บทที่2

การสื่อสารข้อมูล

การสื่อสารข้อมูล หมายถึง การรับส่งข้อมูลระหว่างอุปกรณ์ที่อยู่ต่างสถานที่กัน การรับส่งข้อมูลในปัจจุบันจะอยู่ในสภาพของเครือข่ายข้อมูล (Data Networks) ซึ่งจะเป็นการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ซึ่งเชื่อมโยงกันเป็นเครือข่าย หากจะพิจารณาเครือข่ายข้อมูลตามโครงสร้างซึ่งเรียกว่าเลเยอร์(Layer) ตามระบบของไอเอสโอ (ISO หรือที่เรียกว่า International Standards Organization) จะสามารถแบ่งออกได้เป็น 7 เลเยอร์โดยแต่ละเลเยอร์จะเป็นมอดูล (module) ที่ทำหน้าที่ต่าง ๆ กันในระบบเครือข่ายข้อมูล ความสัมพันธ์ระหว่างเลเยอร์ต่าง ๆ แสดงได้ดังรูป 2.1 โครงสร้างที่เห็นนี้เป็นแบบจำลองที่เรียกว่าโอเอสไอ (Open System Interconnection) ระบบเครือข่ายที่มีอยู่ในปัจจุบันจะแตกต่างจากระบบโอเอสไอบ้าง อย่างไรก็ตามในที่นี้ขอกล่าวถึงเลเยอร์ในระบบโอเอสไอเท่านั้น

เลเยอร์ของเครือข่ายข้อมูล

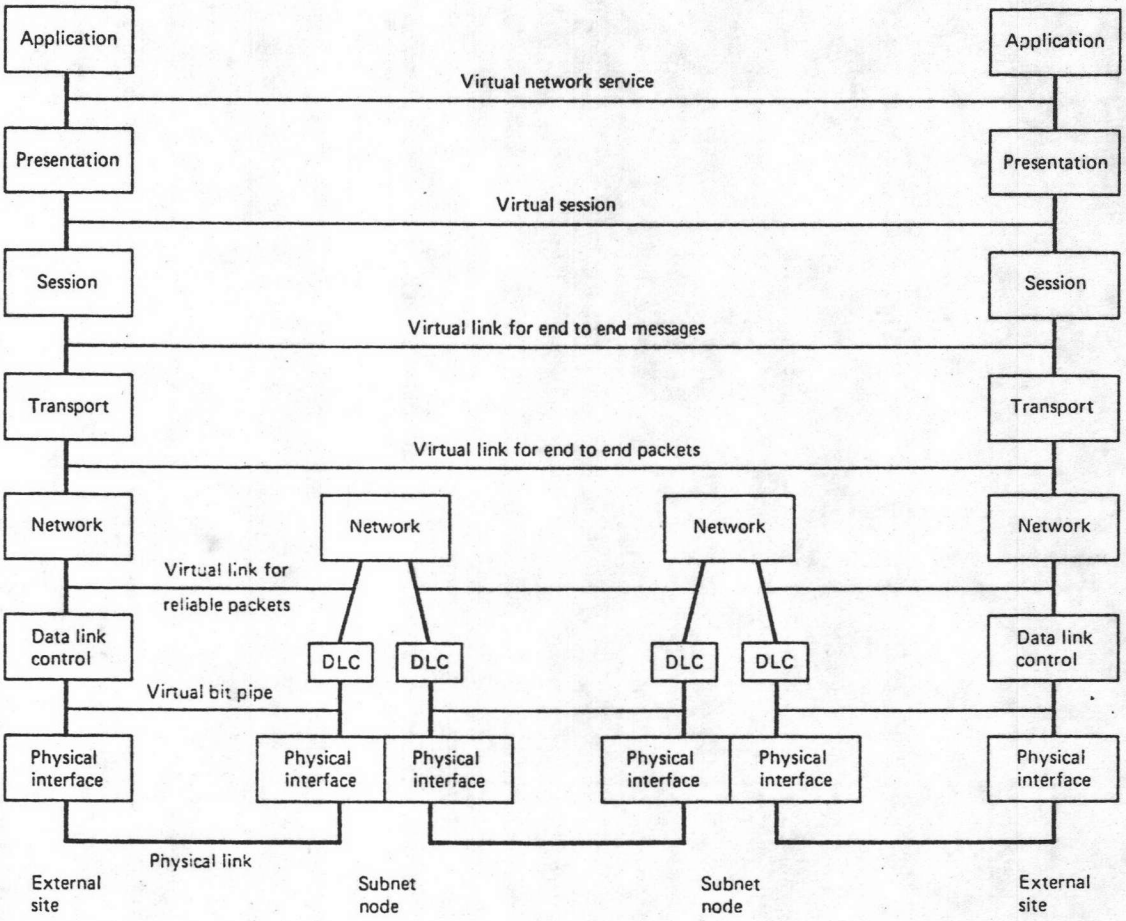
ไอเอสโอได้แบ่งเลเยอร์ของเครือข่ายข้อมูลออกเป็น 7 เลเยอร์ดังนี้ (2)

1. ฟิสิคัลเลเยอร์ (Physical Layer) หน้าที่ของฟิสิคัลเลเยอร์คือการส่วนที่เชื่อมโยงการสื่อสารข้อมูลระหว่างโนด (Node) โดยรับส่งข้อมูลที่ส่งมาในลักษณะเป็นบิตจากเลเยอร์ถัดไปคือดาตาลิงค์คอนโทรลเลเยอร์ และเปลี่ยนให้เป็นสัญญาณที่เหมาะสมกับอุปกรณ์ที่ใช้ในการเชื่อมโยง หรือเปลี่ยนสัญญาณให้กลับเป็นข้อมูล โดยปกติอุปกรณ์ที่ทำหน้าที่นี้จะเป็นอุปกรณ์เรียกว่า โมเดม (Modem)

การรับส่งข้อมูลในเลเยอร์นี้อาจพิจารณาตามลักษณะของข้อมูลได้เป็น 3 ชนิด

1.1 แบบซิงโครนัส มีการรับข้อมูลในช่วงเวลาสม่ำเสมอเช่น 1 บิต ต่อ 1 วินาที โดยมอดูลของเลเยอร์ถัดไปคือ ดีแอลซีมอดูลจะต้องส่งข้อมูลในอัตราเดียวกันกับการส่งนี้ไม่ว่าจะมีหรือไม่มีข้อมูลที่ต้องการส่งหรือไม่

1.2. แบบอินเตอร์มิทเทนท์ซิงโครนัส (Intermittent synchronous) โดยดีแอลซีมอดูลจะส่งข้อมูลในอัตราเดียวกันกับการส่งแบบซิงโครนัสเมื่อมีข้อมูลที่ต้องการและจะหยุดส่งข้อมูลเมื่อไม่มีข้อมูลที่ต้องส่ง



รูปที่ 2.1 แสดงเลเยอร์ของเน็ทเวอร์คตามโอเอ็สไอ

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.3 แบบอะซิงโครนัสคาเรคเตอร์ (asynchronous characters) โดยปกติใช้ในเครื่องไมโครคอมพิวเตอร์และเทอร์มินัลความเร็วต่ำ โดยตัวอักษรและรหัสควบคุมจะถูกแบ่งเป็นชุดของบิตที่ยาวเท่า ๆ กัน ซึ่งมักจะมีขนาด 8 บิตตามรหัสแอสกี (ASCII Code) การส่งข้อมูลจะส่งตามตัวอักษรที่ได้รับในขณะนั้น ๆ

การเชื่อมโยงระหว่างดีแอลซีมอดูล และ โมเดม โดยปกติจะเชื่อมโยงโดยอาศัยอาร์เอส232 หรือเอ็ก 21 อาร์เอส 232 จะมีจุดต่อสัญญาณอยู่ 21 จุด แต่ในทางปฏิบัติจะไม่ได้ใช้สัญญาณทั้งหมดคงใช้เฉพาะเท่าที่จำเป็นแค่นั้น การใช้งานจะเป็นการเชื่อมโยงระหว่างดีซีอี (Data Communication Equipment) ในที่นี้คือ โมเดมและ ดีทีอี (Data Terminal Equipment) ในที่นี้คือ ดีแอลซีเลเยอร์

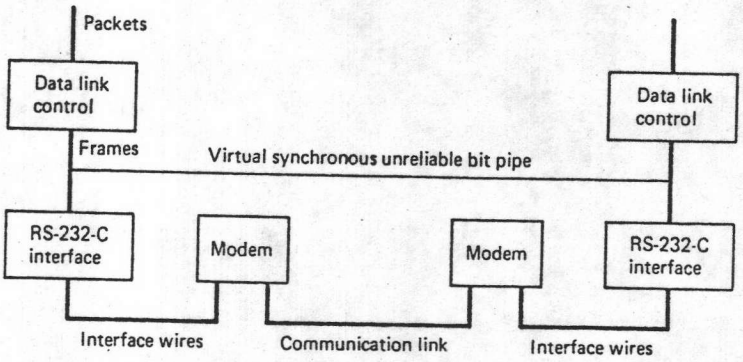
ตัวอย่างของการเชื่อมโยง สมมุติว่า ดีทีอี ต้องการส่งข้อมูล ดีทีอี จะส่งสัญญาณไปยัง ดีซีอี โดยผ่านสายสัญญาณอาร์ทีเอ็ส (Request to send) ดีซีอี จะตอบกลับด้วยสัญญาณ ซีทีเอ็ส (clear to send) เมื่อโมเดมพร้อมที่จะทำงานจะส่งสัญญาณดีซีอีพร้อม (DCE ready) และเมื่อโมเดมพบสัญญาณพาหะจากโมเดมอีกด้านหนึ่งจะส่งสัญญาณตรวจพบพาหะ (carrier detect) ดีทีอีจะต้องพบสัญญาณเหล่านี้จึงจะเริ่มทำการส่งข้อมูลระหว่าง ดีทีอี และดีซีอี การเชื่อมโยงของมอดูลแสดงได้ดังรูป 2.2

ส่วนเอ็ก 21 จะคล้ายกันกับอาร์เอส 232 แต่จะมีสายสัญญาณน้อยกว่าคือใช้สายสัญญาณจำนวน 8 สาย (แม้ว่าจะมีขั้วต่อสัญญาณ 15 จุด) โดยนิยมใช้กับโปรโตคอลเอ็ก 25

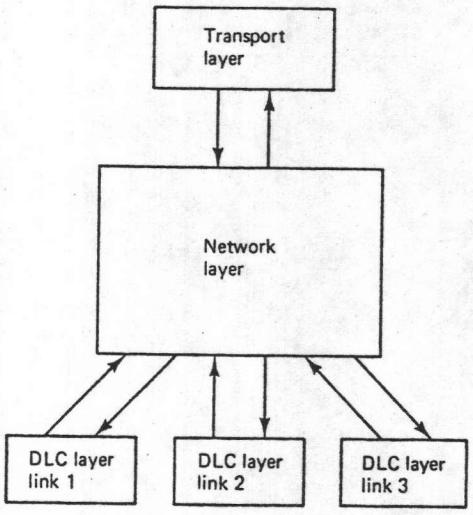
2. <u>ดาต้าลิงค์คอนโทรลเลเยอร์</u> (ดีแอลซี หรือ DLC)

เป็นเลเยอร์ที่สองตามรูป 2.1 การเชื่อมโยงเป็นแบบจุดต่อจุด (Point to Point) โดยจะมีดาต้าลิงค์คอนโทรลมอดูลที่ปลายของการเชื่อมโยงเสมอ วัตถุประสงค์ของดาต้าลิงค์คอนโทรลคือ ทำการเปลี่ยนข้อมูลจากเลเยอร์แรกไปยังเลเยอร์ที่สูงขึ้นไปอีกขั้นหนึ่ง และ ทำการส่งแพคเก็ท (packets) ระหว่างการเชื่อมโยง ซึ่งสำหรับดีแอลซีนั้นแพคเก็ทคือชุดของบิตที่ได้รับมาจากเลเยอร์สูงกว่าที่อยู่ถัดไป

เมื่อดีแอลซีมอดูลจะส่งข้อมูลจะมีการเพิ่มข้อมูลสำหรับการควบคุมที่ส่วนต้นและส่วนท้ายของแพคเก็ทและให้ผลลัพธ์เป็นชุดของข้อมูลที่เรียกว่าเฟรม (Frame) บิตที่ใช้ในการควบคุมนี้จะประกอบไปด้วยส่วนสำหรับบอกความผิดพลาดในการส่งข้อมูล จุดเริ่มต้นและสิ้นสุดของเฟรม และ การขอร้องให้ส่งข้อมูลใหม่เมื่อเกิดข้อผิดพลาด การตรวจสอบต่าง ๆ นี้จะตรวจสอบโดยโปรโตคอลของเพียร์ดีแอลซีมอดูล (Peer DLC Module)



รูปที่ 2.2 แสดงการเชื่อมโยงระหว่างดีแอลซีและโมเด็ม



รูปที่ 2.3 แสดงการทำงานของเน็ทเวอร์คเลเยอร์

3. เน็ทเวอร์คเลเยอร์ (Network Layer)

สำหรับเน็ทเวอร์คเลเยอร์นี้จะมีหน้าที่ในการควบคุมเส้นทางการรับส่งข้อมูลและการไหลของข้อมูลในเน็ทเวอร์ค จากรูป 2.3 จะเห็นว่าเมื่อแพคเก็ทผ่านเข้ามาที่โนดทางฟิสิคัลเลเยอร์แล้วจะเดินทางมาถึงดีแอลซีเลเยอร์ ถ้าดีแอลซีเลเยอร์ยอมรับว่าข้อมูลที่ได้รับถูกต้องก็จะส่งข้อมูลที่ได้รับเข้าสู่เน็ทเวอร์คเลเยอร์ของโนดนั้นต่อไป เน็ทเวอร์คเลเยอร์จะทำการกำหนดเส้นทางและจุดหมายของแพคเก็ทที่ได้รับมาจากเลเยอร์ที่สูงกว่า และ ส่งไปยังดีแอลซีเลเยอร์ตามเส้นทางที่ถูกต้อง ในทำนองเดียวกันเมื่อเน็ทเวอร์คเลเยอร์ได้รับข้อมูลจากดีแอลซีเลเยอร์ก็จะส่งข้อมูลนั้นไปยังทรานสปอร์ตเลเยอร์

4. ทรานสปอร์ตเลเยอร์ (Transport Layer)

หน้าที่ของทรานสปอร์ตเลเยอร์มีอยู่หลายประการแต่โดยปกติเน็ทเวอร์คไม่จำเป็นต้องมีคุณสมบัติเหล่านี้ทั้งหมด คุณสมบัติของทรานสปอร์ตเลเยอร์มีดังนี้คือ

4.1 ทำหน้าที่ตัดทอนข่าวสารให้เป็นแพคเก็ทสำหรับด้านส่งข้อมูลและทำการรวบรวมแพคเก็ทให้เป็นข่าวสารที่ด้านรับข้อมูล การรวบรวมข้อมูลนี้จะสามารถทำได้ง่ายถ้ามีบัฟเฟอร์ที่มีขนาดเพียงพอ แต่จะทำได้ยากขึ้นถ้าแพคเก็ทที่รับไม่อยู่ในลำดับที่เรียงกันหรือกรณีที่บัฟเฟอร์มีขนาดจำกัด

4.2 ทำการมัลติเพลกซ์เซสชั่น (multiplex session) ที่มีอัตราการส่งข้อมูลต่ำจากแหล่งกำเนิดหรือรับข้อมูล ให้เป็นเซสชั่นเดียวกันที่เน็ทเวอร์คเลเยอร์

4.3 ทำการเชื่อมโยงเน็ทเวอร์คที่แตกต่างกันเข้าด้วยกัน โดยปกติมักจะเชื่อมโยงกันด้วยโนดที่เรียกว่าเกทเวย์ (gateways) ซึ่งจะทำการรับแพคเก็ทจากเน็ทเวอร์คเลเยอร์ชนิดหนึ่งและเปลี่ยนให้อยู่ในรูปที่เหมาะสมสำหรับเน็ทเวอร์คอีกชนิดหนึ่ง ในกรณีที่แพคเก็ทมีขนาดแตกต่างกัน ทรานสปอร์ตเลเยอร์ก็จะต้องทำการแบ่งแพคเก็ทนั้นออกเป็นแพคเก็ทที่มีขนาดเล็กลง

5. เซสชั่นเลเยอร์ (Session Layer)

เซสชั่นเลเยอร์มีหน้าที่คล้ายกับส่วนไดเรกทอรีโดยจะเก็บข้อมูลต่าง ๆ ที่จำเป็นสำหรับผู้ใช้ในการขอรับบริการจากเน็ทเวอร์ค นอกจากนี้ยังเป็นส่วนที่ตรวจสอบสิทธิ์ในการสร้างเซสชั่น อาจกล่าวได้ว่าเซสชั่นเลเยอร์ทำหน้าที่ควบคุมการติดต่อระหว่างจุดสองจุดในการสร้างเซสชั่น

6. พรีเซนเตชั่นเลเยอร์ (Presentation Layer)

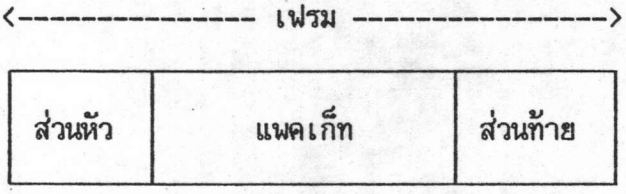
หน้าที่หลักของเลเยอร์นี้คือการทำการเข้ารหัสลับข้อมูล (data encryption) การอัดข้อมูล (data compression) และ การแปลงรหัสของข้อมูล (code conversion) เช่น การเปลี่ยนจากรหัสแอสกีไปเป็นเอ็บซีดิค (ASCII to EBCDIC)

7. แอปพลิเคชั่นเลเยอร์ (Application Layer)

หน้าที่ของเลเยอร์นี้คือการทำงานทั้งหมดที่นอกเหนือจากหน้าที่ของเลเยอร์ที่ผ่านมาทั้ง 6 ชนิด ซึ่งตามปกติแล้วแต่ละแอปพลิเคชั่นจะมีซอฟแวร์ของตนเองในการทำงาน

การตรวจสอบความผิดพลาด

การส่งข้อมูลตามปกติจะมีการส่งในลักษณะที่เรียกว่าเฟรม ซึ่งจะประกอบไปด้วยส่วนหัว (Header) แพคเก็ท และส่วนท้าย (Trailer) การแก้ไขข้อผิดพลาดโดยปกติจะทำด้วยวิธีที่เรียกว่าเออาร์คิว (ARQ หรือ Automatic Repeat Request) ซึ่งวิธีนี้จะอาศัยการขอข้อมูลใหม่จะอาศัยส่วนที่เป็นส่วนหัวและส่วนท้ายของเฟรมดังรูปที่ 2.4



รูปที่ 2.4 ลักษณะของเฟรม

การตรวจสอบความผิดพลาดของข้อมูลในเฟรมสามารถตรวจสอบได้ 4 วิธีคือ

1. การตรวจพาริตี้เดี่ยว (Single Parity Checks)

การตรวจสอบด้วยวิธีนี้ใช้วิธีการเพิ่มบิตหนึ่งบิตต่อท้ายข้อมูล เราเรียกบิตที่เพิ่มขึ้นมานี้ว่าพาริตี้บิต ค่าของพาริตี้บิตจะเป็น 1 ถ้าจำนวนบิตที่เป็น 1 ของข้อมูลเป็นเลขคี่ และ 0 เมื่อเป็นเลขคู่ การตรวจสอบด้วยวิธีนี้จะพบข้อผิดพลาดในกรณีที่จำนวนบิตที่เกิดผิดพลาดเป็นเลขคี่เท่านั้น สำหรับกรณีที่จำนวนบิตที่ผิดพลาดเป็นเลขคู่นั้นไม่สามารถตรวจสอบได้(2) การตรวจสอบโดยวิธีนี้อาจพบได้ในการส่งข้อมูลที่เป็นรหัสแอสกี ซึ่งจะประกอบด้วยข้อมูลและพาริตี้บิตดังรูปที่ 2.5

S1	S2	S3	S4	S5	S6	S7	Parity
1	0	1	1	0	0	0	1

รูปที่ 2.5 แสดงพาริตี้สำหรับแอสกีขนาด 7 บิต

2. <u>การตรวจสอบพาริตี้แนวนอนและแนวตั้ง</u> (Horizontal and Vertical Parity Checks)

เมื่อต้องการให้การตรวจสอบความผิดพลาดได้ดีขึ้นจะสามารถทำได้โดยการจัดให้ข้อมูลอยู่ในลักษณะข้อมูลแบบอะเรย์ 2 มิติ ดังรูปที่ 2.6 ก โดยมีพาริตี้สำหรับตรวจสอบทั้งในแต่ละแถวและสดมภ์ ในกรณีที่เกิดการผิดพลาดของจำนวนบิตที่เป็นจำนวนคู่ในแต่ละแถวจะสามารถตรวจพบได้โดยอาศัยพาริตี้ในแนวสดมภ์ ในทำนองเดียวกันการผิดพลาดในแนวสดมภ์ก็ตรวจสอบได้โดยพาริตี้ในแนวแถว แต่ข้อผิดพลาดที่ยังคงเหลืออยู่คือ กรณีที่เกิดข้อผิดพลาดใน 2 แถว 2 สดมภ์ ดังแสดงโดยรูปที่ 2.6 ข ถ้าบิตในวงกลมเกิดการเปลี่ยนแปลงไปจะไม่สามารถตรวจสอบได้

3. <u>การตรวจสอบพาริตี้เช็คโค้ด</u> (Parity Check Codes)

การตรวจสอบโดยวิธีนี้อาศัยหลักทางคณิตศาสตร์ในการสร้างพาริตี้โดยแบ่งข้อมูลที่ต้องการส่งตามจำนวนบิต และสร้างพาริตี้แทรกไประหว่างชุดของบิตนั้นก่อให้เกิดชุดของบิตชุดใหม่ดังแสดงในรูปที่ 2.7 การใช้วิธีนี้ โดยปกติจะใช้เพื่อแก้ไขข้อมูลให้ความถูกต้องดังเดิมได้เมื่อเกิดข้อผิดพลาดในการส่ง การแก้ไขด้วยวิธีนี้มักทำในระดับของฟิสิคัลเลเยอร์มากกว่าการทำในระดับดีแอลซีเลเยอร์

4. <u>การตรวจสอบไซคลิกรีดันแดนซี</u> (Cyclic Redundancy checks)

วิธีตรวจสอบพาริตี้ที่นิยมใช้กันมากในปัจจุบันคือวิธีตรวจสอบแบบไซคลิกรีดันแดนซีหรือซีอาร์ซี (CRC) ถ้าให้ K เป็นความยาวของบิตสำหรับข้อมูล (ประกอบด้วยส่วนหัว และแพคเก็ทของเฟรม) เราสามารถแทนบิตของข้อมูลทั้งหมดด้วย

$$a_{k-1}\ ,\ a_{k-2}\ ,\ \ldots\ldots,\ a_1\ ,\ a_0$$

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
1	0	1	1	1	1	1	0

Horizontal checks

Vertical checks

ก.

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	(1)	0	0	(0)	1	0
1	0	0	0	1	1	1	0
0	0	(1)	1	0	(0)	1	1
1	0	1	1	1	1	1	0

ข.

รูปที่ 2.6 แสดงการตรวจสอบพาริตี้แนวนอนและแนวตั้ง

s_1	s_2	s_3	c_1	c_2	c_3	c_4	
1	0	0	1	1	1	0	
0	1	0	0	1	1	1	$c_1 = s_1 + s_3$
0	0	1	1	1	0	1	$c_2 = s_1 + s_2 + s_3$
1	1	0	1	0	0	1	$c_3 = s_1 + s_2$
1	0	1	0	0	1	1	$c_4 = s_2 + s_3$
1	1	1	0	1	0	0	
0	0	0	0	0	0	0	

รูปที่ 2.7 แสดงการตรวจสอบพาริตี้เช็คโค้ด

และแทนข้อมูลนี้ด้วยโพลิโนเมียล

$$M(x) = a_{k-1}\, x + ...a_1 x + a_o$$

เมื่อทำการคำนวณโพลีโนเมียลนี้ด้วยโพลีโนเมียลกำเนิด P(X) โดยอาศัยคณิตศาสตร์แบบมอดูโลสอง (modulo-2 arithmatic) การคำนวณแบบมอดูโลสองมีหลักการดังแสดงในตารางที่ 2.1 ดังนี้

ค่าแรก	ค่าที่สอง	ค่าผลบวก	ค่าผลต่าง	ผลเอกคลูซีฟออร์
0	0	0	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	0

ตารางที่ 2.1 เปรียบเทียบมอดูโลสองและเอกคลูซีฟออร์

การเลือกโพลีโนเมียล P(X) ต้องเลือกให้มีดีกรี (degree) น้อยกว่าโพลีโนเมียล M(X) และต้องมีอย่างน้อย 2 เทอม เมื่อเลือกโพลีโนเมียลกำเนิดที่ต้องการได้แล้วจะสามารถหาซีอาร์ซีได้ โดยขั้นตอนต่อไปนี้ (3)

4.1 คูณ M(X) ด้วย X^r เมื่อ r เป็นดีกรีของโพลีโนเมียล P(X) ผลลัพธ์ของการคูณจะเป็นการเพิ่มบิต 0 จำนวน r บิตที่ตำแหน่งบิตลำดับต่ำ(low order bit) ได้สตริงขนาด N บิตซึ่ง N = r+k โดยที่บิตลำดับสูง (high order bit) จำนวน k บิต ไม่มีการเปลี่ยนแปลง

4.2 หาร $M(X).X^r$ ด้วย P(X) ได้ผลจากการหาร Q(X) และเศษเหลือ R(X) การหาเศษเหลือนี้โดยปกติจะทำโดยอาศัยฮาร์ดแวร์ ตัวอย่างการหารโพลีโนเมียล $X^5 + 0X^4 + X^3 + 0X^2 + 0X + 1$ ซึ่งเขียนเป็นรูปย่อได้เป็น 1 0 1 0 0 1 ด้วยค่าโพลิโนเมียลกำเนิด $X^2 + 0X + 1$ ซึ่งเขียนเป็นรูปย่อได้เป็น 1 0 1 จะแสดงได้ดังนี้

```
                1   0   0   0      = ผลการหาร
          ┌──────────────────
101       │ 1   0   1   0   0   1
          │ 1   0   1
          │ ─────────
          │     0   0   0
          │     0   0   0
          │     ─────────
          │         0   0   0
          │         0   0   0
          │         ─────────
          │             0   0   1  = เศษเหลือ
```

4.3 ข้อความที่ต้องส่งจะสามารถแทนได้ด้วย T(X) ซึ่งแสดงได้โดย

$$T(X) = M(X).X^{r} + R(X)$$

จากขั้นตอนทั้ง 3 ที่ผ่านมาจะพบว่า ข้อความที่จะส่งจะมีค่าเท่ากับการส่งข่าวสารจำนวน K bit และตามด้วยเศษเหลือ R(X) ที่ได้จากขั้นตอนที่ 3 ซีอาร์ซี มาตรฐานสำหรับ 16 บิตประกอบด้วย ซีอาร์ซี-16 (CRC-16) และซีอาร์ซี-ซีซีไอทีที (CRC-CCITT) ซึ่งมีโพลีโนเมียลกำเนิดดังนี้

$P(X) = X^{16} + X^{15} + X^{2} + 1$ สำหรับซีอาร์ซี-16

$P(X) = X^{16} + X^{12} + X^{5} + 1$ สำหรับซีอาร์ซี-ซีซีไอทีที

<u>การส่งข้อมูลใหม่แบบเออาร์คิว</u>

วิธีการเบื้องต้นในการแก้ไขข้อมูลเมื่อเฟรมที่รับมีข้อมูลผิดพลาดคือ การขอให้ส่งข้อมูลใหม่โดยอัตโนมัติ (เออาร์คิว) เมื่อดีแอลซีมอดูลฝ่ายรับข้อมูลพบว่าเฟรมที่ได้รับมีข้อมูลผิดพลาดจะทำการขอร้องให้ดีแอลซีมอดูลฝ่ายส่งข้อมูลทำการส่งข้อมูลที่ผิดพลาดนั้นใหม่ จากที่ได้กล่าวมาแล้วว่าเมื่อแพคเก็ทจากเน็ทเวอร์คเลเยอร์เข้าสู่ดีแอลซีเลเยอร์ จะมีการเพิ่มส่วนหัวและท้ายของแพคเก็ทโดยดีแอลซีมอดูลทำให้เกิดเป็นเฟรม และจะทำการส่งเฟรมโดยผ่านฟิสิคัลเลเยอร์ เมื่อพบข้อผิดพลาดในเฟรมจะเกิดการส่งข้อมูลเฟรมนั้นใหม่ ดังนั้นจึงเป็นไปได้ว่าการส่งข้อมูลจะเป็นลักษณะระดับนี้คือ แพคเก็ทแรก แพคเก็ทสอง แพคเก็ทแรกที่ถูกส่งใหม่ เป็นต้น ในการส่งแพคเก็ทใหม่นี้ ส่วนหัวและท้ายของเฟรมไม่จำเป็นต้องเหมือนกับเฟรมที่เกิดความผิดพลาดโดยขึ้นอยู่กับโปรโตคอลที่ใช้ในการรับส่งข้อมูล

ในการพิจารณานี้จะสมมุติว่าเฟรมที่ส่งนี้มีการล่าช้าระหว่างการรับส่งได้ต่าง ๆ กัน นอกจากนี้บางเฟรมอาจสูญหายไปได้ โดยที่ไม่ถึงฝ่ายรับ อย่างไรก็ตามเฟรมที่ฝ่ายรับรับได้ จะมีลำดับของเฟรมตามลำดับในการส่ง ดังแสดงในรูปที่ 2.8

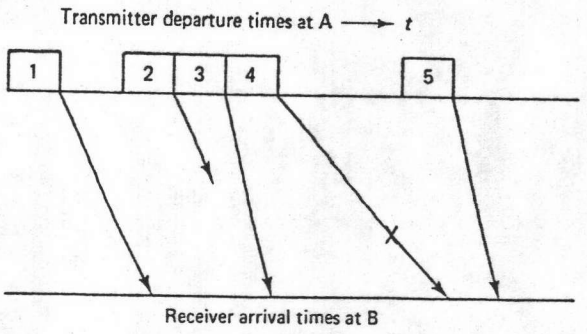
1. เออาร์คิวหยุดและคอย (stop and wait ARQ)

โปรโตคอลที่ง่ายที่สุดในการส่งข้อมูลใหม่คือแบบหยุดและคอย แนวความคิดพื้นฐานนี้ใช้หลักการที่ว่า แต่ละแพคเก็ทที่ได้รับจะต้องถูกต้องเสียก่อนจึงจะส่งแพคเก็ทถัดไปได้ ดังนั้นการส่งแพคเก็ทจากจุด A ไปยัง B นั้นแพคเก็ทแรกจะถูกส่งไปในเฟรมแรก จากนั้นดีแอลซีฝ่ายส่งจะอยู่ในสภาวะรอคอย ฝ่าย B เมื่อรับเฟรมนั้นได้จะตรวจสอบว่า เฟรมนั้นมีข้อมูลผิดพลาดหรือไม่ ถ้าถูกต้องจะส่งการตอบรับ (acknowledgment หรือ ACK) กลับไปยัง A ถ้าข้อมูลผิดพลาดจะส่งการตอบปฏิเสธ (negative acknowledgment หรือ NAK) ไปยัง A เช่นกัน เนื่องจากระหว่างการส่งข้อมูลจาก B ไปยัง A หรือจาก A ไปยัง B สามารถเกิดความผิดพลาดได้เสมอ ดังนั้นจึงมีการส่งซีอาร์ซี พร้อมกับการตอบรับหรือปฏิเสธนี้ด้วย

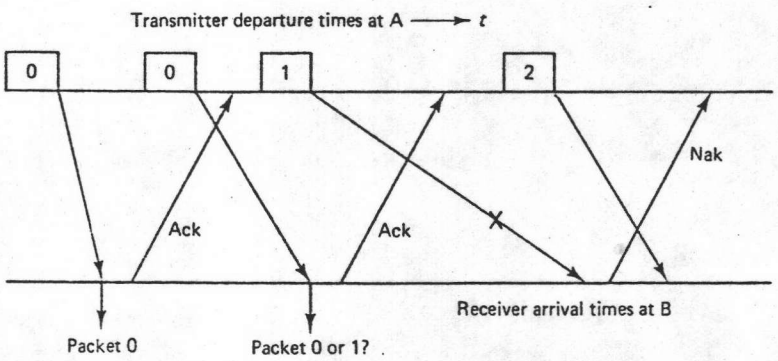
ในกรณีที่การรับข้อมูลที่ B ไม่มีข้อมูลผิดพลาด A จะได้รับการตอบรับ จากนั้น A จะส่งแพคเก็ทใหม่ในเฟรมถัดไป ถ้าเกิดความผิดพลาดในข้อมูลจะเกิดการส่งแพคเก็ทเดิมในเฟรมใหม่ ในกรณีที่เฟรมหรือการตอบรับหรือปฏิเสธเกิดสูญหายไป A จะรออยู่ช่วงเวลาหนึ่งและทำการส่งแพคเก็ทนั้นใหม่ ในกรณีที่ความล่าช้าในการส่งจะยาวนานเท่าไรก็ได้นั้นมีความเป็นไปได้ที่ A จะรอจนสิ้นสุดระยะเวลาที่กำหนด (time out) และส่งแพคเก็ทเดิมใหม่ ต่อจากนั้นจึงได้รับ ACK หรือ NACK จากแพคเก็ทที่ส่งไปในครั้งก่อน ดังรูปที่ 2.9

จึงเป็นไปได้ว่าแพคเก็ทแรกที่ B รับได้ไม่มีข้อผิดพลาด ต่อจากนั้นก็ได้รับแพคเก็ทเดิมอีกในเฟรมที่สองดังรูปที่ 2.9 จากการที่เรากำหนดว่าแพคเก็ทมีลักษณะเป็นบิตสตริงที่มีค่าอะไรก็ได้นั้น เมื่อแพ็คเก็ทแรกและแพคเก็ทสองที่ได้รับนี้เหมือนกัน ทำให้ B ไม่สามารถบอกความแตกต่างระหว่างการส่งแพคเก็ทใหม่หรือเป็นแพคเก็ทที่สองได้ ปัญหานี้สามารถแก้ไขได้อย่างง่าย ๆ โดยการเพิ่มลำดับที่ของแพคเก็ทเข้าไปในดีแอลซีมอดูล ของฝ่ายส่งข้อมูลซึ่งในที่นี้คือ A

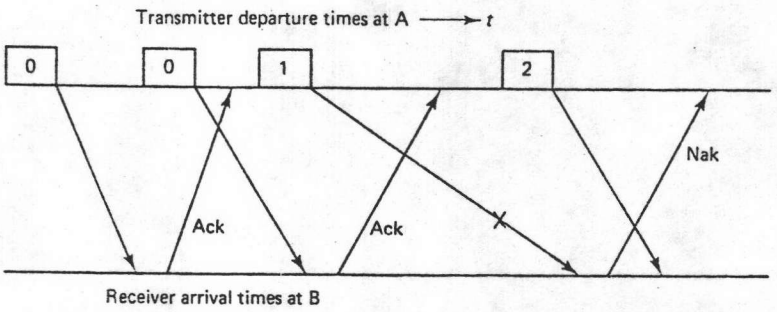
ปัญหาอีกชนิดหนึ่งอาจเกิดขึ้นได้เมื่อการตอบกลับเกิดการสูญหายดังรูป 2.10 เมื่อ B ได้รับแพคเก็ท 0 ถึง 2 ครั้งและข้อมูลที่ได้รับถูกต้อง ขณะที่ B ส่งการตอบรับกลับไปยัง A นั้นฝ่าย A ไม่สามารถจะทราบได้ว่าการตอบรับนั้นเป็นของเฟรมใดแน่ ซึ่งจะก่อให้เกิดข้อผิดพลาดในการรับส่งข้อมูลได้ ทางแก้ปัญหานี้อาจกระทำได้โดยให้ดีแอลซีฝ่ายรับแทนที่จะส่งเฉพาะ ACK หรือ NAK กลับมาจะมีการส่งหมายเลขของแพคเก็ทถัดไปที่รอคอยอยู่ ซึ่งจะ



รูปที่ 2.8 แสดงแบบจำลองในการรับส่งข้อมูล



รูปที่ 2.9 แสดงความกำกวมในการส่งเฟรมที่ไม่มีหมายเลขลำดับ



รูปที่ 2.10 แสดงความกำกวมในการตอบรับที่ไม่มีหมายเลขลำดับ

15661

ช่วยลดความกำกวมในการตอบรับว่าเป็นของเฟรมใดได้ แนวทางแก้ไขอีกวิธีหนึ่งคือ การส่งหมายเลขของแพคเก็ทที่ได้รับนั้นกลับมาด้วย

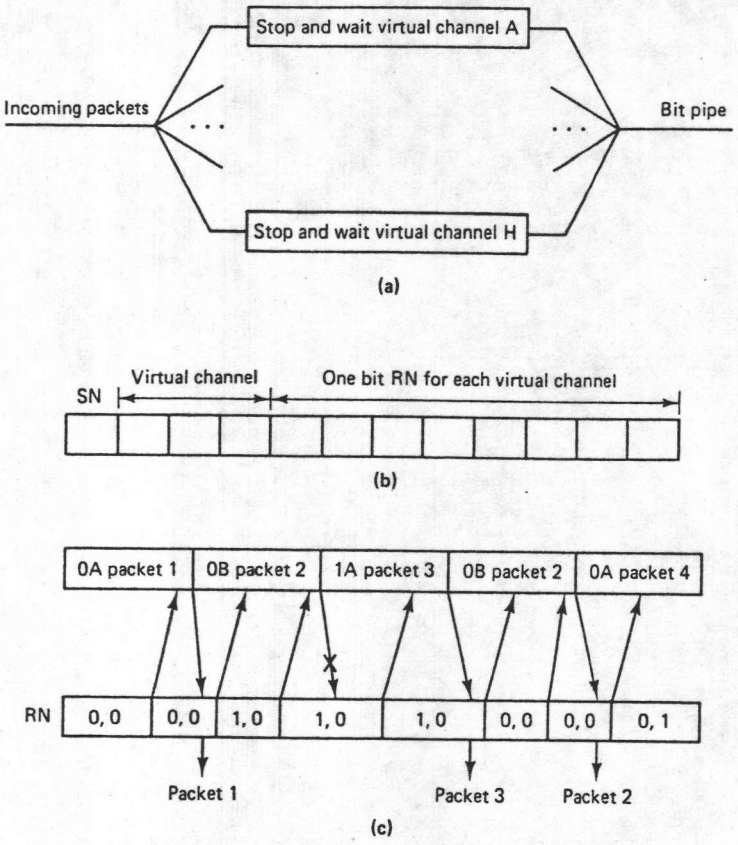
ดังนั้นถ้าการตรวจสอบของ ซีอาร์ซีถูกต้องเราจะพบว่าวิธีการหยุดและคอย โดยที่ฝ่ายส่งได้ส่งหมายเลขส่ง(SN) สำหรับแต่ละแพคเก็ทและฝ่ายรับส่งหมายเลขที่รอคอย(RN) กลับมาด้วยจะทำให้การรับส่งข้อมูลนั้นถูกต้อง เนื่องจาก A จะไม่ส่งแพคเก็ทใหม่นอกจาก B จะขอมาและ B สามารถทราบได้ว่าแพคเก็ทที่ตนรับนั้นเป็นแพคเก็ทใด ปัญหาที่จะอย่างเดียวคือขนาดของหมายเลขส่งและหมายเลขรอคอยจะมีขนาดใหญ่มากถ้ามีการรับส่งข้อมูลเรื่อย ๆ จึงต้องปรับปรุงวิธีการส่งแบบหยุดและคอยนี้เล็กน้อย โดยการใช้การส่งมอดูลัสของ SN และ RN แทน ซึ่งจะพบว่าถ้ามอดูลัสสองนี้มีความเพียงพอซึ่งจะพบได้จากตัวอย่างของการรับส่งแบบไบซิงค์

วิธีการแบบหยุดและคอยนี้ไม่ค่อยได้พบมากนักในการสื่อสารข้อมูลในปัจจุบัน เนื่องจากการที่ไม่สามารถใช้อุปกรณ์ในการสื่อสารให้มีประสิทธิภาพ ปัญหาที่สังเกตุได้คือ การสูญเปล่าระหว่างการรอคอยการตอบรับ ดังนั้นจึงได้มีการปรับปรุงวิธีการแบบหยุดและคอยนี้เพิ่มเติมก่อให้เกิดวิธีใหม่อีก 3 วิธีที่มีประสิทธิภาพดีกว่า คือ เออาร์คิวอาร์ปาเนท (ARPANET ARQ) เออาร์คิวย้อนกลับเอ็น (go back n ARQ) และเออาร์คิวเลือกการส่งกลับ (Selective Repeat ARQ)

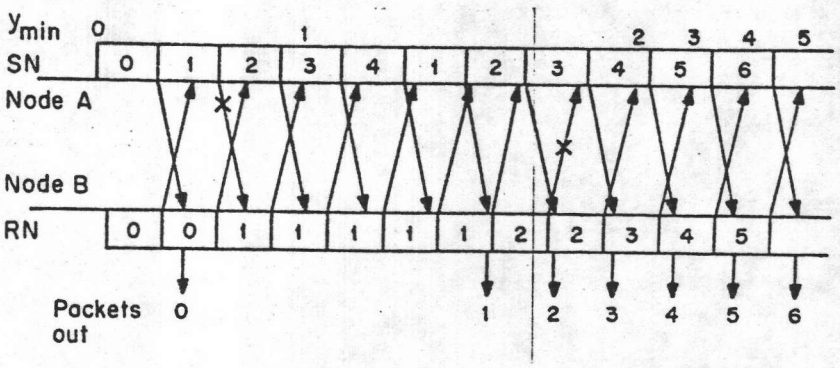
2. <u>เออาร์คิวอาร์ปาเนท</u> (ARPANET ARQ)

วิธีการของอาร์ปาเน็ทในการเพิ่มประสิทธิภาพของการรับส่งคือ การใช้วิธีแบบหยุดและคอย 8 ชุดทำงานพร้อมกันแบบขนาด โดยอาศัยการมัลติเพลกซ์ในที่นี้จะก่อให้เกิดช่องทางเสมือน (virtual channel) 8 ช่องทาง เมื่อจะรับส่งแพคเก็ทจะส่งโดยผ่านช่องทางเสมือนเหล่านี้เมื่อช่องทางเสมือนอยู่ในสภาวะที่ว่างงาน (Idle) ในกรณีที่ช่องทางเสมือนทุกช่องทางอยู่ในสภาวะทำงานทั้งหมดแพคเก็ทที่จะส่งจะต้องรอคอยอยู่โดยไม่ได้เข้ามาในส่วนของดีแอลซีดังรูปที่ 2.11 แต่ละช่องทางเสมือนที่อยู่ในสภาวะทำงานจะถูกมัลติเพลกซ์เข้ากับบิตไปพ์ (bit pipe) โดยแต่ละเฟรมของแต่ละช่องทางเสมือนจะถูกส่งทีละเฟรมต่อกันไป ลำดับของการส่งสำหรับอาร์ปาเนทไม่ใช่สิ่งสำคัญในการส่งจึงนิยมใช้วิธีแบบราวด์โรบิน (Round robin) ในกรณีที่วนมาถึงช่องทางเสมือนที่ส่งข้อมูลไปแล้วแต่ยังไม่ได้รับการตอบรับจะทำการส่งข้อมูลสำหรับแพคเก็ทนั้นอีกครั้งหนึ่ง สำหรับช่องทางเสมือนที่ได้รับการตอบรับแล้วจะกลับสู่สภาพที่ว่างงานอีกครั้งหนึ่งและพร้อมที่จะรับแพคเก็ทใหม่จากเลเยอร์อื่น

เวลาสูญเปล่า (overhead) ที่เพิ่มขึ้นจากโปรโตคอลแบบหยุดและคอยคือ แต่ละเฟรมจะต้องมีข้อมูลของหมายเลขของช่องทางเสมือน (ใช้ 3 บิต) และหมายเลขลำดับมอดูโล



รูปที่ 2.11 แสดงเออาร์คิวอาร์ปาเน็ท



รูปที่ 2.12 แสดงโปรโตคอลเออาร์คิวย้อนกลับ 4

สองของแพคเก็ทบนช่องทางเสมือน (ใช้ 1 บิต) ในด้านการตอบรับนั้นข้อมูลของทั้ง 8 เฟรมจะถูกตอบรับรวมมาในเฟรมเดียวกัน โดยแสดงถึงหมายเลขที่รอคอยอยู่ของแต่ละช่องทางเสมือน

คุณสมบัติที่เป็นประโยชน์ของวิธีการนี้อย่างหนึ่งคือ ข้อมูลสำหรับตอบรับจะได้รับอยู่เสมอเนื่องจากการตอบรับ 1 เฟรมจะได้รับข้อมูลสำหรับทุกช่องทางเสมือน โดยปกติแล้วการส่งข้อมูลใหม่ของวิธีนี้จะส่งเพียง 1 ครั้ง ต่อเฟรม เมื่อเกิดการผิดพลาด

คุณสมบัติที่ไม่ค่อยดีนักของวิธีนี้คือ อาร์ปาเนทไม่มีการจัดลำดับของแพคเก็ทที่ได้รับ จึงทำให้ลำดับของแพคเก็ทระหว่างดีแอลซีที่รับและส่งแตกต่างกัน ทำให้การใช้โปรโตคอลชนิดนี้ไม่เป็นที่นิยมแม้ว่าจะมีประสิทธิภาพมาก

3. <u>เออาร์คิวย้อนกลับเอ็น</u> (goback N ARQ)

วิธีย้อนกลับเอ็นนี้เป็นโปรโตคอลสำหรับเออาร์คิวที่นิยมกันมากจึงปรากฏในโปรโตคอลมาตรฐานทั่วไป เช่น เฮดดีแอลซี (HDLC) เอสดีแอลซี (SDLC) เป็นต้น แนวความคิดพื้นฐานของวิธีย้อนกลับเอ็นมีดังนี้ สมมุติว่าต้องการส่งข้อมูลจาก A ไป B แพคเก็ทที่ส่งจะถูกจัดลำดับหมายเลขอย่างต่อเนื่องกันและส่งหมายเลขส่งกับส่วนหัวของทุก ๆ เฟรม โดยมีข้อแตกต่างจากแบบหยุดและคอย คือ แพคเก็ทจะถูกส่งต่อ ๆ กันไปโดยไม่รอว่าจะมีการขอให้ส่งหรือไม่ ดีแอลซีฝ่ายรับที่ B จะส่งหมายเลขที่รอคอยกลับไปยัง A เพื่อขอแพคเก็ทหมายเลข RN และทำให้ทุกแพคเก็ทก่อน RN ถือว่าได้รับอย่างถูกต้อง

โปรโตคอลแบบย้อนกลับเอ็นนี้ N จะเป็นจำนวนแพคเก็ทที่จะส่งได้อย่างต่อเนื่องโดยไม่ต้องมีการขอให้ส่งจากอีกฝ่ายหนึ่ง ดังรูปที่ 2.12 แสดงให้เห็นกรณีของการย้อนกลับ 4

4. <u>เออาร์คิวเลือกการส่งกลับ</u> (Selective Repeat ARQ)

วิธีการนี้จะช่วยลดการส่งข้อมูลใหม่ของวิธีย้อนกลับเอ็น เนื่องจากวิธีย้อนกลับเอ็น จะต้องส่งข้อมุลใหม่ N เฟรมเมื่อเฟรมใดเฟรมหนึ่งผิดพลาด ในกรณีที่โอกาสเกิดข้อผิดพลาดมีน้อยกว่า 10^{-4} การส่งข้อมูลใหม่หลายเฟรมนี้มีผลน้อยมากต่อประสิทธิภาพของการส่ง แต่ในกรณีที่การส่งเกิดข้อผิดพลาดได้มากหรือการส่งมีช่วงเวลาประวิง (delay) มาก (เช่นการส่งโดยผ่านดาวเทียม) จำเป็นต้องเพิ่มประสิทธิภาพโดยโปรโตคอลแบบการเลือกการส่งกลับ แนวความคิดพื้นฐานคือ ให้ฝ่ายรับยอมรับแพคเก็ทที่ไม่เรียงลำดับและขอให้ส่งข้อมูลใหม่เฉพาะที่เกิดขึ้นผิดพลาดเท่านั้น โดยวิธีนี้ยังมีการใช้ค่า N ตามแบบย้อนกลับเอ็น เมื่อได้รับข้อมูลที่ถูกต้องแล้วดีแอลซีมอดูลจะต้องจัดลำดับของแพคเก็ทให้ถูกต้อง