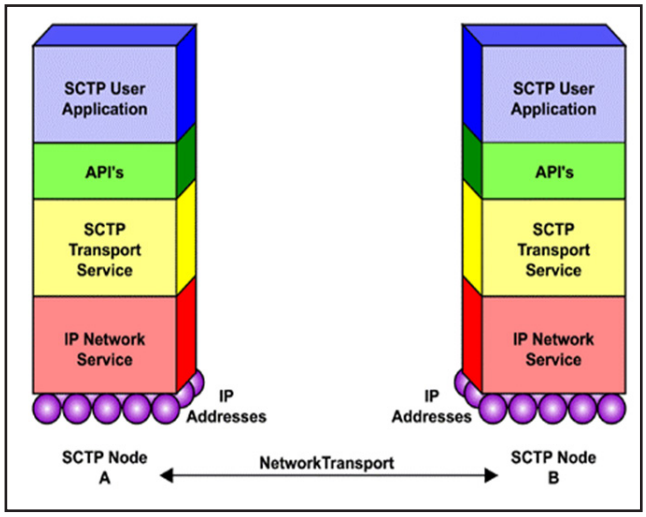
อย่างยาวนานตั้งแต่ปี ค.ศ. 1980 และค.ศ. 1981 ตามลำดับ ยังมีความสามารถไม่เพียงพอต่อความต้องการดังกล่าว จึงมีการพัฒนาโพรโทคอลตัวใหม่ที่ชื่อ Stream Control Transmission Protocol (SCTP) (RFC 4960) [5] ที่มีคุณลักษณะต่าง ๆ ที่ดีขึ้นในการสนับสนุนการนำส่งสัญญาณ เช่น SCTP ถูกออกแบบมาให้มีประสิทธิภาพให้ทนทาน (Robust) ต่อการที่ลิงค์เสียหาย (Link failures) และช่วยบรรเทาปัญหาเกี่ยวกับเรื่องการให้บริการแบบรับส่งข้อมูลภายในเวลาที่กำหนด (Timeliness) ของ TCP ที่มีในปัจจุบัน นอกเหนือจากแอพพลิชันระบบสัญญาณ (Signaling applications) เรายังสามารถใช้ประโยชน์จากฟังก์ชันต่าง ๆ ของ SCTP ได้ โดย IETF ได้ตัดสินใจที่จะสร้างมาตรฐานของ SCTP สำหรับการใช้งานอินเทอร์เน็ตทั่วไปด้วย ซึ่งทำให้ SCTP นั้นมีความที่จำเป็นในการใช้ทรัพยากรร่วมกันกับผู้ใช้เครือข่าย TCP อย่างเป็นธรรม โดย SCTP จะสามารถทำงานร่วมกับ TCP (TCP-friendly) ได้ ดังนั้น IETF จึงตัดสินใจที่จะรวมกลไกต่าง ๆ ที่ทำงานเหมือน TCP (TCP-like mechanisms) เข้าไปเป็นส่วนหนึ่งของ SCTP ด้วย ในบทความนี้จึงได้นำเสนอเรื่องราวเกี่ยวกับโพรโทคอล SCTP ในแง่มุมต่าง ๆ กัน

2. โพรโทคอล SCTP

2.1 ภาพรวมของ SCTP

ชั้นให้บริการทรานสปอร์ตของ SCTP (SCTP transport service layer) แสดงในภาพที่ 1 [6] อยู่ในตำแหน่งระหว่างแอพพลิเคชันของผู้ใช้ที่ใช้ SCTP (SCTP user application layer) และบริการเครือข่าย (Network service layer) ที่กำลังใช้อยู่ เนื่องด้วย SCTP นั้นจะขึ้นอยู่กับการอินเตอร์เฟซกับปลายทางหรือเรียกว่า endpoint ที่เป็น SCTP (SCTP end-points) 2 จุด โดยมีส่วนที่ติดต่อกับการเขียนโปรแกรมแอพพลิเคชันที่ชัดเจน (APIs: application programming interfaces) ที่ทำงานในระหว่างชั้น SCTP transport service และชั้น SCTP user application นอกจากนี้แต่ละ endpoint ยังสามารถมีได้หลายหมายเลข IP address

SCTP สามารถใช้งานได้หลาย ๆ เส้นทาง (Multiple paths) และหลาย ๆ สตรีม (Multiple streams) พร้อมกันเพื่อส่งข้อความข้ามระหว่าง 2 endpoint โดยใน SCTP ข้อมูลจะถูกส่งผ่านระหว่าง endpoint โดยผ่านการเชื่อมต่อที่เรียกว่า "association" โดยที่ association นั้นจะเริ่มต้นการทำงานด้วย "การเริ่มต้น (initiation)" และมีการบำรุงรักษาจนกว่าข้อมูลทั้งหมดได้ถูกส่งไปและได้รับเรียบร้อยแล้ว เมื่อข้อมูลทั้งหมดนั้นได้รับแล้ว association นั้นก็จะสิ้นสุดผ่าน "การปิด (shutdown)"



ภาพที่ 1 *ตำแหน่งที่อยู่ของ SCTP ใน IP Stack*

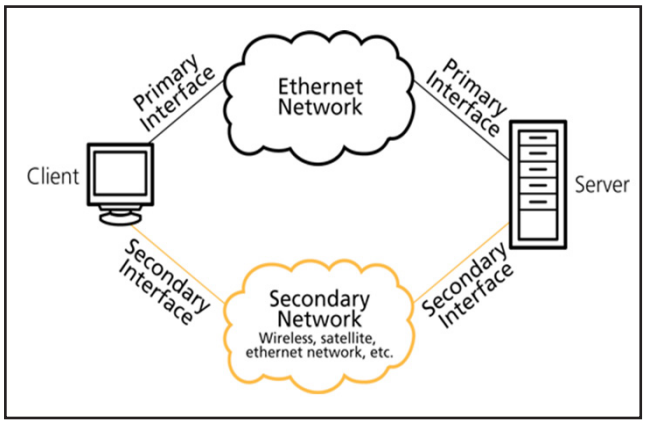
ภายใน SCTP ข้อมูลของผู้ใช้และข่าวสารเกี่ยวกับการควบคุม (Control information) จะถูกประกอบอยู่ในชังค์ (Chunk) ซึ่งอาจจะมีมากกว่า 1 ชังค์ โดยชังค์คือหน่วยของข้อมูล (Unit of information) ที่ใช้ในแพ็คเก็ต SCTP โดยที่ชังค์และส่วนหัวที่ใช้ร่วมกัน (Common Header) นั้นจะประกอบด้วยหน่วยข้อมูลโพรโทคอล (Protocol data unit: PDU) หรือเรียกว่า "แพ็คเก็ต SCTP" (SCTP packet) ในแพ็คเก็ต SCTP นั้นมีชังค์ของข้อมูล (Data chunks) และชังค์ของการควบคุม (Control chunks) โดย SCTP ได้จัดเตรียมการส่งข้อความแบบเรียงลำดับ (Ordered message delivery) ภายใน "สตรีม SCTP" (SCTP streams) และการสนับสนุนเครือข่ายที่ทนทานต่อความบกพร่อง (Fault tolerance) ในสภาพแวดล้อมของ multi-homing (ดูในหัวข้อที่ 2.1.1)

SCTP ถูกพัฒนาขึ้นเพื่อรองรับการส่งข้อมูลที่ไม่เหมาะที่จะส่งด้วย TCP โดย IETF-SIGTRAN ซึ่งเป็นกลุ่มการทำงานเพื่อแก้ปัญหาการขนส่งของสัญญาณสำหรับการสื่อสารด้วยแพ็คเก็ตบนระบบโทรศัพท์บนเครือข่าย IP (Packet-based PSTN signaling over IP Networks) โดยได้เผยแพร่ออกสู่สาธารณะชนเมื่อเดือนตุลาคม ค.ศ.2000 ใน RFC2960 [5] โดย SCTP มีลักษณะการทำงานที่คล้ายคลึงกับ TCP แต่ได้เพิ่มคุณสมบัติหลายประการเพื่อแก้ปัญหาต่าง ๆ ที่พบใน

TCP ซึ่งคุณสมบัติเหล่านั้นได้แก่

2.1.1 Multi-homing

SCTP นั้นออกแบบมาเพื่อจัดการกับระบบสัญญาณในระบบโทรคมนาคมบน IP ซึ่งเป็นระบบที่ต้องมีความล่าช้าของสัญญาณ (Delay) ต่ำเป็นอย่างมาก ซึ่งนับหน่วยเป็นมิลลิวินาที Multi-homing (ภาพที่ 2 [7]) อนุญาตให้ในเครื่องคอมพิวเตอร์ 1 เครื่องมีได้หลายๆ อินเตอร์เฟซ (Multiple interfaces) สำหรับการทำเส้นทางสำรอง (Redundant link) เพื่อที่จะใช้ทดแทนในกรณีที่เส้นทางหลัก (Primary path) เสียหาย โดยไม่ต้องรอเป็นเวลานาน การทำงานของ SCTP จะมีอินเตอร์เฟซหลักที่ใช้งานเป็นเส้นทางหลักซึ่งเรียกว่า primary path และอินเตอร์เฟซที่เหลือจะเป็นเส้นทางที่สองซึ่งเรียกว่า secondary path หรือ alternative paths ในกรณีที่มีหลายๆ อินเตอร์เฟซ ถ้าเส้นทางของ primary path เสียหายไม่สามารถใช้งานได้ไม่ว่าจะด้วยเหตุผลใดก็ตาม secondary path ก็จะถูกใช้งานแทน

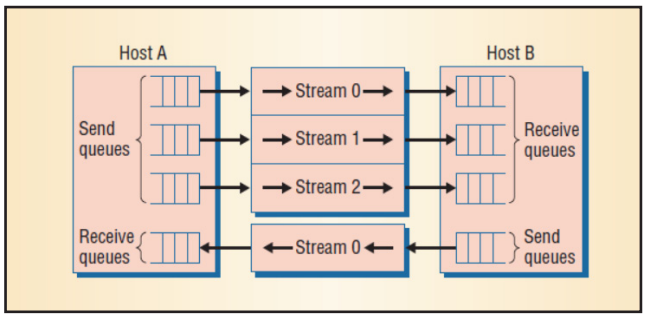


ภาพที่ 2 *Multi-homing*

เมื่อ primary path สามารถใช้งานได้อีกครั้ง การสื่อสารก็จะกลับมาใช้เส้นทางหลัก โดยที่แอพพลิเคชันไม่จำเป็นต้องสนใจกับประเด็นนี้ ในขณะที่สร้างการเชื่อมต่อ (Connection) อินเตอร์เฟซที่เป็น primary และ secondary จะถูกตรวจสอบและเฝ้าดูอยู่ตลอดเวลา โดยใช้ขบวนการของ heartbeat/heartbeat acknowledgement ซึ่งจะทำหน้าที่ตรวจสอบแอดเดรส (Validates addresses), และดูแลรักษาการคำนวณค่า Round Trip Time (RTT) สำหรับแต่ละแอดเดรส โดยค่า RTT แสดงให้เห็นว่า primary path นั้นทำงานช้ากว่า secondary path หรือไม่ และอนุญาตให้การสื่อสารย้ายไปยังอินเตอร์เฟซที่เป็น secondary ได้

2.1.2 Multi-streaming

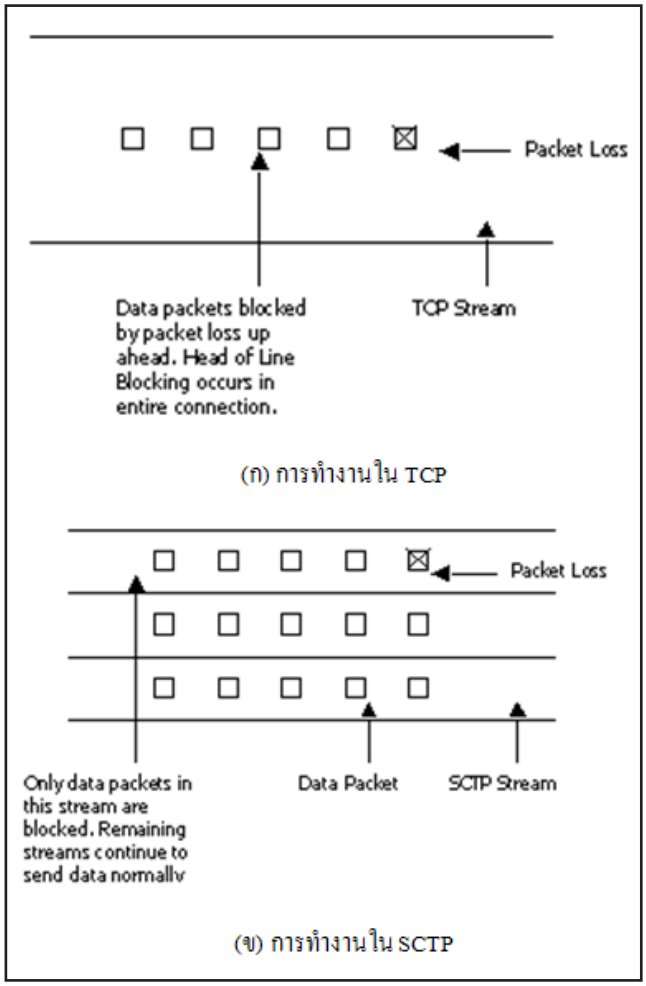
การทำงานของ TCP ใน 1 connection จะมีได้เพียงแค่ data stream เดียวเท่านั้น ทำให้ข้อมูลทั้งหมดต้องผ่านไปบนสตรีมนั้นเท่านั้น ส่วนการทำงานของ SCTP อนุญาตให้ใน 1 connection หรือ 1 association มีหลายสตรีมพร้อมกันได้ (Multiple simultaneous data streams) (ภาพที่ 3 [8]) ซึ่งเรียกวิธีการนี้ว่า Multi-streaming [6] แต่ละ message ที่ส่ง data stream สามารถที่จะมีปลายทางที่แตกต่างกันได้ อย่างไรก็ตาม แต่ละ message ก็ต้องดูแลรักษาขอบเขตของข้อความ (Message boundaries) ของตนเอง เช่น ระบบไม่สามารถส่งส่วนที่เหมือนกันของข้อความไปยังสตรีมอื่นๆ ได้ หนึ่งข้อความต้องส่งผ่านไปยัง 1 สตรีมเท่านั้น



ภาพที่ 3 *Multi-streaming*

ใน TCP เมื่อมีการรันระบบการส่งข้อมูลแบบเรียงลำดับ (Ordered data delivery system) ถ้ามีแพ็คเก็ตใดแพ็คเก็ตหนึ่งผิดลำดับไป (Out of order) หรือหายไป จะทำให้สตรีมนั้นถูกบล็อก (Block) (ภาพที่ 4-ก [9]) เพื่อจะได้ทำการแก้ปัญหาของลำดับข้อมูล ซึ่งเรียกว่า “Head-of-Line Blocking” ใน SCTP การใช้งาน multi-streams (ภาพที่ 4-ข [9]) จะมีเพียงแค่สตรีมที่ได้รับผลกระทบเท่านั้นที่จะถูกบล็อก ส่วนสตรีมอื่นๆ ยังคงสามารถใช้ส่งข้อมูลได้ปรกติ

โดยการใช้ multi-streams กับ SCTP นั้น ประเด็นเกี่ยวกับเว็บเบราเซอร์ที่มีเพียงแค่ความสามารถในการจัดการกับการเชื่อมต่อ 2 การเชื่อมต่อพร้อมกันได้ถูกกำจัดออกไป ไคลเอ็นต์หรือเว็บเซิร์ฟเวอร์สามารถเปิดสตรีมเพิ่มเติมขึ้นมา และส่งภาพ ข้อความ ฯลฯ ผ่านแต่ละสตรีม โดยลดค่าเวลาหน่วง (Latency) โดยรวม นอกจากนี้ยังสามารถลดค่าใช้จ่าย (Overhead) ที่เซิร์ฟเวอร์ซึ่งมักจะมีการเชื่อมต่อที่แบ่งแยกเป็นจำนานมากเพื่อต้องการที่จะตอบสนองการร้องขอ



ภาพที่ 4 *Head-of-Line Blocking*

สตรีมมิ่งของ SCTP ยังจะช่วยให้มีโครงสร้างพื้นฐาน (Infrastructures) ที่เชื่อมต่อกับวิธีการของการสื่อสารที่หลากหลายพร้อมๆ กันได้ เช่น ในภาพที่ 3 สตรีมที่ 1 อาจจะเป็นการสื่อสารด้วยเสียง สตรีมที่ 2 อาจจะเป็นการส่งวิดีโอ สตรีมที่ 3 อาจเป็นโปรแกรมที่ใช้ร่วมกัน และอื่นๆ ต่อไป โดยการเปิดใช้ SCTP เพื่อตอบสนองความต้องการในอนาคตของกลุ่ม SIGTRAN ทำให้นักพัฒนาสามารถทำงานเชื่อมต่อกับ SCTP ได้ง่ายขึ้น โดยสิ่งที่นักพัฒนาต้องทำคือเพียงแค่เกี่ยวข้องกับสตรีมที่เหมาะสมที่ใช้ส่งข่าวสารไปยังแอพพลิเคชันเท่านั้น

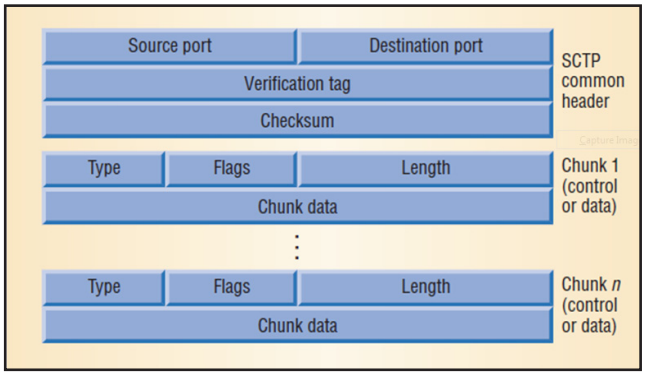
2.2 รูปแบบของแพ็คเก็ต SCTP

ส่วนนี้จะอธิบายถึงรูปแบบของแพ็คเก็ต SCTP (SCTP packet format) ซึ่งรายละเอียดแสดงดังภาพที่ 5 [8]

ส่วนหัว (Common header) ประกอบด้วย [6]:

- แอดเดรสของพอร์ตที่ต้นทาง (Source port address)
- แอดเดรสของพอร์ตที่ปลายทาง (Destination port address)
- แท็กการตรวจสอบ (Verification tag)
- การตรวจสอบของแพ็คเก็ตทั้งหมด (The checksum of the entire packet)

หมายเลขพอร์ตต้นทาง (Source port) จะถูกใช้โดย endpoint ที่ได้รับแพ็คเก็ตเพื่อระบุให้ทราบว่าแพ็คเก็ต SCTP นั้นขึ้นอยู่กับ association ใด หมายเลขพอร์ตปลายทาง (Destination port) คือพอร์ตของผู้รับ SCTP ที่ต้องส่งแพ็คเก็ตไป แต่ละ endpoint จะกำหนดแท็กการตรวจสอบ (มีค่า 32 บิต) ที่ระบุ association โดยค่าการตรวจสอบ (Checksum) ทำหน้าที่เป็นเครื่องมือตรวจสอบความสมบูรณ์ของข้อมูลสำหรับแต่ละแพ็คเก็ต SCTP



ภาพที่ 5 *รูปแบบของแพ็คเก็ต SCTP*

ฟิล์ดภายในชังค์สามารถอธิบายได้ดังนี้

- ฟิล์ดชนิดของชังค์ (Chunktype): ระบุชนิดของฟิล์ดที่ถูกส่ง
- ฟิล์ดแฟล็กของชังค์ (Chunk flag): ระบุว่าบิตใดบ้างจะถูกนำมาใช้ใน association
- ความยาวของชังค์ (Chunk length): ระบุขนาดของชังค์ทั้งหมดมีค่ากี่ไบต์
- ข้อมูลของชังค์ (Chunk data): รวมข้อมูลจริง (Actual data payload) ของชังค์ทั้งหมด

ตามที่ระบุไว้ในภาพที่ 5 มีสมาชิก N ชังค์ (จำนวนของชังค์) ที่ระบุไว้ในแพ็คเก็ต SCTP 1 แพ็คเก็ต จำนวน N จะถูกกำหนดโดยขนาดของหน่วยส่งสูงสุด (MTU: maximum transmission unit) ที่ส่งผ่านได้เส้นทางที่ใช้ส่งนั้น ใน SCTP อนุญาตให้มีการมัลติเพล็กซ์หลาย ๆ ชังค์ในหนึ่งแพ็คเก็ตที่มีขนาดเท่ากับความจุเต็ม MTU (Chunks are multiplexed in one packet to full MTU capacity) โดยมีข้อยกเว้นสำหรับชังค์ของการเริ่มต้น (INIT) และชังค์ตอบกลับการเริ่มต้น

(INIT ACK) เราสามารถแบ่งชังค์ออกได้เป็น 14 ชนิด ประกอบไปด้วย ชังค์ของข้อมูลและชังค์สำหรับควบคุมอีก 13 ชนิด ชังค์ของข้อมูลนั้นประกอบไปด้วยข้อมูลที่เกิดขึ้นจริง ความหมายและพารามิเตอร์ของชังค์ควบคุมได้สรุปไว้ใน [6]

2.3 ฟังก์ชัน SCTP: ติดตามขบวนการการส่ง (Tracking the Transmission Process) [6]

ส่วนนี้จะอธิบายขั้นตอนการขนส่งข้อมูลซึ่งประกอบด้วย 3 ขั้นตอนหลักของ SCTP association: การเริ่มต้น (initiation) (ภาพที่ 6 [6]) การส่งข้อมูล (data transmission) (ภาพที่ 7 [6]) และการปิดหรือการสิ้นสุด (shutdown) (ภาพที่ 8 [6]) โดยในที่นี้ endpoint ที่เริ่มต้นสร้าง association จะเรียกว่า “โหนด A” ส่วน endpoint ที่ได้รับการร้องขอการจัดตั้ง association จะเรียกว่า “โหนด B”

หมายเหตุ: ชังค์ยกเลิก (ABORT) และชังค์ข้อผิดพลาด (ERROR) สามารถสร้างโดย endpoint ฝั่งใดก็ได้และสามารถส่งในเวลาใดก็ได้ในระหว่างช่วง association ซึ่งทั้งสองกรณีจะทำให้ endpoint ปิดทันที

2.3.1 ขั้นตอนเริ่มต้น Association (Association Initiation)

1. โหนด A สร้างชังค์ INIT และส่งไปยังโหนด B ในขณะเดียวกัน โหนด A จะเริ่มใช้ตัวนับเวลา INIT (INIT timer)

2. ถ้าโหนด B ประสงค์ที่จะยอมรับ association นี้ก็สร้างชังค์ ACK INIT พร้อมด้วยคุกกี้ จากนั้นจะส่งชังค์ ACK INIT พร้อมกับคุกกี้กลับไปยังโหนด A

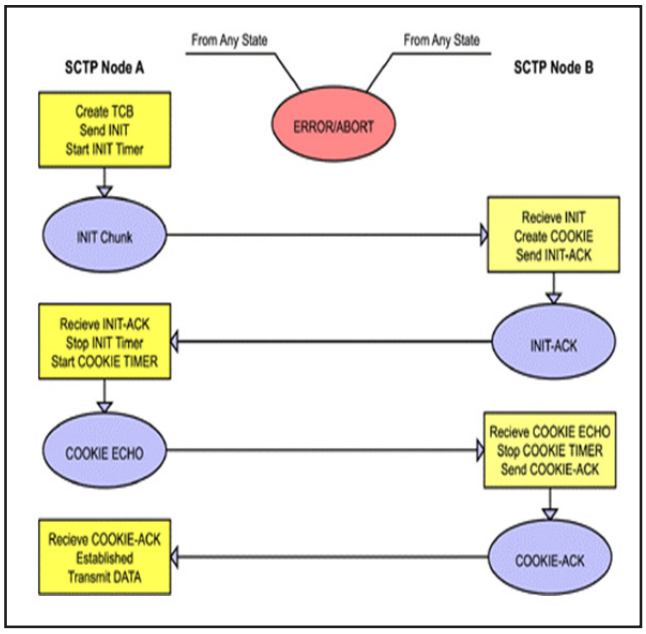
3. โหนด A ได้รับชังค์ ACK INIT และหยุดตัวจับเวลา INIT จากนั้น โหนด A จะสร้างชังค์ COOKIE ECHO ส่งกลับยังโหนด B และโหนด A จะเริ่มตัวจับเวลาคุกกี้ (cookie timer) ในขณะที่ส่งไปนั้น ชังค์ข้อมูลอาจจะรวมไปในแพ็คเก็ตนี้ด้วย

4. โหนด B ตรวจสอบความถูกต้องของคุกกี้ หลังจากการตรวจสอบก็จะส่ง COOKIE ACK กลับไปยังโหนด A

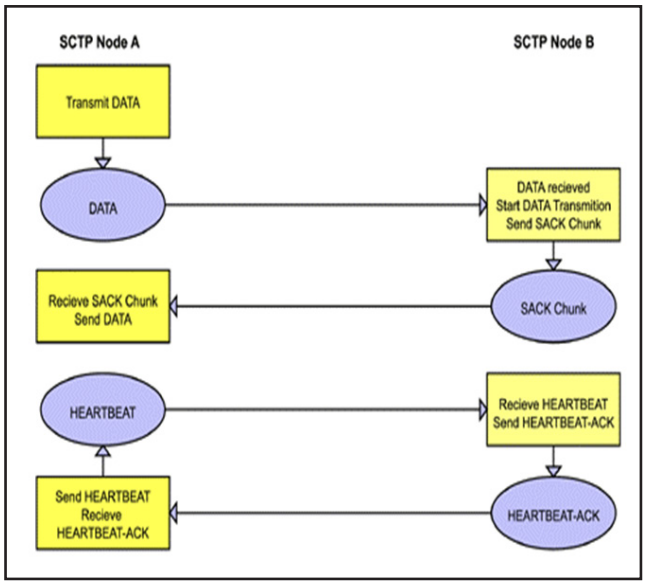
5. โหนด A ได้รับ COOKIE ACK และเข้าสู่ขั้นตอนต่อไปของการส่งข้อมูล

2.3.2 ขั้นตอนการส่งข้อมูลใน Association (Association Data Transmission)

โหนด A และโหนด B สามารถส่งข้อมูลได้อย่างต่อเนื่อง ตลอดกระบวนการการส่งข้อมูล ชังค์ HEARTBEAT และ



ภาพที่ 6 *การเริ่มต้น association ใน SCTP โดยใช้วิธีการแบบ Four-Way Handshake*



ภาพที่ 7 *ขบวนการการส่งข้อมูลใน SCTP*

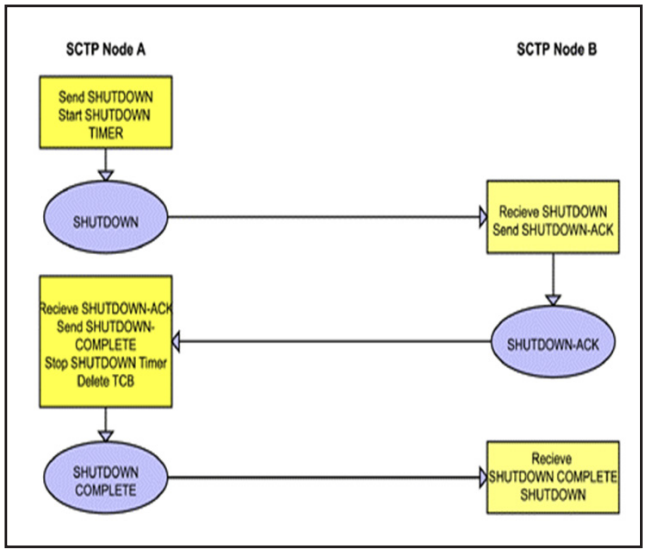
ชังค์ HEARTBEAT ACK จะถูกแลกเปลี่ยนระหว่างโหนดในทุกๆ ช่วงเวลาที่แน่นอนซึ่งควบคุมโดยตัวจับเวลาการเต้นของหัวใจ (Heartbeat timer) โดยชังค์เหล่านี้ใช้เพื่อทดสอบการเชื่อมต่อของ endpoint เพื่อที่จะรักษาความถูกต้องของการส่งข้อมูล

6. โหนด A และโหนด B แลกเปลี่ยนชังค์ข้อมูล

7. หลังจากที่ได้รับแต่ละชังค์ข้อมูล ปลายทางที่ได้รับจะตอบกลับด้วยชังค์ SACK (Selective Acknowledgement)

เพื่อแจ้งให้ผู้รับทราบ

8. ข้อมูลจะถูกส่งจนกว่าจะมีปลายทางฝั่งใดฝั่งหนึ่งตัดสินใจที่จะปิด association โดยการส่งชังค์ SHUTDOWN รวมไปเป็นส่วนหนึ่งของแพ็คเก็ตนั้น



ภาพที่ 8 *ขบวนการการปิด association ใน SCTP*

2.3.3 การปิดหรือการสิ้นสุด Association (Association Shutdown)

ชังค์ SHUTDOWN สามารถที่จะส่งโดยโหนดใดก็ได้ระหว่างสองโหนดที่สื่อสารกันอยู่ ในที่นี้ พิจารณาที่โหนด A ว่าเป็นผู้เริ่มต้นขบวนการการปิด (Shutdown process)

9. โหนด A ส่งชังค์ SHUTDOWN ไปยังโหนด B และเริ่มตัวนับเวลา shutdown (Shutdown timer)

10. โหนด B จะตอบกลับหลังจากได้รับชังค์ SHUTDOWN โดยการสร้างชังค์ SHUTDOWN ACK ส่งกลับไปยังโหนด A

11. เมื่อโหนด A ได้รับ SHUTDOWN ACK และตอบสนองด้วยการปิดตัวนับเวลา shutdown จากนั้นโหนด A จะสร้างชังค์ SHUTDOWN COMPLETE ส่งกลับไปยังโหนด B

ในทั้ง 3 ขั้นตอน คือ การเริ่มต้น การส่งข้อมูลและการปิด จะทำงานผ่าน SCTP API ที่เชื่อมต่อกับชั้นแอพพลิเคชันและชั้นทรานสปอร์ต

โพรโทคอลชั้นทรานสปอร์ตในเครือข่าย IP เช่น TCP และ UDP ได้รับประโยชน์จากซ็อกเก็ต API โดยที่ซ็อกเก็ต API ได้เตรียมการออกแบบที่เป็นมาตรฐานที่สามารถอินเตอร์เฟซข้ามหลากหลายระบบปฏิบัติการและหลากหลายแพลตฟอร์มได้ โดยสามารถดูรายละเอียดของซ็อกเก็ตของ API ที่ใช้ในการแนะนำบริการของ SCTP ในแอพพลิเคชันที่ใช้ TCP และ UDP เป็นพื้นฐานได้ที่ [6]

2.4 ประโยชน์ที่เพิ่มขึ้นของ SCTP เมื่อเทียบกับ TCP

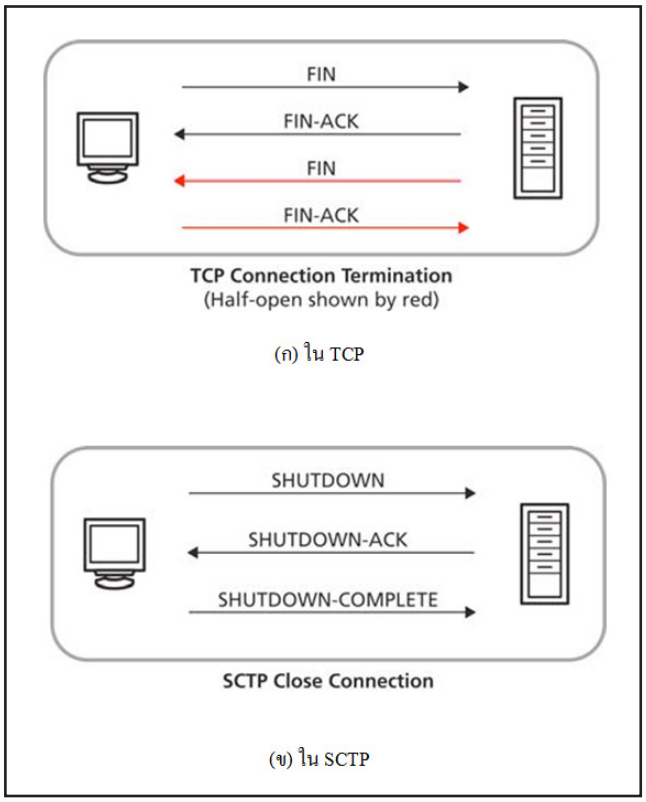
รายละเอียดการเปรียบเทียบระหว่าง TCP UDP และ SCTP นั้น สามารถอ่านเพิ่มเติมได้จาก [8] ในบทความนี้จะนำเสนอส่วนที่สำคัญดังนี้ [7]

2.4.1 การอนุญาตให้การเชื่อมต่อแบบปิดได้ครึ่งหนึ่ง (Allow half-closed connections)

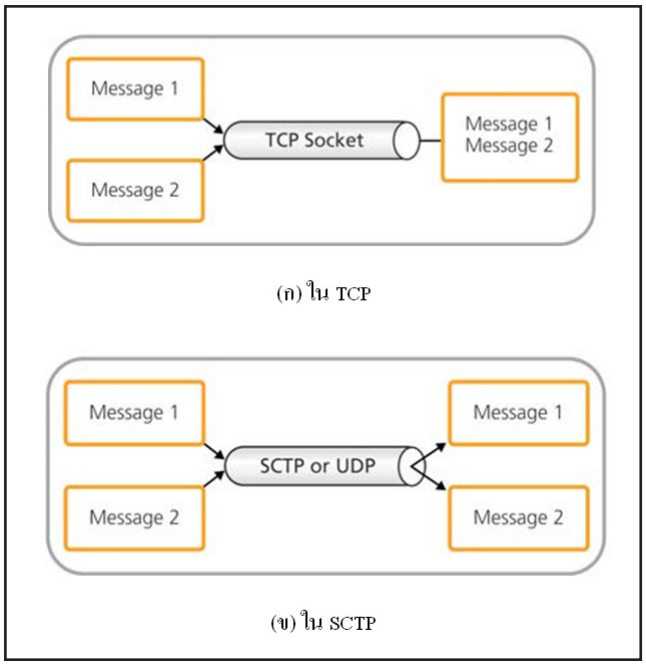
การเชื่อมต่อแบบปิดได้ครึ่งหนึ่งสามารถเกิดขึ้นได้เมื่อด้านหนึ่งของการสนทนาเชื่อว่าการเชื่อมต่อนั้นถูกปิดไปแล้ว แต่ในขณะที่อีกฝั่งหนึ่งเชื่อว่ามันยังคงเปิดอยู่ ใน TCP ใช้คำสั่งที่ต้องการสิ้นสุดในลักษณะ 4 ทิศทาง (four-way) ประกอบด้วยการส่งข้อความ FIN และ FIN-ACK ทั้งสองทิศทาง การเชื่อมต่อแบบเปิดได้ครึ่งหนึ่ง (Half-open) อาจยังคงมีอยู่ได้ ถ้าสองข้อความสุดท้าย คือ FIN และ FIN-ACK ไม่ได้ส่ง (ภาพที่ 9-ก) ทำให้ SCTP กำจัดโอกาสที่จะเกิดเหตุการณ์นี้ขึ้น โดยใช้วิธีการปิดแบบ 3 ทาง (three-way shutdown) (ภาพที่ 9-ข) ซึ่งประกอบด้วยการขอปิดการเชื่อมต่อ (SHUTDOWN) การยืนยันการปิดการเชื่อมต่อ (SHUTDOWN-ACK), และการแสดงปิดการเชื่อมต่อเรียบร้อยแล้ว (SHUTDOWN-COMPLETION) เมื่อขบวนการนี้เริ่มขึ้น ทั้งสองฝ่ายจะยุติการสื่อสารทันที หากต้องการส่งข้อมูลเพิ่มเติมจะต้องทำการสร้างการเชื่อมต่อใหม่อีกครั้ง

2.4.2 การสงวนขอบเขตของข้อความ (Preservation of message boundaries)

หากไคลเอ็นต์ส่งข้อความขนาด 100 ไบต์และขนาด 50 ไบต์ ข่าวสารจะถูกนำเสนอไปยังเซิร์ฟเวอร์กับขอบเขตของข้อความที่สงวนไว้ (Preserved message boundaries) แต่ในการทำงานกับ TCP ข้อความอาจถูกส่ง/รับ โดยข้อความจะมีขนาด 150 ไบต์ ดังตัวอย่างภาพที่ 10-ก ข้อความที่ได้รับจะมีเพียงแค่ 1 ข้อความ ซึ่งการทำแบบนี้จะบังคับให้แอพพลิเคชันต้องแยกข้อความกลับไปยังรูปแบบเดิมที่มาจากต้นทาง แต่ในการทำงานกับ SCTP และ UDP (ภาพที่ 10-ข) ข้อความจะถูกส่งเป็น 100 ไบต์และ 50 ไบต์ ตามลำดับ โดยในการทำงานกับ SCTP และ UDP จะมีการดูแลรักษาขอบเขตของข้อความและแอพพลิเคชันไม่ต้องทำการแยกข้อความนี้ที่ปลายทางอีกครั้ง



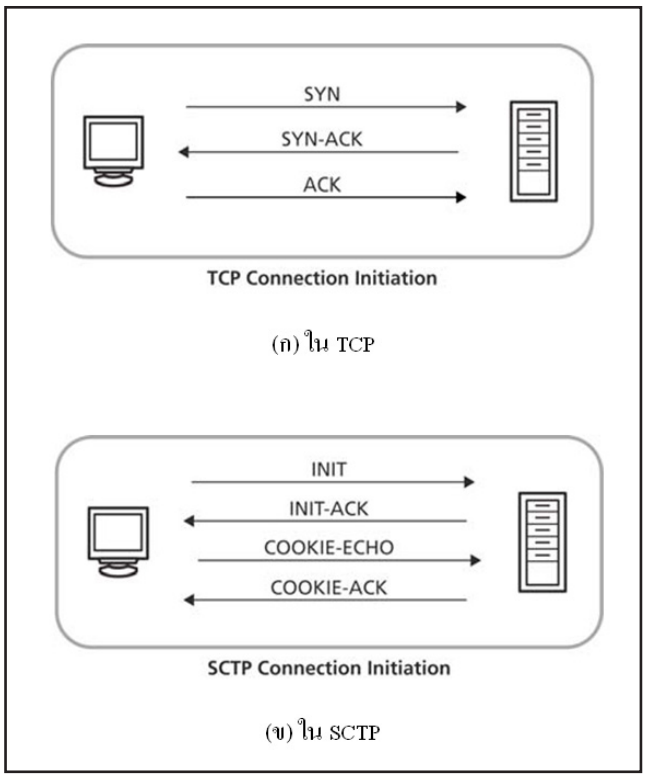
ภาพที่ 9 *การปิดการเชื่อมต่อ*



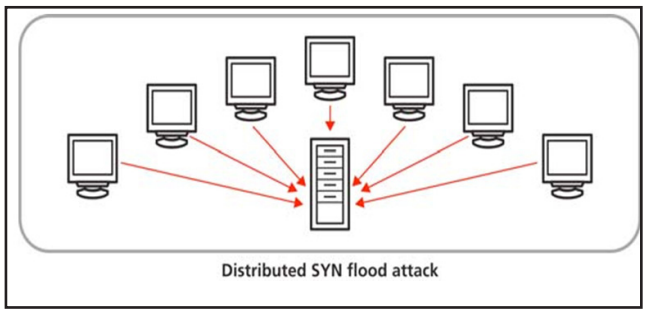
ภาพที่ 10 *การสงวนขอบเขตของข้อความ*

2.4.3 การป้องกันโจมตีด้วยการส่ง SYN เป็นจำนวนมาก (Protect against SYN flooding attacks)

เพื่อให้เข้าใจถึงการโจมตีด้วยการส่ง SYN เป็นจำนวนมาก เราต้องดูที่วิธีการเชื่อมต่อที่เป็นแบบฉบับที่จัดตั้งขึ้น โดยการใช้งาน TCP ไคลเอ็นต์เริ่มสื่อสารโดยการส่งการร้องขอซิงโครไนซ์ หรือแพ็คเก็ต SYN ทางฝั่งเซิร์ฟเวอร์จะตอบสนองด้วยการยอมรับโดยส่งกลับแพ็คเก็ต SYN-ACK และในที่สุดไคลเอ็นต์ก็จะตอบกลับแพ็คเก็ตที่ได้รับ โดยส่งแพ็คเก็ต ACK กลับไปที่เซิร์ฟเวอร์ ซึ่งเราเรียกขบวนการนี้ว่า “การจับมือแบบสามทาง (Three-way handshake)” (ภาพที่ 11-ก) เมื่อทั้งสามขั้นตอนเสร็จสมบูรณ์จะสามารถเริ่มต้นการสื่อสารได้ ในทางตรงกันข้ามกับ SCTP ซึ่งใช้ “การจับมือแบบสี่ทาง (Four-way handshake)” (ภาพที่ 11-ข) แต่อาจจะเริ่มส่งข่าวสารได้ตั้งแต่ขั้นตอนที่สาม โดยการทำงานคือไคลเอ็นต์ที่ใช้ SCTP จะเริ่มการสื่อสารโดยการส่งแพ็คเก็ต INIT เมื่อฝั่งเซิร์ฟเวอร์ได้รับจะตอบกลับไปด้วยแพ็คเก็ต INIT-ACK และ cookie (บริบทเฉพาะที่ระบุการเชื่อมต่อ) ฝั่งไคลเอ็นต์เมื่อได้รับแล้ว จะส่งค่า cookie ของเซิร์ฟเวอร์ที่ได้รับมาตอนแรกกลับไปยังเซิร์ฟเวอร์ นอกจากนี้ฝั่งไคลเอ็นต์ยังสามารถส่งข่าวสารเพิ่มเติมหลังจากการส่ง COOKIE-ECHO ทางฝั่งเซิร์ฟเวอร์เมื่อได้รับ COOKIE-ECHO แล้วก็จะตอบกลับด้วย COOKIE-ACK กลับไปยังฝั่งไคลเอ็นต์



ภาพที่ 11 *การสร้างการเชื่อมต่อ*



ภาพที่ 12 *การโจมตีด้วยการส่ง SYN เป็นจำนวนมาก*

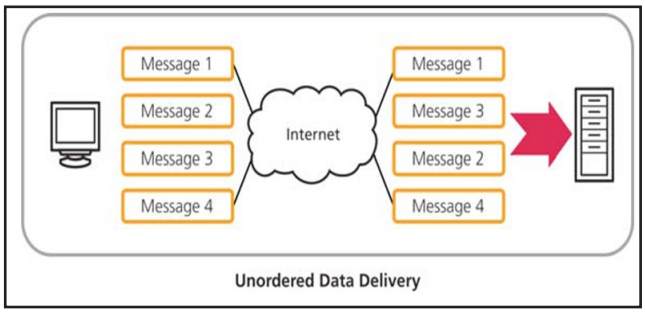
การโจมตีด้วยการส่ง SYN (ภาพที่ 12) เป็นจำนวนมาก เกิดขึ้นเมื่อไคลเอ็นต์ 1 เครื่องหรือหลายๆ เครื่องส่งแพ็คเก็ต SYN ไปยังเซิร์ฟเวอร์ ซึ่งจะทำให้เซิร์ฟเวอร์เป้าหมายนั้นจัดสรรทรัพยากรไว้ให้สำหรับแต่ละการร้องขอและมักจะเกินขีดความสามารถของเซิร์ฟเวอร์ที่จะรับได้ ทำให้เซิร์ฟเวอร์ต้องมีการรีบูตเครื่องหรืออาจมีผลที่เลวร้ายกว่านั้นได้ เมื่อทำงานกับ TCP เซิร์ฟเวอร์จะมีการจัดสรรทรัพยากรให้เรียบร้อยแล้วสำหรับแต่ละการเชื่อมต่อ ภายในการเชื่อมต่อแบบ TCP endpoint ทั้งสองฝั่งต้องทำการจัดสรรพอร์ตชั่วคราว (Ephemeral port) หน่วยความจำ และการประมวลผลของ CPU สำหรับการเชื่อมต่อใหม่แต่ละการเชื่อมต่อ การเชื่อมต่อด้วย TCP เริ่มต้นเมื่อมีการรับแพ็คเก็ต SYN แล้วจะมีการจัดสรรทรัพยากรสำหรับการเชื่อมต่อนั้น ในขณะที่การทำงานด้วย UDP ซึ่งเป็นโพรโทคอลแบบ connection-less ไม่ได้ใช้ขบวนการแบบนี้ ส่วนการใช้ SCTP เซิร์ฟเวอร์จะไม่ทำการจัดสรรทรัพยากรสำหรับการเชื่อมต่อนั้นจนกว่าจะได้รับ COOKIE-ECHO แล้วหลังจากนั้น ไคลเอ็นต์ที่ใช้ SCTP ต้องทำการจัดสรรทรัพยากรที่ใช้เพื่อให้สามารถที่จะส่งข่าวสารได้

2.4.4 การเลือกตอบกลับ (Selective acknowledgements)

ในมาตรฐาน TCP ทุกๆ ข้อความหรือแพ็คเก็ตของข่าวสารต้องมีการพิจารณาโดยต้องมีการยืนยันการได้รับข้อความหรือแพ็คเก็ตนั้น (Acknowledgement) ต้องมีการส่งใหม่ (Resend) เมื่อจำเป็นและต้องถูกประมวลผลในการลำดับ (In the order) ที่แพ็คเก็ตเหล่านั้นถูกส่งมา ส่วนการใช้งาน SCTP นั้น ทางฝั่งผู้รับมีความสามารถในการเลือกตอบกลับสำหรับข้อความที่สูญหายไป (Missing) ไม่เป็นตามลำดับที่ส่งมาจากต้นทาง (Disordered) หรือซ้ำ (Duplicated) เนื่องจากธรรมชาติของการสื่อสารโทรคมนาคม แอพพลิเคชันส่วนใหญ่จะจบลงที่การทิ้งข้อความใดๆ ที่ไม่ซิงโครไนซ์กัน (Unsynchronized) หรือเรียกง่ายๆ ว่าข้อมูลที่ไม่ตรงกันนั้นจะถูกทิ้งไป ดังนั้นจึงมีความจำเป็นที่ต้องมีการทิ้งข้อมูลบางส่วนจากการส่งและข้อมูลที่ได้รับ ซึ่งหมายถึงว่าบางส่วนของคำบางส่วนของวิดีโอ หรือบางชิ้นส่วนของการรีเฟรช (refresh) ไวท์บอร์ดจะถูกละเลยไป แอพพลิเคชันและผู้ใช้อาจสังเกตเห็นเพียงเล็กน้อยในการขาดหายไปของเสียง วิดีโอหรือการรีเฟรช ซึ่งสิ่งนี้ถูกเรียกว่าเวลาหน่วง (Jitter) ในโลกการสื่อสารโทรคมนาคม ผลรวมเวลาหน่วงที่น้อยมากๆ มักเป็นที่ต้องการมากกว่าการกว่าการส่งแพ็คเก็ตใหม่อีกครั้ง และการที่ต้องประมวลผลใหม่อีกครั้งนั้นจะทำผลรวมของเวลาหน่วงมีค่าเพิ่มขึ้นเป็นสองเท่า ซึ่งโดยปรกติแล้ว ผู้ใช้มักสังเกตเห็นความล่าช้านี้ได้อย่างชัดเจน

2.4.5 การนำส่งข้อมูลแบบไม่เรียงลำดับ (Unordered data delivery)

เนื่องจากธรรมชาติของเครือข่าย มีความเป็นไปได้ว่าแพ็คเก็ตทั้งหมดที่ถูกส่งไปอาจเดินทางข้ามเครือข่ายโดยไม่ได้ใช้เส้นทางเดียวกัน อาจมีเวลาล่าช้าเนื่องจากการใช้เส้นทางหนึ่งที่มีความล่าช้ามากกว่าเวลาล่าช้าที่เส้นทางอื่นๆ ฝั่งผู้รับเมื่อได้รับข้อความนั้นอาจได้รับข้อความที่มีลำดับไม่เหมือนกับข้อความเดิมที่ส่งมา การนำส่งข้อมูลแบบไม่เรียงลำดับที่เกิดในเหตุการณ์นี้ จะสามารถแก้ไขปัญหาได้โดยการเรียงลำดับข้อความนั้นใหม่ให้ถูกต้อง ในการใช้งานคุณลักษณะการถ่ายโอนข้อมูลที่เชื่อถือ (Reliable data transfer) ได้ของ TCP ต้องการให้แพ็คเก็ตเหล่านั้นถูกประมวลผลตามลำดับที่จัดส่งมา ถ้ามีแพ็คเก็ตใดหายไปหรือผิดลำดับไป ต้องมีการเรียงลำดับแพ็คเก็ตใหม่ก่อนที่การประมวลผลจะสามารถดำเนินการต่อไปได้



ภาพที่ 13 *การส่งข้อมูลที่ไม่เรียงลำดับ*

ภาพที่ 13 ถ้าเราใช้ TCP เมื่อข้อความที่ 3 ได้รับก่อน การประมวลผลทั้งหมดจะหยุด และรอข้อความที่ 2 เมื่อได้รับข้อความที่ 2 ก็เริ่มการประมวลผล จากนั้น ข้อความที่ 3

จึงจะถูกประมวลผล ในการทำงานของ SCTP อนุญาตให้มีการนำส่งข้อมูลแบบไม่เรียงลำดับและเนื่องจากมีการส่งแบบหลายสตรีม ในกรณีที่เกิดปัญหานี้ จะมีเพียงแค่สตรีมเดียวเท่านั้นที่ได้รับผลกระทบและถูกบล็อกชั่วคราว SCTP จะประมวลผลข้อความในลำดับที่ได้รับข้อความนั้น จะไม่รอการเรียงลำดับของข้อความเหล่านั้นที่ส่งมา ในโหมดการการถ่ายโอนแบบมีความน่าเชื่อถือของ SCTP จะมีหลายๆ โซลูชันดิสก์แบบเครือข่าย (Networked disk solutions) ที่ได้มีการจัดเตรียมบริการเรียงลำดับข้อมูลไว้ ความสามารถของ SCTP เพียงแค่ส่งข้อมูลแบบง่ายๆ เมื่อลดภาระโอเวอร์เฮดที่ไม่จำเป็นในการเรียงลำดับข้อมูลใหม่ของเซิร์ฟเวอร์

2.5 Concurrent Multi-streaming (CMT) SCTP

การทำงานของ SCTP แบบดั้งเดิมนั้น จะส่งข้อมูลโดยใช้อินเตอร์เฟซใดอินเตอร์เฟซหนึ่งในการส่งข้อมูล ซึ่งเรียกว่าเส้นทางหลัก (Primary path) ส่วนเส้นทางอื่นๆ จะเป็นเส้นทางสำรอง (Secondary path) ซึ่งจะมีการใช้งานเมื่อเส้นทางหลักไม่สามารถใช้งานได้ จะเห็นได้ว่าเราใช้เส้นทางสำรองนั้นไม่ได้เต็มประสิทธิภาพ จึงมีแนวความคิดในการใช้เส้นทางทั้งหมดที่มีให้ส่งข้อมูลได้พร้อม ๆ กัน ซึ่งเรียกว่า Concurrent Multi-streaming (CMT) [10] หรือ CMT-SCTP ซึ่งเป็นการพัฒนาต่อยอดความสามารถของ SCTP แบบดั้งเดิม โดย CMT-SCTP จะสามารถทำงานได้ดีในกรณีที่เส้นทาง (Paths) ที่ใช้งานนั้นมีแบนด์วิดท์ ดีเลย์ (Delay) และค่าอัตราความผิดพลาด (Error rates) ที่เหมือนกันหรือใกล้เคียงกัน การที่เส้นทางใดเส้นทางหนึ่งมีคุณสมบัติแตกต่างออกไป ซึ่งจริงๆ แล้วก็เปรียบเสมือนกับกรณีที่เกิดขึ้นจริงในการใช้งาน [11] คือ การที่เราใช้คอมพิวเตอร์เครื่องเดียวเชื่อมต่อกับผู้ให้บริการ (Internet service provider: ISP) หลายๆ รายพร้อมกัน ยังคงเป็นประเด็นที่ต้องได้รับการแก้ไข และยังมีประเด็นอื่นๆ อีกมาก

2.6 การนำไปใช้งานและคุณลักษณะที่สนับสนุน

มีการนำ SCTP (SCTP implementations) (รายละเอียดแสดงดังตารางที่ 1 [11]) ไปใช้งานในระบบหลายๆ ปฏิบัติการ เช่น Linux (mainline kernel 2.6.36), FreeBSD (release kernel 8.12), MacOS X, Windows (SctpDrv) และ Solaris (OpenSolaris 2009.06) รวมถึง sctplib (เป็นไลบารีสำหรับ SCTP และไลบารีสำหรับซ็อกเก็ต API ที่มีชื่อเสียงที่รู้จักเป็นอย่างดีในการนำ SCTP ไปใช้งาน) นอกจากนี้ยังรวมถึงโปรแกรมจำลองการทำงานของเครือข่าย (Network Simulator) เช่น OMNeT++/INET simulation model ซึ่งได้เตรียมฟังก์ชันต่างๆ ของ SCTP ไว้สำหรับการทดสอบ

3. สรุป

การทำงานของโพรโทคอลในชั้นทรานสปอร์ตเป็นสิ่งที่จำเป็นอย่างยิ่งที่มีผลสำเร็จต่อการส่งข้อมูลในระบบเครือข่าย ในบทความนี้ได้นำเสนอตั้งแต่ความเป็นมา โครงสร้างของแพ็คเก็ต SCTP คุณลักษณะต่างๆ ของ SCTP เมื่อเทียบกับ TCP และ UDP และสุดท้ายได้พูดถึง CMT-SCTP ที่สามารถส่งข้อมูลได้หลายๆ อินเตอร์เฟซพร้อมกันๆ ซึ่งคุณสมบัติเหล่านี้ทำให้ SCTP มีประสิทธิภาพสูงกว่า TCP

ตารางที่ 1 *การนำ SCTP ไปใช้งานและคุณลักษณะที่สนับสนุน*

Feature	Linux	FreeBSD	MacOS X	Windows	Solaris	sctplib	OMNeT++
Standard SCTP (RFC 4960)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Explicit congestion notif. (ECN)	Yes	Yes	Yes	Yes	No	No	No
Partial reliability (PR-SCTP)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Chunk authentication (SCTP-AUTH)	Yes	Yes	Yes	Yes	Yes	No	Yes
Dynamic address reconfiguration	Yes	Yes	Yes	Yes	Yes	Experimental	Yes
Packet drop rreporting (PKTDROP)	No	Yes	Yes	Yes	No	No	Yes
SACK immediately	No	Yes	Yes	Yes	No	No	Yes
Advanced stream schedulers	No	In progress	In progress	In progress	No	No	Yes
Stream reset	No	Yes	Yes	Yes	No	No	Yes
Non-revokable SACKs (NR-SACK)	No	Yes	Yes	Yes	No	No	Yes
Potentially failed (PF) path state	No	Yes	Yes	Yes	No	No	No
CMT-SCTP	No	Yes	Yes	Yes	No	No	Yes
CMT/RP-SCTP	No	In progress	In progress	In progress	No	No	Yes
Buffer splitting	No	In progress	In progress	In progress	No	No	Yes
Chunk rescheduling	No	In progress	In progress	In progress	No	No	Yes

และ UDP โดยบทความนี้ช่วยให้ผู้สนใจหรือผู้ศึกษาเกี่ยวกับระบบเครือข่ายได้รู้จักโพรโทคอลใหม่ ซึ่งในที่นี้คือ SCTP ในชั้นทรานสปอร์ต นอกเหนือจาก TCP และ UDP ที่รู้จักและมีการใช้งานมาอย่างยาวนาน ซึ่ง SCTP นี้ถือว่าเป็นแนวโน้มและทิศทางในอนาคตของชั้นทรานสปอร์ตที่จะมีการนำไปใช้งานในเครือข่ายประเภทต่างๆ อย่างกว้างขว้าง ดังจะเห็นได้ว่าหลายๆ ระบบปฏิบัติการหลักในท้องตลาดมีการติดตั้งโพรโทคอล SCTP ให้ใช้งานกันแล้ว

4. เอกสารอ้างอิง

[1] J. Postel. “Internet Protocol.” RFC 791, *Internet Engineering Task Force*, September, 1981.

[2] L. Ong, et al. “Framework architecture for signaling transport.” *RFC 2719*, *Internet Engineering Task Force*, October, 1999.

[3] J. Postel. “User Datagram Protocol.” *RFC 768*, *Internet Engineering Task Force*, August, 1980.

[4] J. Postel. “Transmission Control Protocol.” *RFC 793*, *Internet Engineering Task Force*, September, 1981.

[5] R. Stewart. “Stream Control Transmission Protocol.” *RFC 4960*, *Internet Engineering Task Force*, September, 2007.

[6] The International Engineering Consortium, “Stream Control Trans-mission Protocol (SCTP).” Web ProForum Tutorials.

[7] P. Stalvig. “Introduction to the Stream Control Transmission Pro-tocol (SCTP): The next generation of the Transmission Control Protocol (TCP).” White Paper, F5 Networks, Inc., October, 2007.

[8] A. L. Caro Jr. et al. “SCTP: A Proposed Standard for Robust In-ternet Data Transport”, *IEEE Computer Society*, Vol. 36 , Issue. 11, pp. 56- 63, November, 2003.

[9] R. Stewart and Q. Xie. *Stream Control Transmission Protocol (SCTP): A Reference Guide*. Addison Wesley, 2002.

[10] J. R. Iyengar, P. D. Amer, and R. Stewart. “Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths.” *IEEE/ACM Trans. Net*., Vol. 14, No 5, pp. 951–964, October, 2006.

[11] T. Dreibholz et al. “Stream Control Transmission Protocol: Past, Current, and Future Standardization Activities.” *IEEE Communications Magazine,* Vol. 49, Issue 4, pp. 82–88, April, 2011.