บทที่ 2
หลักการและทฤษฎี

ในบทนี้จะกล่าวถึงหลักการ และทฤษฎีที่เกี่ยวกับการสร้างโปรแกรมจัดการระเบียน แสดงผลการศึกษาผ่านทางเครือข่ายอินเทอร์เน็ต โดยประกอบไปด้วยทฤษฎีที่เกี่ยวกับระบบเครือข่าย การสื่อสารบนอินเทอร์เน็ต การออกแบบฐานข้อมูล การเขียนโปรแกรมติดต่อกับฐานข้อมูลบนอินเทอร์เน็ต

2.1 ทฤษฎีระบบเครือข่าย

2.1.1 รูปแบบเครือข่ายมาตรฐานสากล

องค์การ International Standards Organization (ISO) ได้กำหนดรูปแบบโครงสร้างมาตรฐานสากลสำหรับการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ไว้เรียกว่า Open Systems Interconnection (OSI) ซึ่งมีอยู่ทั้งหมด 7 ชั้นสื่อสาร ตัวโครงสร้างเองได้เน้นความสำคัญของรูปแบบการติดต่อสื่อสาร ระหว่างระบบเปิด (Open systems) กับระบบปิด จึงสามารถนำไปใช้อ้างอิงได้ในระดับสากลอย่างแท้จริง

แนวความคิดของการกำหนดมาตรฐานเป็นแบบชั้นสื่อสาร (layers) คือ

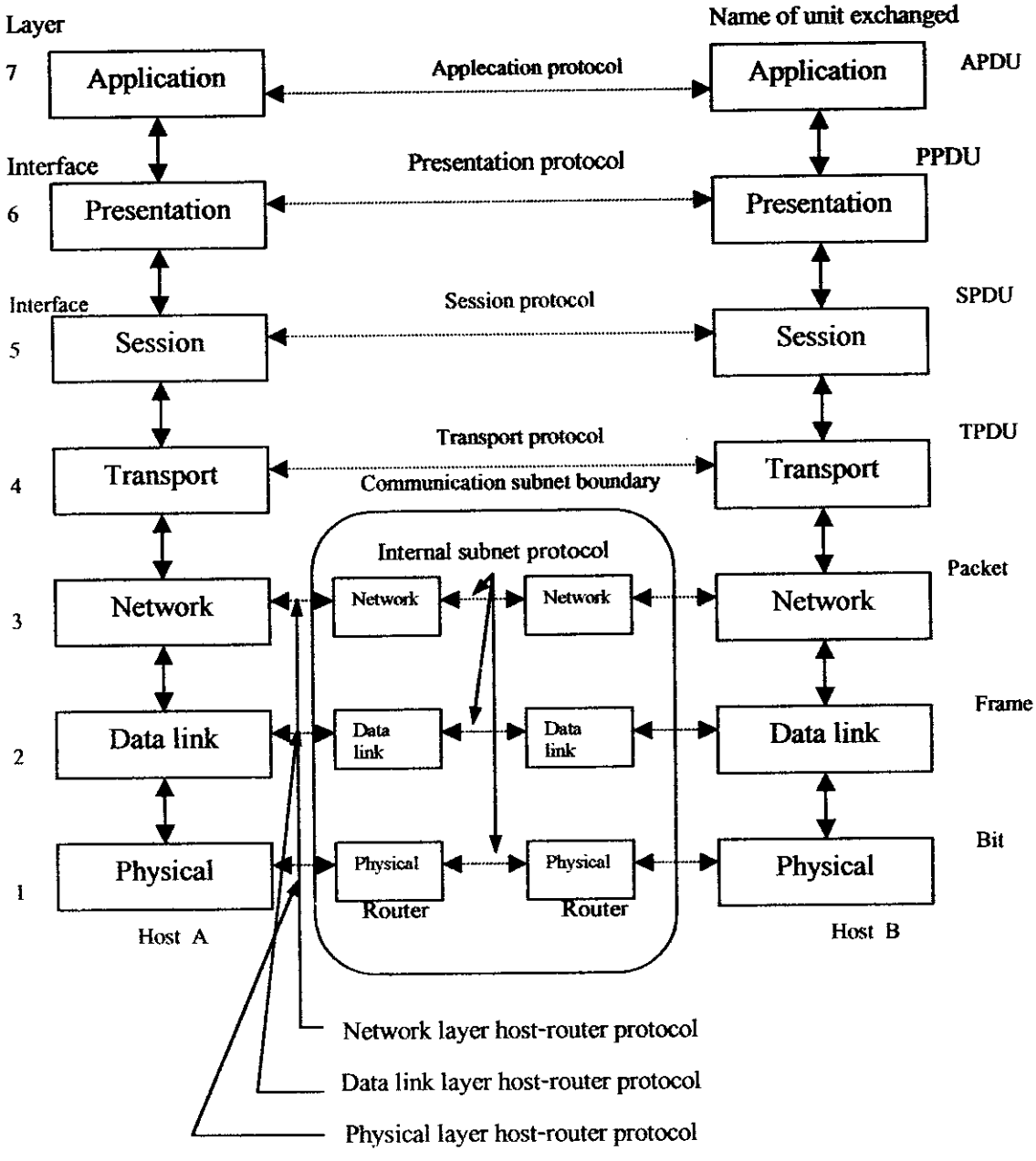
1.ชั้นสื่อสารแต่ละชั้นถูกกำหนดขึ้นมาตามบทบาทที่แตกต่างกัน

2.แต่ละชั้นสื่อสารต้องทำหน้าที่ตามที่ได้รับมอบหมายอย่างดียิ่ง

3.แต่ละฟังก์ชันในชั้นสื่อสารใดๆ จะต้องกำหนดขึ้นมาโดยใช้แนวความคิดในระดับสากลเป็นวัตถุประสงค์หลัก

4.ขอบเขตความรับผิดชอบของแต่ละชั้นสื่อสาร จะต้องกำหนดขึ้นมาเพื่อจำกัดปริมาณการแลกเปลี่ยนข้อมูลและผลกระทบข้างเคียงระหว่างการติดต่อให้มีน้อยที่สุด

5.จำนวนของชั้นสื่อสารจะต้องมีมากพอที่จะแยกฟังก์ชันการทำงานที่แตกต่างกันให้อยู่คนละชั้น แต่จะต้องไม่มีมากเกินความจำเป็น



รูปที่ 2.1 รูปแบบโครงสร้าง OSI

(ที่มา: Computer Network, Andrew S. Tanenbaum)

ชั้นสื่อสารกายภาพ (The Physical Layer)

ชั้นกายภาพเป็นชั้นระดับล่างสุดที่เกี่ยวข้องโดยตรงกับอุปกรณ์สื่อสารต่างๆ ทำหน้าที่ในการกำหนดวิธีควบคุมการรับและการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ในระดับบิต ได้แก่การส่งบิต 0 จะแทนด้วยกระแสไฟฟ้ากี่โวลต์ และบิต 1 จะต้องใช้กี่โวลต์, แต่ละบิตจะใช้ระยะเวลาในการ

ส่งนานเท่าไร, การส่งเป็นแบบทางเดียวหรือสองทาง, จะเริ่มติดต่ออย่างไร, การติดต่อจะสิ้นสุดอย่างไร, และสายเคเบิ้ลมีกี่เส้นแต่ละเส้นใช้เพื่ออะไร เป็นต้น จะเห็นได้ว่ากฎระเบียบ สำหรับชั้นนี้จะเกี่ยวพันโดยตรงกับการทำงานของอุปกรณ์สัญญาณไฟฟ้า(หรือสัญญาณใดๆ) ขั้นตอนในการใช้อุปกรณ์เหล่านั้น และความสัมพันธ์กับสื่อที่ใช้รับ-ส่งสัญญาณ

ชั้นสื่อสารเชื่อมต่อข้อมูล (Data Link Layer)

หน้าที่หลักของชั้นเชื่อมต่อข้อมูลคือ ทำการรวบรวมข้อมูลต่อจากชั้นกายภาพ ตรวจสอบความถูกต้องของข้อมูล แล้วส่งข้อมูลที่ปราศจากข้อผิดพลาดนี้ให้กับชั้นควบคุมเครือข่ายต่อไป โดยปกติผู้ส่งข้อมูลจะแบ่งข้อมูลที่มีความยาวมากออกเป็นกลุ่มข้อมูลย่อยๆแต่ละส่วนย่อยเรียกว่า ดาต้าเฟรม (Data frame) ซึ่งจะมีขนาดคงที่ประมาณสองหรือสามร้อยไบต์ หรืออย่างมากก็ไม่เกินสองถึงสามพันไบต์ชุดของดาต้าเฟรมสำหรับข้อมูลที่ต้องการส่งไปให้ผู้รับก็จะถูกส่งไปทีละเฟรมตั้งแต่เฟรมแรกไปจนครบทุกเฟรม ข้างฝ่ายผู้รับจะตอบสนองโดยการส่งดาต้าเฟรมพิเศษ เรียกว่าเฟรมตอบรับ (Acknowledgement frame) ไปถึงผู้ส่งเพื่อเป็นการบอกให้ทราบว่าได้รับข้อมูลครบแล้ว กระบวนการรับ-ส่งข้อมูล ข้อมูลชุดนี้ก็จะเสร็จสิ้นสมบูรณ์

การรับ-ส่งข้อมูลในชั้นกายภาพนั้นจะไม่รับรู้ในเรื่องโครงสร้างข้อมูล คือจะมองเห็นข้อมูลว่าเป็นบิต 0 หรือบิต 1 กลุ่มหรือชุดหนึ่งที่เรียงตามลำดับ เรียกว่า กระแสบิต (Bit stream) จึงเป็นหน้าที่ของโปรแกรมในชั้นเชื่อมต่อข้อมูลจะต้องทำการตรวจสอบความถูกต้อง ซึ่งทำได้โดยการเพิ่มข้อมูลสำหรับการตรวจสอบติดไว้กับข้อมูลทุกเฟรม เช่น การเพิ่มข้อมูลส่วนหัวและส่วนหาง (Header and tailer) เข้าไปกับทุกเฟรมซึ่งจะใช้เป็นตัวกำหนดขอบเขตของดาต้าเฟรมด้วย

การส่งข้อมูลผ่านระบบใดๆก็ตาม ข้อมูลที่ส่งนั้นมีโอกาสจะเสียหายหรือสูญหายไปเลยก็ได้ โปรแกรมในชั้นเชื่อมต่อข้อมูลจะต้องสามารถตรวจสอบความผิดปกตินี้ได้เอง หรืออาจตอบสนองต่อการตรวจพบโดยโปรแกรมในชั้นกายภาพ เมื่อพบความผิดปกตินี้แล้วก็จะต้องมีวิธีการแก้ไข เช่น แจ้งให้ผู้ส่งข้อมูลได้ส่งข้อมูลชุดเดิมกลับมาใหม่ (เรียกเฟรมนี้ว่า Duplicate frame) อย่างไรก็ตามการส่งข้อมูลซ้ำทำให้เกิดปัญหาตามมาในกรณีที่ชุดข้อมูลไม่ได้สูญหายไปไหนเพียงแต่ใช้เวลาเดินทางมากกว่าปกติ ดังนั้นข้อมูลชุดเดียวกันก็จะมาถึงผู้ใช้ทั้งสองเฟรม โปรแกรมในชั้นนี้จะต้องหาวิธีตรวจสอบและต้องกำจัดเฟรมที่ซ้ำออกไป

ปัญหาอื่นๆ ที่ต้องจัดการให้เรียบร้อย ได้แก่ การรักษาความสมดุล การรับ-ส่งข้อมูลเมื่อผู้ส่งพยายามส่งข้อมูลด้วยความเร็วสูงเกินกว่าที่ผู้รับจะทำงานได้ทัน หรือการแก้ปัญหาการช่วงชิงช่องสื่อสารระหว่างผู้รับและผู้ส่งในระบบการสื่อสารสายเดียวแต่สามารถส่งข้อมูลได้ทั้งสองทิศทาง ทั้งนี้นอกจากผู้รับจะต้องรับข้อมูลแล้ว ยังจะต้องส่งข้อมูลเฟรมตอบรับกลับไปยังผู้ส่งด้วย

ชั้นสื่อสารควบคุมเครือข่าย (Network Layer)

ชั้นควบคุมเครือข่ายมีหน้าที่รับผิดชอบในการควบคุมการติดต่อรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ (เรียกว่าโหนด Node) ต่างๆ ในระบบเครือข่ายให้เป็นไปด้วยความเรียบร้อย สิ่งที่สำคัญที่สุดคือการกำหนดเส้นทางเดินของข้อมูลจากโหนดผู้ส่งข้อมูลไปตามโหนดต่างๆ จนถึงโหนดผู้รับข้อมูลในที่สุดโฮสต์บางกลุ่มจะกำหนดเส้นทางการเดินข้อมูลโดยศึกษาระบบเครือข่ายแล้วสร้างตารางเส้นทางเดินข้อมูลแบบถาวร โฮสต์บางกลุ่มจะกำหนดเส้นทางการเดินข้อมูลในตอนเริ่มต้นของการสื่อสารดังนั้นการสื่อสารในครั้งต่อไป (ติดต่อกับโหนดเดิม) อาจจะเปลี่ยนไปใช้เส้นทางอื่นได้ โฮสต์ในกลุ่มที่มีวิธีการซับซ้อนมากจะกำหนดเส้นทางเดินข้อมูลในระดับแพ็กเก็ต กรณีที่มีผู้ส่งข้อมูลพร้อมๆ กันหลายจุดจะทำให้เกิดความคับคั่งของข้อมูลคล้ายกับสภาวะการจราจรในชั่วโมงเร่งด่วนซึ่งมีปริมาณรถยนต์มากทำให้การจราจรติดขัด โฮสต์ในกลุ่มนี้ก็จะปรับเส้นทางการเดินข้อมูลของแต่ละแพ็กเก็ตให้เหมาะสมกับสภาวะของระบบเครือข่ายตลอดเวลา

การส่งผ่านข้อมูลในระบบเครือข่ายอาจมีการบันทึก ผู้ส่ง ผู้รับ และปริมาณข้อมูลที่ไหลผ่านโฮสต์หรือเราเตอร์ต่างๆ เพื่อประโยชน์ทางด้านการคิดค่าบริการ ซึ่งจะมีความซับซ้อนมากขึ้นถ้าข้อมูลไหลผ่านระบบเครือข่ายย่อยที่มีการคิดอัตราค่าบริการที่แตกต่างกัน

เมื่อแพ็กเก็ตเดินทางผ่านเครือข่ายย่อยระบบหนึ่งไปยังอีกระบบหนึ่งอาจเกิดปัญหาความแตกต่างระหว่างกันในด้านต่างๆ อาทิเช่น การใช้กฎการสื่อสารข้อมูลไม่เหมือนกัน หรือการใช้วิธีการกำหนดตำบลที่อยู่ไม่เหมือนกันปัญหาที่กล่าวถึงนี้เป็นความรับผิดชอบของโปรแกรมในชั้นควบคุมเครือข่ายที่จะต้องหาทางแก้ไขหรือปรับความแตกต่างระหว่างเครือข่ายต่างๆ ให้สามารถเข้าใจกันได้ ท้ายที่สุด การส่งข้อมูลแบบกระจายข่าว (Broadcasting) ที่มีใช้ในบางระบบนั้นจะไม่มีปัญหาที่กล่าวถึงอยู่เลย ดังนั้นโปรแกรมในชั้นนี้จึงมีหน้าที่การทำงานน้อยมากหรืออาจไม่มีเลยก็ได้

ชั้นจัดการนำส่งข้อมูล (Transport Layer)

โปรแกรมในชั้นนำส่งข้อมูลมีหน้าที่หลักในการรับข้อมูลมาจากชั้นควบคุมหน้าต่างสื่อสารซึ่งอาจต้องแบ่งข้อมูลออกเป็นแพ็กเก็ตขนาดย่อม (ในกรณีที่ข้อมูลมีปริมาณมาก) หลายๆแพ็กเก็ตแล้วจึงส่งข้อมูลทั้งชุดต่อไปให้โปรแกรมในชั้นควบคุมเครือข่าย ทางด้านโปรแกรมชั้นนำส่งข้อมูลของผู้รับก็จะทำหน้าที่ประกอบแพ็กเก็ตชุดนี้ให้กลับมารวมเป็นข้อมูลเดิม

ในภาวะปกติ การเชื่อมต่อการสื่อสารจะเป็นการจัดตั้งหน้าต่างสื่อสาร (Session) ระหว่างผู้ส่งและผู้รับตามที่เกิดขึ้น ถ้าต้องการเพิ่มประสิทธิภาพก็อาจสร้างโพรเซสของโปรแกรมนำส่งข้อมูลขึ้นมาหลายๆ โพรเซสเพื่อช่วยกันจัดส่งข้อมูลให้เร็วขึ้นแต่ถ้าเน้นในด้านความประหยัดก็อาจทำใน

ทางตรงกันข้ามนั่นคือการยุบรวมโพรเซสให้เหลือจำนวนน้อยลงแล้วจึงจัดการให้โพรเซสที่เหลืออยู่ทำการส่งข้อมูลทั้งหมดโดยการใช้ช่องสื่อสารร่วมกัน

โปรแกรมในชั้นนี้เป็นผู้กำหนดประเภทของการให้บริการต่างๆ รวมไปถึงการอำนวยความสะดวกในการใช้ระบบเครือข่ายซึ่งแบ่งออกได้เป็น 3 ประเภท ประเภทแรกเป็นการให้บริการแบบจุด-ต่อ-จุด โดยเน้นการรับประกันความถูกต้องของข้อมูลเป็นสำคัญ ประเภทที่สองเน้นการให้บริการข้อมูล ข้อมูลในระดับแพ็กเก็ตซึ่งแม้ว่าจะไม่รับประกันการสูญหายของข้อมูลแต่ก็ให้ความคล่องตัวสูงกว่าแบบแรก (การรับประกันความถูกต้องของข้อมูลสามารถทำในชั้นอื่นได้) ประเภทที่สามเป็นการส่งข้อมูลแบบกระจายข่าวเพื่อประโยชน์ในการส่งข้อมูลชุดเดียวกันไปยังผู้ใช้หลายจุดพร้อมกัน

โปรแกรมในชั้นนำส่งข้อมูลติดต่อถึงกันผ่านช่องสัญญาณเสมือน (Virtual channel) ระหว่างผู้ส่งและผู้รับโดยตรงเรียกว่าเป็นการติดต่อแบบ End-to-End Connection ในขณะที่โปรแกรมในสามชั้นแรกนั้นเป็นการติดต่อแบบจุด-ต่อ-จุด ซึ่งผู้รับอาจไม่ใช่ผู้รับข้อมูลที่แท้จริงแต่เป็นเพียงโหนดตัวกลางในการรับแล้วส่งข้อมูลต่อไปตามเส้นทางเดินข้อมูลที่ถูกกำหนดไว้ รายละเอียดแสดงไว้ในรูป 1-16

เครื่องโฮสต์ส่วนมากจะใช้ระบบปฏิบัติการที่ให้บริการแบบมัลติโปรแกรมมิ่ง (Multiprogramming) คือสามารถสร้าง และใช้งานโพรเซสในชั้นการนำส่งข้อมูลได้หลายโพรเซสในขณะเดียวกัน จึงมีความจำเป็นที่จะต้องเพิ่มข้อมูลส่วนหัว (คือส่วน H4 ในรูป 1-11) เข้าไปกับข้อมูลแต่ละแพ็กเก็ตเพื่อบอกให้ระบบปฏิบัติการของโฮสต์ทราบว่าแพ็กเก็ตที่รับมานั้นเป็นของโพรเซสใด

นอกจากการใช้ช่องสื่อสารร่วมกันแล้วโปรแกรมในชั้นนำส่งข้อมูลจะต้องมีความสามารถในการจัดตั้งหน้าต่างสื่อสารกับโหนดอื่นๆ ในระบบเครือข่าย และจัดการยกเลิกเมื่อการสื่อสารสิ้นสุดลง โปรแกรมในชั้นนี้ยังต้องมีวิธีการกำหนดการตั้งชื่อให้แก่ตนเอง และแนะนำให้ผู้อื่นในระบบฯ ได้รู้จัก รวมทั้งเรียนรู้การกำหนดตำบลที่อยู่ของโหนดอื่นได้ อีกสิ่งหนึ่งที่จำเป็นมากคือการควบคุมการไหลของข้อมูล (Flow control) ซึ่งมีทั้งในระดับโฮสต์ และระดับเราเตอร์โดยมีวัตถุประสงค์ในการควบคุมการรับ และส่งข้อมูลโดยเฉพาะในกรณีที่ผู้ส่งจัดการส่งข้อมูลเร็วเกินกว่าผู้ใช้จะทำงานได้ทัน

ชั้นสื่อสารควบคุมหน้าต่างสื่อสาร (Session Layer)

ชั้นควบคุมหน้าต่างสื่อสารเป็นผู้กำหนดวิธีการควบคุมการเชื่อมต่อระหว่างผู้รับข้อมูล และผู้ส่งข้อมูลตั้งแต่เริ่มต้นการสื่อสารไปจนยุติการสื่อสาร เช่น การติดต่อขอใช้โฮสต์จากเครื่อง

คอมพิวเตอร์ที่อยู่ไกลออกไป (Remote login) หรือการส่งแฟ้มข้อมูลระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายโดยภาพรวมแล้วการให้บริการในชั้นนี้จะคล้ายกับบริการที่มีให้ในชั้นนำส่งข้อมูลแต่ในชั้นนี้จะให้บริการหลายอย่างที่เป็นประโยชน์มากว่าสำหรับการประยุกต์ใช้งานบางประเภท

หน้าที่สำคัญอย่างหนึ่งคือบริหารการแลกเปลี่ยนข่าวสาร (Dialogue control) อันได้แก่การกำหนดให้การแลกเปลี่ยนข่าวสารเป็นไปแบบสองทางในเวลาเดียวกัน (Full duplex) หรือถ้าเป็นการสื่อสารแบบทางเดียวแต่สลับทิศได้ (Half duplex) ก็จะต้องเป็นผู้จัดลำดับให้ทั้งผู้ใช้ และผู้ส่งทำการส่งข้อมูลได้คล้ายกับการควบคุมสับหลีกรถไฟ

สำหรับการสื่อสารประเภทที่ต้องใช้ โทเก้น (Token) โปรแกรมในชั้นนี้จะเป็นผู้บริหารการใช้โทเก้นเพื่อให้โหนดต่างๆ ในระบบนั้นผลัดเปลี่ยนการครอบครองโทเก้นอย่างเป็นธรรม หรือถูกต้องตามลำดับความสำคัญ (priority)

หน้าที่อีกประการหนึ่งได้แก่การแก้ปัญหาความล้มเหลวในการส่งข้อมูลขนาดใหญ่มากระหว่างโหนดต่างๆ ในกรณีที่การส่งข้อมูลเกิดล้มเหลวกลางคนโดยไม่มีการแก้ไขใดๆ โหนดทั้งสองก็จะต้องเริ่มต้นใหม่หมด ถ้าเกิดการล้มเหลวขึ้นอีกก็จะต้องเริ่มต้นใหม่อีก วิธีการแก้ไขวิธีหนึ่งคือการแทรกจุดตรวจสอบความถูกต้อง (Checkpoints) เข้าไปจำนวนหนึ่ง (ขึ้นอยู่กับปริมาณข้อมูล) ในระหว่างการส่งข้อมูล จุดตรวจสอบทั้งหมดจะต้องถูกแทรกเข้าไปในข้อมูลที่ตำแหน่งเดียวกันของทั้งผู้ส่ง และผู้รับซึ่งเรียกว่าการ synchronization หากเกิดการล้มเหลวขึ้นโปรแกรมในชั้นนี้ของผู้รับก็จะค้นหาจุดตรวจสอบจุดสุดท้ายก่อนการล้มเหลวเพื่อลบข้อมูลส่วนที่อยู่หลังจุดตรวจสอบนั้นทิ้งไป แล้วแจ้งให้ผู้ส่งเริ่มต้นการส่งข้อมูลใหม่จากจุดตรวจสอบนั้นแทนที่จะต้องเริ่มต้นใหม่ทั้งหมด

ชั้นสื่อสารนำเสนอข้อมูล (Presentation layer)

โปรแกรมที่ทำงานในระดับชั้นควบคุมต้นๆ ที่กล่าวมานั้นจะให้ความสนใจในประสิทธิภาพของการรับ-ส่งข้อมูล และมองเห็นว่าข้อมูลคือกระแสบิต (Bit stream) หรือกระแสไบต์ (Byte stream) เท่านั้น โปรแกรมในชั้นนำเสนอข้อมูลจะมองข้อมูลว่าเป็นสิ่งที่มีรูปแบบ (Syntax) และความหมาย (Semantics) มากกว่ากระแสของบิตหรือไบต์ เช่น ข้อมูลที่เป็นตัวเลขบอกจำนวนเงิน ข้อมูลที่เป็นชื่อ และข้อมูลที่เป็นใบเรียกเก็บเงิน เป็นต้น ความแตกต่างของการให้ความหมายข้อมูลของเครื่องคอมพิวเตอร์ในระบบต่างๆ เป็นปัญหาที่จะต้องได้รับการแก้ไขในระดับส่วนรวมไม่ใช่ให้แต่ละฝ่ายแก้ปัญหาโดยลำพัง การควบคุมรูปแบบ และความหมายของข้อมูล การใช้รหัสแทนข้อมูล เช่น รหัส ASCII หรือ Unicode หรือการแทนข้อมูลด้วยระบบ Little endian หรือ Big endian

รวมถึงการเข้ารหัส และถอดรหัสข้อมูลสิ่งต่างๆ ที่กล่าวมานี้ล้วนแต่เป็นความรับผิดชอบของโปรแกรมในชั้นนี้

ชั้นสื่อสารการประยุกต์ (Application Layer)

ในปัจจุบัน มีจอภาพเทอร์มินัล (Terminals) อยู่หลายร้อยชนิดทั่วโลกซึ่งส่วนใหญ่จะไม่สามารถใช้ทดแทนหรือใช้งานร่วมกันได้ การติดต่อระหว่างเครื่องคอมพิวเตอร์ที่อยู่คนละระบบเครือข่ายย่อยจึงไม่อาจสื่อสารกันได้โดยสมบูรณ์ โปรแกรมในชั้นการประยุกต์จึงเข้ามามีบทบาทสำคัญสองด้านคือ การเป็นตัวกลาง หรือส่วนติดต่อระหว่างโปรแกรมประยุกต์ (Application programs) กับโปรแกรมใน 6 ชั้นที่เหลือ และการกำหนดแบบมาตรฐานของจอ (Terminal type)

การกำหนดแบบมาตรฐานของจอนั้นไม่ได้เป็นการกำหนดวิธีสร้างจอเทอร์มินัลให้เหมือนกันแต่จะคล้ายกับการสร้างจอเทอร์มินัลเสมือน (Virtual terminal) ขึ้นบนจอเทอร์มินัลจริงทั้งนี้เพื่อทำให้จอเทอร์มินัลทุกชนิดในโลกมีความเข้าใจตรงกันเช่น ขนาดบริเวณที่ในการแสดงผล การเคลื่อนย้ายตำแหน่งเคอร์เซอร์ (Cursor) และการแสดงตัวอักษร ณ ตำแหน่งต่างๆ บนจอภาพเป็นต้น จึงทำให้การใช้จอเทอร์มินัลเพื่อการสื่อสารบนระบบเครือข่ายเกิดขึ้นได้แม้ว่าจะใช้จอเทอร์มินัลต่างแบบกันก็ตาม

การสำเนาแฟ้มข้อมูลหรือการคัดลอก (Copy) แฟ้มข้อมูลผ่านระบบเครือข่ายก็อาจเกิดปัญหาได้ตัวอย่าง เช่น ในระบบปฏิบัติการหลายระบบมีวิธีการกำหนดชื่อแฟ้มข้อมูลที่แตกต่างไปจากวิธีที่ใช้ในระบบอื่น จึงจำเป็นจะต้องมีการแก้ไขหรือปรับแต่งส่วนที่ต่างกันให้สามารถเข้ากันได้ ถ้าหากปล่อยให้เป็นหน้าที่ของผู้ใช้โดยตรงแล้วก็จะทำให้เกิดความยุ่งยากมาก ทั้งนี้เพราะผู้ใช้บางส่วนอาจไม่มีความรู้มากพอที่จะแก้ไขได้ นอกจากนี้การส่งจดหมายอิเล็กทรอนิกส์ การใช้เทอร์มินัลสำหรับป้อนข้อมูลจากระยะไกล หรือการดูบัญชีรายชื่อแฟ้มข้อมูล (ในเครื่องคอมพิวเตอร์อื่น ล้วนเป็นหน้าที่ของโปรแกรมชั้นการประยุกต์ที่จะต้องอำนวยความสะดวกให้แก่ผู้ใช้ทั้งสิ้น

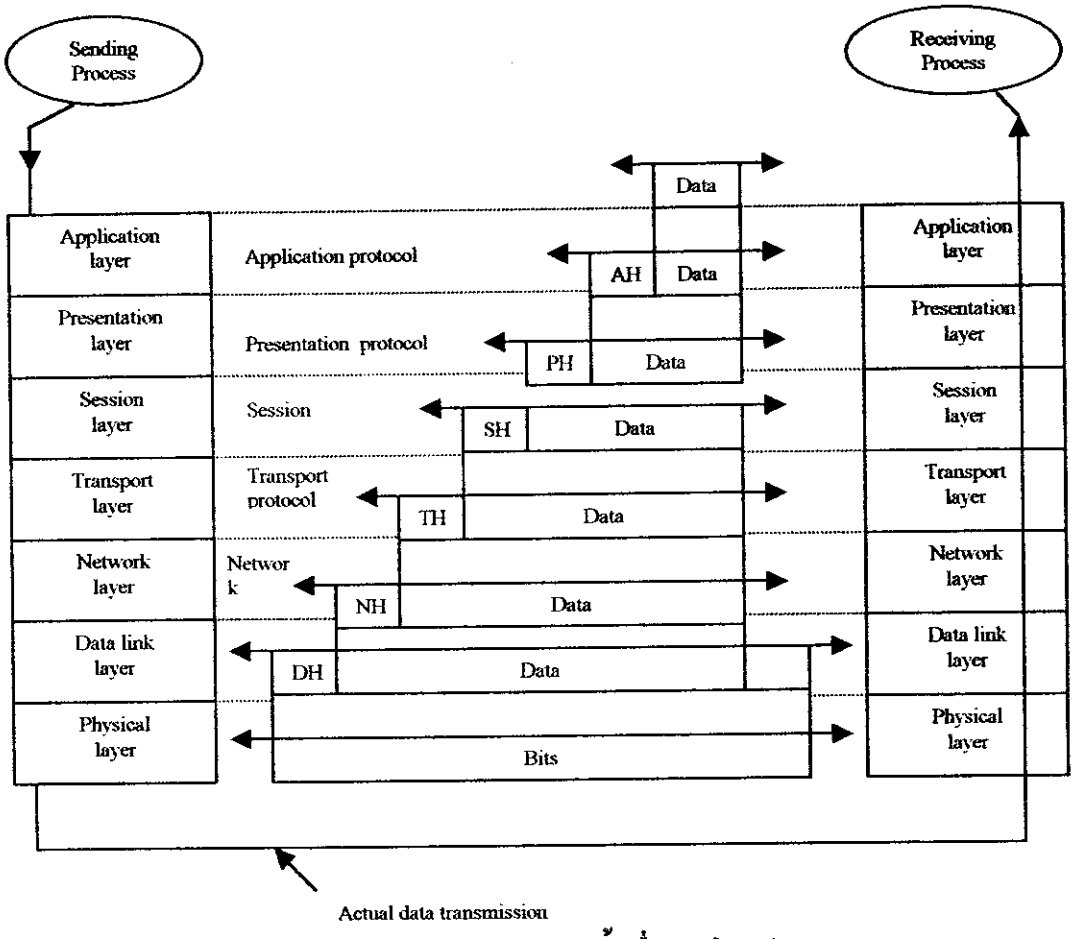
การส่งข้อมูลในรูปแบบมาตรฐานของ OSI

รูป 2.2 แสดงให้เห็นขั้นตอนที่ข้อมูลได้ถูกส่งจากผู้ส่งข้อมูลไปให้ผู้รับข้อมูล กระบวนการส่งข้อมูลเริ่มต้นจากโพรเซสที่ต้องการส่งข้อมูลส่งมอบข้อมูลให้กับโปรแกรมชั้นการประยุกต์ซึ่งจะเติมข้อมูลส่วนหัว (คือส่วน AH ในรูป 2.2) ที่ต้องการเข้าไปกับข้อมูลจริงแล้วส่งต่อไปให้โปรแกรมชั้นนำเสนอข้อมูล โปรแกรมในชั้นนี้จะเข้าใจว่าข้อมูลที่ส่งมาทั้งหมดนั้นคือข้อมูลจริง (จะไม่ทราบว่า มีข้อมูลส่วน AH แทรกเข้ามาด้วยแล้ว) จึงเพิ่มเติมข้อมูลในส่วนที่ตนเองต้องการลงไปกับข้อมูลทั้งหมด (คือส่วน PH ในรูป 2.2) แล้วส่งต่อให้กับโปรแกรมในชั้นกำหนดหน้าต่างสื่อสาร กระบวน

การนี้จะทำซ้ำไปเรื่อยๆ เมื่อผ่าน โปรแกรมในแต่ละชั้นสื่อสารจนถึงชั้นกายภาพซึ่งจะมองเห็นว่าข้อมูลที่ส่งมาจากชั้นเชื่อมต่อข้อมูลนั้นเป็น เพียงกระแสบิต (Bit stream) ที่จะต้องส่งออกไปเท่านั้น

ทางด้านผู้ใช้ข้อมูลจะเริ่มกระบวนการรับข้อมูลโดยชั้นกายภาพรับข้อมูลเข้ามาในลักษณะของกระแสบิตแล้วส่งต่อไปให้โปรแกรมชั้นเชื่อมต่อข้อมูล โปรแกรมนี้จะอาศัยข้อมูลส่วนหัว (DH) เพื่อทำความเข้าใจกับลักษณะของข้อมูลที่รับมารวมทั้งการตรวจสอบความถูกต้อง เสร็จแล้วจึงลบข้อมูลส่วนนี้ออกไป และส่งข้อมูลที่เหลือไปให้กับโปรแกรมในชั้นควบคุมเครือข่าย กระบวนการก็จะเป็นไปอย่างนี้จนในที่สุดข้อมูลก็ถูกส่งมาถึงชั้นการประยุกต์ ในชั้นนี้ข้อมูลส่วนหัวที่โปรแกรมในแต่ละชั้น (ของผู้ส่ง) เพิ่มเติมเข้าไปนั้นได้ถูกโปรแกรมในชั้นนั้นๆ (ของผู้ใช้) ดึงออกไปใช้จนเหนือแต่ข้อมูลส่วนหัวของชั้นการประยุกต์ (AH) เท่านั้น โปรแกรมในชั้นนี้ก็จะดึงออกไปใช้จนเหลือแต่ข้อมูลส่วนหัวของชั้นการประยุกต์ (AH) เท่านั้นโปรแกรมในชั้นนี้ก็จะดึงข้อมูลส่วนสุดท้ายนี้ออกไปใช้งานแล้วจึงส่งข้อมูลที่เหลืออยู่ (คือข้อมูลจริง) ให้กับโพรเซสของผู้รับนำข้อมูลไปใช้งานต่อไป

แนวความคิดที่เป็นหลักสำคัญของกระบวนการนี้คือ การที่โปรแกรมในแต่ละชั้นนั้นคิดว่าการสื่อสารเป็นไปตามแนวนอน แม้ว่าในความเป็นจริงนั้นจะเป็นไปตามแนวตั้งดังที่แสดงในรูป 2.2 ทั้งนี้หมายความว่าโปรแกรมในแต่ละชั้นของทางด้านผู้ส่งจะแทรกข้อมูลที่ตนเองต้องการเข้าไปกับข้อมูลโดยมีวัตถุประสงค์ให้โปรแกรมในชั้นเดียวกันของทางผู้รับได้นำข้อมูลนั้นไปใช้ จึงเสมือนกับว่าโปรแกรม ในแต่ละชั้นของทางผู้ส่งจะทำงานกับโปรแกรมในชั้นเดียวกันของทางฝ่ายผู้รับ โดยไม่สนใจว่าจะมีโปรแกรมในชั้นอื่นๆ อยู่ด้วย การทำงานของโปรแกรมในแต่ละชั้นจึงเป็นอิสระจากโปรแกรมในชั้นอื่นๆ อย่างแท้จริง และจะปฏิบัติกับข้อมูลที่รับมาราวกับว่าเป็นข้อมูลจริงทั้งหมด (แม้ว่าจะมีข้อมูลของชั้นบนๆ แทรกเข้ามาด้วยก็ตาม)



รูป 2.2 ตัวอย่างการทำงานของชั้นสื่อสารในรูปแบบ OSI

(ที่มา: Computer Network, Andrew S. Tanenbaum)

2.1.2 รูปแบบระบบเครือข่าย TCP/IP

ระบบเครือข่ายระดับโลกที่มีใช้อยู่ในปัจจุบันคือระบบอินเทอร์เน็ตนั้นมีกำเนิดมาจากระบบเครือข่ายชื่อ APPANET ซึ่งได้รับการสนับสนุนให้ดำเนินการวิจัยโดยมีกระทรวงกลาโหมประเทศสหรัฐอเมริกาเป็นผู้ออกค่าใช้จ่าย ในยุคแรกๆ นั้นเป็นระบบเครือข่ายที่เชื่อมการติดต่อระหว่างเครื่องคอมพิวเตอร์ในมหาวิทยาลัย และสถานที่ราชการหลายร้อยแห่งในสหรัฐอเมริกาเข้าด้วยกัน โดยใช้สายโทรศัพท์เช่า (Leased lines) เป็นสายสื่อสารหลัก ต่อมาเมื่อระบบการสื่อสารแบบคลื่นวิทยุความถี่สูง และการสื่อสารดาวเทียมเริ่มเข้ามามีบทบาท และนำมาใช้ในระบบมากขึ้น ทำให้กฎการสื่อสาร (Protocol) ที่เคยใช้ได้ผลดีนั้นเกิดปัญหาไม่สามารถใช้งานได้อีกต่อไป กฎการสื่อสารรุ่นต่อมาจึงได้รับการออกแบบเพื่อนำมาใช้ทดแทนแบบเก่าโดยมีวัตถุประสงค์ในการเชื่อมการติดต่อระหว่างระบบที่มีความแตกต่างกันเป็นเรื่องหลัก ผลที่ได้รับคือกฎการสื่อสารที่เรียกว่า

กฎสื่อสารมาตรฐานแบบ TCP/IP ซึ่งได้รับการปรับปรุงจนนำมาใช้งานจริงได้ในปี ค.ศ. 1974 การปรับปรุงรุ่นต่อมาสำเร็จในปี ค.ศ. 1988 รายละเอียดสามารถหาอ่านได้ใน (Clark, 1988)

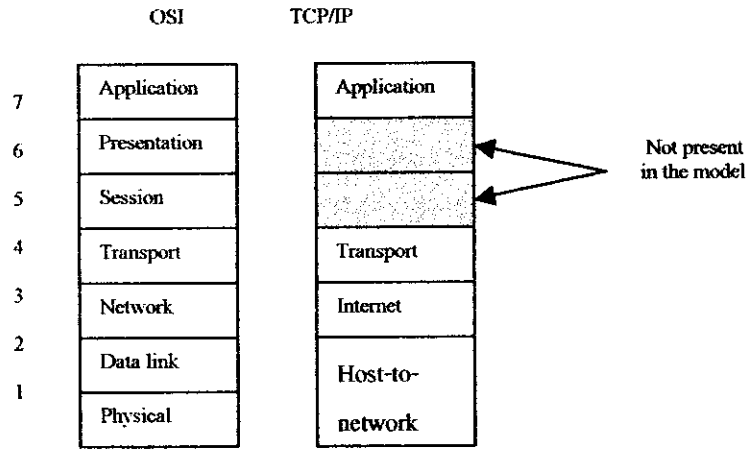
กฎสื่อสารมาตรฐานแบบ TCP/IP ยังมีวัตถุประสงค์หลักอีกสองข้อสำคัญคือ ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่ง และผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งข้อมูลเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดการหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ ข้อที่สองคือจะต้องมีความอ่อนตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ Real-time หรือการสื่อสารแบบโทรศัพท์ (Voice)

ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

จากความต้องการทั้งหมดที่กล่าวถึงทำให้เกิดเป็นวิธีการส่งข้อมูลแบบหนึ่งเรียกว่าระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็กเก็ต (Packet-switching network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่าแพ็กเก็ตสามารถไหลจากโหนดผู้ส่งแพ็กเก็ตไปตามโหนดต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ หากว่ามีการส่งแพ็กเก็ตออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่าย แพ็กเก็ตแต่ละตัวในชุดนี้ก็จะเป็นอิสระแก่กันและกัน ดังนั้นแพ็กเก็ตที่มาถึงจุดหมายเป็นตัวแรกอาจไม่ใช่แพ็กเก็ตที่ถูกส่งออกมาเป็นตัวแรกก็ได้ จึงเป็นหน้าที่ของโปรแกรมในชั้นสูงขึ้นไปที่จะต้องรวบรวมแพ็กเก็ตทั้งหมดแล้วจัดเรียงลำดับให้ถูกต้องก่อนที่จะส่งมอบให้กับโพรเซสผู้รับข้อมูล

ระบบนี้เปรียบเทียบได้กับระบบไปรษณีย์ระหว่างประเทศ กรมไปรษณีย์ในแต่ละประเทศจะมีการจักระเบียบต่างๆ เป็นของตนเองซึ่งอาจเหมือนหรือไม่เหมือนกับประเทศอื่นก็ได้ คนที่ต้องการส่งจดหมายต้องรับทราบระเบียบต่างๆ เฉพาะบริเวณบังคับใช้โดยกรมไปรษณีย์ในประเทศตนเองเท่านั้นเมื่อส่งจดหมายไปแล้วไปรษณีย์ในประเทศของผู้ส่งก็จะจัดการส่งจดหมายฉบับนั้นไปตามที่อยู่ผู้รับซึ่งอยู่ในประเทศอื่น เป็นไปได้ว่าจดหมายจะได้รับการฝากส่งไปตามไปรษณีย์ของหลายๆ ประเทศก่อนที่จะไปถึงยังไปรษณีย์ของประเทศผู้รับจะเห็นได้ว่าในทันทีที่จดหมายเดินทางออกนอกประเทศไปแล้วนั้นจะไม่อยู่ในความควบคุมของไปรษณีย์ประเทศผู้ส่งอีกต่อไป ในขณะเดียวกันไปรษณีย์ของประเทศต่างๆ ที่จดหมายเดินทางผ่านอาจมีกฎระเบียบต่างกันออกไป เช่น มีแสตมป์เป็นของตนเอง แต่เนื่องจากได้มีการตกลงกันไว้แล้วจึงยอมรับและช่วยส่งจดหมายไปยังจุดต่อไปให้ กฎระเบียบที่แตกต่างกันนี้ ผู้ส่งจดหมายไม่มีความจำเป็นที่จะต้องรับทราบเลย

ชั้นสื่อสารบนอินเทอร์เน็ตกำหนดรูปแบบแพ็กเก็ตและกฎการสื่อสารเรียนว่า IP (Internet Protocol) ดังแสดงในรูป 2.3 ซึ่งเปรียบเทียบ TCP/IP กับรูปแบบระบบเครือข่าย OSI งานสำคัญของ TCP/IP คือการนำแพ็กเก็ตไปส่งยังจุดหมายปลายทางให้ได้



รูป 2.3 รูปแบบโครงสร้าง TCP/IP

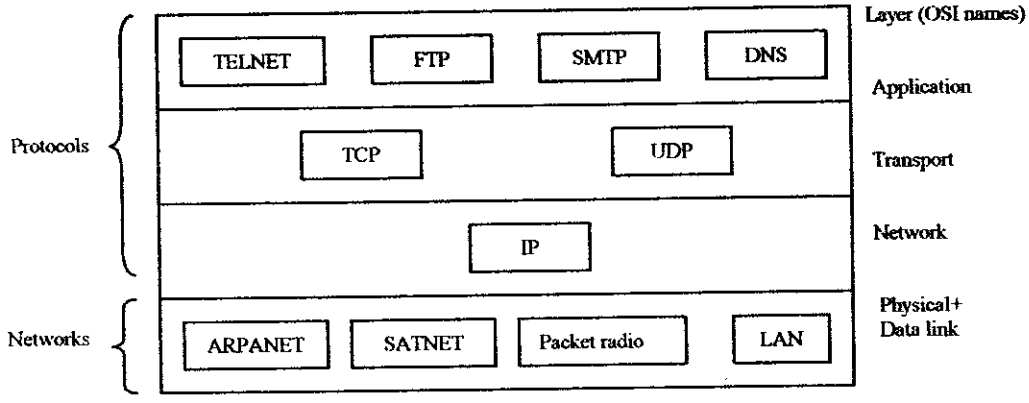
(ที่มา: Computer Network, Andrew S. Tanenbaum)

ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

ชั้นสื่อสารที่อยู่เหนือชั้น IP คือชั้นนำส่งข้อมูลซึ่งมีหน้าที่การทำงานเหมือนกันกับชั้นจัดการนำส่งข้อมูลของมาตรฐาน OSI แบ่งออกเป็นโพรโตคอลสองประเภท ประเภทแรกเรียกว่า TCP (Transmission Control Protocol) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (Connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลในเป็นกระแสไบต์ (Byte stream) ที่ไว้ใจได้ (Reliable) โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า Message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ Message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่งส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

กฎการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) ไม่มีการตรวจสอบความถูกต้องของข้อมูลเหมือนกับแบบ TCP อย่างไรก็ตามวิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูลจึงนิยมใช้ในระบบผู้ให้ และผู้ใช้บริการ (Client/Server system) ซึ่งมีการสื่อสารแบบถาม/ตอบ (Request/Reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (Voice) ทางอินเทอร์เน็ต รูป 2.4 แสดงให้เห็นความ

สัมพันธ์ของ IP, TCP และ UDP กฎการสื่อสารแบบ IP นี้เป็นระบบบที่ได้รับความนิยมมากและได้รับการนำไปใช้ในระบบสื่อสารหลายระบบ



รูป 2.4 กฎการสื่อสารในรูปแบบ TCP/IP

(ที่มา: Computer Network, Andrew S. Tanenbaum)

ชั้นสื่อสารการประยุกต์ (Application Layer)

ตามมาตรฐาน TCP/IP ไม่มีการกำหนดชั้นนำเสนอข้อมูลและชั้นควบคุมหน้าต่างสื่อสารตามที่ปรากฏในรูปแบบมาตรฐาน OSI ทั้งนี้จากประสบการณ์ที่ผ่านมาของการกำหนดมาตรฐานของ OSI พบว่าโปรแกรมสำหรับชั้นควบคุมการสื่อสารสองชั้นนี้จะมีประโยชน์ในการใช้งานจริงน้อยมากระบบเครือข่าย TCP/IP จึงตัดทั้งสองชั้นนี้ออกไป ดังนั้นชั้นการประยุกต์จึงกลายเป็นชั้นที่อยู่เหนือชั้นนำส่งข้อมูล

โปรแกรมในชั้นการประยุกต์จะรวมเอาหน้าที่การทำงานที่จะเป็นของสองชั้นที่หายไปไว้ที่นี่ อันได้แก่ โพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โพรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า FTP และโพรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP ดังรูป 2.4 โพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไปโดยผ่านทางอินเทอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้น โพรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่าย หรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆ ก็ได้ โพรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

นอกจากโพรโตคอลที่กล่าวถึงแล้วยังมีโพรโตคอลอีกจำนวนมากที่ได้รับการพัฒนาขึ้นมาใช้งาน เช่น โพรโตคอลสำหรับการบริหารที่อยู่ของโหนดต่างๆ ในระบบเครือข่าย เรียกว่า DNS (Domain Name Service) จะช่วยในการเก็บสำเนารายชื่อของโหนดต่างๆ ในระบบที่เป็นทั้งชื่อในภาษาอังกฤษที่ผู้ใช้ทั่วไปรู้จัก และสามารถจดจำได้ง่าย และชื่อที่เป็นตัวเลขที่ใช้อ้างอิงในระหว่างการสื่อสาร โพรโตคอล NNTP ใช้ในการให้บริการข่าวสาร (news) แก่สมาชิกในระดับกลุ่ม และโพรโตคอล HTTP ซึ่งช่วยในการสร้างรูปแบบหน้าจอขั้นก้าวหน้าซึ่งนิยมใช้ในระบบ WWW (World Wide Web) เป็นต้น

ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

โพรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูลทางด้านผู้รับก็จะทำงานในทางกลับกันคือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร IP

การที่ไม่กำหนดรายละเอียดเอาไว้อาจเนื่องมาจากเหตุผลสามประการ ประการแรกคือ จากประสบการณ์ที่ได้รับในการกำหนดมาตรฐาน OSI พบว่าเทคโนโลยีทางด้านการรับ-ส่งข้อมูลผ่านสื่อประเภทต่างๆนั้นมีการเปลี่ยนแปลงอยู่เสมอ หากมีการกำหนดรายละเอียดก็อาจจะต้องแก้ไขกันอยู่บ่อยๆ ประการที่สองคือ การสื่อสารที่เกิดขึ้นในระดับนี้เป็นการส่งข้อมูลในระดับกระแสบิต (Bit stream) ซึ่งไม่มีความเกี่ยวข้องกับโครงสร้างข้อมูลในระดับบนแต่อย่างใด ประการสุดท้ายการทำงานระดับนี้จะติดต่อหรือมีความสัมพันธ์กับอุปกรณ์สื่อสารข้อมูลโดยตรง ซึ่งอุปกรณ์เหล่านี้ล้วนแล้วแต่มีมาตรฐานสากลรองรับอยู่อย่างเหลือเฟือแล้ว

การเปรียบเทียบระหว่างมาตรฐาน OSI และ TCP

รูปแบบระบบเครือข่าย TCP/IP มีลักษณะทั่วไปคล้ายคลึงกับรูปแบบระบบเครือข่าย OSI อย่างมาก รูปแบบระบบทั้งสองมีรากฐานบนแนวความคิดเดียวกันคือการสร้างชั้นสื่อสารที่เป็นอิสระแก่กัน และกันอย่างสิ้นเชิง รวมทั้งหน้าที่การทำงานในชั้นสื่อสารต่างๆ ก็มีความใกล้เคียงกันมาดังจะเห็นได้จากชั้นสื่อสารที่อยู่ในระดับบนสุดลงมาถึงชั้นจัดการนำส่งข้อมูลจะมีการติดต่อจากผู้ส่งถึงผู้รับแบบเสมือนว่าเป็นการติดต่อกันโดยตรง ซึ่งความแตกต่างของระบบเครือข่ายย่อยต่างๆ จะถูกปรับให้สามารถสื่อสารกันได้อย่างเรียบร้อย ในขณะที่ชั้นสื่อสารที่สูงกว่าชั้นจัดการนำส่งข้อมูลจะมีความเกี่ยวข้องกับโปรแกรมประยุกต์มากกว่าชั้นสื่อสารอื่นๆ

อย่างไรก็ตามมาตรฐานทั้งสองแบบก็มีข้อแตกต่างกันอยู่พอสมควร ในที่นี้จะมุ่งความสนใจไปยังความแตกต่างของมาตรฐานทั้งสองแบบ ส่วนความแตกต่างทางด้านโพรโตคอลนั้นจะได้กล่าวถึงในบทต่อๆ ไป

องค์ประกอบความคิดพื้นฐานที่สำคัญของรูปแบบระบบเครือข่าย OSI คือ

1. ส่วนการให้บริการ (Service)
2. ส่วนการติดต่อ (Interface)
3. โพรโตคอล (Protocols)

วัตถุประสงค์หลักของรูปแบบระบบเครือข่าย OSI คือการแยกขอบเขตความรับผิดชอบและหน้าที่ขององค์ประกอบทั้งสามออกจากกันอย่างเด็ดขาด โดยที่แต่ละชั้นสื่อสารจะทำหน้าที่ให้บริการแก่ชั้นที่อยู่เหนือขึ้นไปหนึ่งขั้น การให้บริการจะบอกให้ทราบว่าชั้นสื่อสารนั้นๆ ให้บริการอะไรบ้าง แต่ไม่ได้บอกรายละเอียดในวิธีการให้บริการและวิธีการติดต่อขอใช้บริการ

ส่วนการติดต่อ เป็นการอธิบายถึงวิธีการที่ผู้ใช้คือโปรแกรมในชั้นสื่อสารที่อยู่เหนือขึ้นไปหนึ่งขั้นจะสามารถเรียกใช้บริการที่มีอยู่ได้อย่างไร โดยจะต้องมีการกำหนดชื่อบริการ ประเภทและจำนวนพารามิเตอร์ หรือข้อมูลที่ส่งเข้ามา และเภทและจำนวนของผลลัพธ์

กฎการสื่อสาร หรือโพรโตคอล เป็นการกำหนดรายละเอียดการทำงานของบริการแต่ละชนิดซึ่งจะมีอิสระเต็มที่ในการเลือกวิธีการทำงานอย่างไรก็ได้เพื่อทำงานให้บรรลุตามวัตถุประสงค์ที่กำหนดไว้

องค์ประกอบที่กล่าวถึงนี้ แม้ว่าจะเกิดขึ้นมานานมากแล้ว เมื่อเปรียบเทียบกับเทคโนโลยีใหม่ในปัจจุบัน เช่น เทคโนโลยีการโปรแกรมเชิงวัตถุ (Object-oriented programming) กลับมีความสอดคล้องกันอย่างกลมกลืน นั่นคือ วัตถุ (Object) สามารถเปรียบเทียบได้กับชั้นสื่อสารแต่ละชั้นซึ่งจะมีฟังก์ชันในการบริการต่างๆ ที่วัตถุอื่น (ชั้นสื่อสารที่อยู่เหนือไปอีกชั้นหนึ่งชั้น) สามารถเรียกใช้ได้ วัตถุมีการกำหนดความหมายของฟังก์ชันที่ให้บริการ ตัวแปรสำหรับฟังก์ชัน รวมทั้งผลลัพธ์ที่ได้ ซึ่งก็เปรียบได้กับส่วนการติดต่อ ท้ายที่สุดกฎการสื่อสารหรือโพรโตคอลก็คือการกำหนดรายละเอียดคำสั่งต่างๆ สำหรับแต่ละฟังก์ชันซึ่งจะไม่มีวัตถุอื่นเข้ามาก้าวก่ายนอกจากตัววัตถุนั้นเอง

ในส่วนของรูปแบบ TCP/IP นั้นไม่ได้มีการแยกความหมายและหน้าที่ของส่วนการให้บริการส่วนการติดต่อ และโพรโตคอลให้ชัดเจนตั้งแต่เริ่มต้น มีความพยายามที่จะกำหนดนิยามที่ชัดเจนให้กับรูปแบบ TCP/IP ให้เหมือนกับรูปแบบ OSI แต่ก็สามารถทำได้กับบริการเพียงสองชนิดคือ บริการส่ง IP แพ็กเก็ต และบริการรับ IP แพ็กเก็ต ความแตกต่างในข้อนี้แสดงให้เห็นว่า OSI นั้นมีการจัดโครงสร้างภายในที่ดีกว่า TCP/IP ซึ่งทำให้การหาโปรแกรมที่ใช้เทคโนโลยีใหม่มาทดแทนโปรแกรมที่ใช้อยู่เดิมในระบบ OSI นั้นสามารถทำได้ง่ายกว่ามาก

รูปแบบของ OSI ได้รับการออกแบบมาก่อนที่โพรโตคอลหลายอย่างจะได้รับการออกแบบที่สมบูรณ์ และสร้างโปรแกรมขึ้นมาใช้งานจริง รูปแบบของ OSI จึงได้รับการออกแบบมาอย่างมีอิสระทางด้านความคิดอย่างเต็มที่โดยไม่ผูกติดกับแนวความคิดของโพรโตคอลใดๆ อย่างไรก็ตามผลเสียที่เกิดขึ้นจากการไม่คำนึงถึงโพรโตคอลอื่นๆ ทำให้การกำหนดหน้าที่การทำงานหลายส่วนไม่มีความเหมาะสมในการใช้งานจริง ตัวอย่างที่เกิดขึ้นได้แก่ปัญหาในชั้นเชื่อมต่อข้อมูล แรกทีเดียวชั้นสื่อสารนี้กำหนดไว้สำหรับการสื่อสารแบบจุด-ต่อ-จุด ต่อมาวิธีการสื่อสารแบบกระจายข่าวเริ่มเป็นที่นิยมมากขึ้น แต่ก็ไม่อาจจะนำเทคโนโลยีการสื่อสารแบบนี้แทรกเข้าไปในรูปแบบ OSI คาดว่าจะมีระบบเครือข่ายย่อยเพียงหนึ่งเครือข่ายภายในแต่ละประเทศ แต่ความจริงที่ปรากฏก็คือประเทศที่มีประสบการณ์การใช้งานระบบเครือข่ายจะจัดตั้งระบบเครือข่ายย่อยขึ้นใช้งานมากมายทำให้วิธีการตั้งชื่อโฮสต์และโหนดต่างต้องได้รับการปรับปรุงใหม่

ทางด้าน TCP/IP นั้นกลับอยู่ในสถานการณ์ตรงกันข้าม นั่นคือโพรโตคอลต่างๆ ได้รับการพัฒนามาเป็นอย่างดีก่อนที่รูปแบบเครือข่าย TCP/IP จะได้รับการพัฒนาขึ้นมาใช้งาน ดังนั้นการผสมผสานโพรโตคอลต่างๆ เข้ากับรูปแบบ TCP/IP จึงทำได้อย่างง่ายดาย อย่างก็ตามความอ่อนตัวที่มีอยู่นี้ทำให้รูปแบบ TCP/IP ไม่มีการกำหนดโครงสร้างที่ชัดเจน การเปรียบเทียบ TCP/IP กับรูปแบบเครือข่ายอื่นๆ จึงเป็นเรื่องที่เป็นไปไม่ได้

ข้อแตกต่างที่เป็นรูปธรรมของทั้งสองรูปแบบนี้คือรูปแบบ OSI ประกอบด้วยชั้นสื่อสารเจ็ดชั้นในขณะที่ TCP/IP มีเพียงสี่ชั้น นอกจากนั้นรูปแบบ OSI ให้บริการส่งข้อมูลแบบไม่ต่อเนื่องและแบบต่อเนื่องในชั้นควบคุมเครือข่าย แต่ให้บริการเฉพาะแบบต่อเนื่องในชั้นนำส่งข้อมูล ส่วนรูปแบบTCP/IP ให้บริการแบบไม่ต่อเนื่องในชั้นควบคุมเครือข่าย แต่ให้บริการทั้งสองชนิดในชั้นนำส่งข้อมูล เนื่องจากการให้บริการในชั้นนำส่ง ข้อมูลเป็นบริการที่ผู้ใช้สามารถนำไปใช้ได้ รูปแบบ TCP/IP จึงเปิดโอกาสให้ผู้ใช้สามารถเลือกวิธีการส่งข้อมูลที่ต้องการได้ ในขณะที่ผู้ใช้ OSI ไม่มีทางเลือก ซึ่งวิธีการติดต่อนี้มีส่วนสำคัญคือ การใช้โพรโตคอลแบบร้องขอ-และ-ตอบสนอง (Request-response) เป็นอย่างมาก

2.2 ทฤษฎีการสื่อสารบนอินเทอร์เน็ต

ก่อนที่คอมพิวเตอร์จะติดต่อสื่อสารกันได้ทั้ง 2 เครื่องจะต้องใช้ภาษาเดียวกันก่อน โดยเครื่องคอมพิวเตอร์ทุกเครื่องที่ทำการติดต่อกับอินเทอร์เน็ตนั้นจะมีการใช้โพรโตคอล TCP/IP เพื่อให้คอมพิวเตอร์ทุกรุ่น ทุกแบบ สามารถติดต่อสื่อสารกันได้อย่างถูกต้องและเมื่อคอมพิวเตอร์ต่างๆ พูดคุยกันรู้เรื่องโดยใช้โพรโตคอลเดียวกัน การติดต่อสื่อสารก็จะเริ่มขึ้นโดยการติดต่อสื่อสารจะมีอยู่ 2 แบบคือแบบส่งข้อมูล และแบบรับข้อมูล โดยเครื่องคอมพิวเตอร์ที่ทำการส่งข้อมูลเราจะเรียก

ว่าเครื่องให้บริการ (Server) และเครื่องคอมพิวเตอร์ที่รับข้อมูลเราจะเรียกว่าเครื่องรับบริการ (Client) ซึ่งการติดต่อในลักษณะนี้เราเรียกว่าการติดต่อแบบ Client-Server

2.2.1 โดเมนเนม (Domain Name)

ทุกๆ สถานที่ในอินเทอร์เน็ตจะมี "ที่อยู่" เฉพาะ ฉะนั้นการที่จะเข้าไปชมเว็บเพจนั้นก็จำเป็นที่จะต้องทราบที่อยู่ของเว็บนั้นเสียก่อน ซึ่งจะมีลักษณะเฉพาะที่เรียกว่า URL (Uniform Resource Locator) ดังนั้นแต่ละเว็บเพจจึงขึ้นต้นด้วย http:// ซึ่งย่อมาจาก Hypertext Transfer Protocol

ส่วนโดเมนเนมก็คือชื่อสำหรับไว้เรียกเว็บไซต์ที่จะเข้าไปซึ่งมักจะพบในรูปแบบ เช่น www.yahoo.com หรืออาจเทียบโดเมนเนมเหมือนบ้านเลขที่เพราะแต่เดิมนั้นโดเมนเนมจะเป็นตัวเลข เช่น 123.456.7.8 แต่ตัวเลขยากต่อการจำจึงได้มีการใช้ตัวอักษรแทนชุดของตัวเลขซึ่งทำให้ง่ายต่อการใช้งานและจำง่ายด้วย โดยโดเมนเนมจะแยกรายละเอียดได้ดังนี้

www.***.com หมายถึง Top-Level-Domain Name

www.***.com หมายถึง ชื่อโดเมนที่ขอจดไว้ (Second-Level-Domain Name)

www.***.com หมายถึง สับย่อย ของโดเมนเนม (Sub Domain Name)

ซึ่งปกติถ้าไม่มีการสร้างสับย่อยเพิ่มก็มักจะเป็นแบบ www.***.com แต่ถ้ามีการสร้างเพิ่มขึ้นก็จะเป็นแบบ yourname.***.com

หรือ yourcompany.***.com

หรือ www.yourcompany.***.com

2.2.2 การกำหนดหมายเลขไอพี

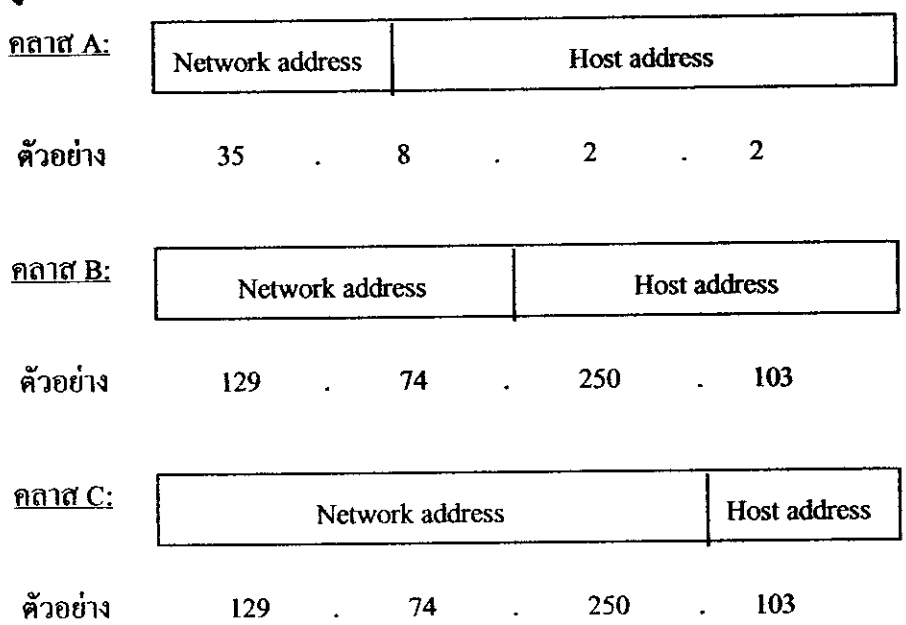
ถึงแม้การใช้โดเมนเนมจะเป็นวิธีที่ดีที่สุดในอินเทอร์เน็ตยุคปัจจุบันแต่ความเข้าใจเรื่องวิธีกำหนดหมายเลขไอพีเวิร์กสเตชันเข้าสู่อินเทอร์เน็ตได้ และในการค้นข้อมูลในอินเทอร์เน็ต

หมายเลขไอพีประกอบด้วยเลขสี่ส่วน หมายเลขนี้หมายถึงคอมพิวเตอร์เครื่องใดเครื่องหนึ่งบนอินเทอร์เน็ต วิธีตั้งหมายเลขไอพีใช้แนวทางที่ทำให้เครือข่ายขนาดใหญ่จัดการหมายเลขจำนวนมาก และเครือข่ายขนาดเล็กจัดการเฉพาะส่วนเล็กๆ ใน "หมายเลขไอพีทั้งหมด" คล้ายกับในเรื่องของบริการโทรศัพท์ ที่บริษัทโทรศัพท์อนุญาตให้องค์กรที่ใหญ่มากออกหมายเลขโทรศัพท์ของทั้งจังหวัด และให้องค์กรเล็กออกหมายเลขโทรศัพท์ในบริเวณของชุมสายหนึ่งๆ

หมายเลขไอพีแบ่งเป็นระดับหรือคลาส ส่วนซ้ายสุดของหมายเลขไอพีเป็นตัวกำหนดคลาส ถ้ามีค่าตั้งแต่ 0 ถึง 127 แสดงว่าเครื่องนั้นอยู่ในระดับเครือข่ายคลาส A ตามทฤษฎีเครือข่ายคลาส A

มีโฮสต์คอมพิวเตอร์ได้ถึง 16,777,216 เครื่อง เครือข่ายระดับถัดมาคือคลาส B มีหมายเลขไอพีเริ่มจาก 128 ถึง 191 ตามทฤษฎีมีโฮสต์ได้ 65,536 เครื่อง หมายเลขในคลาส C เริ่มจาก 192 ถึง 233 แต่ละเครือข่ายมีโฮสต์ได้ถึง 256 เครื่อง รูปที่ 4.2 แสดงการแบ่งหมายเลขไอพีออกเป็นคลาสต่างๆ

อินเทอร์เน็ตเป็นเครือข่ายของเครือข่าย เครือข่ายแต่ละเครือข่ายที่ประกอบกันขึ้นเป็นอินเทอร์เน็ตจะมีผู้ดูแลเครือข่าย (Network administrator) ของตนเอง งานอย่างหนึ่งของผู้ดูแลคือกำหนดหมายเลขไอพีให้กับโฮสต์จากกลุ่มหมายเลขไอพีที่ถูกกำหนดมาให้กับเครือข่ายนั้นๆ ในทางปฏิบัติ ขั้นตอนการกำหนดหมายเลขไอพีสำคัญต่อผู้ดูแลระบบมาก แต่แทบไม่มีความหมายต่อผู้ใช้เลย เมื่อผู้ดูแลเครือข่ายเลือกหมายเลขไอพีสำหรับคอมพิวเตอร์เครื่องใหม่ที่จะเชื่อมเข้ากับเครือข่ายได้แล้ว ผู้ดูแลยังต้องเลือกชื่อโฮสต์แบบโดเมนด้วย จากนั้นต้องบันทึกทั้งหมายเลขไอพี และชื่อโฮสต์ (รวมทั้งข้อมูลอื่นที่จำเป็น) ลงในเซิร์ฟเวอร์บริการชื่อโดเมนสำหรับเครือข่ายนั้น เมื่อเสร็จแล้วผู้ใช้ก็สามารถใช้หมายเลขไอพีและเริ่มใช้ TCP/IP ได้



รูปที่ 2.5 คลาสของเครือข่ายบนอินเทอร์เน็ต

(ที่มา: The internet for everyone, Richard W.Wiggins)

2.2.3 TCP/IP ส่งข้อมูลให้คุณได้อย่างไร

กระบวนการหรือขั้นตอนในการย้ายข้อมูลจากคอมพิวเตอร์ของคุณไปยังคอมพิวเตอร์อีกเครื่องบนเครือข่ายที่คุณใช้หรือบนอินเทอร์เน็ตเป็นเรื่องที่ค่อนข้างซับซ้อน และรายละเอียดว่าขั้น

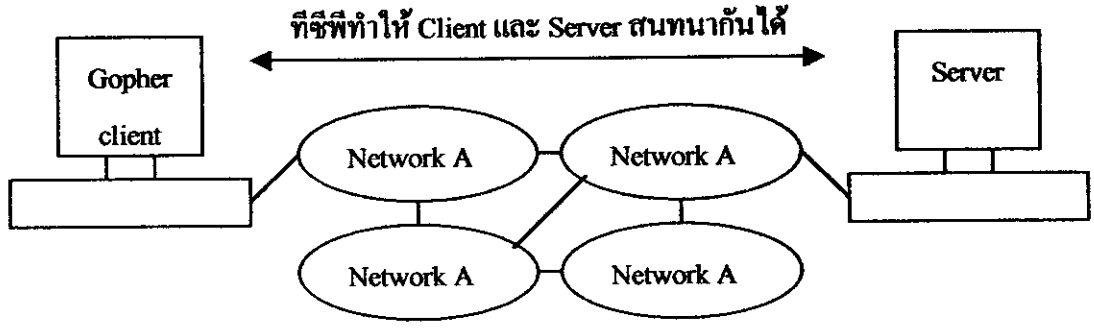
ตอนนี้ทำงานอย่างไรเป็นเรื่องยุ่งยากจนดูเหมือนว่าถ้าเอาข้อความที่ต้องการส่งพร้อมกับที่อยู่ปลายทางผูกติดกับลูกโป่งและปล่อยให้ลอยไปจะง่ายกว่า ถึงแม้ขั้นตอนนี้จะซับซ้อนแต่เชื่อถือได้

โพรโตคอลหลักที่ใช้กันอยู่บนอินเทอร์เน็ตมีชื่อว่า TCP/IP ความจริง ทีซีพี และไอพีเป็นโพรโตคอลสองโพรโตคอลที่มีหน้าที่ต่างกัน หน้าที่ของทีซีพีคือสร้างกลไกในการส่งข้อมูลไปมาระหว่างคอมพิวเตอร์ที่เชื่อถือได้ ซึ่งหมายถึงว่า ตัวข้อมูลที่ส่งผ่านมาต้องถูกต้องครบถ้วนสมบูรณ์ เรียงลำดับข้อมูลอย่างถูกต้อง และไม่มีการซ้ำซ้อนของข้อมูล หน้าที่ของไอพีคือเคลื่อนย้ายกลุ่มของข้อมูลที่เรียกว่า "ดาต้าแกรม" (Datagram) ให้ผ่านเครือข่ายที่โยงใยกันอย่างซับซ้อน (เรียกว่า "อินเทอร์เน็ตเวิร์ก", Internetwork) ซึ่งคั่นระหว่างคอมพิวเตอร์สองเครื่องที่ต้องการติดต่อกันอยู่

เมื่อคอมพิวเตอร์เครื่องหนึ่งต้องการสนทนากับคอมพิวเตอร์อีกเครื่องบนอินเทอร์เน็ต คอมพิวเตอร์จะเริ่มเปิดการเชื่อมต่อทีซีพีกับคอมพิวเตอร์อื่น พูดในทางปฏิบัติคือ เมื่อโปรแกรมประยุกต์เช่นเทลเน็ต หรือโกเฟอร์เซิร์ฟเวอร์ มันจะเริ่มเปิดการเชื่อมต่อทีซีพี ขั้นตอนนี้อาจเทียบได้กับการทำงานของเครือข่ายโทรศัพท์ เมื่อคนกดปุ่มโทรศัพท์ โทรศัพท์ไปยังที่อยู่หนึ่ง (หมายเลขโทรศัพท์) จะมีมือลึกลับ (ระบบสวิทชิ่ง) เรียกไปยังที่อยู่นั้นบนเครือข่ายให้ หลังจากที่มีผู้รับโทรศัพท์ คนก็สามารถแลกเปลี่ยนข้อมูลกันได้จนกว่าทั้งคู่ตัดสินใจวางโทรศัพท์

การเชื่อมต่อระหว่างคอมพิวเตอร์สองเครื่องโดยใช้ทีซีพีทำให้สามารถส่งข้อมูลได้พร้อมกันสองทาง อาจนึกภาพว่าขั้นตอนนี้เป็นการส่งข้อมูลสองสายจากปลายด้านหนึ่งไปยังอีกด้าน ข้อมูลในสายหนึ่งวิ่งในทิศตรงข้ามกับอีกสาย ทีซีพียังมีวิธีการให้ด้านหนึ่งบอกกับอีกด้านว่า "นี่! ข้อมูลนี้เป็นข้อมูลสำคัญนะ" วิธีการนี้มีประโยชน์มาก เช่น ทำให้โปรแกรมไคลเอ็นต์ของเทลเน็ตบอกโปรแกรมเซิร์ฟเวอร์ว่ากดแป้น Enter แล้ว และต้องการให้เซิร์ฟเวอร์ประมวลผลคำสั่งได้ ทีซีพียังมีวิธีการ "ควบคุมการไหลของข้อมูล" เพื่อไม่ให้คอมพิวเตอร์ที่รับข้อมูลได้รับข้อมูลจากฝ่ายส่งมากเกินไป

ทีซีพีช่วยให้โปรแกรมคอมพิวเตอร์สองโปรแกรมแลกเปลี่ยนข้อมูลกันบนเครือข่าย หรือบนกลุ่มเครือข่าย (อินเทอร์เน็ต) ได้ ทีซีพีต้องใช้บริการของไอพีในการย้ายข้อมูลระหว่างส่วนย่อยๆ ของเครือข่าย ข้อมูลที่ส่งจะถูกแบ่งเป็นกลุ่มย่อยเรียก ดาต้าแกรม ซึ่งข้อมูลหนึ่งชุดอาจแบ่งเป็นหนึ่งดาต้าแกรมหรือมากกว่านั้น ดาต้าแกรมแต่ละตัวต้องวิ่งผ่านเครือข่ายหลายเครือข่ายก่อนจะถึงจุดหมายเลขไอพี บนแต่ละเครือข่ายมีอุปกรณ์ที่เรียกว่า "เราเตอร์" (Router) เราเตอร์เป็นตัวตรวจสอบว่าหมายเลขไอพีปลายทางอยู่บนเครือข่ายอื่นหรือไม่ ถ้าอยู่เราเตอร์จะส่งดาต้าแกรมต่อไปยังเครือข่ายอื่น การแบ่งงานระหว่างทีซีพี และไอพีแสดงอยู่ในรูปที่ 2.6



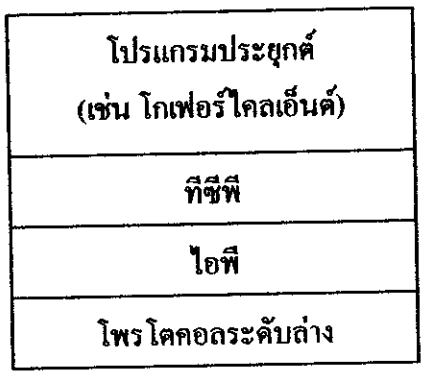
รูปที่ 2.6 ไอพีเป็นผู้ส่งข้อมูลผ่านเครือข่ายที่โยงอยู่ระหว่างทั้งสองเครื่อง
(ที่มา: The internet for everyone, Richard W.Wiggins)

ข้อมูลของคุณต้องถูกแปลงหลายครั้งหลายคราวกว่าที่จะถึงมือผู้รับ เช่นเมื่อคุณเลือกเอกสารในโกเฟอร์ โปรแกรมไคลเอ็นต์ของคุณจะเปิดการเชื่อมต่อแบบทีซีพีไปยังเซิร์ฟเวอร์พร้อมกับขอให้เซิร์ฟเวอร์ส่งเอกสารกลับมาโดยใช้การเชื่อมต่อนั้น คำขอของคุณอาจส่งไปตามเส้นทางหนึ่งบนอินเทอร์เน็ต แต่เอกสารที่ส่งกลับมาอาจใช้เส้นทางใหม่ที่ไม่เหมือนเดิมก็ได้ ความจริงไอพีอาจแบ่งดาต้าแกรมออกเป็นส่วนย่อยลงไปอีก เรียก แฟรกเม้นต์ (Fragment) เพื่อให้ส่งได้อย่างมีประสิทธิภาพ แฟรกเม้นต์อาจถูกส่งแยกไปตามเส้นทางต่างๆ กัน ไอพีอาจส่งดาต้าแกรมโดยไม่เรียงลำดับ นอกจากนี้ไอพีไม่สามารถรับรองว่าดาต้าแกรมส่งถึงจุดหมายได้เรียบร้อย เป็นหน้าที่ของทีซีพีที่ต้องตรวจสอบว่าข้อมูลถึงจุดหมายทั้งหมดหรือไม่ ถ้าไม่ถึงจะต้องขอให้ส่งข้อมูลมาใหม่

เหตุใดจึงมีการอนุญาตให้ส่งข้อมูลแยกออกจากกันไปหลายเส้นทางก่อนถึงจุดหมายเลขไอพี เหตุผลหนึ่งคือ ลักษณะของเครือข่าย เช่นนี้มีประสิทธิภาพมากกว่า และเรียกกันว่า "แพ็คเก็ตสวิชชิ่ง" (Packet-Switching) ถ้าจะเปรียบเทียบกับถนน ถ้าเส้นทางใดติดขัด และมีเส้นทางอื่นที่สามารถไปถึงจุดหมายเดียวกันได้ ก็ควรไปทางนั้น นอกจากนี้ถ้าเส้นทางไหนเกิดใช้ไม่ได้ การทำงานแบบนี้ทำให้ข้อมูลถูกส่งไปทางอื่นได้ เครือข่ายที่มีลักษณะเช่นนี้เหมาะอย่างยิ่งสำหรับใช้ในการทหาร คอมพิวเตอร์พิเศษที่มีหน้าที่ตัดสินใจว่าจะส่งดาต้าแกรมไปเส้นทางใดคือ เราเตอร์ งานของเราเตอร์คือส่งดาต้าแกรมไปยังเครือข่ายอื่นเมื่อจุดหมายปลายทางของดาต้าแกรมไม่ได้อยู่บนเครือข่ายของตน เมื่อมีเส้นทางให้เลือกหลายเส้น เราเตอร์เลือกใช้เส้นทางบางเส้นมากกว่าเส้นทางนั้นเร็วกว่าหรือเพราะผู้ดูแลระบบกำหนด

การส่งข้อมูลของคุณยังต้องใช้การทำงานในอีกระดับคือ ข้อมูลต้องส่งผ่านสื่อกลาง เช่น สายอีเธอร์เนต (Ethernet) สายเคเบิ้ลไฟเบอร์ออพติก หรือสายโทรศัพท์ ไอพีต้องใช้บริการของ

ซอฟต์แวร์สำหรับ "ควบคุมการใช้สื่อกลาง" (Media access control) ในการส่งและรับข้อมูลผ่านสื่อเหล่านี้ รายละเอียดว่าสื่อกลางเหล่านี้ใช้งานได้อย่างไรเป็นเรื่องที่ไม่สำคัญต่อผู้ใช้ พูดคร่าวๆ คือ ไอพีต้องใช้ความสามารถของ "โพรโตคอลระดับล่าง" อย่างเช่นอีเธอร์เน็ตเพื่อส่งข้อมูลไปตามสายการสื่อสาร เหมือนกับที่ทีซีพีต้องใช้ความสามารถของไอพีในการสร้างและส่งดาต้าแกรม แนวความคิดนี้อธิบายได้จากขั้นของโพรโตคอลในรูปที่ 2.7 ซึ่งใช้อธิบายการสื่อสารแบบ TCP/IP ได้อย่างคร่าวๆ โชคดีที่เราไม่จำเป็นต้องเข้าใจการทำงานนี้ละเอียดทุกขั้นตอนเพื่อใช้อินเทอร์เน็ต แต่ความเข้าใจเรื่องนี้มีประโยชน์เมื่อคุณใช้เครือข่าย เช่นเมื่อคุณส่งข้อมูลแล้วมีการหยุดเป็นบางครั้ง ก็แสดงว่าดาต้าแกรมอาจวนหาทางออกอยู่ในเส้นทางที่ติดขัด นอกจากนี้ยังเป็นประโยชน์ในการเลือกวิธีเชื่อมต่อกับอินเทอร์เน็ตแบบต่างๆ ซึ่งจะได้พูดถึงในบทต่อไป



รูปที่ 2.7 ชั้นของโพรโตคอล

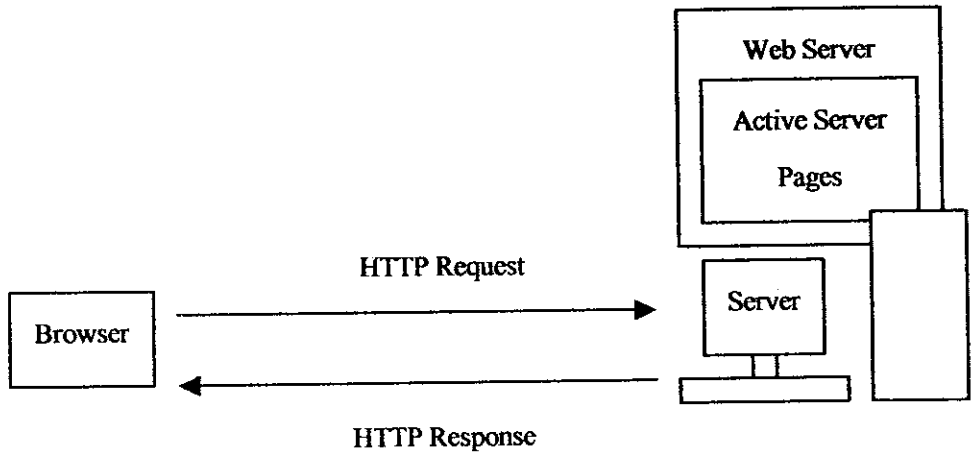
(ที่มา: The internet for everyone, Richard W.Wiggins)

2.3 Active Server Pages (ASP)

ASP หรือ Active Server Pages เป็นโปรแกรมที่พัฒนาขึ้นโดยบริษัทไมโครซอฟต์ เพื่อใช้งานทางด้านอินเทอร์เน็ต โดย ASP จะทำหน้าที่ตีความเอกสารที่เขียนด้วยภาษาสคริปต์ เช่น VBScript โดยที่มี ASP tag (คือ คำสั่งที่มีเครื่องหมาย <% %>) กำกับอยู่ ซึ่งเบราเซอร์ทั่วไป เช่น Netscape Navigator หรือ Internet Explorer ไม่สามารถนำไปแสดงผล จากนั้นจึงสร้างเอกสารผลลัพธ์เป็นเอกสาร HTML อันเป็นเอกสารที่ประกอบด้วย HTML tag ต่างๆ (คือ คำสั่งที่มีเครื่องหมาย <>) กำกับอยู่ ซึ่งเบราเซอร์ทั่วไปดังกล่าวสามารถนำไปสร้างเป็นเว็บเพจขึ้นเพื่อใช้แสดงผลได้

การทำงานของโปรแกรม ASP จะเกิดขึ้นเฉพาะทางฝั่งเซิร์ฟเวอร์เท่านั้น เราเรียกว่า เป็นการทำงานแบบ server side จากนั้นผลลัพธ์ที่ได้จะถูกส่งไปให้เว็บเซิร์ฟเวอร์ แล้วเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารดังกล่าวต่อไปยังเบราเซอร์อีกทีหนึ่ง เมื่อเบราเซอร์ได้รับเอกสารนั้นแล้ว เบราเซอร์ก็จะ

สามารถแสดงผลได้ถูกต้องครบถ้วน โดยการทำงานของเบราเซอร์ทางฝั่งของผู้ใช้นี้ เราเรียกว่าเป็นการทำงานแบบ client side



รูปที่ 2.8 กระบวนการทำงานของ ASP

(ที่มา: เพิ่มพลังอินเทอร์แอคทีฟให้เว็บเพจด้วย ASP, กิตติภูมิ วรฉัตร)

การทำงานทั้งหมดจะเริ่มจาก เบราเซอร์ร้องขอเอกสาร HTML ไปยังเว็บเซิร์ฟเวอร์ผ่านทาง HTTP (HTTP request) โดยที่เอกสารที่ขอไปจะเป็นแฟ้มข้อมูลที่มีนามสกุลเป็น .asp (เช่น search.asp ฯลฯ) เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอดังกล่าว ก็จะส่งเอกสารนั้นไปให้ ASP ตีความ จากนั้น ASP ก็จะสร้างเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปยังเบราเซอร์ และใช้แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP response) ซึ่งการทำงานของ ASP นี้แทบไม่แตกต่างไปจากหลักการทำงานของ โปรแกรม CGI (Common Gateway Interface) จนอาจกล่าวได้ว่า ASP ก็เป็นโปรแกรม CGI ประเภทหนึ่งเช่นกัน

การเขียนโปรแกรมเพื่อสร้างเอกสารที่จะทำงานกับ ASP นั้น (ในที่นี้ขอเรียกว่า เอกสาร ASP) ไม่จำเป็นต้องอาศัยโปรแกรมเฉพาะในการเขียน เราสามารถนำ โปรแกรมประเภท text editor ทั่วไปมาใช้งานได้ทันที เช่น โปรแกรม Notepad ฯลฯ หรือจะใช้โปรแกรมที่เขียนเอกสาร ASP โดยเฉพาะก็ได้ เช่น Visual Inter Dev เป็นต้น

เอกสารASP แตกต่างกับเอกสาร HTML ทั่วไปตรงที่มีส่วนของคำสั่ง ASP อยู่ในเอกสารด้วย โดยทั่วไปหากเรานำเอกสาร HTML มาเปลี่ยนเป็นเอกสาร ASP เลยก็ทำได้ นั่นคือวิธีการสร้างเอกสาร ASP แบบง่ายๆ เช่น เราสามารถเปลี่ยนเอกสาร HTML ที่ชื่อ index.html ไปเป็น index.asp ได้เลย โดยที่เมื่อโปรแกรม ASP ตีความส่วนใดของเอกสารที่มี HTML Tag กำกับอยู่ ก็จะไม่เกิดความเปลี่ยนแปลงใดๆ ในเอกสารนั้นเลย แต่หากว่าส่วนใดมี ASP tag กำกับ ASP ก็จะเปลี่ยน

เอกสารส่วนดังกล่าวไปอยู่ในรูปข้อความทั่วไปหรือเป็น HTML tag แทน เช่น หากในเอกสารมีคำสั่งดังนี้

<%response.write("Hello" & now)%>

ก็จะถูกเปลี่ยนเป็น

Hello แล้วตามด้วยวัน และเวลาปัจจุบัน

Active Server Pages ยังมีคุณลักษณะ (features) ที่สำคัญ 4 อย่างที่ทำให้โดดเด่นนั้น คือ

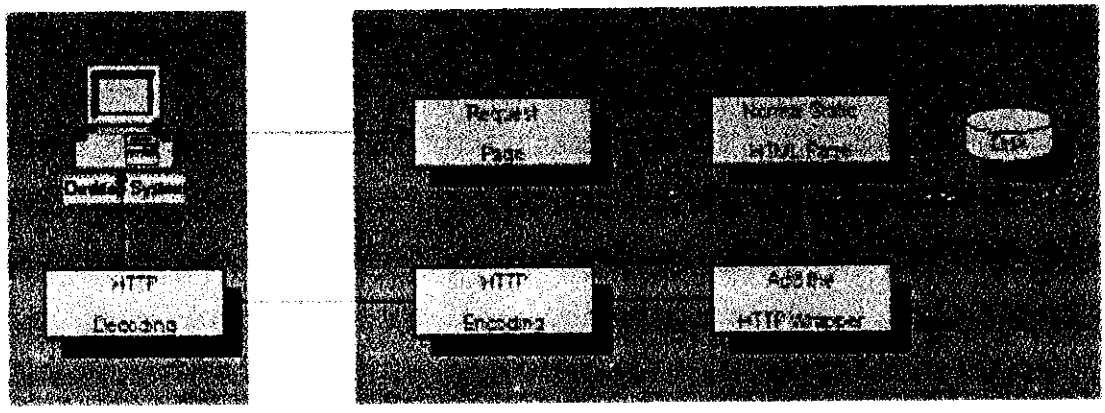
1. Active Server Pages สามารถบรรจุ Script ที่ใช้ประมวลผลทาง Server ได้ การทำเช่นนี้ได้ทำให้เกิดประโยชน์มาก เพราะจะทำให้เราสามารถสร้างหน้าเอกสารที่เป็น Dynamic ได้ ตัวอย่างง่ายๆ ที่แสดงถึงประโยชน์ของคุณลักษณะนี้คือเราสามารถที่จะสร้างเอกสาร (Page) ที่แสดงข้อความทักทายที่แตกต่างกันไป ในแต่ละเวลาของวันได้

2. Active Server Pages ได้เตรียม built-in object มากมาย การที่มี Built-in object ใน Active Server Pages ช่วยให้ Script ของเรามีประสิทธิภาพในการใช้งานมากขึ้น นั้นเพราะ Object ต่างๆ เหล่านี้จะทำให้เราสามารถรับ-ส่ง ข้อมูล (Data) ระหว่าง Server กับ Client (Browser) ได้ ตัวอย่างเช่น การใช้ Object "Request" เราสามารถรับข้อมูลจากผู้ใช้ที่ส่งมาทางฟอร์ม (Form) ของ HTML และส่งข้อมูลนั้นต่อไปให้กับส่วนของ Script ที่ต้องการได้อย่างง่ายดาย

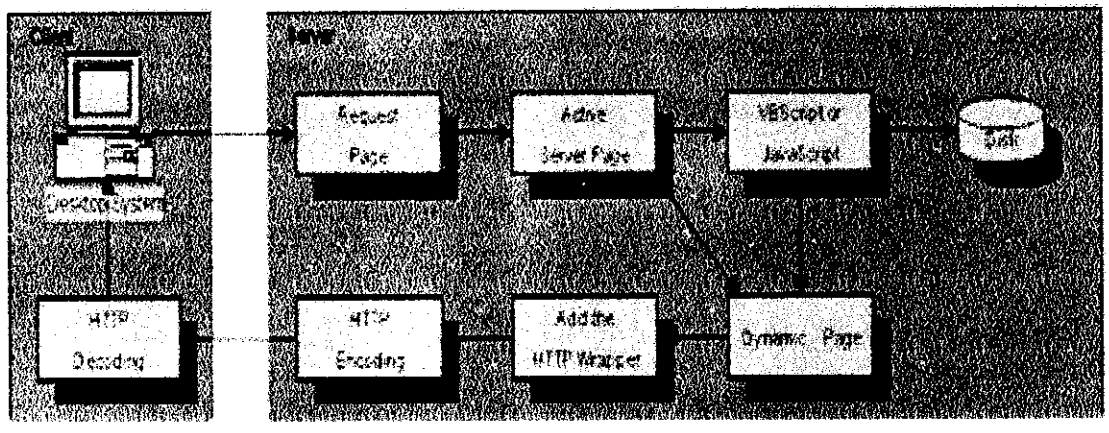
3. Active Server Pages สามารถเพิ่ม Component ที่ต้องการได้ ไม่เฉพาะ Component มาตรฐานที่ Active Server Pages ได้เตรียมไว้ตอนที่ Install เท่านั้น Active Server Pages ยังสามารถทำการเพิ่ม Component ที่ต้องการเข้าไปได้อีก

4. Active Server Pages สามารถทำการติดต่อกับฐานข้อมูล (Database) ดังเช่น Microsoft SQL Server หรือ Microsoft Access ได้เป็นอย่างดี โดยการใช้ชุดของ Object พิเศษ (Object เหล่านี้มีเป็นมาตรฐานอยู่แล้วใน Active Server Pages) ที่เรียกว่า ActiveX Data Object (ADO) คุณลักษณะในข้อนี้ทำให้ Active Server Pages มีประสิทธิภาพมากที่สุดในการที่จะนำไปใช้งาน

ดังนี้ด้วยคุณลักษณะทั้ง 4 ข้อที่กล่าวมา ทำให้กล่าวได้ว่า Active Server Pages นั้นคือหน้าเอกสาร HTML (pages) มาตรฐานที่ได้เพิ่มการทำงานของ Script ที่ประมวลผลทาง Server โดยมี Object และ Component เพิ่มเข้ามาช่วยในการทำงาน ทำให้สามารถสร้าง Web Site ที่มีหน้าเอกสารแบบ Dynamic ได้



รูปที่ 2.9 แสดงการทำงานระหว่าง Server และ Client ของ Static HTML Page



รูปที่ 2.10 แสดงการทำงานระหว่าง Server และ Client ของ Active Server Pages

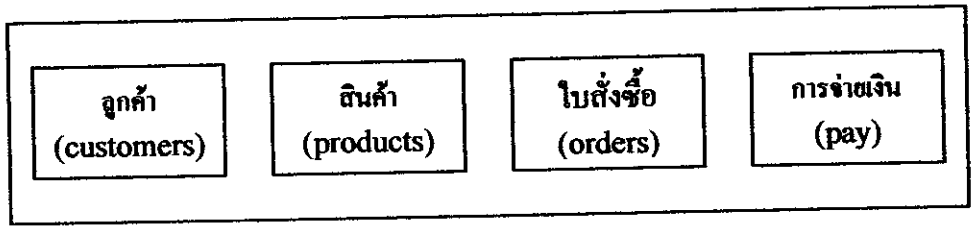
2.4 <u>ทฤษฎีการออกแบบฐานข้อมูลและตาราง</u>

2.4.1 ERDs (Entity-relationship diagrams)

ERDs ถูกใช้เป็นตัวแบบข้อมูล โดยเฉพาะสำหรับเป็นตัวแบบของโครงสร้างข้อมูลซึ่งในที่สุดก็จะถูกดำเนินการในฐานข้อมูล ส่วนประกอบของ ERDs จำเป็นได้แก่

1. เอนทิตี้ (Entity)

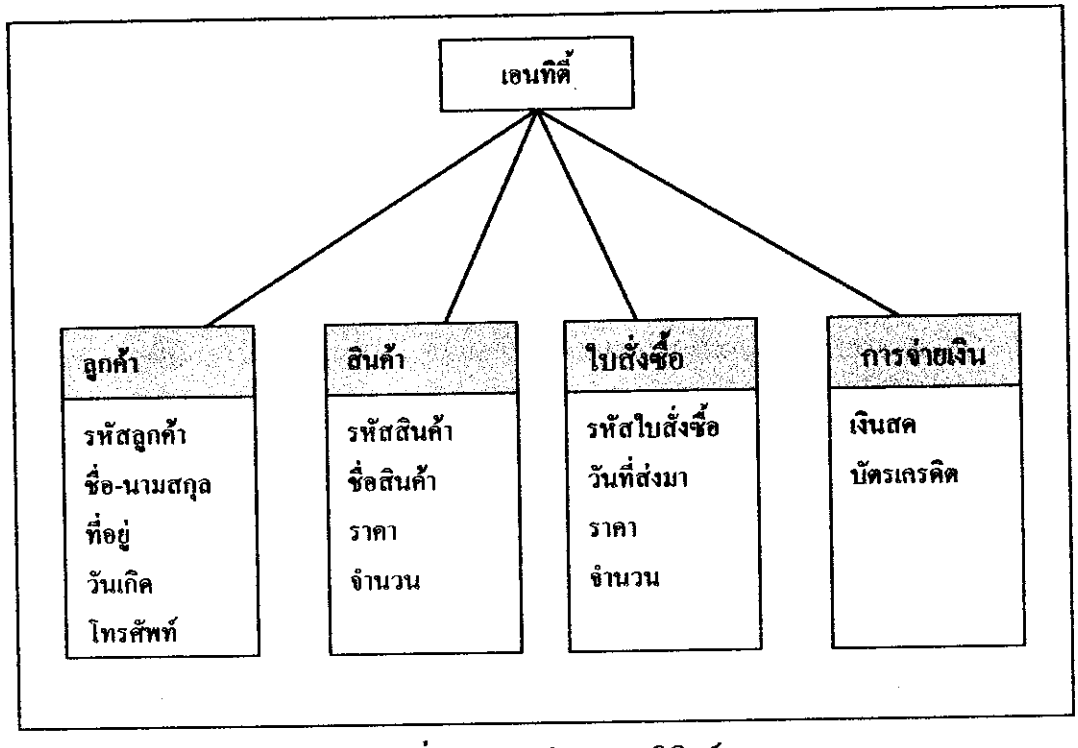
จะหมายถึง คน สถานที่ วัตถุ หรือสิ่งต่างๆ เช่น พนักงาน ลูกค้า สินค้า ราคา ใบสั่งของ ฯลฯ ตัวอย่างเช่นการออกแบบสร้างฐานข้อมูลเกี่ยวกับประวัติการซื้อสินค้าของลูกค้าเอนทิตี้ในส่วนนี้จะมี ลูกค้า ใบสั่งซื้อ การจ่ายเงิน เป็นต้น



รูปที่ 2.11 ตัวอย่าง เอนทิตี้

2. แอททริบิวท์ (Attribute)

จะหมายถึง ข้อมูลแต่ละเอนทิตี้ที่ใช้แสดงรายละเอียด เกี่ยวกับเอนทิตี้นั้นๆ เช่น แอททริบิวท์เอนทิตี้ของสินค้าจะประกอบไปด้วย รหัส ชื่อสินค้า ราคา แอททริบิวท์เอนทิตี้ของใบสั่งซื้อจะประกอบไปด้วย รหัสใบสั่งซื้อ วันที่ ราคา จำนวนสินค้า และแอททริบิวท์เอนทิตี้ของการจ่ายเงินจะประกอบไปด้วย เงินสด บัตรเครดิต



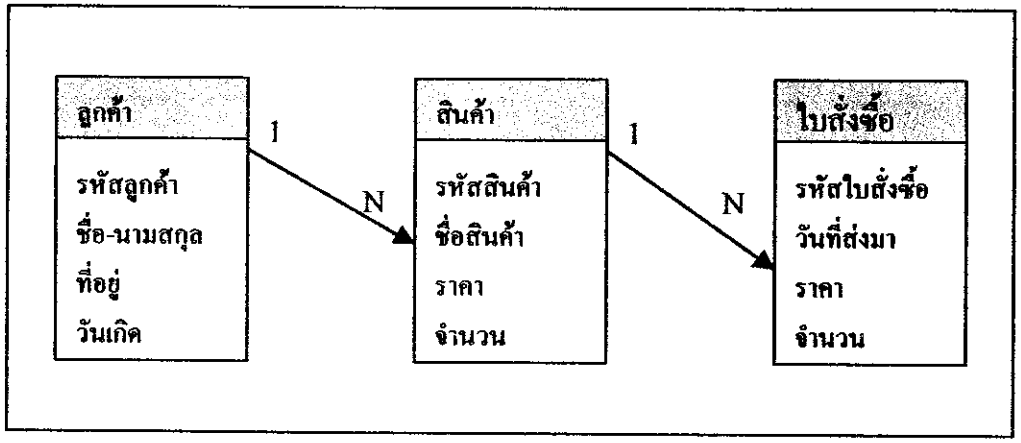
รูปที่ 2.12 ตัวอย่างแอททริบิวท์

3.ความสัมพันธ์ (Relationships)

จะหมายถึง ความสัมพันธ์ของเอนทิตี้ที่มีต่อกันบนระบบ ตัวอย่างเช่นประวัติการซื้อสินค้าของลูกค้าก็จะประกอบไปด้วยเอนทิตี้ลูกค้า เอนทิตี้สินค้า และเอนทิตี้ใบสั่งซื้อ จะเห็นว่ามีความสัมพันธ์กันระหว่างทั้ง 3 เอนทิตี้ โดยเอนทิตี้ลูกค้าเชื่อมความสัมพันธ์กับสินค้า และสินค้าก็มีความ

สัมพันธ์กับใบสั่งซื้อ ฐานข้อมูลเชิงสัมพันธ์จึงเป็นการรวมเอนทิตี้ที่มีความสัมพันธ์ระหว่างกันบนระบบเอาไว้ในฐานข้อมูล ปกติแล้วฐานข้อมูลเชิงสัมพันธ์จะทำงานร่วมกับข้อมูล 2 ตาราง หรือมากกว่า เพื่อเชื่อมความสัมพันธ์ระหว่างตาราง รูปแบบของความสัมพันธ์ มีอยู่ 3 แบบด้วยกันคือ

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One Relationships)
- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many Relationships)
- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-Many Relationships)



รูปที่ 2.13 ตัวอย่างความสัมพันธ์

2.4.2 ส่วนประกอบของภาษาฐานข้อมูล

ภาษาฐานข้อมูล SQL จะทำการกำหนดโครงสร้างของข้อมูล จัดการข้อมูล ค้นหาข้อมูลโดยการติดต่อและควบคุมระบบจัดการฐานข้อมูล (DBMS) อีกทีหนึ่ง ทางทฤษฎีแล้วภาษาฐานข้อมูลจะมีส่วนประกอบสำคัญอยู่ 3 ส่วนคือ

1.ส่วนภาษานิยามข้อมูล (Data Definition Language : DDL)

เป็นส่วนของคำสั่งที่ใช้กำหนดโครงสร้างข้อมูล สร้างฐานข้อมูล การสร้างตาราง (Base Tables) การกำหนดดัชนี (Indexes) การสร้างวิว (View) เพื่อดูผลลัพธ์ นอกจากนั้นยังใช้ในการกำหนดกฎเกณฑ์ต่างๆ ให้ DBMS ใช้ในการตรวจสอบความถูกต้องของข้อมูลด้วย

2.ส่วนภาษาจัดการข้อมูล (Data Manipulation Language: DML)

เป็นส่วนของคำสั่งในการค้นหาข้อมูล (Retrive) เพิ่มเติมข้อมูล (Insert) เปลี่ยนแปลงแก้ไขข้อมูล (Update) การลบข้อมูล (Delete)

3.ส่วนภาษาในการควบคุมข้อมูล (Data Control Language: DCL)

เป็นส่วนของคำสั่งที่ใช้ควบคุมเข้าถึงข้อมูล (Access control) ของผู้ใช้ โดยจะกำหนดว่าผู้ใช้คนใดสามารถเพิ่ม ลบ แก้ไข-เปลี่ยนแปลงข้อมูล ผู้ใช้คนใดทำได้เพียงเรียกดู-ค้นหาข้อมูล เป็นการกำหนดสิทธิ์ (Privileges) ให้แก่ผู้ใช้งาน เพื่อความปลอดภัยให้กับข้อมูล

2.4.3 ส่วนประกอบพื้นฐานของฐานข้อมูลเชิงสัมพันธ์

ระบบฐานข้อมูลเชิงสัมพันธ์จะมีส่วนประกอบที่ใช้ในการทำงาน คือ ตาราง ซึ่งภายในจะมีทั้งแถว และ คอลัมน์ รวมเป็น เรคคอร์ด (Record) แล้วยังมีคีย์หลัก-คีย์นอก ซึ่งสร้างจากคอลัมน์ตลอดจนโดเมนที่ใช้กำหนดขอบเขต-ประเภทข้อมูล

1. ตาราง (Table)

ฐานข้อมูลเชิงสัมพันธ์จะบรรจุตารางไว้จำนวนมาก และในฐานข้อมูลจะเก็บข้อมูลที่สัมพันธ์กันทุกตาราง ในการเก็บข้อมูลทั้งในคอลัมน์และแถวจะมีหลักการดังนี้

- ตารางจะต้องมีชื่อไม่ซ้ำ (Unique name) และประกอบไปด้วยแถวและคอลัมน์
- ทุกคอลัมน์จะบรรจุข้อมูลได้เพียงชนิดเดียว
- ข้อมูลในแต่ละแถวจะต้องไม่ซ้ำกัน

2. คอลัมน์ (Column)

เป็นการเก็บรายละเอียดต่างๆ ของข้อมูลและกำหนดประเภทข้อมูลที่จะใช้งานลงไปเช่น คอลัมน์ Name ให้ประเภทของข้อมูลเป็น Character (ตัวอักษร) และคอลัมน์ Salary ให้ประเภทข้อมูลเป็น Integer/Numeric ในตารางจะประกอบไปด้วยหลายคอลัมน์ เพื่อบรรจุรายละเอียดของชื่อ ที่อยู่ ตำแหน่ง เงินเดือน ประวัติการทำงาน ฯลฯ

3. แถว (Row)

แถวจะประกอบไปด้วยหลายคอลัมน์ คือใน 1 แถวจะมีข้อมูลต่างๆที่อยู่ในแต่ละคอลัมน์ เช่น รหัสพนักงาน ชื่อ-นามสกุล ตำแหน่ง เงินเดือน วันลาป่วย ประวัติพนักงาน ฯลฯ และในตารางข้อมูลจะประกอบไปด้วยหลายแถวเช่นกัน

4. คีย์หลัก (Primary Key)

คีย์หลัก หมายถึง คอลัมน์ที่มีข้อมูลไม่ซ้ำกัน (Unique) เป็นเรื่องสำคัญในระบบฐานข้อมูล คอลัมน์ที่มีคุณสมบัติเป็นคีย์หลักได้จะต้องมีค่าของข้อมูลไม่ซ้ำกันเลยเช่น คอลัมน์รหัสพนักงาน ซึ่งตามปรกติแล้วพนักงานแต่ละคนจะมีรหัสประจำตัวต่างกันแน่นอนข้อกำหนดของคีย์หลักมีดังนี้

- ในทุกตารางจะต้องมีคีย์หลัก (Logical)
- ใน 1 ตารางจะมีคีย์หลักได้เพียง 1 คีย์เท่านั้น

- ในคีย์หลักค่าของข้อมูลในคอลัมน์จะต้องไม่เป็นค่าว่าง หรือ Null โดยจะต้องกำหนดเป็น Not Null เสมอ

5. คีย์นอก (Foreign Key)

คีย์นอกจะเป็นคอลัมน์ หรือคอมบิเนชั่น (Combination) ของตารางหนึ่ง ซึ่งมีความสัมพันธ์กับคอลัมน์ในตารางอื่น และสามารถจะเชื่อมโยงข้อมูลระหว่างกันได้ คีย์นอกสามารถจะมีค่าว่างได้

6. โดเมน (Domain)

โดเมน คือ การกำหนดขอบเขต-ประเภทของข้อมูลหรือค่าของข้อมูล ที่ให้ใช้ในคอลัมน์ หรือ ในฟิลด์ เช่น คอลัมน์ Salary มีการกำหนดขอบเขต-ประเภทของข้อมูลให้เป็น Integer และคอลัมน์อื่นๆ เช่น

Name : กำหนดให้เฉพาะตัวอักษรใหญ่-เล็ก (A....Z, a....z) เท่านั้น

Salary : กำหนดให้เฉพาะเลขจำนวนเต็มบวก (Integer) มีค่าอยู่ระหว่าง 0-999,999.00

Address: กำหนดให้มีทั้งตัวอักษร (Character) และตัวเลข (Numeric)

2.5 การเข้ารหัสข้อมูลแบบ RSA (Rivest-Shamir-Adelman Encryption)

การเข้ารหัสข้อมูลแบบนี้ถูกประดิษฐ์ขึ้นในปี ค.ศ. 1978 และจนถึงทุกวันนี้ยังสามารถใช้รักษาความปลอดภัยของข้อมูลได้เป็นอย่างดี หลักการทำงานของการเข้ารหัสข้อมูลแบบนี้ก็คือความยากในการหาส่วนประกอบที่เป็นตัวเลขไพรม์ (Prime Number) ของตัวเลขไพรม์ขนาดใหญ่

การประดิษฐ์วิธีการเข้ารหัสแบบ RSA นี้ทำให้การเข้ารหัสแบบ Public-Key Encryption สามารถนำไปใช้ได้จริงในทางปฏิบัติ การเข้ารหัสแบบ RSA นี้ให้ความปลอดภัยสูงมากและมีการนำไปใช้อย่างแพร่หลายในปัจจุบัน

2.5.1 ขั้นตอนในการเข้ารหัสแบบ RSA

1. หาค่าตัวเลขไพรม์ขนาดใหญ่ p และ q (ตัวอย่างเช่น ตัวเลขไพรม์ ที่มีขนาด 400 บิต) แล้วคำนวณหา n โดยที่

   $$n = pq$$

2. เลือกตัวหารร่วมมาก (ห.ร.ม.) หรือที่เรียกว่า gcd — Greatest Common Divisor และกำหนดให้ค่านี้เป็นกุญแจถอดรหัสที่เรียกว่า Decryption Exponent หรือ d ซึ่งตัวเลข d นั้นต้องไม่มีตัวหารร่วมใดๆ กับค่า (p-1) และ (q-1) และตัวเลขนี้จะทำหน้าที่เป็นส่วนประกอบของ Private Key โดยอาศัยสูตรดังต่อไปนี้เพื่อกำหนดความสัมพันธ์ระหว่างค่าต่างๆ ดังนี้คือ

$$gcd(d,(p-1)(q-1)) = 1$$

นั้นคือตัวหารร่วมมาก (ห.ร.ม.) ระหว่างค่า d,(p-1),(q-1) มีค่าที่มากที่สุดคือ 1

3. คำนวณหาค่า Encryption Exponent หรือ ค่า e ซึ่งจะทำหน้าที่เป็นส่วนประกอบของ Public Key โดยกำหนดความสัมพันธ์ระหว่างค่าต่างๆ ดังนี้คือ

$$ed = 1 \bmod (p-1)(q-1)$$

4. กำหนดค่า Public Key = (e , n) โดยค่า n = pq และค่า Public Key นี้สามารถที่จะแจกจ่ายออกไปได้
5. ในการเข้ารหัสข้อมูลนั้นจะใช้ Public Key = (e , n) มาทำการเข้ารหัสข้อมูล m ได้ดังนี้คือ

$$c = m^e \bmod n$$

โดยข้อมูลที่ได้หลังจากการเข้ารหัสคือ c = Cipher Text ซึ่งสามารถที่จะส่งออกไปสู่ระบบเครือข่ายภายนอกได้อย่างปลอดภัย เนื่องจากในการถอดรหัสนั้นจะต้องใช้กุญแจ d ที่ถูกเก็บไว้เป็น Private Key

6. ในการถอดรหัสนั้นจะใช้ Private Key = d มาใช้ในการถอดรหัสเพื่อให้ได้ข้อความเดิม m กลับมา

$$m = cd \bmod n$$

นั่นคือการนำเอาข้อความที่เป็น Cipher Text, c มาทำการถอดรหัสเพื่อให้ได้ข้อความ m กลับคืนมา โดยใช้กุญแจรหัส d ที่ถูกเก็บไว้เป็นความลับและไม่แจกจ่ายให้ใคร

7. สาเหตุที่การเข้ารหัสและถอดรหัสโดยวิธีการ RSA สามารถทำการเข้ารหัสและถอดรหัสข้อมูลได้นั้น สามารถดูได้จากขั้นตอนทางคณิตศาสตร์ ดังต่อไปนี้คือ

$n = pq$; โดยที่ q คือตัวเลข Prime

ดังนั้น

$x(p-1)(q-1) \equiv 1 \pmod n$

ซึ่งหมายความว่า ถ้า s เป็นตัวเลข Integer ตัวหนึ่งแล้ว

$n = s*[(p-1)(q-1)] + 1$; โดยที่ $n = pq$

$cd = (me)d = ms(p-1)(q-1) = med \pmod n$

8. ตัวอย่างของการคำนวณ RSA

$p = 7 \quad q = 11$

$pq = 77$

$(p-1)(q-1) = (7-1)(11-1) = (6)(10) = 60$

เลือก d = 13 ดังนั้น ตัว ห.ร.ม. ของ 13 กับ 60 คือ 1
ดังนั้น e = 37 เนื่องจากว่า

$$37*13 = 481 \ (1 \pmod{60})$$

สมมติว่า ข้อความ m = 2 (โดยที่ $2 < n = 77$) ดังนั้น
$c = m^e \bmod n = 237 \bmod 77 = 51$ --→ Cipher Text
$m = c^d \bmod n = 5113 \bmod 77 = 2$ --→ Plain Text
ดังนั้นเราได้ m = 2 กลับมาเหมือนเดิม

2.5.2 ความปลอดภัยของการเข้ารหัสแบบ RSA

- ความปลอดภัยในการเข้ารหัสแบบ RSA ขึ้นกับความยากในการคำนวณหาค่า d จากค่า n และ e ที่มีอยู่
- เพราะหากเราสามารถทำการถอดรูดหาตัวประกอบ (factorize) ของค่า n แล้วเราก็สามารถที่จะคำนวณหาค่า p และ q ได้ และจากนั้นเราก็สามารถหาค่า d ซึ่งเป็น Private Key ได้ด้วย
- หากทำการถอดรูดหาตัวประกอบของค่าตัวเลขนั้นๆ ไม่มีความยากแล้วจะทำให้การเข้ารหัสแบบ RSA นั้นไม่มีความปลอดภัย คือผู้ไม่ประสงค์ดีสามารถทำการคำนวณหาค่า d ได้ และนำค่า d ไปใช้ในการคำนวณถอดรหัสหาข้อความเดิม m ซึ่งเป็น Plain Text จากข้อความ c ที่ขโมยมาที่เป็น Cipher Text โดยไม่ได้รับอนุญาต
- ด้วยเทคโนโลยีปัจจุบัน เราสามารถทำการถอดรูดหาตัวประกอบตัวเลขที่มีขนาดใหญ่มากที่สุดได้ถึง 400 Bits แต่ก็ได้มีการวิจัยอย่างกว้างขวางที่จะพยายามเพิ่มขนาดให้ถึง 512 Bits ซึ่งเป็นขนาดที่ใช้ในการเข้ารหัสแบบ RSA
- คำถามที่สำคัญคือ ความปลอดภัยของการเข้ารหัสแบบ RSA ขึ้นอยู่กับความยากในการถอดหาตัวประกอบของตัวเลขไพรม์อย่างเดียวหรือไม่ เพราะอาจมีวิธีอื่นๆ อีกที่สามารถนำมาใช้ในการคำนวณหา Plain Text จาก Cipher Text ได้

2.5.3 คุณสมบัติของการเข้ารหัสแบบ RSA

- การเข้ารหัสแบบ RSA นั้นเป็นการเข้ารหัสแบบ Block Cipher
- ปกติแล้วการเข้ารหัสแบบ RSA จะช้ากว่าการเข้ารหัสแบบอื่นๆ มาก เนื่องจากว่าต้องใช้การคำนวณที่สลับซับซ้อนและขนาดกุญแจที่ใช้มีขนาดใหญ่มากเมื่อเปรียบเทียบกับการเข้ารหัสแบบ DES แล้วนั้น การเข้ารหัสแบบ RSA จะช้ากว่าประมาณ 1,000 เท่า

- เนื่องจากความช้าในการเข้ารหัสข้อมูล จึงไม่นิยมเอา RSA ไปใช้ในการเข้ารหัสข้อความที่มีขนาดใหญ่ แต่จะนำเอาไปใช้ในการเข้ารหัสข้อมูลขนาดเล็กที่ต้องการความปลอดภัยสูงมากๆ เช่น ใช้ในการเข้ารหัสและแจกจ่าย Secret Key ที่ใช้เป็น Session Key ในการติดต่อสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ในแต่ละครั้ง