บทที่ 2

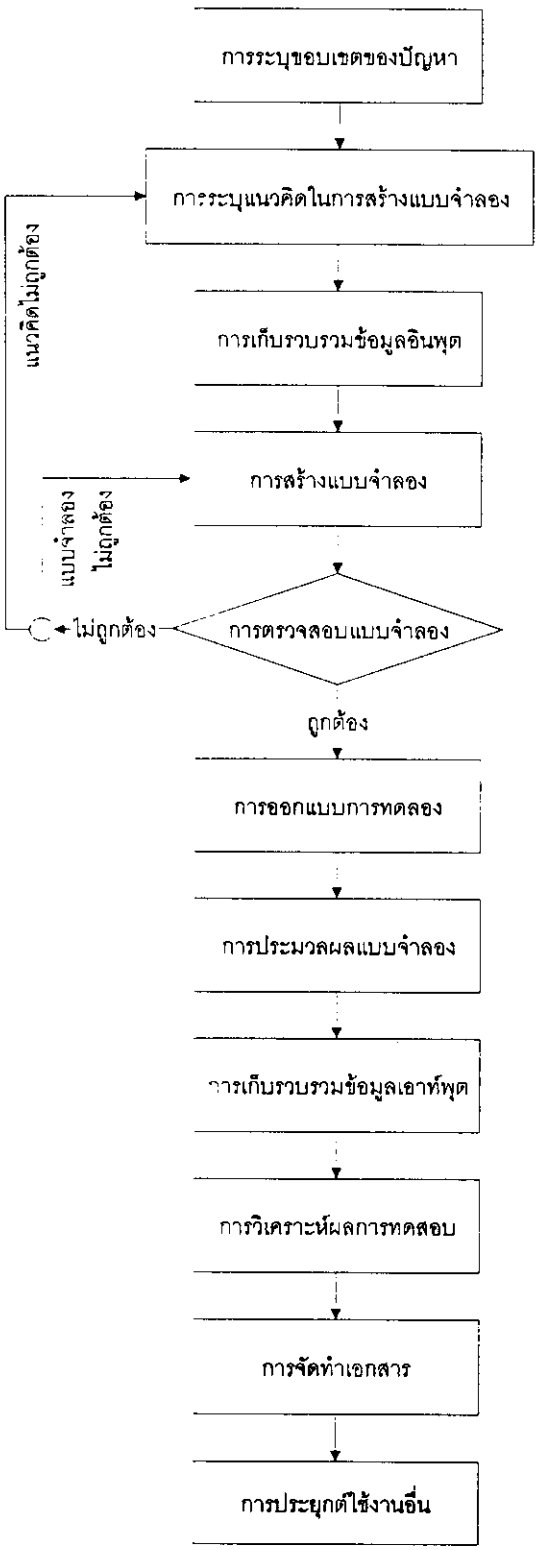
ตัวจำลองระบบเครือข่าย

ระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันมีขนาดใหญ่และมีความซับซ้อนมากขึ้น อีกทั้งจำนวนเครื่องลูกข่ายก็เพิ่มสูงขึ้น ดังนั้นเมื่อเกิดปัญหาขึ้นในระบบเครือข่าย จึงเป็นเรื่องยากที่จะทำการทดสอบโดยการแก้ไขหรือปรับปรุงในระบบที่ใช้งานอยู่ เพราะอาจส่งผลกระทบต่อระบบเครือข่ายทั้งระบบ ตัวจำลองระบบเครือข่ายได้เข้ามามีบทบาทสำคัญและเป็นอีกทางเลือกหนึ่งที่ใช้เป็นเครื่องมือช่วยในการศึกษาและทดสอบสมมุติฐานต่าง ๆ เนื่องจากมีข้อดีหลายประการ อาทิ เช่น มีค่าใช้จ่ายที่ต่ำและช่วยประหยัดระยะเวลาในการทดสอบ สามารถศึกษาปัญหาที่ซับซ้อนผ่านเงื่อนไขต่าง ๆ ที่กำหนดได้ง่ายและรวดเร็ว รวมทั้งมีความยืดหยุ่นในการปรับเปลี่ยนและทำซ้ำเพื่อหาผลการทดสอบที่ดีที่สุด และในปัจจุบันตัวจำลองระบบเครือข่ายมีทั้งที่สร้างขึ้นในเชิงพาณิชย์ ยกตัวอย่างเช่น OPNET และที่สร้างขึ้นโดยไม่หวังผลตอบแทนและสามารถนำมาใช้งานได้โดยไม่ต้องเสียค่าธรรมเนียมใด ๆ เช่น REAL 5.0, NS 2.0 เป็นต้น ต่อไปนี้จะได้กล่าวถึงทฤษฎีและหลักการในการจำลองระบบเครือข่ายคอมพิวเตอร์ ตัวจำลองระบบเครือข่ายที่มีในปัจจุบัน รวมถึงลักษณะการนำตัวจำลองแต่ละตัวไปใช้งาน ส่วนประกอบทางซอฟต์แวร์ และการสร้างตัวจำลอง

2.1 ทฤษฎีและหลักการ

การจำลองระบบเครือข่ายคอมพิวเตอร์ ประกอบด้วยองค์ประกอบหลัก 2 อย่างคือ แบบจำลองเครือข่าย ซึ่งอาจประกอบด้วยอุปกรณ์ที่ใช้ในระบบเครือข่าย (Network Element) เช่น สวิตช์ เราท์เตอร์ เป็นต้น รวมถึงโปรโตคอลที่ใช้ในการรับส่งข้อมูลจากต้นทางมายังปลายทาง และอีกองค์ประกอบหนึ่งคือ ค่าพารามิเตอร์ต่าง ๆ ของแบบจำลอง เช่น จำนวนของแพ็กเกตที่ถูกทิ้ง (Drop) ค่าหน่วงเวลาในการเข้าคิว ค่าหน่วงเวลาในการส่งข้อมูล และค่าพารามิเตอร์อื่นที่มีผลต่อประสิทธิภาพของระบบเครือข่าย

ตัวจำลองระบบเครือข่ายถือเป็นเครื่องมือที่มีความจำเป็นอย่างมาก เพื่อใช้ในการวิเคราะห์ประสิทธิภาพของระบบเครือข่าย โดยกระบวนการในการพัฒนาการจำลอง แสดงดังภาพประกอบ 2-1



ภาพประกอบ 2-1 กระบวนการในการพัฒนาการจำลอง

(ที่มา : ปรับปรุงจาก Roger D. Smith, 2000)

2.1.1 การระบุขอบเขตของปัญหา

ขั้นตอนแรกของกระบวนการในการพัฒนาการจำลอง คือ การกำหนดขอบเขตของปัญหาให้ชัดเจน วัตถุประสงค์และความต้องการของการจำลอง รวมถึงผลลัพธ์ที่ต้องการต้องถูกกำหนดอย่างชัดเจน โดยจะไม่สามารถทำการจำลองได้หากความต้องการยังไม่ชัดเจน

2.1.2 การระบุแนวคิดในการสร้างแบบจำลอง

การระบุแนวคิดในการสร้างแบบจำลองซึ่งอาจมีมากกว่า 1 วิธี โดยในขั้นตอนนี้เป็นการรวบรวมอัลกอริทึมต่าง ๆ เพื่อนำมาใช้ในการอธิบายระบบ อินพุตที่ต้องใช้ และผลลัพธ์ที่จะได้จากการจำลอง รวมถึงข้อจำกัดและสมมุติฐานต่าง ๆ ที่สามารถเป็นไปได้ทั้งหมด และนำข้อมูลที่ได้มาสรุปเพื่อหาผลลัพธ์เพียงผลลัพธ์เดียวที่ถูกต้องตามวัตถุประสงค์และความต้องการมากที่สุด

2.1.3 การเก็บรวบรวมข้อมูลอินพุต

การเก็บรวบรวมข้อมูลอินพุต โดยข้อมูลอินพุตเหล่านี้รวมไปถึงพฤติกรรมการทำงานของระบบและข้อมูลทางสถิติ ซึ่งข้อมูลดังกล่าวจะถูกนำมาใช้เป็นค่าพารามิเตอร์อินพุต เพื่อใช้ในการพัฒนาอัลกอริทึม และใช้ในการประเมินประสิทธิภาพการทำงานของการจำลอง การเก็บรวบรวมข้อมูลอินพุตที่มีความถูกต้องเป็นส่วนที่ยากที่สุดในกระบวนการทำงานของการจำลอง

2.1.4 การสร้างแบบจำลอง

ในการสร้างแบบจำลองจะขึ้นอยู่กับผลลัพธ์ที่ต้องการ (ข้อมูลเอาท์พุต) และข้อมูลอินพุตที่มีอยู่ โดยการเขียนโปรแกรมการทำงานบนเครื่องคอมพิวเตอร์ หรือการนำสมการทางคณิตศาสตร์และทางตรรกศาสตร์มาใช้ในการอธิบายระบบแทน และทำการทดสอบบนเครื่องคอมพิวเตอร์

2.1.5 การตรวจสอบแบบจำลอง

ขั้นตอนนี้ถือเป็นขั้นตอนที่มีความสำคัญมากขั้นตอนหนึ่ง ทั้งนี้เพื่อให้มั่นใจได้ว่าอัลกอริทึมของระบบ ข้อมูลอินพุต และสมมุติฐานที่ได้ออกแบบไปมีความถูกต้อง และสามารถนำผลลัพธ์ที่ได้ไปแก้ปัญหาที่ต้องการได้

2.1.6 การออกแบบการทดลอง

การออกแบบการทดลองเป็นขั้นตอนของการออกแบบวิธีการที่เหมาะสมในการทดสอบแบบจำลองว่าสามารถใช้ในการหาคำตอบที่ต้องการได้ ในการออกแบบดังกล่าวต้องแน่ใจได้ว่าเป็นวิธีที่ใช้ต้นทุนต่ำที่สุดและใช้ระยะเวลาสั้น ซึ่งสามารถนำวิธีการในทางสถิติมาใช้ช่วยในการออกแบบการทดลอง เพื่อให้ได้ข้อมูลที่มีความถูกต้องและแม่นยำ

2.1.7 การประมวลผลแบบจำลอง

เป็นการทดสอบแบบจำลองที่ได้ออกแบบไว้ เพื่อให้ได้ผลลัพธ์ที่ต้องการซึ่งนำไปสู่คำตอบในการแก้ปัญหา โดยอาจต้องทำการทดสอบหลายครั้งเพื่อให้ได้ผลลัพธ์ที่มีความน่าเชื่อถือ

2.1.8 การเก็บรวบรวมข้อมูลเอาท์พุต

ขั้นตอนนี้จะเกิดขึ้นพร้อมกับขั้นตอนของการประมวลผลแบบจำลอง โดยข้อมูลเอาท์พุตจะถูกรวบรวมและบันทึกไว้ เพื่อใช้ในการหาคำตอบที่ดีที่สุดของแบบจำลอง

2.1.9 การวิเคราะห์ผลการทดสอบ

ในการเก็บรวบรวมข้อมูลเอาท์พุตระหว่างที่ทำการทดสอบแบบจำลองมีจำนวนมาก และข้อมูลถูกเก็บกระจายในหลายช่วงเวลา ดังนั้นต้องมีการนำข้อมูลที่ได้มาวิเคราะห์ โดยในการวิเคราะห์ผลที่ได้อาจนำเสนออยู่ในรูปแบบของตาราง กราฟ แผนที่ ภาพเคลื่อนไหว หรือข้อความ ซึ่งทำให้ง่ายในการทำความเข้าใจ

2.1.10 การจัดทำเอกสาร

ทำการประเมินและสรุปผลลัพธ์ที่ได้จากการทดสอบแบบจำลอง โดยต้องมีการจัดเก็บเป็นเอกสาร เพื่อการปรับปรุงแบบจำลองในอนาคต

2.1.11 การประยุกต์ใช้งานอื่น

แบบจำลองที่สร้างขึ้นสามารถนำมาปรับปรุง แก้ไข เพื่อประยุกต์ใช้กับงานอื่นได้

2.2 ตัวจำลองระบบเครือข่ายที่มีในปัจจุบัน

2.2.1 NETSIM เวอร์ชัน 3.1 ของ MIT

NETSIM 3.1 เป็นตัวจำลองระบบเครือข่ายที่ขึ้นกับเหตุการณ์ (Event Driven) ที่ใช้กับระบบเครือข่ายแบบแพ็กเกตสวิตชิ่ง ถูกออกแบบโดย Massachusetts Institute of Technology Laboratory for Computer Science Advanced Network Architecture Group (MIT LCS ANA Group) เพื่อใช้งานเฉพาะกลุ่ม สำหรับผู้ที่มีความสนใจสามารถนำไปใช้งานได้โดยไม่ต้องเสียค่าธรรมเนียมใด ๆ แต่ปัจจุบันคู่มือการใช้งานไม่สามารถหาแบบออนไลน์ได้

การนำไปใช้งาน

สามารถจำลองระบบเครือข่ายได้ทุกประเภท โดย NETSIM 3.1 ทำงานเป็นแบบกระบวนการเดี่ยว (Single Process) เขียนด้วยโปรแกรมภาษาซี

ส่วนประกอบทางซอฟต์แวร์

โครงร่างของตัวจำลองระบบเครือข่ายประกอบด้วย รายการเหตุการณ์ต่าง ๆ (Schedule Event) และวิธีการติดต่อสื่อสารกับผู้ใช้เป็นแบบกราฟฟิก (Graphic User Interface) เพื่อแสดงโทโปโลยีและข้อมูลการทำงานของระบบเครือข่าย โดยส่วนติดต่อกับผู้ใช้จะแสดงเป็นอ้อปเจ็ก (Object) ต่าง ๆ แบบกราฟฟิก เช่น ปุ่มคอมโพเนนต์ต่าง ๆ (Components) หน้าต่างของข้อมูล (Information Window) เครื่องวัด (Meters) เป็นต้น และข้อมูลของอ้อปเจ็ก เช่น ค่าพารามิเตอร์ต่าง ๆ และลำดับการจัดคิว ซึ่งค่าพารามิเตอร์ของแต่ละอ้อปเจ็กสามารถทำการแก้ไขระหว่างที่ตัวจำลองกำลังทำงานอยู่ได้

แพ็กเกต ประกอบด้วย ตัวจัดการเหตุการณ์ (Event Manager) ส่วนของอินพุตและเอาท์พุต (I/O Routine) เครื่องมือต่าง ๆ เช่น เครื่องมือที่ใช้ในการจัดคิวหรือรายการที่จะใช้ในการสร้างคอมโพเนนต์ต่าง ๆ เป็นต้น และ Toolkit ที่เป็นไลบรารี (Library) ของฟังก์ชันต่าง ๆ ที่เขียนด้วยภาษาซี ซึ่งง่ายในการจัดการแต่ละคอมโพเนนต์

ในการสร้างแบบจำลองการทำงาน ผู้ใช้ต้องเขียนคอมโพเนนต์ขึ้นมาใหม่ด้วยภาษาซี โดยทำการแก้ไขไฟล์บางไฟล์แล้วทำการคอมไพล์ (Compile) และเชื่อมต่อแต่ละคอมโพเนนต์เข้าด้วยกัน และสร้างเป็นตัวจำลองโดยใช้ Toolkit

การสร้างตัวจำลอง

คอมโพเนนต์ต่าง ๆ อาจหมายถึง ลิงค์ (Link) โฮสต์ (Host) สวิตช์ (Switch) หรือตัวเชื่อมต่อระหว่างเครือข่าย ซึ่งทุก ๆ คอมโพเนนต์จะประกอบไปด้วยคลาส (Class) และประเภท (Type) โดยที่แต่ละคอมโพเนนต์จะต้องคอยจัดการกับเหตุการณ์ต่าง ๆ

เมื่อมีเหตุการณ์ (Event) เกิดขึ้นกับคอมโพเนนต์ ฟังก์ชันการทำงานจะถูกเรียกใช้ และทำงานตามเงื่อนไขที่ได้ถูกกำหนดค่าไว้ รวมถึงเงื่อนไขที่ผู้ใช้ทำการเขียนโปรแกรมเพื่อจัดการกับคอมโพเนนต์ต่าง ๆ โดยคอมโพเนนต์จะคอยจัดการกับเหตุการณ์ 3 ประเภท คือ Command, Regular และ Private ซึ่งเหตุการณ์แบบ Private จะเป็นเหตุการณ์ที่คอมโพเนนต์ทำการส่งข้อมูลถึงตัวเอง ส่วนเหตุการณ์แบบ Regular เป็นเหตุการณ์ที่เกี่ยวข้องกับการทำงานของตัวจำลอง และเหตุการณ์ที่เป็นแบบ Command เป็นเหตุการณ์ที่เกี่ยวกับคำสั่งที่ใช้ในการควบคุมการทำงาน

ค่าพารามิเตอร์ต่าง ๆ เป็นข้อมูลที่แสดงถึงคุณลักษณะของแต่ละคอมโพเนนต์ ซึ่งสามารถแสดงอยู่ในรูปของไฟล์ข้อมูล หรือให้แสดงบนหน้าจอคอมพิวเตอร์ (โดยผู้ใช้เป็นผู้ป้อนค่าอินพุต ส่วนเอาท์พุตจะได้จากการทำงานของตัวจำลอง) สามารถกำหนดค่าพารามิเตอร์ของอินพุตขณะที่ทำการสร้างคอมโพเนนต์ และสามารถแก้ไขในขณะที่ตัวจำลองกำลังทำงานอยู่ได้ คอมโพเนนต์ที่

NETSIM 3.1 มีให้ เช่น ลิงค์แบบอีเทอร์เน็ต (Ethernet Link) ลิงค์แบบจุดต่อจุด (Point-to-Point Link) สวิตช์ โฮสต์ และ Packet Sink เป็นต้น

2.2.2 The National Institute of Standards and Technology (NIST)

NIST ได้พัฒนาตัวจำลองระบบเครือข่ายแบบ ATM ที่ชื่อ NIST เพื่อใช้ในการศึกษาและประเมินประสิทธิภาพของเครือข่ายแบบ ATM โดยใช้ NETSIM เป็นต้นแบบ ส่วนคู่มือการใช้งานและตัวโปรแกรมต้นฉบับ (Source Code) สามารถหาได้โดยไม่เสียค่าธรรมเนียมใด ๆ

การใช้งาน

ตัวจำลองตัวนี้มีการนำไปใช้งานหลัก ๆ อยู่ 2 อย่างคือ Planning Tool และ Protocol Analysis Tool โดย Planning Tool ทำหน้าที่ในการวางแผนระบบเครือข่ายแบบ ATM ตัวจำลองสามารถทำงานได้โดยการกำหนดค่า Configuration และค่า Traffic Load ซึ่งจะให้ค่าผลลัพธ์ที่เป็นค่าทางสถิติ เช่น ค่า Link Utilization ค่าอัตราการรับส่งข้อมูล (Throughput) ส่วน Protocol Analysis Tool ใช้ในการวิเคราะห์ประสิทธิภาพของโปรโตคอลที่ใช้ในเครือข่ายแบบ ATM เป็นการศึกษาผลกระทบของโปรโตคอลแต่ละชนิดที่มีผลต่อระบบโดยรวม เช่น ในเรื่องของการจัดสรรแบนด์วิดท์ (Bandwidth Allocation) เป็นต้น

ส่วนประกอบทางซอฟต์แวร์

หลักการทำงานของตัวจำลอง NIST ไม่แตกต่างจากตัวจำลอง NETSIM 3.1 ของ MIT โดยเครื่องมือต่าง ๆ เขียนด้วยโปรแกรมภาษาซี รวมถึงมีส่วนติดต่อกับผู้ใช้เป็นแบบกราฟฟิก ทำงานบนระบบปฏิบัติการยูนิกซ์ โทโปโลยีของระบบเครือข่ายและค่าพารามิเตอร์ต่าง ๆ ที่เกี่ยวข้องของแต่ละคอมโพเนนต์ที่ใช้ในการจำลอง จะแสดงในหน้าต่างของข้อมูล โดยผู้ใช้สามารถสร้างโทโปโลยีของระบบเครือข่ายได้เอง ส่วนการทำงานของเครือข่ายจะถูกบันทึกไว้ที่หน้าต่างของเครื่องวัด อาจแสดงในรูปของกราฟแท่ง หรือฮีสโทแกรม หรืออื่น ๆ ตัวจำลองจะประกอบด้วยตัวจัดการเหตุการณ์ ส่วนของอินพุตและเอาท์พุต และเครื่องมือที่ใช้ในการสร้างคอมโพเนนต์ต่าง ๆ

การสร้างตัวจำลอง

ในการสร้างตัวจำลองทำได้โดยการนำคอมโพเนนต์มาต่อกัน และให้คอมโพเนนต์หนึ่งส่งข้อมูล (Message) ไปยังอีกคอมโพเนนต์ ซึ่งจะทำงานตามฟังก์ชันการทำงานและโครงสร้างของข้อมูลที่มี โดยทุก ๆ คอมโพเนนต์จะประกอบด้วยคลาสและประเภท

คลาสของคอมโพเนนต์ที่มีอยู่ในตัวจำลองตัวนี้ ได้แก่ ฟิสิคัลลิงค์ (Physical Link) สวิตช์แบบ ATM, Broadband Terminal Equipment (BTE) และงานประยุกต์ต่าง ๆ ของ ATM ยกตัวอย่างเช่น ตัวสร้างปริมาณข้อมูล (Traffic Generator) งานที่ใช้โปรโตคอล TCP/IP การบริการ

แบบ Constant Bit Rate (CBR) การบริการแบบ VBR เช่น Batch หรือ Poisson และการบริการแบบ ABR

2.2.3 CPSIM

CPSIM เป็นตัวจำลองระบบเครือข่ายที่ถูกสร้างขึ้นเพื่อวัตถุประสงค์ในการใช้งานทั่วไป โดย Groselj Boyan (BoyanTech Inc.)

การใช้งาน

CPSIM เป็นตัวจำลองที่จำลองเหตุการณ์แบบไม่ต่อเนื่องแบบขนาน (Parallel Discrete Event Simulation) เหมาะสำหรับการนำไปใช้งานในการจำลองเหตุการณ์ที่มีขนาดใหญ่ เช่น ระบบเครือข่ายคอมพิวเตอร์ สำหรับตัวโปรแกรมสามารถหาใช้งานได้โดยไม่เสียค่าธรรมเนียมใด ๆ แต่มีข้อจำกัดในเรื่องของความสามารถในการจำลองอ้อปเจ็กได้ 256 อ้อปเจ็ก ซึ่งถ้าต้องการแบบเต็มรูปแบบสามารถหาซื้อได้

ในการจำลองระบบเครือข่ายแบบขนาน (Parallel Simulation) มีวิธีการอยู่ 2 วิธี คือ Optimistic และ Conservative โดยในกรณีของ Optimistic เมื่อมีเหตุการณ์เข้ามา อ้อปเจ็กจะทำการประมวลผลเหตุการณ์ทันที แต่ในกรณีของ Conservative เหตุการณ์จะถูกประมวลผลที่เวลา (Simulation Time) เท่ากับ t เท่านั้น

ส่วนประกอบทางซอฟต์แวร์

จุดประสงค์ในการออกแบบตัวจำลองนี้ก็คือเรื่องของประสิทธิภาพ ตัวจำลองตัวนี้ไม่มีส่วนติดต่อกับผู้ใช้แบบกราฟฟิก และไม่มีฟังก์ชันภายใน (Built-in Function) ผู้ใช้สามารถทำการออกแบบตัวจำลองตัวนี้ได้อย่างอิสระ ดังนั้นตัวจำลองตัวนี้ถูกออกแบบมาเหมาะสำหรับผู้ที่มีความเชี่ยวชาญทางด้านการจำลองโดยเฉพาะ

การสร้างตัวจำลอง

CPSIM มีการแบ่งระหว่างส่วนของ Kernel และส่วนของไลบรารี โดย Kernel มี 2 เวอร์ชัน คือ สำหรับสถาปัตยกรรมแบบขนาน (Parallel Architectures) และสำหรับตัวประมวลผลแบบเดี่ยว (Uniprocessor) Kernel มีไว้สำหรับการ Synchronization Scheduling การป้องกันปัญหาการล็อคตาย (Deadlock) และการส่งผ่านข้อมูลบนระบบปฏิบัติการที่หลากหลาย ส่วนไลบรารีจะประกอบด้วยฟังก์ชันต่าง ๆ

CPSIM เป็นตัวจำลองที่เขียนด้วยโปรแกรมภาษาซี การลิงค์กันของอ้อปเจ็กต่าง ๆ ในตัวจำลองจะเป็นแบบกราฟฟิก โดยในระหว่างที่ตัวจำลองกำลังทำงานจะไม่สามารถทำการแก้ไขหรือเปลี่ยนแปลงค่าต่าง ๆ ได้

ในการสร้างแบบจำลองผู้ใช้จะเป็นผู้กำหนดรายละเอียดส่วนย่อย ๆ ขอ งอ้อปเจ็กเอง และต้องทำการกำหนดโครงสร้างของข้อมูล พร้อมทั้งเขียนโปรแกรมสำหรับเหตุการณ์ที่ต้องการให้เกิดขึ้นด้วย

เหตุการณ์ต่าง ๆ จะถูกจัดการโดยตัวประมวลผลที่เป็นอิสระต่อกัน เมื่อมีเหตุการณ์เข้ามายังอ้อป่เจ็ก เหตุการณ์จะถูกเพิ่มเข้าไปที่ส่วนท้ายสุดของคิว อ้อปเจ็กอาจมีเหตุการณ์หลายเหตุการณ์รออยู่ในคิว แต่ Scheduler จะจัดการกับเหตุการณ์ที่มี Time-Stamp น้อยที่สุดก่อน ในการทำงานระหว่างอ้อปเจ็กหลาย ๆ อ้อปเจ็กต้องมีตัวประสานงานเพื่อให้สามารถทำงานเข้าจังหวะกันได้ ก็คือ Null Event ซึ่งจะเป็นตัวจัดการว่าจะให้อ้อปเจ็กไหนเป็นคิวต่อไป

2.2.4 Network Simulation Testbed (NEST) **เวอร์ชัน** 2.5

NEST ถูกพัฒนาที่คณะวิทยาศาสตร์สาขาคอมพิวเตอร์ของ Columbia University ทำงานบนระบบปฏิบัติการยูนิกซ์ สามารถหาโปรแกรมต้นฉบับและคู่มือประกอบการใช้งานได้โดยไม่ต้องเสียค่าธรรมเนียมใด ๆ

การใช้งาน

วัตถุประสงค์ของตัวจำลองตัวนี้คือ สำหรับใช้ในการจำลองและสร้างอัลกอริทึมและระบบต้นแบบ NEST 2.5 อนุญาตให้ผู้ใช้ทำงานที่เกี่ยวข้องกับโปรโตคอลระดับสูง ๆ ได้ หรือใช้งานในระบบประมวลผลแบบกระจาย (Distributed System) และมีส่วนติดต่อกับผู้ใช้แบบกราฟฟิกสำหรับใช้ในการแสดงผลการจำลอง หรือแสดงรายละเอียดของเครือข่าย

การแสดงโทโปโลยีของระบบเครือข่ายจะอยู่ในรูปแบบของกราฟฟิก ทำให้ผู้ใช้สามารถสร้างหรือแก้ไขค่า Configuration ของเครือข่ายได้ โดยใช้เครื่องมือทางกราฟฟิก (Graphical Tool) ทำให้ง่ายในการใช้งาน

ตัวจำลองตัวนี้มีความสามารถในการจำลองโทโปโลยีแบบต่าง ๆ ของเครือข่ายที่แตกต่างกัน ซึ่งมีประโยชน์มากสำหรับงานที่ต้องการศึกษาอัลกอริทึมในการหาเส้นทางของเครือข่าย หรืองานที่ต้องการศึกษารูปแบบการไหลของข้อมูลในระบบเครือข่าย

ส่วนประกอบทางซอฟต์แวร์

หัวใจสำคัญของ NEST 2.5 คือ ไลบรารีของตัวจำลอง (Simulation Library) ซึ่งประกอบด้วย ฟังก์ชันต่าง ๆ ที่จำเป็นในการจำลองระบบประมวลผลแบบกระจาย โดยโปรแกรมที่ใช้ในการจำลองเกิดจากการนำไลบรารีของฟังก์ชันต่าง ๆ ที่เขียนด้วยภาษาซีมาเชื่อมต่อกัน ฟังก์ชันที่เด่นชัดของ NEST 2.5 คือ ฟังก์ชันเซิร์ฟเวอร์ ซึ่งทำหน้าที่เป็นเครื่องเซิร์ฟเวอร์ที่ใช้ติดต่อกับเครื่อง

ไคลเอนต์ และควบคุมผ่านทางโปรแกรมที่ใช้ติดต่อกับผู้ใช้เป็นแบบกราฟฟิก ฟังก์ชันดังกล่าวมีข้อดีคือ สามารถควบคุมตัวจำลองบนเครื่องเซิร์ฟเวอร์จากสถานีงานได้ ทำให้ประหยัดทรัพยากร

ตัวจำลองจะทำงานเป็นแบบกระบวนการเดี่ยวบนยูนิกซ์ (Single Unix Process) โดยจะมีส่วนของ Overhead น้อยเมื่อเทียบกับการทำงานแบบมัลติทาสก์กิ้ง (Multi-tasking) NEST 2.5 จะใช้กลไกในการประมวลผลแบบ Lightweight Process Mechanism ทำให้ง่ายในการจำลองระบบประมวลผลแบบกระจายที่มีความซับซ้อน (มีจำนวนโหนดตั้งแต่ 100 ถึง 1,000 โหนด)

การตรวจสอบแก้ไขจุดบกพร่องของตัวโปรแกรมสามารถทำได้หลายวิธี วิธีแรกโดยการแทรกคำสั่ง printf เข้าไปยังตำแหน่งของโปรแกรมที่ต้องการทำการตรวจสอบจุดบกพร่อง แต่เนื่องจากตัวจำลองทำงานเป็นแบบกระบวนการเดี่ยวบนยูนิกซ์ ทำให้สามารถใช้ dbx ซึ่งถือเป็นเครื่องมือที่มีประโยชน์มากที่ใช้ในการตรวจสอบจุดบกพร่องของโปรแกรม

NEST 2.5 เป็นตัวจำลองสำหรับเหตุการณ์แบบต่อเนื่อง โดยผู้ใช้สามารถกำหนดหรือแก้ไขความยาวของเหตุการณ์ที่ต้องการได้ และสามารถทำการแก้ไขค่า Configuration ได้ตลอดเวลา

2.2.5 Realistic And Large (REAL) เวอร์ชัน 5.0

REAL เป็นตัวจำลองระบบเครือข่ายที่เขียนขึ้นโดย S. Keshav ที่ Cornell University โดยการนำ NEST 2.5 มาเป็นต้นแบบ

การใช้งาน

REAL สร้างเพื่อการใช้งานในด้านการศึกษาพฤติกรรมแบบพลวัต (Dynamic Behavior) ของวิธีการควบคุมการไหลและความแออัดของข้อมูลในเครือข่ายแบบแพ็กเกตสวิตชิ่ง โดยผู้ใช้สามารถจำลองระบบเครือข่ายขึ้นมา แล้วสังเกตพฤติกรรมต่าง ๆ ที่เกิดขึ้นบนระบบเครือข่ายได้ พร้อมทั้งมีโปรแกรมต้นฉบับให้สำหรับผู้ที่มีความสนใจที่จะนำไปปรับปรุง และแก้ไขให้เหมาะสมกับงานของตนเอง แต่ REAL ไม่มีส่วนติดต่อกับผู้ใช้แบบกราฟฟิก

ระบบฮาร์ดแวร์และซอฟต์แวร์ที่สนับสนุนคือ ทำงานบนระบบปฏิบัติการยูนิกซ์ SunOS, Solaris, IRIX, BSD4.3, Ultrix และ UMIPS

ส่วนประกอบทางซอฟต์แวร์

REAL แบ่งออกเป็นโมดูลทั้งหมด 30 โมดูล ซึ่งเขียนด้วยภาษาซี โดยทำหน้าที่ในการจัดการฟังก์ชันการทำงานที่เกี่ยวกับโปรโตคอลที่ใช้ในการควบคุมการไหลของข้อมูล เช่น โปรโตคอล TCP และเกี่ยวกับอัลกอริทึมในการจัดคิว 5 ประเภท อาทิเช่น การจัดคิวแบบ First In First Out การจัดคิวแบบ Fair Queuing เป็นต้น

อินพุตเป็นแบบ Scenario ประกอบด้วย โทโปโลยีของเครือข่าย โปรโตคอล Workload เช่น เวลาที่แพ็กเกตมาถึง (Packet Arrival Time) และค่าพารามิเตอร์ที่ใช้ในการควบคุมต่าง ๆ เช่น ขนาดของแพ็กเกต (Packet Size) โดยในการกำหนดค่าต่าง ๆ ของระบบที่ต้องการใช้ในการจำลองจะใช้ภาษาแอสกี้อย่างง่าย (Simple ASCII) หรือที่เรียกว่า NetLanguage

เอาท์พุตเป็นแบบข้อมูลทางสถิติ (Statistic) เช่น จำนวนของแพ็กเกตที่ส่งออกไป ค่าหน่วงเวลาในการเข้าคิว จำนวนของแพ็กเกตที่ถูกทิ้ง และที่ถูกส่งใหม่ (Retransmit)

เนื่องจาก NEST 2.5 ไม่มีในเรื่องของไทม์เมอร์ (Timer) แต่ในส่วนของ REAL จะแก้ไขปัญหาตรงจุดนี้ด้วยการส่งแพ็กเกตของไทม์เมอร์ (Timer Packet) ออกไปจากต้นทาง แล้วส่งกลับมาหาตัวเองตามระยะเวลาที่ได้ถูกกำหนดไว้ แต่ยังมีข้อเสียคือวิธีนี้จะไม่สามารถทำการกำหนดค่าของไทม์เมอร์ใหม่ได้

2.2.6 Network Simulator (NS) เวอร์ชัน 2.0

NS เป็นตัวจำลองที่จำลองเหตุการณ์แบบไม่ต่อเนื่องในรูปแบบของอ้อปเจ็ก (Object-oriented Discrete Event Simulator) สำหรับงานวิจัยระบบเครือข่าย มีต้นแบบมาจาก REAL เริ่มแรกเป็น NS เวอร์ชัน 1.0 ถูกพัฒนาโดยกลุ่มที่ทำงานวิจัยเกี่ยวกับระบบเครือข่าย คือ Lawrence Berkeley National Laboratory (LBNL) แต่ได้มีการพัฒนามาเป็น NS เวอร์ชัน 2.0 ซึ่งเป็นส่วนหนึ่งของโครงการ Virtual InterNetwork Testbed (VINT)

การใช้งาน

NS 2.0 เหมาะสำหรับใช้ในการจำลองระบบเครือข่ายอินเตอร์เน็ตที่หลากหลาย โดยสามารถจำลองการทำงานของโปรโตคอลต่าง ๆ เช่น Transmission Control Protocol (TCP), User Datagram Protocol (UDP) รวมถึงโปรโตคอลที่ใช้ในระดับชั้น Medium Access Control (MAC) ของเครือข่ายแลน และในงานประยุกต์ต่าง ๆ เช่น Hyper Text Transfer Protocol (HTTP), Teletype Network Protocol (Telnet), File Transfer Protocol (FTP) เป็นต้น อัลกอริทึมที่ใช้ในการจัดคิวแบบต่าง ๆ เช่น FIFO, Random Early Detection (RED) และ Class Based Queuing (CBQ) เป็นต้น และอัลกอริทึมต่าง ๆ ที่ใช้ในการหาเส้นทาง เช่น อัลกอริทึม Dijkstra รวมทั้งการรับส่งข้อมูลแบบมัลติคาสท์ และ NS 2.0 สามารถจำลองโหนดได้ถึง 1,000 โหนด

คุณลักษณะอื่นของ NS 2.0 คือสามารถจำลอง Error Model ได้ โดยหน่วยของ Error อาจอยู่ในรูปของจำนวนแพ็กเกต จำนวนบิต หรือจำนวนขึ้นกับเวลา

ระบบฮาร์ดแวร์และซอฟต์แวร์ที่สนับสนุน ทำงานบนระบบปฏิบัติการยูนิกซ์ ได้แก่ FreeBSD, Linux, SunOS, Solaris และสามารถทำงานภายใต้ระบบปฏิบัติการวินโดวส์ได้ ทั้ง Win95/98/NT

ส่วนประกอบทางซอฟต์แวร์

NS 2.0 เขียนด้วยโปรแกรมภาษา C++ และ OTcl (Object-oriented Tool command language) โดยแพ็กเกตจะประกอบด้วย Class Hierarchy ของอ้อปเจ็กต่าง ๆ ที่เขียนด้วยโปรแกรมภาษา C++ และภาษา OTcl ผู้ใช้สามารถสร้างอ้อปเจ็กใหม่ได้โดยผ่านทาง OTcl Interpreter

Tcl เป็นกระบวนการที่มีความคล่องตัวและมีประโยชน์มากที่ถูกนำมาใช้ในการจำลอง เช่น เหตุการณ์แบบ Start and Stop หรือ Network Failure โดยเขียนโปรแกรมด้วยภาษา Tcl เพื่อสร้างโทโปโลยีของระบบเครือข่าย โดยการใช้ลิงค์ โหนด และ Agent (ซึ่งทำงานร่วมกับโหนด) ในการสร้าง

การสร้างตัวจำลอง

ในการควบคุมและกำหนดค่า Configuration ต่าง ๆ ของตัวจำลองสามารถทำโดยผ่านทางคลาสของ OTcl ซึ่งคลาสจะมีกระบวนการในการสร้างและการจัดการโทโปโลยีต่าง ๆ รวมถึงการกำหนดรูปแบบเริ่มต้นของแพ็กเกต และการเลือกตัว Scheduler ด้วย เช่น ในการสร้างโทโปโลยีจะใช้คลาสโหนด คลาสลิงค์ และคลาส Agent ของ OTcl โดยฟังก์ชันการทำงานของโหนดคือ รับแพ็กเกตเข้ามา ตรวจสอบแพ็กเกต และส่งแพ็กเกตต่อไปยังโหนดถัดไป ส่วนลิงค์มีหลายชนิดให้เลือกใช้ เช่น แบบจุดต่อจุด แบบบรอดคาสท์ (Broadcast) หรือแบบไร้สาย (Wireless) เป็นต้น โดยสามารถกำหนดคุณลักษณะในรูปแบบของค่าหน่วงเวลา (Delay) และค่าแบนด์วิดท์ได้ และสุดท้าย Agent เป็นคอมโพเนนต์อีกประเภทหนึ่งของโหนด แต่ Agent จะถูกสร้างให้อยู่ตรงตำแหน่งปลายสุดของเครือข่าย ผู้ใช้สามารถสร้าง Source หรือ Sink ใหม่ได้จากคลาส Agent โดย NS 2.0 มี Agent ที่สามารถรองรับโปรโตคอลได้หลากหลาย ไม่ว่าจะเป็น TCP, CBR หรือ UDP รวมถึงโปรโตคอลอื่น ๆ เช่น Real Time Protocol (RTP), Real Time Control Protocol (RTCP), Scalable Reliable Multicast (SRM)

NS 2.0 มีส่วนติดต่อกับผู้ใช้เป็นแบบกราฟฟิก เรียกว่า Network Animator (NAM) โดยสามารถแสดงค่าอัตราการรับส่งข้อมูล และจำนวนของแพ็กเกตที่ถูกทิ้งที่แต่ละลิงค์ได้

2.2.7 Objective Modular Network Testbed in C++ (OMNeT++)

OMNeT++ ถูกพัฒนาโดย András Varga ที่ Technical University of Budapest, Department of Telecommunications (BME-HIT)

การใช้งาน

OMNeT++ เป็นตัวจำลองที่จำลองเหตุการณ์แบบไม่ต่อเนื่องในรูปแบบของอ้อปเจ็กโมดูล (Object-oriented Modular Discrete Event Simulator) ซึ่งสามารถนำไปใช้ในการจำลองโปรโตคอลต่าง ๆ ที่ใช้ในการติดต่อสื่อสาร ระบบเครือข่ายคอมพิวเตอร์และแบบจำลองทราฟฟิก มัลติโปรเซสเซอร์ (Multi-processor) และระบบประมวลผลแบบกระจาย ระบบการจัดการ (Administrative System) และระบบอื่น ๆ ที่เป็นเหตุการณ์แบบไม่ต่อเนื่อง

ส่วนประกอบทางซอฟต์แวร์

OMNeT++ เป็นตัวจำลองระบบเครือข่ายคอมพิวเตอร์ที่มีความยืดหยุ่นในการทำงาน ถูกพัฒนาขึ้นด้วยภาษา C++ และทำงานบนระบบปฏิบัติการยูนิกซ์ และยังสามารถทำงานบนระบบปฏิบัติการวินโดวส์ได้ ทั้ง Win3.1/95/NT

ในโปรแกรม OMNeT++ มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถทำการโปรแกรมได้เร็วขึ้น โดยจะมีส่วนของ Graphical NED Editor (GNED) ที่ใช้จำลองโมดูลแบบกราฟฟิก และแบบ Editor ซึ่งจะสามารถสร้างโมดูล และกำหนดคุณสมบัติต่าง ๆ ในโหมดของกราฟฟิก และสามารถทำการแก้ไข NED Language เพื่อเชื่อมโยงโมดูลต่าง ๆ เข้าด้วยกัน

โปรแกรม OMNeT++ มีส่วนติดต่อกับผู้ใช้แบบกราฟฟิก เพื่อแสดงผลของการเปลี่ยนแปลงตัวแปรต่าง ๆ ในขณะที่ตัวจำลองกำลังทำงาน ซึ่งมีความสำคัญมากในการตรวจสอบแก้ไขจุดบกพร่องของโปรแกรมว่าแสดงผลได้ตามที่ต้องการหรือไม่ โดยมีส่วนติดต่อกับผู้ใช้ 2 แบบด้วยกันสำหรับบนระบบปฏิบัติการยูนิกซ์ ดังนี้

Cmdenv : เป็นแบบบรรทัดคำสั่ง (Command-Line) การทำงานแบบนี้จะเป็นการทำงานแบบดำเนินต่อไปจนครบตามที่กำหนดใน Configuration File แต่หากการทำงานหยุดไป ในขณะที่ยังทำงานไม่เสร็จนั้นแสดงว่าโปรแกรมมีข้อผิดพลาด (Error) เกิดขึ้น ซึ่งทำให้ผู้ใช้ทราบว่าต้องทำการดีบักที่ส่วนใดของโปรแกรม

Tkenv : เป็นแบบกราฟฟิก โดยแสดงเป็นหน้าต่าง (Window User Interface) การทำงานในโหมดนี้จะแสดงผลทิศทางการไหลของข้อมูลในแต่ละโมดูลย่อย (Submodule) และแสดงเหตุการณ์ที่เกิดขึ้นในแต่ละโมดูลย่อย โดยระหว่างการทำงานสามารถหยุดและทำงานใหม่ได้ นอกจากนั้นส่วนติดต่อกับผู้ใช้สามารถใช้งานได้ง่าย และยังสามารถตรวจสอบค่าได้

2.2.8 Internet Simulated ATM Networking Environment (INSANE)

INSANE เป็นตัวจำลองระบบเครือข่ายที่ถูกออกแบบที่มหาวิทยาลัยของแคลิฟอร์เนีย ซึ่งตั้งอยู่ที่ Berkeley ในปี 1996

การใช้งาน

INSANE มีจุดประสงค์หลัก คือ การประเมินประสิทธิภาพของอินเตอร์เน็ตโปรโตคอลที่หลากหลายบนเครือข่ายแบบ ATM โดยสามารถทำการจำลองได้มากกว่า 1,000 โฮสต์ แต่ปัจจุบันไม่ได้มีการพัฒนาต่อไป

ระบบฮาร์ดแวร์และซอฟต์แวร์ที่สนับสนุนคือ ทำงานบนระบบปฏิบัติการยูนิกซ์ Sun, Sparcstation, Solaris, FreeBSD, NetBSD และ IRIX

ส่วนประกอบทางซอฟต์แวร์

INSANE เป็นตัวจำลองที่จำลองเหตุการณ์แบบไม่ต่อเนื่องในรูปแบบของอ้อปเจ็ก เขียนด้วยโปรแกรมภาษา C++ โดยจะมีอ้อปเจ็กต่าง ๆ ให้มา (Built-in Objects) ได้แก่ คิวประเภทต่าง ๆ และโมดูลของโปรโตคอลต่าง ๆ ได้แก่ IP, TCP และ UDP และงานประยุกต์ต่าง ๆ เช่น FTP, Telnet, World Wide Web (WWW), Browser, Audio, Video และยังสามารถใช้กลุ่มของคำสั่งสร้างอ้อปเจ็กใหม่ ๆ ได้ เช่น ในการสร้างสวิตช์แบบ ATM จะใช้ Tcl Script ที่ถูกอ้างจาก Configuration File

ตัวจำลองตัวนี้ทำงานเป็นแบบกระบวนเดี่ยว แต่ในการทำงานของตัวจำลองที่มีขนาดใหญ่จะใช้ส่วนติดต่อกับผู้ใช้แบบกราฟฟิกที่เรียกว่า INSANEMON ซึ่งทำให้ง่ายในการแสดงผลที่ได้จากการทำงานของตัวจำลองหลาย ๆ ตัวบนเครื่องต่าง ๆ

2.2.9 Optimized Network Engineering Tools (OPNET)

OPNET เป็นตัวจำลองเชิงพาณิชย์ของ MIL3 Inc. ซึ่งได้มีการพัฒนามาเป็นเวลากว่า 15 ปี

การใช้งาน

OPNET มีส่วนติดต่อกับผู้ใช้เป็นแบบกราฟฟิก และสามารถจำลองการทำงานของเครือข่ายได้เป็นจำนวนหลายร้อยโหนด ทางบริษัท Thomson-CSF และ CNET ได้นำ OPNET ไปใช้ในการจำลองระบบเครือข่ายแบบ ATM และใช้กับโปรโตคอลในชั้นต่าง ๆ วิธีการจำลองของ OPNET เป็นการจัดการโครงสร้างแบบระดับขั้น (Hierarchical Structure)

ส่วนประกอบทางซอฟต์แวร์

ซอฟต์แวร์ตัวนี้เขียนด้วยภาษาซี ประกอบด้วยเครื่องมือมากมายและแบ่งออกเป็นหลายส่วน ดังนี้ ส่วนออกแบบและวางแผนตัวจำลอง (OPNET Modeler and OPNET Planner) ไลบรารีของแบบจำลอง (Model Library) และเครื่องมือที่ใช้ในการวิเคราะห์ (Analysis Tool) โดยส่วนออกแบบตัวจำลองมีไว้สำหรับการจำลองและวิเคราะห์ประสิทธิภาพของระบบเครือข่ายที่มีขนาดใหญ่ ระบบคอมพิวเตอร์ และงานประยุกต์อื่น ๆ โดยทั่วไปใช้ในการประเมินความเป็นไปได้ของระบบที่ได้ออกแบบใหม่ พัฒนาระบบการสื่อสารให้มีความเหมาะสม และเพื่อประเมินประสิทธิภาพของระบบเครือข่าย และในส่วนวางแผนตัวจำลอง จะเป็นโปรแกรมประยุกต์ที่อนุญาตให้ผู้ดูแลระบบสามารถประเมินประสิทธิภาพของระบบเครือข่ายได้ โดยไม่ต้องทำการโปรแกรมหรือคอมไพล์ใหม่ ผู้ใช้ไม่สามารถกำหนดแบบจำลองใหม่ได้ แต่หากต้องการแก้ไขต้องทำการติดต่อกับ MIL3

ไลบรารีของแบบจำลอง เช่น ไลบรารีของ ATM ได้แก่ ฟังก์ชันการทำงานของเครือข่ายแบบ ATM, ATM Adaptation Layer (AAL) และส่วนติดต่อกับอินเตอร์เน็ตโปรโตคอล ไว้สำหรับการจัดการเรื่องของบัฟเฟอร์ (Buffer Management) การควบคุมความแออัดของข้อมูลในเครือข่าย การแบ่งย่อยข้อมูลและการนำกลับมารวมกันของข้อมูล (Segmentation and Reassembly) ส่วนไลบรารีของอินเตอร์เน็ตโปรโตคอล ไว้สำหรับติดต่อกับโปรโตคอลอื่น ๆ เช่น โปรโตคอลที่ใช้ในเครือข่าย ATM หรือเครือข่ายอีเทอร์เน็ต

สุดท้ายเครื่องมือที่ใช้ในการวิเคราะห์ เป็นส่วนแสดงผลที่ได้จากการทำงานของตัวจำลอง ซึ่งเป็นแบบกราฟฟิก โดยสามารถวิเคราะห์ผลลัพธ์ที่ได้ของอุปกรณ์ที่ใช้ในระบบเครือข่ายแต่ละตัว

2.2.10 ตัวจำลองอื่น ๆ

มีตัวจำลองระบบเครือข่ายอีกมากมาย ยกตัวอย่างเช่น

2.2.10.1 Parallel Simulation Environment for Complex systems (PARSEC) เป็นตัวจำลองเวอร์ชันใหม่ของ Maisie (Maisie เป็นตัวจำลองระบบเครือข่ายที่เขียนด้วยภาษาซี เหมาะสำหรับการจำลองเหตุการณ์แบบไม่ต่อเนื่อง) ถูกพัฒนาโดย Parallel Computing Laboratory ที่ University of California Los Angeles (UCLA) PARSEC เป็นซอฟต์แวร์ที่ไม่ใช่สิทธิของสาธารณะแต่สามารถเปลี่ยนแปลงและแก้ไขโปรแกรมเพื่อการศึกษา งานวิจัย และองค์กรที่ไม่ได้ทำเพื่อหวังผลกำไรได้ โดยต้องทำการลงทะเบียนก่อน PARSEC เป็นตัวจำลองแบบขนาน เขียนด้วยโปรแกรมภาษาซี และทำการคอมไพล์โดยใช้ PARSEC Compiler (pcc) ตัวจำลองตัวนี้สามารถทำการจำลองโหนดได้ถึง 1,000 โหนด ทำงานบนระบบปฏิบัติการยูนิกซ์ รองรับในเรื่อง

ของการตรวจสอบแก้ไขจุดบกพร่องของตัวโปรแกรม โดยใช้ Unix C Debugger พื้นฐาน เช่น dbx หรือ gdb ของระบบปฏิบัติการยูนิกซ์

2.2.10.2 COMNET III เป็นตัวจำลองเพื่อการพาณิชย์ และเป็นเครื่องมือที่ใช้ในการวิเคราะห์ประสิทธิภาพของระบบเครือข่าย โดยตัวจำลองตัวนี้สามารถจำลองการทำงานของเครือข่ายแบบแลน เครือข่ายแบบแวน และเครือข่ายแบบแมน สามารถทำงานบนระบบปฏิบัติการวินโดวส์ได้ ทั้ง Win95/NT และในการสร้างแบบจำลองสามารถทำได้โดยการเลือกรูปไอคอนซึ่งแสดงแทนอุปกรณ์ต่าง ๆ เช่น โหนด ลิงค์ เป็นต้น ส่วนตำแหน่งของอุปกรณ์สามารถใช้เมาส์ในการเลื่อนไปวางยังตำแหน่งที่ต้องการได้ ทำให้ง่ายต่อการออกแบบแบบจำลอง

2.2.10.3 System for Modeling Unslotted Real-time Phenomena (SMURPH) เป็นตัวจำลองที่จำลองเหตุการณ์แบบไม่ต่อเนื่อง ใช้สำหรับจำลองการทำงานของโปรโตคอลต่าง ๆ ที่ใช้ในการสื่อสารในระดับชั้น MAC (MAC Layer) หรือในระดับต่ำ (Low Level) โดยเขียนด้วยภาษา C++

2.2.10.4 Maryland Routing Simulator (MaRS) เป็นตัวจำลองที่จำลองเหตุการณ์แบบไม่ต่อเนื่อง ถูกออกแบบมาสำหรับการศึกษาโปรโตคอลที่ใช้ในการหาเส้นทางของระบบเครือข่าย ซึ่งสามารถเพิ่มโมดูลของโปรโตคอลที่ใช้ในการหาเส้นทาง (Routing Protocol) และรองรับโปรโตคอลที่ใช้ในระดับชั้นทรานสปอร์ต (Transport Layer Protocol) รวมถึงมีส่วนติดต่อกับผู้ใช้เป็นแบบกราฟฟิก 2 ส่วน คือ Xlib และ Motif แต่มีข้อจำกัดในเรื่องของแบบจำลองงานประยุกต์ต่าง ๆ

2.2.10.5 NETSIM จากมหาวิทยาลัยของ Richmond เป็นตัวจำลองที่จำลองเหตุการณ์แบบไม่ต่อเนื่องแบบกระบวนการเดียว (Single Process Discrete Event Simulator) เขียนด้วยภาษาซี ถูกออกแบบสำหรับการศึกษาประสิทธิภาพของระบบเครือข่ายแบบแลน

2.2.10.6 Block Oriented Network Simulator (BONeS) เป็นเครื่องมือเพื่อการพาณิชย์ และเป็นตัวจำลองที่ขึ้นกับเหตุการณ์ ที่ใช้สำหรับการออกแบบและจำลองระบบเครือข่ายระดับสูง (High Level) เพื่อประเมินผลกระทบที่มีต่อประสิทธิภาพของระบบ แบ่งการทำงานออกเป็น 2 ส่วน คือ BONeS Designer และ Signal Processing Worksystem (SPW) ทำงานบนระบบปฏิบัติการวินโดวส์

2.2.10.7 PROPHESY เป็นเครื่องมือเพื่อการพาณิชย์เช่นกัน เหมาะสำหรับการจำลองเครือข่ายแบบแลนและแวน ทำงานบนระบบปฏิบัติการวินโดวส์ 3.1 ขึ้นไป

2.3 สรุป

ตัวจำลองระบบเครือข่ายในปัจจุบันมีอยู่มากมาย ทั้งที่เป็นแบบ Open Source คือ สามารถหาโปรแกรมต้นฉบับมาใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย พร้อมทั้งสามารถนำมาปรับปรุงและแก้ไขเพื่อให้เหมาะสมกับงานได้ ยกตัวอย่างเช่น NETSIM เวอร์ชัน 3.1 ซึ่งเหมาะในการนำมาจำลองระบบเครือข่ายแบบแพ็กเกตสวิตชิ่ง NIST เป็นตัวจำลองที่สร้างขึ้นเพื่อศึกษาและประเมินประสิทธิภาพของเครือข่ายแบบ ATM INSANE เป็นตัวจำลองระบบเครือข่ายที่ถูกออกแบบสำหรับใช้ในการทดสอบประสิทธิภาพของโปรโตคอลที่หลากหลาย เช่น IP, TCP และ UDP บนเครือข่ายแบบ ATM เขียนด้วยภาษา C++ NEST เวอร์ชัน 2.5 เหมาะสำหรับใช้ในการจำลองและสร้างอัลกอริทึมและระบบต้นแบบ ส่วน REAL เวอร์ชัน 5.0 เป็นตัวจำลองระบบเครือข่ายที่ถูกสร้างขึ้นเพื่อศึกษาพฤติกรรมแบบพลวัตของวิธีการควบคุมการไหลและความแออัดของข้อมูลในเครือข่ายแบบแพ็กเกตสวิตชิ่ง เขียนด้วยภาษาซี โดยใช้ NEST เวอร์ชัน 2.5 เป็นต้นแบบ OMNeT++ เป็นตัวจำลองเหตุการณ์แบบไม่ต่อเนื่อง ที่เขียนด้วยภาษาซี โดย András Varga และตัวจำลองอีกตัวที่ได้รับความนิยมคือ NS เวอร์ชัน 2.0 เป็นตัวจำลองเหตุการณ์แบบไม่ต่อเนื่อง ที่เขียนด้วยภาษา C++ และ OTcl เหมาะสำหรับใช้ในการจำลองเครือข่ายอินเตอร์เน็ตที่หลากหลาย โดยสามารถจำลองการทำงานของโปรโตคอลต่าง ๆ เช่น TCP, UDP และโปรโตคอลที่ใช้ในระดับชั้น MAC ของเครือข่ายแลน และในงานประยุกต์ต่าง ๆ เช่น HTTP, Telnet, FTP เป็นต้น รวมถึงความสามารถในการจำลองกลไกการทำงานในการจัดลำดับคิวแบบต่าง ๆ และการรับส่งข้อมูลแบบมัลติคาสท์ โดยทำงานบนระบบปฏิบัติการยูนิกซ์ และยังสามารถทำงานบนระบบปฏิบัติการวินโดวส์ได้ด้วย

ส่วนตัวจำลองที่เป็นแบบเชิงพาณิชย์ ได้แก่ OPNET ซึ่งถือเป็นตัวจำลองเพื่อการพาณิชย์ที่มีความสมบูรณ์แบบมากที่สุดตัวหนึ่ง แต่มีราคาค่อนข้างสูง เขียนด้วยภาษาซี นอกจากนี้ยังมีตัวจำลอง BONeS และตัวจำลอง PROPHESY เป็นต้น สรุปได้แสดงดังตาราง 2-1

ตาราง 2-1 ตัวจำลองระบบเครือข่ายที่มีในปัจจุบันโดยสรุป

1. ตัวจำลองระบบเครือข่ายแบบ Open Source								
ตัวจำลอง	NETSIM 3.1	NIST	INSANE	NEST 2.5	REAL 5.0	OMNeT++	NS 2.0	PARSEC
ลักษณะการใช้งาน	เพื่อจำลองระบบเครือข่ายแบบแพ็กเกตสวิตชิ่ง เขียนด้วยภาษาซี	เพื่อศึกษาและประเมินประสิทธิภาพของเครือข่ายแบบ ATM เขียนด้วยภาษาซี	ใช้ในการทดสอบประสิทธิภาพของโปรโตคอลที่หลากหลาย เช่น IP, TCP และ UDP บนเครือข่ายแบบ ATM เขียนด้วยภาษา C++	ใช้ในการจำลองและสร้างอัลกอริทึมและระบบต้นแบบ	เพื่อศึกษาพฤติกรรมแบบพลวัตของวิธีการควบคุมการไหลและความแออัดของข้อมูลในเครือข่ายแบบแพ็กเกตสวิตชิ่ง เขียนด้วยภาษาซี โดยใช้ NEST 2.5 เป็นต้นแบบ	ใช้ในการจำลองโปรโตคอลต่าง ๆ ที่ใช้ในระบบเครือข่ายคอมพิวเตอร์ มัลติโปรเซสเซอร์ และระบบการจัดการ เขียนด้วยภาษาซี	ใช้ในการจำลองเครือข่ายอินเตอร์เน็ตที่หลากหลาย โดยสามารถจำลองการทำงานของโปรโตคอล เช่น TCP, UDP และโปรโตคอลที่ใช้ในระดับชั้น MAC ของเครือข่ายแลน และในงานประยุกต์ต่าง ๆ เช่น HTTP, Telnet, FTP เป็นต้น รวมถึงการจัดลำดับคิวแบบต่าง ๆ เขียนด้วยภาษา C++ และ Otcl	เป็นตัวจำลองแบบขนาน เขียนด้วยโปรแกรมภาษาซี สามารถแก้ไขเพื่อการศึกษา งานวิจัย และองค์กรที่ไม่ได้ทำเพื่อหวังผลกำไร
ระบบปฏิบัติการ	Unix	Unix	Unix Sun Sparcstation Solaris FreeBSD NetBSD IRIX	Unix	Unix SunOS Solaris IRIX BSD4.3 Ultrix UMIPS	Unix Win3.1/95/NT	Unix Linux FreeBSD SunOS Solaris Win95/98/NT	Unix

ตาราง 2-1 (ต่อ)

2. ตัวจำลองระบบเครือข่ายแบบ Commercial				
ตัวจำลอง	OPNET	BONeS	PROPHESY	COMNET III
ลักษณะการใช้งาน	เป็นตัวจำลองเพื่อการพาณิชย์ที่มีความสมบูรณ์แบบมากที่สุดตัวหนึ่ง แต่มีราคาค่อนข้างสูง เขียนด้วยภาษาซี	ใช้สำหรับการออกแบบและจำลองระบบเครือข่ายระดับสูง เพื่อประเมินผลกระทบที่มีต่อประสิทธิภาพของระบบ	เหมาะสำหรับการจำลองเครือข่ายแบบแลนและแวน	ใช้ในการวิเคราะห์ประสิทธิภาพของระบบเครือข่าย โดยสามารถจำลองการทำงานของเครือข่ายแบบ รวมถึงเครือข่ายแบบแวน และเครือข่ายแบบแมน
ระบบปฏิบัติการ	Windows	Windows	Win3.1 ขึ้นไป	Win95/NT