บทที่ 5
สถาปัตยกรรมคอมพิวเตอร์

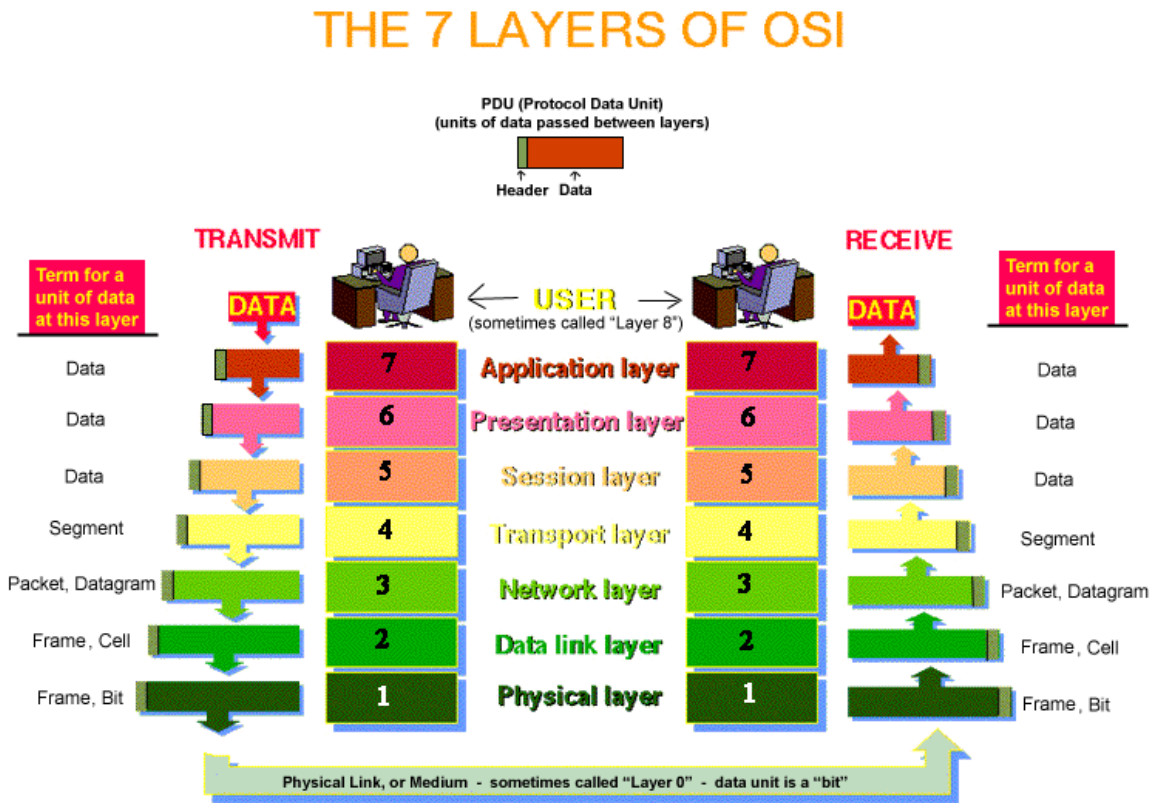
โปรโตคอลคืออะไร ?

การเชื่อมต่อคอมพิวเตอร์ให้เป็นเครือข่ายด้วยสายสัญญาณนั้น เป็นขั้นตอนที่ง่ายซึ่งในการสร้างเครือข่ายในแต่ส่วนที่ท้าทายก็คือ การพัฒนามาตรฐานเพื่อให้คอมพิวเตอร์และอุปกรณ์เครือข่ายที่ผลิตโดยบริษัทต่างๆ ที่จะสามารถติดต่อสื่อสารกันได้ ซึ่งเป็นมาตรฐานนี้คือ โปรโตคอล (Protocol) หรือสรุปสั้นๆ โปรโตคอลคือ กฎ ขั้นตอน และรูปแบบของข้อมูลที่ใช้ในการสื่อสารระหว่างคอมพิวเตอร์สองเครื่องใดๆ ที่เชื่อมต่อกันเป็นเครือข่ายโปรโตคอลของเครือข่าย บางทีอาจเรียกว่า “สถาปัตยกรรมเครือข่าย (Network Architecture)” เนื่องจากระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันเป็นระบบที่มีความซับซ้อนมากทำให้ยากต่อการออกแบบโดยคนๆ เดียวหรือคนกลุ่มเดียว เพื่อให้การพัฒนาระบบเป็นไปอย่างมีประสิทธิภาพและง่ายขึ้น จึงมีการแบ่งโปรโตคอลออกเป็นชั้นๆ หรือเลเยอร์ (Layer) การทำงานในแต่ละเลเยอร์จะไม่ซ้ำซ้อนกัน ซึ่งเลเยอร์ที่อยู่ต่ำกว่าจะทำหน้าที่ให้บริการ (Service) กับชั้นที่อยู่สูงกว่า โดยเลเยอร์ที่อยู่สูงกว่าไม่จำเป็นต้องทราบรายละเอียดว่าเลเยอร์ที่อยู่ต่ำกว่ามีวิธีให้บริการอย่างไร เพียงแค่รู้ว่ามีบริการอะไรบ้างและแต่ละบริการคืออะไรก็เพียงพอ ซึ่งแนวความคิดนี้จะเรียกว่า “เทคโนโลยีเลเยอร์ (Layer Technology)”

แบบอ้างอิง OSI

องค์การมาตรฐานนานาชาติ (The International Organization for Standardization) และใช้อักษรย่อว่า “ISO” ซึ่งคนส่วนใหญ่เข้าใจว่าย่อมาจาก “International Standard Organization” แต่จริงๆ แล้วไม่ใช่ อย่างไรก็ตาม ISO เป็นองค์กรที่ออกแบบโปรโตคอล ISO (Open System Interconnect) หรือโปรโตคอลการเชื่อมต่อเครือข่ายแบบเปิด จุดมุ่งหมายของการพัฒนามาตรฐานนี้ เพื่อให้คอมพิวเตอร์และอุปกรณ์เครือข่ายที่ผลิตโดยบริษัทต่างๆ สามารถทำงานร่วมกันได้ ชุดโปรโตคอลนี้ส่วนใหญ่จะเรียกว่า “แบบอ้างอิง OSI (OSI Reference Model)” เนื่องจากโปรโตคอลนี้ไม่ได้ถูกใช้งานอย่างแพร่หลายเหมือนโปรโตคอลอื่นๆ เช่น โปรโตคอล TCP/IP ที่ใช้อย่างแพร่หลายในระบบเครือข่ายอินเตอร์เน็ต

แบบอ้างอิงนี้จะแบ่งขั้นตอนการสื่อสารระหว่างคอมพิวเตอร์ออกเป็น 7 ขั้นตอน หรือเลเยอร์ (Layer) ดังแสดงในรูป



รูปที่ 5.1.1 เลเยอร์ของการสื่อสาร 7 ขั้นตอน

ที่มา : http://bfindarto.files.wordpress.com/2008/03/osi3.gif

ชั้นที่ 7 แอพพลิเคชันเลเยอร์ (Application Layer)

โปรโตคอลชั้นที่อยู่บนสุดของแบบอ้างอิง OSI คือชั้นที่ 7 แอพพลิเคชั่นเลเยอร์ (Application Layer) ถึงแม้ชื่อจะเป็นแอพพลิเคชันเลเยอร์ แต่ก็ไม่ได้รวมถึงแอพพลิเคชั่นของผู้ใช้ด้วย (User Application) แต่โปรโตคอลในชั้นนี้จะเป็นจุดเชื่อมต่อระหว่างแอพพลิเคชันของผู้ใช้กับกระบวนการการสื่อสารผ่านเครือข่าย ชั้นนี้อาจถือได้ว่าเป็นชั้นที่เริ่มกระบวนการติดต่อสื่อสาร เช่น เมื่อผู้ใช้ต้องการส่งอีเมลล์ โปรแกรมที่ผู้ใช้ใช้ส่งอีเมลล์จะติดต่อกับโปรโตคอลในชั้นประยุกต์เพื่อเริ่มกระบวนการทั้งหมด ตัวอย่างของโปรโตคอลที่ทำงานในเลเยอร์นี้ เช่น

- File Transfer, Access and Management (FTAM) : ให้บริการเกี่ยวกับการถ่ายโอนไฟล์ระหว่างคอมพิวเตอร์ และการอ่าน การเขียน หรือแม้กระทั่งการลบไฟล์ที่อยู่ในอีกเครื่องหนึ่งได้
- Virtual Terminal Protocol (VTP) : ให้บริการเกี่ยวกับการเข้าใช้แอพพลิเคชั่นที่อยู่อีกเครื่องหนึ่งโดยการจำลองเทอร์มินอลของเครื่องที่อยู่ห่างไกลให้กับผู้ใช้

- Message Handling Service (MHS) : ให้บริการเกี่ยวกับการรับส่งอีเมลล์
- Directory Service (DS) : ให้บริการเกี่ยวกับการจับคู่ระหว่างชื่อและที่อยู่ของคอมพิวเตอร์
- Common Management Information Protocol (CMIP) : ให้บริการข้อมูลเกี่ยวกับการจัดการเครือข่าย

ชั้นที่ 6 พรีเซนเตชันเลเยอร์ (Presentation Layer)

โปรโตคอลในชั้นนี้จะรับผิดชอบเรื่องเกี่ยวกับรูปแบบของข้อมูลที่รับส่งผ่านเครือข่ายเนื่องจากคอมพิวเตอร์ที่ต้องการแลกเปลี่ยนข้อมูลกันนั้นอาจมีวิธีการเข้ารหัส (Encoding) ที่ต่างกัน เช่น คอมพิวเตอร์บางเครื่องอาจใช้การเข้ารหัสแบบ ASCII (American Code for Information Interchange) หรือบางเครื่องอาจใช้การเข้ารหัสแบบ EBCDIC (Extended Binary Coded Decimal Interchange Code) ดังนั้นก่อนการส่งข้อมูลโปรโตคอลในเลเยอร์นี้ก็จะแปลงข้อมูลให้อยู่ในรูปแบบที่เป็นมาตรฐาน ส่วนทางด้านฝ่ายรับก็จะทำการแปลงกลับไปเป็นรูปแบบที่คอมพิวเตอร์เครื่องนั้นเข้าใจ

ชั้นที่ 5 เซสชันเลเยอร์ (Session Layer)

ชั้นเซสชัน (Session Layer) ทำหน้าที่ควบคุมการสื่อสารผ่านเครือข่ายที่กำลังเกิดขึ้นระหว่างสองฝั่ง การสื่อสารที่กำลังเป็นไปในช่วงขณะใดขณะหนึ่งจะเรียกว่า “เซสชัน (Session)” แอพพลิเคชั่นทั้งสองฝั่งสามารถแลกเปลี่ยนข้อมูลหรือรับส่งแพ็กเก็ตถึงกันและกันได้ในช่วงเวลาที่เซสชันยังอยู่ โดยเซสชันเลเยอร์จะรับผิดชอบเกี่ยวกับการสร้างเซสชัน ควบคุมการแลกเปลี่ยนข้อมูล และยกเลิกเซสชันเมื่อการสื่อสารสิ้นสุด

อีกฟังก์ชั่นหนึ่งของเซสชั่นเลเยอร์คือ การควบคุมจังหวะการรับส่งข้อมูล (Synchronization) ฟังก์ชั่นนี้มีประโยชน์ในกรณีอย่างเช่น สมมุติว่ากำลังมีการถ่ายโอนไฟล์ระหว่างสองเครื่อง แล้วเครื่องใดเครื่องหนึ่งเกิดล้มเหลวกระทันหัน การถ่ายโอนไฟล์นั้นอาจต้องเริ่มใหม่ แต่เซสชั่นเลเยอร์มีฟังก์ชั่นที่กำหนดจุดเริ่มต้นของกระบวนการถ่ายโอนไฟล์ได้ โดยเมื่อเปิดเครื่องใหม่ก็เริ่มกระบวนการถ่ายโอนไฟล์ต่อจากเมื่อคราวก่อนหน้านี้ได้

ชั้นที่ 4 ทรานสปอร์ตเลเยอร์ (Transport Layer)

ชั้นเคลื่อนย้ายข้อมูล หรือทรานสปอร์ตเลเยอร์ (Transport Layer) รับผิดชอบในการเคลื่อนย้ายข้อมูลระหว่างโพรเซสของผู้รับและโพรเซสของผู้ส่ง โพรเซสส์ในที่นี้จะหมายถึงโปรแกรมที่กำลังรันบนเครื่องคอมพิวเตอร์ ซึ่งในขณะใดขณะหนึ่งอาจจะมีหลายโพรเซสที่กำลังรันอยู่ ดังนั้นชั้นนี้จะรับผิดชอบในการรับส่งข้อมูลให้ถึงโพรเซสส์ที่ต้องการ หน้าที่อีกอย่างของโปรโตคอลในชั้นนี้คือ การตรวจเช็คแพ็กเก็ตที่ละทิ้งโดยเราท์เตอร์ และทำการส่งข้อมูลใหม่อีกครั้ง

หน้าที่ที่สำคัญอีกอย่างหนึ่งของชั้นนี้คือ การจัดเรียงแพ็กเก็ตข้อมูลที่อาจเดินทางถึงฝ่ายรับโดยไม่เป็นลำดับ เนื่องจากแพ็กเก็ตแต่ละแพ็กเก็ตเดินทางคนละเส้นทาง หรือบางแพ็กเก็ตอาจสูญหายระหว่างทางแล้วมีการส่งใหม่อีกครั้ง

โปรโตคอลในชั้นนี้แบ่งออกเป็น 2 ประเภทคือ

- คอนเน็กชั่นโอเรียนเต็ด (Connection-Oriented)
- คอนเน็กชั่นเลส (Connectionless)

โปรโตคอลในเลเยอร์นี้สามารถให้บริการได้หลายๆ แอพพลิเคชันในเวลาเดียวกัน เพื่อการกำหนดที่อยู่เพื่อใช้ติดต่อกับแอพพลิเคชันที่อยู่อีกฝั่งหนึ่ง โดยที่อยู่ในที่นี้ส่วนใหญ่จะเรียกว่า “พอร์ต (Port)” แต่การเชื่อมต่อเข้ากับพอร์ตเรียกว่า “ซ็อกเก็ต (Socket)”

ชั้นที่ 3 เน็ตเวิร์คเลเยอร์ (Network Layer)

ชั้นเครือข่าย (Network Layer) จะรับผิดชอบในการจัดเส้นทางให้กับข้อมูลระหว่างสถานีส่งและสถานีรับ ถ้ามีเส้นทางเดียว เช่น ถ้ามีคอมพิวเตอร์แค่สองเครื่องเชื่อมต่อกันโดยตรง การจัดเส้นทางคงไม่ยาก เพราะมีแค่เส้นทางเดียว แต่ถ้าเป็นเครือข่ายที่ซับซ้อนการจัดเส้นทางก็ไม่ใช่เรื่องง่ายนัก ในชั้นนี้จะไม่มีกลไกใดๆ ในการตรวจสอบข้อผิดพลาดของข้อมูล ดังนั้นฟังก์ชันนี้จึงเป็นหน้าที่ของชั้นเชื่อมโยงข้อมูล

การให้บริการในเลเยอร์นี้จะแบ่งออกเป็น 2 ประเภทคือ

- Connectionless Network Service : การส่งข้อมูลแบบไม่มีการสร้างการเชื่อมต่อก่อนโปรโตคอลที่ให้บริการแบบนี้เช่น CLNP (Connectionless Network Protocol) และ CLNS (Connectionless Network Service)
- Connection-Oriented Network Service : ก่อนที่จะมีการส่งข้อมูลทุกครั้ง จะมีการสร้างเส้นทางการเชื่อมต่อระหว่างสองสถานีก่อน ละเมื่อรับส่งข้อมูลเสร็จก็จะมีการยกเลิกเส้นทางการเชื่อมต่อดังกล่าว โปรโตคอลที่ให้บริการแบบนี้ เช่น CONP (Connection-Oriented Network Protocol) และ CMNS (Connection-Mode Network Service)

ชั้นที่ 2 ดาต้าลิงค์เลเยอร์ (Data Link Layer)

ชั้นนี้ก็มีหน้าที่เหมือนกันชั้นอื่นๆ คือรับและส่งข้อมูล ซึ่งชั้นนี้จะรับผิดชอบในการรับส่งข้อมูลและมีการตรวจสอบความถูกต้องข้อมูลด้วย ทางด้านสถานีที่ส่งข้อมูลจะจัดข้อมูลให้เป็นเฟรม (Frame) ซึ่งในเฟรมจะมีข้อมูลที่ทำให้เฟรมสามารถส่งไปยังสถานีรับผ่านเครือข่ายท้องถิ่น (LAN) อย่างถูกต้องและสำเร็จ การส่งข้อมูลสำเร็จในที่นี้หมายถึงการที่เฟรมข้อมูลส่งถึงปลายทางที่สถานีส่งต้องการโดยที่เฟรมข้อมูลไม่มีข้อผิดพลาด ดังนั้นในเฟรมต้องมีข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของเฟรมข้อมูลนั้นๆ ด้วย การส่งข้อมูลสำเร็จนั้นเหตุการณ์ต่อไปนี้ต้องเกิดขึ้น

- สถานีรับ เมื่อได้รับเฟรมแล้วต้องตรวจสอบข้อผิดพลาดของข้อมูลแล้วแจ้งให้สถานีส่งทราบ
- สถานีส่ง ต้องได้รับการตอบรับจากสถานีรับว่าได้รับเฟรมข้อมูลถูกต้องแล้ว
- โปรโตคอลมาตรฐานที่ทำงานในชั้นนี้ ก็เหมือนกับชั้นกายภาพคือ มี IEEE 802.2LLC, IEEE 802.3 Ethernet, IEEE 802.5 Token ring, FDDI และ X.25

ชั้นที่ 1 ฟิสิคอลเลเยอร์ (Physical Layer)

เลเยอร์ที่อยู่ล่างสุดคือ ชั้นกายภาพ (Physical Layer) เลเยอร์นี้จะรับผิดชอบเกี่ยวกับการส่งข้อมูลที่เป็นบิต หรือ 0 กับ 1 ในระบบเลขฐานสอง (Binary) ชั้นนี้จะรับข้อมูลจากเลเยอร์ที่ 2 หรือชั้นเชื่อมโยงข้อมูล (Data Link Layer) ซึ่งข้อมูลชุดหนึ่งจะเรียกว่า “เฟรม (Frame)” และทำการส่งเฟรมของข้อมูลนี้ที่ละบิตแบบเรียงตามลำดับ เหตุการณ์นี้จะเกิดขึ้นทางฝั่งสถานีที่ส่งข้อมูล ส่วนทางฝั่งสถานีรับข้อมูลชั้นกายภาพก็

จะทำการรับข้อมูลที่ส่งมาทีละบิตและจัดส่งผ่านข้อมูลเป็นบิตนี้ต่อไปยังชั้นเชื่อมโยงข้อมูลเพื่อทำการโพรเซสส์ต่อไป

สรุปแบบอ้างอิง OSI

Application Layer - ชั้นที่เจ็ดเป็นชั้นที่อยู่ใกล้ผู้ใช้มากที่สุดและเป็นชั้นที่ทำงานส่งและรับข้อมูลโดยตรงกับผู้ใช้ ตัวอย่างเช่น ซอร์ฟแวร์โปรแกรม ต่างๆ ที่อาศัยอยู่บนเลเยอร์นี้ เช่น DNS, HTTP, Browser เป็นต้น

Presentation Layer - ชั้นที่หกเป็นชั้นที่รับผิดชอบเรื่องรูปแบบของการแสดงผลเพื่อโปรแกรมต่างๆ ที่ใช้งานระบบเครือข่ายทำให้ทราบว่าข้อมูลที่ได้เป็นประเภทใด เช่น รูปภาพเอกสาร ไฟล์วีดีโอ

Session Layer - ชั้นที่ห้านี้ทำหน้าที่ในการจัดการกับเซสชั่นของโปรแกรม ชั้นนี้เองที่ทำให้ในหนึ่งโปรแกรม ยกตัวอย่างเช่น โปรแกรมค้นดูเว็บ (Web browser) สามารถทำงานติดต่ออินเทอร์เน็ตได้พร้อมๆกันหลายหน้าต่าง

Transport Layer - ชั้นนี้ทำหน้าที่ดูแลจัดการเรื่องของความผิดพลาดที่เกิดขึ้นจากการสื่อสาร ซึ่งการตรวจสอบความผิดพลาดนั้นจะพิจารณาจากข้อมูลส่วนที่เรียกว่า checksum และอาจมีการแก้ไขข้อผิดพลาดนั้นๆ โดยพิจารณาจาก ฝั่งต้นทางกับฝั่งปลายทาง (End-to-end) โดยหลักๆแล้วชั้นนี้จะอาศัยการพิจารณาจาก พอร์ต (Port) ของเครื่องต้นทางและปลายทาง Network Layer - ชั้นที่สามจะจัดการการติดต่อสื่อสารข้ามเน็ตเวิร์ค ซึ่งจะเป็นการทำงานติดต่อข้ามเน็ตเวิร์คแทนชั้นอื่นๆ ที่อยู่ข้างบน

Data Link Layer - ชั้นนี้จัดเตรียมข้อมูลที่จะส่งผ่านไปบนสื่อตัวกลาง

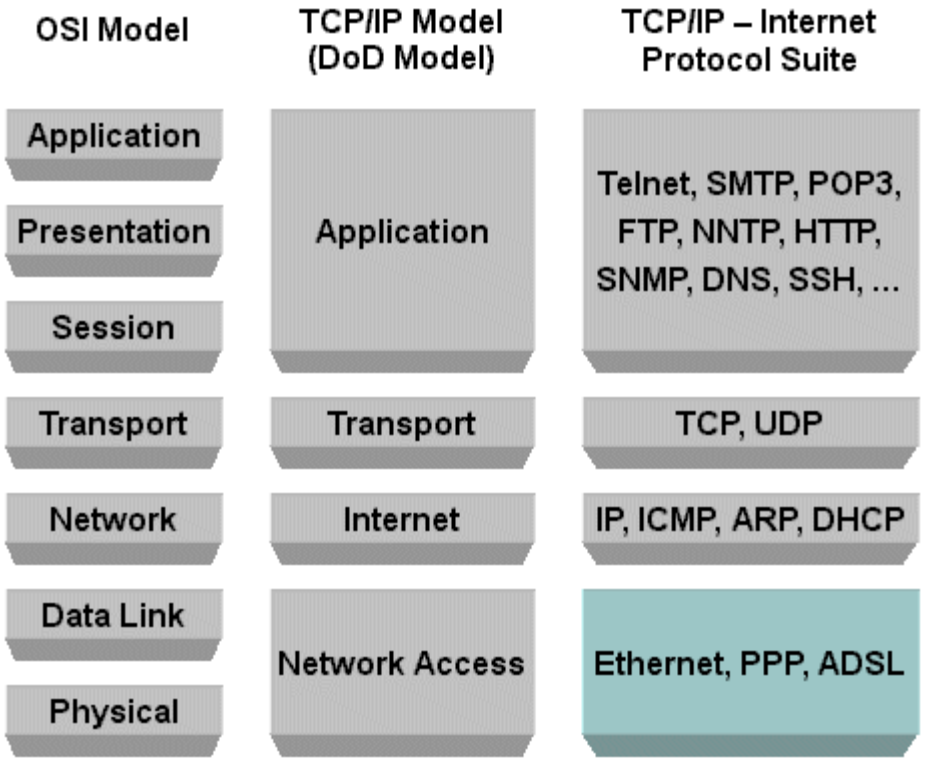
Physical Layer - ชั้นสุดท้ายเป็นชั้นของสื่อที่ใช้ในการติดต่อสื่อสาร ซึ่งอาจจะเป็นทั้งแบบที่ใช้สายหรือไม่ใช้สาย ตัวอย่างของสื่อที่ใช้ได้แก่ Shield Twisted Pair (STP), Unshield Twisted Pair (UTP), Fibre Optic และอื่นๆ

โปรโตคอล TCP/IP

ชุดโปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) ได้ถูกพัฒนามานานแล้วกว่า 20 ปีซึ่งเริ่มจากการวิจัยที่สนับสนุนโดยกระทรวงกลาโหมสหรัฐฯ จุดประสงค์ของการวิจัยนี้ก็เพื่อเชื่อมต่อคอมพิวเตอร์ที่ต่างแพลตฟอร์มกันให้สามารถสื่อสารกันผ่านเครือข่ายได้ ซึ่งสามารถทำได้โดยการ

แบ่งโปรโตคอลเป็นชั้นและเป็นการแยกการทำงานของแอพพลิเคชัน ของผู้ใช้ออกจากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลผ่านเครือข่าย ชุดโปรโตคอลนี้จะมีการจัดรูปแบบที่แตกต่างจากแบบอ้างอิง OSI เล็กน้อย

การออกแบบชุดโปรโตคอล TCP/IP จะมุ่งเน้นไปที่การเชื่อมต่อระหว่างระบบที่ต่างกันในขณะที่แบบอ้างอิง OSI จะเน้นไปที่การแบ่งการทำงานของโปรโตคอลออกเป็นชั้นๆ การออกแบบ TCP/IP ยังคงเป็นแบบชั้นๆ เหมือนกัน แต่เมื่อถึงตอนทำจริงๆ ก็ให้ขึ้นอยู่กับการตัดสินใจของผู้ออกแบบ ซึ่งเป็นผลให้ชุดโปรโตคอล OSI เหมาะสำหรับใช้อธิบายการสื่อสารระหว่างคอมพิวเตอร์ในเครือข่ายได้ดีกว่า ในขณะที่ชุดโปรโตคอล TCP/IP เป็นที่นิยมมากกว่าการนำไปใช้จริง



รูปที่ 5.1.2 เลเยอร์ของการสื่อสาร 7 ขั้นตอน

ที่มา : http://bit.kuas.edu.tw/~csshieh/teach/np/tcpip/dod.gif

แอพพลิเคชันเลเยอร์ (Application Layer) จะเป็น Application protocol ที่ทำหน้าที่เชื่อมต่อกับผู้ใช้และให้บริการต่างๆ เช่น FTP, Telnet, SNMP ฯลฯ

โฮสต์ทูโฮสต์เลเยอร์ (Host-to-Host Layer)

จะเป็น TCP หรือ UDP ที่ทำหน้าที่คล้ายกับ Layer ที่ 4 ของ OSI Model คือ ควบคุมการรับส่งข้อมูลจากปลายด้าน ส่งถึงปลายด้านรับข้อมูลและตัดข้อมูลออกเป็นส่วนย่อยให้เหมาะสมกับเครือข่ายที่ใช้รับส่ง

ข้อมูล รวมทั้งประกอบข้อมูลส่วนย่อยๆ นี้เข้าด้วยกันเมื่อถึงปลายทาง ซึ่งในชั้นนี้จะมี 2 โปรโตคอลคือ TCP (Transmission Control Protocol) และ UDP (User Datagram Protocol) ซึ่งมีลักษณะการรับส่งข้อมูลที่แตกต่างกัน

โปรโตคอล TCP จะใช้การรับส่งข้อมูลแบบ Connection-Oriented คือสร้างการเชื่อมต่อจะส่งข้อมูลเพื่อให้แน่ใจว่าจะส่งข้อมูลถึงปลายทางแน่นอน

ส่วนโปรโตคอล UDP จะใช้การรับส่งข้อมูลแบบ Connectionless คือไม่มีการสร้างการเชื่อมต่อปลายทางก่อน ข้อมูลจะถูกส่งออกไปทันที ซึ่งมีการคาดหวังว่าเครื่องปลายทางจะได้รับข้อมูลที่ส่งไป

อินเทอร์เน็ตเลเยอร์ (Internet Layer)

การทำงานในชั้นนี้จะเทียบเท่ากับการทำงานในชั้นเน็ตเวิร์คเลเยอร์ (Network Layer) ของแบบอ้างอิง OSI ซึ่งในชั้นนี้จะทำหน้าที่ในการส่งข้อมูลผ่านเครือข่ายต่างๆ ตามเส้นทางให้ถึงจุดหมาย ชุดข้อมูลที่อยู่ในชั้นนี้จะเรียกว่า “แพ็กเก็ต (Packet)” ซึ่งหน้าที่ของโปรโตคอลในชั้นนี้ก็ คือ ส่งแพ็กเก็ตข้อมูลนี้ให้ถึงปลายทางโดยการจัดเส้นทางที่เหมาะสมให้กับแพ็กเก็ต โปรโตคอลหลักที่ทำงานในชั้นนี้คือ IP (Internet protocol)

นอกจากโปรโตคอล IP แล้ว ยังมีโปรโตคอลอื่นๆ ที่ช่วยให้การทำงานของโปรโตคอล IP เป็นไปด้วยดี เช่น

- ICMP (Internet Control Message Protocol) : ใช้สำหรับการรายงานข้อผิดพลาดในระหว่างการรับส่งแพ็กเก็ต IP
- IGMP (Internet Group Message Protocol) : ใช้สำหรับการรายงานโฮสต์ที่เป็นสมาชิกในกลุ่มของมัลติคาสต์ (Multicast)
- ARP (Address Resolution Protocol) : ใช้สำหรับการแปลงหมายเลข IP เป็นที่อยู่ในเลเยอร์ที่ 2 (MAC Address)
- RARP (Address Resolution Protocol) : ทำงานในทางตรงกันข้ามกับ ARP
- ปัจจุบันโปรโตคอล IP ที่ใช้งานอยู่ในเครือข่ายอินเตอร์เน็ตจะเป็นเวอร์ชั่น 4 หรือเรียกสั้นๆ ว่า “IPV4”

เน็ตเวิร์คแอ็กเซสเลเยอร์ (Network Access Layer)

โปรโตคอล TCP/IP สามารถใช้ได้กับเน็ตเวิร์คหลายประเภท โดยเน็ตเวิร์คที่นิยมใช้งานมากที่สุดคือ อีเธอร์เน็ตนั่นเอง นอกจากนี้แพ็กเก็ตของ TCP/IP ยังสามารถส่งผ่านเน็ตเวิร์คอื่นๆ ได้ เช่น FDDI, ATM, X.25, FRAME Relay, PPP, SLIP และ ISDN เป็นต้น

ชุดโปรโตคอล IPX/SPX

บริษัทโนเวลล์ ได้พัฒนาชุดโปรโตคอล IPX/SPX (Internet Packet Exchange/Sequenced Packet Exchange) โปรโตคอลชุดนี้ได้พัฒนามาจากโปรโตคอล XNS (Xerox's Network System) ซึ่งเป็นโปรโตคอลที่ใช้ในอีเธอร์เน็ตในช่วงเริ่มแรก ชุดโปรโตคอล IPX/SPX เป็นที่นิยมมากในช่วงทศวรรษ 1980 โดยโปรโตคอล นี้เป็นส่วนหนึ่งของระบบปฏิบัติการเน็ตแวร์ (NetWare) ของบริษัท โนเวลล์ เน็ตแวร์ ถือได้ว่าเป็นระบบปฏิบัติการเครือข่ายมาตรฐานในช่วงแรกๆ ของการใช้งานเครือข่าย

ชุดโปรโตคอล Apple Talk

ความนิยมในการใช้คอมพิวเตอร์ของบริษัท แอปเปิ้ล (Apple) ได้เพิ่มขึ้นเรื่อยๆ ดังนั้นความต้องการที่จะเชื่อมต่อคอมพิวเตอร์เหล่านี้ให้เป็นเครือข่ายก็เป็นสิ่งที่หลีกเลี่ยงไม่ได้ บริษัทแอปเปิ้ลจึงได้พัฒนาโปรโตคอลแอปเปิ้ลทอล์ค (Apple Talk) ขึ้นมา ซึ่งโปรโตคอลชุดนี้รวมทั้งฮาร์ดแวร์ที่จำเป็นจะถูกติดตั้งกับคอมพิวเตอร์ของบริษัท แอปเปิ้ลทุกเครื่องก่อนที่จะขาย แนวคิดของบริษัท แอปเปิ้ลในการผลิตคอมพิวเตอร์ก็คือ จะผลิตคอมพิวเตอร์ให้ผู้ใช้สามารถใช้งานได้ง่ายที่สุด (User Friendly) การออกแบบเครือข่ายก็เช่นกัน การเชื่อมต่อคอมพิวเตอร์เข้ากับเครือข่ายง่ายขนาดเสียบสายสัญญาณเครือข่ายและเปิดเครื่องก็สามารถใช้ได้เลย

โปรโตคอลแอปเปิ้ลทอล์คเป็นโปรโตคอลที่ใช้สร้างเครือข่ายแบบเพียร์ทูเพียร์ และจะให้บริการเกี่ยวกับการแชร์ไฟล์และเครื่องพิมพ์ ดังนั้นคอมพิวเตอร์แต่ละเครื่องจะเป็นได้ทั้งเซิร์ฟเวอร์และไคลเอนท์ในเวลาเดียวกัน มีหลายบริษัทที่ได้นำเอาโปรโตคอลนี้ไปใช้ในระบบปฏิบัติการของตัวเอง ซึ่งเป็นผลให้คอมพิวเตอร์แอปเปิ้ลสามารถเชื่อมต่อเข้ากับเครือข่ายคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการอื่นได้ด้วย

ชุดโปรโตคอล NetBUEI

โปรโตคอล NetBUEI (ย่อมาจาก NetBIOS Extended User Interface ส่วน NetBIOS ย่อมาจาก Network Basic Input/Output System) ซึ่งเป็นชุดโปรโตคอลที่พัฒนาโดย IBM และเริ่มใช้เมื่อปี 1985 โปรโตคอลนี้มีขนาดเล็กเมื่อเทียบกับโปรโตคอลอื่นๆ แต่มีประสิทธิภาพสูง และเหมาะสำหรับเครือข่ายขนาดเล็ก

โปรโตคอล NetBUEI จะเทียบได้กับชั้นเครือข่ายและชั้นเคลื่อนย้ายข้อมูลของแบบอ้างอิง OSI โปรโตคอลนี้จะสร้างการเชื่อมต่อระหว่างคอมพิวเตอร์สองเครื่อง แล้วทำให้สองเครื่องนี้สามารถแลกเปลี่ยนข้อมูลกันได้

IP ADDRESS

หมายเลข IPADDRES เป็นแอดเดรสที่ผู้ติดตั้งระบบเครือข่ายจำเป็นต้องกำหนดให้กับเครื่องคอมพิวเตอร์ เพื่อใช้บ่งบอกตำแหน่งที่อยู่ของเครื่องคอมพิวเตอร์ในระบบ

มาตรฐานของ IP ADDRESS ปัจจุบันเป็นมาตรฐานเวอร์ชั่น 4.0 ซึ่งได้กำหนด IP ADDRESS มีทั้งหมด 32 บิต หรือ 4 ไบต์ แต่ละไบต์จะถูกคั่นด้วจุด (.) ตัวอย่างเช่น 192.168.5.11 แต่อย่างไรก็ดี ภายในหมายเลขที่เราเห็นนี้ยังถูกแบ่งออกเป็น 2 ส่วน ดังนี้

- ส่วนแรกเราเรียกว่า หมายเลข Network Address หรือ Subnet Address
- ส่วนที่สองเราเรียกว่า หมายเลข Host Address

Subnet Mask

เป็นพารามิเตอร์อีกตัวหนึ่งที่ต้องระบุควบคู่กับหมายเลข IP Address หน้าที่ของ Subnet Mask ก็คือการช่วยในการแยกแยะว่าส่วนใดภายในหมายเลข IP Address เป็น Network Address และส่วนใดเป็นหมายเลข Host Address

ข้อกำหนดเกี่ยวกับการกำหนดเน็ตเวิร์คแอดเอรส

เครื่องคอมพิวเตอร์ทุกเครื่องที่เชื่อมต่ออยู่ในเน็ตเวิร์คเซ็กเมนต์เดียวกัน จะต้องได้รับการกำหนดให้มีหมายเลข Network Address เหมือนกัน และหมายเลข Network Address ดังกล่าวจะต้องไม่ซ้ำกับเครื่องคอมพิวเตอร์ในเซกเมนต์อื่น

IP ADDRESS ในคลาสต่าง ๆ

เพื่อความเป็นระเบียบ ทางองค์กรกลางที่ดูแลเรื่องของ IP Address จึงได้มีการจัดคลาส (Class) หรือหมวดหมู่ของหมายเลข IP Address ไว้ทั้งหมด 5 คลาส โดยคลาสของแอดเดรสจะเป็นตัวกำหนดว่าบิตใดบ้างในหมายเลข IP Address ที่จะต้องถูกใช้เพื่อเป็น Network Address และบิตใดบ้างที่ต้องถูกใช้เพื่อเป็น Host Address นอกจากนั้นคลาสยังเป็นตัวกำหนดอีกด้วยว่า จำนวนของเน็ตเวิร์คเซ็กเมนต์ที่มีได้ในคลาสนั้นๆ มีเท่าไร และจำนวนของเครื่องคอมพิวเตอร์ที่สามารถมีได้ภายในเน็ตเวิร์คเซ็กเมนต์นั้นๆ มีเท่าไร

ตารางต่อไปนี้จะแสดงให้เห็นถึงตำแหน่งของไบต์ที่ถูกนำมาใช้เป็น Network Address และ Host Address ของหมายเลข IP Address ในคลาส A, B และ C

คลาส (Class)	IP Address	Network Address	Host Address
A	w.x.y.z	w	x.y.z
B	w.x.y.z	w.x	y.z
C	w.x.y.z	w.x.y	z

คลาส A

แอดเดรสในคลาส A จะถูกนำไปกำหนดให้กับระบบเครือข่ายขนาดใหญ่มากที่มีจำนวนเครื่องคอมพิวเตอร์อยู่เป็นจำนวนมาก ข้อกำหนดของคลาส A มีอยู่ว่า

- ไบต์แรกที่อยู่ด้านซ้ายสุด (8 บิต) จะถูกกันไว้เป็น Network Address และสามไบต์สุดท้ายที่เหลือ (อีก 24 บิต) จะถูกใช้เป็น Host Address
- บิตซ้ายสุดในไบต์แรกด้านซ้ายสุดจะต้องมีค่าเป็นศูนย์เสมอ ส่วนอีก 7 บิตที่เหลือในไบต์แรกด้านซ้ายสุดจะถูกใช้คำนวณเป็น Network Address

คลาส B

แอดเดรสในคลาส B มักถูกนำไปกำหนดให้กับระบบเครือข่ายขนาดปานกลางไปจนถึงขนาดใหญ่ที่มีจำนวนเครื่องคอมพิวเตอร์อยู่มากพอสมควร ข้อกำหนดของคลาส B มีอยู่ว่า

- สองไบต์แรกที่อยู่ด้านซ้ายสุด (16 บิต) จะถูกกันไว้เป็น Network Address และสองไบต์สุดท้ายถัดมา (อีก 16 บิต) จะถูกใช้เป็น Host Address
- บิตซ้ายสุดสองบิตแรกในไบต์แรกด้านซ้ายสุดจะต้องมีค่าเป็น 1 0 เสมอ ส่วนอีก 14 บิตที่เหลือในสองไบต์แรกด้านซ้ายสุดจะถูกใช้คำนวณเป็น Network Address

คลาส C

แอดเดรสในคลาส C มักถูกนำไปกำหนดให้กับระบบเครือข่ายขนาดเล็กที่มีจำนวนเครื่องคอมพิวเตอร์อยู่ไม่มากนัก ข้อกำหนดของคลาส C มีอยู่ว่า

- สามไบต์แรกที่อยู่ด้านซ้ายสุด (24 บิต) จะถูกกันไว้เป็น Network Address ส่วนอีก 1 ไบต์สุดท้ายที่เหลือ (8 บิต) จะถูกใช้เป็น Host Address
- 3 บิตซ้ายสุดในไบต์แรกที่อยู่ด้านซ้ายสุดจะต้องมีค่าเป็น 1 1 0 ตามลำดับ ส่วนอีก 21 บิตที่เหลือจะถูกใช้คำนวณเป็น Network Address

คลาส D

แอดเดรสในคลาส D จะไม่ถูกนำมาใช้กำหนดให้กับเครื่องคอมพิวเตอร์ทั่วไป แต่จะถูกใช้สำหรับการส่งข้อมูลแบบมัลติคาสก์ (Multicast) ของบางแอพพลิเคชั่น

ข้อกำหนดของคลาส D มีอยู่ว่า บิตซ้ายสุด 4 บิตแรกในไบต์ซ้ายสุดจะต้องมีค่าเป็น 1 1 1 0 เสมอ ส่วนอีก 28 บิตที่เหลือจะถูกใช้กำหนด “แอดเดรสของกลุ่มเครื่อง” ที่ต้องการเข้ามาอยู่ในกลุ่มมัลติคาสก์เดียวกัน (Multicast Group)

แอดเดรสในคลาสนี้จะไม่มีการแบ่งแยกว่าบิตไหนเป็น Network Address หรือ Host Address เราจะสังเกตได้ว่า ไบต์ซ้ายสุดของแอดเดรสในคลาส D จะต้องมีค่าเป็น 224 เสมอ

คลาส E

คลาส E เป็นแอดเดรสที่ถูกสงวนเอาไว้ก่อน ยังไม่ได้ถูกใช้งานจริง แต่อาจถูกใช้ในอนาคต ข้อกำหนดมีอยู่ว่า 4 บิตซ้ายสุดในไบต์แรกด้านซ้ายจะต้องมีค่าเป็น 1 1 1 1 วิธีการสังเกตอย่างรวดเร็วว่า IP Address ที่ได้มาอยู่คลาสอะไร

- ถ้าไบต์แรกด้านซ้ายสุดเป็นตัวเลข 1-126 แสดงว่าเป็นหมายเลข IP Address ที่อยู่ในคลาส A
- ถ้าไบต์แรกด้านซ้ายสุดเป็นตัวเลข 128-191 แสดงว่าเป็นหมายเลข IP Address ที่อยู่ในคลาส B
- ถ้าไบต์แรกด้านซ้ายสุดเป็นตัวเลข 192-223 แสดงว่าเป็นหมายเลข IP Address ที่อยู่ในคลาส C

Default Subnet Mask ของแต่ละคลาส

IP Address แต่ละคลาสจะมีค่า Subnet Mask เป็นของตนเอง ดังนี้

- คลาส A จะมี Subnet Mask เป็น 255.0.0.0
- หรือในเลขฐานสองคือ 11111111.00000000.00000000.00000000
- คลาส B จะมี Subnet Mask เป็น 255.255.0.0
- หรือในเลขฐานสองคือ 11111111. 11111111.00000000.00000000
- คลาส C จะมี Subnet Mask เป็น 255.255.255.0
- หรือในเลขฐานสองคือ 11111111. 11111111. 11111111.00000000

การทำ Subnetting

การทำ Subnetting เป็นการนำเอา Network Address ที่มีอยู่ 1 Address มาแบ่งซอยออกเป็นหลายๆ Sub-Network Address เพื่อให้สามารถนำไปกำหนดให้กับเน็ตเวิร์คแต่ละเซ็กเมนต์ได้ เราได้ทราบแล้วว่าเน็ตเวิร์คแต่ละเซ็กเมนต์ที่ใช้โปรโตคอล TCP/IP ต้องการหมายเลข Network Address เป็นของตนเอง ซึ่งต้องเป็นค่าเฉพาะตัวที่ไม่ซ้ำกันกับ Network Address ของเซ็กเมนต์อื่นดังนั้นเราจึงกล่าวได้ว่าการทำ Subnetting คือการนำเอา Network Address ที่มีอยู่มาแบ่งซอยย่อยออกเป็นหลายๆ Subnet Address โดยให้จำนวนของ Subnet Address มากกว่าหรือเท่ากับจำนวนของเน็ตเวิร์กเซ็กเมนต์ที่มีอยู่

ตัวอย่างเช่น เราตั้งใจออกแบบให้หมายเลข IP Address ในเน็ตเวิร์กภายในมีการใช้งาน Private Address ที่ขึ้นต้นด้วย 10.0.0.0/8 โดยเน็ตเวิร์คภายในมีการแบ่งออกเป็นหลายๆ เซ็กเมนต์เราสามารถนำเอา 10.0.0.0/8 มาคำนวณแตกออกเป็นซับเน็ตแอดเดรสย่อยๆ เพื่อกำหนดให้กับเครื่องคอมพิวเตอร์ในแต่ละซับเน็ตได้

หลักพื้นฐานของการทำ Subnet

หลักการของการทำ Subnet มีอยู่ว่า ต้องขอยืมเอาบิต (bit) ในตำแหน่งที่แต่เดิมเคยเป็น Host Address มาใช้เป็น Sub-network Address ด้วยการแก้ไขค่า Subnet Mask ให้เป็นค่าใหม่ที่เหมาะสม

เมื่อมีการขอยืมเอาตำแหน่งบิตที่เคยเป็น Host Address มาใช้เป็น Sub-network Address ก็จะลดลงด้วย นอกจากนั้น ตำแหน่งที่เป็น Network Address ก็จะเปลี่ยนไป โดยจำนวนบิตที่ถูกกันไว้ให้เป็น Network Address ก็จะเพิ่มมากขึ้นด้วย เพราะได้ไปขอยืมตำแหน่งบิตจากฝั่งของ Host Address มาใช้ ดังนั้น ในเมื่อตำแหน่งของ Host Address และ Network Address เปลี่ยนไป ค่าของ Subnet Mask ก็จะต้องเปลี่ยนตามไปด้วยเพื่อให้สอดคล้องกัน

ปัญหาของการออกแบบให้มีค่าของ Subnet Mask เพียงค่าเดียวทั่วทั้งเน็ตเวิร์ค

การออกแบบให้เน็ตเวิร์คทุกๆ เซ็กเมนต์มีค่าของ Subnet Mask ที่เท่ากันตลอดนี้เราเรียกว่า Fixed Length Subnet Mask (FLSM) ซึ่งเป็นหลักการในการออกแบบแอดเดรสแบบเก่าที่ใช้กันมาแต่ดั้งเดิมในสมัยแรกของการถือกำเนิดของระบบเน็ตเวิร์ค ในทางปฏิบัติปัจจุบัน เราพบว่าหากยึดถือตามแนวทางการออกแบบแบบ FLSM นี้ในทุกๆ สถานการณ์ ผู้ดูแลเน็ตเวิร์คอาจพบกับ “ความไม่คล่องตัว” บางอย่างได้ เช่น ค่าของ Subnet Mask ที่คำนวณได้สามารถให้จำนวนของซับเน็ตแอดเดรสเพียงพอ แต่ไม่สามารถให้จำนวนของโฮสต์ต่อหนึ่งซับเน็ตได้อย่างเพียงพอ หรือในทางกลับกันค่าของ Subnet Mask ที่คำนวณได้สามารถให้จำนวนของโฮสต์ต่อหนึ่งซับเน็ตได้เพียงพอ แต่ไม่สามารถให้จำนวนของซับเน็ตแอดเดรสที่ต้องการได้ ในอีกกรณีหนึ่งก็คือ การที่จำนวนเครื่องคอมพิวเตอร์หรือเครื่องโฮสต์ในแต่ละเซ็กเมนต์มีค่าต่างกันมาก การออกแบบโดยให้ค่าของ Subnet Mask คงที่เท่ากันหมดทุกๆ เซ็กเมนต์ อาจเป็นการสิ้นเปลืองหมายเลขแอดเดรสในบางซับเน็ตไปโดยไม่จำเป็น

“ความไม่คล่องตัว” ข้างต้นสามารถถูกขจัดลงไปได้ด้วยการใช้เทคนิคการออกแบบSubnet mask แบบที่เรียกว่า “Variable Subnet Mask(VLSM)” ซึ่งเป็นการออกแบบโดยไม่จำเป็นต้องให้ค่าของ Subnet Mask มีค่าเท่ากันทุกๆ ซับเน็ต ในแต่ละซับเน็ต เราสามารถออกแบบให้ค่าของ Subnet mask มีค่าแตกต่างกันได้

Private Address

เน็ตเวิร์คเซ็กเมนต์ขององค์กรที่ต้องติดต่อกับอินเตอร์เน็ตภายนอก จะต้องใช้หมายเลข Network Address ที่เป็นหมายเลข Public IP Address ที่ได้รับการจัดสรรจาก ISP หรือจากหน่วยงานที่มีหน้าที่รับผิดชอบเรื่องแอดเดรสบนอินเตอร์เน็ต

สำหรับเน็ตเวิร์คภายในที่ไม่ได้เชื่อมต่อกับอินเตอร์เน็ตโดยตรง เราสามารถใช้หมายเลขแอดเดรสที่ขึ้นต้นด้วย IP Address ต่อไปนี้ได้ แอดเดรสดังกล่าวจะถูกสงวน (Reserved) ไว้สำหรับใช้ในเน็ตเวิร์คภายในเท่านั้น โดยบนเครือข่ายอินเตอร์เน็ตส่วนใหญ่นั้นจะไม่มีการใช้งานแอดเดรสที่ถูกสงวนไว้เป็น Private Address ดังกล่าวนี้

ช่วงของ IP Address	**คลาสของเน็ตเวิร์ค**	**จำนวนของเน็ตเวิร์คที่เป็นไปได้**
10.0.0.0 – 10.255.255.255	A	1 คลาส A
172.16.0.0 – 172.31.255.255	B	16 คลาส B
192.168.0.0 – 192.168.255.255	C	256 คลาส C

Private Address ข้างต้นได้รับการกำหนดไว้ในมาตรฐาน RFC หมายเลข 1918

ในการเลือกว่าจะใช้งานแอดเดรสกลุ่มไหนขึ้นกับปริมาณของเน็ตเวิร์คเซ็กเมนต์ และปริมาณโฮสต์ต่อหนึ่งเซ็กเมนต์ที่อยู่ภายในเป็นหลัก เราสามารถนำเอาหลักการของการทำ Subnetมาใช้งานกับแอดเดรสภายในกลุ่มต่างๆที่กล่าวมาข้างต้นนี้ได้เช่นกัน ตัวอย่างเช่น เน็ตเวิร์คภายในองค์กรเป็นเน็ตเวิร์คที่มีขนาดใหญ่ มักนิยมใช้งานแอดเดรส 10.0.0.0/8 แล้วใช้การแบ่ง Subnet เข้ามาช่วยแบ่งแอดเดรส 10 ออกเป็นซับเน็ตย่อยๆ เพื่อกำหนดให้กับแต่ละเซ็กมนต์อีกครั้งฃ

อย่างไรก็ดีเนื่องจากแอดเดรสที่กล่าวมาข้างต้นนี้ถูกกำหนดไว้ให้ภายในองค์กรของตนเท่านั้น ไม่สามารถติดต่อสื่อสารกับเน็ตเวิร์คอื่นๆ บนอินเตอร์เน็ตจริงได้ ดังนั้น ในการทำให้เครื่องคอมพิวเตอร์บนเน็ตเวิร์คภายในสามารถติดต่อกับอินเตอร์เน็ตได้ เราต้องใช้เทคนิควิธีการทำ Network Address

Translation (NAT) เข้ามาช่วยแปลงหมายเลข IP Address ต้นทางให้กลายเป็น Public IP Address จริงๆ เสียก่อน ก่อนถูกเร้าท์ต่อไปยังเครือข่ายอินเตอร์เน็ตได้

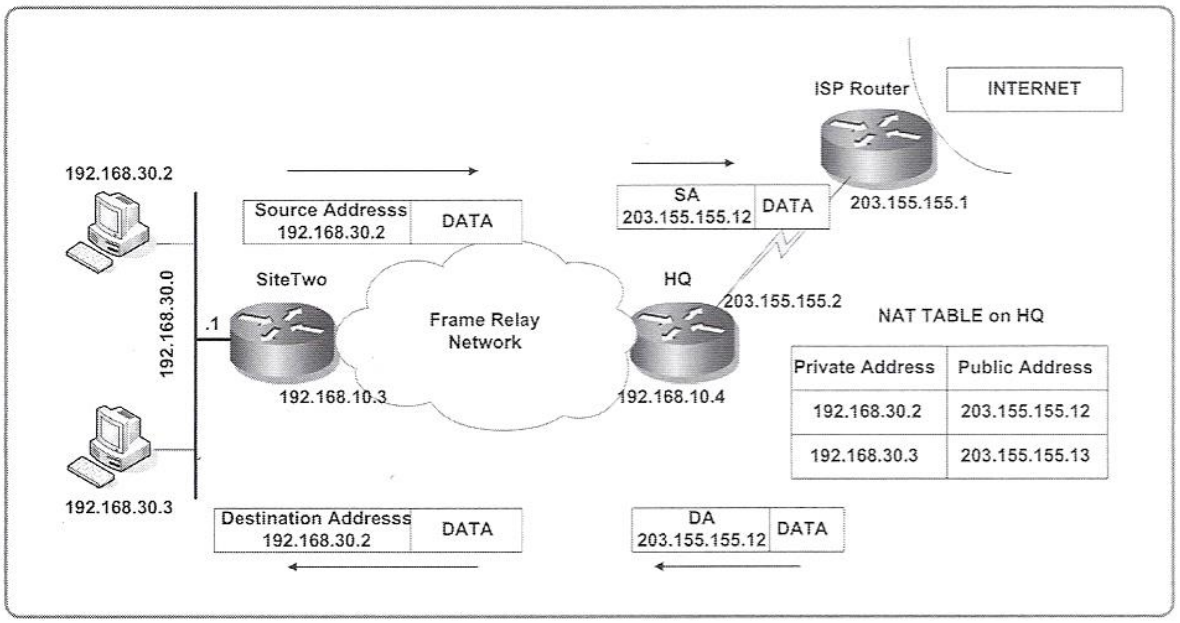
Network Address Translation (NAT)

เทคนิดการทำ NAT เป็นการแทนที่หมายเลข IP Address ต้นทางของแพ็กเก็ต IP ที่วิ่งผ่านอุปกรณ์ออกไปให้เป็นหมายเลข IP Address ที่กำหนด

มีอุปกรณ์อยู่ด้วยกันหลายประเภทที่สามารถทำ NAT ได้ ได้แก่ เร้าท์เตอร์ และไฟล์วอลล์เป็นต้น โดยมีลักษณะการทำ NAT กว้างๆ เหมือนกันคือ

- Static NAT : เป็นการทำ NAT แบบหนึ่งต่อหนึ่ง (One to One) ความหมายก็คือ กำหนดให้หมายเลข Private Address เบอร์นี้ จะถูกแทนที่ (Translate) ออกไปเป็นหมายเลข Public Address เบอร์อะไร
- Dynamic NAT : เป็นการทำ NAT แบบหลายๆ แอดเดรสต่อหนึ่งแอดเดรส (Many to One) ความหมายก็คือ มีการสร้าง NAT Pool ขึ้นมาหนึ่ง Pool แชร์ร่วมกัน ซึ่งภายใน Pool นั้นประกอบด้วยหลายๆ IP Address เมื่อแพ็กเก็ต IP ที่มี Private Address ต่างๆ วิ่งผ่านอุปกรณ์ที่ทำ NAT เข้ามา แพ็กเก็ต IP แรกจะถูกแทนที่ด้วยแอดเดรสใน Pool แอดเดรสหนึ่ง แพ็กเก็ต Ip ที่สองก็จะถูกแทนที่ด้วยอีกแอดเดรสหนึ่งที่อยู่ใน Pool ไปเรื่อยๆ
- Port Address Translation (PAT) หรือ NAT Overloading : เป็นการทำ NAT โดยอาศัย IP Address เพียงแอดเดรสเดียวแชร์ร่วมกันสำหรับทุกๆ Private Addressโดยอุปกรณ์ที่ทำ NAT ในลักษณะนี้จะใช้หมายเลข TCP Port เป็นตัวแบ่งแยก Private Address แต่ละแอดเดรสเอง

หลักการทำงานของ Static NAT



รูปที่ 11.1 หลักการพื้นฐานของ NAT

ที่มา : หนังสือเรียนรู้ระบบเน็ตเวิร์กจากอุปกรณ์ของ Cisco, 2548 : 498

จากรูปแสดงหลักการพื้นฐานของ NAT โดยแพ็กเก็ตที่วิ่งออกจากเน็ตเวิร์คภายในซึ่งใช้งาน Private Address (192.168.x.x) จะถูกแทนที่หรือ Translate หมายเลข Source IP Address จาก 192.168.x.x ให้กลายเป็นหมายเลข Public Address (ขึ้นต้นด้วย 203.155.155.x) ซึ่งทาง ISP จัดสรรมาให้

สำหรับกรณีของ Static Nat จะเห็นได้ว่าบนเร้าท์เตอร์ตัวริมขาออก (เร้าท์เตอร์ HQ) ได้ถูกเซ็ตคอนฟิกูเรชั่นไว้ให้มีการแมปแบบหนึ่งต่อหนึ่งคือ ถ้าหมายเลข Source Address เป็น 192.168.30.2 ก็ให้แทนที่ด้วย Source Address เป็น 203.155.155.12 และถ้าหมายเลข Source Address เป็น 192.168.30.3 ก็ให้แทนที่ด้วย Source Address เป็น 203.155.155.13

เมื่อแพ็กเก็ตถูกส่งกลับมาจากอินเตอร์เน็ต เร้าท์เตอร์ตัวริมก็จะเช็คดูในตาราง NAT Table และแทนที่ Destination Address 203.155.155.12 ด้วย Private Address จริงคือ 192.168.30.2 ให้โดยอัตโนมัติและส่งกลับเข้าไปยังเน็ตเวิร์คภายในให้ถึงเครื่องโฮสต์นั้นๆ

Static NAT นี้มักนิยมใช้กับเซิร์ฟเวอร์ที่ต้องการเปิดให้บริการต่อสาธารณะภายนอกซึ่งต้องการหมายเลข Public Address เป็นของตนเองที่แน่นอน