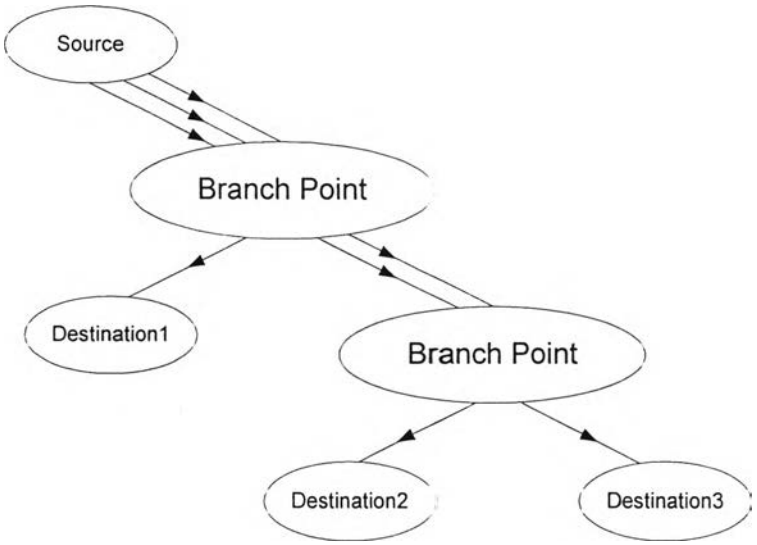
บทที่ 3
การส่งข้อมูลประเภท ABR จากหนึ่งไปจุดสู่หลายจุดในโครงข่ายเอทีเอ็ม

3.1 กล่าวนำ

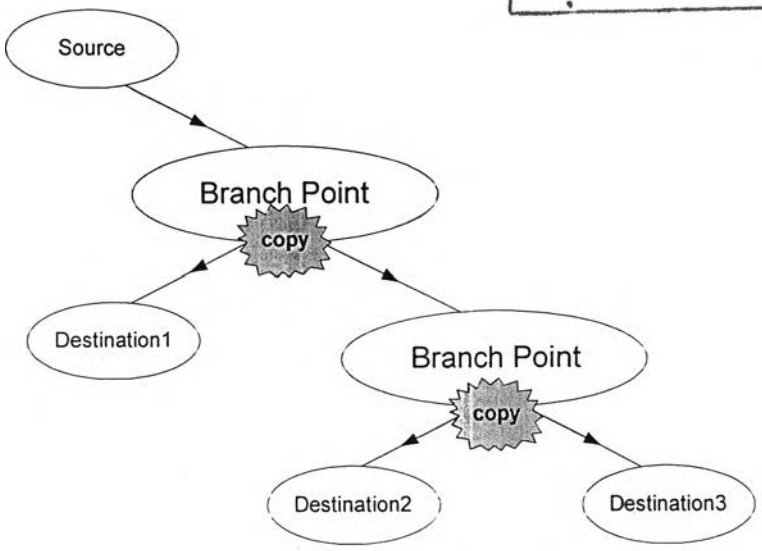
โครงข่ายเอทีเอ็มนั้น ถูกออกแบบมาเพื่อรองรับการเชื่อมต่อจากหนึ่งจุดไปสู่หลายจุด เพื่อรองรับการใช้งานหลายประเภท อาทิ เช่น video conferancing , การ multicast ใน LANE หรือ การสอนระยะทางไกล ฯลฯ การส่งแบบหนึ่งจุดไปสู่หลายจุดในโครงข่ายเอทีเอ็มนั้นต้องส่งข้อมูลเข้าสู่โครงข่ายโดยไม่เพิ่มภาระให้กับโครงข่ายโดยไม่จำเป็น ด้วยหลักการที่ว่า ผู้ส่งต้องไม่มีการส่งข้อมูลที่ซ้ำเดิมไปในเส้นทางเดียวกันไปสู่ปลายทางต่าง ๆ กัน ดังรูปที่ 3.1 ดังนั้นการคัดลอกเซลล์ที่สวิตช์ระหว่างทางจึงเป็นสิ่งจำเป็น เพื่อให้ได้ประสิทธิภาพในการส่งมากที่สุดดังรูปที่ 3.2 ด้วยเหตุนี้ แหล่งกำเนิดจำเป็นต้องทราบรูปแบบโครงข่ายเป็นอย่างดี เพื่อสามารถสร้าง Multicast tree สำหรับส่งข้อมูล ในที่นี้ Multicast tree นั้นประกอบด้วย root นั่นคือแหล่งกำเนิดนั่นเอง , branch point ได้แก่ อุปกรณ์สวิตช์ที่ทำหน้าที่คัดลอกข้อมูล, branch คือสายนำสัญญาณที่นำข้อมูลไปสู่ปลายทาง และ leave คือ อุปกรณ์ปลายทางที่มีหน้าที่รับข้อมูล



รูปที่ 3.1 การส่งแบบหนึ่งจุดไปสู่หลายจุดที่ไม่มีประสิทธิภาพ

สำหรับการให้บริการแบบ ABR นั้นต้องอาศัยการป้อนกลับข่าวสารความคับคั่ง ดังนั้นหากต้องการให้การส่งจากหนึ่งจุดไปสู่หลายจุดนั้นมีประสิทธิภาพจึงจำเป็นต้องมี การควบคุมการส่งแบบหนึ่งจุดไปสู่หลายจุดสำหรับการให้บริการแบบ ABR

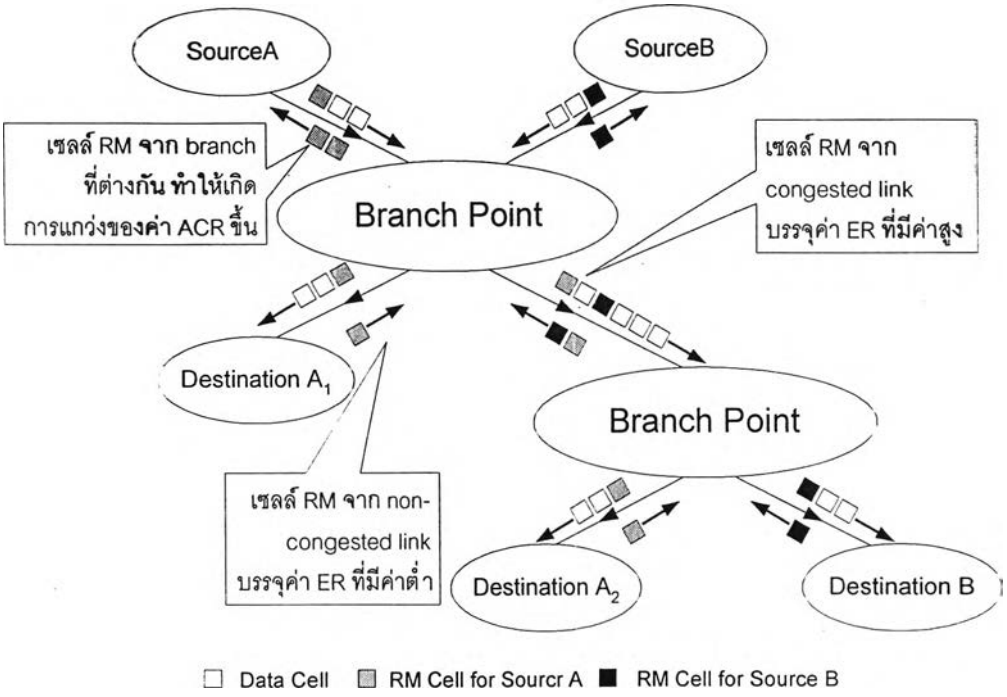
หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.2 การส่งแบบหนึ่งจุดไปสู่หลายจุดที่มีประสิทธิภาพ

3.2 การควบคุมการส่งแบบหนึ่งจุดไปสู่หลายจุดสำหรับการให้บริการแบบ ABR

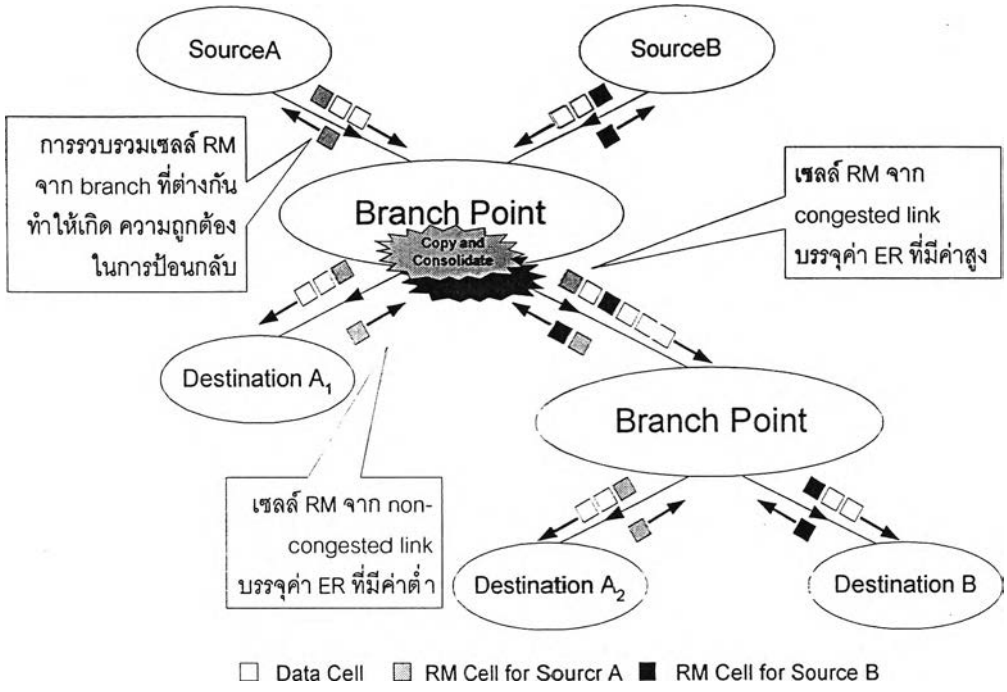
หลักการการควบคุมการส่งแบบหนึ่งจุดไปสู่หลายจุด (Point-to-multipoint connection) นั้นได้รับการพัฒนามาจาก การส่งแบบหนึ่งจุดไปสู่หนึ่งจุด โดยนำวิธีควบคุมความคับคั่งที่ใช้ในการส่งแบบหนึ่งจุดไปสู่หนึ่งจุด มาใช้ โดยการให้สวิตช์นั้นเป็นผู้คัดลอกเซลล์ต่าง ๆ ออกไปตาม branch ต่าง ๆ อย่างไรก็ตามการรวบรวมข่าวสารความคับคั่งที่ป้อนกลับจาก branch ที่ต่างกัน สำหรับแต่ละ branch point เป็นเรื่องสำคัญ เนื่องจากข่าวสารที่ป้อนกลับไปหาแหล่งกำเนิดไม่ควรจะเพิ่มขึ้นตามจำนวนของปลายทางสำหรับการเชื่อมต่อนั้น ปัญหาดังกล่าวเรียกว่า feedback implosion โดยสามารถแสดงได้ดังรูปที่ 3.3 จะเห็นว่าเซลล์ FRM ที่สวิตช์ทำหน้าที่คัดลอกไปสู่ปลายทางที่ต่างกันนั้น ล้วนถูกตอบกลับมาจากปลายทางต่าง ๆ และหากสวิตช์นั้นปล่อยให้เซลล์ BRM นั้นไปสู่แหล่งกำเนิดนั้น จะทำให้ค่า ACR นั้นเกิดการแกว่ง เนื่องจากค่าข่าวสารความคับคั่งที่ป้อนกลับจากแต่ละ branch นั้น ไม่เหมือนกัน ทำให้แหล่งกำเนิดได้ข่าวสารที่ไม่แน่นอน และ ค่าอัตราส่งที่แหล่งกำเนิดนั้นจะเกิดการแกว่งขึ้นมา โดยปัญหาที่เกิดขึ้นนี้เรียกว่าเกิด consolidation noise ขึ้น และจะทำให้โครงข่ายเกิดความคับคั่งขึ้น อันจะทำให้เกิดการสูญเสียเซลล์ข้อมูลได้



รูปที่ 3.3 ความผิดพลาดจากการไม่รวบรวมข่าวสารการป้อนกลับ

วิธีการควบคุมความคับคั่งในการส่งจากจุดเดียวสู่จุดเดียวจะใช้การปรับอัตราส่งที่ต่ำสุด ที่สวิตช์ตลอดเส้นทางรองรับได้ ดังนั้นในการส่งจากหนึ่งจุดไปสู่หลายจุดนั้นควรปรับอัตราส่งให้เป็นค่าต่ำที่สุดสำหรับที่ทุกเส้นทางจากต้นทางสู่ปลายทางทุกอัน สามารถรองรับได้ ดังนั้น branch point จึงจำเป็นต้องมีวิธีการรวบรวมข้อมูลความคับคั่งจากเซลล์ BRM เพื่อให้ข้อมูลที่จะส่งกลับไปพร้อมเซลล์ RM มีข้อมูลความคับคั่งของโครงข่ายที่ดีที่สุดจากทุก branch วิธีการดังกล่าวเรียกว่า วิธีการรวบรวมข่าวสารการป้อนกลับ (consolidation algorithm) การรวบรวบข้อมูลความคับคั่งที่ดีควรทำให้อัตราส่งที่แหล่งกำเนิดมีค่าต่ำที่สุดที่สวิตช์ทุกตัวใน multicast tree สามารถรองรับได้ เพื่อป้องกันการเกิดความคับคั่งขึ้นและ พยายามลด consolidation noise ให้มากที่สุด

อย่างไรก็ตามการป้อนกลับที่ถูกต้องสู่แหล่งกำเนิดนั้นอาจนำไปสู่ การรอข่าวสารที่จำเป็นทั้งหมดของแต่ละ branch ซึ่งจะทำให้ความเร็วในการป้อนกลับข่าวสารดังกล่าวช้ามาก โดยเฉพาะในสภาวะที่โครงข่ายมีจำนวนเซลล์ RM น้อย ทำให้แหล่งกำเนิดไม่สามารถปรับอัตราส่งให้เข้าสู่อัตราส่งที่แหล่งกำเนิดสามารถรองรับได้อย่างแท้จริง และ อาจก่อให้เกิดปัญหาความคับคั่ง หรือ ปัญหาการใช้งานสายสัญญาณไม่คุ้มค่าขึ้นในโครงข่าย ปัญหาดังกล่าวเรียกว่าเกิด consolidation delay โดยวิธีรวบรวมข่าวสารความคับคั่งที่ดีนั้นต้องมีผลตอบสนองชั่วครู่ที่รวดเร็ว เพื่อลดปัญหา consolidation delay และทำให้แหล่งกำเนิดสามารถตอบสนองต่อสภาวะของโครงข่ายที่เปลี่ยนไปได้ การทำงานของ consolidation algorithm สามารถแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 การทำงานของวิธีรวบรวมข่าวสารความคับคั่ง

3.3 วิธีการรวบรวมข่าวสารความคับคั่ง

วิธีการรวบรวมข่าวสารการป้อนกลับมีหลายวิธีดังได้เสนอมาก่อนหน้านี้ในบทความ [8-11] หลักการในการพิจารณาวิธีการรวบรวมข้อมูลความคับคั่งนั้นจะพิจารณาจาก ความเร็วในการป้อนกลับ และ ความถูกต้องของข้อมูลที่ป้อนกลับ ซึ่งวิธีการที่เน้นความเร็วในการป้อนกลับนั้น จะมีเวลาในการป้อนกลับที่เร็วทำให้แหล่งกำเนิดสามารถปรับอัตราส่งให้ทันต่อสภาวะที่เปลี่ยนไปของโครงข่ายได้ แต่ก็ทำให้ branch point ไม่สามารถรวบรวมข่าวสารความคับคั่งจากทุก branch ก่อนการป้อนกลับ ทำให้เกิดการผิดพลาดในการป้อนกลับอันนำไปสู่การเกิด Consolidation noise ในทางกลับกันหากวิธีรวบรวมความคับคั่งนั้นเน้นความถูกต้องในการป้อนกลับ โดยการรวบรวมข่าวสารที่จำเป็นมากที่สุดก่อนการป้อนกลับนั้น ซึ่งในวิธีการแรก ๆ ที่เสนอนั้นจะพยายามให้ได้การป้อนกลับที่เร็วโดยที่ไม่สนใจความถูกต้องมากนัก แต่วิธีการใหม่ ๆ พยายามพัฒนาจากหลักการที่เน้นให้มีความถูกต้องของการป้อนกลับมากที่สุด และพัฒนาให้ได้การป้อนกลับที่เร็วขึ้นโดยไม่ต้องรอข้อมูลมากนัก อย่างไรก็ตาม จุดมุ่งหมายสำคัญของวิธีการรวบรวมข่าวสารความคับคั่งที่สำคัญคือ มีความถูกต้องในการป้อนกลับ ในขณะที่มีความรวดเร็วในการป้อนกลับ ซึ่งจะทำให้แหล่งกำเนิดสามารถตอบสนองต่อสภาวะที่เปลี่ยนไปของโครงข่ายได้อย่างทันท่วงที วิธีรวบรวมข่าวสารความคับคั่งที่พัฒนาขึ้นมานั้นมีมากมายหลายวิธี อาทิเช่น Robert algorithm, TS algorithm, RSS algorithm, Wait-for-all algorithm และ Immediate Rate Calculation

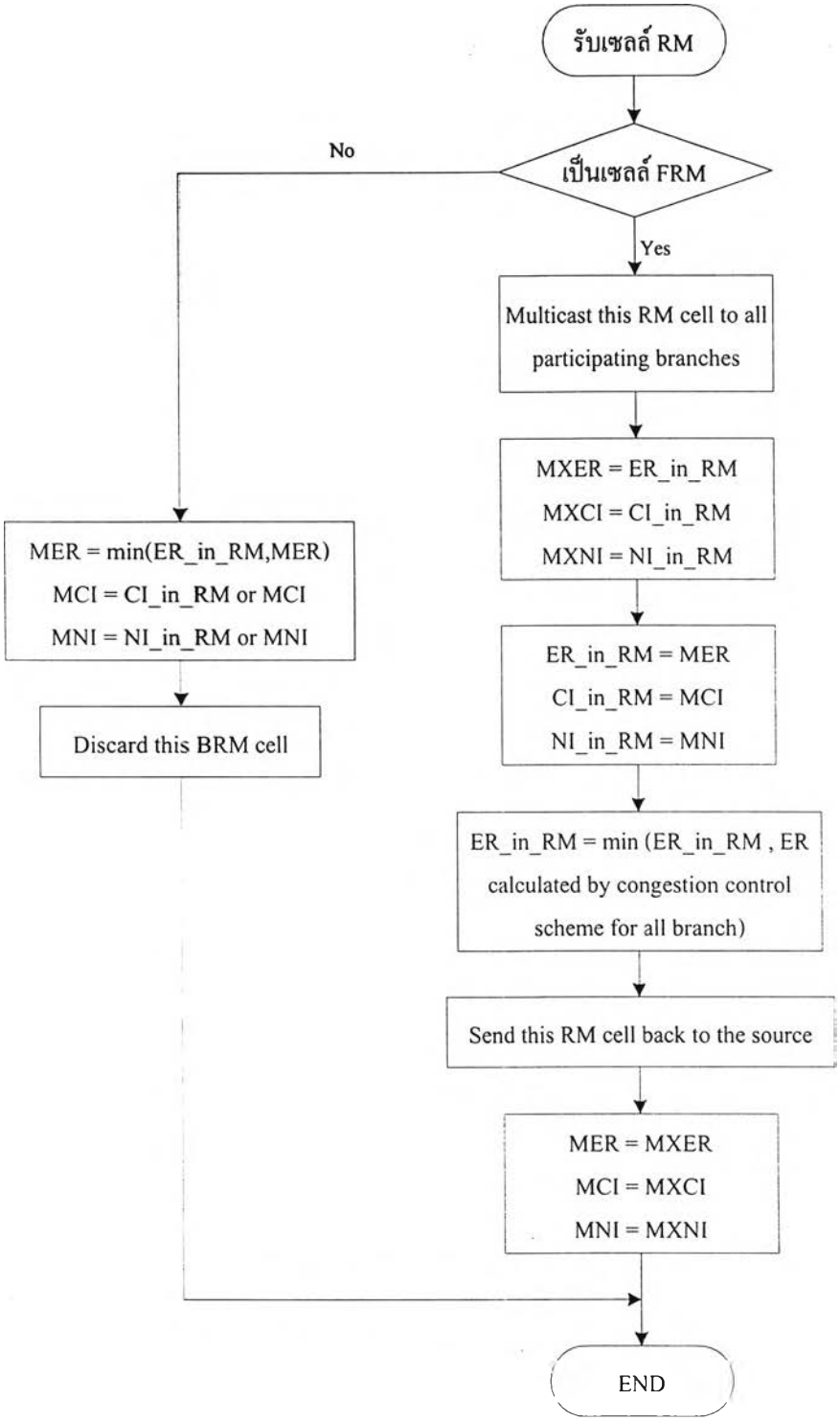
3.3.1 Robert Algorithm

Branch point จะรวบรวมข่าวสารการป้อนกลับซึ่งได้แก่ ค่า ER, CI และ NI จากเซลล์ BRM ทุกครั้งเมื่อได้รับเซลล์ BRM โดยค่าต่าง ๆ ข้างต้นจะทำการรวบรวมเก็บไว้ในตัวแปร MER, MCI และ MNI ตามลำดับ ตัวแปรทั้งสามนี้จะจัดการโดยเป็นตัวแปรประเภท per-VC นั่นคือ สวิตช์จะมีตัวแปรดังกล่าวข้างต้นนี้เป็นจำนวนเท่ากับปริมาณการเชื่อมต่อจากหนึ่งจุดไปสู่หลายจุดที่ผ่านสวิตช์นั้น และจะจัดการค่าต่าง ๆ แยกเป็นแต่ละการเชื่อมต่อ โดยค่า MER จะเป็นค่า ER น้อยสุดจากทุก branch สำหรับในการเชื่อมต่อนั้น ส่วน MCI และ MNI จะมีค่าเป็น 1 ก็ต่อเมื่อค่า CI และ NI จาก branch ใด branch หนึ่งมีค่าเป็น 1 ตามลำดับ

เมื่อสวิตช์ได้รับเซลล์ FRM จะสร้างเซลล์ BRM เพื่อป้อนกลับข่าวสารความคับคั่ง เซลล์ BRM ที่สร้างขึ้นมาใหม่นี้จะถูกบรรจุด้วยข่าวสารความคับคั่งที่รวบรวมไว้ก่อนแล้วนั่นคือ MER, MCI และ MNI นั่นเอง จากนั้นสวิตช์จะคำนวณค่า ER สำหรับการป้อนกลับจากทุก branch โดยใช้วิธีควบคุมความคับคั่งที่มีอยู่ ค่าต่าง ๆ ที่ได้จะนำมาเปรียบเทียบกับค่าที่อยู่ในเซลล์ BRM โดยค่าที่น้อยที่สุดจะถูกบรรจุอยู่ในเซลล์ BRM ก่อนการส่งเซลล์ BRM ตัวนี้ไปสู่แหล่งกำเนิด เมื่อสวิตช์ทำการส่งเซลล์ BRM ดังกล่าวไปสู่แหล่งกำเนิดแล้ว ตัวแปร MER จะถูกลบล้างค่าให้เป็นค่า ER ที่อยู่ในเซลล์ FRM ที่เพิ่งได้รับนั่นเอง ส่วน MCI และ MNI จะถูกล้างค่าให้เป็น 0 วิธีนี้นั้นมีความเร็วในการป้อนกลับสูงมาก แต่ไม่มีความถูกต้องในการป้อนกลับ โดยเมื่อ branch point ได้รับเซลล์ FRM แล้วตอบกลับข่าวสารไปแล้วนั้น หากยังไม่ได้รับเซลล์ BRM ก่อนที่ได้จะรับเซลล์ FRM ตัวถัดไปก็จะทำการป้อนกลับข่าวสารที่ผิดพลาดได้ ลำดับการของวิธีการรวบรวมข่าวสารความคับคั่งวิธีนี้จะเป็นดังรูปที่ 3.5 และแผนผังการทำงานจะอยู่ในรูปที่ 3.6

```
Upon the receipt of a forward RM(ER,CI,NI) cell :
    Multicast this RM cell to all participating branches;
    Let  MXER = ER , MXCI = CI , MXNI = NI;
    Let  ER = MER , CI = MCI , NI = MNI;
    Let  ER = min (ER , ER calculated by congestion control scheme for all branch)
    Send this RM cell back to the source;
    Let  MER = MXER , MCI = MXCI , MNI = MXNI;
Upon the receipt of a backward RM(ER,CI,NI) Cell
    Let  MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
    Discard the BRM Cell
```

รูปที่ 3.5 ลำดับการทำงานของ Robert algorithm



รูปที่ 3.6 แผนผังการทำงานของ Robert algorithm

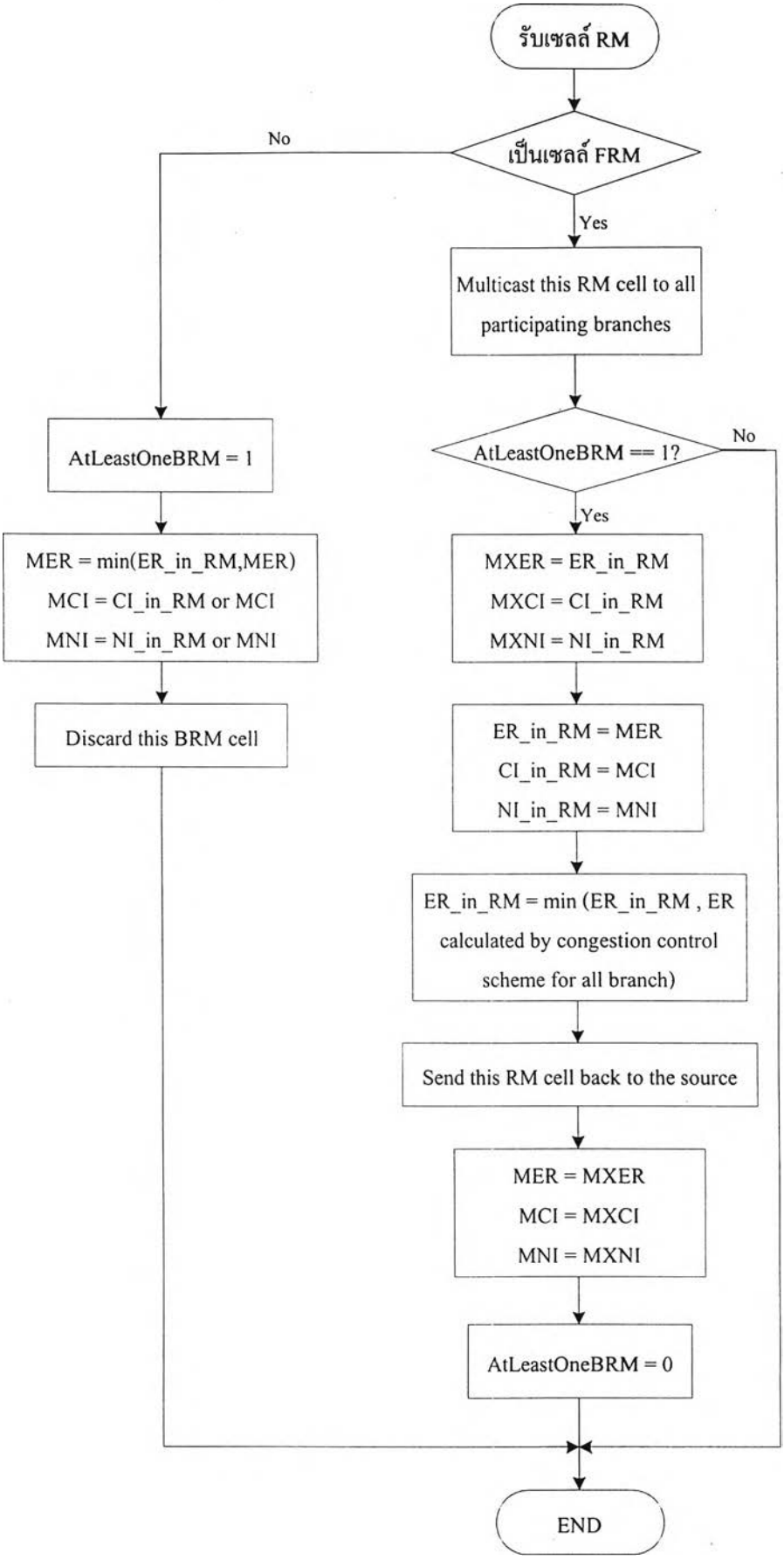
3.3.2 TS Algorithm

วิธีรวบรวมข่าวสารความคับคั่งวิธีนี้พัฒนามาจาก Robert Algorithm โดยวิธีการดังกล่าวนั้นเกิด Consolidation noise ขึ้น เนื่องจากความไม่ถูกต้องในการรวบรวมข่าวสารความคับคั่ง ซึ่งบางกรณีนั้นอาจมีการสร้างและส่งเซลล์ BRM ไปสู่แหล่งกำเนิด โดยใช้ค่า MER ที่ยังไม่ได้รับการปรับแต่งจากการรับเซลล์ BRM จึงมีโอกาสที่ค่า MER นั้นจะเป็นค่า ER ที่มาจาก FRM ครั้งก่อน ซึ่งส่วนใหญ่จะเป็น

ค่า PCR นั่นเอง ดังนั้นจึงมีการเสนอวิธีให้ สวิตช์นั้นจะไม่สามารถป้อนกลับข่าวสารความคับคั่งได้ หาก ยังไม่ได้รับเซลล์ FRM และ ยังไม่ได้รับข่าวสารจากเซลล์ BRM อย่างน้อยหนึ่งตัวจาก branch ใด branch หนึ่งก่อน ซึ่งจะทำให้ค่า MER ที่รวบรวมไว้ที่สวิตช์นั้นจะได้รับการปรับแต่งจากการรับเซลล์ BRM ก่อนอย่างน้อยหนึ่งครั้ง เทคนิคดังกล่าวสามารถเพิ่มความถูกต้องในการรวบรวมข่าวสารการป้อนกลับได้ เนื่องจากทำให้ข่าวสารที่รวบรวมไว้ใน MER มีความถูกต้องมากขึ้นก่อนการป้อนกลับข่าวสารนั้น อย่างไรก็ตาม หาก MER ยังไม่รับการปรับแต่งจากการรับเซลล์ BRM จาก branch ที่มีปัญหาความคับคั่งจริง ๆ ก็ทำให้ค่า MER ที่เกิดขึ้นนั้นยังคงมีความผิดพลาดอยู่ นอกจากนี้จากผลการศึกษาต่อมาพบว่าหากสวิตช์ต้องเป็นฝ่ายตอบกลับเซลล์ FRM ลำดับการของวิธีการรวบรวมข่าวสารความคับคั่งวิธีนี้จะเป็นดังรูปที่ 3.7 และแผนผังการทำงานจะอยู่ในรูปที่ 3.8

```
Upon the receipt of a forward RM(ER,CI,NI) cell :
    Multicast this RM cell to all participating branches;
    If AtLeastOneBRM then
        Let  MXER = ER , MXCI = CI , MXNI = NI;
        Let  ER = MER , CI = MCI , NI = MNI;
        Let  ER = min (ER , ER calculated by congestion control scheme for all branch)
        Send this RM cell back to the source;
        Let  MER = MXER , MCI = MXCI , MNI = MXNI;
        Let  AtLeastOneBRM = 0;
Upon the receipt of a BRM(ER,CI,NI) Cell
    Let  AtLeastOneBRM = 1;;
    Let  MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
    Discard the BRM Cell
```

รูปที่ 3.7 ลำดับการทำงานของ TS algorithm



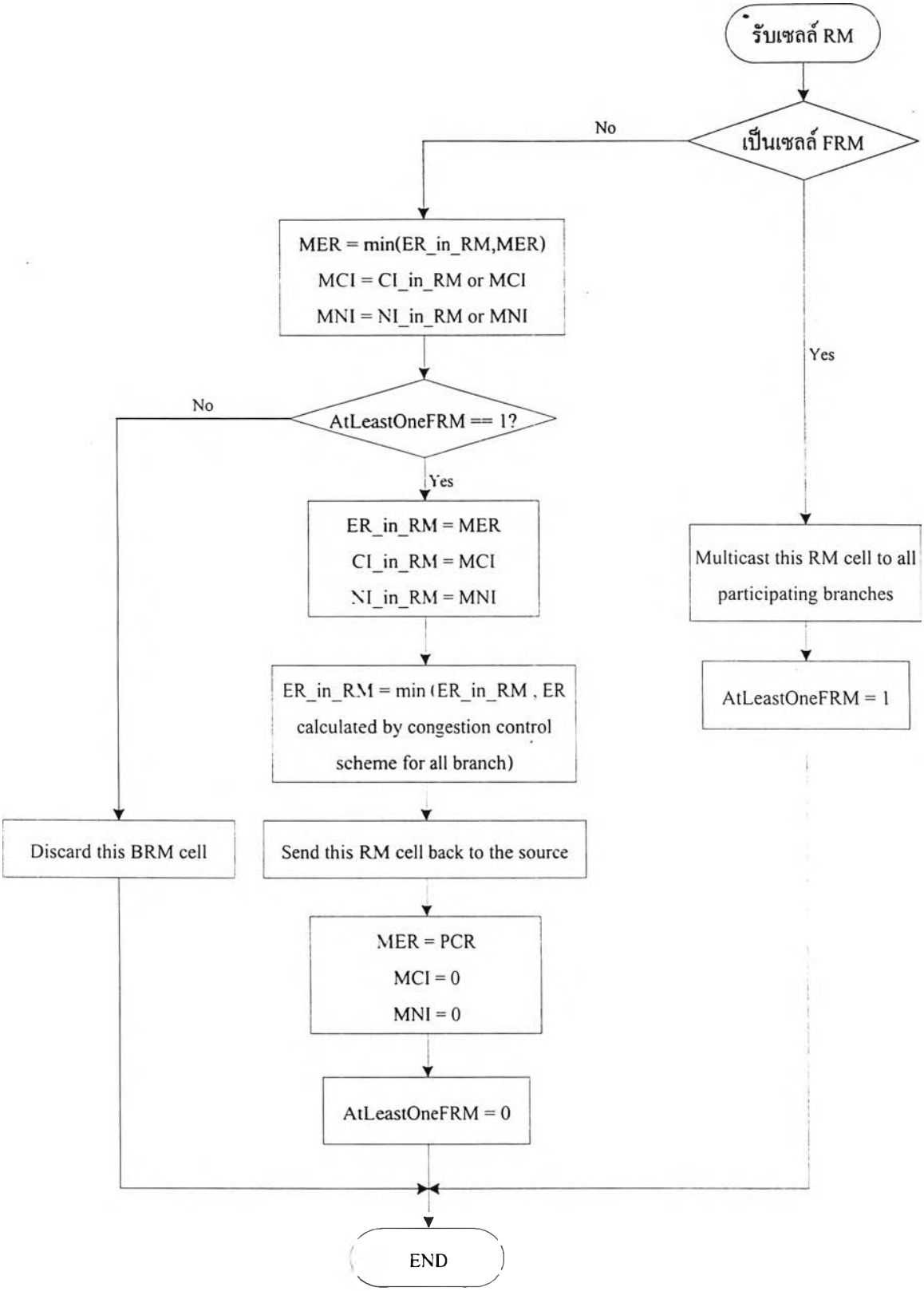
รูปที่ 3.8 แผนผังการทำงานของ Robert algorithm

3.3.3 RSS Algorithm

วิธีรวบรวมข่าวสารความคับคั่งวิธีนี้พัฒนามาจากวิธีการ Robert Algorithm และ TS Algorithm อีกทีหนึ่ง โดยเสนอว่าการสร้างเซลล์ BRM เพื่อตอบกลับหลังจากได้รับเซลล์ FRM ทุกครั้งนั้นเป็นการเพิ่มความซับซ้อนให้กับสวิตช์ขึ้นมาก โดยสวิตช์ควรทำหน้าที่สลับเส้นทางและปรับแต่งค่าในเซลล์เพียงเท่านั้น ดังนี้ RSS Algorithm จึงได้ถูกพัฒนาขึ้น โดยใช้วิธีการตรวจจับว่าหากมีเซลล์ FRM ผ่านสวิตช์นั้นแล้วจะรอให้มีเซลล์ BRM ตัวต่อไปเพื่อปรับแต่งค่า ER, CI และ NI ในเซลล์ BRM ดังกล่าว โดยพิจารณาจากค่า MER, MCI และ MNI ที่รวบรวมไว้ที่สวิตช์ และ ค่าที่คำนวณได้จากวิธีควบคุมความคับคั่งของสวิตช์ จากนั้นจะส่งเซลล์ BRM นั้นผ่านไปหาแหล่งกำเนิดต่อไป โดยลำดับการทำงานและแผนผังการทำงานเป็นดังรูปที่ 3.9 และ 3.10 ตามลำดับ

```
Upon the receipt of a forward RM(ER,CI,NI) cell :
    Multicast this RM cell to all participating branches;
    Let  AtLeastOneFRM = 1;
Upon the receipt of a backward RM(ER,CI,NI) Cell
    Let  MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
    If AtLeastOneFRM then
        Let  ER = MER , CI = MCI , NI = MNI;
        Let  ER = min (ER , ER calculated by congestion control scheme for all branch)
        Send this RM cell back to the source;
        Let  MER = PCR , MCI = 0 , MNI = 0;
        Let  AtLeastOneFRM = 0;
    else
        Discard the BRM Cell
```

รูปที่ 3.9 ลำดับการทำงานของ RSS algorithm



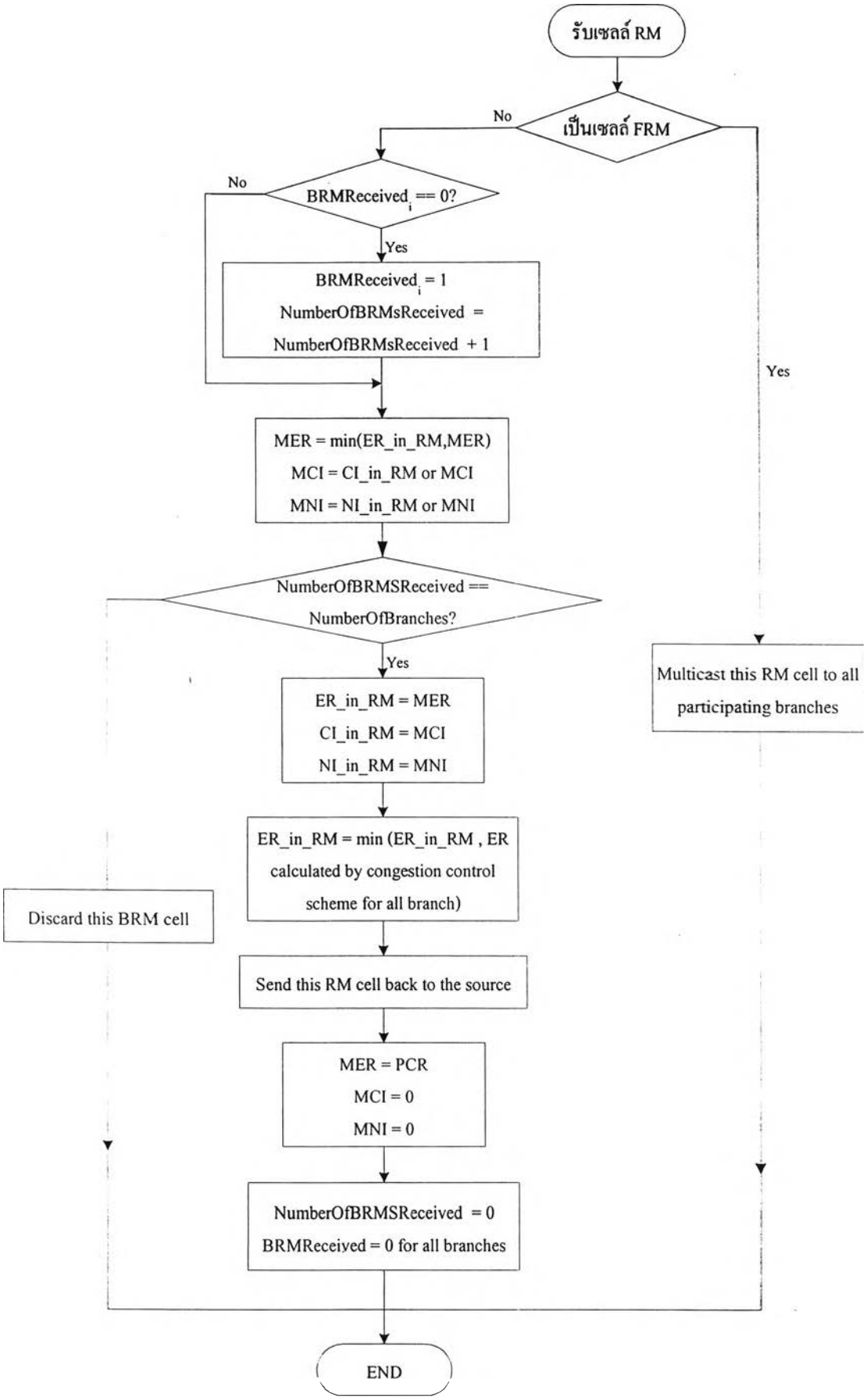
รูปที่ 3.10 แผนผังการทำงานของ RSS algorithm

3.3.4 Wait-for-all Algorithm

Branch point จะรวบรวมเซลล์ BRM ให้ครบตามจำนวนให้เท่ากับจำนวน branch ก่อนโดยการเก็บค่า ER ที่น้อยที่สุดจากทุก ๆ เซลล์ RM ที่เข้ามาไว้ในตัวแปร MER และเก็บค่า CI และ NI ที่แสดงถึงความคับคั่งไว้ จากนั้นจะส่งผ่านเซลล์ RM ตอบกลับไปหาแหล่งกำเนิด โดยค่า ER ที่ตอบกลับ คือค่าน้อยที่สุดระหว่างค่า MER ค่า ER ที่คำนวณได้จากวิธีการควบคุมความคับคั่ง การที่จะทำให้สวิตช์สามารถตรวจสอบปริมาณเซลล์ BRM ที่รับได้นั้น ต้องมีการเพิ่มตัวแปร NumberOfBRMsReceived เพื่อการตรวจนับเซลล์ BRM ที่ได้รับมาแล้ว โดยตัวแปรดังกล่าวมีการจัดการเป็น Per-VC ส่วนตัวแปร BRMReceived นั้นมีการจัดการเป็น Per-output นั่นคือมีการจัดการตัวแปรต่อหนึ่งเส้นทางขาออก โดยตัวแปร BRMReceived มีไว้เพื่อตรวจสอบการรับเซลล์ BRM สำหรับเส้นทางขาออกนั้น เมื่อจำนวนเซลล์ BRM ที่รับได้ที่สวิตช์มีค่าเท่ากับจำนวน branch ของการเชื่อมต่อนั้น สวิตช์จะทำการส่งเซลล์ BRM ที่เพิ่งได้รับนั้นไปสู่แหล่งกำเนิด โดยมีการปรับแต่งค่า ER ก่อนการป้อนกลับ จากนั้นตัวแปร MER จะถูกลบล้างเป็นค่า PCR ของการเชื่อมต่อนั้น ส่วน CI, NI และ NumberOfBRMsReceived จะได้รับการลบล้างให้เป็น 0 สำหรับทุก branch และ BRMReceived จะถูกลบล้างเป็น 0 สำหรับวิธีการนี้ทำมีความถูกต้องของการป้อนกลับ แต่มีการตอบกลับที่ช้ามาก เนื่องจากต้องรอเซลล์ RM ให้กลับมาครบทุก branch ก่อน ดังนั้นหากว่ามีปลายใดปลายหนึ่งที่มี propagation delay มากจะทำให้แหล่งกำเนิดต้องรอให้เซลล์ BRM นั้นเดินทางกลับมาจากปลายทางที่ไกลที่สุดก่อน อันจะเป็นการถ่วงให้เวลาในการป้อนกลับในช่วงแรกนั้นช้ามาก โดยเฉพาะในกรณีที่มีเซลล์ RM จำนวนน้อยอยู่ในโครงข่าย ยกตัวอย่างเช่นในช่วงเริ่มต้นการส่งข้อมูล ลำดับการทำงานและแผนผังการทำงานของวิธีการนี้จะเป็นดังรูปที่ 3.11 และ 3.12 ตามลำดับ

```
Upon the receipt of a forward RM(ER,CI,NI) cell :
    Multicast this RM cell to all participating branches;
Upon the receipt of a backward RM(ER,CI,NI) Cell from branch i :
    If not BRMReceived_i then
        Let BRMReceived_i = 1;
        Let NumberOfBRMsReceived = NumberOfBRMsReceived + 1;
    Let MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
    If (NumberOfBRMSReceived == NumberOfBranches ) then
        Let ER = MER , CI = MCI , NI = MNI;
        Let ER = min (ER , ER calculated by congestion control scheme for all branch)
        Send this RM cell back to the source;
        Let MER = PCR , MCI = 0 , MNI = 0;
        Let NumberOfBRMSReceived = 0;
        Let BRMReceived = 0 for all branch;
    else
        Discard the BRM Cell
```

รูปที่ 3.11 ลำดับการทำงานของ Wait-for-all algorithm



รูปที่ 3.12 แผนผังการทำงานของ Wait-for-all algorithm

3.3.5 Immediate Rate Calculation Algorithm

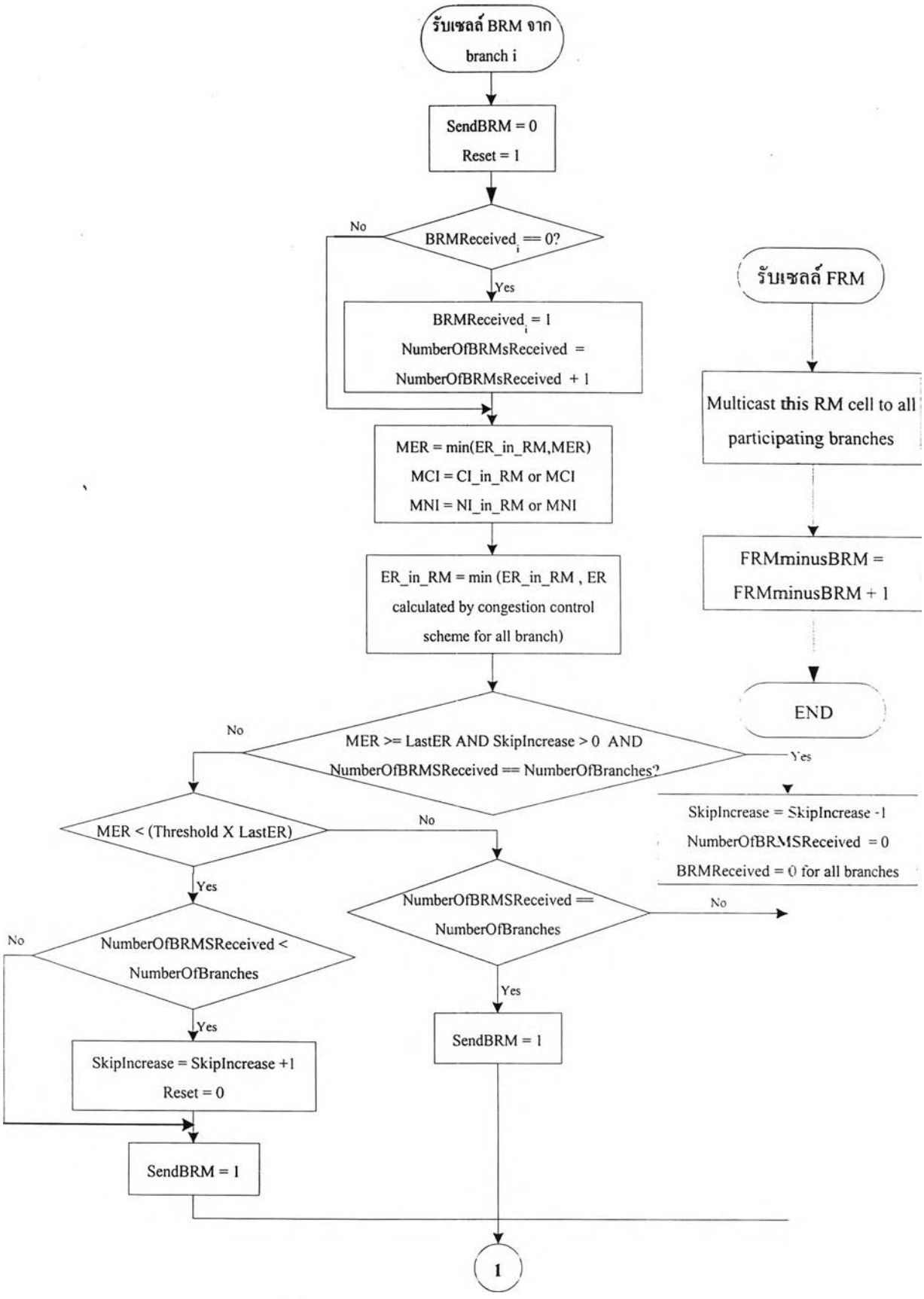
วิธีรวบรวมข่าวสารความคับคั่งนี้พัฒนามาจาก Wait-for-all Algorithm หากแต่เพิ่มความสามารถในการตรวจจับสภาวะโหลดเกินของโครงข่าย (Overload detected) โดยมีวิธีการตรวจสอบโดยให้สวิตช์เปรียบค่า ER ในเซลล์ BRM ที่เพิ่งได้รับกับค่า ER ที่ใช้ในการป้อนกลับครั้งที่แล้ว (Last_ER) ทุกครั้งที่ได้รับเซลล์ BRM และหากค่า ER ในเซลล์ BRM นั้นมีค่าน้อยกว่าค่า Last_ER คูณค่ากับ Threshold หนึ่ง ๆ แสดงว่าโครงข่ายเริ่มเกิดความคับคั่งขึ้น เนื่องจากค่าอัตราส่งที่สวิตช์แนะนำมีค่าน้อยลง ซึ่งสามารถบ่งชี้ถึงความคับคั่งของโครงข่ายได้ ดังนั้นสวิตช์จะคำนวณค่า ER ใหม่จากค่า MER และค่าที่ได้จากการใช้วิธีควบคุมความคับคั่ง เพื่อปรับแต่งค่า ER ในเซลล์ BRM ที่เพิ่งได้รับ และทำการส่งผ่านเซลล์ BRM ที่เพิ่งได้รับนั้นกลับไปสู่แหล่งกำเนิดทันที โดยไม่รอให้มีการรับเซลล์ BRM จากทุก branch ก่อน

อย่างไรก็ตาม การส่งเซลล์ BRM ที่เกิดจากการใช้ Overload detected จะทำให้ปริมาณเซลล์ BRM ที่ป้อนกลับมีจำนวนมากขึ้น และอัตราส่วนระหว่างจำนวนเซลล์ FRM ที่แหล่งกำเนิดส่งไป และจำนวนเซลล์ BRM ที่แหล่งกำเนิดได้รับ จะมีค่าเกิน 1 อันจะทำให้วิธีควบคุมข่าวสารป้อนกลับที่แหล่งกำเนิดอาจเกิดปัญหาได้ เนื่องจากแหล่งกำเนิดนั้นมีการนับปริมาณเซลล์ FRM ที่ส่งไปและยังไม่ได้รับการตอบกลับไว้ ดังนั้นหากสวิตช์มีการส่งเซลล์ BRM กลับสู่แหล่งกำเนิดมากกว่าจำนวนเซลล์ FRM ที่ได้รับอาจทำให้เกิดปัญหาในการควบคุมความคับคั่งได้ ด้วยเหตุนี้ วิธีการนี้จึงแก้ปัญหาด้วยการนับจำนวนการส่งเซลล์ BRM จากการใช้ Overload detected ไว้ และในรอบการป้อนกลับถัดไปเมื่อค่า ER ที่เก็บรวบรวมไว้ได้จากทุก Branch ไม่แสดงความคับคั่ง (ไม่เกิด Overload Indication) ก็จะงดการส่งเซลล์ BRM นั้น เพื่อชดเชยการส่งเซลล์ BRM จากการใช้ Overload Indication ไป

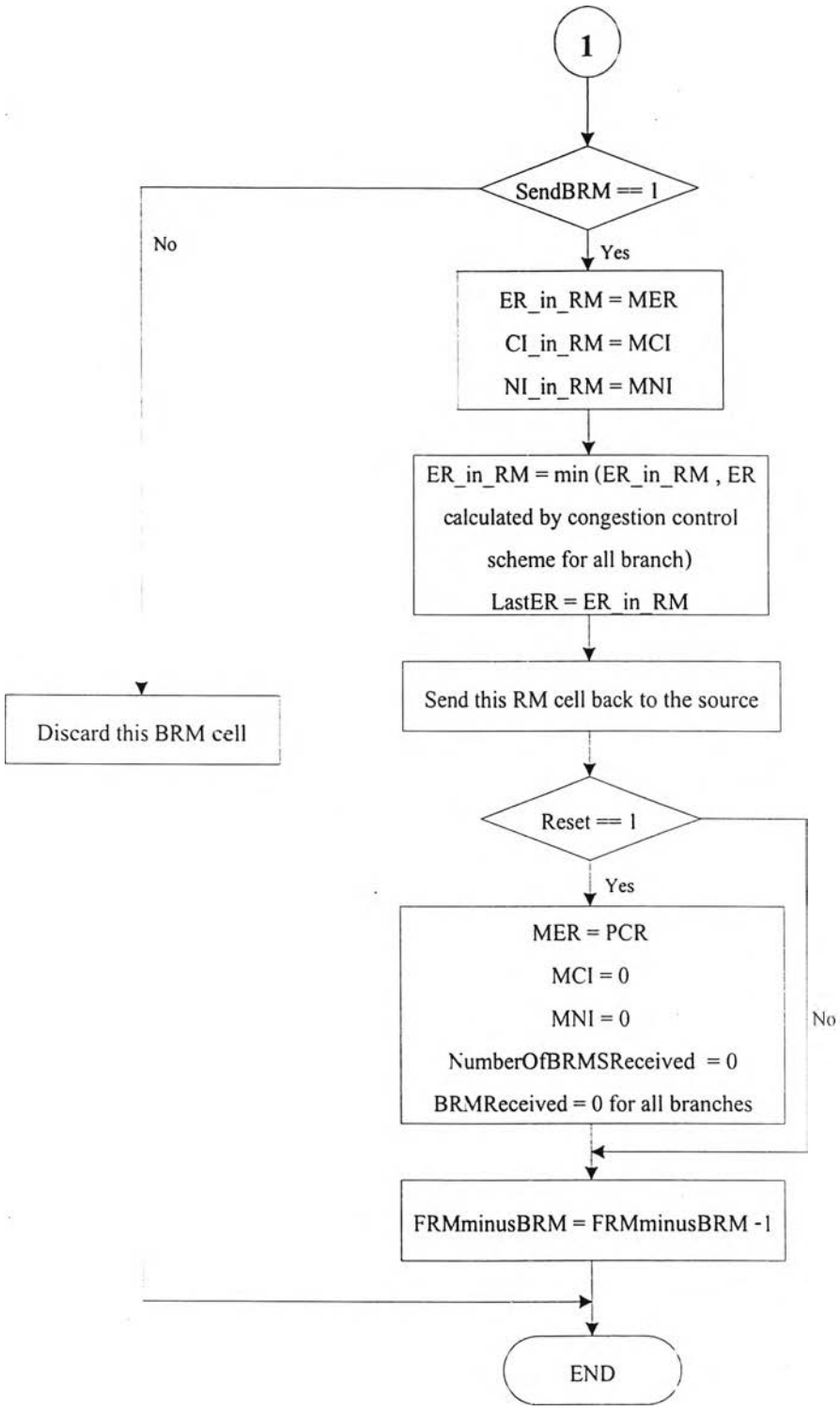
การใช้เทคนิค Overload detected นั้นสามารถช่วยเพิ่มความเร็วในการป้อนกลับในสภาวะโหลดเกินแก่วิธี Wait-for-all algorithmได้เป็นอย่างดี อย่างไรก็ตามเทคนิคดังกล่าวยังไม่สามารถเพิ่มความเร็วในการป้อนกลับได้มากนัก เนื่องจากการตรวจสอบสภาวะ Overload detected นั้น มีการตรวจสอบจากค่า ER ที่ได้จาก branch โดยไม่ตรวจสอบสภาวะของสวิตช์ตัวเองก่อนการตัดสินใจ ดังนั้นเพื่อเพิ่มความเร็วในการป้อนกลับขึ้นมา วิธี Immediate rate calculation จึงเพิ่มให้มีการคำนวณสภาวะความคับคั่งของสวิตช์ตัวเองก่อน โดยการให้มีการคำนวณค่า ER โดยใช้วิธีควบคุมความคับคั่งที่มี ก่อนการตรวจสอบ Overload detected ซึ่งวิธีนี้ทำให้ตรวจสอบความคับคั่งได้อย่างรวดเร็วขึ้น โดยสามารถตรวจสอบสถานะความคับคั่งของสวิตช์ตนเอง และ สถานะความคับคั่งจาก branch ต่าง ๆ ว่าเกิดสภาวะโหลดเกินขึ้นในโครงข่ายหรือไม่ และดำเนินการส่งเซลล์ BRM พิเศษเพื่อป้อนกลับความคับคั่งสู่แหล่งกำเนิด ลำดับการทำงานและแผนผังการทำงานของวิธีการนี้จะเป็นดังรูปที่ 3.13 และ 3.14 ตามลำดับ

```
Upon the receipt of a forward RM(ER,CI,NI) cell :
    Multicast this RM cell to all participating branches;
    Let   FRMminusBRM = FRMminusBRM +1;
Upon the receipt of a backward RM(ER,CI,NI) Cell from branch  i  :
    Let   SendBRM = 0;
    Let   Reset = 1;
    If not BRMReceived_i then
        Let BRMReceived_i = 1;
        Let NumberOfBRMsReceived = NumberOfBRMsReceived + 1;
    Let   MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI;
    Let   MER = min (MER , minimum ER calculated by congestion control scheme for all branch)
                                                        (*** for faster overload detect ***)
    If (MER ≥ LastER) and (SkipIncrease > 0) and (NumberOfBRMSReceived == NumberOfBranches ) then
    {
        Let   SkipIncrease = SkipIncrease –1 ;
        Let   NumberOfBRMSReceived = 0;
        Let BRMReceived = 0 for all branch;
    }
    else if (MER < (Threshold x  LastER)) then        (***** overload is detected *****)
    {
        if (NumberOfBRMsReceived < NumberOfBranches) then
            Let   SkipIncrease = SkipIncrease + 1;
            Let   Reset = 0;
        Let SendBRM = 1;
    }
    else if (NumberOfBRMsReceived == NumberOfBranches) then
        Let SendBRM = 1;
    If (SendBRM) then
    {
        Let   ER = MER , CI = MCI , NI = MNI;
        Let   ER = min (ER , ER calculated by congestion control scheme for all branch)
        Send this RM cell back to the source;
        Let LastER = ER;
        If (Reset) then
            Let   MER = PCR , MCI = 0 , MNI = 0;
            Let NumberOfBRMsReceived = 0;
            Let   BRMReceived = 0 for all branchs
        Let   FRMminumBRM = FRMminumBRM –1;
    }
    else
        Discard the BRM Cell
```

รูปที่ 3.13 ลำดับการทำงานของ Immediate Rate Calculation algorithm



รูปที่ 3.14 แผนผังการทำงานของ Immediate rate calculation algorithm



รูปที่ 3.14 แผนผังการทำงานของ Immediate rate calculation algorithm (ต่อ)

3.4 ปัญหาการทำงานของวิธีควบคุมความคับคั่งร่วมกับวิธีรวบรวมข่าวสารความคับคั่ง

การศึกษาที่ผ่านมาทางด้านการให้บริการประเภท ABR แบบหนึ่งจุดไปสู่หลายจุดที่ผ่านมานั้น เน้นการศึกษาสำหรับใช้งานกับวิธี ERICA ซึ่งยังไม่มีการศึกษาให้นำมาใช้กับวิธีการควบคุมความคับคั่งอื่น ๆ วิทยานิพนธ์ฉบับนี้จึงเสนอการนำวิธีรวบรวมข่าวสารความคับคั่งแบบต่าง ๆ มาใช้กับวิธี

I18952768

FMMRA เพื่อศึกษาปัญหา Consolidation noise และความเร็วในการป้อนกลับ ซึ่งการเกิด consolidation noise ในบางสภาวะนั้นอาจมีสาเหตุมากจากการใช้งานร่วมกันระหว่าง ERICA และ วิธีรวบรวมข่าวสารความคับคั่งที่ไม่ดี

สำหรับวิธีการรวบรวมข่าวสารความคับคั่งใหม่ ๆ ที่มีอยู่ในปัจจุบันนี้เน้นที่รักษาความถูกต้องของการรวบรวมข่าวสารความคับคั่งจากเซลล์ BRM เท่านั้น ทำให้มีผลตอบสนองชั่วครู่ และ consolidation delay ที่ช้ามาก วิธี Immediate rate calculation algorithm นั้นพัฒนาเทคนิคในการตรวจจับสภาวะโหลดเกินเพื่อเพิ่มความเร็วในการป้อนกลับในสภาวะโหลดเกิน อย่างไรก็ตามเทคนิคดังกล่าวเป็นการแก้ไขปัญหา consolidation delay ที่ปลายเหตุ ซึ่งสังเกตได้ว่า หากโครงข่ายอยู่ในสภาวะโหลดต่ำ วิธีการดังกล่าวยังคงมีผลตอบสนองที่ช้า ซึ่งทำให้เกิดปัญหาการใช้งานสายสัญญาณที่ไม่คุ้มค่าขึ้น วิทยานิพนธ์ฉบับนี้เสนอสาเหตุของการเกิด consolidation noise และพยายามหาวิธีการแก้ไขปัญหาดังกล่าว โดยแนวทางการแก้ไขปัญหานั้นจะเน้นที่พัฒนาให้มีผลตอบสนองชั่วครู่ที่รวดเร็ว เพื่อขจัด consolidation delay เช่นเดียวกับในวิธีการแรก ๆ หากแต่พัฒนาวิธีการใหม่ ๆ เพื่อเพิ่มความถูกต้องให้กับข่าวสารที่เก็บรวบรวมไว้ที่ branch point นั่นเอง

3.5 วิธีรวบรวมความคับคั่งที่เสนอแบบที่ 1

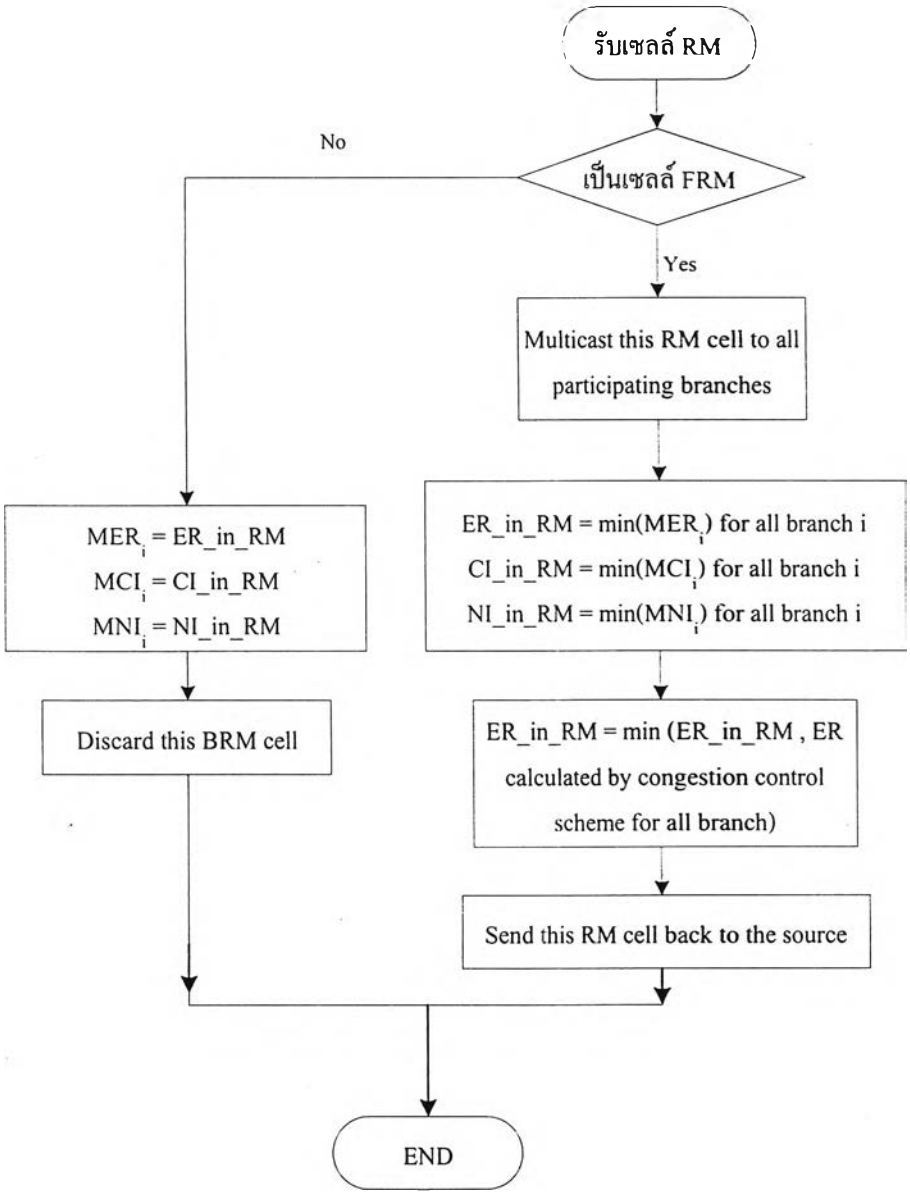
วิธีการรวบรวมความคับคั่งแบบต่าง ๆ ที่เสนอมานั้น โดยส่วนใหญ่แล้วจะพยายามพัฒนาเฉพาะทางด้านความเร็ว หรือ ความถูกต้องของข่าวสารความคับคั่ง อย่างใดอย่างหนึ่งเท่านั้น วิธีการแรก ๆ นั้นเน้นความเร็วในการป้อนกลับในขณะที่ไม่สนใจความถูกต้องในการป้อนกลับ ส่วนวิธีใหม่ ๆ ที่พัฒนาขึ้นมานั้นเน้นให้มีความถูกต้องของข้อมูลที่ใช้ในการป้อนกลับ แต่มีผลตอบสนองที่ช้า จึงได้มีการพัฒนาเพื่อเพิ่มความเร็วในการป้อนกลับสำหรับวิธีนี้ขึ้นโดยใช้เทคนิค overload detected ที่สามารถตรวจสอบสภาวะโหลดเกิน และทำการป้อนกลับทันที ดังที่ได้เสนอในวิธี Immediate rate calculation อย่างไรก็ตาม วิธีดังกล่าวยังมีข้อเสียในด้านผลตอบสนองชั่วครู่ในสภาวะโหลดต่ำ (Underload condition) ทำให้โครงข่ายนั้นมีการใช้งานสายสัญญาณที่ไม่คุ้มค่า

วิธีการที่เสนอจึงเน้นที่ความเร็วในการป้อนกลับ ในขณะที่ปรับปรุงความถูกต้องของข่าวสารความคับคั่งที่เก็บรวบรวมไว้ที่ branch point เนื่องจากผลตอบสนองชั่วครู่ของ Robert algorithm นั้นให้ผลตอบที่เร็วที่สุด วิธีการใหม่จึงนำหลักการของ Robert algorithm มา โดยนำเงื่อนไขในการป้อนกลับเซลล์ BRM มาใช้ branch point จะทำการส่งเซลล์ BRM ก็ต่อเมื่อได้รับ FRM แล้วเท่านั้น ส่วนความถูกต้องของข่าวสารที่เก็บไว้นั้น ได้พัฒนาขึ้นมาโดยการเพิ่มจำนวนตัวแปรของข่าวสารที่รวบรวมไว้ให้มีลักษณะการเก็บ โดยเก็บแยกแต่ละ VC และ เก็บค่าจากทุก ๆ branch (per-VC and per-branch basis) ทำให้ข่าวสารที่เก็บรวมรวมไว้มีความถูกต้องและมีความข้อมูลที่ใหม่อยู่เสมอ โดยการตัดสินใจในการป้อนกลับค่า ER นั้นจะใช้ค่า ER ที่น้อยที่สุดระหว่างค่า ER ที่เก็บรวบรวมไว้จากทุก branch นั่นเอง ซึ่ง

หลักการนี้ถูกนำมาใช้สำหรับค่า CI และ NI เช่นกัน ลำดับขั้นตอนการทำงานและแผนผังการทำงานของวิธีการนี้จะเป็นดังรูปที่ 3.15 และ 3.16 ตามลำดับ

Upon the receipt of a forward RM(ER,CI,NI) cell :
Multicast this RM cell to all participating branches;
Let ER = min(MER_i) , CI = or(MCI_i) and NI = or(MNI_i) for all participating branches;
Let ER = min (ER , ER calculated by congestion control scheme for all branches)
Send this RM cell back to the source;
Upon the receipt of a backward RM(ER,CI,NI) Cell from branch i:
Let MER_i =ER , MCI_i = CI , MNI_i = NI;
Discard the BRM Cell

รูปที่ 3.15 ลำดับการทำงานของ วิธีการที่เสนอแบบที่ 1



รูปที่ 3.16 แผนผังการทำงานของ วิธีการที่เสนอแบบที่ 1

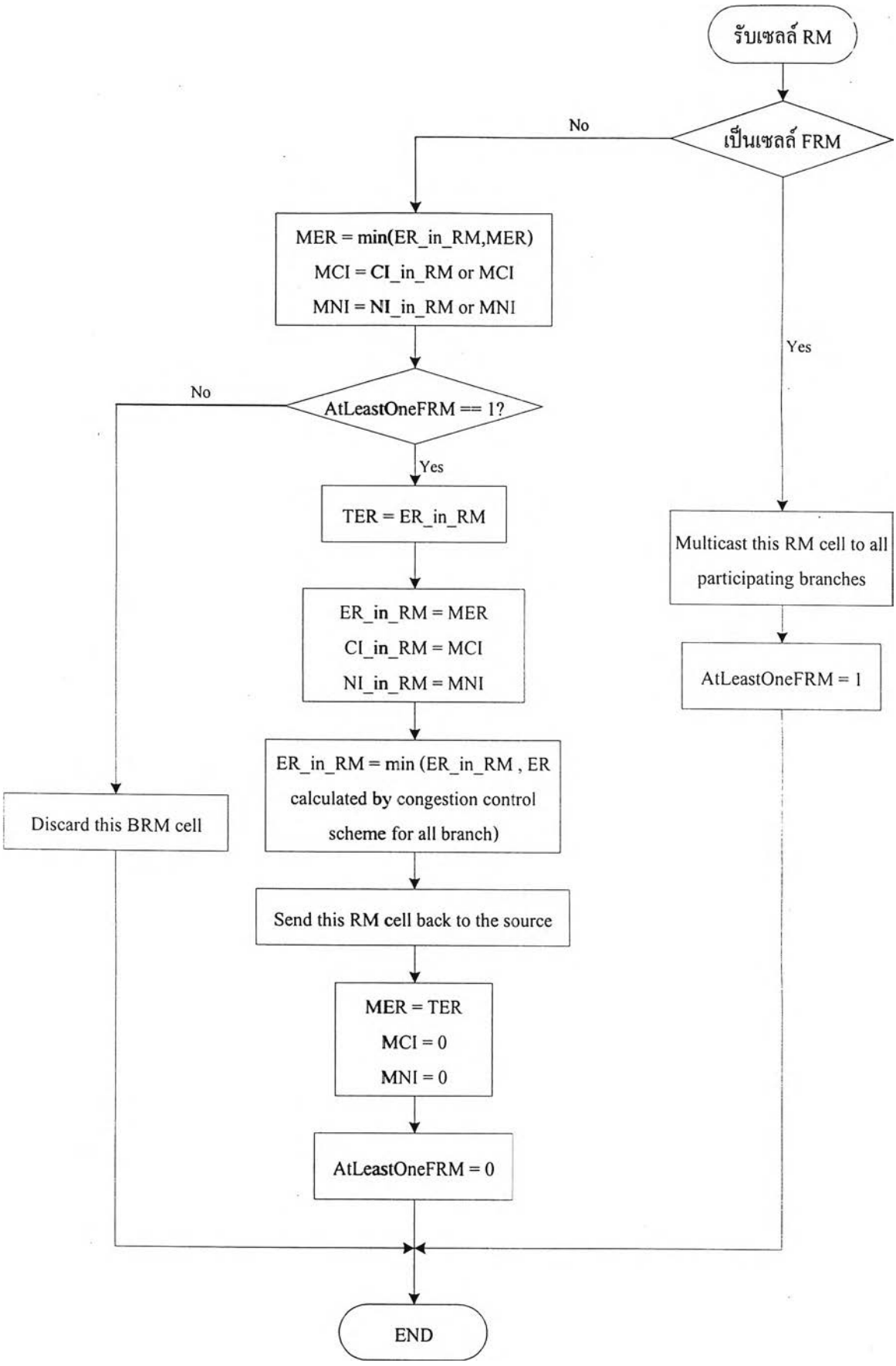
3.6 วิธีรวบรวมความคับคั่งที่เสนอแบบที่ 2

ในวิธีรวบรวมความคับคั่งของ Robert TS และ RSS นั้นให้ผลตอบชั่วครู่ที่รวดเร็วมาก แต่ยังคงมี Consolidation noise อยู่ สำหรับวิธีรวบรวมความคับคั่งที่ตอบกลับเซลล์ BRM จากการตรวจสอบเซลล์ FRM นั้น หากจำนวนเซลล์ FRM ที่ branch point ได้รับมากกว่าเซลล์ BRM ที่ได้รับจาก branch หนึ่ง ๆ จะทำให้เกิดการตอบกลับเซลล์ BRM มากขึ้น สาเหตุหนึ่งของการเกิด Consolidation noise นั้นเนื่องจากมีการลบล้างค่า MER ที่รวบรวมไว้จากทุก Branch หลังจากส่งเซลล์ BRM ไปแล้ว เมื่อมีการตอบกลับเซลล์ BRM มากขึ้นจะทำให้ข่าวสารที่รวบรวมไว้เกิดการผิดพลาดขึ้นเนื่องจากการลบล้างค่า MER นั่นเอง

วิธีการใหม่ที่เสนออีกหนึ่งวิธีนี้จึงพัฒนามาจาก RSS algorithm ที่เน้นให้มีความเร็วในการป้อนกลับ โดยที่มีหลักการทำงานพื้นฐานเหมือนกับ RSS algorithm หากแต่พัฒนาไม่ให้มีการลบล้างค่า MER แต่นำค่า ER ในเซลล์ BRM ที่ได้รับก่อนจะตอบกลับเซลล์ RM มาเป็นค่า MER แทน ซึ่งถือเป็นการป้อนกันการเกิด consolidation noise อย่างหนึ่ง ตัวแปร TER นั้นถูกสร้างขึ้นมาเพื่อบันทึกค่า ER ที่ได้รับจากเซลล์ BRM ตัวล่าสุด และนำค่าดังกล่าวไปเป็นค่า MER ทันทีหลังจากป้อนกลับเซลล์ BRM กลับสู่แหล่งกำเนิดแล้ว ลำดับขั้นตอนการทำงานของวิธีการนี้จะเป็นดังนี้

```
Upon the receipt of a forward RM(ER,CI,NI) cell :
    Multicast this RM cell to all participating branches;
    Let  AtLeastOneFRM = 1;
Upon the receipt of a backward RM(ER,CI,NI) Cell
    Let  MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
    If AtLeastOneFRM then
        Let  TER = ER;
        Let  ER = MER , CI = MCI , NI = MNI;
        Let  ER = min (ER , ER calculated by congestion control scheme for all branch)
        Send this RM cell back to the source;
        Let  MER = TER , MCI = 0 , MNI = 0;
        Let  AtLeastOneFRM = 0;
    else
        Discard the BRM Cell
```

รูปที่ 3.17 ลำดับการทำงานของ วิธีการที่เสนอแบบที่ 2



รูปที่ 3.18 แผนผังการทำงานของ วิธีการที่เสนอแบบที่ 2