บทที่ 4

ข่ายงานติดต่อสื่อสารของระบบดีดีพี

4.1 บทนำ

จากบทที่ 3 ได้กล่าวแล้วว่าระบบประมวลข้อมูลย่อยสามารถเชื่อมโยงติดต่อกับหน่วยประมวลผลข้อมูลส่วนกลางโดยข่ายการสื่อสาร ระบบติดต่อสื่อสาร เป็นระบบที่ใช้เคลื่อนย้ายข่าวสารที่ต้องการไปยังอีกสถานที่หนึ่งที่อยู่ห่างไกลออกไป การส่งข่าวสารนี้ทำได้โดยการส่งผ่านสัญญาณไฟฟ้า ผ่านตัวกลางต่าง ๆ เช่น คลื่นวิทยุ สายโทรศัพท์ หรือไมโครเวฟ การใช้คอมพิวเตอร์ในยุคแรก ๆ มักจะใช้คอมพิวเตอร์เครื่องเดียวทำงานให้แก่องค์การทั้งหมด ซึ่งโดยทั่ว ๆ ไปมักจะส่งงานในรูปของแบทช์โปรเซสซิง ผู้ใช้ที่อยู่ห่างไกลออกไปหรือผู้ใช้ที่อยู่คนละแห่งที่คอมพิวเตอร์ติดตั้งอยู่ไม่สามารถติดต่อกับคอมพิวเตอร์ได้โดยตรง ต่อมาเมื่อเทคโนโลยีเจริญก้าวหน้ามากขึ้นพร้อมทั้งความต้องการในการใช้คอมพิวเตอร์มากขึ้น เป็นผลให้ลักษณะการใช้คอมพิวเตอร์เริ่มเปลี่ยนแปลงไป ระบบออนไลน์ถูกนำเข้ามาใช้อย่างแพร่หลาย เช่น ระบบออนไลน์แบบเซนหรัลไลส์ ซึ่งเมื่อใช้งานไปก่อให้เกิดความไม่สอดคล้อง หรือพบปัญหาดังที่ได้กล่าวแล้ว แนวทางในการใช้คอมพิวเตอร์จึงเปลี่ยนแปลงต่อไปอีก เข้าสู่ระบบดีดีพีเพราะว่าเนื่องจากตั้งแต่ ค.ศ.1970 ราคาของคอมพิวเตอร์ถูกลงมาก การส่งข้อมูลเข้าประมวลผลยังหน่วยประมวลข้อมูลส่วนกลางจะทำให้เสียค่าใช้จ่ายในการติดต่อสื่อสาร ดังนั้นคอมพิวเตอร์จึงถูกนำไปติดตั้ง ณ.แหล่งที่ต้องการใช้คอมพิวเตอร์ แล้วส่งเพียงข้อมูลที่สรุปแล้วเข้ามายังหน่วยประมวลผลข้อมูลส่วนกลาง ทำให้ค่าใช้จ่ายในการติดต่อสื่อสารลดลงอย่างมาก เมื่อเทียบกับค่าใช้จ่ายในการติดต่อสื่อสาร เมื่อใช้คอมพิวเตอร์เครื่องใหญ่เพียงเครื่องเดียว จากที่กล่าวมาแล้วจะเห็นว่าระบบการติดต่อสื่อสารยังเป็นส่วนหนึ่งที่จำเป็นของระบบ แม้ว่าลักษณะของระบบจะถูกเปลี่ยนแปลงไปตามเทคโนโลยีก็ตาม

4.2 นิยามของระบบ

LEONARD KEINROCK [18] ได้ให้นิยามของข่ายงานติดต่อสื่อสารไว้ดังนี้

ข่ายงาน คือ การรวบรวมโนดจำนวนหนึ่ง โนดเหล่านี้จะถูกต่อเข้ากับโนดอื่น ๆ โดยการเชื่อมโยง

การเชื่อมโยง (link) คือ ช่องทางในการรับส่งข้อมูลของข่ายงาน

โนด คือ ศูนย์การติดต่อสื่อสารที่ทำหน้าที่รับ, เก็บ และส่งผ่านข่าวสาร

Andrew S.Tanenbaum [12] ได้ให้นิยามข่ายงานคอมพิวเตอร์ว่า เป็นการรวบรวมคอมพิวเตอร์ที่เป็นอิสระมาต่อเข้าด้วยกัน

Tanenbaum ได้ให้คำอธิบายเกี่ยวกับคอมพิวเตอร์ที่เป็นอิสระว่า ถ้ามีคอมพิวเตอร์เครื่องหนึ่งสามารถสั่งให้คอมพิวเตอร์อีกเครื่องหนึ่งเริ่มต้นทำงาน หยุดการทำงาน หรือควบคุมการทำงานของคอมพิวเตอร์เครื่องหลังนี้ได้ แสดงว่าคอมพิวเตอร์เครื่องหลังนี้ไม่เป็นอิสระสำหรับลักษณะของคอมพิวเตอร์ที่ต่อเข้าด้วยกัน Tanenbaum กล่าวว่า เมื่อคอมพิวเตอร์ทั้งสองเครื่องสามารถแลกเปลี่ยนข่าวสารกันได้ จากนิยามของ Tanenbaum เขาได้กล่าวว่าข่ายงานติดต่อสื่อสารของระบบดีดีพีนั้นเป็นเพียงกรณีพิเศษของข่ายงานคอมพิวเตอร์ที่เขาได้นิยามไว้

วัตถุประสงค์โดยทั่ว ๆ ไปของการใช้ข่ายงานติดต่อสื่อสารข้อแรกก็เพื่อสามารถใช้รีซอสร่วมกันหรือแบ่งกันใช้งาน ลองพิจารณาองค์กรที่มีคอมพิวเตอร์หลาย ๆ เครื่องติดตั้งอยู่ ณ. ที่ต่าง ๆ ที่ห่างไกลกัน เมื่อผู้บริหารต้องการข้อมูลต่าง ๆ ขององค์กรทั้งหมด จึงมีความจำเป็นที่ต้องต่อคอมพิวเตอร์เหล่านี้เข้าด้วยกันเพื่อแลกเปลี่ยนข่าวสารที่จำเป็น หรือส่งผ่านข่าวสารที่ต้องการใช้งานจากแห่งหนึ่งไปยังอีกแห่งหนึ่งได้ทันตามความต้องการ เมื่อต่อคอมพิวเตอร์เหล่านี้เข้าด้วยกันเป็นผลให้ทุก ๆ โปรแกรม ข้อมูลต่าง ๆ และรีซอสอื่น ๆ สามารถใช้ร่วมกันภายในข่ายงานได้โดยไม่ต้องคำนึงถึงที่อยู่จริง ๆ ของรีซอส หรือผู้ใช้ นอกจากนี้จะทำให้มีการแบ่งเบาภาระงานของแต่ละคอมพิวเตอร์ ให้กับคอมพิวเตอร์อื่น ๆ ในข่ายงานที่มีภาระงานน้อยกว่า อาจจะกล่าวได้ว่าวัตถุประสงค์ข้อนี้ก็คือ พยายามลดอิทธิพลของปัญหาในเรื่องสภาพทางภูมิศาสตร์ (tyrany of geography)

วัตถุประสงค์อีกข้อหนึ่ง ก็คือทำให้ระบบมีความเชื่อถือได้ (reliability)

มากขึ้น โดยมีแหล่งที่สามารถทำงานได้มากกว่าหนึ่งแห่ง เช่น ระบบที่มีคอมพิวเตอร์หลาย ๆ เครื่อง แต่เครื่องคอมพิวเตอร์เหล่านี้ไม่ได้ต่อเข้าด้วยกัน ถ้ามีข้อผิดพลาดของฮาร์แวร์ขึ้นที่จุดใดจุดหนึ่งผู้ใช้ที่ตำแหน่งนี้จะต้องรอให้การซ่อมเสร็จ แม้ว่าจะมีคอมพิวเตอร์อื่น ๆ ที่ว่างพอที่จะทำงานได้อยู่อีกที่แห่งหนึ่ง แต่สำหรับระบบที่มีการใช้ข่ายงานติดต่อสื่อสารข้อผิดพลาดดังกล่าวจะไม่ทำให้การทำงานหยุดชงัก เพราะผู้ใช้สามารถทำงานต่อไปโดยใช้คอมพิวเตอร์เครื่องอื่นจนกว่าคอมพิวเตอร์ ณ.ที่นั้นจะถูกซ่อมเสร็จ

4.3 ประเภทของสายในการรับส่งข้อมูล

จากที่กล่าวแล้วจะเห็นว่าข่ายสื่อสารของระบบดีดีพี จะเกิดขึ้นได้เมื่อมีการเชื่อมโยงคอมพิวเตอร์ที่อยู่ตามตำแหน่งต่าง ๆ ของระบบเข้าด้วยกัน สายในการรับส่งข้อมูลจึงเป็นส่วนประกอบสำคัญอันหนึ่งของระบบดีดีพี สำหรับในหัวข้อนี้จะไม่กล่าวถึงตัวกลางที่ใช้ส่งผ่านข้อมูล แต่จะกล่าวถึงลักษณะของสายโดยจะแบ่งลักษณะของไลน์ตามความเร็วในการส่งผ่านข้อมูล โดยทั่ว ๆ ไปแล้วสามารถแยกออกเป็น 3 ประเภท คือ

– ซับวอยซเกรด (Subvoice grade) เป็นสายที่ออกแบบสำหรับเครื่องส่งโทรเลขหรือเครื่องจักรอื่น ที่มีลักษณะคล้ายคลึงกันสำหรับไลน์ประเภทนี้ในสหรัฐอเมริกามีความเร็วในการส่งผ่านตั้งแต่ 45 – 150 บิทต่อวินาที สำหรับบางประเทศมีสายที่มีความเร็วในการส่งมากกว่าการส่งของโทรเลข แต่ยังไม่มีความเร็วในการส่งผ่านเท่ากับวอยซไลน์ ตัวอย่างเช่น ประเทศอังกฤษจะมีข่ายสื่อสารที่ชื่อว่า Datel 2oo Service ที่มีความสามารถในการส่งผ่าน 200 บิทต่อวินาที

– วอยซเกรด (Voice grade) ได้แก่สายโทรศัพท์ในปัจจุบันความสามารถในการส่งผ่านจาก 600 – 4800 บิทต่อวินาที สายโทรศัพท์แบบไดอัลอัพ (dial-up) ทุกวันนี้มีความสามารถในการส่งผ่าน 1200 หรือ 2400 บิทต่อวินาที

– ไวดแบนด์ (Wideband) เป็นสายที่มีความสามารถในการส่งผ่านสูงกว่าวอยซไลน์ สำหรับไวดแบนด์ไลน์มีความเร็วในการส่งผ่านสูงถึง 500,000 บิทต่อวินาที ในการใช้งานปกติถ้าต้องการความเร็วในการส่งผ่านสูงขึ้นอีกก็สามารถทำได้

เนื่องจากการจะเลือกใช้สายในการรับส่งข้อมูลไม่สามารถพิจารณาจากความเร็วใน

การส่งผ่านข้อมูลอย่างเดียวได้ ต้องพิจารณาองค์ประกอบอื่น ๆ ร่วมด้วย ลักษณะของสายที่จะต้องพิจารณา เมื่อมีการ เลือกใช้

– สายแบบต่อชั่วขณะ (Switched line) สายชนิดนี้ค่าใช้จ่ายจะถูก ถ้าปริมาณการใช้ไม่สูงมากนัก สายประ เภทนี้จะ เป็นสายสาธารณะซึ่งตัวอย่างก็ได้แก่ สายโทรศัพท์ ข้อดีของสายประ เภทนี้ก็คือกระจายไปอย่างกว้างขวาง สามารถติดต่อกับผู้ใช้ที่อยู่ ณ.ที่ต่าง ๆ ได้สะดวก แต่เนื่องจากเป็นสายสาธารณะอาจจะมีคลื่นรบกวน ทำให้สัญญาณสูญ เสียไป.

– สายแบบต่อตลอดเวลา (Private line) สายแบบนี้เป็นสายโทรศัพท์ที่เช่าจากชุมสายไว้ใช้งาน ค่าใช้จ่ายจะถูกกว่าถ้ามีปริมาณการใช้งานมาก ๆ

– ลักษณะของสาย เป็นแบบซิมเพกซ์ ซึ่งส่งข้อมูลได้เพียงทิศทางเดียว หรือแบบฮาฟดูเพกซ์ที่ส่งข้อมูลได้ทั้งสองทิศทาง แต่ไม่สามารถส่งข้อมูลพร้อมกันได้ หรือแบบฟูลดูเพกซ์ที่จะส่งข้อมูลพร้อมกันในสองทิศทางได้โดยปกติแล้วการใช้งานส่วนมากมักเลือกใช้แบบฮาฟดูเพกซ์ หรือฟูลดูเพกซ์

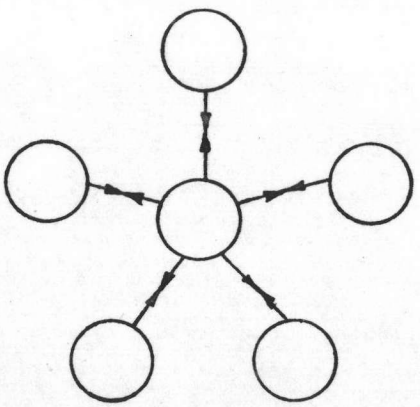
ลักษณะของสายที่แสดงให้เห็นพยายามเสนอในรูปที่จะ เป็นไปได้ ในประ เทศไทยเนื่องจากการใช้การติดต่อสื่อสารในวงการธุรกิจ หรือวงการอื่น ๆ จะต้องอยู่ภายใต้การควบคุมขององค์การสื่อสารแห่งประ เทศไทย และภายใต้กฎหมายที่ไม่ยอมให้มีการสร้างข่ายงานขององค์กรนั้น ๆ เองได้ ข่ายสื่อสารในประ เทศไทยจึงมักจะ เป็นสายโทรศัพท์เป็นส่วนใหญ่

4.4 รูปแบบของข่ายงาน

ข่ายงานติดต่อสื่อสารของระบบดีดีพี จะ เกิดขึ้นก็ต่อเมื่อมีการ เชื่อมโยงโนทต่าง ๆ ในระบบเข้าหากัน สำหรับในหัวข้อนี้จะกล่าวถึงรูปแบบหลักของข่ายงาน ซึ่งพอจะแบ่งออกเป็น 5

รูปแบบ ได้แก่ แบบดาว (Star) แบบลำดับชั้น (Heirarchy) แบบเส้นตรง (Linear) แบบวงแหวน (Ring) แบบเชื่อมโยงสมบูรณ์ (Fullyconnected) และแบบข่ายงาน (Network)

4.4.1 รูปแบบดาว สำหรับรูปแบบของข่ายงานแบบดาว จะเกิดขึ้นโดยมีหนึ่งโนดหรือมากกว่า ต่อเข้ากับมาสเตอร์โนดในลักษณะการต่อแบบพอยท์ทูพอยท์ การควบคุมการส่งผ่านของข้อมูลจะถูกจัดการโดยมาสเตอร์โนด ตัวอย่างของโครงสร้างแบบดาว ได้แก่ การต่อของเทอร์มินัลเข้ากับคอมพิวเตอร์ศูนย์กลาง โดยสายที่ใช้ส่งผ่านข้อมูล สำหรับเทอร์มินัลที่ต่ออยู่กับสายที่ใช้ส่งข้อมูล อาจจะมีหลาย ๆ เทอร์มินัล หรือเทอร์มินัลเดียวก็ได้ การส่งผ่านของข้อมูลเป็นในลักษณะพอยท์ทูพอยท์ กล่าวคือส่งข้อมูลจากเทอร์มินัลสู่คอมพิวเตอร์ หรือส่งข้อมูลจากคอมพิวเตอร์สู่เทอร์มินัล การควบคุมการส่งผ่านของข้อมูลจะจัดการโดยคอมพิวเตอร์ที่ศูนย์กลาง ซึ่งจะใช้ซอฟแวร์ที่จัดการเกี่ยวกับเทอร์มินัล รูปที่ 4.1 แสดงให้เห็นการจัดรูปแบบของข่ายงานแบบดาว

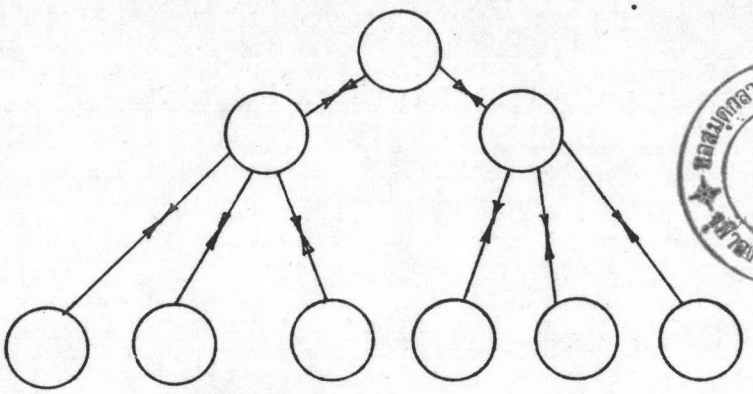


รูปที่ 4.1 แสดงการจัดรูปแบบของข่ายงานแบบดาว

รูปแบบของข่ายงานแบบดาว เป็นรูปแบบที่พบมากที่สุดในการใช้งาน แต่ละโนดที่ติดต่อกับมาสเตอร์โนท เรียกว่าแซ็ทที่ไลทโนด (Satellite node) ข่ายงานแบบดาว อาจจะกล่าวได้ว่า เป็นข่ายงานแบบลำดับชั้นที่มีเพียงชั้นเดียว ข้อดีของข่ายงานแบบดาว ก็คือสามารถทำงานได้ดี เนื่องจากได้ใช้สายในการรับส่งข้อมูลเพียงโนดเดียวทำให้ไม่ต้องใช้ร่วมกับโนดอื่น ๆ ทำให้ลดการรบกวนกันที่อาจจะ เกิดขึ้นได้ ข้อเสียของข่ายงานแบบดาว ก็คือหากเกิดข้อผิดพลาดขึ้นที่มาสเตอร์โนดจะทำให้ระบบหยุดชงัก บางครั้งอาจจะ เกิดคันเห็นชัน (contention) ขึ้นที่มาสเตอร์โนดได้ เนื่องเพราะมาสเตอร์โนดต้องรับภาระแซ็ทที่ไลทโนดหลายโนด นอกจากนี้ข่ายงานแบบดาว อาจจะเสียค่าใช้จ่ายสูงกว่าข่ายงานแบบอื่น เนื่องจากแต่ละโนดไม่ได้ใช้สายในการรับส่งข้อมูลร่วมกัน และในกรณีที่สายในการรับส่งข้อมูลเสีย จะทำให้การติดต่อกับมาสเตอร์โนดเป็นไปไม่ได้

โดยปกติความสัมพันธ์ของมาสเตอร์โนดกับแซ็ทที่ไลทโนด จะเป็นในลักษณะมาสเตอร์สเลฟ (master - slave) ในบางกรณีมาสเตอร์โนดอาจจะ เป็นเพียงสวิทชิงโนด แซ็ทที่ไลท-โนดโดยทั่ว ๆ ไปจะ เป็นมินิคอมพิวเตอร์ที่มีลักษณะเหมือนกัน ทำหน้าที่เป็นหน่วยประมวลผลทราน-แซกชัน สำหรับมาสเตอร์โนดมักจะใช้คอมพิวเตอร์ขนาดกลาง หรือคอมพิวเตอร์ขนาดใหญ่

4.4.2 แบบลำดับชั้น โครงสร้างของข่ายงานแบบลำดับชั้น บางทีเรียกว่าทรี (Tree) เป็นส่วนที่เพิ่มเติมออกมาจากข่ายงานแบบดาวซึ่งจะถูกพิจารณาว่า เป็นข่ายงานแบบลำดับชั้นที่มีระดับเดียว สำหรับข่ายงานแบบลำดับชั้นจะ เสมือนกับนำข่ายงานแบบดาว มาต่อเข้าด้วยกันหลาย ๆ ชั้นมีลักษณะคล้าย ๆ กับการแตกกิ่งก้านสาขาของต้นไม้ การควบคุมการทำงานจะถูกจัดให้อยู่ที่ศูนย์กลาง และกระจายอำนาจการควบคุมให้กับแต่ละศูนย์กลางย่อยที่ห่างไกลออกไปจากศูนย์กลาง การควบคุมการส่งผ่านข้อมูลของแต่ละโนด จะมีลักษณะ เป็นสเลฟกันตามลำดับกับโนดที่อยู่ใกล้ศูนย์กลางมากกว่า ตัวอย่างของข่ายงานแบบลำดับชั้น ได้แก่ การต่อของเทอร์มินัลเข้ากับคอมพิวเตอร์ศูนย์กลางโดยผ่านฟรอนเอนด์โปรเซสเซอร์ คอมพิวเตอร์ศูนย์กลางจะควบคุมการส่งผ่านข้อมูลของฟรอนเอนด์โปรเซสเซอร์ จากนั้นฟรอนเอนด์โปรเซสเซอร์จึงควบคุม การส่งผ่านข้อมูลของเทอร์มินัลอีกต่อหนึ่ง รูปที่ 4.2 แสดงให้เห็นลักษณะของข่ายงานแบบลำดับชั้น ซึ่งประกอบด้วยชุดของโนด โดยแต่ละโนดจะมีโนดที่อยู่ภายใต้การควบคุม



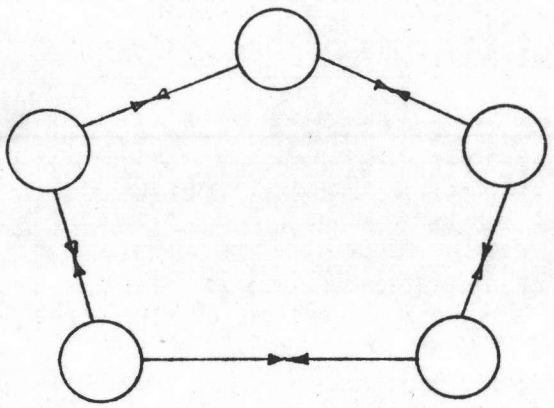
หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 4.2 แสดงข่ายงานแบบลำดับชั้น

เรียกว่า ชิลดรีนโนด (Children nodes) หลายโนด และแต่ละโนดจะมีแพรีนทโนด (Parent node) เพียงโนดเดียว ยกเว้นโนดที่เรียกว่า รูทโนด (Root node) ไม่มีแพรีนทโนด จะมีแต่ชิลดรีนโนด โนดซึ่งมีแต่แพรีนทโนดแต่ไม่มีชิลดรีนโนด เรียกว่า ลีฟโนด (Leaf node) ข้อดีของข่ายงานแบบลำดับชั้นคล้ายคลึงกับข่ายงานแบบดาว คือสามารถทำงานได้มีประสิทธิภาพดี ข้อดีของข่ายงานแบบลำดับชั้นอีกข้อหนึ่งก็คือ การประมวลผลข้อมูลจะถูกแบ่งความรับผิดชอบให้กับแพรีนทโนดในระดับที่ต่ำกว่า ดังนั้นจึงเป็นการแบ่งเบาภาระบางส่วน หรือเกือบทั้งหมดจากเซนทรัลโนด โดยทั่ว ๆ ไปลักษณะการจัดรูปแบบของแพรีนทโนดกับชิลดรีนโนดมักจะเป็นแบบมาสเตอร์สเลฟ สำหรับขนาดของโนดจะเป็นสิ่งกำหนดจำนวนระดับของลำดับชั้นของข่ายงาน

4.4.3 แบบวงแหวน รูปแบบของข่ายงานแบบที่ 3 ได้แก่ ข่ายงานแบบวงแหวน โดยที่ข่ายงานแบบนี้จะประกอบด้วยโนด ตั้งแต่ 2 โนดขึ้นไป ต่อเข้าด้วยกัน โดยที่แต่ละโนดจะต่อเข้ากับอีก 2 โนด ที่อยู่ใกล้กันแบบพอยท์ทูพอยท์ แต่ละโนดจะมีความรับผิดชอบเกี่ยวกับการควบคุมการส่งผ่านข้อมูลเท่า ๆ กัน โดยไม่ได้มีความสัมพันธ์กันแบบมาสเตอร์สเลฟ แต่ละโนดจะมีฟังก์ชันที่ทำหน้าที่ในการติดต่อสื่อสาร การส่งผ่านข้อมูลจะส่งผ่านจากโนดหนึ่งไปยังอีกโนดหนึ่ง ถ้าข้อมูลนั้นไม่ใช่ข่าวสารสำหรับโนดนี้ ข่าวสารก็จะถูกส่งต่อไปยังโนดต่อไป ตัวอย่าง ของการใช้ข่ายงานแบบวงแหวน ได้แก่ เทอร์มินัลที่ต่ออยู่กับ เทอร์มินัลคอนโทรลเลอ

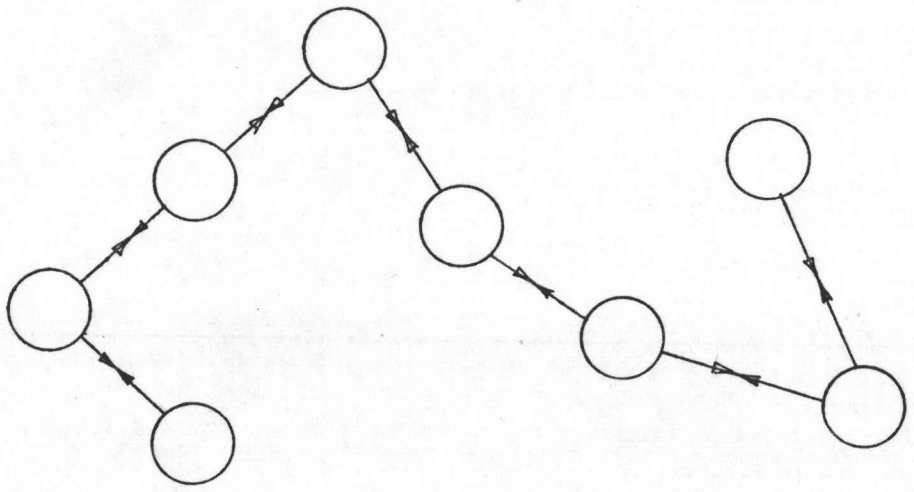
แต่ละ เทอร์มินัลจะต่อเข้ากับสายที่เป็นวงรอบ ซึ่งต่อออกมาจากคอนโทรลเลอแล้ว วกกลับเป็นวงรอบไปจบที่คอนโทรลเลออีก รูปแบบของข่ายงานแบบวงแหวน แสดงได้ตามรูปที่ 4.3 ซึ่งข่ายงานแบบวงแหวนนี้จะ เป็นรูปแบบที่ใช้งานกันมากอีกแบบหนึ่ง



รูปที่ 4.3 แสดงข่ายงานแบบวงแหวน

โนดในระบบจะถูกต่อเข้ากับอีก 2 โนดที่อยู่ใกล้ที่สุด เพื่อก่อให้เกิดวงแหวน การติดต่อสื่อสารของข่ายงานแบบวงแหวนอาจจะ เป็นแบบทิศทางเดียว หรือสองทิศทาง ข่ายงานแบบวงแหวน จะใช้ค่าใช้จ่ายน้อยกว่าข่ายงานแบบดาว เนื่องจากมีการใช้สายในการรับส่งข้อมูลร่วมกัน แต่ประสิทธิภาพในการทำงานอาจจะไม่ดีเท่าข่ายงานแบบดาว เพราะต้องส่งผ่านข้อมูลผ่านหลายโนด ก่อนจะถึงจุดหมายที่ต้องการ โดยปกติข่ายงานแบบวงแหวนมักจะมีการติดต่อสื่อสารแบบสองทิศทาง เพราะหากว่ามีการติดต่อสื่อสารแบบทิศทางเดียว หากมีโนดใดในระบบเกิดขัดข้อง จะทำให้ระบบไม่สามารถจะทำงานได้ โดยทั่ว ๆ ไปการติดต่อสื่อสารมักจะ เป็นหน้าที่ของฟรอนเอนด์โปรเซสเซอร์ หากข้อผิดพลาดที่เกิดขึ้นของโนดเกิดจากโฮสจะไม่ทำให้การติดต่อสื่อสารหยุดชงัก

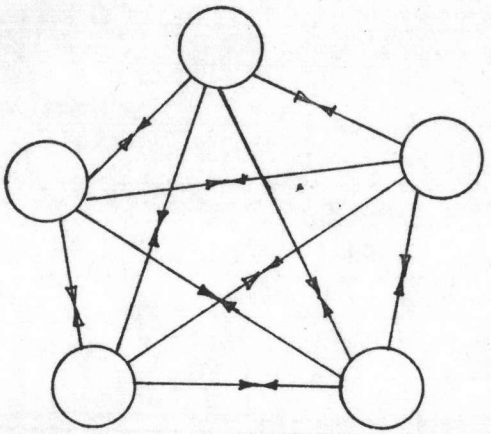
4.4.4 แบบเส้นตรง รูปแบบของข่ายงานแบบเส้นตรง แสดงได้ตามรูปที่ 4.4 เราอาจจะกล่าวได้ว่าข่ายงานแบบเส้นตรง เป็นข่ายงานแบบวงแหวนที่มีปลายเปิด



รูปที่ 4.4 แสดงข่ายงานแบบเส้นตรง

ข่ายงานแบบเส้นตรง เป็นที่รู้จักกว้างขวางมากในระบบการติดต่อสื่อสารระหว่างชาติ เพราะว่าเป็นรูปแบบที่ใช้ค่าใช้จ่ายต่ำในการเชื่อมโยงโนดที่กระจัดกระจายอยู่อย่างกว้างขวาง ถ้าหากการเชื่อมโยงของโนดไม่มีการซ้ำซ้อน ข่ายงานแบบเส้นตรงจะไม่สามารถติดต่อกันได้ หากมีข้อขัดข้องเกิดขึ้นกับสายที่ใช้รับส่งข้อมูล สำหรับในระบบที่มีจำนวนโนดมาก ๆ ข่ายงานแบบเส้นตรงจะถูกนำมาใช้ในการเชื่อมโยงกลุ่มย่อยของโนด ที่เชื่อมต่อเข้าด้วยกันแล้วในรูปแบบของข่ายงานแบบต่าง ๆ เช่น ข่ายงานแบบดาว และข่ายงานแบบวงแหวน

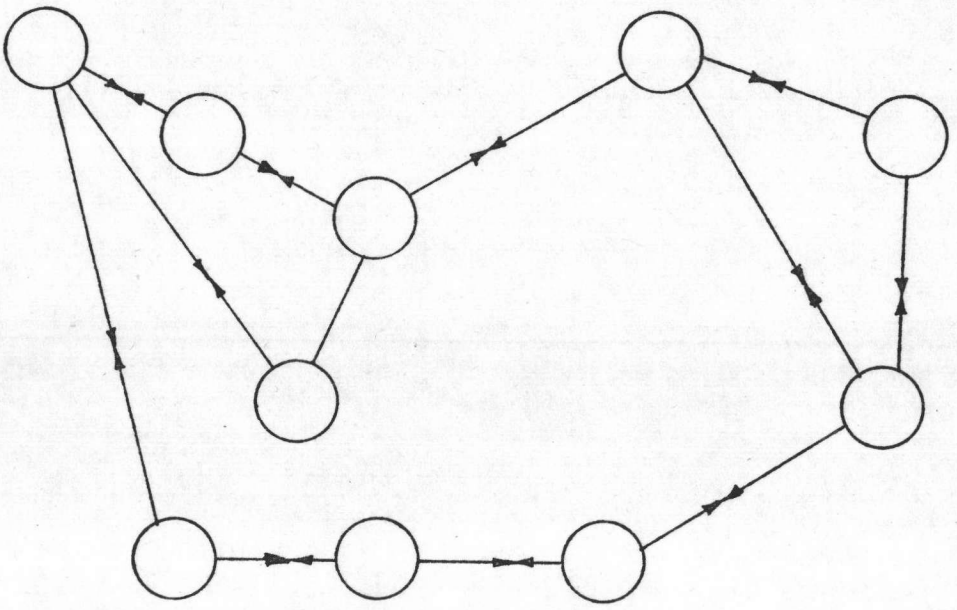
4.4.5 แบบเชื่อมโยงสมบูรณ์ ข่ายงานแบบเชื่อมโยงสมบูรณ์เกิดขึ้นจากทุก ๆ โนดเชื่อมโยงเข้าหากันได้หมด กล่าวคือแต่ละโนดจะมีสายสำหรับส่งข้อมูลเชื่อมโยงกับทุก ๆ โนดในระบบ ลักษณะของการควบคุมการส่งผ่านของข้อมูลอาจจะเป็นแบบพอยท์ทูพอยท์ หรือแบบของข่ายงานแบบวงแหวน เพราะว่าในข่ายงานแบบเชื่อมโยงสมบูรณ์จะมีการเชื่อมโยงของโนดเป็นวงรอบซึ่งเป็นลักษณะของข่ายงานแบบวงแหวน การส่งผ่านข้อมูลสามารถเลือกใช้เส้นทางได้หลายเส้นทาง รูปที่ 4.5 แสดงข่ายงานแบบเชื่อมโยงสมบูรณ์ จะเห็นว่าทุก ๆ โนดจะมีสายที่ใช้รับส่งข้อมูลต่อโดยตรงกับโนดอื่น ๆ ที่เหลือในระบบ



รูปที่ 4.5 แสดงข่ายงานแบบเชื่อมโยงสมบูรณ์

ข่ายงานแบบเชื่อมโยงสมบูรณ์มีข้อดีที่สามารถทำงานได้มีประสิทธิภาพสูง เพราะไม่มีการใช้สายสำหรับส่งข้อมูลร่วมกัน นอกจากนี้หากสายสำหรับส่งข้อมูลบางส่วนชัดข้องก็สามารถทำงานได้ต่อไป เพราะแต่ละโนดมีการเชื่อมโยงเข้าหาทุก ๆ โนดที่เหลือในระบบ ค่าใช้จ่ายของข่ายงานแบบเชื่อมโยงสมบูรณ์จะสูงมาก โดยเฉพาะอย่างยิ่งในระบบที่มีจำนวนโนดมาก เพราะว่าจำนวนของการเชื่อมโยงจะเป็น $\frac{n}{2}(n-1)$ เมื่อ n เป็นจำนวนโนด โดยปกติแล้วข่ายงานแบบเชื่อมโยงสมบูรณ์จะหากรณีที่จะใช้ยาก แต่การใช้งานจริง ๆ จะตัดการเชื่อมโยงบางส่วนเหลือแต่การเชื่อมโยงที่จำเป็น เพื่อให้ระบบมีความสามารถทำงานได้ตามต้องการ และมีค่าใช้จ่ายถูกลงบ้าง

4.4.6 แบบข่ายงาน ข่ายงานแบบข่ายงานซึ่งได้แสดงไว้ในรูปที่ 4.6 ก็คือกรณีทั่วไปในการเชื่อมโยงของโนด และรวมไปถึงข่ายงานแบบอื่น ๆ ทั้งหมดซึ่งเป็นกรณีพิเศษ โดยทั่ว ๆ ไป รูปแบบของข่ายงานนั้นไม่สามารถกำหนดได้ว่าจะเป็นรูปแบบใดได้แน่นอน เพราะรูปแบบของข่ายงานจะต้องประกอบด้วยองค์ประกอบต่าง ๆ รวมไปถึงการพิจารณาความเหมาะสมทางด้านค่าใช้จ่ายด้วย ส่วนมากในการใช้งานจึงมักจะใช้รูปแบบผสมระหว่างรูปแบบต่าง ๆ ของข่ายงานที่กล่าวมาแล้ว



รูปที่ 4.6 แสดงข่ายงานแบบข่ายงาน

4.5 หน้าที่พื้นฐานของข่ายงาน

เราได้เกี่ยวข้องกับข่ายงานสำหรับติดต่อสื่อสารระหว่างเครื่องจักรกล 2 เครื่อง ที่มีลักษณะ เป็นแบบอินเทลลิเจนท์ ซึ่งบางเครื่องอาจจะมีความสามารถสูงกว่าอีกเครื่องหนึ่ง เช่นเดียวกันกับระบบติดต่อสื่อสารของโทรศัพท์ ที่เป็นข่ายงานสำหรับการติดต่อสื่อสารระหว่างประชาชน

บุคคลที่ต้องการใช้โทรศัพท์จะต้องทราบหมายเลขโทรศัพท์ที่ต้องการจะติดต่อ เขาจะต้องมีวิธีการติดต่อไปยังเลขหมายที่ต้องการ เช่น ยกหูโทรศัพท์แล้วหมุนหมายเลขหลังจากการรอชั่วขณะหนึ่งข่ายงานก็จะบอกกับผู้ติดต่อว่าสามารถติดต่อได้ ถ้าหากไม่สามารถติดต่อได้ก็จะบอกว่าเป็นสาเหตุหนึ่งสาเหตุใดต่อไปนี้ เช่น โทรศัพท์ที่ถูกเรียกกำลังใช้งานอยู่หมุนหมายเลขผิด หรือเกิดข้อขัดข้องของอุปกรณ์ที่ใช้ติดต่อ เป็นต้น หลังจากที่วงจรไฟฟ้าถูกต่อแล้วก็จะเริ่มการติดต่อระหว่างบุคคล ขณะนี้อาจกล่าวได้ว่าบุคคลที่กำลังติดต่ออยู่นั้นมีเซ็ชชัน (Session) ต่อกัน

บทความนี้จะใช้คำว่า เซ็ทชันสำหรับเครื่องจักรกลที่กำลังติดต่อกัน สำหรับขบวนการ เมื่อมีเซ็ทชันระหว่างเครื่องจักรกลที่ติดต่อกัน 5 ขบวนการ คือ

- ขบวนการแรกก่อนจะมีการติดต่อสื่อสารกันได้ จะต้องมีการต่อสายที่จะใช้รับส่งข้อมูล เพื่อให้เป็นเส้นทางสำหรับการส่งผ่านข้อมูลระหว่างผู้ใช้ที่ปลายทางทั้งสองฝ่าย
- เมื่อสามารถต่อเส้นทางสำหรับการส่งผ่านข้อมูลแล้วก็จะ เริ่มจัดสร้างชุดของข้อมูลที่ต้องการส่งให้กับอีกฝ่ายหนึ่ง
- หลังจากเตรียมข้อมูลที่จะส่งพร้อมแล้ว ก็ส่งข้อมูลออกไปตามเส้นทางสำหรับส่งข้อมูลที่ได้ต่อไว้แล้ว
- เมื่อส่งข้อมูลที่ต้องการ และได้รับการตอบรับข้อมูลเรียบร้อยแล้ว ก็จะหยุดการส่งข้อมูล
- ขบวนการสุดท้ายก็คือ ตัดสายที่ต่อเข้าไว้เป็นเส้นทางสำหรับส่งข้อมูล เนื่องจากไม่ต้องการใช้เส้นทางเหล่านี้แล้ว

4.5.1 ระบบย่อยสำหรับส่งผ่านข้อมูล (Transport Subsystem) ในขณะที่ผู้ใช้ที่ปลายทางทั้งสองเกิดเซ็ทชันต่อกัน ผู้ใช้ที่ปลายทางจะไม่ต้องสนใจลักษณะการต่อสายที่ใช้เป็นเส้นทางในการส่งผ่านข้อมูล ลักษณะการส่งข้อมูลในรูปของบิท การป้องกันความผิดพลาดจากการส่งข้อมูล การสูญหายของข้อมูล หรือความซ้ำซ้อนของข้อมูล ซึ่งหน้าที่เหล่านี้จะ เป็นหน้าที่ของระบบย่อยสำหรับส่งผ่านข้อมูลที่จะจัดการให้ผู้ใช้ได้รับข้อมูลในลักษณะ เดียวกับผู้ใช้ที่อีกปลายทางหนึ่งต้องการส่งให้ แต่ผู้ใช้ที่ปลายทางจะ เกี่ยวข้องกับ เรื่อง เมื่อไรที่ต้องการให้มี เซ็ทชัน เมื่อไรต้องการยก เลิก เซ็ทชัน การ เข้าใจข้อมูลที่ได้รับจากผู้ใช้อีกปลายหนึ่ง และการนำข้อมูลที่ได้รับไปใช้งาน ซึ่งการทำงานเหล่านี้จะ เป็นการให้บริการโดยชุดของซอฟแวร์ที่เรียกว่า เซ็ทชัน เซอวิส (Session Service)

4.5.2 ระบบย่อยสำหรับปลายทาง (Termination Subsystem) เมื่อเทอร์มินัลส่งข่าวสารที่ไม่มีความซับซ้อนมาก จะต้องมีสิ่งอำนวยความสะดวกต่าง ๆ เช่น บัฟเฟอ (buffer)

สำหรับการประมวลผลข้อมูลแบบกระจาย จะต้องใช้สิ่งอำนวยความสะดวกหลายชนิด เพื่อให้ผู้ใช้ที่ปลายทางเข้าใจความหมายในการติดต่อกัน โดยเฉพาะอย่างยิ่งผู้ใช้ที่มีความแตกต่างกันในข่ายงาน ซึ่งสิ่งอำนวยความสะดวกเหล่านี้จะถูกจัดหาโดย ซอฟแวร์และฮาร์ดแวร์ ที่ได้กระจายไว้ ณ.ที่ตั้งต่าง ๆ วิธีที่ดีที่สุดในการใช้ ฟังค์ชันต่าง ๆ ที่ถูกกระจายไปยังโนดต่าง ๆ ในระบบก็คือ การที่จะให้โนดเหล่านั้นทำงานร่วมกัน บางโนดจะส่งข้อมูลโดยทำการ เปลี่ยนแปลงขนาดของข้อมูล เพื่อให้ข้อมูลมีขนาดเล็กลง เพื่อให้การส่งผ่านข้อมูลประหยัดค่าใช้จ่าย เมื่อส่งข้อมูลไปแล้วโนดที่รับข้อมูลก็จะ เปลี่ยนแปลงรูปร่างของข้อมูลให้กลับสู่รูปเดิมที่ส่งออกมา บางโนดก็จะจัดรูปแบบของข้อมูลที่ได้รับจากโนดอื่นให้เป็นตัวอักษร เป็นต้น โดยปกติแล้วซอฟแวร์ที่ใช้สำหรับหน้าที่ต่าง ๆ เช่น ซอฟแวร์ที่ใช้สำหรับการทำงานของคาตา เบส จะมีความซับซ้อนมาก ก่อนที่จะนำระบบดีดีพีมาใช้ ซอฟแวร์เหล่านี้ จะถูกเก็บเอาไว้ที่คอมพิวเตอร์ศูนย์กลางทั้งหมด ต่อมาจึงได้พบว่าควรจะกระจายฟังค์ชันต่าง ๆ ออกไปยังผู้ใช้ ในการที่จะใช้ซอฟแวร์ที่กระจายอยู่ตามที่ตั้งต่าง ๆ นี้ จำเป็นจะต้องแน่ใจว่าในแต่ละโนดที่จะติดต่อกันนั้นจะต้องใช้ฟังค์ชันที่เข้ากันได้ เมื่อเกิดเซ็ทชันขึ้น

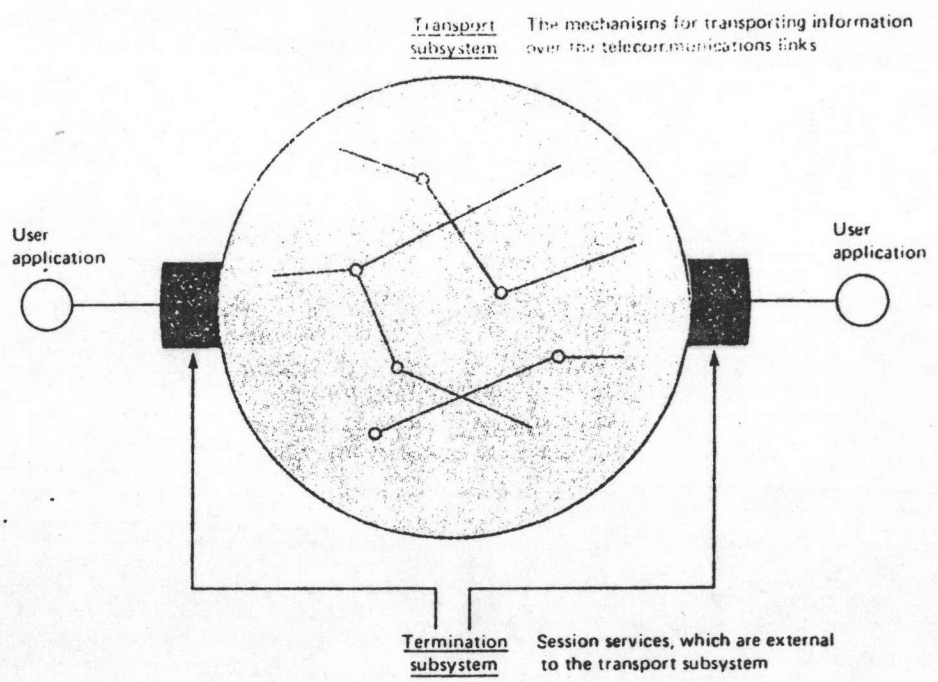
เทอร์มินัลจะถูกใช้แอกเซสซอฟแวร์ที่แตกต่างกัน ที่เวลาต่าง ๆ บางขณะ เทอร์มินัลอาจจะติดต่อกับระบบคาตา เบส หรืออาจจะติดต่อกับซอฟแวร์อื่น ๆ ลักษณะ เช่นนี้ก็คล้ายกับการติดต่อระหว่างคอมพิวเตอร์ที่จะมีซอฟแวร์ที่มีชนิดต่าง ๆ กัน และความต้องการที่แตกต่างกัน

การสนทนาระหว่างเครื่องจักรที่โปรโตคอลแตกต่างกัน จะถูกมองในลักษณะที่ว่า ใครพูดอะไร พูดเมื่อไร ในบางกรณีเครื่องจักรที่ติดต่อกันจะส่งข่าวสารสลับกัน กล่าวคือขณะที่เครื่องหนึ่งส่งผ่านข่าวสารอีกเครื่องหนึ่งจะ เป็นผู้รับ บางครั้งเครื่องจักรทั้งสองสามารถส่งข่าวสารได้พร้อม ๆ กัน หรือบางเครื่องสามารถชักจังหวะอีกเครื่องหนึ่งได้ อย่างไรก็ดีก่อนที่จะมีการแลกเปลี่ยนข่าวสาร จะต้องมีการกำหนดโปรโตคอลที่จะใช้ในการติดต่อสื่อสาร

เนื่องจากโปรโตคอล และสิ่งอำนวยความสะดวกที่ถูกนำมาใช้งานสำหรับผู้ใช้แต่ละคนจะแตกต่างกันในข่ายงาน โดยเฉพาะการติดต่อของข่ายงานต่างชนิดกันจะมีความแตกต่างระหว่างผู้ใช้มาก เมื่อเกิดเซ็ทชันขึ้นก็จะมีการนำเอาสิ่งอำนวยความสะดวกมาใช้ และจะต้องมีการตกลงเกี่ยวกับโปรโตคอล และฟังค์ชันอื่น ๆ ที่แตกต่างกันระหว่างผู้ใช้ให้เข้ากันได้ เพื่อให้ผู้ใช้

ที่ปลายทางสามารถติดต่อกันได้ และเข้าใจความหมายในการติดต่อกัน เราจะเรียกการรวบรวมฟังค์ชันที่นอกเหนือไปจากระบบย่อย สำหรับส่งผ่านข้อมูลว่า เป็นระบบย่อยสำหรับปลายทาง

ระบบย่อยสำหรับปลายทางจะประกอบด้วยซอฟแวร์ หรือบางครั้งอาจจะเป็นไปได้ที่จะเป็นฮาร์ดแวร์ หรือไมโครโคด (microcode) ซึ่งจะอยู่ในสถานที่ต่าง ๆ ซึ่งแสดงไว้ในรูปที่ 4.7



(ก)

Number	Places
1	A host computer
2	A front-end communication controller
3	A terminal controller
4	An intelligent terminal

(ข)

รูปที่ 4.7 (ก) แสดงระบบย่อยสำหรับปลายทาง และระบบย่อยสำหรับส่งผ่านข้อมูล
(ข) แสดงสถานที่ซึ่งซอฟแวร์ที่เป็นระบบย่อยสำหรับปลายทางอยู่

บางครั้งระบบย่อยสำหรับส่งผ่านข้อมูล และระบบย่อยสำหรับปลายทาง จะอยู่คนละเครื่อง จะต้องมีเครื่องจักรอีกเครื่องหนึ่งที่ทำหน้าที่เป็นอินเทอเฟส(interface)ระหว่างระบบย่อยทั้งสอง แต่ในบางกรณีส่วนของระบบย่อย ทั้งสองอยู่ในเครื่องเดียวกันโดยใช้ซอฟแวร์อินเทอเฟสระหว่างระบบย่อยทั้งสอง ระบบย่อยสำหรับปลายทางจะเป็นผู้จัดหาชุดของเซ็ทชันเซอวิส ซึ่งแต่ละชุดจะแตกต่างกันตามระบบงานที่ใช้

4.5.3 เซ็ทชันเซอวิส เป็นการให้บริการเกี่ยวกับเซ็ทชันซึ่งจะแบ่งแบบคร่าว ๆ ได้ 2 ประเภท ประเภทแรกใช้เมื่อเริ่มต้นของเซ็ทชัน ส่วนประเภทที่สองจะใช้เมื่อเซ็ทชันกำลังดำเนินไป โดยส่วนมากระบบโทรศัพท์ที่ใช้อยู่ในปัจจุบันไม่ได้ใช้เซอวิสอื่น ๆ นอกเหนือจากระบบย่อยสำหรับส่งผ่านข้อมูลเมื่อสามารถติดต่อกันได้แล้ว แต่สำหรับข่ายงานของระบบดีดีพี หรือระบบคอมพิวเตอร์ทั่ว ๆ ไปจะมีการใช้เซอวิสต่าง ๆ หลายชนิดร่วมกันนอกเหนือจากระบบย่อยสำหรับส่งผ่านข้อมูล ตัวอย่างเช่น การตรวจสอบความถูกต้องของข้อมูล , คาตา เบสเซอวิส และการตรวจสอบรหัสลับเพื่อให้เกิดความปลอดภัยของข้อมูล เซ็ทชันเซอวิสถูกใช้สำหรับการแลกเปลี่ยนของข้อมูล แต่ก่อนจะมีการแลกเปลี่ยนข้อมูล เซ็ทชันเซอวิสต้องตรวจสอบว่าทั้งสองฝ่ายที่จะติดต่อกันต้องมีคุณสมบัติ ดังนี้

– ต้องเป็นผู้ได้รับอนุญาตเท่านั้น ในขณะที่เทอร์มินัล หรือโปรแกรมใช้งานต้องการที่จะใช้คาตา เบสที่อยู่ห่างไกลออกไป ระบบแฟ้มข้อมูล หรือคอมพิวเตอร์ เนื่องจากข่ายงานยินยอมให้เทอร์มินัล หรือคอมพิวเตอร์ ณ.ที่ใดก็ได้ใช้งานสิ่งเหล่านี้ได้ ถ้าหากผู้ไม่ได้รับอนุญาตสามารถใช้งานสิ่งเหล่านี้ได้ อาจจะทำให้เสียค่าใช้จ่ายเพิ่มขึ้น อาจจะทำให้ข้อมูลไม่เป็นความลับ หรืออาจจะ เป็นผลให้เป็นการทำลายข้อมูล หรือโปรแกรม

– มีฟะซิลลิที (facility) ที่ต้องการ ฟะซิลลิทีซึ่งผู้ใช้ต้องการใช้สำหรับการประมวลผล เมื่อมีการติดต่อกันของแต่ละหน่วยของผู้ใช้จะแตกต่างกันอย่างมาก ทั้งสองฝ่ายที่ติดต่อกันอาจจะมีฮาร์ดแวร์ที่แตกต่างกัน หรือซอฟแวร์ที่แตกต่างกัน และเป็นไปได้ที่อาจจะมีการ เลือกใช้ซอฟแวร์ที่ต่างกัน ดังนั้นก่อนที่จะมีการส่งผ่านข้อมูลทั้งสองฝ่ายจะ ต้องทราบแน่ว่าฝ่ายที่จะติดต่อด้วยมีฟะซิลลิทีที่ต้องการ

– ต้องตกลงเกี่ยวกับโปรโตคอลก่อนจะ เริ่มต้นเซ็ทชัน เนื่องจากผู้ใช้แต่ละคนจะใช้โปรโตคอลในการติดต่อแตกต่างกัน

บางข่ายงานทุก ๆ โนดมีความสามารถในการ เริ่มต้นเซ็ทชัน แต่บางข่ายงานจะมีเพียงบางโนด หรืออาจจะมีเพียงโนดเดียวที่ทำได้ สำหรับข่ายงานแบบฮอริซัลทัลทุก ๆ โนดจะมีความสามารถในการทำงานเท่ากัน ยกเว้นบางโนดที่ทำหน้าที่เกี่ยวกับการบริหาร เช่น

ทำหน้าที่เก็บสถิติเกี่ยวกับการติดต่อสื่อสาร และเก็บข่าวสารเกี่ยวกับสถานะของข่ายงานสำหรับในข่ายงานแบบเวอทิคัลโนดต่าง ๆ จะมีความสามารถแตกต่างกันอย่างมาก เครื่องที่อยู่ในลำดับที่ต่ำกว่าจะมีหน่วยความจำน้อยกว่า และมีความสามารถในการคำนวณน้อยกว่าการเริ่มต้นเซ็ทชันอาจจะทำโดยเครื่องที่อยู่ในระดับที่สูงกว่า เมื่อเครื่องที่อยู่ในระดับต่ำกว่าต้องการจะใช้เซ็ทชัน ก็จะต้องติดต่อกับเครื่องจักรที่มีหน้าที่รับผิดชอบในการต่อเซ็ทชัน

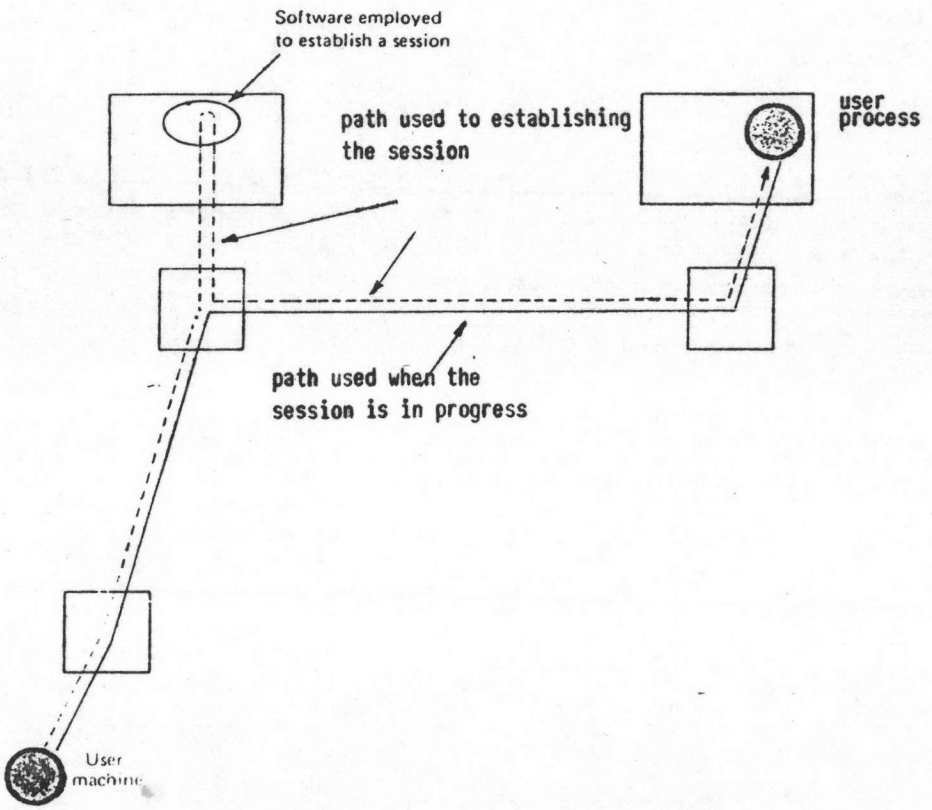
พิจารณา เมื่อเทอร์มินัลต้องการให้เกิดเซ็ทชันกับคอมพิวเตอร์ที่อยู่ห่างไกลออกไป เทอร์มินัลจะติดต่อกับโลคอลคอมพิวเตอร์ เพื่อร้องขอให้เริ่มต้นเซ็ทชันกับคอมพิวเตอร์ที่ห่างไกลออกไป ซอฟแวร์ที่มีหน้าที่ในการสร้างเซ็ทชันของคอมพิวเตอร์เครื่องแรกก็จะจัดการสร้างเซ็ทชันกับคอมพิวเตอร์เครื่องต่อไป หลังจากนั้นคอมพิวเตอร์เครื่องแรกก็จะแยกตัวออกปล่อยให้การส่งผ่านข้อมูลดำเนินต่อไปดังรูปที่ 4.8 (ก)

บางครั้งเครื่องจักรที่สร้างเซ็ทชันจะเป็นเครื่องเดียวกับเครื่องที่เริ่มต้นเลือกเส้นทางในการส่งผ่านข้อมูล สำหรับกรณีที่เกิดขึ้นตามรูป 4.8 (ก) นั้นมักจะพบเมื่อเส้นทางในการส่งผ่านข้อมูลซับซ้อน สำหรับในรูปที่ 4.8 (ข) โนดที่แลเงาจะเป็นผู้หาเส้นทางสำหรับส่งผ่านข้อมูล

เป็นไปได้ที่จะให้ฟรอนเอนด์โปรเซสเซอร์ ทำหน้าที่เกี่ยวกับการสร้างเซ็ทชันเพื่อให้โฮสคอมพิวเตอร์เป็นอิสระกับซอฟแวร์ที่ใช้เกี่ยวกับข่ายงาน หากโฮสมีข้อขัดข้องข่ายงานก็ยังสามารถดำเนินงานต่อไปได้

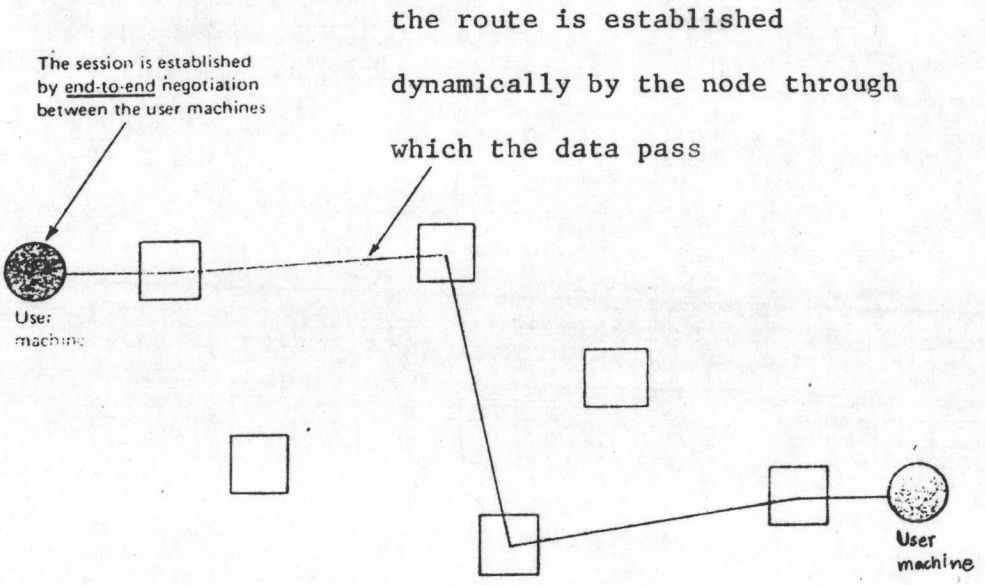
เราได้กล่าวถึงคำว่าผู้ใช้ของข่ายงานบ่อย ๆ คำว่าผู้ใช้เราหมายถึงจุดเริ่มต้นหรือจุดหมายปลายทางที่ข่าวสารถูกส่งผ่านภายในข่ายงาน ผู้ใช้ที่กล่าวถึงนี้จะเป็นผู้ใช้เซ็ทชันเซอวิส และความสามารถในการส่งผ่านข้อมูล

ก่อนที่จะเริ่มมีการนำเอาเครื่องจักรที่มีลักษณะแบบอินเทลลิเจนท์กระจายไปยังจุดต่าง ๆ คำว่า "ผู้ใช้" จะมีความหมายชัดเจนเกี่ยวกับเรื่องการติดต่อสื่อสารข้อมูล คำว่าผู้ใช้จะหมายความถึงบุคคลผู้ที่ใช้เทอร์มินัล หรืออุปกรณ์อื่น ๆ ที่คล้ายกับพิมพ์ดีด ต่อมาในปัจจุบันทั้งเทอร์มินัล และเทอร์มินัลคอนโทรลเลอร์ก็สามารถบรรจุโปรแกรมใช้งานได้ทั้งสิ้น ซึ่งโปรแกรมเหล่านี้จะกลายเป็น "ผู้ใช้" ของความสามารถในการส่งผ่านข้อมูลและเซ็ทชันเซอวิส



A third party sets up a session and establishes a route for it. After the session is established the third party drops out, but may be responsible for recovery actions.

(n)



A session set up without the assistance of a third party or manager. The route is selected dynamically by the nodes which relay the data to its destination.

(ข)

รูปที่ 4.8 แสดงการสร้างเซ็ทชัน

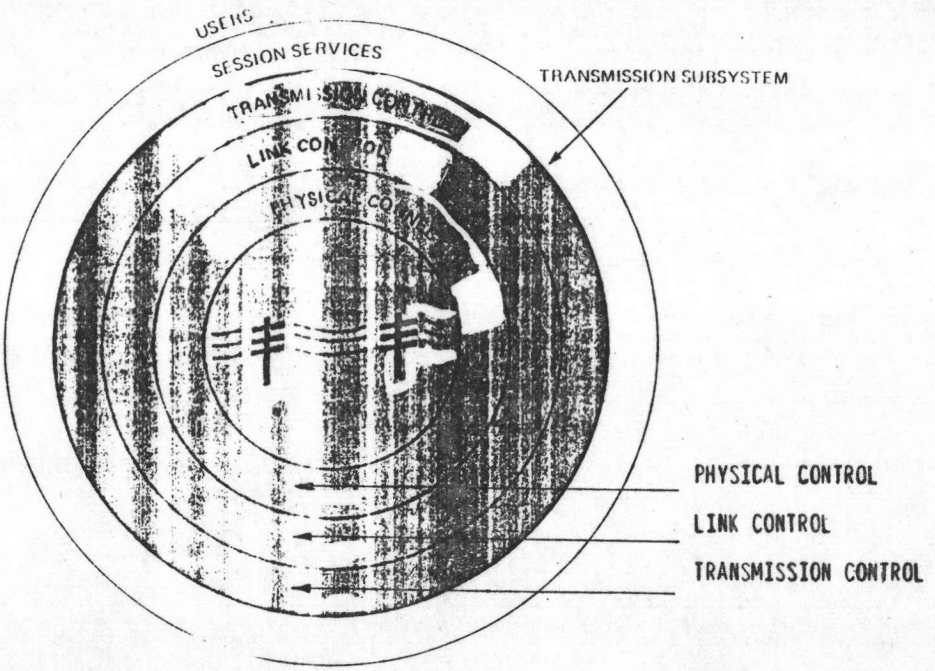
โปรแกรมใช้งานอาจจะอยู่ในโฮสคอมพิวเตอร์ หรืออุปกรณ์นำข้อมูลเข้า หรือนำข้อมูลออก ซึ่งทั้งสองกรณีโปรแกรมเหล่านี้ก็จะเป็น "ผู้ใช้" ของงาน

Number	Type of user
1	A person at a simple terminal
2	An application program in a terminal or controller
3	An application program in a host computer
4	A function built into a chip or microcode

รูปที่ 4.9 แสดงประเภทต่าง ๆ ของผู้ใช้

4.6 สถาปัตยกรรมของข่ายงาน (Network Architecture)

ระบบคอมพิวเตอร์รุ่นใหม่มักจะประกอบไปด้วยซอฟแวร์ ซึ่งสร้างขึ้นในลักษณะชั้นของผิวหนังที่ห่อหุ้มร่างกายมนุษย์ แต่ละชั้นของซอฟแวร์ที่เพิ่มขึ้นเพื่อเพิ่มความสะดวกในการใช้คอมพิวเตอร์นั้น ๆ หรือพยายามที่จะลดความยุ่งยากซับซ้อนของซอฟแวร์ที่ใช้โดยการสร้างชุดของซอฟแวร์ที่แยกจากกันเป็นชั้น ๆ โดยแต่ละชั้นจะทำหน้าที่ของตนเองอย่างเป็นอิสระ รูป 4.11 แสดงถึงการเพิ่มชั้นของซอฟแวร์ล้อมรอบซีพียู

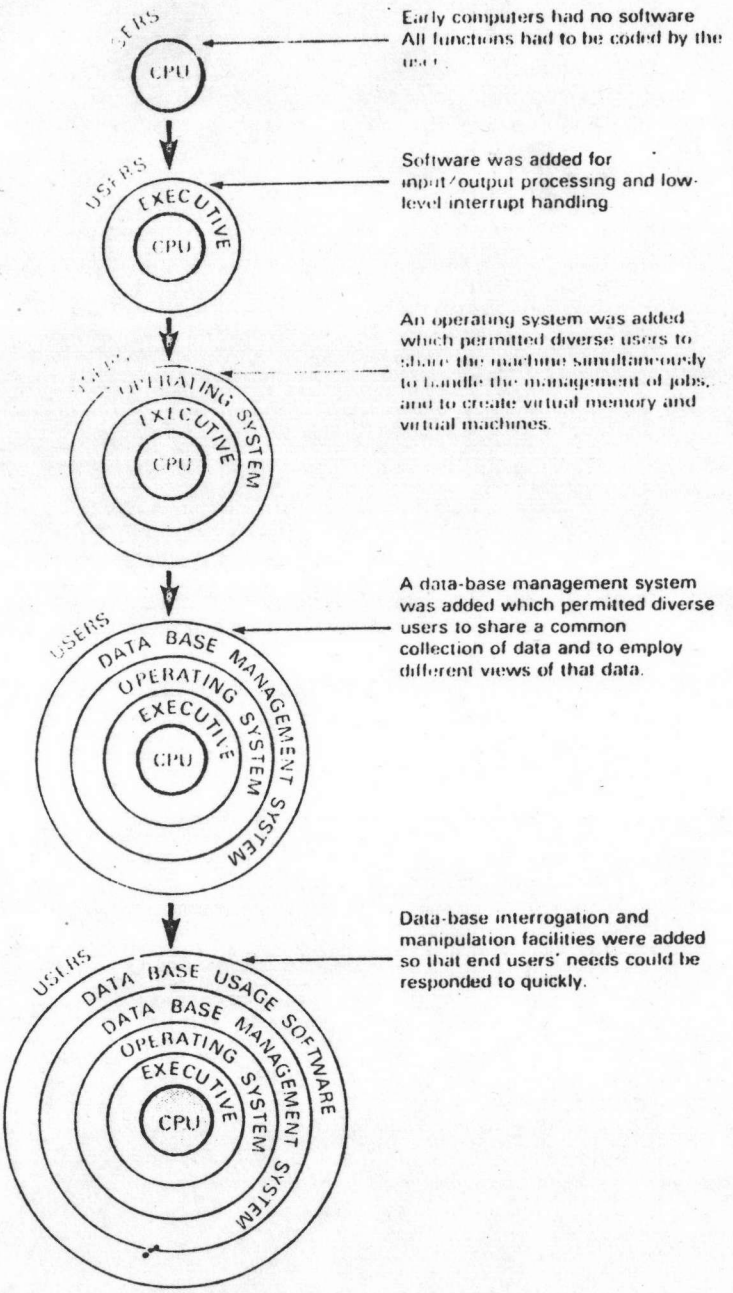


รูปที่ 4.10 แสดงแผนภาพของระบบการติดต่อสื่อสารแบบง่าย ๆ

คอมพิวเตอร์รุ่นแรกยังไม่มีซอฟแวร์ การทำงานทุก ๆ ชนิด ผู้ใช้จะต้องเขียนเป็นโปรแกรมในลักษณะภาษาเครื่อง การทำงานทุก ๆ ชนิดรวมทั้งอุปกรณ์นำข้อมูลเข้า และนำข้อมูลออกผู้ใช้ก็ต้องสั่งงานเอง ต่อมามีซอฟแวร์เพิ่มเข้ามาเพื่อจัดการเกี่ยวกับอุปกรณ์นำข้อมูลเข้า และนำข้อมูลออก ทำให้ผู้ใช้มีความสะดวกมากขึ้น ต่อมาเมื่อมีระบบควบคุมปฏิบัติงาน ทำให้ผู้ใช้หลายคนสามารถใช้งานซีพียูหลายคนพร้อม ๆ กันได้ นอกจากนี้ยังจัดการเกี่ยวกับการเข้าทำงานของแต่ละงานตามลำดับก่อนหลัง หรือความสำคัญ และรวมไปถึงการสร้างหน่วยความจำเสมือนให้กับซีพียูด้วย เมื่อมีการพัฒนาซอฟแวร์จนกระทั่งได้มีการเพิ่มดีบีเอ็มเอสซอฟแวร์ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เก็บเอาไว้ร่วมกันในรูปแบบต่าง ๆ ที่ต้องการ ขั้นสุดท้ายก็จะมีซอฟแวร์เพื่อช่วยให้การใช้ดาต้าเบสมีประสิทธิภาพมากยิ่งขึ้น สำหรับข่ายงานของระบบดีดีพีก็เช่นเดียวกัน จะมีลักษณะเป็นชั้นของซอฟแวร์ที่ทำหน้าที่ต่าง ๆ เพื่อให้ผู้ใช้ข่ายงานลดความยุ่งยากในการใช้ข่ายงาน นอกจากนี้ยังเป็นการแยกฟังค์ชันต่าง ๆ ออกเป็นส่วนเล็ก ๆ ทำให้สามารถควบคุมดูแลได้ง่าย รูปที่ 4.10 แสดงให้เห็นถึงระบบการติดต่อสื่อสารแบบง่าย ๆ เนื่องจากข่ายงานติดต่อสื่อสารประกอบด้วยชั้นต่าง ๆ ที่ทำหน้าที่ต่างกัน ซึ่งชั้นต่าง ๆ เหล่านี้จะต้องติดต่อกันด้วยกฎเกณฑ์ของแต่ละชั้นเพื่อสามารถทำความเข้าใจซึ่งกันและกันได้ กฎเกณฑ์เหล่านี้เราเรียกว่าโปรโตคอล ชุดของชั้นและโปรโตคอลเรารวมเรียกว่าสถาปัตยกรรมของข่ายงาน เนื่องจากเรามีชั้นหลายชั้นทำให้โปรโตคอลต้องมีลักษณะเป็นลำดับชั้นเช่นเดียวกัน ซึ่งลักษณะการติดต่อกันโดยใช้ลำดับชั้นของโปรโตคอลจะได้กล่าวต่อไป

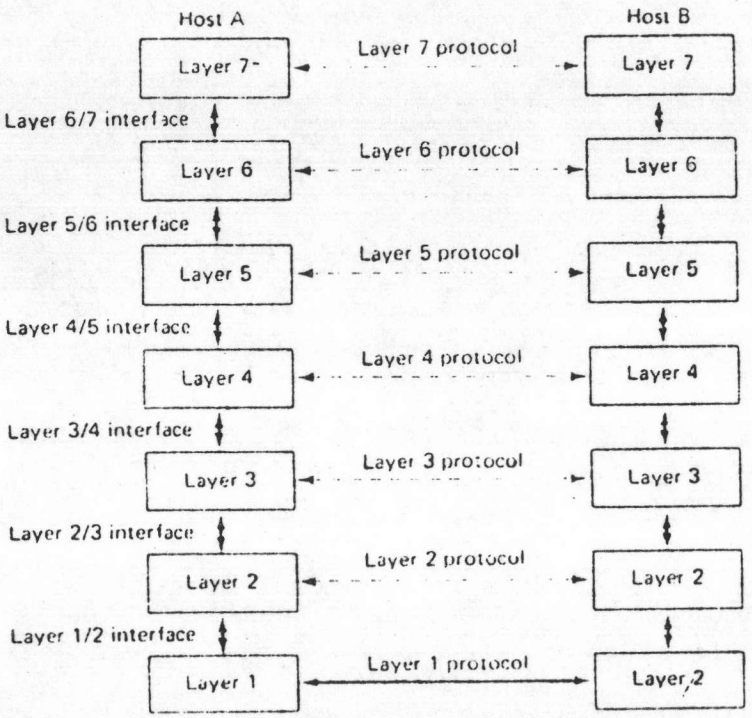
4.6.1 ลำดับชั้นของโปรโตคอล (Protocol Hierarchies) เพื่อเป็นการลดความยุ่งยากและซับซ้อนในการออกแบบข่ายงานส่วนใหญ่จะสร้างขึ้นในรูปของชั้น (layer) หรือระดับ (level)

โดยแต่ละชั้นที่สร้างขึ้นจะขึ้นกับชั้นที่ต่ำกว่าจำนวนชั้น ชื่อของแต่ละชั้น และหน้าที่ของแต่ละชั้นจะแตกต่างกันจากข่ายงานหนึ่งไปยังอีกข่ายงานหนึ่ง วัตถุประสงค์ของแต่ละชั้นก็เพื่อที่จะให้บริการกับชั้นที่อยู่สูงกว่า และไม่ประสงค์ที่จะให้ชั้นที่อยู่สูงกว่าต้องรับทราบรายละเอียดในการทำงาน แต่จะทำงานให้เมื่อมีการร้องขอ



Layers of software. Each layer that was added made the machine more capable and more useful.

รูปที่ 4.11 แสดงชั้นของซอฟแวร์ที่ล้อมรอบซีพียู



Layers, protocols, and interfaces.

รูปที่ 4.12 แสดงชั้นโปรโตคอล และอินเทอเฟส

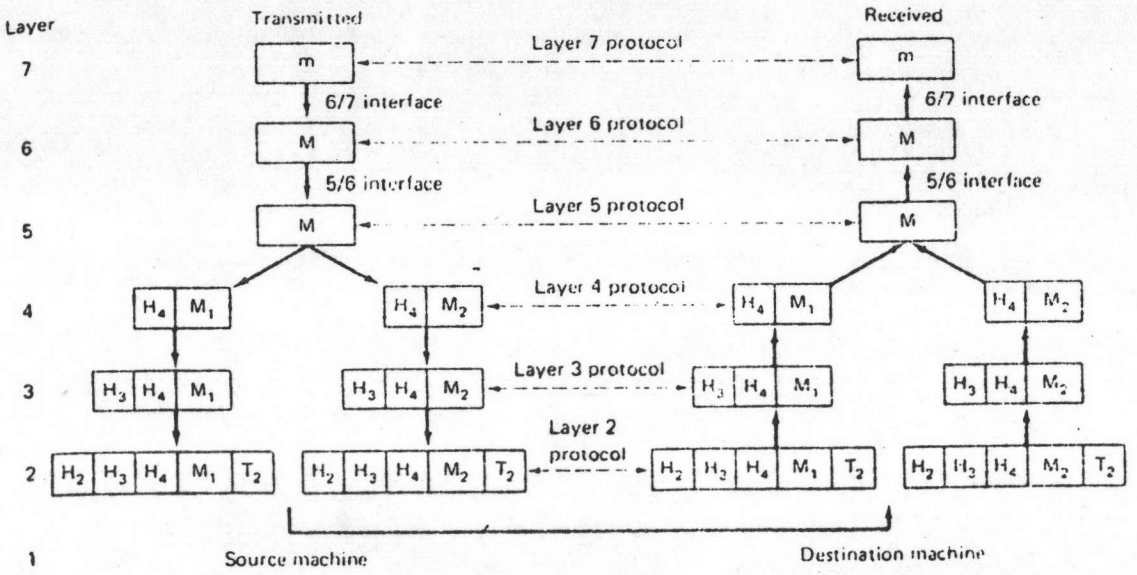
ถ้าชั้นที่ n ของเครื่องจักรหนึ่งกำลังติดต่ออยู่กับชั้นที่ n ของเครื่องจักรอีกเครื่องหนึ่ง กฎเกณฑ์ที่ใช้เพื่อบ่งบอกให้เครื่องจักรทั้งสองเข้าใจจุดมุ่งหมาย หรือข้อความที่ส่งจากอีกเครื่องจักรหนึ่งเรียกว่าโปรโตคอลของชั้นที่ n รูปที่ 4.12 แสดงให้เห็นช่ายงานที่มี 7 ชั้นของโปรโตคอล

แท้ที่จริงแล้วจะไม่มีข้อมูลใดถูกส่งผ่านโดยตรงจากชั้นที่ n ของเครื่องจักรหนึ่งไปยังชั้นที่ n ของอีกเครื่องจักรหนึ่งนอกจากจะเป็นชั้นที่ต่ำที่สุด แต่ละชั้นจะส่งข้อมูลและข่าวสารที่ใช้สำหรับการควบคุมให้กับชั้นที่อยู่ถัดลงไปข้างล่าง จนกระทั่งถึงชั้นที่ต่ำที่สุด ที่ชั้นที่ต่ำที่สุดก็จะมีการติดต่อสื่อสารแบบฟิสิคัล (Physical Communication) กับเครื่องจักรอื่นโดยการส่งสัญญาณทางไฟฟ้าไปยังอีกเครื่องจักรหนึ่ง สำหรับการติดต่อของชั้นที่อยู่สูงกว่า เราเรียกว่าเป็นการติดต่อสื่อสารแบบเวอชวล (Virtual Communication)ซึ่งแสดงในรูป 4.13 ด้วยเส้นประ ส่วนเส้นทึบแสดงถึงการติดต่อสื่อสารแบบฟิสิคัล

ระหว่างชั้นที่อยู่ติดกันจะมีอินเทอเฟส ซึ่งอินเทอเฟสนี้จะเป็นตัวกำหนดว่าการทำงานใดหรือเซอวิสใดบ้างที่ชั้นที่ต่ำกว่าต้องให้บริการกับชั้นที่อยู่สูงกว่า การกำหนดหน้าที่ของอินเทอ - เฟสให้แน่นอนนี้เป็นข้อสำคัญในการพิจารณาออกแบบช่ายงาน ซึ่งถ้าสามารถกำหนดหน้าที่ของอินเทอเฟสให้แน่นอน ก็เท่ากับกำหนดให้แต่ละชั้นทำงานตามหน้าที่ของตนเองเท่านั้น ในกรณีที่ต้องการให้ข้อมูลที่ถูกส่งผ่านระหว่างชั้นน้อยที่สุด อินเทอเฟสที่ดีจะต้องมีความสามารถตัดส่วนของข้อมูลที่ไม่จำเป็น และทำให้เป็นรูปที่ง่ายพอที่จะเปลี่ยนชั้นของซอฟแวร์ที่อยู่ต่ำกว่าแล้วไม่ทำให้เกิดผลกระทบต่อชั้นที่อยู่สูงกว่า

ตัวอย่างที่จะแสดงให้เห็นการติดต่อสื่อสารโดยชั้นของซอฟแวร์หลาย ๆ ชั้น สมมติว่ามีนักปรัชญาชาวเคนยา และชาวอินโดนีเซียต้องการที่จะติดต่อกัน เป็นการติดต่อกันในชั้นที่ 3 แต่เนื่องจากเคนยา และอินโดนีเซียไม่มีภาษากลางที่ใช้ร่วมกัน ดังนั้นต้องใช้ตัวแปลภาษาเป็นตัวติดต่อให้ ซึ่งเป็นการติดต่อกันในชั้นที่ 2 โดยแปลงภาษาเคนยาเป็นภาษาอินโดนีเซีย ซึ่งจะขึ้นกับโปรโตคอลของชั้นที่ 2 จากนั้นก็ส่งข้อมูลให้วิศวกร เพื่อให้ส่งผ่านข้อมูลไปยังวิศวกรของอีกฝ่ายหนึ่ง ตามข้อตกลงที่ให้กันไว้ หรือโปรโตคอลชั้นที่ 1 เมื่อข้อมูลถูกส่งมาแล้วก็จะส่งผ่าน แล้วแปลงเป็นภาษาอินโดนีเซีย เพื่อให้นักปรัชญาทั้งสองฝ่ายสามารถติดต่อกันได้

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



Actual information flow supporting virtual communication in layer 7.

รูปที่ 4.13 แสดงการส่งข้อมูล เมื่อเกิดการติดต่อสื่อสารแบบเวอชวลในชั้นที่ 7

จากรูปที่ 4.13 ต้องการส่งข่าวสาร m จากชั้นที่ 7 ของเครื่องจักรหนึ่งไปยังชั้นที่ 7 ของอีกเครื่องจักรหนึ่ง โดยเริ่มแรกจะส่งผ่าน 6/7 อินเทอเฟส สำหรับตัวอย่างนี้ชั้นที่ 6 จะแปลงรูปแบบของข่าวสารด้วยวิธีการใดวิธีการหนึ่ง เช่น เท็คชทคัมเพร็ชชัน (Text-Compression) แล้วส่งต่อข่าวสารใหม่ M ให้กับชั้นที่ 5 โดยผ่าน 5/6 อินเทอเฟสชั้นที่ 5 จะไม่เปลี่ยนแปลงรูปร่างของข่าวสารแต่จะทำหน้าที่ควบคุมการส่งข่าวสารให้กับชั้นที่ 6 ไม่ให้มีข่าวสารให้กับชั้นที่ 6 มากเกินไปในขณะที่ชั้นที่ 6 ยังไม่ว่าง

ข่ายงานที่ใช้ส่วนมากจะไม่มีข้อจำกัดของขนาดของข่าวสารที่ชั้นที่ 4 จะรับได้แต่จะมีข้อจำกัดในชั้นที่ 3 สำหรับตัวอย่างชั้นที่ 4 จะแบ่งข่าวสารที่ส่งเข้ามาให้เป็นหน่วยที่เล็กกว่าแล้วใส่เฮ็คเคอ (Header) ให้กับแต่ละหน่วย เฮ็คเคอจะประกอบด้วยข่าวสารที่ใช้สำหรับการควบคุม เช่น เลขลำดับของข่าวสาร เพื่อให้เครื่องจักรที่ปลายทางสามารถนำข่าวสารมาเรียงได้ตามลำดับ ถ้าชั้นที่ต่ำกว่าไม่จัดการกับลำดับของข่าวสาร ชั้นที่ 3 จะตัดสินใจว่าจะใช้ไลน์ใดส่งข่าวสารออกไป จากนั้นก็จะใส่เฮ็คเคอของชั้นที่ 3 แล้วส่งผ่านข้อมูลไปยังชั้นที่ 2 ชั้นที่ 2 นอกจากจะใส่เฮ็คเคอแล้วยังใส่เทรเลอ (Trailer) จากนั้นก็จะส่งผลลัพท์ให้กับชั้นที่ 1 เพื่อส่งผลลัพท์นั้นไปยังอีกเครื่องจักรหนึ่งโดยการติดต่อสื่อสารแบบฟิสิคัล

สิ่งที่น่าสนใจก็คือความสัมพันธ์ระหว่างการติดต่อสื่อสารแบบฟิสิคัล และการติดต่อสื่อสารแบบเวอชวล และข้อแตกต่างระหว่างโปรโตคอลและอินเทอเฟส ตัวอย่างเช่น การติดต่อสื่อสารระหว่างชั้นที่ 4 ของเครื่องจักรสองเครื่อง ความคิดเบื้องต้นก็คือ เครื่องจักรทั้งสองติดต่อกันด้วยโปรโตคอลของชั้นที่ 4 แต่ความเป็นจริงแล้วจะติดต่อกันโดยผ่านอินเทอเฟส

4.6.2 ปัญหาในการออกแบบของชั้น เราจะกล่าวถึงปัญหาที่ต้องพบในการออกแบบโปรโตคอลที่แตกต่างกันอย่างสิ้น ๆ ลุก ๆ ชั้นจะต้องมีกลไกที่จะติดต่อกันกับเครื่องจักรอื่นโดยทั่ว ๆ ไปข่ายงานจะประกอบด้วยคอมพิวเตอร์หลายเครื่อง จึงจำเป็นต้องมีฟังค์ชันที่ทำหน้าที่บอกที่อยู่ของเครื่องจักรที่ต้องการติดต่อด้วย

ชุดของปัญหาที่จะต้องตัดสินใจในการออกแบบคือ วิธีการส่งผ่านข้อมูล วิธีการส่งข้อมูลเป็นไปได้สามแบบคือ ซิมเพกซ์ ฮาฟดูเพกซ์ และฟูลดูเพกซ์ โปรโตคอลจะต้องกำหนดว่าจะใช้ล็อจิคัลแชนแนล (Logical Channel) จำนวนเท่าใด และลำดับความสำคัญ

ของแต่ละแชนแนล โดยปกติแล้วข่ายงานทั่ว ๆ ไปจะกำหนดการใช้ล็อจิคัลแชนแนลอย่างน้อย 2 แชนแนล สำหรับข้อมูลปกติ และข้อมูลที่ต้องการความรีบด่วน

การควบคุมข้อผิดพลาดที่เกิดขึ้นก็เป็นเรื่องสำคัญที่ต้องคำนึงถึง เมื่อเกิดข้อผิดพลาดเนื่องจากการติดต่อสื่อสารแบบฟิสิคัล ผู้รับจะต้องมีวิธีการที่จะบอกกับผู้ส่งว่า ข่าวสารใดบ้างที่ได้รับถูกต้องแล้ว และข่าวสารใดบ้างที่จะต้องส่งใหม่อีกครั้งหนึ่ง

บางครั้งคอมมิวนิเคชันแชนแนลไม่สามารถจัดลำดับของข่าวสารที่ถูกส่งเข้ามาได้ ดังนั้นโปรโตคอลต้องมีความสามารถที่จะให้ผู้รับรวบรวมส่วนของข่าวสารให้เรียงลำดับตามลักษณะที่ส่งออกมา วิธีแก้ปัญหาง่าย ๆ ก็คือนับส่วนของข่าวสารมีจำนวนเท่าใดในขณะที่ส่งข่าวสารออกไป แต่จะพบปัญหาอีกว่าจะทำอย่างไร ถ้าหากส่วนของข่าวสารเหล่านั้นถูกส่งเข้ามาอย่างไม่เรียงลำดับ

ปัญหาอีกปัญหาหนึ่งที่สำคัญที่จะเกิดกับทุก ๆ ชั้น ก็คือจะทำอย่างไรที่จะทำให้ผู้ส่งที่มีความเร็วในการส่งข้อมูลสูงสามารถส่งข้อมูลไปยังผู้รับข้อมูลที่มีความเร็วในการรับข้อมูลต่ำโดยไม่ทำให้ข้อมูลสูญหาย หรือทำให้การทำงานมีประสิทธิภาพลดลง อาจจะแก้ปัญหาโดยมีการส่งข้อมูลหรือสัญญาณจากผู้รับกลับมายังผู้ส่งว่า ขณะนี้ผู้รับพร้อมที่จะรับข้อมูลหรือไม่

จากที่กล่าวมาแล้วเป็นส่วนหนึ่งของปัญหาที่เกิดขึ้นในข่ายงานที่ใช้สถาปัตยกรรมแบบชั้น จากปัญหาที่ยกเป็นตัวอย่างก็จะแสดงออกถึงความซับซ้อนในการติดต่อสื่อสารของข่ายงาน แม้ว่าการใช้สถาปัตยกรรมแบบชั้นจะมีปัญหา แต่ก็สามารถจัดการได้ เพราะได้แบ่งออกเป็นส่วนย่อย ๆ ที่ทำหน้าที่ของตนเอง

ข้อดีของสถาปัตยกรรมของข่ายงานแบบชั้น

- สามารถเปลี่ยนแปลงแต่ละชั้น หรือปรับปรุงให้ดีขึ้นโดยไม่ส่งผลกระทบกับชั้นอื่น ๆ
- ทำให้การออกแบบทั้งหมดง่ายขึ้น เนื่องจากแบ่งหน้าที่ออกเป็นส่วน ๆ ที่ชัดเจนให้กับแต่ละชั้น
- สามารถแบ่งงานให้กับคณะออกแบบหลาย ๆ คณะได้ เนื่องจากแต่ละชั้นถูกกำหนดหน้าที่อย่างชัดเจน จึงสามารถแบ่งงานให้คณะทำงานหลาย ๆ คณะได้
- เครื่องจักรหลายชนิดสามารถต่อกับข่ายงานที่ชั้นต่าง ๆ ได้ ตามแต่ชนิดของ

เครื่องจักรนั้น ๆ ทำให้มีการติดต่อกับผู้ใช้ได้กว้างขวางยิ่งขึ้น

- การทำงานของชั้นที่ต่ำกว่าสามารถถูกใช้ร่วมกันโดยผู้ใช้ที่แตกต่างกันในระดับที่สูงกว่า เช่น จะมีผู้ใช้ที่ติดต่อกับชายงานในระดับที่ 4 และระดับที่ 3 ดังนั้นชั้นที่ 1 และ 2 จะถูกใช้ร่วมกันโดยผู้ใช้ทั้งสองได้
- ฟังค์ชันต่าง ๆ สามารถเปลี่ยนจากซอฟแวร์เป็นฮาร์ดแวร์หรือไมโครโคด
- การติดต่อกับเครื่องจักรสองเครื่องโดยผู้ผลิตที่แตกต่างกันสามารถทำได้ง่าย

ข้อเสียของสถาปัตยกรรมของชายงานแบบชั้น

- เนื่องจากมีชั้นต่าง ๆ มากจะทำให้การทำงานซับซ้อนขึ้น เช่น ถ้าเรามีชั้นต่าง ๆ 7 ชั้น เราจะต้องเสียเนื้อที่สำหรับใส่ข่าวสารสำหรับการควบคุมทั้ง 7 ชั้น ซึ่งอาจจะมีความยาวมากกว่าข้อมูลที่ต้องการจะส่งจริงก็ได้

4.6.3 ไอเอสโอโมเดล (Iso Model) โมเดลนี้ถูกพัฒนาโดย International Standards Organization หรือที่เรียกย่อ ๆ ว่า ISO โมเดลที่พัฒนานี้เรียกว่า Open Systems Interconnection (OSI) เพื่อให้เป็นมาตราฐานสำหรับโปรโตคอลที่แตกต่างกัน สำหรับโมเดลนี้จะมี 7 ชั้น Zimmermann [20] ได้อธิบายกฎเกณฑ์เบื้องต้นเกี่ยวกับโมเดลไว้ดังนี้

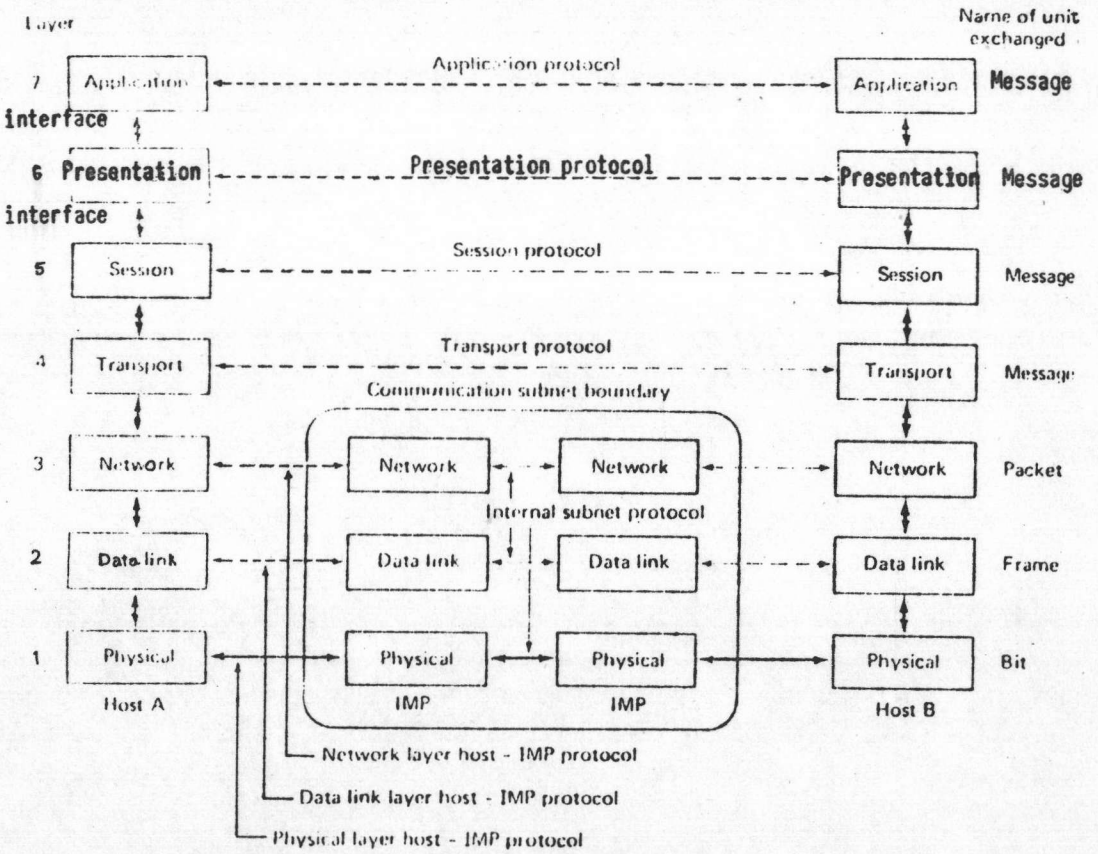
- ชั้นหนึ่งจะถูกสร้างขึ้นเมื่ออีกชั้นหนึ่งต้องการ
- แต่ละชั้นจะทำงานตามหน้าที่ ที่กำหนดเท่านั้น
- ฟังค์ชันในแต่ละชั้นจะต้องเข้าได้กับโปรโตคอลที่เป็นมาตราฐานสากล
- แต่ละชั้นจะต้องสร้างสิ่งแวดล้อมที่จะทำให้การส่งผ่านข้อมูลผ่านอินเทอเฟสน้อยที่สุด
- จำนวนของชั้นจะต้องมีมากพอที่จะทำให้ฟังค์ชันที่แตกต่างกันไม่จำเป็นต้องอยู่ในชั้นเดียวกัน และจำนวนชั้นต้องมีไม่มากเกินไปจนทำให้อาคิเท็คเชอใหญ่เกินไป

รูปที่ 4.14 แสดงสถาปัตยกรรมของชายงานที่มีจำนวนชั้น 7 ชั้น ซึ่งเป็นโมเดลของไอเอสโอ

ที่เรียกว่า โอเอสไอโมเดล โอเอสไอโมเดลจะประกอบด้วยชั้นต่าง ๆ 7 ชั้น คือ

4.6.3.1 ชั้นฟิสิคัล (Physial Layer ; Physical Control)

สำหรับชั้นนี้จะเกี่ยวข้องกับการส่งบิทของข้อมูลผ่านสายที่ใช้รับส่งข้อมูลไปยังผู้รับ ปัญหาในการออกแบบก็คือต้องให้แน่ใจว่า หากส่งบิทหนึ่งแล้วต้องได้รับบิทหนึ่งด้วย การออกแบบในชั้นนี้จะเกี่ยวข้องกับขบวนการทางไฟฟ้า และวิธีการที่จะจัดตั้งการติดต่อ การบำรุงรักษา และตัดการติดต่อของฟิสิคัลลิงค์ ตัวอย่างเช่น จะต้องใช้แรงดันกี่โวลท์ที่จะใช้แทนบิทหนึ่งและแรงดันกี่โวลท์สำหรับบิทศูนย์ การส่งผ่านข้อมูลสามารถกระทำได้ทั้งสองทิศหรือไม่ และจะทำอย่างไรบ้างเมื่อต้องการเริ่มต้นการติดต่อ และจะทำอย่างไรเมื่อติดต่อเสร็จเรียบร้อยแล้ว



รูปที่ 4.14 แสดงสถาปัตยกรรมของชายงานของไอเอสไอ

4.6.3.2 ชั้นดาต้าลิงก์ (Data Link Layer ; Data link Control) หน้าที่ของชั้นนี้ก็คือนำบิทที่ต้องการส่งมาเปลี่ยนแปลงรูปร่างให้สามารถส่งออกไปยังชั้นที่ 3 ของผู้รับโดยไม่มีข้อผิดพลาดในการส่งผ่านข้อมูล ชั้นนี้จะทำงานโดยแบ่งข้อมูลที่เข้ามาออกเป็นเฟรมข้อมูล (Data frame) หลายเฟรมจากนั้นก็ส่งผ่านเฟรมเหล่านี้ตามลำดับ จากนั้นผู้รับก็จะส่งเฟรมตอบรับ (acknowledgement frame) กลับไปให้กับผู้ส่งเพื่อแสดงว่าได้รับเฟรมข้อมูลเรียบร้อย

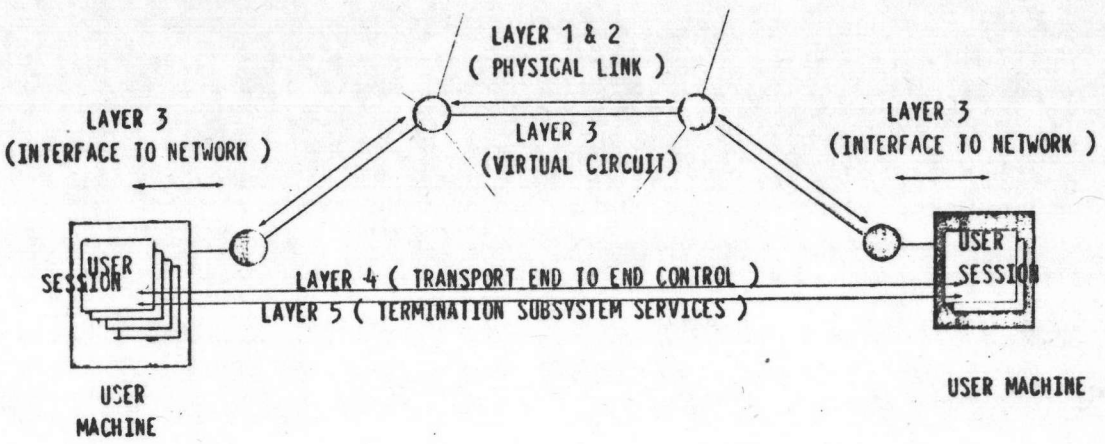
สำหรับในกรณีที่มีเสียงรบกวน (noise) ในสายอาจจะทำลายเฟรมที่ส่งไปได้ในกรณีนี้ซอฟแวร์ของชั้นที่ 2 ของเครื่องจักรที่เป็นผู้ส่งจะต้องส่งเฟรมข้อมูลใหม่ อย่างไรก็ดีการส่งเฟรมที่เหมือนกันหลาย ๆ ครั้ง อาจจะเป็นผลให้เกิดเฟรมที่ซ้ำซ้อนขึ้นได้ เนื่องจากเฟรมตอบรับถูกทำลายไป จึงทำให้เกิดเฟรมที่ซ้ำ เป็นหน้าที่ของชั้นที่ 2 เกี่ยวกับปัญหาที่เกิดจากการถูกทำลาย การสูญหาย และการซ้ำซ้อนของเฟรม ดังนั้นชั้นที่ 3 จะถูกสมมุติให้ทำงานโดยไม่มีข้อผิดพลาดของสาย ซึ่งชั้นที่ 2 จะเป็นผู้ทำให้ข้อผิดพลาดต่าง ๆ หมดไป แล้วจึงส่งข้อมูลให้กับชั้นที่ 3 ทำงาน

ปัญหาอีกปัญหาหนึ่งที่เกิดขึ้นในชั้นที่ 2 และเกิดขึ้นกับชั้นอื่นที่สูงกว่าเกือบทั้งหมดก็คือ จะทำอย่างไรที่จะทำให้ผู้ส่งที่มีความเร็วในการส่งผ่านข้อมูลสูง สามารถส่งข้อมูลให้กับผู้รับที่มีความเร็วในการรับข้อมูลต่ำโดยไม่ทำให้ข้อมูลสูญหาย บางวิธีที่นำมาใช้ก็คือการบอกให้ผู้ส่งรู้ว่าขณะนี้ผู้รับมีบัฟเฟอเหลืออยู่เท่าไรในขณะนั้น ยิ่งในกรณีที่ใช้สายในการรับส่งข้อมูลเป็นแบบฟูลดูเพกซ์จะเป็นผลให้ซอฟแวร์ที่ใช้ในชั้นที่ 2 ยุ่งยากมากขึ้นไปอีก

4.6.3.3 ชั้นเน็ทเวอร์ค (Network Layer,Network Control) ชั้นเน็ทเวอร์ค จะทำหน้าที่ควบคุมการทำงานของระบบย่อยสำหรับส่งผ่านข้อมูล ชั้นที่ 3 นี้จะเกี่ยวข้องกับวงจรเสมือน (Virtual circuit) หรือบางครั้งเรียกว่าล็อจิคัลลิงก์ ซึ่งเป็นวงจรที่ไม่มีจริงในความเป็นจริง แต่ชั้นที่ 3 จะแกล้งเสนอให้ชั้นที่สูงกว่ารู้ว่ามีจริง

รูปที่ 4.15 แสดงการติดต่อของคอมพิวเตอร์สองเครื่อง ฟิสิคัลไลน์ที่ต่ออยู่ระหว่างคอมพิวเตอร์จะใช้วิธีการสำหรับชั้นที่ 1 และชั้นที่ 2 เพื่อแลกเปลี่ยนข้อมูล ผู้ใช้ไม่ต้องการจะรู้ว่าเส้นทางที่จะส่งผ่านข้อมูลเป็นอย่างไร หรือจะต้องใช้ฟิสิคัลไลน์เป็น

จำนวนเท่าไรในการส่งผ่านข้อมูล ผู้ใช้ต้องการ เพียงการติดต่อแบบง่าย ๆ กับวงจร เสมือน ชั้นที่ 3 จะเป็นผู้สร้างวงจร เสมือน และจัดการให้ชั้นที่สูงกว่าติดต่อกับวงจร เสมือน



รูปที่ 4.15 แสดงการติดต่อของคอมพิวเตอร์สองเครื่อง

สำหรับในบางระบบ เส้นทางในการส่งผ่านข้อมูลระหว่างผู้ใช้สองคนจะ เปลี่ยนแปลงได้ เครื่องจักรที่ใช้ข่ายงานอาจจะต้องการนำข่าวสารที่จะส่งผ่านมาแบ่ง เป็นส่วน เล็ก ๆ เรียกว่า แพ็คคิท(packet) มีความยาวไม่เกินขนาดที่กำหนดไว้ แพ็คคิท เหล่านี้จะถูกรวบรวมเข้า เป็นข่าวสารอีกครั้งหนึ่งหลังจากส่งผ่านข้อมูล เหล่านี้ออกไปแล้ว ในบางข่ายงาน แพ็คคิทที่ส่งออกไปอาจจะไม่เรียงลำดับในขณะทำการส่งข้อมูล สำหรับหน้าที่ของชั้นที่ 3 ก็คือ ต้องทำให้ข่ายงานสามารถส่งแพ็คคิทให้กับผู้รับ เรียงลำดับ เช่น เดียวกับลำดับของแพ็คคิทที่ถูกส่งโดยผู้ส่ง

4.6.3.4 ชั้นแทร์นสพอท (Transport Layer;Transport Control) จากรูปที่ 4.15 แสดงให้เห็นว่าชั้นที่ 3 จะเกี่ยวข้องกับการอินเทอเฟส ระหว่างเครื่องคอม-พิวเตอร์ของผู้ใช้กับข่ายงาน สำหรับชั้นที่ 4 และชั้นที่สูงกว่าจะเกี่ยวข้องกับการทำงานของผู้ใช้ทั้งสองที่ปลายทางของข่ายงานที่มีต่อกัน ฟังค์ชันที่ทำงานในชั้นที่ 4 รวมไปถึงการควบคุมความถูกต้องและครบถ้วนของทรานแซคชันระหว่างผู้ใช้ที่ปลายทางทั้งสอง เพื่อป้องกันการสูญหายของทรานแซคชัน หรือการประมวลผลทรานแซคชันที่ซ้ำกัน ควบคุมการส่งทรานแซคชัน และค้นหาที่อยู่ของเครื่องจักรที่อยู่อีกปลายหนึ่ง สำหรับชั้นที่ 4 และชั้นที่ต่ำกว่าลงมาจะทำหน้าที่ให้บริการในการส่งผ่านข้อมูล ซึ่งเกี่ยวข้องกับการส่งผ่านกลุ่มของบิทจากผู้ใช้คนหนึ่งไปยังผู้ใช้อีกคนหนึ่งที่ปลายทางโดยไม่ได้จัดการใด ๆ กับบิทเหล่านี้ ชั้นที่อยู่สูงกว่าชั้นที่ 4 จะเป็นส่วนที่จัดการกับบิทเหล่านี้ สำหรับอินเทอเฟสของชั้นที่อยู่สูงกว่าชั้นที่ 4 ขึ้นไปก็เพื่อจะให้ผู้ใช้บริการในการส่งผ่านข้อมูลไม่ขึ้นกับชนิดของข่ายงานที่ใช้

4.6.3.5 ชั้นเซ็ทชัน (Session Layer;Session Control) การเริ่มต้นเซ็ทชันระหว่างการทำงานของผู้ใช้จะซับซ้อน เนื่องจากมีหลายทางที่เครื่องจักรจะทำงานร่วมกัน สำหรับชั้นที่ 5 จะทำหน้าที่เริ่มต้นเซ็ทชัน และสิ้นสุดเซ็ทชันในขณะที่กำลังเกิดเซ็ทชันแล้วมีสิ่งผิดพลาดเกิดขึ้น ชั้นที่ 5 ต้องเริ่มต้นเซ็ทชันใหม่ได้โดยไม่สูญเสียข้อมูล หรือในกรณีที่ไม่สามารถสิ้นสุดเซ็ทชันได้ตามปกติ ชั้นเซ็ทชันจะต้องมีฟังค์ชันสำหรับตรวจสอบข้อผิดพลาด และริคัฟเวอรีฟังค์ชัน (recovery function) เพื่อให้ระบบกลับเข้าสู่สภาพปกติก่อนจะมีข้อผิดพลาดเกิดขึ้น บางประเภทของเซ็ทชันเครื่องจักรที่ติดต่อกันทั้งสองเครื่องสามารถส่งข่าวสารได้ทั้งสองเครื่อง และโปรโตคอลจะบอกว่าใครจะส่งข่าวสารเมื่อไร และเป็นเวลานานเท่าไร บางครั้งเครื่องจักรทั้งสองจะส่งข่าวสารสลับกันไป บางครั้งเครื่องจักรหนึ่งสามารถส่งข่าวสารได้หลายข่าวสารก่อนที่อีกเครื่องจักรหนึ่งจะตอบรับ หรือบางครั้งเมื่อเกิดเซ็ทชันแล้วเครื่องจักรหนึ่งสามารถชักจูงหวะอีกเครื่องหนึ่งได้ ดังนั้นเมื่อเริ่มต้นเซ็ทชันจะต้องมีการตกลงกันเกี่ยวกับการส่งข่าวสาร

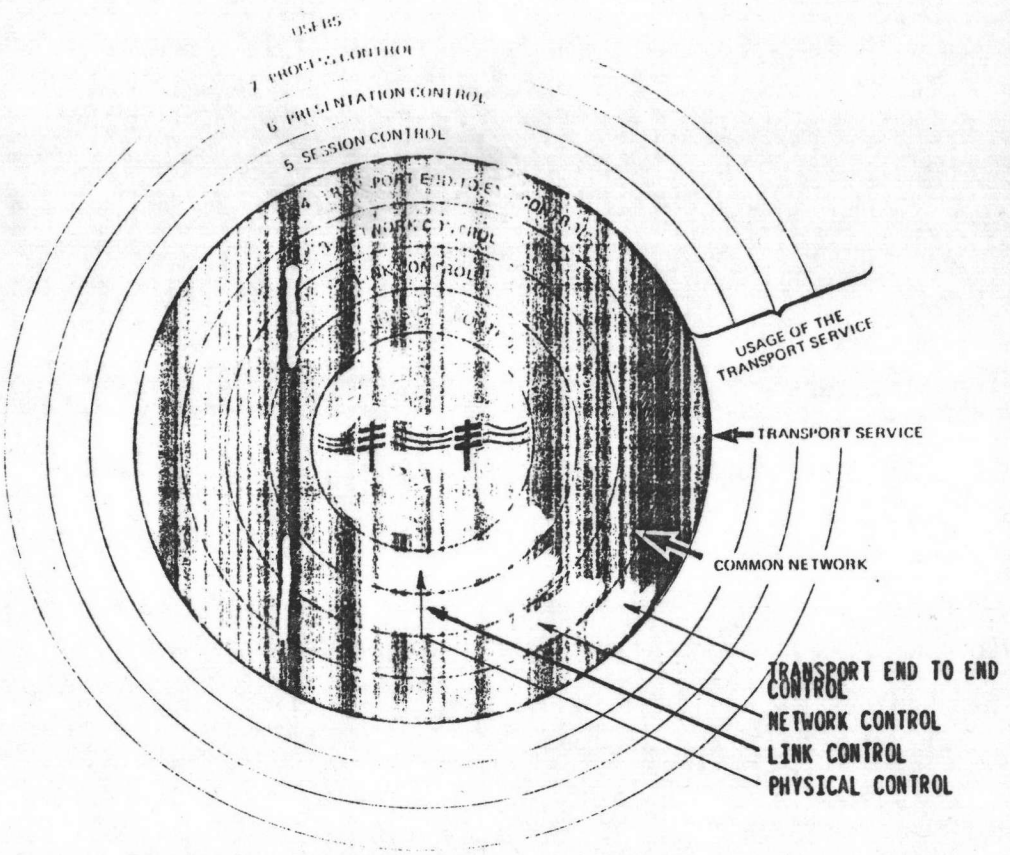
4.6.3.6 ชั้นเพร็ซเซ็นเทชัน (Presentation Layer;Presention Control) สำหรับชั้นที่ 6 นี้จะมีฟังค์ชันที่เกี่ยวกับชุดของตัวอักษร และรหัสข้อมูลที่ใช้และวิธี

การที่ข้อมูลจะถูกแสดงให้เห็นทางเครื่องพิมพ์ หรือเทอร์มินัล เมื่อตัวอักษรถูกส่งมาถึงปลายทางที่เป็นเทอร์มินัล ชุดของตัวอักษรเหล่านี้จะถูกแสดงออกให้ผู้ใช้ที่ปลายทางเห็นแล้วน่าสนใจ เช่น อาจจะมีการพิมพ์หัวข้อเรื่อง ย่อหน้า เว้นบรรทัด ข้อมูลบางชุดต้องพิมพ์ให้เห็นชัดเพื่อให้อ่านง่าย และอาจจะมีการใช้เครื่องหมายต่าง ๆ เช่น จุดภาค การแสดงผลบนจอภาพ จะมีรูปแบบต่าง ๆ เพื่อให้ผู้ใช้สามารถนำข้อมูลเข้าโดยสะดวก แต่จะส่งผ่านเฉพาะข้อมูลที่ใช้จริง ๆ เท่านั้น เลขรหัสที่ส่งมายังเทอร์มินัลจะเป็นสิ่งบอกการเลือกรูปแบบการแสดงผล โดยปกติแล้วชั้นที่ 6 จะทำหน้าที่เกี่ยวกับชุดของตัวอักษร รหัสของตัวอักษร และวิธีที่จะนำชุดของตัวอักษรเหล่านี้ไปใช้ นอกจากนี้ชั้นที่ 6 จะมีฟังก์ชันต่าง ๆ อีก เช่น การเปลี่ยนแปลงรหัสของตัวอักษร เพื่อให้อุปกรณ์ที่มีรหัสของตัวอักษรแตกต่างกันสามารถติดต่อกันได้ บางครั้งตัวอักษรที่จะถูกส่งออกไปจะถูกทำให้เป็นสายของบิทที่เล็กกว่าเพื่อให้ประหยัดค่าใช้จ่ายในการส่งผ่านข้อมูล และเพื่อให้เกิดความปลอดภัยของข้อมูลชั้นที่ 6 จะมีการใส่รหัสลับ และการถอดรหัสลับ

4.6.3.7 ชั้นแอพพลิเคชัน หรือโปรเซสคอนโทรล (Application Layer or Process Control) สำหรับชั้นที่ 7 นี้จะเป็นฟังก์ชันที่ทำหน้าที่สนับสนุนการทำงาน หรือกิจกรรมของระบบ (System activity) ตัวอย่างเช่น สนับสนุนการทำงานของผู้สั่งงานเครื่อง (operator) การใช้ข้อมูลที่อยู่ห่างไกลออกไป การควบคุมการส่งผ่านแฟ้มข้อมูล และกิจกรรมของดิสทริบิวท์ดาต้าเบส

เมื่อมีการใช้ดิสทริบิวท์ดาต้าเบสจำเป็นต้องใช้การควบคุมต่าง ๆ เพื่อป้องกันปัญหาเรื่องการปิดตายของระบบ (deadlock) ความครบถ้วน และถูกต้องของข้อมูล (integrity) ซึ่งจะต้องเป็นหน้าที่ของชั้นที่ 7 ที่จะต้องจัดการ

ตั้งแต่ชั้นที่ 4 ลงไปจะเกี่ยวข้องกับการส่งผ่านบิทของข้อมูลระหว่างผู้ใช้ที่ปลายทาง เราอาจจะเรียกได้ว่าทั้งหมดนี้เป็นระบบย่อยสำหรับส่งผ่านข้อมูล สำหรับชั้นที่ 5 , 6 และ 7 จะทำหน้าที่ให้บริการหลายชนิด ซึ่งจะถูกใช้งานโดยผู้ใช้ที่มีเซ็ทชันอยู่ เราอาจจะเรียกชุดของชั้นเหล่านี้ว่าเป็นเซ็ทชันเซอวิส สำหรับบริษัทผู้ผลิตเครื่องคอมพิวเตอร์ก็จะมีความแตกต่างในการออกแบบคอมพิวเตอร์เพื่อใช้ในระบบดีดีพี รูปที่ 4.17 แสดงการเปรียบเทียบสถาปัตยกรรมของไอเอสโอกับบริษัทผู้ผลิตต่าง ๆ



The International Standards Organization's seven layers of control for distributed processing.

รูปที่ 4.16 แสดงโมเดลของไอเอสโอ

<table>
<tr><th>Layer</th><th>ISC</th><th>ARPANET</th><th>SNA</th><th>DECNET</th></tr>
<tr><td>7</td><td>Application</td><td>User</td><td>End user</td><td rowspan="2">Application</td></tr>
<tr><td>6</td><td>Presentation</td><td>Telnet, FTP</td><td>NAU services</td></tr>
<tr><td>5</td><td>Session</td><td>(None)</td><td>Data flow control</td><td rowspan="2">(None)</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Transport</td><td>Host-host</td><td rowspan="2">Transmission control</td></tr>
<tr><td rowspan="2">Source to destination IMP</td><td rowspan="2">Network services</td></tr>
<tr><td>3</td><td>Network</td><td rowspan="2">Path control</td><td>Transport</td></tr>
<tr><td>2</td><td>Data link</td><td>IMP-IMP</td><td>Data link control</td><td>Data link control</td></tr>
<tr><td>1</td><td>Physical</td><td>Phys</td><td>Physical</td><td>Physical</td></tr>
</table>

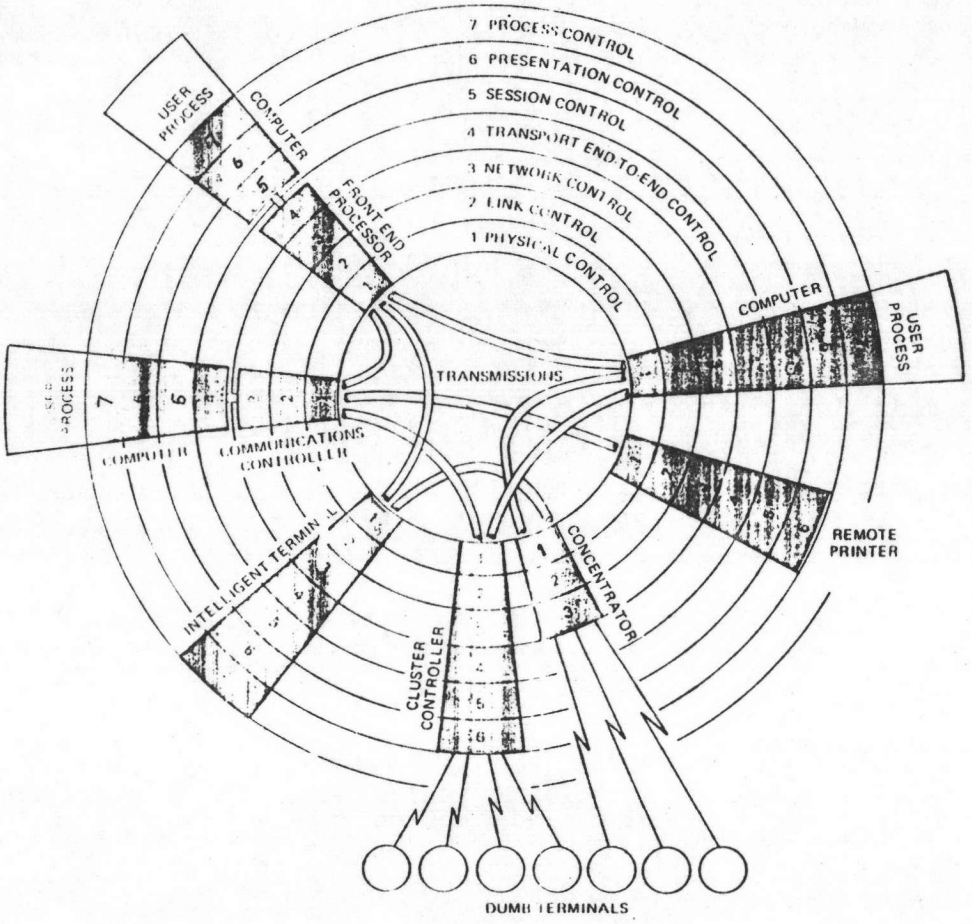
รูปที่ 4.17 แสดงการเปรียบเทียบสถาปัตยกรรมของไอเอสโอกับผู้ผลิตอื่น ๆ

สำหรับข่ายงานของระบบดีดีพี ชั้นต่าง ๆ จะถูกกระจายไปในเครื่องจักรที่แตกต่างกัน รูปที่ 4.18 แสดงประเภทของเครื่องจักรซึ่งมีหลายประเภท สำหรับหน่วยประมวลผลข้อมูลส่วนกลางมักจะถูกออกแบบให้ประกอบด้วยชั้นทั้ง 7 ซึ่งเหมือนกับคอมพิวเตอร์ที่แสดงทางซ้ายมือของรูปที่ 4.18 บางทีอาจจะดีกว่านั้นโดยมีการแยกบางชั้นออกเป็นฟรอนเอนด์โปรเซสเซอร์ ซึ่งจะทำหน้าที่จัดการกับสามชั้นแรก หรือบางครั้งอาจจะจัดการฟังก์ชันต่าง ๆ ของชั้นที่ 4 ด้วย ส่วนหน่วยประมวลผลข้อมูลย่อย อาจจะมีลักษณะคล้าย ๆ กับหน่วยประมวลผลข้อมูลส่วนกลางแต่มีความสามารถน้อยกว่า ซึ่งลักษณะของหน่วยประมวลผลนี้จะขึ้นกับงานที่จะนำไปใช้

สำหรับเทอร์มินัลที่ประกอบด้วยไมโครโปรเซสเซอร์ ก็จะมีความสามารถในการจัดการเกี่ยวกับชั้นต่าง ๆ ทั้ง 7 ชั้น ซึ่งก็คืออินเทลลิเจนท์เทอร์มินัลในรูปที่ 4.18 ซอฟแวร์ที่ใช้กับอินเทลลิเจนท์เทอร์มินัลจะซับซ้อนน้อยกว่าซอฟแวร์ที่ใช้กับคอมพิวเตอร์ เพราะว่าเทอร์มินัลจะสนับสนุนการทำงานเพียงเซ็ทชันเดียว ใช้ล็อจิคัลแชนแนลเพียงแชนแนลเดียวและประกอบด้วยฟังก์ชันที่ใช้สำหรับการจัดการเพียงไม่กี่ฟังก์ชัน ในบางกรณีเทอร์มินัลจะเป็นเพียงอุปกรณ์ที่ไม่ซับซ้อน และต่ออยู่กับคลัสเทอคอนโทรลเลอ (Cluster Controller) ซึ่งคอนโทรลเลอจะประกอบด้วยซอฟแวร์ที่ใช้สำหรับข่ายงาน ดังได้แสดงไว้ในรูป 4.18 ตอนใต้

ตัวอย่างของข่ายงานของบริษัทผู้ผลิตที่จะแสดงให้เห็น ได้แก่ ข่ายงานที่ชื่อว่าเอสเอนเอ (SNA ; System Network Architecture) ซึ่งเป็นข่ายงานของบริษัทไอบีเอ็มที่สร้างขึ้นเพื่อให้ผู้ใช้สามารถสร้างข่ายงานของตัวเองได้ เอสเอ็นเอเริ่มใช้งานมาหลายปีแล้ว เริ่มต้น เมื่อ ค.ศ.1974 เอสเอ็นเอเริ่มใช้งานโดยสามารถใช้งานกับข่ายงานแบบเซนทรัลไลส์เท่านั้น โดยรูปแบบของข่ายงานจะเป็นแบบทรี และประกอบด้วยโฮสเดียวกับเทอร์มินัล ต่อมาก็มีการเปลี่ยนแปลง จนกระทั่ง ค.ศ.1976 เอสเอ็นเอประกอบด้วยหลายโฮส แต่รูปร่างของข่ายงานยังเป็นแบบทรี การติดต่อภายในของโนดในทรี จะทำได้ระหว่างรูทของทรีเท่านั้น จนกระทั่ง ค.ศ.1979 ข้อจำกัดต่าง ๆ ลดน้อยลงทำให้การติดต่อภายในข่ายงานทำได้ดีขึ้น

ข่ายงานแบบเอสเอ็นเอประกอบด้วยการรวบรวมเครื่องจักรที่อยู่ตามตำแหน่งต่าง ๆ เรียกว่าโนด โนดในข่ายงานแบบเอสเอ็นเอมีสี่ประเภท โนดประเภทที่หนึ่งได้แก่ เทอร์มินัล โนดประเภทที่สองได้แก่ คอนโทรลเลอร์ ซึ่งทำหน้าที่ควบคุมการทำงานของเทอร์มินัล หรือ



The layers of control are allocated between machines in different ways.

รูปที่ 4.18 แสดงประเภทของเครื่องจักรที่ใช้ข่ายงาน

อุปกรณ์อื่น ๆ โนดประเภทที่สี่ ได้แก่ ฟรอนเอนด์โปรเซสเซอร์ ซึ่งมีหน้าที่แบ่งเบาภาระของซีพียูในเรื่องการจัดการเกี่ยวกับการติดต่อสื่อสารโดยผ่านข่ายงาน โนดประเภทที่ห้า ได้แก่ โฮส หรือไมโครโปรเซสเซอร์ หรือ คอนโทรลเลอร์ที่มีคุณสมบัติคล้ายโฮส สำหรับโนดประเภทที่สามไม่มีการแบ่งประเภทเอาไว้ในข่ายงานแบบเอสเอ็นเอ

แต่ละโนดจะประกอบด้วย เอ็น เอ ยู (NAU;Network Addressable Unit) หนึ่งหน่วยหรือมากกว่า เอ็นเอยูเป็นส่วนหนึ่งของซอฟแวร์ ซึ่งอนุญาตให้การประมวลผล

ของผู้ใช้สามารถใช้ข่ายงานได้ เอ็นเอยูแต่ละหน่วยจะมีที่อยู่ของตัวเองในข่ายงาน เมื่อผู้ใช้ต้องการใช้ข่ายงานและต้องติดต่อกับเอ็นเอยู เมื่อติดต่อแล้วผู้ใช้สามารถติดต่อไปยังเอ็นเอยูอื่น ๆหรือถูกติดต่อโดยเอ็นเอยูอื่นได้เช่นเดียวกัน อาจกล่าวได้ว่าเอ็นเอยูเป็นจุดเริ่มต้นที่จะเข้าสู่ข่ายงานของผู้ใช้

เอ็นเอยูสามารถแบ่งออกได้เป็นสามประเภท ประเภทแรก ได้แก่แอลยู (Lu ; Logical unit) เป็นเอ็นเอยูที่ผู้ใช้ทั่ว ๆ ไปสามารถติดต่อได้ ประเภทที่สองได้แก่พียู (PU ; Physical unit) เป็นเอ็นเอยูประเภทพิเศษซึ่งจะต่ออยู่กับโนด พียูจะถูกใช้โดยข่ายงานเพื่อทำให้โนดมีลักษณะออนไลน์ ตรวจสอบการทำงานของโนด ทำให้โนดมีลักษณะออฟไลน์ นอกจากนี้ยังทำงานในฟังก์ชันที่เกี่ยวกับการบริหาร พียูจะเตรียมการให้ข่ายงานสามารถอ้างถึงอุปกรณ์ที่ต้องการใช้จริง (Physical device)โดยไม่สนใจว่าการประมวลผลใดที่กำลังใช้อุปกรณ์เหล่านี้ ประเภทที่สามของเอ็นเอยู ได้แก่ เอสเอสซีพี (SSCP ; System Service Control Point) ซึ่งเอสเอสซีพีจะมีสำหรับโนดประเภทที่ห้าเท่านั้น เอสเอสซีพีจะมีหนึ่งหน่วยต่อโนดประเภทที่ห้าหนึ่งโนด เอสเอสซีพีจะทำหน้าที่ควบคุมฟรอนเอนด์โปรเซสเซอร์ คอนโทรลเลอ และเทอร์มินัลที่ต่อเข้ากับโฮส ทั้งซอฟแวร์และฮาร์ดแวร์ที่รวบรวมกันเข้าและถูกจัดการโดยเอสเอสซีพี เรียกว่า โดเมน (domain) รูปที่ 4.19 แสดงโดเมนอย่างง่าย ๆ สองโดเมนสำหรับข่ายงานแบบเอสเอ็นเอ

รูปที่ 4.20 แสดงลำดับชั้นของโปรโตคอลของข่ายงานแบบเอสเอ็นเอ ชั้นต่ำที่สุดทำหน้าที่เกี่ยวกับการส่งผ่านบิทของข้อมูลจากเครื่องจักรหนึ่งไปยังอีกเครื่องจักรหนึ่ง ชั้นต่อมาเรียกว่าคาตาลิงก์คอนโทรล มีหน้าที่สร้างเฟรมจากสายของบิทโดยค้นหา และริคัฟเวอร์ ข้อผิดพลาดในการส่งผ่านข้อมูล เพื่อส่งเฟรมที่ถูกต้องให้กับชั้นที่สูงกว่า ข่ายงานหลายชนิดได้ลอกแบบโปรโตคอลชั้นที่ 2 ของข่ายงานแบบเอสเอ็นเอโดยตรงและโดยอ้อม โปรโตคอลสำหรับชั้นที่ 2 ของข่ายงานแบบเอสเอ็นเอ มีชื่อเรียกว่า เอสดีแอลซี (SDLC ; Synchronous Data link Control) ชั้นที่ 3 ของข่ายงานแบบเอสเอ็นเอ เรียกว่า พาทคอนโทรล (path Control) ซึ่งจะทำหน้าที่เกี่ยวกับการหาเส้นทางในการส่งผ่านข้อมูล และการควบคุมต่าง ๆ เกี่ยวกับระบบย่อยสำหรับส่งผ่านข้อมูล นอกจากนี้ยังสามารถนำเพ็คคิท

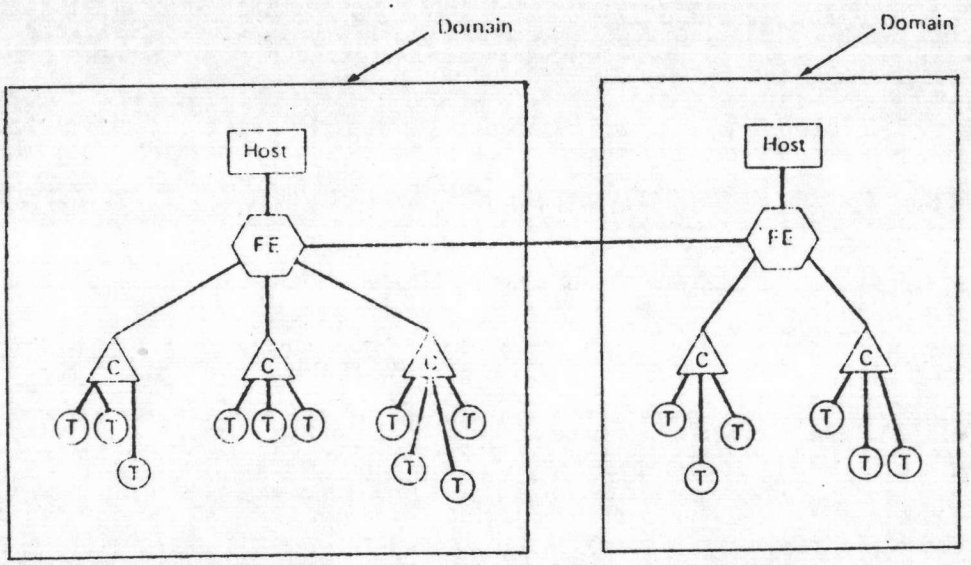
ที่ไม่มีความสัมพันธ์กันมาสร้างเป็นเฟรม เพื่อทำให้การส่งผ่านของข้อมูลมีประสิทธิภาพมากขึ้น และยังเกี่ยวข้องกับการบอกที่อยู่แบบลำดับชั้น ที่ใช้ในข่ายงานแบบเอสเอ็นเอ

ชั้นต่อมาเรียกว่า แทร็นซมิชชันคอนโทรล (transmission control) งานของชั้นนี้ได้แก่ การสร้างการติดต่อเพื่อส่งผ่านข้อมูลที่เรียกว่าเซ็ทชัน ตัดขาดการติดต่อ เมื่อส่งผ่านข้อมูลเรียบร้อยแล้ว และจัดการ การติดต่อในขณะที่กำลังติดต่อกัน ชั้นนี้จะเป็นอินเทอเฟสให้กับชั้นที่อยู่สูงกว่าอย่างสม่ำเสมอโดยไม่ขึ้นกับคุณสมบัติของระบบย่อยสำหรับส่งผ่านข้อมูล ขณะที่กำลังเกิดเซ็ทชันแทร็นซมิชชันคอนโทรลจะทำการควบคุม การส่งผ่านข้อมูลให้ผู้ส่งที่มีความเร็วในการส่งผ่านข้อมูลสูงสามารถส่งข้อมูลให้ผู้รับที่มีความเร็วในการรับข้อมูลต่ำได้ ควบคุมการแบ่งส่วนบัฟเฟอ จัดข่าวสารที่มีคุณสมบัติแตกต่างกัน จัดการเกี่ยวกับมัลทิเพล็กซิง (multiplexing) และดีมัลทิเพล็กซิง (demultiplexing) ของข้อมูลและข่าวสารสำหรับการควบคุม เพื่อให้เป็นประโยชน์กับชั้นที่สูงกว่า และหากต้องการก็สามารถใส่รหัสลับ และถอดรหัสลับได้ เพื่อความปลอดภัยของข้อมูล

ชั้นต่อมาเรียกว่า คาตาโฟลคอนโทรล (data flow Control) ทำหน้าที่ควบคุมการเคลื่อนไหวของข้อมูล คุณสมบัติที่แตกต่างไปก็คือไม่มีเอ็กเคอสำหรับชั้นนี้ เมื่อทำการติดต่อกับชั้นเดียวกันของอีกเครื่องหนึ่งโดยที่จริงแล้วข่าวสารที่จะใช้เป็นเอ็กเคอจะถูกส่งให้กับชั้นของแทร็นซมิชชันคอนโทรลในรูปของพารามีเตอร์ และบางส่วนจะรวมอยู่กับแทร็นซมิชชันเอ็กเคอ

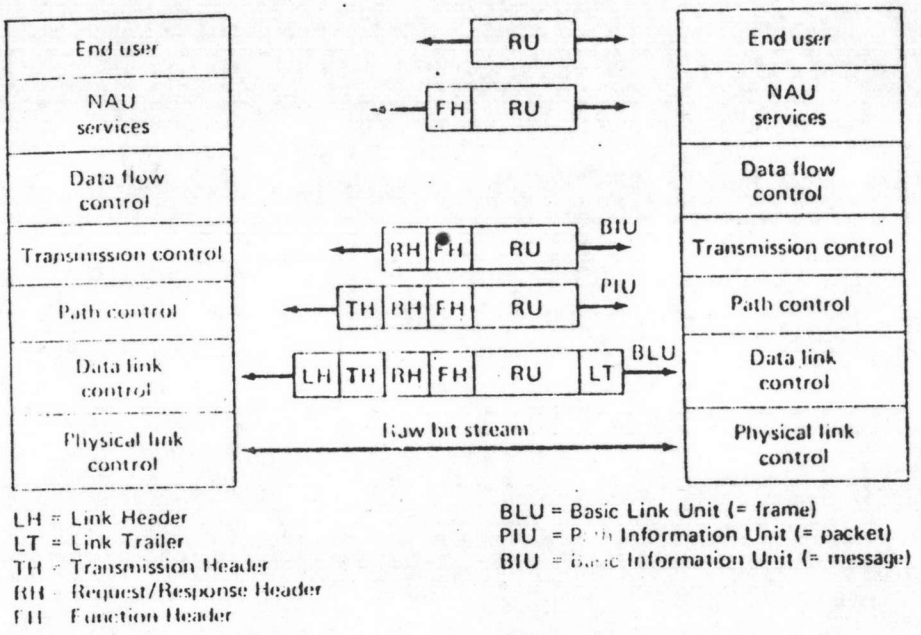
ชั้นที่ 6 ของข่ายงานแบบเอสเอ็นเอ เรียกว่า เอ็นเอยูเซอวิส เอ็นเอยูเซอวิสจะเตรียมการให้บริการสองประเภท ให้แก่การประมวลผลของผู้ใช้ ประเภทแรกเรียกว่าเพร็ซเซ็นเทชันเซอวิส ตัวอย่าง การให้บริการ ได้แก่ เท็กซทคัมเพร็ซชัน ประเภทที่สอง ได้แก่ เซ็ทชันเซอวิส มีหน้าที่ให้บริการจัดการติดต่อระหว่างผู้ใช้ บางครั้งก็จะเพิ่มเน็ทเวอร์คเซอวิส ซึ่งมีหน้าที่ให้กับการบริการเกี่ยวกับการทำงานของข่ายงานทั้งหมด

4.6.3.8 อินเทอเฟส ส่วนสำคัญของสถาปัตยกรรมแบบชั้นก็คืออินเทอเฟสระหว่างชั้น อินเทอเฟสเหล่านี้จะต้องถูกกำหนดลักษณะแน่นอน เพื่อให้เป็นมาตรฐานในการใช้งาน แต่ละชั้นในระบบดีดีพีที่ติดต่อกับชั้นเดียวกันของอีกเครื่องจักรหนึ่งที่ปลายทาง เราอาจจะคิดว่าเครื่องจักรที่กำลังติดต่อมีลักษณะคล้ายชั้นของขนมเค็ก แต่ละชั้นจะประกอบด้วยชุดของ



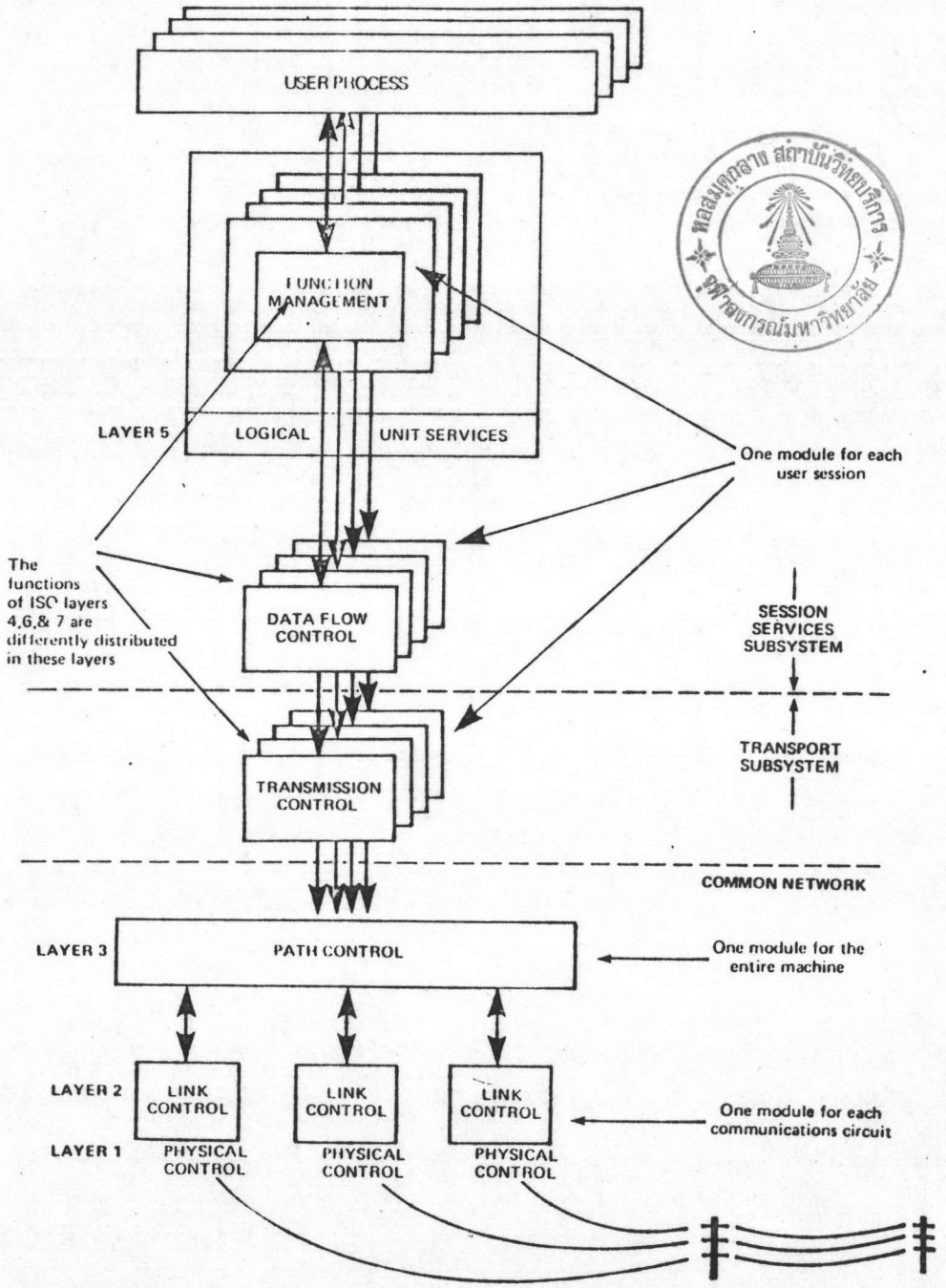
A two domain SNA network. FE = Front End, C = Controller, T = Terminal.

รูปที่ 4.19 แสดงโดเมนของข่ายงานแบบเอสเอ็นเอ

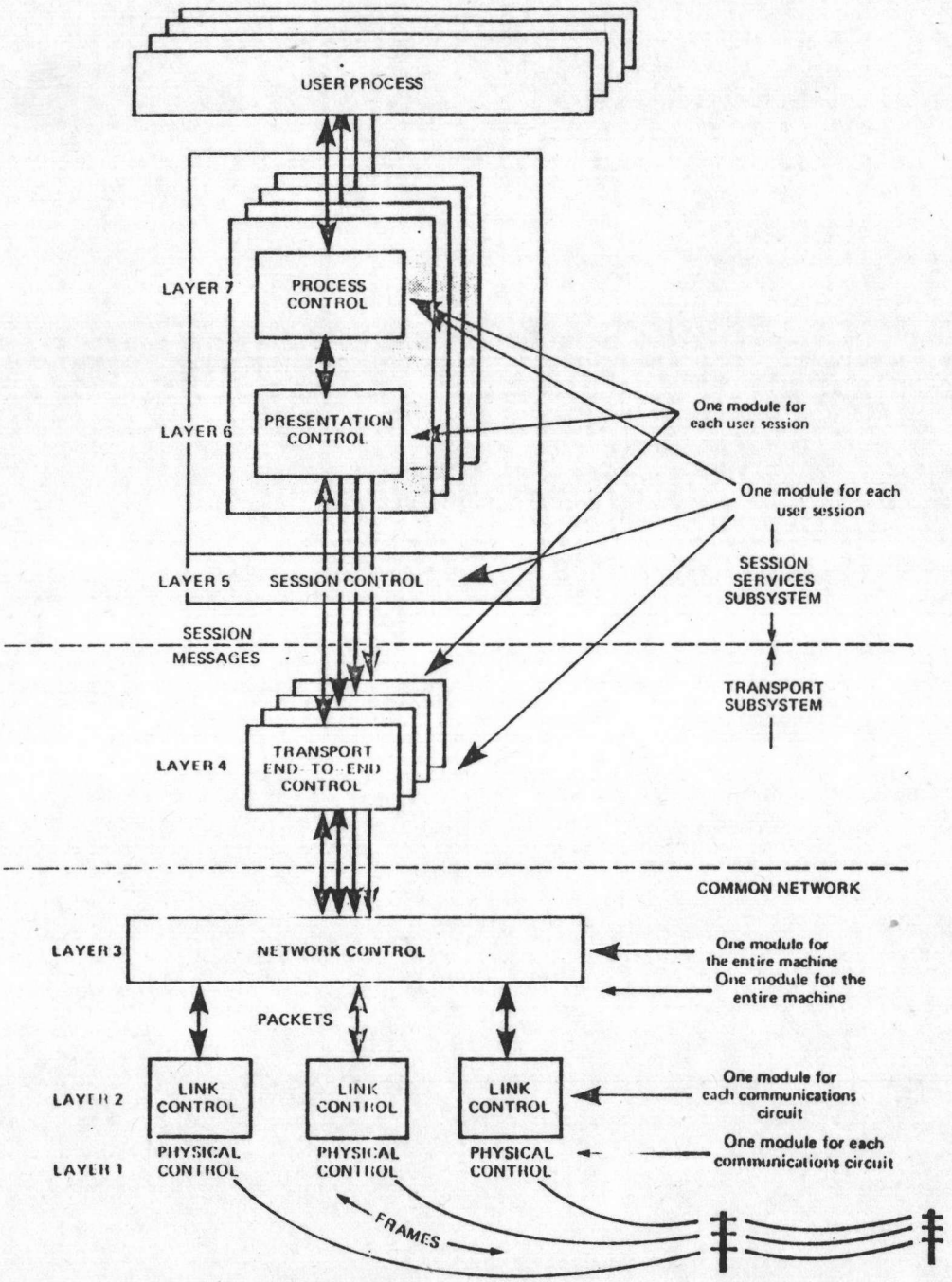


Protocol hierarchy and units exchanged in SNA.

รูปที่ 4.20 แสดงลำดับชั้นของโปรโตคอลของข่ายงานแบบ เอส เอ็น เอ



รูปที่ 4.21 แสดงโนดที่ใช้สถาปัตยกรรมของเอสเอ็นเอ



A full-function node of a distributed system using the ISO 7-layer architecture.

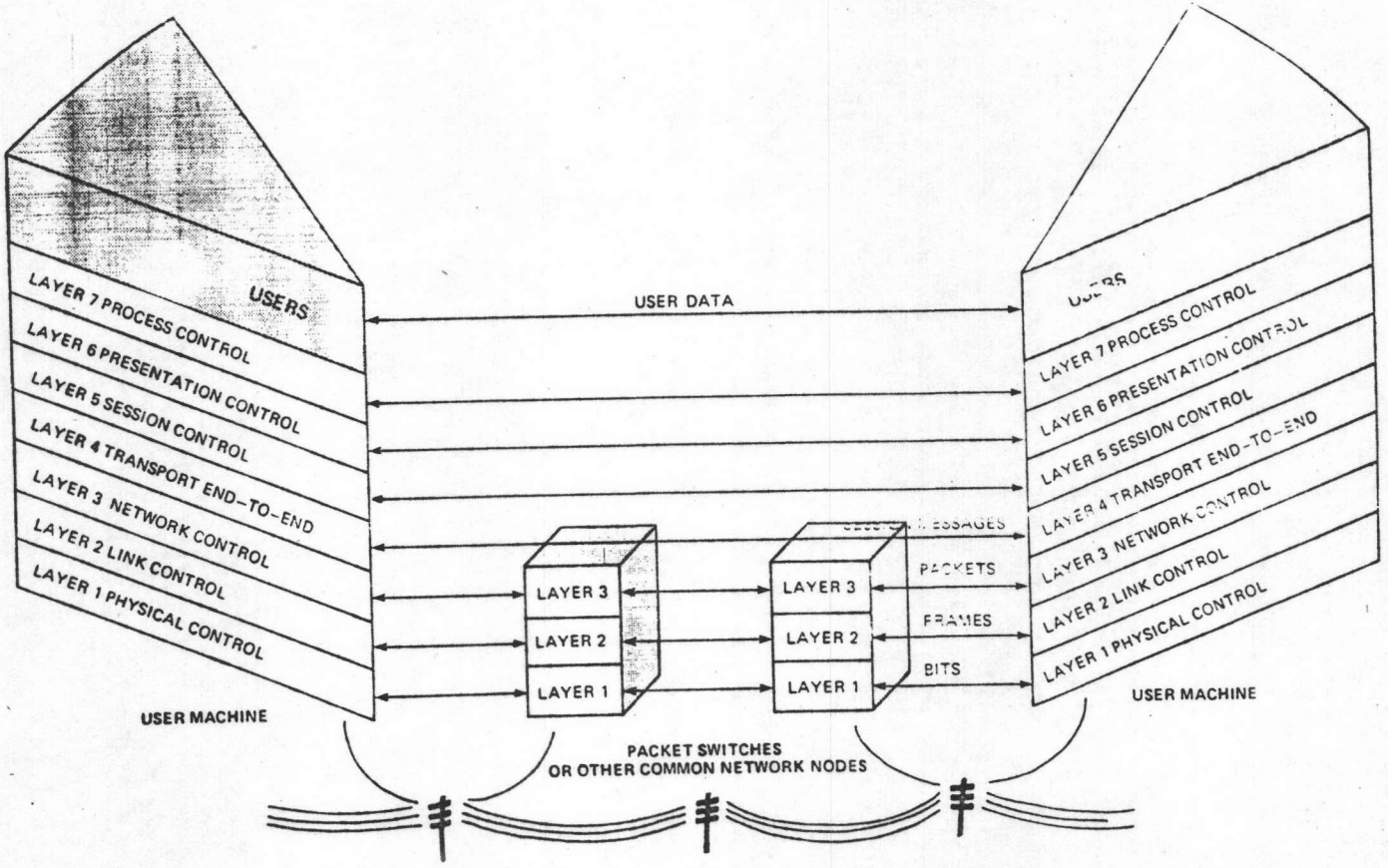
รูปที่ 4.22 แสดงโนดที่ใช้สถาปัตยกรรมของไอเอสโอ

ฟังก์ชันที่แตกต่างกัน ดังแสดงในรูปที่ 4.23 (ก) เมื่อเกิดเซ็ทชันขึ้นระหว่างการประมวลผลของผู้ใช้ ชั้นที่อยู่สูงกว่าได้แก่ ชั้นที่ 5 ถึงชั้นที่ 7 จะเกี่ยวข้องกับเซ็ทชันที่เกิดขึ้นสำหรับชั้นที่ต่ำกว่าชั้นที่ 5 จะไม่เกี่ยวข้องกับเซ็ทชันแต่จะทำหน้าที่เกี่ยวกับการเคลื่อนย้ายข้อมูลผ่านข่ายงาน ซึ่งข่ายงานเหล่านี้จะถูกใช้ร่วมกันระหว่างเครื่องจักรหลายเครื่อง ดังแสดงในรูปที่ 4.23 (ข) ชั้นที่ 4 ถึงชั้นที่ 7 จะทำหน้าที่เกี่ยวกับการติดต่อของปลายทางทั้งสองในขณะที่เกิดเซ็ทชันของผู้ใช้ทั้งสอง สำหรับชั้นที่ 1 ถึงชั้นที่ 3 จะทำหน้าที่ติดต่อสื่อสารระหว่างโนดของข่ายงานซึ่งถูกใช้ร่วมกัน โนดเหล่านี้อาจจะเป็นเพ็คคิทสวิทช์ คอมมิวนิเคชันคอนโทรลเลอ หรือ เครื่องจักรแบบอื่นที่ถูกออกแบบให้ทำงานกับข้อมูลของข่ายงาน จากรูปที่ 4.23 (ข)จะเห็นว่าเป็นการส่งผ่านข้อมูลระหว่างเครื่องจักรของผู้ใช้สองเครื่อง โดยโนดของเน็ทเวอร์ค ซึ่งประกอบด้วยซอฟแวร์ของระบบย่อยสำหรับส่งผ่านข้อมูล

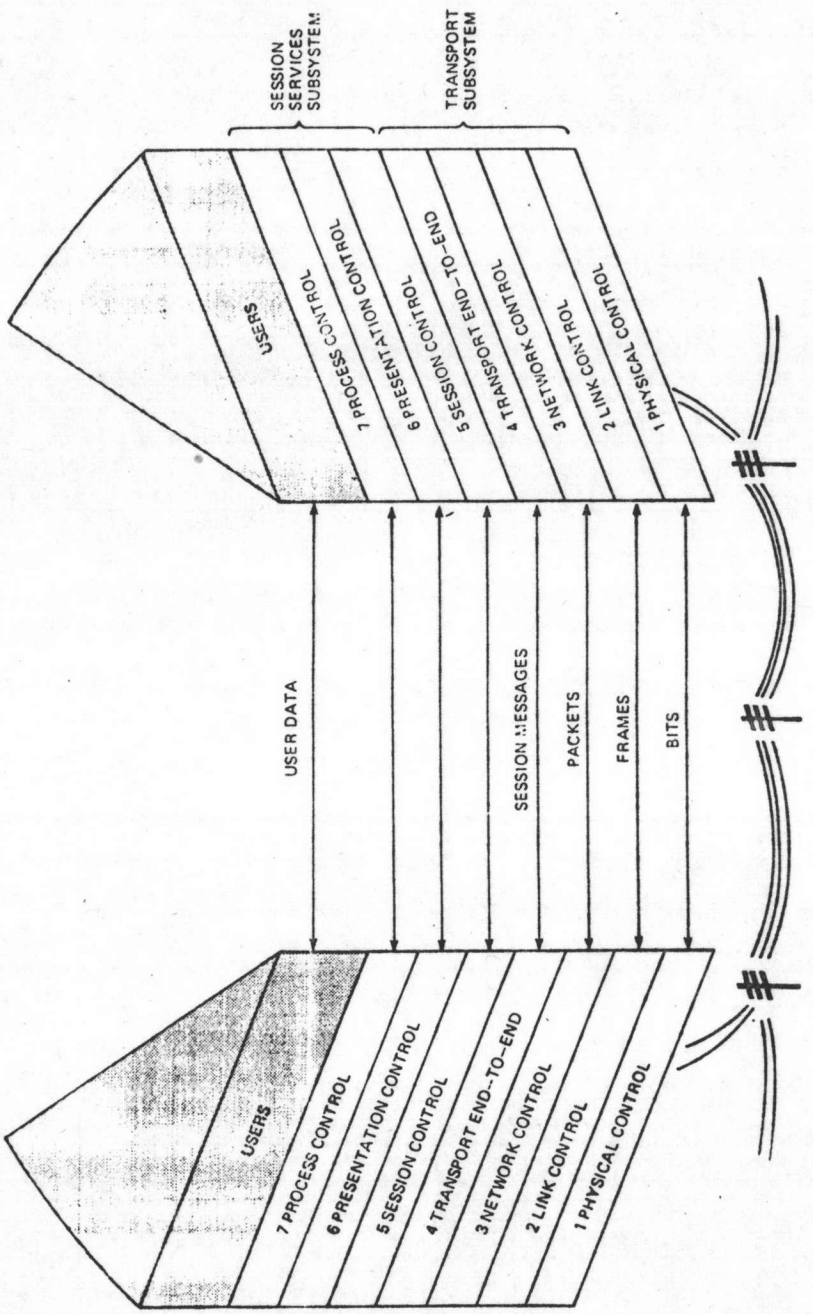
การติดต่อระหว่างชั้นของเครื่องจักรที่แยกจากกันสามารถทำได้สองแบบ คือ การใช้เฮ็ดเดอ และข่าวสารเพื่อการควบคุม (Control message)

– เฮ็ดเดอ แต่ละชั้นของอาคิเท็คเชอแบบชั้น ยกเว้น ชั้นที่ 1 จะเพิ่มเฮ็ดเดอให้กับข่าวสารที่จะถูกส่งออกไป และเฮ็ดเดอเหล่านี้จะถูกแปลความหมายโดยชั้นที่มีคุณสมบัติเหมือนกับชั้นของผู้ส่งที่ปลายทาง เฟรมของชั้นที่ 2 จะประกอบด้วยเฮ็ดเดอ และข้อมูล ซึ่งเฮ็ดเดอเหล่านี้จะถูกใช้โดยกลไกของชั้นที่ 2 ของเครื่องจักรที่อีกปลายทาง นอกจากจะมีเฮ็ดเดอแล้วจะมีเทรเลอ (trailer) ซึ่งถูกใช้เพื่อบอกการจบของเฟรม และใช้ตรวจสอบว่าเฟรมนี้มีข้อผิดพลาดในการส่งผ่านข้อมูลหรือไม่ เพ็คคิทของชั้นที่ 3 จะประกอบด้วยเฮ็ดเดอซึ่งจะเป็นตัวบ่งให้เพ็คคิทไปยังจุดหมายที่ต้องการ เฮ็ดเดอเหล่านี้จะถูกใช้โดยชั้นที่ 3 ที่ปลายทางนั้น ข่าวสารของชั้นที่ 4 จะประกอบด้วยเฮ็ดเดอซึ่งมีวัตถุประสงค์ เพื่อให้ชั้นที่ 4 ของอีกเครื่องจักรหนึ่งใช้ โดยทั่ว ๆ ไปเฮ็ดเดอของชั้นที่ N จะไม่ถูกตรวจสอบโดยชั้นที่ N-1 และในขณะที่ข้อมูลถูกส่งผ่านไปชั้นที่ N-1 จะเพิ่มเฮ็ดเดอของตัวเองเข้ากับข้อมูลด้วย ดังแสดงในรูปที่ 4.24

– ข่าวสารเพื่อการควบคุม ข่าวสารเพื่อการควบคุมจะถูกส่งออกไปคล้าย ๆ กับข้อมูลที่ส่งออกไป โดยทั่ว ๆ ไป แล้วข่าวสารที่เกี่ยวกับการควบคุมที่เกิดขึ้นบ่อย ๆ จะถูก

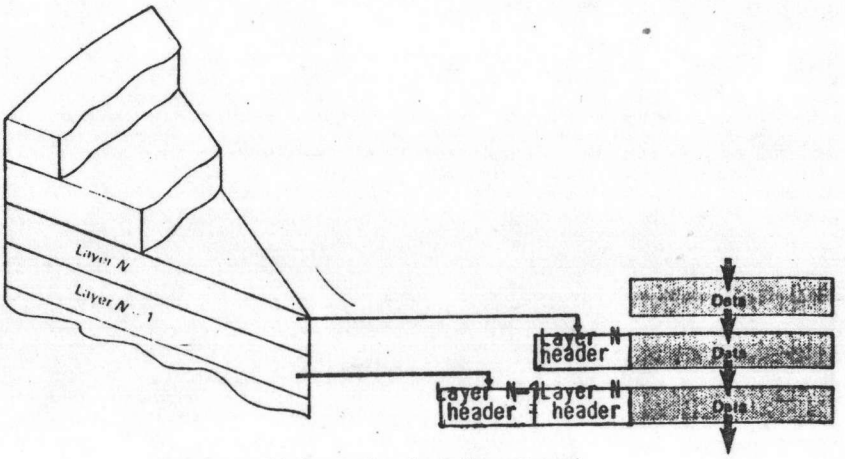


Layers 4 to 7 provide end-to-end communication between session software.
Layers 1 to 3 provide an interface to a shared network.

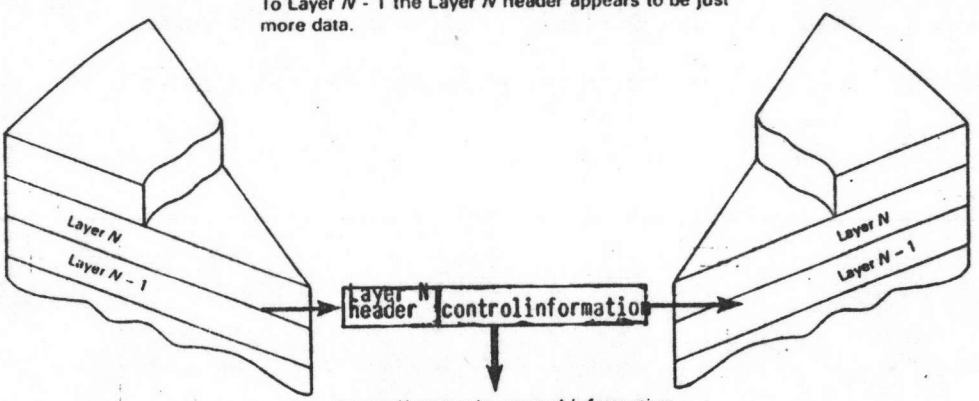


(ข)

รูปที่ 4.23 แสดงการติดต่อระหว่างผู้ใช้โดยใช้สถาปัตยกรรมของไอเอสโอ



Each layer adds its own header to the message.
To Layer *N* - 1 the Layer *N* header appears to be just more data.



Layer *N* transmits control information to its complementary layer.
To Layer *N* - 1 this also appears to be just more data.

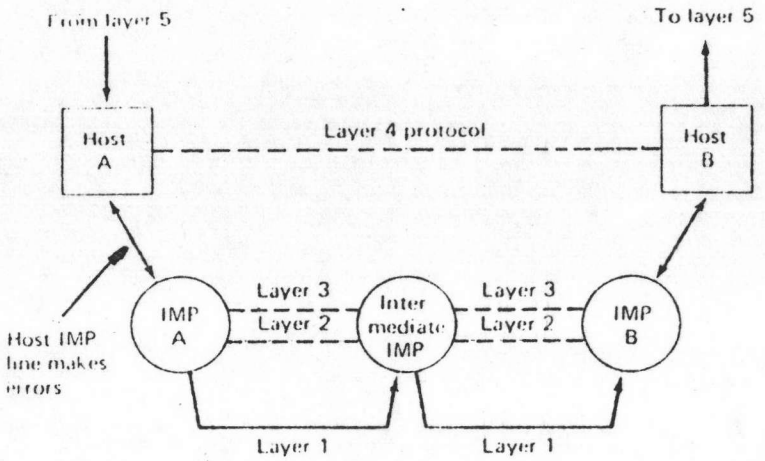
There are two firms of communication between equivalent layers: message headers for that layer, and control messages passed between the layers. These are, or will become, the basis of international standards.

รูปที่ 4.24 แสดงการใช้เฮ็ดเดอและข่าวสารสำหรับการควบคุม

รวมเข้าไว้ในเอ็กเคอ แต่สำหรับข่าวสารเกี่ยวกับการควบคุมที่เกิดขึ้นเพียงไม่กี่ครั้ง หรือมีขนาดที่ใหญ่กว่าขนาดของเอ็กเคอ ก็จะถูกส่งออกไปต่างหากเรียกว่า ข่าวสารเพื่อการควบคุม

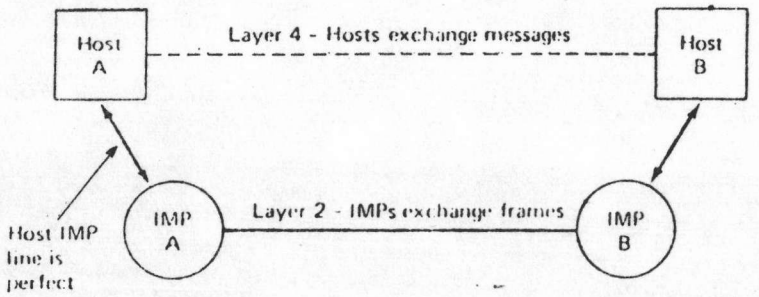
ข่าวสารเพื่อการควบคุมจะถูกใช้เมื่อมีความจำเป็นโดยจะส่งไปยังปลายทางโดยไม่มีการเสียเวลาการส่งข่าวสารเพื่อการควบคุม จะมีลำดับความสำคัญมากกว่าการส่งข้อมูลปกติ เช่น จะไม่สนใจแถวคอยของข้อมูลปกติที่รออยู่ที่ปลายทาง ซึ่งลักษณะเช่นนี้คล้ายกับรถพยาบาลที่พยายามจะไปถึงจุดหมายให้ทันเวลา ไอเอสโอเป็นผู้กำหนดลักษณะของเอ็กเคอ และข่าวสารเพื่อการควบคุมที่แต่ละชั้นใช้อยู่ซึ่งก็เท่ากับเป็นการกำหนดว่าแต่ละชั้นจะติดต่อกันได้อย่างไร คอมพิวเตอร์ที่มาจากบริษัทผู้ผลิตต่าง ๆ อาจจะติดต่อกันได้ถ้าเครื่องเหล่านั้นใช้เอ็กเคอและข่าวสารเพื่อการควบคุมเหมือนกันรวมทั้งการแปลความหมายที่เหมือนกันด้วย

4.6.3.9 ความรู้เบื้องต้นเกี่ยวกับดาตาลิงค์โปรโตคอล สำหรับหัวข้อนี้จะได้ศึกษาเกี่ยวกับวิธีการที่จะทำให้เกิดความเชื่อถือได้ และการติดต่อสื่อสารอย่างมีประสิทธิภาพระหว่างไอเอ็มพี (IMP;Interface Message Processor) ที่ต่อเข้าด้วยกัน หรือระหว่างโฮสกับไอเอ็มพีที่ต่อเข้าด้วยกัน ไอเอ็มพีก็คือส่วนประกอบของข่ายงานที่ทำหน้าที่สวิทชิง ซึ่งคำว่าไอเอ็มพีนี้เป็นศัพท์ของ ARPANET การต่อเข้าด้วยกัน หมายถึงว่าไอเอ็มพีทั้งสองต่อเข้าด้วยกันแบบฟิสิคัลโดยช่องทางในการรับส่งข้อมูล เช่น สายโทรศัพท์ไมโครเวฟ ลักษณะสำคัญของช่องทางนี้ก็คือต้องมีลักษณะเหมือนต่อไว้ด้วยสาย กล่าวคือ บิทที่ถูกส่งออกไปต้องเรียงลำดับเหมือนกับบิทที่ต้องการจะส่ง จากไอเอสโอโมเดลที่ใช้ ผู้ใช้จะส่งข่าวสารลงมาให้ชั้นที่ 6 และส่งเรื่อยมาจนถึงชั้นที่ 4 ก็คือแทร์นสพอทเลเออ ซอฟแวร์ของแทร์นสพอทเลเออจะแยกข้อมูลออกเป็นเม็สเสจที่ต้องการแล้วใส่แทร์นสพอทเอ็กเคอไว้ข้างหน้าของแต่ละเม็สเสจ จากนั้นจะส่งเม็สเสจไปให้ไอเอ็มพีที่ต่ออยู่กับโนดนี้ ไอเอ็มพีจะย้ายตำแหน่งของแพ็คคิทของโฮสและเฟรมเอ็กเคอ แล้วจะใส่เอ็กเคอของตัวเองไปข้างหน้าของแต่ละเม็สเสจ เพื่อสร้างแพ็คคิทที่ใช้ติดต่อระหว่างไอเอ็มพี แพ็คคิทเอ็กเคอของชั้นที่ 3 ที่ใช้ติดต่อระหว่างไอเอ็มพี อาจจะประกอบด้วยที่อยู่ของผู้รับ และผู้ส่งข้อมูล หรือข่าวสารที่ใช้สำหรับเลือกเส้นทางในการส่งผ่านข้อมูล เมื่อแพ็คคิทถูกส่งมายังชั้นที่ 2 จะมีการใส่เอ็กเคอและเทรเลอกับเฟรมเพื่อเตรียมสำหรับการส่งผ่านข้อมูล ต่อจากนั้นเฟรมจะถูกส่งไปยังไอเอ็มพีที่อยู่ถัดไปลักษณะของโมเดลทั่ว ๆ ไปแสดงได้ตามรูปที่ 4.25



รูปที่ 4.25 แสดงโปรโตคอลของชั้นที่ 2 , 3 และ 4 ในกรณีทั่วไป

สำหรับในหัวข้อนี้เราจะสนใจเกี่ยวกับโปรโตคอลของชั้นที่ 2 ซึ่งเราจะใช้โมเดลแบบง่าย ๆ ตามรูปที่ 4.26 เราจะไม่สนใจชั้นที่ส่งข้อมูลเป็นแพ็คคิท คือชั้นที่ 3 แต่จะสนใจเกี่ยวข้องกับเม็สเสจและเฟรม ความคิดเบื้องต้นเกี่ยวกับหัวข้อนี้ก็คือซอฟแวร์ของชั้นที่ 2 จะทำหน้าที่รับข้อมูลเข้ามาแล้วส่งข้อมูลเหล่านี้ออกไปโดยไม่มีข้อผิดพลาด แม้ว่าจะมีเสียงรบกวนในช่องในการรับส่งข้อมูล โดยไม่สนใจว่าข้อมูลที่ได้รับเข้ามาจะมาจากชั้นที่ 3หรือชั้นที่ 4



(b)

รูปที่ 4.26 แสดงโมเดลแบบง่าย ๆ ที่ใช้ในหัวข้อนี้

ความคิดเบื้องต้นเกี่ยวกับโมเดลก็คือถ้าโฮส A ต้องการส่งข้อมูลที่มีความยาวมาให้กับโฮส B ต่อมาพิจารณากรณีที่โฮส B ต้องการส่งข้อมูลให้โฮส A พร้อม ๆ กับที่โฮส A ส่งข้อมูลให้โฮส B แต่ขณะนี้เราจะพิจารณาในลักษณะการส่งผ่านข้อมูลแบบซิมเพล็กซ์จากโฮส A ไปยังโฮส B

สมมติให้โฮส A มีข้อมูลที่พร้อมที่จะส่งออกไปตลอดเวลา เมื่อไอเอ็มพีต้องการข้อมูลโฮส ก็พร้อมที่จะส่งให้ทันที ซึ่งข้อสมมติฐานนี้จะตัดออกภายหลัง เมื่อเสร็จของโฮสจะเป็นข้อมูลจริง ทุก ๆ บิทจะถูกส่งให้กับโฮสหนึ่งซึ่งตามความจริงแล้วโฮสที่เป็นผู้รับข้อมูลจะแปลความหมายส่วนหนึ่งของข้อมูลที่รับเข้ามาว่าเป็นเอ็กเคอ ซึ่งไม่เกี่ยวข้องกับซอฟแวร์ของชั้นที่ 2

เมื่อไอเอ็มพีได้รับเมสเสจก็จะใช้เมสเสจเหล่านี้สร้างเฟรมขึ้น ซึ่งแต่ละเฟรมก็จะเพิ่มคอนโทรลอินฟอเมชัน (Control information) เข้าด้วย จากนั้นเฟรมเหล่านี้

ก็ถูกส่งต่อไปให้อีกไอเอ็มพีหนึ่ง สมมติเรามีโพรซีเซอ (procedure) ที่อยู่ในคอมพิวเตอร์ที่สามารถเรียกใช้ได้ทันทีที่ชื่อว่า getf และ sendf ทำหน้าที่รับและส่งเฟรมตามลำดับ จากนั้นฮาร์แวร์ที่ใช้สำหรับการส่งผ่านข้อมูลจะทำการตรวจสอบข้อผิดพลาดในการส่งผ่านข้อมูล โดยที่ไอเอ็มพีซอฟแวร์ไม่ต้องเกี่ยวข้อง อาจจะกล่าวได้ว่าเป็นหน้าที่ในการทำงานของชั้นที่ 1

เริ่มแรกผู้รับจะไม่ต้องทำอะไรแต่จะรอให้เกิดเหตุการณ์นั้นขึ้น ไอเอ็มพีจะรอคอยให้เกิดเหตุการณ์ขึ้นโดยโพรซีเซอที่ชื่อว่า Wait (event) โพรซีเซอนี้จะทำงานเมื่อมีเหตุการณ์เกิดขึ้น เช่น มีเฟรมเข้ามาจากอีกโฮสหนึ่ง เมื่อเกิดเหตุการณ์ขึ้นตัวแปรที่ชื่อว่า event จะเป็นตัวบ่งชี้ว่าเหตุการณ์อะไรเกิดขึ้น เหตุการณ์ต่าง ๆ ที่จะเกิดขึ้นจะแตกต่างกันสำหรับหลาย ๆ โปรโตคอล ซึ่งเหตุการณ์เหล่านี้จะได้กำหนดและอธิบายไว้สำหรับแต่ละโปรโตคอลสำหรับการทำงานจริง ๆ แล้วไอเอ็มพีจะไม่อยู่เฉย ๆ เพื่อคอยเหตุการณ์ตามสมมติฐานของเรา แต่ไอเอ็มพีจะทำงานอื่นหรือให้บริการกับช่องรับส่งข้อมูลอื่น ๆ ต่อเมื่อได้รับสัญญาณอินเทอรัพท (interrupt) ซึ่งจะบอกให้หยุดงานที่ทำอยู่เพื่อรับเฟรมที่เข้ามา เพื่อให้สามารถเข้าใจง่ายขึ้นเราจะตั้งสมมติฐานอีกว่าไอเอ็มพีให้บริการกับช่องรับส่งข้อมูลช่องเราเพียงช่องเดียว

เมื่อเฟรมเข้ามาถึงฮาร์ดแวร์ของผู้รับจะตรวจสอบข้อผิดพลาดของข้อมูลที่ส่งเข้ามา ถ้าหากมีข้อผิดพลาดไอเอ็มพีจะจัดให้ตัวแปร event เป็น CksumErr ถ้าเฟรมถูกรับเข้ามาได้โดยไม่เสียหายไอเอ็มพีจะจัดให้ตัวแปร event เป็น FrameArrival จากนั้นจะตรวจสอบเฟรมที่เข้ามาด้วยโพรซีเซอ getf โดยตรวจสอบคอนโทรลอินฟอเมชันถ้าทุกอย่างเรียบร้อยก็จะส่งส่วนของเมสเสจที่มากับเฟรมให้กับโฮส สำหรับเฮ็กเดอของเฟรมจะไม่ถูกส่งให้กับโฮส

สำหรับเหตุผลที่ดีสำหรับการไม่ส่งเฮ็กเดอให้กับโฮส เพราะต้องการให้โปรโตคอลของโฮส และของไอเอ็มพีแยกกันโดยเด็ดขาด ตราบใดที่โฮสไม่รู้จักรูปแบบของเฟรม และดาตาลิงค์คอนโทรลโปรโตคอล การเปลี่ยนแปลงของสิ่งเหล่านี้สามารถทำได้โดยไม่ต้องเปลี่ยนแปลงซอฟแวร์ของโฮส การใช้อินเทอเฟสระหว่างโฮส และไอเอ็มพีจะทำให้การออกแบบของซอฟแวร์ง่ายขึ้น เพราะโปรโตคอลที่ใช้สำหรับการติดต่อสื่อสาร และโปรโตคอลของโฮสเป็น

อิสระแก่กัน จะมีปัญหามากมายถ้าหากมีการเปลี่ยนแปลงโปรโตคอลของชั้นที่อยู่ต่ำกว่าเพียงเล็กน้อย แล้วมีผลให้ต้องเปลี่ยนแปลงระบบควบคุมการปฏิบัติงานของโฮส โดยทั่ว ๆ ไปแล้วไอเอ็มพีในข่ายงานส่วนใหญ่มักจะถูกกำหนดให้เป็นชิ้นของฮาร์แวร์ การเปลี่ยนซอฟแวร์ของไอเอ็มพีก็จะทำให้มีการเปลี่ยนแปลงเพียงหนึ่งโปรแกรมแล้วส่งโปรแกรมที่แก้ไขแล้วนี้ไปยังไอเอ็มพีต่าง ๆ ตลอดข่ายงาน การเปลี่ยนแปลงโปรโตคอลของโฮส จะเป็นงานที่ซับซ้อนมากในกรณีที่โฮสในข่ายงานมีลักษณะแตกต่างกันซึ่งในกรณีนี้จะเป็นไปได้มาก ดังนั้นการเปลี่ยนแปลงที่จะทำให้ต้องแก้ไขระบบควบคุมการปฏิบัติงานทุก ๆ ระบบในข่ายงาน จึงเป็นงานที่ยุ่งยากกว่าการเปลี่ยนแปลงโปรแกรม ของมินิคอมพิวเตอร์ หรือไมโครคอมพิวเตอร์

รูปที่ 4.27 แสดงโพรซีเซอและตัวแปรต่าง ๆ ที่ใช้ร่วมกันสำหรับหลาย ๆ โปรโตคอล ซึ่งจะได้กล่าวรายละเอียดต่อไป ข้อมูลที่ได้กำหนดโครงสร้างไว้มี 5 ประเภท ได้แก่ bit,SequenceNr, message , FrameKind และ frame bit จะมีค่าเป็นศูนย์หรือหนึ่ง SequenceNr เป็นเลขบวกที่ใช้นับจำนวนเฟรม โดยที่ MaxSeq เป็นค่าคงที่ ที่บ่งบอกลำดับที่มากที่สุดซึ่งจะถูกกำหนดในแต่ละโปรโตคอล message เป็นหน่วยของข่าวสารระหว่างโฮส และไอเอ็มพีหรือระหว่างโฮสกับโฮส สำหรับโมเดลที่ใช้ความยาวของ message จะมีความยาวแน่นอน แต่สำหรับการใช้งานจริงความยาวของ message จะเปลี่ยนแปลงได้

เฟรมจะประกอบด้วย 4 เขตพิกัด คือ kind , seq , ack และ info สำหรับสามเขตพิกัดแรกจะเป็นคอนโทรลอินฟอเมชัน ส่วนเขตพิกัดสุดท้ายเป็นข้อมูลจริงที่ต้องการจะส่งเขตพิกัดต่าง ๆ ที่เป็นคอนโทรลอินฟอเมชันรวมกัน เรียกว่า เฮ็ดเดอของเฟรม kind จะบอกว่ามีข้อมูลในเฟรมหรือไม่เพราะบางโปรโตคอลจะแยกเฟรมที่มีคอนโทรลอินฟอเมชันและเฟรมที่มีข้อมูลออกจากกัน seq และ ack จะใช้บอกลำดับและแอ็คน็อลเล็ดมีนทของเฟรมตามลำดับ info ของคาดาเฟรมจะประกอบด้วยแทร์นสพอทเม็สเสจเดียว สำหรับคอนโทรลเฟรมจะไม่ใช้เขตพิกัดนี้ สำหรับการใช้งานจริงความยาวของเขตพิกัดนี้จะเปลี่ยนแปลงได้ และสำหรับคอนโทรลเฟรมก็จะไม่ใช้เขตพิกัดนี้เหมือนกัน

โพรซีเซอต่าง ๆ ที่จำเป็นต้องใช้ได้แสดงไว้ในรูปที่ 4.27 โพรซีเซอ wait

```
{Some data types and procedures common to a number of the protocols.}

const LastBit = ... ;                    {determines message size}
      doomsday = false;                  {used to repeat forever}
      MaxSeq = ... ;                     {MaxSeq = highest seq = 2^n - 1}

type bit = 0 .. 1;
     SequenceNr = 0 .. MaxSeq;           {used to number the frames}
     message = packed array [0 .. LastBit] of bit;   {a host-host message}
     FrameKind = (data, ack, nak);
     frame = packed record
       kind: FrameKind;
       seq: SequenceNr;
       ack: SequenceNr;
       info: message
     end;

procedure wait(var event: EvType);
begin {Wait for an event to happen; return its type in event.} end;

procedure FromHost(var m: message);
begin {Fetch information from the host for transmission on the channel.} end;

procedure ToHost(m: message);
begin {Deliver information from an inbound frame to the host.} end;

procedure getf(var r: frame);
begin {Go get an inbound frame and copy it to r for the receiver.} end;

procedure sendf(s: frame);
begin {Transmit the frame s over the communication channel.} end;

procedure StartTimer(k:SequenceNr);
begin {Start the clock running and enable TimeOut event.} end;

procedure StopTimer(k:SequenceNr);
begin {Stop the clock and disable TimeOut event.} end;

procedure StartAckTimer;
begin {Start an auxiliary timer for sending separate acks.} end;

procedure StopAckTimer;
begin {Stop the auxiliary timer and disable HostIdle event.} end;

procedure EnableHost;
begin {Allow the host to cause a HostReady event.} end;

procedure DisableHost;
begin {Forbid the host from causing a HostReady event.} end;

procedure inc(var k: SequenceNr);
begin {Increment k circularly.} if k < MaxSeq then k := k + 1 else k := 0 end;
```

รูปที่ 4.27 แสดงถึงโพรซีเซอต่าง ๆ ที่กำหนดขึ้น

จะอยู่ที่ตอนต้นเพื่อรอให้เกิดเหตุการณ์ต่าง ๆ ขึ้นตามที่กล่าวมาแล้ว สำหรับโพรซีเซอ ToHost และ FromHost จะถูกใช้โดยไอเอ็มพีเพื่อส่งเม็สเสจไปยังโฮส และรับเม็สเสจจากโฮสตามลำดับ getf และ sendf จะใช้สำหรับแลกเปลี่ยนเฟรมระหว่างไอเอ็มพี ในขณะที่ ToHost และ FromHost ใช้สำหรับแลกเปลี่ยนเม็สเสจระหว่างไอเอ็มพีกับโฮส

ในโปรโตคอลส่วนมากจะมีช่องทางในการรับส่งข้อมูลที่มีเสียงรบกวน ซึ่งจะทำให้เฟรมเสียไปเพื่อริคัฟเวอร์เฟรมที่ส่งออกไป ไอเอ็มพีที่ส่งเม็สเสจจะจับเวลาขณะที่ส่งเฟรมออกไป ถ้าไม่ได้รับคำตอบในช่วงเวลาที่กำหนดไว้ว่าได้รับเฟรมถูกต้องแล้ว นาฬิกาที่ตั้งเวลาไว้จะตัดเวลาและส่งสัญญาณอินทีรัพท์ให้กับไอเอ็มพี

สำหรับโปรโตคอลของเราจะจัดเกี่ยวกับเรื่องนี้โดยใช้โพรซีเซอ Wait แล้วจัดตัวแปร event เป็น TimeOut โพรซีเซอ StartTimer และ StopTimer ใช้สำหรับเปิดและปิดนาฬิกาที่ใช้จับเวลาตามลำดับ โพรซีเซอ StartAckTimer และ StopAckTimer ใช้ควบคุมไทเมอ (timer) ที่ใช้สร้างการตอบรับภายใต้เงื่อนไขที่แน่นอน

โพรซีเซอ EnableHost และ DisableHost จะใช้สำหรับโปรโตคอลที่มีความซับซ้อน คือเราไม่มีสมมติฐานว่าโฮสมีเม็สเสจพร้อมที่จะส่งตลอดเวลา เมื่อไอเอ็มพีต้องการจะติดต่อกับโฮสของตัวเองในกรณีที่มีเม็สเสจที่ต้องการจะส่งโฮสก็จะอนุญาตให้อินทีรัพท์ จากเหตุการณ์นี้เราจะจัดให้ตัวแปร event เป็น HostReady ในกรณีที่ไม่สามารถติดต่อกับโฮสได้ก็จะไม่มีการจัดค่าให้กับตัวแปร event

ลำดับของเฟรมจะเริ่มต้นตั้งแต่ 0 ไปจนถึง MaxSeq ซึ่งค่า Maxseq จะแตกต่างกันไปในแต่ละโปรโตคอล โดยทั่ว ๆ ไปการเพิ่มค่าของการนับเลขลำดับของเฟรม จะเพิ่มขึ้นครั้งละหนึ่ง ซึ่งการทำงานนี้จะทำโดยโพรซีเซอ inc

– โปรโตคอลแบบ Utopia เราจะเริ่มต้นพิจารณาโปรโตคอลแบบง่าย ๆ โดยที่ข้อมูลจะถูกส่งออกไปได้เพียงทิศทางเดียว โฮสพร้อมเสมอที่จะรับและส่งข้อมูล เวลาที่ใช้ในการประมวลผลน้อยมาก บัฟเฟอมีขนาดใหญ่มาก และช่องทางในการรับส่งข้อมูลระหว่างไอเอ็มพีจะไม่มีเสียงรบกวนที่จะทำให้เฟรมเสียหาย ข้อจำกัดเหล่านี้ตามความเป็นจริงแล้วล้วนเป็นปัญหาที่สำคัญยิ่งในการทำงานของโปรโตคอล แต่สำหรับหัวข้อนี้เราจะตั้งสมมติฐาน

เหล่านี้เอาไว้ก่อน เพื่อให้เข้าใจแนวความคิดเบื้องต้นในการทำงานของโปรโตคอล โปรโตคอลนี้ให้ชื่อว่า Utopia ดังแสดงในรูปที่ 4.28 โปรโตคอลจะประกอบด้วยโพรซีเซอ คือ sender และ receiver โดยที่โพรซีเซอ sender จะทำงานอยู่บนไอเอ็มพีที่จะส่งข้อมูล ส่วนโพรซีเซอ receiver จะทำงานอยู่บนไอเอ็มพีที่เป็นผู้รับข้อมูล ไม่จำเป็นต้องนับเลขลำดับของเฟรม หรือใช้การตอบรับ ดังนั้น MaxSeq ก็ไม่จำเป็นต้องใช้ ตัวแปรตัวเดียวที่ใช้ก็คือ FrameArrival ซึ่งจะบอกให้รู้ว่าเฟรมที่เข้ามาไม่เสียหาย

```
{Protocol 1 ("Utopia") provides for data transmission in one direction only,
 from sender to receiver. The communications channel is assumed to be
 error free, and the receiver is assumed to be able to process all the input
 infinitely fast. Consequently, the sender just sits in a loop pumping data
 out onto the line as fast as it can.}

type EvType = (FrameArrival);

procedure sender1;
var s: frame;                          {buffer for an outbound frame}
    buffer: message;                   {buffer for an outbound message}
begin
  repeat
    FromHost(buffer);                  {go get something to send}
    s.info := buffer;                  {copy it into s for transmission}
    sendf(s);                          {send it on its way}
  until doomsday                       {tomorrow, and tomorrow, and tomorrow,
                                        creeps in this petty pace from day to day,
                                        to the last syllable of recorded time;
                                                        - Macbeth, V, v}
end; {sender1}


procedure receiver1;
var r: frame;
    event: EvType;                     {filled in by wait, but not used here}
begin
  repeat
    wait(event);                       {only possibility is FrameArrival}
    getf(r);                           {go get the inbound frame}
    ToHost(r.info)                     {pass the data to the host}
  until doomsday
end; {receiver1}
```

An unrestricted simplex protocol.

รูปที่ 4.28 แสดงโปรโตคอลแบบ Utopia

สำหรับโพรซีเซอ sender จะทำงานไปเรื่อย ๆ จนกว่าตัวแปร doomsday จะมีค่าเป็นจริงซึ่งเป็นไปไม่ได้ นั่นคือจะมีการส่งข้อมูลออกไปเรื่อย ๆ ภายในโพรซีเซอจะประกอบด้วย การทำงาน 3 ส่วน คือ นำเม็สเสจมาจากโฮส สร้างเฟรมที่จะส่งออกไปโดยใช้ตัวแปร s และส่งเฟรมออกไป สำหรับโปรโตคอลนี้เซตพิกัดที่ใช้จะใช้แต่เซตพิกัด info เนื่องจากเซตพิกัดอื่นใช้สำหรับตรวจสอบข้อผิดพลาด และควบคุมการส่งผ่านของข้อมูล ซึ่งในโปรโตคอลนี้ไม่มีข้อจำกัดแบบนี้

สำหรับโพรซีเซอ receiver ก็จะเป็นแบบง่าย ๆ โดยเริ่มต้นจะรอให้เกิดเหตุการณ์ขึ้น เมื่อเฟรมมาถึงก็กำหนดค่าให้ตัวแปร event เป็น FrameArrival จากนั้นโพรซีเซอ getf จะนำเฟรมที่เข้ามาจากบัฟเฟอให้กับตัวแปร r และท้ายที่สุดส่วนของข้อมูลจะถูกส่งผ่านไปยังโฮส จากนั้นไอเอ็มพีก็จะกลับไปรอให้เกิดเหตุการณ์ต่อไป

– โปรโตคอลแบบ Stop - and Wait เราจะยกเลิกข้อจำกัดที่ไม่สอดคล้องกับความเป็นจริงจากโปรโตคอลแบบแรกไว้ก่อน ได้แก่ ความสามารถในการประมวลผลข้อมูลที่รวดเร็วมากของโฮสที่เป็นผู้รับข้อมูล หรือไอเอ็มพีที่มีบัฟเฟอขนาดใหญ่มาก เป็นต้น

ปัญหาหลักที่จะเกี่ยวข้องด้วยได้แก่ จะทำอย่างไรให้ผู้ส่งไม่ส่งเม็สเสจให้กับผู้รับมากเกินไปจนไม่สามารถทำงานได้ ถ้าผู้รับต้องใช้เวลา t ในการทำงานโพรซีเซอ getf และ ToHost ผู้ส่งจะต้องส่งข้อมูลในช่วงเวลา t มีค่าเฉลี่ยน้อยกว่าหนึ่งเฟรม ถ้าเราสมมติว่าฮาร์ดแวร์ของผู้รับไม่สามารถจัดการเกี่ยวกับบัฟเฟอ และแถวคอยอย่างอัตโนมัติ ผู้ส่งจะส่งเฟรมใหม่ไม่ได้จนกว่าเฟรมเก่าจะถูกนำไปใช้งานโดยโพรซีเซอ getf เพื่อป้องกันเฟรมใหม่จะถูกเขียนทับลงบนเฟรมเก่า

สำหรับข้อจำกัดอื่น ๆ ก็อาจจะเป็นไปได้ที่จะทำให้ผู้ส่งทำงานได้ช้าลงจนผู้รับทำงานได้ทัน โดยทั่ว ๆ ไปแล้วแต่ละไอเอ็มพี จะมีช่องทางในการรับส่งข้อมูลหลายช่องมาต่ออยู่ ช่วงเวลาที่เฟรมจะเข้ามา และเวลาช่วงเวลาที่เฟรมกำลังถูกประมวลผล อาจเปลี่ยนแปลงไปได้มาก ถ้าผู้ออกแบบซอฟแวร์สามารถคำนวณเวลาที่ผู้รับใช้ในการทำงานมากที่สุด ก็จะสามารถโปรแกรมผู้ส่งให้ส่งข้อมูลช้าลงพอที่จะทำให้ทุก ๆ เฟรม เกิดการเสีย

เวลาน้อยที่สุด แต่สำหรับวิธีนี้จะทำให้การใช้การติดต่อสื่อสารไม่เหมาะสม

วิธีแก้ไขที่นิยมกันมากก็คือผู้รับจะส่งสัญญาณกลับไปยังผู้ส่งให้รับรู้ หลังจากที่ส่งเม็สเสจให้โฮสแล้ว ผู้รับจะส่งเฟรมที่เรียกว่า เฟรมตอบรับ เพื่อบอกให้ผู้ส่งว่าให้ส่งเฟรมใหม่ได้ ดังนั้น หลังจากส่งเฟรมไปแล้ว ผู้รับจะถูกกำหนดโดยโปรโตคอลให้คอยเฟรมที่ถูกส่งกลับมา

โปรโตคอลที่ผู้ส่ง ส่งเฟรมออกไปแล้วรอคอยการตอบรับ ก่อนที่จะส่งเฟรมใหม่เรียกว่า Stop - and - Wait รูปที่ 4.29 แสดงตัวอย่างของ Stop - and - Wait protocol สำหรับในโปรโตคอลแบบที่หนึ่ง ผู้ส่งจะเริ่มต้นด้วยการไปนำเอาเม็สเสจมาจากโฮส เพื่อนำมาสร้างเฟรมแล้วส่งออกไป แต่สำหรับโปรโตคอลแบบที่สอง จะไม่เหมือนกับโปรโตคอลแบบที่หนึ่ง ผู้ส่งจะต้องคอยจนเฟรมตอบรับ มาถึงก่อนจึงจะเริ่มทำงานต่อไป

สำหรับผู้รับของโปรโตคอลแบบที่ 1 และโปรโตคอลแบบที่ 2 จะแตกต่างกันที่หลังจากรับเม็สเสจส่งต่อให้โฮสแล้ว ผู้รับของโปรโตคอลแบบที่ 2 จะส่งเฟรมตอบรับให้กับผู้ส่งก่อนที่จะไปรอเพื่อรับเฟรมใหม่ เฟรมที่ส่งกลับไปให้กับผู้ส่งไม่จำเป็นต้องมีข้อมูล เพราะผู้ส่งจะสนใจเพียงมีเฟรมถูกส่งกลับไปเท่านั้น

- โปรโตคอลแบบ PAR พิจารณาลักษณะของช่องในการรับส่งข้อมูลที่ใช้จริง คือจะมีข้อผิดพลาดเกิดขึ้น เฟรมที่ส่งอาจจะหายไปเพียงบางส่วนหรือถูกทำลายทั้งหมดก็ได้ เราสมมติว่าถ้าเฟรมถูกทำลายแต่ฮาร์ดแวร์ของผู้รับจะตรวจสอบข้อผิดพลาดได้ แต่ในกรณีที่เฟรมถูกทำลายไปแล้ว แต่ไม่สามารถตรวจสอบข้อผิดพลาดได้ โปรโตคอลที่ใช้งานอยู่จะส่งข้อมูลที่ผิดให้กับโฮส ซึ่งแสดงว่าโปรโตคอลนี้ไม่สามารถทำงานได้

ดูเหมือนว่าถ้าดัดแปลงโปรโตคอลแบบที่ 2 เล็กน้อยจะสามารถทำงานได้ ผู้ส่งจะส่งเฟรมของข้อมูล แต่ผู้รับจะส่งเพียงเฟรมตอบรับในกรณีที่ได้รับข้อมูลอย่างถูกต้อง ถ้าเฟรมที่เสียหายเข้ามาถึงผู้รับ ก็จะไม่สนใจ หลังจากนั้นชั่วขณะหนึ่งผู้ส่งก็จะตัดเวลาแล้วส่งเฟรมใหม่อีก ขบวนการนี้จะทำต่อไปเรื่อย ๆ จนกระทั่งเฟรมสุดท้ายเข้ามาครบ

```
{Protocol 2 ("stop-and-wait") also provides for a one directional flow of data
 from sender to receiver. The communications channel is once again assumed
 to be error free, as in protocol 1. However, this time, the receiver has only
 a finite buffer capacity and a finite processing speed, so the protocol must
 explicity prevent the sender from flooding the receiver with data faster than
 it can be handled.}
type EvType = (FrameArrival);

procedure sender2;
var s: frame;
    buffer: message;
    event: EvType;
begin
  repeat
    FromHost(buffer);          {fetch information from the host}
    s.info := buffer;          {copy it into s for subsequent transmission}
    sendf(s);                  {bye bye little frame}
    wait(event);               {do not proceed until given go ahead}
  until doomsday
end; {sender2}

procedure receiver2;
var r,s: frame;
    event: EvType;
begin
  repeat
    wait(event);               {only possibility is FrameArrival}
    getf(r);                   {go get the frame}
    ToHost(r.info);            {give the message to the host}
    sendf(s);                  {send a dummy frame as a go ahead signal}
  until doomsday
end; {receiver2}
```

A simplex stop-and-wait protocol.

รูปที่ 4.29 แสดง Stop - and - Wait Protocol

จากที่กล่าวมาแล้วมีข้อเสียมาก ลองพิจารณาว่าส่วนใดผิด โฮส A ส่งเม็สเส็จให้กับไอเอ็มพีของตัวเอง ต่อจากนั้นเน็ทเวอร์คจะเป็นตัวรับรองให้เม็สเส็จนี้ส่งผ่านไอเอ็มพีของ B ให้กับโฮส B อย่างถูกต้องในลำดับเดียวกัน แต่โฮส B จะไม่รู้ว่าเม็สเส็จที่ได้รับครบตามจำนวนหรือไม่ หรือมีเม็สเส็จที่ซ้ำกันหรือไม่ ลองพิจารณาเหตุการณ์ต่อไปนี้

- โฮส A ส่งเม็สเส็จให้ไอเอ็มพี A จากนั้นเม็สเส็จถูกส่งไปยังโฮส B โดยผ่านไอเอ็มพี B ไอเอ็มพี B จะส่งเฟรมตอบรับให้กับไอเอ็มพี A
- ถ้าเฟรมตอบรับสูญหาย โดยที่ช่องที่ใช้รับส่งข้อมูลไม่สามารถบอกข้อผิดพลาดนี้ได้เนื่องจากช่องเหล่านี้จะจัดการเกี่ยวกับเฟรมข้อมูลไม่สามารถจัดการเกี่ยวกับเฟรมที่ใช้สำหรับการควบคุม
- ไอเอ็มพี A จะตักเวลาเนื่องจากไม่ได้รับเฟรมตอบรับแล้วก็จะส่งเม็สเส็จเดิมออกไปอีก
- บังเอิญเม็สเส็จที่ส่งไปใหม่ไม่ถูกทำลาย ทำให้โฮส B รับเม็สเส็จที่ซ้ำกันแสดงว่าโปรโตคอลทำงานผิดพลาด

จากที่กล่าวมาแล้ว เห็นได้ชัดว่าควรจะมีวิธีสำหรับผู้รับที่จะแยกเฟรมที่พึ่งส่งมาครั้งแรก หรือเฟรมที่ส่งมาอีกครั้งเมื่อเฟรมเก่าเสียหาย วิธีง่าย ๆ ก็คือให้ผู้ส่งใส่เลขลำดับลงในเฮ็ดเดอของทุก ๆ เฟรมที่ส่งออกมา ทำให้ผู้รับสามารถตรวจสอบเลขลำดับได้ว่ามีเฟรมที่ซ้ำกันหรือไม่

เนื่องจากขนาดของเฮ็ดเดอต้องการขนาดเล็ก จะมีปัญหาเกิดขึ้นว่าจะต้องใช้กี่บิทสำหรับเลขลำดับ สำหรับโปรโตคอลนี้จะมีความคลุมเคลือในการส่งเฟรมที่ m และ m+1 ถ้าเฟรมที่ m สูญหายผู้รับจะไม่ส่งเฟรมตอบรับ ทำให้ผู้ส่งต้องพยายามส่งเฟรมที่ m ใหม่ ดังนั้นการจะส่งเฟรมที่ m หรือ m+1 ขึ้นกับว่าเฟรมตอบรับ ถูกรับอย่างถูกต้องหรือไม่

เลขลำดับใช้เพียงหนึ่งบิทก็เพียงพอ ในช่วงเวลาหนึ่งผู้รับจะได้เลขลำดับที่แตกต่างกันคือ 0 และ 1 ถ้ารับเลขลำดับเป็น 0 ต่อไปเลขลำดับต้องเป็น 1 ถ้าเลขลำดับที่ส่งเข้ามาเหมือนกันทั้งสองครั้งแสดงว่าเฟรมที่ส่งเข้ามานี้ซ้ำ สำหรับตัวอย่างของโปรโตคอลนี้แสดงได้ดังรูปที่ 4.31 ซึ่งคล้าย ๆ กับโปรโตคอลแบบที่ 2 ซึ่งสามารถส่งข้อมูลในทิศทางเดียว

```
{Protocol 3 ("par") allows data to be transmitted in one direction over a
 noisy communications channel that garbles and even loses frames.}

const MaxSeq = 1;

type EvType = (FrameArrival, CksumErr, TimeOut);

procedure sender3;
var NextFrameToSend: SequenceNr; {sequence number of next outgoing frame}
    s: frame;                    {scratch variable}
    buffer: message;             {buffer for outbound message}
    event: EvType;
begin
  NextFrameToSend := 0;          {initialize outbound sequence numbers}
  FromHost(buffer);              {fetch first message}
  repeat
    s.info := buffer;            {construct frame for transmission}
    s.seq := NextFrameToSend;    {insert sequence number in frame}
    sendf(s);                    {send it on its way}
    StartTimer(s.seq);           {if answer takes too long, time out}
    wait(event);                 {possibilities: FrameArrival, CksumErr, TimeOut }
    if event = FrameArrival then
      begin                      {an acknowledgement has arrived intact}
        FromHost(buffer);        {fetch the next one to send}
        inc(NextFrameToSend);    {invert NextFrameToSend}
      end
  until doomsday
end; {sender3}


procedure receiver3;
var FrameExpected: SequenceNr;   {FrameExpected = 0 or 1}
    r,s: frame;                  {scratch variables}
    event: EvType;
begin
  FrameExpected := 0;
  repeat
    wait(event);                 {possibilities: FrameArrival, CksumErr}
    if event = FrameArrival then
      begin                      {a valid frame has arrived}
        getf(r);                 {accept inbound frame}
        if r.seq = FrameExpected then
          begin                  {this is what we have been waiting for}
            ToHost(r.info);      {pass the data to the host}
            inc(FrameExpected)   {next time expect the other sequence nr}
          end;
        sendf(s)                 {none of the fields are used!}
      end
  until doomsday
end; {receiver3}
```

. A Positive Acknowledgement/Retransmission protocol.

รูปที่ 4.30 แสดงโปรโตคอลแบบ PAR

แม้ว่าจะสามารถจัดการกับเฟรมที่สูญหายได้โดยใช้วิธีตัดเวลาแล้วเริ่มส่งเฟรมออกมาใหม่ ซึ่งช่วงเวลาก่อนการตัดเวลาต้องนานพอสมควรไม่เช่นนั้นจะเกิดการตัดเวลาในขณะที่ เฟรมตอบรับกำลังจะไปถึง จะทำให้ผู้ส่ง ส่งข้อมูลที่ซ้ำกัน ถ้าการตอบรับของเฟรมแรกถูกส่งเข้ามา แต่ผู้ส่งเข้าใจผิดคิดว่าเป็นการตอบรับของเฟรมที่สองที่เพิ่งส่งออกไป และไม่รู้ว่ามีเฟรมตอบรับอีกเฟรมหนึ่งกำลังจะมาถึง ต่อมาถ้าส่งเฟรมที่สามออกไปแต่เฟรมที่สามสูญหาย แต่พอดีเฟรมตอบรับของเฟรมที่สองเข้ามาพอดี ผู้ส่งก็จะเข้าใจผิดจะไม่ส่งเฟรมที่สามที่สูญหายออกไปใหม่ ทำให้โปรโตคอลทำงานผิดพลาดใช้ไม่ได้ผล ซึ่งโปรโตคอลต่อ ๆ ไปในเฟรมตอบรับจะมีข้อมูลเพื่อป้องกันข้อผิดพลาดแบบนี้

โปรโตคอลแบบที่สาม จะแตกต่างจากที่กล่าวแล้วทั้งผู้รับ และผู้ส่ง โดยจะมีตัวแปรที่จะบันทึกเมื่อไอเอ็มพีอยู่ในระหว่างรอคอยให้เหตุการณ์เกิดขึ้น `NextFrameTosend` จะเป็นตัวแปรที่บอกเลขลำดับของเฟรมที่จะส่งต่อไป ผู้รับจะจำเลขลำดับของเฟรมที่จะรับต่อไปด้วยตัวแปร `FrameExpected` ก่อนที่จะเข้าวงรอบจะมีการกำหนดค่าต่าง ๆ ในตอนต้น ๆ ของโพรซีเซอ

หลังจากส่งเฟรมออกไปแล้ว ผู้ส่งจะเริ่มจับเวลาโดยจะเริ่มกำหนดค่าของช่วงเวลาซึ่งจะนานพอที่เฟรมจะไปถึงผู้รับรวมกับเวลาที่ผู้รับใช้ประมวลผล และเวลาที่เฟรมตอบรับจะกลับไปยังผู้ส่ง ถ้าช่วงเวลาผ่านพ้นไปแล้วจะคิดว่าเฟรมที่ส่งออกไป หรือเฟรมตอบรับสูญหาย ผู้ส่งก็จะเริ่มส่งเฟรมเดิมออกไปใหม่

หลังจากส่งเฟรมออกไปแล้ว ผู้ส่งจะคอยให้เกิดเหตุการณ์ขึ้น ซึ่งเหตุการณ์ที่จะเกิดขึ้นเป็นไปได้ 3 ทาง คือ เฟรมตอบรับกลับมาได้โดยถูกต้อง เฟรมตอบรับกลับมาโดยไม่ถูกต้อง หรือไทเมอจะปิด ถ้าเฟรมตอบรับที่ถูกต้องกลับมาถึง ผู้ส่งจะเอาเม็สเสจต่อไปมาจากโฮสเก็บลงในบัฟเฟอหับเม็สเสจเก่า และเพิ่มเลขลำดับแล้วสร้างเฟรมใหม่ส่งออกไป ถ้าไม่มีเฟรมถูกส่งกลับมาหรือเฟรมที่ถูกทำลายถูกส่งกลับมา จะไม่มีการเปลี่ยนแปลงข้อมูลในบัฟเฟอ และเลขลำดับ จากนั้นก็ส่งเฟรมเดิมออกไปอีกครั้งหนึ่ง

สำหรับผู้รับถ้าเฟรมที่ถูกต้องมาถึง จะมีการตรวจสอบเลขลำดับว่าซ้ำกับเฟรมที่ได้รับก่อนหน้านี้หรือไม่ ถ้าไม่ซ้ำกันก็จะส่งต่อให้โฮส แล้วส่งเฟรมตอบรับกลับไปให้แก่ผู้ส่ง สำหรับเฟรมที่ซ้ำ หรือเสียหายจะไม่ส่งให้กับโฮส

สำหรับโปรโตคอลที่กล่าวมาทั้งสามแบบ เป็นโปรโตคอลแบบง่าย ๆ ที่มีข้อจำกัด สำหรับการใช้งานจริงโปรโตคอลจะซับซ้อนมาก เนื่องจากข้อจำกัดที่ได้สมมติไว้จะถูกตัดออกไป การที่นำโปรโตคอลทั้งสามแบบนี้มาแสดงให้เห็นก็เพื่อให้เข้าใจ หลักการทำงานเบื้องต้นของโปรโตคอล

ในระบบดีดีพีหน่วยประมวลผลข้อมูลส่วนกลาง และหน่วยประมวลผลข้อมูลย่อยต้องใช้สถาปัตยกรรมของข่ายงานที่เข้ากันได้ สำหรับแต่ละหน่วยจะมีลักษณะอย่างไร หรือใช้จำนวนกี่ชั้นก็ต้องขึ้นกับการนำไปใช้งาน