# การเพิ่มประสิทธิภาพกระบวนการเปลี่ยนถ่ายจาก IPv4 ไปสู่ IPv6
# Enhancement of IPv4/IPv6 Migration Mechanisms

ณ ภัทร ช่วงชุณห์ส่อง
Napat Chuangchunsong

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Computer Engineering
Prince of Songkla University

2558

**ชื่อวิทยานิพนธ์** การเพิ่มประสิทธิภาพกระบวนการเปลี่ยนถ่ายจาก IPv4 ไปสู่ IPv6
**ผู้เขียน** นายณ ภัทร ช่วงชุณห์ส่อง
**สาขาวิชา** วิศวกรรมคอมพิวเตอร์

---

**อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก**

.....
(รองศาสตราจารย์ทศพร กมลภิวงศ์)

**คณะกรรมการสอบ**

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.นิษฐิดา เอลซ์)

.....กรรมการ
(รองศาสตราจารย์ทศพร กมลภิวงศ์)

.....กรรมการ
(รองศาสตราจารย์ ดร.สินชัย กมลภิวงศ์)

.....กรรมการ
(ดร.เฉลิมพล ชาญศรีภิญโญ)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....
(รองศาสตราจารย์ ดร.ธีระพล ศรีชนะ)
คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ.....
(รองศาสตราจารย์ทศพร กมลภิวงศ์)
อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ.....
(นายณ ภัทร ช่วงชุณห์ส่อง)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....
(นายณ ภัทร ช่วงชุณห์ส่อง)
นักศึกษา

**ชื่อวิทยานิพนธ์** การเพิ่มประสิทธิภาพกระบวนการเปลี่ยนถ่ายจาก IPv4 ไปสู่ IPv6
**ผู้เขียน** นายณ ภัทร ช่วงชุณห์ส่อง
**สาขาวิชา** วิศวกรรมคอมพิวเตอร์
**ปีการศึกษา** 2557

## บทคัดย่อ

แม้ว่ากระบวนการเปลี่ยนถ่ายจะมีวัตถุประสงค์หลักเพื่อให้บริการ IPv4 และ IPv6 ควบคู่ไปด้วยกันในระหว่างการเปลี่ยนแปลงจากการใช้งานอินเทอร์เน็ตด้วย IPv4 ให้เป็น IPv6 แต่กระบวนการเปลี่ยนถ่ายแต่ละกระบวนการยังคงมีหลักการทำงานและจุดมุ่งหมายรองที่แตกต่างกันออกไป งานวิจัยชิ้นนี้มีวัตถุประสงค์เพื่อนำเสนอกระบวนการเปลี่ยนถ่ายที่มีประสิทธิภาพและความยืดหยุ่นในการใช้งาน อีกทั้งยังสามารถให้บริการได้จนกระทั่งยกเลิกการใช้งาน IPv4 ในที่สุด กระบวนการเปลี่ยนถ่ายที่นำเสนอเรียกว่า “Enhancement of Lightweight 4over6” ซึ่งเป็นกระบวนการที่พัฒนาต่อยอดจาก Lightweight 4over6 โดยปรับปรุงให้รองรับการระบุอุโมงค์สื่อสารไปยังปลายทางภายในเครือข่ายได้โดยตรง เพื่อลดความคับคั่งของข้อมูลในเส้นทางหลัก งานวิจัยได้ทดสอบประสิทธิภาพของกระบวนการเปลี่ยนถ่ายที่นำเสนอเปรียบเทียบกับกระบวนการเปลี่ยนถ่ายซึ่งประกอบไปด้วย 4over6, 4rd และ Lightweight 4over6 ด้วยโปรแกรมจำลองเครือข่าย OPNET Modeler 16.0 บนเครือข่ายที่จำลองจาก UniNet ผลการทดลองพบว่ากระบวนการที่นำเสนอมีประสิทธิภาพการเชื่อมต่อภายในเครือข่ายสูงกว่า Lightweight 4over6 อย่างเห็นได้ชัด และมีประสิทธิภาพสูงกว่า 4over6 และ 4rd เล็กน้อย เนื่องจากทั้งสองกระบวนการสามารถระบุอุโมงค์สื่อสารไปยังปลายทางภายในเครือข่ายได้โดยตรงเช่นกัน อย่างไรก็ตามเมื่อนำกระบวนการที่นำเสนอมาเปรียบเทียบประสิทธิภาพการเชื่อมต่อภายนอกเครือข่ายสังเกตได้ว่ากระบวนการที่นำเสนอมีประสิทธิภาพลดลงเพียงเล็กน้อยเมื่อเปรียบเทียบกับ Lightweight 4over6 เนื่องจากขั้นตอนการคัดกรองแพ็กเก็ต IPv4 บนอุปกรณ์ฝั่งผู้ใช้งานที่เพิ่มขึ้น แต่การประมวลผลดังกล่าวเป็นการประมวลผลบนอุปกรณ์ฝั่งผู้ใช้งานจึงไม่ส่งผลกระทบมากนัก แม้ว่าแพ็กเก็ตในระบบเครือข่ายจะเพิ่มสูงขึ้น

**Thesis Title** Enhancement of IPv4/IPv6 Migration Mechanisms
**Author** Mr.Napat Chuangchunsong
**Major Program** Computer Engineering
**Academic Year** 2014

## ABSTRACT

Although major aim of transitions is intended to provide both IPv4 and IPv6 in parallel during the changing process from IPv4 to IPv6 completely, minor aims of transitions still have the differences. This research aims to propose the enhancement of transition that has more performance and flexibility. It can also provide internet connectivity until disable IPv4 eventually. The proposed transition is called "Enhancement of Lightweight 4over6" which is improved from Lightweight 4over6 by creating direct tunnels to destinations inside transition domain to avoid critical paths. The research compares performance of the proposed transition with current transitions, which consist of 4over6, 4rd and Lightweight 4over6 on UniNet's network by using network simulator (OPNET Modeler 16.0). Based on simulation results, proposed method has higher performance in terms of intra-communication than lw4over6 significantly and higher performance than 4over6 and 4rd slightly because both transitions support mesh connectivity as proposed method. For inter-communication, performance of proposed method is lower than lw4over6 slightly. The reduced performance is affected by IPv4 packet filtering on customer equipment. However, the IPv4 packet filtering in proposed method does not operate on provider equipment. It also reduces little performance.

# กิตติกรรมประกาศ

ขอขอบพระคุณรองศาสตราจารย์ทศพร กมลภิวงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก และรองศาสตราจารย์ ดร.สินชัย กมลภิวงศ์ ที่ได้กรุณาให้คำปรึกษา คำแนะนำ และให้แนวคิดในการทำวิจัย ตลอดจนช่วยตรวจสอบและแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.นิษฐิดา เอลซ์ ประธานกรรมการสอบวิทยานิพนธ์ และดร.เฉลิมพล ชาญศรีภิญโญ กรรมการสอบวิทยานิพนธ์ (ผู้ทรงคุณวุฒิ) ที่ได้กรุณาให้คำแนะนำที่เป็นประโยชน์ในการปรับปรุงวิทยานิพนธ์ให้สมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณอาจารย์ธัชชัย เอ้งฉ้วน และอาจารย์โรเบิร์ต เอลซ์ ที่ได้กรุณาให้คำปรึกษา ชี้แนะแนวทาง ตลอดจนให้ความรู้ต่างๆ ในการดำเนินการวิจัยที่เป็นประโยชน์มาโดยตลอด

ขอขอบพระคุณ คณาจารย์ และบุคลากรในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ให้ความรู้และให้ความช่วยเหลือระหว่างการดำเนินการวิจัย

ขอขอบพระคุณ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ที่ได้สนับสนุนทุนการศึกษาในระดับอุดมศึกษา และระดับบัณฑิตศึกษา ตลอดจนสนับสนุนทุนในการเผยแพร่ผลงานวิจัยแก่ข้าพเจ้า

ขอขอบคุณเพื่อนนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ทุกคนที่ได้ช่วยเหลือ และเป็นกำลังใจในการทำงานเป็นอย่างดีเสมอมา

ณ ภัทร ช่วงชุณห์ส่อง

# สารบัญ

หน้า

# สารบัญ (ต่อ)

หน้า

# รายการภาพประกอบ

หน้า

# รายการภาพประกอบ (ต่อ)

หน้า

# รายการภาพประกอบ (ต่อ)

หน้า

# รายการตาราง

หน้า

# บทที่ 1
# ความสำคัญและที่มาของหัวข้อวิจัย

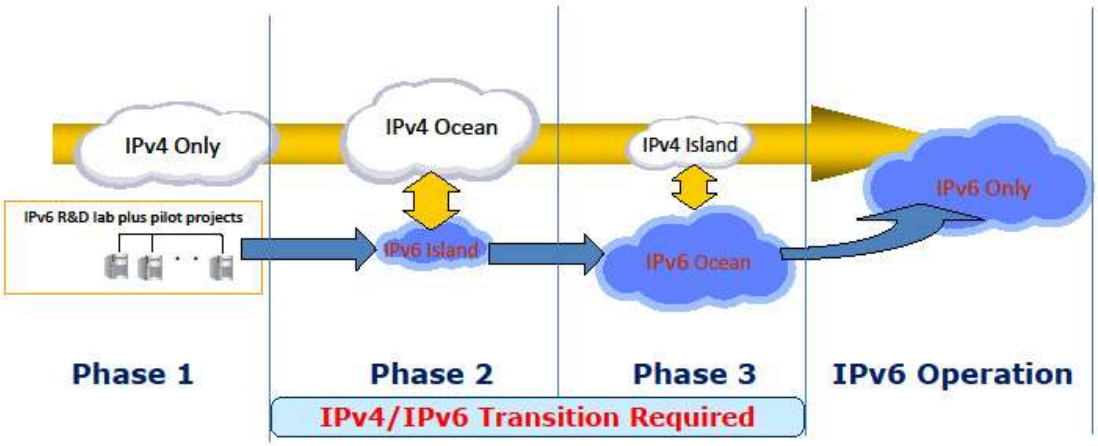
## 1.1 ความสำคัญและที่มาของหัวข้อวิจัย

จากปัญหาความขาดแคลนหมายเลขโพรโตคอลอินเทอร์เน็ตรุ่นที่ 4 (Internet Protocol version 4 หรือ IPv4) คณะทำงานเฉพาะกิจด้านวิศวกรรมอินเทอร์เน็ต (Internet Engineering Task Force) ได้มีการพัฒนาโพรโตคอลอินเทอร์เน็ตรุ่นที่ 6 (Internet Protocol version 6 หรือ IPv6) ขึ้นเพื่อใช้งานทดแทน IPv4 ดังนั้น ระบบเครือข่ายต้องเปลี่ยนแปลงจาก IPv4 ไปสู่ IPv6 แต่การเปลี่ยนแปลงจาก IPv4 ไปเป็น IPv6 นั้นไม่สามารถทำได้โดยง่าย เนื่องจาก IPv6 ไม่สามารถติดต่อสื่อสารกับ IPv4 ได้โดยตรง คณะทำงานเฉพาะกิจด้านวิศวกรรมอินเทอร์เน็ตจึงพัฒนากระบวนการที่เรียกว่า “กระบวนการเปลี่ยนถ่าย” (Transition) เพื่อที่จะทำให้สามารถใช้งาน IPv4 และ IPv6 ร่วมกันได้ในระหว่างช่วงของการเปลี่ยนแปลงนี้

กระบวนการเปลี่ยนถ่ายมีด้วยกันหลายวิธีการ ซึ่งสามารถแบ่งตามเทคนิคหลักๆ ที่นำมาใช้ได้ 3 วิธีด้วยกันได้แก่ Dual Stack, Tunneling และ Translation [1] วิธีการ Dual Stack ช่วยให้อุปกรณ์ภายในเครือข่ายสามารถใช้งานทั้ง IPv4 และ IPv6 ควบคู่ไปด้วยกันแบบขนาน ดังนั้นอุปกรณ์สามารถเชื่อมต่อกับเครือข่าย IPv4 โดยใช้หมายเลข IPv4 และเชื่อมต่อกับเครือข่าย IPv6 โดยใช้หมายเลข IPv6 โดยทั้ง IPv4 และ IPv6 ถูกจัดสรรอย่างอิสระแต่ยังคงเชื่อมต่อบนโครงข่ายเดียวกัน วิธีการ Tunneling ใช้โพรโตคอลอินเทอร์เน็ตรุ่นใดรุ่นหนึ่งเป็นหลัก และสร้างการเชื่อมต่อของโพรโตคอลอินเทอร์เน็ตอีกรุ่นโดยผ่านเครือข่ายที่ยังไม่รองรับการใช้งานโพรโตคอลอินเทอร์เน็ตทั้งสองโดยใช้อุโมงค์สื่อสาร อุโมงค์สื่อสารห่อหุ้มแพ็กเก็ตของโพรโตคอลอินเทอร์เน็ตต้นฉบับซึ่งระบบเครือข่ายยังไม่รองรับลงในแพ็กเก็ตของโพรโตคอลอินเทอร์เน็ตอีกรุ่น เพื่อให้สามารถส่งแพ็กเก็ตผ่านเครือข่ายดังกล่าวไปยังปลายทางได้ และวิธีการ Translation ช่วยให้อุปกรณ์ภายในเครือข่ายที่รองรับเพียง IPv4 หรือ IPv6 สามารถเชื่อมต่อกับเครือข่ายที่รองรับโพรโตคอลอินเทอร์เน็ตอีกรุ่นได้ หากอุปกรณ์ภายในเครือข่ายต้องการติดต่อสื่อสารกับเครื่องปลายทางที่มีโพรโตคอลอินเทอร์เน็ตแตกต่างกัน แพ็กเก็ตจะถูกแปลงให้มีโพรโตคอลอินเทอร์เน็ตตรงกับเครื่องปลายทาง เพื่อให้สามารถติดต่อสื่อสารระหว่างกันได้

ความต้องการให้บริการอินเทอร์เน็ตผ่านกระบวนการเปลี่ยนถ่ายยังคงมีต่อไปจนกระทั่งสามารถเปลี่ยนแปลงการใช้งานจาก IPv4 ไปสู่ IPv6 อย่างสมบูรณ์ ช่วงเวลาในการเปลี่ยนแปลงการใช้งานนี้สามารถแบ่งออกเป็นระยะย่อยๆ 4 ระยะ ดังแสดงในรูปที่ 1-1 ระยะแรกมีเพียงการให้บริการอินเทอร์เน็ตผ่าน IPv4 เพียงโพรโตคอลเดียว (IPv4 only) ในระยะนี้ไม่ได้รับผลกระทบจากความขาดแคลนหมายเลข IPv4 มากนัก ระยะที่สองเป็นช่วงเริ่มต้นของการเปลี่ยนแปลงจึงมีเครือข่าย IPv6 จำนวนไม่มากนัก การใช้งาน IPv6 ในระยะนี้ให้บริการโดยใช้กระบวนการเปลี่ยนถ่ายเพื่อสร้างการเชื่อมต่อ IPv6 ผ่านเครือข่าย IPv4 (IPv4 ocean) เช่น 6to4 [2] และ 6rd [3] เป็นต้น ในระยะนี้ถูกมองเสมือนว่าเครือข่าย IPv6 นั้นเป็นเกาะที่ถูกล้อมรอบด้วยเครือข่าย IPv4 ระยะที่สามให้บริการอินเทอร์เน็ตโดยใช้กระบวนการเปลี่ยนถ่ายเพื่อสร้างการเชื่อมต่อ IPv4 ผ่านเครือข่าย IPv6 (IPv6 ocean) เนื่องจากอุปกรณ์เครือข่ายรองรับการใช้งาน IPv6

มากขึ้น เครือข่ายแกนหลักของ IPv6 มีขนาดใหญ่ขึ้น และได้รับผลกระทบจากความขาดแคลนหมายเลข IPv4 รุนแรงมากยิ่งขึ้น ดังนั้นผู้ให้บริการอินเทอร์เน็ตต้องการวิธีที่สามารถจัดสรรหมายเลข IPv4 ได้อย่างมีประสิทธิภาพสูงสุด ผู้ให้บริการอินเทอร์เน็ตต้องลดการใช้หมายเลข IPv4 ในส่วนที่ไม่จำเป็นให้ได้มากที่สุด เมื่อมองภาพรวมของระบบเครือข่ายในระยะนี้เครือข่าย IPv4 กลายเป็นเกาะซึ่งถูกล้อมรอบด้วยเครือข่าย IPv6 แทน ในระยะสุดท้ายของการเปลี่ยนแปลง ระบบเครือข่ายทั้งหมดใช้งานเพียง IPv6 เท่านั้น (IPv6 only) โดยไม่นำ IPv4 มาใช้งานอีกต่อไป




รูปที่ 1-1 ระยะการเปลี่ยนแปลงการใช้งานจาก IPv4 ไปสู่ IPv6

ที่มา : Evolution Towards IPv6 [4]

ในปัจจุบัน การเปลี่ยนแปลงจาก IPv4 ไปสู่ IPv6 เริ่มเข้าสู่การเปลี่ยนแปลงในระยะที่ 3 IPv6 ocean เนื่องจากองค์การกำหนดหมายเลขอินเทอร์เน็ต (Internet Assigned Numbers Authority) ได้จัดสรร IPv4 address ชุดสุดท้ายออกไป [5] ส่งผลให้การขยายการให้บริการอินเทอร์เน็ตต่อจากนี้จำเป็นต้องขยายการให้บริการด้วย IPv6 เป็นหลัก อย่างไรก็ตามผู้ใช้งานยังคงต้องการติดต่อสื่อสารกับเครือข่าย IPv4 อีกเป็นจำนวนมาก เนื่องจากสัดส่วนปริมาณข้อมูลของ IPv6 เทียบกับปริมาณข้อมูลทั้งหมดในเดือนเมษายน ปีพ.ศ. 2557 มีเพียง 2.80% เท่านั้น และเมื่อพิจารณาปริมาณการใช้งาน IPv6 ของแต่ละประเทศพบว่า มีเพียงไม่กี่ประเทศเท่านั้นที่มีปริมาณ IPv6 เพิ่มขึ้นอย่างมีนัยสำคัญ เช่น เบลเยียม เยอรมนี และสหรัฐอเมริกา [6] ยิ่งกว่านั้น นักวิเคราะห์มองว่า IPv4 และ IPv6 ยังคงมีการใช้งานร่วมเป็นระยะเวลานาน ด้วยสาเหตุสำคัญสองประการ ประการแรกคือ IPv4 และ IPv6 ใช้งานร่วมกันบนโครงข่ายเดียวกันได้ผ่านกระบวนการเปลี่ยนถ่ายต่างๆ และอีกสาเหตุหนึ่งคือปริมาณข้อมูลการใช้งาน IPv4 ยังคงมีอยู่เป็นจำนวนไม่น้อยในปัจจุบัน ส่งผลให้การเปลี่ยนจากการใช้งาน IPv4 เป็น IPv6 ต้องเปลี่ยนแปลงแบบค่อยเป็นค่อยไป [7] จากข้อมูลดังกล่าวแสดงให้เห็นว่า ความต้องการใช้งาน IPv4 ไม่ได้ลดลงจากเดิมมากนัก แม้ว่าองค์การกำหนดหมายเลขอินเทอร์เน็ตจะไม่สามารถจัดสรรหมายเลข IPv4 เพิ่มเติมได้แล้วก็ตาม

ยิ่งกว่านั้น การใช้งาน IPv4 ยังคงมีความต้องการจากองค์กรขนาดใหญ่ ซึ่งมีสาขาจำนวนมาก เนื่องจากองค์กรเหล่านี้อาจมีการพัฒนาโปรแกรมประยุกต์เพื่อใช้งานขึ้นเองโดยเฉพาะหรือจัดซื้อโปรแกรมประยุกต์ตามท้องตลาดมาใช้งาน โดยโปรแกรมประยุกต์เหล่านี้อาจยังไม่รองรับ

การใช้งาน IPv6 ในขณะนั้น หากต้องการให้โปรแกรมประยุกต์เหล่านี้สามารถใช้งานได้ร่วมกับ IPv6 โปรแกรมประยุกต์ต้องปรับปรุงให้รองรับการใช้งาน IPv6 ในกรณีที่ไม่สามารถปรับปรุงให้รองรับการใช้งาน IPv6 องค์กรเหล่านี้ต้องจัดหาโปรแกรมประยุกต์ใหม่เพื่อนำมาใช้งานทดแทน ส่งผลให้ต้องใช้งบประมาณเพิ่มขึ้นจำนวนมาก แต่หากผู้ให้บริการอินเทอร์เน็ตขององค์กรเหล่านี้ เลือกที่จะให้บริการ IPv4 และ IPv6 ควบคู่ไปด้วยกันจนกว่าจะเลิกใช้งาน IPv4 อย่างสมบูรณ์ ผู้ให้บริการอินเทอร์เน็ตสามารถช่วยยืดเวลาสำหรับปัญหานี้ออกไป เพื่อให้ลูกค้ามีเวลารับมือกับปัญหามากยิ่งขึ้น แต่ผู้ให้บริการอินเทอร์เน็ตต้องรับภาระในการให้บริการ IPv4 และ IPv6 ควบคู่ไปด้วยกันบนโครงข่ายที่มีอยู่เดิม ยิ่งกว่านั้นผู้ให้บริการต้องวางแผนเพื่อรองรับการขยายตัวของโครงข่ายเพิ่มเติมในอนาคตภายใต้จำนวนหมายเลข IPv4 ที่มีอยู่อย่างจำกัดซึ่งถือว่าเป็นงานที่ไม่ง่ายนัก

ด้วยเหตุนี้ ผู้ให้บริการอินเทอร์เน็ตจึงเริ่มให้ความสำคัญกับการให้บริการ IPv4 ผ่านกระบวนการเปลี่ยนถ่ายซึ่งสามารถใช้งานทั้ง IPv4 และ IPv6 ควบคู่ไปด้วยกัน โดยกระบวนการเปลี่ยนถ่ายดังกล่าวต้องจัดสรรหมายเลข IPv4 ที่เหลืออยู่ได้อย่างมีประสิทธิภาพ และลดการใช้งาน IPv4 ในส่วนที่ไม่จำเป็นออกไปให้ได้มากที่สุด ยิ่งกว่านั้นกระบวนการเปลี่ยนถ่ายในระยะที่ 3 IPv6 ocean ถือเป็นช่วงที่ผู้ให้บริการต้องให้ความสำคัญมากที่สุด เนื่องจากการเปลี่ยนแปลงในระยะนี้ใช้ระยะเวลามากที่สุด อีกทั้งยังเป็นช่วงสุดท้ายก่อนที่จะยกเลิกการใช้งาน IPv4 อย่างสมบูรณ์ ในกรณีที่ผู้ให้บริการที่มีหมายเลข IPv4 ไม่เพียงพอต่อการให้บริการอินเทอร์เน็ตด้วยวิธีการ Dual-Stack ผู้ให้บริการจำเป็นต้องให้บริการอินเทอร์เน็ตด้วยกระบวนการเปลี่ยนถ่ายอย่างหลีกเลี่ยงไม่ได้ ซึ่งการตัดสินใจเลือกกระบวนการเปลี่ยนถ่ายอย่างเหมาะสมนั้นไม่สามารถทำได้โดยง่าย เพราะกระบวนการเปลี่ยนถ่ายในปัจจุบันมีหลากหลายกระบวนการด้วยกัน ซึ่งแต่ละกระบวนการก็มีวิธีการดำเนินงานที่แตกต่างกันออกไป

แม้ว่าในปัจจุบันมีงานวิจัยที่นำกระบวนการเปลี่ยนถ่ายต่างๆ มาวิเคราะห์จุดเด่นและข้อจำกัดอยู่บ้าง แต่ก็ยังไม่ครอบคลุมกระบวนการเปลี่ยนถ่ายที่ถูกนำเสนอใหม่เท่าที่ควร ดังนั้นเพื่อให้ได้รับผลการวิเคราะห์ที่ครอบคลุมและสามารถนำไปใช้ในปรับปรุงกระบวนการเปลี่ยนถ่ายให้มีประสิทธิภาพมากยิ่งขึ้น กระบวนการเปลี่ยนถ่ายที่นำมาเปรียบเทียบควรมีช่วงระยะเวลาในการเปลี่ยนแปลงที่ยาวนาน และครอบคลุมไปถึงกระบวนการเปลี่ยนถ่ายที่มีหลักการทำงานใกล้เคียงกัน ด้วยเหตุนี้ ในวิทยานิพนธ์ฉบับนี้จึงนำเสนอผลการวิเคราะห์ และวิธีการเพิ่มประสิทธิภาพกระบวนการเปลี่ยนถ่ายที่อยู่ในช่วงการเปลี่ยนแปลงการใช้งานจาก IPv4 ไปสู่ IPv6 ในช่วงระยะที่ 3 IPv6 ocean กระบวนการเปลี่ยนถ่ายที่นำมาวิเคราะห์ใช้วิธีการ Tunneling เพื่อสร้างการเชื่อมต่อ IPv4 ผ่านเครือข่าย IPv6 นอกจากนี้วิทยานิพนธ์ฉบับนี้ได้จำลองอุปกรณ์เครือข่ายซึ่งรองรับการดำเนินการของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการเพื่อนำผลลัพธ์ที่ได้จากการจำลองมาเปรียบเทียบ และวิเคราะห์ผลที่เกิดขึ้นบนเครือข่ายอย่างละเอียด ซึ่งช่วยให้สามารถสรุปจุดเด่นและข้อจำกัดของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการได้อย่างครบถ้วน จากผลของการจำลองสามารถหาข้อสรุปเพื่อใช้ในการตัดสินใจเลือกกระบวนการเปลี่ยนถ่ายสำหรับให้บริการอินเทอร์เน็ตสำหรับระบบเครือข่ายแต่ละเครือข่ายได้อย่างเหมาะสม ยิ่งกว่านั้นยังสามารถนำข้อสรุปดังกล่าวมาใช้สำหรับพัฒนากระบวนการเปลี่ยนถ่ายที่มีอยู่เดิมให้มีประสิทธิภาพมากยิ่งขึ้น และสามารถก้าวข้ามข้อจำกัดที่มีอยู่เดิมได้

## 1.2 วัตถุประสงค์ของโครงการ

1 เพื่อวิเคราะห์ประสิทธิภาพ และเปรียบเทียบข้อดีข้อเสียของระบบเครือข่ายที่มีการให้บริการ IPv4 ด้วยกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 โดยใช้การจำลองระบบเครือข่าย

2 เพื่อวิเคราะห์รูปแบบการให้บริการ IPv4 ที่เหมาะสมกับกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 แต่ละกระบวนการ

3 เพื่อพัฒนากระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 รูปแบบใหม่ซึ่งมีประสิทธิภาพในการเชื่อมต่อสูง และสามารถลดผลกระทบจากปัญหาความขาดแคลนหมายเลข IPv4 ในอนาคตลงได้

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

1 เป็นข้อมูลในการวางแผนการให้บริการ IPv4 ด้วยกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6

2 ได้รับกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 ใหม่ที่มีประสิทธิภาพสูงและสามารถลดผลกระทบจากปัญหาความขาดแคลนหมายเลข IPv4 ในอนาคตลงได้

## 1.4 ขอบเขตของการวิจัย

1 วิเคราะห์ประสิทธิภาพของกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6แต่ละกระบวนการ โดยใช้แบบจำลองระบบเครือข่ายด้วยโปรแกรม OPNET Modeler

2 พัฒนากระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 ที่มีประสิทธิภาพสูง และสามารถลดผลกระทบจากปัญหาความขาดแคลนหมายเลข IPv4 ในอนาคตลงได้

# บทที่ 2
# ทฤษฎีและหลักการ

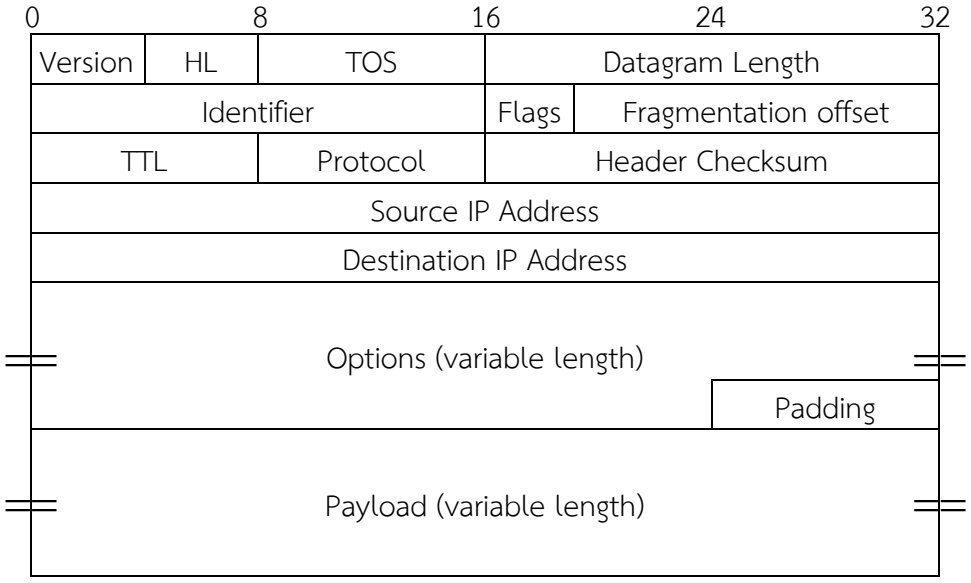
บทนี้เป็นการกล่าวถึงทฤษฎีและหลักการของกระบวนการเปลี่ยนถ่ายที่เกี่ยวข้องในวิทยานิพนธ์ โดยเนื้อหาจะกล่าวถึงที่มาและรายละเอียดของ IPv4 และ IPv6 จากนั้นนำเข้าสู่รายละเอียดในการทำงานของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการ ท้ายที่สุดเป็นการทบทวนวรรณกรรมที่เกี่ยวข้องเพิ่มเติมเพื่อให้เนื้อหาของทฤษฎีและหลักการมีความครบถ้วนสมบูรณ์

## 2.1 โพรโตคอลอินเทอร์เน็ต (Internet Protocol)

โพรโตคอลอินเทอร์เน็ตเป็นโพรโตคอลสำคัญสำหรับใช้ในการติดต่อสื่อสารในชั้นที่ 3 ของ OSI model (network-layer) ฟิลด์ข้อมูลสำคัญของโพรโตคอลอินเทอร์เน็ตคือหมายเลขโพรโตคอลอินเทอร์เน็ต (IP address) ซึ่งใช้สำหรับระบุตัวตนสำหรับอุปกรณ์ต่างๆ ในเครือข่าย หมายเลขโพรโตคอลช่วยให้เราเตอร์สามารถส่งต่อแพ็กเก็ตไปยังเครื่องปลายทางได้อย่างถูกต้อง โพรโตคอลอินเทอร์เน็ตที่แพร่หลายกันในปัจจุบันมีด้วยกันสองรุ่น ได้แก่ IPv4 และ IPv6

### 2.1.1 โพรโตคอลอินเทอร์เน็ตรุ่นที่ 4 (IPv4)

IPv4 เป็นโพรโตคอลอินเทอร์เน็ตที่ถูกออกแบบและนำมาใช้งานตั้งแต่ยุคเริ่มต้นของอินเทอร์เน็ต ฟิลด์ข้อมูลที่ใช้สำหรับระบุหมายเลข IPv4 มีขนาด 32 บิต หรือคิดเป็นจำนวน 4 ไบต์ จำนวนหมายเลข IPv4 ที่สามารถกำหนดได้ทั้งหมดเท่ากับ $2^{32}$ เพื่อให้ง่ายต่อการอ่านหมายเลข IPv4 ของมนุษย์ IPv4 สามารถเขียนแต่ละไบต์ของหมายเลข IPv4 จากเลขฐานสองเป็นเลขฐานสิบ




รูปที่ 2-1 รูปแบบฟิลด์ข้อมูลของโพรโตคอล IPv4

โดยแต่ละไบต์ถูกเขียนคั่นด้วยสัญลักษณ์จุด (dot) การเขียนเช่นนี้เรียกว่า “dotted-decimal notation” ในขณะที่ IPv4 ถูกพัฒนาขึ้นมานั้น เทคโนโลยีด้านการส่งข้อมูลยังไม่มีประสิทธิภาพมากนัก IPv4 จึงให้ความสำคัญกับการตรวจสอบความถูกต้องของแพ็กเก็ตเพื่อเลือกส่งต่อเพียงแพ็กเก็ตที่ปราศจากข้อผิดพลาดเท่านั้น และอนุญาตให้แพ็กเก็ตถูกแบ่งออกเป็นแพ็กเก็ตย่อย (fragmentation) โดยเราเตอร์ที่อยู่ระหว่างทางเพื่อให้สามารถส่งข้อมูลไปยังปลายทางผ่านลิงค์ที่รองรับขนาดของแพ็กเก็ตสูงสุดที่ต่ำกว่าได้ โดยไม่จำเป็นต้องเริ่มต้นส่งใหม่ หลังจากนั้นแพ็กเก็ตย่อยจะถูกรวมกลับเป็นแพ็กเก็ตต้นฉบับโดยเครื่องปลายทาง รูปแบบของฟิลด์ข้อมูลของ IPv4 แสดงรายละเอียดดังรูปที่ 2-1

จากรูปที่ 2-1 แสดงรายละเอียดฟิลด์ข้อมูลทั้งหมดของ IPv4 ซึ่งฟิลด์ข้อมูลดังกล่าวมีหน้าที่ดังต่อไปนี้

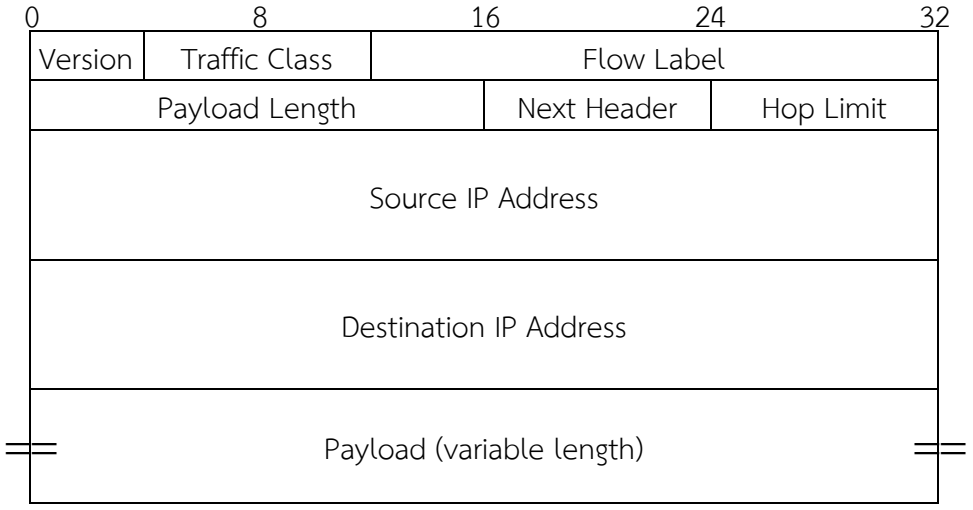
- Version number: มีขนาด 4 บิต และถูกใช้สำหรับระบุรุ่นของโพรโตคอลอินเทอร์เน็ต โดยโพรโตคอลอินเทอร์เน็ตแต่ละรุ่นสามารถกำหนดรูปแบบและขนาดของฟิลด์ข้อมูลที่แตกต่างกันได้ซึ่งเป็นไปตามการนิยามของแต่ละรุ่น เราเตอร์ต้องใช้ข้อมูล version number เพื่อกำหนดรูปแบบการอ่านฟิลด์ข้อมูลส่วนที่เหลือต่อไป
- Header Length: มีขนาด 4 บิต และถูกใช้สำหรับระบุความยาวของ header เนื่องจากความยาวของ header มีขนาดเพิ่มขึ้นตามขนาดของ option header length มีค่าเท่ากับความยาวของ header ในหน่วยบิตเป็นตัวตั้งแล้วหารด้วย 32 ดังนั้นในกรณีที่ header ไม่มี option เพิ่มเติม header length จะมีค่าน้อยที่สุดเท่ากับ 160/32 = 5
- Type of service (TOS): มีขนาด 8 บิต และถูกใช้สำหรับระบุคุณสมบัติที่ต้องการสำหรับการส่งข้อมูลโดยโปรแกรมประยุกต์ ยกตัวอย่างเช่น กำหนดให้ข้อมูลที่ส่งมีค่าหน่วงเวลาในการส่งข้อมูลต่ำ หรือความน่าเชื่อถือสูง เป็นต้น เพื่อให้ได้รับคุณภาพของข้อมูลตรงตามความต้องการในการส่งข้อมูลของโปรแกรมประยุกต์
- Datagram Length: มีขนาด 16 บิต และถูกใช้สำหรับระบุความยาวทั้งหมดของแพ็กเก็ต IPv4 ในหน่วยไบต์ ดังนั้นความยาวสูงสุดของแพ็กเก็ต IPv4 สูงสุดตามทฤษฎีต้องมีค่าไม่เกิน 65,535 ไบต์ อย่างไรก็ตามโดยทั่วไปความยาวสูงสุดของแพ็กเก็ต IPv4 มีค่าไม่เกิน 1,500 ไบต์ และไม่ต่ำกว่า 576 ไบต์
- Identifier, Flags, Fragmentation offset: มีขนาด 16, 3 และ 13 บิต ตามลำดับ โดยฟิลด์ข้อมูลทั้งสามถูกใช้สำหรับการแบ่งแพ็กเก็ตออกเป็นแพ็กเก็ตย่อย ดังนั้น เราเตอร์สามารถลดขนาดของแพ็กเก็ตลง เพื่อให้แพ็กเก็ตมีความยาวไม่เกินความยาวสูงสุดของแพ็กเก็ตในลิงค์ที่ต้องการส่งต่อข้อมูล

- Time-to-live (TTL): มีขนาด 8 บิต และถูกใช้เพื่อป้องกันไม่ให้แพ็กเก็ตถูกส่งต่ออย่างไม่รู้จบ โดยทุกครั้งที่เราเตอร์ส่งต่อแพ็กเก็ต ค่า time-to-live จะถูกลดลงครั้งละ 1 หากค่า time-to-live ของแพ็กเก็ตใดถูกลดลงเท่ากับศูนย์ แพ็กเก็ตดังกล่าวจะถูกละทิ้งในที่สุด
- Protocol: มีขนาด 8 บิต และถูกใช้สำหรับระบุว่าข้อมูลของแพ็กเก็ตดังกล่าวเป็นข้อมูลของโพรโตคอลใดใน transport-layer ยกตัวอย่างเช่น หาก protocol มีค่าเท่ากับ 6 ข้อมูลของแพ็กเก็ตจะถูกส่งต่อให้กับ Transmission Control Protocol (TCP) หรือหาก protocol มีค่าเท่ากับ 17 ข้อมูลของแพ็กเก็ตจะถูกส่งต่อให้กับ User Datagram Protocol (UDP) เป็นต้น
- Header Checksum: มีขนาด 16 บิต และถูกใช้สำหรับตรวจสอบความถูกต้องของข้อมูลภายในส่วนของ header โดย header checksum คำนวณโดยแบ่งข้อมูลของ header ออกเป็นชุดๆ แต่ละชุดมีขนาด 16 บิตเท่ากับขนาดของ header checksum จากนั้นบวกข้อมูลแต่ละชุดเข้าด้วยกัน แล้วนำผลลัพธ์ที่ได้ไปดำเนินการ 1’ complement เมื่อเราเตอร์ได้รับแพ็กเก็ตใหม่เข้ามา เราเตอร์ต้องตรวจสอบความถูกต้องของ header โดยบวกชุดข้อมูลทั้งหมดเข้าด้วยกันซึ่งรวมถึง header checksum หากผลลัพธ์ที่ได้มีค่าเท่ากับ $1111111111111111_2$ แสดงว่าข้อมูลของ header ไม่ถูกเปลี่ยนแปลงระหว่างการส่ง แต่หากข้อมูลของ header เกิดการเปลี่ยนแปลงระหว่างการส่ง เราเตอร์ก็จะละทิ้งแพ็กเก็ตดังกล่าว ดังนั้นเราเตอร์ต้องคำนวณ header checksum ใหม่ทุกครั้ง เมื่อค่า time-to-live หรือค่าอื่นๆ ภายใน header เกิดการเปลี่ยนแปลง
- Source, Destination IP address: หมายเลขโพรโตคอลอินเทอร์เน็ตของ IPv4 มีขนาด 32 บิต และถูกใช้ในการระบุตัวตนสำหรับอุปกรณ์ภายในระบบเครือข่าย โดยหมายเลข IPv4 ที่ถูกกำหนดให้กับแต่ละอุปกรณ์ต้องไม่ซ้ำซ้อนกันเพื่อให้สามารถระบุตัวตนได้อย่างถูกต้อง source IP address ถูกใช้สำหรับระบุเครื่องต้นทางซึ่งทำหน้าที่เป็นผู้ส่งและ destination IP address ถูกใช้สำหรับระบุเครื่องปลายทางซึ่งทำหน้าที่เป็นผู้รับ
- Options: มีขนาดตั้งแต่ 0 จนถึง 320 บิตและต้องหารด้วย 32 ลงตัว options ถูกออกแบบสำหรับเป็นส่วนขยายเพื่อให้ IPv4 สามารถประยุกต์ใช้คุณสมบัติพิเศษเพิ่มเติมในอนาคต
- Payload: มีค่าเท่ากับ total length ลบด้วย header length โดย payload ถูกใช้สำหรับบรรจุข้อมูลของ transport-layer ที่ต้องการส่งไปยังเครื่องปลายทาง

### 2.1.2 โพรโตคอลอินเทอร์เน็ตรุ่นที่ 6 (IPv6)

IPv6 เป็นโพรโตคอลอินเทอร์เน็ตที่ถูกพัฒนาขึ้นเพื่อนำมาใช้งานแทน IPv4 แม้ว่า IPv4 สามารถใช้งานได้เป็นอย่างดี แต่จำนวนหมายเลข IPv4 ไม่เพียงพอต่อความต้องการใช้งานอินเทอร์เน็ตที่เพิ่มขึ้นอย่างรวดเร็วในปัจจุบัน ดังนั้นวัตถุประสงค์หลักในการพัฒนา IPv6 คือการออกแบบให้หมายเลข IPv6 เพียงพอต่อความต้องการในปัจจุบันและในอนาคต หมายเลข IPv6 ใช้เลขฐานสองจำนวน 128 บิตในการระบุตัวตน ส่งผลให้ IPv6 สามารถแก้ไขปัญหาความขาดแคลนที่เกิดขึ้นกับ IPv4 ได้

ยิ่งกว่านั้น IPv6 ยังถูกออกแบบให้มีความเหมาะสมกับเทคโนโลยีเครือข่ายที่เปลี่ยนไปเพื่อให้ได้รับประสิทธิภาพในการใช้งานสูงสุด ฟิลด์ข้อมูลของ IPv6 จึงปรับลดลงจาก IPv4 โดยสิ่งที่พัฒนาขึ้นมาอย่างเห็นได้ชัดคือ ลิงค์ของระบบเครือข่ายในปัจจุบันมีจำนวนมากขึ้นและรองรับอัตราการส่งข้อมูลสูงขึ้นต่างจากในอดีต การประมวลผลที่ล่าช้าส่งผลให้ไม่สามารถใช้งานลิงค์ได้อย่างเต็มประสิทธิภาพ IPv6 จึงลดการประมวลผลที่ไม่จำเป็นออกไปให้ได้มากที่สุด โดยคุณสมบัติเดิมของ IPv4 ที่ถูกตัดออกไปใน IPv6 ได้แก่ การตรวจสอบ header ของแพ็กเก็ต และการแบ่งแพ็กเก็ตออกเป็นแพ็กเก็ตย่อย สำหรับการตรวจสอบ header ของแพ็กเก็ต IPv6 อนุญาตให้ส่งต่อแพ็กเก็ตที่อาจเกิดการเปลี่ยนแปลงในระหว่างการส่งได้ เพราะอย่างไรก็ตามแพ็กเก็ตดังกล่าวก็จะถูกตรวจพบข้อผิดพลาดและถูกละทิ้งโดยเครื่องปลายทางอย่างแน่นอน การยกเลิกการตรวจสอบ header ส่งผลให้เราเตอร์มีการประมวลผลที่ลดลง ซึ่งช่วยให้มีความสามารถในการส่งข้อมูลสูงขึ้น และสำหรับการแบ่งแพ็กเก็ตเป็นแพ็กเก็ตย่อย IPv6 มองว่าเครื่องต้นทางและเครื่องปลายทางควรเป็นผู้ดำเนินการได้ตั้งแต่เริ่มต้นส่งข้อมูล โดยที่เราเตอร์ระหว่างทางไม่จำเป็นทำหน้าที่ลดขนาดของแพ็กเก็ต ดังนั้นฟิลด์ข้อมูลของ IPv4 ที่ไม่ปรากฏใน IPv6 ประกอบด้วย header checksum, Identifier, Flags และ Fragmentation offset นอกจากนี้ฟิลด์ข้อมูลที่ถูกตัดออกไปอีก 2 ฟิลด์ ได้แก่ Header length และ Options เนื่องจาก IPv6 กำหนดขนาดของ header ให้มีค่าคงที่เท่ากับ 40 ไบต์จึงไม่จำเป็นต้องระบุค่า header length ส่วนกรณีที่ IPv6 ต้องการใช้งาน option IPv6 สามารถระบุ option เพิ่มเติมด้วยฟิลด์ next header รูปแบบของฟิลด์ข้อมูลของ IPv6 มีรายละเอียดแสดงดังในรูปที่ 2-2




รูปที่ 2-2 รูปแบบฟิลด์ข้อมูลของโพรโตคอล IPv6

จากรูปที่ 2-2 แสดงรายละเอียดฟิลด์ข้อมูลทั้งหมดของ IPv6 ซึ่งฟิลด์ข้อมูลดังกล่าวมีหน้าที่ดังต่อไปนี้

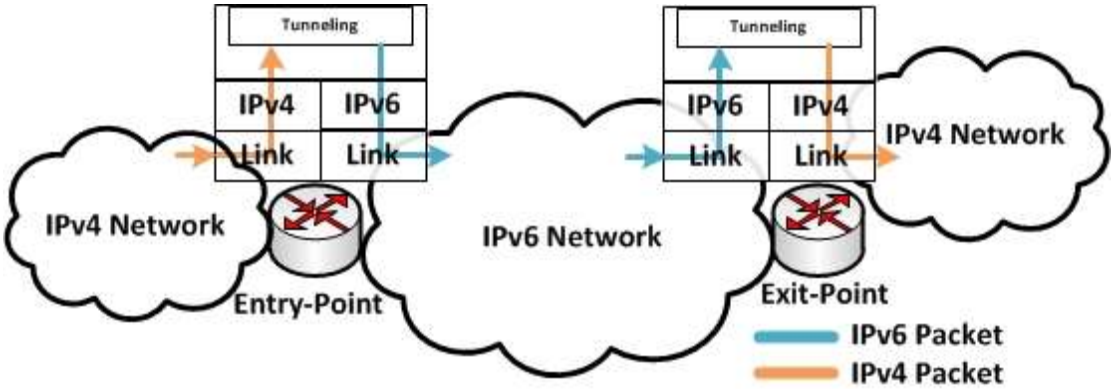
- Version number: มีขนาด 4 บิต และถูกใช้สำหรับระบุรุ่นของโพรโตคอลอินเทอร์เน็ตเช่นเดียวกับ IPv4
- Traffic class: มีขนาด 4 บิต และถูกใช้งานเช่นเดียวกับ TOS ใน IPv4
- Flow label: มีขนาด 20 บิต และถูกใช้สำหรับระบุรูปแบบในการส่งต่อแพ็กเก็ตของเราเตอร์ ซึ่งผู้ส่งมีความต้องการควบคุมการส่งเป็นพิเศษ
- Payload length: มีขนาด 16 บิต และเปลี่ยนแปลงจาก total length ของ IPv4 เนื่องจาก IPv6 มีขนาด header คงที่ ขนาดของ payload ก็ไม่จำเป็นต้องรวมความยาวของ header เหมือนกับ total length
- Next header: มีขนาด 8 บิต และถูกใช้สำหรับระบุว่า payload ในแพ็กเก็ตนี้เป็นของโพรโตคอลใดซึ่งเหมือนกับ protocol ใน IPv4 ยิ่งกว่านั้นค่าที่ใช้สำหรับระบุโพรโตคอลซึ่งเป็นเจ้าของ payload ใน next header ใช้ค่าเหมือนกับ protocol อีกด้วย
- Hop limit: มีขนาด 8 บิต และถูกใช้สำหรับกำหนดจำนวนครั้งที่สามารถส่งต่อแพ็กเก็ตได้เช่นเดียวกับ time-to-live ของ IPv4
- Source , Destination address: หมายเลขโพรโตคอลของ IPv6 มีขนาด 128 บิต และถูกใช้สำหรับระบุตัวตนของผู้ส่งและผู้รับ
- Payload: มีความยาวเท่ากับขนาดของ payload และทำหน้าที่บรรจุข้อมูลของโพรโตคอลตามที่ระบุไว้โดย next header

จากที่กล่าวมาเกี่ยวกับโพรโตคอลอินเทอร์เน็ต สรุปได้ว่าการปรับปรุง IPv6 จากข้อจำกัดของ IPv4 ส่งผลให้ทั้ง IPv4 และ IPv6 ไม่สามารถติดต่อสื่อสารระหว่างกันได้โดยตรง เครือข่าย IPv6 จึงกลายเป็นเหมือนเครือข่ายคู่ขนานกับเครือข่าย IPv4 หากเครือข่าย IPv4 ต้องการติดต่อกับเครือข่าย IPv6 เครือข่าย IPv4 จำเป็นต้องปรับปรุงระบบใหม่ให้รองรับ IPv6 ทดแทนส่งผลให้เกิดความยุ่งยากในการใช้งานอย่างมาก ยิ่งกว่านั้นการให้บริการและการใช้งานโปรแกรมประยุกต์ส่วนใหญ่ยังคงรองรับ IPv4 เป็นหลัก ส่งผลให้เปลี่ยนจาก IPv4 สู่ IPv6 จำเป็นต้องใช้เวลาค่อนข้างนาน ด้วยเหตุนี้คณะทำงานเฉพาะกิจด้านวิศวกรรมอินเทอร์เน็ตจึงพัฒนากระบวนการเปลี่ยนถ่ายเพื่อให้สามารถใช้งาน IPv4 และ IPv6 ร่วมกันได้ในระหว่างการเปลี่ยนแปลงที่เกิดขึ้น

## 2.2 วิธีการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 (IPv4-in-IPv6 tunneling)

วิธีการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 เป็นวิธีการที่นิยมนำมาใช้สำหรับสร้างการเชื่อมต่อ IPv4 ในกระบวนการเปลี่ยนถ่ายปัจจุบัน [8] อุโมงค์สื่อสารถูกสร้างขึ้นระหว่างโหนด IPv6 2 โหนดเพื่อส่งแพ็กเก็ต IPv4 เสมือนเป็น payload ของแพ็กเก็ต IPv6 โดยที่อุโมงค์สื่อสารดังกล่าวสามารถเรียกว่า “ลิงค์เสมือน” เนื่องจากเมื่อมองในมุมของ IPv4 แพ็กเก็ต IPv4 ถูกส่งผ่านอุโมงค์สื่อสารในเครือข่าย IPv6 โดยที่แพ็กเก็ต IPv4 ไม่ได้รับการเปลี่ยนแปลงแต่

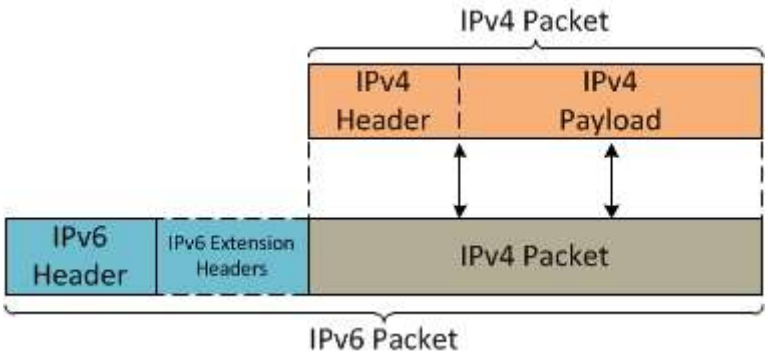
อย่างใดเหมือนการส่งข้อมูลด้วย link-layer ในการส่งข้อมูลแต่ละครั้ง โหนด IPv6 มีหน้าที่ในการดำเนินการโดยเฉพาะ เมื่อโหนดหนึ่งทำหน้าที่ห่อหุ้มแพ็กเก็ต IPv4 ซึ่งรับมาจากเครื่องต้นทางภายในแพ็กเก็ต IPv6 ส่วนอีกโหนดหนึ่งทำหน้าที่นำแพ็กเก็ต IPv4 ดั้งเดิมภายในแพ็กเก็ต IPv6 ออกมาเพื่อนำแพ็กเก็ต IPv4 มาส่งต่อไปยังเครื่องปลายทางต่อไป โหนดที่ทำหน้าที่ห่อหุ้มแพ็กเก็ตถูกเรียกว่า “entry-point” ซึ่งเป็นอุโมงค์สื่อสารต้นทาง และโหนดที่ทำหน้าที่นำแพ็กเก็ตดั้งเดิมที่ถูกห่อหุ้มออกมถูกเรียกว่า “exit-point” ซึ่งเป็นอุโมงค์สื่อสารปลายทาง วิธีการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 แสดงดังรูปที่ 2-3




รูปที่ 2-3 วิธีการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6

แพ็กเก็ต IPv6 ที่ใช้ในการห่อหุ้มแบ่งออกเป็น 3 ส่วน ได้แก่ IPv6 header, IPv6 extension header และ payload (ซึ่งในกรณีนี้ใช้บรรจุแพ็กเก็ต IPv4) โดยแพ็กเก็ต IPv6 ที่ใช้ในการห่อหุ้มมีรายละเอียดดังรูปที่ 2-4 ในการห่อหุ้มแพ็กเก็ต IPv4 ภายในแพ็กเก็ต IPv6 อุโมงค์สื่อสารต้นทางดำเนินการโดยเพิ่ม IPv6 header, IPv6 extension header (อาจมีหรือไม่มีก็ได้) และบรรจุแพ็กเก็ต IPv4 เสมือนเป็น payload ของแพ็กเก็ต IPv6 ซึ่งการกำหนดฟิลด์ข้อมูล IPv6 header ในการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 โดยทั่วไปมีรายละเอียดดังต่อไปนี้

- Version number: กำหนดค่าเท่ากับ 6
- Traffic class: กำหนดตามค่า traffic class ของแพ็กเก็ต IPv4 หรือถูกกำหนดเป็นค่า traffic class สำหรับอุโมงค์สื่อสาร
- Flow label: กำหนดค่าตามที่อุโมงค์สื่อสารต้นทางต้องการ โดยทั่วไปถูกกำหนดค่าเป็น 0
- Payload length: กำหนดค่าเท่ากับขนาดของแพ็กเก็ต IPv4 รวมกับขนาดของ IPv6 extension header
- Hop limit: กำหนดค่าเท่ากับ time-to-live ของแพ็กเก็ต IPv4 หรือถูกกำหนดเท่ากับค่า hop limit ตามที่ได้รับจาก Neighbor Discovery ซึ่งมีค่าเท่ากับ 64




รูปที่ 2-4 แพ็กเก็ต IPv6 ที่ห่อหุ้มแพ็กเก็ต IPv4

- Next header: การกำหนดค่าของ next header แบ่งออกเป็น 2 กรณีด้วยกัน คือ กรณีที่ไม่มี IPv6 extension header และกรณีที่มี IPv6 extension header สำหรับกรณีที่ไม่มี IPv6 extension header กำหนดให้ค่า next header มีค่าเท่ากับค่า next header ของ IPv4 header แต่กรณีที่มี IPv6 extension header กำหนดให้ค่าของ next header มีค่าเท่ากับค่า next header ของ IPv6 extension header นั้นๆ โดยที่ค่า next header ของ IPv6 extension header ชุดสุดท้าย กำหนดให้มีค่าเท่ากับค่า next header ของ IPv4 header
- Source address: กำหนดเป็นหมายเลข IPv6 ของอุปกรณ์ที่ทำหน้าที่เป็นอุโมงค์สื่อสารต้นทาง
- Destination address: กำหนดเป็นหมายเลข IPv6 ของอุปกรณ์ที่ทำหน้าที่เป็นอุโมงค์สื่อสารปลายทาง

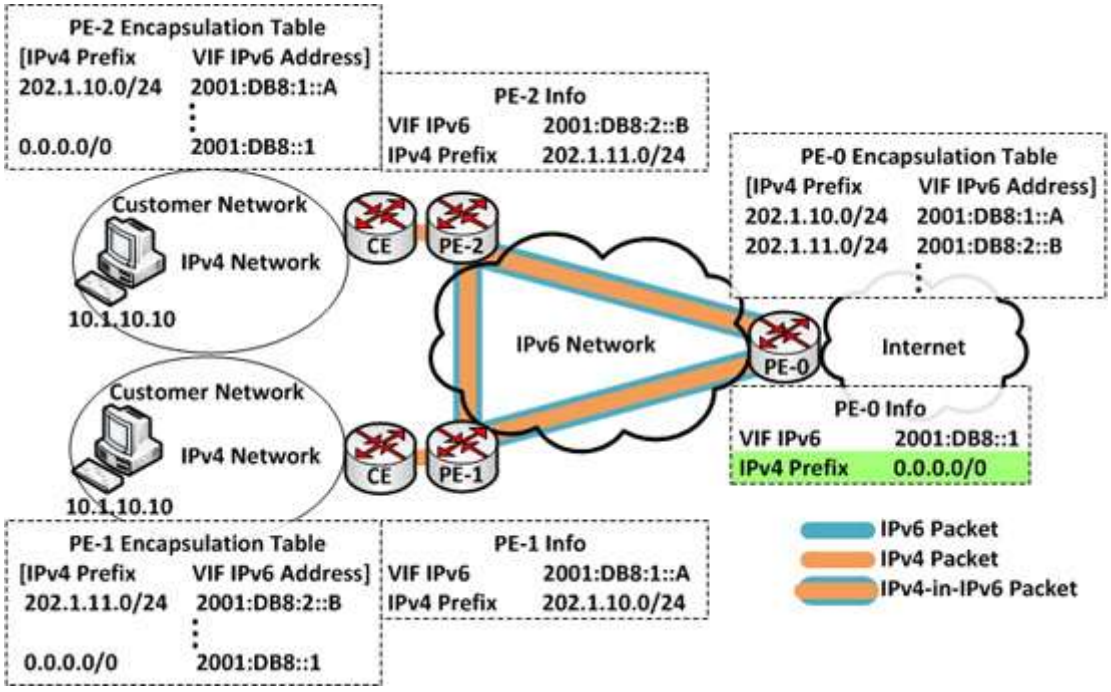
กระบวนการเปลี่ยนถ่ายในปัจจุบันที่ใช้วิธีการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 มีหลากหลายกระบวนการด้วยกัน ซึ่งรายละเอียดของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการมีดังต่อไปนี้

### 2.2.1 4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions

4over6 เป็นกระบวนการเปลี่ยนถ่ายแบบ stateful ซึ่งออกแบบโดยให้ความสำคัญกับความยืดหยุ่นในการใช้งานเป็นหลัก [9] หมายเลข IPv4 และหมายเลข IPv6 ที่ใช้ใน 4over6 สามารถกำหนดได้อย่างอิสระซึ่งแตกต่างจากกระบวนการเปลี่ยนถ่ายแบบแบบ stateless ยกตัวอย่างเช่น 6to4 การกำหนดหมายเลข IPv4 และหมายเลข IPv6 แบบ stateless ของ 6to4 มีข้อกำหนดเข้มงวดและต้องใช้งานร่วมกับ IPv6 prefix พิเศษ 2002::/16 เท่านั้นเพื่อให้สามารถกำหนดหมายเลข IPv6 ได้ครอบคลุมหมายเลข IPv4 อย่างครบถ้วนและไม่ซ้ำซ้อนกับหมายเลข IPv6 สำหรับการใช้งานทั่วไป ยิ่งกว่านั้นจำนวนหมายเลข IPv4 มีน้อยกว่าหมายเลข IPv6 จำนวนมากส่งผลให้ไม่สามารถสร้างความสัมพันธ์จาก IPv6 ไปยัง IPv4 แบบหนึ่งต่อหนึ่งได้ ในทางปฏิบัติต้องสร้างข้อกำหนดเพื่อสร้างความสัมพันธ์หมายเลข IPv6 เพียงบางส่วนให้สอดคล้องกับหมายเลข IPv4 ดังนั้นการกำหนดหมายเลข IPv4 และ IPv6 แบบ stateless โดยใช้เพียง IPv6 prefix เดียวเพื่อให้บริการในระบบ

เครือข่ายแกนหลักที่มีหมายเลข IPv4 หลายช่วงมีข้อเสียสำคัญคือต้องใช้หมายเลข IPv6 จำนวนมากตามไปด้วยส่งผลให้สิ้นเปลืองหมายเลข IPv6 อย่างมาก ด้วยเหตุนี้ทำให้ 4over6 กลายเป็นหนึ่งในต้นแบบของกระบวนการเปลี่ยนถ่ายที่ใช้วิธีการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 แบบ stateful ซึ่งสามารถใช้งานกับระบบเครือข่ายแกนหลักได้อย่างเหมาะสม แม้ว่าหมายเลข IPv4 หรือหมายเลข IPv6 เกิดการเปลี่ยนแปลง เนื่องจากการกำหนดอุโมงค์สื่อสารปลายทางของ 4over6 สามารถเปลี่ยนแปลงตามข้อมูลเครือข่ายใหม่ได้อย่างอัตโนมัติโดยที่ผู้ใช้งานไม่รู้สึกถึงความเปลี่ยนแปลง

อุโมงค์สื่อสารของ 4ove6 ถูกสร้างระหว่างเกตเวย์ของผู้ให้บริการแต่ละแห่ง โดยอุปกรณ์ดังกล่าวจะถูกเรียกว่า “Provider Edge” (PE) ในการกำหนดอุโมงค์สื่อสารของ PE ใช้ข้อมูล 2 ชนิด ได้แก่ IPv4 prefix และหมายเลข IPv6 ซึ่งคล้ายกับข้อมูลที่ใช้สำหรับกำหนดเส้นทางทั่วไป แต่ต่างกันที่ในกรณีนี้ Next Hop คือหมายเลข IPv6 ซึ่งเป็นโพรโตคอลอินเทอร์เน็ตต่างรุ่นกัน ดังนั้น PE ต้องห่อหุ้มแพ็กเก็ต IPv4 ภายในแพ็กเก็ต IPv6 พร้อมกับกำหนดปลายทางเป็นหมายเลข IPv6 ของ PE ของเครือข่ายปลายทาง สำหรับการแลกเปลี่ยนข้อมูลเส้นทางของ IPv4 และ IPv6 4over6 ได้เพิ่มส่วนขยายของ Multiprotocol Extensions for Border Gateway Protocol (MP-BGP) [10] เพื่อใช้สำหรับแลกเปลี่ยนข้อมูล IPv4 prefix และหมายเลข IPv6 ของแต่ละเครือข่าย สำหรับการสร้างการเชื่อมต่อ IPv4 ของ 4over6 สามารถศึกษารายละเอียดได้จากรูปที่ 2-5

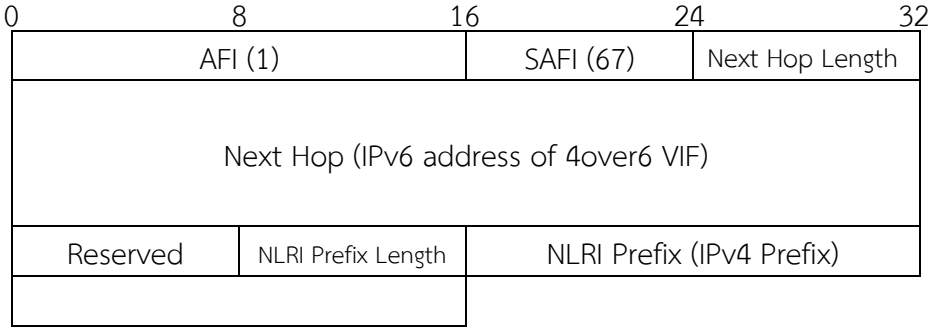



รูปที่ 2-5 การสร้างการเชื่อมต่อ IPv4 ของ 4over6

#### 2.2.1.1 Control plan

อินเทอร์เฟส IPv4 ของ PE สามารถใช้งานร่วมกับโพรโตคอลกำหนดเส้นทางอื่นๆ ได้อย่างหลากหลายเพื่อแลกเปลี่ยนข้อมูลเส้นทางของเครือข่าย IPv4 โดยไม่จำเป็นต้องใช้งานเฉพาะ Border Gateway Protocol (BGP) เมื่อรวบรวมข้อมูลเครือข่าย IPv4 ครบถ้วน ข้อมูลของเครือข่าย IPv4 ทั้งหมดถูกเพิ่มลงไปภายในฐานข้อมูลเครือข่าย IPv4 ที่เชื่อมต่อโดยตรงกับ PE เพื่อใช้สำหรับการประกาศข้อมูลต่อไป PE สามารถมีอินเทอร์เฟส IPv4 มากกว่าหนึ่งอินเทอร์เฟส แต่ต้องมีอินเทอร์เฟส IPv6 อย่างน้อยหนึ่งอินเทอร์เฟสเพื่อทำหน้าที่เป็นอินเทอร์เฟสเสมือนของ IPv4 (VIF) เพื่อใช้สำหรับห่อหุ้มแพ็กเก็ต IPv4 และส่งผ่านเครือข่ายแกนหลัก IPv6 อินเทอร์เฟส IPv6 ของ PE ก็สามารถใช้งานร่วมกับโพรโตคอลกำหนดเส้นทางอื่นๆ เพื่อแลกเปลี่ยนข้อมูลเส้นทางของเครือข่าย IPv6 เดียวกับอินเทอร์เฟส IPv4 แต่โดยส่วนใหญ่นิยมใช้โพรโตคอลกำหนดเส้นทาง BGP อย่างไรก็ตาม แม้ว่า PE สามารถเลือกใช้งานโพรโตคอลกำหนดเส้นทางได้อย่างอิสระ แต่การแลกเปลี่ยนข้อมูลซึ่งเกี่ยวข้องกับ 4over6 เพื่อแลกเปลี่ยนข้อมูลระหว่าง PE อื่นๆ กำหนดให้ต้องดำเนินการผ่าน MP-BGP เท่านั้น การแลกเปลี่ยนข้อมูล 4over6 ระหว่าง PE ด้วย MP-BGP ประกอบด้วย 2 ส่วน คือ การประกาศข้อมูลเส้นทางเชื่อมต่อไปยังเครือข่าย IPv4 และการเรียนรู้ข้อมูลเส้นทางเชื่อมต่อไปยังเครือข่าย IPv4

- ขั้นตอนการประกาศข้อมูลเส้นทางเชื่อมต่อไปยังเครือข่าย IPv4
    1) PE เรียนรู้ข้อมูลของเครือข่าย IPv4 ซึ่งเชื่อมต่อกันโดยตรง จากนั้น PE เพิ่มข้อมูลของเครือข่าย IPv4 ทั้งหมดลงในฐานข้อมูลเครือข่าย IPv4
    2) PE เพิ่มข้อมูลของเครือข่าย IPv4 ลงในตารางกำหนดการห่อหุ้มแพ็กเก็ต โดยตารางดังกล่าวใน 4over6 ถูกเรียกว่า “Encapsulation Table” ข้อมูลภายใน Encapsulation Table ประกอบไปด้วย IPv4 prefix และหมายเลข IPv6 ข้อมูล IPv4 prefix กำหนดเป็นเครือข่าย IPv4 และข้อมูลหมายเลข IPv6 กำหนดเป็นหมายเลข IPv6 ของอินเทอร์เฟสเสมือนของ PE
    3) PE ประกาศข้อมูลภายใน Encapsulation Table ซึ่งมีการเพิ่มเติมหรือปรับปรุงใหม่ออกไปยัง PE อื่นๆ ภายในเครือข่ายแกนหลัก IPv6 โดยใช้ MP-BGP code MP_REACH_NLRI รูปแบบฟิลด์ข้อมูลของ MP_REACH_NLRI มีละเอียดดังแสดงในรูปที่ 2-6




รูปที่ 2-6 รูปแบบฟิลด์ข้อมูลของ MP_REACH_NLRI

จากรูปที่ 2-6 รายละเอียดของการกำหนดค่าฟิลด์ข้อมูลของ MP_REACH_NLRI เพื่อประกาศข้อมูลภายใน Encapsulation Table มีดังต่อไปนี้

- ฟิลด์ข้อมูล AFI ถูกกำหนดค่าเท่ากับ 1 (IPv4)
- ฟิลด์ข้อมูล SAFI ถูกกำหนดค่าเท่ากับ 67 (4over6)
- ฟิลด์ข้อมูล NLRI ถูกกำหนดค่าเท่ากับเครือข่าย IPv4
- ฟิลด์ข้อมูล Next Hop ถูกกำหนดค่าเท่ากับหมายเลข IPv6 ของอินเทอร์เฟสเสมือนของ PE

- การเรียนรู้ข้อมูลเส้นทางเชื่อมต่อไปยังเครือข่าย IPv4
  1) เมื่อ PE ได้รับ MP_REACH_NLRI PE ต้องตรวจสอบความถูกต้องของ MP_REACH_NLRI โดยฟิลด์ข้อมูล AFI ต้องมีค่าเท่ากับ 1 และฟิลด์ข้อมูล SAFI ต้องมีค่าเท่ากับ 67
  2) หาก MP_REACH_NLRI ที่ได้รับถูกต้อง ข้อมูลเครือข่าย IPv4 ภายในฟิลด์ข้อมูล NLRI และหมายเลข IPv6 ภายในฟิลด์ข้อมูล Next Hop ถูกนำออกมาเพื่อเพิ่มเติมหรือปรับปรุง Encapsulation table
  3) ในกรณีที่ข้อมูลเครือข่าย IPv4 ถูกเพิ่มเติม ข้อมูลเครือข่าย IPv4 ใหม่ต้องถูกกระจายไปยังเครือข่าย IPv4 เพื่อประกาศเส้นทางเชื่อมต่อไปยังเครือข่าย IPv4 ดังกล่าว จากนั้นข้อมูลเครือข่าย IPv4 ใหม่ถูกเพิ่มลงในตารางกำหนดเส้นทางภายใน PE ซึ่งได้รับ MP_REACH_NLRI พร้อมกับกำหนดค่า Next Hop เท่ากับ Null และ OUTPUT Interface เท่ากับอินเทอร์เฟสเสมือนของ PE

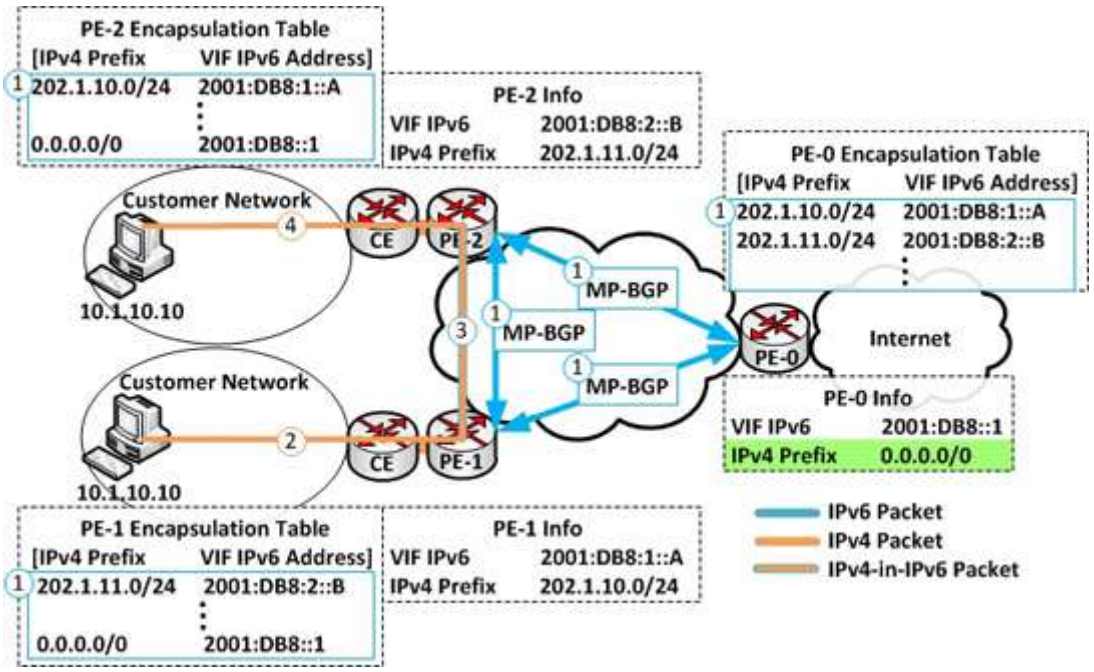
**2.2.1.2 Data plan**

การส่งแพ็กเก็ต IPv4 ผ่านเครือข่ายแกนหลัก IPv6 ของ 4over6 อาศัยหลักการห่อหุ้มแพ็กเก็ตและการกำหนดปลายทางโดยใช้ข้อมูลภายใน Encapsulation Table ซึ่งถูกปรับปรุงด้วยโพรโตคอลกำหนดเส้นทาง BGP ดังที่ได้กล่าวไปแล้ว เมื่อเครื่องต้นทางซึ่งเชื่อมต่อกับ PE-1 ต้องการติดต่อสื่อสารกับเครื่องปลายทางซึ่งเชื่อมต่อกับ PE-2 แพ็กเก็ต IPv4 ถูกส่งจากเครื่องต้นทางไปยัง PE-1 โดยใช้เส้นทางเชื่อมต่อไปยังเครือข่าย IPv4 ปลายทางตามที่ PE-1 ได้ประกาศออกไป เมื่อ PE-1 ได้รับแพ็กเก็ต IPv4 หมายเลข IPv4 ปลายทางถูกอ่านออกมาและนำไปค้นหาภายในตารางกำหนดเส้นทางของ PE-1 หากพบว่า หมายเลข IPv4 ปลายทางดังกล่าวต้องถูกส่งไปยังเครื่องปลายทางผ่านอินเทอร์เฟสเสมือนของ PE-1 แพ็กเก็ตจะถูกดำเนินการห่อหุ้มตามข้อมูลภายใน Encapsulation Table โดยหมายเลข IPv6 ต้นทางถูกกำหนดเป็นหมายเลข IPv6 ของอินเทอร์เฟสเสมือนของ PE-1 และหมายเลข IPv6 ปลายทางถูกกำหนดเป็นหมายเลข IPv6 ในฟิลด์ข้อมูล Next Hop ของ Encapsulation Table ซึ่งมีฟิลด์ข้อมูล IPv4 prefix สอดคล้องกับหมายเลข IPv4 ปลายทาง (ในกรณีนี้กำหนดหมายเลข IPv6 ปลายทางเป็นหมายเลข IPv6 ของอินเทอร์เฟสเสมือนของ PE-2) เมื่อแพ็กเก็ตที่ถูกห่อหุ้มด้วย IPv6 ถูกส่งมาถึง PE-2 หากแพ็กเก็ตที่ถูกห่อหุ้มสอดคล้องกับข้อมูลภายใน Encapsulation Table ของ PE-2 แพ็กเก็ต IPv4 ดั้งเดิมภายในแพ็กเก็ต IPv6ดังกล่าวจะถูกนำออกมาเพื่อส่งต่อไปยังเครื่องปลายทางต่อไป

### 2.2.1.3 หลักการทำงาน

หลักการทำงานโดยภาพรวมของ 4over6 มีขั้นตอนดังแสดงในรูปที่ 2-7

1) ข้อมูลของเครือข่าย IPv4 และ IPv6 ถูกแลกเปลี่ยนด้วย MP-BGP ระหว่าง PE ซึ่งถูกกำหนดไว้ล่วงหน้า ข้อมูลของเครือข่าย IPv4 และ IPv6 ที่ได้จากการแลกเปลี่ยนจะถูกบันทึกลงภายใน Encapsulation Table
2) PE-1 ได้รับแพ็กเก็ต IPv4 ที่ถูกส่งจากเครื่องลูกข่ายภายในเครือข่ายของผู้ใช้งาน
3) แพ็กเก็ต IPv4 ดังกล่าวอาจมีดำเนินการแปลง private IPv4 address ของผู้ส่งเป็น public IPv4 address ขึ้นอยู่กับการดำเนินการของ PE-1 จากนั้นจึงถูกห่อหุ้มภายใน IPv6 และส่งต่อไปยัง PE ปลายทางตามข้อมูลภายใน Encapsulation table ซึ่งในกรณีนี้ PE ปลายทางคือ PE-2
4) เมื่อ PE-2 ได้รับแพ็กเก็ตที่ถูกห่อหุ้ม PE-2 จะนำแพ็กเก็ต IPv4 ดั้งเดิมออกมา ซึ่งอาจมีดำเนินการแปลง public IPv4 address ของผู้รับเป็น private IPv4 address ขึ้นอยู่กับการดำเนินการของ PE-2 แล้วจึงส่งต่อไปยังเครื่องปลายทางในที่สุด




รูปที่ 2-7 หลักการทำงานโดยภาพรวมของ 4over6

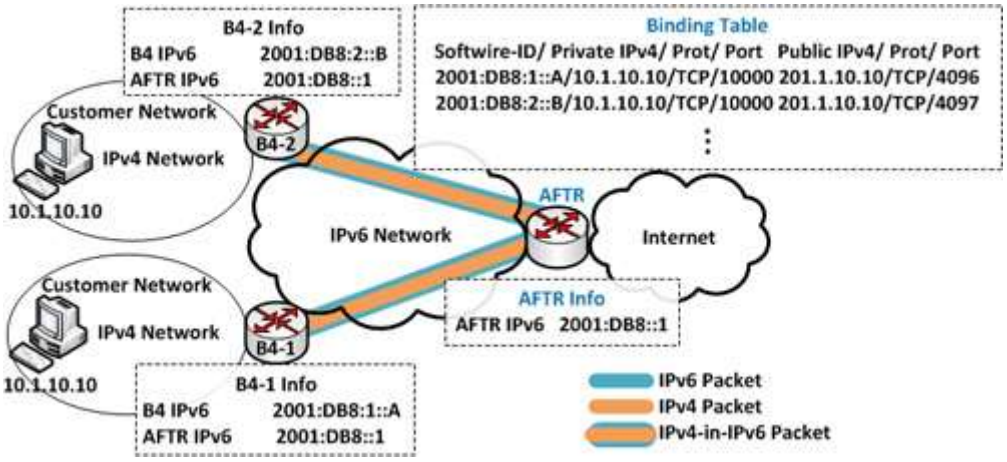
### 2.2.2 Dual-Stack lite (DS-lite)

ในอดีตการเชื่อมต่อกับเครือข่าย IPv4 นั้นนิยมจัดสรร public IPv4 address หนึ่งหมายเลขต่อหนึ่งเครือข่ายของผู้ใช้บริการ เพื่อให้ผู้ใช้งานสามารถสร้างการเชื่อมต่อผ่านกระบวนการ Network Address Translation เพื่อแปลงจาก private IPv4 address เป็น public IPv4 address (NAT44) โดยอุปกรณ์ฝั่งของผู้ใช้บริการเอง แต่เมื่อมีผู้ใช้งานมีจำนวนมากขึ้น และระบบเครือข่าย

ขยายใหญ่ขึ้นทำให้มีจำนวน public IPv4 address ไม่เพียงพอต่อการให้บริการด้วยวิธีการข้างต้น ผู้ให้บริการอินเทอร์เน็ตจึงปรับเปลี่ยนวิธีการจัดสรร public IPv4 address โดยแทนที่ผู้ให้บริการจัดสรร public IPv4 address ผู้ให้บริการจะดำเนินการจัดสรร private IPv4 address ให้แก่เครือข่ายของผู้ใช้บริการ และดำเนินการ Network Address Translation (NAT) จาก private IPv4 address ให้กลายเป็น public IPv4 address โดยฝั่งผู้ให้บริการ การดำเนินการเช่นนี้สามารถเรียกอีกอย่างหนึ่งว่า “large-scale NAT” (LSN) LSN มีประสิทธิภาพในการจัดสรรหมายเลข IPv4 สูงกว่าการดำเนินการแบบแรก แต่ในบางเครือข่ายของผู้ใช้บริการอาจมีการดำเนินการ NAT อีกชั้นหนึ่ง ส่งผลให้มีการดำเนินการ NAT รวมเป็น 2 ครั้งด้วยกัน (NAT444) ซึ่งอาจเกิดปัญหาการซ้ำกันของ private IPv4 address เนื่องจากเครือข่ายของผู้ให้บริการ และเครือข่ายของผู้ใช้บริการต่างก็ใช้ private IPv4 address เช่นเดียวกัน เพื่อป้องกันการเกิดปัญหาดังกล่าวจึงมีการพัฒนากระบวนการที่เรียกว่า “NAT464” กระบวนการนี้ช่วยป้องกันการซ้ำกันของ private IPv4 address โดยเมื่อแพ็กเก็ต IPv4 ถูกส่งออกจากเครือข่ายของผู้ใช้บริการก็จะถูกแปลงเป็นแพ็กเก็ต IPv6 และเมื่อถูกส่งมาถึงเครือข่ายของผู้ให้บริการจะถูกแปลงกลับเป็นแพ็กเก็ต IPv4 ดังเดิม อย่างไรก็ตาม NAT464 กลับสร้างปัญหาใหม่เกี่ยวกับการแปลงระหว่าง IPv4 และ IPv6 ขึ้นมาแทนที่ ท้ายที่สุดกระบวนการ Dual-Stack lite จึงถูกพัฒนาขึ้นมาแทนที่ Dual-Stack lite คล้ายกับ NAT464 แต่ Dual-Stack lite ใช้การห่อหุ้มแพ็กเก็ต IPv4 ลงในแพ็กเก็ต IPv6 แทนการแปลงแพ็กเก็ต IPv4 ให้เป็นแพ็กเก็ต IPv6 จึงไม่ประสบปัญหาจากการแปลงแพ็กเก็ต [11]

Dual-Stack lite สามารถเขียนโดยย่ออีกแบบหนึ่งว่า “DS-lite” ใช้หลักการห่อหุ้มแพ็กเก็ต IPv4 ด้วยแพ็กเก็ต IPv6 เพื่อสร้างเป็นอุโมงค์สื่อสาร โดยอุโมงค์สื่อสารถูกสร้างเพื่อเชื่อมต่อระหว่างเกตเวย์ของฝั่งผู้ใช้งาน และอุปกรณ์ของผู้ให้บริการโดยสร้างผ่านเครือข่ายแกนหลักของผู้ให้บริการ เกตเวย์ของฝั่งผู้ใช้งานถูกเรียกว่า “Basic Bridging BroadBand” (B4) และอุปกรณ์ของผู้ให้บริการถูกเรียกว่า “Address Family Transition Router” (AFTR) การสร้างการเชื่อมต่อ IPv4 ผ่านเครือข่าย IPv6 ช่วยให้อุปกรณ์ภายในเครือข่ายของผู้ให้บริการไม่จำเป็นต้องใช้งาน IPv4 ส่งผลให้สามารถลดการใช้งานหมายเลข IPv4 และลดการใช้ทรัพยากรบนอุปกรณ์ในเครือข่ายของผู้ให้บริการ

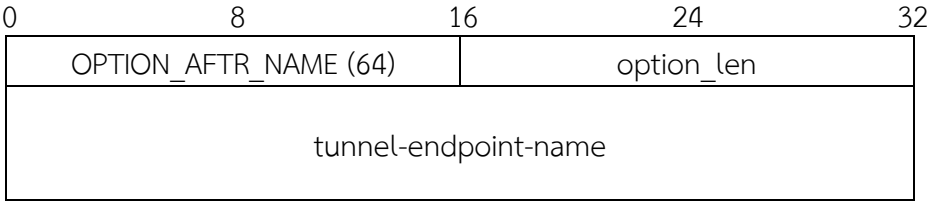
เมื่อ B4 ห่อหุ้มแพ็กเก็ต IPv4 แล้วส่งมายัง AFTR AFTR จะต้องนำแพ็กเก็ต IPv4 ที่บรรจุอยู่ภายในแพ็กเก็ต IPv6 ออกมา แต่เนื่องจากหมายเลข IPv4 ที่ใช้งานนั้นเป็น private IPv4 address ดังนั้น AFTR จำเป็นต้องดำเนินการ NAT เพื่อแปลงจาก private IPv4 address ให้กลายเป็น public IPv4 address ก่อนที่จะส่งต่อแพ็กเก็ต IPv4 ไปยังเครื่องปลายทาง แต่การดำเนินการ NAT ของกระบวนการ DS-lite นั้นไม่สามารถดำเนินการโดยใช้เพียงแค่ข้อมูลของ inside IPv4 source address + port และ outside IPv4 source address + port ได้ เนื่องจากวัตถุประสงค์ของ DS-lite นั้นต้องการให้เครือข่ายของผู้ใช้งานสามารถกำหนดหมายเลข IPv4 ได้อย่างอิสระ หมายเลข IPv4 ของผู้ใช้งานแต่ละคนอาจมีการซ้ำซ้อนกันทำให้ไม่สามารถระบุกลับไปยังต้นทางที่ถูกต้องได้ ดังนั้นการดำเนินการ NAT ของกระบวนการ DS-lite ต้องทำการบันทึกข้อมูลของหมายเลข IPv6 ของ B4 ของแต่ละเครือข่ายผู้ใช้งานเพิ่มเติมเพื่อให้สามารถตรวจสอบกลับไปยังเครื่องต้นทางที่ถูกต้องได้ สำหรับการสร้างการเชื่อมต่อ IPv4 ของ DS-lite สามารถศึกษารายละเอียดได้จากรูปที่ 2-8




รูปที่ 2-8 การสร้างการเชื่อมต่อ IPv4 ของ DS-lite

### 2.2.2.1 Control plan

การควบคุมการให้บริการการเชื่อมต่อ IPv4 ผ่านเครือข่ายแกนหลัก IPv6 ของ DS-lite มีหลักการทำงานที่เรียบง่าย เนื่องจากควบคุมเฉพาะช่วงเริ่มต้นในการสร้างการเชื่อมต่อเท่านั้น รูปแบบการเชื่อมต่อของ DS-lite นั้นมีเพียง AFTR เท่านั้นที่มีอุโมงค์สื่อสารเชื่อมต่อไปยัง B4 ทั้งหมด (Hub & Spoke) โดยที่ B4 ส่งแพ็กเก็ตไปยังเครื่องปลายทางโดยอาศัย AFTR เสมอ B4 จึงไม่ต้องคำนึงถึงเครือข่ายผู้ใช้งานใหม่ที่เพิ่มขึ้นหรือเครือข่ายผู้ใช้งานที่ลดลง เพราะหน้าที่ดังกล่าวนั้นถูกดำเนินการโดย AFTR เพียงอุปกรณ์เดียว เพื่อสร้างการเชื่อมต่อ IPv4 ใน DS-lite B4 ต้องการข้อมูลของ AFTR เพื่อกำหนดเป็นอุโมงค์สื่อสารปลายทาง ดังนั้น DS-lite ได้นิยาม DHCPv6 option เพื่อใช้สำหรับส่งข้อมูลของ AFTR ให้กับ B4 [12] DHCPv6 option ดังกล่าวถูกเรียกว่า “OPTION _AFTR_NAME” รูปแบบ DHCPv6 OPTION_AFTR_NAME มีรายละเอียดดังแสดงดังรูปที่ 2-9




รูปที่ 2-9 รูปแบบ DHCPv6 OPTION_AFTR_NAME

จากรูปที่ 2-9 รูปแบบ DHCPv6 option ทั่วไปมีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต และ option_data สำหรับ DHCPv6 OPTION_AFTR_NAME จึงกำหนดให้ option_code มีค่าเท่ากับ 64 และ option_data บรรจุข้อมูลเพียงฟิลด์ข้อมูล tunnel-endpoint-name เท่านั้น ข้อมูลภายใน tunnel-endpoint-name ใช้สำหรับบ่งบอกอุโมงค์สื่อสารปลายทางของ AFTR ให้กับ B4 ซึ่งข้อมูลดังกล่าวถูกระบุในรูปแบบ Fully Qualified Domain Name (FQDN) เมื่อข้อมูล tunnel-endpoint-name ได้รับในรูปแบบของโดเมนเนม B4 จำเป็นต้องเรียนรู้ข้อมูลของ DNS server ควบคู่ไปด้วย

เพื่อค้นหา AFTR B4 เริ่มต้นโดยดำเนินการร้องขอข้อมูลของ DHCPv6 OPTION _AFTR_NAME และ OPTION_DNS_SERVERS ไปยัง DHCPv6 server เมื่อ B4 ได้รับข้อมูล AFTR

จาก DHCPv6 server B4 เริ่มดำเนินการร้องขอหมายเลข IPv6 จากโดเมนเนมของ AFTR เมื่อ B4 ได้รับหมายเลข IPv6 ของ AFTR B4 นำหมายเลข IPv6 ดังกล่าวกำหนดเป็นอุโมงค์สื่อสารปลายทาง และสามารถเริ่มต้นประกาศ private IPv4 address เพื่อนำไปใช้ภายในเครือข่ายของผู้ใช้บริการ

**2.2.2.2 Data plan**

การส่งแพ็กเก็ต IPv4 ผ่านเครือข่ายแกนหลัก IPv6 ของ DS-lite อาศัยหลักการห่อหุ้มแพ็กเก็ตและการกำหนดปลายทางไปยัง AFTR เพียงปลายทางเดียว ซึ่งข้อมูลของ AFTR ถูกแลกเปลี่ยนด้วย DHCPv6 ส่วนการจัดสรร public IPv4 address และพอร์ตเป็นหน้าที่ของ AFTR อย่างไรก็ตามรูปแบบการเชื่อมต่อของ DS-lite แบ่งออกเป็น 2 รูปแบบด้วยกันเพื่อให้เหมาะกับจำนวนผู้ใช้งานภายในเครือข่ายผู้ใช้งานแต่ละเครือข่าย รูปแบบที่หนึ่งเรียกว่า “Gateway-Based Architecture” และรูปแบบที่สองเรียกว่า “Host-Based Architecture” ทั้งสองรูปแบบการเชื่อมต่อมีรายละเอียดของการทำงานในส่วนของ Data plan ดังต่อไปนี้

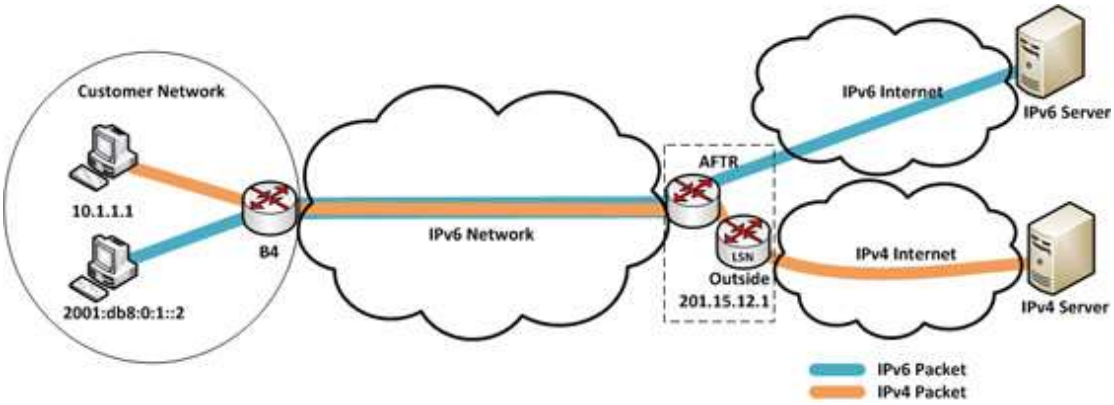
- **Gateway-Based Architecture** เป็นรูปแบบการใช้งานสำหรับให้บริการเครื่องลูกข่ายหลายเครื่อง โดยเครื่องลูกข่ายแต่ละเครื่องจะเชื่อมต่อกับ B4 และ B4 เป็นจุดที่สร้าง tunneling เชื่อมต่อกับ AFTR ของผู้ให้บริการดังแสดงในรูปที่ 2-10

ในการสร้างการเชื่อมต่อ IPv4 สำหรับ DS-lite รูปแบบ Gateway-Based Architecture เมื่อเครื่องต้นทาง IPv4 10.1.1.1 port 10000 ต้องการที่จะติดต่อกับเครื่องปลายทาง 198.51.100.1 port 80 จะมีขั้นตอนการดำเนินการในระหว่างการส่งแพ็กเก็ตดังต่อไปนี้

ขั้นที่ 1: เครื่องต้นทางสร้างแพ็กเก็ต IPv4 เพื่อส่งไปยังเครื่องปลายทางผ่าน B4

ขั้นที่ 2: B4 ทำการห่อหุ้มแพ็กเก็ต IPv4 ด้วยแพ็กเก็ต IPv6 และระบุปลายทางเป็น AFTR

ขั้นที่ 3: AFTR นำแพ็กเก็ต IPv4 ภายในแพ็กเก็ต IPv6 ออกมา แล้วดำเนินการ NAT เพื่อแปลงจาก private IPv4 address เป็น public IPv4 address และส่งไปยังปลายทางต่อไป โดยที่ AFTR จะทำการบันทึกข้อมูลดังแสดงในตารางที่ 2-1 เพื่อใช้สำหรับการดำเนินการ NAT

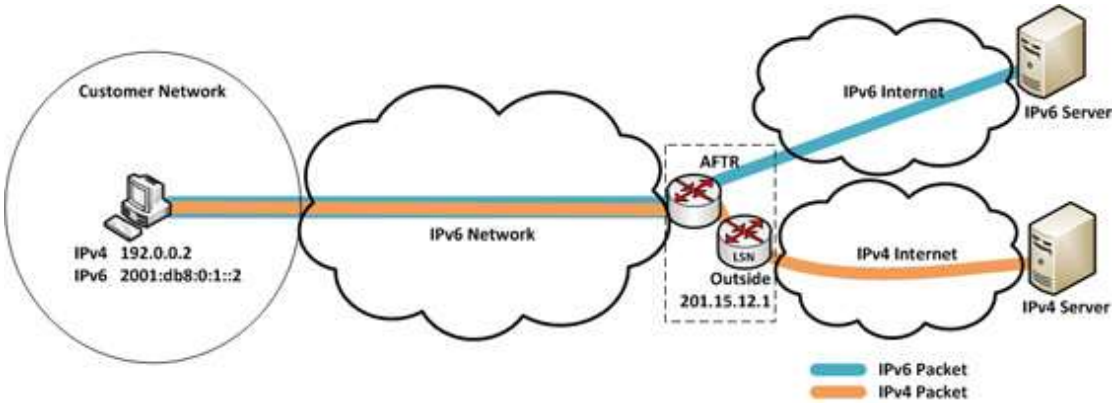



รูปที่ 2-10 DS-lite รูปแบบ Gateway-Based Architecture

ตารางที่ 2-1 ข้อมูลสำหรับดำเนินการ NAT ของ DS-lite แบบ Gateway-Based Architecture

| Softwire-ID/IPv4/Prot/Port | IPv4/Prot/Port |
|---|---|
| 2001:db8:0:1::1/10.1.1.1/TCP/10000 | 201.15.12.1/TCP/5000 |

- **Host-Based Architecture** เป็นรูปแบบการใช้งานสำหรับให้บริการเครื่องลูกข่ายเพียงเครื่องเดียว ซึ่งเหมาะสำหรับเครื่องลูกข่ายที่เชื่อมต่อกับผู้ให้บริการโดยตรง โดยองค์การกำหนดหมายเลขอินเทอร์เน็ตได้สงวน IPv4 subnet 192.0.0.0/29 ไว้สำหรับใช้กำหนดให้กับเครื่องที่ใช้งานรูปแบบนี้ ทุกโฮสที่ใช้การเชื่อมต่อเช่นนี้จะมีหมายเลข IPv4 เหมือนกันทั้งหมด โดยเครื่องลูกข่ายจะต้องสร้างอุโมงค์สื่อสารเพื่อเชื่อมต่อกับ AFTR ของผู้ให้บริการโดยตรงดังแสดงในรูปที่ 2-11




รูปที่ 2-11 DS-lite รูปแบบ Host-Based Architecture

ในการสร้างการเชื่อมต่อ IPv4 สำหรับ DS-lite รูปแบบ Host-Based Architecture เมื่อเครื่องต้นทาง IPv4 192.0.0.2 port 10000 ต้องการที่จะติดต่อกับเครื่องปลายทาง 198.51.100.1 port 80 จะมีขั้นตอนการดำเนินการในระหว่างการส่งแพ็กเก็ตดังต่อไปนี้

ขั้นที่ 1: เครื่องต้นทาง (ในกรณีนี้ก็คือ B4) สร้างแพ็กเก็ต IPv4 และทำการห่อหุ้มแพ็กเก็ต IPv4 ด้วยแพ็กเก็ต IPv6 เพื่อส่งไปยังปลายทาง โดยแพ็กเก็ต IPv6 ที่ห่อหุ้มระบุปลายทางเป็น AFTR

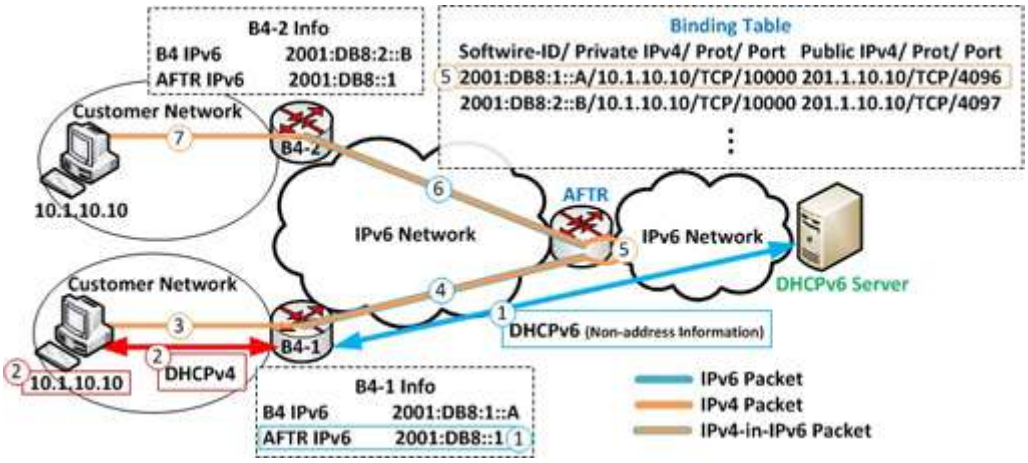
ขั้นที่ 2: AFTR นำแพ็กเก็ต IPv4 ภายในแพ็กเก็ต IPv6 ออกมา แล้วดำเนินการ NAT เพื่อแปลงจาก private IPv4 address เป็น public IPv4 address และส่งไปยังปลายทางต่อไป โดยที่ AFTR จะทำการบันทึกข้อมูลดังแสดงในตารางที่ 2-2 เพื่อใช้สำหรับการดำเนินการ NAT

ตารางที่ 2-2 ข้อมูลสำหรับดำเนินการ NAT ของ DS-lite แบบ Host-Based Architecture

| Softwire-ID/IPv4/Prot/Port | IPv4/Prot/Port |
|---|---|
| 2001:db8:0:1::2/192.0.0.2/TCP/10000 | 201.15.12.1/TCP/5000 |

### 2.2.2.3 หลักการทำงาน

หลักการทำงานโดยภาพรวมของ DS-lite มีขั้นตอนดังแสดงในรูปที่ 2-12




รูปที่ 2-12 หลักการทำงานโดยภาพรวมของ DS-lite

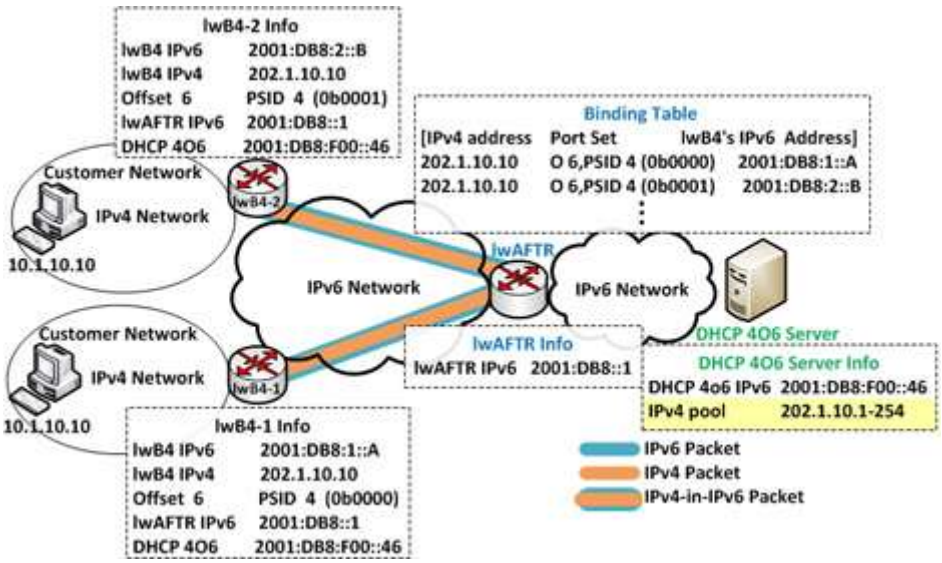
1) B4-1 ร้องขอข้อมูลของ AFTR ด้วย OPTION_AFTR_NAME ไปยัง DHCPv6 server และได้รับข้อมูลของ AFTR กลับมาดังแสดงในตารางข้อมูลของ B4-1
2) จากนั้น B4-1 สามารถกำหนดหมายเลข IPv4 ให้กับเครื่องลูกข่ายภายในเครือข่ายของผู้ใช้งาน
3) แพ็กเก็ต IPv4 ถูกส่งจากเครื่องลูกข่ายภายในเครือข่ายของผู้ใช้งานไปยัง B4-1
4) แพ็กเก็ต IPv4 ดังกล่าวถูกห่อหุ้มภายในแพ็กเก็ต IPv6 และส่งต่อไปยัง AFTR
5) AFTR นำแพ็กเก็ต IPv4 ดั้งเดิมออกมา จากนั้นแปลง private IPv4 address ของผู้ส่งเป็น public IPv4 address และบันทึกลงใน Binding Table แล้วจึงส่งต่อไปยังเครื่องปลายทาง
6) แต่ในกรณีนี้เครื่องปลายทางอยู่ภายในเครือข่ายของผู้ใช้งาน 2 แพ็กเก็ต IPv4 จึงถูกแปลง public IPv4 address ของผู้รับเป็น private IPv4 address และถูกห่อหุ้มภายในแพ็กเก็ต IPv6 แล้วส่งต่อไปยัง B4-2
7) เมื่อ B4-2 ได้รับแพ็กเก็ตที่ถูกห่อหุ้ม แพ็กเก็ตจะถูกนำแพ็กเก็ต IPv4 ออกมาเพื่อส่งต่อไปยังเครื่องปลายทางต่อไป

### 2.2.3 Lightweight 4over6 (lw4over6)

Lightweight 4over6 หรือสามารถเรียกอีกชื่อหนึ่งว่า “lw4over6” เป็นกระบวนการที่ถูกพัฒนาขึ้นเพื่อเป็นส่วนขยายของ DS-lite [13] lw4over6 แก้ไขข้อจำกัดเรื่องการดำเนินการ NAT แบบรวมศูนย์ของ DS-lite ด้วยการกระจาย public IPv4 address และ port-set ให้แก่เครือข่ายของผู้ใช้งาน การดำเนินการ NAT ของ lw4over6 จึงถูกกระจายไปยังอุปกรณ์ของผู้ใช้งานตามไปด้วย อุปกรณ์ในกระบวนการ lw4over6 ถูกเรียกคล้ายกับอุปกรณ์ในกระบวนการ DS-lite โดยอุปกรณ์ฝั่งผู้ใช้งานเรียกว่า “Lightweight Basic Bridging BroadBand” (lwB4) และอุปกรณ์ฝั่งผู้ให้บริการเรียกว่า “Lightweight Address Family Transition Router” (lwAFTR)

การกระจายการดำเนินการ NAT ไปยังอุปกรณ์ของผู้ใช้บริการส่งผลให้อุปกรณ์ของผู้ให้บริการบำรุงรักษาข้อมูลลดลง จากเดิมที่ AFTR ของ DS-lite บันทึกข้อมูล 5 ชนิด แต่ lwAFTR ของ lw4over6 บันทึกข้อมูลเพียง 3 ชนิดเท่านั้น ได้แก่ public IPv4 address, port-set และหมายเลข IPv6 ของ lwB4 ในทางกลับกัน lwB4 ไม่เพียงต้องการหมายเลข IPv6 ของ lwAFTR แต่ยังต้องการข้อมูล public IPv4 address และ port-set สำหรับใช้ในการดำเนินการ NAT อีกด้วย สำหรับการสร้างการเชื่อมต่อ IPv4 ของ lw4over6 สามารถดูรายละเอียดได้จากรูปที่ 2-13

ในการดำเนินการจัดสรร public IPv4 address และ port-set lwAFTR ไม่ได้จัดการโดยตรง แต่ lw4over6 ได้นิยาม DHCPv4 over DHCPv6 Server (DHCP 4o6 Server) เพื่อทำหน้าที่จัดสรรโดยเฉพาะ ดังนั้น lwAFTR ต้องการการดำเนินการที่ช่วยให้สามารถปรับปรุงข้อมูลของการจัดสรร public IPv4 address และ port-set เพิ่มเติม แต่เนื่องจาก lw4over6 ยังอยู่ในขั้นตอนการร่างของ IETF รายละเอียดจึงยังไม่สมบูรณ์มากนัก อย่างไรก็ตาม lw4over6 ให้ความสำคัญการปรับปรุงข้อมูลการจัดสรร public IPv4 address และ port-set ผ่าน DHCP โดยใช้การรับส่งข้อมูลผ่าน multicast และกำหนดให้ lwAFTR อยู่ภายในเส้นทางการส่งข้อมูลระหว่าง DHCP 4o6 Server และ lwB4 เพื่อให้สามารถปรับปรุงข้อมูลการจัดสรร public IPv4 address และ port-set โดยการติดตามข้อมูล DHCP ทั้งหมดที่ถูกส่งผ่าน lwAFTR




รูปที่ 2-13 การสร้างการเชื่อมต่อ IPv4 ของ lw4over6

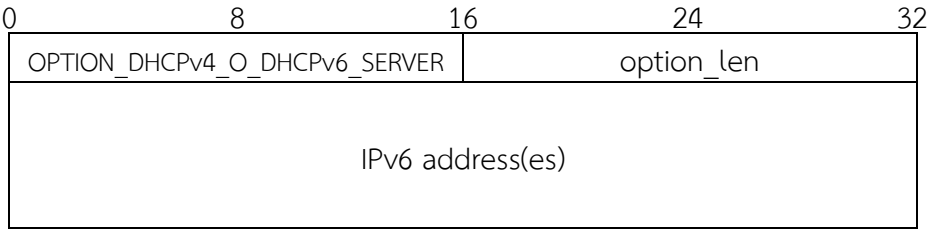
### 2.2.3.1 Control plan

การควบคุมการให้บริการการเชื่อมต่อ IPv4 ผ่านเครือข่ายแกนหลัก IPv6 ของ lw4over6 นั้นควบคุมเฉพาะช่วงเริ่มต้นในการสร้างการเชื่อมต่อเท่านั้น โดยรูปแบบการเชื่อมต่อของ lw4over6 เป็นแบบ Hub & Spoke เช่นเดียวกับ DS-lite ส่งผลให้ lwB4 ไม่ต้องคำนึงถึงเครือข่ายผู้ใช้งานใหม่ที่เพิ่มขึ้นหรือเครือข่ายผู้ใช้งานที่ลดลง โดยหน้าที่ดังกล่าวนั้นถูกดำเนินการโดย lwAFTR เพียงอุปกรณ์เดียว เพื่อสร้างการเชื่อมต่อ IPv4 ใน lw4ove6 lwB4 ไม่เพียงต้องการข้อมูลของ lwAFTR เพื่อกำหนดเป็นอุโมงค์สื่อสารปลายทาง แต่ lwB4 ยังต้องการข้อมูลหมายเลข IPv4 และ port-set เพิ่มเติมเพื่อใช้ในการดำเนินการ NAT สำหรับวิธีการสำหรับประกาศข้อมูลที่จำเป็นให้กับ

lwB4 ใน lw4over6 มีการนำเสนอออกมา 2 รูปแบบด้วยกัน คือ การส่งข้อมูลด้วย DHCPv4 over DHCPv6 [14], [15] และการส่งข้อมูลด้วย DHCPv6 [16] ซึ่งทั้งสองวิธีการมีรายละเอียดดังต่อไปนี้

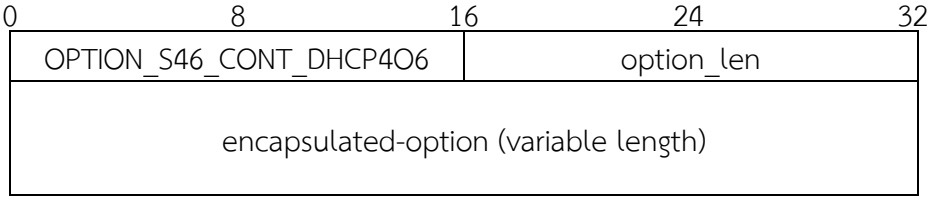
- การจัดหาข้อมูล lw4over6 ด้วย DHCPv4 over DHCPv6

สำหรับการส่งข้อมูลที่จำเป็นไปยัง lwB4 ด้วย DHCPv4 over DHCPv6 lw4over6 ได้นิยาม DHCPv6 option ประกอบด้วย OPTION_DHCPv4_O_DHCPv6_SERVER และ OPTION_S46_CONT_DHCP4O6 [17] โดย OPTION_DHCPv4_O_DHCPv6_SERVER ใช้สำหรับส่งข้อมูลหมายเลข IPv6 ของ DHCP 4O6 Server และ OPTION_S46_CONT_DHCP4O6 ใช้สำหรับร้องขอหมายเลย IPv6 ของ lwAFTR และลงทะเบียนหมายเลข IPv6 ของ lwB4 กับ DHCP 4O6 Server รูปแบบ DHCPv6 OPTION_DHCPv4_O_DHCPv6_SERVER และ OPTION_S46_CONT_DHCP4O6 มีรายละเอียดดังแสดงในรูปที่ 2-14 และรูปที่ 2-15 ตามลำดับ




รูปที่ 2-14 รูปแบบ DHCPv6 OPTION_DHCPv4_O_DHCPv6_SERVER

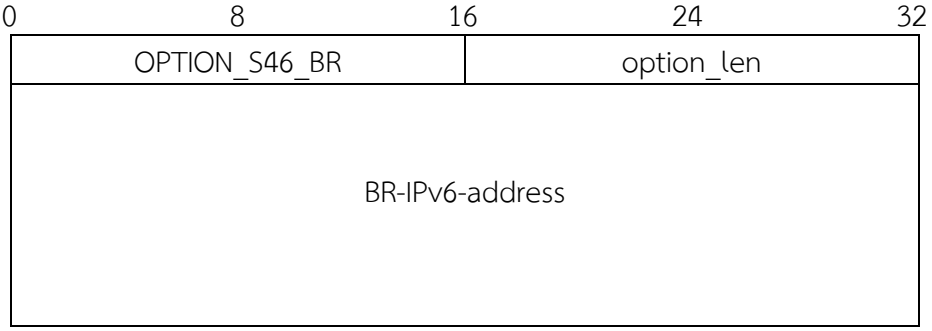
จากรูปที่ 2-14 รูปแบบ DHCPv6 OPTION_DHCPv4_O_DHCPv6_SERVER มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิตและมีค่าเท่ากับ 88, option_length ขนาด 16 บิต และ IPv6 address(es) ขนาดของ option_length มีค่าเป็นจำนวนเท่าของ 16 เนื่องจาก IPv6 address(es) บรรจุด้วยหมายเลข IPv6 ของ DHCP 4O6 Server ซึ่งในบางครั้งอาจมีหลายหมายเลข ในกรณีที่ option_length ถูกกำหนดค่าเท่ากับ 0 หมายความว่า lwB4 สามารถติดต่อไปยัง DHCP 4O6 Server โดยใช้ All_DHCP_Relay_Agents_and_Servers multicast address




รูปที่ 2-15 รูปแบบ DHCPv6 OPTION_S46_CONT_DHCP4O6

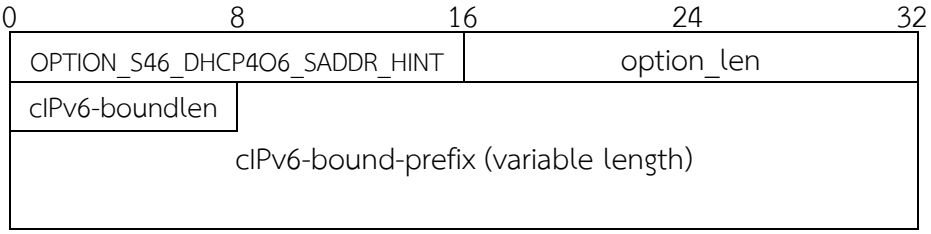
จากรูปที่ 2-15 รูปแบบ DHCPv6 OPTION_S46_CONT_DHCP4O6 มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต และ encapsulated-option โดย encapsulated-option ใช้สำหรับบรรจุ option ที่เกี่ยวข้องกับ lw4over6 3 option ด้วยกันเพื่อใช้ในการร้องขอข้อมูลของ lwAFTR และลงทะเบียนหมายเลข IPv6 เพื่อใช้สำหรับร้องขอหมายเลข IPv4 และ port-set ต่อไป ซึ่ง DHCPv6 option ที่เกี่ยวข้อง ได้แก่

OPTION_S46_BR, OPTION_S46_DHCP4O6_SADDR_HINT และ OPTION_S46_DHCP4O6 _SADDR โดยแต่ละ option มีวัตถุประสงค์และรูปแบบดังต่อไปนี้



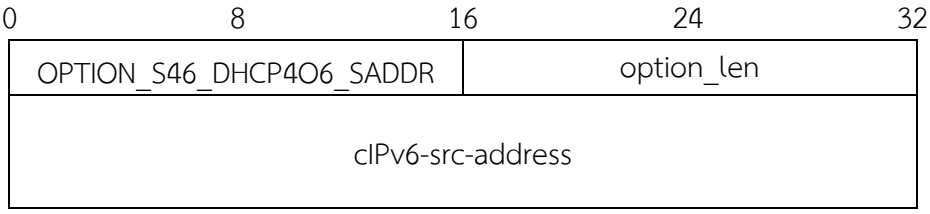

รูปที่ 2-16 รูปแบบ DHCPv6 OPTION_S46_BR

จากรูปที่ 2-16 รูปแบบ DHCPv6 OPTION_S46_BR มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต และ BR-IPv6-address ขนาด 128 บิต DHCPv6 OPTION_S46_BR ใช้สำหรับประกาศข้อมูลของ lwAFTR ซึ่งคล้ายกับ OPTION_AFTR_NAME ของ DS-lite แต่แตกต่างกันที่ DS-lite ระบุข้อมูลของ AFTR ในรูปแบบโดเมนเนม ส่วน OPTION_S46_BR ของ lw4over6 ระบุข้อมูลของ lwAFTR ในรูปแบบหมายเลข IPv6 ส่งผลให้ lwB4 ไม่จำเป็นต้องแปลงจากโดเมนเนมให้กลายเป็นหมายเลข IPv6 ซึ่งช่วยให้ lwB4 นำหมายเลข IPv6 ของ lwAFTR มากำหนดเป็นอุโมงค์สื่อสารปลายทางได้ทันที




รูปที่ 2-17 รูปแบบ DHCPv6 OPTION_S46_DHCP4O6_SADDR_HINT

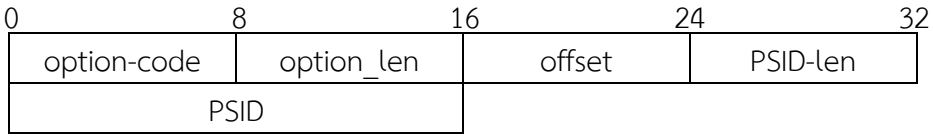
จากรูปที่ 2-17 รูปแบบ DHCPv6 OPTION_S46_DHCP4O6_SADDR_HINT มีฟิลด์ข้อมูล 4 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต, cIPv6-boundlen ขนาด 8 บิต และ cIPv6-bound-prefix ขนาดไม่เกิน 128 บิต โดย cIPv6-bound-prefix คือ IPv6 prefix ของ lwB4 ซึ่ง DHCP 4O6 Server แนะนำให้กับ lwB4 เพื่อเลือกใช้ในขั้นตอนการร้องขอหมายเลข IPv4 และ port-set




รูปที่ 2-18 รูปแบบ DHCPv6 OPTION_S46_DHCP4O6_SADDR

จากรูปที่ 2-18 รูปแบบ DHCPv6 OPTION_S46_DHCP4O6_SADDR มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต และ cIPv6-src-address ขนาด 128 บิต โดย cIPv4-src-address คือหมายเลข IPv6 ของ lwB4 ซึ่ง lwB4 เลือกเพื่อใช้ในการร้องขอหมายเลข IPv4 และ port-set lwB4 ใช้ OPTION_S46_DHCP4O6_SADDR เพื่อส่งข้อมูลหมายเลข IPv6 ของ lwB4 ให้กับ DHCP 4O6 Server

นอกจากนี้ การส่งข้อมูลที่จำเป็นไปยัง lwB4 ด้วย DHCPv4 over DHCPv6 ต้องการ DHCPv4 OPTION ใหม่ เรียกว่า “OPTION_V4_PORTPARAMS” เพื่อใช้ส่งข้อมูล port-set สำหรับหมายเลข IPv4 เพิ่มเติม [18] รูปแบบ DHCPv4 OPTION_V4_PORTPARAMS มีรายละเอียดดังแสดงในรูปที่ 2-19




รูปที่ 2-19 รูปแบบ DHCPv4 OPTION_V4_PORTPARAMS

จากรูปที่ 2-19 รูปแบบ DHCPv4 OPTION_V4_PORTPARAMS มีฟิลด์ข้อมูล 5 ฟิลด์ ประกอบด้วย option_code ขนาด 8 บิต, option_length ขนาด 8 บิต, offset ขนาด 8 บิต, PSID-len ขนาด 8 บิต, และ PSID ขนาด 16 บิต ฟิลด์ข้อมูลของ OPTION_V4_PORTPARAMS ถูกกำหนดตามข้อมูลของ port mapping algorithm ซึ่งสามารถศึกษารายละเอียดเพิ่มเติมได้ในหัวข้อ 2.2.4

- offset: ถูกใช้สำหรับระบุจำนวนบิตของพอร์ตที่ต้องการหลีกเลี่ยงซึ่งสามารถกำหนดค่าตั้งแต่ 0 - 15 offset ถูกใช้เพื่อหลีกเลี่ยงการนำพอร์ตของระบบซึ่งถูกลงทะเบียนแล้วมาใช้ เช่น พอร์ต 0-1023
- PSID-len: ถูกใช้สำหรับระบุจำนวนบิตของ PSID
- PSID: ถูกใช้สำหรับระบุค่าของพอร์ตซึ่งสามารถนำไปกำหนดเพื่อใช้งานได้เฉพาะ lwB4 ซึ่งได้รับการจัดสรรไว้เท่านั้น นอกจากนั้น lwB4 ที่ได้รับ public IPv4 address เดียวกันต้องมี PSID ที่แตกต่างกัน

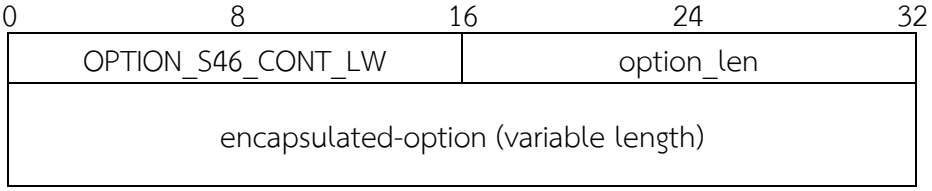
lwB4 เริ่มต้นการจัดหาข้อมูลของ lw4over6 โดยดำเนินการร้องขอข้อมูลของ DHCPv6 OPTION_DHCPv4_O_DHCPv6_SERVER ไปยัง DHCPv6 server จากนั้น DHCPv6 server ตอบกลับข้อมูลของ DHCP 4O6 Server เมื่อ lwB4 ได้รับข้อมูลของ DHCP 4O6 Server lwB4 ดำเนินการร้องขอข้อมูลของ DHCPv6 OPTION_S46_CONT_DHCP4O6 ไปยัง DHCP 4O6 Server โดยระบุ OPTION_CLIENTID และ OPTION_IA_NA เพื่อบ่งบอกเอกลักษณ์ของ lwB4 จากนั้น DHCP 4O6 Server ตอบกลับข้อมูล DHCPv6 OPTION_S46_CONT_DHCP4O6 ซึ่งภายในบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR_HINT จากนั้น lwB4 จึงส่ง DHCPv6 OPTION_CLIENTID, OPTION_IA_NA และ OPTION_S46_CONT_DHCP4O6 ซึ่งภายในบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR เพื่อลงทะเบียน DUID และ IAID

คู่กับหมายเลข IPv6 ของ lwB4 ที่จะใช้ในการร้องขอหมายเลข IPv4 และ port-set ท้ายที่สุด DHCP 4O6 Server ตอบกลับข้อมูล DHCPv6 OPTION_S46_CONT_DHCP4O6 ซึ่งภายในบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR กลับมาจึงเสร็จสิ้นการลงทะเบียนหมายเลข IPv6 ของ lwB4

ในการดำเนินการร้องขอข้อมูล IPv4 ผ่านกระบวนการ DHCPv4 over DHCPv6 lwB4 ร้องขอ DHCPv4 OPTON_V4_PORTPARAMS ไปยัง DHCP 4O6 Server โดยระบุ Client Identifier (Type:255 IAID ,DUID) จากนั้น DHCP 4O6 Server นำ IAID และ DUID มาตรวจสอบกับหมายเลข IPv6 ของ lwB4 ซึ่งได้ทำการลงทะเบียนไว้ หาก IAID และ DUID ถูกต้อง DHCP 4O6 Server จะตอบกลับหมายเลข IPv4 และ port-set ให้กับ lwB4 หลังจากนั้น lwB4 นำ public IPv4 address และ port-set มาใช้สำหรับดำเนินการ NAT ต่อไป หลังจากได้รับข้อมูลของ IPv4 lwB4 เริ่มต้นประกาศ private IPv4 address เพื่อนำไปใช้ภายในเครือข่ายของผู้ใช้บริการ ในกรณีที่การร้องขอข้อมูล IPv4 ดำเนินการผ่าน multicast lwAFTR ต้องอยู่ภายในเส้นทางการแลกเปลี่ยนข้อมูลระหว่าง lwB4 และ DHCP 4O6 Server เพื่อติดตามและปรับปรุงข้อมูลหมายเลข IPv4 และ port-set ให้เป็นปัจจุบัน แต่ในกรณีที่การร้องขอข้อมูล IPv4 ดำเนินการผ่าน unicast DHCP 4O6 Server ต้องส่งข้อมูลการจัดสรร IPv4 และ port-set ให้กับ lwAFTR โดยตรง

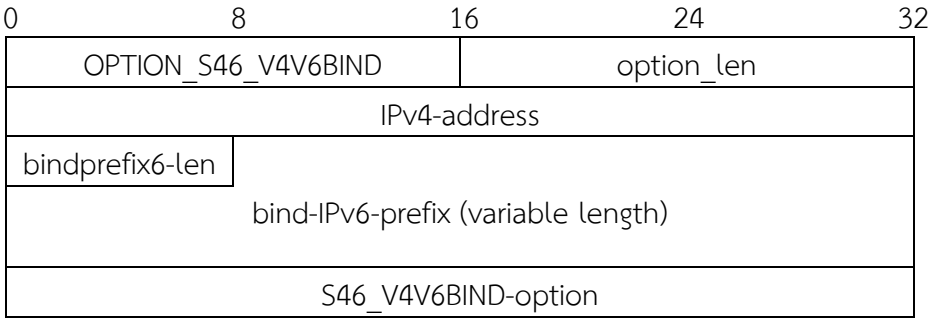
- การจัดหาข้อมูล lw4over6 ด้วย DHCPv6

สำหรับการส่งข้อมูลที่จำเป็นไปยัง lwB4 ด้วย DHCPv6 lw4over6 ได้นิยาม DHCPv6 option ซึ่งเรียกว่า “OPTION_S46_CONT_LW” เพื่อใช้สำหรับส่งข้อมูลทั้งหมดให้กับ lwB4 รูปแบบ DHCPv6 OPTION_S46_CONT_LW มีรายละเอียดดังแสดงในรูปที่ 2-20




รูปที่ 2-20 รูปแบบ DHCPv6 OPTION_S46_CONT_LW

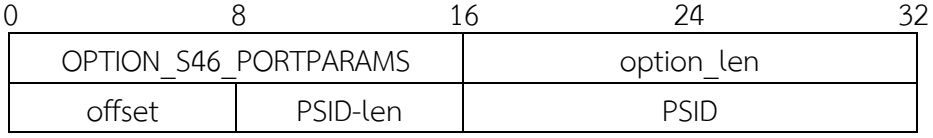
จากรูปที่ 2-20 รูปแบบ DHCPv6 OPTION_S46_CONT_LW มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต และ encapsulated-option แต่เนื่องจากกระบวนการร่างของ lw4over6 ยังไม่แล้วเสร็จ DHCPv6 OPTION_S4_CONT_LW จึงยังไม่มีการกำหนดค่าที่แน่นอนให้กับ option_code นอกจากนั้นภายใน encapsulated-option ยังประกอบด้วย option ที่เกี่ยวข้องกับ lw4over6 3 option ด้วยกัน ได้แก่ OPTION_S46_BR (แสดงรายละเอียดไปในรูปที่ 2-16), OPTION_S46_V4V6BIND และ OPTION_S46_PORTPARAMS โดยแต่ละ option มีวัตถุประสงค์และรูปแบบดังต่อไปนี้




รูปที่ 2-21 รูปแบบ DHCPv6 OPTION_S46_V4V6BIND

จากรูปที่ 2-21 รูปแบบ DHCPv6 OPTION_S46_V4V6BIND มีฟิลด์ข้อมูล 6 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต, IPv4-address ขนาด 32 บิต, bindprefix6-len ขนาด 8 บิต, bind-IPv6-prefix ขนาดไม่เกิน 128 บิต และ S46 _V4V6BIND-option ซึ่งสามารถปรับขนาดได้ (S46_V4V6BIND-option อาจมีหรือไม่มีก็ได้)

- IPv4-address: ถูกใช้สำหรับระบุ public IPv4 address เพื่อจัดสรรให้กับ lwB4
- bindprefix-len: ถูกใช้สำหรับระบุขนาดของ IPv6 prefix ของฟิลด์ข้อมูล bind-IPv6-prefix
- bind-IPv6-prefix: ถูกใช้สำหรับระบุ IPv6 prefix ของ lwB4 เพื่อตรวจสอบความถูกต้องของ IPv6 prefix ที่ใช้เป็นอุโมงค์สื่อสาร ในกรณีที่ขนาดของข้อมูล IPv6 prefix หารด้วย 8 ไม่ลงตัวต้องดำเนินการเติมบิตที่มีค่า 0 ต่อท้าย (Padding)
- S46_V4V6BIND-option: ถูกใช้สำหรับระบุข้อมูลอื่นเพิ่มเติมเพื่อให้สามารถกำหนดความสัมพันธ์ระหว่างหมายเลข IPv4 และ IPv6 เช่น ข้อมูล port-set (OPTION_S46_POTRPARAMS) เป็นต้น




รูปที่ 2-22 รูปแบบ DHCPv6 OPTION_S46_PORTPARAMS

จากรูปที่ 2-22 รูปแบบ DHCPv6 OPTION_S46_PORTPARAMS มีฟิลด์ข้อมูล 5 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต, offset ขนาด 8 บิต, PSID-len ขนาด 8 บิต และ PSID ขนาด 16 บิต ชนิดฟิลด์ข้อมูลของ OPTION_S46 _PORTPARAMS เหมือนกับ DHCPv4 OPTION_V4_PORTPARAMS ทุกประการ ยกเว้นเพียงขนาดของ option_code และ option_length เท่านั้น

- offset: ถูกใช้สำหรับระบุจำนวนบิตของพอร์ตที่ต้องการหลีกเลี่ยงซึ่งสามารถกำหนดค่าตั้งแต่ 0 - 15 offset ถูกใช้เพื่อหลีกเลี่ยงการนำพอร์ตของระบบซึ่งถูกลงทะเบียนแล้วมาใช้ ค่าโดยปริยายของ offset มีค่าเท่ากับ 6 (เพื่อหลีกเลี่ยงพอร์ต 0-1023)
- PSID-len: ถูกใช้สำหรับระบุจำนวนบิตของ PSID
- PSID: ถูกใช้สำหรับระบุค่าของพอร์ตซึ่งสามารถนำไปกำหนดเพื่อใช้งานได้เฉพาะ lwB4 ซึ่งได้รับการจัดสรรไว้เท่านั้น นอกจากนั้น lwB4 ที่ได้รับ public IPv4 address เดียวกันต้องมี PSID ที่แตกต่างกัน

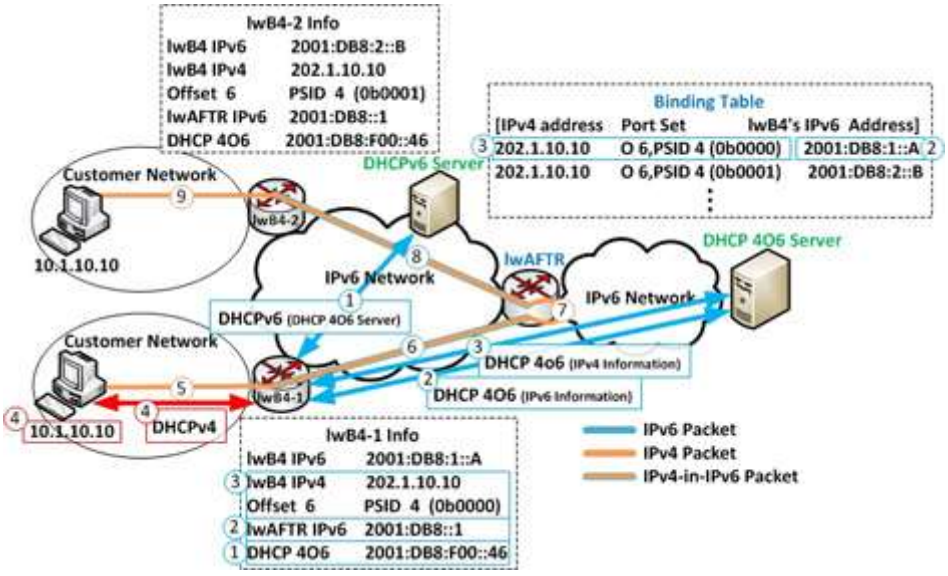
lwB4 เริ่มต้นการค้นหาข้อมูลของ lw4over6 โดยดำเนินการร้องขอข้อมูลของ DHCPv6 OPTION_S46_CONT_LW ไปยัง DHCPv6 server ผ่าน multicast จากนั้น DHCPv6 server ตอบกลับข้อมูล DHCPv6 OPTION_S46_CONT_LW ภายในบรรจุ OPTION_S46_BR, OPTION_S46_V4V6BIND และ OPTION_S46_PORTPARAMS lwAFTR ซึ่งอยู่ภายในเส้นทางการติดต่อสื่อสารระหว่าง DHCPv6 server กับ lwB4 ดำเนินการติดตามและปรับปรุงข้อมูลที่ DHCPv6 server จัดสรรให้กับ lwB4 จากนั้นเมื่อ lwB4 ได้รับ DHCPv6 OPTION_S46_CONT_LW หมายเลข IPv6 ของ lwAFTR จะถูกกำหนดเป็นอุโมงค์สื่อสารปลายทาง และ public IPv4 address และ port-set จะถูกนำมาใช้สำหรับดำเนินการ NAT ต่อไป หลังจากนั้น lwB4 จึงเริ่มต้นประกาศ private IPv4 address เพื่อเริ่มต้นใช้งาน IPv4 ภายในเครือข่ายของผู้ใช้บริการ

**2.2.3.2 Data plan**

การส่งแพ็กเก็ต IPv4 ผ่านเครือข่ายแกนหลัก IPv6 ของ lw4over6 อาศัยหลักการห่อหุ้มแพ็กเก็ตและการกำหนดปลายทางไปยัง lwAFTR เท่านั้น การสร้างการเชื่อมต่อของ lw4over6 จึงไม่จำเป็นต้องปรับปรุงข้อมูลของอุโมงค์สื่อสารปลายทางอื่นๆ เพิ่มเติม เมื่อเครื่องต้นทางซึ่งเชื่อมต่อกับ lwB4-1 ต้องการติดต่อสื่อสารกับเครื่องปลายทางซึ่งเชื่อมต่อกับ lwB4-2 แพ็กเก็ต IPv4 ถูกส่งจากเครื่องต้นทางไปยัง lwB4-1 ซึ่งทำหน้าที่เป็นเกตเวย์ เมื่อ lwB4-1 ได้รับแพ็กเก็ต IPv4 แพ็กเก็ต IPv4 ดังกล่าวจะถูกดำเนินการ NAT ด้วย public IPv4 address และ port-set ตามที่ได้รับการกำหนดไว้ก่อนหน้าและห่อหุ้มลงภายในแพ็กเก็ต IPv6 เพื่อส่งไปยัง lwAFTR โดยหมายเลข IPv6 ต้นทางถูกกำหนดเป็นหมายเลข IPv6 ของ lwB4-1 เมื่อ lwAFTR ได้รับแพ็กเก็ตที่ถูกห่อหุ้ม lwAFTR จะนำแพ็กเก็ต IPv4 ดั้งเดิมออกมาเพื่อส่งต่อไปยังเครื่องปลายทางต่อไป แต่กรณีนี้เครื่องปลายทางเชื่อมต่อกับ lwB4-2 ดังนั้นแพ็กเก็ต IPv4 ดังกล่าวจึงถูกดำเนินการโดย lwAFTR อีกครั้ง lwAFTR ค้นหาหมายเลข IPv6 ของ lwB4 ปลายทางใน Binding Table โดยใช้หมายเลข IPv4 และพอร์ตของเครื่องปลายทาง จากนั้นแพ็กเก็ตจึงถูกห่อหุ้มและส่งต่อไปยัง lwB4 ปลายทางซึ่งมี public IPv4 address และ port-set ตรงกับหมายเลข IPv4 และพอร์ตของเครื่องปลายทาง โดยในกรณีนี้คือ lwB4-2 เมื่อ lwB4-2 ได้รับแพ็กเก็ตที่ถูกห่อหุ้ม lwB4-2 จะนำแพ็กเก็ต IPv4 ดั้งเดิมออกมา จากนั้นแพ็กเก็ต IPv4 ถูกดำเนินการ NAT และส่งต่อไปยังเครื่องปลายทางต่อไป

### 2.2.3.3 หลักการทำงาน

หลักการทำงานโดยภาพรวมของ lw4over6 ซึ่งใช้การจัดหาข้อมูลด้วย DHCPv4 over DHCPv6 มีขั้นตอนดังแสดงในรูปที่ 2-23




รูปที่ 2-23 หลักการทำงานโดยภาพรวมของ lw4over6 ซึ่งจัดหาข้อมูลด้วย DHCPv4 over DHCPv6

1) lwB4-1 ร้องขอข้อมูลของ DHCP 4O6 Server ด้วย OPTION_DHCPv4_O_DHCPv6_SERVER ไปยัง DHCPv6 server และได้รับข้อมูลของ DHCPv4 over DHCPv6 กลับมาดังแสดงในตารางข้อมูลของ lwB4-1
2) lwB4-1 ร้องขอหมายเลข IPv6 ของ lwAFTR และลงทะเบียนหมาเลข IPv6 เพื่อใช้ในการร้องขอหมายเลข IPv4 และ port-set ด้วยการส่ง OPTION_S46_CONT_DHCP4O6 ไปยัง DHCP 4O6 Server จากนั้น DHCP 4O6 Server ตอบกลับหมายเลข IPv6 ของ lwAFTR และบันทึกการลงทะเบียนของ lwB4-1
3) เพื่อร้องขอหมายเลข IPv4 และ port-set lwB4-1 ส่ง DHCPv4 OPTION_V4_PORTPARAMS ด้วย DHCPv4 over DHCPv6 ไปยัง DHCP 4O6 Server จากนั้น DHCP 4O6 Server จัดสรรหมายเลข IPv4 และ port-set แล้วตอบกลับไปยัง lwB4-1
4) หลังจากที่ lwB4-1 ได้รับข้อมูล IPv4 lwB4-1 เริ่มต้นกำหนดหมายเลข IPv4 ให้กับเครื่องลูกข่ายภายในเครือข่ายของผู้ใช้งาน
5) แพ็กเก็ต IPv4 ถูกส่งจากเครื่องลูกข่ายภายในเครือข่ายของผู้ใช้งานไปยัง lwB4-1
6) lwB4-1 ดำเนินการแปลง private IPv4 address ของผู้ส่งเป็น public IPv4 address แล้วจึงห่อหุ้มแพ็กเก็ต IPv4 ภายในแพ็กเก็ต IPv6 และส่งต่อไปยัง lwAFTR

7) lwAFTR จะนำแพ็กเก็ต IPv4 ดั้งเดิมภายในแพ็กเก็ต IPv6 ออกมา แล้วจึงส่งต่อไปยังเครื่องปลายทาง
8) แต่ในกรณีนี้เครื่องปลายทางอยู่ภายในเครือข่ายของผู้ใช้งาน 2 แพ็กเก็ต IPv4 จึงถูกห่อหุ้มภายในแพ็กเก็ต IPv6 แล้วส่งต่อไปยัง lwB4-2
9) เมื่อ lwB4-2 ได้รับแพ็กเก็ตที่ถูกห่อหุ้ม lwB4-2 จะนำแพ็กเก็ต IPv4 ดั้งเดิมออกมาและแปลง public IPv4 address ของผู้รับเป็น private IPv4 address จากนั้นแพ็กเก็ต IPv4 จึงถูกส่งต่อไปยังเครื่องปลายทางต่อไป

### 2.2.4 IPv4 Residual Deployment via IPv6 (4rd)

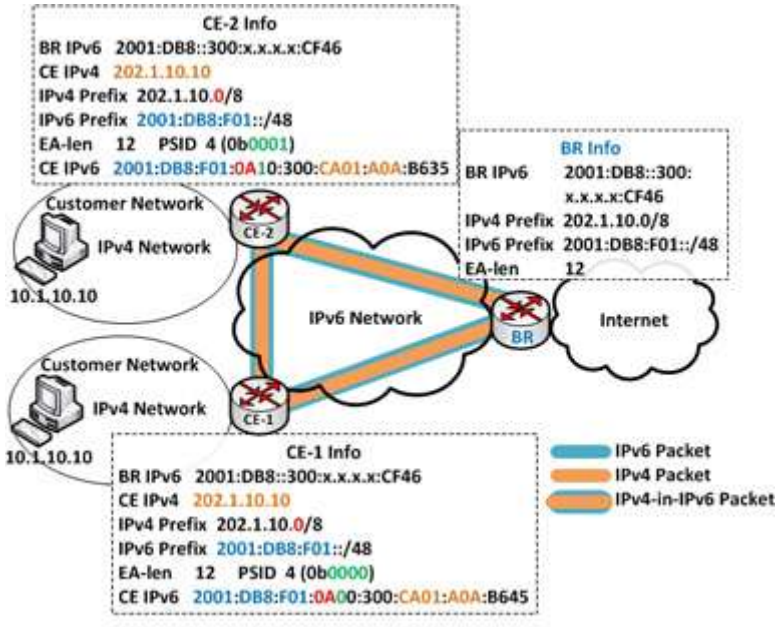
กระบวนการเปลี่ยนถ่ายแบบ stateless ในยุคบุกเบิกคงหนีไม่พ้นกระบวนการ 6to4 หลักการทำงานของ 6to4 ใช้การกำหนดหมายเลข IPv6 โดยบรรจุหมายเลข IPv4 ไว้ภายในเพื่อสร้างความสัมพันธ์ระหว่างหมายเลข IPv6 และหมายเลข IPv4 IPv6 prefix ของ 6to4 สร้างจาก 6to4 prefix 2002::/16, IPv4 address 32 บิต และ Subnet-ID 16 บิต ซึ่งการกำหนดหมายเลข IPv6 เช่นนี้ทำให้ 6to4 relay ทราบได้ทันทีว่า หมายเลข IPv6 ปลายทางที่มีรูปแบบตรงกับหมายเลข IPv6 ของ 6to4 ต้องส่งต่อไปยังเครื่องปลายทางใดต่อไป ในการส่งแพ็กเก็ต IPv6 ด้วยอุโมงค์สื่อสาร 6to4 relay ต้องทำการห่อหุ้มแพ็กเก็ต IPv6 ภายในแพ็กเก็ต IPv4 พร้อมกับกำหนดหมายเลข IPv4 ปลายทางจากหมายเลข IPv4 ที่ถูกบรรจุอยู่ภายในหมายเลข IPv6 การออกแบบการทำงานของ 6to4 ถือได้ว่าเป็นต้นแบบของอุโมงค์สื่อสารอัตโนมัติแบบ stateless ก็ว่าได้ เพื่อเพิ่มความยืดหยุ่นในการกำหนดหมายเลข IPv6 กระบวนการเปลี่ยนถ่ายใหม่จึงถูกออกแบบให้สามารถใช้งานร่วมกับ IPv6 prefix ทั่วไปนอกเหนือจาก 6to4 prefix กระบวนการเปลี่ยนถ่ายดังกล่าวถูกเรียก “6rd” 6rd เป็นกระบวนการที่พัฒนาต่อยอดจาก 6to4 โดยออกแบบให้สามารถกำหนดกฎสำหรับการสร้างอุโมงค์สื่อสารได้เอง ซึ่งกฎเหล่านี้จะแตกต่างกันไปตามการออกแบบของผู้ให้บริการ 6rd อนุญาตให้ผู้ให้บริการสามารถกำหนดจำนวนบิตในแต่ละส่วนของหมายเลข IPv6 ได้ ส่งผลให้สามารถสร้างกฎที่เหมาะสมกับหมายเลข IPv4 และหมายเลข IPv6 ของแต่ละเครือข่าย ต่อมาเมื่อระบบเครือข่าย IPv6 มีขนาดใหญ่ขึ้นและมีความขาดแคลนหมายเลข IPv4 เพิ่มมากขึ้น หลักการทำงาน 6rd จึงถูกนำมาใช้อีกครั้ง แต่เป็นในทิศทางที่กลับกัน โดยเปลี่ยนจากการสร้างอุโมงค์สื่อสารด้วย IPv4 เป็นการสร้างอุโมงค์สื่อสารด้วย IPv6 แทน กระบวนการดังกล่าวถูกเรียกว่า “4rd” เพื่อใช้เป็นทางเลือกสำหรับให้บริการ IPv4 [19]

4rd เป็นชื่อย่อของ IPv4 Residual Deployment via IPv6 ซึ่งเป็นการสร้างอุโมงค์สื่อสารอัตโนมัติ เพื่อที่จะกระจายหมายเลข IPv4 ที่เหลืออยู่ไปยังเครือข่ายของผู้ใช้บริการผ่านเครือข่าย IPv6 อุโมงค์สื่อสารของ 4rd ถูกสร้างเพื่อเชื่อมต่อระหว่างอุปกรณ์เกตเวย์ของฝั่งผู้ใช้งานและอุปกรณ์ของผู้ให้บริการโดยสร้างผ่านเครือข่ายของผู้ให้บริการ โดยอุปกรณ์เกตเวย์ของฝั่งผู้ใช้งานถูกเรียกว่า “Customer Edge” (CE) และอุปกรณ์ของผู้ให้บริการถูกเรียกว่า “Border Relay” (BR) การกระจายหมายเลข IPv4 ไปยังเครือข่ายของผู้ใช้บริการทำให้เครื่องที่ได้รับหมายเลข IPv4 รองรับการใช้งานทั้ง IPv4 และ IPv6 ยิ่งกว่านั้นการให้บริการ IPv4 ผ่าน 4rd ยังช่วยลดความต้องการหมายเลข IPv4 สำหรับอุปกรณ์เครือข่ายที่อยู่ภายในเครือข่ายของผู้ให้บริการ เพราะแพ็กเก็ต IPv4

จะถูกห่อหุ้มอยู่ภายในแพ็กเก็ต IPv6 เพื่อส่งผ่านเครือข่ายแกนหลักของผู้ให้บริการ ดังนั้นเครือข่ายแกนหลักของผู้ให้บริการรองรับการใช้งานเฉพาะ IPv6 ก็เพียงพอแล้ว อุปกรณ์ในเครือข่ายของผู้ให้บริการจึงมีการประมวลผลลดลง เนื่องจากทำการประมวลผลเพียงโพรโตคอลเดียว

4rd ซึ่งออกแบบมาเพื่อใช้งานร่วนกับหมายเลข IPv4 ที่เหลือจึงมีโอกาสที่หมายเลข IPv4 จะมีช่วงที่ไม่ต่อเนื่องกัน และมีขนาดของแต่ละช่วงไม่เท่ากัน ดังนั้น 4rd จึงถูกออกแบบมาให้รองรับกฎมากกว่า 1 กฎเพื่อใช้ในการจับคู่ระหว่างหมายเลข IPv4 ในแต่ละช่วงกับหมายเลข IPv6 การรองรับกฎหลายข้อทำให้กฎแต่ละข้อสามารถกำหนดได้อย่างอิสระ และสามารถออกแบบให้เหมาะสมกับหมายเลข IPv4 ที่เหลือในแต่ละช่วงได้อย่างเหมาะสมส่งผลให้สามารถนำหมายเลข IPv4 ที่เหลือมาจัดสรรได้อย่างมีประสิทธิภาพ ยิ่งกว่านั้น 4rd สามารถใช้งานควบคู่กับหมายเลข IPv4 ได้หลายรูปแบบ เช่น public IPv4 prefix (IPv4 subnet) , public IPv4 address , shared public IPv4 with port-set และ no IPv4 address by NAT64+ สำหรับการสร้างการเชื่อมต่อ IPv4 ของ 4rd สามารถดูรายละเอียดได้จากรูปที่ 2-24

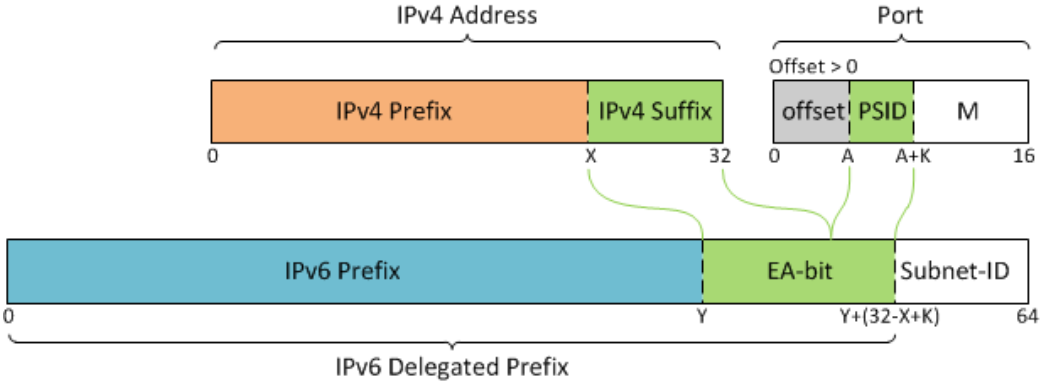
รูปแบบหลักในการให้บริการของ 4rd เป็นการกระจาย public IPv4 address ให้กับ CE โดยตรง ซึ่งการใช้งานในรูปแบบนี้ CE จะต้องรองรับ NAT เพื่อที่จะทำการแปลงแพ็กเก็ตก่อนดำเนินการห่อหุ้มลงในแพ็กเก็ต IPv6 เพื่อส่งไปยัง BR การให้บริการในรูปแบบนี้สามารถรองรับเครื่องภายในเครือข่ายของผู้ใช้งานได้จำนวนมาก แต่ทั้งนี้ก็ขึ้นอยู่กับความสามารถในการดำเนินการ NAT ปัญหาสำหรับการให้บริการในรูปแบบนี้มีโอกาสเกิดขึ้น เมื่อมีจำนวน CE เพิ่มมากขึ้น ผู้ให้บริการอาจต้องมีการจัดสรร public IPv4 address ใหม่ เนื่องจาก public IPv4 address ที่สงวนไว้อาจไม่เพียงพอต่อความต้องการของผู้ใช้งาน ส่วนรูปแบบเสริมในการให้บริการคือ no IPv4 address by NAT64+ การให้บริการด้วยรูปแบบนี้คล้ายกับ LSN เนื่องจากผู้ให้บริการดำเนินการจัดสรร public IPv4 address และพอร์ตให้กับผู้ใช้งาน ผู้ให้บริการจึงต้องรับภาระเกี่ยวกับการดำเนินการ NAT และการบันทึกข้อมูลการใช้งานเป็นจำนวนมาก




รูปที่ 2-24 การสร้างการเชื่อมต่อ IPv4 ของ 4rd

### 2.2.4.1 4rd Mapping Rule

หลักการจับคู่ระหว่างหมายเลข IPv4 และพอร์ตกับหมายเลข IPv6 ใช้การกำหนดส่วนของ IPv4 prefix และ IPv6 prefix ที่เหมือนกันไว้ล่วงหน้า และนำเพียงส่วนที่ต่างกันมาของหมายเลข IPv4 (IPv4 suffix) และ port-set ID บรรจุต่อจาก IPv6 prefix เพื่อสร้างความสัมพันธ์ระหว่างหมายเลข IPv4 และพอร์ตกับหมายเลข IPv6 เมื่อต้องการดำเนินการเปลี่ยนจากหมายเลข IPv4 และพอร์ตเป็นหมายเลข IPv6 หรือเปลี่ยนจากหมายเลข IPv6 เป็นหมายเลข IPv4 และพอร์ต สามารถดำเนินการโดยนำส่วนที่ต่างกันของหมายเลข IPv4 และพอร์ตหรือหมายเลข IPv6 มาต่อท้าย IPv4 prefix หรือ IPv6 prefix ก็จะได้รับหมายเลข IPv4 และ IPv6 ที่ถูกต้อง โดยหมายเลข IPv4, พอร์ต และหมายเลข IPv6 ถูกแบ่งออกเป็นส่วนย่อยดังแสดงในรูปที่ 2-25



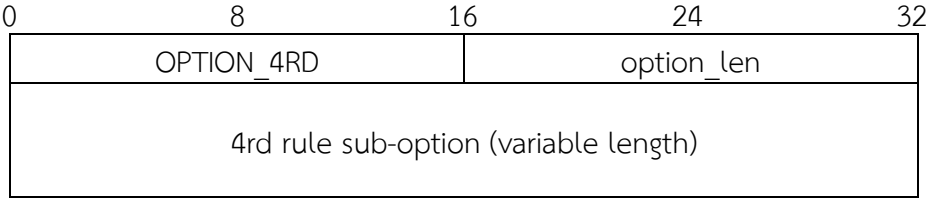

รูปที่ 2-25 หลักการจับคู่ระหว่างหมายเลข IPv4 และพอร์ตกับหมายเลข IPv6

จากรูปที่ 2-25 แสดงให้เห็นถึงรายละเอียดในการแบ่งหมายเลข IPv4, พอร์ต และหมายเลข IPv6 หมายเลข IPv4 ถูกแบ่งออกเป็น 2 ส่วน ได้แก่ IPv4 prefix และ IPv4 suffix ส่วนพอร์ตถูกแบ่งออกเป็น 3 ส่วนตาม port mapping algorithm ได้แก่ offset (A), port-set ID หรือ PSID (K) และ M bits โดยค่าของ offset ถูกกำหนดให้มีค่ามากกว่า 0 เท่านั้นเพื่อไม่อนุญาตให้นำพอร์ตบางส่วนมาใช้งาน ยกตัวอย่างเช่น well-known port (พอร์ต 0 - 1023) จำนวนบิตโดยปริยายของ offset ของ 4rd กำหนดไว้เท่ากับ 4 บิต ซึ่งหมายความว่าไม่อนุญาตให้นำพอร์ต 0 – 4095 มาใช้งาน แต่ 4rd ก็สามารถกำหนดจำนวนบิตของ offset เท่ากับ 0 เพื่อให้สามารถนำพอร์ตทุกพอร์ตมาใช้ได้ในกรณีที่มีการกำหนด WKPs authorized port-set ID เป็นบิตที่ใช้สำหรับบ่งบอกช่วงของพอร์ตที่อุปกรณ์ฝั่งผู้ใช้งานได้รับการจัดสรร ซึ่งอุปกรณ์ฝั่งผู้ใช้งานที่ได้รับหมายเลข IPv4 เหมือนกันต้องมี port-set ID ที่แตกต่างกันเพื่อป้องกันการใช้งานพอร์ตซ้ำซ้อนกัน ในกรณีที่มีจำนวนบิตของ port-set ID มาก จำนวนพอร์ตที่สามารถนำมาใช้งานก็ยิ่งน้อยลง port-set ID จึงบ่งบอกถึงอัตราการแบ่งปันจำนวนพอร์ตอีกด้วย M bits เป็นส่วนที่สามารถกำหนดเป็นค่าใดก็ได้ ดังนั้นเมื่อต้องการใช้งานพอร์ต อุปกรณ์ฝั่งผู้ใช้งานต้องกำหนดค่าของ port-set ID ตามที่ได้รับการกำหนดและสามารถเลือกกำหนดค่าใดก็ได้ สำหรับ M bits และ offset แต่ค่า offset ต้องมีค่ามากกว่า 0 ส่วนหมายเลข IPv6 ถูกแบ่งออกเป็น 4 ส่วน ได้แก่ IPv6 prefix, EA bits, subnet-ID และ interface ID EA bits ได้จากการรวม IPv4 suffix กับ port-set ID ตามลำดับ เมื่อนำ IPv6 prefix รวมกับ EA bits จะถูกเรียกว่า “IPv6 delegated prefix” ในกรณีที่ IPv6 delegated

prefix น้อยกว่า /64 subnet-ID ต้องถูกกำหนดขึ้นเพื่อให้ prefix มีขนาดเท่ากับ /64 สำหรับ interface ID ของ 4rd ถูกแบ่งออกเป็น 3 ส่วนย่อย ได้แก่ 4rd tag, IPv4 address และ CNP โดย 4rd tag มีขนาด 16 บิต และมีค่าเท่ากับ 0x0300 IPv4 address ขนาด 32 บิต และ CNP ย่อมาจาก Checksum-Neutrality Preserver ซึ่ง CNP มีขนาด 16 บิต และมีค่าเท่ากับ 1's complement ของผลรวมขนาด 16 บิตของหมายเลข IPv6 จำนวน 80 บิตแรกซึ่งประกอบไปด้วย IPv6 prefix, EA bits, subnet-ID และ 4rd tag

**2.2.4.2 Control plan**

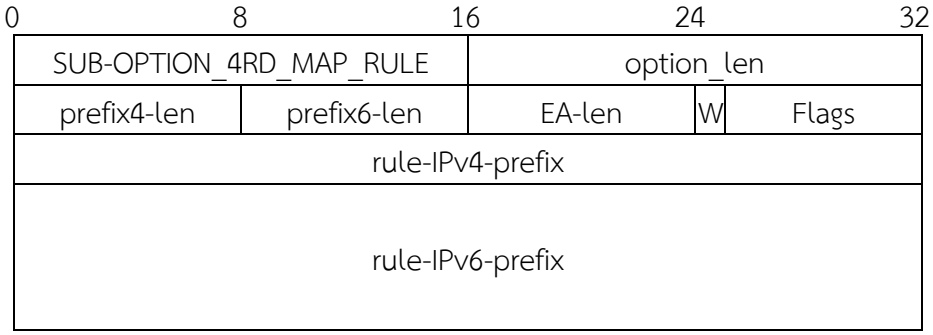
การควบคุมการให้บริการการเชื่อมต่อ IPv4 ผ่านเครือข่ายแกนหลัก IPv6 ของ 4rd นั้นควบคุมเฉพาะช่วงเริ่มต้นในการสร้างการเชื่อมต่อเท่านั้นเช่นเดียวกับ DS-lite และ lw4over6 แม้ว่า 4rd รองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง แต่ 4rd ก็ไม่จำเป็นต้องควบคุมการเชื่อมต่อตลอดเวลา เนื่องจาก 4rd กำหนดกฎในการจับคู่ระหว่างหมายเลข IPv4 และหมายเลข IPv6 ไว้ล่วงหน้า เมื่อมีเครือข่ายผู้ใช้งานใหม่เพิ่มขึ้นมาต้องกำหนดหมายเลข IPv4 และหมายเลข IPv6 ตามกฎที่ได้กำหนดเท่านั้นส่งผลให้สามารถแปลงไปมาระหว่างหมายเลข IPv4 และหมายเลข IPv6 โดยไม่จำเป็นต้องใช้ข้อมูลอื่นนอกเหนือ 4rd rule ซึ่ง 4rd ได้นิยาม DHCPv6 OPTOIN_4RD เพื่อใช้สำหรับส่งข้อมูล 4rd rule ให้กับ CE รูปแบบ DHCPv6 OPTION_4RD มีรายละเอียดดังรูปที่ 2-26




รูปที่ 2-26 รูปแบบ DHCPv6 OPTION_4RD

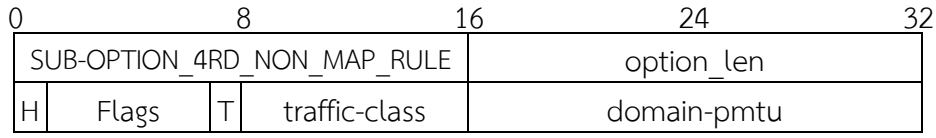
จากรูปที่ 2-26 รูปแบบ DHCPv6 OPTION_4RD มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต และ 4rd rule sub-option แต่เนื่องจากกระบวนการร่างของ 4rd ยังไม่แล้วเสร็จ DHCPv6 OPTION_4RD จึงยังไม่มีการกำหนดค่าที่แน่นอนให้กับ option_code ในส่วนของ 4rd rule sub-option ประกอบด้วย sub-option ที่เกี่ยวข้องอีก 2 sub-option ด้วยกัน ได้แก่ SUB-OPTION_4RD_MAP_RULE และ SUB-OPTION_4RD_NON_MAP_RULE โดย SUB-OPTION_4RD_MAP_RULE สามารถมีได้มากกว่าหนึ่งกฎ แต่ SUB-OPTION_4RD_NON_MAP_RULE ต้องมีเพียงหนึ่งกฎเท่านั้น ซึ่งแต่ละ SUB-OPTION มีวัตถุประสงค์และรูปแบบดังต่อไปนี้

จากรูปที่ 2-27 รูปแบบ DHCPv6 SUB-OPTION_4RD_MAP_RULE มีฟิลด์ข้อมูล 8 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต, prefix4-len ขนาด 8 บิต, prefix6-len ขนาด 8 บิต, EA-len ขนาด 8 บิต, Flags ขนาด 8 บิต, rule-IPv4-prefix ขนาดไม่เกิน 32 บิต และ rule-IPv6-prefix ขนาดไม่เกิน 128 บิต




รูปที่ 2-27 รูปแบบ DHCPv6 SUB-OPTION_4RD_MAP_RULE

- prefix4-len: ใช้สำหรับระบุขนาดของ IPv4 prefix ภายในฟิลด์ข้อมูล rule-IPv4-prefix
- prefix6-len: ใช้สำหรับระบุขนาดของ IPv6 prefix ภายในฟิลด์ข้อมูล rule-IPv6-prefix
- EA-len: ใช้สำหรับระบุจำนวนบิตของฟิลด์ EA
- Flags: ใช้สำหรับระบุรายละเอียดของ 4rd rule เพิ่มเติม ในขณะนี้นิยามเพียง W flags ซึ่งใช้กำหนด WKP authorized
- rule-IPv4-prefix: ใช้สำหรับระบุ IPv4 prefix เพื่อใช้ในกฎการจับคู่ระหว่างหมายเลข IPv4 และหมายเลข IPv6
- rule-IPv6-prefix: ใช้สำหรับระบุ IPv6 prefix เพื่อใช้ในกฎการจับคู่ระหว่างหมายเลข IPv4 และหมายเลข IPv6




รูปที่ 2-28 รูปแบบ DHCPv6 SUB-OPTION_S46_PORTPARAMS

จากรูปที่ 2-28 รูปแบบ DHCPv6 SUB-OPTION_4RD_NON_MAP_RULE มีฟิลด์ข้อมูล 5 ฟิลด์ ประกอบด้วย option_code ขนาด 16 บิต, option_length ขนาด 16 บิต, Flags ขนาด 8 บิต, traffic-class ขนาด 8 บิต และ domain-pmtu ขนาด 16 บิต

- Flags: ใช้สำหรับระบุคุณสมบัติของ 4rd เพิ่มเติม ในขณะนี้นิยามเพียง H flags และ T flags โดย H flags ใช้กำหนดรูปแบบการเชื่อมต่อเป็น Hub & Spoke และ T flags ใช้สำหรับอนุญาตให้สามารถกำหนด traffic class สำหรับอุโมงค์สื่อสาร
- traffic-class: ใช้สำหรับระบุค่าของ traffic class สำหรับอุโมงค์สื่อสาร
- domain-pmtu: ใช้สำหรับระบุค่า PMTU โดยต้องมีค่าอย่างน้อย 1,280

CE เริ่มต้นการทำงานโดยดำเนินการร้องขอข้อมูลของ DHCPv6 OPTION_4RD ไปยัง DHCPv6 server เมื่อ DHCPv6 server ตอบกลับข้อมูล DHCPv6 OPTION_4RD ซึ่งประกอบด้วย SUB-OPTION_4RD_MAP_RULE และ SUB-OPTION_4RD_NON_MAP_RULE โดย 4rd rule ที่ DHCPv6 server ตอบกลับต้องสอดคล้องกับ 4rd rule ซึ่งถูกกำหนดไว้ล่วงหน้าใน BR จากนั้นเมื่อ CE ได้รับ DHCPv6 OPTION_4RD CE นำ 4rd rule ที่ได้รับมาบันทึกและใช้เสมือนเป็นตารางกำหนดเส้นทางเพื่อห่อหุ้มและส่งแพ็กเก็ต IPv4 ยิ่งกว่านั้น CE นำ public IPv4 address และ port-set ที่ได้รับการจัดสรรมาใช้สำหรับดำเนินการ NAT ต่อไป หลังจากนั้น CE จึงเริ่มต้นประกาศ private IPv4 address เพื่อนำไปใช้ภายในเครือข่ายของผู้ใช้บริการ

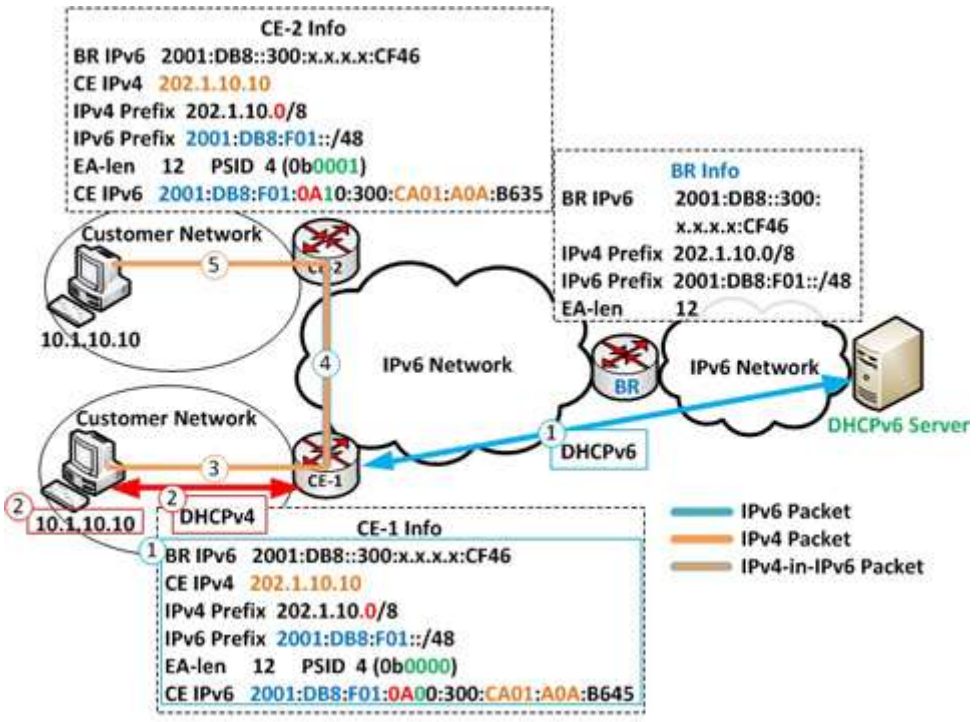
**2.2.4.3 Data plan**

การส่งแพ็กเก็ต IPv4 ผ่านเครือข่ายแกนหลัก IPv6 ของ 4rd อาศัยหลักการห่อหุ้มแพ็กเก็ตและการกำหนดปลายทางตาม 4rd rule โดย CE ต้องใช้งานหมายเลข IPv4 และหมายเลข IPv6 ตามที่กำหนดใน 4rd rule อย่างเคร่งครัด การสร้างการเชื่อมต่อของ 4rd จึงไม่จำเป็นต้องปรับปรุงข้อมูลของอุโมงค์สื่อสารปลายทางอื่นๆ เพิ่มเติม เมื่อเครื่องต้นทางซึ่งเชื่อมต่อกับ CE-1 ต้องการติดต่อสื่อสารกับเครื่องปลายทางซึ่งเชื่อมต่อกับ CE-2 แพ็กเก็ต IPv4 ถูกส่งจากเครื่องต้นทางไปยัง CE-1 ซึ่งทำหน้าที่เป็นเกตเวย์ เมื่อ CE-1 ได้รับแพ็กเก็ต IPv4 แพ็กเก็ต IPv4 ดังกล่าวจะถูกดำเนินการ NAT ด้วย public IPv4 address และ port-set ตามที่ได้รับการกำหนดไว้ก่อนหน้าด้วย 4rd rule จากนั้น CE-1 นำหมายเลข IPv4 ปลายทางมาเทียบกับ 4rd rule โดยใช้กฎ longest matching คล้ายกับที่ใช้กับตารางกำหนดเส้นทาง ในกรณีนี้เครื่องปลายทางเชื่อมต่อกับ CE-2 ดังนั้น 4rd rule ที่ตรงกับอุโมงค์สื่อสารปลายทางคือ CE mapping rule เมื่อดำเนินการเปลี่ยนหมายเลข IPv4 และพอร์ตเป็นหมายเลข IPv6 ตาม CE mapping rule (ซึ่งคล้ายกับตัวอย่างของ 4rd Mapping rule) CE-1 ดำเนินการส่งแพ็กเก็ตที่ถูกห่อหุ้มไปยัง CE-2 หลังจาก CE-2 ได้รับแพ็กเก็ตที่ถูกห่อหุ้ม แพ็กเก็ต IPv4 ดั้งเดิมจะถูกนำออกมาเพื่อดำเนินการ NAT และส่งต่อไปยังเครื่องปลายทางต่อไป

**2.2.4.4 หลักการทำงาน**

หลักการทำงานโดยภาพรวมของ 4rd มีขั้นตอนดังแสดงในรูปที่ 2-29

1) CE-1 ร้องขอข้อมูลหมายเลข IPv6 และ 4rd rule ด้วย DHCPv6 OPTION _4RD ไปยัง DHCPv6 server และได้รับข้อมูลหมายเลข IPv6 และ 4rd rule กลับมาดังแสดงในตารางข้อมูลของ CE-1
2) หลังจากนั้น CE-1 สามารถกำหนดหมายเลข IPv4 ให้กับเครื่องลูกข่ายภายในเครือข่ายของผู้ใช้งาน 1
3) CE-1 ได้รับแพ็กเก็ต IPv4 ที่ถูกส่งจากเครื่องลูกข่ายภายในเครือข่ายของผู้ใช้งาน




รูปที่ 2-29 หลักการทำงานโดยภาพรวมของ 4rd

4) แพ็กเก็ต IPv4 ดังกล่าวถูกดำเนินการแปลง private IPv4 address ของผู้ส่งเป็น public IPv4 address และ port-set ตามที่ CE-1 ได้รับการจัดสรร จากนั้นแพ็กเก็ต IPv4 ดังกล่าวถูกห่อหุ้มภายในแพ็กเก็ต IPv6 และส่งต่อไปยัง PE ปลายทางตาม 4rd rule ซึ่งในกรณีนี้ PE ปลายทางตรงกับ CE mapping rule ดังนั้นหมายเลข IPv6 ปลายทางที่ได้จากคำนวณด้วยหมายเลข IPv4 และพอร์ตของเครื่องปลายทางคือหมายเลข IPv6 ของ CE-2
5) เมื่อ CE-2 ได้รับแพ็กเก็ตที่ถูกห่อหุ้ม CE-2 นำแพ็กเก็ต IPv4 ดั้งเดิมออกมา จากนั้นดำเนินการแปลง public IPv4 address ของผู้รับเป็น private IPv4 address และส่งต่อไปยังเครื่องปลายทางในที่สุด

## 2.3 การทบทวนวรรณกรรม

กระบวนการเปลี่ยนถ่ายที่ถูกพัฒนาออกมาย่อมมีจุดเด่นและข้อจำกัดแตกต่างกันออกไป จากจุดเด่นเหล่านี้ทำให้มีการนำเสนอรูปแบบการให้บริการที่เหมาะของกระบวนการเปลี่ยนถ่ายสำหรับแต่ละระบบเครือข่าย [20], [21] เมื่อพิจารณากระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv4 พบว่า 6rd เป็นกระบวนการที่สามารถให้บริการได้อย่างมีประสิทธิภาพสูงสุด เนื่องจากสามารถให้บริการด้วยเส้นทางที่สั้นที่สุด และสามารถใช้งานร่วมกับ public IPv4 address และ private IPv4 address แต่เมื่อพิจารณากระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 พบว่ายังไม่สามารถสรุปอย่างชัดเจนได้ว่ากระบวนการใดมีความเหมาะสมสำหรับการให้บริการที่สุด เนื่องจากการสร้างความสัมพันธ์จากหมายเลข IPv6 ไปยังหมายเลข IPv4 ค่อนข้างทำ

ได้ยาก เพราะหมายเลข IPv6 มีจำนวนมากกว่าหมายเลข IPv4 อย่างมหาศาลและช่วงของ IPv4 อาจขาดความต่อเนื่อง และอีกหนึ่งสาเหตุคือมีการพัฒนากระบวนการเปลี่ยนถ่ายในช่วงนี้ออกมาอย่างต่อเนื่องส่งผลให้ในขณะนี้ยังไม่สามารถสรุปได้อย่างชัดเจน สำหรับกระบวนการเปลี่ยนถ่ายในช่วงนี้นิยมใช้การสร้างอุโมงค์สื่อสารด้วยการห่อหุ้มแพ็กเก็ต IPv4 ลงในแพ็กเก็ต IPv6 เพื่อใช้ในสร้างการเชื่อมต่อกับเครือข่าย IPv4 แทนการเชื่อมต่อแบบปกติ การสร้างอุโมงค์สื่อสารดังกล่าวเรียกได้ว่าเป็นการสร้างลิงค์เสมือน ข้อดีกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 คือไม่ทำการแก้ไขข้อมูลแพ็กเก็ต IPv4 ขณะที่กำลังส่งผ่านอุโมงค์สื่อสาร ทำให้ปลายอุโมงค์สื่อสารได้รับข้อมูลแพ็กเก็ต IPv4 เหมือนเดิมทุกประการ รูปแบบการเชื่อมต่อของอุโมงค์สื่อสาร สามารถแบ่งออกเป็น 2 รูปแบบ คือ 1) Mesh model สำหรับรูปแบบนี้ เราเตอร์ทุกตัวสามารถสร้างอุโมงค์สื่อสารเชื่อมต่อระหว่างกันโดยตรง ดังนั้นในการส่งต่อแพ็กเก็ต เราเตอร์จะต้องเลือกอุโมงค์สื่อสารที่เหมาะสม เพื่อส่งแพ็กเก็ตไปยังเครื่องปลายทางโดยตรง และ 2) Hub & Spokes model สำหรับรูปแบบนี้ เราเตอร์มีอุโมงค์สื่อสารเพียงอุโมงค์เดียวเท่านั้น โดยอุโมงค์สื่อสารดังกล่าวเชื่อมต่อไปยังเราเตอร์ศูนย์กลางซึ่งอยู่ในฝั่งของผู้ให้บริการ ในรูปแบบนี้แพ็กเก็ต IPv4 ทั้งหมดจะถูกส่งไปยังเราเตอร์ศูนย์กลางก่อนเสมอ ซึ่งเปรียบเสมือนเราเตอร์ศูนย์กลางทำหน้าที่เป็นเกตเวย์เราเตอร์ แต่หากแบ่งกระบวนการเปลี่ยนถ่ายตามรูปแบบของการจัดสรรหมายเลข IPv4 จะพบว่าวิธีการจัดสรรหมายเลข IPv4 ที่นิยมนำมาใช้ประกอบไปด้วย dynamic address และ static address วิธีการจัดสรรแบบ dynamic นิยมใช้ในกระบวนการเปลี่ยนถ่ายแบบ stateful [22],[23] และวิธีการจัดสรรแบบ static จะนำมาใช้ใน stateless transition [24] โดยวิธีการจัดสรรแบบ dynamic จะมีอัตราการใช้งานพอร์ตสูงกว่าเมื่อเทียบกับวิธีการจัดสรรแบบ static แต่ก็มีข้อเสียคือไม่สามารถรับประกันว่าผู้ใช้แต่ละคนจะได้รับการจัดสรรจำนวนพอร์ตอย่างเท่าเทียม นอกจากนี้กระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 ยังประสบปัญหาสำคัญเกี่ยวกับการค้นหาเส้นทางสำหรับเครือข่าย IPv4 เนื่องจากไม่สามารถประกาศเส้นทางของ IPv4 บนเครือข่าย IPv6 ได้โดยตรง การแก้ไขปัญหาดังกล่าวสามารถดำเนินการได้ 2 วิธี วิธีแรกคือการปรับปรุงโพรโตคอลกำหนดเส้นทางซึ่งแบ่งออกเป็น 2 ทางเลือก ทางเลือกที่หนึ่งคืออนุญาตให้โพรโตคอลกำหนดเส้นทาง IPv6 ประกาศ IPv6 prefix พิเศษที่บรรจุข้อมูล IPv4 อยู่ภายในเพื่อใช้ส่งข้อมูลการกำหนดเส้นทาง IPv4 และทางเลือกที่สองคืออนุญาตให้โพรโตคอลกำหนดเส้นทาง IPv4 สามารถกำหนดหมายเลข IPv6 ในฟิลด์ข้อมูล next hop ส่วนวิธีที่สองคือบรรจุข้อมูล IPv4 ลงภายในในหมายเลข IPv6 เพื่อให้ได้รับข้อมูล IPv4 จากหมายเลข IPv6 ในทันที สำหรับปัญหาเกี่ยวกับ MTU เนื่องจากการห่อหุ้มแพ็กเก็ตส่งผลให้แพ็กเก็ต IPv6 มีขนาดใหญ่ขึ้น ปัญหาจาก MTU มีวิธีแก้ไขหลายวิธี เช่น เพิ่มขนาด MTU สำหรับเครือข่าย IPv6, ลดขนาด MTU สำหรับเครือข่าย IPv4 และการแบ่งแพ็กเก็ตออกเป็นแพ็กเก็ตย่อย ดังนั้นกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 จึงถูกออกแบบออกมาอย่างหลากหลายเพื่อดึงจุดเด่นของแต่ละวิธีการออกมา กระบวนการเปลี่ยนถ่ายเหล่านี้ประกอบด้วย DS-lite, 4over6, lw4over6 และ 4rd

กระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารถูกนำมาวิเคราะห์ความสามารถในการให้บริการบนระบบเครือข่าย โดยรูปแบบของระบบเครือข่ายถูกจัดเป็น 3 กลุ่ม ได้แก่ เครือข่ายแกนหลักระหว่างผู้ให้บริการ, เครือข่ายแกนหลัก IPv6 ของผู้ให้บริการ และเครือข่ายแกน

หลัก IPv4 ของผู้ให้บริการ [25] ในการวิเคราะห์กระบวนการเปลี่ยนถ่ายเพื่อให้บริการระหว่างเครือข่ายแกนหลักระหว่างผู้ให้บริการกำหนดให้เครือข่ายแกนหลักใช้งานได้เพียง IPv6 หรือ IPv4 โพรโตคอลเดียวเท่านั้น กระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์ซึ่งรองรับใช้งานในรูปแบบ mesh model และรองรับการเปลี่ยนแปลงในการกำหนดเส้นทางตลอดเวลามีเพียงกระบวนการของ softwire mesh เท่านั้น เพราะ softwire mesh นำใช้ BGP เพื่อปรับปรุงการกำหนดเส้นทางของ IPv6 และ IPv4 ควบคู่ไปด้วยกัน สำหรับเครือข่ายแกนหลัก IPv6 ของผู้ให้บริการมีการวิเคราะห์กระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสาร 3 กระบวนการ ได้แก่ public 4over6, DS-lite และ 4rd ข้อดีของกระบวนการ public 4over6 และ DS-lite คือสามารถกำหนดหมายเลข IPv4 และหมายเลข IPv6 ได้อย่างอิสระ แต่ทั้งสองกระบวนการก็มีข้อจำกัด สำหรับ public 4over6 นั้นสามารถใช้งานเฉพาะ public IPv4 address เท่านั้น การให้บริการด้วย public 4over6 จึงต้องการ public IPv4 address สำหรับเครือข่ายผู้ใช้งานอย่างเพียงพอ สำหรับ DS-lite นั้นสามารถใช้งานเฉพาะ private IPv4 address เท่านั้น เนื่องจากกระบวนการ DS-lite กำหนดให้มีการดำเนินการ NAT โดยผู้ให้บริการ แต่กระบวนการ 4rd ไม่ได้รับผลกระทบจากปัญหาการใช้งานร่วมกับหมายเลข IPv4 ดังเช่น กระบวนการ public 4over6 และ DS-lite เพราะ 4rd สามารถทำงานร่วมกับ public IPv4 address และ private IPv4 address แต่ 4rd ต้องมีการกำหนดหมายเลข IPv4 ให้มีความสัมพันธ์กับหมายเลข IPv6 และกำหนดอุปกรณ์ที่ต้องดำเนินการ NAT ให้สัมพันธ์กับรูปแบบการจัดสรรหมายเลข IPv4 ภายในเครือข่าย ข้อสรุปในบทความดังกล่าวแนะนำว่า หากระบบที่ต้องการให้บริการมีหมายเลข IPv4 เพียงพอควรใช้วิธีการ public 4over6 และ 4rd ซึ่งดำเนินการ NAT โดยเกตเวย์ของผู้ใช้บริการ แต่หากมีหมายเลข IPv4 ไม่เพียงพอแนะนำให้ใช้ DS-lite ที่ดำเนินการ NAT ที่เครือข่ายของผู้ให้บริการ สำหรับเครือข่ายแกนหลัก IPv6 ของผู้ให้บริการไม่ขอกล่าวถึงรายละเอียดเนื่องจากอยู่นอกเหนือขอบเขตของวิทยานิพนธ์นี้ แม้ว่าการวิเคราะห์สามารถดำเนินการได้อย่างรวดเร็ว แต่การวิเคราะห์ยังคงมีข้อจำกัดอยู่ เนื่องจากไม่มีหลักฐานสนับสนุนที่สามารถแสดงให้เห็นรายละเอียดได้อย่างชัดเจนส่งผลให้การสรุปผลการวิเคราะห์อย่างละเอียดนั้นทำได้ค่อนข้างยาก

นอกจากการวิเคราะห์เพียงอย่างเดียวมีการนำเสนอการวิเคราะห์พร้อมผลการจำลองการใช้งาน สำหรับการจำลองการให้บริการด้วยกระบวนการเปลี่ยนถ่ายในเครือข่ายแกนหลักของผู้ให้บริการมีการนำกระบวนการเปลี่ยนถ่ายซึ่งประกอบด้วย 6rd, DS-lite และ IVI มาทำการจำลองด้วย OPNET modeler [26] ในการจำลองมีการวัดค่า Ethernet delay และ http delay เพื่อนำมาใช้ในการเปรียบเทียบประสิทธิภาพ กระบวนการเปลี่ยนถ่ายที่ถูกเลือกออกมาใช้เป็นตัวแทนของกระบวนการเปลี่ยนถ่ายที่มีหลักการทำงานแบบเดียวกันเพื่อใช้ในการเปรียบเทียบ ยกตัวอย่างเช่น 6rd จะใช้เป็นตัวแทนของกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv4 และ IVI จะใช้เป็นตัวแทนของกระบวนการเปลี่ยนถ่ายที่ใช้การแปลงหมายเลขโพรโตคอลอินเทอร์เน็ต เป็นต้น สำหรับผลการเปรียบเทียบประสิทธิภาพ การให้บริการด้วย Dual Stack มีค่า Ethernet delay และ http delay น้อยที่สุด ส่วนการให้บริการด้วย IVI จะมีค่า Ethernet delay มากที่สุด แต่มีค่า http delay น้อยกว่าการให้บริการ 6rd เนื่องจากมีการจัดสรรพอร์ตอย่างมีประสิทธิภาพ นอกจากนี้ยังมีการนำเสนอการเปรียบเทียบกระบวนการเปลี่ยนถ่ายแบบอุโมงค์สื่อสารและแบบแปลงหมายเลขโพรโตคอลอินเทอร์เน็ตซึ่งจำลองโดยใช้ OMNET++ [27] ในการ

จำลองมีการวัดค่า delay และ throughput เพื่อนำมาใช้ในการเปรียบเทียบประสิทธิภาพ สำหรับผลการเปรียบเทียบประสิทธิภาพการให้บริการแปลงหมายเลข IP มีค่า delay น้อยกว่าและมีค่า throughput สูงกว่าการให้บริการแบบอุโมงค์สื่อสาร แต่เมื่อเปรียบเทียบกับบทความซึ่งนำกระบวนการ IVI มาใช้เป็นตัวแทน [26] พบว่าค่า delay ที่วัดผลออกมาได้ไม่มีความสอดคล้องกัน ดังนั้น การเลือกกระบวนการเปลี่ยนถ่ายเพื่อนำมาใช้เป็นตัวแทนของกระบวนการเปลี่ยนถ่ายที่มีหลักการทำงานแบบเดียวกันนั้นเป็นวิธีการที่ไม่เหมาะสมเท่าที่ควร เนื่องจากกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการย่อมมีหลักการทำงานที่แตกต่างกันออกไปจึงไม่สามารถนำมาใช้ทดสอบทดแทนกันได้ในทุกกรณี สำหรับการจำลองการใช้งานด้วยกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารรูปแบบอื่นมีการจำลองกระบวนการเปลี่ยนถ่ายบน Multiprotocol Label Switch (MPLS) ด้วย OPNET modeler โดยการเปลี่ยนถ่ายที่นำมาเปรียบเทียบประกอบด้วย manual tunnel, GRE tunnel, 6to4 tunnel และ IPv4 compatible tunnel โดยมีการเปรียบเทียบการใช้งาน ใน 2 รูปแบบ คือ customer edge และ provider edge นอกจากนี้ยังมีการเปรียบระหว่างการให้บริการด้วย 6PE, Dual Stack และ Native IPv6 [28] ค่าที่นำมาเปรียบเทียบผลลัพธ์ได้แก่ delay, jitter และ throughput เมื่อได้ผลลัพธ์ของแบบจำลองทั้งหมดจึงนำมาวิเคราะห์ด้วยวิธีการทางสถิติ จากนั้นนำผลลัพธ์ทางสถิติมาจัดลำดับเพื่อสรุปกระบวนการให้บริการที่ดีที่สุด โดยวิธีการที่มีให้บริการที่ดีที่สุดก็คือ Dual Stack และรองลงมาก็คือ 6PE กระบวนการ 6PE มีประสิทธิภาพการให้บริการที่ค่อนข้างสูงกว่ากระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารอื่นๆ เนื่องจาก 6PE ไม่ดำเนินการห่อหุ้มแพ็กเก็ตแบบ IP-in-IP แต่ใช้การดำเนินการห่อหุ้มแบบ IP-in-MPLS ซึ่งเป็นการห่อหุ้มในระดับ link-layer ส่งผลให้ส่งแพ็กเก็ตเข้าสู่ภายในเครือข่ายแกนหลักที่ใช้ MPLS ได้โดยตรง แต่ข้อจำกัดของให้บริการโดยใช้กระบวนการ 6PE คือไม่สามารถให้บริการกับเครือข่ายไม่รองรับ MPLS ได้ แม้ว่าการนำแบบจำลองของระบบเครือข่ายเข้ามาช่วยในการวิเคราะห์ทำให้สามารถสรุปผลได้อย่างละเอียด แต่ก็จำเป็นที่จะต้องสร้างแบบจำลองอย่างรอบคอบเพื่อป้องกันไม่ให้ผลการทดลองที่ได้มีความคลาดเคลื่อน

นอกจากการจำลองการให้บริการด้วยกระบวนการเปลี่ยนถ่ายต่างๆ ยังมีการนำกระบวนการเปลี่ยนถ่ายมาทดสอบประสิทธิภาพการใช้งานบนเครือข่ายจริง เพื่อแสดงประสิทธิภาพของการใช้งานที่แท้จริง โดยกระบวนการเปลี่ยนถ่ายที่นำมาทดสอบได้แก่ manual tunnel, 6to4 tunnel และ tunnel broker [29] ในการทดสอบ มีค่าที่ใช้ในการวัดเพื่อทำการประเมินประสิทธิภาพได้แก่ delay, throughput, CPU utilization และ Loss Rate แม้ว่าการนำกระบวนการเปลี่ยนมาทดสอบการให้งานบนเครือข่ายจริงจะแสดงให้เห็นถึงประสิทธิภาพในการใช้งานจริง และสามารถวัดค่าที่ใช้เพื่อประเมินประสิทธิภาพได้อย่างครบถ้วน แต่จุดด้อยคือไม่สามารถดำเนินการได้อย่างรวดเร็ว เนื่องจากต้องพัฒนาบนอุปกรณ์จริง และต้องปรับปรุงเครือข่ายบางส่วนให้รองรับการดำเนินการของกระบวนการเปลี่ยนถ่ายที่นำมาทดสอบ

จากหัวข้อการทบทวนวรรณกรรมสามารถสรุปได้ว่า บทความซึ่งเกี่ยวข้องกับกระบวนการเปลี่ยนถ่ายส่วนใหญ่เป็นการเปรียบประสิทธิภาพในการให้บริการ กระบวนการเปลี่ยนถ่ายที่ถูกนำมาเปรียบเทียบประสิทธิภาพโดยใช้การวิเคราะห์จะมีความทันสมัยและหลากหลาย แต่ประสิทธิภาพจากการวิเคราะห์เพียงอย่างเดียวส่งผลให้ผลการวิเคราะห์ไม่มีความน่าเชื่อถือเท่าที่ควร

ด้วยเหตุนี้ การเปรียบเทียบประสิทธิภาพโดยใช้ผลลัพธ์จากแบบจำลองเครือข่ายจึงเป็นวิธีที่ได้รับความนิยมมากกว่าเนื่องจากมีหลักฐานในการสนับสนุนผลการวิเคราะห์ อีกทั้งกระบวนการเปลี่ยนถ่ายที่ถูกนำมาเปรียบเทียบประสิทธิภาพโดยใช้แบบจำลองก็ยังไม่ล้าสมัยจนเกินไป นอกจากนี้ แบบจำลองกระบวนการเปลี่ยนถ่ายส่วนใหญ่เป็นกระบวนการที่ใช้การสร้างการเชื่อมต่อ IPv6 ด้วยอุโมงค์สื่อสาร IPv4 ซึ่งอยู่ในช่วงระยะที่ 2 IPv4 ocean ของการเปลี่ยนแปลงการใช้งานจาก IPv4 ไปสู่ IPv6 ดังนั้นการเปรียบเทียบประสิทธิภาพโดยใช้ผลลัพธ์จากแบบจำลองเครือข่ายของกระบวนการเปลี่ยนถ่ายใช้การสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 ซึ่งอยู่ในช่วงระยะที่ 3 IPv6 ocean ของวิทยานิพนธ์ฉบับนี้จึงมีความน่าสนใจและเป็นการให้ข้อมูลใหม่ที่เป็นประโยชน์ในการตัดสินใจเลือกกระบวนการเหล่านี้ในการให้บริการ

# บทที่ 3
# การออกแบบกระบวนการเปลี่ยนถ่าย

ในบทที่ 2 กล่าวถึงหลักการทำงานของกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 แต่ละกระบวนการโดยละเอียด ข้อมูลดังกล่าวมีความสำคัญอย่างยิ่งในการนำมาวิเคราะห์ประสิทธิภาพของกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 ในบทนี้จะนำเสนอกระบวนการเปลี่ยนถ่ายซึ่งสามารถให้บริการการเชื่อมต่อ IPv4 ผ่านอุโมงค์สื่อสารเพื่อรองรับการใช้งาน IPv4 จนกระทั่งเลิกใช้งานในที่สุด หัวข้อหลักในบทนี้เริ่มต้นด้วยคุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมในการให้บริการ IPv4 และ IPv6 ควบคู่ไปด้วยกัน ซึ่งวิเคราะห์จากจุดเด่นของกระบวนการเปลี่ยนถ่าย จากนั้นนำเสนอแนวความคิดในการออกแบบกระบวนการเปลี่ยนถ่าย พร้อมทั้งอธิบายหลักการทำงานของ Enhancement of Lightweight 4over6 เมื่อทราบหลักการทำงานทั้งหมดแล้ว หัวข้อถัดไปเป็นการอธิบายโพรโตคอลและข้อมูลที่เกี่ยวข้องใน Enhancement of Lightweight 4over6 และสรุปภาพรวมขั้นตอนการดำเนินการของ Enhancement of Lightweight 4over6

## 3.1 คุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมในการให้บริการ IPv4 และ IPv6

ในหัวข้อนี้นำหลักการทำงานของกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 มาใช้ประกอบในการวิเคราะห์ประสิทธิภาพเพื่อนำไปสู่ข้อสรุปเกี่ยวกับคุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมกับการให้บริการ IPv4 และ IPv6 ควบคู่ไปด้วยกันจนกระทั่งมีการยกเลิกการใช้งาน IPv4 ในที่สุด โดยกระบวนการเปลี่ยนถ่ายที่ถูกนำมาเปรียบเทียบประกอบด้วย 4over6, DS-lite, lw4over6 และ 4rd ประเด็นที่ถูกนำมาเปรียบเทียบประกอบด้วย หลักการทำงาน, รูปแบบ NAT, การจัดสรรหมายเลข IPv4 ให้กับอุปกรณ์ฝั่งผู้ใช้งาน, การระบุอุโมงค์สื่อสารปลายทาง, การปรับปรุงข้อมูลอุโมงค์สื่อสารปลายทาง รูปแบบการเชื่อมต่อ, การแบ่งปันเซสชัน, ความสามารถในรองรับการขยายขนาด, ความซับซ้อน และอุปกรณ์ที่ต้องปรับปรุงในการให้บริการ ซึ่งผลการเปรียบเทียบได้แสดงรายละเอียดดังตารางที่ 3-1

ตารางที่ 3-1 เปรียบเทียบหลักการทำงานของกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6

| ประเด็น | กระบวนการเปลี่ยนถ่าย | | | |
|---|---|---|---|---|
| | 4over6 | DS-lite | lw4over6 | 4rd |
| หลักการทำงาน | ดำเนินการแบบ stateful | ดำเนินการแบบ stateful | ดำเนินการแบบ stateful | ดำเนินการแบบ stateless |
| รูปแบบ NAT | แบบกระจาย | แบบรวมศูนย์ | แบบกระจาย | แบบกระจาย |
| การเชื่อมต่อ | mesh | hub & spoke | hub & spoke | mesh |

ตารางที่ 3-1 (ต่อ) เปรียบเทียบหลักการทำงานของกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6

| ประเด็น | กระบวนการเปลี่ยนถ่าย | | | |
|---|---|---|---|---|
| | 4over6 | DS-lite | lw4over6 | 4rd |
| การจัดสรรหมายเลข IPv4 ให้กับอุปกรณ์ฝั่งผู้ใช้งาน | public IPv4 address | private IPv4 address | public IPv4 address และ port-set | public IPv4 address, public IPv4 address และ port-set |
| การระบุอุโมงค์สื่อสารปลายทาง | กำหนดโดยใช้ข้อมูลใน Encapsulation Table | กำหนดไปยังฝั่งผู้ให้บริการเสมอ | กำหนดไปยังฝั่งผู้ให้บริการเสมอ | กำหนดตาม 4rd rule |
| การปรับปรุงข้อมูลอุโมงค์สื่อสารปลายทาง | อุปกรณ์ฝั่งผู้ใช้งานและฝั่งผู้ให้บริการปรับปรุงข้อมูลด้วยโพรโตคอลกำหนดเส้นทาง | อุปกรณ์ฝั่งผู้ให้บริการปรับปรุงข้อมูลด้วย DHCPv6 | อุปกรณ์ฝั่งผู้ให้บริการปรับปรุงข้อมูลด้วย DHCPv6 | เนื่องจากยึดตาม 4rd rule จึงไม่มีการปรับปรุงข้อมูล |
| การแบ่งปันเซสชัน | แต่ละเครือข่ายมีจำนวนเซสชันตามจำนวนของ public IPv4 address ที่ครอบครองอยู่ | แต่ละเครือข่ายผู้ใช้งานได้รับจำนวนเซสชันตามการดำเนินการ NAT ของอุปกรณ์ฝั่งผู้ให้บริการ | แต่ละเครือข่ายผู้ใช้งานได้รับจำนวนเซสชันอย่างเท่าเทียม แต่ก็สามารถปรับจำนวนเซสชันได้ตามความเหมาะสม | แต่ละเครือข่ายผู้ใช้งานได้รับจำนวนเซสชันอย่างเท่าเทียม |
| ความสามารถในรองรับการขยายขนาด | สามารถเพิ่มจำนวนอุปกรณ์ฝั่งผู้ให้บริการ | - | สามารถเพิ่มจำนวนอุปกรณ์ฝั่งผู้ให้บริการ แต่ต้องแลกเปลี่ยนข้อมูลเพิ่มเติม | สามารถเพิ่มจำนวนอุปกรณ์ฝั่งผู้ให้บริการ |
| ความซับซ้อน | ความซับซ้อนสูง เนื่องจากนำโพรโตคอลกำหนดเส้นทางมาใช้ในการแลกเปลี่ยนข้อมูล | ความซับซ้อนสูง เนื่องจากผนวกรวมการดำเนินการ NAT และการระบุอุโมงค์สื่อสารเข้าด้วยกัน | ความซับซ้อนค่อนข้างสูง เนื่องจากแบ่งการดำเนินการ NAT และการระบุอุโมงค์สื่อสารออกจากกัน | ความซับซ้อนปานกลาง เนื่องจากดำเนินการตาม 4rd rule ที่กำหนดล่วงหน้าเท่านั้น |
| อุปกรณ์ที่ต้องปรับปรุงในการให้บริการ | ต้องปรับปรุงอุปกรณ์ฝั่งผู้ให้บริการและฝั่งผู้ใช้ งานให้รองรับทั้งกระบวนการเปลี่ยนถ่าย และโพรโตคอลกำหนดเส้นทาง | ต้องปรับปรุงอุปกรณ์ฝั่งผู้ให้บริการให้รองรับกระบวนการเปลี่ยนถ่ายและอุปกรณ์ฝั่งผู้ใช้งานให้รองรับอุโมงค์สื่อสาร | ต้องปรับปรุงอุปกรณ์ฝั่งผู้ให้บริการและฝั่งผู้ใช้งานให้รองรับทั้งกระบวนการเปลี่ยนถ่าย และ DHCPv6 | ต้องปรับปรุงอุปกรณ์ฝั่งผู้ให้บริการและฝั่งผู้ใช้งานให้รองรับทั้งกระบวนการเปลี่ยนถ่าย และ DHCPv6 |

ผลลัพธ์จากตารางที่ 3-1 กระบวนการเปลี่ยนถ่ายที่นำมาเปรียบเทียบทุกกระบวนการต่างทำงานบนเครือข่ายแกนหลัก IPv6 ดังนั้นทุกกระบวนการเปลี่ยนถ่ายจึงสามารถให้บริการการเชื่อมต่อ IPv6 ได้โดยตรง ส่วนการให้บริการการเชื่อมต่อ IPv4 นั้นดำเนินการผ่านอุโมงค์สื่อสาร กระบวนการเปลี่ยนถ่ายที่ทำงานบนเครือข่ายแกนหลัก IPv6 เป็นคุณสมบัติที่ดีอย่างหนึ่ง เพราะเครือข่ายสามารถรองรับ IPv6 ได้อย่างสมบูรณ์ อีกทั้งยังสามารถให้บริการ IPv4 ควบคู่ไปด้วยกัน ซึ่งแม้ IPv4 ถูกยกเลิกการใช้งานในอนาคตก็ไม่ส่งผลกระทบต่อการให้บริการแต่อย่างใด ส่วนคุณสมบัติอื่นๆ ที่มีความโดดเด่นล้วนมีความแตกต่างกันไป สำหรับคุณสมบัติเด่นของ 4over6 คือสามารถจัดสรรหมายเลข IPv4 และ IPv6 ได้อย่างอิสระ และรองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง สำหรับคุณสมบัติเด่นของ DS-lite คือสามารถจัดสรรหมายเลข IPv4 และ IPv6 ได้อย่างอิสระเพียงอย่างเดียวเท่านั้น สำหรับคุณสมบัติเด่นของ lw4over6 ซึ่งเป็นกระบวนการที่เพิ่มเติมจาก DS-lite ก็ได้รับการสืบทอดคุณสมบัติจัดสรรหมายเลข IPv4 และ IPv6 ได้อย่างอิสระเช่นเดียวกัน นอกจากนี้ lw4over6 ยังสามารถแบ่งปันจัดสรรหมายเลข IPv4 ในระดับพอร์ตแบบพลวัตได้อีกด้วยซึ่งสามารถช่วยลดปัญหาความขาดแคลนหมายเลข IPv4 ลงได้ สำหรับกระบวนการสุดท้าย คุณสมบัติเด่นของ 4rd คือสามารถแบ่งปันจัดสรรหมายเลข IPv4 ในระดับพอร์ต และรองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง แต่ไม่สามารถจัดสรรหมายเลข IPv4 และ IPv6 ได้อย่างอิสระ

เมื่อนำคุณสมบัติเด่นของแต่ละกระบวนการมาผนวกรวมเข้าด้วยกันสามารถสรุปเป็นคุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมในการให้บริการมีทั้งหมด 4 คุณสมบัติด้วยกัน ซึ่งประกอบด้วย

1) ส่งเสริมให้เครือข่ายของผู้ให้บริการรองรับการให้บริการด้วย IPv6 โดยตรง เพื่อเตรียมพร้อมสำหรับความต้องการใช้งานในอนาคต
2) สามารถจัดสรร IPv4 และ IPv6 ได้อย่างอิสระ เพื่อให้การจัดสรรหมายเลข IPv4 และ IPv6 มีความยืดหยุ่นและสอดคล้องกับความต้องการใช้งานอย่างแท้จริง
3) สามารถแบ่งจัดสรรหมายเลข IPv4 ให้กับผู้ใช้งานได้ในระดับพอร์ตแบบพลวัตเพื่อลดปัญหาความขาดแคลนหมายเลข IPv4
4) รองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง เพื่อให้บริการการเชื่อมต่อ IPv4 ได้อย่างมีประสิทธิภาพสูงสุด

### 3.2 แนวความคิดในการออกแบบ

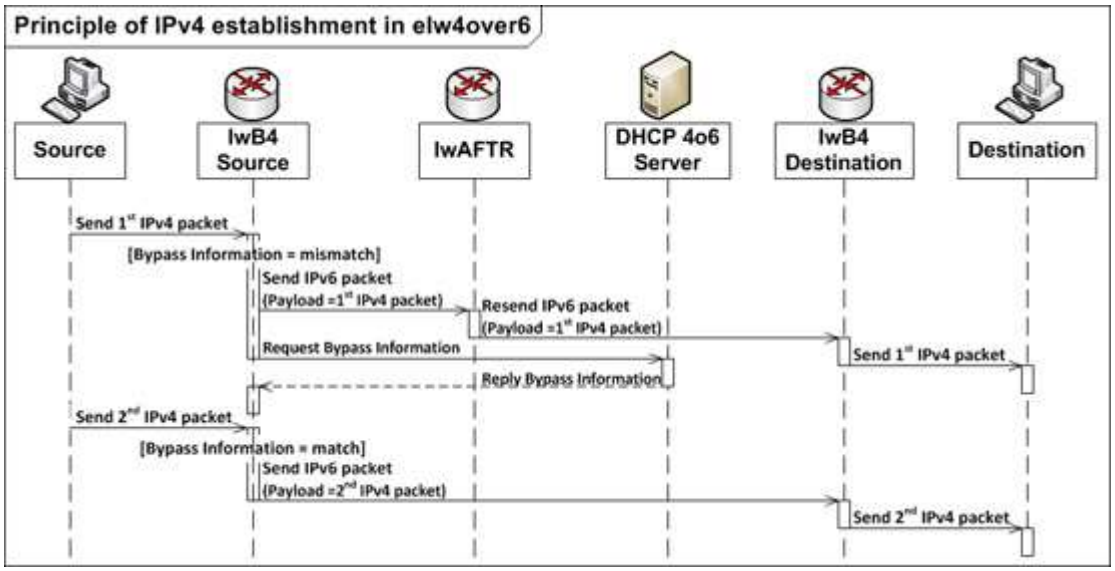
เมื่อพิจารณาตามคุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมในการให้บริการทั้ง 4 ข้อเทียบกับกระบวนการเปลี่ยนถ่ายที่ได้กล่าวมาข้างต้นพบว่า lw4over6 มีคุณสมบัติใกล้เคียงมากที่สุด เพราะ lw4over6 ถูกออกแบบให้มีการดำเนินการ NAT แบบกระจาย เพื่อลดภาระการประมวลผลที่ศูนย์กลางลง โดย lw4over6 กระจายหมายเลข IPv4 และ port-set จากส่วนกลางไปยังเครือข่ายของผู้ใช้งาน วัตถุประสงค์หลักของการดำเนินการดังกล่าวคือให้เครือข่ายของผู้ใช้งานสามารถดำเนินการ NAT ก่อนที่จะห่อหุ้มแพ็กเก็ตเพื่อส่งต่อมายังส่วนกลาง ดังนั้น lw4over6 มีคุณสมบัติ 3 ใน 4 ข้อของกระบวนการเปลี่ยนถ่ายที่เหมาะสมสำหรับให้บริการ IPv4 และ IPv6 ควบคู่ไปด้วยกัน ซึ่งประกอบด้วย 1) ส่งเสริมให้เครือข่ายของผู้ให้บริการรองรับการให้บริการด้วย IPv6 โดยตรง 2) สามารถจัดสรร IPv4 และ IPv6 ได้อย่างอิสระ 3) สามารถแบ่งจัดสรรหมายเลข IPv4 ให้กับผู้ใช้งานได้ในระดับพอร์ตแบบพลวัต อย่างไรก็ตาม lw4over6 ยังขาดคุณสมบัติอีกข้อหนึ่ง นั่นคือการรองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง เนื่องจาก lwB4 มีเพียงอุโมงค์สื่อสารเชื่อมต่อไปยัง lwAFTR เท่านั้น lwAFTR จึงต้องทำหน้าที่ส่งต่อแพ็กเก็ตให้กับ lwB4 เสมอ เพื่อแก้ไขข้อจำกัดดังกล่าว ในบทนี้ขอนำเสนอกระบวนการที่ถูกพัฒนาเพิ่มเติมจาก lw4over6 ซึ่งสามารถให้บริการได้เช่นเดียวกับ lw4over6 ทุกประการ ยิ่งกว่านั้นกระบวนการที่นำเสนอสามารถสร้างการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง โดยอาศัยการร้องข้อข้อมูลอุโมงค์สื่อสารปลายทางจาก DHCP 4o6 server เพื่อให้ lwB4 สามารถเรียนรู้ข้อมูลอุโมงค์สื่อสารโดยตรงที่เชื่อมต่อกับเครือข่ายของเครื่องปลายทาง กระบวนการที่ถูกนำเสนอนี้เรียกว่า “Enhancement Lightweight 4over6” (elw4over6)

การปรับปรุงให้ lw4over6 รองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง ประกอบด้วยประเด็นสำคัญที่ต้องพิจารณา 2 ประเด็น ได้แก่ 1) หมายเลข IPv4 ที่ใช้ในการเชื่อมต่อ และ 2) การระบุอุโมงค์สื่อสารของเครือข่ายปลายทาง สำหรับประเด็นแรก lw4over6 ออกแบบให้รองรับการดำเนินการ NAT แบบกระจาย lwB4 ได้รับ public IPv4 address และ port-set สำหรับใช้ในการเชื่อมต่อ ในประเด็นนี้จึงไม่จำเป็นต้องปรับปรุงเพิ่มเติม สำหรับประเด็นที่สอง lwB4 มีเพียงอุโมงค์สื่อสารเชื่อมต่อไปยัง lwAFTR เท่านั้น lwB4 ไม่มีการบำรุงรักษาข้อมูลของอุโมงค์สื่อสารปลายทางเช่นเดียวกับ lwAFTR ส่งผลให้ lwB4 ต้องพึ่งพา lwAFTR ในการส่งต่อแพ็กเก็ตไปยังเครือข่ายปลายทาง หากปรับปรุงให้ lwB4 สามารถบำรุงรักษาข้อมูลของอุโมงค์สื่อสารปลายทางได้เช่นเดียวกับ lwAFTR lwB4 จะสามารถสร้างอุโมงค์สื่อสารไปยังเครือข่ายปลายทางได้โดยตรง

### 3.3 หลักการทำงานของ Enhancement of Lightweight 4over6

Enhancement of Lightweight 4over6 (elw4over6) ถูกปรับปรุงให้สามารถส่งต่อแพ็กเก็ต IPv4 ไปยังเครื่องปลายทางที่อยู่ภายในขอบเขตการให้บริการด้วย elw4over6 ได้โดยตรง โดยอาศัยข้อมูล Bypass Scope และ Bypass Binding Table เป็นหลัก ข้อมูล Bypass Scope ถูกใช้เพื่อคัดกรองระหว่างแพ็กเก็ต IPv4 ภายในขอบเขตการให้บริการ ส่วนข้อมูล Bypass

Binding Table ซึ่งมีลักษณะคล้าย Binding Table บน lwAFTR ทำหน้าที่บันทึกข้อมูลของ lwB4 ปลายทางที่เคยติดต่อสื่อสาร เพื่อให้การสื่อสารครั้งต่อไปสามารถส่งข้อมูลไปยังเครือข่ายปลายทางได้โดยตรง ในการดำเนินการดังกล่าว elw4over6 ประยุกต์ DHCPv4 Leasequery และ DHCPv4 over DHCPv6 เพื่อแลกเปลี่ยนข้อมูล Bypass Binding Table แม้การดำเนินการเช่นนี้จะเป็นการเพิ่มภาระให้กับอุปกรณ์ฝั่งผู้ใช้งาน แต่ก็สามารถลดปัญหาคอขวดซึ่งเกิดขึ้นบน lwAFTR อีกทั้งยังสามารถใช้เส้นทางที่เหมาะสมที่สุดในการส่งข้อมูลไปยังเครื่องปลายทางภายในขอบเขตการให้บริการของ elw4over6 โดยที่เครื่องต้นทางไม่จำเป็นต้องส่งแพ็กเก็ต IPv4 โดยอาศัย lwAFTR สำหรับหลักการสร้างการเชื่อมต่อ IPv4 ของ elw4over6 เบื้องต้นแสดงรายละเอียดดังจากรูปที่ 3-1




รูปที่ 3-1 หลักการสร้างการเชื่อมต่อ IPv4 ของ elw4over6 เบื้องต้น

อุปกรณ์ในการให้บริการใน elw4over6 มี 3 ชนิดด้วยกันได้แก่ lwAFTR, lwB4 และ DHCP 4o6 server สำหรับ lwAFTR ใน elw4over6 มีหลักการทำงานเหมือนกับ lwAFTR ใน lw4over6 ทุกประการ ส่วน DHCP 4o6 Server ใน elw4over6 มีหลักการทำงานไม่แตกต่างกับ DHCP 4o6 Server ใน lw4over6 โดย DHCP 4o6 Server ใน elw4over6 ปรับปรุงให้รองรับ option ใหม่เพิ่มเติมเพียงเล็กน้อย แต่สำหรับ lwB4 ใน elw4over6 มีการปรับปรุงเพิ่มเติมอย่างมาก เนื่องจาก lwB4 ต้องรองรับการกำหนดอุโมงค์สื่อสารไปยังเครือข่ายปลายทางโดยตรง หลักการทำงานของอุปกรณ์แต่ละชนิดใน elw4over6 มีรายละเอียดดังต่อไปนี้

### 3.3.1 หลักการทำงานของ lwAFTR ใน elw4over6

lwAFTR ใน elw4over6 ทำหน้าที่ปรับปรุงข้อมูลของอุโมงค์สื่อสารปลายทาง และส่งต่อแพ็กเก็ตไปยังเครือข่ายปลายทางโดยอาศัยข้อมูลที่ได้จากการปรับปรุง นอกจากนั้น lwAFTR ต้องทำหน้าที่เป็น DHCPv6 Relay ระหว่าง lwB4 และ DHCP 4o6 Server เพื่อบันทึกข้อมูลการจัดสรร Public IPv4 address และพอร์ตของ lwB4 ทั้งหมดภายในขอบเขตการให้บริการ (lwAFTR อาจไม่จำเป็นต้องทำหน้าที่เป็น DHCPv6 Relay ก็ได้ หากกำหนดให้ DHCP 4o6 Server ทำหน้าที่

ส่งข้อมูลการจัดสรร Public IPv4 address และพอร์ตให้กับ lwAFTR ทันทีที่มีการจัดสรรข้อมูลใหม่) โดย lwAFTR ใน elw4over6 ยังคงมีหลักการทำงานเหมือนกับ lwAFTR ใน lw4over6 ทุกประการ ซึ่งมีรายละเอียดดังต่อไปนี้

**3.3.1.1 lwAFTR Control Plan Behavior**

lwAFTR ปรับปรุงข้อมูลของ lwB4 ภายใน Binding Table ซึ่งประกอบด้วย หมายเลข IPv6, IPv4 และ port-set หน้าที่หลัก 2 ประการของ lwAFTR ได้แก่ การห่อหุ้มแพ็กเก็ต IPv4 ที่ถูกส่งมาจากอินเทอร์เน็ตด้วยแพ็กเก็ต IPv6 แล้วส่งต่อไปยัง lwB4 และการตรวจสอบความถูกต้องของแพ็กเก็ต IPv6 ที่ห่อหุ้มแพ็กเก็ต IPv4 ซึ่งถูกส่งมาจาก lwB4 เพื่อส่งต่อไปยังอินเทอร์เน็ตต่อไป

lwAFTR ต้องประสานข้อมูลของ lwB4 กับ DHCP 4O6 Server เพื่อปรับปรุงข้อมูลอย่างสม่ำเสมอ ยิ่งกว่านั้นข้อมูลของ lwB4 ที่บันทึกต้องมีการระบุ lease time เพื่อให้ lwAFTR สามารถยกเลิกการใช้งานได้เมื่อ lwB4 สิ้นสุดการใช้งาน การประสานข้อมูลของ lwB4 สามารถดำเนินการได้ 2 วิธีการด้วยกัน วิธีการแรกคือกำหนดให้ lwAFTR เป็น DHCPv6 Relay เพื่อให้ lwAFTR สามารถติดตามข้อมูลของ lwB4 ในขณะที่ lwB4 กำลังแลกเปลี่ยนข้อมูลกับ DHCP 4O6 Server และวิธีการที่สองคือกำหนดให้ DHCP 4O6 Server แจ้งปรับปรุงข้อมูลของ lwB4 ไปยัง lwAFTR เมื่อข้อมูลมีการเปลี่ยนแปลง

**3.3.1.2 lwAFTR Data Plan Behavior**

การนำส่งข้อมูลของ lwAFTR แบ่งออกเป็น 3 เหตุการณ์หลัก ได้แก่ การรับแพ็กเก็ต IPv6 ซึ่งบรรจุแพ็กเก็ต IPv4, การรับแพ็กเก็ต IPv4 และ Hairpinning

กรณีที่ lwAFTR ได้รับแพ็กเก็ต IPv6 ซึ่งบรรจุแพ็กเก็ต IPv4 อยู่ภายใน lwAFTR ต้องนำแพ็กเก็ต IPv4 ดั้งเดิมออกมา แล้วนำหมายเลข IPv6, IPv4 และพอร์ตของผู้ส่งมาตรวจสอบกับ Binding Table หากข้อมูลถูกต้อง lwAFTR สามารถส่งต่อแพ็กเก็ต IPv4 ไปยังเครื่องปลายทางได้ แต่หากข้อมูลไม่ถูกต้อง lwAFTR ต้องละทิ้งแพ็กเก็ตดังกล่าว และส่ง ICMPv6 type 1, code 5 (source address failed ingress/egress policy) กลับไปยัง lwB4 ต้นทางต่อไป

กรณีที่ lwAFTR ได้รับแพ็กเก็ต IPv4 หมายเลข IPv4 ปลายทางและพอร์ตจะถูกนำมาตรวจสอบกับ Binding Table หากข้อมูลตรงกับ lwB4 ใด แพ็กเก็ต IPv4 จะถูกห่อหุ้มด้วยแพ็กเก็ต IPv6 แล้วส่งต่อไปยัง lwB4 ดังกล่าว โดยที่หมายเลข IPv6 ต้นทางถูกกำหนดเป็นหมายเลข IPv6 ของ lwAFTR แต่หากข้อมูลไม่ตรงกับ lwB4 ใดเลย แพ็กเก็ต IPv4 จะถูกละทิ้งพร้อมกับส่ง ICMPv4 type 3, code 1 (Destination unreachable) กลับไปยังเครื่องต้นทาง

Hairpinning เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ lwB4 ของเครือข่ายผู้ใช้งาน 2 เครือข่ายภายในขอบเขตการให้บริการของ elw4over6 ต้องการส่งข้อมูลระหว่างกัน lwAFTR ยังคงต้องรองรับการส่งต่อแพ็กเก็ตจาก lwB4 ต้นทางไปยัง lwB4 ปลายทางเช่นเดียวกับ lw4over6 โดยแพ็กเก็ต IPv4 ดั้งเดิมจะนำออกมาจากแพ็กเก็ต IPv6 ที่ส่งมาจาก lwB4 ต้นทาง และถูกห่อหุ้มลงในแพ็กเก็ต IPv6 ที่กำหนดปลายทางเป็น lwB4 ปลายทางอีกครั้ง

### 3.3.2 หลักการทำงานของ lwB4 ใน elw4over6

หลักการทำงานพื้นฐานของ lwB4 คือการดำเนินการ NAT และการสร้างอุโมงค์สื่อสาร อย่างไรก็ตาม lwB4 ใน elw4over6 มีหลักการทำงานเพิ่มขึ้นจาก lwB4 ใน lw4over6 เนื่องจากต้องปรับข้อมูลของอุโมงค์สื่อสารปลายทางให้เป็นปัจจุบันเพื่อใช้สำหรับสร้างการเชื่อมต่อไปยังเครือข่ายของเครื่องปลายทางโดยตรง ข้อมูลของอุโมงค์สื่อสารปลายทางถูกปรับปรุงให้สอดคล้องกับข้อมูลที่บันทึกโดย DHCP 4o6 Server หลักการทำงานของ lwB4 ใน elw4over6 มีรายละเอียดดังต่อไปนี้

#### 3.3.2.1 การจัดเตรียมข้อมูลที่จำเป็นของ lwB4 ใน elw4over6

ข้อมูลที่จำเป็นสำหรับ lwB4 ไม่ได้มีเพียงแค่หมายเลข IPv6 ของ lwAFTR เท่านั้น แต่ยังรวมถึงหมายเลข IPv4 และ port-set ที่ใช้สำหรับดำเนินการ NAT อีกด้วย ยิ่งไปกว่านั้นการที่เครือข่ายแกนหลักของผู้ให้บริการไม่สามารถใช้งาน IPv4 โดยตรงได้ ส่งผลให้การจัดหาข้อมูลสำหรับ lwB4 ทั้งหมดจึงต้องดำเนินการผ่าน DHCPv4 over DHCPv6 เท่านั้น

การแลกเปลี่ยนข้อมูลที่จำเป็นสำหรับ lwB4 สามารถทำได้หลายวิธีการด้วยกัน แต่ elw4over6 ให้ความสำคัญสำหรับประเด็นเกี่ยวกับ lease time เป็นหลัก เนื่องจาก lease time ของหมายเลข IPv6 และ IPv4 ควรเป็นอิสระต่อกันและ lease time ของหมายเลข IPv4 ต้องมีค่าสูงสุดไม่เกินค่า lease time ของหมายเลข IPv6 ดังนั้น elw4over6 จึงสนับสนุนแนวคิดการจัดสรรข้อมูลสำหรับ lwB4 ด้วย DHCPv4 over DHCPv6 เนื่องจากสามารถแยกจัดสรรหมายเลข IPv4 ได้อย่างอิสระ ซึ่งสามารถยกเลิกการใช้งานหมายเลข IPv4 หรือต่ออายุการใช้งานหมายเลข IPv4 โดยไม่ส่งผลกระทบกับการใช้งาน IPv6

สำหรับการจัดสรรข้อมูลสำหรับ lwB4 ด้วย DHCPv4 over DHCPv6 นั้น การแลกเปลี่ยนข้อมูลหมายเลข IPv6 ของ lwAFTR และหมายเลข IPv6 ของ DHCP 4o6 Server ดำเนินการโดยใช้ DHCPv6 ส่วนข้อมูลซึ่งเกี่ยวข้องกับ IPv4 ยกตัวอย่างเช่น หมายเลข IPv4, port-set, Bypass Scope และ lease time ของ IPv4 ดำเนินการโดยใช้ DHCPv4 over DHCPv6 ทั้งหมด

การจัดหาข้อมูล IPv4 ที่เกี่ยวข้องสำหรับ lwB4 ของ elw4over6 มีขั้นตอนหลักเหมือนกับการจัดหาข้อมูลโดยใช้ DHCPv4 over DHCPv6 ของ lw4over6 ทุกประการ ยกเว้นเพียงการร้องขอ Bypass Scope เพิ่มเติมใน DHCPv4 option เท่านั้น lwB4 เริ่มต้นการจัดหาข้อมูลสำหรับ elw4over6 โดยดำเนินการร้องขอข้อมูลของ DHCPv6 OPTION_DHCPv4_O_DHCPv6_SERVER ไปยัง DHCPv6 server เมื่อ lwB4 ได้รับข้อมูลของ DHCP 4o6 server lwB4 ดำเนินการร้องขอข้อมูลของ DHCPv6 OPTION_S46_CONT_DHCP4O6 ไปยัง DHCP 4o6 server โดยระบุ OPTION_CLIENTID และ OPTION_IA_NA เพื่อลงทะเบียนสำหรับใช้ในการร้องขอข้อมูล IPv4 จากนั้น DHCP 4o6 server ตอบกลับข้อมูล DHCPv6 OPTION_S46_CONT_DHCP4O6 ซึ่งภายในบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR_HINT จากนั้น lwB4 จึงส่ง DHCPv6 OPTION_CLIENTID, OPTION_IA_NA และ OPTION_S46_CONT_DHCP4O6 ซึ่งภายในบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR เพื่อลงทะเบียน DUID และ IAID คู่กับหมายเลข IPv6 ของ lwB4 ท้ายที่สุด DHCP 4O6 Server ตอบ

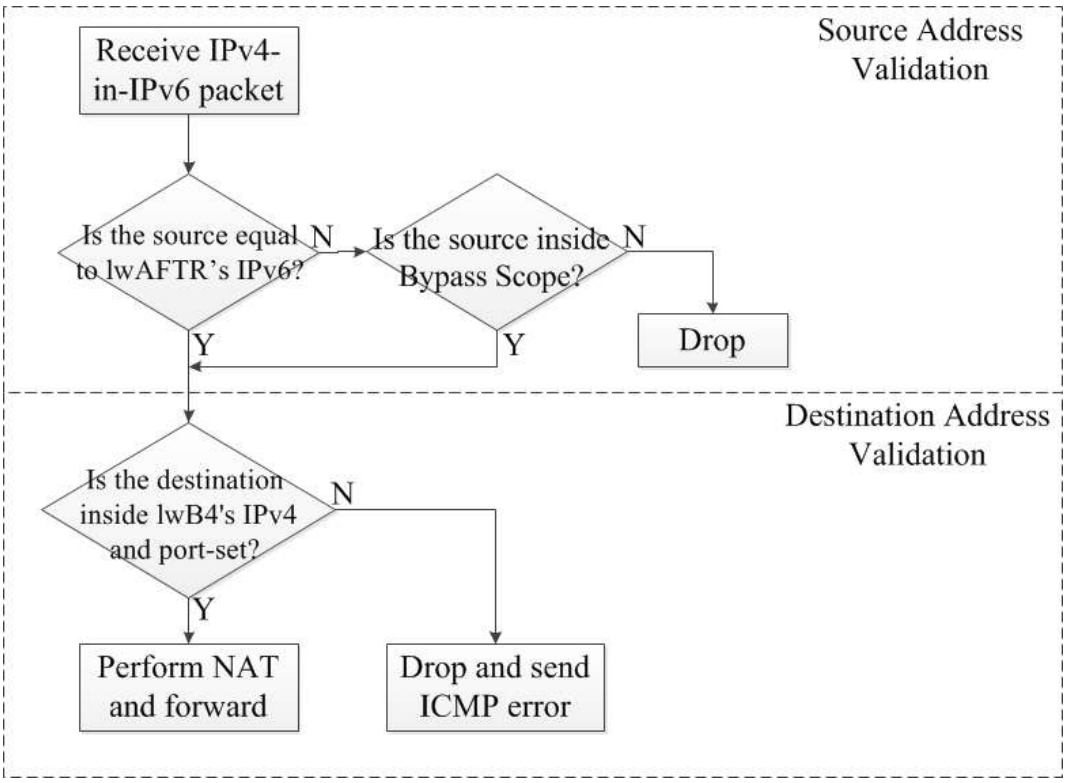
กลับข้อมูล DHCPv6 OPTION_S46_CONT_DHCP4O6 ซึ่งภายในบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR กลับมาจึงเสร็จสิ้นการลงทะเบียนของ lwB4

เพื่อดำเนินการร้องขอข้อมูล IPv4 ผ่านกระบวนการ DHCPv4 over DHCPv6 lwB4 ร้องขอ DHCPv4 OPTON_v4_PORTPARAMS ไปยัง DHCP 4o6 server อีกครั้งโดยระบุ Client Identifier (Type:255 IAID ,DUID) และ Subnet Allocation Option (Bypass Scope) จากนั้น DHCP 4o6 server นำ IAID และ DUID มาตรวจสอบกับหมายเลข IPv6 ของ lwB4 ที่ได้ทำการลงทะเบียนไว้ หาก IAID และ DUID ถูกต้อง DHCP 4O6 Server จะตอบกลับข้อมูลหมายเลข IPv4, port-set, lease time และ Bypass Scope ซึ่งได้รับจัดสรรให้กับ lwB4

**3.3.2.2 lwB4 Data Plan Behavior**

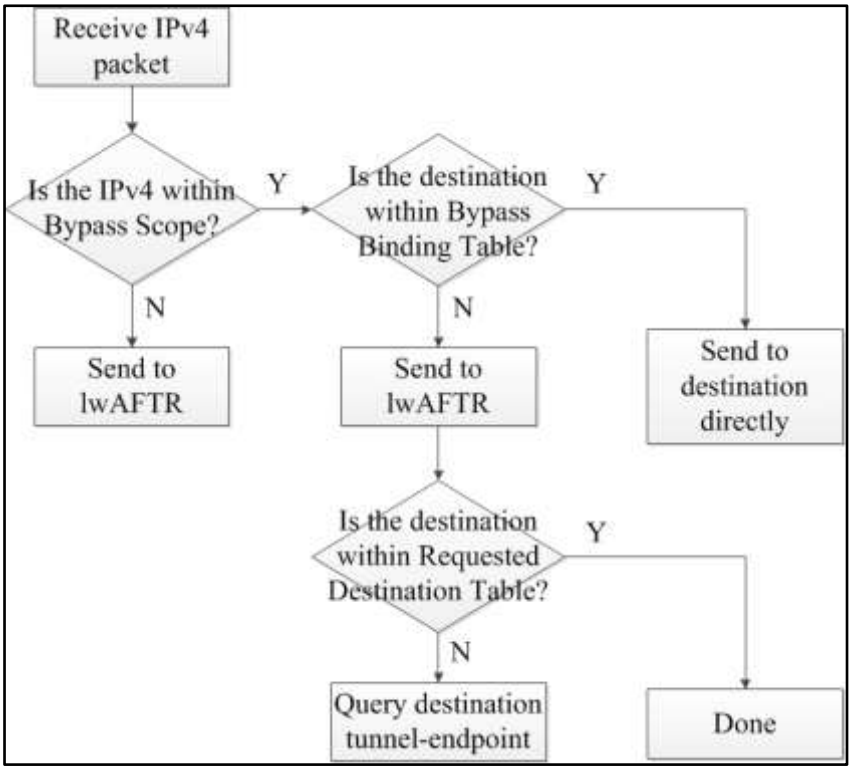
หลักการในการนำส่งข้อมูลของ lwB4 ใน elw4over6 มีการปรับปรุงเพิ่มเติมเพื่อให้รองรับการปรับปรุงข้อมูลของอุโมงค์สื่อสารปลายทางอื่นนอกเหนือจากอุโมงค์สื่อสารที่เชื่อมต่อไปยัง lwAFTR อย่างไรก็ตามการนำส่งข้อมูลของ lwB4 ก็ยังคงแบ่งออกเป็น 2 เหตุการณ์หลัก ได้แก่ การรับแพ็กเก็ต IPv6 ซึ่งบรรจุแพ็กเก็ต IPv4 และการรับแพ็กเก็ต IPv4

เมื่อ lwB4 ได้รับแพ็กเก็ต IPv6 ซึ่งบรรจุแพ็กเก็ต IPv4 การตรวจสอบความถูกต้องของแพ็กเก็ต IPv6 ถูกแบ่งออกเป็น 2 ขั้นตอนด้วยกัน ได้แก่ การตรวจสอบหมายเลขต้นทาง และการตรวจสอบหมายเลขปลายทาง ขั้นตอนแรกคือการตรวจสอบหมายเลขต้นทาง หากหมายเลข IPv6 ต้นทางตรงกับหมายเลข IPv6 ของ lwAFTR แพ็กเก็ตจะผ่านเข้าสู่ขั้นตอนการตรวจสอบหมายเลขปลายทางทันที แต่หากหมายเลข IPv6 ต้นทางไม่ตรงกับหมายเลข IPv6 ของ lwAFTR lwB4 ต้องตรวจสอบหมายเลข IPv4 ต้นทางว่าอยู่ภายในขอบเขตของ Bypass Scope หรือไม่ หากหมายเลข IPv4 ต้นทางอยู่ภายในขอบเขตของ Bypass Scope แพ็กเก็ตจะผ่านเข้าสู่ขั้นตอนการตรวจสอบหมายเลขปลายทางต่อไป แต่หากหมายเลข IPv4 ต้นทางไม่อยู่ภายในขอบเขตของ Bypass Scope แพ็กเก็ตดังกล่าวก็จะถูกละทิ้งโดยไม่มีการส่งการแจ้งเตือนใดๆ กลับไป เมื่อแพ็กเก็ตเข้าสู่ขั้นตอนการตรวจสอบหมายเลขปลายทาง หากหมายเลข IPv4 และพอร์ตของเครื่องปลายทางตรงกับหมายเลข IPv4 และ port-set ซึ่ง lwB4 ได้รับการจัดสรร แพ็กเก็ต IPv4 จะถูกส่งต่อไปยังเครื่องปลายทางต่อไป แต่ถ้าหากหมายเลข IPv4 และ port-set ไม่ตรงกับที่ lwB4 ได้รับการจัดสรร แพ็กเก็ตดังกล่าวจะถูกละทิ้งและส่ง ICMPv4 type 3 code 13 (Communication Administratively Prohibited) กลับไปยังเครื่องต้นทาง โดย ICMPv4 ดังกล่าวสามารถส่งผ่านเครือข่าย IPv6 ตามข้อกำหนดของการสร้างการเชื่อมต่อ IPv4 ด้วยอุโมงค์สื่อสาร IPv6 [8] ขั้นตอนของการรับแพ็กเก็ต IPv6 ซึ่งบรรจุแพ็กเก็ต IPv4 ของ lwB4 มีรายละเอียดดังแสดงในรูปที่ 3-2




รูปที่ 3-2 ขั้นตอนของการรับแพ็กเก็ต IPv6 ซึ่งบรรจุแพ็กเก็ต IPv4 ของ lwB4

เมื่อ lwB4 ได้รับแพ็กเก็ต IPv4 แพ็กเก็ต IPv4 จะถูกส่งต่อไปยังเครื่องปลายทางผ่านอุโมงค์สื่อสาร เพื่อให้สามารถกำหนดอุโมงค์สื่อสารปลายทางได้อย่างถูกต้อง lwB4 ต้องนำหมายเลข IPv4 และพอร์ตของเครื่องปลายทางมาตรวจสอบ หากหมายเลขของ IPv4 ปลายทางไม่อยู่ในภาย Bypass Scope แพ็กเก็ต IPv4 จะถูกห่อหุ้มแล้วส่งต่อไปยัง lwAFTR แต่ถ้าหากหมายเลข IPv4 ปลายทางอยู่ภายใน Bypass Scope หมายเลข IPv4 และพอร์ตดังกล่าวจะถูกนำไปค้นหาใน Bypass Binding Table ในกรณีที่หมายเลข IPv4 และพอร์ตตรงกับข้อมูลภายใน Bypass Binding Table แพ็กเก็ต IPv4 จะถูกห่อหุ้มและส่งต่อไปยังปลายทางโดยตรงตามข้อมูลที่พบใน Bypass Binding Table แต่หากข้อมูลหมายเลข IPv4 และพอร์ตไม่ตรงกับข้อมูลใดเลย แพ็กเก็ต IPv4 จะถูกห่อหุ้มและส่งต่อไปยัง lwAFTR พร้อมกับเริ่มกระบวนการร้องขอข้อมูลของ lwB4 ปลายทางซึ่งเป็นเจ้าของหมายเลข IPv4 และพอร์ตดังกล่าว ในการร้องขอข้อมูลของ lwB4 ปลายทาง lwB4 ต้องตรวจสอบว่าหมายเลข IPv4 และพอร์ตของเครื่องปลายทางดังกล่าวถูกร้องขอไปแล้วหรือไม่ โดยตรวจสอบข้อมูลภายใน Requested Destination Table หากพบข้อมูล IPv4 และพอร์ตภายใน Requested Destination Table แสดงว่าข้อมูลของ lwB4 ปลายทางถูกดำเนินการร้องขอไปแล้ว ไม่จำเป็นต้องดำเนินการร้องขอซ้ำ แต่หากไม่พบข้อมูลของ IPv4 และพอร์ตภายใน Requested Destination Table lwB4 ต้องร้องข้อข้อมูลของ lwB4 ปลายทางไปยัง DHCP 4o6 server โดยใช้ DHCPv4 Leasequery over DHCPv6 และเพิ่มหมายเลข IPv4 และพอร์ตของเครื่องปลายทางลงไปใน Requested Destination Table ขั้นตอนของการรับแพ็กเก็ต IPv4 ของ lwB4 มีรายละเอียดดังแสดงในรูปที่ 3-3




รูปที่ 3-3 ขั้นตอนของการรับแพ็กเก็ต IPv4 ของ lwB4

#### 3.3.2.3 lwB4 Control Plan Behavior

การควบคุมการทำงานของ elw4over6 ต้องดำเนินการควบคุมทั้งขั้นตอนการจัดหาข้อมูลที่จำเป็นสำหรับ lwB4 และขั้นตอนการสร้างอุโมงค์สื่อสารโดยตรงไปยัง lwB4 ปลายทาง ดังนั้น elw4over6 นำทั้ง ICMP และ DHCP มาประยุกต์ใช้ในการควบคุมการสร้างอุโมงค์สื่อสารดังกล่าว โดย ICMP ถูกใช้สำหรับระบุข้อผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลผ่านอุโมงค์สื่อสาร และ DHCP ถูกใช้สำหรับแลกเปลี่ยนข้อมูลของอุโมงค์สื่อสาร โดยการใช้งาน ICMP และ DHCP ที่เกี่ยวข้องทั้งหมดได้อธิบายรายละเอียดในหัวข้อ 3.4 โพรโตคอลและข้อมูลที่เกี่ยวข้อง

### 3.3.3 หลักการทำงานของ DHCP 4o6 Server ใน elw4over6

DHCP 4o6 Server ทำหน้าที่จัดสรรข้อมูล IPv4 และข้อมูลที่จำเป็นอื่นๆ ให้กับ lwB4 ดังนั้น DHCP 4o6 Server ใน elw4over6 ต้องรองรับหลักการทำงานพื้นฐานทั้งหมดของ DHCP 4o6 Server ใน lw4over6 นอกจากนั้น DHCP 4o6 Server ใน elw4over6 ต้องรองรับการร้องขอข้อมูลอุโมงค์สื่อสารที่ใช้สำหรับเชื่อมต่อไปยัง lwB4 ในแต่ละเครือข่ายผู้ใช้งานอีกด้วย ดังนั้น การทำงานของ DHCP 4o6 Server ใน elw4over6 ต้องรองรับ DHCP message และ DHCP option ที่เพิ่มขึ้น ยกตัวอย่างเช่น DHCPv4 Leasequery ซึ่งร้องขอข้อมูลโดยระบุหมายเลข IPv4 และพอร์ต, DHCPv4 Subnet Allocation Option และ DHCPv4 Relay Agent Information ซึ่งการใช้งาน DHCP ได้อธิบายรายละเอียดในหัวข้อที่ 3.4.2 DHCP

## 3.4 โพรโตคอลและข้อมูลที่เกี่ยวข้องใน elw4over6

หัวข้อนี้จะกล่าวถึงโพรโตคอลและข้อมูลที่ถูกนำมาใช้สำหรับปรับปรุงข้อมูลอุโมงค์สื่อสารให้เป็นปัจจุบันและจัดการกับเหตุการณ์ที่เกิดขึ้นใน elw4over6 โพรโตคอลที่ถูกนำมาใช้งานเป็นหลักคือโพรโตคอลที่ถูกนิยามโดย lw4over6 และโพรโตคอลอื่นๆ ถูกนำมาประยุกต์ใช้เพิ่มเติมเพียงบางส่วนเท่านั้น โดยมีข้อมูลบางส่วนที่ถูกนิยามขึ้นใหม่ซึ่งประกอบไปด้วย DHCPv4 option บางอ็อปชันและข้อมูลเพิ่มเติมของ lwB4 ซึ่งอธิบายในหัวข้อ lwB4 Information เพื่อให้สามารถรองรับการแลกเปลี่ยนข้อมูลเพิ่มเติมของ elw4over6 ได้อย่างสมบูรณ์ โพรโตคอลและข้อมูลที่ถูกนำมาใช้ภายใน elw4over6 สามารถแบ่งออกเป็น 3 กลุ่มซึ่งประกอบด้วย ICMP, DHCP และ lwB4 Information โดย ICMP ใช้สำหรับควบคุมการใช้งานของอุโมงค์สื่อสาร ส่วน DHCP ใช้สำหรับแลกเปลี่ยนข้อมูลของอุโมงค์สื่อสารและข้อมูลอื่นๆ ที่จำเป็น และ lwB4 Information เป็นข้อมูลที่ lwB4 ต้องการเพิ่มเติมจากที่ได้นิยามใน lw4over6 ซึ่งรายละเอียดของ ICMP, DHCP และ lwB4 Information มีดังต่อไปนี้

### 3.4.1 ICMP

ICMP ถูกนำมาใช้เพื่อควบคุมความถูกต้องของการใช้งานทั้งหมายเลข IPv6, หมายเลข IPv4 รวมไปถึงการใช้งานระดับพอร์ต ด้วยเหตุนี้ทั้ง ICMPv4 และ ICMPv6 จึงถูกนำมาใช้งานควบคู่ไปด้วยกัน โดย ICMP ถูกกำหนดให้นำมาใช้งานเพิ่มเติมโดย elw4over6 ประกอบด้วย ICMPv6 type 1 code 0, type 4 code 1 เท่านั้น ส่วน ICMP อื่นๆ นอกจากนี้ล้วนถูกกำหนดให้นำมาใช้งานโดย lw4over6 ทั้งสิ้น รายละเอียดการใช้งานของ ICMP ดังกล่าวมีดังต่อไปนี้

#### 3.4.1.1 ICMPv4 type 3 code 1 (Host Unreachable)

ICMPv4 type 3 code 1 ถูกใช้สำหรับตอบกลับไปยังเครื่องต้นทาง เมื่อเราเตอร์ไม่สามารถส่งต่อแพ็กเก็ตไปยังเครื่องปลายทางได้ ICMPv4 type 3 code 1 ใน elw4over6 มีการใช้งานลักษณะเดียวกันกับ lw4over6 โดย ICMPv4 type 3 code 1 ถูกส่งจาก lwAFTR ไปยังเครื่องต้นทางภายในอินเทอร์เน็ต เมื่อเครื่องต้นทางพยายามติดต่อไปยังหมายเลข IPv4 และพอร์ตของเครื่องปลายทางที่ยังไม่ได้ถูกจัดสรรสำหรับใช้งาน

#### 3.4.1.2 ICMPv4 type 3 code 13 (Communication Administratively Prohibited)

ICMPv4 type 3 code 13 ถูกใช้ใน 2 กรณี ได้แก่ กรณีแรกใช้สำหรับตอบจาก lwB4 ปลายทางกลับไปยัง lwB4 ต้นทาง และกรณีที่สองใช้สำหรับตอบจาก lwB4 ปลายทางกลับไปยัง lwAFTR เพื่อบ่งบอกว่าหมายเลข IPv4 หรือพอร์ตของเครื่องปลายทางไม่ถูกต้อง โดย lwB4 ต้นทางที่ได้รับ ICMPv4 type 3 code 13 ต้องตรวจสอบข้อมูลของอุโมงค์สื่อสารดังกล่าวใหม่อีกครั้ง เพื่อป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในการส่งครั้งต่อไป

#### 3.4.1.3 ICMPv6 type 1 code 0 (no route to destination)

ICMPv6 type 1 code 0 ถูกนำมางานใช้เฉพาะใน elw4over6 เท่านั้น โดย ICMPv6 type 1 code 0 ถูกใช้สำหรับตอบกลับไปยัง lwB4 ต้นทาง เมื่อไม่สามารถค้นหาเส้นทางไป

ยัง lwB4 ปลายทางได้ ในกรณีที่ lwB4 ต้นทางพยายามที่จะติดต่อไปยัง lwB4 ปลายทางโดยตรง แต่ lwB4 ปลายทางมีการยุติการใช้งานไปก่อนที่ระยะเวลา lease time จะสิ้นสุดลง เมื่อ lwB4 ต้นทางได้รับ ICMPv6 type 1 code 0 ตอบกลับมา lwB4 จะต้องตรวจสอบข้อมูลของอุโมงค์สื่อสารดังกล่าวใหม่อีกครั้ง เพื่อนำข้อมูลที่ได้มาปรับปรุงข้อมูลภายใน Bypass Binding Table ให้ถูกต้อง

#### 3.4.1.4 ICMPv6 type 1 code 5 (source address failed ingress/egress policy)

ICMPv6 type 1 code 5 ถูกใช้สำหรับตอบกลับจาก lwAFTR ไปยัง lwB4 ต้นทาง เมื่อ lwB4 ต้นทางมีการใช้งานหมายเลข IPv6, IPv4 หรือพอร์ตไม่ตรงกับการจัดสรรที่บันทึกไว้โดย lwAFTR ICMPv6 type 1 code 5 ใน elw4over6 มีการใช้งานลักษณะเดียวกันกับ lw4over6 ICMPv6 type 1 code 5 ถูกส่งจาก lwAFTR ไปยัง lwB4 ต้นทาง เมื่อ lwB4 ต้นทางได้รับ ICMPv6 type 1 code 5 lwB4 ต้องเริ่มต้นขั้นตอนการร้องขอหมายเลข IPv4 และ port-set ใหม่อีกครั้ง

#### 3.4.1.5 ICMPv6 type 4 code 1 (unrecognized Next Header type encountered)

ICMPv6 type 4 code 1 ถูกนำมาใช้งานเฉพาะใน elw4over6 เท่านั้น โดย ICMPv6 type 4 code 1 ถูกส่งจากเราเตอร์ใดๆ ตอบกลับไปยัง lwB4 ต้นทาง เมื่อเราเตอร์ดังกล่าวไม่รองรับการสร้างอุโมงค์สื่อสาร เหตุการณ์นี้จะเกิดขึ้นเมื่อมีการนำหมายเลข IPv6 ของ lwB4 ที่ยุติการใช้งานไปก่อนที่จะสิ้นสุดระยะเวลา lease time ไปจัดสรรให้อุปกรณ์ที่ไม่ใช่ lwB4 เมื่อเหตุการณ์นี้เกิดขึ้น lwB4 จะต้องตรวจสอบข้อมูลของอุโมงค์สื่อสารดังกล่าวใหม่อีกครั้ง เพื่อนำข้อมูลที่ได้มาปรับปรุงข้อมูลภายใน Bypass Binding Table ให้ถูกต้อง

### 3.4.2 DHCP

DHCP ใน lw4over6 ถูกนำมาใช้สำหรับจัดสรรข้อมูลให้กับ lwB4 เฉพาะขั้นตอนการร้องขอหมายเลข IPv4 และ port-set เท่านั้น แต่ DHCP ใน elw4over6 ไม่เพียงถูกใช้ขั้นตอนการร้องขอหมายลเลข IPv4 และ port-set แต่ยังถูกนำมาใช้ในการร้องขอข้อมูลของอุโมงค์สื่อสารปลายทาง เพื่อใช้สำหรับสร้างการเชื่อมต่อโดยตรงอีกด้วย สำหรับ elw4over6 ใช้ DHCPv6 และ DHCPv4 over DHCPv6 เพื่อจัดสรรข้อมูลที่เกี่ยวข้องให้กับ lwB4 เช่นเดียวกับ lw4over6 โดยที่ elw4over6 มีการปรับปรุง DHCP message และ DHCP option เพิ่มเติมเพื่อให้รองรับการสร้างอุโมงค์สื่อสารโดยตรง รายละเอียดของ DHCP ที่เกี่ยวข้องทั้งหมดมีดังต่อไปนี้

#### 3.4.2.1 DHCPv4 over DHCPv6

DHCPv4 over DHCPv6 ถูกออกแบบเพื่อใช้สำหรับจัดสรรข้อมูล IPv4 ผ่าน DHCPv6 ข้อมูลของ DHCPv4 message จะถูกบรรจุภายใน DHCPv6 option ชื่อ OPTION_DHCPv4_MSG อย่างไรก็ตามการใช้งาน DHCPv6 OPTION_DHCPv4_MSG ต้องถูกส่งผ่าน DHCPv6 message ที่นิยามใหม่โดยเฉพาะ โดย DHCPv6 message ที่ถูกนิยามขึ้นทั้ง 2 ชนิด ประกอบด้วย DHCPV4-QUERY และ DHCPV4-RESPONSE นอกจากนี้ DHCPv6 message ใหม่ก็ยังไม่สามารถเริ่มต้นการใช้งานได้ หาก DHCP client ไม่ได้รับข้อมูล DHCPv6 OPTION_DHCP4_O_DHCP6_SERVER กล่าวโดยสรุป DHCPv4 over DHCPv6 คือการแลกเปลี่ยนข้อมูล IPv4 ด้วย

DHCPv6 message และ DHCPv6 option ที่นิยามขึ้นมาสำหรับใช้งานโดยเฉพาะซึ่ง DHCPv6 option ใน DHCPv4 over DHCPv6 ได้แก่ OPTION_DHCP4_O_DHCP6_SERVER และ OPTION _DHCPv4_MSG

### 3.4.2.2 DHCPv6 message

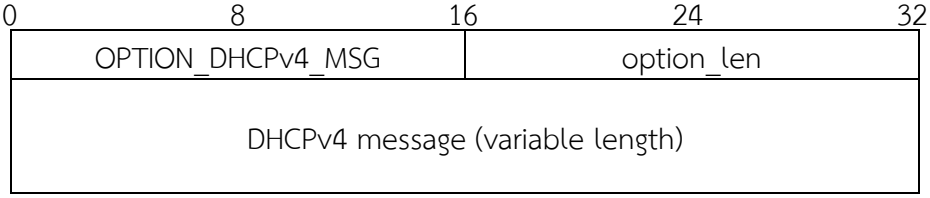
การแลกเปลี่ยนข้อมูลทั้งหมดของ elw4over6 ดำเนินการผ่าน DHCPv6 สำหรับข้อมูล IPv6 สามารถแลกเปลี่ยนผ่าน DHCPv6 message ทั่วไป แต่สำหรับข้อมูล IPv4 ต้องแลกเปลี่ยนผ่าน DHCPv4 over DHCPv6 โดยใช้ DHCPv6 DHCPV4-QUERY message และ DHCPV4-RESPONSE message เท่านั้น DHCPv6 message และ DHCPv6 option ที่เกี่ยวข้องเป็นไปตามข้อกำหนดใน Dynamic IPv4 Provisioning for Lightweight 4over6 [14], DHCPv4-over-DHCPv6 Transport [15] และ Dynamic Host Configuration Protocol for IPv6 โดย DHCPv6 option ที่เกี่ยวข้องแสดงรายละเอียดในหัวข้อถัดไป

### 3.4.2.3 DHCPv6 option

- OPTION_S46_CONT_DHCP4O6 เป็นกล่องของอ็อปชันที่ใช้สำหรับรวบรวมกลุ่มอ็อปชันที่เกี่ยวข้องเข้าด้วยกัน OPTION_S46_CONT_DHCP4O6 ใช้สำหรับบรรจุข้อมูลของ OPTION_S46_BR, OPTION_S46_DHCP4O6_SADDR_HINT และ OPTION_S46_DHCP4O6 _SADDR เพื่อประกาศหมายเลข IPv6 ของ lwAFTR ให้กับ lwB4 และใช้ในการลงทะเบียนหมายเลข IPv6 ของ lwB4 เพื่อใช้ร้องขอหมายเลข IPv4 และ port-set ต่อไป รายละเอียดของ OPTION _S46_CONT_DHCP4O6 สามารถดูได้จากหัวข้อที่ 2.2.3 Lightweight 4over6

- OPTION_DHCPv4_OVER_DHCPv6_SERVER มีค่า option_code เท่ากับ 88 OPTION_DHCPv4_OVER_DHCPv6_SERVER ใช้สำหรับตอบข้อมูลหมายเลข IPv6 ของ DHCP 4O6 Server จาก DHCP server กลับไปยัง DHCP client เพื่อใช้สำหรับติดต่อสื่อสารไปยัง DHCP 4O6 Server รายละเอียดของ OPTION_DHCPv4_OVER_DHCPv6_SERVER สามารถดูได้จากหัวข้อที่ 2.2.3 Lightweight 4over6

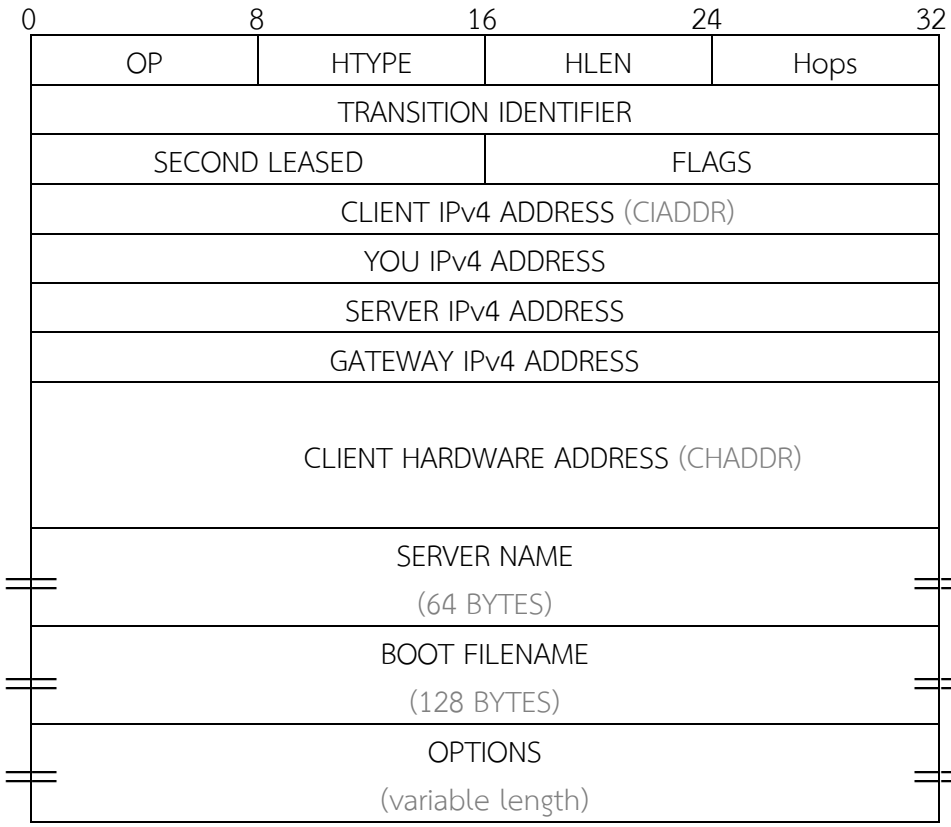
- OPTION_DHCPv4_MSG มีค่า option_code เท่ากับ 87 OPTION_DHCPv4 _MSG ใช้สำหรับส่งข้อมูล DHCPv4 message ระหว่าง DHCP 4o6 client และ DHCP 4o6 server DHCPv4 message ภายใน OPTION_DHCPv4_MSG มีเพียงส่วนของ DHCPv4 message เท่านั้น ไม่รวมถึง IPv4 header และ UDP header รูปแบบ OPTION_DHCPv4_MSG มีรายละเอียดดังรูปที่ 3-4




รูปที่ 3-4 รูปแบบ DHCPv6 OPTION_DHCPv4_MSG

### 3.4.2.4 DHCPv4 message

DHCPv4 ถูกใช้สำหรับแลกเปลี่ยนข้อมูล IPv4 และปรับปรุงข้อมูลของอุโมงค์สื่อสารปลายทางของ elw4over6 โดยอาศัย DHCPv4 over DHCPv6 สำหรับขั้นตอนการกำหนดข้อมูล IPv4 มีการดำเนินการเช่นเดียวกับ lw4over6 การดำเนินการที่ elw4over6 นิยามเพิ่มเติมมีเพียงการปรับปรุงข้อมูลของอุโมงค์สื่อสารปลายทางเพื่อสร้างอุโมงค์สื่อสารโดยตรงเท่านั้น elw4over6 นำ DHCPv4 Leasequery มาประยุกต์ใช้ในการแลกเปลี่ยนข้อมูลของอุโมงค์สื่อสาร แต่เดิมนั้น DHCPv4 Leasequery ถูกออกแบบมาเพื่อให้ DHCPv4 relay ร้องขอข้อมูลที่ต้องการไปยัง DHCPv4 server เพื่อปรับปรุงข้อมูลให้สอดคล้องกัน [30] DHCPv4 Leasequery ประกอบด้วย DHCPv4 message ทั้งหมด 4 ชนิดด้วยกัน ได้แก่ Leasequery, Leaseactive, Leaseunassigned และ Leaseunknown ดังนั้นเพื่อให้การปรับปรุงข้อมูลของอุโมงค์สื่อสารของ lwB4 ใน elw4over6 สามารถดำเนินการได้อย่างถูกต้องและรวดเร็ว การปรับปรุงข้อมูลอุโมงค์สื่อสารดังกล่าวจึงดำเนินการโดยอาศัย DHCPv4 Leasequery ซึ่ง DHCPv4 Leasequery ดังกล่าวมีรายละเอียดการใช้งานดังต่อไปนี้




รูปที่ 3-5 รูปแบบ DHCPv4 message

- Leasequery message ถูกส่งโดย DHCP relay เพื่อร้องข้อข้อมูลของ DHCP client ที่สนใจไปยัง DHCP server การร้องขอข้อมูล DHCP client ที่ต้องการสามารถระบุโดยใช้หมายเลข IPv4, หมายเลข MAC หรือ Client-identifier ซึ่งการร้องข้อโดยใช้หมายเลข IPv4 มีข้อกำหนดว่าฟิลด์ htype, hlen และ chaddr ของ DHCPv4 message ต้องกำหนดค่าเป็น 0 ฟิลด์

ciaddr ของ DHCPv4 message ต้องกำหนดค่าเป็นหมายเลข IPv4 ที่ต้องการร้องขอ และไม่ระบุ Client-identifier option ส่วนการร้องข้อโดยใช้หมายเลข MAC มีข้อกำหนดว่าฟิลด์ htype, hlen และ chaddr ต้องกำหนดเป็นค่าสำหรับหมายเลข MAC ที่ต้องการร้องขอ ฟิลด์ ciaddr ต้องกำหนดค่าเป็น 0 และไม่ระบุ Client-identifier option และสำหรับการร้องข้อโดยใช้ Client-identifier มีข้อกำหนดว่าฟิลด์ htype, hlen และ chaddr ต้องกำหนดค่าเป็น 0 ฟิลด์ ciaddr ต้องกำหนดค่าเป็น 0 และต้องระบุ Client-identifier option

สำหรับ DHCPv4 Leasequery message ใน elw4over6 ต้องการร้องขอโดยใช้หมายเลข IPv4 และพอร์ต โดยประยุกต์จากการร้องขอโดยใช้หมายเลข IPv4 การร้องข้อโดยใช้หมายเลข IPv4 และพอร์ต กำหนดว่าฟิลด์ htype, hlen และ chaddr กำหนดค่าเป็น 0 ฟิลด์ ciaddr ต้องกำหนดค่าเป็นหมายเลข IPv4 ที่ต้องการร้องขอ และไม่ระบุ Client-identifier option เช่นเดียวกับการร้องขอโดยใช้หมายเลข IPv4 แต่การร้องขอโดยใช้หมายเลข IPv4 และพอร์ตต้องระบุพอร์ตเพิ่มเติม สำหรับการระบุพอร์ตกำหนดให้ระบุ DHCPv4 OPTION_v4_PORTPARAMS โดยกำหนดค่าฟิลด์ offset เท่ากับ 0, PSID-len เท่ากับ 16 และ PSID เท่ากับพอร์ตที่ต้องการ ยกเว้นในบางกรณีที่ต้องการร้องขอข้อมูล port-set อนุญาตให้สามารถกำหนด offset และ PSID-len ตามข้อมูลของ port-set ที่ต้องการได้ รูปแบบ DHCPv4 message มีรายละเอียดดังแสดงในรูปที่ 3-5

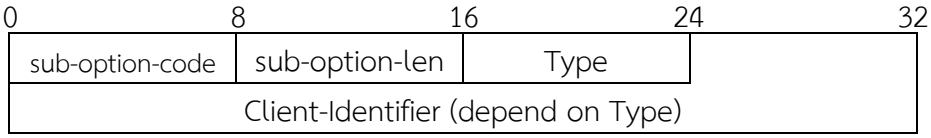
- Leaseactive message ถูกส่งโดย DHCP server เพื่อตอบกลับข้อมูลของ DHCP client ที่ถูกร้องขอไปยัง DHCP relay การตอบกลับด้วย Leaseactive message มีความหมายว่า DHCP client ที่ถูกร้องขอได้รับการจัดสรรหมายเลข IPv4 ออกไปโดย DHCP server และยังไม่หมดเวลาการใช้งาน สำหรับ DHCPv4 Leaseactive message ใน elw4over6 ก็ถูกนำมาใช้ในลักษณะเดียวกัน แต่ใช้สำหรับตอบข้อมูลของอุโมงค์สื่อสารปลายทางซึ่งถูกร้องขอโดยใช้หมายเลข IPv4 และพอร์ต ข้อมูลของอุโมงค์สื่อสารปลายทางจะถูกบรรจุภายใน DHCPv4 OPTION_ASSOCIATED_HOST_INFO ซึ่ง OPTION_ASSOCIATED_HOST_INFO ถูกนิยามขึ้นใหม่ เมื่อ lwB4 ได้รับ Leaseactive message ตอบกลับ lwB4 ต้องเพิ่มข้อมูลของอุโมงค์สื่อสารปลายทางดังกล่าวใน Bypass Binding Table

- Leaseunassigned message ถูกส่งโดย DHCP server เพื่อตอบกลับข้อมูลของ DHCP client ที่ถูกร้องขอไปยัง DHCP relay การตอบกลับด้วย Leaseunassigned message มีความหมายว่า DHCP client ที่ถูกร้องขอยังไม่มีการจัดสรรหมายเลข IPv4 สำหรับใช้งาน หรือถูกยกเลิกการใช้งานไปแล้ว สำหรับ DHCPv4 Leaseunassigned message ใน elw4over6 ก็ถูกนำมาใช้สำหรับตอบข้อมูลของอุโมงค์สื่อสารปลายทางซึ่งถูกร้องขอโดยใช้หมายเลข IPv4 และพอร์ต เพื่อบ่งชี้ว่าอุโมงค์สื่อสารดังกล่าวยังไม่ได้รับการจัดสรรสำหรับใช้งาน เมื่อ lwB4 ได้รับ Leaseunassigned message ตอบกลับ lwB4 ต้องลบข้อมูลอุโมงค์สื่อสารปลายทางที่ตรงกับหมายเลข IPv4 และพอร์ตดังกล่าว พร้อมกับเพิ่มค่า Next Query Time ภายใน Requested Destination Table เพื่อยืดเวลาในการร้องขอข้อมูลครั้งถัดไปออกไป

● Leaseunknown message ถูกส่งโดย DHCP server เพื่อตอบกลับข้อมูลของ DHCP client ที่ถูกร้องขอไปยัง DHCP relay การตอบกลับด้วย Leaseunknown message มีความหมายว่า DHCP client ที่ถูกร้องขอไม่ตรงกับข้อกำหนดของ DHCP client ที่อนุญาตไว้ สำหรับ DHCPv4 Leaseunknown message ใน elw4over6 ก็ถูกนำมาใช้สำหรับตอบข้อมูลของอุโมงค์สื่อสารปลายทางซึ่งถูกร้องขอโดยใช้หมายเลข IPv4 และพอร์ต เพื่อบ่งบอกว่าอุโมงค์สื่อสารดังกล่าวไม่อยู่ภายในช่วงของ Bypass Scope หรือข้อมูลของ lwB4 ที่ทำการร้องขอไม่สอดคล้องกับข้อมูลการจัดสรรที่ lwAFTR ได้ทำการบันทึก ทั้งสองเหตุการณ์แสดงให้เห็นว่า lwB4 มีข้อมูลไม่ตรงกับข้อมูลของ elw4over6 ปัจจุบัน เมื่อ lwB4 ได้รับ Leaseunknown message ตอบกลับ lwB4 ต้องเริ่มต้นขั้นตอนการดำเนินการลงทะเบียนและร้องขอข้อมูล IPv4 ใน elw4over6 ใหม่ทั้งหมด

### 3.4.2.5 DHCPv4 option

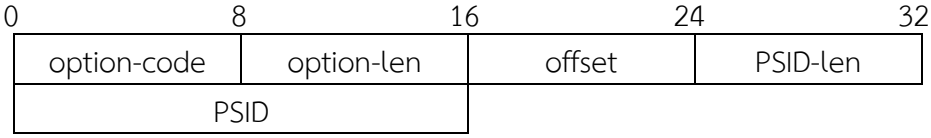
● Relay Agent Information (option code 82) ถูกใช้สำหรับระบุตัวตนสำหรับ DHCP relay ที่เป็นผู้ร้องขอข้อมูล Relay Agent Information ใน elw4over6 ได้นิยาม sub-option ใหม่เรียกว่า “Client-identifier” เพื่อใช้ในการระบุตัวตนของ lwB4 โดย Client-identifier เดิมนั้นได้ถูกนิยามเป็น DHCPv4 option อยู่ก่อนแล้ว แต่เนื่องจาก DHCPv4 Leasequery ได้นิยามการร้องขอข้อมูลด้วย Client-identifier ส่งผลให้ไม่สามารถนำ Client-identifier มาใช้เพื่อระบุตัวตนของผู้ร้องขอได้ ดังนั้น elw4over6 จำเป็นต้องนิยาม Client-identifier เป็น sub-option ของ Relay Agent Information โดย Client-identifier sub-option ของ DHCPv4 Relay Agent Information มีรูปแบบเช่นเดียวกับ Client-identifier option ทุกประการ ซึ่งแสดงรายละเอียดดังรูปที่ 3-6




รูปที่ 3-6 รูปแบบ Client-identifier sub-option ของ DHCPv4 Relay Agent Information

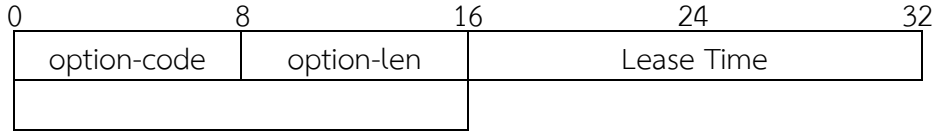
จากรูปที่ 3 6 รูปแบบ Client-identifier sub-option ของ DHCPv4 Relay Agent Information มีฟิลด์ข้อมูล 4 ฟิลด์ ประกอบด้วย sub-option_code ขนาด 8 บิต, sub-option_length ขนาด 8 บิต, Type ขนาด 8 บิตและ Client-Identifier ซึ่งมีขนาดและข้อมูลแปรผันตาม Type โดย Type ของ Client-Identifier ที่นำมาใช้ระบุตัวตนใน elw4over6 คือ Type 255 ซึ่งข้อมูลภายใน Client-Identifier ของ Type 255 บรรจุ IAID และ DUID ตามลำดับ

● OPTION_v4_PORTPARAMS ใช้สำหรับระบุข้อมูลที่เกี่ยวข้องกับพอร์ตของหมายเลข IPv4 ซึ่งกำหนดช่วงของพอร์ตตามหลักการของ port mapping algorithm รูปแบบของ OPTION_v4_PORTPARAMS มีรายละเอียดดังรูปที่ 3-7



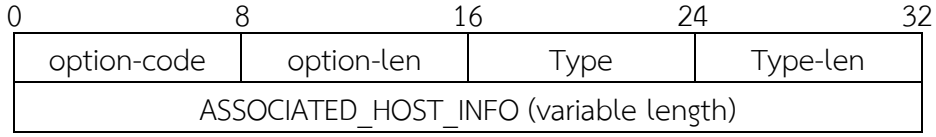

รูปที่ 3-7 รูปแบบ DHCPv4 OPTION_v4_PORTPARAMS

- IP Address Lease Time (option code 51) ถูกใช้สำหรับระบุระยะเวลาที่สามารถครอบครองหมายเลข IPv4 ที่ได้รับการจัดสรร IP Address Lease Time มีค่า option-code เท่ากับ 51, option-length มีค่าเท่ากับ 4 และ Lease Time มีขนาด 32 บิตซึ่งภายในระบุจำนวนเวลาในหน่วยวินาที รูปแบบของ IP Address Lease Time มีรายละเอียดดังรูปที่ 3-8



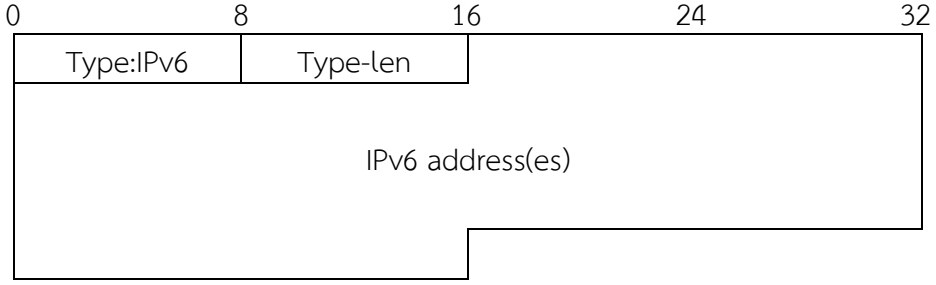

รูปที่ 3-8 รูปแบบ DHCPv4 IP Address Lease Time

- OPTION_ASSOCIATED_HOST_INFO ถูกนิยามขึ้นเพื่อนำมาใช้ใน elw4over6 โดยเฉพาะ OPTION_ASSOCIATED_HOST_INFO ถูกใช้เพื่อตอบกลับข้อมูลเพิ่มเติมที่เกี่ยวข้องกับโฮสที่ถูกร้อง ซึ่งข้อมูลดังกล่าวถูกระบุโดยใช้ Type และ Type-len เพื่อให้สามารถระบุข้อมูลได้อย่างหลากหลายและรองรับข้อมูลรูปแบบใหม่เพิ่มเติมในอนาคต รูปแบบ OPTION_ASSOCIATED_HOST_INFO แสดงรายละเอียดดังรูปที่ 3-9




รูปที่ 3-9 รูปแบบ DHCPv4 OPTION_ASSOCIATED_HOST_INFO

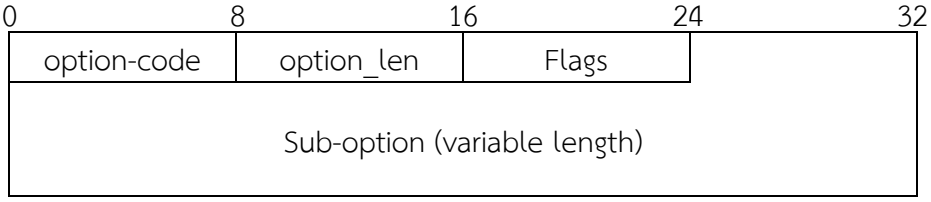
Type ที่ถูกนิยามใน DHCPv4 OPTION_ASSOCIATED_HOST_INFO ในขณะนี้มีเพียง IPv6 ซึ่งนำมาใช้เพื่อระบุข้อมูลหมายเลข IPv6 ของโฮส รูปแบบของ Type: IPv6 มีรายละเอียดดังรูปที่ 3-10




รูปที่ 3-10 รูปแบบ Type:IPv6 ของ DHCPv4 OPTION_ASSOCIATED_HOST_INFO

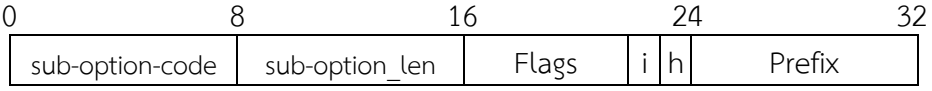
จากรูปที่ 3-10 รูปแบบ Type:IPv6 ของ DHCPv4 OPTION_ASSOCIATED _HOST_INFO มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย Type ขนาด 8 บิต, Type_length ขนาด 8 บิต, และหมายเลข IPv6 ขนาดเท่ากับจำนวนเท่าของ 128 บิต เพื่อใช้ส่งข้อมูลในกรณีที่มี IPv6 หลายหมายเลข

- Subnet Allocation Option (option code 220) ใช้สำหรับจัดสรร IPv4 subnet ให้กับ DHCPv4 client [31] Subnet Allocation Option ถูกนำมาประยุกต์ใช้ใน elw4over6 เพื่อส่งข้อมูล Bypass Scope รูปแบบ Subnet Allocation Option มีรายละเอียดดังแสดงในรูปที่ 3-11



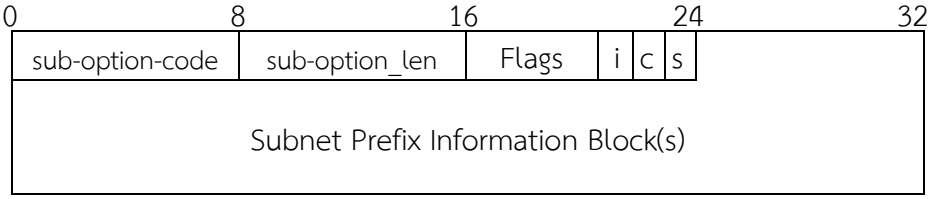

รูปที่ 3-11 รูปแบบ DHCPv4 Subnet Allocation Option

sub-option ของ Subnet Allocation Option ที่เกี่ยวข้องกับ elw4over6 มี 3 sub-option ได้แก่ Subnet-Request, Subnet-Information และ Subnet-Name ซึ่งรายละเอียด sub-option ของ Subnet Allocation Option มีดังต่อไปนี้



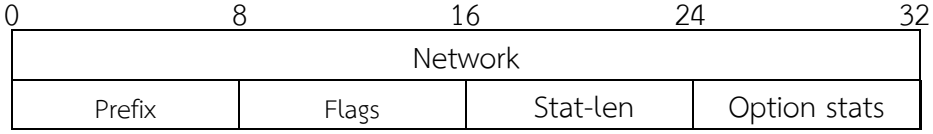

รูปที่ 3-12 รูปแบบ Subnet-Request sub-option ของ DHCPv4 Subnet Allocation Option

จากรูปที่ 3-12 รูปแบบ DHCPv4 Subnet-Request sub-option มีฟิลด์ข้อมูล 4 ฟิลด์ ประกอบด้วย sub-option_code ขนาด 8 บิตและมีค่าเท่ากับ 1, sub-option_length ขนาด 8 บิต, Flags ขนาด 8 บิตและ Prefix ขนาด 8 บิต โดย Flags “i” ใช้ระบุว่าเป็นการร้องขอข้อมูลที่ได้รับการจัดสรรไปแล้วก่อนหน้า และ Flags “h” ใช้ระบุความเป็นลำดับชั้น เมื่อกำหนด Flags “h” เป็น 1 หมายความว่า โหนดที่ร้องขอต้องการจัดสรรหมายเลข IPv4 โดยตรงโดยไม่ผ่าน DHCP server สำหรับ Prefix ใช้ระบุขนาดของ subnet ที่ต้องการ



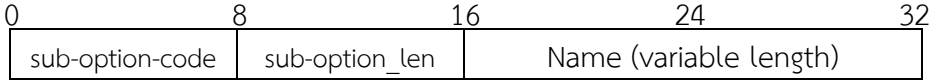

รูปที่ 3-13 รูปแบบ Subnet-Information sub-option
ของ DHCPv4 Subnet Allocation Option

จากรูปที่ 3-13 รูปแบบ DHCPv4 Subnet-Information sub-option มีฟิลด์ข้อมูล 4 ฟิลด์ ประกอบด้วย sub-option_code ขนาด 8 บิตและมีค่าเท่ากับ 2, sub-option_length ขนาด 8 บิต, Flags ขนาด 8 บิตและ Subnet Prefix Information Block โดย Flags “c” ใช้ระบุว่าข้อมูลภายใน Subnet Prefix Information Block เป็นข้อมูลที่ถูกร้องขอและ Flags “s” ใช้ระบุว่าข้อมูลภายใน Subnet Prefix Information Block เป็นข้อมูล subnet ที่ DHCP server จัดสรรเพิ่มเติมให้ นอกจากนี้ elw4over6 ได้นิยาม Flags “i” ใน Subnet-Information sub-option เพิ่มเติมเพื่อบ่งชี้ว่าข้อมูล subnet ที่ได้รับเป็นเพียงข้อมูลเท่านั้น ไม่สามารถนำมาใช้ในการจัดสรรหมายเลข IPv4 ได้ โดยที่ Flags “i” ดังกล่าวถูกนำมาใช้ในการประกาศข้อมูล Bypass Scope ต่อไป




รูปที่ 3-14 รูปแบบ Subnet Prefix Information Block

จากรูปที่ 3-14 รูปแบบ Subnet Prefix Information Block มีฟิลด์ข้อมูล 5 ฟิลด์ ประกอบด้วย Network ขนาด 32 บิต, Prefix ขนาด 8 บิต, Flags ขนาด 8 บิต, Stat-length ขนาด 8 บิตและ Option stats ขนาดเท่ากับ Stat-length แต่โดยทั่วไป Stat-length มีขนาดเท่ากับ 0




รูปที่ 3-15 รูปแบบ Subnet-Name sub-option ของ DHCPv4 Subnet Allocation Option

จากรูปที่ 3-15 รูปแบบ DHCPv4 Subnet-Name sub-option มีฟิลด์ข้อมูล 3 ฟิลด์ ประกอบด้วย sub-option_code ขนาด 8 บิตและมีค่าเท่ากับ 3, sub-option_length ขนาด 8 บิต และ Name ขนาดเท่ากับ sub-option_length โดย Name ถูกใช้สำหรับระบุชื่อของ subnet ในระหว่างการจัดสรร สำหรับการร้องขอข้อมูล Bypass Scope ใน elw4over6 กำหนดให้ระบุค่า Name ให้เท่ากับ “Bypass Scope”

### 3.4.3 lwB4 Information

lwB4 ต้องอาศัยข้อมูลเพิ่มเติมนอกเหนือจากข้อมูลหมายเลข IPv6 ของ lwAFTR, Public IPv4 address และ port-set เพื่อใช้ในการสร้างอุโมงค์สื่อสารโดยตรงของ elw4over6 ซึ่งข้อมูลที่ lwB4 ต้องการเพิ่มเติมประกอบด้วยข้อมูล 3 ชนิด ได้แก่ Bypass Scope, Bypass Binding Table และ Requested Destination Table โดยรายละเอียดของข้อมูลทั้ง 3 ชนิดมีดังต่อไปนี้

#### 3.4.3.1 Bypass Scope

Bypass Scope ใช้สำหรับจำแนกเครื่องปลายทาง เพื่อบ่งชี้ว่าเครื่องปลายทางดังกล่าวสามารถสร้างอุโมงค์สื่อสารโดยตรงได้หรือไม่ Bypass Scope เป็นช่วงของหมายเลข IPv4 ที่ถูกจัดสรรให้กับ lwB4 ใน elw4over6 ทั้งหมดภายในเครือข่ายของผู้ให้บริการ Bypass Scope จึงสามารถกรองแพ็กเก็ตของ lwB4 ใน elw4over6 ภายในเครือข่ายของผู้ให้บริการเพื่อเข้าสู่ขั้นตอนการสร้างอุโมงค์สื่อสารโดยตรง หมายเลข IPv4 และ port-set ภายใน Bypass Scope ต้องถูก

กำหนดให้กับ lwB4 ซึ่งรองรับ elw4over6 เท่านั้น ส่วน lwB4 ใน lw4over6 ให้จัดสรรหมายเลข IPv4 และ port-set ในช่วงนอกเหนือ Bypass Scope เพื่อให้ elw4over6 และ lw4over6 สามารถทำงานควบคู่ไปด้วยกัน

ข้อมูล Bypass Scope ถูกร้องขอผ่าน DHCPv4 Subnet Allocation Option โดยภายใน Subnet Allocation Option ของแพ็กเก็ต DHCP discover บรรจุ Subnet-Request และ Subnet-Name โดย Subnet-Request ต้องกำหนดค่า Flags “i” เท่ากับ 1 และค่า Prefix เท่ากับ 0 เพื่อระบุว่าเป็นการร้องขอข้อมูล subnet ที่ได้จัดสรรไว้ก่อนหน้านี้ ส่วน Subnet-Name กำหนดค่า Name เท่ากับอักขระ “Bypass Scope” เพื่อบ่งบอกไปยัง DHCP server ว่าข้อมูลของ subnet ที่ต้องการคือ Bypass Scope จากนั้น DHCP server ตอบแพ็กเก็ต DHCP offer ซึ่งบรรจุ Subnet-Information โดย Subnet-Information ต้องกำหนดค่า Flags “c” เท่ากับ 1, ค่า Flags “i” เท่ากับ 1 และค่า Subnet Prefix Information Block เป็นข้อมูล subnet ของ Bypass Scope Flags “c” เป็นการระบุว่าเป็นข้อมูลที่ได้รับการจัดสรร และ Flags “i” เป็นการระบุว่า subnet ที่ได้รับต้องนำมาใช้เป็นข้อมูลเท่านั้นไม่สามารถนำไปใช้ในการจัดสรรหมายเลข IPv4 ได้ หลังจากที่ DHCP client ได้รับแพ็กเก็ต DHCP offer DHCP client ส่ง DHCP request ซึ่งบรรจุ Subnet-Information จากนั้น DHCP server ก็จะตอบกลับด้วย DHCP acknowledge ซึ่งบรรจุ Subnet-Information กลับมายัง DHCP client เช่นกัน ในกรณีที่ Bypass Scope ไม่ถูกกำหนด lwB4 ใน elw4over6 สามารถปรับเปลี่ยนการดำเนินการตามหลักการทำงานของ lwB4 ใน lw4over6 เพื่อให้บริการต่อไป

### 3.4.3.2 Bypass Binding Table

Bypass Binding Table ใช้สำหรับบันทึกข้อมูลของอุโมงค์สื่อสารปลายทางต่างๆ ที่ดำเนินการร้องขอเสร็จสิ้น ข้อมูล Bypass Binding Table มีลักษณะคล้ายกับข้อมูลอุโมงค์สื่อสารปลายทางบน lwAFTR โดยทั้ง Bypass Binding Table ของ lwB4 และ Binding Table ของ lwAFTR มีการบันทึก lease time ของอุโมงค์สื่อสารปลายทางแต่ละอุโมงค์ ดังนั้นข้อมูลอุโมงค์สื่อสารปลายทางจะถูกลบทิ้งทันทีเมื่อสิ้นสุดระยะเวลา lease time ข้อมูลของ Bypass Binding Table ประกอบด้วยหมายเลข IPv6, IPv4, port-set และ lease time ซึ่งถูกปรับปรุงผ่านขั้นตอน Initial Bypass lw4over6 Process หรือ Terminative Bypass lw4over6 Process เท่านั้น โดยขั้นตอนการดำเนินการของทั้งสองขั้นตอนได้อธิบายรายละเอียดในหัวข้อ 3.5

### 3.4.3.3 Requested Destination Table

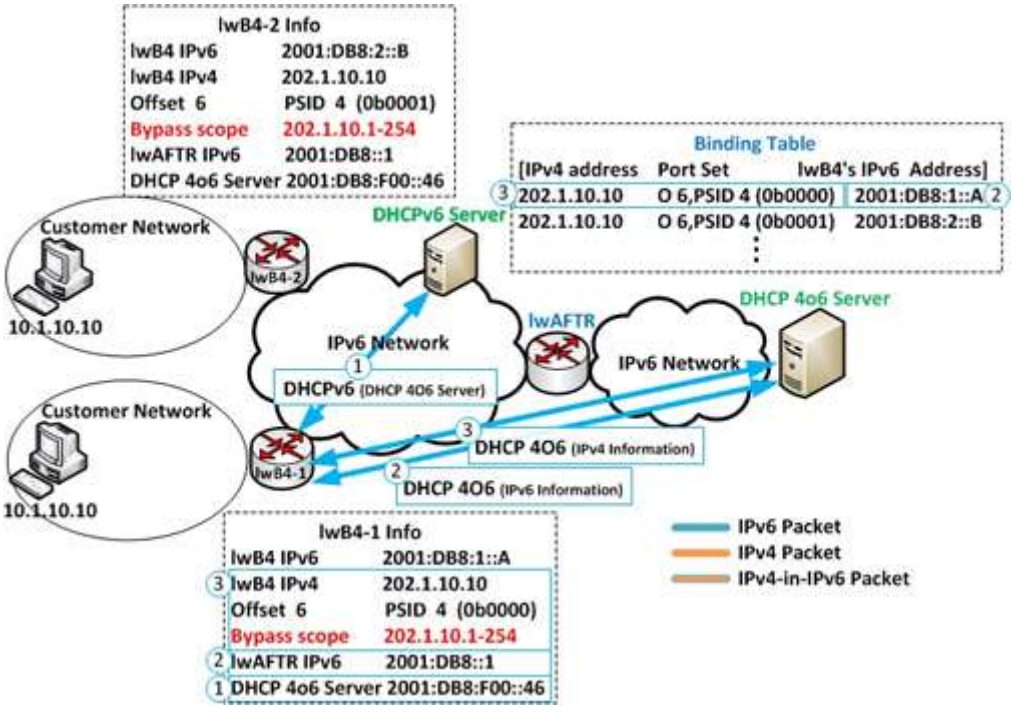
Requested Destination Table ใช้สำหรับบันทึกข้อมูลหมายเลข IPv4 และพอร์ตที่ผ่านการร้องขอข้อมูลอุโมงค์สื่อสารโดย Bypass Binding Table เพื่อป้องกันการร้องขอซ้ำภายในระยะเวลาอันสั้น Requested Destination Table มี Next Query Time เป็นตัวแปรสำหรับกำหนดระยะเวลาในการป้องกันการร้องข้อซ้ำ โดยเมื่อครบกำหนดเวลา ข้อมูลของหมายเลข IPv4 และพอร์ตจะถูกลบออกจากตารางเพื่อให้สามารถร้องขอได้อีกครั้ง ข้อมูลของ Requested Destination Table ประกอบด้วยหมายเลข IPv4, พอร์ตและ Next Query Time

## 3.5 ขั้นตอนการดำเนินการของ Enhancement of Lightweight 4over6

หัวข้อนี้จะอธิบายขั้นตอนการดำเนินการทั้งหมดภายใน elw4over6 ซึ่งขั้นตอนการดำเนินดังกล่าวแบ่งออกเป็น 3 ขั้นตอนด้วยกัน ได้แก่ Initial lw4over6 Process, Initial Bypass lw4over6 Process และ Terminative Bypass lw4over6 Process ขั้นตอนการดำเนินการแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

### 3.5.1 Initial lw4over6 Process

Initial lw4over6 Process ดำเนินการโดยใช้ DHCPv6 และ DHCPv4 over DHCPv6 เพื่อร้องขอข้อมูลที่เกี่ยวข้อง ในขั้นตอนนี้มีส่วนที่แตกต่างไปจาก lw4over6 เพียงส่วนเดียวคือการร้องขอข้อมูล Bypass Scope ผ่าน DHCPv4 Subnet Allocation Option เนื่องจาก Bypass Scope เป็นข้อมูลที่ใช้สำหรับอนุญาตให้สามารถดำเนินการ Initial Bypass lw4over6 Process และ Terminative Bypass lw4over6 Process เพื่อควบคุมการสร้างอุโมงค์สื่อสาร การดำเนินการของ Initial lw4over6 Process แบ่งออกเป็น 3 ขั้นตอนย่อยซึ่งประกอบไปด้วย ขั้นตอนการร้องขอข้อมูล DHCP 4o6 server, ขั้นตอนการร้องขอข้อมูล lwAFTR และลงทะเบียน lwB4 และขั้นตอนการร้องขอข้อมูล IPv4 ซึ่งขั้นตอนการดำเนินการของ Initial lw4over6 Process มีรายละเอียดดังรูปที่ 3-16




รูปที่ 3-16 ขั้นตอนการดำเนินการของ Initial lw4over6 Process

1) ขั้นตอนการร้องขอข้อมูล DHCP 4o6 server

- lwB4 ร้องขอหมายเลข IPv6 ของ DHCP 4o6 server ไปยัง DHCPv6 server ผ่าน DHCPv6 Solicit โดย DHCPv6 option ที่ถูกร้องขอประกอบไปด้วย OPTION_DHCP4_O_DHCP6_SERVER

- DHCPv6 server ตอบกลับ OPTION_DHCP4_O_DHCP6_SERVER ซึ่งบรรจุข้อมูลหมายเลข IPv6 ของ DHCP 4o6 server ผ่าน DHCPv6 Advertise ไปยัง lwB4
- lwB4 ร้องขอข้อมูลหมายเลข IPv6 ของ DHCP 4o6 server ไปยัง DHCPv6 server ผ่าน DHCPv6 Request
- DHCPv6 Server ตอบกลับ OPTION_DHCP4_O_DHCP6_SERVER ซึ่งบรรจุข้อมูลหมายเลข IPv6 ของ DHCP 4o6 server ผ่าน DHCPv6 Reply ไปยัง lwB4 หลังจากนั้น lwB4 จึงสามารถเริ่มต้นการติดต่อสื่อสารกับ DHCP 4o6 server

2) ขั้นตอนการร้องขอข้อมูล lwAFTR และลงทะเบียน lwB4

- lwB4 ร้องขอข้อมูล lwAFTR และลงทะเบียน lwB4 ไปยัง DHCP 4o6 server ผ่าน DHCPv6 Solicit โดยระบุ OPTION_CLIENTID (DUID) และ OPTION_IA_NA (IAID) เพื่อระบุตัวตนของ lwB4 สำหรับ DHCPv6 option ที่ถูกร้องขอประกอบไปด้วย OPTION_S46_CONT_DHCP4O6
- DHCP 4o6 server ตอบกลับข้อมูล lwAFTR และแนะนำหมายเลข IPv6 ของ lwB4 ที่ควรใช้สำหรับลงทะเบียน lwB4 ด้วย OPTION_S46_CONT_DHCP4O6 ซึ่งภายในบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR_HINT ผ่าน DHCPv6 Advertise ไปยัง lwB4
- lwB4 ร้องขอข้อมูล lwAFTR และระบุหมายเลข IPv6 ของ lwB4 ที่ต้องการลงทะเบียนไปยัง DHCP 4o6 server ผ่าน DHCPv6 Request โดยระบุ OPTION_CLIENTID, OPTION_IA_NA และ OPTION_S46_CONT_DHCP4O6
- DHCP 4o6 server ลงทะเบียนหมายเลข IPv6 ของ lwB4 โดยบันทึก DUID และ IAID จากนั้น DHCP 4o6 server จึงตอบกลับ OPTION_S46_CONT_DHCP4O6 ซึ่งบรรจุ OPTION_S46_BR และ OPTION_S46_DHCP4O6_SADDR ผ่าน DHCPv6 Reply ไปยัง lwB4

3) ขั้นตอนการร้องขอข้อมูล IPv4

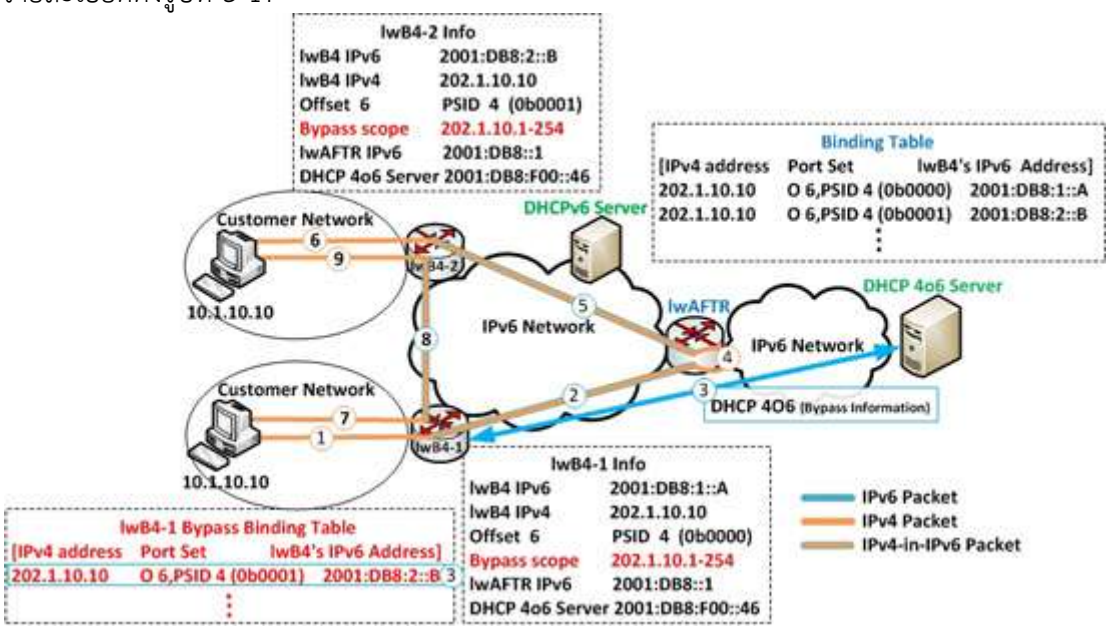
- lwB4 ร้องขอข้อมูล IPv4 ไปยัง DHCP 4o6 server ผ่าน DHCPv4 discover over DHCPv6 โดยระบุ Client Identifier (Type:255 IAID ,DUID) และ Subnet Allocation Option (Bypass Scope) สำหรับ DHCPv4 option ที่ถูกร้องขอได้แก่ OPTON_V4_PORTPARAMS
- จากนั้น DHCP 4o6 server นำ IAID และ DUID มาตรวจสอบกับหมายเลข IPv6 ของ lwB4 ซึ่งผ่านการลงทะเบียน หาก IAID และ DUID ลงทะเบียนอย่างถูกต้อง DHCP 4o6 server ตอบกลับหมายเลข IPv4 ที่

ได้รับการจัดสรร, OPTON_V4_PORTPARAMS, IP Address Lease Time และ Subnet Allocation Option (Bypass Scope) ผ่าน DHCPv4 offer over DHCPv6 ไปยัง lwB4

- lwB4 ร้องขอข้อมูล IPv4 ที่ได้รับการจัดสรรไปยัง DHCP 4o6 server ผ่าน DHCPv4 Request over DHCPv6 โดยระบุ Address Request, Client Identifier (Type:255 IAID ,DUID) และ Subnet Allocation Option (Bypass Scope) สำหรับ DHCPv4 option ที่ถูกร้องขอได้แก่ OPTON_V4_PORTPARAMS
- DHCP 4o6 server ตรวจสอบความถูกต้องของการร้องขอ จากนั้นจึงตอบกลับหมายเลข IPv4 ที่ได้รับการจัดสรร, OPTON_V4_PORTPARAMS, IP Address Lease Time และ Subnet Allocation Option (Bypass Scope) ผ่าน DHCPv4 acknowledge over DHCPv6 ไปยัง lwB4

### 3.5.2 Initial Bypass lw4over6 Process

Initial Bypass lw4over6 Process ถูกออกแบบเพื่อนำมาใช้งานใน elw4over6 โดยเฉพาะ ขั้นตอนนี้จะเริ่มดำเนินการเมื่อ lwB4 ได้รับแพ็กเก็ต IPv4 ที่มีปลายทางอยู่ภายใน Bypass Scope lwB4 ต้องดำเนินการร้องขอข้อมูลของอุโมงค์สื่อสารปลายทางไปยัง DHCP 4o6 server เมื่อ lwB4 ได้รับข้อมูลของอุโมงค์สื่อสารปลายทาง lwB4 จึงสามารถสร้างอุโมงค์สื่อสารไปยังเครือข่ายปลายทางโดยตรงได้ ขั้นตอนการดำเนินการของ Initial Bypass lw4over6 Process มีรายละเอียดดังรูปที่ 3-17




รูปที่ 3-17 ขั้นตอนการดำเนินการของ Initial Bypass lw4over6 Process

1) เมื่อ lwB4 ต้นทางได้รับแพ็กเก็ตที่มีหมายเลข IPv4 ปลายทางอยู่ภายใน Bypass Scope lwB4 ต้นทางต้องตรวจสอบข้อมูลภายใน Bypass Binding Table ว่ามีข้อมูลอุโมงค์สื่อสารโดยตรงไปยัง lwB4 ปลายทางหรือไม่?

2) lwB4 ต้นทางจะไม่มีข้อมูลอุโมงค์สื่อสารโดยตรงภายใน Bypass Binding Table เนื่องจากเป็นการส่งแพ็กเก็ต IPv4 ไปยังเครื่องปลายทางใหม่เป็นครั้งแรก lwB4 ต้องห่อหุ้มและส่งแพ็กเก็ตดังกล่าวไปยัง lwAFTR และตรวจสอบข้อมูลภายใน Requested Destination Table ว่ามีการร้องขอข้อมูลเครือข่ายของเครื่องปลายทางดังกล่าวไปแล้วหรือไม่?

3) เนื่องจากเป็นการส่งแพ็กเก็ตครั้งแรกจึงไม่มีการร้องขอข้อมูลของเครือข่ายปลายทาง lwB4 ต้องดำเนินร้องขอข้อมูลอุโมงค์สื่อสารปลายทางไปยัง DHCP 4o6 server โดยใช้ DHCPv4 Leasequery over DHCPv6 การร้องขอข้อมูลของเครื่องปลายทางด้วย DHCPv4 Leasequery ใช้การระบุหมายเลข IPv4 และพอร์ต ด้วยเหตุนี้ DHCPv4 ที่ถูกส่งไปต้องระบุ OPTION_v4_PORTPARAMS และ Relay Agent Information (Client Identifier) โดย DHCPv4 option ที่ถูกร้องขอประกอบไปด้วย OPTION_ASSOCIATED_HOST_INFO, OPTION _V4_PORTPARAMS เมื่อส่ง DHCPv4 Leasequery over DHCPv6 เสร็จสิ้น lwB4 ต้นทางต้องบันทึกหมายเลข IPv4 และพอร์ตลงใน Requested Destination Table

เมื่อ DHCP 4o6 server ได้รับ DHCPv4 Leasequery over DHCPv6 ขั้นตอนการตรวจสอบความถูกต้องของ lwB4 ต้นทางซึ่งเป็นผู้ร้องขอจึงเริ่มขึ้น หาก lwB4 ต้นทางผ่านการลงทะเบียนอย่างถูกต้อง DHCP 4o6 server จะตอบ DHCPv4 Leaseactive over DHCPv6 กลับมายัง lwB4 ต้นทาง เมื่อ lwB4 ต้นทางได้รับข้อมูลของอุโมงค์สื่อสารปลายทางผ่าน DHCPv4 Leaseactive over DHCPv6 ซึ่งประกอบไปด้วย OPTION_ASSOCIATED_HOST_INFO, OPTION _v4_PORTPARAMS และ IP Address Lease Time ข้อมูลภายใน DHCPv4 Leaseactive over DHCPv6 จะต้องถูกตรวจสอบความถูกต้องก่อนดำเนินการเพิ่มข้อมูลอุโมงค์สื่อสารปลายทางลงใน Bypass Binding Table

4) ในขณะเดียวกัน เมื่อ lwAFTR ได้รับแพ็กเก็ต IPv6 ที่ห่อหุ้มแพ็กเก็ต IPv4 อยู่ภายใน lwAFTR นำแพ็กเก็ต IPv4 ดั้งเดิมออกมาเพื่อส่งต่อไปยังเครื่องปลายทาง

5) แต่ในกรณีนี้เครื่องปลายทางเชื่อมต่อกับ lwB4-2 ซึ่งอยู่ภายในขอบเขตการให้บริการเดียวกัน แพ็กเก็ต IPv4 ต้องถูกห่อหุ้มภายในแพ็กเก็ต IPv6 แล้วส่งต่อไปยัง lwB4-2 ซึ่งเป็นเกตเวย์ของเครื่องปลายทางต่อไป

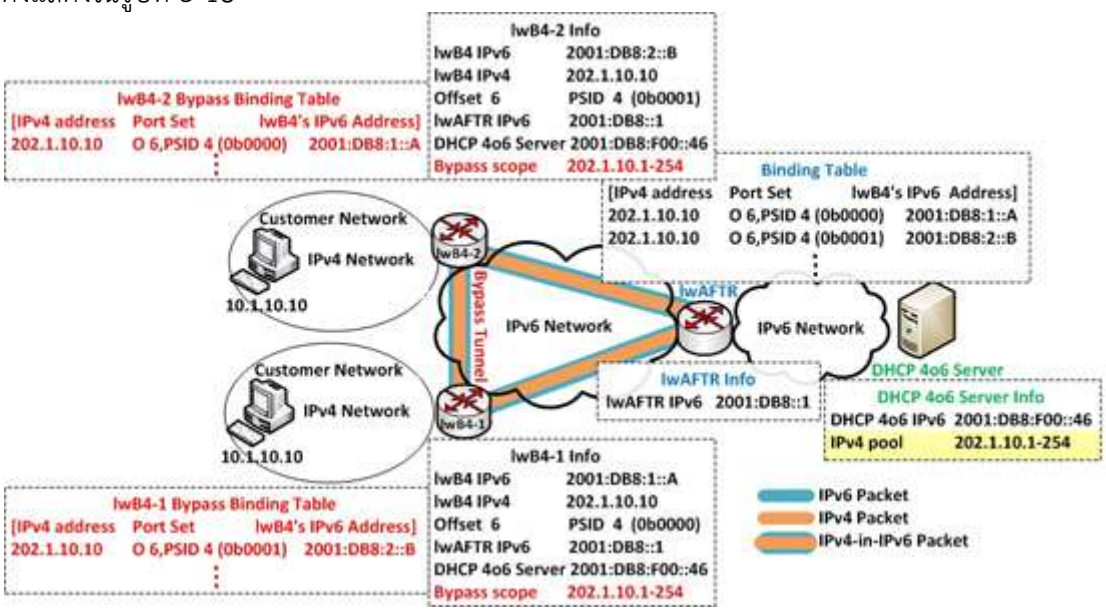
6) เมื่อ lwB4 ปลายทางได้รับแพ็กเก็ตที่ถูกห่อหุ้ม lwB4 ปลายทางต้องตรวจสอบความถูกต้องของเครื่องที่ทำหน้าที่เป็นอุโมงค์สื่อสารต้นทางและนำแพ็กเก็ต IPv4 ดั้งเดิมออกมาเพื่อส่งต่อไปยังเครื่องปลายทางต่อไป

7) เมื่อ lwB4 ต้นทางได้รับแพ็กเก็ตที่มีหมายเลข IPv4 ปลายทางอยู่ภายใน Bypass Scope lwB4 ต้องตรวจสอบข้อมูลภายใน Bypass Binding Table ว่ามีข้อมูลอุโมงค์สื่อสารโดยตรงไปยัง lwB4 ปลายทางหรือไม่? เช่นเดียวกันกับการทำงานในข้อ 1)

8) เนื่องจากข้อมูลของ lwB4 ปลายทางได้ถูกปรับปรุงเรียบร้อยแล้ว lwB4 จึงสามารถห่อหุ้มและส่งแพ็กเก็ตดังกล่าวไปยัง lwB4 ปลายทางโดยตรง

9) เมื่อ lwB4 ปลายทางได้รับแพ็กเก็ตที่ถูกห่อหุ้ม lwB4 ปลายทางต้องตรวจสอบความถูกต้องของเครื่องที่ทำหน้าที่เป็นอุโมงค์สื่อสารต้นทางและนำแพ็กเก็ต IPv4 ดั้งเดิมออกมาเพื่อส่งต่อไปยังเครื่องปลายทางต่อไป

ภายหลังจากดำเนินการขั้นตอน Initial lw4over6 Process และ Initial Bypass lw4over6 Process เสร็จสิ้น อุโมงค์สื่อสารโดยตรงถูกสร้างขึ้นระหว่าง lwB4-1 และ lwB4-2 โดยข้อมูลเกี่ยวข้องในการดำเนินการทั้งหมดของ lwB4, lwAFTR และ DHCP 4o6 server มีรายละเอียดดังแสดงในรูปที่ 3-18




รูปที่ 3-18 ข้อมูลที่เกี่ยวข้องในการสร้างการเชื่อมต่อ IPv4 ของ elw4over6

### 3.5.3 Terminative Bypass lw4over6 Process

Terminative Bypass lw4over6 Process ถูกนิยามเพื่อนำมาใช้งานใน elw4over6 โดยเฉพาะ ขั้นตอนนี้จะเริ่มดำเนินการเมื่อ lwB4 ได้รับ ICMP ที่แสดงข้อผิดพลาดในการส่งแพ็กเก็ตไปยังเครือข่ายปลายทางโดยตรง lwB4 ต้องดำเนินการตรวจสอบความถูกต้องเกี่ยวกับข้อมูลของอุโมงค์สื่อสารดังกล่าวกับข้อมูลของ DHCP 4o6 Server หากข้อมูลของอุโมงค์สื่อสารที่บันทึกไม่สอดคล้องกัน อุโมงค์สื่อสารดังกล่าวจะถูกปรับปรุงให้ถูกต้องหรือยกเลิกการใช้งาน ขั้นตอนการดำเนินการของ Terminative Bypass lw4over6 Process มีดังต่อไปนี้

1) lwB4 ปลายทางหรือเราเตอร์ส่ง ICMP ตอบกลับไปยัง lwB4 ต้นทางเพื่อระบุปัญหาที่เกิดขึ้นในการสร้างอุโมงค์สื่อสารโดยตรง

2) เมื่อ lwB4 ได้รับ ICMPv4 type 3 code 13 หรือ ICMPv6 type 1 code 0, type 4 code 1 lwB4 ต้องนำข้อมูลหมายเลข IPv4 และพอร์ตหรือหมายเลข IPv6 มาตรวจสอบกับข้อมูลภายใน Bypass Binding Table กรณีที่ไม่พบข้อมูลภายใน Bypass Binding Table กำหนดให้ lwB4 เพิกเฉยต่อ ICMP ดังกล่าว แต่ในกรณีที่พบข้อมูลภายใน Bypass Binding Table กำหนดให้ lwB4 ต้องร้องขอข้อมูลของอุโมงค์สื่อสารปลายทางที่เกิดปัญหาไปยัง DHCP 4o6 Server

ผ่าน DHCPv4 Leasequery over DHCPv6 โดยระบุหมายเลข IPv4 และ port-set จากข้อมูลใน Bypass Binding Table เพื่อปรับปรุงข้อมูลของอุโมงค์สื่อสารดังกล่าวให้มีความถูกต้อง

### 3.6 สรุปการออกแบบกระบวนการเปลี่ยนถ่าย Enhancement of Lightweight 4over6

Enhancement of Lightweight 4over6 เป็นกระบวนการที่พัฒนาต่อยอดจาก Lightweight 4over6 เพื่อเพิ่มประสิทธิภาพในการเชื่อมต่อภายในเครือข่าย อุปกรณ์ของ elw4over6 สามารถดำเนินการได้เช่นเดียวกับ lw4over6 ทุกประการ ยิ่งกว่านั้น อุปกรณ์ของ elw4over6 ยังสามารถสร้างอุโมงค์สื่อสารภายในเครือข่ายเพื่อเชื่อมต่อไปยังเครือข่ายปลายทางโดยตรง ซึ่งคุณสมบัติดังกล่าวเป็นคุณสมบัติเฉพาะของ elw4over6 เท่านั้น เพื่อรองรับการสร้างอุโมงค์สื่อสารโดยตรง elw4over6 อาศัยข้อมูลสำคัญ 2 ชนิด ได้แก่ Bypass Scope และ Bypass Binding Table ข้อมูล Bypass Scope เป็นช่วงของหมายเลข IPv4 ของ lwB4 ซึ่งถูกนำมาใช้เพื่อคัดกรองแพ็กเก็ต IPv4 ภายในเครือข่ายออกจากแพ็กเก็ตทั้งหมด ส่วนข้อมูล Bypass Binding Table เป็นตารางบันทึกอุโมงค์สื่อสารปลายทางซึ่งบันทึกโดย lwB4 เพื่อใช้เป็นข้อมูลในการสร้างอุโมงค์สื่อสารไปยังเครือข่ายปลายทาง ข้อมูลของอุโมงค์สื่อสารปลายทางภายใน Bypass Binding Table ถูกปรับปรุงให้สอดคล้องกับข้อมูลการจัดสรรหมายเลข IPv4 และ port-set ของ DHCP 4o6 server โดย lwB4 จะดำเนินการร้องขอข้อมูลของอุโมงค์สื่อสารปลายทางที่ต้องการผ่าน DHCPv4 leasequery over DHCPv6 ซึ่งได้อธิบายรายละเอียดในหัวข้อ 3.4.2 แต่ในบางครั้งการส่งข้อมูลโดยใช้อุโมงค์สื่อสารใน Bypass Binding Table อาจผิดพลาด lwB4 ซึ่งเป็นอุโมงค์สื่อสารต้นทางจะได้รับ ICMP ตอบกลับมา หลังจากนั้น lwB4 สามารถดำเนินการแก้ไขข้อผิดพลาดตามชนิดและข้อมูลใน ICMP จากคุณสมบัติเฉพาะของ elw4over6 ซึ่งรองรับการสร้างอุโมงค์สื่อสารภายในเครือข่ายเพื่อเชื่อมต่อไปยังเครือข่ายปลายทางโดยตรงช่วยให้เครื่องต้นทางสามารถใช้เส้นทางที่เหมาะสมที่สุดเพื่อส่งข้อมูลไปยังเครื่องปลายทางภายในเครือข่าย โดยไม่จำเป็นต้องส่งแพ็กเก็ต IPv4 ไปยังอุปกรณ์ของผู้ให้บริการก่อนเช่นเดียวกับ lw4over6

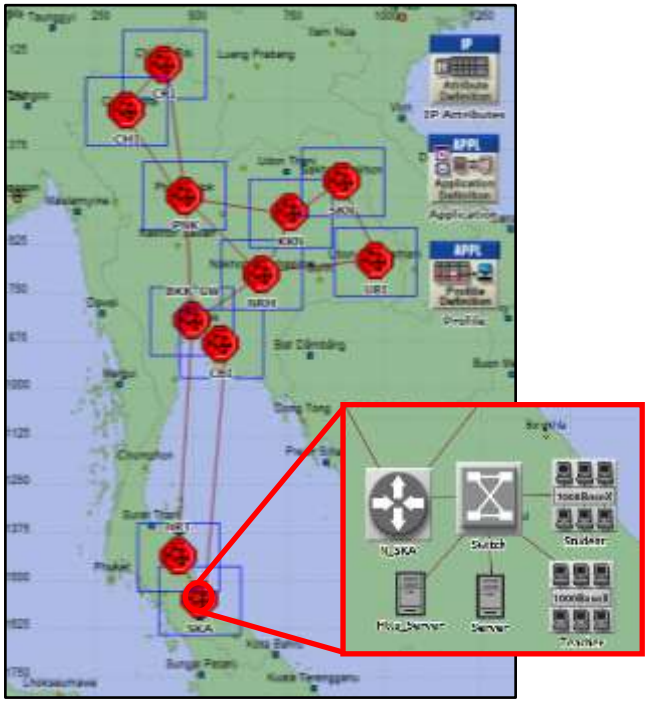
# บทที่ 4
# ผลการทดสอบและบทวิเคราะห์

ในบทที่ 3 ได้นำเสนอกระบวนการเปลี่ยนถ่าย Enhancement of Lightweight 4over6 ซึ่งถูกออกแบบโดยยึดตามคุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมในการให้บริการ IPv4 และ IPv6 ในบทนี้จะนำ Enhancement of Lightweight 4over6 มาเปรียบเทียบประสิทธิภาพในการให้บริการการเชื่อมต่อ IPv4 กับกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 อื่นๆ โดยใช้แบบจำลองระบบเครือข่าย โดยในหัวข้อแรกจะอธิบายเกี่ยวกับรายละเอียดของระบบเครือข่ายจำลอง ซึ่งรวมไปถึงการกำหนดการใช้งานโปรแกรมประยุกต์และตัวชี้วัดที่สนใจในการวัดประสิทธิภาพที่ใช้ในการจำลอง หัวข้อถัดไปเป็นการนำเสนอประสิทธิภาพของกระบวนการเปลี่ยนถ่ายตามตัวชี้วัดในแต่ละชนิด และหัวข้อสุดท้ายเป็นการสรุปผลการทดสอบประสิทธิภาพของกระบวนการเปลี่ยนถ่าย

## 4.1 ระบบเครือข่ายจำลอง

การวิเคราะห์ประสิทธิภาพเบื้องต้นเป็นการวิเคราะห์เชิงเปรียบเทียบโดยมองภาพรวมของระบบทั้งหมด ส่งผลให้ไม่สามารถประเมินประสิทธิภาพของระบบเครือข่ายได้ละเอียดมากนัก ดังนั้นเพื่อให้สามารถวิเคราะห์ประสิทธิภาพของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการอย่างละเอียดต้องอาศัยการวิธีการจำลองระบบเครือข่ายเข้ามาช่วยในการวิเคราะห์ ข้อดีของการใช้แบบจำลองเครือข่ายนั้นคือสามารถวัดประสิทธิภาพของระบบเครือข่ายได้หลากหลายระดับ ยกตัวอย่างเช่น ผลลัพธ์ที่เกิดขึ้นกับระบบเครือข่ายในภาพรวม, ระบบเครือข่ายย่อย, โหนด, ลิงค์ หรือแม้กระทั่งโมดูลภายในโหนดหรือลิงค์ เมื่อสามารถบันทึกผลลัพธ์ที่เกิดขึ้นในทุกระดับชั้น ส่งผลให้สามารถวิเคราะห์ประสิทธิภาพได้ละเอียดมากยิ่งขึ้น อีกทั้งยังสามารถนำข้อมูลในระดับแต่ละระดับที่มีความสัมพันธ์กันมาสนับสนุนผลการวิเคราะห์เพิ่มเติม เพื่อให้ผลการวิเคราะห์มีความน่าเชื่อถือมากยิ่งขึ้น

ระบบเครือข่ายที่ใช้ในการวิเคราะห์ประสิทธิภาพจำลองมาจากระบบเครือข่ายของ UniNet ซึ่งเป็นระบบเครือข่ายที่มีวัตถุประสงค์เพื่อให้บริการอินเทอร์เน็ตสำหรับการศึกษาและวิจัยสำหรับสถาบันทางการศึกษาภายในประเทศไทย สาเหตุที่เลือกระบบเครือข่ายของ UniNet เป็นต้นแบบในการจำลอง เนื่องจาก UniNet ประสบปัญหาในการขยายการให้บริการอินเทอร์เน็ตเพื่อให้รองรับสถาบันทางการศึกษาขนาดกลาง และขนาดเล็กเพิ่มเติม เพราะ UniNet กำลังขาดแคลนหมายเลข IPv4 เพื่อนำมาใช้สำหรับการจัดสรร การให้บริการ IPv4 ผ่านกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 จึงถือเป็นทางออกหนึ่งซึ่งสามารถนำมาใช้แก้ไขปัญหาตรงจุดนี้ ดังนั้นระบบเครือข่ายของ UniNet จึงมีความเหมาะสมเพื่อใช้เป็นต้นแบบสำหรับศึกษาผลกระทบที่เกิดขึ้นเมื่อให้บริการด้วยกระบวนการเปลี่ยนถ่ายดังกล่าว ระบบเครือข่ายของ UniNet ที่ถูกจำลองเบื้องต้นมีลักษณะดังแสดงในรูปที่ 4-1



รูปที่ 4-1 ระบบเครือข่ายที่จำลองมาจากระบบเครือข่ายของ UniNet

จากรูปที่ 4-1 ระบบเครือข่ายจำลองที่ใช้ในการทดสอบประกอบด้วย เราเตอร์จำนวน 11 ตัวซึ่งเชื่อมต่อเครือข่ายทั้งหมดเข้าด้วยกัน โดยแบ่งออกเป็นเราเตอร์ฝั่งผู้ให้บริการ 1 ตัว (Node BKK), เราเตอร์ภายในโครงข่าย 2 ตัว (Node PNK และ NRM) และเกตเวย์ของเครือข่ายย่อยของสถาบันทางการศึกษา 8 ตัว การสร้างระบบเครือข่ายของสถาบันทางการศึกษาเป็นระบบเครือข่ายย่อยช่วยลดความซับซ้อนของแบบจำลองลงได้ในระดับหนึ่ง ภายในระบบเครือข่ายย่อยของสถาบันทางการศึกษาก็จะประกอบด้วยเครื่องแม่ข่าย, เครือข่ายของอาจารย์ และเครือข่ายของนักศึกษา โหนดภายในระบบเครือข่ายของสถาบันทางการศึกษาทำหน้าที่จำลองการใช้งานโปรแกรมประยุกต์เสมือนกับเป็นผู้ใช้งานในระบบ การเชื่อมต่อระหว่างเกตเวย์ของสถาบันทางการศึกษากับเราเตอร์อื่นๆ เชื่อมต่อกันด้วย IPv6 เท่านั้น มีเพียงเราเตอร์ฝั่งผู้ให้บริการที่รองรับการเชื่อมต่อด้วย IPv4 สู่อินเทอร์เน็ตและเครือข่ายของผู้ใช้งานที่รองรับการเชื่อมต่อ IPv4 โดยอาศัยกระบวนการเปลี่ยนถ่าย ซึ่งกระบวนการที่นำมาเปรียบเทียบในการทดสอบประกอบด้วย 4rd, 4over6, lw4over6 และ elw4over6 แม้กระนั้น กระบวนการเปลี่ยนถ่ายที่ไม่ถูกนำมาทดสอบมีเพียงกระบวนการเดียวคือ DS-lite เนื่องจากสาเหตุหลัก 2 ประการ ประการแรก DS-lite ไม่สามารถจัดสรร Public IPv4 address ให้กับอุปกรณ์ฝั่งผู้ใช้งานได้ อุปกรณ์ฝั่งผู้ให้บริการต้องทำหน้าที่จัดสรร Public IPv4 address และพอร์ตให้กับเซสชันแต่ละเซสชันของเครื่องลูกข่าย ส่งผลให้อุปกรณ์ฝั่งผู้ให้บริการมีภาระงานในการประมวลผลสูง ประการที่สอง ผลลัพธ์ของ lw4over6 ซึ่งเป็นกระบวนการที่พัฒนาต่อยอดจาก DS-lite สามารถนำมาใช้เป็นตัวแทนของ DS-lite ได้ lw4over6 ได้รับการปรับปรุงให้มีประสิทธิมากขึ้น โดยกระจายภาระงานในการจัดสรร Public IPv4 address และพอร์ตให้กับเครื่องลูกข่ายในแต่ละเซสชันให้กับอุปกรณ์ฝั่งผู้ใช้งานซึ่งช่วยให้ลดการประมวลผลที่อุปกรณ์ฝั่งผู้ให้บริการลง

คุณสมบัติการให้บริการของกระบวนการเปลี่ยนถ่าย 4rd, 4over6, lw4over6 และ elw4over6 ถูกนำมาเปรียบเทียบในประเด็นต่างๆ โดยประเด็นที่ถูกนำมาเปรียบเทียบประกอบไปด้วย หลักการทำงาน, การจัดสรรหมายเลข IPv4, การระบุอุโมงค์สื่อสาร, การเชื่อมต่อ และการอัพเดตข้อมูลอุโมงค์สื่อสารซึ่งมีรายละเอียดดังแสดงในตารางที่ 4-1

ตารางที่ 4-1 แสดงการเปรียบเทียบหลักการทำงานของกระบวนการสร้างอุโมงค์สื่อสารด้วย IPv6

| ประเด็น | กระบวนการเปลี่ยนถ่าย | | | |
|---|---|---|---|---|
| | 4rd | 4over6 | lw4over6 | elw4over6 |
| หลักการทำงาน | stateless | stateful | stateful | stateful |
| การจัดสรรหมายเลข IPv4 | Public IPv4 address และ port-set แบบถาวร | Public IPv4 address แบบถาวร | Public IPv4 address และ port-set แบบพลวัต | Public IPv4 address และ port-set แบบพลวัต |
| การระบุอุโมงค์สื่อสารปลายทาง | กำหนดตาม 4rd rule | กำหนดโดยใช้ข้อมูลใน Encapsulation Table | กำหนดไปยังฝั่งผู้ให้บริการเสมอ | กำหนดโดยใช้ข้อมูลใน Bypass Binding Table |
| การเชื่อมต่อ | mesh | mesh | hub & spoke | mesh |
| การอัพเดตข้อมูลอุโมงค์สื่อสาร | - | อัพเดตข้อมูลด้วยโพรโตคอลกำหนดเส้นทาง IPv4/IPv6 | - | อัพเดตข้อมูลด้วย DHCPv4 over DHCPv6 |
| ความสามารถในการรองรับการขยายขนาด | ปานกลาง (ต้องเปลี่ยน 4rd rule เพื่อจัดสรรหมายเลขไอพีใหม่) | ปานกลาง (อาจขาดแคลนหมายเลข IPv4 ในการใช้งาน) | ต่ำ (ไม่รองรับการเชื่อมต่อ IPv4 แบบ mesh) | สูง (รองรับการเชื่อมต่อ IPv4 แบบ mesh) |
| ความยืดหยุ่นในการใช้งาน | ต่ำ (ต้องใช้หมายเลข IPv6, IPv4 และพอร์ตตาม 4rd rule) | ปานกลาง (สามารถใช้หมายเลข IPv6 และ IPv4 อย่างอิสระ) | สูง (สามารถหมายเลข IPv6, IPv4 และพอร์ตอย่างอิสระ) | สูง (สามารถหมายเลข IPv6, IPv4 และพอร์ตอย่างอิสระ) |

จากตารางที่ 4-1 4rd เป็นกระบวนที่มีการดำเนินการแบบ stateless และไม่ใช้โพรโตคอลอื่นๆ ในการปรับปรุงข้อมูลอุโมงค์สื่อสาร แต่ 4rd ก็ยังคงรองรับการเชื่อมต่อ IPv4 แบบ mesh เนื่องจากอุปกรณ์ทั้งหมดใน 4rd ต้องใช้งานตามหมายเลข IPv6, IPv4 และพอร์ตตามรูปแบบที่กำหนดล่วงหน้าใน 4rd rule ดังนั้นอุปกรณ์ภายใน 4rd สามารถคำนวณอุโมงค์สื่อสารปลายทางตามรูปแบบที่ถูกกำหนดใน 4rd rule โดยไม่จำเป็นต้องใช้การปรับปรุงข้อมูลอุโมงค์สื่อสารแต่อย่างใด แต่ข้อเสียของการใช้ 4rd rule คือทำให้มีความยืดหยุ่นในการใช้งานต่ำ และในกรณีที่ผู้ให้บริการมีหมายเลข IPv4 ไม่เพียงพอสำหรับขยายขนาดเครือข่าย 4rd จำเป็นต้องกำหนด 4rd rule ใหม่เพื่อรองรับผู้ใช้งานที่เพิ่มขึ้น สำหรับ 4over6 ปรับปรุงข้อมูลอุโมงค์สื่อสารด้วยโพรโตคอลกำหนดเส้นทาง IPv4/IPv6 ส่งผลให้ 4over6 รองรับการเชื่อมต่อ IPv4 แบบ mesh แต่ 4over6 มีความยืดหยุ่นในการใช้งานและความสามารถในการรองรับการขยายขนาดระดับปานกลาง เพราะแม้ว่า 4over6 จะสามารถใช้งานหมายเลข IPv6 และ IPv4 อย่างอิสระ แต่ก็ไม่สามารถจัดสรรการใช้งานหมายเลข IPv4 ในระดับพอร์ตได้ ในกรณีที่ผู้ให้บริการต้องการขยายขนาดเครือข่ายจึงอาจมี

หมายเลข IPv4 ไม่เพียงพอต่อการให้บริการ สำหรับ lw4over6 กำหนดให้อุปกรณ์ฝั่งผู้ใช้งานมีอุโมงค์สื่อสารเพียงอุโมงค์สื่อสารเดียวเพื่อเชื่อมต่อไปยังอุปกรณ์ฝั่งผู้ให้บริการ lw4over6 จึงไม่รองรับการเชื่อมต่อ IPv4 แบบ mesh ส่งผลให้มีความสามารถในการรองรับการขยายขนาดในระดับปานกลาง อย่างไรก็ตาม lw4over6 สามารถใช้งานหมายเลข IPv6, IPv4 และพอร์ตอย่างอิสระเพราะ lw4ove6 มีอุปกรณ์ที่ทำหน้าที่ในการจัดสรรหมายเลย IPv4 และ port-set ให้กับอุปกรณ์ฝั่งผู้ใช้งานโดยเฉพาะ lw4over6 จึงมีความยืดหยุ่นในการใช้งานสูง และสำหรับ elw4over6 มีการประยุกต์ใช้ DHCPv4 over DHCPv6 เพิ่มเติมจาก lw4over6 เพื่อปรับปรุงอุโมงค์สื่อสารปลายทาง elw4over6 จึงรองรับการเชื่อมต่อ IPv4 แบบ mesh ส่งผลให้มีความสามารถในการรองรับการขยายขนาดในระดับสูง นอกจากนี้ elw4over6 ยังคงใช้อุปกรณ์ที่ทำหน้าที่ในการจัดสรรหมายเลย IPv4 และพอร์ตให้กับอุปกรณ์ฝั่งผู้ใช้งานโดยเฉพาะเช่นเดียวกับ lw4over6 ส่งผลให้ elw4over6 สามารถใช้งานหมายเลข IPv6, IPv4 และพอร์ตอย่างอิสระจึงมีความยืดหยุ่นสูงในการใช้งานสูง

### 4.1.1 โปรแกรมประยุกต์ที่ใช้ในการจำลอง

ในการจำลองประสิทธิภาพการให้บริการการเชื่อมต่อ IPv4 โดยใช้กระบวนการสร้างอุโมงค์สื่อสารด้วย IPv6 ได้สร้างแบบจำลองด้วยโปรแกรม OPNET Modeler 16.0 ซึ่งการจำลองด้วยโปรแกรม OPNET Modeler มีจุดเด่นอย่างหนึ่งคือสามารถกำหนดรายละเอียดในการจำลองได้ใกล้เคียงกับความเป็นจริงอย่างมาก เนื่องจากโปรแกรม OPNET Modeler แบ่งการกำหนดการใช้งานโปรแกรมประยุกต์ในการจำลองออกเป็น 3 ส่วน ได้แก่ คุณสมบัติของโปรแกรมประยุกต์ (Application Attribute), ลักษณะการใช้งานของผู้ใช้(User Profile) และการรองรับโปรแกรมประยุกต์ของเครื่องลูกข่ายและเครื่องแม่ข่าย(Application Source & Destination)

สำหรับการกำหนดคุณลักษณะของโปรแกรมประยุกต์เป็นการกำหนดว่าโปรแกรมประยุกต์นั้นมีลักษณะการใช้งานอย่างไร ยกตัวอย่างเช่น โปรแกรมประยุกต์ HTTP สามารถกำหนดข้อมูล HTTP Version, จำนวนการเชื่อมต่อสูงสุด, ขนาด และชนิดของเว็บเพจ ซึ่งโปรแกรมประยุกต์แต่ละชนิดมีคุณลักษณะที่แตกต่างกันไปตามลักษณะพิเศษของโปรแกรมนั้นๆ นอกจากนี้โปรแกรมประยุกต์เดียวกันสามารถกำหนดคุณลักษณะได้มากกว่าหนึ่งรูปแบบ เช่น โปแกรมประยุกต์ HTTP ในรูปแบบแรกอาจเลือกใช้ HTTP Version 1.0 แต่ในรูปแบบที่สองอาจเลือกใช้ HTTP Version 1.1 เป็นต้น สำหรับการกำหนดคุณลักษณะการใช้งานของผู้ใช้เป็นการระบุว่าผู้ใช้งานมีการใช้งานโปรแกรมประยุกต์ใดบ้าง มีการใช้งานบ่อยครั้งแค่ไหน และการใช้งานแต่ละครั้งใช้เวลานานเท่าไร ส่งผลให้สามารถกำหนดลักษณะการใช้งานของผู้ใช้ได้อย่างหลากหลายและใกล้เคียงกับการใช้งานจริงของผู้ใช้ และสุดท้ายสำหรับการกำหนดการรองรับโปรแกรมประยุกต์ของเครื่องลูกข่ายและเครื่องแม่ข่ายเป็นการกำหนดว่าเครื่องลูกข่ายแต่ละเครื่องรองรับลักษะการใช้งานของผู้ใช้แบบใดบ้าง และกำหนดว่าเครื่องแม่ข่ายแต่ละเครื่องรองรับโปรแกรมประยุกต์ใด เครื่องลูกข่ายแต่ละเครื่องนั้นต้องกำหนดเครื่องแม่ข่ายที่ต้องการติดต่อในแต่ละโปรแกรมประยุกต์ โดยที่เครื่องลูกข่ายสามารถกำหนดเครื่องแม่ข่ายของแต่ละโปรแกรมประยุกต์ได้มากกว่า 1 เครื่อง และสามารถกำหนดน้ำหนักในการติดต่อไปยังเครื่องแม่ข่ายแต่ละเครื่องได้

การเลือกโปรแกรมประยุกต์ที่ใช้ในแบบจำลองอาศัยข้อมูลจากบทความของ Cisco มาใช้ควบคู่ในการตัดสินใจ [32] โดยในปี พ.ศ. 2558 Cisco คาดว่าการใช้งานอินเทอร์เน็ตจะมีสัดส่วนการใช้งาน Internet Video 61%, File Sharing (P2P) 24% และ Web/Data 15% การใช้งาน Internet Video สามารถแบ่งย่อยออกเป็น 2 รูปแบบ ได้แก่ Lossy และ Lossless รูปแบบ Lossy ยินยอมให้คุณภาพวีดีโอลดทอนลงไปบ้าง ซึ่งนิยมใช้ในการส่งข้อมูลวีดีโอแบบ real-time โดยวีดีโอแบบ real-time ต้องการความเร็วในการส่งข้อมูลสูงจึงอาศัยการส่งข้อมูลด้วย UDP เป็นหลัก และรูปแบบ Lossless ซึ่งต้องการรักษาคุณภาพวีดีโอให้เทียบเท่าต้นฉบับ ซึ่งนิยมใช้ในการส่งข้อมูลแบบ on-demand จึงต้องการโพรโตคอลที่สามารถควบคุมความถูกต้องในการส่งข้อมูลเช่น TCP สำหรับโพรโตคอลที่ใช้ในการส่งข้อมูลของ File Sharing (P2P) และ Web/Data มีการส่งข้อมูลด้วย TCP เนื่องจากให้ความสำคัญกับความถูกต้องของข้อมูลเป็นอันดับแรก แม้ว่าสัดส่วนการส่งข้อมูลด้วย UDP จะมีปริมาณไม่น้อยกว่า TCP แต่การใช้ UDP ในการส่งข้อมูลมีการดำเนินการที่เรียบง่าย หากนำ UDP มาใช้ในการวัดประสิทธิภาพของกระบวนการเปลี่ยนถ่ายก็ไม่สามารถนำผลลัพธ์มาสรุปเทียบเคียงกับ TCP ซึ่งมีความซับซ้อนในการส่งข้อมูลมากกว่าได้ ดังนั้นในการวัดประสิทธิภาพจึงเลือกใช้ TCP ซึ่งเป็นโพรโตคอลที่มีความซับซ้อนสูงกว่าและมีโปรแกรมประยุกต์ที่นำ TCP มาใช้งานหลากหลายกว่า UDP เพื่อให้สามารถสรุปผลได้ครอบคลุมไปถึงโพรโตคอลและโปรแกรมประยุกต์อื่นๆ ที่มีหลักการทำงานซับซ้อนใกล้เคียงกัน เมื่อพิจารณาโปรแกรมประยุกต์ที่ใช้การส่งข้อมูลด้วย TCP และมีแนวโน้มการใช้งานสูงจากในบทความ พบว่ามี 2 โปรแกรมประยุกต์ด้วยกัน ได้แก่ BitTorrent ใน File Sharing (P2P) และ HTTP ใน Web/Data โดยผู้วิจัยได้เลือกใช้โปรแกรมประยุกต์ HTTP ในการจำลอง เนื่องจากเหตุผล 2 ประการด้วยกัน ประการที่หนึ่ง HTTP เป็นโปรแกรมประยุกต์ที่มีสัดส่วนการใช้งานสูง และเป็นโปรแกรมประยุกต์มาตรฐานในโปรแกรม OPNET Modeler จึงไม่จำเป็นต้องพัฒนาขึ้นใหม่ นอกจากนี้การใช้โปรแกรมประยุกต์มาตรฐานใน OPNET Modeler ยังช่วยให้ผลลัพธ์ที่ได้มีความน่าเชื่อถือสูงกว่าโปรแกรมประยุกต์ที่ต้องพัฒนาขึ้นใหม่ ประการที่สอง HTTP แตกต่างจาก BitTorrent เพียงแค่ BitTorrent สามารถดาวน์โหลดข้อมูลชุดเดียวกันได้จากหลากหลายเครื่องต้นทางในเวลาเดียวกัน ซึ่งรูปแบบการเชื่อมต่อซึ่งมีลักษณะใกล้เคียงกับ BitTorrent ก็สามารถจำลองด้วย HTTP ได้ โดยเพิ่มจำนวนเครื่องแม่ข่ายให้มากขึ้น และแบ่งข้อมูลออกเป็นส่วนย่อยๆ เพื่อกระจายไปยังเครื่องแม่ข่าย จากนั้นก็กำหนดให้เครื่องลูกข่ายดาวน์โหลดข้อมูลทั้งหมดในเวลาใกล้เคียงกัน

ดังนั้น เครื่องลูกข่ายแต่ละเครื่องในการจำลองการให้บริการ IPv4 โดยกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 บนระบบเครือข่ายของ UniNet มีการกำหนดลักษณะการใช้งานของผู้ใช้ที่เป็นนักศึกษา (User Profile: student) โดยนักศึกษามีการใช้งานโปรแกรมประยุกต์ HTTP ใน 2 รูปแบบย่อย ได้แก่ HTTP: Searching และ HTTP: Browsing โดย HTTP: Searching เป็นการจำลองการร้องขอข้อมูลจำนวนน้อยครั้งแต่ข้อมูลที่ถูกร้องขอค่อนข้างมีขนาดใหญ่ HTTP: Searching จึงเว้นระยะเวลาในการร้องขอ Web page แต่ละครั้งห่างกันโดยเฉลี่ยเท่ากับ 10 วินาที page ของ HTTP: Searching มีขนาดตั้งแต่ 2,000 ถึง 5,000 ไบต์และมีจำนวน object เท่ากับ 3 ส่วน HTTP: Browsing เป็นการจำลองการร้องขอข้อมูลจำนวนมากครั้งแต่ข้อมูลที่ถูกร้องขอมีขนาดเล็ก HTTP: Browsing จึงเว้นระยะเวลาในการร้องขอ Web page แต่

ละครั้งห่างกันโดยเฉลี่ยเท่ากับ 4 วินาที page ของ HTTP: Searching มีขนาดตั้งแต่ 650 ถึง 2,650 ไบต์และมีจำนวน object เท่ากับ 6 การใช้งานโปรแกรมประยุกต์ HTTP ของเครื่องลูกข่ายในทุกกระบวนการเปลี่ยนถ่ายถูกควบคุมให้มีอินพุตเหมือนกันทุกประการ สำหรับการใช้งาน TCP ของอุปกรณ์ทั้งหมดภายในแบบจำลองกำหนดให้ใช้ TCP-Reno ที่มีคุณลักษณะแบบ Fast Retransmit และ Fast Recovery นอกจากนี้อุปกรณ์เครือข่ายฝั่งผู้ใช้งานเชื่อมต่อกับเครือข่ายแกนหลักของผู้ให้บริการด้วยลิงค์ขนาด 10 Mbps และมีความเร็วของ CPU ในการประมวลผลเท่ากับ 1 GHz โดยปัจจัยควบคุมในการจำลองต่างๆ และโปรแกรมประยุกต์ HTTP ที่ใช้เป็นอินพุตแสดงรายละเอียดดังตารางที่ 4-2

ตารางที่ 4-2 ปัจจัยที่กำหนดในการจำลอง

| ปัจจัยที่กำหนดในการจำลอง | | ค่า |
|---|---|---|
| Backbone Links | | 10 Mbps |
| Router: CPU speed | | 1 GHz |
| TCP Parameters: | Slow-Start initial count (MSS) | 2 |
| | Fast Retransmit | Enable |
| | Fast Recovery | Reno |
| HTTP Parameters: | HTTP version | 1.1 |
| | HTTP timeout | 10 seconds |
| HTTP (Searching): | Page size | 2,000 – 5,000 Bytes |
| | Objects | 3 |
| | Inter-repetition Time | 10 seconds |
| HTTP (Browsing): | Page size | 650 – 2,650 Bytes |
| | Objects | 6 |
| | Inter-repetition Time | 4 seconds |

### 4.1.2 ตัวชี้วัดที่สนใจ

สำหรับตัวชี้วัดที่น่าสนใจแบ่งออกเป็น 2 กลุ่มด้วยกัน ประกอบด้วย ตัวชี้วัดด้านการประมวลผล และตัวชี้วัดด้านการรับส่งข้อมูล โดยมีสมมุติฐานว่า “กระบวนการเปลี่ยนถ่ายที่มีตัวชี้วัดด้านการประมวลผลที่สูง จะส่งผลให้มีตัวชี้วัดด้านการรับส่งข้อมูลที่ต่ำ” สำหรับตัวชี้วัดด้านการประมวลผลวัดจากอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานและอุปกรณ์ฝั่งผู้ให้บริการ ซึ่งคำนวณเป็นร้อยละ แต่สำหรับตัวชี้วัดด้านการรับส่งข้อมูล เช่น อัตราการรับข้อมูล หรืออัตราการสูญหายของข้อมูลจากการส่ง ไม่เหมาะในการนำมาคำนวณเป็นร้อยละโดยเทียบกับฐานอัตราการส่งข้อมูล เนื่องจากเครื่องแม่ข่ายของ HTTP ซึ่งทำหน้าที่เป็นผู้ส่งไม่ได้ส่งข้อมูลออกมา(Push)ด้วยความเร็วคงที่ตลอดการเชื่อมต่อ แต่เครื่องผู้ส่งในการรับส่งข้อมูลด้วย HTTP จะส่งข้อมูลตามการร้องข้อ(Pull)ของเครื่องผู้รับซึ่งส่งผลให้มีความเร็วในการส่งข้อมูลแปรผันตามรูปแบบในการร้องขอ โดยในบางช่วงอาจมีอัตราการส่งข้อมูลสูงมาก แต่ในบางช่วงก็อาจมีอัตราการส่งข้อมูลเท่ากับศูนย์ เพราะไม่มีการร้องขอจากเครื่องลูกข่าย ทำให้อัตราการส่งข้อมูลด้วย HTTP จึงมีค่าไม่คงที่ตลอดการเชื่อมต่อ ดังนั้นตัวชี้วัดด้านการรับส่งข้อมูลจึงใช้หน่วยในการวัดผลเป็นไบต์ต่อวินาที (bytes/sec)

จึงมีความเหมาะสมมากกว่าการปรับฐานของข้อมูลเทียบเป็นร้อยละ สำหรับตัวชี้วัดอีกตัวหนึ่งที่ไม่ได้นำมาใช้คือ Throughput ซึ่งสามารถปรับฐานของข้อมูลเทียบเป็นร้อยละได้เช่นกัน เนื่องจากการใช้งานโปรแกรมประยุกต์ HTTP ทำงานบน TCP มีการส่งข้อมูลซ้ำเมื่อเกิดการสูญหายของข้อมูล ดังนั้น Throughput จึงไม่สามารถนำมาสรุปได้ว่า กระบวนการเปลี่ยนถ่ายที่มี Throughput สูงกว่านั้นมีสาเหตุมาจากมีความสามารถส่งข้อมูลได้มากกว่าหรือมีการส่งข้อมูลซ้ำที่สูงกว่า ด้วยเหตุนี้ตัวชี้วัดที่ถูกนำมาใช้แทน Throughput คืออัตราการรับข้อมูลด้วย HTTP ซึ่งเป็นตัวชี้วัดใน application-layer ที่วัดเฉพาะข้อมูลที่ได้รับอย่างถูกต้องด้วยเครื่องปลายทางเท่านั้นโดยไม่รวมผลการส่งข้อมูลซ้ำ

สำหรับตัวชี้วัดที่ใช้เป็นเอาต์พุตในการเปรียบเทียบประสิทธิภาพการให้บริการการเชื่อมต่อ IPv4 ด้วยกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 ประกอบไปด้วย ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU, ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP, ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP และค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ซึ่งตัวชี้วัดแต่ละตัวที่ใช้วัดประสิทธิภาพมีรายละเอียดดังต่อไปนี้

#### 4.1.2.1 ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU (Average CPU utilization)

อัตราส่วนการใช้งาน CPU บ่งบอกสัดส่วนในการใช้งาน CPU ที่ใช้ในการประมวลผลในการส่งต่อแพ็กเก็ต และการประมวลผลของโปรแกรมประยุกต์ ถ้าหากผลการทดลองของกระบวนการเปลี่ยนถ่ายใดให้ค่าตัวชี้วัดนี้ต่ำแสดงว่ากระบวนการนั้นใช้เวลาในการประมวลผลน้อย ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU มีหน่วยเป็นร้อยละ (%) ซึ่งคำนวณได้จากสมการที่ 4-1 โดยตัวแปร CPU utilization time หมายถึงระยะเวลาที่ CPU ใช้ในการประมวลผลทั้งหมดในการจำลอง และตัวแปร t หมายถึงระยะเวลาในการจำลองทั้งหมด

$$CPU\ utilization_{Avg} = \frac{CPU\ utilization\ time}{t} \times 100 \qquad (4\text{-}1)$$

#### 4.1.2.2 ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP (Average HTTP traffic received)

อัตราการรับข้อมูลด้วย HTTP บ่งบอกปริมาณข้อมูลที่ได้รับผ่าน HTTP ซึ่งวัดปริมาณข้อมูลจากทั้ง HTTP server และ HTTP client ทั้งหมดภายในระบบที่ถูกส่งจากชั้น transport-layer ไปยังชั้น application-layer ในหนึ่งวินาที ถ้าหากผลการทดลองของกระบวนการเปลี่ยนถ่ายใดให้ค่าตัวชี้วัดนี้สูงแสดงว่ากระบวนการนั้นสามารถส่งข้อมูลได้อย่างมีประสิทธิภาพ ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP มีหน่วยเป็นไบต์ต่อวินาที ซึ่งคำนวณได้จากสมการที่ 4-2 โดยตัวแปร i คือเวลาในการจำลองวินาทีที่ 1, 2, 3, ..., t ตัวแปร t หมายถึงระยะเวลาในการจำลองทั้งหมด และตัวแปร $HTTP\ traffic\ received_i$ หมายถึงอัตราการรับข้อมูลโดยเฉลี่ยในวินาทีที่ i

$$HTTP\ traffic\ received_{Avg} = \frac{\sum_{i=1}^{n} HTTP\ traffic\ received_i}{t} \qquad (4\text{-}2)$$

#### 4.1.2.3 ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP (Average HTTP traffic lost)

อัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP บ่งบอกปริมาณข้อมูลที่เกิดการสูญหายจากการส่งด้วย HTTP ภายในระบบซึ่งวัดปริมาณข้อมูลที่ส่งจากชั้น application-layer ไปยังชั้น transport-layer ของเครื่องต้นทางหักลบด้วยปริมาณข้อมูลที่รับในชั้น application-layer ของเครื่องปลายทางในหนึ่งวินาที ถ้าหากผลการทดลองของกระบวนการเปลี่ยนถ่ายใดให้ค่าตัวชี้วัดนี้ต่ำแสดงว่ากระบวนการนั้นมีความน่าเชื่อถือในการส่งข้อมูลสูง ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP มีหน่วยเป็นไบต์ต่อวินาทีซึ่งคำนวณได้จากสมการที่ 4-3 โดยตัวแปร i คือ เวลาในการจำลองวินาทีที่ 1, 2, 3, ..., t ตัวแปร t หมายถึงระยะเวลาในการจำลองทั้งหมด ตัวแปร HTTP traffic $sent_i$ หมายถึงอัตราการส่งข้อมูลโดยเฉลี่ยในวินาทีที่ i และตัวแปร HTTP traffic $received_i$ หมายถึงอัตราการรับข้อมูลโดยเฉลี่ยในวินาทีที่ i

$$HTTP\ traffic\ lost_{Avg} = \frac{\sum_{i=1}^{t}(HTTP\ traffic\ sent_i - HTTP\ traffic\ received_i)}{t} \quad (4\text{-}3)$$

#### 4.1.2.4 ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object (Average HTTP object download time)

ระยะเวลาในการดาวน์โหลด HTTP object บ่งบอกความเร็วในการดาวน์โหลดข้อมูลด้วย HTTP ซึ่งวัดในชั้น application-layer ถ้าหากผลการทดลองของกระบวนการเปลี่ยนถ่ายใดให้ค่าตัวชี้วัดนี้ต่ำ แสดงว่ากระบวนการนั้นมีความรวดเร็วในการรับส่งข้อมูลสูง ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object มีหน่วยเป็นวินาทีต่อออบเจ็กต์ซึ่งคำนวณได้จากสมการที่ 4-4 โดยตัวแปร i คือ HTTP object ที่ 1, 2, 3, ..., o ตัวแปร o หมายถึงจำนวน HTTP object ทั้งหมด และตัวแปร HTTP object download $time_i$ หมายถึงเวลาในการดาวน์โหลด HTTP object ที่ i

$$HTTP\ object\ download\ time_{Avg} = \frac{\sum_{i=1}^{o} HTTP\ object\ download\ time_i}{o} \quad (4\text{-}4)$$

### 4.2 ประสิทธิภาพของกระบวนการเปลี่ยนถ่าย

ในการเปรียบเทียบประสิทธิภาพได้แบ่งรูปแบบการจำลองออกเป็น 2 รูปแบบตามรูปแบบการเชื่อมต่อ รูปแบบแรกคือรูปแบบที่มีเฉพาะการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายนอกเครือข่าย (Inter-communication) และรูปแบบที่สองคือรูปแบบที่มีเฉพาะการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายในเครือข่าย (Intra-communication) นอกจากนั้นการจำลองทั้ง 2 รูปแบบถูกแบ่งการจำลองออกเป็น 4 รูปแบบย่อยตามจำนวนของเครื่องลูกข่ายภายในเครือข่ายผู้ใช้งานซึ่งประกอบไปด้วย เครื่องลูกข่ายจำนวน 512 1,024 1,536 และ 2,048 เครื่อง ในแต่ละรูปแบบย่อยถูกทดสอบเพื่อวัดประสิทธิภาพด้วยตัวชี้วัดซึ่งประกอบไปด้วย ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU, ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP, ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP และค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object

ตัวชี้วัดค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU แสดงถึงปริมาณการประมวลผลซึ่งประกอบด้วย การประมวลผลในการส่งต่อแพ็กเก็ต และการประมวลผลของโปรแกรมประยุกต์ ซึ่งในการประมวลผลในการสร้างอุโมงค์สื่อสารของกระบวนการเปลี่ยนถ่ายนั้นถูกรวมอยู่ในส่วนของโปรแกรมประยุกต์ การประมวลผลในการสร้างอุโมงค์สื่อสารแบ่งออกเป็น 2 ขั้นตอนย่อย ได้แก่ การระบุอุโมงค์สื่อสารปลายทาง และการห่อหุ้มแพ็กเก็ต ซึ่งแต่ละกระบวนการเปลี่ยนถ่ายมีเพียงแค่การดำเนินการระบุอุโมงค์สื่อสารปลายทางเท่านั้นที่แตกต่างกัน ดังนั้นความแตกต่างของการระบุอุโมงค์สื่อสารปลายทางนั้นเป็นตัวแปรสำคัญที่ส่งผลต่อค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU และนำไปสู่ความแตกต่างของประสิทธิภาพการรับส่งข้อมูลด้วย HTTP ในที่สุด

### 4.2.1 รูปแบบที่มีการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายในเครือข่าย

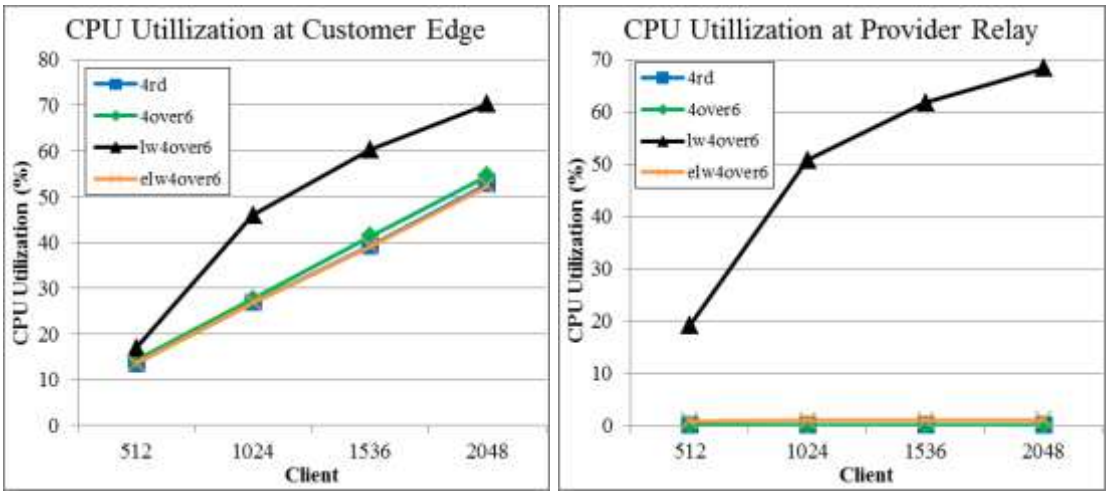
ในการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายในเครือข่ายของทุกกระบวนการเปลี่ยนถ่าย แพ็กเก็ตทั้งหมดถูกส่งระหว่างอุปกรณ์ฝั่งผู้ใช้งานของเครือข่ายผู้ใช้งาน 2 เครือข่าย เมื่ออุปกรณ์ฝั่งผู้ใช้งานต้นทางได้รับแพ็กเก็ต บางกระบวนการอาจส่งแพ็กเก็ตไปยังอุปกรณ์ฝั่งผู้ให้บริการก่อน แต่บางกระบวนการก็อาจส่งแพ็กเก็ตไปยังเครือข่ายปลายทางโดยตรง ซึ่งขึ้นอยู่กับหลักการทำงานของแต่ละกระบวนการเปลี่ยนถ่าย โดยผลการเปรียบเทียบประสิทธิภาพการให้บริการของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการโดยใช้ตัวชี้วัดข้างต้นมีรายละเอียดดังต่อไปนี้

#### 4.2.1.1 อัตราส่วนการใช้งาน CPU เมื่อเครื่องปลายทางที่อยู่ภายในเครือข่าย

ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานและอุปกรณ์ฝั่งผู้ให้บริการบ่งบอกถึงประสิทธิภาพการประมวลผลของกระบวนการแต่ละกระบวนการ โดยค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์แต่ละชนิดแสดงถึงภาระงานต่างๆ ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ในแต่ละกระบวนการเปลี่ยนถ่ายซึ่งมีการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายในเครือข่ายแสดงรายละเอียดดังตารางที่ 4-3 นอกจากนี้ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ถูกนำมาแสดงข้อมูลในรูปแบบกราฟดังรูปที่ 4-2 เพื่อให้ง่ายต่อการศึกษาข้อมูล

ตารางที่ 4-3 ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน
และอุปกรณ์ฝั่งผู้ให้บริการ เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย

| **ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย (ร้อยละ)** | | | | | |
|---|---|---|---|---|---|
| **ชนิดอุปกรณ์** | **จำนวนเครื่องลูกข่าย** | **กระบวนการเปลี่ยนถ่าย** | | | |
| | | **4rd** | **4over6** | **lw4over6** | **elw4over6** |
| อุปกรณ์ฝั่งผู้ใช้งาน | 512 | 13.55 | 14.34 | 16.93 | 13.52 |
| | 1,024 | 26.90 | 27.69 | 45.96 | 26.84 |
| | 1,536 | 39.25 | 41.39 | 60.28 | 39.20 |
| | 2,014 | 52.77 | 54.62 | 70.27 | 52.37 |
| อุปกรณ์ฝั่งผู้ให้บริการ | 512 | 0.20 | 0.20 | 19.22 | 0.77 |
| | 1,024 | 0.20 | 0.20 | 50.80 | 0.98 |
| | 1,536 | 0.20 | 0.20 | 61.77 | 1.21 |
| | 2,014 | 0.20 | 0.20 | 68.34 | 1.43 |




(a) (b)

รูปที่ 4-2 ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน (a) และอุปกรณ์ฝั่งผู้ให้บริการ (b) เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย

จากรูปที่ 4-2 ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการแบ่งออกเป็น 2 ส่วนตามชนิดของอุปกรณ์ ได้แก่ ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานซึ่งแสดงดังรูปที่ 4-2 (a) และค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ฝั่งผู้ให้บริการซึ่งแสดงดังรูปที่ 4-2 (b) กระบวนการที่มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานต่ำที่สุดคือ elw4over6 ผลลัพธ์ที่ดีรองลงมาคือ 4rd, 4over6 และ lw4over6 ตามลำดับ elw4over6 และ 4over6 ดำเนินการแบบ stateful เช่นเดียวกัน แต่ elw4over6 ใช้เวลาในการประมวลผลน้อยกว่าเนื่องจาก elw4over6 บำรุงรักษาข้อมูลอุโมงค์สื่อสารเฉพาะที่มีการเชื่อมต่อเท่านั้น ไม่ได้บำรุงรักษาข้อมูลอุโมงค์สื่อสารทั้งหมดเหมือนกับ 4over6 ในทางกลับกัน 4rd ซึ่งมีการดำเนินการแบบ stateless กลับมีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU สูงกว่า elw4over6 เล็กน้อย เนื่องจากเวลาในการประมวลผลของ elw4over6 เพิ่มขึ้นตามจำนวนของอุโมงค์สื่อสาร ในกรณีที่ elw4over6 มีอุโมงค์สื่อสารมีจำนวนไม่มาก elw4over6 จะสูญเสียเวลาในการประมวลผลเพียงเล็กน้อยเท่านั้น อย่างไรก็ตามจุดที่น่าสนใจที่สุดคือ lw4over6 ซึ่งเป็นกระบวนการสามารถระบุอุโมงค์สื่อสารปลายทางได้รวดเร็วที่สุด เพราะกำหนดอุโมงค์สื่อสารปลายทางตายตัวไปยังอุปกรณ์ฝั่งผู้ให้บริการกลับมีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU สูงสุด เนื่องจากค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ไม่ได้เพิ่มขึ้นจากการประมวลผลของกระบวนการเปลี่ยนถ่ายเพียงอย่างเดียว แต่เพิ่มขึ้นจากการประมวลแพ็กเก็ตที่ส่งต่อผ่านเราเตอร์อีกด้วย ด้วยเหตุนี้การดำเนินการของ lw4over6 ที่อาศัยการส่งแพ็กเก็ตไปยังอุปกรณ์ฝั่งผู้ให้บริการก่อนเสมอ แพ็กเก็ตของ lw4over6 จึงถูกส่งต่อผ่านเราเตอร์จำนวนมากส่งผลให้อุปกรณ์ฝั่งผู้ใช้งานมีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU สูง เมื่อพิจารณาความสัมพันธ์ระหว่างจำนวนเครื่องลูกข่ายกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน พบว่าทุกกระบวนการเปลี่ยนถ่ายต่างมีค่าเฉลี่ยของอัตราการใช้งาน CPU สูงขึ้นเมื่อมีจำนวนเครื่องลูกข่ายเพิ่มมากขึ้น ซึ่งมีสาเหตุจากปริมาณการเชื่อมต่อที่มีจำนวนมากขึ้น เมื่อเพิ่มจำนวนเครื่องลูกข่ายเป็นสองเท่าจากจำนวนเริ่มต้น การใช้งาน HTTP ย่อมเพิ่มขึ้นเป็น

สองเท่าเช่นกัน ในกรณีที่เครือข่ายยังคงสามารถให้บริการได้โดยไม่เกิดภาวะคับคั่ง ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ต้องเพิ่มขึ้นตามสัดส่วนการเพิ่มขึ้นของจำนวนเครื่องลูกข่าย ดังนั้น ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานของ elw4over6, 4rd และ 4over6 มีแนวโน้มเพิ่มขึ้นแบบเส้นตรงแสดงให้เห็นว่า กระบวนการเหล่านี้สามารถประมวลผลแพ็กเก็ตที่เพิ่มขึ้นได้อย่างมีประสิทธิภาพมากกว่า lw4over6 ซึ่งสามารถนำไปใช้อ้างอิงในการเปรียบเทียบค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP ต่อไป

จากรูปที่ 4-2 (b) กระบวนการที่มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของผู้ให้บริการต่ำที่สุดก็คือ 4rd และ 4over6 ผลลัพธ์ที่ดีรองลงมาคือ elw4over6 และ lw4over6 ตามลำดับ สำหรับ 4rd และ 4over6 มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของผู้ให้บริการเท่ากับศูนย์ เนื่องจากทั้งสองกระบวนการสามารถกำหนดอุโมงค์สื่อสารไปยังเครือข่ายปลายทางโดยตรงก่อนเริ่มการติดต่อสื่อสาร โดย 4rd สามารถกำหนดอุโมงค์สื่อสารปลายทางโดยอาศัย 4rd rule ส่วน 4over6 สามารถกำหนดอุโมงค์สื่อสารโดยอาศัยตารางกำหนดเส้นทางแบบพิเศษ สำหรับ elw4over6 อนุญาตให้การติดต่อสื่อสารเกิดขึ้นโดยส่งผ่านอุปกรณ์ของผู้ให้บริการในช่วงเริ่มต้น แต่ภายหลังจากอุปกรณ์ฝั่งผู้ใช้งานทั้งสองปรับปรุงข้อมูลอุโมงค์สื่อสารเสร็จสิ้นจึงสามารถเริ่มการติดต่อสื่อสารโดยตรงได้ สำหรับ lw4over6 เป็นเพียงกระบวนการเดียวที่ต้องส่งข้อมูลไปยังเครื่องปลายทางโดยอาศัยอุปกรณ์ฝั่งผู้ให้บริการเสมอ ส่งผลให้มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการสูงที่สุด เมื่อพิจารณาความสัมพันธ์ระหว่างจำนวนเครื่องลูกข่ายกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการ พบว่ามีเพียง lw4over6 และ elw4over6 เท่านั้นที่มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU เพิ่มขึ้นตามจำนวนเครื่องลูกข่าย สำหรับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของ elw4over6 เพิ่มขึ้นเพียงเล็กน้อยเท่านั้น เนื่องจากอุปกรณ์ฝั่งผู้ให้บริการของ elw4over6 ทำหน้าที่ส่งต่อแพ็กเก็ตเฉพาะช่วงที่อุปกรณ์ฝั่งผู้ใช้งานกำลังดำเนินการสร้างอุโมงค์อุโมงค์สื่อสารเท่านั้น เมื่อสร้างอุโมงค์สื่อสารโดยตรงสำเร็จอุปกรณ์ฝั่งผู้ให้บริการของ elw4over6 ก็ไม่จำเป็นต้องทำหน้าที่เป็นตัวกลางในการส่งต่อแพ็กเก็ตอีกต่อไป ซึ่งแตกต่างจากค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของ lw4over6 ที่มีค่าสูงและเพิ่มขึ้นอย่างมากตามจำนวนเครื่องลูกข่าย เพราะอุปกรณ์ฝั่งผู้ให้บริการของ lw4over6 ต้องทำหน้าที่ส่งต่อแพ็กเก็ตที่ได้รับจากอุปกรณ์ฝั่งผู้ใช้งานไปยังเครื่องปลายทางเสมอ จึงได้รับผลกระทบจากการเพิ่มจำนวนเครื่องลูกข่ายอย่างต่อเนื่อง

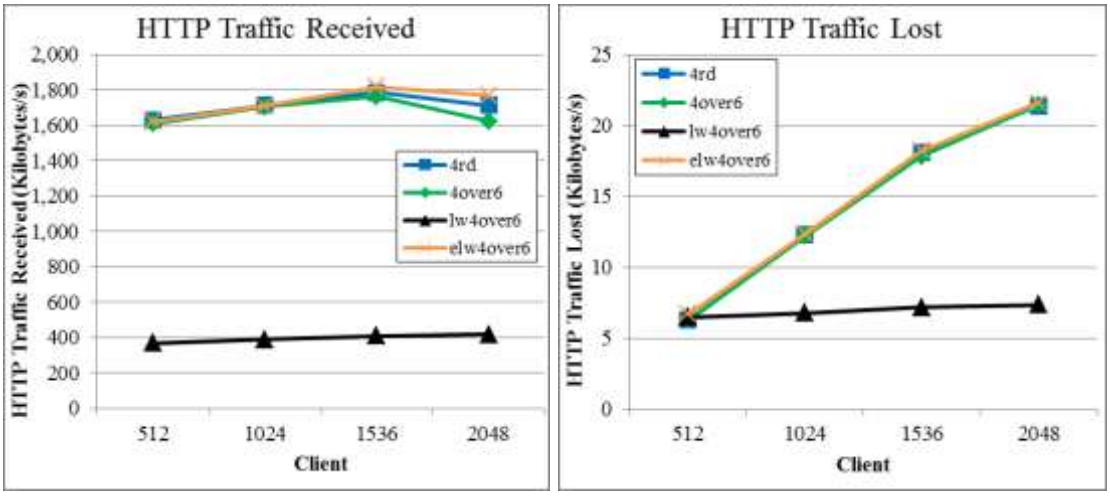
#### 4.2.1.2 ประสิทธิภาพการรับส่งข้อมูลด้วย HTTP เมื่อเครื่องปลายทางที่อยู่ภายในเครือข่าย

ประสิทธิภาพการรับส่งข้อมูลด้วย HTTP เป็นตัวบ่งชี้ประสิทธิภาพที่ชัดเจนที่สุดในการเปรียบเทียบประสิทธิภาพการทำงานของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการ เนื่องจากแสดงถึงประสิทธิภาพในการส่งข้อมูลที่เกิดขึ้นผ่านกระบวนการเปลี่ยนถ่ายอย่างแท้จริง ตัวชี้วัดประสิทธิภาพการรับส่งข้อมูลด้วย HTTP ประกอบด้วย ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP, ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP และค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object โดยผลลัพธ์จากการทดลองของตัวชี้วัดข้างต้นแสดงรายละเอียดดังตารางที่ 4-4 เพื่อให้ง่ายต่อการศึกษาข้อมูลจากตารางที่ 4-4 ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP และค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ถูกนำมาแสดงในรูปแบบกราฟดัง

รูปที่ 4-3 และค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ถูกนำมาแสดงในรูปแบบกราฟดังรูปที่ 4-4

ตารางที่ 4-4 ประสิทธิภาพการรับส่งข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย

| ประสิทธิภาพการรับส่งข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย | | | | | |
|---|---|---|---|---|---|
| ตัวชี้วัด | จำนวนเครื่องลูกข่าย | กระบวนการเปลี่ยนถ่าย | | | |
| | | 4rd | 4over6 | lw4over6 | elw4over6 |
| ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP (Kilobytes/s) | 512 | 1,627.46 | 1,607.07 | 367.83 | 1,618.30 |
| | 1,024 | 1,712.19 | 1,703.97 | 386.51 | 1,710.20 |
| | 1,536 | 1,784.57 | 1,760.81 | 409.03 | 1,815.96 |
| | 2,014 | 1,710.78 | 1,621.94 | 416.18 | 1,766.01 |
| ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP (Kilobytes/s) | 512 | 6.27 | 6.27 | 6.48 | 6.69 |
| | 1,024 | 12.30 | 12.21 | 6.78 | 12.34 |
| | 1,536 | 18.06 | 17.81 | 7.17 | 18.19 |
| | 2,014 | 21.35 | 21.42 | 7.36 | 21.57 |
| ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object (seconds) | 512 | 0.75 | 0.75 | 2.72 | 0.74 |
| | 1,024 | 1.53 | 1.55 | 2.89 | 1.53 |
| | 1,536 | 2.27 | 2.30 | 3.08 | 2.23 |
| | 2,014 | 2.82 | 2.85 | 3.16 | 2.81 |




(a) (b)

รูปที่ 4-3 ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP (a) และค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP (b) เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย

รูปที่ 4-3 (a) แสดงค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP ของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการ เมื่อเครือข่ายผู้ใช้งานมีเครื่องลูกข่ายน้อยกว่า 1,536 เครื่อง กระบวนการ 4rd, elw4over6 และ 4over6 มีค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP ใกล้เคียงกันอย่างมาก มีเพียง lw4over6 เท่านั้นที่มีค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP ต่ำที่สุดซึ่ง

สอดคล้องกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน ยกเว้นเพียงกรณีของ elw4over6 ซึ่งมี HTTP traffic receive ต่ำกว่า 4rd เล็กน้อย แต่เมื่อเครือข่ายผู้ใช้งานมีจำนวนเครื่องลูกข่ายตั้งแต่ 1,536 เครื่องขึ้นไป กระบวนการเปลี่ยนถ่ายที่มีค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP สูงที่สุดคือ elw4over6 ผลลัพธ์ที่ดีรองลงมาได้แก่ 4rd, 4over6 และ lw4over6 ตามลำดับ ซึ่งสอดคล้องกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของฝั่งผู้ใช้งานทุกประการ สาเหตุที่ elw4over6 มี HTTP traffic receive ไม่สอดคล้องกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของฝั่งผู้ใช้งานเมื่อเครือข่ายผู้ใช้งานมีเครื่องลูกข่ายน้อยกว่า 1,536 เครื่อง เนื่องจาก elw4over6 มีการสร้างอุโมงค์สื่อสารโดยตรงไปยังเครือข่ายปลายทางหลังจากเริ่มต้นการติดต่อสื่อสาร ในขณะที่กำลังดำเนินการสร้างอุโมงค์สื่อสาร แพ็กเก็ตจากอุปกรณ์ฝั่งผู้ใช้งานยังคงถูกส่งไปยังเครื่องปลายทางโดยอาศัยอุปกรณ์ฝั่งผู้ให้บริการ ดังนั้นอัตราการส่งของ TCP จึงเพิ่มขึ้นช้ากว่ากระบวนการอื่นๆ แต่เมื่อเพิ่มจำนวนเครื่องลูกข่ายมากขึ้น ส่งผลให้มีปริมาณข้อมูล HTTP เพิ่มขึ้นจนทำให้ระบบเข้าสู่ภาวะคับคั่ง อัตราการส่งข้อมูลของ TCP ของทุกกระบวนการจึงเริ่มลดลง elw4over6 ซึ่งมีเวลาในการประมวลผลต่ำสุดจึงมีอัตราการส่งข้อมูลสูงสุดมากกว่ากระบวนการอื่นๆ ส่งผลให้ elw4over6 มีค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP สูงสุดเมื่อเครือข่ายผู้ใช้งานมีเครื่องลูกข่ายจำนวนมาก

จากค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย แสดงให้เห็นว่า elw4over6 ประสบความสำเร็จในการปรับปรุงประสิทธิภาพ เนื่องจากอัตราการรับข้อมูลด้วย HTTP ของ elw4over6 มีค่าสูงขึ้นกว่า lw4over6 ซึ่งเป็นกระบวนการเปลี่ยนถ่ายต้นแบบอย่างเห็นได้ชัด โดยอัตราการรับข้อมูลด้วย HTTP ของ elw4over6 มีค่าสูงกว่า 4rd และ 4over6 เล็กน้อยซึ่งถือได้ว่าเป็นกระบวนการเปลี่ยนถ่ายที่มีอัตราการรับส่งข้อมูลสูงที่สุดในกรณีที่เครื่องปลายทางอยู่ภายในเครือข่ายผู้ให้บริการเดียวกัน

รูปที่ 4-3 (b) แสดงค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการ ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของกระบวนการเปลี่ยนถ่ายที่มีแนวโน้มเพิ่มขึ้นและเกาะกลุ่มกัน ได้แก่ 4rd, elw4over6 และ 4over6 โดยเรียงลำดับน้อยไปหามาก ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของทั้ง 3 กระบวนการแตกต่างกันเพียงเล็กน้อยเท่านั้น เนื่องจากทั้ง 3 กระบวนการสามารถเชื่อมต่อไปยังเครือข่ายผู้ใช้งานอื่นภายในผู้ให้บริการเดียวกันโดยตรง เมื่อมีจำนวนเครื่องลูกข่ายน้อยกว่า 2,048 เครื่อง ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของทั้ง 3 กระบวนการเพิ่มขึ้นและมีแนวโน้มเป็นเส้นตรง แต่เมื่อเครือข่ายผู้ใช้งานมีเครื่องลูกข่ายจำนวน 2,048 เครื่อง ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของทั้ง 3 กระบวนยังคงเพิ่มขึ้น แต่ก็ยังน้อยกว่าเมื่อเทียบกับแนวโน้มเส้นตรง ส่วนอีกกระบวนการที่เหลือคือ lw4over6 ซึ่งมีผลลัพธ์แตกต่างจากกระบวนการอื่นๆ เนื่องจาก lw4over6 ไม่สามารถเชื่อมต่อไปยังเครือข่ายผู้ใช้งานอื่นภายในผู้ให้บริการเดียวกันโดยตรง อุปกรณ์ฝั่งผู้ใช้งานต้องส่งแพ็กเก็ตไปยังเครื่องปลายทางผ่านอุปกรณ์ฝั่งผู้ให้บริการเท่านั้น ส่งผลให้เส้นทางในการส่งข้อมูลไปยังอุปกรณ์ฝั่งผู้ให้บริการเกิดความคับคั่ง แม้ว่าเครือข่ายผู้ใช้งานมีเครื่องลูกข่ายเพียง 512 เครื่องก็ตาม ดังนั้นค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของ lw4over6 จึงมีแนวโน้มค่อนข้างทรงตัว เนื่องจาก TCP ปรับอัตราการส่งข้อมูลให้มีความสัมพันธ์กับภาวะคับคั่งที่เกิดขึ้นภายในระบบ

แม้ว่าค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของ lw4over6 จะมีค่าน้อยที่สุด แต่ก็เป็นผลสืบเนื่องจากค่าเฉลี่ยของอัตราการับข้อมูลด้วย HTTP ที่มีค่าน้อยที่สุดเช่นกัน เมื่อนำค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP มาคิดเป็นร้อยละโดยเทียบกับค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP ดังตารางที่ 4-5 เห็นได้อย่างชัดเจนว่า lw4over6 มีร้อยละของการสูญหายของข้อมูลจากการส่งด้วย HTTP ต่อการรับข้อมูลด้วย HTTP สูงกว่ากระบวนการอื่นๆ สำหรับค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของ elw4over6 มีค่าใกล้เคียงกับ 4rd และ 4over6 ซึ่งถือได้ว่าเป็นกลุ่มที่มีร้อยละของการสูญหายของข้อมูลจากการส่งด้วย HTTP ต่อการรับข้อมูลด้วย HTTP ต่ำที่สุด

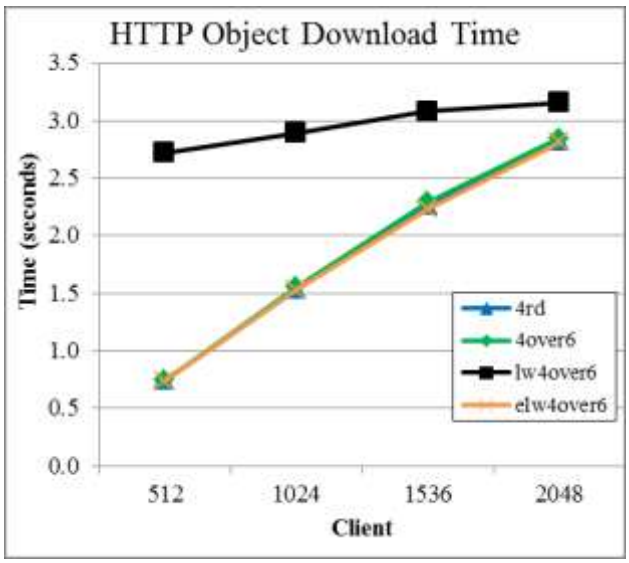
ตารางที่ 4-5 ร้อยละของการสูญหายของข้อมูลจากการส่งด้วย HTTP ต่อการรับข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย

| ตัวชี้วัด | จำนวนเครื่องลูกข่าย | กระบวนการเปลี่ยนถ่าย | | | |
|---|---|---|---|---|---|
| | | 4rd | 4over6 | lw4over6 | elw4over6 |
| ร้อยละของการสูญหายของข้อมูลจากการส่งด้วย HTTP ต่อการรับข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย (%) | 512 | 0.39 | 0.39 | 1.76 | 0.41 |
| | 1,024 | 0.72 | 0.72 | 1.75 | 0.72 |
| | 1,536 | 1.01 | 1.01 | 1.75 | 1.00 |
| | 2,014 | 1.25 | 1.32 | 1.77 | 1.22 |

อย่างไรก็ตาม ค่าเฉลี่ยของอัตราการับข้อมูลด้วย HTTP ที่ต่ำของ lw4over6 นั้นมีความน่าสนใจอย่างมาก โดยอัตราการับข้อมูลที่มีค่าต่ำของโปรแกรมประยุกต์ที่ส่งข้อมูลด้วย TCP หมายความว่า TCP ต้องดำเนินการส่งข้อมูลซ้ำเป็นจำนวนมาก ดังนั้นค่าเฉลี่ยของอัตราการสูญหายจากการส่งข้อมูลในชั้น IP-layer ในตารางที่ 4-6 ถูกนำมาใช้เพิ่มเติม เพื่อวัดอัตราสูญหายจากการส่งข้อมูลทั้งหมด โดยค่าเฉลี่ยของอัตราการสูญหายจากการส่งข้อมูลของ lw4over6 ในชั้น IP-layer มีแนวโน้มสูงขึ้นเมื่อเพิ่มจำนวนเครื่องลูกข่าย อีกทั้งค่าเฉลี่ยของอัตราการสูญหายจากการส่งข้อมูลในชั้น IP-layer ของ lw4over6 มีค่าสูงที่สุด ซึ่งแสดงให้เห็นว่า lw4over6 มีประสิทธิภาพการเชื่อมต่อไปยังเครื่องปลายทางภายในเครือข่ายต่ำกว่ากระบวนการอื่นๆ อย่างมีนัยสำคัญ

ตารางที่ 4-6 ค่าเฉลี่ยของอัตราการสูญหายจากการส่งข้อมูลในชั้น IP-layer เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย

| **ตัวชี้วัด** | **จำนวนเครื่องลูกข่าย** | **กระบวนการเปลี่ยนถ่าย** | | | |
|---|---|---|---|---|---|
| | | **4rd** | **4over6** | **lw4over6** | **elw4over6** |
| ค่าเฉลี่ยของอัตราการสูญหายจากการส่งข้อมูลในชั้น IP-layer (packets/s) | 512 | 2,523.36 | 2,542.44 | 12,704.82 | 2,488.37 |
| | 1,024 | 8,487.81 | 8,528.44 | 39,568.93 | 8,358.54 |
| | 1,536 | 19,235.86 | 19,269.88 | 60,240.01 | 18,991.15 |
| | 2,014 | 45,542.36 | 45,788.10 | 88,572.97 | 44,913.99 |




รูปที่ 4-4 ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object
เมื่อเครื่องปลายทางอยู่ภายในเครือข่าย

จากรูปที่ 4-4 ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ของกระบวนการเปลี่ยนถ่ายอื่นๆ ต่างมีแนวโน้มไปในทิศทางเดียวกัน ยกเว้นเพียง lw4over6 กระบวนการเปลี่ยนถ่ายที่มีค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ต่ำที่สุดคือ elw4over6 ผลลัพธ์ที่ดีรองลงมาได้แก่ 4rd, 4over6 และ lw4over6 ตามลำดับ ซึ่งยังคงสอดคล้องกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน โดยค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ของทั้ง 3 กระบวนการแรกไม่แตกต่างกันมากนัก ยกเว้น lw4over6 เพียงกระบวนการเดียวที่มีค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object สูงกว่ากระบวนการอื่นอย่างเห็นได้ชัด เนื่องจากไม่สามารถสร้างอุโมงค์สื่อสารไปยังเครือข่ายของเครื่องปลายทางได้โดยตรง

แม้ว่าค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ของ elw4over6, 4rd และ 4over6 จะมีแนวโน้มเพิ่มขึ้นอย่างรวดเร็วเมื่อเทียบกับ lw4over6 แต่ทั้ง 3 กระบวนการก็ยังคงมีค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ต่ำกว่า lw4over6 อีกทั้ง elw4over6, 4rd และ 4over6 ยังมีค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP สูงกว่าอีกด้วย ดังนั้นจึงสามารถสรุปได้ว่าทั้ง 3 กระบวนการมีประสิทธิภาพที่สูงกว่า lw4over6 นอกจากนี้ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ของ elw4over6 มีค่าต่ำที่สุด ซึ่งต่ำกว่า 4rd และ 4over6 เล็กน้อย ด้วยเหตุนี้ทำให้ elw4over6 เป็นหนึ่งในกระบวนการเปลี่ยนที่มีประสิทธิภาพในการเชื่อมต่อภายในเครือข่ายผู้ให้บริการที่สูงที่สุดกระบวนการหนึ่ง

### 4.2.2 รูปแบบที่มีการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายนอกเครือข่าย

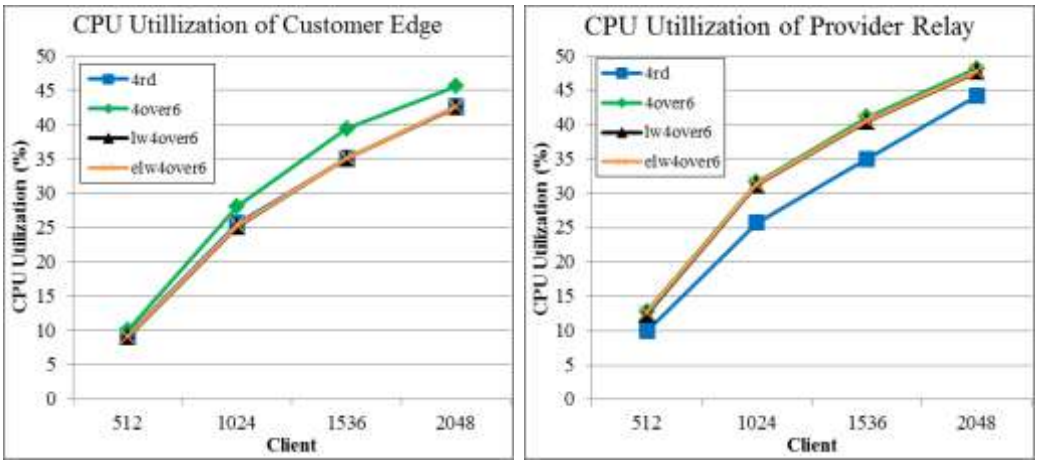
ในการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายนอกเครือข่ายของทุกกระบวนการเปลี่ยนถ่าย แพ็กเก็ตทั้งหมดจะถูกส่งจากอุปกรณ์ฝั่งผู้ใช้งานผ่านอุปกรณ์ฝั่งผู้ให้บริการที่ทำหน้าที่เป็นอุโมงค์สื่อสารปลายทางเพื่อส่งต่อไปยังเครื่องปลายทางที่อยู่ภายนอกเครือข่าย ประสิทธิภาพของรูปแบบการเชื่อมต่อจึงขึ้นอยู่กับอุปกรณ์ฝั่งผู้ให้บริการเป็นหลัก ซึ่งผลการเปรียบเทียบประสิทธิภาพการให้บริการโดยใช้ตัวชี้วัดข้างต้นมีรายละเอียดดังต่อไปนี้

#### 4.2.2.1 อัตราส่วนการใช้งาน CPU เมื่อเครื่องปลายทางที่อยู่ภายนอกเครือข่าย

ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานและอุปกรณ์ฝั่งผู้ให้บริการในแต่ละกระบวนการเปลี่ยนถ่าย ซึ่งมีการติดต่อสื่อสารกับเครื่องปลายทางที่อยู่ภายนอกเครือข่ายแสดงรายละเอียดดังตารางที่ 4-7 นอกจากนี้ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ถูกนำมาแสดงข้อมูลในรูปแบบกราฟดังรูปที่ 4-5 เพื่อให้ง่ายต่อการศึกษาข้อมูล

ตารางที่ 4-7 ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน และอุปกรณ์ฝั่งผู้ให้บริการ เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย

| ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย (ร้อยละ) | | | | | |
|---|---|---|---|---|---|
| ชนิดอุปกรณ์ | จำนวนเครื่องลูกข่าย | กระบวนการเปลี่ยนถ่าย | | | |
| | | 4rd | 4over6 | lw4over6 | elw4over6 |
| อุปกรณ์ฝั่งผู้ใช้งาน | 512 | 9.25 | 9.92 | 8.98 | 9.03 |
| | 1,024 | 25.58 | 28.07 | 25.08 | 25.30 |
| | 1,536 | 35.12 | 39.38 | 35.00 | 35.10 |
| | 2,014 | 42.63 | 45.56 | 42.53 | 42.62 |
| อุปกรณ์ฝั่งผู้ให้บริการ | 512 | 9.91 | 12.81 | 12.38 | 12.70 |
| | 1,024 | 25.77 | 31.60 | 31.27 | 31.41 |
| | 1,536 | 35.06 | 41.10 | 40.51 | 40.60 |
| | 2,014 | 44.21 | 48.17 | 47.66 | 47.74 |




(a) (b)

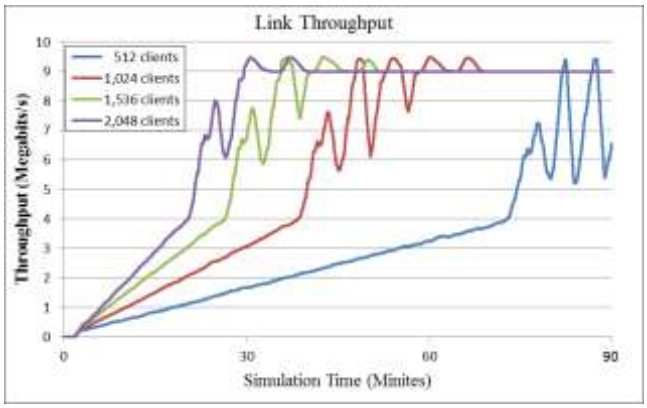
รูปที่ 4-5 ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน (a) และอุปกรณ์ฝั่งผู้ให้บริการ (b) เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย

จากรูปที่ 4-5 ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการแบ่งออกเป็น 2 ส่วนตามชนิดของอุปกรณ์ ได้แก่ ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานซึ่งแสดงดังรูปที่ 4-5 (a) และค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ฝั่งผู้ให้บริการซึ่งแสดงดังรูปที่ 4-5 (b) กระบวนการที่ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของ

อุปกรณ์ฝั่งผู้ใช้งานต่ำที่สุดคือ lw4over6 ผลลัพธ์ที่ดีรองลงมาคือ elw4over6, 4rd และ 4over6 ตามลำดับ โดยค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานของ elw4over6, 4rd และ 4over6 สอดคล้องกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานเมื่อเครื่องปลายทางอยู่ภายในเครือข่าย มีเพียง lw4over6 เท่านั้นที่มีผลลัพธ์ที่ดีขึ้น เนื่องจากในกรณีที่เครื่องปลายทางอยู่ภายนอกเครือข่าย ทุกกระบวนการต้องส่งแพ็กเก็ตทั้งหมดผ่านอุปกรณ์ฝั่งผู้ใช้งานเท่านั้น lw4over6 ซึ่งระบุอุโมงค์สื่อสารตายตัวไปยังอุปกรณ์ฝั่งผู้ให้บริการจึงได้เปรียบกว่ากระบวนการอื่นๆ อย่างไรก็ตามค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของ lw4over6, elw4over6 และ 4rd มีความแตกต่างกันเพียงเล็กน้อยเท่านั้น สำหรับ elw4over6 ใช้ข้อมูล Bypass Scope เพื่อคัดกรองแพ็กเก็ตภายในเครือข่ายออกมา โดยที่แพ็กเก็ตอื่นๆ จะถูกส่งไปยังอุปกรณ์ฝั่งผู้ให้บริการ ส่วน 4rd นั้นใช้ 4rd rule ในการคัดกรองแพ็กเก็ต แต่เนื่องจาก 4rd rule มีความซับซ้อนมากกว่า Bypass Scope ส่งผลให้ 4rd มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ที่สูงกว่า และสำหรับกระบวนการสุดท้าย 4over6 มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU สูงที่สุด เนื่องจากอุปกรณ์ฝั่งผู้ใช้งานของ 4over6 บันทึกอุโมงค์สื่อสารทั้งหมดของทุกเครือข่ายปลายทาง เมื่อมีข้อมูลของอุโมงค์สื่อสารจำนวนมากส่งผลให้ 4over6 ใช้ระยะเวลาในการค้นหาเพื่อระบุอุโมงค์สื่อสารปลายทางมากยิ่งขึ้น เมื่อพิจารณาความสัมพันธ์ระหว่างจำนวนเครื่องลูกข่ายกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน พบว่าทุกกระบวนการเปลี่ยนถ่ายต่างมีค่าเฉลี่ยของอัตราการใช้งาน CPU สูงขึ้นเมื่อมีจำนวนเครื่องลูกข่ายเพิ่มมากขึ้น ซึ่งมีสาเหตุจากปริมาณการเชื่อมต่อที่มีจำนวนมากขึ้น เมื่อเพิ่มจำนวนเครื่องลูกข่ายจาก 512 เครื่องเป็น 1,024 เครื่อง ความชันของค่าเฉลี่ยของอัตราการใช้งาน CPU มีค่ามากที่สุด และค่อยๆ ลดลงจนเริ่มลู่เข้าสู่ค่าคงที่

จากรูปที่ 4-5 (b) กระบวนการที่มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการต่ำที่สุดคือ 4rd ส่วนกระบวนการที่เหลือมีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ใกล้เคียงกัน สาเหตุที่กระบวนการที่เหลือมีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ใกล้เคียงกันและสูงกว่า 4rd เนื่องจากทั้งสามกระบวนการมีการดำเนินการแบบ stateful และมีขั้นตอนการบันทึกข้อมูลอุโมงค์สื่อสารโดยอุปกรณ์ฝั่งผู้ให้บริการที่คล้ายคลึงกัน ซึ่งแตกต่างจาก 4rd ที่มีการดำเนินการแบบ stateless อุปกรณ์ฝั่งผู้ให้บริการของ 4rd จึงไม่จำเป็นต้องบันทึกข้อมูลอุโมงค์สื่อสารเพิ่มเติมส่งผลให้ 4rd มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ต่ำที่สุด เมื่อพิจารณาความสัมพันธ์ระหว่างจำนวนเครื่องลูกข่ายกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการ พบว่าทุกกระบวนการเปลี่ยนถ่ายต่างมีค่าเฉลี่ยของอัตราการใช้งาน CPU สูงขึ้นเมื่อมีจำนวนเครื่องลูกข่ายเพิ่มมากขึ้น ซึ่งมีสาเหตุจากปริมาณการเชื่อมต่อที่มีจำนวนมากขึ้นเช่นเดียวกันกับค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการ เมื่อเพิ่มจำนวนเครื่องลูกข่ายจาก 512 เครื่องเป็น 1,024 เครื่อง ความชันของค่าเฉลี่ยของอัตราการใช้งาน CPU มีค่ามากที่สุดและค่อยๆ ลดลงจนเริ่มลู่เข้าสู่ค่าคงที่เมื่อเพิ่มจำนวนเครื่องลูกข่าย โดยแนวโน้มค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU กับจำนวนเครื่องลูกข่ายสอดคล้องกับค่าเฉลี่ยอัตราการใช้งานลิงค์ของเกตเวย์ของผู้ให้บริการซึ่งเชื่อมต่อไปยังอินเทอร์เน็ตดังแสดงในรูปที่ 4-6 กราฟแต่ละเส้นแสดงถึงจำนวนเครื่องลูกข่าย โดยในแกน X เป็นเวลาที่ทำการบันทึกผลการจำลอง และในแกน Y เป็นค่าเฉลี่ยอัตราการใช้งานลิงค์ จากรูปที่ 4-6 เมื่อเครื่องลูกข่ายมีจำนวน 512 เครื่องใช้เวลาในการเพิ่ม

ค่าเฉลี่ยอัตราการใช้งานลิงค์จนกระทั่งสูงสุดประมาณ 80 นาที แต่เมื่อเพิ่มจำนวนเครื่องลูกข่ายเป็น 1,024 เครื่องใช้เวลาในการเพิ่มค่าเฉลี่ยอัตราการใช้งานลิงค์จนกระทั่งสูงสุดลดลงอย่างมาก แต่เวลาที่ใช้ในการเพิ่มค่าเฉลี่ยอัตราการใช้งานลิงค์จนกระทั่งสูงสุดลดลงก็ค่อยๆ ลดลงเมื่อเพิ่มจำนวนเครื่องลูกข่าย ส่งผลให้ความชันของค่าเฉลี่ยของอัตราการใช้งาน CPU มีค่ามากที่สุด และค่อยๆ ลดลงจนเริ่มลู่เข้าสู่ค่าคงที่ในที่สุด เนื่องจากไม่สามารถเพิ่มอัตราการส่งข้อมูลให้มากกว่าอัตราการรับส่งข้อมูลสูงสุดของลิงค์ได้




รูปที่ 4-6 ค่าเฉลี่ยอัตราการใช้งานลิงค์ของเกตเวย์ของผู้ให้บริการซึ่งเชื่อมต่อไปยังอินเทอร์เน็ต

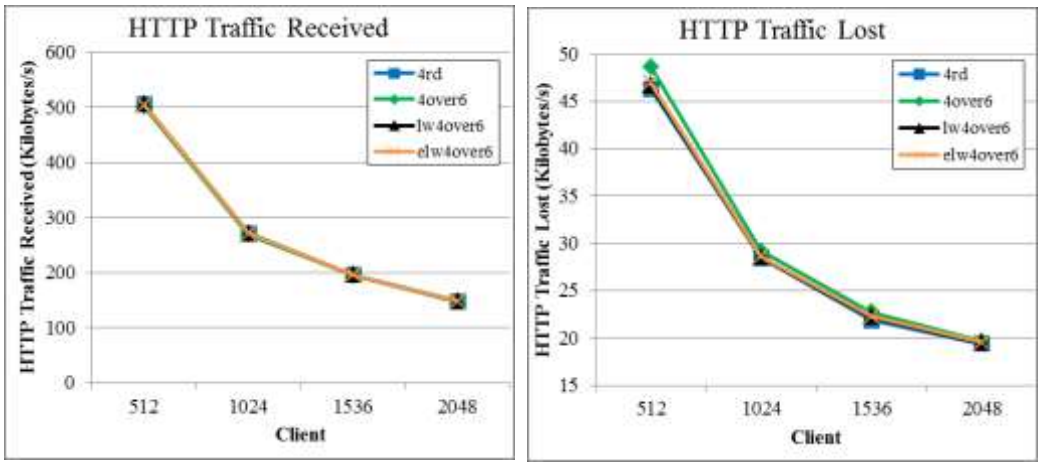
สาเหตุที่ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานและค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ฝั่งผู้ให้บริการของกระบวนการเดียวกันไม่เป็นไปในทิศทางเดียวกัน เนื่องจากอุปกรณ์ฝั่งผู้ใช้งานและอุปกรณ์ฝั่งผู้ให้บริการบางส่วนมีหลักการทำงานที่แตกต่างกัน ยกตัวอย่างเช่น อุปกรณ์ฝั่งผู้ใช้งานของ lw4over6 บันทึกเพียงอุโมงค์สื่อสารตายตัวไปยังอุปกรณ์สื่อสารฝั่งผู้ให้บริการ แต่อุปกรณ์ฝั่งผู้ให้บริการของ lw4over6 ต้องบันทึกข้อมูลอุโมงค์สื่อสารที่เชื่อมต่อไปยังอุปกรณ์ฝั่งผู้ใช้งานทั้งหมด สำหรับ elw4over6 อุปกรณ์ฝั่งผู้ใช้งานกำหนดอุโมงค์สื่อสารปลายทางไปยังอุปกรณ์ฝั่งผู้ใช้งานเมื่อเครื่องปลายทางอยู่ภายนอก Bypass Scope และสามารถกำหนดอุโมงค์สื่อสารโดยตรงหากเครื่องปลายทางอยู่ภายใน Bypass Scope แต่อุปกรณ์ฝั่งผู้ให้บริการของ elw4over6 ต้องบันทึกข้อมูลอุโมงค์สื่อสารที่เชื่อมต่อไปยังอุปกรณ์ฝั่งผู้ใช้งานทั้งหมดไม่ต่างจาก lw4over6 สำหรับ 4over6 ทั้งอุปกรณ์ฝั่งผู้ใช้งานและอุปกรณ์ฝั่งผู้ให้บริการต้องบันทึกข้อมูลอุโมงค์สื่อสารที่เชื่อมต่อไปยังอุปกรณ์ฝั่งผู้ใช้งานทั้งหมด เนื่องจากใช้การปรับปรุงข้อมูลอุโมงค์สื่อสารให้เป็นปัจจุบันผ่านโพรโตคอลกำหนดเส้นทาง และสำหรับ 4rd ซึ่งเป็นเพียงกระบวนการเดียวที่ดำเนินการแบบ stateless ทั้งอุปกรณ์ฝั่งผู้ใช้งานและอุปกรณ์ฝั่งผู้ให้บริการใช้ 4rd rule ในการระบุอุโมงค์สื่อสารปลายทาง จากหลักการทำงานของอุปกรณ์ฝั่งผู้ใช้งานและอุปกรณ์ฝั่งผู้ให้บริการของแต่ละกระบวนการสามารถนำมาสนับสนุนผลการจำลองได้เป็นอย่างดี โดยอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานของ lw4over6 เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่ายควรมีค่าต่ำสุด ผลลัพธ์ที่ดีรองลงมาควรเป็น 4rd หรือ elw4over6 และผลลัพธ์ที่แย่ที่สุดคือ 4over6 ส่วนอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งให้บริการของ 4over6, lw4over6 และ elw4over6 ควรมีใกล้เคียงกัน และสูงกว่า 4rd เนื่องจาก 4rd สามารถใช้ 4rd rule เพื่ออุโมงค์สื่อสารได้โดยไม่ต้องอาศัยตารางบันทึกข้อมูลเหมือนกับกระบวนการอื่นๆ

#### 4.2.2.2 ประสิทธิภาพการรับส่งข้อมูลด้วย HTTP เมื่อเครื่องปลายทางที่อยู่ภายนอกเครือข่าย

ตัวชี้วัดประสิทธิภาพการรับส่งข้อมูลด้วย HTTP ประกอบด้วย ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP, ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP และค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object โดยผลลัพธ์จากการทดลองของตัวชี้วัดข้างต้นแสดงรายละเอียดดังตารางที่ 4-8 เพื่อให้ง่ายต่อการศึกษาข้อมูลจากตารางที่ 4-8 ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP และค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ถูกนำมาแสดงในรูปแบบกราฟดังรูปที่ 4-7 และค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ถูกนำมาแสดงในรูปแบบกราฟดังรูปที่ 4-7

ตารางที่ 4-8 ประสิทธิภาพการรับส่งข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย

| ประสิทธิภาพการรับส่งข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย | | | | | |
|---|---|---|---|---|---|
| ตัวชี้วัด | จำนวนเครื่องลูกข่าย | กระบวนการเปลี่ยนถ่าย | | | |
| | | 4rd | 4over6 | lw4over6 | elw4over6 |
| ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP (Kilobytes/s) | 512 | 504.97 | 501.75 | 504.57 | 503.61 |
| | 1,024 | 271.23 | 268.60 | 270.83 | 270.80 |
| | 1,536 | 195.46 | 194.52 | 195.21 | 195.17 |
| | 2,014 | 147.47 | 146.94 | 147.29 | 147.25 |
| ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP (Kilobytes/s) | 512 | 46.34 | 48.66 | 46.68 | 46.96 |
| | 1,024 | 28.48 | 29.21 | 28.54 | 28.65 |
| | 1,536 | 21.90 | 22.76 | 22.22 | 22.30 |
| | 2,014 | 19.41 | 19.65 | 19.50 | 19.54 |
| ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object (seconds) | 512 | 2.37 | 2.55 | 2.45 | 2.48 |
| | 1,024 | 7.88 | 8.00 | 7.94 | 7.98 |
| | 1,536 | 12.11 | 12.15 | 12.12 | 12.14 |
| | 2,014 | 12.60 | 12.65 | 12.62 | 12.63 |




(a) (b)

รูปที่ 4-7 ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP (a) และค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP (b) เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย

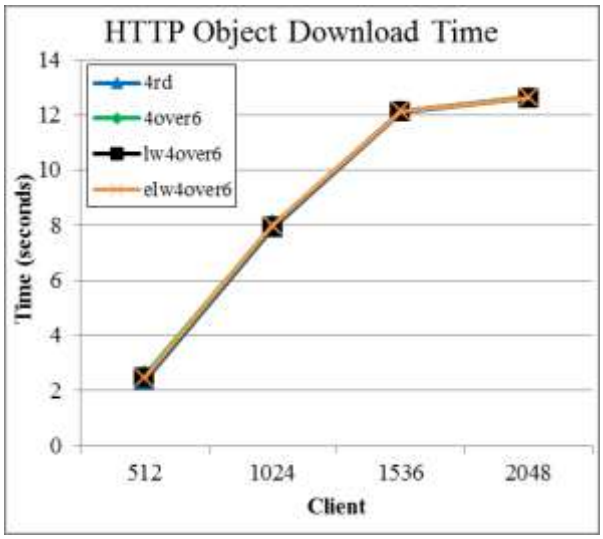
จากรูปที่ 4-7 (a) ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP ของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการมีแนวโน้มลดลงเมื่อเพิ่มจำนวนเครื่องลูกข่าย เนื่องจากในขณะที่เกิดภาวะคับคั่ง TCP จะปรับลดอัตราการส่งข้อมูล ดังนั้นยิ่งมีเครื่องลูกข่ายจำนวนมากทำให้เกิดภาวะคับคั่งอย่างรวดเร็ว ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP จึงมีแนวโน้มลดลงเมื่อเพิ่มจำนวนเครื่องลูกข่าย กระบวนการเปลี่ยนถ่ายที่มีค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP สูงที่สุดคือ 4rd ผลลัพธ์ที่ดีรองลงมาได้แก่ lw4over6, elw4over6 และ 4over6 ตามลำดับ 4rd เป็นกระบวนการที่มีค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP สูงที่สุด เนื่องจากกระบวนการอื่นๆ มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการสูงกว่า 4rd จึงได้รับผลกระทบจากปัญหาคอขวดน้อยกว่า

จากค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย แสดงให้เห็นว่า elw4over6 มีอัตราการรับข้อมูลด้วย HTTP ไม่แตกต่างกับกระบวนการเปลี่ยนถ่ายอื่นๆ แม้ว่าอัตราการรับข้อมูลด้วย HTTP ของ elw4over6 มีค่าต่ำกว่า 4rd และ lw4over6 เล็กน้อยก็ตาม สาเหตุที่ค่าเฉลี่ยของอัตราการับข้อมูลด้วย HTTP ของ elw4over6 มีค่าต่ำกว่าของค่าเฉลี่ยของอัตราการับข้อมูลด้วย HTTP ของ lw4over6 ซึ่งเป็นกระบวนการต้นแบบ เนื่องจาก elw4over6 ถูกเพิ่มเติมขั้นตอนในการคัดกรองแพ็กเก็ตซึ่งอยู่ภายในเครือข่ายผู้ใช้งานออกจากแพ็กเก็ตอื่นๆ จึงส่งผลให้ elw4over6 มีประสิทธิภาพลดลงเล็กน้อยเมื่อเชื่อมต่อไปยังเครื่องปลายทางที่อยู่ภายนอกเครือข่าย

จากรูปที่ 4-7 (b) ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการมีแนวโน้มลดลงเช่นเดียวกับอัตราการรับข้อมูลด้วย HTTP สาเหตุที่ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ปรับลดลงอย่างต่อเนื่อง เนื่องจากโอกาสเกิดการสูญหายจากการส่งข้อมูลด้วย HTTP ย่อมลดลงตามปริมาณการรับส่งข้อมูลด้วย HTTP ที่มีค่าลดลง กระบวนการเปลี่ยนถ่ายที่มีค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ต่ำที่สุดคือ 4rd ผลลัพธ์ที่ดีรองลงมาได้แก่ lw4over6, elw4over6 และ 4over6 ตามลำดับ เนื่องจาก lw4over6, elw4over6 และ 4over6 มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการสูงกว่า 4rd จึงได้รับผลกระทบจากปัญหาคอขวดที่อุปกรณ์ฝั่งผู้ให้บริการมากกว่า สำหรับ lw4over6 มีความเร็วในการประมวลผลของอุปกรณ์ฝั่งผู้ใช้งานสูงที่สุดเป็นปัจจัยที่เข้ามาชดเชยส่งผลให้มีอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ดีเป็นอันดับสองรองจาก 4rd ส่วน elw4over6 มีอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP มากกว่า lw4over6 เล็กน้อย เนื่องจากการประมวลผลที่เพิ่มขึ้นในการคัดกรองแพ็กเก็ตของอุปกรณ์ฝั่งผู้ใช้งาน โดยค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของ lw4over6, elw4over6 และ 4over6 เรียงลำดับตามค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU

แม้ว่าค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของ elw4over6 จะไม่ได้มีค่าน้อยที่สุด แต่เมื่อเพิ่มจำนวนเครื่องลูกข่ายพบว่า ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ของ elw4over6 แทบไม่แตกต่างจากกระบวนการเปลี่ยนถ่ายอื่นๆ ทั้งที่ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP ควรมีความแตกต่าง

อย่างชัดเจนมากยิ่งขึ้น แสดงให้เห็นว่า elw4over6 มีประสิทธิภาพในระดับเดียวกับกระบวนการเปลี่ยนถ่ายอื่นๆ และไม่ได้มีประสิทธิภาพต่ำกว่า lw4over6 มากนัก




รูปที่ 4-8 ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object
เมื่อเครื่องปลายทางอยู่ภายนอกเครือข่าย

จากรูปที่ 4-8 ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการมีค่าเฉลี่ยใกล้เคียงกันอย่างมาก แต่กระบวนการเปลี่ยนถ่ายที่มีค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ต่ำที่สุดคือ 4rd ผลลัพธ์ที่ดีรองลงมาได้แก่ lw4over6, elw4over6 และ 4over6 ตามลำดับ ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ในกรณีที่เครื่องปลายทางอยู่ภายนอกเครือข่ายประเมินได้จากค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการและฝั่งผู้ใช้งาน โดยให้ความสำคัญอุปกรณ์ฝั่งผู้ให้บริการเป็นหลัก สังเกตได้ว่า 4rd มีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการต่ำที่สุด ส่วนสามกระบวนการที่เหลือนั้นมีค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการไม่แตกต่างกันจึงต้องนำค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งานมาเปรียบเทียบต่อโดย lw4over6 มีค่าต่ำที่สุดรองลงมาคือ elw4over6 และ 4over6 ตามลำดับ ซึ่งสอดคล้องกับผลลัพธ์ของค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object อย่างไรก็ตาม ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object ก็เป็นอีกหนึ่งตัวชี้วัดที่ทุกกระบวนการเปลี่ยนถ่ายแทบไม่ความแตกต่างกัน แม้ว่าจะทดสอบด้วยการเพิ่มจำนวนเครื่องลูกข่ายก็ตาม

### 4.3 สรุปผลการทดสอบประสิทธิภาพของกระบวนการเปลี่ยนถ่าย

บทนี้นำเสนอการเปรียบประสิทธิภาพการเชื่อมต่อในกรณีที่เครื่องปลายทางอยู่ภายในเครือข่าย และในกรณีที่เครื่องปลายทางอยู่ภายนอกเครือข่ายของผู้ให้บริการของกระบวนการเปลี่ยนถ่าย 4rd, 4over6, lw4over6 และ elw4over6 ผลลัพธ์ของแต่ละตัวชี้วัดถูกนำมาบันทึกรวมกันในตารางที่ 4-9 โดยใช้ผลการจำลองที่มีจำนวนของเครื่องลูกข่ายภายในเครือข่ายผู้ใช้งาน 2,048 เครื่อง

ตารางที่ 4-9 สรุปผลประสิทธิภาพของตัวชี้วัดที่สนใจในกรณีที่เครื่องปลายทางอยู่ภายในเครือข่าย และในกรณีที่เครื่องปลายทางอยู่ภายนอกเครือข่าย

| **รูปแบบการเชื่อมต่อ** | **ตัวชี้วัด** | **กระบวนการเปลี่ยนถ่าย** | | | |
|---|---|---|---|---|---|
| | | **4rd** | **4over6** | **lw4over6** | **elw4over6** |
| เครื่องปลายทางอยู่ภายในเครือข่าย | ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน (ร้อยละ) | 52.77 | 54.62 | 70.27 | 52.37 |
| | ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการ (ร้อยละ) | 0.20 | 0.20 | 68.34 | 1.43 |
| | ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP (Kilobytes/s) | 1,710.78 | 1,621.94 | 416.18 | 1,766.01 |
| | ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP (Kilobytes/s) | 21.35 | 21.42 | 7.36 | 21.57 |
| | ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object (seconds) | 2.82 | 2.85 | 3.16 | 2.81 |
| เครื่องปลายทางอยู่ภายนอกเครือข่าย | ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ใช้งาน (ร้อยละ) | 42.63 | 45.56 | 42.53 | 42.62 |
| | ค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU ของอุปกรณ์ฝั่งผู้ให้บริการ (ร้อยละ) | 44.21 | 48.17 | 47.66 | 47.74 |
| | ค่าเฉลี่ยของอัตราการรับข้อมูลด้วย HTTP (Kilobytes/s) | 147.47 | 146.94 | 147.29 | 147.25 |
| | ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP (Kilobytes/s) | 19.41 | 19.65 | 19.50 | 19.54 |
| | ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object (seconds) | 12.60 | 12.65 | 12.62 | 12.63 |

เมื่อพิจารณาประสิทธิภาพการเชื่อมจากตารางที่ 4-9 พบว่า กระบวนการ elw4over6 ที่พัฒนามาจาก lw4over6 มีประสิทธิภาพการเชื่อมต่อภายในเครือข่ายสูงที่สุด และสูงขึ้นอย่างเห็นได้ชัดเมื่อเปรียบเทียบกับ lw4over6 ไม่ว่าจะเปรียบเทียบค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU,ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP,ค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object อย่างไรก็ตามเมื่อนำ elw4over6 มาเปรียบเทียบประสิทธิภาพการเชื่อมต่อภายนอกเครือข่ายจะเห็นได้ว่า elw4over6 มีประสิทธิภาพลดลงเพียงเล็กน้อย สำหรับ

4over6 และ lw4over6 มีจุดเด่นแตกต่างกันอย่างสิ้นเชิง โดย lw4over6 มีประสิทธิภาพการเชื่อมต่อภายนอกเครือข่ายสูงที่สุด แต่ก็สูงกว่ากระบวนการ elw4over6 และ 4rd เพียงเล็กน้อยเท่านั้น อย่างไรก็ตาม lw4over6 มีประสิทธิภาพการเชื่อมต่อภายในเครือข่ายต่ำกว่ากระบวนการอื่นๆ อย่างเห็นได้ชัด ยิ่งไปกว่านั้น lw4over6 เป็นเพียงกระบวนการเดียวที่การเชื่อมต่อไปยังเครื่องปลายทางที่อยู่ภายในเครือข่ายของผู้ให้บริการจำเป็นต้องอาศัยอุปกรณ์ฝั่งผู้ให้บริการ ส่งผลให้อุปกรณ์ฝั่งผู้ให้บริการต้องส่งต่อทั้งแพ็กเก็ตที่ภายในเครือข่ายและภายนอกเครือข่าย ซึ่งทำให้ประสิทธิภาพในการให้บริการโดยรวมของ lw4over6 ลดลง ส่วน 4over6 ใช้ตารางกำหนดเส้นทางแบบพิเศษช่วยให้สามารถกำหนดอุโมงค์สื่อสารปลายทางภายในเครือข่ายได้โดยตรง 4over6 จึงมีประสิทธิภาพการเชื่อมต่อภายในเครือข่ายที่สูงกว่า lw4over6 แต่การใช้ตารางกำหนดเส้นทางกลับกลายเป็นดาบสองคม เพราะทำให้ 4over6 มีประสิทธิภาพการเชื่อมต่อภายนอกเครือข่ายต่ำกว่ากระบวนการอื่นๆ สำหรับ 4rd เป็นกระบวนการที่มีประสิทธิภาพสูงมาก ทั้งการเชื่อมต่อภายในเครือข่ายและการเชื่อมต่อภายนอกเครือข่าย เนื่องจาก 4rd ดำเนินการแบบ stateless แต่ถึงกระนั้น 4rd ยังขาดความยืดหยุ่นในการจัดสรรหมายเลข IPv4 และหมายเลข IPv6

# บทที่ 5
# สรุปผลการวิจัยและข้อเสนอแนะ

หลังจากที่ได้นำเสนอกระบวนการเปลี่ยนถ่าย Enhancement of Lightweight 4over6 และเปรียบเทียบประสิทธิภาพกับกระบวนการเปลี่ยนถ่ายอื่นๆ ในบทนี้จะกล่าวถึงการสรุปผลการวิจัย, ประโยชน์ที่ได้รับจากงานวิจัย, การนำกระบวนการเปลี่ยนถ่ายที่นำเสนอไปใช้งาน ตลอดจนปัญหาจากการดำเนินการวิจัยและข้อเสนอแนะเพื่อเป็นประโยชน์ต่อผู้ที่ต้องการนำงานวิจัยชิ้นนี้ไปศึกษาและพัฒนางานวิจัยที่เกี่ยวข้องกับกระบวนการเปลี่ยนถ่ายต่อไป

## 5.1 สรุปผลการวิจัย

การขยายตัวอย่างรวดเร็วของอินเทอร์เน็ตส่งผลให้หมายเลข IPv4 ไม่เพียงพอต่อความต้องการในปัจจุบัน แม้ว่า IPv6 จะสามารถแก้ไขปัญหาดังกล่าวได้ แต่ IPv4 และ IPv6 ก็ยังคงไม่สามารถติดต่อสื่อสารระหว่างกันได้โดยตรง ด้วยเหตุนี้จึงมีการพัฒนากระบวนการเปลี่ยนถ่ายขึ้นเพื่อให้สามารถใช้งาน IPv4 และ IPv6 ควบคู่ไปด้วยกันจนกระทั่งสามารถเปลี่ยนแปลงการใช้งานจาก IPv4 ไปสู่ IPv6 อย่างสมบูรณ์ ผู้วิจัยเลือกศึกษากระบวนการเปลี่ยนถ่ายซึ่งถูกพัฒนาเพื่อใช้งานในช่วงเวลาของการเปลี่ยนแปลงการใช้งานจาก IPv4 ไปสู่ IPv6 ในระยะที่ 3 IPv6 ocean เนื่องจากเป็นช่วงที่ใช้ระยะเวลาในการเปลี่ยนแปลงยาวนานที่สุด

กระบวนการเปลี่ยนถ่ายที่ถูกนำมาศึกษาประกอบด้วย 4over6, DS-lite, lw4over6 และ 4rd จากการวิเคราะห์กระบวนการเปลี่ยนถ่ายเหล่านี้พบว่า กระบวนการเปลี่ยนถ่ายแต่ละกระบวนการต่างก็มีจุดเด่นแตกต่างกันออกไป ยกตัวอย่างเช่น กระบวนการเปลี่ยนถ่ายแบบ stateful มีความยืดหยุ่นในการจัดสรร IPv4 และ IPv6 สูงกว่ากระบวนการเปลี่ยนถ่ายแบบ stateless แต่กระบวนการเปลี่ยนถ่ายแบบ stateless มีประสิทธิภาพในการเชื่อมต่อ IPv4 ที่สูงกว่าเข้ามาทดแทน เป็นต้น ข้อแตกต่างของแต่ละกระบวนการช่วยให้ผู้วิจัยสามารถสรุป คุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมในการให้บริการ IPv4 และ IPv6 เพื่อกำหนดเป็นคุณสมบัติสำคัญของกระบวนการเปลี่ยนถ่ายที่นำเสนอ โดยคุณสมบัติดังกล่าวประกอบด้วย 1) ส่งเสริมให้เครือข่ายของผู้ให้บริการรองรับการให้บริการด้วย IPv6 โดยตรง 2) สามารถจัดสรร IPv4 และ IPv6 ได้อย่างอิสระ 3) สามารถแบ่งจัดสรรหมายเลข IPv4 ให้กับผู้ใช้งานได้ในระดับพอร์ตแบบพลวัต และ 4) รองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง

เมื่อนำคุณสมบัติของกระบวนการเปลี่ยนถ่ายที่เหมาะสมในการให้บริการ IPv4 และ IPv6 มาเปรียบเทียบกับกระบวนการเปลี่ยนถ่ายต่างๆ พบว่า lw4over6 มีคุณสมบัติใกล้เคียงกับคุณสมบัติที่ต้องการมากที่สุด โดยคุณสมบัติของ lw4over6 ที่ขาดไปมีเพียงข้อเดียวคือการรองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง ดังนั้นกระบวนการเปลี่ยนถ่ายที่นำเสนอจึงปรับปรุง lw4over6 ให้รองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรง ซึ่งกระบวนการที่นำเสนอถูกเรียกว่า “Enhancement of Lightweight 4over6” กระบวนการเปลี่ยนถ่ายที่นำเสนอถูกปรับปรุงให้

รองรับการเชื่อมต่อ IPv4 ไปยังเครื่องปลายทางภายในเครือข่ายของผู้ให้บริการเดียวกันได้โดยตรงโดยใช้ Bypass Scope และ Bypass Binding Table เพื่อบำรุงรักษาข้อมูลอุโมงค์สื่อสารปลายทางของอุปกรณ์ฝั่งผู้ใช้งาน สำหรับ Bypass Scope เป็นช่วงของหมายเลข IPv4 ทั้งหมดซึ่งถูกจัดสรรให้กับอุปกรณ์ฝั่งผู้ใช้งาน Bypass Scope จึงถูกนำมาใช้เพื่อคัดกรองระหว่างแพ็กเก็ต IPv4 ภายในเครือข่าย และแพ็กเก็ต IPv4 ภายนอกเครือข่าย และสำหรับ Bypass Binding Table ซึ่งมีลักษณะคล้ายตารางบันทึกอุโมงค์สื่อสารปลายทางของอุปกรณ์ฝั่งผู้ให้บริการ ทำหน้าที่บันทึกอุโมงค์สื่อสารที่เคยติดต่อสื่อสารเพื่อให้การสื่อสารครั้งต่อไปสามารถส่งข้อมูลไปยังเครื่องปลายทางได้โดยตรง ในการแลกเปลี่ยนข้อมูลอุโมงค์สื่อสารใน Bypass Binding Table กระบวนการเปลี่ยนถ่ายที่นำเสนอประยุกต์ใช้ DHCP leasequery เพื่อแลกเปลี่ยนข้อมูล เมื่อข้อมูลอุโมงค์สื่อสารถูกปรับปรุงให้เป็นปัจจุบัน อุปกรณ์ฝั่งผู้ใช้งานสามารถใช้เส้นทางที่เหมาะสมที่สุดในการส่งข้อมูลไปยังเครื่องปลายทางภายในเครือข่าย โดยที่เครื่องต้นทางไม่จำเป็นต้องส่งแพ็กเก็ต IPv4 ไปยังอุปกรณ์ของผู้ให้บริการก่อน

จากผลการทดสอบ กระบวนการเปลี่ยนถ่ายที่นำเสนอสามารถทำงานได้อย่างถูกต้อง โดยกระบวนการที่นำเสนอมีประสิทธิภาพการเชื่อมต่อภายในเครือข่ายสูงที่สุด และสูงขึ้นอย่างเห็นได้ชัดเมื่อเปรียบเทียบกับ lw4over6 ไม่ว่าจะเปรียบเทียบค่าเฉลี่ยของอัตราส่วนการใช้งาน CPU, ค่าเฉลี่ยของอัตราการสูญหายของข้อมูลจากการส่งด้วย HTTP หรือค่าเฉลี่ยของระยะเวลาในการดาวน์โหลด HTTP object อย่างไรก็ตามเมื่อนำกระบวนการที่นำเสนอมาเปรียบเทียบประสิทธิภาพการเชื่อมต่อภายนอกเครือข่ายสังเกตได้ว่า กระบวนการที่นำเสนอมีประสิทธิภาพลดลงเพียงเล็กน้อยเมื่อเปรียบเทียบกับ lw4over6 เนื่องจากขั้นตอนการคัดกรองแพ็กเก็ต IPv4 บนอุปกรณ์ฝั่งผู้ใช้งานด้วย Bypass Scope ที่เพิ่มขึ้น แต่การประมวลผลดังกล่าวเป็นการประมวลผลบนอุปกรณ์ฝังผู้ใช้งานจึงไม่ส่งผลกระทบมากนัก แม้ว่าแพ็กเก็ตในระบบเครือข่ายจะเพิ่มสูงขึ้น ดังนั้นกระบวนการเปลี่ยนถ่ายที่นำเสนอถือได้ว่าเป็นทางเลือกที่ดีอีกทางเลือกหนึ่งในการให้บริการการเชื่อมต่อ IPv4 ด้วยกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6

## 5.2 ประโยชน์ที่ได้รับจากงานวิจัย

งานวิจัยนี้แสดงให้เห็นถึงประสิทธิภาพของกระบวนการเปลี่ยนถ่ายในการให้บริการการเชื่อมต่อ IPv4 โดยใช้อุโมงค์สื่อสาร IPv6 ข้อมูลข้างต้นสามารถใช้ในการวางแผนการให้บริการ IPv4 ด้วยกระบวนการเปลี่ยนถ่ายที่ใช้การสร้างอุโมงค์สื่อสารด้วย IPv6 ได้เป็นอย่างดี โดยในหัวข้อนี้จะสรุปเกี่ยวกับประสิทธิภาพในการให้บริการ และเครือข่ายที่เหมาะสมในการนำกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการไปใช้ ซึ่งข้อสรุปของกระบวนการเปลี่ยนถ่ายแต่ละกระบวนการมีดังต่อไปนี้

DS-lite เป็นกระบวนการที่จัดสรร Public IPv4 address และพอร์ตโดยอุปกรณ์ฝั่งผู้ให้บริการ โดย DS-lite ดำเนินการ NAT และบันทึกข้อมูลอุโมงค์สื่อสารควบคู่ไปด้วยกัน ด้วยเหตุนี้อุปกรณ์ฝั่งผู้ใช้งานจึงถูกปรับปรุงเพิ่มเติมเพียงเล็กน้อย เพื่อให้รองรับการสร้างอุโมงค์สื่อสารเท่านั้น ประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายนอกเครือข่ายของ DS-lite อยู่ในระดับปานกลาง เนื่องจาก DS-lite มีการดำเนินการ NAT แบบรวมศูนย์ แต่ด้านการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายในเครือข่าย DS-lite มีประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทาง

ภายในเครือข่ายในระดับต่ำ เพราะ DS-lite ไม่รองรับการเชื่อมต่อ IPv4 แบบ mesh และมีดำเนินการ NAT แบบรวมศูนย์ ยิ่งกว่านั้น DS-lite มีสามารถในการขยายขนาดการให้บริการในระดับต่ำ เพราะประสิทธิภาพในการให้บริการของ DS-lite จะลดลงตามความสามารถในการดำเนินการ NAT ของอุปกรณ์ฝั่งผู้ให้บริการ อย่างไรก็ตาม DS-lite ก็มีข้อดีด้านความยืดหยุ่น เนื่องจาก DS-lite จัดสรร Public IPv4 address และพอร์ตโดยอุปกรณ์ฝั่งผู้ให้บริการทำให้ DS-lite มีความยืดหยุ่นในการใช้งานสูง ดังนั้น DS-lite จึงเหมาะสำหรับให้บริการกับเครือข่ายขนาดเล็กที่มีงบประมาณน้อยหรือเครือข่ายที่ผู้ใช้งานเป็นเจ้าของอุปกรณ์เอง ซึ่งไม่ต้องการเปลี่ยนอุปกรณ์ฝั่งผู้ใช้งานใหม่ สำหรับประสิทธิภาพในการให้บริการการเชื่อมต่อ IPv4 ของ DS-lite ค่อนข้างต่ำกว่ากระบวนการอื่น อีกทั้งยังไม่สามารถสร้างการเชื่อมต่อ IPv4 ภายในเครือข่ายได้โดยตรง ดังนั้นผู้ให้บริการที่ต้องการใช้ DS-lite ต้องมั่นใจว่าโปรแกรมประยุกต์สำคัญๆ ที่ใช้ภายในเครือข่ายนั้นสามารถใช้งานด้วย IPv6 และใช้การเชื่อมต่อ IPv4 เฉพาะกับเครื่องปลายทางที่ไม่สามารถใช้งาน IPv6 ได้จะเป็นรูปแบบที่เกิดประโยชน์สูงสุด DS-lite จึงเหมาะสำหรับใช้งานบนเครือข่ายที่มีการใช้งาน IPv4 ในปริมาณไม่มากนัก แต่ยังคงต้องการรักษาการเชื่อมต่อ IPv4 ไว้

4over6 ใช้โพรโตคอลกำหนดเส้นทางในการปรับปรุงข้อมูลอุโมงค์สื่อสารปลายทาง โดยแต่ละเครือข่ายผู้ใช้งานจะประกาศ Public IPv4 address ของแต่ละเครือข่ายออกมาเพื่อใช้เป็นข้อมูลในการสร้างอุโมงค์สื่อสาร การดำเนินการ NAT ของ 4over6 จึงดำเนินการโดยอุปกรณ์ฝั่งผู้ใช้งาน 4over6 จึงมีประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายนอกเครือข่ายในระดับสูง และมีประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายในเครือข่ายในระดับสูงเช่นกัน เนื่องจากรองรับการเชื่อมต่อ IPv4 แบบ mesh เมื่อพิจารณาด้านความความสามารถในการรองรับการขยายขนาด 4over6 มีความสามารถในการรองรับการขยายขนาดในระดับปานกลาง แม้ว่า 4over6 จะรองรับการดำเนินการ NAT แบบกระจายและรองรับการเชื่อมต่อ IPv4 แบบ mesh แต่อุปกรณ์ฝั่งผู้ใช้งานของ 4over6 จำเป็นต้องใช้ Public IPv4 address ซึ่งอาจส่งผลให้ Public IPv4 address ไม่เพียงพอต่อการใช้งาน เมื่อผู้ใช้งานเพิ่มจำนวนสูงขึ้น ด้านความยืดหยุ่น 4over6 ก็มีความยืดหยุ่นในระดับปานกลาง เนื่องจากไม่สามารถแบ่งปันการใช้งาน Public IPv4 address ในระดับพอร์ต ดังนั้น 4over6 จึงเหมาะสำหรับเครือข่ายขนาดใหญ่ซึ่งมีเครือข่ายผู้ใช้งานในระดับองค์กร

4rd เป็นกระบวนการเปลี่ยนถ่ายที่มีการดำเนินการแบบ stateless 4rd กำหนดให้เครือข่ายผู้ใช้งานใช้หมายเลข IPv6, IPv4 และ port-set ตามกฎที่ออกแบบไว้ล่วงหน้า 4rd จึงสามารถสร้างอุโมงค์สื่อสารปลายทางเพื่อเชื่อมต่อไปยังอุปกรณ์ฝั่งผู้ให้บริการและอุปกรณ์ฝั่งผู้ใช้งานภายในเครือข่ายได้โดยตรง โดยไม่จำเป็นต้องใช้การปรับปรุงข้อมูลอุโมงค์สื่อสาร ส่งผลให้ 4rd มีประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายนอกเครือข่ายและเครื่องปลายทางภายในเครือข่ายในระดับสูง แต่กระนั้น 4rd กลับมีความยืดหยุ่นในการใช้งานต่ำ เนื่องจากต้องใช้งานหมายเลข IPv6, IPv4 และ port-set ตามกฎที่ออกแบบอย่างเคร่งครัด ข้อเสียดังกล่าวยังส่งผลให้ 4rd มีความสามารถในการรองรับการขยายขนาดในระดับปานกลางเท่านั้น แม้ว่า 4rd จะมีประสิทธิภาพในการใช้งานสูงก็ตาม เนื่องจากผู้ให้บริการต้องจัดสรรหมายเลข IPv6, IPv4 และ port-set ของเครือข่ายผู้ใช้งานใหม่ทุกครั้งที่มีการขยายขนาดของเครือข่าย ด้วยเหตุนี้ 4rd จึง

เหมาะสำหรับเครือข่ายขนาดใหญ่ซึ่งมีเครือข่ายผู้ใช้งานในระดับผู้ใช้งานทั่วไปจนไปถึงระดับองค์กร อีกทั้งเครือข่ายดังกล่าวควรเป็นระบบเครือข่ายที่มีการวางแผนการขยายขนาดการให้บริการที่แน่นอน เพื่อลดผลกระทบจากการเปลี่ยนแปลงหมายเลข IPv6, IPv4 และ port-set ในเครือข่ายผู้ใช้งาน

lw4over6 ออกแบบเพื่อแก้ไขปัญหาของ DS-lite ซึ่งมีการดำเนินการ NAT แบบรวมศูนย์ lw4over6 ใช้การกระจาย Public IPv4 address และ port-set ให้กับอุปกรณ์ผู้ใช้งาน เพื่อให้อุปกรณ์ฝั่งผู้ใช้งานเป็นผู้ดำเนินการ NAT ประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายนอกเครือข่ายของ lw4over6 จึงอยู่ในระดับสูง ในทางกลับกัน lw4over6 กลับมีประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายในเครือข่ายในระดับปานกลางเท่านั้น เพราะอุปกรณ์ฝั่งผู้ใช้งานของ lw4over6 มีอุโมงค์สื่อสารเพียงอุโมงค์เดียวเพื่อเชื่อมต่อไปยังอุปกรณ์ฝั่งผู้ให้บริการ ทำให้ lw4over6 ไม่รองรับการเชื่อมต่อ IPv4 แบบ mesh ด้านความสามารถในการขยายขนาดของ lw4over6 ก็อยู่ในระดับต่ำ เนื่องจาก lw4over6 ไม่รองรับการเชื่อมต่อ IPv4 แบบ mesh ซึ่งส่งผลให้ lw4over6 มีปริมาณแพ็กเก็ตในเส้นทางหลักที่ใช้สำหรับเชื่อมต่อไปยังอุปกรณ์ฝั่งผู้ให้บริการเพิ่มสูงขึ้น อย่างไรก็ตาม lw4over6 มีความยืดหยุ่นในการใช้งานสูง เพราะสามารถจัดสรรหมายเลข IPv6, IPv4 และ port-set ได้อย่างอิสระ ดังนั้นเครือข่ายที่เหมาะสมในการให้บริการด้วย lw4over6 ควรเป็นเครือข่ายขนาดปานกลางซึ่งมีเครือข่ายผู้ใช้งานในระดับผู้ใช้งานทั่วไป โดยโปรแกรมประยุกต์สำคัญๆ ที่ใช้ภายในเครือข่ายนั้นควรรองรับใช้งานด้วย IPv6 และใช้การเชื่อมต่อ IPv4 เฉพาะกับเครื่องปลายทางที่ไม่สามารถใช้งาน IPv6 ได้เพื่อลดผลกระทบจากการไม่รองรับการเชื่อมต่อ IPv4 แบบ mesh ให้ได้มากที่สุด ดังนั้น lw4over6 จึงเหมาะสำหรับใช้งานบนเครือข่ายที่มีการใช้งาน IPv4 ในปริมาณไม่มากนัก แต่ยังคงต้องการรักษาการเชื่อมต่อ IPv4 ไว้เช่นเดียวกับ DS-lite

สำหรับกระบวนการสุดท้าย elw4over6 เป็นกระบวนการที่พัฒนาต่อยอดจาก lw4over6 เพื่อเพิ่มเติมคุณสมบัติการรองรับการเชื่อมต่อ IPv4 แบบ mesh โดย elw4over6 ประยุกต์ใช้ DHCP ในการปรับปรุงข้อมูลอุโมงค์สื่อสารปลายทาง elw4ove6 จึงได้รับคุณสมบัติพื้นฐานมาจาก lw4over6 ไม่ว่าจะเป็นการจัดสรร IPv4 และ IPv6 ได้อย่างอิสระและสามารถแบ่งปันการใช้งาน Public IPv4 address ในระดับพอร์ต ดังนั้น elw4over6 จึงมีประสิทธิภาพการเชื่อมต่อ IPv4 กับเครื่องปลายทางภายนอกเครือข่ายและเครื่องปลายทางภายในเครือข่ายในระดับสูง เพราะ elw4over6 รองรับการเชื่อมต่อ IPv4 แบบ mesh นอกจากนี้ elw4over6 ยังมีความยืดหยุ่นในการใช้งาน และมีความสามารถในการรองรับการขยายขนาดการให้บริการในระดับสูง เนื่องจาก elw4over6 สามารถจัดสรรหมายเลข IPv6, IPv4 และ port-set ได้อย่างอิสระ, รองรับการดำเนินการ NAT แบบกระจาย และรองรับการเชื่อมต่อ IPv4 แบบ mesh ดังนั้น elw4ove6 จึงเหมาะสำหรับเครือข่ายขนาดใหญ่ซึ่งมีเครือข่ายผู้ใช้งานในระดับผู้ใช้งานทั่วไปจนไปถึงระดับองค์กร และยังเหมาะสำหรับเครือข่ายที่มีการขยายขนาดบ่อยครั้ง เนื่องจาก elw4over6 สามารถจัดสรร IPv4 และ IPv6 แยกจากกันได้อย่างอิสระ

## 5.3 การนำกระบวนการเปลี่ยนถ่ายที่นำเสนอไปใช้งาน

กระบวนการเปลี่ยนถ่ายที่นำเสนอ (Enhancement of Lightweight 4over6) มีคุณสมบัติที่เหมาะสมในการนำไปใช้งานจริง ไม่ว่าจะเป็นด้านความยืดหยุ่นในการใช้งาน และด้านความสามารถในการรองรับการขยายขนาดของเครือข่าย ในด้านประสิทธิภาพ ขั้นตอนในการประมวลผลที่เพิ่มขึ้นล้วนดำเนินการโดยอุปกรณ์ฝั่งผู้ใช้งานซึ่งไม่ได้เป็นการประมวลผลโดยส่วนกลาง ส่งผลให้ประสิทธิภาพในการให้บริการไม่ได้ลดลงอย่างมีนัยยะสำคัญ สำหรับอุปกรณ์ภายในกระบวนการเปลี่ยนถ่ายที่ผู้วิจัยได้นำเสนอประกอบไปด้วย DHCP 4o6 Server, lwAFTR และ lwB4 โดย lwAFTR เป็นอุปกรณ์ที่ไม่จำเป็นต้องปรับปรุงหลักการทำงานใดเพิ่มเติม สำหรับ DHCP 4o6 Server และ lwB4 มีการปรับปรุงหลักการทำงานบางส่วนตามที่ได้นำเสนอไปในบทที่ 3 โดยขั้นตอนที่ถือเป็นหัวใจสำคัญในการทำงานก็คือการแลกเปลี่ยนข้อมูลอุโมงค์สื่อสารระหว่าง lwB4 และ DHCP 4o6 Server ซึ่ง lwB4 ต้องทำหน้าที่ตรวจสอบข้อมูลของเครื่องปลายทางว่าสามารถสร้างอุโมงค์สื่อสารโดยตรงได้หรือไม่ หากสามารถสร้างอุโมงค์สื่อสารโดยตรงได้ lwB4 จะร้องขอข้อมูลอุโมงค์สื่อสารปลายทางไปยัง DHCP 4o6 Server จากนั้น DHCP 4o6 Server จะตอบข้อมูลอุโมงค์สื่อสารปลายทางกลับไปยัง lwB4 เมื่อผู้ให้บริการสามารถปรับปรุงการทำงานของ DHCP 4o6 Server และ lwB4 ตามข้อกำหนดดังกล่าว ผู้ให้บริการก็สามารถให้บริการการเชื่อมต่อ IPv4 ด้วยกระบวนการเปลี่ยนถ่ายที่นำเสนอได้

อย่างไรก็ตาม กระบวนการเปลี่ยนถ่ายที่นำเสนอยังต้องการการนิยาม DHCPv4 option และรูปแบบการร้องขอข้อมูลโดยระบุหมายเลข IPv4 และพอร์ตใน DHCPv4 leasequery โดยองค์การกำหนดหมายเลขอินเทอร์เน็ตเพิ่มเติม เพื่อให้การร้องขอข้อมูลอุโมงค์สื่อสารเป็นไปอย่างเหมาะสมโดยไม่ขัดต่อคำแนะนำใน RFC สำหรับสิ่งที่ต้องการกำหนดเพิ่มเติมใน DHCPv4 option ประกอบไปด้วย DHCPv4 OPTION_ASSOCIATED_HOST_INFO, DHCPv4 sub-option Client-identifier ใน Relay Agent Information และ Flag “i” ของ DHCPv4 sub-option Subnet-Information ใน Subnet Allocation Option

## 5.4 ปัญหาและข้อเสนอแนะ

### 5.4.1 ปัญหาจากการดำเนินงานวิจัย

เพื่อให้การศึกษาครอบคลุมกระบวนการเปลี่ยนถ่ายในระยะเดียวกันอย่างครบถ้วน การศึกษากระบวนการเปลี่ยนถ่ายทุกกระบวนการล้วนมีความสำคัญ แม้ว่าบางกระบวนการยังอยู่ในขั้นตอนการร่างก็ตาม โดยกระบวนการที่ยังอยู่ในขั้นตอนการร่าง ประกอบด้วย lw4over6 และ 4rd กระบวนการเปลี่ยนถ่ายเหล่านี้มีข้อมูลบางส่วนซึ่งยังไม่อธิบายในรายละเอียดส่งผลให้เนื้อหาบางส่วนขาดความชัดเจน ยิ่งกว่านั้นในบางครั้งแบบร่างชุดใหม่มีการเปลี่ยนแปลงเพิ่มเติมอย่างมากจึงจำเป็นต้องศึกษาทบทวนข้อมูลเพิ่มเติม ส่งผลให้ต้องใช้เวลาในการศึกษาและปรับปรุงแบบจำลองเครือข่ายเพิ่มขึ้น อย่างไรก็ตาม ในวิทยานิพนธ์ฉบับนี้พยายามอ้างอิงแบบร่างของกระบวนการเปลี่ยนถ่ายให้มีความทันสมัยมากที่สุด โดยหลักการทำงานของกระบวนการเปลี่ยนถ่ายภายในวิทยานิพนธ์ฉบับนี้ยึดถือตามแบบร่างของกระบวนการเปลี่ยนถ่ายที่ระบุไว้ในบรรณานุกรมซึ่งอาจมีรายละเอียดแตกต่างจากหลักการทำงานของกระบวนการเปลี่ยนถ่ายฉบับสมบูรณ์บ้างเล็กน้อย

### 5.4.2 ข้อเสนอแนะ

การแลกเปลี่ยนข้อมูลอุโมงค์สื่อสารด้วย DHCPv4 leasequery over DHCPv6 ของกระบวนการเปลี่ยนถ่ายที่นำเสนอสามารถนำมาประยุกต์ใช้ในการกู้คืนข้อมูลอุโมงค์สื่อสารปลายทางของอุปกรณ์ฝั่งผู้ให้บริการของ lw4over6 ได้ในกรณีที่อุปกรณ์หยุดการทำงานอย่างไม่คาดคิด การกู้คืนข้อมูลอุโมงค์สื่อสารช่วยให้อุปกรณ์ฝั่งผู้ใช้งานสามารถใช้งานด้วยหมายเลข IPv4 และหมายเลข IPv6 เดิมโดยไม่ต้องดำเนินการร้องขอใหม่ การกู้คืนข้อมูลอุโมงค์สื่อสารอาจกู้คืนครั้งละหนึ่งอุโมงค์สื่อสาร โดยใช้รูปแบบการแลกเปลี่ยนข้อมูลอุโมงค์สื่อสารดังที่นิยามในกระบวนการเปลี่ยนถ่ายที่นำเสนอ หรือกู้คืนอุโมงค์สื่อสารแบบพิเศษซึ่งสามารถกู้คืนข้อมูลทั้งหมดในการแลกเปลี่ยนข้อมูลเพียงครั้งเดียว ในการกู้คืนอุโมงค์สื่อสารแบบพิเศษจำเป็นต้องมีการกำหนดรูปแบบการแลกเปลี่ยนข้อมูลเพิ่มเติม เช่น ระบุหมายเลข IPv4 และพอร์ตเท่ากับศูนย์ เพื่อบ่งบอกว่าเป็นการร้องขอแบบพิเศษ, แก้ไขให้ DHCPv4 OPTION_v4_PORTPARAMS สามารถตอบกลับข้อมูลพอร์ตได้มากกว่า 1 ช่วง และเพิ่มข้อกำหนดด้านความปลอดภัยของอุปกรณ์ที่มีสิทธิในการกู้คืนข้อมูลแบบพิเศษ ประเด็นเกี่ยวกับการกู้คืนข้อมูลหลังจากที่อุปกรณ์หยุดการทำงานอย่างไม่คาดคิดจึงเป็นประเด็นที่มีความน่าสนใจ และมีความท้าทายในการพัฒนาอยู่ไม่น้อยเลยทีเดียว

# เอกสารอ้างอิง


[1] R. E. Gilligan and E. Nordmark, “Basic Transition Mechanisms for IPv6 Hosts and Routers,” RFC 4213, October 2005.

[2] K. Moore and B. E. Carpenter, “Connection of IPv6 Domains via IPv4 Clouds,” RFC 3056, February 2001.

[3] W. Townsley and O. Troan, “IPv6 Rapid Deployment on IPv4 Infrastructures (6rd),” RFC 5969, August 2010.

[4] “Evolution Towards IPv6,” Oct-2009. [Online]. Available: http://www.ipv6.org.sa/sites/default/files/Evolution Towards IPv6, STC High-level Plan - STC.pdf. [Accessed: 30-Oct-2014].

[5] M. Townsley, “IPv4 Run-Out and IPv4-IPv6 Co-Existence Scenarios,” RFC 6127, May 2011.

[6] “IPv6 – Google.” [Online]. Available: https://www.google.com/intl/en/ipv6 /statistics.html. [Accessed: 30-Oct-2014].

[7] S. M. Kerner, “IPv6 & IPv4 Will Co-Exist for a Long Time.” [Online]. Available: http://www.enterprisenetworkingplanet.com/netsp/ipv6-ipv4-will-co-exist-for-a-long-time.html. [Accessed: 30-Oct-2014].

[8] A. Conta and S. Deering, “Generic Packet Tunneling in IPv6 Specification,” RFC 2473, December 1998.

[9] Y. Cui, J. Wu, X. Li, M. Xu, and C. Metz, “4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions,” RFC 5747, March 2010.

[10] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, “Multiprotocol Extensions for BGP-4,” RFC 4760, January 2007.

[11] A. Durand, J. Woodyatt, R. Droms, and Y. L. Lee, “Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion,” RFC 6333, August 2011.

[12] T. Mrugalski and D. Hankins, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite,” RFC 6334, August 2011.

[13] Y. Cui, Q. Sun, T. Tsou, I. Farrer, Y. Lee, and M. Boucadair, “Lightweight 4over6: An Extension to the DS-Lite Architecture draft-ietf-softwire-lw4over6-10,” Internet Draft, June 2014.

[14] C. Liu, Q. Sun, and J. Wu, “Dynamic IPv4 Provisioning for Lightweight 4over6 draft-liu-softwire-lw4over6-dhcp-deployment-04,” Internet Draft, July 2014.

[15] Y. Cui, S. Krishnan, Q. Sun, I. Farrer, and M. Siodelski, “DHCPv4-over-DHCPv6 (DHCP 4o6) Transport,” RFC 7341, August 2014.

[16] C. Bao, W. Dec, L. Yeh, T. Mrugalski, I. Farrer, O. Troan, S. Perreault, and X. Deng, “DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients draft-ietf-softwire-map-dhcp-07,” Internet Draft, March 2014.

[17] Y. Cui, Q. Sun, and I. Farrer, “DHCPv4 over DHCPv6 Source Address Option draft-fsc-softwire-dhcp4o6-saddr-opt-00,” Internet Draft, June 2014.

[18] Y. Cui, Qiong, I. Farrer, Q. Sun, Y. Lee, and M. Boucadair, “Dynamic Allocation of Shared IPv4 Addresses draft-ietf-dhc-dynamic-shared-v4allocation-01,” Internet Draft, July 2014.

[19] M. Chen, G. Chen, S. Jiang, Y. Lee, R. Despres, and R. Penno, “IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd) draft-ietf-softwire-4rd-08,” Internet Draft, April 2014.

[20] P. Wu, Y. Cui, J. Wu, J. Liu, and C. Metz, “Transition from IPv4 to IPv6: A State-of-the-Art Survey,” IEEE Commun. Surv. Tutor., vol. 15, no. 3, pp. 1407–1424, Third 2013.

[21] J. Arkko and F. Baker, “Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment,” RFC 6180, May 2011.

[22] J. Wu, Y. Cui, X. Li, and C. Metz, “The Transition to IPv6, Part 1: 4over6 for the China Education and Research Network,” IEEE Internet Comput., vol. 10, no. 3, pp. 80–85, May 2006.

[23] Y. Cui, J. Wu, X. Li, M. Xu, and C. Metz, “The Transition to IPv6, Part II: The Softwire Mesh Framework Solution,” IEEE Internet Comput., vol. 10, no. 5, pp. 76–80, Sep. 2006.

[24] M. Chen, X. Li, A. Li, and Y. Cui, “Forwarding IPv4 Traffics in Pure IPv6 Backbone with Stateless Address Mapping,” in Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, 2006, pp. 260–270.

[25] Y. Cui, J. Dong, P. Wu, J. Wu, C. Metz, Y. L. Lee, and A. Durand, “Tunnel-Based IPv6 Transition,” IEEE Internet Comput., vol. 17, no. 2, pp. 62–68, Mar. 2013.

[26] L. Gong, H. Le, and R. Yu, “Analyses on IPv6 Evolution Technologies,” J. Comput. Inf. Syst., vol. 8, pp. 5859–5865, Jul. 2012.

[27] R. AlJa’afreh, J. Mellor, and I. Awan, “A Comparison Between the Tunneling Process and Mapping Schemes for IPv4/IPv6 Transition,” in International Conference on Advanced Information Networking and Applications Workshops, 2009. WAINA ’09, 2009, pp. 601–606.

[28] P. Grayeli, S. Sarkani, and T. Mazzuchi, “Performance Analysis of IPv6 Transition Mechanisms over MPLS,” Int. J. Commun. Netw. Inf. Secur. IJCNIS, vol. 4, no. 2, Aug. 2012.

[29] D. S. Punithavathani and K. Sankaranarayanan, “IPv4/IPv6 Transition Mechanisms,” Eur. J. Sci. Res., vol. 34, no. 1, pp. 110–124, Jul. 2009.

[30] K. Kinnear and R. Woundy, “Dynamic Host Configuration Protocol (DHCP) Leasequery,” RFC 4388, February 2006.

[31] R. Johnson, K. Kinnear, and M. Stapp, “Description of Cisco Systems’ Subnet Allocation Option for DHCPv4,” RFC 6656, July 2012.

[32] “Broadband Access in the 21st Century: Applications, Services, and Technologies,” Cisco. [Online]. Available: http://cisco.com/c/en/us/solutions /collateral/service-provider/service-provider-strategy/white_paper_c11-690395.html. [Accessed: 11-Feb-2014].

# ภาคผนวก

**ภาคผนวก ก**
**ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์**


ICOIN 2014
Sponsored by KIISE Information Networking Society
Technically Co-sponsored by IEEE Computer Society and IEICE Communication Society
Korean Institute of Information Scientists and Engineers
IEEE
IEEE computer society
IEICE

# Performance Evaluation of IPv4/IPv6 Transition Mechanisms: IPv4-in-IPv6 Tunneling Techniques

N. Chuangchunsong
Dep. of Computer Engineering
Prince of Songkla University
Songkla, Thailand
napat.chu@gmail.com

S. Kamolphiwong, T. Kamolphiwong and R. Elz
Dep. of Computer Engineering
Prince of Songkla University
Songkla, Thailand
{ksinchai, kthossaporn, kre}@coe.psu.ac.th

P. Pongpaibool
NECTEC, NSTDA
Bangkok, Thailand
panita@nectec.or.th

***Abstract*—Exhaustion of IPv4 address space is highly aware for most internet players, not only Internet Service Providers (ISPs), but also Telco and Content Providers. A number of IPv4/IPv6 migration/transition tools and mechanisms have been proposed, deployed/implemented world-wide. To make IPv4 networks be able to connect to IPv6 world, 4over6, DS-lite, and 4rd seems to be the most attractive solution according to their features and functions benefits. Beside such benefits, in this paper, we investigate their performance in terms of delay time, and reliability in both inter and intra-communications. Comparison results and analysis of these 3 solutions will be given. We conclude that these figures will provide the factors of scalability and quality-of-service (QoS).**

***Keywords—IPv6, DS-Lite, 4over6, 4rd, IPv4/IPv6 transition***

## I. INTRODUCTION

Exhaustion of IPv4 address space is now highly aware by all the Internet players. At least two regional regions; Asia and Europe, have run of IPv4 addresses, while the rest are following (except Africa Region). A number of transitions and migrations tools and mechanisms have been proposed and implemented, e.g. transitions done by IETF [1]. Most of them have pros and cons depending on their usage proposes. The most limited factor on IPv4 and IPv6 transition is the IPv6 incompatible with IPv4.

Demanding of transition mechanisms will go on from now until the changing from IPv4 to IPv6 completed. The changing state can be divided into 3 phases as follows: Phase I, IPv6 is an island in IPv4 ocean, where IPv4 still dominates on the global networking. Phase II, after some years later, IPv4 become an island while IPv6 will be ocean. This means that in this stage, IPv6 is much bigger than IPv4. The final phase, Phase III, most of networks are in IPv6 native. Today, a few percentage of IPv6 traffic has been seen, e.g. 2% of IPv6 traffic seen by Google. We expect that within next 3-5 years, IPv6 traffic will increase to between 30-50%. So, we do need some solutions to work on for next 3-5 years. This means that we will keep IPv4 networks running on IPv6 networks where IPv4 is a small portion of the global network. Recently, some well-known mechanisms are considered to be the solutions; they are: 4over6 [2], DS-lite [3] and 4rd [4]. Most technical papers proposed and investigated there features and functions. In this paper, we investigate transitions mechanism using "Tunneling Technique", which is deployed in [2], [3], and [4]. We will present simulation results on performance of such transition mechanism in terms of processing delay/overhead, and reliability. These performance considered will affect their scalability and quality of service (QoS).

This paper is organised as follows: In section 2, transition mechanism of IPv4-in-IPv6 tunneling will be given. Simulation scenarios are described in Section 3. Simulation results, comparisons and analysis are presented in Section 4. We conclude our paper in the last section.

## II. IPV4-IN-IPV6 TUNNELING TRANSITION

In the beginning of transition development, tunneling technique was intended to extend IPv6 connectivity through IPv4 backbone network. This means that IPv6 networks were both ends. Later, however, this technique then is designed to support IPv4 networks at both ends while IPv6 is a backbone network. Some improvements have been developed. By using this mechanism, all IPv6 hosts can connect to IPv4 destination hosts, and IPv4 addresses can be allocated effectively [5],[6].

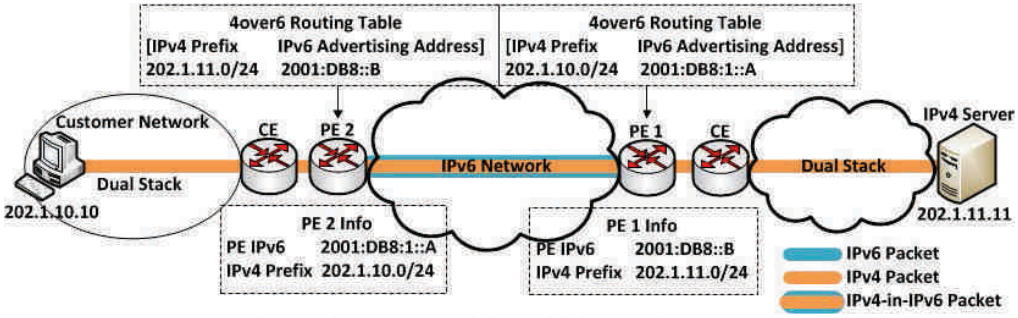



Fig. 1. The IPv4 connectivity establishment using 4over6.

### A. 4over6

4over6 is one of the IPv4-in-IPv6 tunneling mechanisms, which is taken to further develop in order to apply in various networks [7],[8],[9],[10]. The 4over6 is taken into account for a consideration of its flexibility in deployment. IPv4 and IPv6 addresses in the mechanism are not necessarily correlated. Moreover, if the IPv4 address or IPv6 address prefix in end nodes are changed, it does not affect the tunnel-endpoint specification. Therefore, the most advantage of 4over6 deployment is it keeps the existing network without any changes.

In this mechanism, communication tunnel is created between provider's gateways which are so called "Provider Edge (PE)". In the tunnel-endpoint specification of PE, it uses 2 information types which consist of IPv4 prefix and IPv6 address. It is similar to the information of general routing table. However, if Next Hop is a different IP version using its original IP address, PE will encapsulate IPv4 packet into IPv6 packet and specify tunnel-endpoint with the PE's IPv6 address of the destination network. To update 4over6 routing table, 4over6 adds new extension of Multiprotocol Extensions for BGP (MP-BGP) which is designed to carry routing information for multiple network layer protocols. A sample of IPv4 connectivity establishment using 4over6 is shown in Fig. 1.

### B. Dual Stack lite

Dual-Stack lite or DS-lite, is another mechanism of IPv4-in-IPv6 tunneling transition. The tunnel is built to connect between customer's gateway and provider's equipment through the service provider's network. The customer's gateway is called Basic Bridging BroadBand (B4) and the provider's equipment is called Address Family Transition Router (AFTR).

When B4 encapsulates the IPv4 packet and send it to the AFTR, AFTR then de-encapsulates to retrieve IPv4 packet. IPv4 address of the packet is private IPv4 address. It must perform NAT before sent to it to the Internet. Nevertheless, the NAT operation of DS-lite cannot perform by only using information of IPv4 source address + port and outside IPv4 source address + port because DS-lite, can allocate any private IPv4 addresses independently. The private IPv4 address of each customer may be overlap and cannot trackback to the right IPv4 address. Therefore, the NAT operation of DS-lite requires a recording of additional information which is the B4's IPv6 addresses of customer, for identify each private IPv4 address. The IPv4 connectivity establishment by using DS-lite is shown in Fig. 2.

### C. 4rd

4rd is automatic tunnel mechanism in order to distribute the remaining IPv4 addresses to the customer's network through IPv6 network. Customer's network obtains IPv4 addresses to complete Dual Stack transition. 4rd is designed for using to the remain IPv4 addresses which each ranges of remaining IPv4 address may not be equal and continuous hence it is support more than one rule in 4rd domain. Each rule is mapped between the remaining IPv4 addresses and IPv6 addresses appropriately. This technique can allocate the ranges of remaining IPv4 addresses effectively and can be used with difference formats of IPv4 address, such as public IPv4 prefix (IPv4 subnet), public IPv4 address, shared public IPv4 with port-set, private IPv4 address and no IPv4 address by NAT64+. The IPv4 connectivity establishment using 4rd is shown in Fig. 3.

4rd deployment can be classified by NAT format into 2 scenarios which are 4rd:NAT Distribution and 4rd:NAT Centralization. For 4rd:NAT Distribution, it provides the public IPv4 address for each Customer Edge (CE) directly. CE also supports NAT44 for translating IPv4 packet before encapsulating into IPv6 packet, and sent to the Border Relay (BR). A number of users within the network can support depends on ability of NAT performance and management on the CE. The limitation of this scenario will occur when increase amount CEs because the public IPv4 address on each CE must be re-allocated. In this case, 4rd is deployed by including public IPv4 prefix (IPv4 subnet), public IPv4 address, and shared public IPv4 with port-set. For 4rd:NAT Centralization, it does not provides the public IPv4 address for each Customer Edge (CE). BR also supports NAT44 for translating IPv4 packet. A tunnel is established as 4rd:NAT Distribution scenario. It just changes all 4rd rules of public IPv4 address to private IPv4 address. In this scenario, public IPv4 address is assigned to only one equipment on provider network. When a number of Customer's network increase, old Customer's network are not affected, and it does not require to re-allocate public IPv4 address as 4rd:NAT Distribution. In this case, 4rd deployment uses private IPv4 address.

## III. SIMULATION SCENARIOS

Our simulation was performed by using OPNET. In our simulation scenarios, we have native IPv6 links between 4 routers on backbone network. Four transition scenarios are used: 4over6, DS-lite, 4rd:NAT Centralization, and 4rd:NAT

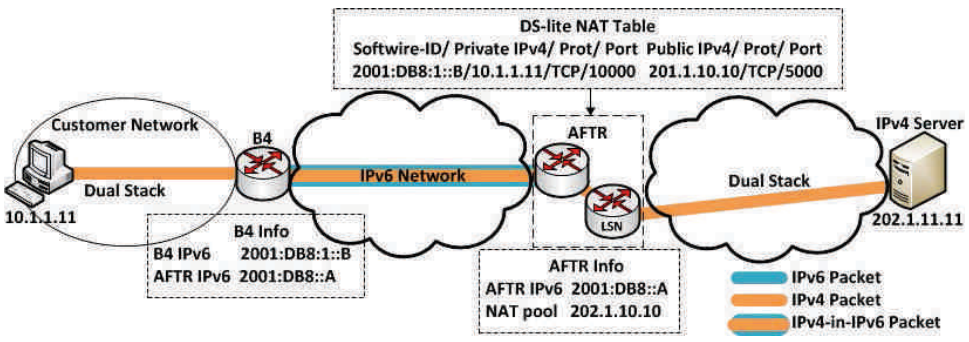



Fig. 2. The IPv4 connectivity establishment using DS-lite.

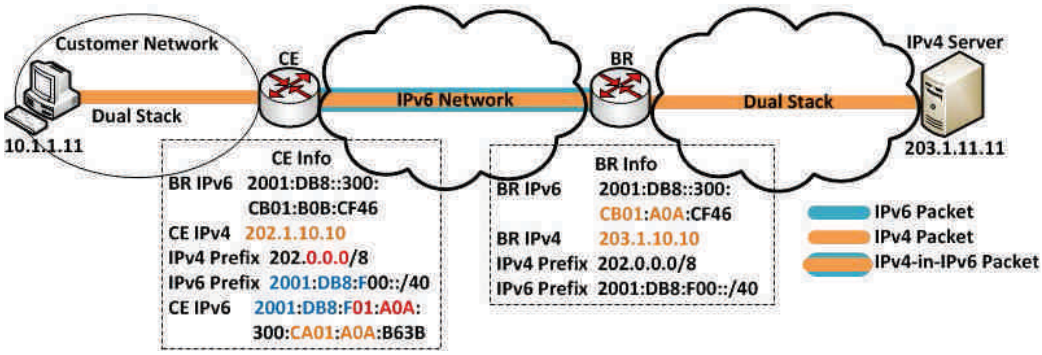



Fig. 3. The IPv4 connectivity establishment using 4rd.

Distribution. Each router provides to support all IPv4-in-IPv6 tunneling transitions by modify IP routing process in IP module. The routers connect together and connect with 3 sub-networks: student subnet, teacher subnet and local server. The application profiles of student and teacher subnets are FTP, HTTP, E-mail and Remote Login. A simulation scenario is shown in Fig. 4.

The following performance parameters are used in our simulations:

IP processing Information of Router

| | |
|---|---|
| Processing scheme | : Central Processing |
| Datagram Switching Rate | : 500,000 packet/sec |
| Datagram Forwarding Rate | : 50,000 packet/sec |
| Memory Size | : 16 MB |

TCP Parameter

| | |
|---|---|
| Receive Buffer | : 8760 bytes |
| Maximum ACK delay | : 0.2 sec |
| Maximum ACK segment | : 2 |
| Slow-Start initial count (MSS) | : 2 |
| Fast Retransmit | : Enable |
| Fast Recovery | : Reno |
| Duplicate ACK threshold | : 3 |
| Initial RTO | : 3.0 sec |
| Minimum RTO | : 1.0 sec |
| Maximum RTO | : 64.0 sec |

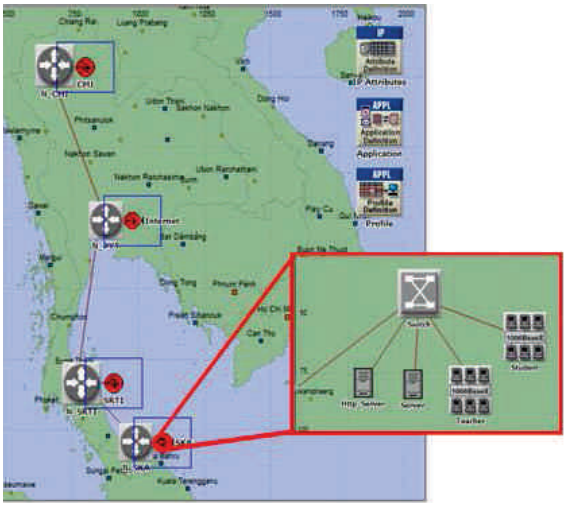


Fig. 4. The simulation scenario.

## IV. Simulation Results and Analysis

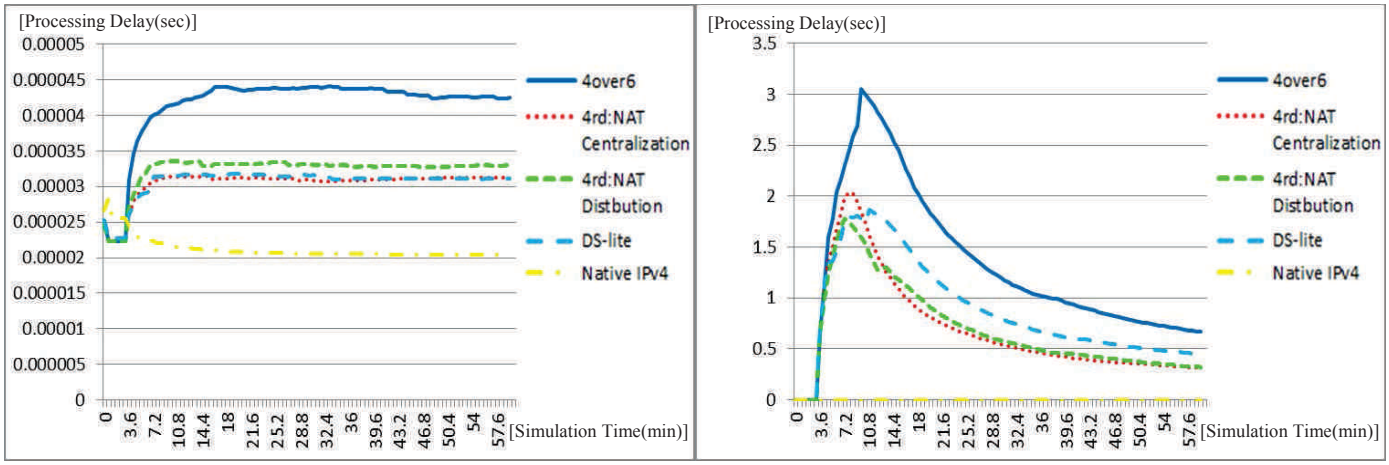
In this section, we will present the simulation results by comparing all 4 transition scenarios. We investigate on performance and reliability in both inter-communication and intra-communication and then we analyze the complexity in the last issue.

### A. Result on Performance

Performance analysis of IPv4-in-IPv6 tunneling transition focus on the time for establishing a communication tunnel. We divided the test into 2 communication types: Inter-communication and Intra-communication.

*1) Inter-communication:* Fig. 5(a) shows processing delay on equipment for establishing a communication tunnel. The equipment, which establishes the communication tunnel for all traffic transition on customer's network, is called Customer Edge (CE). The equipment, which establishes the communication tunnel on provider's network, is called Provider Relay (PR). In this result, the transition which has the highest processing delay on CE is 4over6 because CE of 4over6 must performs NAT and stateful tunneling by using special routing table. The transition which has lower processing delay is 4rd:NAT Distribution. Although 4rd:NAT Distribution performs NAT as 4over6, but the tunnel-endpoint specification of 4rd:NAT Distribution is stateless. For 4rd:NAT Centralization, the tunnel-endpoint specification is also stateless and does not perform NAT on CE. So, 4rd:NAT Centralization has less processing delay than the CE 4rd:NAT Distribution slightly. Transition which has lowest processing delay is DS-lite because it does not perform NAT and tunnel-endpoint on CE of DS-lite always pre-specify to PR. It also does not waste the time for tunnel-endpoint specification.

Fig. 5(b) shows a trend of processing delays of each transition technique. We can see that processing delays are variation because they are the summation of the queue delay and service time. If the packets in queue increase, processing delay also increases. Transition which has the highest processing delay is 4over6. Although, it does not perform NAT on PR, but it specify the tunnel-endpoint by using special routing table which is still time-consuming. Transition which has lower processing delay is DS-lite because




(a) (b)

Fig. 5. Processing delay of inter-communication (a) at Customer Edge (b) at Provider Relay.

DS-lite uses the special NAT table for performed NAT and specified tunnel-endpoint simultaneously. Transitions which have the lowest processing delay are 4rd:NAT Distrubution and 4rd:NAT Centralization because the tunnel-endpoint specification of 4rd is stateless. However, with carefully analysis, we found that at the beginning of the simulation, all hosts in the network have high transmission rates. 4rd:NAT Centralization which perform NAT on PR has a higher Processing delay. Later, if host has high TCP delay, host will decrease the transmission rate as soon as TCP delay increase. Queue size of the other transitions, except 4rd:NAT Distribution also decrease rapidly. Therefore, Processing delay of 4rd:NAT Centralization is less than 4rd:NAT Distribution slightly, as shown in Fig. 5(b).

In Fig. 6, FTP Download Response Time illustrates delay on a real application in the network. The result of FTP Download Response Time is consistent with the processing delay on PR because processing delay on PR is the highest delay when compared with other delays.

*2) Intra-communication:* From the result shown in Fig. 7 (a), Processing delay on CE of intra-communication is similar to processing delay on CE of inter-communication as shown Fig. 5(a). A different result is processing delay of 4over6 because in this communication type, the destination locates within same transition domain. 4over6 does not search for specified tunnel-endpoint in the routing table until the last record. Therefore, processing delay of 4over6 has decrease.

In Fig. 7(b), 4over6 and 4rd (in both NAT distribution and NAT centralization) have a processing delay on PR similar to Native IPv4 because it can specify tunnel-endpoint to CE of destination network directly, unlike the DS-lite that necessary to send packet to the PR first. Therefore, DS-lite has the highest processing delay on PR.

In Fig. 8, transition which has the highest FTP Download Response Time is DS-lite because it cannot send packets to the destination network directly. It always sends packets to PR to perform NAT first. Whereas, FTP Download Response Time of 4rd:NAT Distribution, 4over6 and 4rd:NAT Centralization are less delay respectively. The results of 3 transitions have slightly difference and consistent with the processing delay on CE as shown in Fig. 7(a).

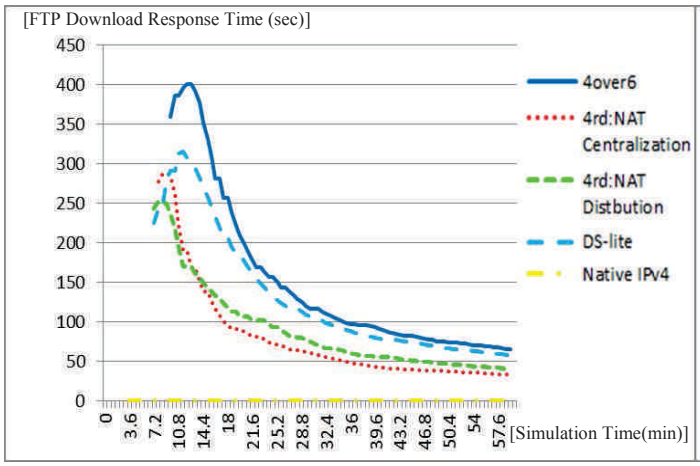



Fig. 6. FTP Download Response Time of inter-communication.

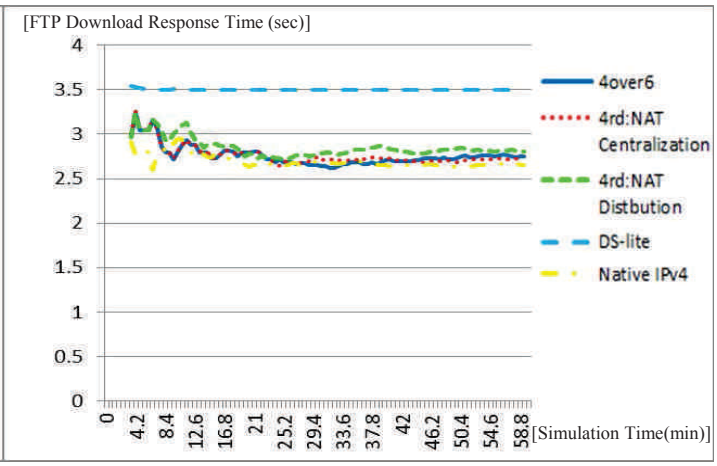
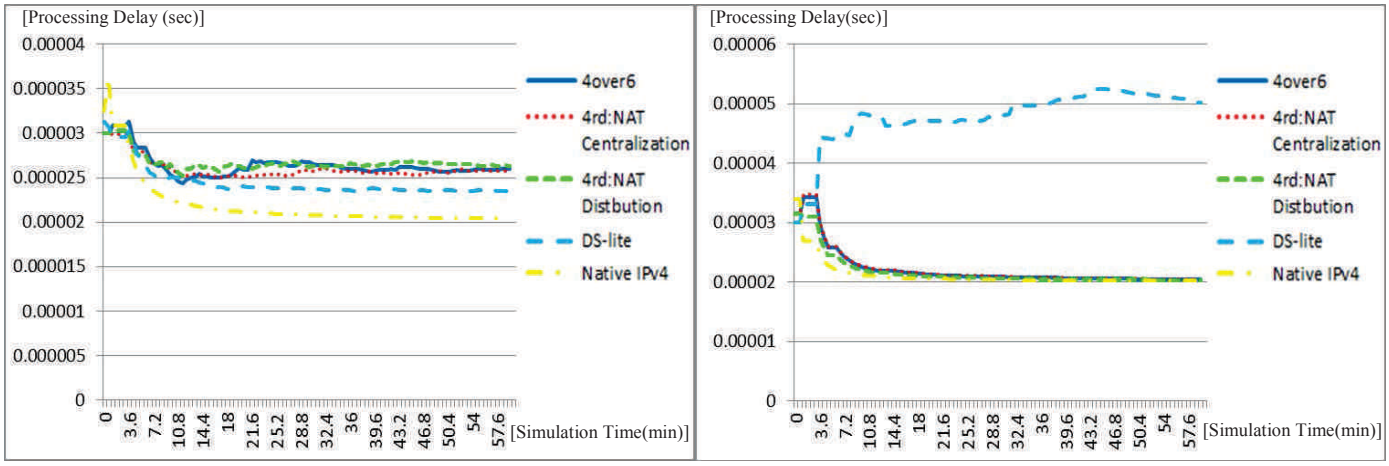



Fig. 8. FTP Download Response Time of intra-communication.




(a) (b)

Fig. 7. Processing delay of intra-communication (a) at Customer Edge (b) at Provider Relay.

*B. Result on Reliability*

Reliability is the most important factor for transition selection criterions. We use a number of retransmission data to measure reliability, and provide analysis trend of retransmission when network has customers growing. We provide 5 sub-scenarios of each transition by using percentage of customer's required per available port. The retransmission results of each transition are shown below:

*1) Inter-communication:* In Fig. 9, when the network has small percentage of customer's required port per available port, the retransmission of each transition is slightly different. When, the network has a large percentage of customer's required port per available port, it illustrates a distinction clearly. Transition that has the lowest number of retransmission is 4rd:NAT Centralization, DS-lite, 4over6 and 4rd:NAT Distribution respectively. 4rd:NAT Centralization and DS-lite have lower number of retransmission because they are NAT centralization mechanism. So, port utilisation is higher than other transition mechanisms. When comparing between 4rd:NAT Centralization and DS-lite, DS-lite has higher retransmission than 4rd:NAT Centralization because port allocation process of DS-lite has more complexity. Whereas, 4rd:NAT Distribution has the highest number of retransmission because it is NAT distribution mechanism, and gives very low End-to-End delay. Retransmission time is also low as well.

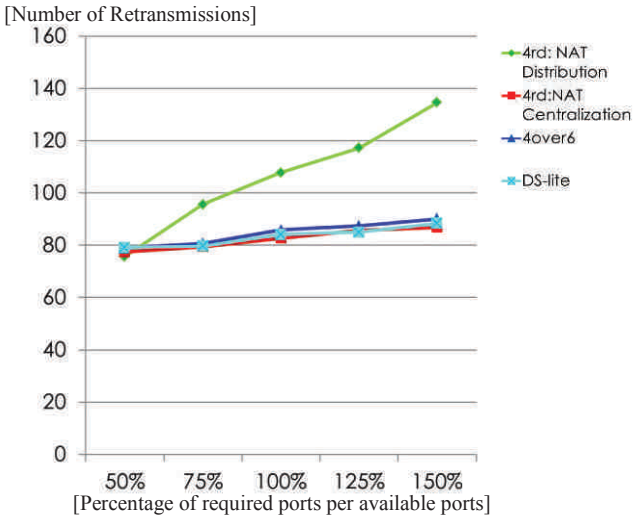



Fig. 9. Number of retransmission for inter-communication of each transition when network has customers increasingly.

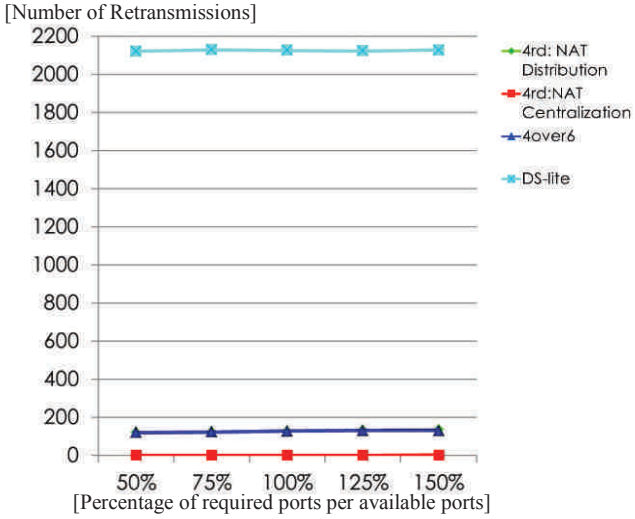



Fig. 10. Number of retransmission for intra-communication of each transition when network has customers increasingly.

*2) Intra-communication:* In Fig. 10, number of retransmissions for intra-connection of each transition are significantly difference. 4rd:NAT Centralization rarely retransmit because it contacts within transition domain by using private IPv4 address directly. On the contrary, 4over6 and 4rd:NAT Centralization have higher number of retransmissions. Since, they are NAT distribution. Whereas, DS-lite has highest number of retransmissions because it must send packets to the PR first. It also causes bottleneck problem on PR.

### *C. complexity*

The operation of each transition is not similarity. They also have a different complexity. When the transitions have a more complexity, the compatibility of transition is decrease. For 4over6, 4over6 perform the tunnel-endpoint specification with the special routing table, which use both IPv4 and IPv6 information together. The new MP-BGP extension is designed to be used for exchange the information of special routing table. 4over6 can update the special routing table quickly when network topology changes. However 4over6 require the routing information exchange regularly. The CE and PR equipment in 4over6 must be improved to support the tunnel-endpoint specification with the special routing table and support the new extension of the MP-BGP. Therefore, 4over6 has high complexity and low compatibility. For 4rd, 4rd perform the tunnel-endpoint specification with 4rd rule. 4rd rule is the translation rule between IPv4 address and IPv6 address, which translate the IPv6 address of the destination tunnel-endpoint immediately by using only IPv4 address and 4rd rule. CE equipment can obtain the 4rd rule information through pre-configuration or DHCPv6. So, the minimum function of PR and CE equipment in 4rd is the tunnel-endpoint specification by 4rd rule and then both PR and CE equipment must be assigned IPv4 address and IPv6 address in accordance with the 4rd rule strictly. Accordingly, 4rd has high complexity and low compatibility. For DS-lite, DS-lite is only transition that is incompatible with mesh connectivity. DS-lite also has low effective in intra-communication. However, the incompatibility of mesh connectivity has the advantage because CE equipment require only one static tunnel-endpoint to PR equipment. DS-lite also has high compatibility. The tunnel-endpoint to CE equipment is recoded by PR equipment. PR uses the special NAT table to recode both NAT session and CE destination simultaneously. Therefore, if CE already supports static IPv4-in-IPv6 Tunneling, ISP can improve only PR equipment in order to provide DS-lite transition.

## V. CONCLUSIONS

In this paper, we present an investigation of performance evaluation of IPv4-in-IPv6 tunneling techniques. Four transition scenarios have been evaluated: 4over6, DS-lite, 4rd:NAT Centralization, and 4rd:NAT Distribution. Two issues are investigated: performance, and reliability, in both communication types: Inter-communication and Intra-communication.

Each transition mechanism has different pros and cons, need to understand their features and functions clearly in order to select the transition to serve on their network appropriately. This paper provides a focus on performance evaluation of such transition mechanism.

In our simulation results, we have found that 4rd:NAT Centralization has the high performance and high reliability. In addition, it has no limitation for communication types. However, it lacks of flexibility in IP address allocation. Moreover, private IPv4 address may not be sufficient, if a network requires to serve a large number of customers. As a result this transition mechanism may be deployed in a small network. For 4rd:NAT Distribution, it gives a high performance and relatively high reliability. It also has no limitation about communication types. However, it lacks of flexibility in IP address allocation similar to 4rd:NAT Centralization. Nevertheless, the inflexibility and new tunnel-endpoint specification of 4rd cause high complexity and low compatibility.

On the other hand, 4over6 has relatively high reliability and also has a high flexibility in IP address allocation. Moreover, it has no limitation of communication types but it has lower performance compared to other transition mechanisms. However, 4over6 require new MB-BGP extension and new tunnel-endpoint specification. It also has high complexity and low compatibility.

Whereas, DS-lite has high flexibility but it has relatively high performance and reliability only on inter-communication. For intra-communication, The DS-lite has low performance and low reliability because it cannot communicate to destination within same transition domain directly. However, the static tunneling of CE in DS-lite causes low complexity and high compatibility.

## REFERENCES


[1] R. Gilligan, E. Nordmark, Basic Transition Mechanisms for IPv6 Hosts and Routers, RFC 4213, October 2005

[2] J. Wu, Y. Cui, X. Li, M. Xu, and C. Metz, “4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions” 2011, IETF RFC5747.

[3] A. Durand, R. Droms, J. Woodyatt, and Y. Lee, “Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion,” 2011, IETF RFC6333.

[4] R. Despres and R. Penno and Y. Lee and G. Chen and S. Jiang , “IPv4 Residual Deployment via IPv6 – a unified Stateless Solution (4rd),” 2013, IETF draft.

[5] J. Arkko and F. Baker, “Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment.” 2011, IETF RFC6180.

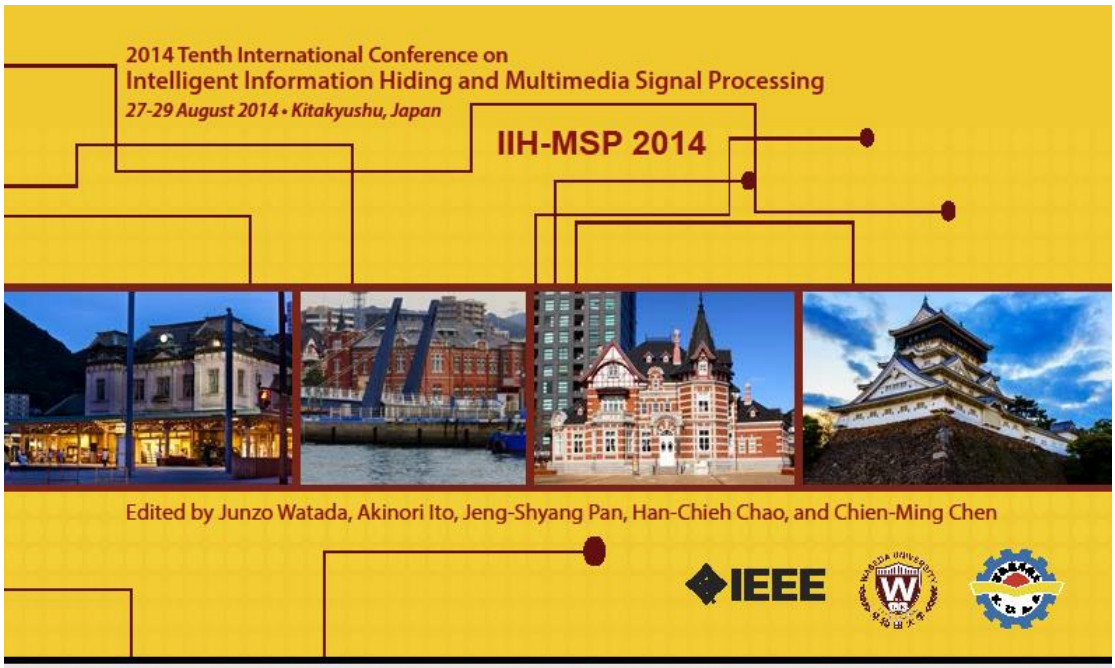
[6] P. Wu, Y. Cui, J. Wu, J. Liu, and C. Metz, “Transition from IPv4 to IPv6: A State-of-the-Art Survey,” IEEE Communications Surveys Tutorials, vol. PP, no. 99, pp. 1 –18, 2012.

[7] J. Wu et aI., "The Transition to IPv6, Part I: 4over6 for the China Education and Research Network," IEEE Internet Computing, May 2006.

[8] Y. Cui et aI., "The Transition to IPv6, Part II: The Softwire Mesh Framework Solution," IEEE Internet Computing, Sept/Oct 2006.

[9] Y. Cui, J. Wu, P. Wu, C. Metz, O. Vautrin, and Y. Lee, “Public IPv4 over Access IPv6 Network,” 2011, IETF draft.

[10] Y. Cui, P. Wu, M. Xu, J. Wu, Y. L. Lee, A. Durand, and C. Metz, “4over6: network layer virtualization for IPv4-IPv6 coexistence,” IEEE Network, vol. 26, no. 5, pp. 44 –48, Oct. 2012.


2014 Tenth International Conference on
Intelligent Information Hiding and Multimedia Signal Processing
27-29 August 2014 • Kitakyushu, Japan
IIH-MSP 2014
Edited by Junzo Watada, Akinori Ito, Jeng-Shyang Pan, Han-Chieh Chao, and Chien-Ming Chen
IEEE
CONFERENCE INFORMATION
PAPERS BY SESSION
PAPERS BY AUTHOR
GETTING STARTED
TRADEMARKS
SEARCH

2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

# An Enhancement of IPv4-in-IPv6 Mechanism

N. Chuangchunsong
Dep. of computer engineering
Prince of Songkla University
Songkla, Thailand
napat.chu@gmail.com

S. Kamolphiwong
Dep. of computer engineering
Prince of Songkla University
Songkla, Thailand
ksinchai@coe.psu.ac.th

T. Kamolphiwong
Dep. of computer engineering
Prince of Songkla University
Songkla, Thailand
kthossaporn@coe.psu.ac.th

R. Elz
Dep. of computer engineering
Prince of Songkla University
Songkla, Thailand
kre@coe.psu.ac.th

***Abstract*—From global Internet information, IPv6 traffic grows slowly. A large portion of service providers are still using IPv4 even IPv6 co-exists with IPv4 for long time. However, IPv6 traffic are gradually growing especially most of network backbone sides. This paper proposes an enhancement mechanism of IPv4-in-IPv6 which supports IPv6 natively and provides IPv4 connectivity by using IPv4-in-IPv6 tunneling, so called "Lightweight 4over6 (lw4over6)". The performance metrics used in this evaluation are: system throughput and packet dropped. Based on our computer simulation results, when the system has a mix of intra and inter-communications, we have found that; with light traffic load, the proposed method performs better than lw4over6 and 4over6 slightly. However, with more heavy traffic load the proposed method gives higher performance than lw4over6 significantly.**

***Keywords-IPv6; IPv4-over-IPv6; lw4over6; IPv6 migration***

## I. INTRODUCTION

The demand for IPv4 slightly reduced even IANA runs out of IPv4 address space. Demonstrated by percentages of IPv6 traffic per the all traffic in the last year [1], it increased doubly from 1.18% to 2.80%. The IPv6 traffic also increase only 1.68% in this year. When we analyze the adoption of IPv6 traffic in each country, there are some countries that have increased the IPv6 traffic explicitly, most are in EU and US. Moreover, the analysts believe that IPv4 and IPv6 co-exist for a long time [2]. There are 2 main reasons. The first is as IPv4 and IPv6 can operate over shared network by using transition. The other is as re-using IPv4 addresses is still widely deployed. Therefore, changing from IPv4 to IPv6 does not cutoff immediately.

Our previous work analysed the performance of IPv4/IPv6 transitions by using network simulation models [3]. Those IPv4-in-IPv6 tunneling transitions were 4rd, Dual Stack lite and 4over6. The simulation results demonstrated that 4rd has the highest performance of IPv4 connectivity both inter-communication and intra-communication, but it still lacks of flexibility in IP address allocation. Since 4rd performs stateless operation, IPv4 and IPv6 address must be allocated based on 4rd rules strictly. If 4rd's network reallocate either IPv4 or IPv6 address, it will affect the allocation of another IP version immediately. The secondary of analysed transition is 4over6. Performance of 4over6 less than 4rd because 4over6 performs stateful operation. 4over6 update the IPv4 and IPv6 over MP-BGP. The advantage of tunnel-endpoint maintenance by using routing protocol is always current information that is not affected by changing of IPv4 or IPv6 address. However 4over6 cannot share public IPv4 address between multiple customer networks because 4over6 cannot route single IPv4 address to multiple destination. This cause may affect IPv4 address shortage for expanded network in the future. The last one is DS-lite. DS-lite performs stateful operation as 4over6. But tunnel-endpoint maintenance of DS-lite relies on special NAT table which records both the IPv6 address of tunnel-endpoint and NAT parameters concurrently. When NAT equipment of providers that support DS-lite operation receive the encapsulated packet from customer gateway, it decapsulates and records NAT parameters and IPv6 addresses of customer gateway before forwarding the inside packet to destination. Therefore, when the network has higher intra-communication, performance of DS-lite is likely to decrease.

The results of previous papers demonstrated the advantages and limitations of each transition technique. Transition that can provide both IPv4 and IPv6 in long-term should has the following features: 1) facilitate the ISP core network to support IPv6 natively, 2) allocate IPv4 and IPv6 address independently, 3) share IPv4 address with restrict port-set dynamically and 4) support mesh IPv4 connectivity.

The rest of this paper is organised as follows: Section II will describe IPv4-in-IPv6 mechanisms used in this paper. Section III will give our proposed solution, how it works. Next, all simulation scenarios will be described. In section V, the simulation results and analysis will be discussed with comparison between our proposed solution and others. We conclude our paper in the last section.

## II. TRANSITION MECHANISMS

### A. 4over6

4over6 is one of IPv4-in-IPv6 tunneling. Tunnel-endpoint information is updated automatically by IPv4 and IPv6 information exchange via routing protocols [4],[5]. When IPv4 or IPv6 address changes, 4over6 can update tunnel-endpoint information properly after updating routing information. 4over6 apply Multiprotocol Extensions for BGP (MP-BGP) to exchange the routing information. Routing information of tunnel-endpoint consist of destination IPv4 prefix and Next Hop's IPv6 address. Equipment that supports IPv4-in-IPv6 tunneling in 4over6 is called "Provider Edge (PE)". When PE receives IPv4 packet that toward the destination IPv4 prefix in tunnel-endpoint information, the IPv4 packet is encapsulated into an IPv6 packet and specify the IPv6 destination depending on the Next Hop's IPv6 address.

978-1-4799-5390-5/14 $31.00 © 2014 IEEE
DOI 10.1109/IIH-MSP.2014.18

### B. *Lightweight 4over6*

Lightweight 4over6 or lw4over6 is developed as extension of DS-lite [6]. The lw4over6 is designed to reduce the impact of NAT centralization in DS-lite. The lw4over6 distributes public IPv4 address and port set to the customer's network. NAT operation of lw4over6 also is distributed to the gateway of customer's network. Name of lw4over6 equipment is similar to the DS-lite equipment. So, the customer's gateway is called Lightweight Basic Bridging BroadBand (lwB4) and the provider's equipment is called Lightweight Address Family Transition Router (lwAFTR).

To support NAT distribution, lwAFTR of lw4over6 maintains only 3 information types such as public IPv4 address, port-set and lwB4's IPv6 address. On the other hand, lwB4 requires addition information not only lwAFTR's IPv6 address but also public IPv4 address and port-set in order to perform NAT functions. To allocate public IPv4 address and port-set, lwAFTR does not allocate public IPv4 address directly. The lw4over6 defines DHCP 4o6 server to serve this allocation particularly [7],[8],[9]. Therefore, lwAFTR must track all public IPv4 addresses and port-set which DHCP 4o6 server allocates to lwB4.

## III. PROPOSED METHOD

Base on consideration results of exist transitions and features of long-term support transition, major features of long-term support transition is similar to the lw4over6 features. Since lw4over6 provides native IPv6 connectivity, it can allocate IPv4 and IPv6 addresses independently and support share public IPv4 address with restrict port-set dynamically. Lack feature of lw4over6 is mesh IPv4 connectivity. Consequently, this paper proposes enhancement of lw4over6 that is improved to support mesh IPv4 connectivity in order to increase performance of IPv4.

### A. *Implementation*

Proposed method applies DHCPv4 over DHCPv6 which defines for exchange IPv4 information between lwB4 and DHCP 4o6 server in lw4over6 to exchange tunnel-endpoint information. The exchanged tunnel-endpoint information includes public IPv4 address, port-set and lwB4's IPv6 address. IPv4 connectivity establishment of proposed method is shown in Fig. 1.

To query tunnel-endpoint information, lwB4 source just knows only public IPv4 address and port of the destination. So, the appropriate queried method is DHCPv4 leasequery because it is designed to query the required information between DHCPv4 Relay and DHCPv4 Server by using IPv4 address or MAC address [10],[11]. The queried method in failover system of lw4over6 is also planned to use DHCPv4 leasequery. Although, the failover system does not specify the details [12]. DHCPv4 leasequery supports query method by IPv4 address only, port excluded. Therefore, proposed method has been defined DHCPv4 message and option to support new DHCPv4 leasequery and addition information.

The DHCPv4 Leasequery message in proposed method has been defined to query by IPv4 address and port that is applied from query by IPv4 address. New DHCPv4 leasequery message defines the query format of IPv4 address and port as the htype field must be set to 0, the hlen field must be set to 2 (size of port), the 2 bytes at the beginning of chaddr fiend is the port value and the remaining bytes must be set to 0 (padding), the ciaddr field must set to the required IPv4 address and Client-identifier option must not set.

The query by the IPv4 address and port is divided into two query types: specific query and all query. For specific query, IPv4 address and port is set as IPv4 address and port of the required host. The returned information is only the information of the specified host. For all query, IPv4 address and port is set to 0. The returned information is recorded by the replied DHCP server. All query is designed for failover system to recover all the information at once.

The information of lwB4 identity and required tunnel-endpoint which are called "lw4over6_ID" and "lw4over6 _lwB4_Information" respectively are exchanged by sub-option of the DHCPv4 Relay Agent Information option (option 82). The lw4over6_ID sub-option is used to specify identity of lwB4 requester. Information inside sub-option lw4over6_ID consists of lwB4's IPv4 address, IPv4 port-set and lwB4's IPv6 address. lw4over6_lwB4_Information sub-option is used to reply the information of requested lwB4 from DHCP 4o6 Server to lwB4. Information inside lw4over6_lwB4_Information is similar to lw4over6_ID that consists of lwB4's IPv4 address, IPv4 port-set, lwB4's IPv6 address and lease time.

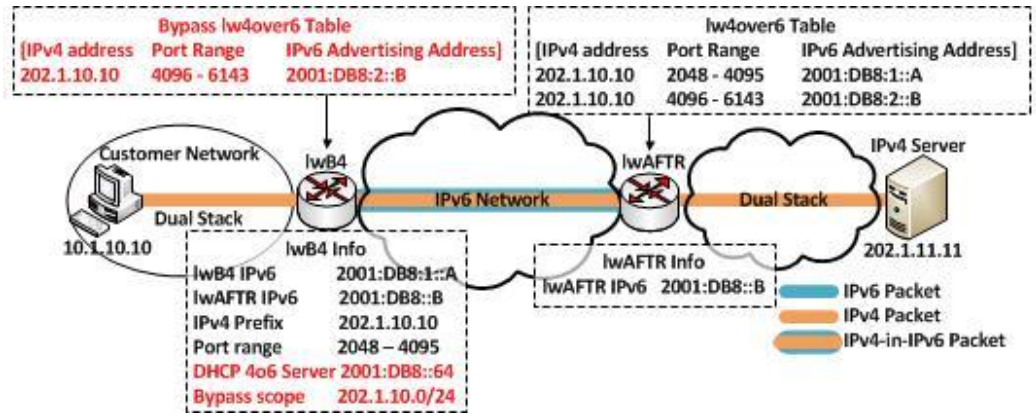



Figure 1. The IPv4 connectivity establishment of proposed method.

Moreover, proposed method required Bypass Scope information that can communicate to destination network directly. Proposed method defines new DHCPv4 option ADDRESS_POOL to declare the Bypass Scope. The Option-value of BYPASS_SCOPE is IPv4 address pools which consist of the couple of start IPv4 address and end IPv4 address. The option-len also is equal to multiple of 8 bytes.

### *B. The lwB4 Behaviour*

The lwB4 in proposed method has more the additional operations than lwB4 in lw4over6. Since, lwB4 in proposed method requires to maintain tunnel-endpoint information in order to communication with destination network directly. New lwB4 behaviour is detail in the following.

Main behaviour of lwB4 in proposed method divided into two events which are IPv4-in-IPv6 packet reception and IPv4 packet reception. When the lwB4 receives an IPv4-in-IPv6 packet from the lwAFTR, the validation is divided into two stages: source address validation and destination address validation. In source address validation, If IPv6 source address is equal to lwAFTR's IPv6. The packet will step into the destination address validation. If the IPv6 source address is not equal to lwAFTR's IPv6, the packet may be send by lwB4 which perform Bypass lw4over6 Process. Then, the packet must be verified the IPv4 source address. If IPv4 source address is in Bypass Scope, the packet will step into the destination address validation. In the other hand, if IPv4 source address is not in Bypass Scope, the packer must be drop silently. In destination address validation, if IPv4 destination address and port match lwB4's IPv4 address and port. Then, the packet is performed NAPT44 translation on the destination address and port, based on the available information in its local NAPT44 table and forward to the IPv4 destination. However, If IPv4 destination address and port do not match. Then, the packet must be dropped. An ICMPv6 error message (type 1 code 1 - communication with destination administratively prohibited) is sent back to the lwAFTR.

When the lwB4 receives such an IPv4 packet, it performs NAPT44 function on the source address and port by using public IPv4 address and a port number from the allocated port-set. Then, if the IPv4 destination and port is not in Bypass Scope, it encapsulates the packet with an IPv6 header and forwards the encapsulated packet to the configured lwAFTR. If the IPv4 destination is in Bypass Scope, the IPv4 destination and port must be checked with Bypass lw4over6 Table. If the IPv4 destination and port matches to a single entry, the lwB4 forwards the encapsulated packet to the lwB4's IPv6 address of the matched entry. If no match is found, the lwB4 must forwards the encapsulated packet to the configured lwAFTR and then the IPv4 destination and port must be checked with Requested Destination Table. If match is found (the IPv4 destination and port is already requested for bypass tunnel), the lwB4 do nothing. If no match is found, the lwB4 must query the information of destination tunnel-endpoint and add IPv4 destination address and port into Requested Destination Table.

### *C. The lwB4 Information*

Creating of direct tunnel in proposed method, lwB4 requires additional information more than ordinary lw4over6 information: IPv6 address of lwAFTR, public IPv4 address of lwB4 and IPv4 port-set. Additional information of lwB4 in proposed method are three types such as Bypass Scope, Bypass lw4over6 Table and Requested Destination Table. Bypass lw4over6 Table is used to record the requested tunnel-endpoint information as tunnel-endpoint information of lwAFTR. Requested Destination Table is used to record IPv4 address and port of destination which is already requested the tunnel-endpoint information.

## IV. SIMULATION SCENARIOS

Simulation scenarios are based on Thai UniNet's network which aims to provide internet service for educational agencies in Thailand. The simulation network has 11 sub-networks. Each sub-network consists of one ISP gateway network, 8 customer networks and 2 ISP core networks. Gateway of each subnet connects with each other by using IPv6 only. Except for provider relay in ISP gateway network and customer edge in customer network, they support IPv4 connectivity that are provided through the transition. Comparative transitions include 4over6, lw4over6 and proposed method that are compared in Table I. In addition, applications in this test are HTTP and FTP which has the TCP parameters, as TCP Reno and fast retransmit.

## V. SIMULATION RESULTS AND ANALYSIS

In this section, we compare all 3 transitions, to investigate in terms of transmission success rate and traffic dropped. Each transition is simulated in 5 sub-scenarios by varying the intra-communication rate and the inter-communication rate. The 5 sub-scenarios consist of Intra-communication 0%, 25%, 50%, 75% and 100% (Intra-communication 25% means that 25% of destinations locate inside ISP network and 75% of destinations locate outside ISP network).

### *A. Transmission Success Rate*

Fig. 2 shows the transmission success rate that is calculated by the throughput per transmitted traffic (excludes retransmission and header). The highest transmission

TABLE I. COMPARISON OF TUNNELING MECHANISM

| Issues | IPv4-in-IPv6 tunneling mechanism | | |
|---|---|---|---|
| | ***4over6*** | ***lw4over6*** | ***Proposed method*** |
| mechanism | stateful transition and stateful NAT on customer network | stateful transition and stateful NAT on customer network | stateful transition and stateful NAT on customer network |
| IPv4 address allocation | public IPv4 address | public IPv4 address with port set | public IPv4 address with port set |
| tunnel-endpoint specification | specify tunnel-endpoint depend on 4over6 routing table | specify static tunnel-endpoint to lwAFTR | specify tunnel-endpoint depend on Bypass lw4over6 Table |
| connectivity model | mesh model | hub and spoke model | mesh model |

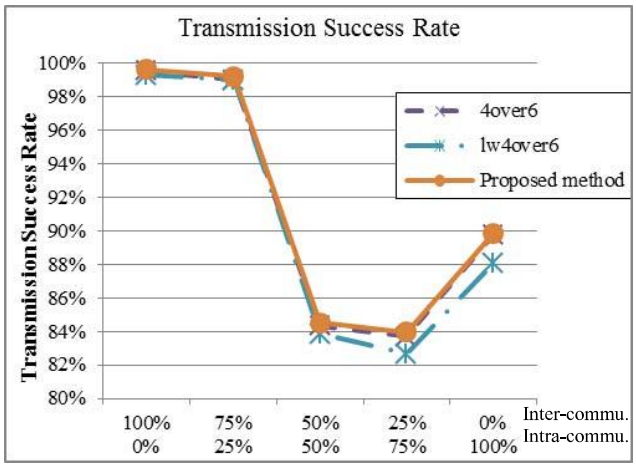



Figure 2. Transmission success rate.

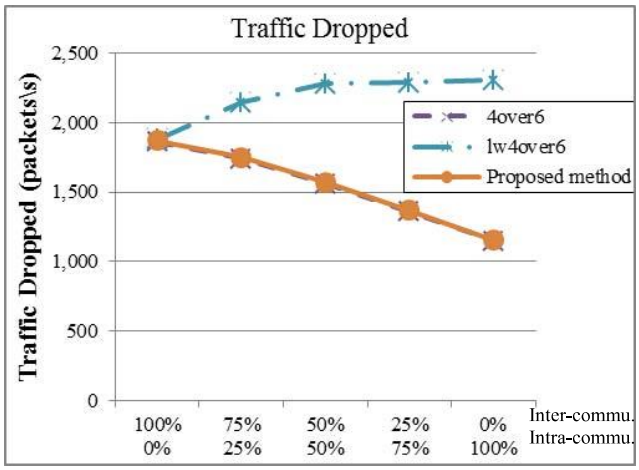



Figure 3. Traffic dropped.

success rate is proposed method, followed by 4over6 and lw4over6 respectively. While intra-communication rate decrease, transmission success rate in proposed method, lw4over6 and 4over6 have slightly different. Even if intra-communication rate increase, transmission success rate in each transition have different significantly. Proposed method and 4over6 have higher transmission success rate than lw4over6 because both transitions support mesh IPv4 connectivity. Proposed method has highest transmission success rate, since it maintains only tunnel-endpoints that already communicate to destination network. Contrast with 4over6, it maintain all tunnel-endpoints. Therefore, proposed method has lower amount of tunnel-endpoint than 4over6.

### *B. Traffic dropped*

Traffic dropped is shown in Fig. 3. Transition that has the increasing trend of traffic dropped is lw4over6 because lw4over6 does not support mesh IPv4 connectivity. When the intra-communication increase, lw4over6 is affected by the increasing IPv4-in-IPv6 traffic that is sent from customer edge to provider relay before forwarded to destination. In contrast, the traffic dropped of remaining transitions are likely to decrease. Since traffic of proposed method and 4over6 can be directly sent to the destination network without provider relay of ISP. Path of traffic in both transitions is less and appropriate. The highest traffic dropped is lw4over6 followed by propose method and 4over6, respectively. The lw4over6 has the highest traffic dropped because it always sends traffic to provider relay first. Traffics in main path to central are the more density. While traffic dropped of 4over6 less than the proposed method slightly because proposed method has throughput more than 4over6.

## VI. CONCLUSIONS

This paper proposes the enhancement of lw4over6 to support mesh IPv4 connectivity by using DHCPv4 leasequery over DHCPv6. The lwB4 of the proposed method is more complexity because the lwB4 is modified some behaviours to support mesh feature. However, performance of proposed method in pure inter-communication less than lw4over6 and 4over6 slightly. Nevertheless, when the system has higher intra-communication, proposed method has higher performance than lw4over6 significantly because it can communicate to the customer network directly. Moreover, proposed method has more performance than 4over6 slightly because tunnel-endpoints of proposed method are queried only tunnels that communicate to the required customer network. Contrast with 4over6, it maintain all tunnel-endpoints. Consequently, proposed method is the one of best transition for the networks that focuses on the quality of IPv4 connectivity.

## REFERENCES

[1] Google, (2014, April 12). Google IPv6 Statistics [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html.

[2] S. Kerner, (2014, April 9). IPv6 & IPv4 Will Co-Exist for a Long Time [Online]. Available: http://www.enterprisenetworkingplanet.com/netsp/ipv6-ipv4-will-co-exist-for-a-long-time.html.

[3] N. Chuangchunsong et al. "Performance Evaluation of IPv4/IPv6 Transition Mechanisms: IPv4-in-IPv6 Tunneling Techniques," in *International Conference on Information Networking (ICOIN)*, pp.238-243, February 2014.

[4] J. Wu et al. "4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions," in *IETF RFC5747*, March 2010.

[5] Y. Cui et al. "4over6: network layer virtualization for IPv4-IPv6 coexistence," in *Network, IEEE*, vol.26, no.5, pp.44-48, October 2012.

[6] Y. Cui et al. "Lightweight 4over6: An Extension to the DS-Lite Architecture," in *IETF draft*, March 2014.

[7] Y. Cui and Q. Sun, "Lightweight 4over6 Deployment with DHCPv4 over DHCPv6," in *IETF draft*, October 2013.

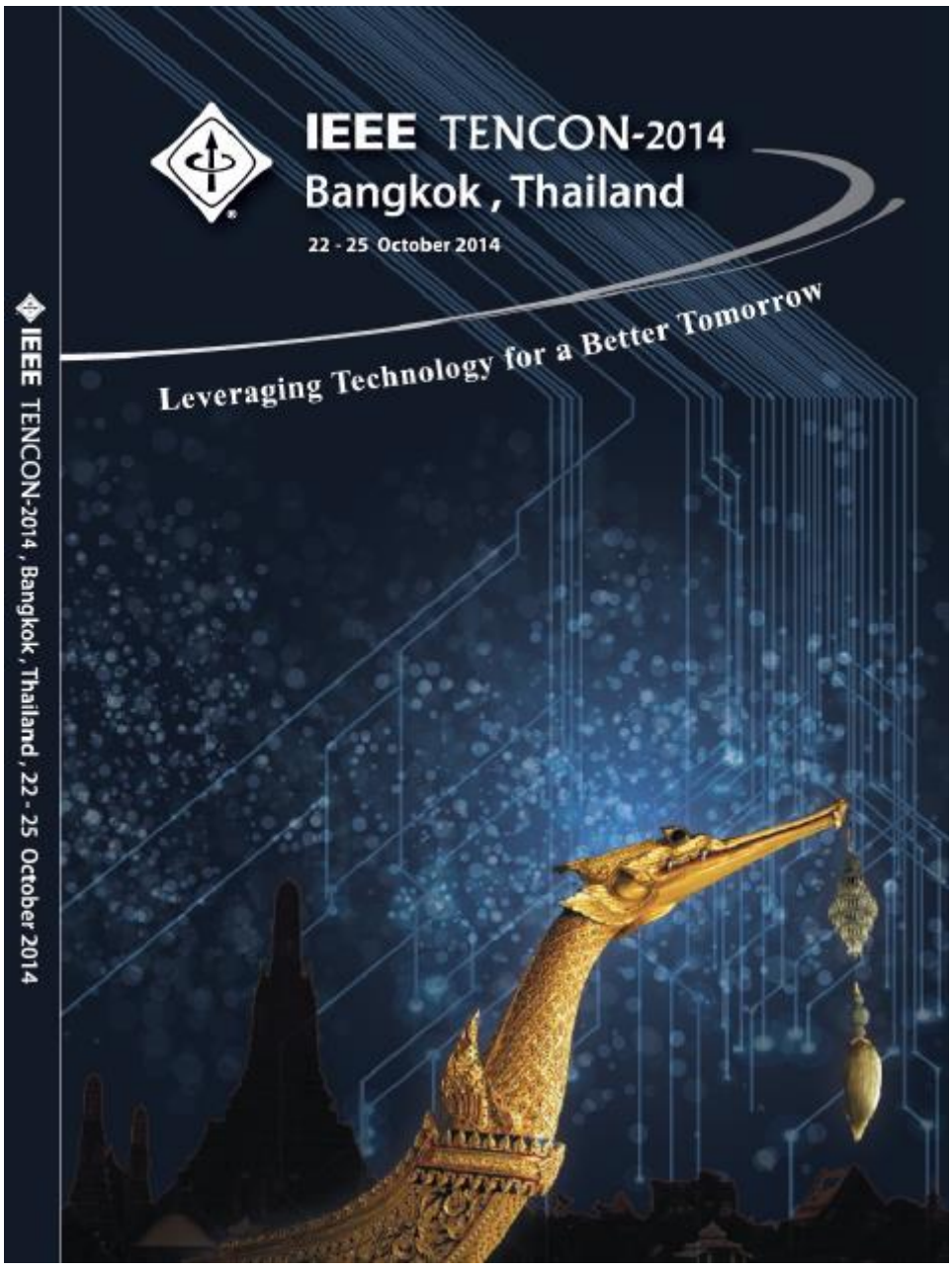
[8] C. Xie et al. "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Lightweight 4over6," in *IETF draft*, July 2013.

[9] Q. Sun et al. "DHCPv4 over DHCPv6 Transport," in *IETF draft*, February 2014.

[10] R. Woundy and K. Kinnear, "Dynamic Host Configuration Protocol (DHCP) Leasequery," in *IETF RFC4388*, February 2006.

[11] P. Kurapati et al. "DHCPv4 Lease Query by Relay Agent Remote ID," in *IETF RFC6148*, February 2011.

[12] Y. Lee et al. "Simple Failover Mechanism for Lightweight 4over6," in *IETF draft*, July 2013.


IEEE TENCON-2014
Bangkok , Thailand
22 - 25 October 2014
Leveraging Technology for a Better Tomorrow
IEEE TENCON-2014 , Bangkok , Thailand , 22 - 25 October 2014

# Performance of Intra and Inter communications of IPv4-in-IPv6 Tunneling Mechanisms

N. Chuangchunsong
Department of Computer Engineering
Prince of Songkla University
Songkla, Thailand
napat.chu@gmail.com

T. Kamolphiwong
Department of Computer Engineering
Prince of Songkla University
Songkla, Thailand
kthossaporn@coe.psu.ac.th

T. Angchuan
Department of Computer Engineering
Prince of Songkla University
Songkla, Thailand
touch@coe.psu.ac.th

***Abstract*—In order to reach the customers thoroughly, internet contents need to support both IPv4 and IPv6. In this case, ISPs should provide their customers with IPv4 and IPv6 access seriously until the IPv4 is disabled eventually. Each transition solution needs to concern not only high working performance but also network supporting in the future. This paper investigates the performance of various transition schemes; IPv4 Residual Deployment across IPv6-Service networks (4rd), 4over6, Lightweight 4over6 (Iw4over6) and Enhancement of Lightweight 4over6 (elw4over6). Computer simulations are divided into two scenarios: intranet (intra-communication) and internet (inter-communication) based on the following performance metrics: CPU utilization, HTTP traffic received, HTTP traffic lost and HTTP object response time. We have found that elw4over6 is the best one for the networks that require flexibility of IP address allocation and providing good quality of IPv4 connectivity.**

***Keywords—IPv6; IPv4-over-IPv6; lw4over6; IPv6 migration***

## I. INTRODUCTION

The adoption of IPv6 has increased at present globally. However, IPv6 deploying in many countries does not increase significantly. Internet contents still need to support both IPv4 and IPv6 to reach every group of people [1]. Another support factor is that IPv4 and IPv6 address can operate over shared network by using appropriate transition mechanism without changing the existing topology. This will help deployment of changing from IPv4 to IPv6 gradually.

Based on our previous work by comparing the performance of IPv4-in-IPv6 tunneling transitions [2], those transitions were 4rd, Dual Stack lite and 4over6. Results of such study have been used to improve our proposed transition in order to obtain the best one of transitions. The proposed transition is developed from Lightweight 4over6 (lw4over6). It is called Enhancement of Lightweight 4over6 (elw4over6). So, this paper compare the IPv4-in-IPv6 tunneling transition, including 4rd, 4over6, lw4over6 and elw4over6 within 2 networks: intranet and internet.

## II. TRANSITION MECHANISMS

### *A. IPv4 Residual Deployment across IPv6-Service networks*

IPv4 Residual Deployment across IPv6-Service networks (4rd) which is stateless transition, aim to distribute the remaining IPv4 addresses to the customer's network by using IPv4-in-IPv6 tunneling [3]. Stateless operation is performed depending on 4rd rule. Thus, IPv6 address and IPv4 address with port set must accord with 4rd rule strictly. Moreover, the design of 4rd takes into account the remaining public IPv4 address of each network that may not be equal and continuous. A 4rd domain may have multiple 4rd rules in order to cover all IPv4 ranges. By using 4rd rules, 4rd supports the difference formats of IPv4 addresses, such as public IPv4 prefix, public IPv4 with port-set, private IPv4 and no IPv4 with NAT64+.

### *B. 4over6*

4over6 is one of stateful transition applying the routing protocol for exchange network information. By the operation, 4over6 can use the exchanged information to specify the IPv4-in-IPv6 tunneling, and the destination IP accordingly [4],[5]. Routing protocol used in 4over6 is Multiprotocol Extensions for BGP (MP-BGP). Since, MP-BGP supports different type of addresses. It is used for exchange IPv4 routing information over IPv6 conveniently. Special routing information in 4over6 consists of destination IPv4 prefix and next hop's IPv6 address. When router which is configured 4over6 transition, receives IPv4 packet, IPv4 packet must be encapsulated into IPv6 packet and assign IPv6 destination as next hop's IPv6 address of matched IPv4 prefix. Exchanged routing information enables the tunnel-endpoint specification to destination network correctly. Moreover, the exchange routing information must be updated before the connectivity is established.

### *C. Lightweight 4over6*

Lightweight 4over6 (lw4over6) is stateful transition as 4over6, and is intended to serve as an extension of DS-lite [6]. So, customer's gateway is called Lightweight Basic Bridging BroadBand (lwB4) and provider's equipment is called Lightweight Address Family Transition Router (lwAFTR), both are similar to equipment in DS-lite. In order to enable distributed processing, lw4over6 allocates public IPv4 address with port set to lwB4 by using DHCP 4o6 server [7],[8],[9]. LwAFTR contains only 3 information types which are: Public IPv4 address, Port-set, and lwB4's IPv6 address instead of the 5 information types as the DS-lite. Since lwAFTR does not perform NAT, it can reduce processing time at centralization reasonably. Although port utilization of lw4over6 is not quite equivalent to DS-lite. However, lw4over6 is still drafting by the IETF. Some operations of lw4over6 are still incomplete.

978-1-4799-4075-2/14/$31.00 ©2014 IEEE

### D. *Enhancement of Lightweight 4over6*

Enhancement of Lightweight 4over6 (elw4over6) is a modification of lw4over6, to reduce hairpin that occurs when the pair of IPv4 host within same lw4over6 domain which needs to exchange data between them [10]. The source must send IPv4 packet to provider's equipment first, this causes a bottleneck at provider's equipment. Moreover, provider's equipment relays not only IPv4 packets within provider network, but also IPv4 packets that sent to the outside destination of the provider network as well. Therefore, the occurred bottleneck reduces the overall performance inevitably.

The name of Elw4over6 equipment is the same as lw4over6; customer's gateway is called Lightweight Basic Bridging BroadBand (lwB4) and provider's equipment is called Lightweight Address Family Transition Router (lwAFTR). Elw4over6 is applied to send IPv4 packets to the destination within same transition domain directly by using bypass scope and bypass lw4over6 table essentially. Bypass scope is the range of IPv4 address, which can allocate to the customer. So, the bypass scope is used to filter the packets within elw4over6 domain and IPv4 packets outside provider network. Next, bypass lw4over6 table is similar tunneling table on the lwAFTR. It is used to record the established tunneling of each lwB4 in order to connect to destination directly in next time. To record the information, elw4over6 applies DHCP lease query [11] to exchange tunnel-endpoint information in bypass lw4over6 table. Although, this function increases the burden on lwB4. However, it can reduce the bottleneck on lwAFTR, and select the appropriate path to the destination within elw4over6 domain directly. Thus, the source can send IPv4 packets to destination network without lwAFTR. IPv4 connectivity establishment of elw4over6 is shown in Fig. 1.

## III. Simulation Scenarios

Simulation topology is based on Thai UniNet's network layout which aims to provide internet service for educational agencies in Thailand. The simulation network consists of ISP gateway network (BKK_GW), 2 ISP core networks (PNK, NRM) and 8 customer networks as shown in Fig. 2. There is only a provider relay in ISP gateway network, and customer edge in customer network provides IPv4 connectivity by using transitions such as 4rd, 4over6, lw4over6 and elw4over6. The transition details are shown in Table I. Moreover, this experiment uses HTTP Browsing and HTTP Searching in the simulation. Both HTTP behaviors use TCP Reno and fast retransmit mechanisms.

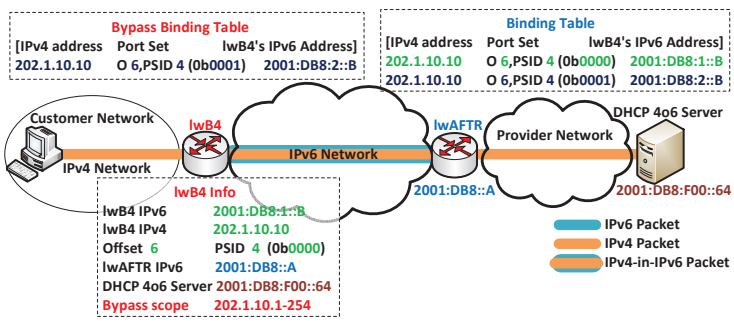



Fig. 1. IPv4 connectivity establishment of proposed method.

TABLE I. Comparison of Transition Mechanisms

| Issues | IPv4-in-IPv6 tunneling mechanism | | | |
|---|---|---|---|---|
| | *4rd* | *4over6* | *lw4over6* | *elw4over6* |
| Mechanism | stateless | stateful | stateful | stateful |
| IP address allocation | static public IPv4 address with port set | public IPv4 address | dynamic public IPv4 address with port set | dynamic public IPv4 address with port set |
| Tunnel-endpoint specification | depend on 4rd rule of each domain | depend on 4over6 routing table | static tunnel-endpoint to lwAFTR | depend on bypass lw-4over6 Table |
| Connecti-vity model | mesh | mesh | hub & spoke | mesh |
| Flexibility | 4RD use multiple static 4rd rules and 4rd rule is re-assigned by DHCPv6 | 4over6 update tunnel-endpoint by IPv4/IPv6 routing protocol | lwAFTR update new tunnel-point by re-initial IPv4 request process | lwB4 exchange tunnel-endpoint information with DHCP 4o6 server |

## IV. Simulation Results and Analysis

The simulation model is divided into two major scenarios depending on location of destination nodes. The first one has the destination nodes in a provider's network (intra-communication or intranet) and another one has the destination nodes outside a provider's network (inter-communication or internet). Moreover, each scenario is divided into 4 sub-scenarios according to number of clients within the network, such as number of clients (e.g. 512, 1,024, 1,536 and 2,048). All scenarios are compared and investigated in terms of CPU utilization, HTTP traffic received, HTTP traffic lost and HTTP object response time.

### A. *Pre-analysis performance*

In pre-analysis of the performance, approximate results of basic performance metrics are shown in Table II. We compare them in terms of hop count, link bandwidth utilization and beginning point of congestion. Hop count is a summation hop count between servers and clients in the system, and hop count per client is the average hop value for a client. Next, beginning point of congestion indicates time that congestion occurs. The last value, link hop count is a summation hop count between servers and clients in the system utilization is calculated by the sum of maximum bandwidth used in all links divided by link capacity of the system.

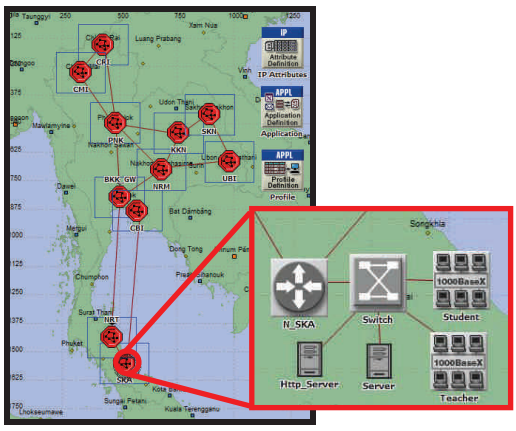



Fig. 2. The simulated network topology.

TABLE II. PERFORMANCE COMPARISON OF TRANSITIONS

| Issues | IPv4-in-IPv6 tunneling mechanism | | | |
|---|---|---|---|---|
| | *4rd* | *4over6* | *lw4over6* | *elw4over6* |
| Intra-communication | | | | |
| - Hop count | 144 | 144 | 322 | 144 |
| - Hop count per client | 2.57 | 2.57 | 5.75 | 2.57 |
| - Beginning point of congestion (Second) | 3005 | 2998 | 1082 | 3042 |
| - Link bandwidth utilization (%) | 45.72 | 45.78 | 26.47 | 45.64 |
| Inter-communication | | | | |
| - Hop count | 161 | 161 | 161 | 161 |
| - Hop count per client | 2.875 | 2.875 | 2.875 | 2.875 |
| - Beginning point of congestion (Second) | 934 | 922 | 938 | 926 |
| - Link bandwidth utilization (%) | 30.01 | 32.14 | 29.89 | 30.86 |

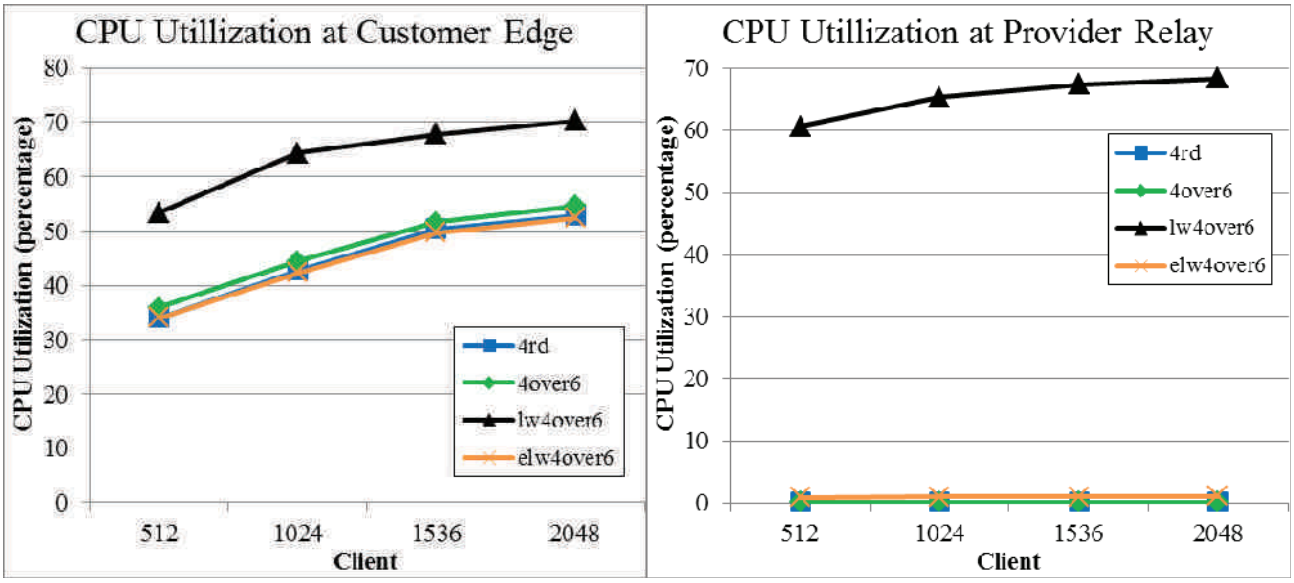
The distinct result of intra-communication is lw4over6. Lw4over6 has maximum hop count. Lw4over6 network accumulated packets and congest rapidly. Other transitions have lower hop count. Moreover, they have higher link utilization because of mesh connectivity. On the other hand, the results of inter-communication are closed to each other. Therefore, performance of transitions in inter-communication might have a slightly difference.

### *B. Intra-communication*

*1) CPU utilization:* CPU utilization of intra-communication is shown in Fig. 3. Fig. 3(a) shows the CPU utilization at customer edge which is customer's gateway. Fig. 3(b) shows the CPU utilization at provider relay which is provider's equipment. The lowest CPU utilization of intra-communication at customer edge is elw4over6, followed by 4rd, 4over6 and lw4over6 respectively. Elw4over6 and 4over6 are stateful operation. Elw4over6 takes time less than 4over6 because elw4over6 maintains only communicated tunnel, but not all tunnels as 4over6. On the other hand, 4rd which is stateless operation has higher CPU utilization than elw4over6 slightly because processing time of elw4over6 depends on the number of communicated tunnels. If the number of tunnels is small, elw4over6 requires a little time to process. However, the most interesting is lw4over6 takes the lowest processing time. This is because it specifies the static tunnel-endpoint to provider relay. It has a maximum CPU utilization. Since, increasing of CPU utilization is not only a process from transition operation but also from forwarding packet in the network. Lw4over6 behavior always forwards packet to provider relay first. So, it has a number of hop count more than other transitions.

In Fig. 3(b), the lowest CPU utilization of intra-communication at provider relay is 4rd and 4over6, followed by elw4over6 and lw4over6 respectively. CPU utilization of 4rd and 4over6 are equal to 0% because both transitions have pre-assigned tunneling before a connection established. 4rd specifies tunnel-endpoint depending on 4rd rule, and 4over6 specifies tunnel-endpoint depending on a special routing table. While elw4over6 assigns a tunnel after it establishes a connection, it has a little CPU utilization. However, the highest CPU utilization at provider relay is from lw4over6 because provider relay of lw4over6 always forwards all packets in the network.

*2) HTTP traffic received:* HTTP traffic received of intra-communication is shown in Fig. 4. HTTP traffic received is the amount of data received by all HTTP servers and HTTP clients. It excludes packet overhead and retransmission packets. The highest HTTP traffic received is elw4over6, followed by 4rd, 4over6 and lw4over6 respectively. If a number of clients is less than 2,048, HTTP traffic received of elw4over6, 4rd and 4over6 are in linear incremental, and do not differ from each significantly. In turn, if system has 2048




(a) (b)

Fig. 3. CPU utilization of intra-communication (a) at Customer Edge (b) at Provider Relay.

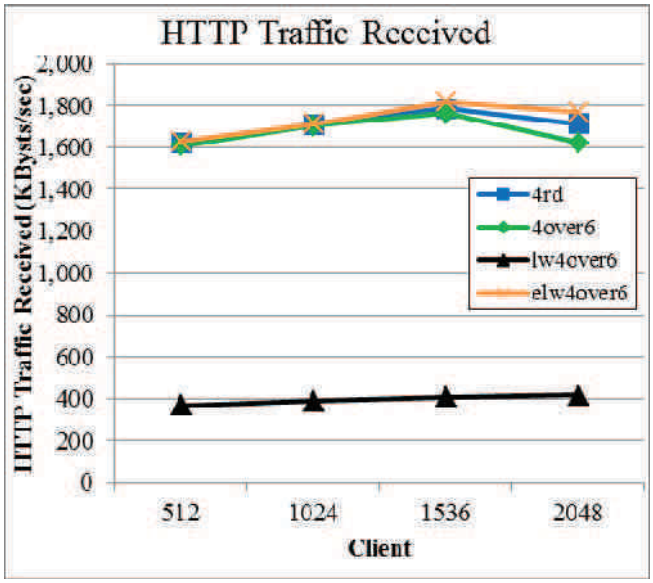



Fig. 4. HTTP traffic received of intra-communication.

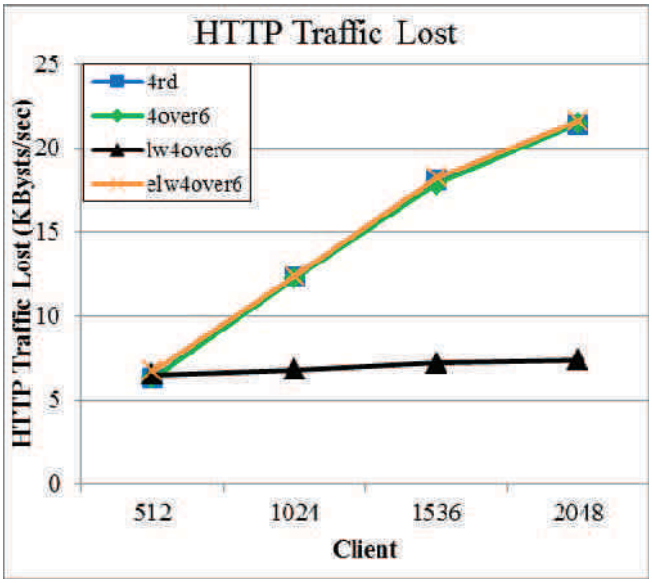



Fig. 5. HTTP traffic lost of intra-communication.

clients, their HTTP traffic received decrease because of congestion effect. Moreover, elw4over6 which has lowest increasing rate of HTTP traffic received has highest HTTP traffic received because elw4over6 established a communication tunnel after source lwB4 started sending first packet to the new destination lwB4. While tunnel establishment is not achieved, packets are forwarded to destination network by lwAFTR. Therefore, TCP transmission rate of elw4over6 increases less than other transitions. However, when the congestion occurred in network, TCP transmission rate in all transitions is decreased. A minimum processing time of elw4over6 is affected when elw4over6 has a maximum TCP transmission rate than other transitions. In contrast, lw4over6 has minimum HTTP traffic received. Since, lw4over6 always sent packet to lwAFTR first. So, HTTP traffic in central links has more density.

*3) HTTP traffic lost:* HTTP traffic lost of intra-communication is shown in Fig. 5. HTTP traffic lost is amount of lost data while HTTP servers and HTTP clients are transmitting. HTTP traffic lost of 4rd, 4over6 and elw4over6 are clustered together and likely to increase. HTTP traffic lost of these transitions differ slightly to each other because the connection of these transitions support mesh model. Once the client is lower than 2048 clients, HTTP traffic lost increase in a straight line as HTTP traffic received. When the system has 2048 clients, network is affected from further congestion. HTTP traffic lost still increase but not a straight line because of TCP congestion avoidance. Other hand, the connection of lw4over6 does not support mesh model. The system began to congest, since the client in system has 512 clients. So, HTTP traffic lost of lw4over6 is as low as HTTP traffic received.

*4) HTT object response time:* HTTP object response time of intra-communication is shown in Fig. 6. HTTP object response time is the time that takes to download a HTTP object. The lowest HTTP object response time is elw4over6, followed by 4rd, 4over6 and lw4over6 respectively that is similar to the CPU utilization at customer edge. HTTP object response time of the best three transitions are little different. Lw4over6 is only one transition that has HTTP object response time more than other transitions clearly because it cannot specify tunnel-endpoint to the destination network directly. When elw4over6, 4rd and 4over6 have the number of clients less than 2,048 clients, HTTP object response time of each transition is likely to be a straight line. It illustrates that the system began to congest at 2,048.

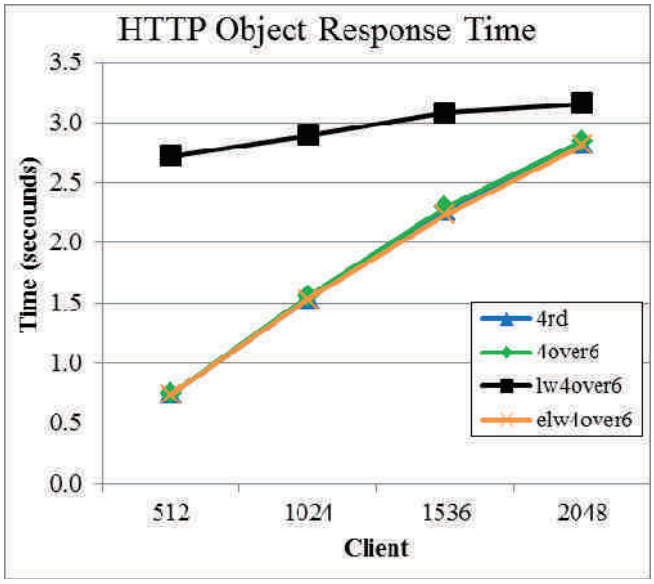
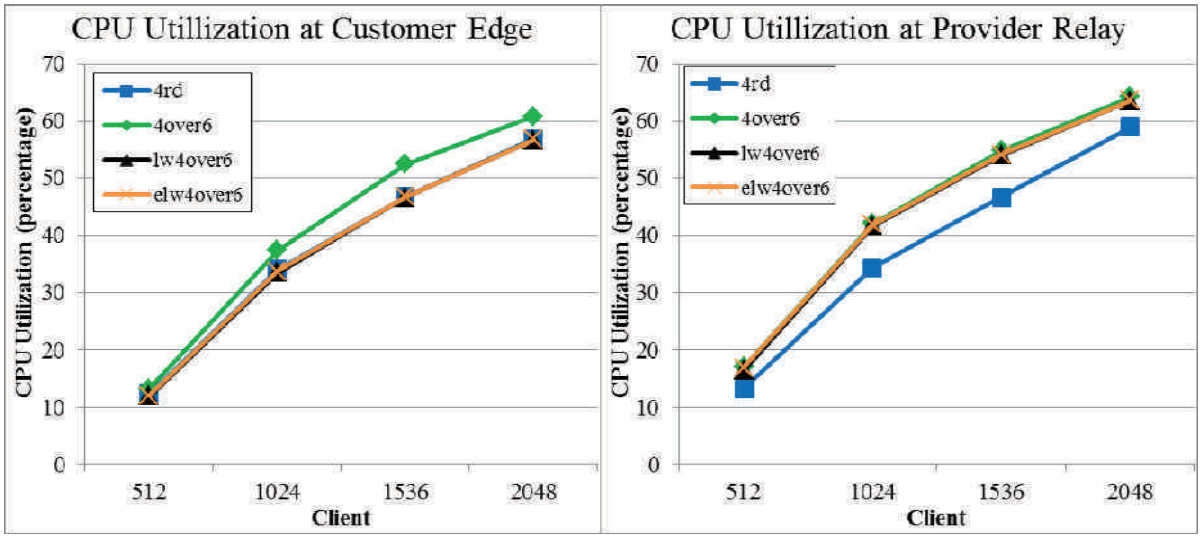



Fig. 6. HTTP object response time of intra-communication.




(a) (b)

Fig. 7. CPU utilization of inter-communication (a) at Customer Edge (b) at Provider Relay.

### C. *Inter-communication*

*1) CPU utilization:* CPU utilization of inter-communication is shown in Fig. 7. Fig. 7(a) shows the CPU utilization at customer edge. The lowest CPU utilization of inter-communication at customer edge is lw4over6, followed by elw4over6, 4rd and 4over6 respectively. Chart of CPU utilization of each transition depends on the its processing time. Although the CPU utilization of lw4over6, elw4over6 and 4rd only differ slightly to each other, lw4over6 still takes lowest processing time because it specifies the static tunnel-endpoint to provider relay only. Elw4over6 uses bypass scope to filter packets within the provider network. Other packets will be sent to the provider relay directly as well. 4rd use 4rd rule to filter packets within the provider network. But, 4rd rule has a little complexity than bypass scope. It also has higher CPU utilization than elw4over6 slightly. The last one, 4over6 has the highest CPU utilization because it specifies tunnel-endpoint by using the special routing table. It must search a destination node in the routing table until default route is found, to send packets to a destination node outside the provider network.

Fig. 7(b) shows the CPU utilization of inter-communication at provider relay. The lowest CPU utilization of inter-communication at customer edge is 4rd. The remained transitions have CPU utilization similarly because these transitions are stateful operation. They have a similar functionality and record all tunneling at provider relay. Thus, these transitions have higher CPU utilization than 4rd, which is stateless operation.

*2) HTTP traffic received:* HTTP traffic received of inter-communication is shown in Fig. 8. HTTP traffic received is likely to decrease, when clients increase because the system has begun to congest. So, TCP reduce the transmission rate exponentially. Although all transitions have similar HTTP traffic received, the highest HTTP traffic received is 4rd, followed by lw4over6, elw4over6 and 4over6 respectively. 4rd also has the highest HTTP traffic received because it has lower CPU utilization at provider relay than other transitions. Therefore, 4rd is affected by bottlenecks less than the others.

*3) HTT traffic lost:* HTTP traffic lost of inter-communication is shown in Fig. 9. HTTP traffic lost is likely to increase until it is the highest when the system has 1,024 clients, and then it has continuous decrease respectively. HTTP traffic lost affects the TCP transmission rate directly. So, HTTP traffic lost rate is reduced as HTTP traffic received rate. While HTTP traffic lost is not up to the peak, the lowest HTTP traffic lost is lw4over6, followed by 4rd, elw4over6 and 4over6 respectively because lw4over6 has the lowest processing time at customer edge. It also has minimum dropped packet at customer edge. However, 4rd compensate for the lower processing time at customer edge with the fastest processing at provider relay. While, the remained transitions have processing time at provider relay same as lw4over6.

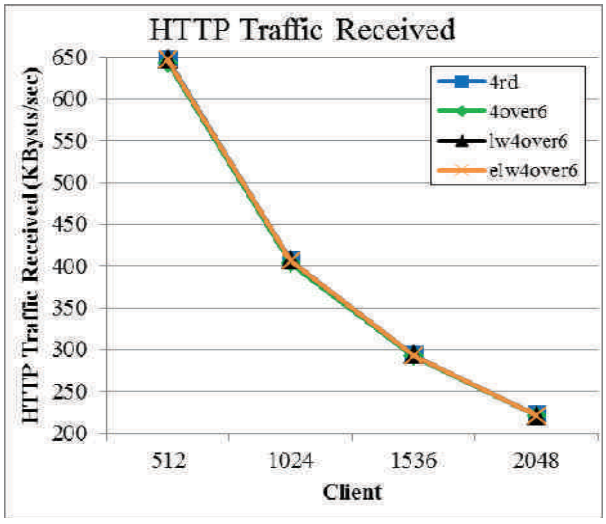



Fig. 8. HTTP traffic received of inter-communication.

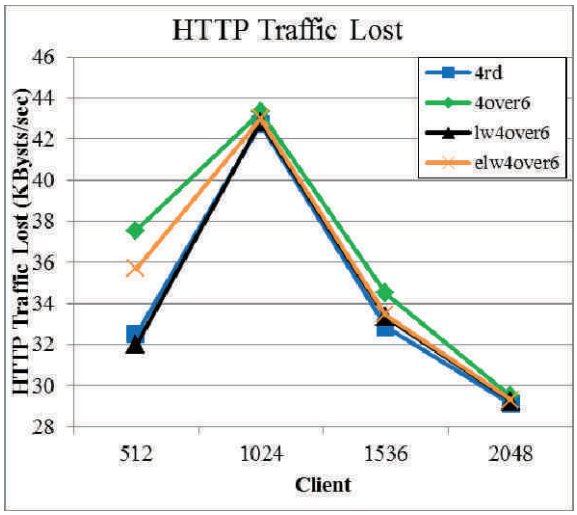



Fig. 9. HTTP traffic lost of inter-communication.

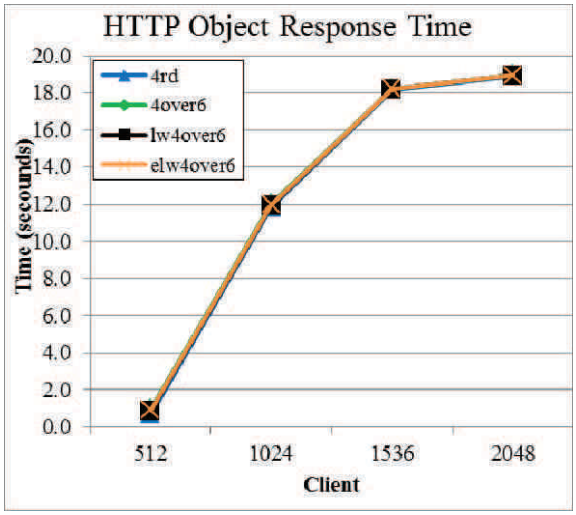



Fig. 10. HTTP object response time of inter-communication.

So, HTTP traffic lost of these transitions depend on the processing time on customer edge. Whereas, HTTP traffic lost peak and has begun to decrease, the lowest HTTP traffic lost is 4rd, followed by lw4over6, elw4over6 and 4over6 respectively because lw4over6, elw4over6 and 4over6 are affected by bottleneck at provider relay more than 4rd. Moreover, HTTP traffic lost of these transitions still depend on the processing time on customer edge.

*4) HTT object response time:* HTTP object response time of inter-communication is shown in Fig. 10. Although HTTP object response time of each transition is likely to increase and is clustered together, the lowest HTTP object response time is 4rd, followed by lw4over6, elw4over6 and 4over6. This chart can explain by CPU utilization in both customer edge and provider relay. HTTP object response time of inter-communication depends on CPU utilization at provider relay and customer edge by focusing on provider relay primarily. The simulation results demonstrated that 4rd has the lowest CPU utilization at provider relay and the other transitions have similar CPU utilization at provider relay. Therefore, 4rd must has the minimum HTTP object response time. Then, CPU utilization at customer edge of remain transitions are compared. Lw4over6 has the lowest CPU utilization at customer edge, followed by elw4over6 and 4over6 respectively that accord with the result of HTTP object response time.

## V. CONCLUSIONS

This paper presents the comparative performance of intra-communication and inter-communication between various transition mechanisms: 4rd, 4over6, lw4over6, and elw4over6. Based on our simulation results, elw4over6 which is enhancement of lw4over6 performances the best highest performance in terms of intra-communication among others. When comparing to lw4over6, elw4over6 performs better performance in terms of CPU utilization, HTTP traffic lost and HTTP object response time. However, for inter-communication, performance of elw4over6 is slightly lower than lw4over6. Next, 4over6 and lw4over6 have different feature entirely. 4over6 uses a specific routing table to specify tunnel-endpoint within provider network. It also has powerful for intra-communication. However, such routing table becomes a double-edged sword because for inter-communication, it always search for a specify tunnel-endpoint until found a default route. This takes a lot of time. In contrast, lw4over6 has the highest performance for inter-communication because of its simple functionality on customer's equipment. However, it has lower performance in intra-communication than other transitions significantly. For the last one, 4rd has high performance in intra-communication and inter-communication because it performs stateless operation. This is inflexible for IP address allocation. Consequently, elw4over6 is the best one for the networks that require flexibility of IP address allocation and focus on the quality of IPv4 connectivity.

## REFERENCES

[1] S. Kerner, (2014, April 9). IPv6 & IPv4 Will Co-Exist for a Long Time [Online]. Available: http://www.enterprisenetworkingplanet.com/netsp/ipv6-ipv4-will-co-exist-for-a-long-time.html.

[2] N. Chuangchunsong et al. "Performance Evaluation of IPv4/IPv6 Transition Mechanisms: IPv4-in-IPv6 Tunneling Techniques," in International Conference on Information Networking (ICOIN), pp.238-243, February 2014.

[3] R. Despres and R. Penno and Y. Lee and G. Chen and S. Jiang , "IPv4 Residual Deployment via IPv6 – a unified Stateless Solution (4rd)," in IETF draft, October 2013.

[4] J. Wu et al. "4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions," in IETF RFC5747, March 2010.

[5] Y. Cui et al. "4over6: network layer virtualization for IPv4-IPv6 coexistence," in Network, IEEE, vol.26, no.5, pp.44-48, October 2012.

[6] Y. Cui et al. "Lightweight 4over6: An Extension to the DS-Lite Architecture," in IETF draft, March 2014.

[7] Y. Cui and Q. Sun, "Lightweight 4over6 Deployment with DHCPv4 over DHCPv6," in IETF draft, October 2013.

[8] C. Xie et al. "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Lightweight 4over6," in IETF draft, July 2013.

[9] Q. Sun et al. "DHCPv4 over DHCPv6 Transport," in IETF draft, February 2014.

[10] N. Chuangchunsong et al. "An Enhancement of IPv4-in-IPv6 Mechanism," in Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp.45-48, August 2014.

[11] R. Woundy and K. Kinnear, "Dynamic Host Configuration Protocol (DHCP) Leasequery," in IETF RFC4388, February 2006.

# ประวัติผู้เขียน

**ชื่อ สกุล** นายณ ภัทร ช่วงชุณห์ส่อง

**รหัสประจำตัวนักศึกษา** 5510120064

**วุฒิการศึกษา**

| วุฒิ | ชื่อสถาบัน | ปีที่สำเร็จการศึกษา |
|---|---|---|
| วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์) | มหาวิทยาลัยสงขลานครินทร์ | 2554 |

**ทุนการศึกษา**

ทุนศิษย์ก้นกุฏิ คณะวิศวกรรมศาสตร์

**การตีพิมพ์เผยแพร่ผลงาน**

N. Chuangchunsong, S. Kamolphiwong, T. Kamolphiwong, R. Elz, and P. Pongpaibool, “Performance evaluation of IPv4/IPv6 transition mechanisms: IPv4-in-IPv6 tunneling techniques,” in *International Conference on Information Networking (ICOIN)*, February 2014, pp. 238–243.

N. Chuangchunsong, S. Kamolphiwong, T. Kamolphiwong and R. Elz, "An Enhancement of IPv4-in-IPv6 Mechanism," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, August 2014, pp.45-48.

N. Chuangchunsong, T. Kamolphiwong, and T. Angchuan, "Performance of Intra and Inter communications of IPv4-in-IPv6 Tunneling Mechanisms," in *TENCON 2014 IEEE Region 10 Conference*, October 2014.