บทที่ 4

การออกแบบและการสร้างแบบจำลอง

จากเนื้อหาในบทที่ 2 และบทที่ 3 ที่กล่าวถึงโครงสร้าง สถาปัตยกรรม องค์ประกอบและการทำงานภายในโปรโตคอลชั้นที่ 2 ของระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 และหลักการส่งข้อมูลวีดีทัศน์บนเครือข่าย UMTS ในบทนี้จะกล่าวถึงการออกแบบและสร้างแบบจำลองเพื่อใช้ในการทดสอบประเมินสมรรถนะการทำงานของโปรโตคอล ซึ่งเป็นหนึ่งในวัตถุประสงค์ของงานวิจัยนี้ที่ต้องการให้ได้มาซึ่งแบบจำลองต้นแบบที่สามารถประเมินสมรรถนะการทำงานของโปรโตคอลชั้นที่ 2 ของระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 ได้ เนื้อหาในบทนี้ประกอบด้วย หัวข้อ 4.1 อธิบายการออกแบบสร้างระบบทดสอบ เหตุผลในการเลือกโปรโตคอล ประเภทของข้อมูลที่ส่ง กลไกการทำงน และพารามิเตอร์สำหรับทดสอบผลกระทบ หัวข้อ4.2 กล่าวถึงเครื่องมือที่ใช้สร้างแบบจำลองตามความต้องการ วิธีการเพิ่มเติมโมดูลการทำงานของระบบ UMTS ลงในระบบเดิมที่เครื่องมือสนับสนุนการทำงานอยู่ก่อนแล้ว และบทสรุปอยู่ในหัวข้อ 4.3

4.1 การออกแบบสร้างระบบทดสอบ

จากวัตถุประสงค์ของงานวิจัยที่ต้องการมีแบบจำลองต้นแบบที่สามารถประเมินสมรรถนะการทำงานของโปรโตคอลได้ ดังนั้นเนื้อหาในหัวข้อนี้จะกล่าวถึง การออกแบบระบบที่จะใช้ในการทดสอบสมรรถนะการทำงานของโปรโตคอลชั้นที่ 2 ซึ่งประกอบด้วยหลายส่วนการทำงาน การออกแบบเป็นไปดังต่อไปนี้

4.1.1 การเลือกกลไกและพารามิเตอร์ของโปรโตคอลชั้นที่ 2

4.1.1.1 การเลือกโปรโตคอล RLC

โปรโตคอลชั้นที่ 2 ของระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 สามารถแบ่งออกเป็นโปรโตคอลชั้นย่อย ๆ ได้แก่ PDCP, BMC, RLC และ MAC แต่ละชั้นย่อยจะมีหน้าที่แตกต่างกันไปดังต่อไปนี้

- ชั้น PDCP มีหน้าที่ในการบีบอัด header ของข้อมูลแพ็กเกตที่มาจากชั้นบน เป็นการช่วยลดประมาณของข้อมูลที่จะทำการส่งออกไปในช่องสัญญาณไร้สายที่มีทรัพยากรอยู่อย่างจำกัด

- ชั้น BMC ทำหน้าที่รับส่งข้อมูลจุดเชื่อมต่อทางวิทยุ (radio interface messages) ที่มาจากศูนย์การกระจายเซล (Cell Broadcast Center) แล้วทำการส่งต่อไปยังชั้น RLC เพื่อทำงานต่อไป
- ชั้น MAC จะให้บริการในการเชื่อมช่องสัญญาณไปยังชั้น RLC ผ่านทางช่องสัญญาณลอจิก (logical channel) ซึ่งลักษณะช่องสัญญาณลอจิก นี้จะกำหนดโดยชนิดของข้อมูลที่จะทำการส่ง นอกจากนี้ยังทำหน้าที่นำเอาแพ็กเกตจากชั้น RLC มาจัดเรียงใส่ใน transport block เพื่อทำการส่งต่อไปให้ชั้น physical ผ่านทางช่องสัญญาณขนส่ง (transport channel)
- ชั้น RLC ทำหน้าที่หลักในการแบ่งและต่อข้อมูลให้มีขนาดเล็กลง จัดส่งข้อมูล และควบคุมความผิดพลาดในการส่งข้อมูลของโปรโตคอลชั้นบนโดยการตรวจสอบข้อผิดพลาดของข้อมูลและการส่งใหม่เพื่อให้ข้อมูลมีความถูกต้องสร้างความน่าเชื่อถือของการรับส่ง

จากโปรโตคอลทั้งหมดที่อยู่ภายในชั้นที่ 2 นั้น โปรโตคอล RLC จะเป็นโปรโตคอลหลักที่ทำหน้าที่ในการจัดการควบคุมความผิดพลาดในการส่งข้อมูลของโปรโตคอลชั้นบน ในขณะที่โปรโตคอลตัวอื่นภายในชั้นก็จะทำงานไปตามหน้าที่ ไม่ได้มีการปรับตัวใด ๆ เพื่อให้การทำงานหรือการส่งข้อมูลเป็นไปได้ดีขึ้น เช่น PDCP ทำหน้าที่ในการบีบอัด header ของแพ็กเก็ตที่มาจากชั้นบน ให้มีขนาดเล็กลงก่อนทำการส่งต่อให้ชั้น RLC ซึ่งการบีบอัด header ก็ช่วยในเรื่องปริมาณข้อมูลที่ต้องส่งไปในช่องสัญญาณเท่านั้น ไม่ได้มีผลต่อความถูกต้องหรือน่าเชื่อถือของข้อมูลที่ส่งนัก สำหรับโปรโตคอล MAC ก็ทำหน้าที่เชื่อมต่อช่องสัญญาณ นำเอาแพ็กเกตจากชั้น RLC มาจัดเรียงใส่ใน transport block แล้วส่งต่อไปให้ชั้น physical การตัดสินใจทำงานใด ๆ ของชั้น MAC นี้ ก็ไม่สามารถตัดสินใจได้ภายในชั้นตนเอง แต่เป็นการส่งผลการทำงานไปยังชั้น RRC ซึ่งเป็นชั้นควบคุม แล้วให้ RRC ส่งคำสั่งในการตัดสินที่จะเปลี่ยนแปลงการทำงานหรือไม่กลับมายังชั้น MAC จากนั้นชั้น MAC ก็จะปรับตัวเพื่อให้ทำงานได้ตรงกับคำสั่งที่ RRC ส่งมาให้ เป็นต้น ดังนั้น จึงเกิดสมมติฐานว่า โปรโตคอล RLC นี้น่าจะเป็นโปรโตคอลหลักภายในชั้นที่ 2 ที่เป็นปัจจัยสามารถทำให้สมรรถนะการทำงานของระบบโดยรวมเปลี่ยนแปลงไป อันเนื่องมาจากกลไกการทำงานและการกำหนดค่าพารามิเตอร์ของโปรโตคอล ดังนั้น การออกแบบแบบจำลองและการทดลองจึงอ้างอิงจากโปรโตคอล RLC เป็นหลัก

4.1.1.2 การเลือกโหมดการทำงานของโปรโตคอล RLC

โปรโตคอล RLC สามารถทำงานได้ 3 โหมดด้วยกัน คือ Transparent Mode (TM) Unacknowledged Mode (UM) และ Acknowledged Mode (AM) ทั้ง 3 โหมด RLC ประกอบด้วย 2 ส่วนคือ RLC ฝั่งส่ง และ RLC ฝั่งรับ ซึ่งฝั่งส่งจะทำหน้าที่ส่ง RLC Protocol Data

Units (PDUs) และฝั่งรับก็จะทำหน้าที่รับ RLC PDU โดยโหมดของบริการที่จะเลือกใช้ดูจากชนิดของข้อมูลที่จะทำการส่ง ลักษณะการทำงานของแต่ละโหมด คือ

Transparent Mode หรือ โหมด TM เป็นโหมดในการส่งข้อมูลที่ไม่มีการใส่ header เพิ่มเข้าไป หากข้อมูลที่ส่งเกิดข้อผิดพลาดก็จะถูกตัดทิ้งไป การส่งข้อมูลจะใช้สำหรับการส่งสตรีมจากชั้นบนที่ไม่มีการแบ่งข้อมูลมา เช่น ข้อมูลเสียง เป็นต้น ในการส่งข้อมูลในโหมด TM นั้นจะขึ้นกับขนาดของการส่งข้อมูล โดยที่หากมีขนาดใหญ่มากจะต้องทำการแบ่งข้อมูลในฝั่งส่งและต่อข้อมูลในฝั่งรับ

Unacknowledged Mode หรือโหมด UM ใช้ในการส่งข้อมูลที่ไม่ต้องการรับประกันการส่งถึงไปยังฝั่งรับ หากข้อมูลที่ส่งเกิดข้อผิดพลาดก็จะถูกตัดทิ้งไปเช่นกัน โครงสร้างของ PDU หรือแพ็กเกตที่ส่งจะมีการเพิ่ม header ที่ระบุ Sequence Number (SN) ของแพ็กเกตไว้ตรวจสอบข้อมูลที่หายไปได้ แต่ก็ไม่สามารถกู้คืนข้อมูลที่หายนั้น การบริการที่เหมาะสำหรับโหมดนี้ได้แก่ Cell Broadcast Service และ Voice over IP (VoIP) เป็นต้น

Acknowledged Mode หรือโหมด AM ใช้ในการจัดส่งข้อมูลที่ต้องการรับประกันการส่งข้อมูลว่าถึงผู้รับแน่นอน โดยจะมีกระบวนการส่งซ้ำของข้อมูลเมื่อเกิดข้อผิดพลาดหรือสูญหาย ซึ่งใช้กระบวนการ Automatic Repeat reQuest (ARQ) เพื่อให้ข้อมูลเกิดความถูกต้อง ในการให้บริการของโหมดนี้จะเป็นการส่งข้อมูลแพ็กเกตที่ต้องการความแน่นอน เช่น อินเตอร์เน็ต การส่งจดหมายอิเล็กทรอนิกส์ การดาวโหลดข้อมูล เป็นต้น

การเปรียบเทียบบริการของแต่ละโหมดสามารถแสดงได้ดังตารางที่ 4.1 หากยกตัวอย่างโปรแกรมประยุกต์ที่มีใช้ในปัจจุบันสำหรับการสื่อสารทางโทรศัพท์เคลื่อนที่ ได้แก่ การสื่อสารด้วยเสียง การสื่อสารด้วยวีดิทัศน์ การรับส่งจดหมายอิเล็กทรอนิกส์ การเปิดเว็บไซต์ หรือการเรียกดูไฟล์ตัวอย่างภาพยนต์ เป็นต้น โปรแกรมประยุกต์แต่ละประเภทก็จำเป็นต้องใช้บริการต่าง ๆ ของโหมดการทำงาน RLC ที่ต่างกันออกไปขึ้นกับเป้าหมายของโปรแกรมประยุกต์นั้น ๆ ตัวอย่างเช่น การส่งข้อมูลเสียงที่ต้องการการส่งที่รวดเร็วมากที่สุด โดยไม่สนใจความถูกต้องของการส่งข้อมูลมากนัก ดังนั้นจึงไม่จำเป็นต้องเรียกใช้บริการในการตรวจสอบและแก้ไขข้อผิดพลาดการกู้คืนของข้อมูล ในขณะที่โปรแกรมประยุกต์สำหรับการรับส่งจดหมายอิเล็กทรอนิกส์ หรือการเปิดเว็บไซต์ที่ต้องมีการโต้ตอบระหว่างโปรแกรมและผู้ใช้ มีความจำเป็นอย่างมากที่ต้องมีความถูกต้องของข้อมูลที่รับส่ง ซึ่งมีบริการด้านการควบคุมความผิดพลาดของการส่งข้อมูลอยู่ในโหมด AM จากตารางเปรียบเทียบบริการจะเห็นว่าโหมด AM สามารถทำงานได้ครอบคลุมการทำงานของทุกโหมด และสามารถรองรับการทำงานของโปรแกรมประยุกต์ได้เกือบทั้งหมด จึงเหมาะสำหรับการเลือกมาเป็นระบบทดสอบ แล้วประยุกต์ผลการทดสอบเพื่อใช้วิเคราะห์กับโหมดอื่น ๆ ได้ แต่หากเลือกโหมด TM หรือ UM มาทำการทดสอบแล้ว จะไม่สามารถประยุกต์ผลใช้งานกับโหมด AM ได้ เนื่องจากการทำงานของ 2 โหมดนี้ไม่ครอบคลุมสำหรับโหมด AM

ตารางที่ 4.1 บริการที่มีในแต่ละโหมดการทำงานของชั้น RLC

บริการ	โหมดการทำงาน		
	TM	UM	AM
การส่งข้อมูล	✓	✓	✓
การแบ่งและรวมกลับของข้อมูล	✓	✓	✓
การเติมเต็มข้อมูล		✓	✓
การเข้ารหัส		✓	✓
การตัดทิ้งของ SDU	✓	✓	✓
การตรวจสอบหมายเลขลำดับ		✓	
การต่อข้อมูล		✓	✓
การตรวจสอบและแก้ไขข้อผิดพลาด			✓
การส่งเป็นลำดับของ PDU ของชั้นบน			✓
การตรวจสอบการซ้ำกันของข้อมูล			✓
การควบคุมการไหลของข้อมูล			✓
การตรวจสอบและการกู้คืนกลับมาของข้อผิดพลาดของโปรโตคอล			✓

นอกจากนี้ใน 2 โหมดแรก เป็นการทำงานที่ไม่การันตีความถูกต้องของการรับ-ส่งข้อมูล ไม่มีการส่งข้อมูลซ้ำหากเกิดความผิดพลาด ทำหน้าที่เพียงแค่ส่งผ่านข้อมูลเท่านั้น ดังนั้นจึงไม่เหมาะที่จะเลือกมาทดสอบสมรรถนะ เพราะไม่สามารถปรับแต่งรูปแบบการทำงานใด ๆได้ โหมดที่เลือกนำมาใช้ในการทดลองจึงเป็น RLC ที่ทำงานในโหมด AM ซึ่งในโหมดการทำงานนี้เอง ก็ประกอบด้วย กลไกและพารามิเตอร์ที่ซับซ้อนเป็นจำนวนมาก จึงเป็นการยากที่จะทำการทดสอบผลกระทบต่อสมรรถนะการทำงานของกลไกและพารามิเตอร์ทั้งหมดได้ และจากการศึกษาหาข้อมูลก่อนการออกแบบ พบว่า กลไกและพารามิเตอร์บางส่วนของโปรโตคอล RLC ได้มีผู้วิจัยรายอื่นทำการทดสอบวัดผลกระทบที่มีต่อสมรรถนะการทำงานของระบบไปบ้างแล้ว ดังนั้น ผู้วิจัยจึงเลือกออกแบบการทดลองเพื่อวัดผลกระทบของกลไกและพารามิเตอร์บางตัวจากกลุ่มที่เหลือ โดยในหัวข้อถัดไปจะกล่าวถึงเหตุผลในการเลือกกลไกและพารามิเตอร์ในการทดสอบ

4.1.1.3 การเลือกกลไกและพารามิเตอร์ของโปรโตคอล RLC และข้อมูลทดสอบ

จากเนื้อหาในบทที่ 3 จะเห็นว่า RLC ประกอบไปด้วยกลไกและพารามิเตอร์จำนวนมาก บางกลไกหรือพารามิเตอร์ก็มีการทำงานปฏิสัมพันธ์กันระหว่าง RLC ฝั่งรับ และ

RLC ฝั่งส่ง หากมองในภาพรวมทั้งหมดคงยากที่จะตัดสินใจว่าจะเลือกตัวใดมาทำการทดสอบวัดผลกระทบที่มีต่อสมรรถนะการทำงานของระบบ เพราะทุกตัวสามารถทำให้การทำงานของระบบมีการเปลี่ยนแปลงได้ทั้งสิ้น แต่หากแยกวิเคราะห์เฉพาะการทำงานที่มีผลต่อประสิทธิภาพของระบบในส่วนที่ทำให้ความเร็วในการรับส่ง และความน่าเชื่อถือถูกต้องของข้อมูลมีเพิ่มมากขึ้น สามารถพิจารณาได้ดังนี้

เนื่องจากประเด็นหลักที่สนในคือ ผลของตัวแปรใน RLC ต่อสมรรถนะการรับส่งข้อมูลในชั้น RLC ดังนั้น ประการแรกที่ต้องสนใจคือ ลักษณะข้อมูลที่จะเลือกนำมาทดสอบการส่ง จากที่ได้กล่าวไปแล้วในบทที่ 2 เรื่องของการแบ่งประเภทของข้อมูลที่ส่ง ข้อมูลกลุ่มที่ต้องการความถูกต้องของข้อมูลเป็นสำคัญแต่ไม่สนใจเวลาหน่วงมากนัก เช่น เว็บไซต์ และกลุ่มเบื้องหลัง เช่น การดาวน์โหลดไฟล์ การส่งอีเมล์ เป็นต้น สำหรับประเภทข้อมูลกลุ่มที่เน้นความเร็วในการรับส่งเป็นสำคัญ ได้แก่ กลุ่มเชิงสนทนา การสื่อสารด้วยเสียง กลุ่มสตรีมมิ่ง และกลุ่มที่มีการส่งข้อมูลอย่างต่อเนื่อง เช่น การส่งข้อมูลวีดิทัศน์แบบเวลาจริง เป็นต้น มีหลายงานวิจัยที่นำพารามิเตอร์ของ RLC มาทดสอบวัดประสิทธิภาพกับกลุ่มข้อมูล 2 ประเภทแรก[3][4][5][35] เพราะเป้าหมายที่งานวิจัยเหล่านั้นต้องการ ก็เพื่อศึกษาว่ามีพารามิเตอร์ใดบ้างที่ทำให้ข้อมูลที่รับส่งมีความถูกต้อง แม่นยำ แต่ไม่ได้สนใจเรื่องเวลาในการส่งมากเท่าใดนัก ด้วยเหตุที่จุดสนใจของงานวิจัยนี้ต้องการเรื่องของความเร็วที่ใช้ส่งด้วย จึงเลือกข้อมูลส่งกลุ่มสตรีมมิ่งแต่ไม่ใช่แบบเวลาจริง กล่าวคือ เป็นการร้องขอเปิดดูไฟล์วีดิทัศน์มาเป็นข้อมูลทดสอบ เนื่องจากข้อมูลประเภทนี้ นอกจากต้องการความเร็วในการส่งแล้ว ยังต้องการความถูกต้องด้วย เพราะแม้ว่าจะส่งได้เร็ว แต่วีดิทัศน์ที่ได้รับไม่สมบูรณ์ ก็อาจทำให้ผู้ใช้งานเกิดความไม่พอใจได้เช่นกัน ดังนั้นจึงเป็นที่น่าสนใจที่จะเลือกข้อมูลกลุ่มนี้มาเป็นข้อมูลทดสอบระบบ

ประเด็นถัดมาในเรื่องของการวิเคราะห์เลือกกลไก RLC มาทำการทดสอบ หากพิจารณาถึงกลไกหลักของ RLC ฝั่งส่งที่มีไว้เพื่อจัดการควบคุมการส่งข้อมูล (flow control) ของโปรโตคอลชั้นบน นั้นก็คือ กลไกการ Poll และกลไกการละทิ้ง SDU สำหรับกลไกการละทิ้ง SDU นั้นจะยอมให้ฝั่งส่งละทิ้ง AMD PDU ที่เกี่ยวข้องกับ SDU จากบัฟเฟอร์ส่งและบัฟเฟอร์ส่งซ้ำ กลไกนี้เกิดขึ้นเมื่อการส่ง AMD PDU ไม่สำเร็จตามช่วงเวลาหรือจำนวนครั้งของการส่งซ้ำที่อนุญาตให้มีได้ จะเลือกใช้การทำงานวิธีใดขึ้นอยู่กับการกำหนดใช้งานตามความต้องการของคุณภาพการบริการ ซึ่งหากต้องการให้เกิดการละทิ้งตามช่วงเวลาก็ต้องกำหนดค่าแก่พารามิเตอร์ *Timer based discard* วิธีนี้จะช่วยควบคุมระยะเวลาหน่วงสูงสุดของแพ็กเกตให้ไม่เกินค่าเวลาที่กำหนด เพราะหากแพ็กเกตเหล่านั้นคงอยู่ในบัฟเฟอร์ส่งหรือส่งซ้ำเกินระยะเวลาที่กำหนด แพ็กเกตเหล่านั้นก็จะถูกละทิ้งไปทันที และอีกวิธีหนึ่งคือ ควบคุมการอยู่ในบัฟเฟอร์ส่งด้วยการกำหนดจำนวนครั้งที่มากที่สุดที่แพ็กเกตนั้น ๆ สามารถถูกทำการส่งซ้ำได้ ไว้ในพารามิเตอร์ *MAXDAT* หากแพ็กเกตใดมีจำนวนครั้งการส่งซ้ำเท่ากับค่าพารามิเตอร์นี้ แพ็กเกตเหล่านั้นก็จะถูกทำการละทิ้งไปเช่นกัน ซึ่งการละทิ้งจากทั้ง 2 วิธี แพ็กเกตที่ถูกละทิ้งจะถือเป็นแพ็กเกตที่สูญหาย ไม่

สามารถทำการกู้คืนได้อีกต่อไป หากเป็นข้อมูลประเภทวีดิทัศน์ ผู้รับชมก็จะไม่สามารถเห็นเฟรมภาพในส่วนที่ถูกละทิ้งไปนั้นเอง ประโยชน์ของกลไกนี้คือ สามารถหลีกเลี่ยงการล้นของบัฟเฟอร์ในชั้น RLC และลดเวลาหน่วงสูงสุดของการส่ง แต่เนื่องจากว่า ประเด็นที่งานวิจัยต้องการ มีเรื่องของความถูกต้องของข้อมูลที่ส่งด้วย หากให้กลไกการละทิ้งนี้ทำงาน อาจเป็นผลให้ข้อมูลที่ส่งบางส่วนสูญหายไปอันเนื่องจากถูกละทิ้งออกจากบัฟเฟอร์ ดังนั้น ในการออกแบบการทดลองจึงตัดการทำงานของกลไกนี้ไป แต่ก็มีงานวิจัยที่ทำการทดสอบผลกระทบของกลไกการละทิ้ง SDU ต่อประสิทธิภาพการส่งข้อมูลเช่นเดียว [3] ซึ่งให้ผลโดยสรุปว่า การกำหนดค่า *Timer based discard* หรือ *MAXDAT* ค่าต่ำจะช่วยลดค่าเวลาหน่วงในการส่งข้อมูล แต่ก็จะนำมาซึ่งการสูญหายของแพ็กเกตที่เพิ่มมากขึ้นเช่นกัน สามารถดูข้อมูลอ้างอิงได้ในบทที่ 1 หัวข้อการตรวจเอกสาร

สำหรับกลไกการ Poll ก็จะมีการทำงานคู่กับกลไกการส่ง STATUS ณ RLC ฝั่งรับ ซึ่งทำหน้าที่ในการร้องขอสถานะการรับข้อมูล และส่งสถานะการรับข้อมูล ตามลำดับ การส่งซ้ำของฝ่ายส่งจะเกิดขึ้นโดยขึ้นอยู่กับการรับ STATUS report ที่ส่งมาจากฝ่ายรับ ทั้งนี้ RLC เป็นโปรโตคอลที่ไม่อนุญาตให้มีการส่งซ้ำนอกจากจะได้รับการตอบกลับสถานะว่าเกิดความผิดพลาดการส่ง STATUS report สามารถทำได้ทั้งจากฝ่ายส่งและฝ่ายรับ โดย RLC ฝั่งส่งสามารถที่จะร้องขอสถานะการรับข้อมูลจากฝั่งรับเอง หรือฝั่งรับอาจส่งสถานะการรับข้อมูลไปให้ฝั่งส่งเองโดยไม่มีการร้องขอก็ได้แต่สำหรับกลไกการ poll ฝ่ายส่งจะส่งบิต poll ไปร้องขอ และฝ่ายรับจะตอบกลับโดยส่ง STATUS report กลับมายังฝั่งส่ง การได้รับ STATUS report จะทำให้เกิดการส่งซ้ำของแพ็กเกตที่ผิดพลาด หากฝั่งส่งสามารถรับรู้สถานะการรับข้อมูลของฝั่งรับได้เร็ว การกู้คืนข้อมูลที่ส่งไม่สำเร็จและต้องการการส่งซ้ำจะเกิดได้เร็วขึ้นเช่นกัน จะเห็นว่ากลไกการ poll มีส่วนอยู่ในกระบวนการกู้คืนของข้อมูลของโปรโตคอล และความถี่ในการ poll ก็น่าจะมีผลต่อความเร็วในการกู้คืนข้อมูล ดังนั้น การออกแบบแบบจำลองจึงเลือกสนใจผลกระทบของกลไกนี้ต่อสมรรถนะโดยรวมของระบบ และเพื่อทดสอบความถี่ในการ poll ว่ามีผลกระทบต่อสมรรถนะในการส่งข้อมูลอย่างไร เมื่อข้อมูลทดสอบเป็นข้อมูลกลุ่มสตรีมมิ่งแต่ไม่ใช่แบบเวลาจริง และอีกเหตุผลหนึ่งที่เลือกสนใจกลไกการ poll ก็เนื่องจากว่า หลังจากได้ข้อสรุปว่ากลไกและพารามิเตอร์ใดมีผลต่อสมรรถนะการทำงานของระบบ ต้องการที่จะออกแบบวิธีการในการปรับปรุงประสิทธิภาพในการส่งข้อมูลให้ดีขึ้น ซึ่งการส่งข้อมูลหรือกู้ข้อมูลเกิดขึ้นที่ฝั่งส่ง และกลไกการ poll ก็ทำงานที่ฝั่งส่ง ดังนั้นการมุ่งจุดสนใจหลักไปที่ฝั่งส่งจึงเป็นประเด็นหลักในการทดสอบ

ทั้งนี้ภายในกลไกการ poll เองนั้นก็ประกอบด้วย trigger ทั้งหมด 8 ตัวด้วยกันที่ใช้เพื่อกำหนดค่าบิต poll ใน header ของ PDU ที่จะส่งออกไป แต่ไม่จำเป็นว่า trigger ทุกตัวจะต้องมีการใช้งานพร้อมกัน trigger ทั้ง 8 ตัว ได้แก่

- ***Poll Timer*** หลังจากที่ trigger ตัวนี้หมดอายุ บิต poll ใน header ของ AMD PDU ตัวที่กำลังจะถูกส่งจะถูกกำหนดค่าเป็น 1 เพื่อทำการร้องขอสถานะการรับ PDU ของฝั่งรับ โดย trigger ตัวจับเวลานี้จะหยุดการจับเวลาหลังจาก

ได้รับ status PDUs ตอบกลับจากฝั่งรับ แต่ถ้าตัวจับเวลาหมดอายุก่อนที่จะได้รับ status PDUs ใด ๆจากฝั่งรับ ฝั่งส่งก็จะทำการกำหนดบิต poll ใน AMD PDU แล้วเริ่มการจับเวลาของ trigger ตัวนี้อีกครั้ง

- ***Poll Prohibit Timer*** เริ่มต้นเมื่อบิต poll ที่อยู่ใน AMD PDU ถูกส่งไปยังชั้นล่างของฝั่งส่ง จากเวลาที่มีการ poll อันเนื่องมาจาก trigger ใด ๆ เกิดขึ้นจนกระทั้งถึงก่อนเวลาที่ซึ่ง Poll Prohibit Timer หมดอายุ จะไม่สามารถเกิดการ poll ใด ๆ ส่งออกไปได้ หากเกิดการ trig ของการ poll เกิดขึ้นในช่วงที่กำลังถูกห้าม การส่ง poll ครั้งใหม่จะถูกหน่วงไว้จนกระทั่งตัวจับเวลานี้หมดอายุ ถ้ามีมากกว่า 1 poll ที่ถูก trig ในขณะที่การ poll ถูกห้าม เมื่อ Poll Prohibit Timer หมดอายุ จะมีเพียงการ poll ตัวเดียวเท่านั้นที่จะถูกส่งได้
- ***Poll Periodic Timer*** ตัวจับเวลานี้จะเริ่มต้นนับเมื่อมีการสร้างเอ็นทิที้ RLC และเมื่อตัวจับเวลาหมดอายุจะมีการตรวจสอบจำนวน PDUs ที่อยู่ในบัฟเฟอร์ส่งซ้ำถ้ามีมากเท่ากับจำนวน threshold ที่กำหนดก็จะเกิดการ poll ขึ้น แต่หากมีจำนวนไม่ถึงก็จะไม่เกิดการ poll จากนั้นตัวจับเวลานี้ก็จะกลับไปที่ค่าเริ่มต้นใหม่เพื่อทำการจับเวลาอีกครั้ง
- ***Last PDU in buffer*** เป็น transmission-buffer-based trigger ทุกครั้งที่ AMD PDU ตัวสุดท้ายที่อยู่ในบัฟเฟอร์ส่ง ที่จะส่งครั้งแรกกำลังจะถูกส่งออกไป บิต poll ใน header ของ PDU นั้นจะถูกกำหนดค่าเป็น 1
- ***Last PDU in retransmission buffer*** เป็น retransmission-buffer-based trigger โดยบิต poll จะถูกกำหนดค่าเป็น 1 ถ้า AMD PDU ที่ถูกส่งซ้ำเป็น AMD PDU ตัวสุดท้ายที่เก็บในบัฟเฟอร์ส่งซ้ำ
- ***Poll_PDU_PDU*** จะมีการ poll ทุก ๆการส่ง PDU ได้ตามจำนวนที่กำหนด เช่น กำหนดให้มีการ poll เกิดขึ้นทุก ๆ 200 PDUs เป็นต้น
- ***Poll_SDU_SDU*** จะมีการ poll ทุก ๆการส่ง SDU ได้ตามจำนวนที่กำหนด เช่น กำหนดให้มีการ poll เกิดขึ้นทุก ๆ 100 SDUs เป็นต้น
- ***Window based polling*** เป็น transmission-window-based trigger จะเกิดการ poll เมื่อมีจำนวน PDUs อยู่ใน transmission window มากกว่าเปอร์เซ็นของขนาด window ที่กำหนด

ซึ่ง trigger ทั้ง 8 ตัว เป็นตัวกำหนดความถี่ในการ poll ทั้งสิ้น แต่มีเงื่อนไขและวัตถุประสงค์ในการเกิดความถี่แตกต่างกัน สามารถแบ่งออกเป็น 3 กลุ่มตามเงื่อนไข คือ

1) กลุ่มที่ใช้ตัวจับเวลาเป็นตัวกำหนดความถี่ในการ poll ได้แก่ poll timer, poll periodic timer และ poll prohibit timer
2) กลุ่มที่สนใจจำนวน PDU ในบัฟเฟอร์ ได้แก่ Last PDU in buffer และ

Last PDU in retransmission buffer และ

3) กลุ่มที่ใช้จำนวนการส่งออกของแพ็กเกต(PDU/SDU) เป็นตัวกำหนดความถี่ ได้แก่ Poll_PDU_PDU, Poll_SDU_SDU และ Window based polling

โดยปกติเมื่อเกิดการ poll จะทำให้ฝั่งรับตอบสนองโดยการส่ง status report กลับไปยังฝั่งส่ง เพื่อฝั่งส่งจะได้ทราบว่าควรทำการนำแพ็กเกตใดบ้างออกจากบัฟเฟอร์อันเนื่องมาจากการส่งสำเร็จ หรือควรทำการส่งซ้ำแพ็กเกตใดกรณีที่สถานะการตอบกลับแจ้งว่าส่งแพ็กเกตนั้นไม่สำเร็จ ดังนั้น หากฝั่งส่งสามารถที่จะรับรู้ได้เร็ว การส่งซ้ำของแพ็กเกตที่ผิดพลาดก็จะเกิดขึ้นได้เร็ว แต่ถ้าฝั่งส่งรับรู้ช้า การส่งซ้ำของแพ็กเกตที่ผิดพลาดก็จะเกิดขึ้นช้าได้ด้วยเช่นกัน และการที่ฝั่งส่งจะสามารถรับรู้สถานะการรับแพ็กเกตได้เร็วมากน้อยเพียงใดนั้น ก็ขึ้นอยู่กับความถี่ในการร้องขอ status report ซึ่งสามารถกำหนดได้ด้วยค่าของ trigger ที่มีอยู่ในกระบวนการ poll นั้นเอง

เนื่องจากประเด็นที่งานวิจัยสนใจก็คือ ความถี่ในการได้รับ status report มีผลอย่างไรกับสมรรถนะการรับส่งข้อมูลของระบบ และพิจารณาผลของกระบวนการที่ดำเนินการต่อหลังจากได้รับสถานะแล้วกับแพ็กเกตที่ผิดพลาด สูญหาย หรือรับสำเร็จ ดังนั้น การเลือกใช้ trigger เพื่อใช้ทดสอบปรับเพิ่มหรือลดประมาณความถี่ในการ poll เพื่อให้ได้มาซึ่งปริมาณ status report และการดำเนินการต่อนั้นต้องพิจารณาจากวัตถุประสงค์ของ trigger แต่ละตัวให้เหมาะกับประเด็นที่งานวิจัยสนใจ และจากข้อมูลที่ส่งเป็นข้อมูลสตรีมมิ่ง ที่ซึ่งสนใจเวลาในการส่งเป็นสำคัญ ดังนั้นเพื่อให้สอดคล้องกับกลุ่มข้อมูลที่ส่ง จึงเลือก trigger ในกลุ่มที่ใช้ตัวจับเวลาเป็นตัวกำหนดความถี่การ poll ที่สามารถกำหนดใช้เชิงของเวลาปรับให้เกิดการร้องขอ status report มากน้อยตามความเร็วที่ต้องการ ในขณะที่ trigger กลุ่มอื่นจะไม่ได้ใช้เวลาเป็นตัวกำหนดความถี่ หากพิจารณา trigger ภายในกลุ่มที่ใช้ตัวจับเวลาเป็นตัวกำหนดความถี่ ที่ประกอบด้วย trigger 3 ตัว ได้แก่ Poll Timer, Poll Prohibit Timer และ Poll Periodic Timer สามารถแยกการวิเคราะห์การเลือกใช้ได้เป็น

- สำหรับ poll prohibit timer ใช้เพื่อจัดการกับปัญหาการส่ง poll ที่มากเกินไป การ poll ที่ถี่ไปจะทำให้ฝั่งรับ ส่งสถานะหรือ status report กลับมามากเกินความจำเป็น การป้องกันการส่งสถานะที่มากเกินไปสามารถทำได้โดยการกำหนด poll prohibit timer ที่ฝั่งส่ง หรือกำหนด status prohibit timer ที่กลไกการส่ง STATUS ที่ฝั่งรับ แต่เนื่องจากฝั่งส่งไม่สามารถรับรู้สถานะการรับข้อมูลของฝั่งรับ การปรับค่าป้องกันนะฝั่งส่งจึงไม่ได้ประโยชน์เท่าที่ควร ดังนั้นจึงกำหนดใช้ status prohibit timer ที่ฝั่งรับแทน
- การ poll โดยใช้ Poll Periodic Timer เนื่องจาก Poll Periodic Timer สนใจว่า มีปริมาณแพ็กเกตที่ค้างอยู่เพื่อรอการส่งซ้ำมากน้อยเพียงใด ถ้ามากถึงปริมาณที่

กำหนดจึงจะทำให้เกิดการ poll ที่ซึ่งการ poll เพื่อร้องขอสถานะการรับข้อมูลในลักษณะนี้ ไม่ได้เป็นไปตามวัตถุประสงค์ของงานวิจัย ที่ต้องการให้การส่งสถานะก็เพื่อเป็นการปรับปรุงข้อมูลสถานะการทำงานระหว่างฝั่งรับและฝั่งส่ง เพื่อให้ฝั่งส่งดำเนินการต่อกับแพ็กเกตที่ส่งสำเร็จหรือส่งผิดพลาดต่อไป และจะเห็นว่าการ poll ด้วย Poll Periodic Timer จะเกิดขึ้นเฉพาะเมื่อมีแพ็กเกตค้างอยู่ในบัฟเฟอร์เพื่อรอการส่งซ้ำถึงปริมาณหนึ่งเท่านั้น หากมีปริมาณไม่ถึงกำหนดก็จะไม่เกิดการ poll ที่ซึ่งทำให้ไม่มีการส่งสถานะกลับจากฝั่งรับ การนำแพ็กเกตที่ส่งสำเร็จออกจากบัฟเฟอร์ส่งก็ไม่สามารถเกิดได้

- ในขณะที่ Poll Timer จะเป็นการ poll เพื่อร้องขอสถานะการรับข้อมูลของฝั่งรับ เพื่อที่ฝั่งส่งเองจะสามารถรับรู้ได้ว่าแพ็กเกตใดส่งสำเร็จ แพ็กเกตใดต้องการการส่งซ้ำ ซึ่งจะเห็นว่า หากไม่มีการใช้งาน Poll Timer จะไม่สามารถเกิดการปรับปรุงข้อมูลสถานะการทำงานการส่งข้อมูล ไม่สามารถรับรู้ว่าแพ็กเกตใดส่งได้สำเร็จและต้องนำออกจากบัฟเฟอร์ แพ็กเกตใดต้องการการส่งซ้ำ แต่หากไม่มีการใช้งาน poll periodic timer กลไกอื่น ๆ ก็ยังคงทำงานได้ต่อไป เพราะกลไกอื่นไม่ได้สนใจจำนวนแพ็กเกตที่ต้องการการส่งซ้ำที่ค้างอยู่ในบัฟเฟอร์ การส่งซ้ำสามารถรับรู้ได้จากการใช้งาน poll timer trigger ได้ นอกจากนี้ความถี่ในการรับสถานะที่บ่อยครั้งช่วยให้การนำแพ็กเกตที่ส่งสำเร็จออกจากบัฟเฟอร์ส่งเกิดได้เร็วขึ้น ลดเหตุการณ์การเกิดการล้นของบัฟเฟอร์ได้
- และได้มีการกำหนดใช้ trigger ในกลุ่มที่สนใจจำนวน PDU ในบัฟเฟอร์ ทั้ง last PDU in transmission buffer และ last PDU in retransmission buffer ด้วย เพื่อหลีกเลี่ยงการเกิดปัญหาการที่แพ็กเกตติดค้างอยู่ในบัฟเฟอร์โดยที่ไม่มีโอกาสจะได้ส่งออกหรือละทิ้งไป[3] แต่ทั้งนี้แบบจำลองที่สร้างขึ้นสามารถรองรับการกำหนดค่าและจำลองผลของ trigger ทั้ง 8 ตัว ดังนั้นหากผู้วิจัยท่านอื่นต้องการนำไปทดสอบกับ trigger อื่นก็สามารถทำได้

สำหรับกลไกการส่ง STATUS ที่ฝั่งรับก็สามารถนำมากำหนดใช้กับแบบจำลองเพื่อทดสอบผลกระทบต่อสมรรถนะของระบบได้เช่นกัน สาเหตุที่ระบบทดสอบไม่ได้กำหนดใช้งาน trigger ตัวจับเวลาในกลไกการส่ง STATUS ก็เนื่องมาจากว่า ระบบที่ออกแบบต้องการแสดงให้เห็นถึงผลกระทบของความถี่ในการ poll ต่อสมรรถนะการส่งข้อมูลของระบบ ซึ่งความถี่ในการ poll นำมาซึ่งความถี่ของการส่ง status report ของฝั่งรับเช่นกัน ดังนั้น หากให้ฝั่งรับมีการใช้งาน trigger ตัวจับเวลาเพื่อส่ง status report ด้วย ผลการทดสอบก็จะไม่สามารถยืนยันได้แน่นอนว่าเกิดจากความถี่ในการ poll เพราะผลที่ได้อาจมีผลของ status report ที่ถูกส่งอันเนื่องจาก trigger ภายในฝั่งรับปะปนมาด้วย เป็นการยากในการวิเคราะห์ผล ในทางกลับกันสามารถเลือกกำหนดใช้ trigger ตัวจับเวลาภายในกลไกการส่ง STATUS แทนการใช้ trigger ตัวจับเวลาของกลไกการ poll

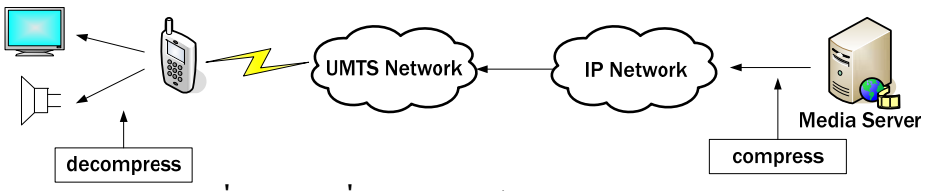
เพื่อทดสอบความถี่ในการส่ง status report โดยปราศจากการร้องขอจากฝั่งส่ง แต่วิธีนี้หากต้องการปรับปรุงสมรรถนะในการส่งให้ดียิ่งขึ้น อาจต้องออกแบบอัลกอริทึ่มและพัฒนาที่ฝั่งรับแทน เพราะผลการทดสอบที่ได้เป็นผลของกลไกที่เกิดขึ้น ณ ฝั่งรับนั่นเอง

4.1.1.4 อัตราการให้บริการของช่องสัญญาณ

เนื่องจากว่าระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 รองรับอัตราการรับส่งข้อมูลที่หลากหลาย ขึ้นกับสภาพและพื้นที่ใช้งาน กล่าวคือ อัตราการรับส่งข้อมูลที่ 144 กิโลบิตต่อวินาที ในสภาพแวดล้อมชนาดใหญ่ พื้นที่รอบนอกที่ห่างไกล และการใช้งานที่มีการเคลื่อนที่ด้วยความเร็วสูง เช่น ในการเคลื่อนที่ของยานพาหนะ อัตราการรับส่งข้อมูลที่ 384 กิโลบิตต่อวินาที สำหรับพื้นที่ครอบคลุมทั่วไป ในสภาพแวดล้อมขนาดเล็ก มีการเคลื่อนที่ไม่เร็วจนเกินไป เช่น ในการเดินทางด้วยเท้า และอัตราการรับส่งข้อมูลที่ 2 เมกะบิตต่อวินาที ในสภาพแวดล้อมขนาดเล็กมาก เช่น ในตึกอาคาร หรือสำนักงาน [10] เพราะฉะนั้นการออกแบบการทดสอบเพื่อให้ครอบคลุมกรณีทั่วไปให้ได้มากที่สุด เพื่อประโยชน์ในการนำผลไปประยุกต์ใช้จริง จึงเลือกใช้ความสามารถของช่องสัญญาณที่ความเร็ว 384 กิโลบิตต่อวินาทีเป็นสภาวะแวดล้อมของระบบทดสอบ ซึ่งเป็นกรณีที่จะเกิดขึ้นได้บ่อยในการใช้งานจริงของผู้ใช้ส่วนใหญ่ เพราะถึงผู้ใช้งานจะนั่งใช้งานภายในอาคารที่มีอัตราการรับส่งข้อมูลที่ 2 เมกะบิตต่อวินาทีแต่เมื่อผู้ใช้เกิดการเคลื่อนที่ เช่น ลุกขึ้นเดิน อัตราการรับส่งข้อมูลก็จะลดลงเหลือ 384 กิโลบิตต่อวินาที ทั้งนี้แล้วแม้ว่าการทดลองจะไม่ได้ออกแบบที่จะเปลี่ยนแปลงค่าอัตราการให้บริการ แต่ก็สามารถนำผลจากการทดสอบที่ให้ค่าอัตราการให้บริการเป็น 384 กิโลบิตต่อวินาทีไปทำนายผลสำหรับค่าอัตราการให้บริการอื่นได้ว่าผลจะออกมามีแนวโน้มเป็นอย่างไร เพราะการที่อัตราการให้บริการสูงขึ้นก็เปรียบเสมือนการมีท่อส่งน้ำขนาดใหญ่ขึ้น สามารถส่งน้ำได้มากกว่าและเร็วกว่าท่อขนาดเล็กนั่นเอง

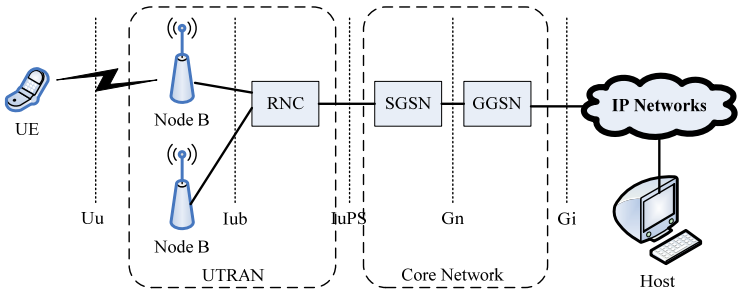
4.1.2 การออกแบบเครือข่ายจำลอง

เนื่องจากกรณีจำลองที่เลือกทดสอบเป็นการส่งข้อมูลวิดีทัศน์จากเครื่องบนเครือข่ายอินเตอร์เน็ตไปยังเครื่องโทรศัพท์เคลื่อนที่ของผู้ใช้ ดังนั้น เส้นทางการส่งผ่านข้อมูลสามารถแสดงได้ดังรูปที่ 4.1 เครือข่ายโทรศัพท์เคลื่อนที่เป็นเครือข่ายที่มีทรัพยากรจำกัด ข้อมูลวิดีทัศน์ที่ส่งผ่านจึงเป็นต้องมีการบีบอัดก่อนทำการส่ง และเมื่อข้อมูลเดินทางไปถึงยังฝั่งรับ ณ ฝั่งรับก็จำเป็นจะต้องรองรับวิธีการคลายการบีบอัดนั้น ๆ เพื่อทำการแสดงผลออกทางหน้าจอและเครื่องขยายเสียง



รูปที่ 4.1 การสื่อสารวิดีทัศน์ผ่านเครือข่าย UMTS

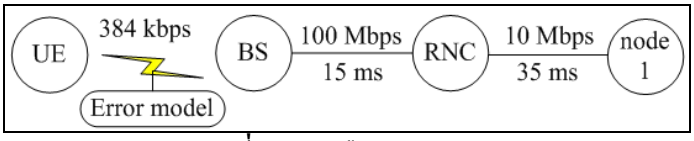
สำหรับเครือข่ายส่วนที่เป็น UMTS Network ภายในสามารถอธิบายโครงสร้างได้โดยอ้างอิงจากสถาปัตยกรรมของเครือข่าย UMTS ที่แบ่งออกเป็น 3 ส่วนใหญ่ ๆ คือ User Equipment (UE), UMTS Terrestrial Radio Access Network (UTRAN) และ Core Network ดังรูปที่ 4.2 [11] โดย UTRAN ประกอบด้วย Bass station (BS) หลาย ๆ โหนดเชื่อมต่อไปยัง Radio Network Controller (RNC) สำหรับ Core network ประกอบ Serving GPRS Support Node (SGSN) และ Gateway GPRS Support Node (GGSN) ซึ่งมีหน้าที่หลักเพื่อทำการค้นหาเส้นทางและส่งข้อมูลผู้ใช้ นอกจากนี้ยังเก็บฐานข้อมูลและฟังก์ชันการจัดการเครือข่ายไว้อีกด้วย และ UE ก็คือสถานีเคลื่อนย้าย (mobile station) ที่เชื่อมต่อกับ Node B บนจุดเชื่อมต่อทางวิทยุของ UMTS (Uu)



รูปที่ 4.2 สถาปัตยกรรม UMTS

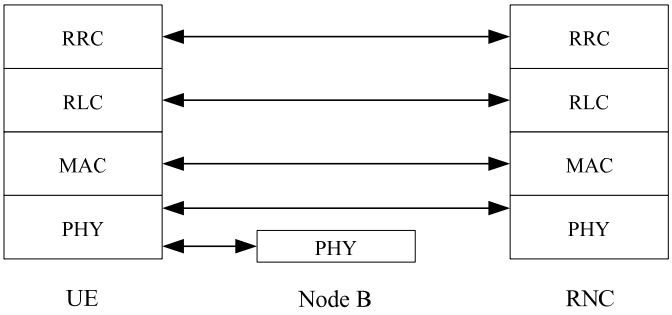
เนื่องจากโปรโตคอลที่สนใจทำงานอยู่บนจุดเชื่อมต่อ Uu ดังนั้นเพื่อลดประเด็นพิจารณาที่ไม่เกี่ยวข้อง จึงออกแบบเครือข่ายจำลองโดยตัดส่วนของ Core Network ออกไป เหลือไว้เพียง UE, UTRAN และ Host ที่อยู่บน IP Network สำหรับเป็นแหล่งส่งข้อมูลให้กับ UE ดังนั้น รูปแบบเครือข่ายจำลองที่ใช้ในการทดสอบจะประกอบด้วยโหนด 4 โหนดที่ทำหน้าที่เป็น UE, Base station, RNC และโหนดบนเครือข่ายอินเตอร์เน็ต (node 1) ลิงค์ระหว่างโหนด Base station จนถึงโหนด node 1 สามารถกำหนดให้มีแบนด์วิดท์และเวลาที่ใช้ส่งข้อมูล (latency) เท่าใดก็ได้ด้วยเหตุเพราะอยู่นอกเหนือจุดเชื่อมต่อ Uu ที่งานวิจัยสนใจ แต่เพื่อสะดวกแก่การคำนวณและวิเคราะห์ผลภายหลัง จึงให้ลิงค์ระหว่างแต่ละโหนดมีแบนด์วิดท์และเวลาที่ใช้ส่งข้อมูลดังแสดงในรูปที่ 4.3 พร้อมกับบังคับไม่ให้มีความผิดพลาดเกิดขึ้นในลิงค์อื่น ๆ ยกเว้นระหว่าง UE และ BS ที่เชื่อมต่อกันผ่านช่องสัญญาณ DCH ของ UMTS ที่ความเร็ว 384 kbps เท่านั้น

ทั้งนี้ก็เพื่อไม่ให้ความผิดพลาดจากส่วนอื่นที่อยู่นอกเหนือจุดสนใจมารบกวนกับผลลัพธ์ที่ได้

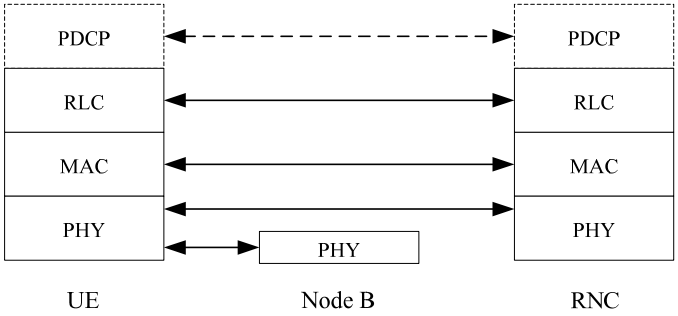


รูปที่ 4.3 เครือข่ายจำลอง

ภายในโหนด UE, BS และ RNC จะด้วยโปรโตคอลสแทคที่ทำหน้าที่ต่าง ๆ ตามมาตรฐานของระบบ UMTS ดังรูปที่ 4.4 และรูปที่ 4.5 ที่แสดงการสิ้นสุดของโปรโตคอลสำหรับช่องสัญญาณ DCH ในระนาบควบคุมและระนาบผู้ใช้ ตามลำดับ ในกรณีที่ระบบมีขนาดใหญ่ มีความหลากหลายอยู่ร่วมกัน ชั้น physical จะไปสิ้นสุดที่ RNC แต่ถ้าเป็นระบบขนาดเล็กไม่มีความหลากหลายในการทำงาน ชั้น physical ก็จะสิ้นสุดที่ Node B การจำลองกำหนดให้มีการส่งข้อมูลผ่าน Node B หรือ Base station ก่อนแล้วจึงส่งต่อให้ยัง RNC การส่งข้อมูลจะทำงานบนระนาบผู้ใช้ ดังนั้นการวัดสมรรถนะก็จะทำการวัดบนระนาบผู้ใช้นี้ นอกจากนี้การจำลองจะไม่มีการใช้งานโปรโตคอล PDCP สำหรับการบีบอัด header ของแพ็กเกต

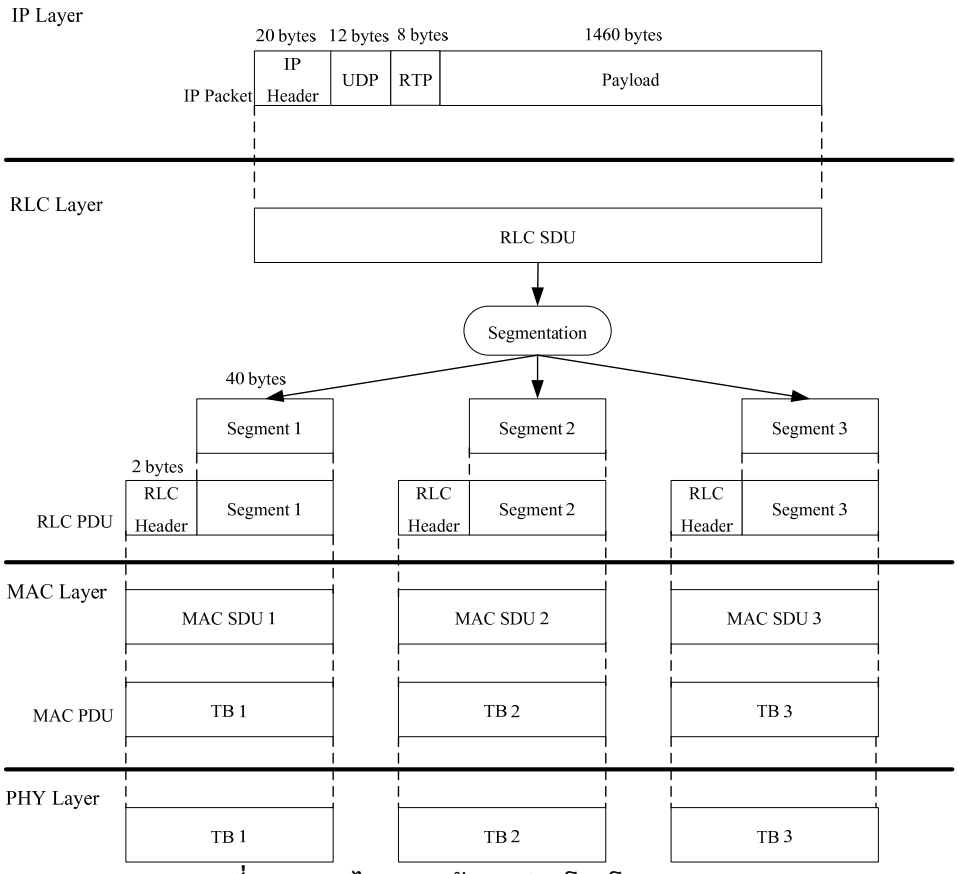


รูปที่ 4.4 การสิ้นสุดของโปรโตคอลสำหรับช่องสัญญาณ DCH ส่วนระนาบควบคุม (control plane) [12]



รูปที่ 4.5 การสิ้นสุดของโปรโตคอลสำหรับช่องสัญญาณ DCH ส่วนระนาบผู้ใช้ (user plane) [12]

การส่งผ่านข้อมูลผ่านโปรโตคอลสแทคจากชั้นโปรแกรมประยุกต์ ไปจนถึงชั้น Physical สามารถแสดงได้ดังรูปที่ 4.6 ที่ซึ่ง RLC ทำงานในโหมด AM และชั้น MAC ไม่มีการเพิ่ม header ลงไปในแพ็กเกต เนื่องจาก UE มีเพียงช่องสัญญาณ dedicated เพียงช่องเดียว ดังนั้น ชั้น MAC อาจไม่จำเป็นต้องเพิ่ม header ใส่ลงไปในแพ็กเกตก็ได้ [38] ด้วยเหตุที่ MAC ไม่สามารถตัดแบ่ง SDU ขนาดใหญ่ให้มีขนาดเป็นไปตามที่รูปแบบแพ็กเกตการส่งข้อมูล เพราะฉะนั้นการตัดแบ่ง SDU จึงจำเป็นที่ RLC ต้องจัดการให้เรียบร้อยก่อนนำส่งผ่านชั้น MAC โดยโปรโตคอล MAC ใช้ transport block เป็นยูนิตในการขนส่ง ขนาดของ transport block สำหรับแต่ละช่องสัญญาณถูกกำหนดจากชั้น physical ในรูปแบบของการให้นิยามชุดของรูปแบบการส่ง (transport format set definition) โปรโคตอลชั้น MAC ทำการมัลติเพล็ก PDU จากโปรโตคอลชั้นบนเข้าสู่ชุด transport block แล้วขนย้ายผ่านช่องสัญญาณ dedicated และโดยปกติสำหรับทุกโหมดการทำงานของ RLC การค้นหาข้อผิดพลาด CRC จะทำที่ชั้น physical และผลของการตรวจสอบ CRC จะถูกส่งให้ชั้น RLC พร้อมกับข้อมูลแท้จริง แต่ในแบบจำลองจะไม่มีการเรียกใช้งาน CRC ของชั้น physical เนื่องจากต้องการตรวจสอบกลไกการกู้คืนข้อมูลของ RLC โดยที่ไม่มีกลไกอื่นใดที่ช่วยในการกู้คืนข้อมูลมาเกี่ยวข้อง



รูปที่ 4.6 การไหลของข้อมูลผ่านโปรโตคอลสแทค

4.1.3 กรณีศึกษาที่ทำการทดสอบ

เหตุการณ์จำลองที่เลือกทดสอบ คือ ผู้ใช้งานโทรศัพท์มือถือทำการเรียกดูไฟล์ข้อมูลวิดีโอ MPEG-4 จากเครือข่ายอินเตอร์เน็ต ด้วยเนื่องจากการใช้งานของผู้ใช้โทรศัพท์ในยุคต่อไปจะเป็นมากกว่าแค่การสื่อสารด้วยเสียงเท่านั้น และเป้าหมายหลักของระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 ก็มุ่งเน้นให้บริการด้านสื่อประเภทมัลติมีเดียด้วย สื่อประเภทนี้กำลังเป็นที่นิยมเพิ่มสูงขึ้นเรื่อย ๆ ในปัจจุบัน หากคุณภาพการสื่อสารด้วยสื่อมัลติมีเดียนี้ ไม่เป็นที่น่าพึงพอใจ ก็จะไม่มีผู้ใช้งาน ทำให้เป้าหมายการให้บริการของระบบล้มเหลว ดังนั้น งานวิจัยจึงเลือกทดสอบประสิทธิภาพด้วยการส่งข้อมูลประเภทวิดีโอที่มีการบีบอัดแบบ MPEG-4 ซึ่งเหมาะกับการส่งบนเครือข่ายไร้สายที่มีทรัพยากรอย่างจำกัด ทั้งนี้ ก็ได้มีงานวิจัยส่วนนึงที่เลือกสื่อทดสอบเป็นข้อมูลการเรียกดูเว็บไซต์ และการดาวน์โหลดไฟล์จากอินเตอร์เน็ต[4][5] ซึ่งสามารถหาข้อมูลเพื่อทำการศึกษาเพิ่มเติมได้เช่นกัน

4.1.3.1 ข้อมูลจำลองวิดีโอ MPEG-4

เนื่องจากการทดสอบ ผลลัพธ์ที่ได้จากการจำลองจะมีข้อมูลด้านตัวเลขค่อนข้างมาก การใช้ไฟล์วิดีทัศน์จริงในการทดสอบคงเป็นการยากสำหรับการคำนวณและวิเคราะห์ผล ดังนั้นผู้วิจัยจึงสร้างวิดีโอ MPEG-4 จำลองขึ้นเพื่อเป็นข้อมูลสำหรับส่งผ่านเครือข่ายจำลองที่ออกแบบไว้ โดยที่วิดีโอจำลอง MPEG-4 นั้น ไม่ได้เป็นข้อมูลวิดีโอจริงแต่เป็น trace file ที่ถูกสร้างขึ้นเพื่อใช้ในการส่งบน ns-2 ซึ่งจำลองให้มีชนิดของเฟรมวิดีโอเป็น i-frame และ p-frame เช่นเดียวกับรูปแบบวิดีโอจริงเมื่อถูกบีบอัดเป็นชนิด mpeg-4 แต่ความยาวของเฟรมแต่ละชนิดจะคงที่เสมอ ซึ่งเหตุจำลองเบื้องต้นต้องการให้เมื่อเริ่มต้นทำการส่งข้อมูลบนช่องสัญญาณที่ไม่มีอัตราความผิดพลาดใด ๆ ข้อมูลที่ส่งออกไปนั้นไม่ได้ใช้งานช่องสัญญาณจนเต็ม เพื่อเว้นส่วนที่ว่างไว้พิจารณาผลเมื่อมีอัตราความผิดพลาดเกิดขึ้นในช่องสัญญาณ จึงกำหนดให้ ความยาวของ i-frame เท่ากับ 9000 ไบท์ และความยาวของ p-frame เท่ากับ 2000 ไบท์ ไฟล์วิดีโอที่มีระยะห่างระหว่างเฟรมหลัก (key frame interval) เท่ากับ 5 มีอัตราการส่ง 10 เฟรมต่อวินาที ทั้งนี้อาจกำหนดความยาวของ i-frame และ p-frame รวมถึงระยะห่างระหว่างเฟรมหลักและอัตราการส่งเฟรมเป็นค่าอื่นที่ยังคงทำให้ช่องสัญญาณว่างพอสำหรับการรองรับอัตราความผิดพลาดของช่องสัญญาณที่จะเกิดขึ้นก็ได้ แต่สำหรับค่าระยะห่างระหว่างเฟรมหลักจะไม่นิยมใช้ค่าที่ต่ำกว่า 5 ลงมา เนื่องจาก ค่าที่ต่ำกว่านี้ทำให้ผิดวัตถุประสงค์ของการบีบอัด เพราะข้อมูลจะถูกบีบอัดได้น้อย โดยที่มีการพิสูจน์แล้วว่า ค่าระยะห่างระหว่างเฟรมหลักตั้งแต่ 5 ขึ้นไป ก็สามารถให้ค่า PSNR ที่สูงได้ [25] ซึ่งค่า PSNR นี้เป็นค่าที่บอกถึงความชัดเจนของภาพในแต่ละเฟรมวีดิทัศน์ ถ้าค่า PSNR มากแสดงว่าเฟรมวีดิทัศน์ที่ส่งให้กับผู้รับนั้นมีคุณภาพใกล้เคียงกับวีดิ

ทัศน์ต้นฉบับมาก โดยรายละเอียดข้อมูลวิดีโอที่ใช้ในการทดลองแสดงไว้ในตารางที่ 4.2

ตารางที่ 4.2 พารามิเตอร์สำหรับวิดีโอ MPEG-4

จำนวนเฟรมที่ส่ง (เฟรม)	1200
ระยะห่างระหว่างเฟรมหลัก (เฟรม)	5
ความยาวของ I-frame (ไบท์)	9000
ความยาวของ P-frame (ไบท์)	2000
อัตราการส่ง (เฟรมต่อวินาที)	10
เวลาที่ใช้ส่ง (วินาที)	120

แต่หากมีการปรับอัตราการให้บริการของช่องสัญญาณ รายละเอียดข้อมูลวิดีโอที่ใช้ก็ต้องเปลี่ยนแปลงไป โดยค่าที่จะเลือกนำมาใช้ต้องมีอัตราส่วนระหว่างข้อมูลวิดีโอที่ส่งและความจุของช่องสัญญาณเท่าเดิม เนื่องจากการวิเคราะห์ผลจะพิจารณาข้อมูลวิดีโอตามอัตราส่วนของความจุของช่องสัญญาณที่มี

4.2 เครื่องมือในการสร้างแบบจำลอง

เนื่องจากเป็นการยากที่จะใช้ระบบจริงของโทรศัพท์เคลื่อนที่ยุคที่ 3 ในการทดสอบสมรรถนะการทำงานของโปรโตคอล นั้นดังทางเลือกที่เป็นไปได้ คือ การสร้างแบบจำลองที่สามารถจำลองทำงานของระบบได้ โดยใช้โปรแกรมสร้างแบบจำลองทางเครือข่ายคอมพิวเตอร์ที่มีให้เลือกมากมายในปัจจุบัน แต่สำหรับงานวิจัยนี้เลือกใช้โปรแกรมจำลอง Network Simulator หรือ ns ด้วยเหตุผลที่ โปรแกรมจำลอง ns เป็นโปรแกรมจำลองที่พัฒนาขึ้นแบบโอเพนซอร์สและใช้แนวคิดของการโปรแกรมเชิงอ็อบเจกต์ จึงเปิดโอกาสให้นักพัฒนาได้สร้างแบบจำลองของโปรโตคอลทางเครือข่ายคอมพิวเตอร์ แล้วนำโปรโตคอลเหล่านั้นมาทำงานเป็นระบบร่วมกับโปรโตคอลอื่น ๆ เพื่อการศึกษาและเปรียบเทียบสมรรถนะหรือข้อจำกัดของโปรโตคอลได้ในสภาพแวดล้อมที่กำหนดขึ้นได้ โปรแกรม ns ยังไม่ได้มีการจัดเตรียมแบบจำลองสำหรับเครือข่าย UMTS ไว้ก่อน ดังนั้นจึงจำเป็นต้องมีการสร้างโมดูลเพิ่มเติมเพื่อปรับแต่งโปรแกรม ns ให้สามารถจำลองการทำงานของโปรโตคอลของระบบ UMTS ตามที่งานวิจัยต้องการได้ โดยโมดูลที่นำมาใช้จำลองระบบ UMTS นั้น เป็นโมดูลสำเร็จ เพราะโปรโตคอลของระบบ UMTS มีความซับซ้อนสูง หากสร้างเองต้องใช้เวลานาน และต้องทดสอบจนมั่นใจว่าโมดูลที่สร้างทำงานได้ถูกต้อง ครบถ้วน มีหลายกลุ่มวิจัยที่พยายามสร้างโมดูลจำลองการทำงานของโปรโตคอลของระบบ UMTS ได้แก่ โมดูลที่พัฒนาโดย Alfredo Todini และ FrancescoVacirca จากภาควิชา INFOCOM มหาวิทยาลัยแห่งโรม ประเทศอิตาลี [31], โมดูลที่พัฒนาโดย Pablo Martin และ

Paula Ballester จาก Strathclyde University of Glasgow ประเทศสก็อตแลนด์ [32] และ EURANE หรือ Enhanced UMTS Radio Access Network ถูกพัฒนาภายใต้โครงการ SEACORN สำหรับ Ericsson Telecommunicate B.V. [33] โมดูลท้ายสุดถูกใช้กันอย่างหลากหลาย เนื่องจาก สามารถจำลองการทำงานของระบบ UMTS ได้ครอบคลุม พัฒนารองรับการทำงานทั้ง 3 โหนดด้วยกันคือ Radio Network Controller (RNC), Base station (BS) และ User Equipment (UE) สนับสนุนการทำงานของช่องสัญญาณขนส่งหรือ transport channels ต่างได ได้แก่ FACH, RACH, DCH และ HS-DSCH รองรับการส่งข้อมูลจากหลายโปรแกรมประยุกต์ ในขณะที่ 2 โมดูลแรกมีความซับซ้อนสูง และสามารถรองรับการส่งข้อมูลที่ส่งจาก UDP เท่านั้น สำหรับรายละเอียดของโมดูลสามารถอ่านได้จากภาคผนวก ก.

4.3 สรุป

เครื่องมือที่ใช้สร้างแบบจำลองเพื่อทดสอบการทำงานของเครือข่ายที่ใช้งานอยู่ในปัจจุบันมีอยู่หลากหลาย ns ก็เป็นเครื่องมือนึงที่นิยมใช้กันอย่างมากในปัจจุบัน เนื่องจากเป็นโปรแกรมชนิด Open Source ที่มีความซับซ้อนไม่สูงนัก จึงถูกเลือกมาใช้ในการทดสอบการทำงานของโปรโตคอลระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 เนื้อหาภายในบทยังได้กล่าวถึงการเพิ่มเติมโมดูลที่จำลองการทำงานระบบ UMTS ลงในโปรแกรม ns เพื่อใช้โมดูลที่ทำการเพิ่มเติมนี้ในการจำลองการทำงานการส่งข้อมูลผ่านโปรโตคอล RLC และเนื่องด้วยการทำงานทั้งหมดระบบมีความซับซ้อนสูง ดังนั้นเพื่อให้ได้มาซึ่งแบบจำลองต้นแบบที่สามารถประเมินสมรรถนะการทำงานของโปรโตคอลชั้นที่ 2 นั้น จำเป็นต้องสนใจและเลือกใช้งานเพียงบางส่วน สำหรับเหตุผลในการเลือกแต่ละส่วนนั้นได้ถูกกล่าวไปแล้ว ซึ่งเงื่อนไขที่ถูกเลือกทั้งหมดจะถูกสร้างเป็นแบบจำลองและใช้ในการทดลองซึ่งจะกล่าวถึงรูปแบบการทดลองในบทที่ 5