การเปรียบเทียบประสิทธิภาพของเกมออนไลน์บนระหว่างโปรโตคอลไอพีรุ่นที่ 4 และรุ่นที่ 6

โดย

นายอำนาจ ช้างเขียว

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาคอมพิวเตอร์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2551

การเปรียบเทียบประสิทธิภาพของเกมออนไลน์บนระหว่างโปรโตคอลไอพีรุ่นที่ 4 และรุ่นที่ 6

โดย

นายอำนาจ ช้างเขียว

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาคอมพิวเตอร์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2551

PERFORMANCE COMPARISON OF GAME ONLINE ON BETWEEN IPV4 PROTOCOL AND IPV6

By

Amnart Changkeaw

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

Department of Computing

Graduate School

SILPAKORN UNIVERSITY

2008

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร อนุมัติให้วิทยานิพนธ์เรื่อง " การเปรียบเทียบประสิทธิภาพของเกมออนไลน์บนระหว่างโปรโตคอลไอพีรุ่นที่ 4 และรุ่นที่ 6 " เสนอโดย นายอำนาจ ช้างเขียว เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

…….....

(รองศาสตราจารย์ ดร.ศิริชัย ชินะตังกูร)

คณบดีบัณฑิตวิทยาลัย

วันที่.....เดือน..... พ.ศ.....

อาจารย์ที่ปรึกษาวิทยานิพนธ์

อาจารย์ ดร.สุนีย์ พงษ์พินิจภิญโญ

คณะกรรมการตรวจสอบวิทยานิพนธ์

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์)

...../...../.....

..... กรรมการ

(อาจารย์ ดร.วสันต์ ภัทรอธิคม)

...../...../.....

..... กรรมการ

(อาจารย์ ดร.สุนีย์ พงษ์พินิจภิญโญ)

...../...../.....

47307316 : สาขาวิชาวิทยาการคอมพิวเตอร์
คำสำคัญ : เกมออนไลน์ / การเปรียบเทียบ / โปรโตคอลไอพีรุ่นที่ 6

อำนาจ ช้างเขียว : การเปรียบเทียบประสิทธิภาพของเกมออนไลน์บนระหว่างโปรโตคอลไอพีรุ่นที่ 4 และรุ่นที่ 6. อาจารย์ที่ปรึกษาวิทยานิพนธ์ : อ.ดร.สุนีย์ พงษ์พินิจภิญโญ. 59 หน้า.

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อศึกษาหลักการและวิธีการในการจัดการข้อมูล และเปรียบเทียบประสิทธิภาพในการทำงานระหว่างเครือข่ายโปรโตคอล IPv4 และเครือข่ายโปรโตคอล IPv6 โดยในการวิจัยได้มีการจัดทำเครือข่ายสำหรับทำการทดลองเป็น 3 แบบได้แก่ 1) เครือข่ายโปรโตคอล IPv4 2) เครือข่ายโปรโตคอล IPv4แบบผ่านการทำ NAT 3) เครือข่ายโปรโตคอล IPv6 และพัฒนาโปรแกรมจำลองการทำงานของเกมออนไลน์ขึ้นมาเพื่อใช้ในการทดสอบ ซึ่งการทดสอบจะมีการควบคุมให้มีการส่งข้อมูลจำนวนที่เท่ากัน และรูปแบบของข้อมูลเหมือนกันกับเครือข่ายสำหรับการทดลองทั้ง 3 แบบ และใช้จำนวนเครื่องตั้งแต่ 1 เครื่อง 10 เครื่อง และ 30 เครื่องทดลองกับเครือข่ายทั้ง 3 แบบหลายๆ ครั้งเพื่อหาค่าเฉลี่ยความเร็วในการรับส่งข้อมูล

ผลการวิจัยพบว่า หากมีการใช้จำนวนเครื่องน้อยความเร็วในการรับส่งข้อมูลเครือข่ายโปรโตคอล IPv4 และ เครือข่ายโปรโตคอล IPv6 จะมีความเร็วแตกต่างกันเล็กน้อยโดย เครือข่ายโปรโตคอล IPv4 จะเร็วกว่าเล็กน้อย แต่เมื่อมีการใช้เครื่องมาทำการทดลองจำนวนมากขึ้นความเร็วในการรับส่งข้อมูล ประสิทธิภาพของเครือข่ายโปรโตคอล IPv6 มีแนวโน้มดีกว่าของเครือข่ายโปรโตคอล IPv4

ภาควิชาคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ปีการศึกษา 2551

ลายมือชื่อนักศึกษา .....

ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์ .....

47307316: MAJOR : COMPUTER SCIENCE
KEY WORDS : GAME ONLINE/ COMPARISON / PROTOCOL IPV6
AMNART CHANGKEAW : PERFORMANCE COMPARISON OF GAME ONLINE ON BETWEEN IPV4 PROTOCOL AND IPV6. THESIS ADVISOR : SUNEE PONGPINIGPINYO, Ph.D. 59 pp.

This research aims to study the principles and methods in data management. The comparison of the performance between IPv4 network protocols and IPv6 network protocols in the research are conducted for three types of network: 1) IPv4 network protocol 2) IPv4 network protocol with NAT 3) IPv6 network protocols which are developed on game simulation program comes to online tests. The same pattern and same amount of experimental data are used to test on those three types of network. The experiments are also tested several times on 1 machine 10 machines and 30 machines respectively to find the average speed of data transfers.

The experimental results show that a small amount of data transfer speed IPv4 network protocols and IPv6 network protocols are slightly different speed. IPv4 network protocol is slightly faster. Conversely, the trial came to a growing number of machines when performances of IPv6 network protocols are better than the network protocol IPv4 performance.

Department of Computing Graduate School, Silpakorn University Academic Year 2008
Student's signature .....
Thesis Advisor's signature .....

กิตติกรรมประกาศ

ในการวิจัยครั้งนี้สำเร็จลุล่วงไปด้วยดีนั้นผู้วิจัยต้องขอขอบพระคุณอาจารย์ที่ปรึกษา อาจารย์ ดร.สุนีย์ พงษ์พินิจภิญโญ ประธานกรรมการ ผู้ช่วยศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์ และกรรมการผู้ทรงคุณวุฒิ อาจารย์ ดร.วสันต์ ภัทรอธิคม ที่กรุณาให้คำปรึกษา และตรวจสอบความถูกต้องของงานวิจัย และขอขอบคุณเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือ เป็นกำลังใจซึ่งกันและกัน และสุดท้ายนี้ต้องขอบพระคุณคุณพ่อคุณแม่ที่สนับสนุนทุนการศึกษา คอยให้กำลังใจ และเป็นแรงผลักดันให้ผู้วิจัยได้ศึกษาต่อจนสำเร็จการศึกษา

สารบัญ

สารบัญตาราง

สารบัญภาพ

สารบัญแผนภูมิ

บทที่ 1
บทนำ

ปัจจุบันการใช้เทคโนโลยีเครือข่ายอินเตอร์เน็ตได้รับความนิยมอย่างแพร่หลายทั่วโลก และการเติบโตของเครือข่ายอินเตอร์เน็ตเป็นไปอย่างรวดเร็ว ทำให้จำนวนหมายเลขอินเตอร์เน็ตโปรโตคอล (IP Address) ที่มีอยู่ปัจจุบันซึ่งเป็นอินเตอร์เน็ตโปรโตคอลรุ่นที่ 4 (Internet Protocol version 4; IPv4) มีแนวโน้มที่จะหมดไปในอนาคตอันใกล้ อินเตอร์เน็ตโปรโตคอลรุ่นที่ 6 (Internet Protocol version 6; IPv6) จึงได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาสำคัญดังกล่าวโดยมีการปรับปรุงโครงสร้างของตัวโปรโตคอลจากอินเตอร์เน็ตโปรโตคอลรุ่นที่ 4 ที่ใช้งานอยู่อย่างแพร่หลายในปัจจุบัน ให้มีจำนวน IP Address เพิ่มมากขึ้น เพื่อรองรับการขยายตัวของเครือข่ายอินเตอร์เน็ตในอนาคตได้อย่างเพียงพอ นอกจากนี้ยังมีการปรับปรุงในเรื่องของประสิทธิภาพและความปลอดภัย เพื่อให้สามารถตอบสนองตามความต้องการในการใช้งานเทคโนโลยีเครือข่ายอินเตอร์เน็ตในปัจจุบันและอนาคต หลายประเทศได้เริ่มนำอินเตอร์เน็ตโปรโตคอลรุ่นที่ 6 มาใช้งานจริง ในงานหลายๆ ด้าน งานด้านพัฒนาเกมก็เป็นอีกส่วนที่มีความสำคัญเนื่องจากในปัจจุบันซอฟต์แวร์ประเภทนี้สามารถสร้างรายได้ให้กับผู้ผลิตเป็นจำนวนมาก และมีการขยายตัวอย่างรวดเร็ว

1. ความเป็นมาและความสำคัญของปัญหา

การเชื่อมต่อเครือข่ายอินเตอร์เน็ตปัจจุบันส่วนใหญ่อยู่บนพื้นฐานการทำงานของอินเตอร์เน็ตโปรโตคอลรุ่นที่ 4 (Internet Protocol Version 4; IPv4) ซึ่งเป็นมาตรฐานในการสื่อสารบนเครือข่ายอินเตอร์เน็ตตั้งแต่ปี ค.ศ. 1981 องค์ประกอบสำคัญในการทำงานของอินเตอร์เน็ตโปรโตคอลได้แก่ หมายเลขอินเตอร์เน็ต (IP Address) ที่ใช้ในการอ้างอิงกันระหว่างเครื่องคอมพิวเตอร์และอุปกรณ์เครือข่ายต่างๆ ทั่วโลก เปรียบเสมือนการใช้งานโทรศัพท์ที่ต้องมีหมายเลขโทรศัพท์ในการอ้างอิง เครื่องคอมพิวเตอร์บนเครือข่ายอินเตอร์เน็ตต้องมีหมายเลขอินเตอร์เน็ตที่ได้รับการจัดสรรและไม่ซ้ำกัน

เนื่องจากจำนวนหมายเลขอินเตอร์เน็ตของ IPv4 ที่ใช้งานกันอยู่ในปัจจุบันมีอย่างจำกัด และการขยายตัวของเครือข่ายในช่วงที่ผ่านมามีอัตราการเจริญเติบโตอย่างรวดเร็ว ทำให้จำนวนหมายเลขอินเตอร์เน็ตของ IPv4 กำลังจะถูกใช้หมดไป ไม่เพียงพอกับการใช้งานอินเตอร์เน็ตในอนาคต หากเหตุการณ์ดังกล่าวเกิดขึ้นหมายความว่าเราจะไม่สามารถเชื่อมต่อเครือข่ายเข้าสู่อินเตอร์เพิ่มขึ้นได้ ดังนั้นคณะทำงานของ IETF (The Internet Engineering Task Force) จึงได้พัฒนา

อินเตอร์เน็ตโปรโตคอลรุ่นใหม่ขึ้นคือ IPv6 โดยมีวัตถุประสงค์ เพื่อปรับปรุงโครงสร้างของอินเตอร์เน็ตโปรโตคอลให้สามารถรองรับหมายเลข IP Address จำนวนมาก และเพื่อปรับปรุงคุณลักษณะอื่นอีกหลายประการ ทั้งในเรื่องของความปลอดภัย การลองรับโปรแกรมแอพพลิเคชั่นใหม่ๆ ที่เกิดขึ้นในอนาคต และการเพิ่มประสิทธิภาพในการประมวลผลแพ็คเกจให้ดีขึ้น ทำให้สามารถตอบสนองการขยายตัวและความต้องการใช้งานเทคโนโลยีเครือข่ายอินเตอร์เน็ตในอนาคตได้เป็นอย่างดี

การผลักดันการนำเอา IPv6 มาใช้งานจริงนั้นจะอยู่ที่ทวีปยุโรปและเอเชียเป็นหลัก สาเหตุที่สำคัญประการแรกคือ ในปัจจุบันทวีปอเมริกาเหนือมีส่วนแบ่งของ IPv4 Address มากถึงร้อยละ 70 ของ IP Address ทั้งหมดที่มีอยู่ในโลก ดังนั้นทวีปอเมริกาจึงยังไม่ค่อยเห็นความสำคัญของ IPv6 มากนัก ซึ่งตรงกันข้ามกับทางทวีปยุโรปและเอเชียต่างก็พบกับปัญหาการมี IP Address ไม่เพียงพอต่อการใช้งานของผู้ใช้อินเตอร์เน็ต สาเหตุอีกประการก็เนื่องมาจากเทคโนโลยีโทรศัพท์เคลื่อนที่ยุคที่ 3 (3G Wireless Technology) ทั้งทางยุโรปและเอเชียต่างก็มีความต้องการสูงทางด้านเทคโนโลยี 3G ซึ่งเทคโนโลยีนี้ทำให้เกิดความต้องการ IP Address เพิ่มมากขึ้น จะพบว่าผู้ผลิตฮาร์ดแวร์ ซอฟต์แวร์ และองค์การที่ทำหน้าที่มาตรฐานต่างๆ ในทวีปยุโรปและทวีปเอเชียต่างส่งสัญญาณที่จะแก้ปัญหาการขาดแคลน IP Address

อินเตอร์เน็ตโปรโตคอลมีการนำมาใช้งานกันอย่างแพร่หลาย ในงานหลายๆ ด้าน นอกเหนือจากการนำมาใช้งานอินเตอร์เน็ต งานอีกประเภทที่มีการนำมาใช้งานอย่างเห็นได้ชัดมากที่สุดก็คือ งานด้านความบันเทิงได้แก่เกมต่างๆ จะเห็นได้ว่าในปัจจุบันเกมที่เล่นผ่านเครือข่าย รวมถึงเกมออนไลน์จะได้รับความนิยมจากผู้บริโภคเป็นอย่างมาก จากการที่มีผู้ให้บริการเกมออนไลน์นำเกมต่างๆ มาเปิดให้บริการเพิ่มมากขึ้นในปัจจุบัน และมีอัตราการขยายตัวอย่างรวดเร็ว เกมออนไลน์ต่างๆ ที่นำมาให้บริการเหล่านี้สามารถทำรายได้ให้กับผู้เปิดบริการเป็นจำนวนมาก อย่างไรก็ตามการให้บริการเกมออนไลน์ในปัจจุบันยังคงให้บริการบนอินเตอร์เน็ตโปรโตคอลรุ่นที่ 4 (IPv4)

ในอนาคตเมื่อมีการนำเอาอินเตอร์เน็ตโปรโตคอลรุ่นที่ 6 (IPv6) มาใช้งานแทนอินเตอร์เน็ตโปรโตคอลรุ่นเก่าทำให้เกิดการเปลี่ยนแปลงในหลายด้านทั้งในด้านซอฟต์แวร์และฮาร์ดแวร์ จะเห็นได้ว่าในส่วนของฮาร์ดแวร์ผู้ผลิตอุปกรณ์ได้มีการผลิตอุปกรณ์เครือข่ายรองรับการทำงานอินเตอร์เน็ตโปรโตคอลรุ่นที่ 6 ออกมาวางจำหน่ายเป็นจำนวนมากขึ้น เพื่อรองรับการขยายตัวของเครือข่ายอินเตอร์เน็ตในอนาคต ที่มีแนวโน้มการขยายตัวอย่างรวดเร็ว ในด้านซอฟต์แวร์ก็มีหลายๆ บริษัทที่มีการพัฒนาซอฟต์แวร์ของตนเองให้รองรับการทำงานของไอพีรุ่นที่ 6 รวมทั้งเกมออนไลน์ก็ต้องมีการพัฒนาให้สามารถรองรับการทำงานของไอพีรุ่นที่ 6 ด้วยเช่นกัน

2. วัตถุประสงค์ของงานวิจัย

2.1 เพื่อศึกษาหลักการและวิธีการในการจัดการข้อมูลและกระบวนการรับส่งข้อมูลบนโปรโตคอล IPv6 และการพัฒนาโปรแกรมบนโปรโตคอล IPv6

2.2 เพื่อทดสอบและประเมินผลเกมออนไลน์ที่พัฒนาขึ้น เทียบประสิทธิภาพที่เกิดขึ้นระหว่างโปรโตคอล IPv4 และโปรโตคอล IPv6

3. ขอบเขตของการศึกษา

งานวิจัยนี้เป็นการพัฒนาเกมออนไลน์บนโปรโตคอลไอพีรุ่นที่ 6 โดยขอบเขตในการศึกษาและวิจัยเพื่อพัฒนาชุดคำสั่งสำหรับใช้ในการพัฒนาเกมบนเครือข่าย IPv6 ชุดคำสั่งดังกล่าวสามารถที่จะนำไปใช้ประโยชน์ในการพัฒนาโปรแกรมสำหรับใช้งานบนเครือข่าย IPv6 อีกด้วย ดังนั้นในการวิจัยจะมุ่งเน้นไปในส่วนของการจัดการ IPv6 เป็นหลัก งานวิจัยแบ่งการทำงานเป็น 2 ส่วนคือในส่วนของ Game Server และ Game Client โดยการทำงานจะทำงานบนเครือข่าย IPv6 เพื่อทดสอบและประเมินผลเกมออนไลน์ที่พัฒนาขึ้น เทียบประสิทธิภาพที่เกิดขึ้นระหว่าง IPv4 และ IPv6

4. ขั้นตอนการศึกษา

ในงานวิจัยนี้สามารถแบ่งขั้นตอนในการศึกษาได้ 6 ขั้นตอน ดังนี้

4.1 เก็บรวบรวมข้อมูลจากเอกสารและข้อมูลที่เกี่ยวข้อง ในลักษะทั่วไปของโปรโตคอลไอพีรุ่นที่ 6 ทฤษฎีต่างๆ ที่ใช้ในการพัฒนาเกมออนไลน์บนไอพีรุ่นที่ 6 และเทคนิคการควบคุมข้อมูลของเกมออนไลน์บนเครือข่าย

4.2 วิเคราะห์และเลือกใช้ทฤษฎีและอัลกอริทึมที่เหมาะสม

4.3 เขียนโปรแกรมและทำการทดลองเกมออนไลน์บนเครือข่ายไอพีรุ่นที่ 6

4.4 วิเคราะห์ผลการทดลองเปรียบเทียบประสิทธิภาพในการทำงานของเกมออนไลน์เมื่อมีการใช้งานบนเครือข่ายไอพีรุ่นที่ 6

4.4.1 ศึกษาหลักการและวิธีการในการจัดการข้อมูลและกระบวนการรับส่งข้อมูลบนเครือข่าย IPv6 และการพัฒนาโปรแกรมบนเครือข่าย IPv6

4.4.2 ศึกษาประสิทธิภาพการจัดการกับข้อมูลของเกมออนไลน์บนเครือข่ายไอพีรุ่นที่ 6

4.5 สรุปผลการทดลอง

4.6 รวบรวมข้อเสนอแนะ

5. นิยามศัพท์เฉพาะ

5.1 IPv6 (Internet Protocol version 6) เป็นเวอร์ชันล่าสุดของ Internet Protocol และได้รวมผลิตภัณฑ์ที่สนับสนุน IP มาเป็นส่วนหนึ่งด้วย รวมถึงระบบปฏิบัติการหลัก IPv6 ได้รับการเรียกว่า "IPng" (IP Next Generation) โดยปกติ IPv6 เป็นกลุ่มของข้อกำหนดจาก Internet Engineering Task Force (IETF) โดย IPv6 ได้รับการออกแบบให้ปฏิรูปกลุ่มของการปรับปรุง IP เวอร์ชัน 4 โดย host ของเครือข่ายและ node แบบ intermediate ซึ่ง IPv4 หรือ IPv6 สามารถดูแลแพ็คเกจของ IP เวอร์ชันอื่น ผู้ใช้และผู้ให้บริการสามารถปรับรุ่นเป็น IPv6 โดยอิสระ

การปรับปรุงที่ชัดเจนของ IPv6 คือความยาวของ IP address เปลี่ยนจาก 32 เป็น 128 การขยายดังกล่าวเพื่อรองรับการขยายของอินเตอร์เน็ต และเพื่อหลีกเลี่ยงการขาดแคลนของตำแหน่งเครือข่าย

IPv6 ได้กำหนดกฎในการระบุตำแหน่งเป็น 3 ประเภทคือ unicast (host เดี่ยวไปยัง host เดี่ยวอื่น ๆ) anycast (host เดี่ยวไปยัง host หลายตัวที่ใกล้ที่สุด) multicast (host เดี่ยวไปยัง host หลายตัว) ส่วนเพิ่มที่พิเศษของ IPv6 คือ

5.1.1 ตัวเลือกในการระบุส่วนขยายของส่วนหัว ได้รับการตรวจสอบเฉพาะจุดหมาย ดังนั้นความเร็วของระบบเครือข่ายสูงขึ้น

5.1.2 ตำแหน่ง anycast ทำให้มีความเป็นไปได้ของการส่งข้อความไปยังหลาย ๆ gateway ที่ใกล้ที่สุดด้วยแนวคิดว่าให้บุคคลใด ๆ บริหารการส่งแพ็คเกจไปยังบุคคลอื่น anycast สามารถใช้ในการปรับปรุงตาราง routing ตลอดเส้นทาง

5.1.3 แพ็คเกจได้รับการระบุให้มีการไหลชนิดพิเศษได้ ทำให้แพ็คเกจที่เป็นส่วนของมัลติมีเดียที่ต้องการ นำเสนอแบบ real time สามารถมีคุณภาพการให้บริการที่สูง

5.1.4 ส่วนหัวของ IPv6 รวมถึงส่วนขยายยินยอมให้แพ็คเกจระบุกลไกแหล่งต้นทาง สำหรับการรวมข้อมูล และรักษาความลับ

5.2 Multicast ความหมาย เป็นการสื่อสารระหว่างผู้ส่ง 1 รายกับผู้รับหลายรายบนระบบเครือข่าย การใช้โดยทั่วไป รวมถึงการปรับปรุงจากสำนักงาน และเอกสารตามระยะเวลา ของจดหมายข่าว เมื่อรวมกับ anycast, unicast และmulticast ซึ่งเป็นประเภทแพ็คเกจใน Internet Protocol Version 6 (IPV 6)

Multicast สนับสนุนเครือข่ายข้อมูลแบบไร้สาย ซึ่งเป็นส่วนหนึ่งของเทคโนโลยี cellular digital packet data (CDPD)

Multicast ใช้สำหรับโปรแกรม MBone เป็นระบบที่ยินยอมให้ผู้ใช้ ที่ตำแหน่ง bandwidth สูงบนอินเตอร์เน็ต โดย Mbone multicast ใช้โปรโตคอล ที่ยอมให้สัญญาณจับกลุ่มเป็นแพ็คเกจ TCP/IP เมื่อผ่านส่วนของอินเตอร์เน็ต แต่ไม่สามารถดูแลโปรโตคอล Multicast โดยตรง

5.3 Anycast ความหมายใน Internet protocol เวอร์ชัน 6 (IPV 6) anycast เป็นการติดต่อระหว่างผู้ส่งรายเดียวกันกับผู้รับหลายรายที่เป็นกลุ่มที่ใกล้ที่สุด เป็นคำศัพท์ที่แตกต่างจาก multicast ที่หมายถึงการติดต่อระหว่างผู้ส่งรายเดียวกับผู้รับหลายราย และ unicast เป็นการติดต่อระหว่างผู้ส่งรายเดียวกับผู้รับรายเดียวในเครือข่าย

Anycasting เป็นการออกแบบที่ให้เครื่อง host 1 เครื่อง กำหนดประสิทธิภาพในการปรับตาราง router สำหรับกลุ่มของ host โดย IPV 6 สามารถให้ gateway host ที่ใกล้ที่สุดและส่งแพ็คเกจไปยัง host โดยการติดต่อแบบ unicast ในอีกลักษณะหนึ่ง host สามารถใช้ anycast ไปยัง host อื่น ๆ ในกลุ่มจนกระทั่งตาราง router ทั้งหมดมีการปรับปรุง

5.4 Unicast หมายถึง การสื่อสารระหว่างผู้ส่งรายเดียว กับผู้รับรายเดียวบนเครือข่าย คำนี้แตกต่างจาก multicast ซึ่งเป็นการสื่อสารระหว่างผู้ส่งรายเดียว กับผู้รับหลายราย และ anycast ซึ่งเป็นการสื่อสารระหว่างผู้ส่งหลายราย กับกลุ่มของผู้รับที่ใกล้ที่สุดในเครือข่าย คำก่อนหน้า คือ การสื่อสารแบบ Point-to-Point มีความหมายคล้ายกับ unicast สำหรับ Internet Protocol Version 6 (IPv6) สนับสนุน unicast, anycast และ multicast

5.5 Tunneling หมายถึง ความสัมพันธ์กับอินเตอร์เน็ต โดยใช้อินเตอร์เน็ต เป็นส่วนของเครือข่ายความปลอดภัยส่วนตัว "tunnel" เป็นเส้นทางเฉพาะที่ให้ข่าวสาร หรือไฟล์ของบริษัทเดินทางผ่านอินเตอร์เน็ตโปรโตคอล หรือกลุ่มของกฎการสื่อสารที่เรียกว่า Point-to-Point Tunneling Protocol ได้รับการเสนอขึ้นในการสร้าง virtual private network โดยผ่าน "tunnel" บนอินเตอร์เน็ต ซึ่งหมายความว่าบริษัทอาจจะไม่จำเป็นต้องมี lease line ของตัวเอง สำหรับการสื่อสารในพื้นที่กว้าง แต่สามารถใช้เครือข่ายสาธารณะได้ PPTP สนับสนุนโดย Microsoft และบริษัทอื่น และ Layer 2 Forwarding ซึ่งเสนอโดย CISCO system เป็นข้อเสนอหลัก สำหรับมาตรฐานใหม่ของ Internet Engineering Task Force (IETF) โดย PPTP ซึ่งเป็นส่วนขยายของโปรโตคอล Point-to-Point Protocol ทำให้ผู้ใช้คอมพิวเตอร์ส่วนบุคคลที่สนับสนุน PPP client จะสามารถใช้ผู้ให้บริการอินเตอร์เน็ตอิสระ ในการเชื่อมต่ออย่างปลอดภัย กับเครื่องแม่ข่ายของบริษัทในทุกที่

6. ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย มีดังต่อไปนี้

6.1 หลักการพัฒนาเกมออนไลน์ที่ทำงานบนเครือข่าย IPv6

6.2 การพัฒนาโปแกรมบนเครือข่าย IPv6 สามารถทำได้ง่ายและสะดวกสบายขึ้น

6.3 ได้เรียนรู้หลักและวิธีการในการจัดการข้อมูลและกระบวนการรับส่งข้อมูลบนเครือข่าย IPv6

6.4 ได้เรียนรู้การนำเอา IPv6 ไปประยุกต์ใช้งานจริง

6.5 ผลการเปรียบเทียบประสิทธิภาพที่เกิดขึ้นระหว่างโปรโตคอลไอพีรุ่นที่ 4 และโปรโตคอลไอพีรุ่นที่ 6

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื่องจากงานวิจัยทางด้านโปรโตคอลไอพีรุ่นที่ 6 มีเป็นจำนวนมากมาย ผู้วิจัยจึงได้ศึกษาถึงขั้นตอนวิธีการและขั้นตอนต่างๆ เพื่อนำไปสู่การวิจัยการนำเอาคุณสมบัติเด่นของโปรโตคอลไอพีรุ่นที่ 6 มาใช้กับเกมออนไลน์ โดยมีการนำเอาทฤษฎีและงานวิจัยที่เกี่ยวข้องและมีความใกล้เคียงโดยเฉพาะกับงานวิจัยเกี่ยวกับโปรโตคอลไอพีรุ่นที่ 6 และงานวิจัยเกี่ยวกับเกมต่างๆ เพื่อเป็นแนวทางในการศึกษา โดยมีการแยกเป็นกลุ่มที่เกี่ยวข้องดังต่อไปนี้

1. ทฤษฎีเกี่ยวกับเครือข่ายและโปรโตคอล
2. ทฤษฎีเกี่ยวกับฐานข้อมูล
3. งานวิจัยที่เกี่ยวกับ IPv6 และ Quality of Service (QoS)
4. งานวิจัยที่เกี่ยวกับ IPv6 และ Mobility
5. งานวิจัยที่เกี่ยวกับ Game Traffic
6. งานวิจัยเกี่ยวกับ IPv6 ในประเทศไทย

1. ทฤษฎีเกี่ยวกับเครือข่ายและโปรโตคอล

1.1 สถาปัตยกรรมเครือข่าย

การที่มนุษย์สามารถสื่อสารกันได้อย่างมีประสิทธิภาพนั้น เนื่องจากใช้ภาษาเดียวกัน เช่น ภาษาไทย ภาษาอังกฤษ เป็นต้น ถ้าใช้คนละภาษาก็คงสื่อสารกันไม่ได้ความ คอมพิวเตอร์ก็เช่นเดียวกันกับมนุษย์ การที่เครื่องคอมพิวเตอร์เครื่องหนึ่งจะสามารถสื่อสารกับคอมพิวเตอร์อีกเครื่องหนึ่งได้จำเป็นต้องใช้ “ภาษา” เดียวกัน ภาษาที่ว่านี้ศัพท์ทางคอมพิวเตอร์นี้เรียกว่า “โปรโตคอล” ดังนั้นคอมพิวเตอร์ที่สื่อสารกันได้ต้องใช้โปรโตคอลเดียวกัน เช่น คอมพิวเตอร์ที่เชื่อมต่อกับอินเตอร์เน็ตจะใช้ “ภาษา” หรือโปรโตคอล TCP/IP ส่วนคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการอินเตอร์เน็ตแวร์ (Netware) ก็จะใช้ “ภาษา” หรือโปรโตคอล IPX/SPX ในการสื่อสารกัน เป็นต้น

ในปัจจุบันฮาร์ดแวร์และซอฟต์แวร์ที่ใช้กับระบบคอมพิวเตอร์มีหลายชนิด บางชนิดก็ใช้งานร่วมกันได้ แต่บางชนิดก็ใช้ร่วมกันไม่ได้ ผู้ใช้บางคนอาจมีความจำเป็นต้องสื่อสารกับผู้ใช้ที่เชื่อมต่อกับเครือข่ายอื่น เครือข่ายคอมพิวเตอร์ในปัจจุบันส่วนใหญ่มักจะแตกต่างกันทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ บางครั้งก็สื่อสารกันไม่ได้ เนื่องจากคอมพิวเตอร์ที่อยู่ในเครือข่ายต่างประเภท

กันจะใช้ภาษาคนละภาษา ในการที่จะทำให้การติดต่อสื่อสารเป็นไปได้นั้นคอมพิวเตอร์เหล่านั้นจำเป็นต้องใช้ภาษาเดียวกัน ด้วยเหตุนี้จึงได้มีการพัฒนาภาษา หรือโปรโตคอลขึ้นมาเพื่อให้คอมพิวเตอร์ที่กล่าวมานี้สามารถสื่อสารกันได้

1.2 ความหมายของโปรโตคอล

การเชื่อมต่อคอมพิวเตอร์ให้เป็นเครือข่ายด้วยสายสัญญาณนั้นเป็นขั้นตอนง่ายสำหรับสร้างเครือข่ายแต่ส่วนที่ท้าท้ายคือ การพัฒนามาตรฐานเพื่อให้คอมพิวเตอร์และอุปกรณ์เครือข่ายที่ผลิตโดยบริษัทต่างๆ สามารถติดต่อสื่อสารกันได้ ซึ่งมาตรฐานนี้คือ โปรโตคอล หรือสรุปสั้นๆ โปรโตคอล คือ กฎ ขั้นตอน และรูปแบบของข้อมูลที่ใช้ในการสื่อสารระหว่างคอมพิวเตอร์สองเครื่องใดๆ ที่เชื่อมต่อกันเป็นเครือข่าย

ตัวอย่างที่เห็นได้ชัดของโปรโตคอล เช่น การสื่อสารโดยใช้โทรศัพท์ ซึ่งจะมีขั้นตอนต่างๆ ที่จะต้องทำก่อนที่จะพูดคุยกันได้ เช่น โดยส่วนใหญ่คำแรกที่พูดเมื่อใช้โทรศัพท์คือ “ฮัลโหล” คือ คำทักทายของภาษาท้องถิ่นอื่นๆ การทักทายกันนี้เป็นสัญญาณให้คู่สนทนาทราบว่าการเชื่อมต่อกันสำเร็จ ขั้นตอนต่อไป คือ อีกฝ่ายจะตอบด้วย “ฮัลโหล” เช่นกัน ซึงเป็นสัญญาณบอกให้ทราบอีว่าการติดต่อสื่อสารเป็นไปได้ทั้งสองทาง ถ้าทั้งสองฝ่ายที่สนทนากันรู้จักกันมาก่อน การสนทนาก็จะเข้าสู้เรื่องได้ทันที แต่ถ้าหากว่าทั้งสองฝ่ายยังไม่รู้จักกัน ก็จะมีขั้นตอน หรือ โปรโตคอลอื่นเพิ่มอีก เพื่อช่วยให้ทั้งสองฝ่ายรู้จักกันก่อนที่จะเริ่มเรื่องที่จะสนทนากันจริงๆ

การสนทนากันของคอมพิวเตอร์ก็ไม่ได้แตกต่างจากตัวอย่างข้างต้นมากนัก การเชื่อมต่อกันของคอมพิวเตอร์เป็นเพียงส่วนหนึ่งของการสร้างระบบเครือข่าย แต่การสื่อสารที่มีความหมาย เช่น การแชร์กันใช้ทรัพยากรของแต่ละฝ่าย ทำให้เครือข่ายคอมพิวเตอร์สมบูรณ์ วิวัฒนาการของเครือข่ายถือได้ว่าเป็นการปฏิวัติครั้งใหญ่ของโครงสร้างของเทคโนโลยีสารสนเทศ

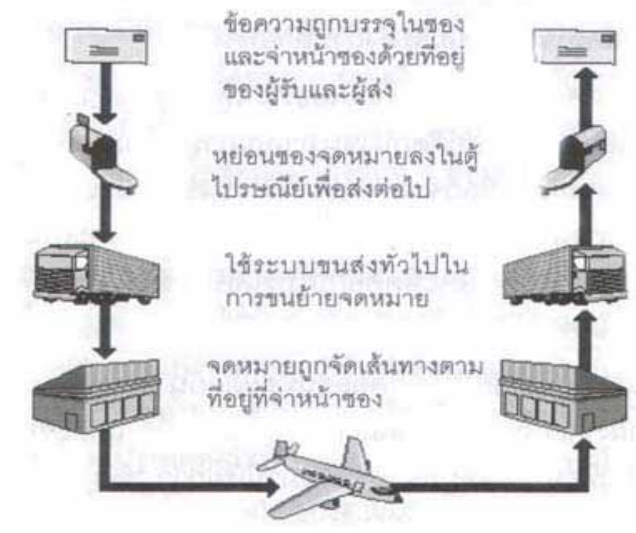
โปรโตคอลของเครือข่ายบางที่อาจเรียกว่า “สถาปัตยกรรมเครือข่าย (Network Architecture)” เนื่องจากเครือข่ายคอมพิวเตอร์ในปัจจุบันเป็นระบบที่ซับซ้อนมาก ทำให้ยากต่อการออกแบบโดยคนๆเดียว หรือคนกลุ่มเดียว เพื่อให้การพัฒนาระบบเป็นไปอย่างมีประสิทธิภาพและง่ายขึ้น จึงมีการแบ่งโปรโตคอลออกเป็นชั้นๆ หรือ เลเยอร์ การทำงานในแต่ละเลเยอร์จะไม่ซ้ำซ้อนกัน ซึ่งเลเยอร์ที่อยู่ต่ำกว่าจะทำหน้าที่ให้บริการ (Service) กับชั้นที่สูงกว่า โดยเลเยอร์ที่สูงกว่าไม่จำเป็นต้องทราบถึงรายละเอียดว่าเลเยอร์ที่อยู่ต่ำกว่ามีวิธีให้บริการอย่างไร เพียงแค่รู้ว่ามีบริการอะไรบ้าง แต่ละบริการคืออะไรก็เพียงพอ ซึ่งแนวคิดนี้จะเรียกว่า “เทคโนโลยีเลเยอร์ (Layer Technology)”

ในตอนต่อไปขอยกตัวอย่างชุดโปรโตคอลต่างๆ ที่ใช้จริงในระบบเครือข่ายปัจจุบัน ชุดโปรโตคอลที่จะกล่าวถึงมีแบบอ้างอิง OIS และชุดโปรโตคอล TCP/IP ที่ใช้ในอินเตอร์เน็ต

1.3 แบบอ้างอิง OSI

องค์การมาตรฐานนานาชาติ (The International Organization for Standardization) และใช้อักษรย่อ “ISO” ซึ่งคนส่วนใหญ่เข้าใจว่าย่อมาจาก “International Standard Organization” แต่จริงๆแล้วไม่ใช่ อย่างไรก็ตาม ISO เป็นองค์กรที่ออกแบบโปรโตคอล OSI (Open System Interconnect) หรือโปรโตคอลการเชื่อมต่อเครือข่ายแบบเปิด จุดมุ่งหมายของการพัฒนามาตรฐานนี้เพื่อให้คอมพิวเตอร์และอุปกรณ์เครือข่ายที่ผลิตโดยบริษัทต่างๆ สามารถทำงานร่วมกันได้ ซึ่งโปรโตคอลนี้ส่วนใหญ่จะเรียกว่า “แบบอ้างอิง OSI (OSI Reference Model)” เหตุที่เรียกแบบอ้างอิงเนื่องจากโปรโตคอลชุดนี้ไม่ได้ถูกใช้งานอย่างแพร่หลาย เหมือนโปรโตคอลชุดอื่นๆ เช่น โปรโตคอล TCP/IP ที่ใช้อย่างแพร่หลาย เช่นในระบบเครือข่ายอินเตอร์เน็ต แต่เนื่องจากแบบอ้างอิง OSI มีการออกแบบโครงสร้างค่อนข้างสมบูรณ์มากที่สุด ด้วยเหตุนี้จึงใช้โปรโตคอลชุดนี้เป็นแบบอ้างอิงในการพัฒนาโปรโตคอลชุดอื่นๆ อีกทั้งยังเป็นระบบที่ง่ายต่อการอธิบายถึงกลไกการทำงานของโปรโตคอลเครือข่าย ดังนั้นเมื่อกล่าวถึงเครือข่าย ส่วนใหญ่จะใช้โปรโตคอลนี้เป็นแบบอ้างอิงในการอธิบาย ซึ่งเป็นที่มาของการเรียกโปรโตคอลชุดนี้ว่า แบบอ้างอิง OSI

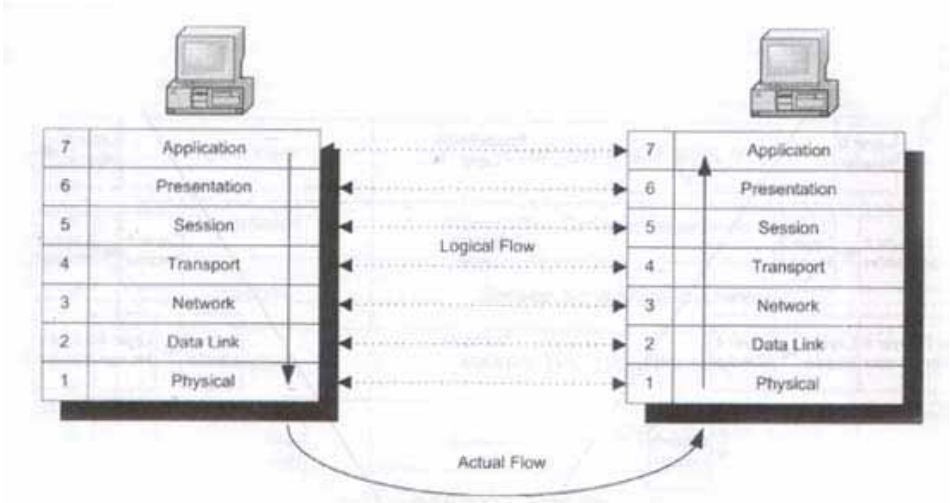
แบบอ้างอิงนี้จะแบ่งขั้นตอนการสื่อสารระหว่างคอมพิวเตอร์ออกเป็น 7 ขั้นตอน หรือเลเยอร์ การแบ่งเป็นขั้นตอนต่างๆเหล่านี้จะยึดหลักเหมือนกับการสื่อสารทั่วไป เช่น การรับส่งจดหมายทางไปรษณีย์ มีขั้นตอนคร่าวๆ ดังภาพที่ 1



ภาพที่ 1 ระบบการขนส่งจดหมาย

ที่มา : จตุชัย แพงจันทร์ และ อนุโชต วุฒิพรพงษ์, เจาะระบบ Network (กรุงเทพมหานคร : โรงพิมพ์ด่านสุทธาการพิมพ์, 2547), 31.

จากตัวอย่างการสื่อสารด้วยจดหมายนั้น จะเห็นว่าขั้นตอนของผู้ส่งกับขั้นตอนของผู้รับนั้นจะคล้ายๆกัน เพียงแต่ลำดับเหตุการณ์นั้นตรงกันข้ามกันเท่านั้น การสื่อสารระหว่างคอมพิวเตอร์ก็จะเป็นไปในลักษณะคล้ายๆกัน กล่าวคือ จะแบ่งการทำงานออกเป็นขั้นตอนต่างๆ เพื่อให้ง่ายต่อการจัดการข้อความที่ส่ง



ภาพที่ 2 แบบอ้างอิง OSI

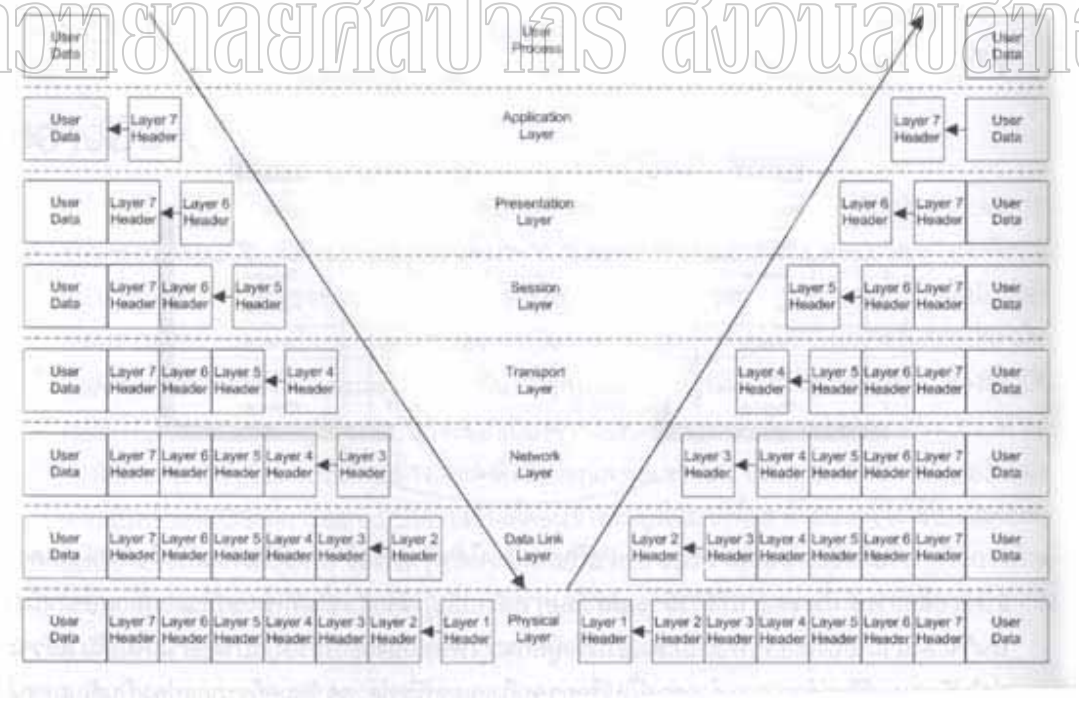
ที่มา : จตุชัย แพงจันทร์ และ อนุโชต วุฒิพรพงษ์, เจาะระบบ Network (กรุงเทพมหานคร : โรงพิมพ์ด่านสุทธาการพิมพ์, 2547), 31.

การสื่อสารระหว่างคอมพิวเตอร์สองเครื่องใดๆ ในเครือข่ายจะมีขั้นตอนการสื่อสารดังแสดงดังภาพที่ 2 การสื่อสารจะเริ่มจากการที่ผู้ใช้มีข้อมูลที่ต้องการส่งไปยังผู้ใช้อีกคนหนึ่งที่ใช้คอมพิวเตอร์อีกเครื่องหนึ่ง ข้อมูลนั้นจะส่งผ่านไปยังเลเยอร์ที่ 7 ในเลเยอร์นี้จะถูกดัดแปลงและใส่ข้อมูลบางอย่างเพิ่มเติม แล้วจะถูกส่งต่อไปยังเลเยอร์ที่อยู่ต่ำกว่า จะทำอย่างนี้ไปเรื่อยๆ จนถึงเลเยอร์ที่อยู่ต่ำสุด ข้อมูลก็จะถูกแปลงเป็นสัญญาณเพื่อส่งผ่านสายสัญญาณ หรือสื่อกลางที่เชื่อมกันระหว่างคอมพิวเตอร์สองเครื่องนี้ จนถึงเครื่องรับ ส่วนกระบวนการในการรับข้อมูลของเครื่องระบบนั้นก็จะทำในทางตรงกันข้ามกับเครื่องที่ส่งข้อมูล กล่าวคือ จะเริ่มกระบวนการรับข้อมูลจากเลเยอร์ 1 ก่อน และส่งต่อไปเรื่อยๆ จนถึงเลเยอร์ 7 และส่งต่อให้แอพพลิเคชันของผู้ใช้ต่อไป

การทำงานในแต่ละเลเยอร์จะเป็นไปในรูปแบบที่ว่าแต่ละเลเยอร์จะคอยให้บริการให้กับเลเยอร์ที่อยู่สูงกว่าและติดกัน โดยที่เลเยอร์ที่ใช้บริการไม่สนใจในรายละเอียดว่าเลเยอร์ที่ให้บริการจะมีวิธีการทำงานอย่างไร เพียงแค่รู้ว่าข้อมูลนั้นจะส่งถึงปลายทางก็พอ

การจัดเรียงโปรโตคอลเป็นชั้นๆ หรือเลเยอร์นี้ก็เพื่อจำลองการไหลของข้อมูลจากเครื่องส่งถึงเครื่องรับแต่ละชั้นจะส่งข้อมูลต่อไปยังยังชั้นที่อยู่ติกัน เช่น ถ้าเป็นการส่งข้อมูล ข้อมูลจะถูกส่งต่อไปยังชั้นที่อยู่ต่ำกว่าถัดลงไป แต่ถ้าเป็นการรับข้อมูล ข้อมูลก็จะส่งจากข้างล่างขึ้นบน แต่ละชั้นจะมีจุดเชื่อมต่อกับชั้นที่อยู่ใกล้เคียงเพื่อให้การติดต่อสื่อสารสำเร็จได้ การติดต่อสื่อสารของแต่ละชั้นจะเป็นแบบเพียร์ทูเพียร์ (Peer-to-Peer) หมายความว่า โปรโตคอลชั้นที่ฝั่งส่งจะติดต่อกับโปรโตคอลชั้นเดียวกันที่อยู่ฝั่งรับ ข้อมูลที่อยู่ในชั้นนี้จะมีความหมายเฉพาะกับโปรโตคอลที่อยู่ระดับเดียวกันของฝั่งตรงข้ามเท่านั้น

ถึงแม้ว่าข้อมูลจะถูกส่งไปยังโปรโตคอลชั้นที่อยู่ติดกัน แต่ละชั้นจะคิดว่าเป็นเหมือนกับการส่งข้อมูลไปยังชั้นเดียวกันที่อยู่ฝั่งหนึ่ง เพื่อให้การสื่อสารแบบนี้เกิดขึ้นได้ แต่ละชั้นจะทำการเพิ่มข้อมูลบางอย่างให้กับข้อมูลที่ได้รับมาจากชั้นที่อยู่บน ข้อมูลส่วนที่เพิ่มมานี้เรียกว่า “ข้อมูลส่วนหัว (Header)” ซึ่งข้อมูลในส่วนนี้จะเข้าใจและนำไปใช้เฉพาะโปรโตคอลที่อยู่ระดับเดียวกันของอีกฝั่งเท่านั้น ทางฝั่งส่งข้อมูลส่วนหัวจะถูกเพิ่ม ส่วนฝั่งรับข้อมูลส่วนหัวจะถูกนำออก กระบวนการนี้แสดงดังภาพที่ 3



ภาพที่ 3 การใช้ข้อมูลส่วนหัวในการสื่อสารระหว่างชั้น

ที่มา : จตุชัย แพงจันทร์ และ อนุโชต วุฒิพรพงษ์, เจาะระบบ Network (กรุงเทพมหานคร : โรงพิมพ์ด่านสุทธาการพิมพ์, 2547), 32.

จากภาพที่ 3 ข้อมูลในชั้นที่ 4 จะประกอบด้วยข้อมูล ที่ส่งผ่านมาจากชั้นที่ 5 แล้วเพิ่มข้อมูลส่วนหัวเพื่อเป็นข้อมูลให้ชั้นที่ 4 ของอีกฝั่งตรงข้าม เสร็จแล้วข้อมูลชุดใหม่นี้จะถูกส่งต่อไปยังชั้นที่ 3 เมื่อชั้นที่ 3 ได้รับข้อมูลก็จะทำการแบ่งข้อมูลนี้เป็นแพ็กเกจ (Packet) ย่อยๆ แล้วเพิ่มข้อมูลส่วนหัวให้แต่ละแพ็กเกจ ซึ่งข้อมูลส่วนหัวนี้จะมีที่อยู่ของคอมพิวเตอร์ที่ต้องการส่งข้อมูลนี้ไปถึงยู่ หลังจากนั้นแพ็กเกจเหล่านี้ก็จะถูกส่งต่อไปยังชั้นที่ 2 หรือชั้นเชื่อมโยงข้อมูล ซึ่งชั้นนี้จะเพิ่มข้อมูลส่วนหัว และเปลี่ยนรูปแบบแพ็กเกจให้เป็นเฟรม(Frame) ซึ่งในแต่ละเฟรมก็จะมีที่อยู่ของคอมพิวเตอร์ในชั้นนี้ แล้วเฟรมก็จะถูกส่งต่อไปยังชั้นกายภาพ ซึ่งจะทำการแปลงเฟรมให้เป็นบิตต่อเนื่องเพื่อทำการส่งต่อไปบนสื่อนำสัญญาณไปยังปลายทางต่อไป ส่วนการทำงานของฝั่งสถานีรับก็จะตรงข้ามกับสถานีส่ง กล่าวคือ แต่ละชั้นจะเอาส่วนที่เป็นข้อมูลส่วนหัวที่ถูกเพิ่มโดยชั้นที่อยู่ระดับเดียวกันออก เมื่อข้อมูลมาถึงชั้นที่ 4 ข้อมูลก็จะอยู่ในรูปแบบเดิมเหมือนกับตอนที่อยู่ในชั้นที่ 4 ของฝั่งสถานีส่ง

ในปัจจุบันระบบเครือข่ายมีโปรโตคอลที่ใช้หลายประเภทซึ่งพัฒนาโดยหลายองค์กรหรือบางบริษัท โครงสร้างของโปรโตคอลเหล่านี้จะแบ่งเป็นชั้นๆ หรือเลเยอร์คล้ายกับแบบอ้างอิง OSI แต่อาจจะไม่เหมือนกันทุกเลเยอร์ บางชุดโปรโตคอลอาจแบ่งขั้นตอนการรับส่งข้อมูลแค่ 4-5 ชั้นเท่านั้น แทนที่จะเป็น 7 ชั้น เหมือนแบบอ้างอิง OSI ซึ่งการทำงานของแต่ละชั้นอาจจะไม่เหมือนของแบบ OSI ทุกอย่าง อย่างไรก็ตาม แบบอ้างอิง OSI ก็ถือได้ว่าเป็นชุดโปรโตคอลที่เป็นต้นแบบของการศึกษาโปรโตคอลชุดอื่นได้ดี

ชุดโปรโตคอล OSI ประกอบด้วยโปรโตคอลมาตรฐานหลายโปรโตคอล โปรโตคอลเหล่านี้เป็นส่วนหนึ่งของโครงการนานาชาติเพื่อพัฒนาโปรโตคอลและมาตรฐานอื่นๆ เพื่อเอื้ออำนวยให้อุปกรณ์เครือข่ายที่ผลิตจากบริษัทต่างๆ สามารถทำงานร่วมกันได้ ข้อกำหนดของมาตรฐาน OSI ถูกจัดทำโดย 2 องค์กร คือ ISO และITU-T ซึ่งสามารถสรุปการทำงานของแต่ละเลเยอร์ดังภาพที่ 4

OSI Reference Model		OSI Protocols
7	Application	CMIP, DS, FTAM, MHS, VTP
6	Presentation	Presentation Service/Presentation Protocol
5	Session	Session Service/Session Protocol
4	Transport	TP0, TP1, TP2, TP3, TP4, TP5
3	Network	CONP/CMNS, CLNP/CLNS, IS-IS, ES-IS
2	Data Link	IEEE 802.2, IEEE 802.3, IEEE 802.5, FDDI, X.25
1	Physical	IEEE 802.2, IEEE 802.3, IEEE 802.5, FDDI, X.25 Hardware Hardware Hardware Hardware

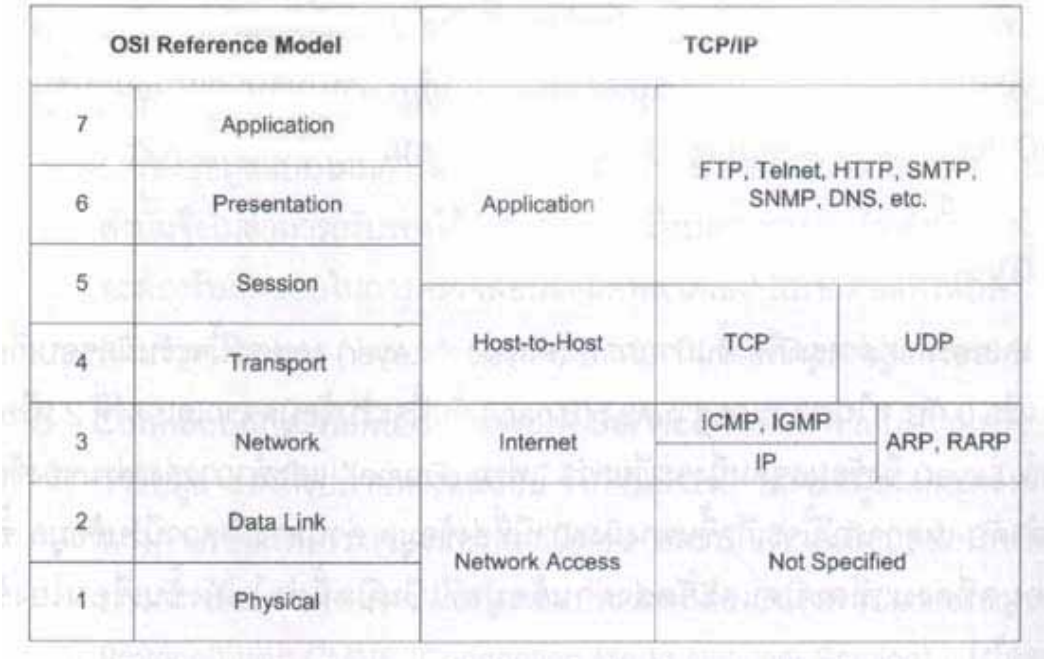
ภาพที่ 4 แสดงชุดโปรโตคอล OSI ที่ทำงานของแต่ละเลเยอร์

ที่มา : จตุชัย แพงจันทร์ และ อนุโชต วุฒิพรพงษ์, เจาะระบบ Network (กรุงเทพมหานคร : โรงพิมพ์ด่านสุทธาการพิมพ์, 2547), 33.

1.4 ชุดโปรโตคอล TCP/IP

ชุดโปรโตคอล TCP/IP (Transmission Control Protocol) ได้ถูกพัฒนามากว่า 20 ปี ซึ่งเริ่มจากการวิจัยที่สนับสนุนโดยกระทรวงกลาโหมสหรัฐฯ จุดประสงค์ของการวิจัยนี้ก็เพื่อเชื่อมคอมพิวเตอร์ที่ต่างแพลตฟอร์มกัน ให้สามารถสื่อสารกันผ่ายเครือข่ายได้ ซึ่งสามารถทำได้โดยการแบ่งโปรโตคอลเป็นชั้นและแยกการทำงานของแอพพลิเคชันของผู้ใช้จากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลผ่านเครือข่าย ชุดโปรโตคอลนี้จะมีการจัดรูปแบบที่แตกต่างจากแบบอ้างอิง OSI เล็กน้อย

การออกแบบชุดโปรโตคอล TCP/IP จะมุ่งเน้นไปที่การเชื่อมต่อระหว่างระบบที่ต่างกัน ในขณะที่แบบอ้างอิง OSI จะเน้นไปที่การแบ่งการทำงานของโปรโตคอลเป็นชั้นๆ การออกแบบ TCP/IP ยังคงเป็นแบบชั้นๆ เหมือนกัน แต่เมื่อถึงตอนทำจริงๆก็จะให้ขึ้นอยู่กับการตัดสินใจของผู้ออกแบบ ซึ่งเป็นผลให้ชุดโปรโตคอล OSI เหมาะสำหรับใช้อธิบายการสื่อสารระหว่างคอมพิวเตอร์ในเครือข่ายได้ดีกว่า ในขณะที่ชุดโปรโตคอล TCP/IP เป็นที่นิยมมากกว่าในการนำไปใช้จริง



ภาพที่ 5 เปรียบเทียบแบบอ้างอิง OSI และ TCP/IP
ที่มา : จตุชัย แพงจันทร์ และ อนุโชต วุฒิพรพงษ์, เจาะระบบ Network (กรุงเทพมหานคร : โรงพิมพ์ด่านสุทธาการพิมพ์, 2547), 38.

จากภาพที่ 5 แสดงการเปรียบเทียบชั้นโปรโตคอลระหว่างแบบอ้างอิง OSI และ TCP/IP แบบอ้างอิง TCP/IP จะแบ่งโปรโตคอลเป็น 4 ชั้น คือ ชั้นประยุกต์ใช้งาน (Application Layer) ชั้นเชื่อมต่อระหว่างโฮส (Host-to-host-Layer), ชั้นอินเตอร์เน็ต (Internet Layer) และชั้นเข้าถึงเครือข่าย (Network Access Layer) การแบ่งชั้นการทำงานของโปรโตคอล TCP/IP เมื่อเปรียบเทียบกับแบบอ้างอิง OSI แล้วอาจจะไม่คล้ายกันมากนัก

1.5 ไอพีแอดเดรสรุ่นที่ 6

Internet Protocol Version 6 หรือ IPv6 คือ อินเทอร์เน็ตโปรโตคอลรุ่นใหม่ที่ออกแบบเพื่อที่จะนำมาใช้แทนอินเทอร์เน็ตโปรโตคอลรุ่นปัจจุบันไอพีแอดเดรสรุ่นที่ 4 IP Version 4 (IPv4)โดยที่มีไอพีแอดเดรสเพิ่มจาก 32 บิตเป็น 128 บิตหรือประมาณ 3.4^{38} อินเทอร์เน็ตโปรโตคอลเวอร์ชันที่ใช้อยู่ในปัจจุบันเป็นเวอร์ชันสี่ (IPv4) ซึ่งกำหนดให้มีขนาดของแอดเดรสเพียง 32 บิต เมื่อเวลาผ่านไปความต้องการที่จะใช้งานหมายเลขไอพีแอดเดรสก็มีมากขึ้นเรื่อยๆ ทำให้คาดว่าอีกไม่นานในอนาคตหมายเลขไอพีแอดเดรสคงมีไม่เพียงพอที่จะแจกจ่ายได้อีกต่อไป และถ้ารอให้ถึงวันนั้นเครือข่ายอินเทอร์เน็ตคงถึงทางตันเพราะว่าไม่สามารถขยายขอบเขตการทำงานได้

อีก จึงมีหน่วยงานที่เกี่ยวข้องกับอินเทอร์เน็ตโปรโตคอลคิดที่จะสร้างอินเทอร์เน็ตโปรโตคอลเวอร์ชันใหม่ขึ้นและเพิ่มขนาดของเลขที่อยู่ให้มากกว่าเดิมเพื่อแก้ปัญหานี้เป็น 128 บิต ซึ่งสามารถที่จะแจกจ่ายให้กับคนทุกคนบนโลกนี้คนละหลายๆ หมายเลขได้ โดยมีการพิจารณาว่าแอดเดรสขนาด 128 บิตนี้ มีความเหมาะสมมากที่สุด เพราะว่าถ้าขยายขนาดแอดเดรสให้มีขนาดมากกว่านี้แล้วจะทำให้ขนาดของเฮดเดอร์ขยายตามไปด้วยโดยปริยาย ซึ่งอาจทำให้เกินความจำเป็นและอาจทำให้ขนาดของส่วนเฮดเดอร์มีขนาดใหญ่กว่าในส่วนของข้อมูลจริง โดยไอพีแอดเดรสรุ่นที่ 6 เฮดเดอร์ (Header) ของข้อมูลแบบไอพีแอดเดรสรุ่นที่ 6 ถูกออกแบบมาให้มีขนาดคงที่และมีภาพแบบที่ง่ายต่อการใช้งานเฮดเดอร์จะประกอบด้วยตำแหน่งต่างๆ (Field) ที่จำเป็นต้องใช้ในการประมวลผลแพ็คเกจ (Packet) ที่เราท์เตอร์ (Router) ส่วนตำแหน่งที่อาจจะถูกประมวลผลเฉพาะที่ต้นทางหรือปลายทางที่เราท์เตอร์จะถูกแยกออกมาไว้ที่ส่วนขยายของเฮดเดอร์ (Extended Header) ไอพีแอดเดรสรุ่นที่ 6 เพิ่มฟีเจอร์ใหม่ๆ ทำการปรับปรุงเฮดเดอร์เพื่อให้ประมวลผลได้มีประสิทธิภาพมากขึ้นในส่วนของเฮดเดอร์นั้นได้มีการเพิ่มประสิทธิภาพในการประมวลผลให้ดียิ่งขึ้น

โดยทำการตัดฟิลด์บางฟิลด์ที่ไม่จำเป็นออกไปปรับปรุงฟิลด์ที่มีอยู่แล้วให้ทำงานได้ดียิ่งขึ้นและเพิ่มฟิลด์ใหม่เข้ามาอีกสองฟิลด์เพื่อสนับสนุนการทำงานในลักษณะต่างๆ โดยที่ขนาดของเฮดเดอร์จะคงที่ตลอด คือ 40 ไบต์ ส่วนขนาดของเฮดเดอร์ของอินเทอร์เน็ตโปรโตคอลรุ่นปัจจุบันนั้นมีขนาดไม่คงที่แน่นอน ภาพแบบ (Header) ใหม่ (IPv6 Header) ได้ถูกออกแบบให้มีขนาด(Header) ลดน้อยลง โดยทำการย้ายฟิลด์ที่ไม่จำเป็น หรือที่เพิ่มออก โดยวางไว้หลัง IPv6 Header และมีการใช้เป็น Streamline Header ซึ่งมีประสิทธิภาพในการดำเนินการติดต่อกับอุปกรณ์ (Router) ได้ IPv4 Header กับ IPv6 Header ไม่สามารถใช้ง่ายร่วมกันได้ ซึ่งในการวางระบบทั้ง IPv4 และ IPv6 ต้องทำทั้งคู่เพื่อให้รู้จักภาพแบบของ Header ซึ่ง Header ของไอพีเลขที่อยู่รุ่นที่ 6 ใหญ่กว่าของไอพีแอดเดรสเวอร์ชันสี่สองเท่าและตำแหน่งที่อยู่ใหญ่กว่าถึง 4 เท่า

ขนาดแอดเดรส (IP Address) มีจำนวนเพิ่มมากขึ้นไอพีแอดเดรสรุ่นที่ 6 (IPv6) จะมีการกำหนดตำแหน่งที่อยู่ผู้ติดต่อ และผู้รับการติดต่อเป็น 128 บิต ซึ่งมีจำนวนที่อยู่ถึง 3.4×10^{38} ทำให้มีการออกแบบเป็นหลายลำดับชั้น และจองที่อยู่สำหรับ (Internet Backbone) เพื่อแยกจากการใช้งานเครือข่ายในองค์กรซึ่งมีเพียงไม่กี่เปอร์เซ็นต์ที่ใช้สำหรับตำแหน่งโฮสต์ และมีที่อยู่จำนวนมากที่ใช้ในอนาคต ทำให้อาจจะไม่จำเป็นที่ต้องมีการใช้การแบบเปลี่ยนไอพีแอดเดรสเครือข่ายย่อย (NATs) ในเครือข่ายอนาคตก็ได้

การกำหนดที่อยู่เป็นลำดับชั้นและกำหนดโครงสร้างสำหรับการหาเส้นทางของอุปกรณ์สำหรับ (IPv6 Global Addresses)ใช้บนไอพีเลขที่อยู่รุ่นที่ 6 สามารถที่สร้างและกำหนดลำดับชั้นได้อย่างมีประสิทธิภาพสำหรับการหาเส้นทาง และสิ่งที่เกิดขึ้นหลายลำดับชั้นสำหรับผู้ให้บริการ

อินเทอร์เน็ตบน (IPv6 Internet และ Backbone Routers) ทำให้ขนาดข้อมูลใน (Routing Table) เล็กลง เพิ่มขีดความสามารถในการเลือกเส้นทาง และสนับสนุนโมบายโฮสต์ IPv6 ได้ทำการจัดภาพแบบของเครือข่ายให้เป็นแบบลำดับชั้นได้แก่ Link-Local ซึ่งจะสามารถติดต่อได้กับโฮสต์เฉพาะที่ถูกเชื่อมต่อกันโดยตรงเท่านั้น (Sit Local) ขอบเขตของการติดต่อสื่อสาร คือ อยู่ในเครือข่ายเดียวกันเท่านั้นซึ่งจะมีลักษณะคล้ายกับไพรเวทไอพีแอดเดรสของอินเทอร์เน็ตโพรโตคอลรุ่นปัจจุบัน และลำดับชั้นสุดท้ายก็ คือ (Global Address) โดยแอดเดรสประเภทนี้มีลักษณะคล้ายกับไอพีจริงของ IPv4 เพราะสามารถถูกแสดงไปยังเครือข่ายต่างๆ ได้ ซึ่งการจัดภาพแบบของหมายเลขไอพีแอดเดรสในภาพแบบเป็นลำดับชั้นนี้ ทำให้ประสิทธิภาพความรวดเร็วในการหาเส้นทางเพิ่มมากขึ้นและช่วยลดปริมาณข้อมูลของตารางการหาเส้นทางลงได้อีกด้วย นอกจากนี้ได้เพิ่มประสิทธิภาพของทำให้การทำงานแบบโมบายโฮสต์มีประสิทธิภาพยิ่งขึ้น

ไม่จำเป็นที่ต้องแจ้งที่อยู่ก่อนหรือกำหนดที่อยู่ไว้ก่อนได้ เป็นการกำหนดค่าโฮสต์ ซึ่ง IPv6 รองรับทั้งกำหนดค่าที่แจ้งไว้ก่อน (State full) เช่น การใช้ DHCP Server และการกำหนดค่าที่อยู่โดยไม่แจ้งไว้ก่อนได้ (Stateless) ในกรณีที่ไม่มี DHCP Server เครื่องโฮสต์บนลิงค์นี้จะกำหนดค่าอัตโนมัติในตัวเองด้วย IPv6 addresses สำหรับลิงค์ (Link Local Addresses) และการกำหนดค่าที่อยู่โดยนำมาจากค่าประกาศด้านหน้าของ (Router) แม้ว่าไม่มี (Router) โฮสต์ก็สามารถที่ลิงค์ได้โดยกำหนดค่าที่อยู่ในลิงค์ท้องถิ่นเอง ด้วย (Link Local Addresses) และการสื่อสารโดยไม่ต้องกำหนดค่าที่อยู่ด้วย Quality Of Service (QoS) โดยมีฟิลด์ใหม่ใน IPv6 Header ที่กำหนดสำหรับรองรับการระบุ ซึ่งระบุการจราจร โดยใช้ฟิลด์ (Flow Label) ใน IPv6 Header อนุญาตให้ Router ทำการระบุและดูแลแพ็คเกจที่ไหล การไหลที่เป็นชุดของแพ็คเกจระหว่างต้นทาง ไปยังปลายทาง โดยรองรับ QoS ทำให้ง่ายต่อการติดต่อให้บรรลุเป้าหมายเมื่อมี (Packet Payload) ถูกเข้ารหัสด้วย (IPSec) ฝังความปลอดภัยไว้ภายใน ทำให้การสื่อสารระหว่างเครื่องมีความปลอดภัย

เพิ่มระบบรักษาความปลอดภัยนอกจากเพิ่มประสิทธิภาพในการประมวลผลเกี่ยวกับแพ็คเกจแล้ว อินเทอร์เน็ตโปรโตคอลเวอร์ชันใหม่นี้ยังเพิ่มประสิทธิภาพด้านความปลอดภัยของข้อมูลที่จะส่งออกไปผ่านเครือข่ายต่างๆ ในส่วนของการรักษาความปลอดภัยนั้น IPv6 ได้มีการใช้ระบบรักษาความปลอดภัยสองชนิด คือ การพิสูจน์ตัวจริง (Authentication) และ การเข้ารหัสข้อมูล (Encryption) ซึ่งในปัจจุบันจะมีการใช้ไฟล์วอลล์เป็นมาตรฐาน ส่วนการรักษาความปลอดภัยจะเป็นออพชั่นพิเศษแล้วแต่ว่าใครจะติดตั้งหรือไม่ติดตั้ง

สนับสนุนการทำงานแบบเวลาจริง (Real-Time Services) IPv6 ได้มีการสนับสนุนการทำงานแบบ (Real Time) โดยการปรับปรุงโดยการเพิ่มฟิลด์ (Flow Label) เพื่อให้การส่งข้อมูลนั้นมีประสิทธิภาพมากยิ่งขึ้น มีระบบติดตั้งแอดเดรสแบบอัตโนมัติ ถึงแม้ว่าปัจจุบันจะมีการใช้ DHCP

ช่วยในการติดตั้งหมายเลขไอพีแอดเดรสก็ตามแต่ว่าเรายังไม่ถือเป็นการติดตั้งแบบอัตโนมัติ IPv6 มีการติดตั้งหมายเลขไอพีแอดเดรสแบบอัตโนมัติ คือเป็นลักษณะแบบติดตั้งเพิ่มเติม (Plug And Play) โดยที่ผู้ใช้ไม่จำเป็นต้องกำหนดค่าพารามิเตอร์ใดๆ ทั้งสิ้นเพราะระบบจะเป็นผู้จัดการให้ทั้งหมด ซึ่งการติดตั้งแบบอัตโนมัตินี้แบ่งเป็นสามชนิดคือ แบบเฉพาะที่แบบไร้สถานะ และแบบเต็มสถานะ การติดต่อกับเครื่องข้างเคียง (Neighbor Discovery Protocol) สำหรับไอพีวีหก (IPv6) เป็นชุด (Internet Control Message Protocol : ICMPv6) ซึ่งจัดการโหนดเพื่อนบ้านที่อยู่ในลิงค์เดียวกันด้วยการจัดการนี้ทำให้มาแทนที่ Address Resolution Protocol (ARP), ICMPv6, Router Discovery และ ICMPv4 Redirect ที่ส่งด้วย Multicast และ Unicast โดยมีการรองรับเป็นหน้าที่เพิ่มขึ้นมา

1.5.1 ประเภทของหมายเลขไอพีแอดเดรสเวอร์ชันสี่ (IPV4) ที่ใช้กันอยู่ในปัจจุบันนั้นมีอยู่ 2 ชนิด คือยูนิคาสต์แอดเดรส ซึ่งแอดเดรสที่เราใช้กันอยู่ทั่วไป และอีกชนิด คือมัลติคาสต์แอดเดรสซึ่งจะเป็นการส่งข้อมูลให้กับกลุ่มของผู้รับ ส่วนแอดเดรสของ IPv6 แบ่งออกเป็น 3 ประเภทดังนี้

1.5.1.1 ยูนิคาสต์แอดเดรส (Unicast) เป็นแอดเดรสที่ถูกกำหนดอินเตอร์เฟสเครื่องต่อเครื่องเท่านั้น

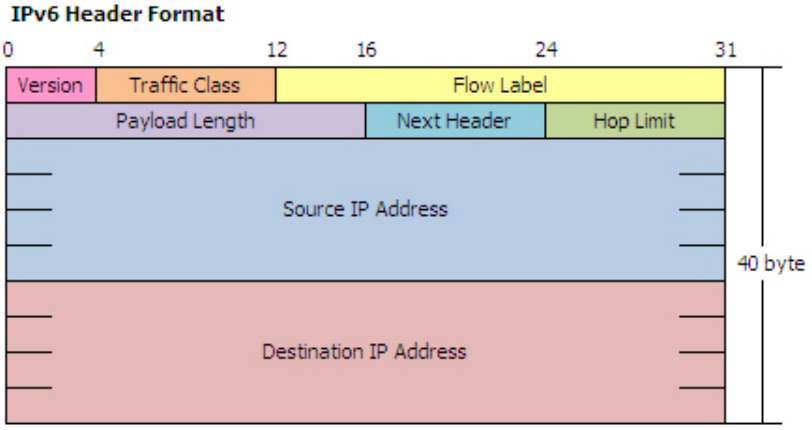
1.5.1.2 มัลติคาสต์แอดเดรส (Multicast) เป็นการกำหนดกลุ่มของอินเตอร์เฟสจะทำการส่งแพ็คเกจไปหาเครื่องทุกเครื่องในกลุ่มอินเตอร์เฟสปลายทาง

1.5.1.3 เอนิคาสต์แอดเดรส (Anycast) เป็นแอดเดรสที่มีการทำงานระหว่างยูนิคาสต์แอดเดรส และ มัลติคาสต์แอดเดรส โดยเป็นการกำหนดกลุ่มของอินเตอร์เฟสที่จะส่ง แต่ว่าการส่งข้อมูลจะเสร็จสิ้นเมื่อมีการส่งข้อมูลเสร็จสิ้นเพียงอินเตอร์เฟสเดียวจากกลุ่มของอินเตอร์เฟสจึงมักใช้กับเราท์เตอร์เพื่อส่งข้อมูลไปให้เราท์เตอร์ที่อยู่ใกล้ที่สุด

ภาพแบบการเขียนแอดเดรส หมายเลขไอพีแอดเดรสของ IPv4 นั้นจะเขียนโดยใช้เลขฐานสิบจำนวนสี่หลักโดยใช้จุดคั่นระหว่างแต่ละหลัก แต่ใน IPv6 นั้นจะเขียนหมายเลขไอพีแอดเดรสแทนด้วยเลขฐานสิบหกจำนวนแปดหลักและใช้เครื่องหมาย “:” คั่นระหว่างชุดตัวเลขแต่ละหลักดังตัวอย่างต่อไปนี้ "EFDC :BA98 :7654 :3210 :EFDC :BA98 :7654 :3210" ซึ่งการใช้เลขฐานสิบหกแทนที่จะใช้เลขฐานสิบทำให้เขียนหมายเลขไอพีแอดเดรสได้กะทัดรัดกว่าเดิม แต่ว่าจดจำได้ยุ่งยากผู้ใช้ IPv6 จึงต้องพึ่งพา Service DNS

1.5.2 ไอพีเฮดเดอร์ของ IPv4 จะมีขนาดไม่คงที่ซึ่งทำให้เปลี่ยนแปลงตามขนาดของ Option ส่วนเฮดเดอร์ของ IPv6 ได้มีการปรับปรุงเฮดเดอร์จาก IPv6 โดยการตัดฟิลด์ที่ไม่จำเป็นทิ้งไป 6 ฟิลด์ และทำการปรับปรุงฟิลด์ที่มีอยู่ให้มีประสิทธิภาพมากขึ้น 5 ฟิลด์ การเพิ่มฟิลด์ใหม่ขึ้นอีก 2 ฟิลด์ ทำให้เฮดเดอร์ของ IPv6 จะมีขนาดคงที่ คือ 40 ไบต์ ซึ่งประกอบด้วยฟิลด์กำหนดการ

ทำงาน 8 ไบต์ และ แอดเดรสต้นทางและปลายทางอีก 32 ไบต์ ซึ่งแต่ละฟิลด์มีความหมายดังแสดงในภาพที่ 6



ภาพที่ 6 รูปแบบเฮดเดอร์ไอพีรุ่นที่ 6

1.5.2.1 Version (4-bit) ใช้สำหรับระบุหมายเลขเวอร์ชัน มัลติคาสต์แอดเดรส (Multicast) เป็นการกำหนดกลุ่มของอินเตอร์เฟสจะทำการส่งแพ็คเกจไปหาเครื่องทุกเครื่องในกลุ่มอินเตอร์เฟสปลายทาง

1.5.2.2 Traffic Class (8-bit) ระบุชนิดของข้อมูลในชุดข้อมูลใช้ระบุว่าแพ็คเกจนี้อยู่ในกลุ่มใดและมีระดับความสำคัญเท่าไหร่ เพื่อที่เราท์เตอร์จะจัดลำดับขั้นการส่งแพ็คเกจให้เหมาะสม

1.5.2.3 Flow Label (20-bit) เป็นการกำหนดลำดับของการไหล ใช้ระบุลักษณะการไหลเวียนการจราจรในเครือข่ายระหว่างต้นทางและปลายทาง เช่น แอพพลิเคชั่นแบบ Video Conference มีการจราจรในเครือข่าย (Traffic) หลายลักษณะ เช่น ภาพ เสียง ตัวอักษร ฯลฯ โดยในแอพพลิเคชั่นหนึ่งจะสามารถสร้าง Flow Label ได้หลายลักษณะและสามารถแยก Flow ของภาพและเสียงออกจากกันได้

1.5.2.4 Payload Length (16-bit) จะพิจารณาความยาวข้อมูลส่วนของชุดข้อมูลหลัง Header เพื่อทำการระบุขนาดของ Payload ในหน่วย Octet (Byte) ดังนั้นขนาดของ Payload สูงสุดจะเป็น 65,535 Octets

1.5.2.5 Next Header (8-bit) กำหนดชนิดของเฮดเดอร์ที่ต่อจากนี้ ซึ่งใช้เป็นตัวบอกว่า extended header ตัวถัดไปเป็นเฮดเดอร์ประเภทไหน เช่น IPSec เป็นต้น

1.5.2.6 Hop Limit (8-bit) กำหนดจำนวนการอนุญาต ถ้าเป็น 0 ก็จะนำชุดข้อมูลนี้ทิ้ง TTL ระบุเวลาที่แพ็คเกจหน่วยเป็นวินาที โดยระบุว่าแต่ละเราท์เตอร์ต้องลด TTL ลงอย่างน้อย 1 วินาทีเราท์เตอร์จึงลด TTL ครั้งละ1 หน่วยเสมอแม้ว่าจะใช้เวลาประมวลผลแพ็คเกจน้อยกว่านั้นทำให้ไม่ตรงกับความหมายของ TTL ดังนั้นจึงถูกเปลี่ยนเป็น Hop Limit เพื่อให้ตรงกับความหมายจริงๆ ซึ่งเหมาะสมและง่ายต่อการประมวลผล

1.5.2.7 Source Address (128-bit) เป็นการระบุที่อยู่ของโฮสต์ หรืออุปกรณ์ที่ส่งข้อมูล

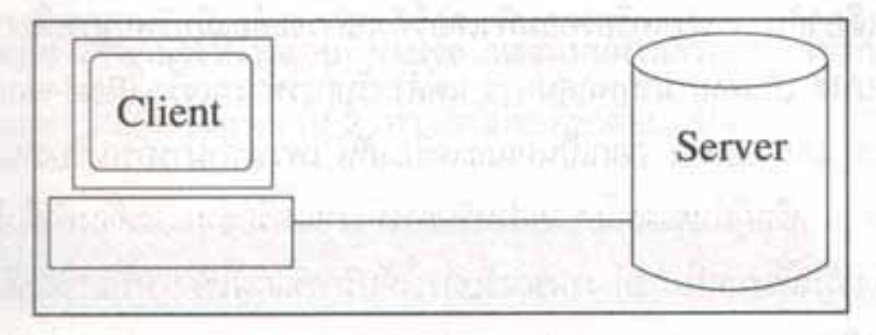
1.5.2.8 Destination Address (128-bit) เป็นการระบุที่อยู่ของโฮสต์ หรืออุปกรณ์ที่รับข้อมูล

2. ทฤษฎีเกี่ยวกับฐานข้อมูล

MySQL จัดเป็นระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกอินเตอร์เน็ต เพราะว่า MySQL เป็นฟรีแวร์ทางฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในตลาดปัจจุบัน ที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาฐานข้อมูลที่เคยใช้ MySQL ต่างยอมรับในความสามารถ ความรวดเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมหาศาล ทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย เช่น Unix, OS/2, Mac OS หรือ Windows นอกจากนี้ MySQL ยังสามารถใช้งานร่วมกับ Web Development Platform ทั้งหลาย เช่น C, C++, Java, Perl, PHP, Python, Tcl หรือ ASP เป็นต้น จากสาเหตุนี้จึงทำให้ MySQL เป็นที่นิยมอย่างมากในปัจจุบันและมีแนวโน้มสูงยิ่งๆขึ้นไปในอนาคต

2.1 สถาปัตยกรรมของ MySQL

สถาปัตยกรรม หรือโครงสร้างภายในของ MySQL คือ การออกแบบการทำงานในลักษณะของ Client/Server ซึ่งประกอบด้วยส่วนหลักๆ สองส่วน คือ ส่วนของผู้ให้บริการ (Server) และส่วนของผู้ใช้บริการ (Client) โดยในแต่ละส่วนจะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของตน ดังแสดงในภาพที่ 7



ภาพที่ 7 การทำงานแบบ Client/Server
ที่มา : สงกรานต์ ทองสว่าง, MySQL ระบบฐานข้อมูลสำหรับอินเตอร์เน็ต (กรุงเทพมหานคร : โรงพิมพ์ซีเอ็ดยูเคชั่น, 2544), 19.

ส่วนของผู้ให้บริการ หรือ (Server) จะเป็นส่วนที่ทำหน้าที่บริหารจัดการฐานข้อมูลในที่นี้หมายถึงตัว MySQL Server และเป็นที่เก็บข้อมูลทั้งหมด ข้อมูลที่เก็บไว้มีทั้งข้อมูลที่จำเป็นสำหรับการทำงานของระบบฐานข้อมูล และข้อมูลที่เกิดจากการที่ผู้ใช้แต่ละคนสร้างขึ้นมา

ส่วนของผู้ใช้บริการ หรือ (Client) คือ ผู้ใช้ โดยโปรแกรมสำหรับใช้งานในส่วนนี้ได้แก่ MySQL Client, Access, Web Development Platform ต่างๆ เช่น Java, Perl, PHP, ASP เป็นต้น

หลักการทำงานในลักษณะ Client/Server มีดังนี้

2.1.1 ที่ฝั่งของ Server จะมีโปรแกรมหรือระบบสำหรับจัดฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการใช้บริการจาก Client

2.1.2 เมื่อมีการร้องขอใช้บริการเข้ามา Server จะทำการตรวจสอบตามวิธีของตน เช่น อาจจำมีการให้ผู้ใช้บริการระบุชื่อและรหัสผ่าน และสำหรับ MySQL สามารถกำหนดได้ว่าจะอนุญาตหรือปฏิเสธ Client ใดๆ ในระบบที่จะเข้าใช้บริการ

2.1.3 ถ้าผ่านการตรวจสอบ Server จะอนุมัติการใช้บริการแก่ Client ที่ร้องขอการใช้บริการนั้นๆ ต่อไป และถ้าในกรณีที่ไม่ได้รับการอนุมัติ Server จะส่งข่าวสารความผิดพลาดแจ้งกลับไปที่ Client ที่ร้องขอใช้บริการนั้น

เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client หรือ Server อาจจะอยู่บนเครื่องเดียวกัน หรือแยกเครื่องกันก็ได้ ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบ ตามปกติถ้าเป็นการทำงานในลักษณะ Web-based มีการใช้ฐานข้อมูลขนาดไม่ใหญ่นัก ตัว MySQL Server และ Client มักจะอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการ

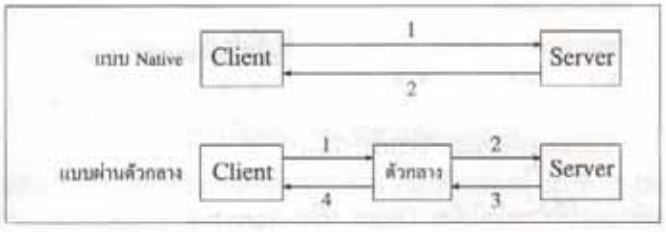
ทำงานมากพอสมควร แต่สำหรับการใช้งานจริง (Real-world Application) มักจะแยก Client และ Server ออกเป็นคนละเครื่อง เพราะสามารถรองรับงานได้ดีกว่า มากกว่า ดังนั้น ผู้บริหารระบบ หรือผู้กำหนดนโยบายสำหรับการทำงานเครือข่าย จะต้องคำนึงถึงเรื่องเกี่ยวข้องเหล่านี้ เพื่อที่จะทำให้ระบบมีการทำงานรองรับให้บริการแก่ผู้ใช้อย่างมีประสิทธิภาพ และข้อมูลมีความปลอดภัยมากที่สุด

2.2 วิธีการเชื่อมต่อจาก Client เข้าสู่ Server

จะกล่าวถึงลักษณะและวิธีการเชื่อมต่อ 2 แบบ คือ Native และ แบบผ่านตัวกลาง ดังนี้

2.2.1 แบบ Native เป็นแบบที่นิยมใช้กันมากในกรณีที่ระบบปฏิบัติการของ MySQL Server เป็น Unix เป็นลักษณะวิธีการเชื่อมต่อที่มีการทำงานเร็วที่สุด เพราะทำงานกันภายใน โดยลักษณะการทานประเภทนี้ได้แก่ การใช้งาน MySQL ร่วมกับ Web Development Platform ทั้งหลาย

2.2.2 แบบผ่านตัวกลาง คือ ODBC (Open Database Connectivity) ซึ่งส่วนใหญ่จะใช้กับ Server ที่ใช้ Windows Platform เป็นระบบปฏิบัติการ การทำงานประเภทนี้จะทำงานช้ากว่า แบบ Native เพราะการทำงานในแต่ละครั้งระหว่าง Client และ server ต้องผ่านตัวกลางก่อน แต่ ODBC มีข้อได้เปรียบในเรื่องฐานผู้ใช้ Windows Platform มากกว่า และด้วย ODBC ทำให้เราสามารถใช้ Client Development Tools ยอดนิยม เช่น Access หรือ ASP เพื่อเชื่อมต่อเข้าหา MySQL Server ได้



ภาพที่ 8 การทำงานแบบ Native และแบบผ่านตัวกลาง

ที่มา : สงกรานต์ ทองสว่าง, MySQL ระบบฐานข้อมูลสำหรับอินเตอร์เน็ต (กรุงเทพมหานคร : โรงพิมพ์ซีเอ็ดยูเคชั่น, 2544), 20.

จากภาพที่ 8 แสดงการเปรียบเทียบการทำงานระหว่างแบบ Native และแบบผ่านตัวกลาง จะเห็นได้ว่าแบบผ่านตัวกลางจะใช้ขั้นตอนมากกว่าแบบ Native ถึงสองเท่า แต่ก็มีข้อดีข้อเสียแตกต่างกันดังแสดงในตารางที่ 1

ตารางที่ 1 เปรียบเทียบการทำงานระหว่าง แบบ Native และแบบผ่านตัวกลาง

แบบ Native	**แบบผ่านตัวกลาง**
1. มีการทำงานรวดเร็วกว่า เพราะสื่อสารกันภายใน	1. มีการทำงานช้ากว่า เพราะมีตัวกลางเพิ่มขึ้นมาอีก 1 ขั้นตอน
2. Client ที่จะมาเชื่อมต่อต้องมีการฝังของส่วนโปรแกรมของ Server บางส่วนไว้สำหรับใช้งานหมายถึงเราต้องปรับปรุง Client เพิ่มเติม	2. ไม่ต้องปรับปรุงClient เพิ่มเติม เพียงแค่สนับสนุน ODBC ก็สามารถทำงานได้
3. ส่วนใหญ่มักไม่มีข้อจำกัดในการทำงาน	3. มีข้อจำกัดขึ้นกับตัวกลาง หรือ ODBC ที่ใช้
4. Server Platform ส่วนใหญ่มักเป็น Unix	4. ส่วนใหญ่เป็น Windows Platform
5. Client ส่วนใหญ่ใช้งานในลักษณะ Web –based เช่น Java, Perl, PHP เป็นต้น	5. รองรับทั้ง Web –based หรือการใช้ Client Development Tools อื่นๆ เช่น Access, VB, ASP

ที่มา : สงกรานต์ ทองสว่าง, MySQL ระบบฐานข้อมูลสำหรับอินเตอร์เน็ต (กรุงเทพมหานคร : โรงพิมพ์ซีเอ็ดยูเคชั่น, 2544), 21.

3. งานวิจัยที่เกี่ยวกับ Quality of Service (QoS)

3.1 IPv6 deployment: Real time applications and QoS aspects

(Bouras, Gkamas, Primpas and Stamos 2006 : 1397-1401)

เป็นงานวิจัยที่กล่าวถึงกลไกความสัมพันธ์ของ QoS ที่อยู่ภายใต้ IPv6 และการเปลี่ยนของโปรแกรมประยุกต์ไปสู่อินเตอร์เน็ตโปรโตคอลใหม่ งานวิจัยได้กล่าวถึงการทดสอบ QoS บนระบบเครือข่าย IPv6 ซึ่งอยู่บนพื้นฐานการทำงานในแบบ DiffServ และมีการเข้มงวดในเรื่องของความสำคัญของ packet ที่มีผลมาจาก real time application ได้มีการนำไปใช้และทดสอบกับระบบ 6NET ที่เป็นเครือข่าย IPv6 ขนาดใหญ่ และมีการเพิ่มการทดสอบในระดับที่สูงขึ้นอีกที่เป็นการการทำในการทดสอบแบบ local งานวิจัยได้มีการนำเสนอผลจากการวิเคราะห์ภายใต้ตัวเลขที่มีความแตกต่างกัน งานวิจัยได้มุ่งผลในเรื่องการเปลี่ยนโปรแกรมประยุกสู่ IPv6 และมีกรณีศึกษาการเปลี่ยนของโปรโตคอล OpenH323 ที่สนับสนุน IPv6

3.2 Estimation of Perceived Quality of Service for Applications on IPv6 Networks

(Xiaoming Zhou, Rob E. Kooij, Henk Uijterwaal and Piet Van Miegham 2006 : 74-81)

เป็นงานวิจัยที่กล่าวถึงการจัดหาบริการที่มีคุณภาพสูงที่เป็นโปรแกรมประยุกต์ในอนาคต การวัดประสิทธิภาพการทำงานของ IPv6 จึงเป็นสิ่งที่ขาดไม่ได้ อย่างไรก็ตามเพื่อให้เกิดความรู้ดีที่สุด การหาค่าความหน่วงของ IPv6 และการสูญเสียประสิทธิภาพที่เกิดขึ้นและผลกระทบบนโปรแกรมประยุกต์ เนื่องมาจากไม่ได้มีการศึกษากันอย่างกว้างขวาง ในงานวิจัยนี้ได้มีการวิเคราะห์ เส้นทาง IPv6 จากจุดเริ่มต้นถึงจุดสิ้นสุดมากกว่า 600 เส้นทาง โดยประมาณ 26 กลุ่มตัวอย่างของ RIPE NCC (Réseaux IP Européens Network Coordination Centre) มากกว่า 2 ปีที่ผ่านมา และมีการเปรียบเทียบค่าความหน่วงและค่าประสิทธิภาพการทำงานที่สูญเสียไปใน IPv6 กับข้อมูลเดียวกันบน IPv4 งานวิจัยได้มีการนำเสนอกระบวนการวิธีการหาค่า และแสดงหลักฐานผลของการวิจัยว่า IPv6 Network มีค่าความหน่วงและการสูญเสียมากกว่า IPv4 โดยงานวิจัยได้มีการได้มีการประเมินคุณภาพมาจากเรียลไทม์โปรแกรมประยุกต์ทั้งหมด 3 ตัว ได้แก่ VoIP, Video-over-IP และ การสื่อสารข้อมูลบนพื้นฐานของ TCP โดยงานวิจัยได้มีการเริ่มต้นงานวิจัยตั้งแต่เดือนตุลาคม 2003 จนกระทั่งถึงเดือนตุลาคม 2005 ซึ่งผลที่เกิดขึ้นเป็นที่มาของการนำเอา QoS มาใช้บน IPv6 Networks มันเป็นสิ่งที่สำคัญมากที่จะทำให้เห็นความแตกต่างของการสูญเสียข้อมูลระหว่าง IPv4 และ IPv6 หลังจากที่มีการนำมาใช้งานแล้วทำให้เห็นความแตกต่างของค่าความหน่วงและการสูญเสียข้อมูลระหว่าง IPv4 และ IPv6 โดยประสิทธิภาพจะดีขึ้นโดยตระหนักถึงปริมาณงานที่เกิดขึ้น

3.3 IPv6 QoS Testing on Dual Stack Network

(Christos Bouras, Dimitris Primpas and Kostas Stamos 2006 : Article No. 3)

งานวิจัยนี้ได้มีการนำเสนอในเรื่องของการทดสอบและการหาค่ากลไกการทำงานของ DiffServ QoS บน IPv6 และ IPv4 บนพื้นฐานการทำงานของซอฟต์แวร์ Dual stack โดยเป็นที่รู้กันว่าใน IPv6 ได้มีการเพิ่มคุณสมบัติบางประการเข้าไปและปัจจุบันได้มีการนำเอากลไกการทำงานของ QoS มาใช้งานกับ IPv6 ซึ่งมีลักษณะใกล้เคียงกับ QoS ที่สนับสนุนใน IPv4 เพราะอย่างนั้นจำนวนการทดสอบกับกลไกที่เกี่ยวข้องกับ DiffServ QoS บน IPv6 Traffic ได้มีการบรรจุรูปแบบการทดสอบที่พิเศษสำหรับจุดมุ่งหมายในครั้งนี้ ซึ่งอยู่ในกระบวนการและขั้นตอนที่มีเหตุผล และมีการหาค่าประสิทธิภาพในการทำงานรวมของอุปกรณ์ค้นหาเส้นทาง (Router) โดยมีการหาค่าให้ครอบคลุมถึงตัวอุปกรณ์จากกลไกการนำไปใช้ในระดับของการสนับสนุนสำหรับคุณสมบัติ QoS IPv6 และมีการเปรียบเทียบประสิทธิภาพบน IPv4 และ IPv6

3.4 Quality of Service and Mobility for the Wireless Internet

(J. Antonio García-Macías, Franck Rousseau, Gilles Berger-Sabbatel, Leyla Toumi and Andrzej Duda 2003 : 341 - 352)

งานวิจัยที่ได้มีการสำรวจออกมาของการทำอย่างไรที่จะหาความเหมาะสมกลไกคุณภาพของการบริการที่รวบรวมจากใกล้ๆ กับความยืดหยุ่นของการจัดการโมบิลิตี้ในเครือข่ายไร้สาย งานวิจัยได้พิจารณาทางเลือกต่างๆ ของเครือข่ายเพื่อให้เกิดประสิทธิภาพสูงสุด การนำเสนอของงานวิจัยจะเป็นแบบลำดับขั้นของสถาปัตยกรรม QoS ที่ยึดตามหลักของ DiffServ ถึงโฮสต์ที่เคลื่อนที่ในสภาวะแวดล้อมที่ไร้สาย โดยตัวงานวิจัยเองได้มีการควบคุมค่าหลายๆ อย่างของเครือข่ายไร้สาย เรียได้ว่าการจำกัดระยะทางในทางภูมิศาสตร์ เพื่อทำให้อัตราการส่งข้อมูลสูงไปยังโฮสต์ทุกเครื่อง มีการบังคับอัตราจุดเริ่มต้นของการรับส่งข้อมูล เพื่อจำกัดการใช้งาน ของช่องทางในฟังก์ชั่นที่ต้องการ QoS และจำกัดจำนวนโฮสต์ที่มีการใช้งานให้เพียงพอ การจัดการ QoS นั้นคู่กับการจัดการโมบิลิตี้ในระดับ IP งานวิจัยได้มีแผนงานการนำเอาไมโครโมบิลิตี้มาใช้ใน IPv6 กับไม่มีการเคลื่อนที่ระหว่างเซลล์ใกล้ๆ ไมโครโมบิลิตี้ได้มีการหลบหลีกการเปลี่ยนที่อยู่ traffic tunneling และไม่มีการเคลื่อนที่ งานวิจัยได้ให้รายละเอียดของการทดลองเพื่อแสดงถึงคุณภาพของการบริการที่แตกต่างกันบนเครือข่าย 802.11b

4. งานวิจัยที่เกี่ยวกับ Mobility

4.1 Mobility Support in IPv6

(Perkins and Johnson 1996 : 27-37)

งานวิจัยจะกล่าวถึงหมายเลขไอพีเวอร์ชั่น 6 (IPv6) ที่ได้รับการออกแบบโดย IETF (Internet Engineering Task Force) โดยมีวัตถุประสงค์เพื่อที่จะนำมาใช้งานแทนที่หมายเลขไอพีเวอร์ชั่นเก่า (IPv4) ได้มีการออกแบบเพื่อเพิ่มประสิทธิภาพของโปรโตคอลเข้าไปใน IPv6 เป็นที่รู้กันก็คือ Mobile IPv6 ซึ่งมีการยอมให้มีการจัดเส้นทางของแพ็คเก็จ IPv6 ส่งไปให้โหนดเคลื่อนที่เป็นข้อดีและมีโอกาสเป็นไปได้ที่จะนำไปออกแบบหมายเลขไอพีเวอร์ชั่นใหม่ ใน Mobile IPv6 โหนดเคลื่อนที่จะถูกกำหนดโดย home address ตลอดเวลา โดยไม่ต้องคำถึงจุดที่มีการเชื่อต่ออินเตอร์เน็ตอยู่ งานวิจัยนี้จะกล่าวกี่ยวกับ Mobility ใน IPv6 และสิ่งที่เกี่ยวข้องเป็นหลัก

4.2 Mobile IPv6 network: implementation and application

(Jiann-Liang Chen, Yu-Feng Lee and Yao-Chung Chang 2006 : 29-43)

งานวิจัยได้กล่าวถึงในอดีตที่ผ่านมาไม่นานโทรศัพท์เคลื่อที่ได้รับความนิยมเพิ่มมากขึ้นและผู้คนทั่วไปเริ่มมีการสื่อสารกันด้วยเครือข่ายไร้สาย Mobile IPv6 ทำให้สามารถสื่อสารกันได้

ถึงแม้จะมีการเคลื่อนที่ไปด้วยก็ตาม ในที่นี้ เครือข่าย Mobile IPv6 ถูกนำไปใช้ด้วยคุณสมบัติของ IEEE 802.11 และได้มีการวิเคราะห์ประสิทธิภาพเมื่อนำไปใช้งานกับโปรแกรมประยุกต์ TCP และ UDP โดยมีการแสดงผลการวิเคราะห์ เปรียบเทียบให้เห็นความแตกต่างระหว่างการนำเอา Mobile IPv6 มาใช้กับโปรแกรมประยุกต์ TCP และ UDP ซึ่งมีการสรุปว่านำมาใช้กับโปรแกรมประยุกต์ TCP จะดีกว่านำไปใช้กับโปรแกรมประยุกต์ UDP เล็กน้อย

5. งานวิจัยที่เกี่ยวกับ Game Traffic

5.1 Game Traffic Analysis: An MMORPG Perspective

(Kuan-Ta Chen, Polly Huang, Chun-Ying Huang and Chin-Laung Lei 2005 : 19-24)

งานวิจัยได้กล่าวถึงเกมออนไลน์ว่าเป็นหนึ่งในธุรกิจที่ทำกำไรให้ผู้ประกอบการอย่างมากบนอินเตอร์เน็ต ในประเภทของเกมออนไลน์ประเภทที่ได้รับความนิยมเป็นพิเศษก็คือ เกมประเภท MMORPG (Massive Multiplayer Online Role Playing Games) ซึ่งมีความโดดเด่นมาในเอเชีย ทางเลือกที่ดีที่จะเข้าใจเกมทราฟิกเพิ่มมากขึ้น งานวิจัยได้มีการวิเคราะห์แพ็คเก็จ 1,356 ล้านแพ็คเก็จจากเกม ShenZhou Online ที่เป็นเกม MMORPG ที่มีขนาดใหญ่พอสมควร และสิ่งแรกที่ทางทีมวิจัยทำก็ในเรื่องของการวิเคราะห์เครื่องแม่ข่าย MMORPG

งานวิจัยได้มีการค้นพบว่าเกมประเภท MMORPG และ FPS (First-Person Shooting) ที่มีความเหมือนกันในเรื่องการส่งแพ็คเก็จที่มีขนาดเล็กและมีความต้องการช่องทางในการรับส่งข้อมูลต่ำ โดยเฉพาะช่องทางในการรับส่งข้อมูลยิ่งน้อยจะทำให้มีความเป็นเรียลไทม์ในเวลาเล่นเกม นอกจากนี้ยังมีลักษณะอื่นที่กล่าวไว้ในงานวิจัยเกี่ยวกับเกมออนไลน์ ในท้ายที่สุดได้มีการนำเสนอผลการวิจัยและมีการเปรียบเทียบปริมาณในการรับส่งข้อมูลระหว่างเครื่องแม่ข่ายและเครื่องลูกข่ายไว้ในเอกสารด้วย

5.2 Network Game Traffic Modeling

(Färber 2002 : 53-57)

งานวิจัยได้กล่าวถึงเครื่องหมายแสดงว่าส่วนแบ่งข้อมูลการจราจรอินเตอร์เน็ตในทุกวันนี้เกิดขึ้นโดยเกมที่เล่นกันบนเครือข่าย ประเภทของข้อมูลการจราจรเป็นสิ่งที่น่าสนใจในการพิจารณาถึงความเป็นไปได้ของตลาดพร้อมกับความต้องการใช้งานแบบเรียลไทม์บนเครือข่าย สำหรับสิ่งที่ต้องคำนึงถึงของข้อมูลการจราจรเกมในมุมมองของเครือข่าย รูปแบบของข้อมูลการจราจรคือความต้องการที่ยอมรับให้เกิดขึ้นเป็นลักษณะเฉพาะที่ถูกนำเข้ามาวิเคราะห์ หรือสร้างแบบจำลอง หาค่าประสิทธิภาพของเครือข่าย ในงานวิจัยนี้ได้มีการหาค่าเกมแอคชั่นที่เล่นกันแบบหลายคน คือเกม Counter Striker จาก 36 ชั่วโมงในเครือข่าย LAN มีการวัดและการนำเสนอ

รูปแบบของข้อมูลการจราจรสำหรับเครื่องลูกข่ายและเครื่องแม่ข่าย งานวิจัยยังกล่าวอีกว่าในตอนท้ายได้มีการสังเกตบนการใช้งานรูปแบบข้อมูลการจราจรเกมบนแบบจำลองและบน QoS สำหรับการหาค่าความเหมาะสมของผลลัพธ์แบบจำลอง

6. งานวิจัยเกี่ยวกับ IPv6 ในประเทศไทย

6.1 Performance Analysis Of Mobile IPv6 For Linux Testbed System

(วโรดม วีระพันธ์ และ อภิเนตร อูนากูล 2546 : 1-13)

งานวิจัยได้กล่าวถึงสิ่งที่ท้าทายของการทำโมบิลิตี้ ในชั้นของ IP คือ การอนุญาตให้โฮสต์นั้นสามารถใช้แอดเดรสเดิมอยู่ได้ โดยไม่มีการกระจาย ตารางเส้นทางการส่งข้อมูลแบบเฉพาะเจาะจง ไปยังอุปกรณ์ชี้เส้นทางหลาย ๆ ตัว หรือมีการตัดการเชื่อมต่อสื่อสารในขณะนั้น Mobile IPv6 ออกแบบมาให้สามารถทำการเคลื่อนและเปลี่ยนที่ในชั้น IP ได้ คือตราบเท่าที่การเชื่อมต่อของ TCP ยังไม่ได้ตัดขาดการสื่อสาร แม้ว่าจะมีการเคลื่อนย้าย IP แต่การเชื่อมต่อสื่อสารนั้นก็ยังคงอยู่และสามารถใช้งานได้ต่อเนื่อง ในงานวิจัยนี้ ได้นำเสนอเกี่ยวกับ การทดลอง วิเคราะห์และปรับปรุงประสิทธิภาพซึ่งใช้ Mobile IPv6 ที่นำไปใช้งานบนลีนุกส์

6.2 IPv6 Testbed

(Centre for Network Research 2550)

เป็นโครงการงานวิจัยที่กำลังวิจัยกันอยู่ที่ทาง CNR (Centre for Network Research) ได้มีการจัดทำขึ้นมาโดยมีวัตถุประสงค์ เนื่องจากการพัฒนาอย่างรวดเร็ซทางด้านไอที ทำให้จำนวนหมายเลข IP กำลังจะหมดไปในไม่ช้า มีการคาดว่าในอนาคต อุปกรณ์เกือบทุกชนิด ไม่ว่าจะเป็นอุปกรณ์สื่อสาร ตู้เย็น โทรทัศน์ หรือแม้แต่ยานพาหนะต่างๆ จะมีหมายเลข IP เพื่อเชื่อมต่อเข้าสู่ระบบเครือข่ายได้ การทำงานของโทรศัพท์ นอกจากนี้การใช้งานด้านต่างๆ จะใช้ความสามารถของ IPv6 มากขึ้น เช่น การความสามารถในการเคลื่อนและเปลี่ยนที่ (Mobility) ความปลอดภัย (Security) เป็นต้น

ห้องปฏิบัติการ CNRนับได้ว่าเป็นที่แรกที่เชื่อมต่อและให้บริการการเข้าสู่ระบบเครือข่ายหลักของ IPv6 (6Bone) และเป็นห้องวิจัยที่มีการวิจัยและพัฒนาระบบ IPv6 ที่เหมาะสมสำหรับประเทศไทย การเตรียมและรวบรวมและเผยแพร่องค์ความรู้ โครงการวิจัยที่ดำเนินการอยู่ในปัจจุบัน

6.3 VoIP:CNR-SIP

(Centre for Network Research 2550)

เป็นโครงการงานวิจัยที่กำลังวิจัยกันอยู่ที่ทาง CNR (Centre for Network Research) ได้มีการจัดทำขึ้นมาโดยงานวิจัยที่เกี่ยวข้องกับ IPv6 ก็คือ SIP over IPv6 with mobility ซึ่งจะเกี่ยวกับ VoIP ที่ทำการรับส่งข้อมูลเสียง ภาพ และวีดีโอ เพื่อการติดต่อสื่อสารบนระบบเครือข่ายอินเตอร์เน็ตที่เป็นที่แพร่หลายในปัจจุบัน ทั้งนี้ แรงผลักดันที่ทำให้เทคโนโลยี IP Telephony เป็นหนึ่งในงานวิจัยที่สำคัญ ก็คือ เทคโนโลยีเอื้ออำนวยต่อการสร้างบริการใหม่ในการสื่อสาร ไม่ว่าจะเป็น การโทรศัพท์ที่สามารถเห็นหน้าของฝั่งตรงข้าม การโอนสาย กล่องไปรษณีย์เสียง (Voice mail) การเคลื่อนย้าย และอื่นๆ นอกจากนี้เหตุผลสำคัญอีกอย่างหนึ่งคือ การสื่อสารทางไกลจะทำได้ถูกลง เนื่องจากไม่มีการคิดค่าโทรทางไกล ผู้ใช้สามารถที่จะโทรศัพท์ภายในประเทศ หรือโทรข้ามโลกได้ในราคาที่ประหยัดมาก

6.4 Investigation of Duplicate Address Detection in Mobile IPv6

(พหล โสตถิวิรัช, พนิตา พงษ์ไพบูลย์, สุขุมาล กิติสิน และ ชวลิต ศรีสถาพรพัฒน์ 2007 : 1-10)

ในระบบเครือข่ายแบบเคลื่อนที่ที่ใช้ IPv6 (Mobile IPv6 Network) ที่มีการเคลื่อนที่ของอุปกรณ์ภายในเครือข่ายและการเคลื่อนที่ระหว่างเครือข่ายอยู่ตลอดเวลา เพื่อการติดต่อสื่อสารที่มีการเคลื่อนที่อยู่ตลอดเวลาให้สามารถทำงานได้อย่างราบรื่น ทำให้การติดตั้ง IPv6 address แบบอัตโนมัติเข้ามามีบทบาท เพราะช่วยให้ผู้ใช้งานไม่ต้องสนใจการติดตั้ง IPv6 address เมื่อมีการเคลื่อนย้ายอุปกรณ์ระหว่างเครือข่ายที่มีกลุ่มของ IPv6 address คนละชุดที่เปลี่ยนแปลงอยู่ตลอด โดยปกติแล้วขั้นตอนการตั้งค่า IPv6 address สามารถทำได้อัตโนมัติ โดย Router จะมีหน้าที่แจกจ่าย Network prefix ให้กับเครื่องลูกข่าย และเครื่องลูกข่ายแต่ละเครื่องจะเลือก host address ของตนเองนำมารวมเข้ากับ Network prefix ที่ได้รับ กลายเป็น IPv6 address ใหม่ เนื่องจากมีความเป็นไปได้ที่เครื่องลูกข่ายสองเครื่องอาจเลือกใช้ host address ชุดเดียวกัน มีผลให้ทั้งสองเครื่องมี IPv6 address ซ้ำกัน ดังนั้น IPv6 จึงกำหนดให้มีการตรวจสอบว่า IPv6 address ที่แต่ละเครื่องเลือกขึ้นมานั้นยังไม่มีใครใช้อยู่ ขั้นตอนนี้เรียกว่า Duplicate Address Detection (DAD) โดยการถามทุก node ว่ามีการใช้แล้วหรือไม่ และรอข้อความตอบกลับเป็นเวลา 1000 ms หากไม่มีใครตอบกลับมาแสดงว่ายังไม่มีใครใช้สามารถใช้ได้เลย บทความนี้จะเน้นการศึกษาขั้นตอนการทำงานที่ส่งผลต่อความล่าช้าในการตรวจสอบ IPv6 address ใหม่และเทคนิคที่สามารถนำมาใช้เพื่อลดความล่าช้าในส่วนนี้ลง

บทที่ 3
วิธีดำเนินงานวิจัย

งานวิจัยนี้ได้มีการพัฒนาเกมออนไลน์เพื่อวัดประสิทธิภาพการทำงานของเครือข่ายโปรโตคอล IPv4 และ IPv6 โดยทำการทดลอง วิเคราะห์ และปรับปรุงประสิทธิภาพโดยได้มีการกำหนดขั้นตอนและระยะการดำเนินการวิจัยดังตารางที่ 2

ตารางที่ 2 ขั้นตอนและระยะเวลาในการดำเนินการวิจัย

ที่	ขั้นตอนการดำเนินการวิจัย	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4	เดือนที่ 5	เดือนที่ 6
1	จัดเตรียมข้อมูลและเอกสารต่างๆ	↔					
2	ออกแบบโครงสร้างของระบบ ฐานข้อมูล	↔	↔				
3	ออกแบบเครือข่ายสำหรับใช้ในการทดสอบ		↔	↔			
4	พัฒนาระบบเกมออนไลน์			↔	↔	↔	
5	ทดสอบระบบเกมออนไลน์				↔	↔	
6	วิเคราะห์และประเมินผลการทดสอบระบบ				↔	↔	
7	สรุปผลการวิจัยและจัดทำรายงานวิทยานิพนธ์						↔

จากขั้นตอนการดำเนินงานวิจัยตามตารางที่ 2 สามารถอธิบายรายละเอียดได้ดังนี้

1. จัดเตรียมข้อมูลและเอกสารต่างๆ

การเตรียมเอกสารงานวิจัยและข้อมูลสำหรับจัดทำตัวระบบ โดยจะเป็นข้อมูลเกี่ยวกับการออกแบบฐานข้อมูล ข้อมูลเกี่ยวกับการสร้างเครือข่าย IPv6 สำหรับใช้ในการทดสอบเกมออนไลน์และข้อมูลการนำเอา IPv6 ไปประยุกต์ใช้งานในด้านต่างๆ

2. ออกแบบโครงสร้างของระบบ ฐานข้อมูล

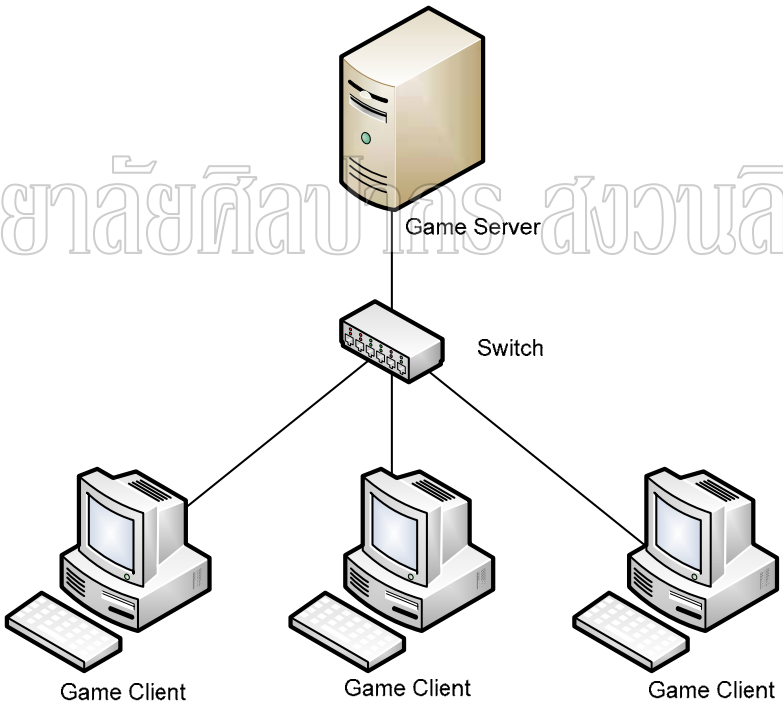
ทำการออกแบบฐานข้อมูลที่ใช้ในการพัฒนาระบบ ซึ่งการเออกแบบจะเก็บข้อมูลของผู้เล่นและเหตุการณ์ต่างที่เกิดขึ้นในเกม

3. ออกแบบเครือข่ายสำหรับใช้ในการทดสอบ

การออกแบบเครือข่ายสำหรับใช้ในการออกแบบระบบจะมีการออกแบบเครือข่ายทั้งหมด 3 แบบโดยดูจากสภาพการใช้งานเครือข่ายจริงในปัจจุบัน มีรายละเอียดดังนี้

3.1 เครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียว

เป็นเครือข่ายที่ออกแบบมาสำหรับการทดสอบเบื้องต้นเนื่องจากโปรโตคอล IPv4 มีการใช้งานกันอยู่แล้วซึ่งจะเทียบกับเครือข่ายขนาดที่ไม่ใหญ่มากนัก

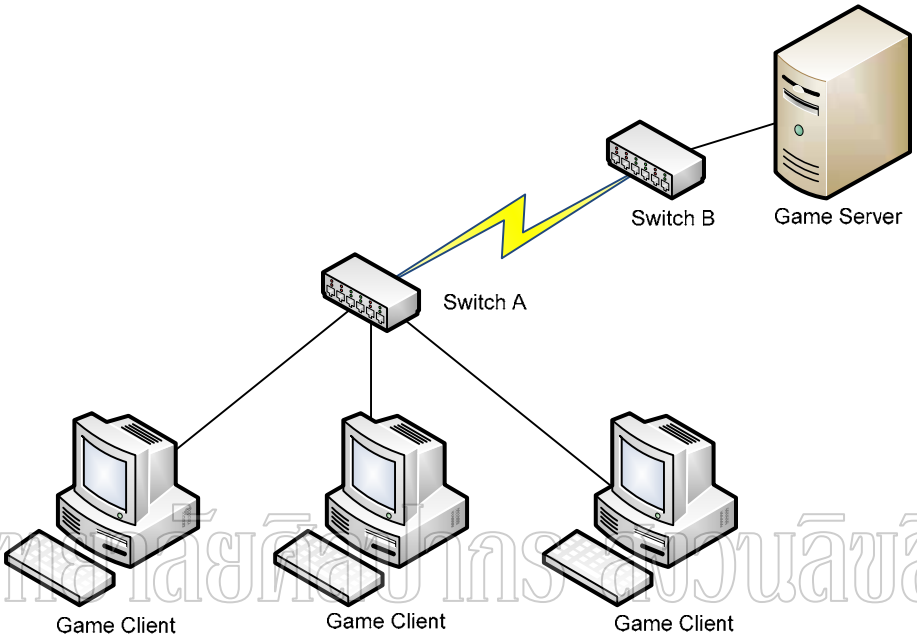


ภาพที่ 9 ภาพเครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียว

จากภาพที่ 9 จะเป็นการเชื่อมต่อของเครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียว ซึ่งมีการจัดสรร IP Address ให้กับเครื่องแต่ละเครื่องโดยวิธีการกำหนด IP Address หรือ ให้ระบบกำหนดให้อัตโนมัติหรือที่เรียกว่า DHCP ก็ได้เช่นกัน

3.2 เครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียวที่มีการทำ NAT

เครือข่ายแบบที่ 2 เปรียบเสมือนเครือข่ายขนาดใหญ่ของโปรโตคอล IPv4 ที่มีการใช้งานกันในปัจจุบัน คือจะมีการนำระบบ NAT (Network Address Translation) เข้ามาใช้งานอันเนื่องมาจาก IP Address โปรโตคอล IPv4 ที่แจกจ่ายให้กับกลุ่มองค์กรมีไม่เพียงพอ

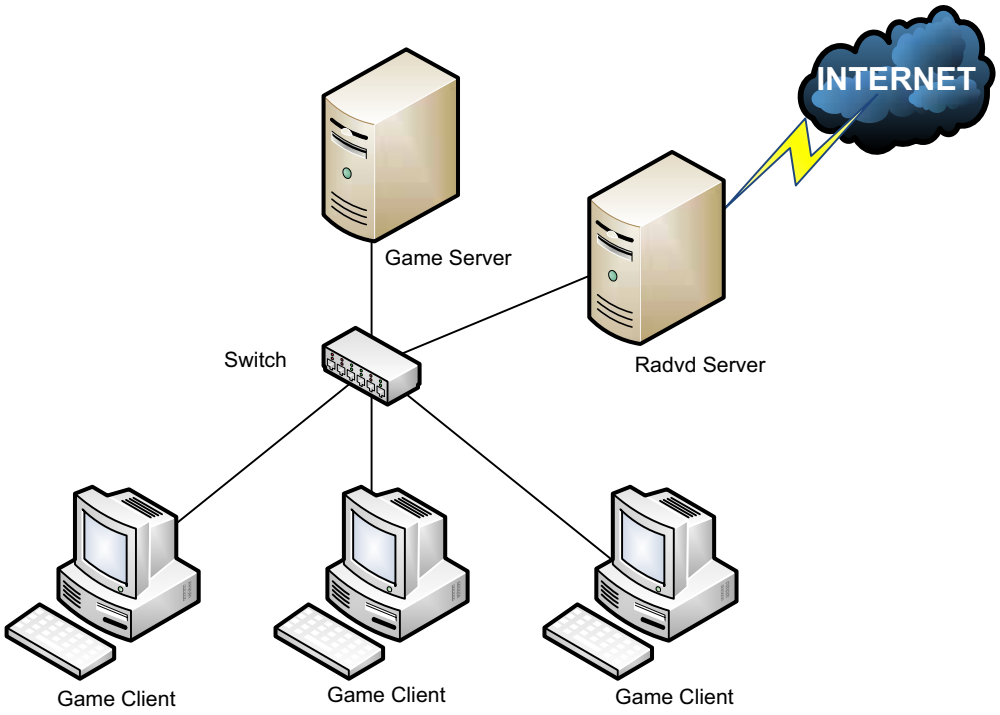


ภาพที่ 10 เครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียวที่มีการทำ NAT

จากภาพที่ 10 จะเป็นการเชื่อมต่อเครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียวที่มีการทำ NAT โดยการเชื่อมต่อเครือข่ายแบบนี้จุดประสงค์เพื่อเป็นการประหยัด IP Address โดยสามารถใช้ Private IP Address ภายในกลุ่มเครือข่ายและใช้ Switch, Gateway, Firewall หรือ Router ในการทำ NAT เพื่อให้ออกไปยังเครือข่ายอื่นเพียง IP Address เดียว จากภาพที่ 10 จะเห็นได้ว่า IP Address ของ Game Client ทั้งหมดจะถูกเปลี่ยนไปเป็น IP อีกชุดหนึ่งที่ Switch A แล้วส่งไปยังเครื่อง Game Server

3.3 เครือข่ายที่มีการใช้งานโปรโตคอล IPv6 เพียงอย่างเดียว

การเชื่อมต่อแบบที่ 3 เป็นการเชื่อมต่อของเครือข่ายโปรโตคอล IPv6 เปรียบเสมือนการเชื่อมต่อเครือข่ายโปรโตคอล IPv6 ที่ใช้งานกันในปัจจุบัน



ภาพที่ 11 ภาพเครือข่ายที่มีการใช้งานโปรโตคอล IPv6 เพียงอย่างเดียว

จากภาพที่ 11 เป็นเครือข่ายที่มีการใช้งานโปรโตคอล IPv6 เพียงอย่างเดียว ซึ่งมีการใช้ Radvd Server (Linux IPv6 Router Advertisement Daemon) มาเพื่อแจก IP Address ให้กับเครื่อง Client ที่อยู่ภายในเครือข่าย ซึ่ง IP Address ของแต่ละเครื่องจะมีการแปลงมาจาก MAC Address โดยรับมาจาก Radvd Server 64 bit และสร้างมาจาก MAC Address อีก 64 bit ตามหลักของ EUI-64 ตัวอย่างเช่น

34-56-78-9A-BC-DE

เปลี่ยนเป็น

34-56-78-FF-FE-9A-BC-DE

แล้วนำมารวมกับ 64 บิตแรกที่เครื่อง Radvd Server สร้างขึ้นมา

4. พัฒนาระบบเกมออนไลน์

พัฒนาระบบเกมออนไลน์โดยจุดประสงค์หลักเป็นการพัฒนาให้รองรับการทำงานของ IPv6 แต่เพื่อให้เห็นถึงประสิทธิภาพการทำงานของตัวระบบจะมีการเปรียบเทียบกับระบบของเกมที่อยู่บน โปรโตคอล IPv4 โดยสรุปจะมีการพัฒนาเพื่อเปรียบเทียบตามตารางที่ 3

ตารางที่ 3 รายการระบบที่มีการพัฒนาเพื่อเทียบประสิทธิภาพ

	IPv4	IPv4 with NAT	IPv6
เวลาในการส่งข้อมูล	✓	✓	✓

สาเหตุที่มีการพัฒนาระบบออกมาใน 3 รูปแบบตามตารางที่ 3.2 ก็เพื่อให้การทดสอบเป็นไปโดยมาตรฐานเดียวกันและให้เกมที่พัฒนาบนเครือข่าย IPv6 มีการทำงานที่เกิดประสิทธิภาพสูงสุด

5. ทดสอบระบบเกมออนไลน์

ทดสอบระบบแล้วเก็บเวลาในการรับส่งข้อมูลจำนวนแพ็คเกจจำนวนหนึ่ง โดยให้มีการส่งข้อมูลในปริมาณที่เท่ากันในเครือข่ายแต่ละแบบ แล้วหาความเร็วในการรับและส่งข้อมูลที่เกิดขึ้น โดยใช้จำนวนเครื่องไม่เท่ากันเช่น 1 เครื่อง 10 เครื่อง และ 30 เครื่องในเครื่องข่ายแต่ละแบบที่มีการออกแบบมาสำหรับการทดสอบในครั้งนี้ แล้วนำข้อมูลเหล่านี้มาวิเคราะห์หาค่าที่เหมาะสมในการนำเอาไปพัฒนาปรับปรุงระบบ

6. วิเคราะห์และประเมินผลการทดสอบระบบ

นำผลที่ได้จากการทดสอบมาวิเคราะห์ประสิทธิภาพของตัวระบบ โดยเทียบเวลาที่ใช้ในการรับส่งข้อมูลของเครือข่ายทั้ง 3 แบบได้แก่ เครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียว เครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียวที่มีการทำ NAT (Network Address Translation) และ เครือข่ายที่มีการใช้งานโปรโตคอล IPv6 เพียงอย่างเดียว

7. สรุปผลการวิจัยและจัดทำรายงานวิทยานิพนธ์

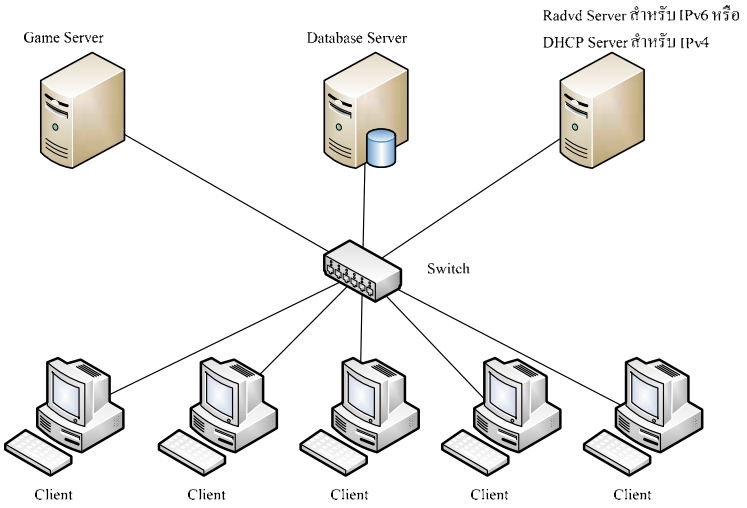
เมื่อการทำวิจัยสำเร็จตามวัตถุประสงค์เรียบร้อยแล้ว ก็สรุปผลการดำเนินการวิจัยและทำรายงานวิทยานิพนธ์

บทที่ 4
ผลการดำเนินการวิจัย

ในการดำเนินการวิจัย ผู้วิจัยได้พัฒนาโปรแกรมขึ้นด้วย Microsoft Visual C++ 6.0 เพื่อให้งานวิจัยบรรลุตามวัตถุประสงค์ที่ตั้งไว้ คือศึกษาและพัฒนาตัวแบบเพื่อใช้ในการเปรียบเทียบประสิทธิภาพการทำงานระหว่างอินเตอร์เน็ตโปรโตคอล IPv4 และ IPv6 ซึ่งมีการพัฒนาโปรแกรมเพื่อจำลองการทำงานของเกมออนไลน์ในสภาวะการทำงานของ IPv4 และ IPv6 โดยมีการควบคุมในเรื่องของจำนวนเครื่องและจำนวนของข้อมูลที่ส่งในเครือข่ายโดยแบ่งออกเป็นกรณีต่างๆ เพื่อให้การวัดประสิทธิภาพมีความชัดเจน รายละเอียดการดำเนินการวิจัยมีการแบ่งรูปแบบการทดลองออกเป็นดังนี้

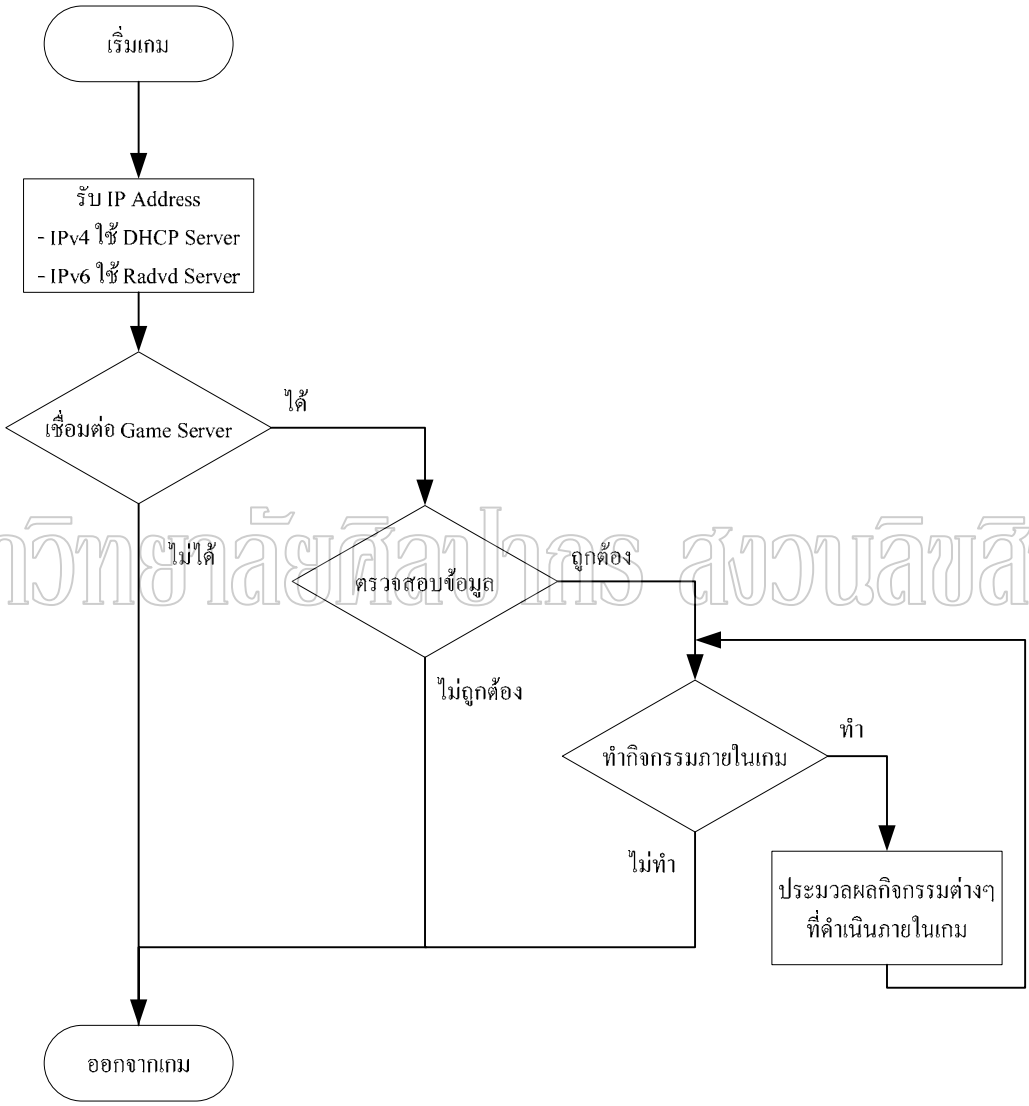
1. การทดลองแบบที่ 1 เครื่องลูกข่ายจำนวน 1 เครื่อง
2. การทดลองแบบที่ 2 เครื่องลูกข่ายจำนวน 10 เครื่อง
3. การทดลองแบบที่ 3 เครื่องลูกข่ายจำนวน 30 เครื่อง
4. การประเมินผลการทดลอง

การออกแบบการทำงานของระบบมีการออกแบบให้มีความใกล้เคียงกันมากที่สุดแล้วให้แตกต่างกันเฉพาะโปรโตคอลที่ใช้งานเท่านั้น การออกแบบระบบจะเป็นไปตามภาพที่ 12



ภาพที่ 12 แสดงการเชื่อมต่อเครือข่ายสำหรับการทดสอบ

ส่วนการออกแบบเครือข่ายสำหรับทดสอบแบบโปรโตคอล IPv4 ที่ผ่านการทำ NAT นั้นจะมีการใช้ Switch เป็นตัว NAT ซึ่งในการทดสอบแบบโปรโตคอล IPv4 และโปรโตคอล IPv6 จะไม่มีการกำหนดค่า Switch ให้เป็น NAT โดยรูปแบบการทำงานของโปรแกรมที่นำมาใช้ในการทดสอบจะเป็นไปภาพที่ 13



ภาพที่ 13 แสดงขั้นตอนการทำงานของโปแกรมที่ใช้ทำการทดสอบ

จากภาพที่ 13 จะเห็นได้ว่าการทำงานของโปรแกรมที่ออกแบบมาสำหรับการทดสอบจะมีลักษณะเหมือนเกมออนไลน์ทั่วไปที่ให้บริการ ซึ่งเกมจะมีการเชื่อมต่อเพื่อส่งข้อมูลกันตลอดเวลาหากมีการประมวลผลกิจกรรมต่างๆ ภายในเกม การทดสอบจึงมีการออกแบบข้อมูลและ

กิจกรรมต่างๆ เหมือนกันทั้ง 3 กรณี เพื่อให้สามารถวัดประสิทธิภาพการทำงานได้ เนื่องจากการเล่นเกมออนไลน์จะมีการส่งข้อมูลที่ไม่แน่นอน ทำให้ไม่สามารถวัดออกมาได้ นอกจากนี้ยังมีการควบคุมโปรแกรมอื่นๆ บนเครื่อง Game Client ที่มีการส่งข้อมูลบนเครือข่ายอีกด้วยให้มีลักษณะเดียวกัน และคุณสมบัติเครื่องเหมือนกัน และแยกการจับเวลาการทำงานเป็น 2 แบบคือ แบบที่ 1 ใช้โปรแกรมช่วยในการจับเวลา โปรแกรมที่ใช้คือ โปรแกรม Comview และแบบที่ 2 จับเวลาเองภายในโปรแกรมที่พัฒนาขึ้น

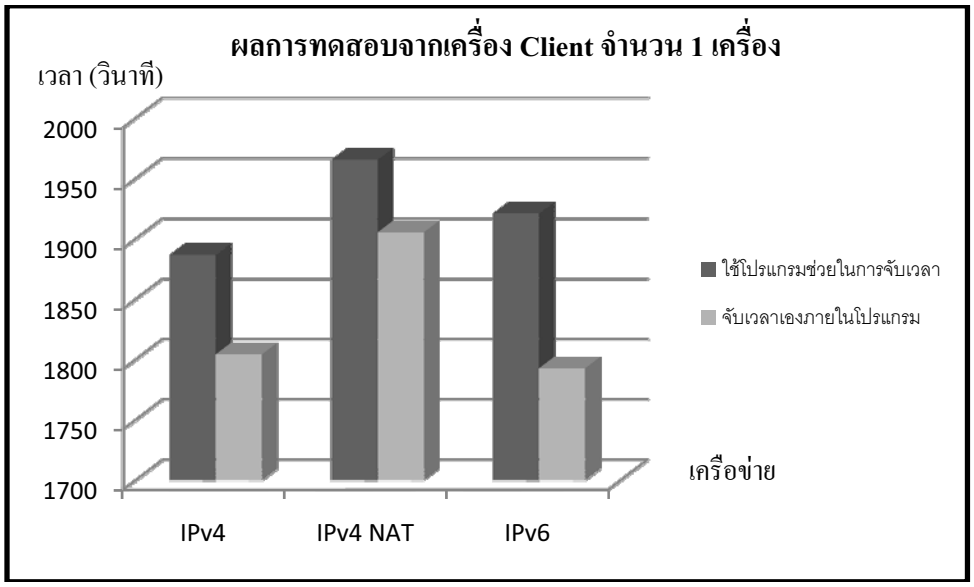
ขั้นตอนการทดสอบแบบต่างๆ สามารถอธิบายได้ดังนี้

1. การทดสอบแบบที่ 1 เครื่องลูกข่ายจำนวน 1 เครื่อง

การทดลองแบบที่ 1 ใช้เครื่องลูกข่ายจำนวน 1 เครื่องใช้ขนาดของข้อมูล 256KByte ต่อแพ็คเกจโดยมีการส่งทั้งหมด 5,000 ครั้งแล้วตรวจสอบผลการทำงาน ได้ผลการทดลองดังตารางที่ 4

ตารางที่ 4 แสดงผลการทดลองแบบที่ 1 เครื่องลูกข่ายจำนวน 1 เครื่อง

	IPv4	IPv4 NAT	IPv6
ใช้โปรแกรมช่วยในการจับเวลา	1886.83 วินาที	1965.56 วินาที	1921.08 วินาที
จับเวลาเองภายในโปรแกรม	1804.32 วินาที	1905.63 วินาที	1792.78 วินาที



แผนภูมิที่ 1 แสดงผลการทดลองแบบที่ 1 เครื่องลูกข่ายจำนวน 1 เครื่อง

จากแผนภูมิที่ 1 แสดงผลการทดลองในแบบที่ 1 จากการทดลองได้มีการทดลองกับเครือข่ายแบบต่างๆ 3 รูปแบบได้แก่ โปรโตคอล IPv4 โปรโตคอล IPv4 ผ่านการทำ NAT (Network Address Translation) และ โปรโตคอล IPv6 ได้ผลการทดลองดังนี้

สำหรับการจับเวลาโดยใช้โปรแกรมเข้ามาช่วยได้ผลดังนี้ โปรโตคอล IPv4 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 1886.83 วินาที โปรโตคอล IPv4 ผ่านการทำ NAT ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 1965.56 วินาที และ โปรโตคอล IPv6 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 1921.08 วินาที

สำหรับการจับเวลาโดยจับเวลาเองภายในโปรแกรมได้ผลดังนี้ โปรโตคอล IPv4 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 1804.32 วินาที โปรโตคอล IPv4 ผ่านการทำ NAT ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 1905.63 วินาที และ โปรโตคอล IPv6 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 1792.78 วินาที

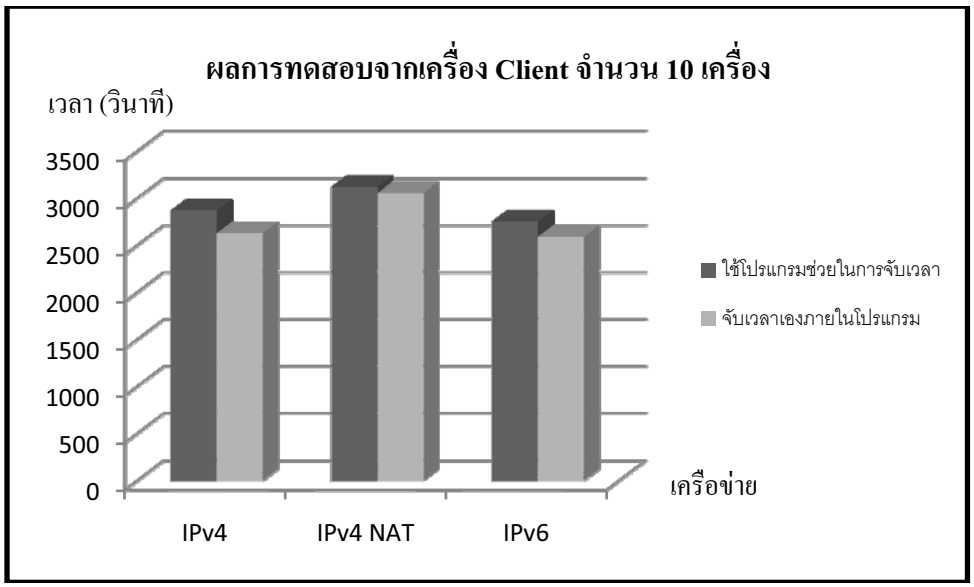
สรุปผลการทดลองในแบบที่ 1 ความเร็วในการรับส่งข้อมูลมีความเร็วแตกต่างกันเล็กน้อยโดยที่โปรโตคอล IPv4 และโปรโตคอล IPv6 มีความเร็วไม่ต่างกันมากนักซึ่งจะเห็นได้ว่าการใช้โปรแกรมมาช่วยในการจับเวลาจะทำให้การรับส่งข้อมูลช้าขึ้นอีกเล็กน้อย และโปรโตคอล IPv4 ผ่านการทำ NAT มีความล่าช้าในการส่งข้อมูลมากที่สุด จะเห็นได้ว่าขนาด Header ของโปรโตคอลมีผลในการรับส่งข้อมูล Header มีขนาดเล็กจะทำให้สามารถส่งข้อมูลได้อย่างรวดเร็ว ซึ่งในส่วนของการทำ NAT จะเสียเวลาในการเพิ่ม Header ของ NAT ทำให้โปรโตคอล IPv4 ผ่านการทำ NAT ช้าที่สุดในการทดลองชุดนี้

2. การทดสอบแบบที่ 2 เครื่องลูกข่ายจำนวน 10 เครื่อง

การทดลองแบบที่ 2 ใช้เครื่องลูกข่ายจำนวน 10 เครื่องใช้ขนาดของข้อมูล 256KByte ต่อแพ็คเกจโดยมีการส่งทั้งหมด 5,000 ครั้งแล้วตรวจสอบผลการทำงาน ได้ผลการทดลองดังตารางที่ 5

ตารางที่ 5 แสดงผลการทดลองแบบที่ 2 เครื่องลูกข่ายจำนวน 10 เครื่อง

	IPv4	IPv4 NAT	IPv6
ใช้โปรแกรมช่วยในการจับเวลา	2875.39 วินาที	3127.70 วินาที	2758.62 วินาที
จับเวลาเองภายในโปรแกรม	2632.21 วินาที	3061.32 วินาที	2594.87 วินาที



แผนภูมิที่ 2 แสดงผลการทดลองแบบที่ 2 เครื่องลูกข่ายจำนวน 10 เครื่อง

จากแผนภูมิที่ 2 แสดงผลการทดลองในแบบที่ 2 จากการทดลองได้มีการทดลองกับเครือข่ายรูปแบบเดียวกับการทดลองแบบที่ 1 ได้แก่ โปรโตคอล IPv4 โปรโตคอล IPv4 ผ่านการทำ NAT (Network Address Translation) และ โปรโตคอล IPv6 ได้ผลการทดลองดังนี้

สำหรับการจับเวลาโดยใช้โปรแกรมเข้ามาช่วยได้ผลดังนี้ โปรโตคอล IPv4 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 2875.39 วินาที โปรโตคอล IPv4 ผ่านการทำ NAT ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 3127.70 วินาที และ โปรโตคอล IPv6 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 2758.62 วินาที

สำหรับการจับเวลาโดยจับเวลาเองภายในโปรแกรมได้ผลดังนี้ โปรโตคอล IPv4 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 2632.21 วินาที โปรโตคอล IPv4 ผ่านการทำ NAT ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 3061.32 วินาที และ โปรโตคอล IPv6 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 2594.87 วินาที

สรุปผลการทดลองในแบบที่ 2 ความเร็วในการรับส่งข้อมูลเริ่มมีความชัดเจนมากขึ้น โดย โปรโตคอล IPv6 มีความเร็วในการรับส่งข้อมูลมากที่สุด อันดับที่สองจะเป็นโปรโตคอล IPv4 และ โปรโตคอล IPv4 ผ่านการทำ NAT มีความล่าช้าในการส่งข้อมูลมากที่สุด จากการทดลองในแบบที่ 2 เมื่อเทียบกับการทดลองในแบบที่ 1 จะเห็นความแตกต่างของโปรโตคอล IPv6 ที่จากการทดลองในแบบที่ 1 ช้ากว่า โปรโตคอล IPv4 แต่กลับทำงานได้เร็วกว่าในการทดลองแบบที่ 2 จะเห็นได้ว่าความซับซ้อนของ Header มีส่วนสำคัญต่อการประมวลผล และส่งข้อมูลในเครือข่ายยิ่ง

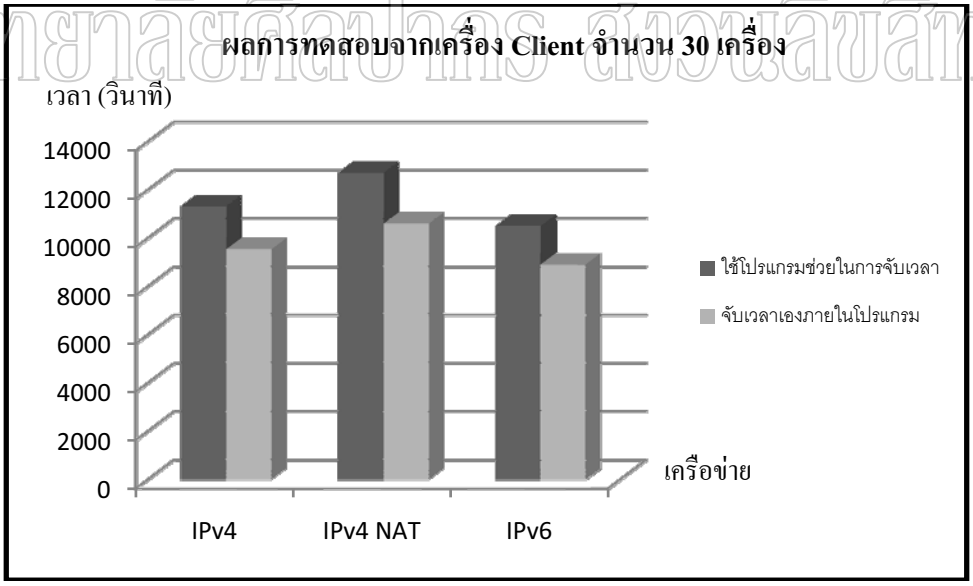
Header มีความซับซ้อนมาก การรับส่งข้อมูลจำนวนมากก็จะทำให้มีความล่าช้ากว่าการส่งโดยการใช้โปรโตคอลที่มี Header ไม่ซับซ้อน ซึ่ง IPv6 ได้มีการตัด Header ที่ไม่จำเป็นออกทำให้การประมวลผลในส่วนนี้ลดลง จึงทำให้รับส่งข้อมูลได้เร็วกว่า IPv4

3. การทดสอบแบบที่ 3 เครื่องลูกข่ายจำนวน 30 เครื่อง

การทดลองแบบที่ 3 ใช้เครื่องลูกข่ายจำนวน 30 เครื่องใช้ขนาดของข้อมูล 256KByte ต่อแพ็คเกจโดยมีการส่งทั้งหมด 5,000 ครั้งแล้วตรวจสอบผลการทำงาน ได้ผลการทดลองดังตารางที่ 6

ตารางที่ 6 แสดงผลการทดลองแบบที่ 3 เครื่องลูกข่ายจำนวน 30 เครื่อง

	IPv4	IPv4 NAT	IPv6
ใช้โปรแกรมช่วยในการจับเวลา	11333.02 วินาที	12722.30 วินาที	10516.83 วินาที
จับเวลาเองภายในโปรแกรม	9554.34 วินาที	10615.69 วินาที	8908.4 วินาที



แผนภูมิที่ 3 แสดงผลการทดลองแบบที่ 3 เครื่องลูกข่ายจำนวน 30 เครื่อง

จากแผนภูมิที่ 3 แสดงผลการทดลองในแบบที่ 3 จากการทดลองได้มีการทดลองกับเครือข่ายรูปแบบเดียวกับการทดลองแบบที่ 1 และการทดลองแบบที่ 2 ได้แก่ โปรโตคอล IPv4

โปรโตคอล IPv4 ผ่านการทำ NAT (Network Address Translation) และ โปรโตคอล IPv6 ได้ผลการทดลองดังนี้

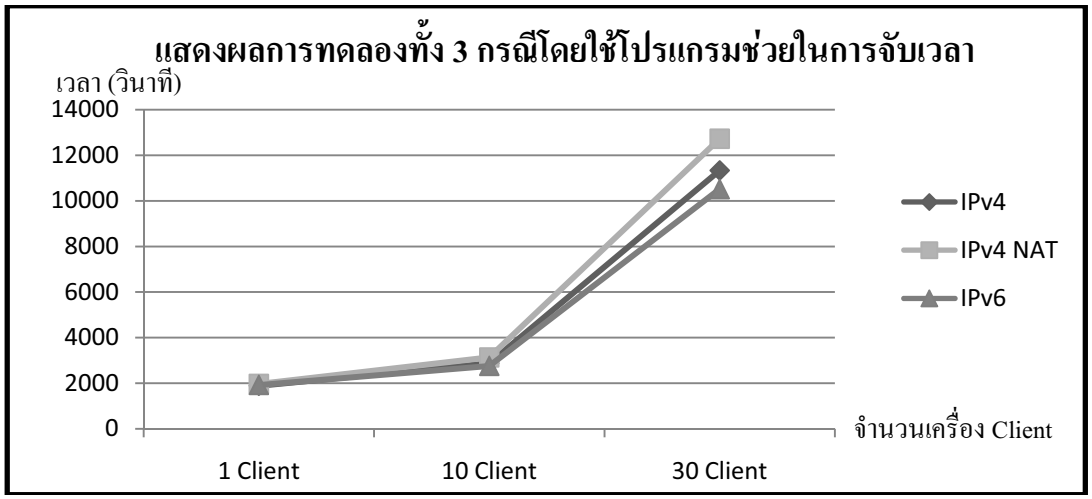
สำหรับการจับเวลาโดยใช้โปรแกรมเข้ามาช่วยได้ผลดังนี้โปรโตคอล IPv4 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 11333.02 วินาที โปรโตคอล IPv4 ผ่านการทำ NAT ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 12722.30 วินาที และ โปรโตคอล IPv6 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 10516.83 วินาที

สำหรับการจับเวลาโดยจับเวลาเองภายในโปรแกรมได้ผลดังนี้ โปรโตคอล IPv4 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 9554.34 วินาที โปรโตคอล IPv4 ผ่านการทำ NAT ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 10615.69 วินาที และ โปรโตคอล IPv6 ใช้ความเร็วในการรับส่งข้อมูลทั้งหมดเฉลี่ย 8908.4 วินาที

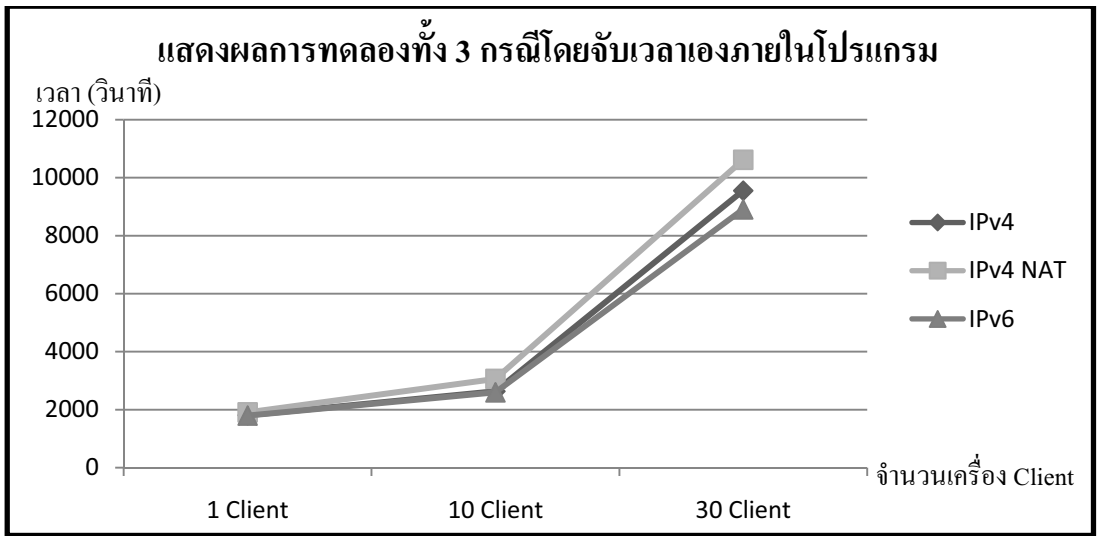
สรุปผลการทดลองในแบบที่ 3 ความเร็วในการรับส่งข้อมูลเริ่มมีความชัดเจนมากกว่าการทดลองในแบบที่ 1 และการทดลองในแบบที่ 2 โดย โปรโตคอล IPv6 มีความเร็วในการรับส่งข้อมูลมากที่สุด อันดับที่สองจะเป็นโปรโตคอล IPv4 และ โปรโตคอล IPv4 ผ่านการทำ NAT มีความล่าช้าในการส่งข้อมูลมากที่สุด จากการทดลองนี้เป็นการยืนยันได้ว่าความซับซ้อนของ Header มีส่วนสำคัญต่อการประมวลผล และส่งข้อมูลในเครือข่าย

4. การประเมินผลการทดสอบระบบ

จากการทดลองทั้งสามแบบเมื่อนำมาแสดงเป็นกราฟเส้นเพื่อให้เกิดความชัดเจนมากยิ่งขึ้นจะได้ตามแผนภูมิที่ 4 และ แผนภูมิที่ 5



แผนภูมิที่ 4 แสดงผลการทดลองทั้ง 3 กรณีโดยใช้โปรแกรมช่วยในการจับเวลา



แผนภูมิที่ 5 แสดงผลการทดลองทั้ง 3 กรณีโดยจับเวลาเองภายในโปรแกรม

จากการทดลองทั้ง 3 แบบเมื่อนำมาเปรียบเทียบกันจะเห็นได้ว่าการทำงานของโปรโตคอล IPv6 เมื่อเทียบกับโปรโตคอล IPv4 นั้น หากทำงานโดยใช้เครื่องคอมพิวเตอร์จำนวนน้อยประสิทธิภาพในการทำงานของโปรโตคอล IPv6 และโปรโตคอล IPv4 มีความแตกต่างกันไม่มากนักโดย แต่เมื่อมีการเพิ่มจำนวนเครื่องคอมพิวเตอร์มากขึ้นการทำงานของโปรโตคอล IPv4 ประสิทธิภาพจะลดลง เมื่อเทียบกับโปรโตคอล IPv6 ดังนั้นได้ว่าโปรโตคอล IPv6 จะทำงานได้กับเครือข่ายขนาดใหญ่ที่มีเครื่องคอมพิวเตอร์จำนวนมากได้ดีกว่าโปรโตคอล IPv4 เพราะจากการออกแบบของโปรโตคอล IPv6 ได้มีการลด Header ที่ไม่จำเป็นออกแล้วนำส่วนที่ไม่ค่อยได้ใช้งานไปเป็นส่วน Option ซึ่งต่างกับ IPv4 ที่มี Header ที่ไม่จำเป็นและไม่ค่อยได้ใช้งานอยู่ และการทำ NAT (Network Address Translation) ใน IPv4 ยังเป็นการเพิ่ม Header ให้กับแพ็คเกจในการรับส่งข้อมูลอีกด้วย จากการทดลองจะเห็นได้อย่างชัดเจนว่าโปรโตคอล IPv6 จะทำงานได้ดีเมื่อเครือข่ายมีขนาดใหญ่ขึ้นเมื่อเทียบกับโปรโตคอล IPv4

บทที่ 5
สรุป อภิปรายผลและข้อเสนอแนะ

จากการดำเนินการพัฒนาระบบเกมออนไลน์เพื่อใช้ในการทดสอบประสิทธิภาพการทำงานของเครือข่ายโปรโตคอล IPv4 และ IPv6 ทำให้ได้โปรแกรมจำลองการทำงานของเกมออนไลน์ซึ่งได้มีการนำไปทดลองกับเครือข่ายที่ออกแบบมาสำหรับการทำงานแบบต่างๆ ซึ่งมีการควบคุมตัวแปรต่างๆ ที่จำเป็นและมีผลต่อการทดลองเช่น ขนาดของข้อมูลที่มีการรับและส่งกันภายในเครือข่าย รวมถึงปริมาณที่ใช้ส่งให้เท่ากัน ซึ่งเหล่านี้เป็นตัวแปรควบคุมที่มีผลต่อตัวแปรต้นที่ต้องการหาคำตอบ แล้วนำผลที่ได้มาวิเคราะห์เพื่อ เปรียบเทียบการทำงานในรูปแบบที่ทำการทดลอง โดยสรุปได้ดังนี้

1. การบรรลุวัตถุประสงค์การวิจัย

การเปรียบเทียบประสิทธิภาพของเกมออนไลน์บนระหว่างโปรโตคอล IPv4 และ IPv6 ได้กำหนดวัตถุประสงค์ไว้ดังนี้

1.1 เพื่อศึกษาหลักการและวิธีการในการจัดการข้อมูลและกระบวนการรับส่งข้อมูลบนโปรโตคอล IPv6 และการพัฒนาโปรแกรมบนโปรโตคอล IPv6

1.2 เพื่อทดสอบและประเมินผลเกมออนไลน์ที่พัฒนาขึ้น เทียบประสิทธิภาพที่เกิดขึ้นระหว่างโปรโตคอล IPv4 และโปรโตคอล IPv6

เมื่อการพัฒนาระบบเสร็จสิ้น และได้ทดสอบการทำงานของระบบ ทำให้ระบบงานนี้บรรลุวัตถุประสงค์ตามที่ตั้งไว้คือ

1.1 มีโปรแกรมจำลองการทำงานของเกมออนไลน์สำหรับทดสอบประสิทธิภาพการทำงานของเครือข่ายที่ใช้งานโปรโตคอล IPv4 และโปรโตคอล IPv6

1.2 มีการสร้างเครือข่ายสำหรับทดสอบการทำงานของเกมออนไลน์ที่พัฒนาขึ้นโดยมีการแยกออกเป็นกรณีต่างๆ 3 กรณีได้แก่ เครือข่ายที่มีการใช้งานโปรโตคอล IPv4เพียงอย่างเดียว เครือข่ายที่มีการใช้งานโปรโตคอล IPv4 เพียงอย่างเดียวที่มีการทำ NAT (Network Address Translation) และ เครือข่ายที่มีการใช้งานโปรโตคอล IPv6 เพียงอย่างเดียว

จากผลการทดลองกับเครือข่ายแบบต่างๆ นี้ได้มีงานวิจัยของต่างประเทศได้มีการทำงานวิจัยในลักษณะเดียวกันนี้แต่จะเป็นการชี้เฉพาะเจาะจงลงไปในเรื่องของข้อมูลการจราจรของเกมคือ Traffic Modelling for Fast Action Network Games (Johannes Färber 2000 : 53-57)

2. สรุปผลการวิจัย

จากการวิจัยที่มีการออกแบบระบบเครือข่ายเป็นแบบต่างๆ ทั้ง 3 แบบได้แก่ แบบที่ 1 เครือข่ายโปรโตคอล IPv4 แบบที่ 2 เครือข่ายโปรโตคอล IPv4แบบผ่านการทำ NAT และแบบที่ 3 เครือข่ายโปรโตคอล IPv6 และมีการทดสอบในแบบต่างๆ ตามจำนวนเครื่องเริ่มตั้งแต่ 1 เครื่อง 10 เครื่อง และ 30 เครื่องพบว่า หากมีการใช้จำนวนเครื่องน้อยความเร็วในการรับส่งข้อมูลเครือข่ายโปรโตคอล IPv4 และ เครือข่ายโปรโตคอล IPv6 จะมีความเร็วแตกต่างกันเล็กน้อยโดย เครือข่ายโปรโตคอล IPv4 จะเร็วกว่าเล็กน้อย แต่เมื่อมีการใช้เครื่องมาทำการทดลองจำนวนมากขึ้นความเร็วในการรับส่งข้อมูล ประสิทธิภาพของเครือข่ายโปรโตคอล IPv6 มีแนวโน้มดีกว่าของเครือข่ายโปรโตคอล IPv4

3. ข้อเสนอแนะ

การเปรียบเทียบประสิทธิภาพของเกมออนไลน์บนระหว่างโปรโตคอล IPv4 และ IPv6 เป็นเพียงจุดเริ่มต้นที่จะมีการนำเอาโปรโตคอล IPv6 มาใช้งาน ผู้วิจัยมีข้อเสนอแนะบางประการเพื่อให้ตัวแบบที่สร้างขึ้นและโปรแกรมที่พัฒนาขึ้นมีประสิทธิภาพดีขึ้น และสามารถนำไปพัฒนาต่อในอนาคตได้ดังนี้

3.1 ศึกษาพัฒนาระบบที่มีการใช้งานความสามารถอื่นๆ ของเครือข่ายโปรโตคอล IPv6 ให้สามารถนำไปใช้งานกับเกมออนไลน์ได้อย่างมีประสิทธิภาพ

3.2 พัฒนาเกมออนไลน์ให้สามารถใช้งานบนเครือข่ายโปรโตคอล IPv6 ในเชิงพานิช

บรรณานุกรม

ภาษาไทย

Centre for Network Research. IPv6 Testbed [ออนไลน์]. เข้าถึงเมื่อ 10 มกราคม 2550. เข้าถึงได้จาก http://cnr.coe.psu.ac.th/ipv6_1_t.htm

Centre for Network Research. VoIP:CNR-SIP [ออนไลน์]. เข้าถึงเมื่อ 10 มกราคม 2550. เข้าถึงได้จาก http://cnr.coe.psu.ac.th/cnrsip_1_t.htm

จตุชัย แพงจันทร์ และ อนุโชต วุฒิพรพงษ์. เจาะระบบ Network. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : โรงพิมพ์ด่านสุทธา การพิมพ์, 2547.

พหล โสตถิวิรัช และคณะ. Investigation of Duplicate Address Detection in Mobile IPv6 [ออนไลน์]. เข้าถึงเมื่อ 5 พฤษภาคม 2552. เข้าถึงได้จาก http://wiki.nectec.or.th/ngiwiki/pub/Project/ ImproveDAD/ Investigation_of_Duplicate_Address_Detection_in_Mobile_IPv6.pdf

วโรดม วีระพันธ์ และ อภิเนตร อูนากูล. Performance Analysis Of Mobile IPv6 For Linux Testbed System. กรุงเทพมหานคร : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546.

สงกรานต์ ทองสว่าง. MySQL ฐานข้อมูลสำหรับอินเตอร์เน็ต. กรุงเทพมหานคร : บริษัทซีเอ็ดยูเคชั่น จำกัด, 2544.

ภาษาต่างประเทศ

Bouras, Christos., Primpas, Dimitris and Stamos, Kostas. “IPv6 QoS Testing on Dual Stack Network.” In Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications, Article No. 3. Italy : ACM, 2006.

C.Bouras, A.Gkamas, D.Primpas and K.Stamos. “IPv6 deployment: Real time applications and QoS aspects.” In Computer Communications 29, 1393–1401. Greece : Science direct, 2006.

Färber, Johannes. “Network Game Traffic Modeling.” In Proceedings of the 1st workshop on Network and system support for games, 53-57. Germany : ACM, 2002.

García-Macías, J. Antonio., Rousseau, Franck., Berger-Sabbatel, Gilles., Toumi, Leyla and Duda, Andrzej. “Quality of Service and Mobility for the Wireless Internet.” Wireless Networks, Volume 9 (July 2003) : 341-352.

Jiann-Liang Chen, Yu-Feng Lee and Yao-Chung Chang. “Mobile IPv6 network: implementation and application.” International Journal of Network Management, Volume 16 (January 2006) : 29-43.

Kuan-Ta Chen, Polly Huang, Chun-Ying Huang and Chin-Laung Lei. “Game Traffic Analysis: An MMORPG Perspective.” In Proceedings of the international workshop on Network and operating systems support for digital audio and video, 19-24. Washington : ACM, 2005.

Perkins, Charles and Johnson, David. “Mobility Support in IPv6.” In Proceedings of the 2nd annual international conference on Mobile computing and networking, 27-37. New York : ACM, 1996.

Xiaoming Zhou, Rob E. Kooij, Henk Uijterwaal and Piet Van Miegham. “Estimation of Perceived Quality of Service for Applications on IPv6 Networks.” In Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks, 74-81. Spain : ACM, 2006.

ภาคผนวก

ภาคผนวก ก

รายละเอียดการพัฒนาโปรแกรม

รายละเอียดการพัฒนาโปรแกรม

จากการพัฒนาระบบทำให้ได้โปรแกรมสำหรับทดสอบประสิทธิภาพการทำงานของเกมออนไลน์บนอินเตอร์เน็ตโปรโตคอล IPv4 และ IPv6 โดยโปรแกรมจำทำงานในลักษณะจำลองการทำงานของเกมออนไลน์ เพื่อให้การวัดประสิทธิภาพการทำงานของอินเตอร์เน็ตโปรโตคอล IPv4 และ IPv6 มีความชัดเจนมากยิ่งขึ้น และการจำลองการทำงานนี้ทำให้สามารถควบคุมตัวแปรที่มีผลต่อการทำงานได้ดียิ่งขึ้น รวมถึงการทดสอบระบบที่สะดวก ทำให้สามารถจัดเก็บข้อมูลที่จำเป็นในการวิเคราะห์ประสิทธิภาพในการทำงานของเกมออนไลน์ได้ดียิ่งขึ้น

การพัฒนาโปรแกรมพัฒนาโดยการใช้ Microsoft VC++ 6.0 และใช้ Network API เป็น DirectPlay ซึ่งอยู่ในชุดพัฒนา DirectX 9.0 SDK ของบริษัท Microsoft เนื่องจากเป็น Network API ที่เป็นที่นิยมและมีข้อมูลสนับสนุนในการพัฒนาโปรแกรมมาก จากการใช้งาน DirectPlay เป็นหลักในการพัฒนาโปรแกรมทำให้การทำงานของตัวโปรแกรมต้องทำงานบนระบบปฏิบัติการ Windows เท่านั้น ซึ่งการทำงานจะมีหลักการเช่นเดียวกับเกมออนไลน์ที่มีการให้บริการทั่วๆ ไป โดยมีการแยกการทำงานเป็นออกสองส่วนหลักดังต่อไปนี้

- Game Server
- Game Client
- ข้อมูลที่มีการรับและส่งระหว่าง Game Server และ Game Client

การทำงานของเกมออนไลน์แต่ละส่วนอธิบายได้ดังนี้

1. Game Server

ในการทำงานของ Game Server นั้นเริ่มต้นด้วยการให้ Network API DirectPlay เชื่อมต่อกับโปรโตคอล TCP/IP เพื่อจองหน่วยความจำและกำหนดสภาพแวดล้อมที่จำเป็นในการทำงาน รวมทั้งกำหนดหน้าต่างหลักในการทำงานโดยมีคำสั่งในการทำงานดังนี้

```
if(CoInitialize(NULL) != S_OK)
        MessageBox(hmaindlg,"Server Initialize failed.","Error report",MB_OK);
```

```
CoCreateInstance(CLSID_DirectPlay8Server,NULL,
CLSCTX_INPROC_SERVER,IID_IDirectPlay8Server,(LPVOID*)&ds);
        ds->Initialize(NULL,DPlayMessageMap,0);
CoCreateInstance(CLSID_DirectPlay8Address,NULL,
        CLSCTX_INPROC_SERVER,IID_IDirectPlay8Address,(LPVOID*)&devicead
dress);
deviceaddress->SetSP(&CLSID_DP8SP_TCPIP);
DialogBox(hinstance,MAKEINTRESOURCE(IDD_MAINDIALOG),NULL,
(DLGPROC)MainDlgProc);
```

หลังจากการเชื่อมต่อกับโปรโตคอล TCP/IP และจองหน่วยความจำบางส่วนเรียบร้อยแล้ว ซึ่งในส่วนการทำงานนี้จะมีการระบุฟังก์ชันในการตรวจจับ Message ของระบบปฏิบัติการ Windows คือ ฟังก์ชัน MainDlgProc และฟังก์ชันในการตรวจจับ Message ของ Network API DirectPlay คือฟังก์ชัน DPlayMessageMap

ขั้นตอนการทำงานอันดับต่อมาก็ได้แก่การเริ่มการทำงานของ Game Server โดยการทำงานในส่วนนี้จะเป็นการกำหนดชื่อของ Game Server จำนวนผู้เล่นสูงสุดที่สามารถเชื่อมต่อเข้ามายังเครื่อง Game Server ได้ รวมไปถึง Port ในการเชื่อมต่อด้วย โดยฟังก์ชันในการทำงานดังนี้


```
HRESULT StartServer(char servername[256],int maxplayer){
        HRESULT         hr=S_OK;
        DPN_PLAYER_INFO    playerinfo;
        WCHAR           wservername[256];
        DXUtil_ConvertGenericStringToWide(wservername, servername);
        ZeroMemory( &playerinfo, sizeof(DPN_PLAYER_INFO));
        playerinfo.dwSize = sizeof(DPN_PLAYER_INFO);
        ...
        ...
        hr = ds->Host(&appdesc,&deviceaddress,1,NULL,NULL,NULL,NULL);
        if(FAILED(hr)){
                return hr;
```

```
        }
        return hr;
}
```

จากการทำงานของฟังก์ชัน StartServer หากทำงานสำเร็จจะส่งค่าคืนเพื่อบอกให้ทราบว่าพร้อมให้เครื่อง Game Client เชื่อมต่อเข้ามาได้ แล้วรอรับการเชื่อมต่อจาก Game Client เพื่อประมวลผลข้อมูลต่างๆ ที่ส่งมาแล้วตอบกลับไปโดยฟังก์ชัน DPlayMessageMap ในการทำงาน

```
HRESULT WINAPI DPlayMessageMap(PVOID UsetContext,DWORD MessageID,
PVOID MessageData)
```

ฟังก์ชันทำหน้าที่ตรวจจับ Message การทำงานของ Network API DirectPlay

Game Client

ในการทำงานของ Game Client นั้นเริ่มต้นด้วยการให้ Network API DirectPlay เชื่อมต่อกับโปรโตคอล TCP/IP เพื่อจองหน่วยความจำและกำหนดสภาพแวดล้อมที่จำเป็นในการทำงาน รวมทั้งกำหนดหน้าต่างหลักในการทำงาน ซึ่งการทำงานในส่วนนี้เหมือนการทำงานของ Game Server โดยมีคำสั่งในการทำงานดังนี้

```
if(CoInitialize(NULL) != S_OK)
        MessageBox(hmaindlg,"Server Initialize failed.","Error report",MB_OK);
CoCreateInstance(CLSID_DirectPlay8Client, NULL,
        CLSCTX_INPROC_SERVER, IID_IDirectPlay8Client,(LPVOID*)&dc);
dc->Initialize(NULL,DPlayMessageMap,0);
CoCreateInstance(CLSID_DirectPlay8Address,NULL,CLSCTX_INPROC_SERVER,
IID_IDirectPlay8Address,(LPVOID*)&deviceaddress);
deviceaddress->SetSP(&CLSID_DP8SP_TCPIP);
CoCreateInstance(CLSID_DirectPlay8Address,NULL,CLSCTX_INPROC_SERVER,
IID_IDirectPlay8Address,(LPVOID*)&hostaddress);
hostaddress->SetSP(&CLSID_DP8SP_TCPIP);
```

```
DialogBox(hinstance,MAKEINTRESOURCE(IDD_MAINDIALOG),NULL,
(DLGPROC)MainDlgProc);
```

หลังจากการเชื่อมต่อกับโปรโตคอล TCP/IP และจองหน่วยความจำบางส่วนเรียบร้อยแล้ว ซึ่งการทำงานจะเหมือนกับ Game Server มีทำงานในส่วนนี้เรียบร้อยแล้ว Game Client จึงเชื่อมต่อไปยังเครื่อง Game Server โดยฟังก์ชันในการทำงานดังนี้

```
HRESULT Connect(char ipaddress[256],char playername[256]){
        HRESULT         hr=S_OK;
        DPN_PLAYER_INFO    playerinfo;
        WCHAR           wpeername[256];
        DXUtil_ConvertGenericStringToWide(wpeername, playername);
        ZeroMemory(&playerinfo, sizeof(DPN_PLAYER_INFO));
        playerinfo.dwSize = sizeof(DPN_PLAYER_INFO);
        playerinfo.dwInfoFlags = DPNINFO_NAME;
        playerinfo.pwszName = wpeername;
        hr = dc->SetClientInfo(&playerinfo, NULL, NULL,
DPNSETCLIENTINFO_SYNC);
        …
        …
        hr = dc->Connect(&appdesc, hostaddress, deviceaddress, NULL, NULL, NULL,
NULL, NULL, &ConnectAsyncOp, NULL);
        if(FAILED(hr)){
                return hr;
        }
        return hr;
}
```


จากการทำงานของฟังก์ชัน Connect หากมีการเชื่อมต่อกับ Game Server สำเร็จ Game Client ก็สามารถเริ่มรับส่งข้อมูลในลักษณะของเกมออนไลน์ โดยข้อมูลที่มีการรับส่งจะเป็นข้อมูล

ที่ออกแบบมาสำหรับใช้ทดสอบประสิทธิภาพการทำงานของอินเตอร์เน็ตโปรโตคอล IPv4 และ IPv6

ข้อมูลที่มีการรับและส่งระหว่าง Game Server และ Game Client

ข้อมูลที่มีการรับส่งในโปรแกรมถูกออกแบบเพื่อทำหน้าที่ต่างๆ ภายในเกม เช่น เคลื่อนย้ายตำแหน่งของตัวละคร เพิ่มและลดจำนวนเงิน พลังชีวิต และค่าอื่นๆ ที่ใช้ภายในเกม รวมถึงใช้สำหรับทดสอบด้วย รายละเอียดมีดังต่อไปนี้

```
struct CHATPACKET{              //packet id = 1
        int flag;
        char message[256];
};

struct LOGINPACKET{             //packet id = 2
        int flag;
        char loginname[256];
        char password[256];
};

struct LOGINPASSPACKET{ //packet id = 3
        int flag;
        BOOL logincomplete;
};
// MONEY
struct UPDATE_MONEY{ //packet id = 4
        int flag;
        char account[40];
        char money_num[15];
};
```

```
struct MONEY_RECEIVE{ //packet id = 5
        int flag;
        char account[40];
        BOOL money_update_complete;
        char money_num[15];
};

// POSITION (MAP)
struct UPDATE_MAP{ //packet id = 6  : UPDATE POSITION
        int flag;
        char account[40];
        char mapname[50];
        char pos_x[3];
        char pos_y[3];
};
```



```
struct MAP_RECEIVE{ //packet id = 7     : UPDATE POSITION RECEIVE
        int flag;
        char account[40];
        BOOL map_update_complete;
        char mapname[50];
        char pos_x[3];
        char pos_y[3];
};

// HP
struct UPDATE_HP{ //packet id = 8
        int flag;
        char account[40];
        char hp[3];
```

```
};

struct HP_RECEIVE{ //packet id = 9
        int flag;
        char account[40];
        BOOL hp_update_complete;
        char hp[3];
};

// MP
struct UPDATE_MP{ //packet id = 10
        int flag;
        char account[40];
        char mp[3];
};

struct MP_RECEIVE{ //packet id = 11
        int flag;
        char account[40];
        BOOL mp_update_complete;
        char mp[3];
};

// STATUS STR
struct UPDATE_STATUS_STR{ //packet id = 12
        int flag;
        char account[40];
        char str_num[3];
};
```

```
struct STATUS_STR_RECEIVE{ //packet id = 13
        int flag;
        char account[40];
        BOOL str_update_complete;
        char str_num[3];
};

// STATUS AGI
struct UPDATE_STATUS_AGI{ //packet id = 14
        int flag;
        char account[40];
        char agi_num[3];
};

struct STATUS_AGI_RECEIVE{ //packet id = 15
        int flag;
        char account[40];
        BOOL agi_update_complete;
        char agi_num[3];
};

// STATUS VIT
struct UPDATE_STATUS_VIT{ //packet id = 16
        int flag;
        char account[40];
        char vit_num[3];
};

struct STATUS_VIT_RECEIVE{ //packet id = 17
        int flag;
```

```
        char account[40];
        BOOL vit_update_complete;
        char vit_num[3];
};

// STATUS INT
struct UPDATE_STATUS_INT{ //packet id = 18
        int flag;
        char account[40];
        char int_num[3];
};

struct STATUS_INT_RECEIVE{ //packet id = 19
        int flag;
        char account[40];
        BOOL int_update_complete;
        char int_num[3];
};

// STATUS DEX
struct UPDATE_STATUS_DEX{ //packet id = 20
        int flag;
        char account[40];
        char dex_num[3];
};

struct STATUS_DEX_RECEIVE{ //packet id = 21
        int flag;
        char account[40];
        BOOL dex_update_complete;
```

```
        char dex_num[3];
};

// STATUS LUK
struct UPDATE_STATUS_LUK{ //packet id = 22
        int flag;
        char account[40];
        char luk_num[3];
};

struct STATUS_LUK_RECEIVE{ //packet id = 23
        int flag;
        char account[40];
        BOOL luk_update_complete;
        char luk_num[3];
};

// UPDATE SAVEMAP
struct UPDATE_SAVEMAP{ //packet id = 24
        int flag;
        char account[40];
        char savemapname[50];
        char spos_x[3];
        char spos_y[3];
};

struct SAVEMAP_RECEIVE{ //packet id = 25
        int flag;
        char account[40];
        BOOL savemap_update_complete;
```

```
        char savemapname[50];
        char spos_x[3];
        char spos_y[3];
};

// CHARACTER STATUS
struct CHARSTATUS{   //packet id = 26
        int flag;
        char account[40];
        char char_hp[6];
        char char_mp[6];
        char char_str[3];
        char char_agi[3];
        char char_vit[3];
        char char_int[3];
        char char_dex[3];
        char char_luk[3];
        char char_mapname[50];
        char char_posx[3];
        char char_posy[3];
        char char_money[15];
};

// Send BITMAP
struct IMG_BITMAP{ // packet id = 27
        int flag;
        char account[40];
        HBITMAP hBitmap;
};
```

```
//Large TEXT
struct LARGETEXT{ // packet id = 28
        int flag;
        char account[40];
        char LText[262144];
};
```

ประวัติผู้วิจัย

ชื่อ-สกุล	นายอำนาจ ช้างเขียว
ที่อยู่	124 หมู่ 8 ตำบลด่านช้าง อำเภอด่านช้าง จังหวัดสุพรรณบุรี 72180
ที่ทำงาน	ศูนย์คอมพิวเตอร์ มหาวิทยาลัยราชภัฏนครปฐม 85 ถนนมาลัยแมน อำเภอเมือง จังหวัดนครปฐม 73000
ประวัติการศึกษา	
พ.ศ. 2545	สำเร็จการศึกษาปริญญาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยราชภัฏนครปฐม
พ.ศ. 2547	ศึกษาต่อระดับปริญญามหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร
ประวัติการทำงาน	
พ.ศ. 2549-ปัจจุบัน	เจ้าหน้าที่บริหารงานทั่วไป ศูนย์คอมพิวเตอร์ มหาวิทยาลัยราชภัฏนครปฐม