บทที่ 3
วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ

โดยปกติแล้วเมื่อผู้ใช้งานต้องการจะสื่อสารผ่านเครือข่ายหนึ่ง ๆ ก็จะต้องทำการสร้างแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อแล้วส่งมาร้องขอที่เครือข่าย ซึ่งในเครือข่ายอินเทอร์เน็ตโพรโตคอลที่มีการรับประกันคุณภาพในการให้บริการจะมีส่วนควบคุมการอนุมัติการเชื่อมต่อ ซึ่งจะทำการตรวจสอบทรัพยากรทั้งหมดของเครือข่าย ถ้าเครือข่ายมีทรัพยากรเพียงพอที่จะสามารถรองรับการเชื่อมต่อจากผู้ใช้งานเพิ่มได้ ก็จะทำการสร้างแพ็กเกตตอบยอมรับแล้วส่งกลับไปยังผู้ที่ร้องขอมาทำให้สามารถสร้างการเชื่อมต่อและส่งข้อมูลได้ต่อไป แต่ถ้าส่วนควบคุมการอนุมัติการเชื่อมต่อตรวจสอบแล้วพบว่ามีทรัพยากรไม่เพียงพอที่จะให้ผู้ร้องขอทำการเชื่อมต่อได้ ก็จะทำการสร้างแพ็กเกตตอบปฏิเสธแล้วส่งกลับไปยังผู้ร้องขอในทันที และถ้าผู้ที่ร้องขอต้องการที่จะติดต่อสื่อสารผ่านเครือข่ายให้ได้ ผู้ร้องขอจะต้องทำการสร้างแพ็กเกตร้องขอขึ้นมาใหม่แล้วส่งไปยังเครือข่ายเพื่อทำการสร้างการเชื่อมต่อซ้ำ ๆ จนกว่าจะได้รับการตอบยอมรับจากเครือข่าย

สภาวะที่เครือข่ายมีทรัพยากรเพียงพอที่จะสามารถรองรับผู้ใช้งานให้ทำการเชื่อมต่อเพิ่มได้จะเรียกว่า “สภาวะปกติ” ส่วนสภาวะที่เครือข่ายมีทรัพยากรไม่เพียงพอที่จะรองรับการเชื่อมต่อจากผู้ใช้งานเพิ่มได้จะเรียกว่า “สภาวะอิ่มตัว” จากปัญหาที่กล่าวมาข้างต้นนี้ ในสภาวะที่เครือข่ายอิ่มตัวซึ่งผู้ใช้งานต้องการที่จะสื่อสารส่งข้อมูลผ่านเครือข่ายแล้วต้องทำการสร้างและส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อซ้ำ ๆ จนกว่าจะได้รับแพ็กเกตตอบยอมรับ ในวิทยานิพนธ์ฉบับนี้ได้มีแนวคิดที่ต้องการลดความยุ่งยากในการร้องขอจัดตั้งการเชื่อมต่อ โดยทำการหน่วงเวลาแพ็กเกตร้องขอไว้ในคิวรอการเชื่อมต่อ แทนที่จะปฏิเสธแพ็กเกตร้องขอนั้นไปในทันที ซึ่งสามารถลดทราฟฟิกที่เกิดจากการร้องขอซ้ำ ๆ และลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อได้ จึงได้เสนอ “วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ” เพื่อเป็นกลไกในการทำงานของส่วนควบคุมการอนุมัติการเชื่อมต่อในสภาวะที่เครือข่ายอิ่มตัว

วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ (Call Request Delaying Method: CRDM) เมื่อเปรียบเทียบหลักการของส่วนควบคุมอนุมัติการเชื่อมต่อในแบบพื้นฐานแล้ว จะสามารถลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ และทำให้เครือข่ายลดทราฟฟิกที่เกิดจากการส่งแพ็กเกตร้องขอซ้ำ ๆ ได้ นอกเหนือจากนั้นยังได้คำนึงถึงความเสมอภาคของผู้ร้องขอในการได้รับการจัดสรรทรัพยากรจากเครือข่ายด้วย

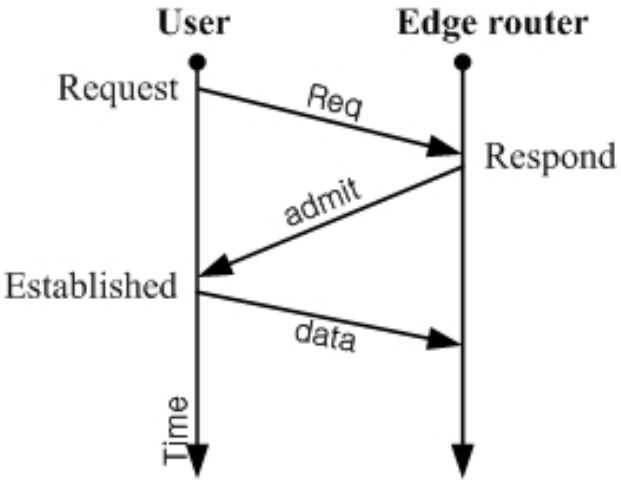
ในบทที่ 3 นี้จะกล่าวถึงการทำงานของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธและการพัฒนาบนโปรแกรมจำลองระบบเครือข่าย

3.1 การทำงานของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ (Call Request Delaying Method: CRDM)

ในวิทยานิพนธ์ฉบับนี้ได้เสนอกลไกการทำงานของส่วนควบคุมการอนุมัติการเชื่อมต่อแบบใหม่ โดยจะแบ่งการทำงานของส่วนควบคุมการอนุมัติการเชื่อมต่อเป็น 2 สภาวะคือ สภาวะปกติและสภาวะอิ่มตัว

3.1.1 ส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธในสภาวะปกติ

ในสภาวะปกติคือ สภาวะที่เครือข่ายมีแบนด์วิดธ์อยู่เพียงพอที่จะสามารถรองรับการเชื่อมต่อจากผู้ใช้งานที่ร้องขอได้ซึ่งส่วนควบคุมการจัดตั้งการเชื่อมต่อในสภาวะนี้จะมีการทำงานเหมือนกับส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบทั่ว ๆ ไปในเครือข่ายที่ยังมีแบนด์วิดธ์



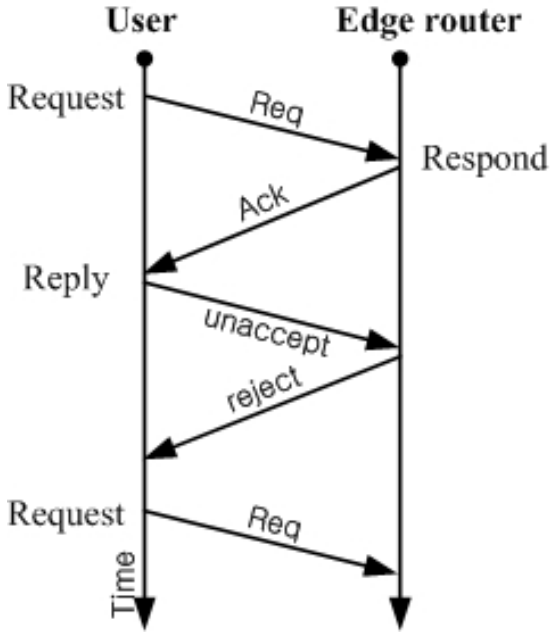
รูปที่ 3.1 การทำงานของ CRDM ในสภาวะปกติ

จากรูปที่ 3.1 เมื่อผู้ใช้งานได้ส่งแพ็กเกตร้องขอ (Req) มายังเครือข่าย ส่วนควบคุมการจัดตั้งการเชื่อมต่อก็จะสามารถอนุญาตให้ผู้ใช้งานทำการเชื่อมต่อได้เลยโดยการส่งแพ็กเกตตอบอนุมัติ (admit) ไปยังผู้ใช้งาน เนื่องจากในสภาวะปกติมีแบนด์วิดธ์อยู่เพียงพอที่จะสามารถทำการรับการเชื่อมต่อจากผู้ใช้งานที่ร้องขอเข้ามาได้

3.1.2 ส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธในสภาวะอิ่มตัว

ในสภาวะอิ่มตัวคือ สภาวะที่เครือข่ายมีแบนด์วิดธ์ไม่เพียงพอที่จะสามารถรองรับการเชื่อมต่อจากผู้ใช้งานที่ร้องขอเข้ามาได้ ซึ่งส่วนควบคุมการจัดตั้งการเชื่อมต่อในสภาวะนี้จะทำงานโดยการหน่วงเวลาการปฏิเสธเมื่อมีการร้องขอเข้ามาและเก็บแพ็กเกตร้องขอไว้ในคิวรอการเชื่อมต่อ โดยจะ

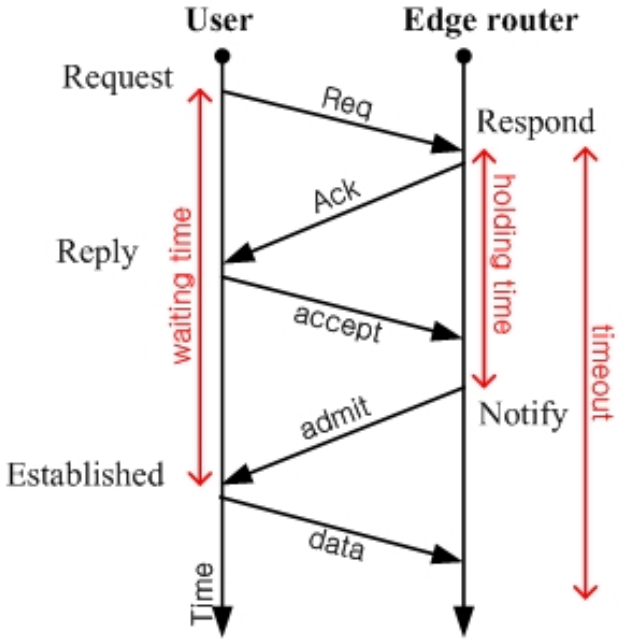
จัดลำดับตามการมาถึงก่อนหลังเพื่อให้เกิดความเสมอภาคในการได้รับการจัดสรรแบนด์วิดธ์ทำให้มีโอกาสในการเชื่อมต่อได้เท่า ๆ กัน โดยแต่ละแพ็กเกตจะถูกตั้งค่าเวลาในการรอคอยของแพ็กเกตไว้เพื่อป้องกันการถูกหน่วงเวลาไปโดยไม่มีที่สิ้นสุด ต่อมาส่วนควบคุมการจัดตั้งการเชื่อมต่อจะทำการส่งแพ็กเกตตอบสนอง (Ack) กลับไปยังผู้ที่ร้องขอเพื่อให้ทราบถึงสถานะการหน่วงเวลาแพ็กเกตไว้ ซึ่งผู้ร้องขอมีสิทธิ์ที่จะยอมรับการหน่วงเวลาจากเครือข่ายหรือไม่ก็ได้ ซึ่งในกรณีที่ไม่ยอมรับการหน่วงเวลา ผู้ใช้งานจะส่งแพ็กเกตตอบปฏิเสธ (unaccept) กลับไปยังเครือข่าย แล้วส่วนควบคุมการจัดตั้งการเชื่อมต่อจะทำการลบข้อมูลแพ็กเกตร้องขอที่อยู่ในคิวรอการเชื่อมต่อและจะส่งแพ็กเกตละทิ้ง (reject) กลับไปยังผู้ร้องขอและเพื่อให้ผู้ใช้งานได้ทราบว่าส่งแพ็กเกตตอบปฏิเสธมาถึงเครือข่ายแล้ว ซึ่งถ้าผู้ร้องขอยังคงต้องการที่จะติดต่อสื่อสารส่งข้อมูลผ่านเครือข่ายก็จะต้องสร้างแพ็กเกตร้องขอและส่งเข้ามาร้องขออีกครั้ง ตามรูปที่ 3.2



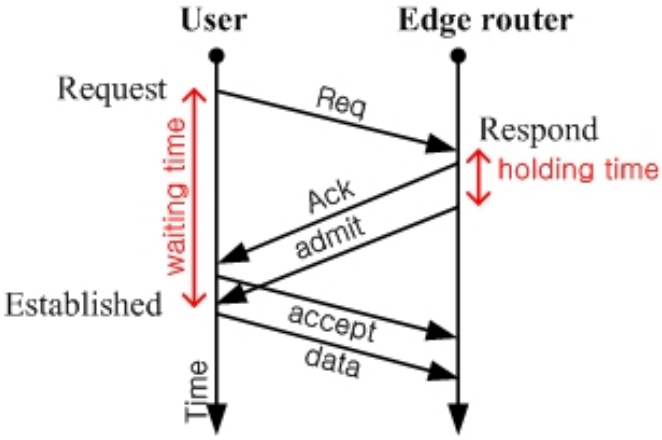
รูปที่ 3.2 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีผู้ร้องขอปฏิเสธ

ในกรณีที่ผู้ใช้งานยอมรับการหน่วงเวลาก็จะส่งแพ็กเกตตอบยอมรับ (accept) กลับไปยังส่วนควบคุมการจัดตั้งการเชื่อมต่อแล้วก็รอการจัดสรรแบนด์วิดธ์จากเครือข่าย เมื่อส่วนควบคุมการจัดตั้งการเชื่อม ต่อได้รับแพ็กเกตตอบยอมรับการหน่วงเวลาแล้ว ก็จะทำการให้แพ็กเกตร้องขอที่อยู่ในคิวรอการเชื่อมต่อรอการจัดสรรแบนด์วิดธ์จากส่วนควบคุมการจัดตั้งการเชื่อมต่อ ในการจัดสรรแบนด์วิดธ์ ในวิทยานิพนธ์นี้ได้คิดและทำการทดลอง 2 รูปแบบคือ การจัดสรรแบนด์วิดธ์ทันทีที่มีแบนด์วิดธ์ว่าง และการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา (อธิบายในหัวข้อที่ 3.1.3) เมื่อถึงเวลาที่จะจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตที่อยู่ในคิวรอการเชื่อมต่อแล้ว ส่วนควบคุมการจัดตั้งการ

เชื่อมต่อจะทำการส่งแพ็กเกตเตือนไปให้แก่ผู้ร้องขอที่ได้รับการจัดสรร และอนุญาตให้ผู้ร้องขอสามารถทำการเชื่อมต่อกับเครือข่ายและส่งข้อมูลได้ต่อไป ตามรูปที่ 3.3

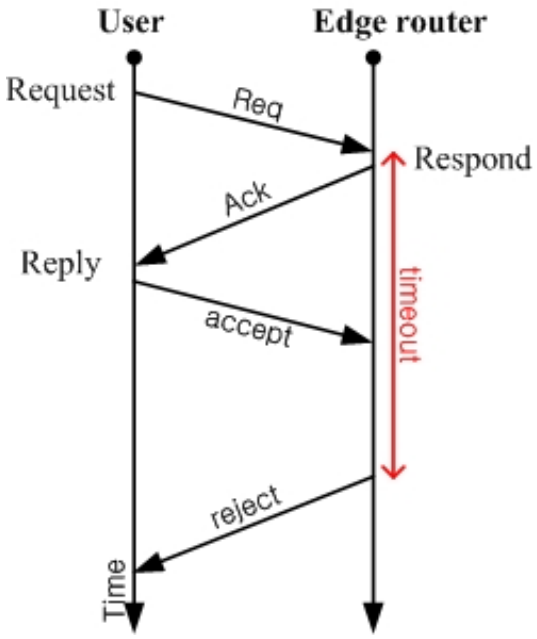


รูปที่ 3.3 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่ 1



รูปที่ 3.4 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่ 2

จากรูปที่ 3.4 จะเป็นการทำงานของส่วนควบคุมการจัดตั้งการเชื่อมต่อในกรณีที่มีการจัดสรรแบนด์วิดธ์ให้แก่ผู้ร้องขอเมื่อได้หน่วงแพ็กเกตร้องขอไว้ไม่นานและยังไม่ได้รับแพ็กเกตตอบรับจากผู้ใช้งาน โดยจะทำการส่งแพ็กเกตตอบยอมรับไปให้แก่ผู้ร้องขอตามไปทันทีและผู้ใช้งานสามารถทำการเชื่อมต่อได้ทันที ซึ่งในกรณีนี้จะใช้เวลาในการหน่วงแพ็กเกตในระยะเวลาที่สั้นกว่าในรูปที่ 3.3



รูปที่ 3.5 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่รอนานเกินกว่าเวลาในการรอคอย

ในรูปที่ 3.5 จะเป็นการทำงานของส่วนควบคุมการจัดตั้งการเชื่อมต่อเมื่อแพ็กเกตที่ถูกหน่วงเวลาไว้ในคิวรอการเชื่อมต่อนานเกินกว่าเวลาในการรอคอยที่ได้ตั้งไว้ ซึ่งเมื่อถึงเวลารอคอยที่ได้ตั้งไว้ส่วนควบคุมการจัดตั้งการเชื่อมต่อจะทำการลบแพ็กเกตที่ถูกหน่วงไว้ในคิวรอการเชื่อมต่อและส่งแพ็กเกตละทิ้งกลับไปยังผู้ร้องขอ ถ้าผู้ใช้งานยังคงต้องการที่จะติดต่อสื่อสารผ่านเครือข่ายก็จะต้องทำการสร้างแพ็กเกตร้องขอขึ้นมาใหม่แล้วส่งไปร้องขอยังเครือข่ายอีกครั้ง

3.1.3 การจัดสรรแบนด์วิดธ์

ในวิทยานิพนธ์นี้ได้ใช้การจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตร้องขอที่ถูกหน่วงเวลาตามที่ได้อธิบายในหัวข้อที่ผ่านมา ซึ่งมีอยู่ 2 แบบคือ การจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง และการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา

3.1.3.1 การจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง

หลังจากที่แพ็กเกตร้องขอได้ถูกหน่วงเวลาและตั้งเวลาในการรอคอยเอาไว้ในคิวรอการเชื่อมต่อแล้วก็จะต้องรอการจัดสรรแบนด์วิดธ์จากส่วนควบคุมการจัดตั้งการเชื่อมต่อ ซึ่งในวิธีนี้จะเป็นการรอคอยจนกว่าจะมีการเชื่อมต่อที่ใช้งานเสร็จแล้วและได้ปล่อยแบนด์วิดธ์กลับสู่เครือข่าย ต่อจากนั้นส่วนควบคุมการจัดตั้งการเชื่อมต่อก็จะทำการจัดสรรให้แก่แพ็กเกตร้องขอที่อยู่ในคิวรอ

การเชื่อมต่อตามลำดับการมาถึงก่อนหลัง (First Come First Served: FCFS) ในทันทีโดยจะส่งแพ็กเกตเตือนไปยังผู้ร้องขอให้สามารถทำการเชื่อมต่อกับเครือข่ายได้ทันที

3.1.3.2 การจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา

ในการจัดสรรแบนด์วิดธ์แบบโดยทันทีที่มีการปลดปล่อยกลับมาสู้เครือข่ายยังมีความไม่เสมอภาคเกิดขึ้น เนื่องจากบางแพ็กเกตมีความต้องการแบนด์วิดธ์จำนวนมากกว่าแพ็กเกตอื่นในการเชื่อมต่อ จึงถูกปฏิเสธการร้องขอจากเครือข่ายอยู่เรื่อย ๆ ในวิทยานิพนธ์นี้จึงได้คิดวิธีที่จะให้มีการรวบรวมแบนด์วิดธ์ในช่วงเวลาหนึ่งก่อนที่จะทำการจัดสรรให้แก่แพ็กเกตร้องขอที่ถูกหน่วงเวลาไว้ในคิวรอการเชื่อมต่อ ในวิธีนี้จะเป็นการนำอัลกอริทึมของการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างมาทำการปรับปรุงเพื่อเพิ่มประสิทธิภาพ โดยให้ส่วนควบคุมอนุมัติการเชื่อมต่อทำการจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตร้องขอตามลำดับการมาถึงก่อนหลังเป็นช่วงคาบเวลาต่ำสุดเพื่อสามารถรวบรวมแบนด์วิดธ์ได้จำนวนหนึ่งและมากพอที่จะสามารถจัดสรรให้แก่การร้องขอที่ต้องการใช้แบนด์วิดธ์จำนวนมาก

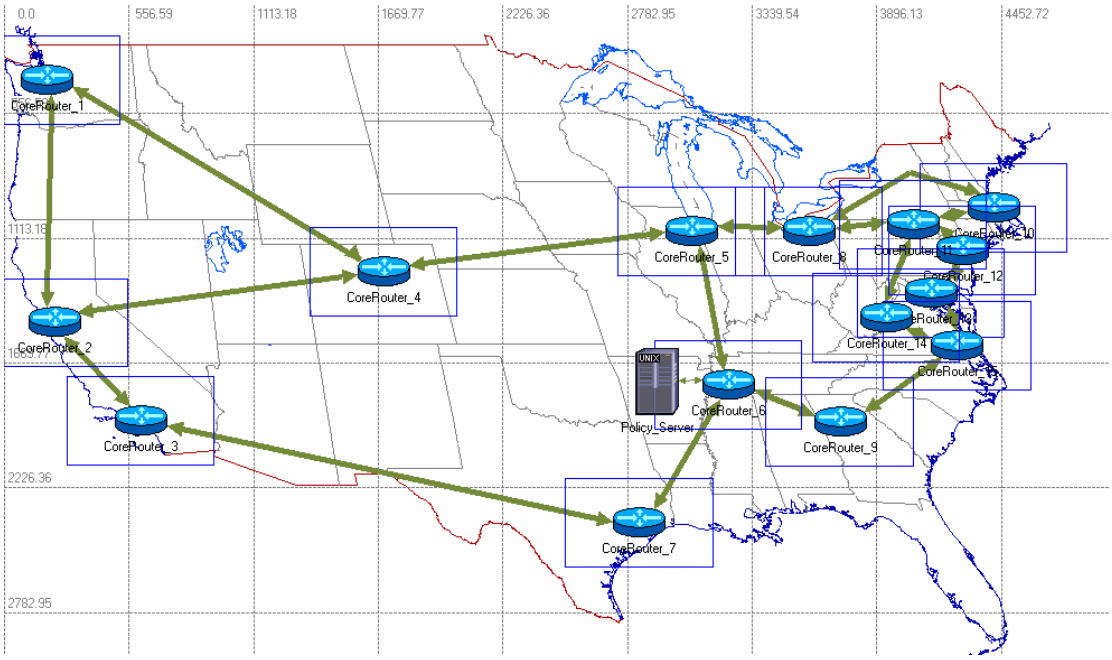
ตารางที่ 3.1 ผลการทดลองเมื่อใช้ช่วงคาบเวลาในการจัดสรรแบนด์วิดธ์

Period Time (sec) (Lifetime 240 sec)	Blocking Rate	Average Setup Time (sec)	CAC Signaling Traffic (Bytes)	Average Response Time (sec)
240	0.011524716	4.860347515	53127168	1.85401E-05
200	0.010054293	4.57568353	52984832	1.85401E-05
100	0.006195438	3.753727242	52917248	1.85401E-05
50	0.002935854	2.867760786	52942848	1.85401E-05
25	0.00229709	2.523446641	53152768	1.85401E-05
20	0.002573589	2.149222968	53075968	1.85401E-05
15	0.000884671	2.130281092	53174272	1.85401E-05
10	**0.00060367**	**1.793002075**	**53112832**	**1.85401E-05**
5	0.001005308	1.710762779	53335040	1.85401E-05
1	0.003058227	3.318515097	55015424	1.85414E-05
0.5	0.003740047	3.740515684	55499776	1.85403E-05

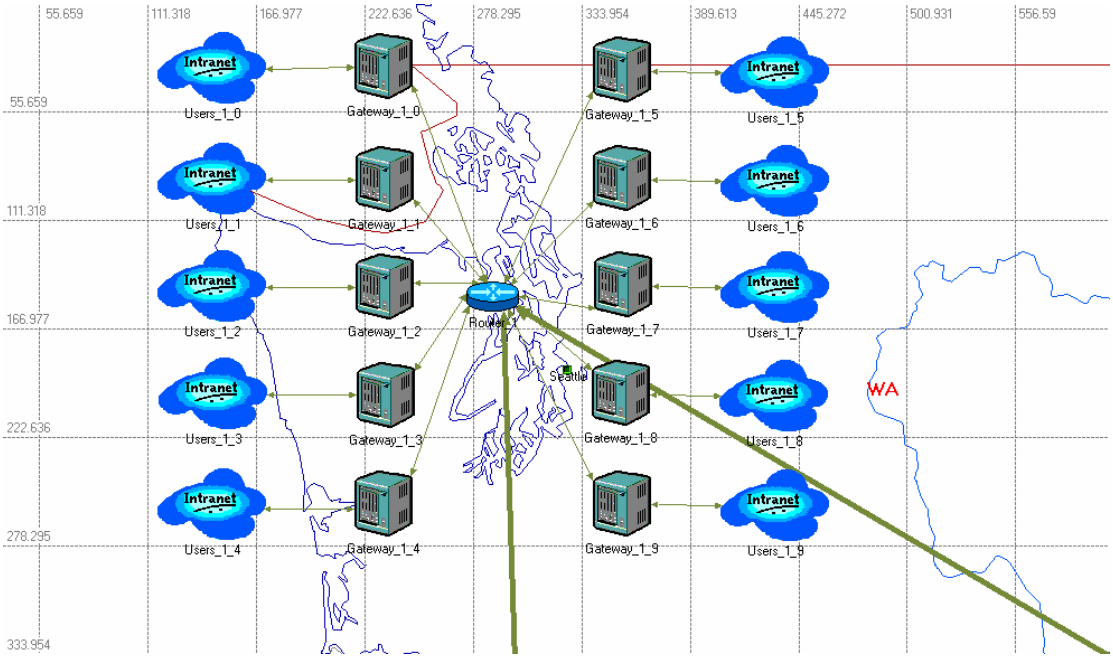
ซึ่งจากผลการทดลองในตารางที่ 3.1 แสดงให้เห็นว่าเมื่อได้ทำการเปลี่ยนช่วงคาบเวลาในการจัดสรรแบนด์วิดธ์ ทำให้อัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อและเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อมีการเปลี่ยนแปลง โดยในบางช่วงจะมีค่าที่เพิ่มขึ้นบางช่วงจะมีค่าที่ลดลง ในขณะที่ค่าพารามิเตอร์ทราฟฟิกที่เกิดจากสัญญาณการจัดตั้งการเชื่อมต่อมีค่าใกล้เคียงกัน ซึ่งค่าคาบเวลาที่มีอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อที่น้อยที่สุดและมีเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อที่ต่ำคือ ค่าคาบเวลาเท่ากับ 10 วินาที เนื่องจากวิทยานิพนธ์นี้ต้องการที่จะทดสอบประสิทธิภาพสูงสุดที่ได้จากการใช้วิธีการจัดตั้งการร้องขอแบบหน่วงเวลาการปฏิเสธ จึงได้เลือกใช้เวลาในการรอคอยสำหรับแพ็กเกตร้องขอที่ถูกหน่วงเวลาเท่ากับ 240 วินาที ซึ่งเป็นเวลาสูงสุดที่สามารถใช้ได้ในการจำลองการทำงานในระบบเครือข่าย

3.2 การสร้างระบบเครือข่ายบนโปรแกรมจำลองระบบเครือข่าย

ตามที่ได้อธิบายในบทที่ 2 ในวิทยานิพนธ์นี้ให้เลือกที่จะใช้ระบบเครือข่ายที่รับประกันคุณภาพในการให้บริการแบบ IntServ over DiffServ ของ Mei Yang [15] ขั้นแรกได้สร้างระบบเครือข่ายที่มีการทำงานแบบ IntServ over DiffServ ตามวิธีของ Mei Yang บนโปรแกรมจำลองระบบเครือข่าย OPNET ในระบบเครือข่ายตามรูปที่ 3.6 จะกำหนดให้มีเราเตอร์หลักทั้งหมด 15 ตัววางไว้ตามตำแหน่งและมีเซิร์ฟเวอร์ 1 ตัวซึ่งต่ออยู่กับเราเตอร์หลักตัวที่ 6



รูปที่ 3.6 เครือข่าย IntServ over DiffServ บนโปรแกรมจำลองระบบเครือข่าย [15]



รูปที่ 3.7 เราเตอร์หลักแต่ละตัวที่รองรับการเชื่อมต่อจากเราเตอร์ขอบและกลุ่มผู้ใช้งาน

ในรูปที่ 3.7 จะเป็นการเชื่อมต่อของเราเตอร์หลักแต่ละตัวที่จะรองรับการเชื่อมต่อจากเราเตอร์ขอบอีก 10 ตัว และที่เราเตอร์ขอบแต่ละตัวจะเชื่อมต่ออยู่กับกลุ่มผู้ใช้งาน ซึ่งกลุ่มผู้ใช้งานนี้จะเป็นส่วนที่ใช้ในการสร้างแพ็กเกตร้องขอและรับส่งข้อมูล

3.3 การสร้างโมเดลจำลองของอุปกรณ์ต่าง ๆ ในระบบเครือข่าย

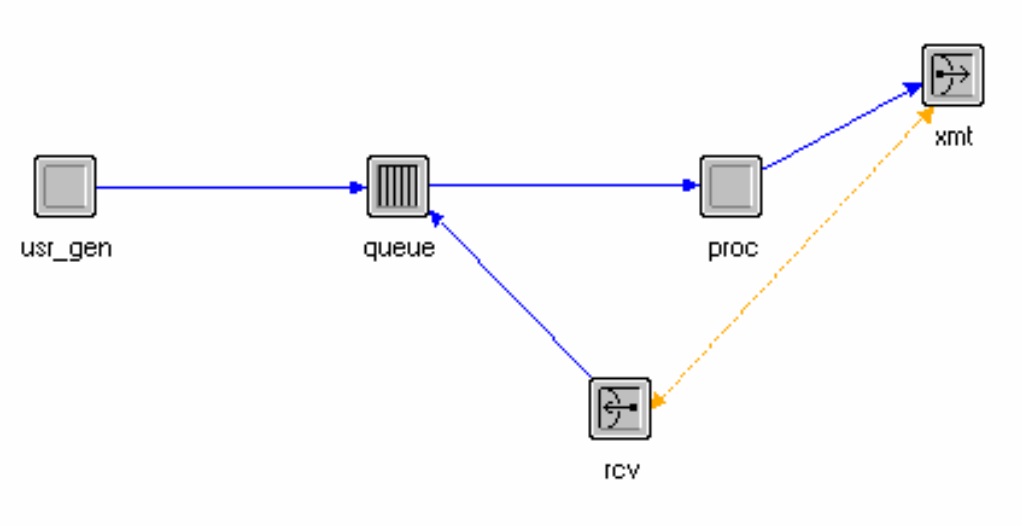
ในระบบเครือข่ายมีโมเดลจำลองของอุปกรณ์ต่าง ๆ คือ กลุ่มผู้ใช้งาน (LANs) เราเตอร์ขอบ (Edge Router) เราเตอร์หลัก (Core Router) ซึ่งในหัวข้อนี้จะอธิบายการทำงานของอุปกรณ์เหล่านี้

3.3.1 กลุ่มผู้ใช้งาน (LANs)

โมเดลของกลุ่มผู้ใช้งานจะเป็นส่วนที่สร้างแพ็กเกตซึ่งเปรียบเสมือนผู้ใช้งานต้นทางและปลายทาง โดยจะมีการทำงานหลัก ๆ คือ สร้างการร้องขอไปตามเราเตอร์ขอบต่าง ๆ โดยใช้วิธีการสุ่ม ซึ่งจะมีลักษณะตามภาพที่ 3.8 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.8 กลุ่มผู้ใช้งานในโปรแกรมจำลองระบบเครือข่าย

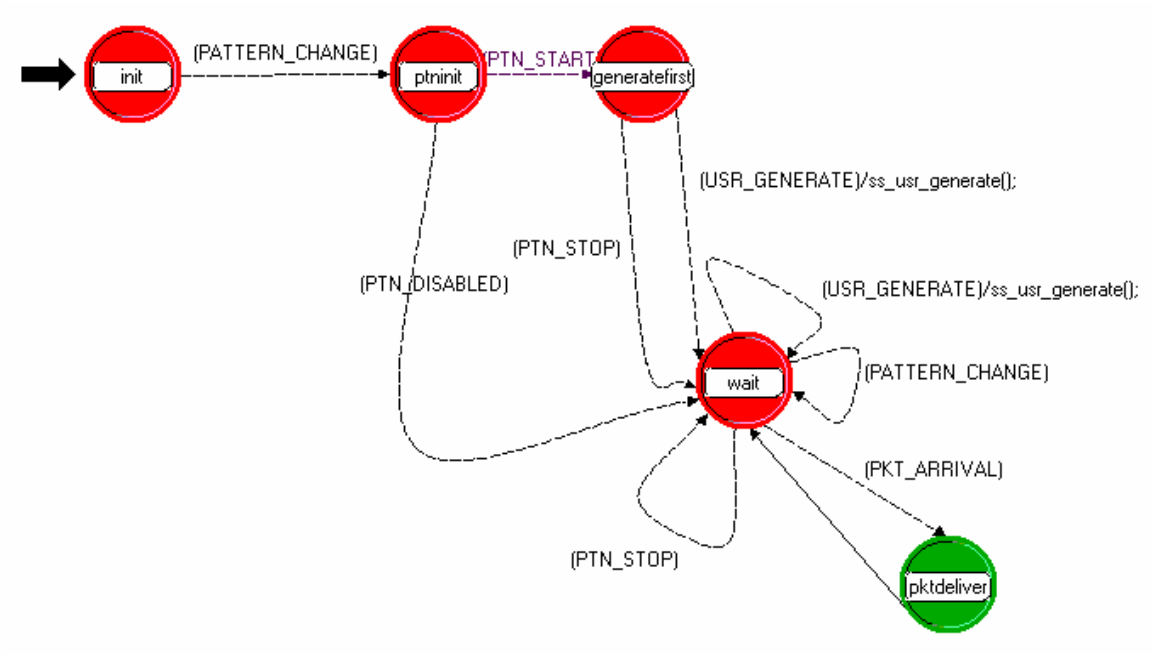
โดยภายในกลุ่มผู้ใช้งานนี้ จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่ง usr_gen จะทำงานเป็นตัวสร้างแพ็กเกตร้องขอขึ้นมาตามลักษณะทราฟฟิกที่จะใช้ในการทดลอง แล้วจะส่งต่อให้ queue เพื่อเป็นตัวในการจัดลำดับตามการมาถึงก่อนหลังก่อนที่จะนำไปประมวลผลที่ proc แล้วจะทำการบันทึกค่าต่าง ๆ ไว้ก่อนจะนำส่งแพ็กเกตออกไปจากโหนดกลุ่มผู้ใช้งานทางพอร์ต xmt ซึ่งในโหนดกลุ่มผู้ใช้งานนี้จะสามารถรับแพ็กเกตข้อมูลได้ผ่านทางพอร์ต rcv แสดงดังรูปที่ 3.9



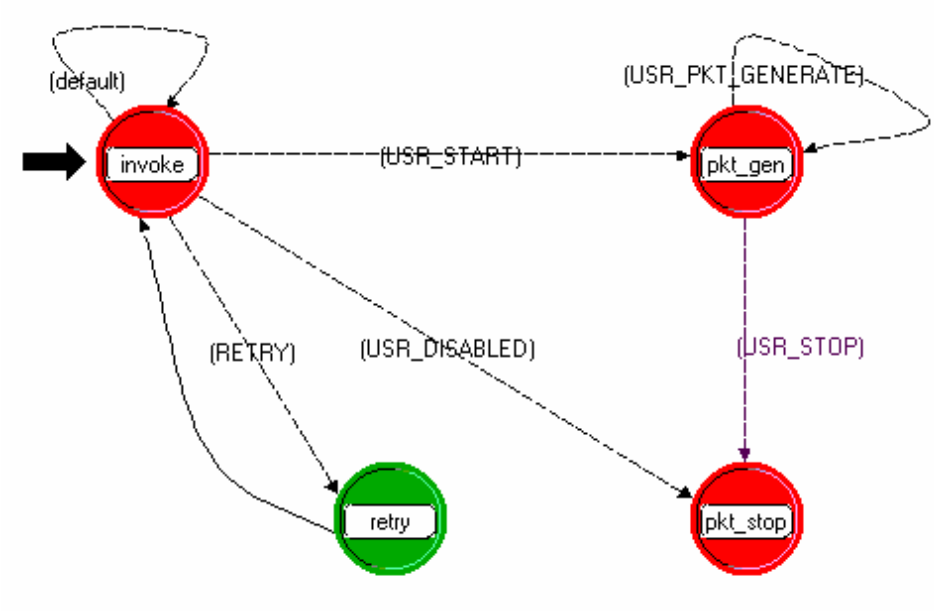
รูปที่ 3.9 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในกลุ่มผู้ใช้งาน

3.3.1.1 การทำงานของโมดูล usr_gen

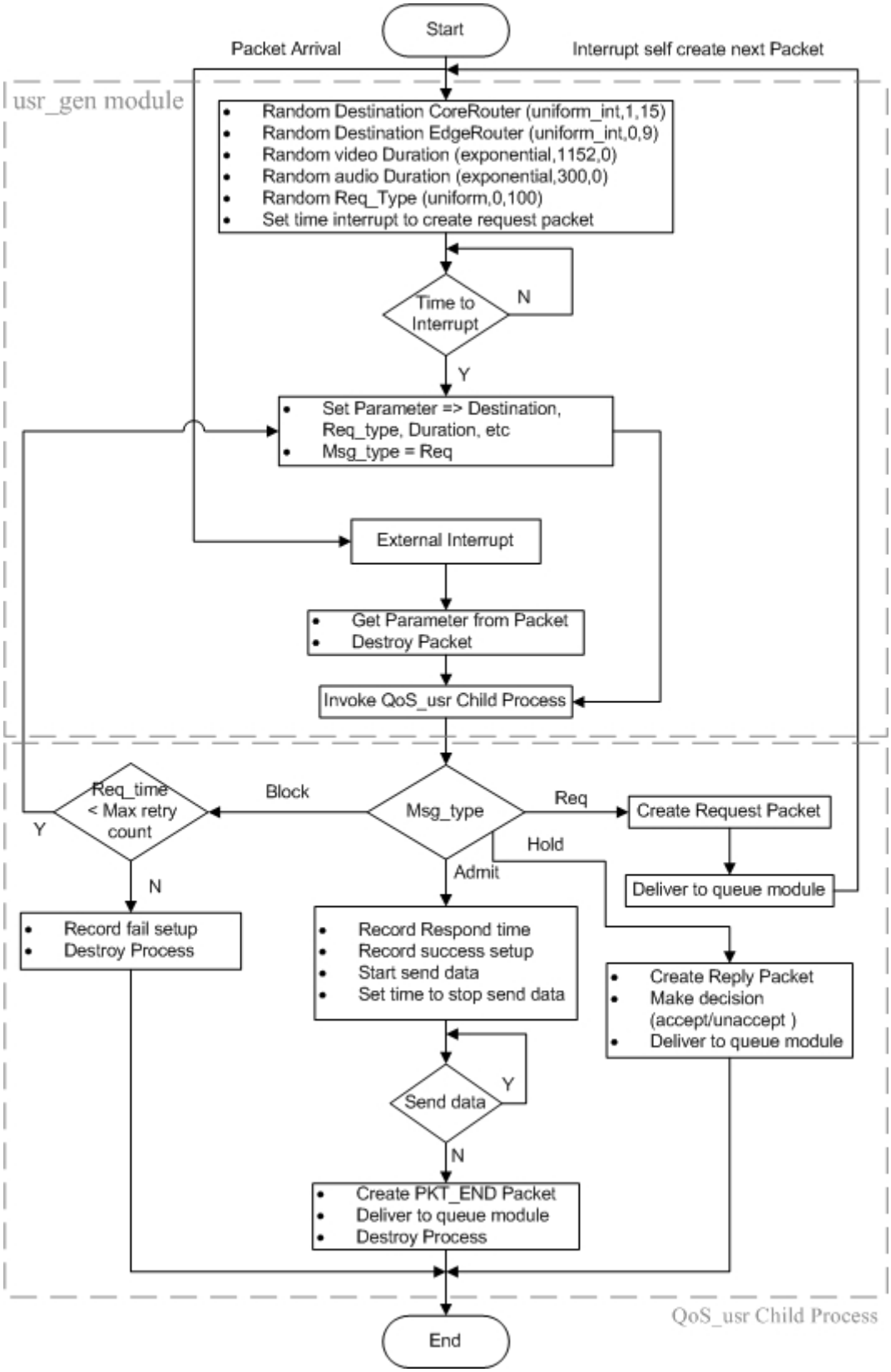
ในส่วนนี้จะทำงานเปรียบเสมือนเป็นผู้ใช้งานหลาย ๆ คนที่มีความต้องการที่จะใช้งานผ่านเครือข่าย โดยจะสร้างแพ็กเกตร้องขอขึ้นมาแล้วส่งไปยังเครือข่ายซึ่งความต้องการในการใช้ประเภทของทราฟฟิกจะมีการสุ่มขึ้นมา โดยภายในโมดูล usr_gen จะมีโครงสร้างการทำงานของ state ภายในดังรูปที่ 3.10 3.11 และลักษณะการทำงานดังรูปที่ 3.12



รูปที่ 3.10 โครงสร้างการทำงานของ state ภายในโมดูล usr_gen



รูปที่ 3.11 โครงสร้างการทำงานของ QoS_usr Child Process ที่ถูกเรียกขึ้นมาใช้งาน

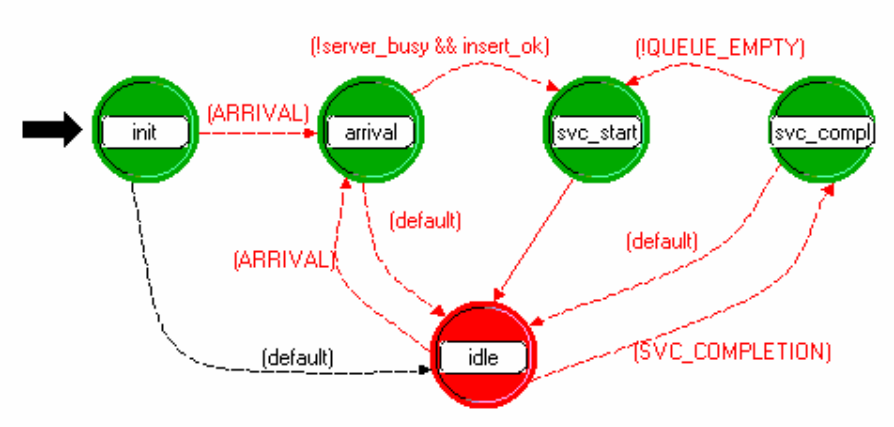


รูปที่ 3.12 ลักษณะการทำงานของโมดูล usr_gen

ส่วนโมดูล usr_gen จะเป็นกระบวนการภายในโหนดของกลุ่มผู้ใช้งานต้นทางและปลายทางมีโครงสร้างดังรูปที่ 3.10 ในขั้นแรกจะทำการสุ่มเลือกเราเตอร์ขอบและเราเตอร์หลักปลายทาง ต่อมาก็จะทำการสุ่มค่าระยะเวลาในการใช้งานของทราฟฟิกประเภทวิดิทัศน์จะใช้วิธีการสุ่มที่มีการกระจายตัวแบบเอกซ์โปเนนเชียลมีค่าเฉลี่ยเท่ากับ 1,152 วินาที (exponential, 1152, 0) และทราฟฟิกประเภทเสียงจะใช้วิธีการสุ่มที่มีการกระจายตัวแบบมีค่าเฉลี่ยเท่ากับ 300 วินาที (exponential, 300, 0) แล้วเก็บในค่าตัวแปรรอการเรียกใช้งาน ต่อมาก็จะสุ่มเลือกประเภทของทราฟฟิกและทำการตั้งค่าเวลาอินเทอร์รัปต์ (interrupt) เพื่อสร้างแพ็กเกตที่ใช้ในการร้องขอจัดตั้งเส้นทาง เมื่อถึงเวลาอิน-เทอร์รัปต์แล้วก็จะตั้งค่าตัวแปรต่าง ๆ ที่จำเป็นจากการสุ่มแล้วเรียกกระบวนการ (invoke process) QoS_usr เพื่อใช้ในการสร้างแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อขึ้นมา แล้วจะส่งต่อไปยังโมดูล queue แล้วจะไปรออยู่ในสถานะรอคอยการเข้ามาของแพ็กเกต ซึ่งใน QoS_usr Child Process จะถูกเรียกขึ้นมาทำงานอีกครั้งเมื่อมีแพ็กเกตถูกส่งเข้ามาที่โมดูล usr_gen และจะแยกไปทำงานตามประเภทของแพ็กเกตซึ่งมีแพ็กเกตร้องขอ (Req) แพ็กเกตอนุมัติ (Admit) แพ็กเกตปฏิเสธ (Block) และแพ็กเกตหน่วงเวลาการปฏิเสธ (Hold) ในกระบวนการสร้างแพ็กเกตร้องขอจะทำการสร้างแพ็กเกตร้องขอใหม่ขึ้น แล้วใส่ข้อมูลที่จำเป็นลงในแพ็กเกตตามที่ได้สุ่มขึ้นมาและส่งไปยังเราเตอร์ขอบเพื่อทำการร้องขอจัดตั้งการเชื่อมต่อ ถ้าได้รับแพ็กเกตอนุมัติมาจะทำการสร้างแพ็กเกตตอบสนอง (Respond) และแพ็กเกตเวลาในการจัดตั้งการเชื่อมต่อ (Setup time) แล้วส่งไปยังโมดูล proc เพื่อทำการบันทึกค่าสถิติ ต่อมาจะเริ่มทำการส่งข้อมูลไปยังปลายทางโดยได้ตั้งอินเทอร์รัปต์ไว้ตามเวลาที่ได้ขอ เมื่อส่งข้อมูลเสร็จสมบูรณ์แล้วจะทำการสร้างแพ็กเกตหยุดการเชื่อมต่อ (PKT_END) เพื่อส่งไปบอกถึงการหยุดส่งข้อมูลแล้วจะทำการปล่อยแบนด์วิดธ์กลับสู่เครือข่าย และจะทำการทำลายโพรเซส (process) นั้นทิ้ง ถ้าได้รับแพ็กเกตปฏิเสธมาจะทำการเปรียบเทียบจำนวนครั้งที่ได้ร้องขอไปแล้วกับจำนวนครั้งสูงสุดในการร้องขอที่กำหนดเอาไว้ ถ้ามีค่าน้อยกว่าก็จะทำตั้งค่าอินเทอร์รัปต์เพื่อส่งแพ็กเกตร้องขอครั้งต่อไป ถ้ามีค่ามากกว่าจะทำการสร้างแพ็กเกตเวลาในการจัดตั้งการเชื่อมต่อ แล้วส่งออกไปยังโมดูล proc ต่อมาก็จะทำการทำลายโพรเซสนั้นทิ้ง ถ้าได้รับแพ็กเกตหน่วงเวลาการปฏิเสธมาจะทำการสร้างแพ็กเกตตอบรับการหน่วงเวลาแล้วทำการตัดสินใจที่จะยอมรับการหน่วงเวลานั้นหรือไม่และจะส่งกลับไปยังเราเตอร์ขอบเพื่อให้ส่วนควบคุมการจัดตั้งการเชื่อมต่อทราบผลการตัดสินใจของผู้ใช้งาน

3.3.1.2 การทำงานของโมดูล queue

ในส่วนนี้จะเป็นจุดแรกที่แพ็กเกตที่เข้ามายังโหนดต่าง ๆ จะต้องผ่านแถวคอยเพื่อทำการจัดระเบียบก่อนที่จะไปทำการประมวลผลในส่วนถัดไป ซึ่งภายในโมดูล queue จะมีโครงสร้างการทำงานของ state ภายในดังรูปที่ 3.13



รูปที่ 3.13 โครงสร้างการทำงานของ state ภายในโมดูล queue

โมดูล queue จะทำหน้าที่เป็นแถวคอย (queueing) โดยจะมีเวลารอในแถวคอยเท่ากับสัดส่วนขนาดของแพ็กเกต (Packet Length) ต่ออัตราเร็วในการให้บริการข้อมูล (Service rate)

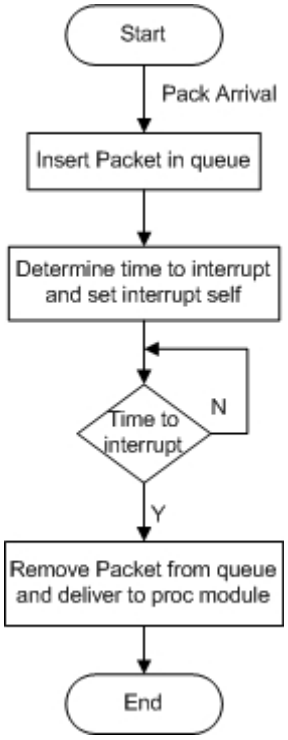
$$t_w = \frac{L_p}{t_x} \tag{3.1}$$

t_w เวลาที่รอในแถวคอย

L_p ขนาดของแพ็กเกต

t_x อัตราเร็วในการให้บริการข้อมูล

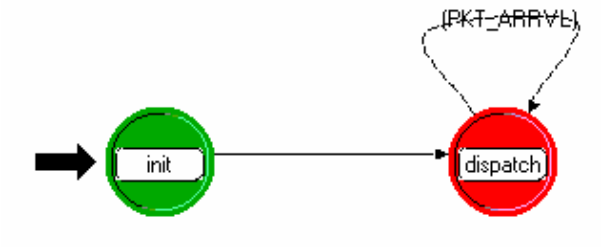
โมดูล queue จะอยู่ในโหนดต่าง ๆ ทุกโหนดในระบบเครือข่าย ซึ่งเมื่อแพ็กเกตถูกส่งมายังโหนดแล้ว แพ็กเกตทุกประเภทจะต้องรออยู่ในแถวคอยซึ่งเป็นเหมือนบัฟเฟอร์ขาเข้าที่เก็บแพ็กเกตไว้รอการบริการแพ็กเกตที่อยู่ในโมดูล proc อยู่ในขณะนี้ให้ได้รับการบริการที่เรียบร้อยก่อนจึงจะสามารถส่งแพ็กเกตที่อยู่ในคิวต่อไปไปยังโมดูล proc ได้ ซึ่งลักษณะการทำงานของโมดูล queue จะเป็นดังรูปที่ 3.14



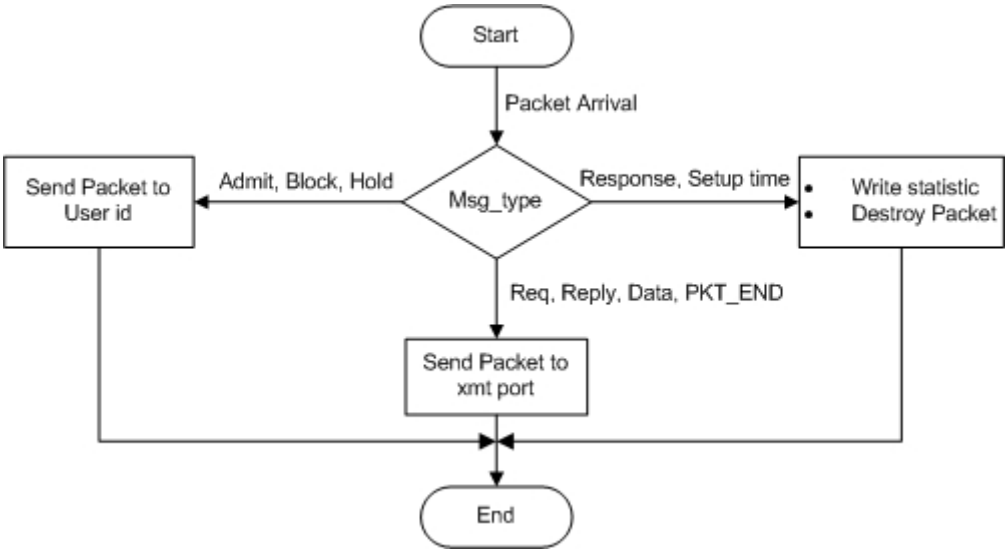
รูปที่ 3.14 ลักษณะการทำงานของโมดูล queue

3.3.1.3 การทำงานของโมดูล proc

ในโมดูล proc ที่อยู่ในกลุ่มผู้ใช้งานนี้จะทำหน้าที่เป็นตัวรับแพ็กเกตจากส่วนที่สร้างแพ็ก-เกตขึ้นมา แล้วส่งแพ็กเกตเข้าไปยังเครือข่ายที่เราเตอร์ขอบขาเข้า และหน้าที่อีกประการหนึ่งของส่วนนี้คือ ทำหน้าที่บันทึกสถิติต่าง ๆ ในการร้องขอจัดตั้งเส้นทางด้วย โดยจะมีโครงสร้างการทำงานของ state ภายในโมดูล proc ดังรูปที่ 3.15



รูปที่ 3.15 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.16 ลักษณะการทำงานของโมดูล proc

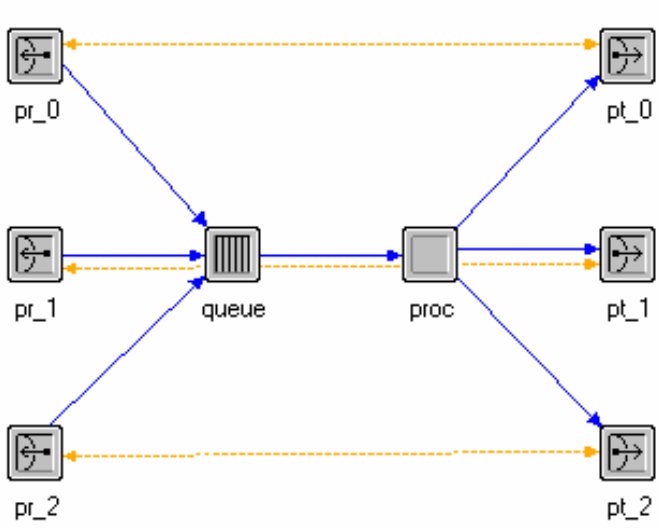
จากรูปที่ 3.16 ในเริ่มแรกกระบวนการทำงานจะไปรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะมีการตรวจสอบประเภทของแพ็กเกตโดยจะแบ่งการทำงานเป็น 3 กลุ่ม ในกลุ่มที่หนึ่งจะประกอบไปด้วยแพ็กเกต Req, Reply, data, PKT_END ซึ่งเมื่อได้รับแพ็กเกตมาจะทำงานโดยการส่งแพ็กเกตต่อไปยังปลายทางโดยผ่านพอร์ต xmt ในกลุ่มที่สองจะประกอบไปด้วยแพ็กเกต Response, Set_up ซึ่งเมื่อได้รับแพ็กเกตมาจะทำการบันทึกสถิติที่ส่งมาด้วยกับข้อมูลในแพ็กเกต และในกลุ่มที่สามจะประกอบไปด้วยแพ็กเกต Admit, Block, Hold ซึ่งเมื่อได้รับแพ็กเกตมาก็จะทำการส่งแพ็กเกตต่อไปยังผู้ใช้งาน

3.3.2 เราเตอร์ขอบ (Edge Router)

โมเดลจำลองของเราเตอร์ขอบนี้จะเป็นส่วนที่จะตัดสินใจที่จะรับหรือไม่รับการร้องขอที่มาจากผู้ใช้งานและทำการจัดการแบนด์วิดธ์ที่ใช้ในการส่งข้อมูล ซึ่งจะมีลักษณะตามรูปที่ 3.17 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.17 เราเตอร์ขอบในโปรแกรมจำลองระบบเครือข่าย

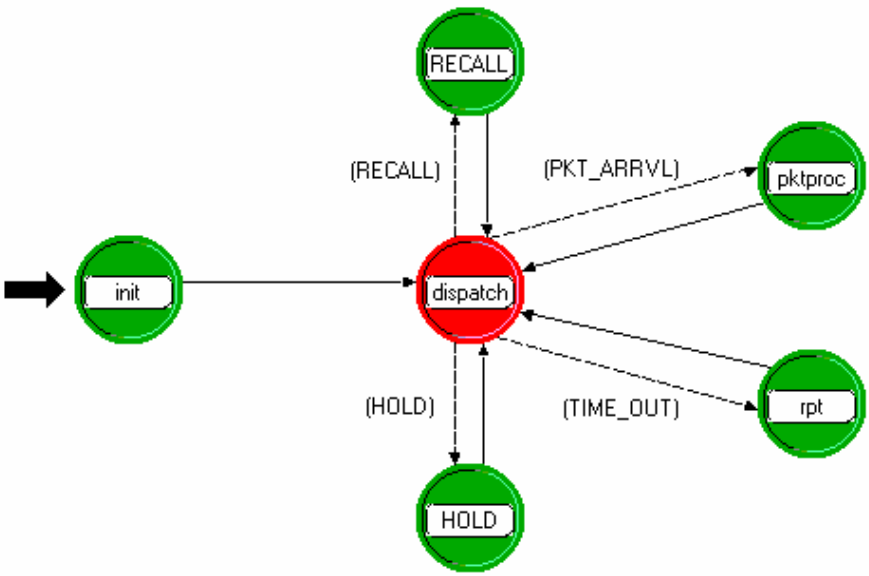
โดยภายในโหนดเราเตอร์ขอบนี้ จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่งจะมีพอร์ตรับและพอร์ตส่งจำนวน 3 ตัว เท่ากันซึ่งพอร์ตรับจะเชื่อมอยู่กับโมดูล queue จะมีหน้าที่รับแพ็กเกตเข้ามาในโหนด ตามรูปที่ 3.18 โดยการทำงานของโมดูล queue จะมีการทำงานเหมือนกับโมดูล queue ในโหนดกลุ่มผู้ใช้งาน และได้อธิบายไว้แล้วในหัวข้อที่ 3.3.1.2



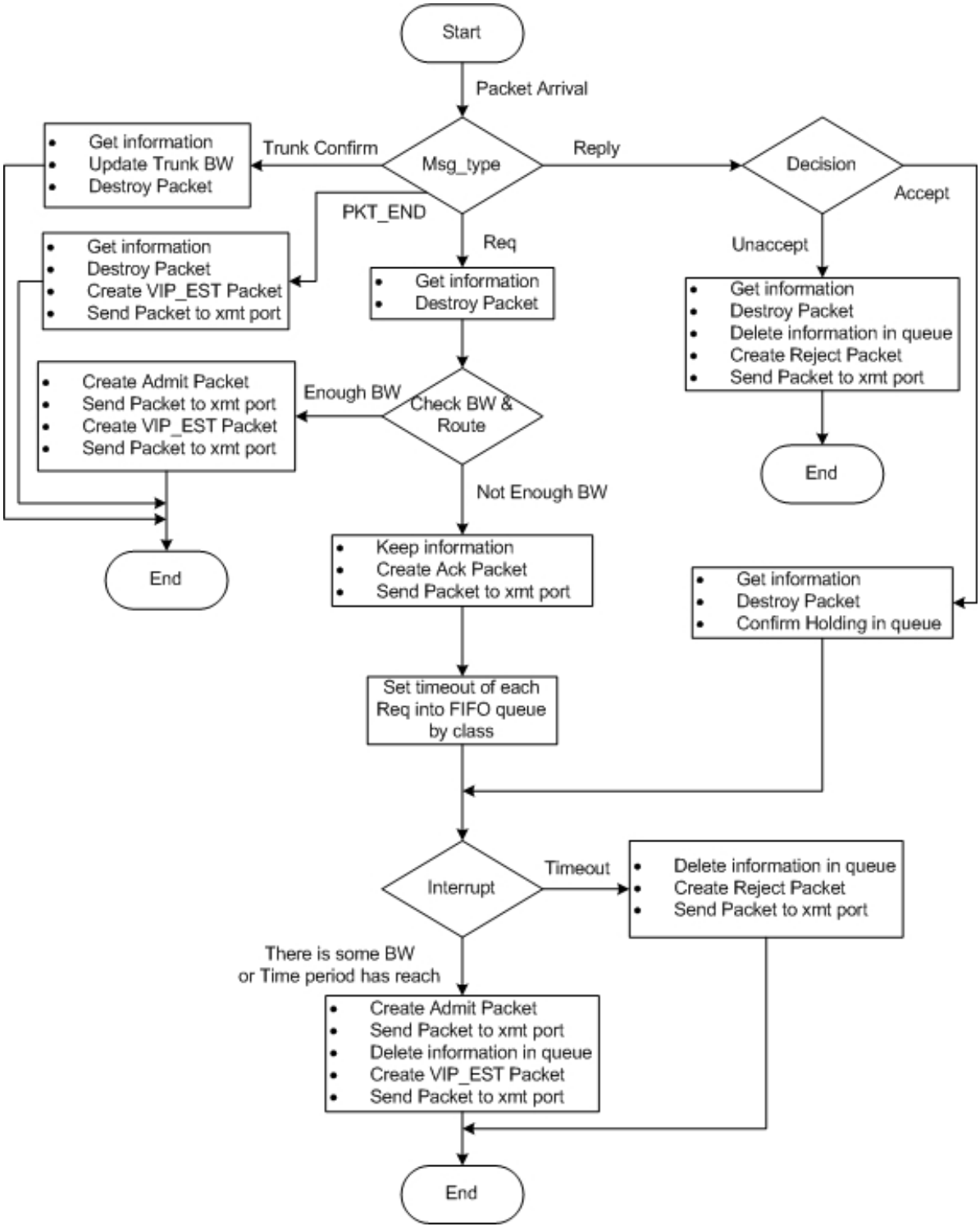
รูปที่ 3.18 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเราเตอร์ขอบ

3.3.2.1 การทำงานของโมดูล proc

ส่วนโมดูล proc ที่อยู่ในเราเตอร์ขอบนี้จะเป็นส่วนที่ควบคุมการร้องขอทั้งหมดที่เข้ามา หรือที่เรียกว่า ส่วนควบคุมการจัดตั้งการเชื่อมต่อ (Call Admission Control: CAC) ซึ่งจะควบคุมจำนวนผู้ที่จะเข้ามาใช้งานเครือข่าย จำนวนแบนด์วิดธ์ การรับประกันคุณภาพในการให้บริการ การรับหรือไม่รับการร้องขอ โดยจะมีโครงสร้างการทำงานของ state ต่าง ๆ ภายในโมดูล proc ตามรูปที่ 3.19



รูปที่ 3.19 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.20 ลักษณะการทำงานของโมดูล proc

เริ่มแรกกระบวนการทำงานจะไปรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะมีการตรวจสอบประเภทของแพ็กเกต ถ้าเป็นแพ็กเกต Trunk Confirm จะทำการรับข้อมูลจากแพ็กเกตแล้วนำข้อมูลนั้นไปใช้ในการปรับปรุงแบนด์วิดธ์ของ Trunk เมื่อเรียบร้อยแล้วจะทำลายแพ็กเกตทิ้ง ถ้า

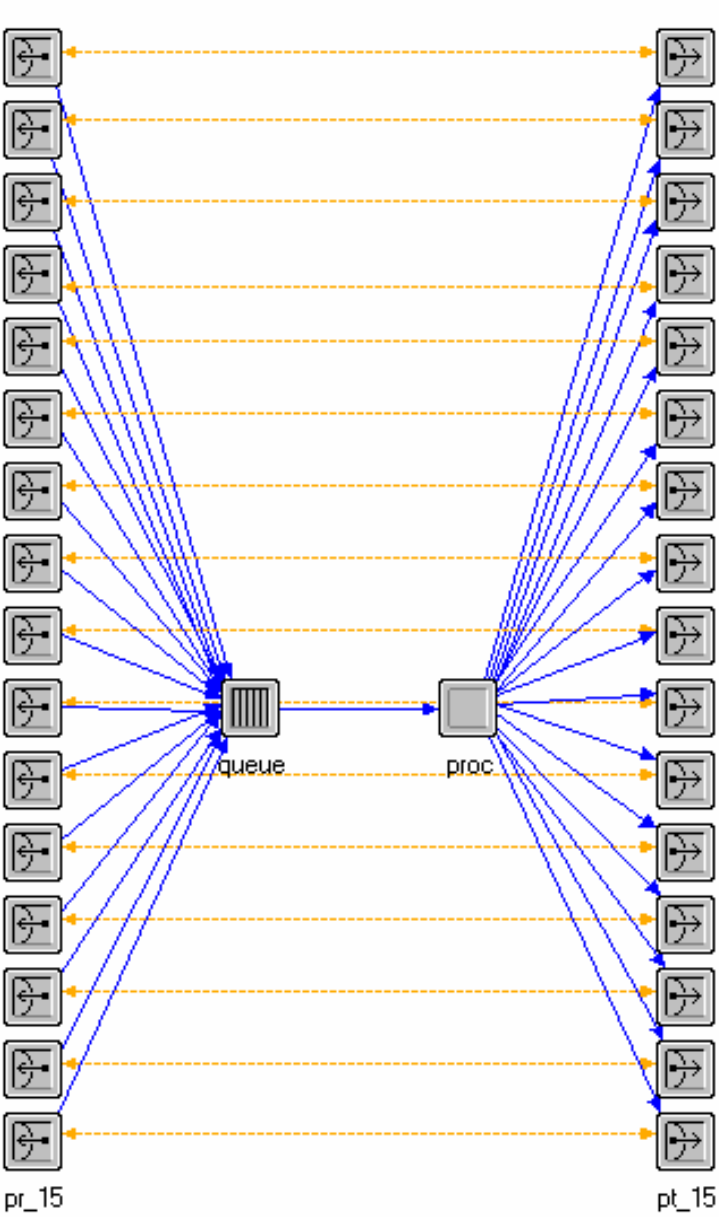
เป็นแพ็กเกต PKT_END จะทำกระบวนการรับข้อมูลจากแพ็กเกต แล้วจะทำการปล่อยแบนด์วิดธ์ที่เคยให้แก่ผู้ใช้งานนั้นกลับสู่เครือข่าย เมื่อเรียบร้อยแล้วจะทำลายแพ็กเกตนั้นทิ้ง ถ้าเป็นแพ็กเกต Reply จะมีการตรวจสอบการตัดสินใจที่ส่งมากับแพ็กเกต ถ้าตัดสินใจยอมรับการหน่วงเวลาจะทำการยืนยันการหน่วงเวลานั้นที่คิวรอการเชื่อมต่อตามข้อมูลที่ส่งมากับแพ็กเกตแต่ถ้าตัดสินใจไม่ยอมรับการหน่วงเวลาจะทำการลบแพ็กเกตที่ถูกหน่วงเวลาไว้ในคิวรอการเชื่อมต่อ และจะทำการสร้างแพ็กเกต Reject ส่งกลับไปยังผู้ใช้งาน ถ้าเป็นแพ็กเกตประเภท Req จะทำการรับข้อมูลจากแพ็กเกตไว้ แล้วทำการตรวจสอบแบนด์วิดธ์ตามที่ได้ร้องขอมากับแพ็กเกต ถ้ามีแบนด์วิดธ์เพียงพอจะทำการสร้างแพ็กเกต Admit ส่งกลับไปยังผู้ใช้งาน และแพ็กเกต VIP_EST ส่งไปยังเราเตอร์ต่าง ๆ ตามเส้นทางที่จะไปสู่ปลายทางเพื่อทำการสำรองทรัพยากร แต่ถ้ามีแบนด์วิดธ์ไม่เพียงพอที่จะให้ทำการเชื่อมต่อได้ก็จะเก็บข้อมูลของแพ็กเกตนั้นไว้แล้วสร้างแพ็กเกต Ack ตอบกลับไปยังผู้ใช้งาน ต่อมาก็จะตั้งเวลาในการรอคอยไว้ให้แก่แพ็กเกตแล้วเก็บแพ็กเกตไว้ในคิวรอการเชื่อม ต่อแบบมาถึงก่อนได้รับการบริการก่อน (First Come First Serve: FCFS) โดยแยกตามประเภทของบริการ แล้วก็จะมารอคอยการจัดสรรแบนด์วิดธ์ ในระหว่างการรอคอยถ้าเวลาในการรอคอยมาถึงแล้วก็จะทำการลบข้อมูลแพ็กเกตนั้นทิ้งไปจากคิวรอการเชื่อมต่อแล้วจะสร้างแพ็กเกต Reject ส่ง กลับไปยังผู้ใช้งาน แต่ถ้ามีแบนด์วิดธ์ว่างหรือเวลาในการจัดสรรมาถึงแล้ว (จัดสรรแบนด์วิดธ์เป็นช่วงคาบเวลา) ก็จะทำการสร้างแพ็กเกต Admit ส่งกลับไปยังผู้ใช้งานที่ได้รับการจัดสรร และสร้างแพ็กเกต VIP_EST ส่งไปสำรองแบนด์วิดธ์ที่เราเตอร์ต่าง ๆ จนถึงปลายทางและทำการลบข้อมูลออกจากคิวรอการเชื่อมต่อ

3.3.3 เราเตอร์หลัก (Core Router)

โมเดลจำลองของเราเตอร์หลักนี้จะทำหน้าที่เป็นส่วนในการสำรองแบนด์วิดธ์และส่งต่อแพ็ก-เกตข้อมูลไปยังเราเตอร์หลักปลายทางให้แก่ผู้ใช้งาน จะมีลักษณะตามภาพที่ 3.21 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.21 เราเตอร์หลักในโปรแกรมจำลองระบบเครือข่าย

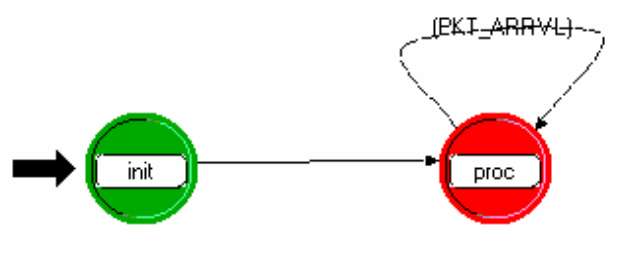
โดยภายในโหนดเราเตอร์หลักนี้จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่งจะมีพอร์ตรับและพอร์ตส่งจำนวน 16 ตัว เท่ากัน ซึ่งพอร์ตรับจะเชื่อมอยู่กับโมดูล queue ซึ่งจะรับแพ็กเกตเข้ามาในโหนดตามรูปที่ 3.22 โดยการทำงานของโมดูล queue จะมีการทำงานเหมือนกับโมดูล queue ในโหนดกลุ่มผู้ใช้งาน และได้อธิบายไว้แล้วในหัวข้อที่ 3.3.1.2



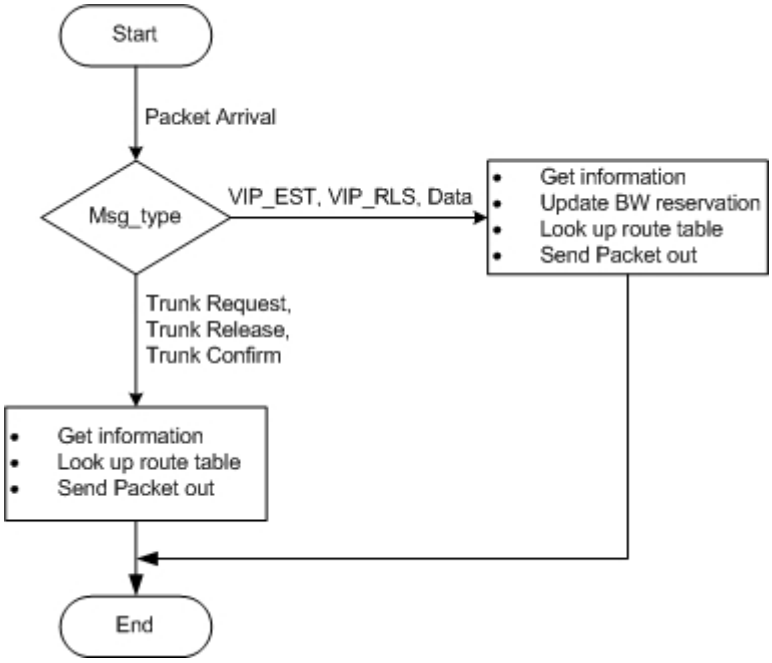
รูปที่ 3.22 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเราเตอร์หลัก

3.3.3.1 การทำงานของโมดูล proc

ในโมดูล proc ในเราเตอร์หลักนี้จะทำหน้าที่ในการส่งต่อแพ็กเกตข้อมูลไปยังปลายทาง และมีการจัดการสำรองแบนด์วิดธ์ตามลำดับชั้น (เช่น การสำรองแบนด์วิดธ์ให้แก่ provisioned link และ trunk) โดยจะมีโครงสร้างการทำงานของ state ต่าง ๆ ภายในโมดูล proc ตามรูปที่ 3.23



รูปที่ 3.23 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.24 ลักษณะการทำงานของโมดูล proc

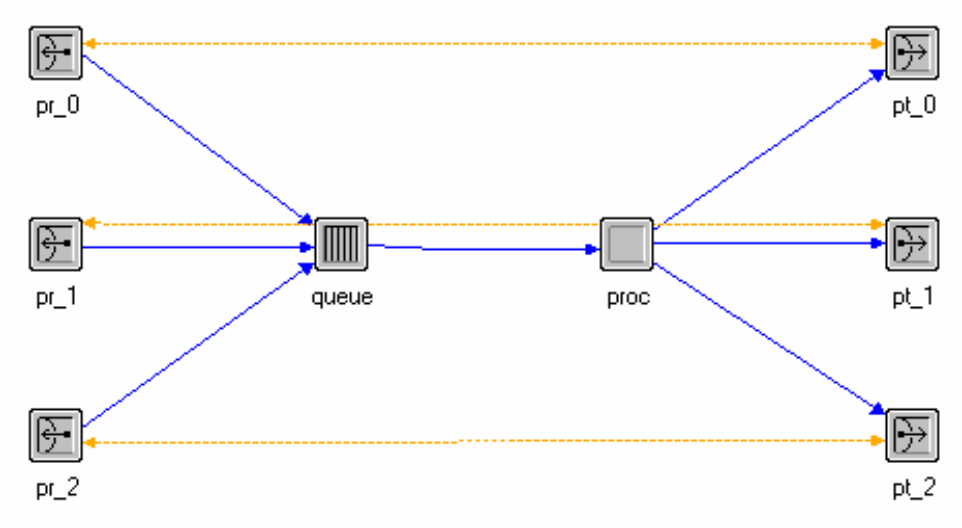
จากรูปที่ 3.24 เริ่มแรกกระบวนการทำงานจะไปรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะทำการตรวจสอบประเภทของแพ็กเกตซึ่งแบ่งออกเป็นสองกลุ่ม กลุ่มแรกจะมีแพ็กเกต VIP_EST, VIP_RLS ซึ่งจะทำการปรับปรุงการสำรองแบนด์วิดธ์ที่เปลี่ยนแปลง ถ้าเป็นแพ็กเกต data จะทำการส่งต่อไปจนถึงปลายทาง ในกลุ่มที่สองจะเป็นแพ็กเกต Trunk Request, Trunk Release, Trunk Confirm จะเป็นการส่งแพ็กเกตต่อไปจนถึงปลายทาง

3.3.4 เซิร์ฟเวอร์ (Server)

โมเดลจำลองของเซิร์ฟเวอร์นี้จะทำหน้าที่เป็นส่วนในการบันทึกการใช้งานแบนด์วิดธ์ทั้งหมดของเครือข่ายและเป็นส่วนกลางในการของใช้แบนด์วิดธ์จากส่วน provisioned link จะมีลักษณะตามภาพที่ 3.25 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.25 เซิร์ฟเวอร์ในโปรแกรมจำลองระบบเครือข่าย

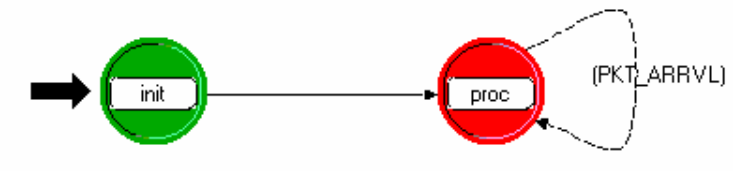
โดยภายในโหนดเซิร์ฟเวอร์นี้ จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่งจะมีพอร์ตรับและพอร์ตส่งจำนวน 3 ตัว เท่ากัน ซึ่งพอร์ตรับจะเชื่อมอยู่กับโมดูล queue ซึ่งจะรับแพ็กเกตเข้ามาในโหนดตามรูปที่ 3.26 โดยการทำงานของโมดูล queue จะมีการทำงานเหมือนกับโมดูล queue ในโหนดกลุ่มผู้ใช้งาน และได้อธิบายไว้แล้วในหัวข้อที่ 3.3.1.2



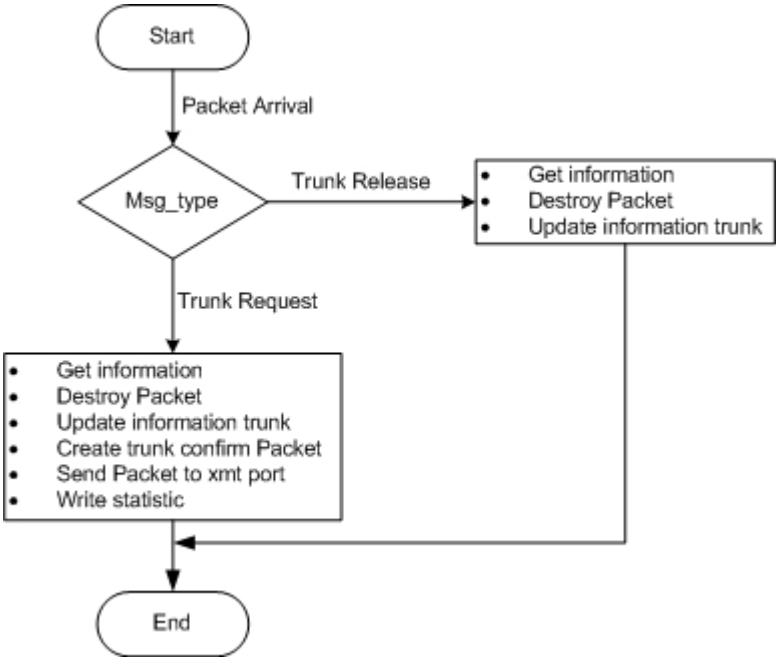
รูปที่ 3.26 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเซิร์ฟเวอร์

3.3.4.1 การทำงานของโมดูล proc

ในโมดูล proc ในเซิร์ฟเวอร์นี้จะทำหน้าที่ โดยจะมีโครงสร้างการทำงานของ state ต่าง ๆ ภายในโมดูล proc ตามรูปที่ 3.27



รูปที่ 3.27 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.28 ลักษณะการทำงานของโมดูล proc

จากรูปที่ 3.28 เริ่มแรกมาจะทำกระบวนการอ่านข้อมูลจากไฟล์ตาราง Provision มาเก็บไว้ในตัวแปรของโปรแกรม แล้วจะรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะทำการตรวจสอบประเภทของแพ็กเกต ถ้าเป็นแพ็กเกต Trunk Request เข้ามา จะทำการรับข้อมูลจากแพ็กเกตเพื่อดูว่ามีการขอขยายแบนด์วิดธ์ trunk เท่าไรและจะสามารถให้ได้หรือไม่ แล้วจะทำการสร้างแพ็กเกต Confirm เพื่อตอบกลับไปยังผู้ที่ส่งมา แล้วทำการบันทึกสถิติการใช้งานแบนด์วิดธ์ที่เปลี่ยนแปลง ถ้าเป็นแพ็กเกต Trunk Release เข้ามา จะทำการรับข้อมูลจากแพ็กเกตเพื่อที่จะดูว่ามีการปล่อยแบนด์วิดธ์กลับคืนสู่เครือข่ายเท่าไร แล้วจะทำการปรับปรุงจำนวนแบนด์วิดธ์ที่เหลืออยู่ทั้งหมด แล้วทำการบันทึกสถิติการใช้งานแบนด์วิดธ์ที่เปลี่ยนแปลง