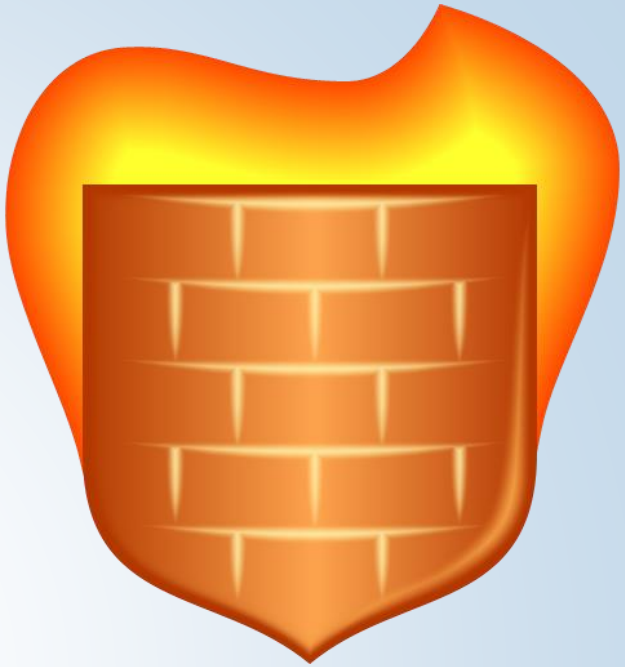
บทที่ 6 : Firewall Part1

สธ412 ความมั่นคงของระบบสารสนเทศ

อาจารย์อภิพงศ์ ปิงยศ

apipong.ping@gmail.com

Outline



- หลักการทำงานของไฟร์วอลล์
- โปรโตคอล TCP/IP
- โปรโตคอลในระดับแอพพลิเคชัน
- โปรโตคอลในระดับทรานสปอร์ต
- โปรโตคอลในระดับเน็ตเวิร์ค

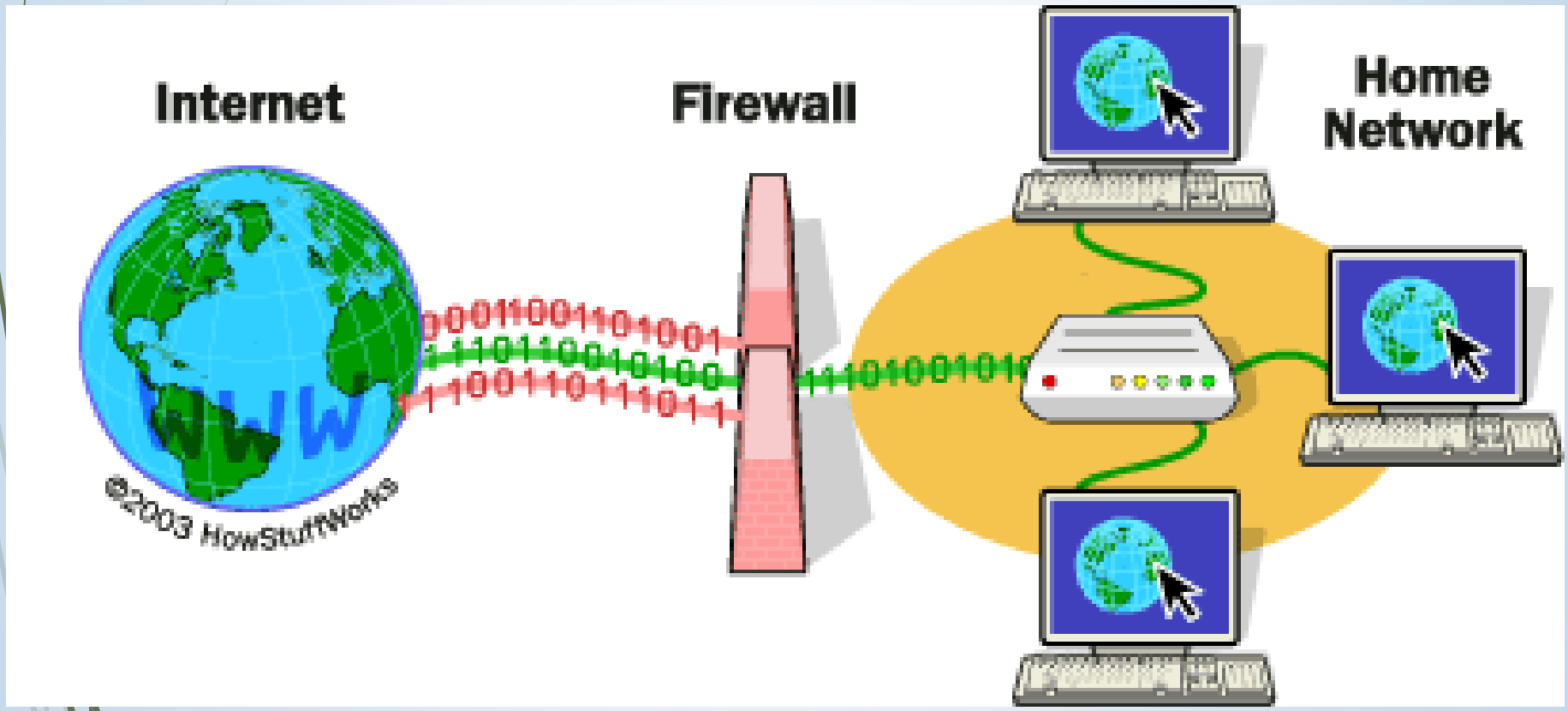
หลักการทำงานของไฟร์วอลล์

- เมื่อเปรียบเทียบการรักษาความปลอดภัยด้านกายภาพ ไฟร์วอลล์เปรียบเสมือนการล็อคประตู มีการใช้บัตรผ่านเข้าออก มียามเฝ้าทางเข้าออก
- หน้าที่หลักมีสองแบบคือ 1) กรองทราฟฟิกที่วิ่งเข้ามายังเครือข่ายจากภายนอก 2) ควบคุมการใช้งานของคอมพิวเตอร์ภายในเครือข่ายที่ต้องการติดต่อกับภายนอก

หลักการทำงานของไฟร์วอลล์ [2]

- หากไม่มีไฟร์วอลล์ก็เปรียบเสมือนการเปิดประตูบ้านทิ้งไว้ การติดตั้งไฟร์วอลล์จะเป็นการเพิ่มความยุ่งยากให้กับผู้บุกรุก
- การลงทุนกับไฟร์วอลล์ควรสัมพันธ์กับความเสียหายที่อาจเกิดขึ้นหากการโจมตีสำเร็จ

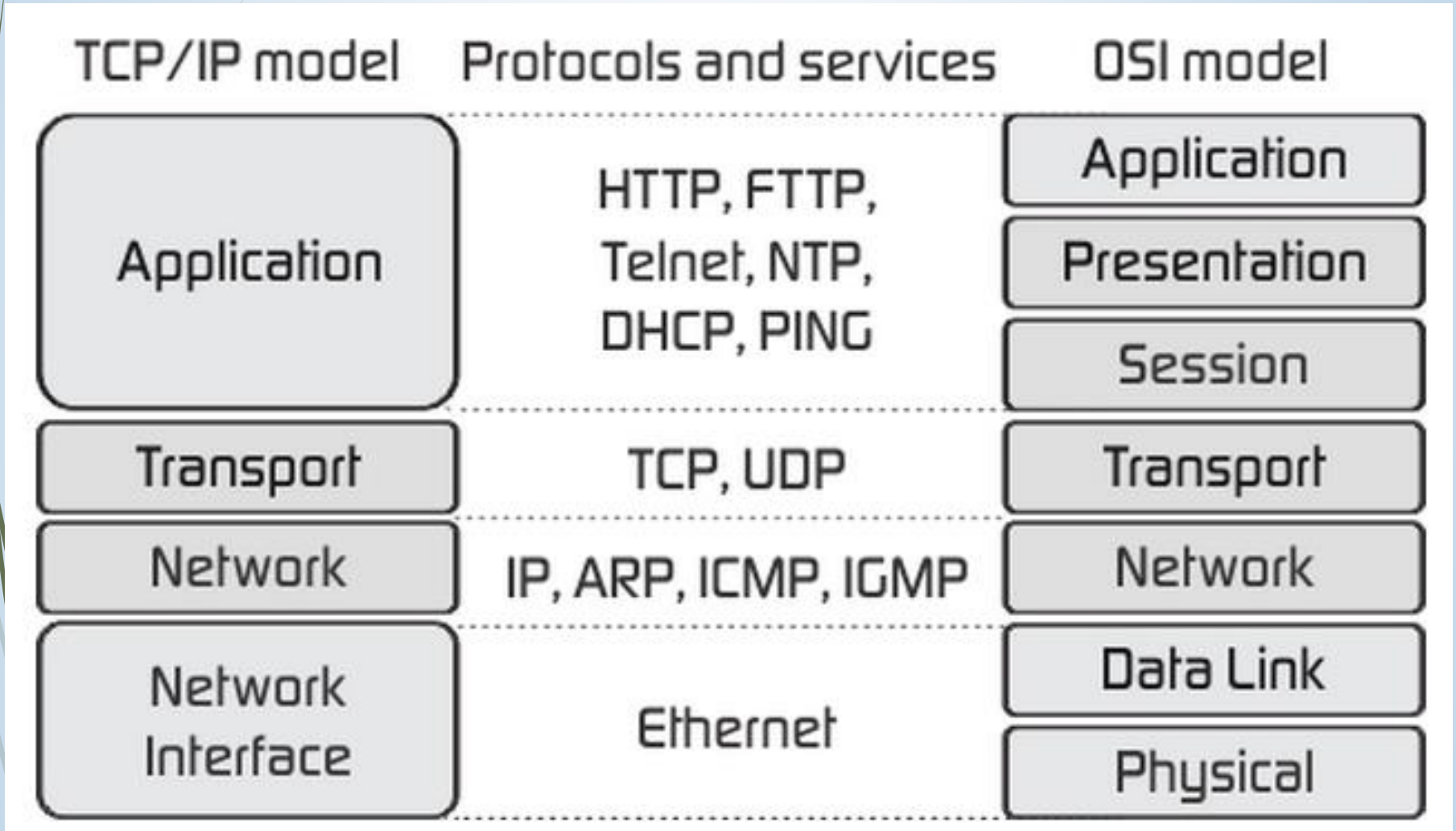
รูปแบบการเชื่อมต่อเน็ตเวิร์คไฟร์วอลล์



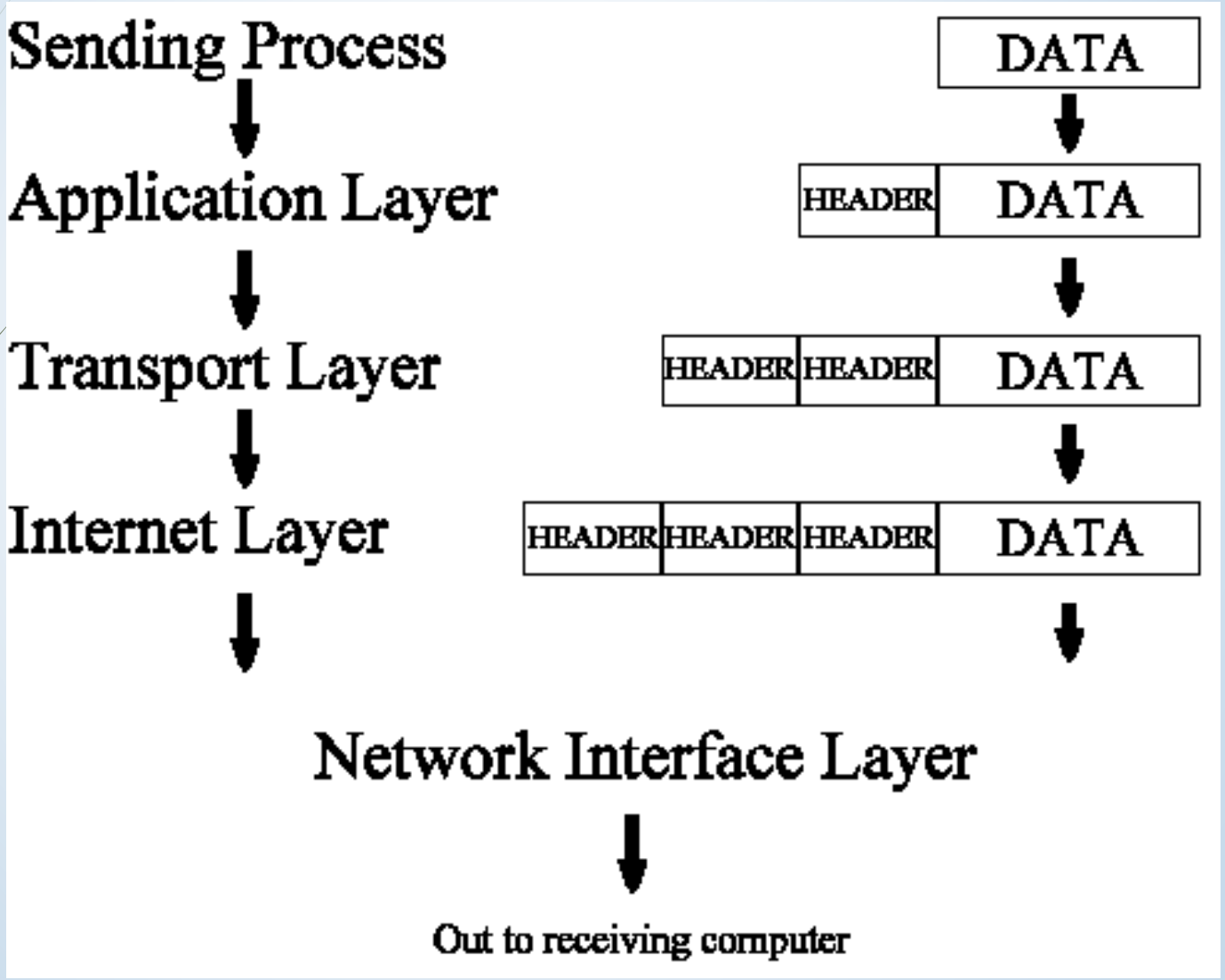
โปรโตคอล TCP/IP

- ก่อนที่จะกล่าวถึงรายละเอียดของไฟร์วอลล์ต่างๆนั้น จำเป็นที่จะต้องทบทวนหลักการทำงานของโปรโตคอล TCP/IP ก่อน
- เพราะไฟร์วอลล์จะใช้ข้อมูลจากแพ็คเก็ต แล้วพิจารณาว่าจะอนุญาตให้แพ็คเก็ตนี้ผ่านไปได้หรือไม่

เปรียบเทียบระหว่าง TCP/IP กับ OSI



การส่งข้อมูลในโปรโตคอล TCP/IP



โปรโตคอล TCP/IP [2]

- การทำความเข้าใจกระบวนการสื่อสารก่อนก็เพื่อที่จะเข้าใจหลักการทำงานของไฟร์วอลล์ในขั้นต่อไป
- ไฟร์วอลล์จะต้องเข้าถึงและวิเคราะห์ข้อมูลในส่วนเฮดเดอร์ในแต่ละเลเยอร์ เพื่อพิจารณาในการอนุญาตส่งผ่านแพ็คเก็ต
- ไฟร์วอลล์ส่วนใหญ่จะพิจารณาเฮดเดอร์ในเลเยอร์ที่ 3 (เน็ตเวิร์คเลเยอร์) แต่มีบางประเภทที่ใช้ข้อมูลในเลเยอร์ที่ 4-7

โปรโตคอลในระดับแอพพลิเคชัน

- เป็นโปรโตคอลในระดับชั้นบนสุด ที่รับผิดชอบฟอร์แมตของข้อมูลที่รับ-ส่ง เป็นต้น
- โปรโตคอลที่ใช้งานบ่อยๆ เช่น DNS, HTTP, HTTPS, SMTP, POP, IMAP,SNMP, FTP เป็นต้น

โปรโตคอลในระดับแอพพลิเคชัน : DNS

- **DNS** (Domain Name System) ทำหน้าที่คล้าย**สมุดโทรศัพท์**
- เมื่อต้องการสื่อสารกับคอมพิวเตอร์เครื่องอื่นในเครือข่าย คอมพิวเตอร์จะทำการสอบถาม IP เครื่องที่ต้องการไปที่ DNS Server
- หาก DNS Server ไม่มีข้อมูลของโดเมนหรือโฮสต์ที่ถูกร้องขอ ก็จะทำการค้นหาข้อมูลมาให้โดยอาจร้องขอไปยังเซิร์ฟเวอร์อื่นก็ได้

โปรโตคอลในระดับแอพพลิเคชัน : HTTP

- เป็นโปรโตคอลในการรับส่งไฟล์ HTML ที่เป็นภาษาในการแสดงเว็บเพจ หรือ WWW
- WWW แอพพลิเคชันทำงานแบบไคลเอนต์/เซิร์ฟเวอร์ คือ จะมีโฮสต์หนึ่งทำงานเป็นเซิร์ฟเวอร์ เรียกว่า Web Server ทำหน้าที่ให้บริการเว็บ ส่วนไคลเอนต์จะใช้โปรแกรม Web Browser ในการร้องขอ HTML และแสดงผลให้กับผู้ใช้

โปรโตคอลในระดับแอพพลิเคชัน : HTTPS

- เป็นโปรโตคอล HTTP ที่พัฒนาเพื่อให้สามารถเข้ารหัสได้ รายละเอียดได้กล่าวเอาไว้แล้วในบทที่ 5 เรื่อง Web Security

โปรโตคอลในระดับแอพพลิเคชัน : SMTP

- SMTP (Simple Mail Transfer Protocol) ทำหน้าที่ส่งอีเมลจากเมลเซิร์ฟเวอร์ของผู้ส่งไปยังเมลเซิร์ฟเวอร์ของผู้รับ

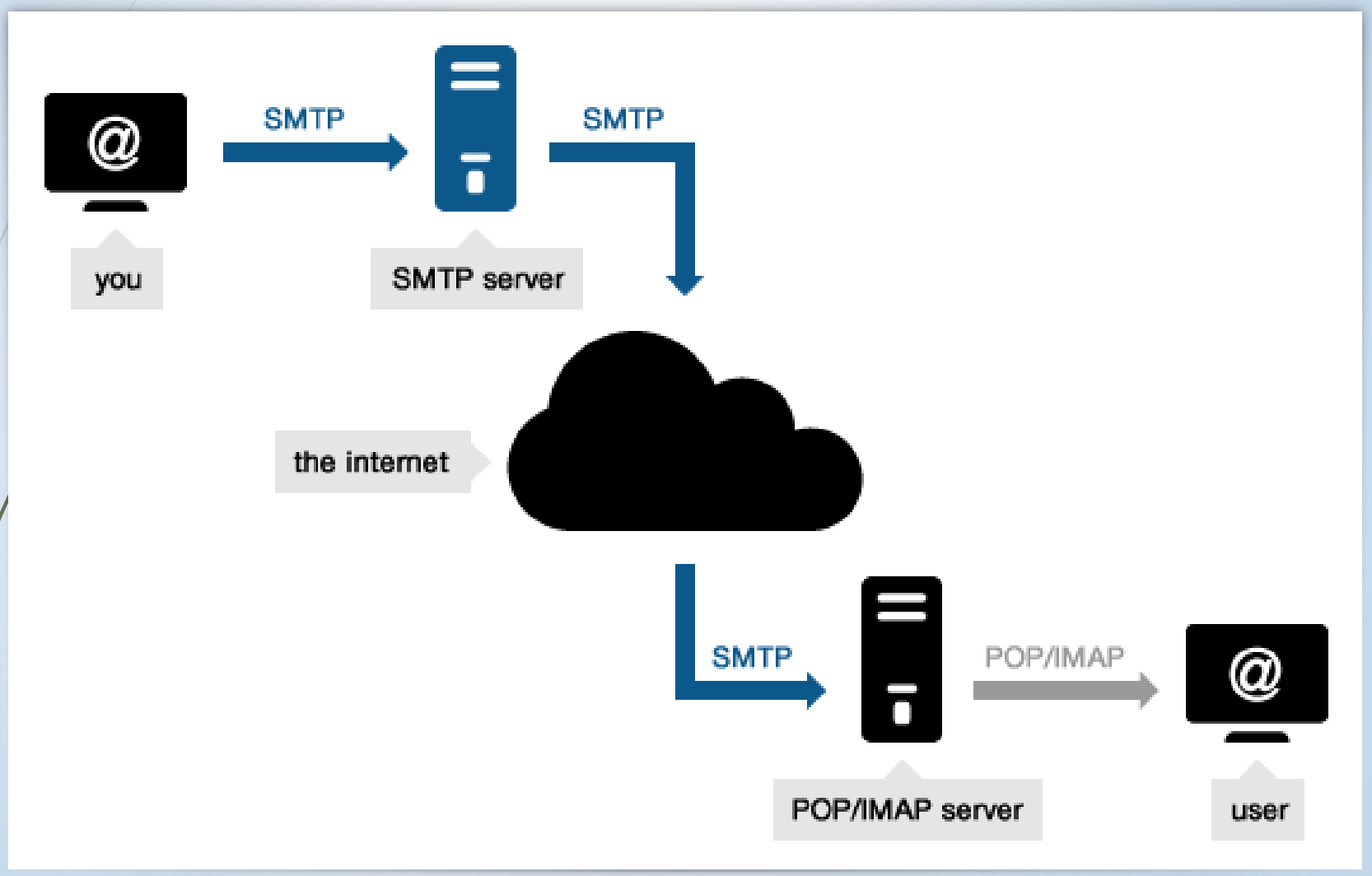
โปรโตคอลในระดับแอพพลิเคชัน : POP

- POP (Post Office Protocol) เป็นโปรโตคอลที่ใช้ในเมลไคลเอนต์ เช่น Outlook ทำหน้าที่ดึงอีเมลจากเมลเซิร์ฟเวอร์มาเก็บเอาไว้ในเครื่องไคลเอนต์ของผู้ใช้
- ปัจจุบันเป็นเวอร์ชัน 3 (POP3)

โปรโตคอลในระดับแอพพลิเคชัน : IMAP

- IMAP (Internet Message Access Protocol) เป็นโปรโตคอลที่ใช้สำหรับจัดการเมลบ็อกซ์
- ถูกคิดค้นมาเพื่อแก้ปัญหาของ POP3 ที่ทำหน้าที่แค่อ่านเมลหรือลบเมลเท่านั้น IMAPมีฟีเจอร์ต่างๆเพิ่มขึ้นมามากมาย เช่นสร้างโฟลเดอร์เพื่อจัดเก็บเมลได้ เก็บรายละเอียดว่าเมลได้ถูกเปิดอ่านไปแล้วหรือยัง เป็นต้น

Mail System Architecture



โปรโตคอลในระดับแอพพลิเคชัน : SNMP

- SNMP (Simple Network Management Protocol) ใช้แลกเปลี่ยนข้อมูลเกี่ยวกับการจัดการเครือข่ายระหว่างอุปกรณ์ต่างๆ
- ช่วยให้ผู้ดูแลระบบสามารถจัดการระบบเครือข่ายได้อย่างมีประสิทธิภาพ รวมไปถึงช่วยในการวิเคราะห์ปัญหาที่เกิดขึ้นภายในระบบด้วย
- สามารถดูข้อมูลเพิ่มเติมได้ที่

https://www.manageengine.com/network-monitoring/what-is-snmp.html#typical-snmp-communication

โปรโตคอลในระดับแอพพลิเคชัน : FTP

- FTP (File Transfer Protocol) ใช้สำหรับถ่ายโอนไฟล์ระหว่าง 2 เครื่อง
- มีการยืนยันสิทธิ์ของผู้ใช้โดยการกรอก Username และ Password เพื่อสร้างการเชื่อมต่อระหว่างเซิร์ฟเวอร์และไคลเอนต์ ให้สามารถดาวน์โหลดหรืออัพโหลดไฟล์ได้

โปรโตคอลในระดับทรานสปอร์ต

- โปรโตคอลในชั้นทรานสปอร์ตเป็นโปรโตคอลที่มีการเชื่อมต่อแบบ Process-to-Process
- มีแอพพลิเคชั่นหลายตัวที่ใช้โปรโตคอล TCP/IP จึงมีการใช้ Port และ Socket ในการช่วยแยกแยะแอพพลิเคชั่นต่างๆ
- ประกอบด้วย 2 โปรโตคอลหลักคือ TCP และ UDP

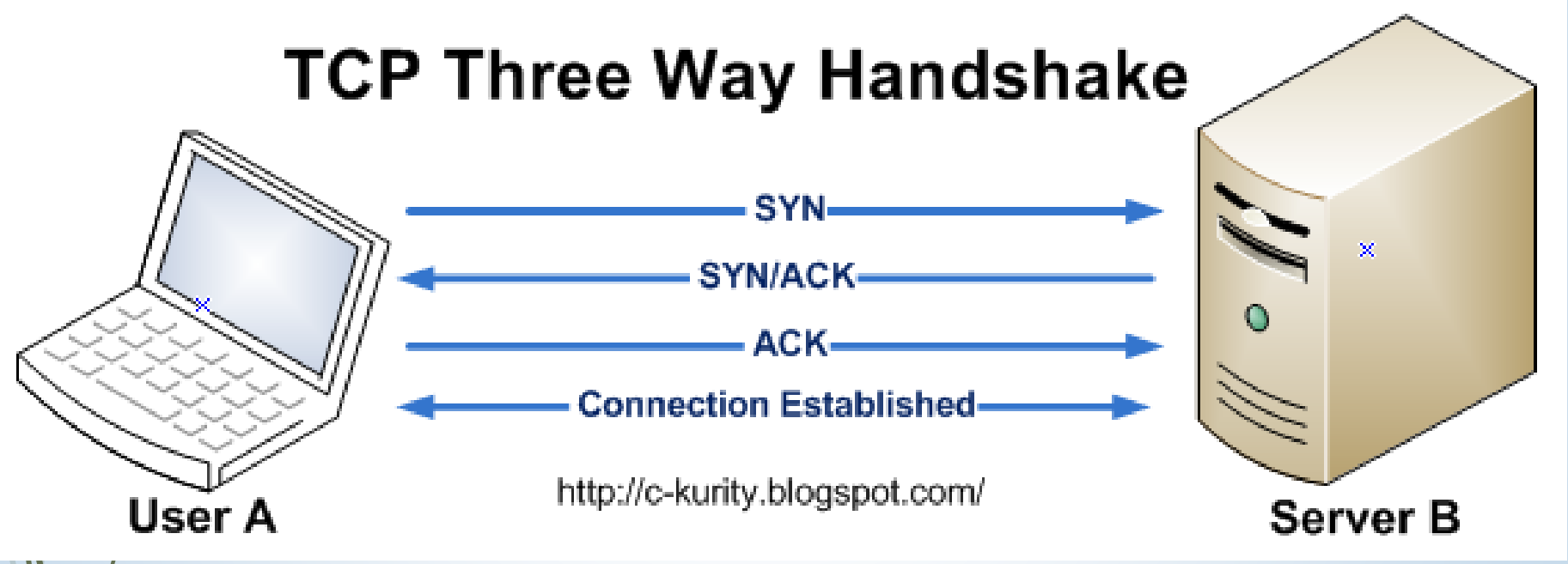
โปรโตคอลในระดับทรานสปอร์ต : TCP

- TCP (Transmission Control Protocol) เป็นโปรโตคอลที่ให้บริการแบบ Connection-Oriented ซึ่งเป็นการส่งข้อมูลที่เชื่อถือได้ รับประกันการส่งข้อมูลทุกแพ็คเก็ตถึงปลายทางได้อย่างแน่นอน
- ถ้าข้อมูลมีขนาดใหญ่จะแบ่งออกเป็นหลายแพ็คเก็ต โปรโตคอล TCP จะทำหน้าที่ควบคุมการรับส่งแพ็คเก็ตเหล่านี้

โปรโตคอลในระดับทรานสปอร์ต : TCP [2]

- การสร้างเซสชั่นของโปรโตคอล TCP เรียกว่า Three-Way Handshake ซึ่งมีหลักการคร่าวๆดังนี้
  - 1. โฮสต์ที่ต้องการส่งข้อมูลจะส่งข้อความไปบอกโฮสต์ปลายทางเพื่อแจ้งให้ทราบว่าต้องการส่งข้อมูล
  - 2. โฮสต์ปลายทางจะตอบตกลงกลับมาพร้อมรหัสที่จะใช้รับ-ส่งข้อมูล
  - 3. โฮสต์ต้นทางจะส่งแพ็คเก็ตพร้อมรหัสที่ได้รับ เพื่อยืนยันการเชื่อมต่อ
- หลังจากสร้างเซสชั่นแล้วจึงจะเริ่มกระบวนการรับ-ส่งข้อมูลจริง

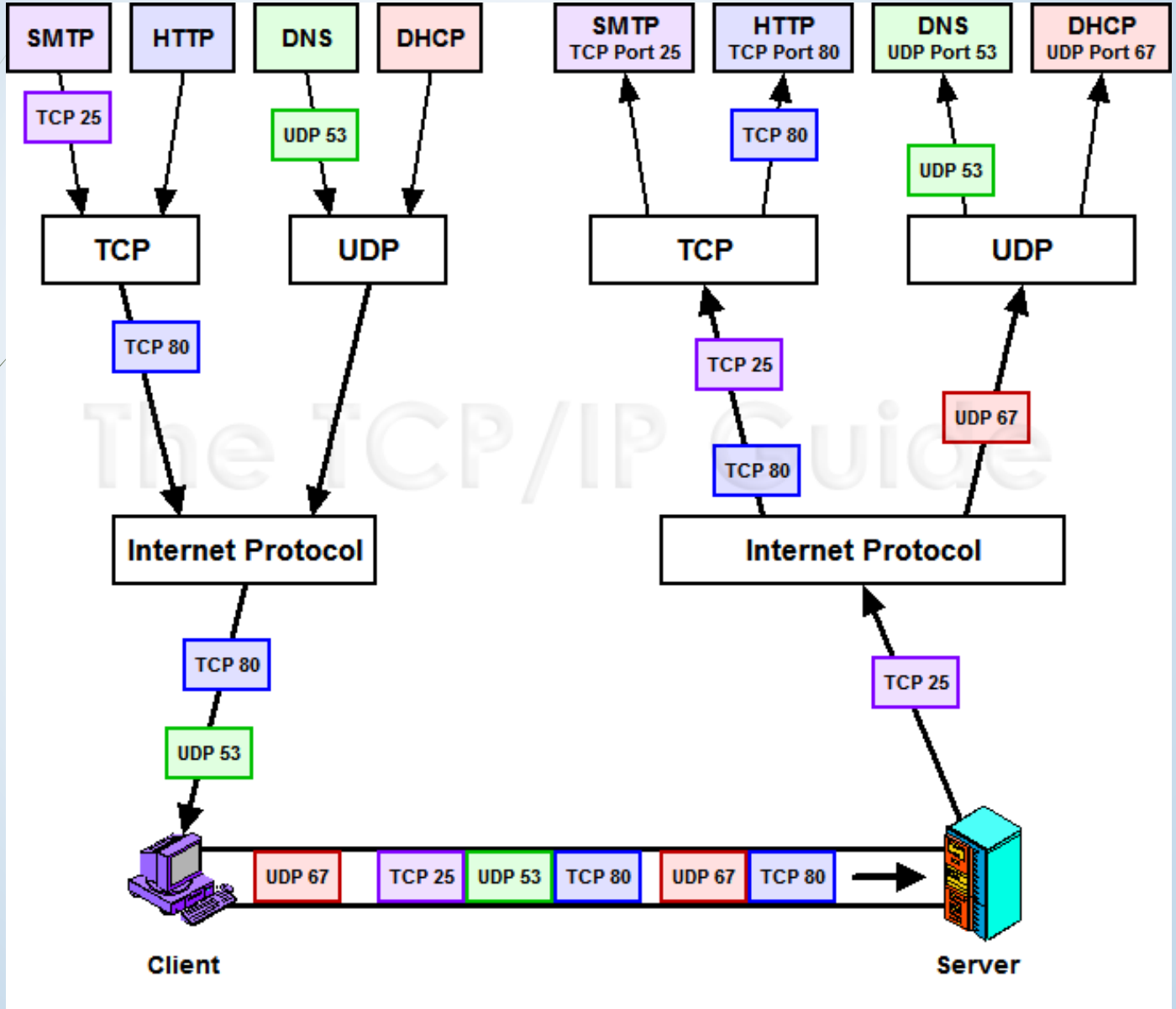
TCP Three-Way Handshake



โปรโตคอลในระดับทรานสปอร์ต : UDP

- UDP (User Datagram Protocol) ให้บริการข้อมูลแบบ Connectionless ซึ่งตรงกันข้ามกับ Connection-Oriented ของโปรโตคอล TCP
- การส่งข้อมูลจะเชื่อถือไม่ได้ ไม่มีการสร้างเซสชั่น และไม่มีกลไกการตอบกลับ
- ข้อดีคือส่งข้อมูลได้อย่างรวดเร็ว และใช้งานได้ดีในการส่งข้อมูลแบบ Broadcast และ Multicast เพราะการส่งข้อมูลในรูปแบบดังกล่าวจะสร้างเซสชั่นไม่ได้

การแบ่งข้อมูลแต่ละแอพพลิเคชั่นออกเป็นพอร์ต



โปรโตคอลในระดับเน็ตเวิร์ค

โปรโตคอลหลักที่ใช้ในชั้นสื่อสารนี้คือ

- IP (Internet Protocol) จัดการเกี่ยวกับการรับ-ส่งแพ็คเก็ต
- ICMP (Internet Control Message Protocol) ใช้วิเคราะห์และบริหารจัดการเครือข่าย

โปรโตคอลในระดับเน็ตเวิร์ค : IP

- IP (Internet Protocol) ทำหน้าที่รับ-ส่งแพ็คเก็ต, ค้นหาเส้นทาง (Routing)
- เป็นโปรโตคอลที่ให้บริการแบบ Connectionless หากมีปัญหาแพ็คเก็ตส่งไม่ถึงปลายทาง จะเป็นหน้าที่ของโปรโตคอลที่อยู่เลเยอร์สูงกว่าในการรับผิดชอบ
- อุปกรณ์ในเลเยอร์นี้คือ Router

โปรโตคอลในระดับเน็ตเวิร์ค : ICMP

- ICMP (Internet Control Message Protocol) ทำหน้าที่รายงานข้อผิดพลาดต่างๆที่เกิดขึ้นในระหว่างการส่งแพ็คเก็ต
- บริการแบบ Connectionless
- ฟังก์ชันสำคัญ เช่น
  - ประกาศข้อผิดพลาดของเครือข่าย
  - ประกาศความคับคั่งของเครือข่าย
  - ช่วยค้นหาข้อผิดพลาด (เช่นคำสั่ง Ping)
  - ประกาศการหมดเวลา (จะใช้ค่า Time to live : TTL เป็นตัววัด)

การใช้คำสั่ง Ping เพื่อค้นหาข้อผิดพลาด