บทที่ 3

โปรโตคอลควบคุมการเชื่อมโยงคลื่นวิทยุ

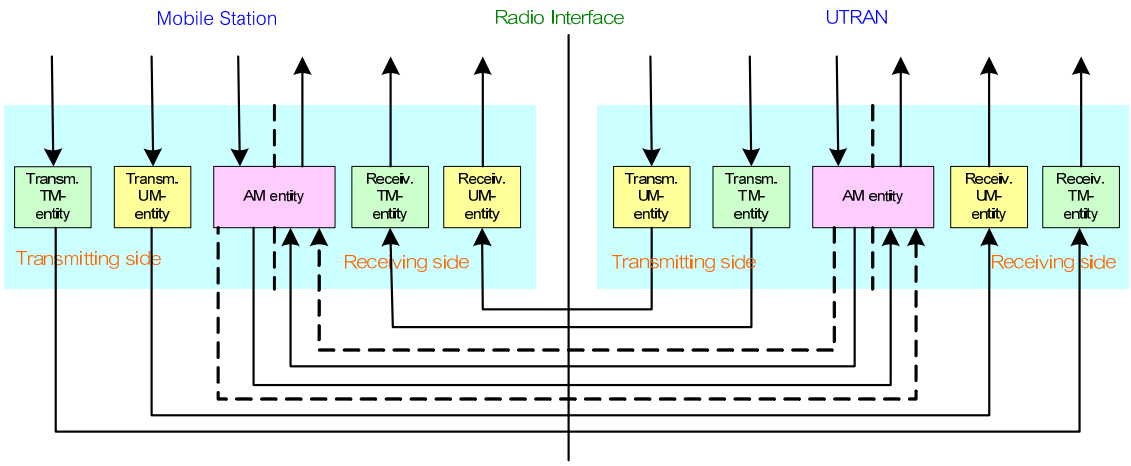
Radio Link Control Protocol: RLC

ในบทที่ 2 ได้กล่าวถึงโครงสร้างและสถาปัตยกรรมโดยภาพรวมของระบบ UMTS ไปแล้ว และเนื่องจากวิทยานิพนธ์นี้สนใจการประเมินประสิทธิภาพของโปรโตคอลชั้นที่ 2 ซึ่งประกอบด้วยโปรโตคอลเอ็นทิที้ย่อย ๆ ได้แก่ RLC, MAC, PDCP และ BMC แต่เนื่องจากโปรโตคอล ในชันที่ 2 ที่มีหน้าที่หลักในการจัดการควบคุมความผิดพลาดในการส่งข้อมูลของโปรโตคอลชั้นบน คือ RLC ดังนั้นผู้จัดทำจึงเลือกโปรโตคอลนี้เป็นหลักในการศึกษาและประเมินประสิทธิภาพในการทำงาน ซึ่งในบทที่ 3 นี้จะกล่าวถึงโครงสร้างและการทำงานโดยละเอียดของโปรโตคอล RLC

Radio Link Control (RLC) Protocol ให้บริการในเรื่องของการตัดแบ่งข้อมูลและการส่งซ้ำสำหรับทั้งข้อมูลที่เป็นสัญญาณควบคุมและข้อมูลที่ใช้งาน RLC แต่ละตัวจะถูกตั้งค่าโดย RRC เพื่อที่จะให้ทำงานใน 3 โหมดคือ Transparent Mode (TM), Unacknowledged Mode (UM) และ Acknowledged Mode (AM) บริการของชั้น RLC ที่จัดให้กับระนาบควบคุมจะถูกเรียกว่า “Signaling Radio Bearer (SRB)” และสำหรับบริการที่จัดให้กับระนาบให้บริการซึ่งเป็นบริการที่ไม่ได้ถูกเรียกใช้โดย PDCP และ BMC จะเรียกว่า “Radio Bearer (RB)” นอกนั้นบริการ RB จะได้มาจาก PDCP และ BMC

3.1 สถาปัตยกรรมของชั้น RLC

ภายในชั้น RLC จะประกอบด้วยเอ็นทิที้หลายอันซึ่งแบ่งออกเป็นสามชนิดตามโหมดการทำงาน คือ TM, UM และ AM ตามที่ได้กล่าวไปแล้วข้างต้น



รูปที่ 3.1 สถาปัตยกรรมของโปรโตคอล RLC [27]

จากรูปที่ 3.1 แสดงโครงสร้างของชั้นโปรโตคอล RLC ซึ่งในแต่ละ TM หรือ UM entity จะถูกติดตั้งเพื่อเป็น RLC entity ที่ใช้ในการส่งหรือรับ RLC PDUs แต่ใน AM entity หนึ่งจะประกอบด้วยฝั่งรับและฝั่งส่งของ PDU ใน entity

ในแต่ละขั้นตอนจะถูกนิยามระหว่างฝั่งรับและฝั่งส่ง ใน UM และ TM โดยที่ entity ที่ใช้ในการส่งเรียกว่าฝั่งส่ง และ entity ที่ใช้ในการรับเรียกว่าฝั่งรับ แต่ใน AM entity จะทำงานเป็นฝั่งส่งและฝั่งรับโดยขึ้นอยู่กับแต่ละ procedure ซึ่งฝั่งส่งและฝั่งรับของทั้ง 3 แบบจะอยู่ทั้งทาง UE และ UTRAN

ในแต่ละ RLC TM และ UM entity จะใช้หนึ่งช่องสัญญาณลอจิกในการส่งหรือรับ PDUs แต่ใน AM entity จะสามารถถูกติดตั้งให้ใช้ช่องสัญญาณลอจิกหนึ่งหรือสองช่อง ในการส่งหรือรับ PDU และ control PDU ถ้าสองช่องสัญญาณลอจิกถูกติดตั้ง จะต้องเป็นชนิดเดียวกัน (DTCH หรือ DCCH) ซึ่งเส้นประในรูปจะแสดงถึงการที่จะสามารถรับและส่ง RLC PDU แยกช่องสัญญาณได้ เช่น control PDU ของช่องสัญญาณหนึ่ง และ PDU ของอีกช่องสัญญาณหนึ่ง โดยหน้าที่การทำงานของ RLC มีดังนี้

- การแบ่งและรวมกลับของข้อมูล (Segmentation and reassembly) ฟังก์ชันนี้จะทำการแบ่งและรวมกลับของข้อมูลของ SDU ที่มีขนาดใหญ่จากชั้นบนให้เป็นหลาย ๆ PDU ที่มีขนาดเล็กลง โดยจะมีการกำหนดขนาดของ PDU ตามอัตราการส่งที่น้อยที่สุดได้สำหรับการบริการของ RLC entity ดังนั้นทำให้สามารถกำหนดอัตราส่งของการบริการได้ ซึ่งหลาย ๆ PDU ของ SDU หนึ่ง ๆจะต้องถูกส่งภายใน Transmission Time Interval (TTI)
- การต่อข้อมูล (Concatenation) ใช้กรณีเนื้อข้อมูลของ SDU ไม่สามารถใส่ได้พอดีในหลาย ๆ PDU ซึ่งส่วนแบ่งแรกของ SDU จะถูกต่อเข้ากับส่วนแบ่งสุดท้ายของ SDU ก่อนหน้า

- การเติมเต็มข้อมูล (Padding) ใช้ในกรณีที่ PDU ที่ไม่สามารถใส่ส่วนแบ่งของ SDU ถัดไปได้ และต้องขนาดของ PDU ที่แน่นอน โดยการเติมเต็มของข้อมูลซึ่งจะใส่ข้อมูลใด ๆลงไปก็ได้ ซึ่งจะถูกเพิกเฉยโดยฝั่งรับ
- การส่งข้อมูลของผู้ใช้ (Transfer of user data) RLC จะสนับสนุนการส่งข้อมูลในสามโหมด คือ TM UM และ AM ซึ่งการส่งข้อมูลจะถูกควบคุมโดยการติดตั้งของคุณภาพการบริการ
- การแก้ไขข้อผิดพลาดให้ถูกต้อง (Error correction) ฟังก์ชันนี้จะทำการแก้ไขโดยการส่งข้อมูลใหม่ในโหมด AM
- การส่งข้อมูลอย่างเป็นลำดับของ PDUs ของชั้นบน (In-sequence delivery of higher layer PDUs) ฟังก์ชันนี้จะถูกสงวนในการส่งข้อมูลใน AM แต่ถ้าฟังก์ชันนี้ไม่ถูกติดตั้งจะให้บริการส่งข้อมูลอย่างไม่เป็นลำดับแทน
- การตรวจสอบการซ้ำกันของข้อมูล (Duplicate detection) จะทำหน้าที่ในการตรวจสอบการซ้ำกันในการรับ PDUs ซึ่งจะส่งเพียงข้อมูลเดียวไปยังชั้นบน
- การควบคุมการไหลของข้อมูล (Flow control) เป็นฟังก์ชันที่ยอมให้ฝั่งรับมีการควบคุมอัตราในฝั่งส่ง โดยการส่งข้อมูลในการควบคุม
- การตรวจสอบหมายเลขของลำดับข้อมูล (Sequence number check) เป็นฟังก์ชันที่รับประกันของการรวมกลับของ PDUs และตรวจสอบความผิดพลาดของข้อมูล SDU โดยอาศัยหมายเลขของลำดับข้อมูลใน PDUs ซึ่งหากพบว่าข้อมูล SDU ผิดพลาด ก็จะทำการตัด SDU ทิ้งไป และรายงานข้อผิดพลาดดังกล่าวไปยังชั้น RRC
- การตรวจสอบข้อผิดพลาดของโปรโตคอลและกู้คืนกลับ (Protocol error detection and recovery) ฟังก์ชันนี้จะทำการตรวจสอบและกู้คืนการทำงานของโปรโตคอล RLC เมื่อพบข้อผิดพลาด
- การเข้ารหัส (Ciphering) จะเป็นการเข้ารหัสข้อมูลโดยมีเวลา และ ciphering key เป็นอินพุต เพื่อให้ข้อมูลมีความปลอดภัยมากยิ่งขึ้น
- การห้ามหรือกลับคืนของการส่งข้อมูล (Suspend/resume function for data transfer) ในการห้ามการส่งข้อมูลนั้นจะถูกใช้ก็ต่อเมื่ออยู่ในกระบวนการควบคุมความปลอดภัย โดย ciphering key ที่เหมือนกันจะถูกใช้ทั้งสองฝั่ง โดยทั้งสองฟังก์ชันจะถูกควบคุมโดยชั้น RRC ผ่านทางอินเตอร์เฟสควบคุม

โดยที่แต่ละโหมดของชั้น RLC จะให้บริการที่มีความแตกต่างกันดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 แสดงบริการที่มีในแต่ละโหมดการทำงานของชั้น RLC

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

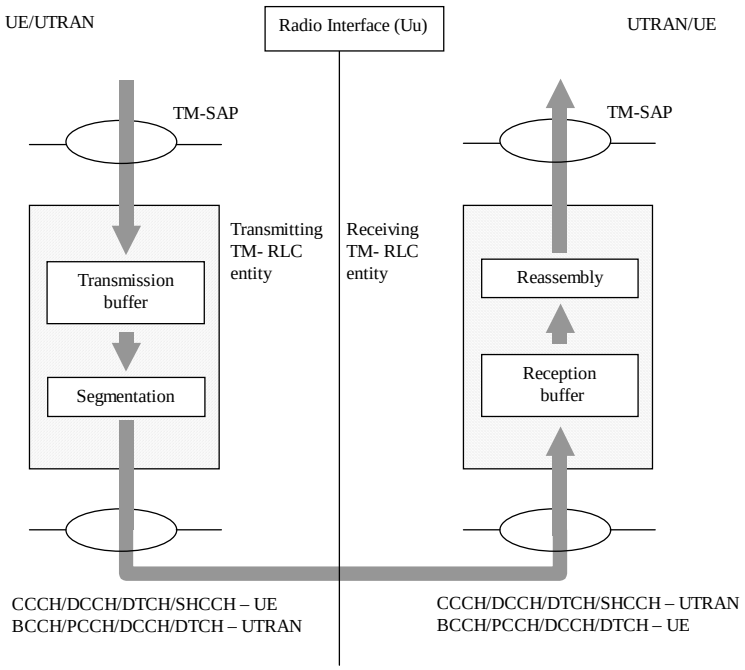
สำหรับองค์ประกอบและการทำงานของแต่ละโหมดจะอธิบายในหัวข้อถัดไป

3.2 องค์ประกอบภายในเอ็นทิที้และบริการ 3 โหมดของ RLC

โปรโตคอล RLC มีบริการสนับสนุนแก่ชั้นบน 3 ชนิด 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 โดยโหมดของบริการที่จะเลือกใช้ดูจากชนิดของ traffic ที่ถูกส่ง เช่น TM ควรจะถูกใช้เมื่อ traffic ที่จะส่งเป็นข้อมูลเสียง และ AM จะถูกปรับแต่งใช้สำหรับบริการ packet data ขบวนการของฝั่งรับและฝั่งส่งของทั้ง 3 โหมดเป็นดังต่อไปนี้

3.2.1 Transparent Mode (TM) RLC Entities

จะเป็นโหมดในการส่งข้อมูลที่ไม่มีการใส่ Header เพิ่มเข้าไป หากข้อมูลที่ส่งเกิดข้อผิดพลาดก็จะถูกตัดทิ้งไป หรือถูกบันทึกข้อผิดพลาดในฝั่งรับ การส่งข้อมูลจะใช้สำหรับการส่งสตรีมจากชั้นบนที่ไม่มีการแบ่งข้อมูลมา เช่น ข้อมูลเสียง เป็นต้น ในการส่งข้อมูลในโหมด TM นั้นจะขึ้นกับขนาดของการส่งข้อมูล โดยที่หากมีขนาดใหญ่มากจะต้องทำการแบ่งข้อมูลในฝั่งส่งและต่อข้อมูลในฝั่งรับ ซึ่งในการส่งข้อมูลระหว่างสองเอ็นทิที้ TM RLC จะเป็นดังรูปที่ 3.2



รูปที่ 3.2 เอ็นทิที้ของ RLC เมื่อทำงานในโหมด Transparent Mode (TM) [27]

3.2.1.1 บริการของโหมด TM

TM บริการสำหรับการส่ง PDU ของชั้นบนโดยไม่มีการเพิ่มรายละเอียดโปรโตคอลใด ๆ การส่งข้อมูลในโหมด TM นั้นจะขึ้นกับขนาดของการส่งข้อมูล หากมีขนาดใหญ่มากจะต้องทำการแบ่งข้อมูล (Segmentation) ในฝั่งส่งและต่อข้อมูล (Reassembly) ในฝั่งรับ โดยโปรโตคอลชั้นบนจะเป็นผู้กำหนดว่าให้ RLC มีการทำ segmentation และ reassembly หรือไม่ก็ได้ ถ้าหากชั้นบนกำหนดให้มีการ segmentation/reassembly และขนาดของ RLC Service Data Unit (SDU) (โดยปกติจะมีขนาดเป็นจำนวนเท่าของ 100 byte) ใหญ่กว่าขนาดของ TMD PDU (ส่วนใหญ่มีขนาดเป็นจำนวนเท่าของ 12 byte) ถูกใช้โดยชั้นต่ำลงมา สำหรับ TTI (Transmission Time Interval) นั้น ๆ ฝั่งส่งจะ segment RLC SDU เพื่อให้พอเหมาะกับ

TMD PDU โดยไม่มีการเติม RLC header โดย TMD PDU ทั้งหมดที่มาจากการ segment RLC SDU ตัวเดียวกันจะถูกส่งภายใน 1 TTI และไม่มี segment จาก RLC SDU อื่นถูกส่งไปใน TTI นั้น ในทางกลับกัน ถ้าขนาดของ RLC SDU เล็กกว่าขนาดของ TMD PDU เสมอและชั้นบนไม่ได้กำหนดให้ RLC จัดการเรื่อง segment/reassembly แล้ว RLC SDU มากกว่า 1 ตัวสามารถถูกส่งได้ใน 1 TTI โดยใส่ RLC SDU หนึ่งลงใน TMD PDU หนึ่ง แต่ TMD PDU ทั้งหมดใน TTI เดียวกันจะต้องมีขนาดเท่ากัน เมื่อฝั่งรับได้รับ TMD PDU ผ่าน logical channel จากชั้นล่าง มันก็จะทำการ reassembly PDU ไปเป็น RLC SDU (ถ้าหากชั้นบนกำหนดไว้ว่าต้องทำ) และส่ง RLC SDU ไปยังชั้นบนทำงานต่อไป จากบริการที่กล่าวมาข้างต้นสามารถแยกเป็นการทำงานของ RLC ฝั่งรับ และ RLC ฝั่งส่งได้ดังนี้

3.2.1.2 เอ็นทิที้ TM RLC ฝั่งส่ง

TM RLC entity ในฝั่งที่ทำการส่งจะรับ RLC SDU จากชั้นบนผ่านทาง Transparent Mode-Service Access Point (TM-SAP) ซึ่งขนาดของ SDU ทั้งหมดจะต้องมีขนาดเป็นจำนวนเท่าของขนาดของ Transparent Mode Data (TMD) PDU หนึ่ง ถ้าการติดตั้งให้มีการแบ่งข้อมูลจากชั้นบนและ SDU มีขนาดมากกว่าขนาดของ TMD PDU เพื่อส่งข้อมูลภายในหนึ่ง TTI แล้ว TM RLC entity ในฝั่งส่งจะทำการแบ่ง RLC SDU เป็น TMD PDU อย่างพอดีโดยไม่มีการใส่ header เข้าไป หลังจากนั้น TMD PDU ทั้งหมดจะถูกส่งออกไปใน TTI เดียวกันโดยที่จะไม่ส่วนแบ่งอื่นจาก RLC SDU อื่นส่งใน TTI เดียวกัน แต่ถ้าไม่มีการติดตั้งการแบ่งข้อมูลจากชั้นบนแล้ว RLC SDU ที่รับจะถูกแทนที่เป็น TMD PDU และจะส่งออกไปใน TTI เดียวกันและทุก ๆ TMD PDU จะใน TTI เดียวกันจะต้องมีขนาดเท่ากัน

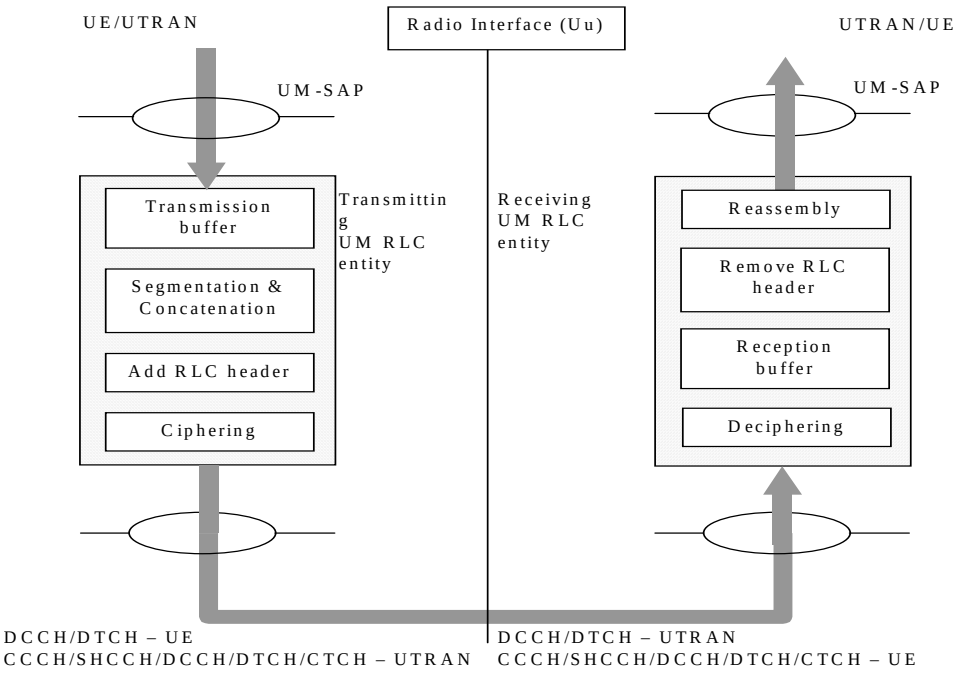
เมื่อมีการจัดการของ RLC SDU เสร็จเรียบร้อยแล้ว TMD PDU ทั้งหมดจะถูกส่งไปยังชั้นล่างผ่านทางช่องสัญญาณลอจิก โดยที่ช่องสัญญาณที่ใช้ขึ้นกับว่า RLC entity อยู่ที่ใด ซึ่งถ้าอยู่ใน user plane จะใช้ช่องสัญญาณ DTCH แต่ถ้าอยู่ใน control plane จะใช้ช่องสัญญาณ CCCH BCCH SHCCH หรือ PCCH ช่องใดช่องหนึ่ง

3.2.1.3 เอ็นทิที้ TM RLC ฝั่งรับ

TM RLC entity ในฝั่งที่ทำการรับ จะรับข้อมูล TMD PDU จากชั้นล่างผ่านทางช่องสัญญาณลอจิก ถ้ามีการติดตั้งของการแบ่งข้อมูลจากชั้นบน ทุก ๆ PDU ที่รับมาใน TTI หนึ่งจะถูก reassembly กลับไปเป็นหนึ่ง SDU แต่ถ้าไม่มีการติดตั้งการแบ่งข้อมูลในแทนแต่ละ PDU เป็น SDU ไปได้ หลังจากที่ SDU มาแล้วจะทำการส่งไปยังชั้นบนโดยผ่าน TM-SAP

3.2.2 Unacknowledged Mode (UM) RLC Entities

ใช้ในการส่งข้อมูลที่ไม่ต้องการรับประกันการส่งถึงไปยังฝั่งรับ หากข้อมูลที่ส่งเกิดข้อผิดพลาดก็จะถูกตัดทิ้งไป หรือถูกบันทึกข้อผิดพลาดในฝั่งรับ โครงสร้างของ PDU จะมี Header ของ Sequence Number (SN) ไว้ตรวจสอบข้อมูลที่หายไปได้ การบริการในโหมดนี้ได้แก่ Cell Broadcast Service และ Voice over IP (VoIP) ซึ่งในการส่งข้อมูลระหว่างสองเอ็นทิ้ UM RLC เป็นดังรูปที่ 3.3



รูปที่ 3.3 เอ็นทิ้ของ RLC เมื่อทำงานในโหมด Unacknowledged Mode (UM) [27]

3.2.2.1 บริการของโหมด TM

UM บริการสำหรับส่ง PDU ของชั้นที่สูงกว่าโดยไม่มีการรับประกันว่าข้อมูลที่จะถูกส่งถึงโหนดปลายทางหรือเอ็นทิ้ RLC ของอีกฝั่งหนึ่ง ถ้า RLC SDU มาจากชั้นบน ฝั่งส่งก็จะทำการตัดแบ่ง RLC SDU เป็น UMD PDU ในขนาดที่เหมาะสม ถ้ามันขนาดใหญ่กว่าพื้นที่ว่างของ UMD PDU ที่มีอยู่แล้ว UMD PDU สามารถ segment และ/หรือ concatenate RLC SDU และ padding bit เพื่อให้มั่นใจว่าขนาดของมันพอดี Length Indicator จะถูกใช้เพื่อกำหนดขอบเขตระหว่าง RLC SDU และ padding ที่ถูกรวมอยู่ใน UMD PDU โปรโตคอลในชั้นบนสามารถกำหนดให้ RLC มีการเข้ารหัส (Ciphering) ได้ หากเป็นเช่นนั้น การ ciphering จะ

เกิดขึ้นก่อนที่ UMD PDU ถูกส่งไปยังชั้นล่าง และ UMD PDU header จะไม่ถูก ciphering หลังจากนั้น UMD PDU ก็จะถูกส่งไปยังชั้นล่างผ่านช่องสัญญาณ logical เมื่อฝั่งรับได้รับ UMD PDU ก็จะทำการ deciphering แล้วเอา RLC header ออก จากนั้นทำการต่อให้เป็น RLC SDU เพื่อส่งต่อไปให้ชั้นบนของฝั่งรับนั้น

3.2.2.2 เอ็นทิ้ UM RLC ฝั่งส่ง

เอ็นทิ้ UM RLC ในฝั่งที่ทำการส่งจะรับ RLC SDU มาจากชั้นบนผ่านทาง UM-SAP ซึ่งเอ็นทิ้ UM RLC จะทำการแบ่งหรือต่อ RLC SDU ให้เป็นข้อมูล UMD PDU ที่เหมาะสม นอกจานี้ยังมีการ padding เพื่อให้ขนาดของ UMD PDU แน่นอนว่าถูกต้อง โดยมี Length indicator (LI) ใช้ในการระบุขอบเขตระหว่าง SDU ใน UMD PDU ซึ่ง Length indicator field สามารถทำการ padding ได้ด้วย

ถ้ามีการติดตั้ง Ciphering UMD PDU จะถูก ciphering ยกเว้น UMD PDU header ก่อนที่จะส่งไปยังชั้นล่าง ผ่านทางช่องสัญญาณลอจิก CCCH SHCCH DCCH CTCH DTCH ช่องสัญญาณใดช่องสัญญาณหนึ่ง

3.2.2.3 เอ็นทิ้ UM RLC ฝั่งรับ

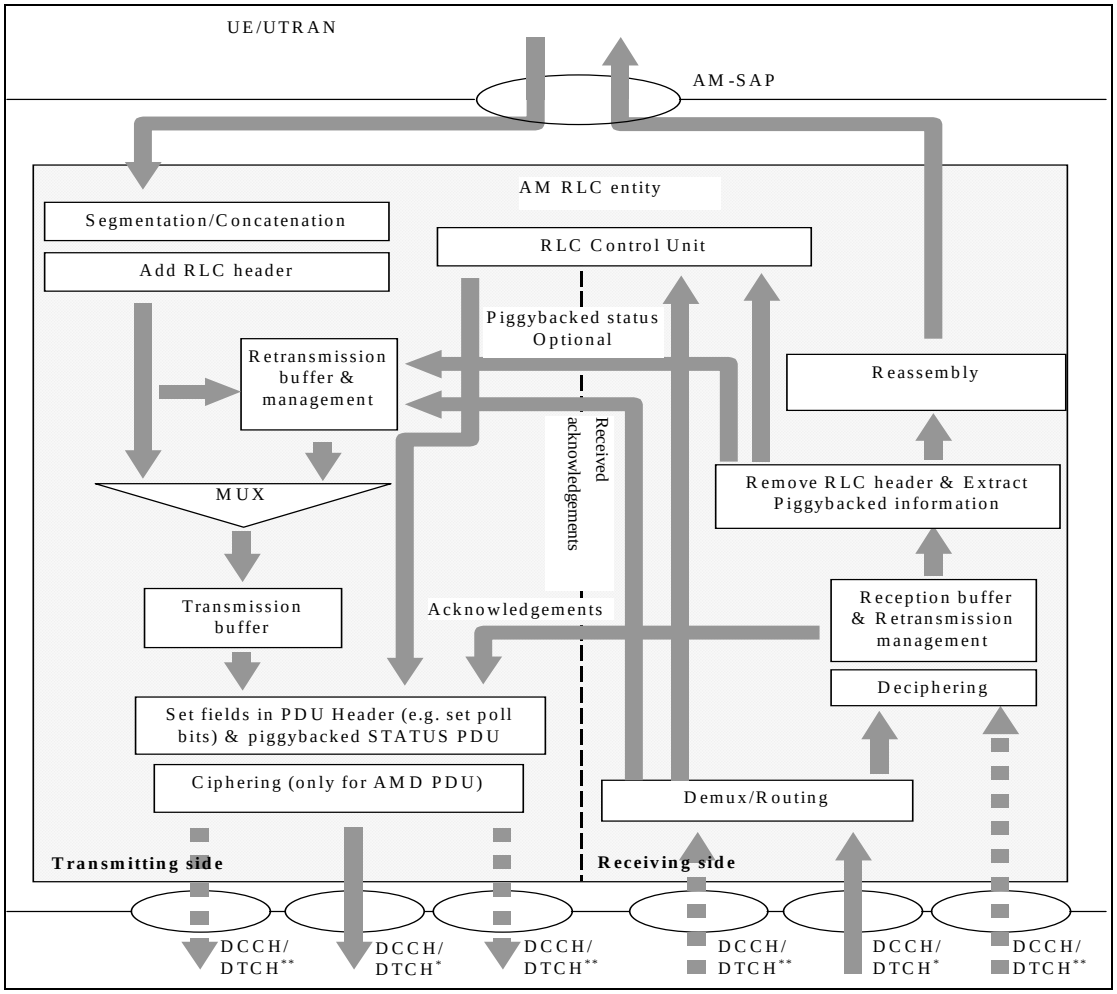
ใน UM RLC entity ในฝั่งที่ทำการรับจะรับ UMD PDU จากช่องสัญญาณลอจิกที่ได้ติดตั้งไว้ก่อนจากชั้นล่าง ถ้ามีการติดตั้ง ciphering ให้ทำการ deciphering PDU นั้นยกเว้นส่วนของ UMD PDU header หลังจากนั้นทำการย้าย header ออกไป และทำการต่อเป็น RLC SDU ในกรณีที่มีการติดตั้งการแบ่งหรือการต่อข้อมูลที่ได้กระทำในฝั่งส่งมาก่อนหน้า

3.2.3 Acknowledged Mode (AM) RLC entities

ใช้ในการบริการที่รับประกันการส่งข้อมูลถึงผู้รับแน่นอน โดยจะมีกระบวนการส่งใหม่ของข้อมูลเมื่อเกิดข้อผิดพลาด ซึ่งใช้กระบวนการ Automatic Repeat reQuest (ARQ) เพื่อให้ข้อมูลเกิดความถูกต้อง ชั้น RRC จะทำการควบคุมประสิทธิภาพ คุณภาพและความล่าช้าของข้อมูล ในการให้บริการของโหมดนี้จะเป็นการส่งข้อมูลแพ็กเกตที่ต้องการความแน่นอน เช่น อินเตอร์เน็ต การส่งเมล์ การดาวโหลดข้อมูล เป็นต้น

จากรูปที่ 3.4 เอ็นทิ้ AM RLC จะถูกติดตั้งใช้ช่องสัญญาณเดียว (เส้นทึบ) หรือ สองช่องสัญญาณ (เส้นประ) จากชั้นบน ใช้ช่องสัญญาณลอจิกเพียงช่องเดียวในกรณีที่ AMD และ Control PDU มีขนาดเท่ากันเท่านั้น และในกรณี downlink จะใช้ช่องสัญญาณใดก็ได้

แต่ uplink จะใช้ช่องสัญญาณที่หนึ่งสำหรับ AMD PDU (ส่งไปยังส่วน deciphering) และช่องสัญญาณที่สองกับ control PDU ซึ่งใช้ในการส่งข้อมูลระหว่างเอ็นทิที้ RLC ที่ทำงานในโหมด AM



รูปที่ 3.4 เอ็นทิที้ของ RLC เมื่อทำงานในโหมด Acknowledged Mode (AM) [27]

3.2.3.1 บริการของโหมด AM

AM บริการสำหรับส่ง PDU ของชั้นที่สูงกว่าและรับประกันว่าข้อมูลที่ส่ง ส่งถึงปลายทางแน่นอน ในโหมดนี้ และจากที่กล่าวไปแล้วว่าทั้งฝั่งรับและฝั่งส่งสามารถถูกปรับแต่งให้สามารถใช้ 1 หรือ 2 ช่องสัญญาณ logical เพื่อเชื่อมต่อไปยังโปรโตคอลชั้นล่าง เพราะว่า AM ถูกออกแบบมาเพื่อจัดเตรียมการส่งที่น่าเชื่อถือของ packet data และเป็นโหมดเริ่มต้นสำหรับ handling โปรแกรมการทำงานของ Interactive และ Background Class (QoS ของ UMTS) ใน

โหมดนี้เป็นการทำงานที่เต็มประสิทธิภาพมากที่สุดของ RLC มีการส่งซ้ำข้อมูล (retransmit) หากข้อมูลนั้นไปไม่ถึงปลายทาง ดังนั้นหากทำการศึกษาเพื่อประเมินสมรรถนะการทำงานของโปรโตคอล RLC ในโหมดนี้ได้ ก็จะสามารถประยุกต์ใช้กับ 2 โหมดที่เหลือได้เช่นกัน

3.2.3.2 เอ็นทิที้ AM RLC ฝั่งส่ง

AM RLC entity ในฝั่งที่ทำการส่งจะรับ SDU มาจากชั้นบนผ่านทาง AM-SAP และ SDUs จะถูกแบ่งเป็น PDUs ซึ่งขนาด AMD PDU จะเป็น semi-static (ติดตั้งผ่านพารามิเตอร์) จะมีการทำ padding และ LI (Length Indicator) นอกจากนี้ LI สามารถระบุว่า Padding และ piggybacked STATUS PDU รวมอยู่ใน AMD PDU เมื่อแบ่งเสร็จแล้วจะถูกเก็บในบัฟเฟอร์ส่งซ้ำ (retransmission buffer) และ MUX โดยที่อยู่ในบัฟเฟอร์จะถูกลบหรือนำกลับมาส่งซ้ำขึ้นกับ STATUS PDU หรือ Piggybacked STATUS PDU ใน status จะมีทั้ง positive และ negative acknowledged ของแต่ละ PDU ในฝั่งรับ เมื่อผ่านส่วนของ MUX แล้วและสามารถใส่ Piggybacked STATUS PDU ลงไปในส่วนของ Padding ได้ใน AMD PDU ซึ่งใน header ของ AMD PDU จะถูก set field โดยรับอินพุตจากส่วน RLC control unit เช่น polling bit

นอกจากนี้มัลติเพล็กซ์ระหว่าง Control PDU ที่มาจาก RLC control unit และ Reception buffer (ใน ciphering เฉพาะ Piggybacked STATUS PDU, DATA และ padding) และสุดท้ายส่ง PDU ออกทาง DTCH หรือ DCCH

3.2.3.3 เอ็นทิที้ AM RLC ฝั่งรับ

AM RLC entity ในฝั่งที่ทำการรับจะรับ PDU จากชั้นล่างผ่านส่วนของการ route ซึ่งหากไปยังส่วน deciphering แล้วส่งไปยัง reception buffer จนกว่า RLC SDU หนึ่งจะสมบูรณ์ซึ่งจะส่ง STATUS PDU ไปยังฝั่งส่งว่ารับสมบูรณ์หรือต้องการให้มีการส่งซ้ำของ PDU ที่หายไป แต่ถ้าพบ Piggybacked status PDU ใน AMD PDU จะส่ง PDU นี้ไปยัง retransmission buffer& management เพื่อลบ PDU เมื่อเป็น positive acknowledged และส่ง ส่ง PDU ใหม่ต่อไป

แต่ถ้าได้ SDU เรียบร้อยแล้วจากการ reassembly แล้วส่งผ่าน am-SAP เพื่อส่งไปยัง upper layer RESET และ RESET ACK PDU จะถูกส่งไปยัง RLC control unit เพื่อประมวลผลโดยต้องการให้มีการตอบสนองของเอ็นทิที้ AM ซึ่งจะส่ง control PDU โดย RLC control unit ไปยังฝั่งส่ง ซึ่งเมื่อรับ STATUS PDU เข้ามาจะถูกส่ง PDU นี้ไปยัง retransmission buffer& management เพื่อลบ PDU เมื่อเป็น positive acknowledged และส่ง PDU ใหม่ต่อไป

3.3 ชนิดของ PDU

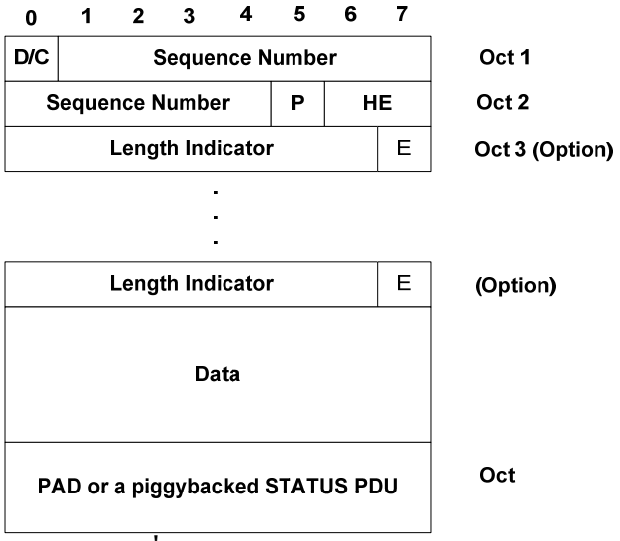
เนื่องจากการทดสอบเลือกใช้การทำงานในโหมด AM ดังนั้น ในหัวข้อนี้ขอกล่าวถึงรายละเอียดชนิดของ PDU ที่ใช้ในโหมดนี้เท่านั้น โดยในโหมด AM มี PDU 5 ชนิดที่ใช้งาน ได้แก่ AMD PDU, STATUS PDU, Piggybacked STATUS PDU, RESET และ RESET ACK PDU ใน AMD PDU จะบรรจุ user data (RLC SDUs) ส่วนใน STATUS PDU และ piggybacked STATUS PDU บรรจุรายละเอียดข้อมูลในการควบคุมโปรโตคอลที่แลกเปลี่ยนระหว่าง RLC ด้วยกัน เช่น การ retransmission การลบหรือทำลายทิ้ง หรือข้อมูลการเลื่อน window RESET PDU ใช้เพื่อสั่งให้โปรโตคอลทั้งหมดมี state, ค่าตัวแปร และตัวนับเวลากลับไปยังสถานะเริ่มต้น และ RESET ACK PDU เป็นการตอบกลับของ RESET PDU ซึ่งรายละเอียดโครงสร้างของแต่ละชนิดเป็นดังต่อไปนี้

3.3.1 AMD PDU

AMD PDU ถูกใช้เพื่อส่ง user data รายงานสถานะการร้องขอ และรายละเอียดการควบคุมโปรโตคอลหากมีการรวม piggyback STATUS PDU เข้าไว้ใน AMD PDU ส่วนสำคัญที่สุดของ AMD PDU คือส่วนของข้อมูลหรือ user data ในการดำเนินการส่งซ้ำ AMD PDU ที่จำเพาะ และการต่อกลับ AMD PDU เป็น RLC SDU แต่ละ AMD PDU จำเป็นต้องมีการแยกแยะระบุตัวเองอย่างชัดเจน ดังนั้นจะมีการกำหนด sequence number ไว้ในส่วนของ header ให้แต่ละ AMD PDU โดยเฉพาะ และเพื่อร้องขอรายงานสถานะ เอ็นทิที้ RLC ฝั่งส่งก็จะมีการกำหนด polling bit ไว้ใน header แต่ละ AMD PDU ด้วยเช่นกัน

จากรูปที่ 3.5 แสดงรูปแบบของ AMD PDUs ความยาวของส่วนที่เป็นข้อมูลต้องเป็นจำนวนเท่าของ 8 bit ใน header ของ AMD PDU 2 ไบท์แรกประกอบด้วย 1 บิต D/C, 12 บิต sequence number, 1 polling บิตและ 2 บิต Header Extension Type (HE) ใน D/C จะเป็นตัวบอกชนิดของ AM PDU จะเป็น ‘0’ ถ้าเป็นชนิด control และเป็น ‘1’ ถ้าเป็นชนิด data ส่วนใน SN เป็นจำนวนต่อไปเรื่อย ๆของแต่ละ AMD PDU ใช้สำหรับการส่งซ้ำและการต่อกลับ บิต polling ใช้สำหรับร้องขอรายงานสถานะจากเอ็นทิที้ RLC ระดับเดียวกัน (peer RLC entity) ถูกกำหนดเป็น ‘1’ ถ้าต้องการร้องขอ กรณีอื่น ๆกำหนดเป็น ‘0’ สำหรับ HE จะเป็นตัวบอกว่า 8 บิตถัดไปจะเป็น data หรือเป็น Length Indicator และบิต E ซึ่ง Length Indicator และ E เป็นส่วนเสริมใช้เฉพาะสำหรับแยก RLC SDU, padding และ piggybacked STATUS PDU ในบิต E เป็นบิตที่ต่อเติมออกมาเพื่อใช้ในการระบุว่า 8 bit ถัดไปเป็น Length Indicator และบิต E อันอื่นอีก ส่วนถัดจากนี้เป็น data ที่บรรจุ RLC SDU หนึ่งอันหรือมากกว่านั้น และขนาดของ data นี้สามารถเปลี่ยนแปลงได้แต่ต้องเป็นจำนวนเท่าของ 8 bit ส่วนสุดท้าย padding หรือ piggybacked

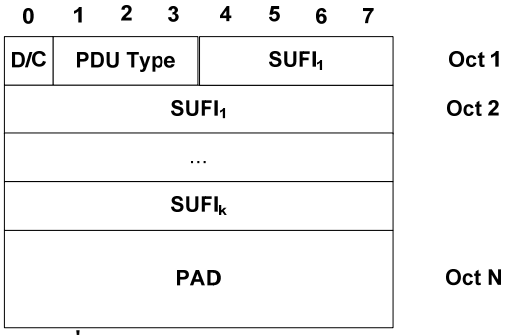
STATUS PDU ซึ่ง padding ควรใช้สำหรับทำให้ขนาดของ AMD PDU มีความถูกต้องและ RLC ที่เป็นตัวรับจะไม่สนใจส่วนนี้ piggybacked STATUS PDU สามารถแทนที่ padding ได้ซึ่งจะนำพารายละเอียดการควบคุมโปรโตคอลสำหรับทำ STATUS PDU และมีขนาดที่เปลี่ยนแปลงได้เพื่อให้เหมาะสมกับขนาดที่มีอยู่ใน AMD PDU



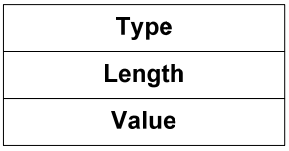
รูปที่ 3.5 รูปแบบของ AMD PDUs

3.3.2 STATUS PDUs

STATUS PDU เป็นตัวนำพารายละเอียดข้อมูลการควบคุมโปรโตคอลใช้แลกเปลี่ยนกันระหว่าง RLC 2 เอ็นทิตี้ ในการเชื่อมต่อ RLC เมื่อจำเป็นต้องมีการส่งซ้ำ RLC PDU ที่หายไป STATUS PDU สามารถที่จะส่งรายการของ sequence number ของ RLC PDU ที่หายไปในช่วงที่การส่งมีข้อผิดพลาด เมื่อฝั่งรับได้รับ AMD PDU และทำการต่อกลับเรียบร้อยแล้ว RLC ฝั่งรับก็จะมีการแจ้งไปบอกฝั่งส่งให้ทำการเลื่อน transmission window ต่อไป และให้ลบ RLC PDU ก่อนหน้าที่ทำการส่งสำเร็จไปแล้วที่อยู่ในบัฟเฟอร์ส่งซ้ำทิ้งไป ดังนั้น STATUS PDU ก็จะมีการส่งรายละเอียดเพื่อบอกให้รู้ถึง RLC PDU ที่ส่งสำเร็จด้วยเช่นกัน ในกรณีที่จำนวนของ RLC PDU ที่ต้องส่งซ้ำมีค่าเท่ากับค่าสูงสุดที่อนุญาตให้มีการส่งซ้ำเกิดขึ้น การกู้คืน RLC PDU ก็จะหยุดทำงาน และ RLC ฝั่งส่งก็จะมีการแจ้งไปบอกถึงความล้มเหลวที่เกิดขึ้น ดังนั้น STATUS PDU ก็สามารถใช้ในการส่งรายเอียดความล้มเหลวเพื่อไปจัดการเกี่ยวกับ reception window ที่ RLC ฝั่งรับได้



รูปที่ 3.6 รูปแบบของ STATUS PDUs



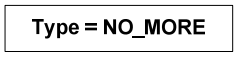
รูปที่ 3.7 โครงสร้างของ Super-Field

จากรูปที่ 3.6 แสดงรูปแบบโครงสร้างของ STATUS PDU ใน D/C มีค่าเป็น '0' จะหมายถึงข้อมูลที่ส่งมานี้เป็นชนิด control สำหรับ PDU Type มี 3 บิตเพื่อบอกชนิดของ Control PDU เช่น STATUS, RESET หรือ RESET ACK ใน SUFI หรือ Super-Field มีขนาดที่เปลี่ยนแปลงได้ตามชนิดของข้อมูลที่บรรจุภายใน รูปที่ 3.7แสดงโครงสร้างของ super-field ขนาดของ sub-field สามารถเป็น 0 ได้ยกเว้น "type" ที่จะต้องมีความยาว 4 บิตซึ่งค่าต่าง ๆที่สามารถเป็นไปได้สำหรับ type แสดงไว้ดังรูปที่ 3.8 เป็นตัวบอกจุดสุดท้ายของส่วนที่เป็น data ของ STATUS PDU และจำเป็นต้องมีตัวนี้อยู่ใน STATUS PDU เป็น SUFI สุดท้าย ACK SUFI (รูปที่ 3.9) เป็นตัวบอกจุดสุดท้ายของส่วนที่เป็น data ของ STATUS PDU และประกอบด้วยชนิดตัวระบุ ACK และ LSN (last SN) ที่จะบ่งบอก sequence number ของ AMD PDU ก่อนหน้าตัวที่ถูกส่งเสร็จถูกต้องเรียบร้อยแล้ว ซึ่งถูกใช้เพื่อแจ้งเอ็นทิที้ RLC ให้เลื่อน transmission window และลบ RLC PDU ก่อนหน้าที่ทำการส่งสำเร็จไปแล้วที่อยู่ในบัฟเฟอร์ส่งซ้ำทิ้งไป LIST SUFI (รูปที่ 3.10) และ BITMAP SUFI (รูปที่ 3.11) ใช้เพื่อร้องขอการส่งซ้ำของ AMD PDU ที่ส่งผิดพลาด MRW SUFI (รูปที่ 3.12) ใช้เพื่อร้องขอให้ RLC entity เลื่อน window รับของมันและยังระบุถึงชุดของ RLC SDU ที่ถูก discard ด้วย MRW_ACK SUFI (รูปที่ 3.13) เป็นการตอบกลับ MRW SUFI รายละเอียดเกี่ยวกับวิธีกำหนดค่าให้กับ sub-field แสดงไว้ใน [27]

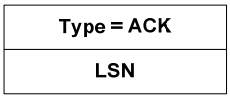
ตารางที่ 3.2 ค่าบิตสำหรับแต่ละชนิดของ Super-Field

บิต	คำอธิบาย
0000	No More Data (NO_MORE)
0001	Window Size (WINDOW)

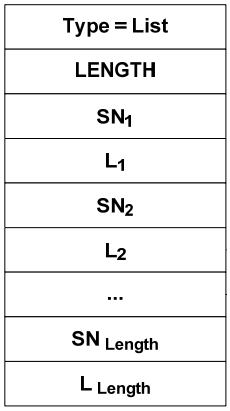
0010	Acknowledgement (ACK)
0011	List (LIST)
0100	Bitmap (BITMAP)
0101	Relative list (Rlist)
0110	Move Receiving Window (MRW)
0111	Move Receiving Window Acknowledgement (MRW_ACK)



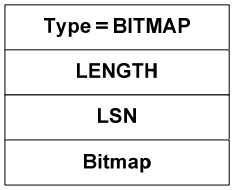
รูปที่ 3.8 super-field : NO_MORE



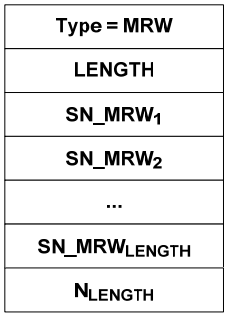
รูปที่ 3.9 super-field : ACK



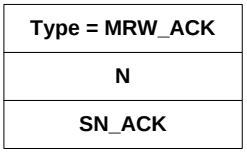
รูปที่ 3.10 super-field : LIST



รูปที่ 3.11 super-field : BITMAP



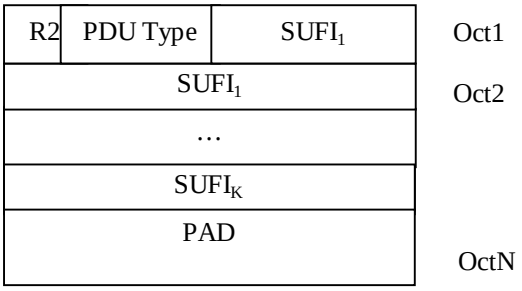
รูปที่ 3.12 super-field :MRW



รูปที่ 3.13 super-field : MRW_ACK

3.3.3 Piggybacked STATUS PDU

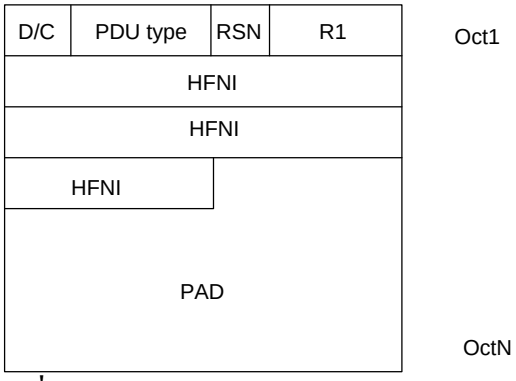
Piggybacked STATUS PDU จะเหมือนกับ STATUS PDU ยกเว้น D/C filed จะถูกแทนด้วย Reserved bit (R2) ซึ่ง PDU นี้สามารถอยู่ใน AMD PDU ถ้าข้อมูลไม่ได้ใส่จนเต็ม AMD PDU PDU type จะถูกเซ็ตเป็น “000” และถ้าเป็นค่าอื่นจะต้องตัดทิ้งออกไปดังรูปที่ 3.14



รูปที่ 3.14 Piggybacked STATUS PDU [27]

3.3.4 RESET, RESET ACK PDU

RESET, RESET ACK PDU จะมีหนึ่งบิต SN field (RSN) เพื่อรู้ว่าเป็นการ retransmission ของ RESET PDU หรือ RESET ACK PDU ก่อนหน้า มีโครงสร้างตามรูปที่ 3.15



รูปที่ 3.15 RESET, RESET ACK PDU [27]

ขนาดของ RESET หรือ RESET ACK PDU จะมีการเปลี่ยนแปลงและขอบเขตบนสุดโดยขนาดของ PDU มากที่สุด ซึ่งจะถูกใช้โดยช่องสัญญาณลอจิกที่ control PDU จะถูกส่งไป Padding จะจัดการให้ครบขนาดของหนึ่ง PDU size ขนาดของ RESET หรือ RESET ACK PDU จะต้องเป็นจำนวนชุดของขนาด 8 บิต

3.4 โปรโตคอลพารามิเตอร์และบัฟเฟอร์

ใน RLC มีโปรโตคอลพารามิเตอร์ 9 ตัว ใช้ในฝั่งส่ง 7 ตัวได้แก่ MAXDAT, POLL_PDU, POLL_SDU, POLL_WINDOW, CONFIGURED_TX_WINDOW_SIZE, MAXMRW และ MAXRST 1 ตัวใช้ในฝั่งรับได้แก่ CONFIGURED_RX_WINDOW_SIZE และตัวสุดท้ายใช้ในทั้ง 2 ฝั่ง ได้แก่ TTI ค่าของพารามิเตอร์อาจถูกกำหนดโดยชั้นบน ณ เวลาที่สร้าง session และจะไม่เปลี่ยนระหว่าง session นั้น ๆ

สำหรับบัฟเฟอร์ใน RLC จะประกอบด้วย 3 ชนิด โดย 2 ชนิด คือบัฟเฟอร์ส่ง (transmission buffer) และบัฟเฟอร์ส่งซ้ำ (retransmission buffer) ถูกใช้เพื่อส่งและส่งซ้ำ RLC PDU ในฝั่งส่ง retransmission buffer เก็บ RLC PDU ทั้งหมดที่ถูกส่งไปและข้อมูลจะถูกเปลี่ยนแปลงใหม่เสมอทุกครั้งที่ได้รับ STATUS PDU ที่รวม ACK SUFI มา เมื่อมีการร้องขอให้ retransmit RLC PDU อันหนึ่งอันใด RLC PDU อันนั้นก็จะถูกคัดลอกจาก retransmission buffer ไปไว้ใน transmission buffer จะเห็นว่า transmission buffer ไม่ได้เก็บเฉพาะ RLC PDU ที่ส่งครั้งแรกเท่านั้น แต่มันยังเก็บ RLC PDU ที่ถูกร้องขอให้ retransmit อีกด้วย อย่างไรก็ตาม transmission buffer เป็นบัฟเฟอร์ที่มีคิวเรียงตามลำดับความสำคัญ โดย RLC PDU ที่ถูกร้องขอให้ retransmit จะมีลำดับความสำคัญสูงกว่า RLC PDU ที่จะถูกส่งออกเป็นครั้งแรก สำหรับบัฟเฟอร์อีกชนิดหนึ่งใน RLC คือ reception buffer ซึ่งเก็บ RLC PDU ที่ได้รับมาถูกต้องเรียบร้อย จะอยู่ในฝั่งรับ บัฟเฟอร์ทั้ง 3 นี้จะว่างในช่วงที่เริ่มการติดต่อ สำหรับรายละเอียดของแต่ละพารามิเตอร์มีดังนี้

1) *TTI* เป็นช่วงเวลาระหว่างการส่งหรือการรับ 2 ลำดับที่ต่อเนื่องกันของ AM PDU ค่าของ TTI ควรเป็นจำนวนเท่าของ 10 มิลลิวินาที และตัวจับเวลาทั้งหมดใน RLC จะอยู่ในส่วนของ TTI
2) *MAXDAT เป็นจำนวนการส่งสูงสุดของ AMD PDU เป็นค่าขอบเขตบนสำหรับตัวแปร state VT(DAT) เมื่อ VT(DAT) เท่ากับค่า MAXDAT ขั้นตอนการทำ SDU discard จะเริ่มต้นขึ้น*
3) *POLL_PDU เป็นค่าขอบเขตบนสำหรับตัวแปร state VT(PDU) จะระบุว่าผู้ส่งควรสำรวจผู้รับบ่อยแค่ไหนถ้า POLL_EVERY_PDU_PDU ถูกตั้งค่า*

ไว้ เมื่อ VT(PDU) เท่ากับค่าของ POLL_PDU บิต poll ใน header ของ AMD PDU จะถูกกำหนดให้เป็น '1'

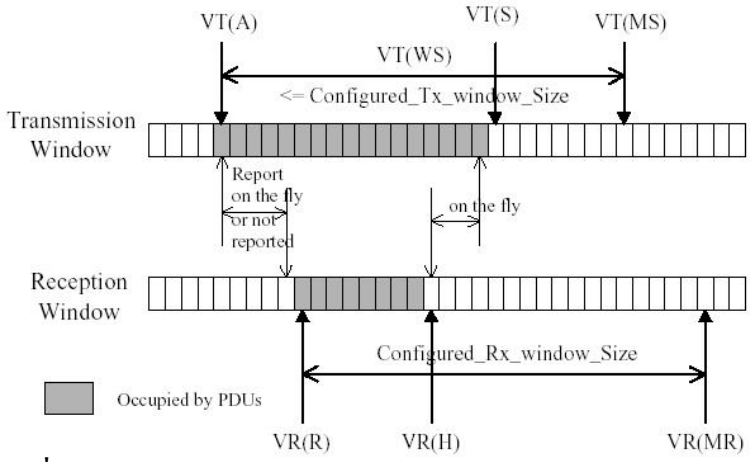
4) *POLL_SDU เป็นค่าขอบเขตบนสำหรับตัวแปร state VT(SDU) จะระบุว่าผู้ส่งควรสำรวจผู้รับบ่อยแค่ไหนถ้า POLL_EVERY_SDU_SDU ถูกตั้งค่าไว้ เมื่อ VT(SDU) เท่ากับค่าของ POLL_SDU บิต poll ใน header ของ AMD PDU จะถูกกำหนดให้เป็น '1'*
5) *POLL_WINDOW จะระบุว่าผู้ส่งควรสำรวจผู้รับบ่อยแค่ไหนถ้า WINDOW_BASED_POLLING ถูกตั้งค่าไว้ การสำรวจค่าจะถูกใช้งานสำหรับแต่ละ AMD PDU เมื่อตัวแปร state J มากกว่าหรือเท่ากับ POLL_WINDOW*
6) *MAXRST เป็นจำนวนครั้งการส่งสูงสุดของ RESET PDU*
7) *CONFIGURED_TX_WINDOW_SIZE เป็นทั้งจำนวนสูงสุดของขนาด transmission window และเป็นค่าสำหรับตัวแปร state VT(WS)*
8) *CONFIGURED_RX_WINDOW_SIZE เป็นขนาดของ reception window*
9) *MAXMRW เป็นจำนวนครั้งสูงสุดในการส่ง MRW SUFI มันเป็นค่าขอบเขตบนสำหรับตัวแปรstate VT(MRW) เมื่อ VT(MRW) เท่ากับค่า MAXMRW ทำให้เกิดขบวนการ RLC RESET*

3.5 ตัวแปร State และการทำงานในฝั่งส่ง

ในฝั่งส่ง RLC คาดหวังแพ็กเกตจากชั้นบนที่เรียกว่า RLC SDU ที่ซึ่งอาจจะถูก segment และ/หรือ concatenate เป็น AMD PDU ที่มีขนาดแน่นอน ใน RLC การ segment เกิดขึ้นถ้าขนาดของ RLC SDU ใหญ่กว่าเนื้อที่ว่างใน AMD PDU แต่ละ AMD PDU ที่จะถูกส่งออกเป็นครั้งแรกจะถูกคัดลอกไปไว้ในบัฟเฟอร์ส่งซ้ำหนึ่งอัน และคัดลอกอีกหนึ่งอันไปไว้ในบัฟเฟอร์ส่ง สำหรับ RLC PDU ที่ถูกร้องขอให้ส่งซ้ำ ตัวของมันที่ถูกคัดลอกไปเก็บไว้ในบัฟเฟอร์ส่งซ้ำก็จะถูกคัดลอกนำมาไว้ในบัฟเฟอร์ส่งพร้อมกับมีลำดับความสำคัญสูงกว่า PDU ที่จะถูกส่งเป็นครั้งแรก ฟิลด์ต่าง ๆ ของ AMD PDU ในบัฟเฟอร์ส่งก็จะถูกกำหนดค่า ตัวอย่างเช่น บิต poll ใน header ของ AMD PDU อาจจะถูกกำหนดค่าเป็น '1' หรือไม่ขึ้นกับค่าของพารามิเตอร์ควบคุม RLC ส่วนของ padding เป็นไปได้ว่าอาจจะถูกแทนที่ด้วย piggybacked status information และการ ciphering จะถูกเรียกใช้หากชั้นบนกำหนดมา เป็นต้น ขั้นตอนสุดท้ายฝั่งส่งทำการส่ง AMD PDU ที่อยู่ในบัฟเฟอร์ส่งไปยังชั้นล่างผ่านช่องสัญญาณลอจิก ขบวนการในการจัดการ PDU แสดงไว้ในฝั่งส่งตามรูปที่ 3.4

3.5.1 ตัวแปร State ในฝั่งส่ง

มีตัวแปรทั้งหมด 8 state ที่ใช้เพื่อเก็บรายละเอียด AM PDU ที่ส่งและส่งซ้ำ ดังแสดงไว้ในรูปที่ 3.16 เพื่อรักษาขนาดของ transmission window (*VT(WS)*) จึงมีการกำหนดให้ *VT(A)* แทนค่าขอบเขตล่าง และ *VT(MS)* แทนค่าขอบเขตบนของ transmission window และเพื่อเก็บรายละเอียดของ PDU ถัดไปที่จะถูกส่งครั้งแรก และบอกถึง RLC PDU ใช้งาน transmission window อยู่เท่าไรจะใช้ตัวแปร *VT(S)* เป็นตัวกำหนด ตัวแปร *VT(DAT)* ใช้ในการนับจำนวนครั้งการส่งของ PDU ตัวนั้นๆ ซึ่งกลไก SDU discard จะนำไปใช้งาน ตัวแปร *VT(MRW)* นับจำนวนครั้งที่ส่ง MRW SUFI ตัวนั้นๆ ที่ถูกเรียกใช้จากขบวนการ RESET การเชื่อมต่อ ตัวแปร *VT(PDU)* ถูกใช้โดยกลไก polling เพื่อนับจำนวน AMD PDU ที่ถูกส่งหลังจากมีการกำหนดบิต polling ตัวล่าสุดขึ้นใช้งาน และ *VT(SDU)* ถูกใช้โดยกลไก polling เพื่อนับจำนวน RLC SDU ที่ถูกส่งหลังจากมีการกำหนดบิต polling ตัวล่าสุดขึ้นใช้งาน



รูปที่ 3.16 ตัวแปรของ transmission และ retransmission window

1. **VT(WS)** เป็นขนาดของ transmission window ที่ถูกกำหนดไว้ตั้งแต่เริ่มต้น session และสามารถเปลี่ยนแปลงได้ระหว่าง session โดยการรับ STATUS PDU ที่มี WINDOW SUFI
2. **VT(A)** เป็น sequence number ตาม sequence number ของลำดับสุดท้ายของ AMD PDU ที่ได้รับการตอบกลับมา เป็นค่าขอบเขตล่างของ transmission window
3. **VT(MS)** เก็บ sequence number ของ AMD PDU แรก ที่สามารถถูกละทิ้งโดยเอ็นทิที้ RLC เพราะว่าเป็นค่าจำกัดของขนาด transmission window เป็นค่าขอบเขตบนของ transmission window และเท่ากับ *VT(A) + VT(WS)* เพราะฉะนั้นค่าควรจะเปลี่ยนแปลงเมื่อ *VT(A)* หรือ *VT(WS)* หรือทั้งสองค่าเปลี่ยนแปลง
4. **VT(S)** เก็บ sequence number ของ AMD PDU ถัดไปที่จะถูกส่งครั้งแรก

5. **VT(DAT)** นับจำนวนครั้งที่ซึ่ง AMD PDU มีกำหนดการเพื่อถูกทำการส่ง ค่าเริ่มต้นของ *VT(DAT)* คือ 0
6. **VT(PDU)** ใช้เมื่อ *POLL_EVERY_POLL_PDU_PDU* ถูกกำหนดให้ใช้งาน เพิ่มค่าครั้งละ 1 หลังจากการส่งแต่ละ AMD PDU ทั้งที่เป็นการส่งครั้งแรกและที่เป็นการส่งซ้ำ และเมื่อค่าของตัวแปรนี้มีค่าเท่ากับค่าของ *POLL_PDU* แล้ว บิต poll ตัวใหม่จะถูกกำหนดค่าเป็น 1 เพื่อร้องขอ STATUS report ค่าเริ่มต้นของ *VT(PDU)* คือ 0
7. **VT(SDU)** ใช้เมื่อ *POLL_EVERY_POLL_SDU_SDU* ถูกกำหนดให้ใช้งาน เพิ่มค่าครั้งละ 1 ทุกครั้งที่ซึ่ง AMD PDU ทั้งหมดที่แบ่งมาจาก RLC SDU นั้นถูกส่งอย่างน้อย 1 ครั้ง และเมื่อมันมีค่าเท่ากับค่าของ *POLL_SDU* แล้วบิต poll ตัวใหม่จะถูกกำหนดค่าเป็น 1 เพื่อร้องขอ STATUS report ค่าเริ่มต้นของ *VT(SDU)* คือ 0
8. **VT(MRW)** นับจำนวนครั้งที่ซึ่ง MRW SUFI ถูกกำหนดเพื่อทำการส่งซ้ำ

3.5.2 กลไกการ Poll (Polling Mechanism)

ในกรณีที่ AMD PDU ผิดพลาดหรือสูญหาย การส่งซ้ำของฝ่ายส่งจะเกิดขึ้นโดยขึ้นอยู่กับการรับ STATUS report ที่ส่งมาจากฝ่ายรับ การส่ง STATUS report สามารถทำได้ทั้งจากฝ่ายส่งและฝ่ายรับ แต่สำหรับกลไกการ poll ฝ่ายส่งจะส่ง polling ไปร้องขอ และฝ่ายรับจะตอบกลับโดยส่ง STATUS report กลับมายังฝั่งส่ง

กลไกการ poll ทำงานโดยกำหนดค่าบิต polling ใน header ของ AMD PDU กี่ตัวก็ได้ ซึ่งมี polling trigger ทั้งหมด 8 ชนิดด้วยกันที่เก็บในฝั่งส่งและใช้เพื่อกำหนดค่าบิต polling เมื่อกลไกการ poll ถูกเริ่ม โดยชั้นบนจะเป็นตัวตัดสินใจว่าจะให้ใช้ trigger ตัวใด

1. **Poll Timer** หลังจากที่ trigger ตัวนี้หมดอายุ บิต poll ใน header ของ AMD PDU ตัวที่กำลังจะถูกส่งจะถูกกำหนดค่าเป็น 1 เพื่อทำการร้องขอสถานะการรับ PDU ของฝั่งรับ โดย trigger ตัวจับเวลาจับนี้จะหยุดการจับเวลาหลังจากได้รับ status PDUs ตอบกลับจากฝั่งรับ แต่ถ้าตัวจับเวลาหมดอายุก่อนที่จะได้รับ status PDUs ใด ๆจากฝั่งรับ ฝั่งส่งก็จะทำการกำหนดบิต poll ใน AMD PDU แล้วเริ่มการจับเวลาของ trigger ตัวนี้อีกครั้ง ซึ่ง trigger ตัวนี้จะถูกนำไปใช้ในการทดลองที่จะกล่าวถึงรายละเอียดในบทที่ 5
2. **Poll Prohibit Timer** เริ่มต้นเมื่อ polling ที่อยู่ใน AMD PDU ถูกส่งไปยังชั้นล่างของฝั่งส่ง จากเวลาที่ polling ถูก trig จนกระทั่งถึงเวลาที่ซึ่ง Poll Prohibit Timer หมดอายุ polling จะถูกขัดขวางห้ามไว้ ถ้า polling หนึ่งถูก trig ในขณะที่ polling กำลังถูกห้าม การส่ง polling ครั้งใหม่จะถูกหน่วงไว้จนกระทั่งตัวจับเวลาหมดอายุ ถ้ามีมากกว่า 1 polling ที่ถูก trig ในขณะที่ polling ถูกห้าม เมื่อ Poll Prohibit Timer

หมดอายุ จะมีเพียง polling ตัวเดียวเท่านั้นที่จะถูกส่งได้

3. **Poll Periodic Timer** เริ่มต้นเมื่อ session ถูกสร้าง เมื่อตัวจับเวลาหมดอายุ polling จะถูก trig ถ้ามี AMD PDU ที่สามารถส่งหรือส่งซ้ำอยู่ และตัวจับเวลาก็จะกลับไปที่ค่าเริ่มต้นใหม่
4. **Last PDU in buffer** เป็น transmission-buffer-based trigger ทุกครั้งที่ AMD PDU สุดท้ายที่อยู่ในบัฟเฟอร์ส่งที่จะส่งครั้งแรกกำลังจะถูกส่งออกไป บิต poll ใน header ของ PDU นั้นจะถูกกำหนดค่าเป็น 1
5. **Last PDU in retransmission buffer** เป็น retransmission-buffer-based trigger โดยบิต poll จะถูกกำหนดค่าเป็น 1 ถ้า AMD PDU ที่ถูกส่งซ้ำเป็น AMD PDU ตัวสุดท้ายที่เก็บในบัฟเฟอร์ส่งซ้ำ
6. **Poll_PDU_PDU** บิต poll จะถูกกำหนดค่าทุกครั้งที่ค่าของ VT(PDU) เท่ากับค่าของ POLL_PDU
7. **Poll_SDU_SDU** บิต poll จะถูกกำหนดค่าทุกครั้งที่ค่าของ VT(SDU) เท่ากับค่าของ POLL_SDU
8. **Window based polling** เป็น transmission-window-based trigger และจะถูกใช้เมื่อ WINDOW_BASED_POLLING ถูกปรับตั้งค่าให้ใช้งาน มีตัวแปร state J แสดงในรูปเปอร์เซ็นเก็บไว้ใน RLC ถูกคำนวนโดยการหารจำนวนของ PDU ที่ถูกส่งภายใน transmission window ปัจจุบันด้วยขนาดของ transmission window เมื่อค่า J มากกว่าเปอร์เซ็น threshold แล้วบิต poll ก็จะถูกกำหนดค่าเป็น 1

3.5.3 กลไกละทิ้ง SDU ของ RLC (RLC SDU Discard Mechanism)

RLC SDU discard mechanism ยอมให้ฝั่งส่ง discard AMD PDU ที่เกี่ยวข้องกับ SDU จากบัฟเฟอร์ส่งและบัฟเฟอร์ส่งซ้ำ กลไกนี้ถูกเริ่มต้นเมื่อการส่ง AMD PDU ไม่สำเร็จตามช่วงเวลาหรือจำนวนครั้งของการส่ง กลไกนี้สามารถหลีกเลี่ยงการล้นของบัฟเฟอร์ในชั้น RLC และลดเวลาหน่วงสูงสุดของการส่ง การทำงานของ RLC SDU discard มี 2 ประเภทที่สามารถถูกกำหนดใช้งานตามความต้องการของคุณภาพการบริการ (QoS) ซึ่งได้แก่ Timer based discard ด้วยสัญญาณที่แน่นอน และ RLC SDU discard หลังจากจำนวนครั้งการส่งเท่ากับ *MAXDAT*

3.5.3.1 Timer based discard ด้วยสัญญาณที่แน่นอน

การละทิ้งวิธีนี้เกิดขึ้นเมื่อ Timer_Discard หมดอายุ วิธีการละทิ้งนี้ไม่สนใจต่อการเปลี่ยนแปลงของอัตรการส่งข้อมูลของช่องสัญญาณและอัตราความผิดพลาด ควบคุมค่าเวลา

หน่วงสูงสุดของแต่ละ RLC SDU อัตราการสูญหายของ SDU เพิ่มขึ้นตาม RLC SDU ที่ถูกละทิ้ง การทำงานเริ่มจากแต่ละ RLC SDU ที่ได้รับจากชั้นบน ตัวนับเวลา Timer_Discard จะเริ่มนับ เมื่อ Timer_Discard ของ SDU หนึ่ง ๆหมดอายุ ฝั่งส่งก็จะละทิ้ง SDU นั้น และใช้สัญญาณที่แน่นอนแจ้งไปบอกยังฝั่งรับถึง SDU ที่ถูกละทิ้งไป

3.5.3.2 RLC SDU discard หลังจากจำนวน transmission เท่ากับ MAXDAT

อีกทางหนึ่งสำหรับการทำ SDU discard คือ SDU discard หลังจากจำนวนครั้งการส่งเท่ากับค่าตัวแปร *MAXDAT* ซึ่งวิธีนี้จะพยายามให้อัตราการสูญหายของ SDU คงที่ อย่างไรก็ตามสมรรถนะเวลาหน่วงของวิธีนี้จะเปลี่ยนแปลงและขึ้นอยู่กับสภาวะของช่องสัญญาณ ถ้า *VT(DAT)* ของ AMD PDU เท่ากับค่า *MAXDAT* ฝั่งส่งจะละทิ้ง RLC SDU ทั้งหมดที่อยู่ใน AMD PDU และใช้สัญญาณที่แน่นอนแจ้งไปบอกฝั่งรับโดยส่ง *MRW SUFI*

3.5.4 Retransmission ของ RLC

3.5.4.1 วิธีการทั่วไป

หลักการทั่วไปประกอบด้วย ฝั่งส่งทำการส่ง poll ไปยังฝั่งรับ และฝั่งรับส่ง status report กลับมาบอกฝั่งส่งเพื่อชี้ว่า RLC PDU ใดได้รับสำเร็จ และ RLC PDU ใดไม่สำเร็จ จาก status report ที่ได้รับมา ฝั่งส่งจะส่งซ้ำ PDU ที่ถูกตอบกลับว่าเกิดความผิดพลาดหรือสูญหายไป มีหลาย ๆ วิธีการที่ถูกเพิ่มเข้าไปใน RLC เพื่อลดเวลาหน่วงของการส่งซ้ำ ป้องกันการส่งซ้ำที่ไม่จำเป็น ลด overhead ของการตอบกลับ และป้องกันการติดตายของโปรโตคอล

3.5.4.2 ชนิดของการส่งซ้ำ

ใน UMTS โปรโตคอล RLC ปัจจุบันใช้ type I hybrid ARQ เป็นวิธีการกู้คืนสำหรับ AM Mode โดยวิธีของ hybrid ARQ มี 3 ชนิดด้วยกัน คือ stop-and-wait (SAW), go-back-N (GBN) และ selective repeat (SR) RLC สามารถใช้วิธีการใดวิธีการหนึ่งเพื่อเป็นรูปแบบในการส่งซ้ำ โดยปรับแต่งค่า transmission window และรูปแบบของการตอบกลับที่เหมาะสม [28]

3.6 ตัวแปร State และการทำงานในฝั่งรับ

ในฝั่งรับคาดหวังที่จะได้รับ AMD PDU และ control PDU ผ่านช่องสัญญาณ

ลอจิกจากชั้นล่าง และจะมีการทำ deciphering หากชั้นบนกำหนดให้ทำ และส่งไปไว้ยังบัฟเฟอร์รับ (reception buffer) โดย AMD PDU จะอยู่ในบัฟเฟอร์รับจนกระทั่ง RLC SDU ถูกส่งไปให้ชั้นบนเสร็จเรียบร้อยแล้ว หลังจากนั้นฝั่งรับจะตอบกลับการรับที่สำเร็จหรือร้องขอการส่งซ้ำของ AMD PDU ที่ผิดพลาดโดยการส่ง STATUS PDU อันนึงหรือมากกว่านั้นไปยัง RLC ฝั่งส่งที่ทำการส่ง PDU นั้นมาให้ ถ้า piggybacked STATUS PDU ถูกพบใน AMD PDU ที่ได้รับก็จะถูกส่งไปยังบัฟเฟอร์ส่งที่ฝั่งส่งเพื่อคัดลอก PDU ที่จำเป็นต้องถูกส่งซ้ำ และนำตัวคัดลอกนั้นไปไว้ในบัฟเฟอร์ส่ง หลังจากนั้น AMD PDU ที่เกี่ยวข้องกับ RLC SDU ที่สมบูรณ์อาจถูกรวมกลับ (reassembly) และส่งไปยังชั้นบน จากนั้น STATUS PDU ถูกส่งไปยังบัฟเฟอร์ส่งซ้ำเพื่อคืนเนื้อที่ของ AMD PDU ที่ส่งสำเร็จเรียบร้อยแล้ว และค้นหา AMD PDU ที่จำเป็นต้องถูกส่งซ้ำ สำหรับ RESET และ RESET ACK PDU ถูกส่งไปยัง RLC control unit เพื่อดำเนินการ ซึ่งการทำงานของฝั่งรับแสดงไว้ในส่วนที่เป็นฝั่งรับในรูปที่ 3.4

3.6.1 ตัวแปร State ในฝั่งรับ

มีตัวแปร state 4 ตัวใช้สำหรับเก็บรายละเอียดของ AMD PDU ที่ได้รับในฝั่งรับ ได้แก่ VR(R), VR(H), VR(MR) และ VR(EP) แสดงไว้ดังรูปที่ 3.16 โดยที่ค่าของแต่ละตัวแปรเป็นดังนี้

1. **VR(R)** เก็บ sequence number ลำดับสุดท้ายของ AMD PDU ที่ได้รับ เป็นค่าขอบเขตล่างของ reception window และจะมีการเปลี่ยนแปลงค่าตามการรับของ AMD PDU ที่มี sequence number เท่ากับ *VR(R)*
2. **VR(H)** เก็บ sequence number สูงสุดของ AMD PDU ที่ได้รับ เป็น sequence number ที่มากที่สุดของ AMD PDU ถัดไปที่คาดว่าจะได้รับ
3. **VR(MR)** เป็นค่า sequence number มากที่สุดของ AMD PDU แรกที่จะถูกปฏิเสธหรือละทิ้งจากฝั่งรับ มีค่าเท่ากับ *VR(R) + CONFIGURED_RX_WINDOW_SIZE* ดังนั้นจึงเป็นค่าขอบเขตบนของ reception window และควรมีการเปลี่ยนแปลงค่าใหม่เมื่อค่าของ *VR(R)* ถูกเปลี่ยน
4. **VR(EP)** ถูกเรียกว่าตัวนับ Estimated PDU เก็บจำนวนของ AMD PDU ที่ซึ่งยังคงคาดหวังการส่งซ้ำตามผลของการส่ง STATUS report ล่าสุด

3.6.2 กลไกการส่ง STATUS (STATUS Transmission Mechanism)

จากการได้รับการร้องขอการ poll ฝั่งรับก็จะส่ง STATUS report กลับไปยังฝั่งส่งเพื่อตอบรับการร้องขอ กลไกนี้เป็นอีกวิธีการหนึ่งของการส่ง STATUS report ที่บังคับโดยฝั่งรับ

แทนที่จะเป็นฝั่งส่งกำหนดให้ส่ง กลไกนี้สามารถยอมให้ฝั่งส่ง ส่ง STATUS report มากกว่าเดิม การมี trigger ที่ช่วยในการดักจับ PDUs ที่สูญหายหรือผิดพลาดไปสามารถทำให้ฝั่งรับส่ง STATUS report ไปร้องขอการส่งซ้ำ เมื่อฝั่งรับพบ PDU ที่ผิดพลาด จากประสิทธิภาพของตัว trigger ทำให้เวลาหน่วงของ RLC SDU ลดลง โดย trigger STATUS Period Timer เป็นตัวควบคุมเวลาของฝั่งรับเพื่อให้ส่ง STATUS report กลับไปยังฝั่งส่งภายในคาบเวลาที่กำหนด นอกจากนี้กลไก EPC (EPC mechanism) ทำให้ฝั่งรับส่ง STATUS report ได้มากขึ้นโดยคำนวณประมาณเวลาที่จำเป็นที่ใช้เพื่อกู้คืน AMD PDU ที่ผิดพลาดรวมไว้ใน STATUS report ตัวสุดท้าย ตัวจับเวลา Status Prohibit Timer ควบคุมความถี่ของการส่ง STATUS report ไปให้ฝั่งส่ง ซึ่ง trigger ทุกตัวของกลไกการส่ง STATUS เป็นตัวเลือกเสริมในการกำหนดใช้งานของโปรโตคอล RLC สำหรับ EPC mechanism และ STATUS Prohibit Timer สามารถถูกกำหนดให้ใช้งานในเวลาเดียวกันได้

1. **การตรวจจับ AMD PDU ที่ผิดพลาด** ทันทีที่ฝั่งรับตรวจพบ AMD PDU ที่ผิดพลาด STATUS PDU จะถูกส่งไปยังฝั่งส่ง เพื่อร้องขอให้มีการส่งซ้ำ AMD PDU ที่ผิดพลาดนั้น ๆทั้งหมดที่ตรวจพบ
2. **Timer-based STATUS Transmission** ฝั่งรับจะส่ง STATUS report ไปยังฝั่งส่งในคาบเวลาที่กำหนด ตัวจับเวลา *STATUS Period Timer* ควบคุมคาบเวลาของการส่ง STATUS report ตัวจับเวลาเริ่มนับเมื่อเอ็นทิที้ RLC ถูกสร้าง และเมื่อมันหมดอายุก็จะเกิดการส่ง STATUS report และตัวจับเวลาก็จะเริ่มต้นนับใหม่
3. **EPC Mechanism** ป้องกันการแลกเปลี่ยน STATUS report ที่มากเกินความจำเป็น *EPC Timer* เป็นช่วงของการเรียงลำดับการส่ง STATUS report โดยจะเริ่มต้นนับทุกครั้งที่ STATUS PDU ตัวแรกของ STATUS report ถูกส่งไปยังชั้นล่าง จากนั้นตัวแปร state *VR(EP)* จะถูกกำหนดเป็นค่าจำนวนของ AMD PDU ทีต้องถูกกู้คืน ถ้าไม่มี AMD PDU ที่ร้องขอการส่งซ้ำ หรือเมื่อ *VR(EP)* มีค่าเป็น 0 STATUS report ตัวใหม่จะส่งจากฝั่งรับ อย่างไรก็ตามถ้าการส่ง STATUS report อื่นถูก trig ในขณะที่ *VR(EP)* ไม่เท่ากับ 0 STATUS report นั้นก็ต้องรอไปจนกระทั้ง *VR(EP)* ลดค่าจนเป็น 0
4. **ตัวแปร State Status Prohibit Timer** เป็นตัวจับเวลาที่ขัดขวางฝั่งรับจากการส่ง STATUS report ต่อเนื่องกัน เริ่มต้นนับทุกครั้งที่ STATUS PDU แรกของ STATUS report ถูกส่งไปยังชั้นล่าง ฝั่งรับไม่ได้รับอนุญาตให้ส่ง STATUS report อื่นจนกว่าตัวจับเวลาจะหมดอายุ
5. **EPC Mechanism และ STATUS Prohibit Timer ถูกติดตั้งค่า** ในกรณีของทั้ง EPC mechanism และ STATUS Prohibit Timer ถูกติดตั้งค่ากำหนดให้ใช้งานในเวลาเดียวกัน การส่ง STATUS report จำเป็นต้องหน่วงไว้จนกระทั้งค่าสูงสุดของ

VR(EP) และ *Status Prohibit Timer* หมดอายุ ถ้าการส่ง STATUS report มากกว่าหนึ่งอันถูก trig ก่อนที่ค่าสูงสุดของ *VR(EP)* และ *Status Prohibit Timer* หมดอายุ จะมีเฉพาะ STATUS report เดียวที่จะถูกส่งออกไปเมื่อมันหมดอายุ

3.7 การทำงานของ TTI

การทำงานของ TTI เกิดขึ้นทุกครั้งที่ TTI ตัวเก่าหมดอายุและอีกตัวหนึ่งเริ่มทำงาน การทำงานรวมถึงการส่ง STATUS PDU และ AMD PDU ไปยังชั้นล่าง ลดค่าตัวจับเวลาทุกตัวที่เกี่ยวข้อง และตรวจสอบทุก ๆครั้งถ้าตัวจับเวลาใด ๆ หมดอายุ เมื่อส่ง AMD PDU ไปยังชั้นล่าง controls PDU มีลำดับความสำคัญสูงกว่า ถัดมาเป็น AMD PDU ที่ร้องขอการส่งซ้ำและ AMD PDU ที่ส่งเป็นครั้งแรกมีลำดับความสำคัญต่ำสุด ก่อนการส่ง AMD PDU ไม่ว่าจะเป็นตัวใหม่ที่สร้างขึ้นหรือตัวที่เป็นการส่งซ้ำก็ตามถ้ามีการกำหนดให้เรียกใช้บิต poll แล้วกลไกการ poll จำเป็นต้องถูกเริ่มต้นทำงานก่อนเพื่อตรวจสอบ หลังจากที่ AMD PDU นั้นถูกส่งออกไปแล้ว ตัวแปร state ที่เกี่ยวข้องในฝั่งส่งจะเกิดการเปลี่ยนแปลงค่า เช่น VT(S), VT(DAT), VT(PDU), VT(SDU) เป็นต้น

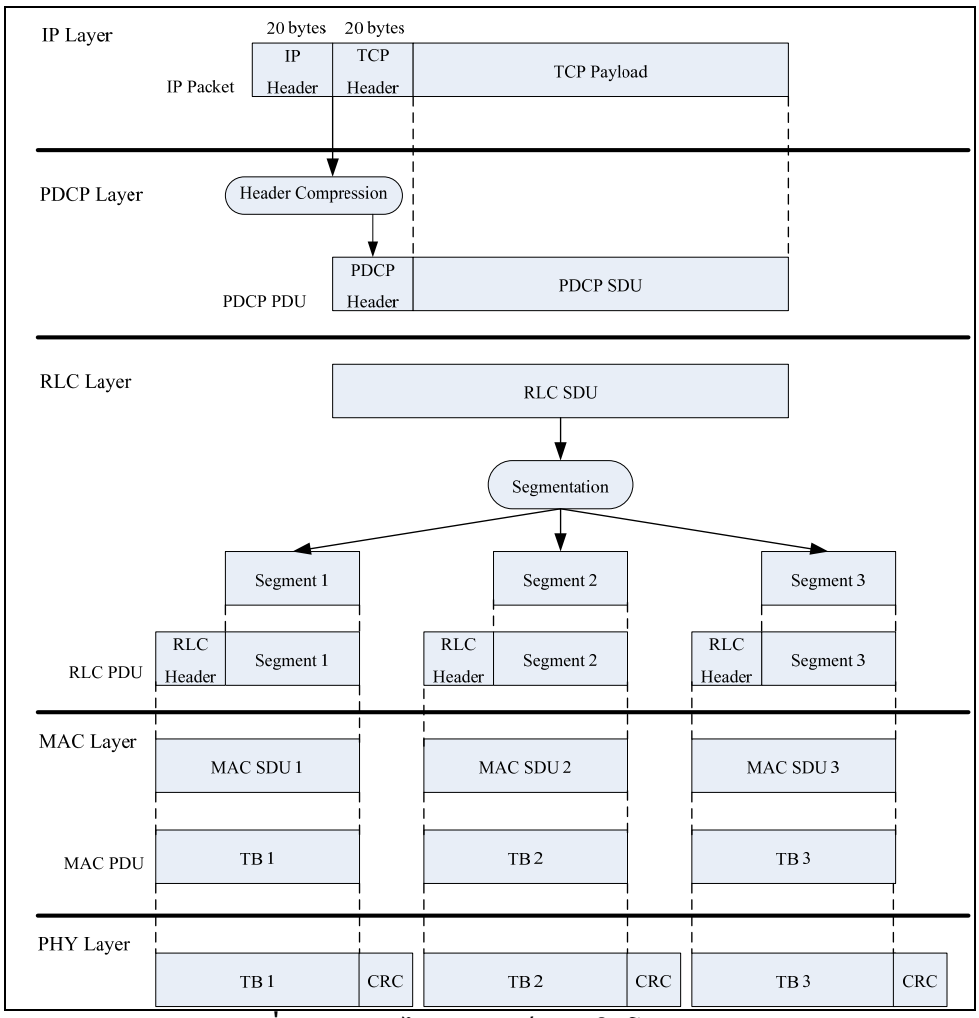
ในขั้นตอนของการลดค่าของตัวจับเวลา ตัวจับเวลาที่ถูกระบุให้ใช้งานทั้งหมดจำเป็นต้องถูกลดค่าครั้งละ 1 ตัวจับเวลาที่เลือกใช้งานอาจจะเป็น EPC Timer หรือ VR(EP), STATUS Prohibit Timer, STATUS Period Timer, Poll Timer, Poll Periodic, Poll Prohibit Timer, Discard Timer สำหรับแต่ละ AMD PDU และ MRW Timer หลังจากลดค่าตัวจับเวลาแล้วการตวรจสอบการหมดอายุของตัวจับเวลาเหล่านี้จะถูกพิจารณาโดยกลไกการ poll, กลไกการส่งSTATUS และกลไกการละทิ้ง SDU ของ RLC ขึ้นอยู่กับว่ากลไกนั้น ๆ เกี่ยวข้องกับตัวจับเวลาตัวใด การตรวจสอบ EPC Timer หรือ VR(EP), Poll Timer, Poll Periodic Timer และ Poll Prohibit Timer จะอยู่ในกลไกการ poll การตรวจสอบ Status Prohibit Timer และ Status Periodic Timer จะอยู่ในกลไกการส่งSTATUS และการตรวจสอบ MRW Timer และ Discard Timer สำหรับแต่ละ AMD PDU จะถูกทำในกลไกการละทิ้ง SDU ของ RLC

3.8 ตัวอย่างการไหลของแพ็กเกต

ลักษณะข้อมูลที่ไหลผ่านโปรโตคอลชั้นที่ 2 จะถูกกำหนดโดยโหมดการทำงานของชั้น RLC (TM, UM หรือ AM) ร่วมกับประเภทของการส่งข้อมูลบนชั้น MAC เช่น ชั้น MAC อาจจะต้องการหรือไม่ต้องการให้มี header ของชั้นก็ได้ เป็นต้น ตัวอย่างเช่น ในบางกรณีหาก UE มีเพียงช่องสัญญาณ dedicated เพียงช่องเดียว header ของชั้น MAC อาจไม่ต้องเพิ่มใส่ลงไปในแพ็กเกต รูปที่ 3.17 แสดงตัวอย่างการไหลของแพ็กเกตเมื่อใช้งานเอ็นทิตี้ AM บน

ช่องสัญญาณ dedicated

โปรโตคอล MAC ที่อยู่ใต้ชั้น RLC จัดการบริการการขนถ่ายข้อมูล ในรูปแบบของ MAC SDUs บนช่องสัญญาณลอจิก เนื่องจากโหมดการขนถ่ายข้อมูลของชั้น MAC เป็นแบบไม่มีการตอบกลับ (unacknowledged) ดังนั้น RLC ที่มีหน้าที่ส่งข้อมูลของโปรโตคอลชั้นที่สูงกว่าจำเป็นต้องมีความน่าเชื่อถือ และด้วยเหตุที่ MAC ไม่สามารถตัดแบ่ง SDU ขนาดใหญ่ให้มีขนาดเป็นไปตามที่รูปแบบแพ็กเกตการส่งข้อมูลที่เป็นไปได้ เพราะฉะนั้นการตัดแบ่ง SDU จึงจำเป็นที่ RLC ต้องจัดการให้เรียบร้อยก่อนนำส่งผ่านชั้น MAC โดยโปรโตคอล MAC ใช้ transport block เป็นยูนิตในการขนส่ง ขนาดของ transport block สำหรับแต่ละช่องสัญญาณถูกกำหนดจากชั้นกายภาพของ WCDMA (WCDMA physical layer) ในรูปแบบของการให้นิยามชุดของรูปแบบการส่ง (transport format set definition) โปรโคตอลชั้น MAC ทำการมัลติเพล็ก PDU จากโปรโตคอลชั้นบนเข้าสู่ชุด transport block แล้วขนย้ายผ่านช่องสัญญาณ dedicated และ common การสับเปลี่ยนระหว่างช่องสัญญาณ common, dedicated และ shared ถูกจัดการโดยชั้น MAC เช่นกันแม้ว่าคำสั่งในการสับเปลี่ยนถูกกำหนดมาจากชั้น RRC และสำหรับทุกโหมดการทำงานของ RLC การค้นหาข้อผิดพลาด CRC จะทำที่ชั้นกายภาพและผลของการตรวจสอบ CRC จะถูกส่งให้ชั้น RLC พร้อมกับข้อมูลแท้จริง



รูปที่ 3.17 การไหลของแพ็กเกตในโหมด AM

3.9 สรุป

จากรายละเอียดข้างต้นจะเห็นว่า RLC เป็นโปรโตคอลที่ยากที่จะเข้าใจเพราะมีพารามิเตอร์จำนวนมากที่จำเป็นจะต้องมีการปรับค่า และเนื่องจากการเปลี่ยนแปลงตามการทำงานของระบบ UMTS ทำให้พารามิเตอร์หลาย ๆตัวต้องมีการกำหนดค่าใหม่โดยเลือกจากสภาวะแวดล้อมของเครือข่ายในเวลานั้นจึงทำให้โปรโตคอลนี้มีความยืดหยุ่นสูง อย่างไรก็ตามการเลือกใช้ค่าที่เหมาะสมเป็นสิ่งที่ท้าทายอย่างมากเพราะเหตุที่ว่าความซับซ้อนของการทำงานร่วมกันของทั้งโปรโตคอล และจำนวนพารามิเตอร์ที่เกี่ยวข้องมีจำนวนมาก การกำหนดค่าที่ไม่เหมาะสมอาจเป็นสาเหตุที่ทำให้ประสิทธิภาพการทำงานของ RLC ลดลงหรือเกิดการติดตาย (deadlock)[29] ด้วยเหตุผลเหล่านี้ทำให้การประเมินประสิทธิภาพของโปรโตคอล RLC ด้วยวิธีการสร้างแบบจำลองจึงเป็นสิ่งจำเป็น