บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 รูปแบบการประมวลผลแบบขนาน

โมเดลของการเขียนโปรแกรมแบบขนาน (parallel programming model) ได้ถูกแบ่งออกเป็น 3 แบบได้แก่

1. การเขียนโปรแกรมแบบขนาน โดยที่ข้อมูลทำงานแบบขนาน (data parallelism) คอมพิวเตอร์ทำงานโดยการสลับข้อมูลแต่ละส่วนผลัดกันเข้าทำงาน

2. การเขียนโปรแกรมแบบขนานโดยที่การควบคุมทำงานแบบขนาน (control parallelism) ผู้เขียนโปรแกรมหรือตัวแปลภาษา (compiler) จะต้องทำการแปลงให้โปรแกรมซึ่งมีลักษณะการทำงานเป็นวงรอบหรือทำซ้ำกลายเป็นทำงานขนานกันไป

3. การเขียนโปรแกรมแบบขนาน โดยใช้ภาษาทำงานแบบขนาน (structure language parallelism) คำสั่งซึ่งอยู่ในตัวโปรแกรมทำงานเป็นแบบขนาน

สถาปัตยกรรมคอมพิวเตอร์ที่สำคัญในการใช้งานโปรแกรมแบบขนาน[3] คือ

1. โมเดลใช้หน่วยความจำร่วมกัน (shared memory model) การทำงานโดยโพรเซสแต่ละโพรเซสมีการแบ่งใช้หน่วยความจำร่วมกันอย่างแน่นแฟ้น(tightly couple) ทำให้การเขียนโปรแกรมง่าย แต่ระบบมีราคาสูง

2. โมเดลใช้หน่วยความจำแบบกระจาย (distributed memory model) เป็นการเชื่อมต่อระหว่างโพรเซสอย่างหลวม ๆ (loosely couple) โพรเซสแต่ละตัวไม่จำเป็นต้องอยู่บนเครื่องเดียวกัน หน่วยความจำของโพรเซสท้องถิ่น (local memory) แต่ละตัวแยกกันอย่างอิสระ ระบบมีราคาถูกเนื่องจากสามารถใช้งานกับคอมพิวเตอร์ที่มีอยู่แล้วได้ แต่การเขียนโปรแกรมยากกว่าแบบแรก

3. โมเดลเรียกใช้หน่วยความจำระยะไกล (remote memory operation) เป็นโมเดลมีคุณสมบัติระหว่างการใช้หน่วยความจำร่วมกัน และแบบแมสเซจพาสซิ่ง โมเดลนี้ใช้แนวคิดแอคทีฟแมสเซจ (active message) ทำให้สามารถเรียกใช้โปรแกรมย่อยจากจากแอดเดรสสเปซ (address space) ของโพรเซสอื่นโดยใช้การทำสำเนาหน่วยความจำระยะไกล (remote memory copy) การรับส่งข้อมูลของโมเดลนี้เป็นแบบทางเดียว (one-sided send and receive model)

4. การเขียนโปรแกรมแบบขนานโดยใช้เทรด (thread) เป็นโมเดลที่โพรเซสทั้งหมดใช้แอดเดรสสเปซเดียวกัน โพรเซสใช้หน่วยความจำตนเองแบบโดยนัย (implicit) แต่เรียกหน่วยความจำที่ใช้ร่วมกันแบบเปิดเผย (explicit) การสื่อสารระหว่างโพรเซสผ่านทางโปรแกรมเค้าเตอร์ (program counter) และ สแตก (stacks)

5. โมเดลแมสเซจพาสซิ่ง (Message Passing Model) เป็นสภาพแวดล้อมในการติดต่อของโพรเซสระหว่างเครื่องแบบ MPMD (Multiple Processor Miltiple Data) ซึ่งใช้โมเดลหน่วยความจำแยกออกจากกัน การทำงานแบบขนานโดยใช้การควบคุมอย่างเปิดเผย (explicit control parallelism) โพรเซสทำงานโดยการส่งและรับแมสเซจระหว่างกัน

จากรายละเอียดข้างต้นสามารถสรุปเป็นตารางได้ดังนี้

Model	**Data Parallel**	**Shared memory**	**Thread**	**Remote operation**	**Message Passing**
Processor	SIMD	SIMD	SIMD,MPMD	SIMD,MPMD	SIMD,MPMD
Programming Algorithm	Sequential	Parallel	Parallel	Parallel	Sequential, Parallel
Data splitting	Compiler	User or compiler	User or compiler	N/A	User
Send/Recv	N/A	N/A	Program counter,stacks	One way	Two ways
Memory operation	Implicit	Implicit	Implicit,shared memory use explicit	Explicit (remote memory copy)	Explicit

ตารางที่ 2.1 การเปรียบเทียบคุณสมบัติของโมเดลการทำงานแบบขนาน

[illegible]

2.2 ระบบเอ็มพีไอ

เอ็มพีไอ (Message Passing Interface) เป็นการเขียนโปรแกรมตามโมเดลแมสเซจพาสซิ่ง สามารถทำงานได้ทั้งในระบบใช้หน่วยความจำร่วมกันหรือหน่วยความจำแยกจากกัน โดยแต่ละโพรเซสจะมีหน่วยความจำของตนเองแยกออกจากกัน แต่ละโพรเซสติดต่อกันโดยการเรียกใช้ไลบรารี่ของโปรแกรมย่อย ข้อมูลในแต่ละโพรเซสได้จากการทำสำเนาข้ามไปยังหน่วยความจำของแต่ละโพรเซสโดยการเรียกใช้โปรแกรมย่อยเอ็มพีไอสามารถใช้งานในระบบเครือข่ายที่มีคอมพิวเตอร์แตกต่างกัน (network of heterogeneous workstation) เพื่อให้มีประสิทธิภาพในการทำงานโดยรวมสูงขึ้น โดยใช้ระบบเครือข่ายเป็นทางเดินข้อมูล และคอมพิวเตอร์แต่ละตัวเสมือนเป็นตัวประมวลผลของระบบเอ็มพีไอ

2.3 ประวัติความเป็นมา [7]

เอ็มพีไอเกิดจากความพยายามเพื่อให้มีมาตรฐานในการส่งแมสเซจ ภายใต้สภาพแวดล้อมแบบหน่วยความจำกระจาย โดยได้รับความสนับสนุนจากศูนย์ศึกษาค้นคว้าการประมวลแบบขนาน (Center for Research on Parallel Computing) โดยฉบับร่างได้ถูกใช้เป็นครั้งแรกในเดือนพฤศจิกายน 1992 ที่เมืองมีนีอาโพลิส (Minneapolis) และมีการนำเสนอร่างเกี่ยวกับมาตรฐานของเอ็มพีไอนี้ ในงานซุปเปอร์คอมพิวติ้ง 1993 คอมเฟอเร้นซ์ ในเดือนพฤศจิกายน 1993 การปรึกษาหารือเป็นส่วนประกอบของเอ็มพีไอฟอรัม (MPI Forum) ซึ่งได้เปิดโอกาสให้สมาชิกของสังคมคอมพิวเตอร์ประสิทธิภาพสูง (High Performance Computing Society) เข้าร่วม

2.4 หลักการพื้นฐาน [3,6,15,16,17,26]

มาตรฐานของเอ็มพีไอถูกพัฒนาขึ้น เพื่อให้การเขียนโปรแกรมสำหรับระบบมัลติคอมพิวเตอร์ เป็นไปได้ง่ายและมีประสิทธิภาพ เอ็มพีไอได้มีการกำหนดคุณสมบัติต่าง ๆ ไว้ดังนี้

1. เป็นการสื่อสารระหว่างจุดต่อจุด (point-to-point)
2. มีคุณสมบัติในการสื่อสารแบบกลุ่ม
3. สามารถกำหนดโครงแบบการเชื่อมต่อเสมือน (virtual topology) ระหว่างโพรเซสในแต่ละกลุ่มได้

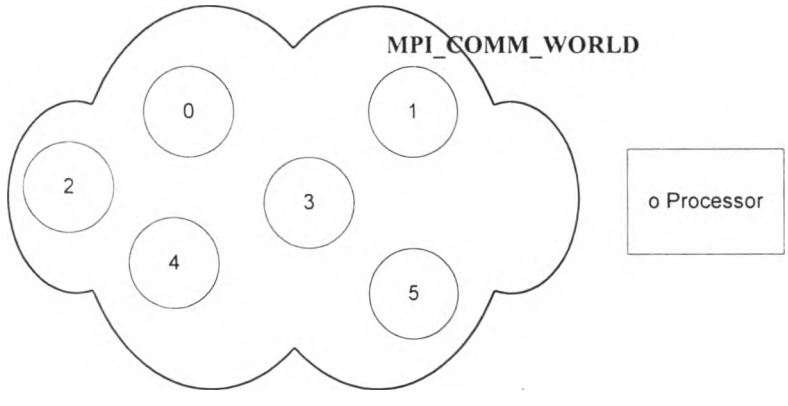
4. สามารถตรวจสอบประสิทธิภาพในระดับระบบแมสเซสพาสซิ่ง ซึ่งจะไม่ต้องเข้าไปตรวจสอบการทำงานในระดับ source code

5. กำหนดลักษณะเชื่อมต่อกับภาษา FORTRAN และ ANSI C ได้โดยตรง

6. มีการจัดการสภาพแวดล้อมและฟังก์ชั่นที่ใช้ในการสอบถาม ในการตรวจสอบข้อผิดพลาด

2.5 ขั้นตอนในการทำงานโดยสังเขป

เอ็มพีไอเริ่มต้นการทำงานโดยใช้คำสั่ง MPI_INIT เป็นการสร้างโครงข่ายการสื่อสารขึ้น ซึ่งเรียกว่า MPI_COMM_WORLD หรือคอมมิวนิเคเตอร์ (communicator) การเรียกใช้เอ็มพีไอแต่ละครั้งต้องเรียกผ่านเสมอ นอกจากนี้โพรเซสภายในเอ็มพีไอทำการสื่อสารถึงกันได้ ต่อเมื่อเป็นโพรเซสที่อยู่ในคอมมิวนิเคเตอร์เดียวกันเท่านั้น



รูปที่ 2.1 ภาพแสดงการสร้างคอมมิวนิเคเตอร์

ภายในคอมมิวนิเคเตอร์ประกอบไปด้วย กลุ่มของโพรเซสอย่างน้อย 1 กลุ่ม กลุ่มโพรเซสเหล่านี้ถูกกำหนดไว้ก่อนมีการสร้างคอมมิวนิเคเตอร์ โพรเซสที่อยู่ในกลุ่มของโพรเซสถูกเรียงลำดับตามตัวเลขโดยเริ่มต้นตั้งแต่ศูนย์ โดยค่าที่ให้แต่ละโพรเซสจะถูกเรียกว่าแร็ง (rank) ค่าของ แร็งนี้เป็นตัวแทนโพรเซสภายในคอมมิวนิเคเตอร์และสามารถมีได้หลายคอมมิวนิเคเตอร์ได้หลายๆ เช่นกัน แต่เมื่อเราต้องการเรียกถึงโพรเซสทั้งหมดของคอมมิวนิเคเตอร์ เราเรียกว่า MPI_COMM_WORLD

เมื่อต้องการจบการทำงานโปรแกรมจะเรียกใช้รูทีน (routine) ชื่อว่า MPI_FINALIZE รูทีนนี้เป็นการลบโครงสร้างข้อมูลที่เอ็มพีไอสร้างขึ้น แต่เอ็มพีไอไม่ยกเลิกคอมมิวนิเคเตอร์ที่ยังทำไม่เสร็จ เพื่อสามารถให้ตรวจสอบการทำงานได้ง่าย เมื่อเรียกใช้คำสั่งนี้แล้วโปรแกรมทั้งหมดก็จะยุติการทำงานโดยสมบูรณ์

2.6 หลักการของ MPICH [27]

ระบบ MPICH ถูกพัฒนาขึ้นโดยมีประสิทธิภาพตามมาตรฐานเอ็มพีไอและมีความสามารถในการเคลื่อนย้ายสู่เครื่องอื่นได้ง่าย (portable) คำว่า "CH" ใน ระบบ MPICH ย่อมาจากคำว่า "คาเมเลียน"(Chameleon) ซึ่งเป็นสัญลักษณ์แห่งการปรับตัวตามสภาพแวดล้อม

ระบบ MPICH เป็นทั้งงานวิจัยและงานพัฒนาซอฟแวร์ ในแง่ของงานวิจัย ระบบ MPICH มุ่งหมายในการหาวิธีลดช่องว่างระหว่างโปรแกรมแบบขนานที่เขียนขึ้นและประสิทธิภาพที่ได้จากฮาร์ดแวร์ เพื่อให้ผู้ใช้มีความเป็นอิสระสามารถพัฒนาใช้งานได้บนระบบคอมพิวเตอร์ที่หลากหลาย ในขณะเดียวกันทางด้านการพัฒนาช่วยให้ผู้ขายเครื่องสามารถปรับเปลี่ยน ระบบ MPICH ให้เข้ากับเครื่องคอมพิวเตอร์ชนิดต่างๆได้ง่าย โครงสร้างของ MPICH ได้รับอิทธิพลการออกแบบมาจากผลิตภัณฑ์ดังต่อไปนี้

1. ระบบ P4 เป็นไลบรารีสำหรับเขียนโปรแกรมแบบขนานในยุคที่ 3
2. คาเมเลียน (Chameleon) เป็นผลิตภัณฑ์ที่มีประสิทธิภาพในการถ่ายโอนสูงสำหรับแมสเซจพาสซิ่งบนเครื่องซูเปอร์คอมพิวเตอร์แบบขนาน
3. ซิปโค้ด (Zipcode) เป็นระบบสำหรับการเขียนไลบรารีที่สามารถปรับเปลี่ยนตามขนาดคอมพิวเตอร์ได้

ซิปโค้ดให้แนวความคิดหลายๆ อย่างต่อการออกแบบมาตรฐานเอ็มพีไอ เช่น คอนเทคท์ (context) กรุ๊ป (groups) และเมลเลอร์ (mailer) ซึ่งเทียบเท่าเอ็มพีไอคอมมิวนิเคเตอร์ ซิปโค้ดยังประกอบไปด้วยส่วนขยายสำหรับคำสั่งรวบรวมข้อมูล (extension collective operation) กับกรุ๊ปสโคป (group scope) เช่นเดียวกับโครงสร้างแบบเสมือน (virtual topology) และโค้ดเหล่านี้ถูกนำมาใช้เป็นหลักสำหรับพัฒนาระบบ MPICH ในรุ่นแรกๆ

2.7 สถาปัตยกรรมของระบบ MPICH

การออกแบบระบบ MPICH เน้นเป้าหมายหลักสองประการคือ

1. ใช้โค้ดที่มีการใช้ร่วมกันเป็นจำนวนมาก แต่ไม่กระทบถึงประสิทธิภาพในการทำงานมากนัก และเป็นโค้ดที่ไม่ขึ้นกับระบบคอมพิวเตอร์

2. ระบบ MPICH สามารถเคลื่อนย้ายไปยังระบบคอมพิวเตอร์แบบอื่นๆได้ง่าย

องค์ประกอบที่สำคัญในการทำให้ระบบ MPICH สามารถทำงานได้ตามเป้าหมายนี้คือ

2.8 สถาปัตยกรรมเอดีไอ (ADI หรือ Abstract Device Interface)

เอดีไอเป็นกลุ่มของคำจำกัดความฟังก์ชั่น (function definition) หรือคำจำกัดความมาโคร (macro definition) เอดีไอกำหนดให้ฟังก์ชันต่างๆ เปลี่ยนแปลงได้ตามอุปกรณ์ (device) ต่างและสามารถทำให้โค้ดต่างๆใช้งานร่วมกันได้ประสิทธิภาพ การทำงานภายในเอดีไอขึ้นอยู่กับผู้ผลิตคอมพิวเตอร์ที่พัฒนาเพื่อให้รองรับกับฮาร์ดแวร์ของตน

คำสั่งเอดีไอประกอบด้วย 4 ส่วนหลักคือ

1. บอกชนิดของการรับหรือส่งแมสเซจ
2. เคลื่อนย้ายข้อมูลระหว่างเอพีไอและฮาร์ดแวร์แมสเซจพาสซิ่ง
3. จัดการรายการแมสเซจที่กำลังมาถึง (ทั้งรับและส่ง)
4. ข้อมูลพื้นฐานสำหรับการทำงาน

ในกรณีฮาร์ดแวร์ของระบบแมสเซจพาสซิ่งอาจไม่สนับสนุนฟังก์ชันทั้งหมด เอดีไอจะจำลองโดยใช้รูทีนย่อยๆที่มีมาประกอบกัน

นอกจากนี้เอดีไอประกอบไปด้วยโค้ดสำหรับการทำแพ็คแมสเซจ (packetizing message) ปะรายละเอียดของหัวแมสเซจ (header information) บริหารมัลติเพิลบับเฟอริ่ง (multiple buffering) นโยบายสำหรับจับคู่แมสเซจที่ได้รับแล้วที่มาถึงแล้วกับแมสเซจที่กำลังมาหรือนำแมสเซจมาเรียงคิว (message queuing) ถ้าจำเป็นและรองรับการสื่อสารที่แตกต่างกัน (heterogenius communication)

2.9 สถาปัตยกรรมแชนแนลอินเตอร์เฟส (Channel Interface)

แชนแนลอินเตอร์เฟสทำหน้าที่รับหรือส่งเอนวีโลป (envelope) สำหรับการควบคุมการทำงาน ทำการรับส่งตัวข้อมูลและส่วนประกอบอื่น ๆ (มีลักษณะเป็นมาโคร) เพื่อเพิ่มประสิทธิภาพในการทำงาน โดยใช้คำสั่งพื้นฐานคอมพิวเตอร์ในการทำงาน ได้แก่ คำสั่งในการเลือก อ่าน เขียนข้อมูล

Channel Interface ประกอบด้วยรูทีนหลัก 5 ชุดคือ

1. รูทีนที่ทำหน้าที่ส่งและรับเอ็นเวโลป (ใช้ในการควบคุม) 3 รูทีน

   MPID_SendControl

   MPID_RecvAnyControl

   MPID_ControlMsgAvail

2. รูทีนสำหรับรับส่งข้อมูล 2 ชุด

   MPID_SendChannel

   MPID_RecvFromChannel

2.10 ประโยชน์ของโมเดลแมสเซจพาสซิ่ง [7]

1. สนับสนุนการใช้งานเชื่อมต่อของโพรเซสบนระบบสื่อสารแบบต่างๆ บนระบบซูปเปอร์คอมพิวเตอร์แบบขนาน ระบบเครือข่ายคอมพิวเตอร์ และโมเดลการใช้หน่วยความจำร่วมกัน
2. ไม่จำเป็นต้องมีระบบควบคุมแบบโมเดลการใช้หน่วยความจำแบบขนาน
3. สนับสนุนการการใช้อัลกอริทึมเพื่อปรับแต่งการกระจายงาน
4. ทนทานต่อความไม่สมดุลของความเร็วในแต่ละโพรเซส
5. เปิดทางเลือกสำหรับคอมไพเลอร์ที่สนับสนุนแคช (cache memory)
6. ในโมเดลใช้หน่วยความจำร่วมกัน โมเดลแมสเซจพาสซิ่งทำให้สามารถควบคุมการใช้ข้อมูลท้องถิ่น (data locality) ได้ดีขึ้น

2.11 หลักการของเครือข่ายท้องถิ่น (LAN) [5,24]

ระบบเครือข่ายท้องถิ่นแตกต่างจากระบบเครือข่ายข้อมูลอื่นๆตรงที่ได้รับการออกแบบให้เหมาะกับเครือข่ายขนาดย่อม ซึ่งมีขนาดตั้งแต่อาคารที่ทำงาน โกดังเก็บของ จนกระทั่งระดับมหาวิทยาลัย มาตรฐาน IEEE 802 ระบุว่า ระบบเครือข่ายท้องถิ่นเป็นการใช้งานร่วมกันของเครือข่ายชนิดเพียร์ทูเพียร์ (peer-to-peer) ขนาดกลาง ซึ่งส่งข้อมูลเพื่อให้ลูกข่ายทุกตัวได้รับ ซึ่งไม่ได้คำนึงถึงความเป็นส่วนตัวของข้อมูลมากนัก

2.12 หลักการของเครือข่ายระยะไกล (WAN) [5,24]

ระบบเครือข่ายระยะไกลถูกออกแบบมาให้ใช้งานในขนาดใหญ่กว่าระบบเครือข่ายท้องถิ่น โดยมีขนาดตั้งแต่หลายช่วงถนนจนถึงระดับประเทศ ระบบเครือข่ายระยะไกลมีปัญหาเกี่ยวกับอัตราความผิดพลาดในการส่งข้อมูลปานกลางจนถึงสูง และมีความล่าช้าเนื่องจากระยะทาง การบริหารงานระบบเครือข่ายระยะไกลโดยส่วยมากเป็นเครือข่ายสาธารณะ และนิยมใช้สำหรับเป็นช่องทางเชื่อมต่อระหว่างระบบเครือข่ายท้องถิ่น

จากหลักการข้างต้นสามารถสรุปเป็นตารางได้ดังนี้

คุณลักษณะ	WAN	LAN
เจ้าของสื่อ	interexchange carrier	องค์การของผู้ใช้
ความเร็วของสื่อ	300 Kbit/s - 1.544 Mbit/s	1 Mbit/s - 400 Mbit/s
ความเร็วของ DTE	150 bit/s - 9.6 Kbit/s	600 bit/s - 56 Kbit/s
ระยะทางระหว่าง DCE กับ DTE	หลายกิโลเมตร - หลายร้อยกิโลเมตร	สองสามฟุต – หลายร้อยฟุต
อัตราความผิดพลาดของการส่ง	1:1000 - 1:10000 bit	1:100,000,000 bit
ลักษณะการทำงาน (Connection)	ไม่ต้องมีการติดต่อก่อน	ต้องมีการติดต่อก่อน

ตารางที่ 2.2 เปรียบเทียบระหว่าง เครือข่ายระยะไกลกับ เครือข่ายท้องถิ่น

2.13 ปัญหาการใช้งานโพรโตคอลทีซีพีสำหรับระบบเครือข่ายระยะไกล [11]

ผลทดสอบที่ผ่านมาพบว่าทีซีพีสามารถทำงานได้ดีบนระบบอินเตอร์เนตหลายรูปแบบ ตั้งแต่ความเร็ว 800 ล้านบิตต่อวินาทีในไอโอแชนแนล (I/O channel) จนกระทั่ง 300 บิตต่อวินาทีในระบบไดอัลอัพโมเดม (dial-up modem) [23] อย่างไรก็ดียังสามารถเกิดคอขวดขึ้นได้ในระบบทีซีพีพื้นฐานสำหรับการถ่ายเทข้อมูลประเภทที่มีความกว้างของแถบสัญญาณมากและมีราวทริปไทม์ดีเลย์ (round-trip delay) สูง (หน่วยเป็นอาร์ทีที (RTT) ใน 1 วินาที โดยค่านี้ได้จากจำนวนข้อมูลที่ทีซีพีใส่ลงไปในสาย) ผลนี้ทำให้ไม่มีการตอบรับข้อมูลที่ส่งไปอย่างทันที (unacknowledged data) ทำให้เกิดปัญหาเกี่ยวกับประสิทธิภาพ ซึ่งจะลดลงเมื่อมีค่านี้มากขึ้นเรื่อยๆ

(จะเห็นได้ชัดเมื่อค่ามีค่ามากกว่า 10^5 บิต) เราเรียกการเกิดลักษณะนี้ว่า "long, fat pipe" และเรียกเครือข่ายที่มีลักษณะอย่างนี้ว่า "LFN" (ออกเสียงว่า "elephan(t)")

ตัวอย่างของ LFN ได้แก่ช่องสัญญาณดาวเทียมความจุสูง (High-capacity packet satellite channels) (DARPA's Wideband Net) แชนแนลความเร็วระดับที-วัน (T1-speed channel) ซึ่งมีค่าความกว้างของแถบสัญญาณ คูณด้วยความล่าช้า มีค่ามากกว่า 10^6 บิต สิ่งเหล่านี้ทำให้มีค่าทีซีพีเซกเม้นท์ (TCP segment) ขนาด 1200 ไบต์ ซึ่งอยู่บนสายมากกว่า 100 เซกเม้นท์ ซึ่งในอนาคตเครือข่ายใยแก้วนำแสงก็ตกอยู่ในฐานะ LFN ด้วย เช่นสายที่วิ่งข้ามเมืองที่มีความล่าช้า 30 มิลลิวินาที ที่ความกว้างของสัญญาณ DS3 (45 ล้านบิตต่อวินาที) ก็มีค่ามากกว่า 10^6 บิตเช่นกัน อัลกอลิทึมที่ดียังไม่เพียงพอที่จะแก้ไขปัญหา LFN นี้ได้ ทำให้จำเป็นต้องมีการเพิ่มส่วนขยายให้กับทีซีพี

ปัญหาพื้นฐาน 3 ประการของทีซีพีบนเครือข่าย LFN คือ

1. ขนาดของหน้าต่าง (window) มีจำกัด

ส่วนหัวทีซีพีปกติใช้ระเบียนขนาด 16 บิตในการแจ้งขนาดของหน้าต่างให้แก่ผู้รับทราบซึ่งไม่เพียงพอต่อการใช้งาน

2. มีการสะสมของการรอตอบรับ

แพคเก็ต (packet)ใดที่มีการสูญหายในระบบ LFN สามารถก่อให้เกิดผลเสียต่ออัตราการส่ง (throughput) อย่างคาดไม่ถึง ผลนี้เกิดจาก เมื่อเกิดเซกเม้นท์ (segment) สูญหาย การส่งทีซีพีจะเกิดการหมดเวลา (timeout) และจะต้องส่งเซกเม้นท์ที่สูญหายนั้นใหม่ แต่เนื่องจากการส่งทีซีพีไม่มีข้อมูลใด ๆ บอกว่าเซกเม้นท์ไหนที่ถึงผู้รับบ้าง และไม่ทราบว่าเซกเม้นท์ไหนบ้างที่ตอนนี้อยู่ระหว่างการเดินทางเพราะว่ายังไม่สิ้นสุดหน้าต่างนั้นๆ ซึ่งอาจเกิดการส่งเซกเม้นท์ที่ไม่จำเป็นเกิดขึ้น

3. ราวทริปไทม์มิ่ง (Round Trip Timing)

ทีซีพีสร้างชุดข้อมูลขึ้นมาเพื่อใช้ในการทดสอบราวทริปไทม์มิ่ง เช่นการใช้เวลาที่ส่ง 1 เซกเม้นท์ไปและรับการตอบรับ (acknowledgment) กลับมา การส่งเซกเม้นท์อีกครั้งสำหรับข้อมูลที่ไม่ได้รับการตอบรับ (unacknowledgment) ภายในจำนวนหลาย ๆ อาร์ทีที จากการศึกษาที่ผ่านมาพบว่า การประมาณที่ถูกต้อง จำเป็นต้องมีการเปลี่ยนแปลงลักษณะความคับคั่งของข้อมูล ระบบเครือข่ายที่มีข้อมูลคับคั่งเป็นสิ่งที่ไม่มีสามารถคาดการณ์ได้[9]

การคำนวณอาร์ทีทีของเซกเม้นท์อาจจะต้องมีการคำนวณที่ยุ่งยากโดยมีในส่วนของทีซีพีเซกเม้นท์เมื่อมีการส่งใหม่ ซึ่งอาจจะมีบางแพคเก็ตจะต้องทำการแพกข้อมูลใหม่ (repacket) ใหม่ และในส่วนของความซับซ้อนเนื่องจากการรวบรวมการตอบรับทีซีพี การคำนวณขั้นต่ำที่สุดทำได้

โดยคำนวณเพียง 1 เซกเม้นท์ต่อหน้าต่างในขณะที่การทำเช่นนี้ใช้ได้กับการประมาณในกรณีที่มีขนาดหน้าต่างเล็กๆ (4 - 8 segment Arpanet window) ซึ่งให้ผลที่ไม่น่าพอใจ

เมื่อมีข้อผิดพลาดเกิดขึ้น ปัญหาทำให้ประสิทธิภาพต่ำลง Zhang[29] , Jain[8] , และ Karn[10] แสดงถึงความเป็นไปไม่ได้ที่จะประมาณอาร์ทีทีถ้ามีแพคเก็ตที่มีการส่งใหม่รวมอยู่ด้วย เนื่องจากข้อมูลทั้งหน้าต่างได้ทำการส่งไปก่อนที่มีการส่งใหม่เกิดขึ้น และข้อมูลทั้งหมดต้องได้รับการตอบรับก่อนที่ตัวอย่างอาร์ทีทีต่อไปส่งมา ซึ่งหมายถึงซึ่งนี่หมายถึงต้องมีหน้าต่างอย่างน้อย 1 หน้าต่างในการคำนวณอาร์ทีทีและเพราะว่าอัตราการผิดพลาดจากการส่งต่อหน้าต่าง (10^{-6} error/bit สำหรับ Wideband Net) มันจะมีผลทำให้การวัดอาร์ทีทีเชื่อถือไม่ได้

2.14 โพรโตคอลในการสื่อสารภายในระบบเครือข่ายระยะไกล

โพรโตคอล CSLIP และ PPP เป็นโพรโตคอลที่ใช้อย่างแพร่หลายบนระบบเครือข่ายสาธารณะทั่วไปเพื่อเชื่อมต่อกับระบบเครือข่ายระยะไกล โพรโตคอลทั้งสองสนับสนุนโพรโตคอลทีซีพี/ไอพี

โพรโตคอล CSLIP

โพรโตคอล SLIP[21] มาจากคำว่า Serial Line IP ส่วนโพรโตคอล CSLIP[10] เป็นลักษณะการลดขนาดส่วนหัวของดาต้าแกรม (datagram) บนการเชื่อมต่อแบบอนุกรม SLIP มีความนิยมเพิ่มขึ้นสำหรับการใช้งานเชื่อมต่อคอมพิวเตอร์ภายในบ้านกับระบบอินเตอร์เนต ผ่านทางพอร์ตสื่อสาร RS-232C ซึ่งพบในคอมพิวเตอร์ส่วนบุคคลทั่วไป

เนื่องจากโพรโตคอล SLIP ถูกใช้ในสายสื่อสารความเร็วต่ำ (ต่ำกว่า 19200 บิตต่อวินาที) และ ถูกใช้ในการสื่อสารแบบโต้ตอบกัน จึงทำให้มีการส่งทีซีพีแพคเก็ตขนาดเล็ก ๆ โต้ตอบกันเป็นจำนวนมาก เนื่องจากทีซีพี/ไอพีที่มีขนาดส่วนหัวไอพี 20 ไบต์และส่วนหัวทีซีพีอีก 20 ไบต์รวมเป็น 40 ไบต์ทำให้เกิดความสิ้นเปลืองแถบสัญญาณเป็นจำนวนมาก โพรโตคอล CSLIP จึงนำมาใช้ทดแทน โดยโพรโตคอล CSLIP จะทำการลดขนาดของส่วนหัวลงเหลือ 3 ถึง 5 ไบต์ โดยสามารถใช้งานการเชื่อมต่อทีซีพีได้พร้อมกัน 16 ช่องทาง

โพรโตคอล PPP

โพรโตคอล PPP[20] มาจากคำว่า Point-to-Point เป็นโพรโตคอลที่สร้างขึ้นเพื่อใช้แก้ไขจุดบกพร่องต่างๆของโพรโตคอล SLIP และ CSLIP

องค์ประกอบของโพรโตคอล PPP มีด้วยกัน 3 ส่วนคือ

1. วิธีการ encapsulate IP datagrams
2. โพรโตคอลสำหรับควบคุมการเชื่อมต่อ (Link Control Protocol - LCP) ใช้ในการสร้างเตรียมระบบและทดสอบการเชื่อมต่อ
3. ชุดโพรโตคอลควบคุมเครือข่าย (Network Control Protocol - NCP) เพื่อที่จะสามารถใช้กับโพรโตคอลระดับเครือข่าย (network layer protocol) ที่แตกต่างกันได้คือ ไอพี (IP) โอเอสไอ (OSI network layer) เดกเน็ท (DECNet) และ แอปเปิ้ลทอล์ก (Apple Talk)

ข้อดีของโพรโตคอล PPP ที่เหนือกว่าโพรโตคอล SLIP คือ

1. สนับสนุนการใช้งานโพรโตคอลได้หลายชนิดในสายสัญญาณเส้นเดียว
2. มี CRC (Cyclic Redundancy Check) อยู่ทุกเฟรม (frame) ใช้ตรวจสอบความถูกต้อง
3. สามารถติดต่อกับไอพีแอดเดรสได้อย่างพลวัฒน์ (dynamic) (โดยการใช้โพรโตคอลควบคุมเครือข่ายชนิดไอพี (IP network control protocol)
4. ลดขนาดส่วนหัวของทีซีพีและไอพีเช่นกับกับโพรโตคอล CSLIP

โพรโตคอล PPP มีข้อด้อยกว่าโพรโตคอล SLIP คือมีขนาดของส่วนหัวที่ใหญ่กว่าโพรโตคอล SLIP แต่เนื่องจากข้อดีของโพรโตคอล PPP มีมากกว่า จึงทำให้โพรโตคอล PPP ได้รับการใช้งานอย่างแพร่หลายในปัจจุบัน

2.15 การใช้ระบบเอ็มพีไอในเครือข่ายระยะไกล

เนื่องจากระบบเอ็มพีไอได้รับการออกแบบมาเพื่อใช้งานในระบบเครือข่ายที่มีความน่าเชื่อถือสูงอย่างเครือข่ายเฉพาะที่ (Local Area Network)[15] จึงไม่มีการป้องกันความผิดพลาดของการทำงานเนื่องจากระบบเครือข่าย ดังนั้นการนำมาใช้งานกับระบบเครือข่ายระยะไกล ซึ่งมีค่าผิดพลาดและมีการสูญหายของข้อมูลค่อนข้างมากทำให้จะต้องมีการแก้ไขและปรับปรุงค่าเริ่มต้นต่างๆ เพื่อให้เอ็มพีไอมีความเหมาะสม และสามารถรองรับการทำงานบนระบบเครือข่ายระยะไกลได้อย่างมีประสิทธิภาพ