

## บทที่ 6

# เครือข่ายคอมพิวเตอร์

# วัตถุประสงค์

---

หลังจากเรียนจบบทที่ 6 แล้ว นักศึกษาต้องสามารถ:

- อธิบายเหตุผลในการที่จะต้องมีเครือข่ายคอมพิวเตอร์
- อธิบายความแตกต่างของเครือข่าย 3 ประเภทคือ: LANs, MANs, และ WANs
- อธิบายแบบจำลอง OSI และ TCP/IP
- อธิบายประเภทต่างๆของอุปกรณ์เชื่อมต่อและ OSI layers ที่ใช้อุปกรณ์แต่ละชนิด
- อธิบายแบบจำลอง client-server

**6.1**

# ประเภทของเครือข่าย



# ขนาดของเครือข่าย

- **เครือข่ายคอมพิวเตอร์** เป็นระบบที่ประกอบด้วยองค์ประกอบย่อยๆ (เช่น เครื่องคอมพิวเตอร์ โมเด็ม ... ) ที่เชื่อมต่อกันโดยสื่อสายส่ง (transmission media) เช่น wire, cable, air
- **เครือข่ายคอมพิวเตอร์** อาจครอบคลุมพื้นที่ตามสภาพทางภูมิศาสตร์ที่มีขนาดเล็ก ขนาดปานกลาง หรือขนาดใหญ่กว้างขวาง
  - \* ถ้าครอบคลุมพื้นที่ขนาดเล็กเรียกว่า**เครือข่ายท้องถิ่น** (Local Area Network : **LAN**)
  - \* ถ้าครอบคลุมพื้นที่ขนาดปานกลางเรียกว่า**เครือข่ายเขตเมือง** (Metropolitan Area Network : **MAN**)



# ขนาดของเครือข่าย (ต่อ)

\* ถ้าครอบคลุมพื้นที่ขนาดใหญ่กว้างขวางเรียกว่าเครือข่าย

ขนาดใหญ่ (Wide Area Network : **WAN**)

- เครือข่ายคอมพิวเตอร์ทั้งสามประเภทสามารถเชื่อมต่อถึงกันและกันได้ ทำให้เกิดเครือข่ายขนาดใหญ่ขึ้นเรียกว่า **Internetwork** หรือ **Internet**
- **Model** คือข้อกำหนดรายละเอียดที่กำหนดโดยองค์มาตรฐานเพื่อเป็นแนวทางในการออกแบบเครือข่าย
- **Protocol** หมายถึงกฎเกณฑ์ที่กำหนดขึ้นเพื่อใช้ควบคุมการเชื่อมต่อระหว่างอุปกรณ์ต่างๆภายในเครือข่ายหรือภายใน Internet

**6.2**

# **แบบจำลอง OSI**

# Open Systems Interconnection : OSI

- เพื่อให้ทุกองค์ประกอบของเครือข่ายทำงานร่วมกันได้อย่างราบรื่น ใน Model จะต้องบ่งบอกและแสดงถึงความสัมพันธ์ระหว่างองค์ประกอบทั้งหมด ตลอดจนหน้าที่ของแต่ละองค์ประกอบด้วย
- Model ดังกล่าวคือ **Open Systems Interconnection** หรือ **OSI Model**
- OSI model ออกแบบโดยองค์กรที่มีชื่อว่า International Organization for Standard (ISO) หลักใหญ่ของ model คือเครื่องคอมพิวเตอร์หรืออุปกรณ์ที่แตกต่างกันต้องสามารถเชื่อมต่อและสื่อสารกันได้

# Seven Layers

- **OSI model** เป็นกรอบของการกำหนดชั้น 7 ชั้น (layer) เพื่อที่จะเป็นแนวทางให้ผู้ออกแบบเครือข่ายกำหนดหน้าที่ของแต่ละ layer ที่แยกกันแต่สัมพันธ์กันได้อย่างชัดเจน ทั้งเจ็ด layer เรียงลำดับจาก layer ที่เกี่ยวข้องกับฮาร์ดแวร์มากที่สุดไปน้อยที่สุดมีดังนี้

1. Physical Layer

2. Data Link Layer

3. Network Layer

4. Transport Layer

5. Session Layer

6. Presentation Layer

7. Application Layer    ดังรูปที่ 6.1





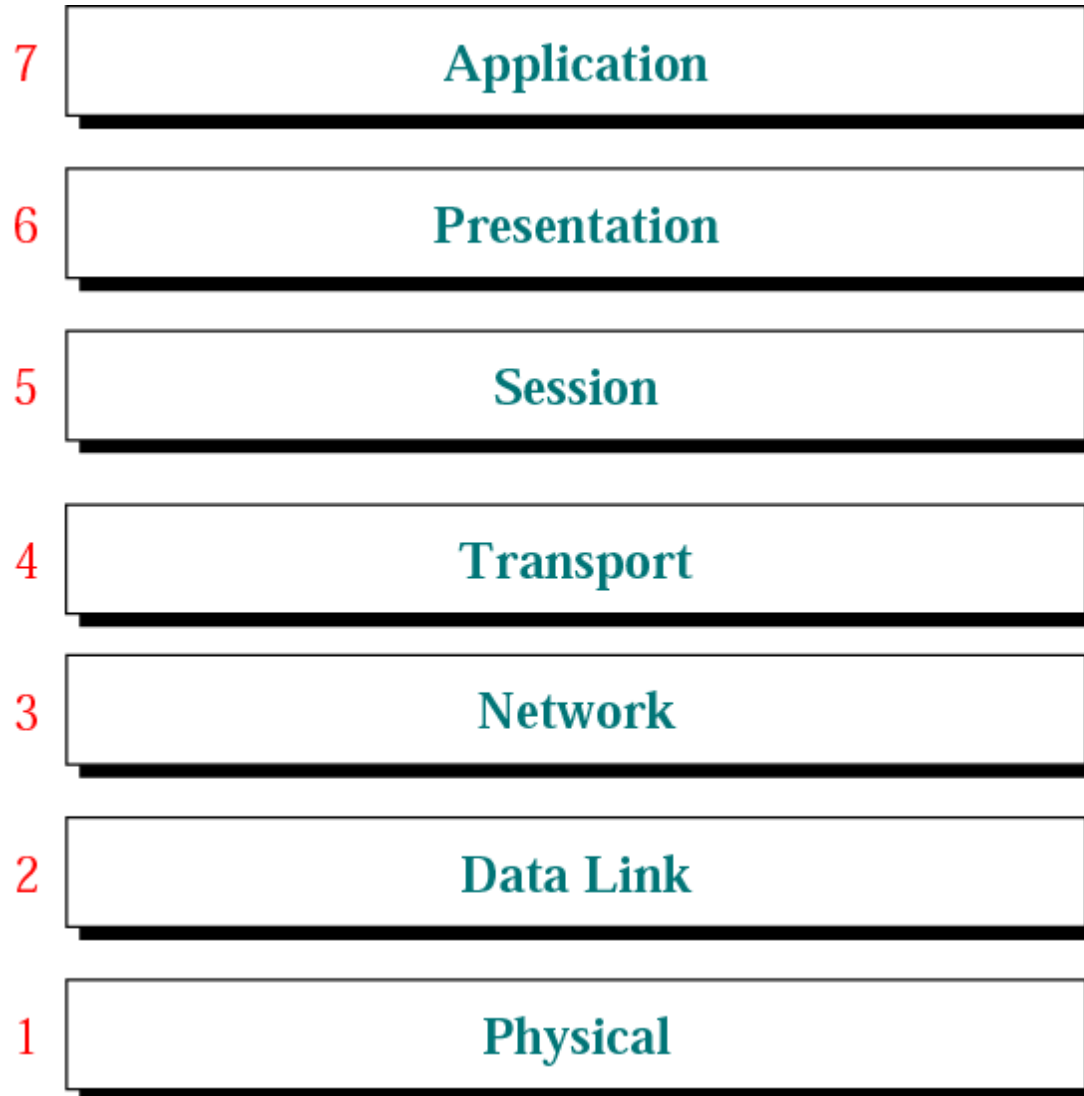


**Note:**

**The Open Systems Interconnection model is a theoretical model that shows how any two different systems can communicate with each other.**



## แบบจำลอง OSI

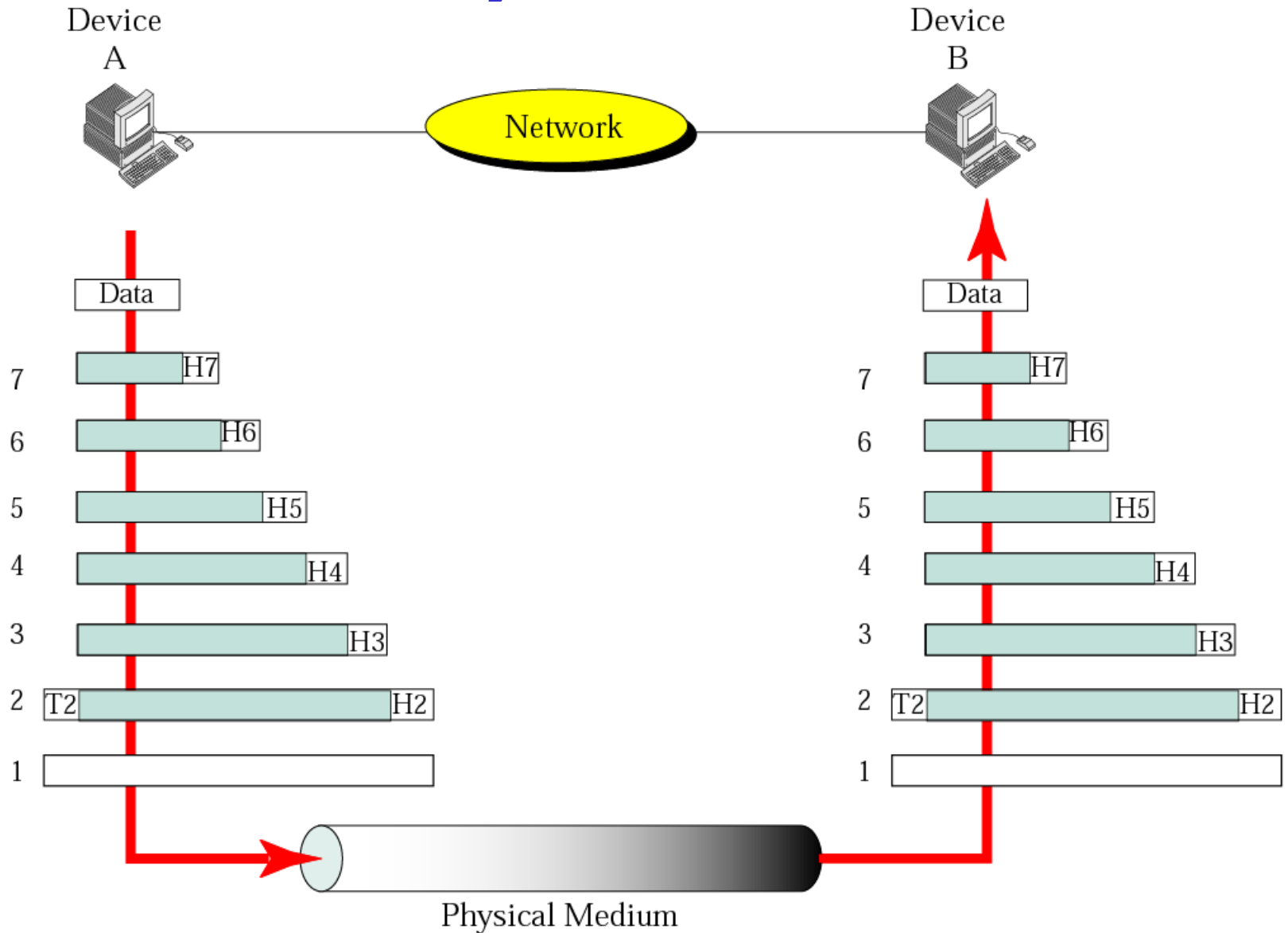


## Seven Layers (ต่อ)

- OSI model ไม่ได้บังคับว่าอุปกรณ์แต่ละชั้นที่เชื่อมต่อกับเครือข่าย จะต้องทำหน้าที่ครบทั้งเจ็ด layer แต่ละอุปกรณ์อาจต้องการเพียง 1 layer หรือ 2 layer หรือ 3 layer หรือทั้งเจ็ด layer ก็ได้ จำนวน layer ขึ้นอยู่กับหน้าที่ของอุปกรณ์และตำแหน่งที่ตั้งของอุปกรณ์นั้นๆ
- รูปที่ 6.2 แสดงให้เห็นถึงบทบาทของแต่ละ layer เมื่อมีการส่งข้อมูล (message) จากอุปกรณ์ A ไปยังอุปกรณ์ B
- ในขณะที่ข้อมูลเดินทางจาก A ไปยัง B ข้อมูลนี้อาจจะต้องผ่านจุดต่างๆ หลายจุด (node) จุดเหล่านี้โดยปกติจะเกี่ยวข้องกับสาม layer แรกใน OSI model เท่านั้น

## รูปที่ 6-2

# การส่งผ่านข้อมูลในแบบจำลอง OSI



# หน้าที่ของทั้งเจ็ดเลเยอร์

- **Application Layer:** มีหน้าที่อำนวยความสะดวกให้ผู้ใช้ (users) ซึ่งอาจเป็นคนหรือซอฟต์แวร์ก็ได้ ให้สามารถเข้าใช้เครือข่ายได้  
เลเยอร์นี้จะกำหนดการประยุกต์อันจะทำงานของผู้ใช้ง่ายขึ้น
- **Presentation Layer:** ทำหน้าที่หลายอย่างดังนี้
  - \* ตรวจสอบความถูกต้องของไวยากรณ์ (syntax) และความหมาย (semantics) ของข้อมูลข่าวสารที่แลกเปลี่ยนกันระหว่างผู้รับ (receiver) และผู้ส่ง (sender)

# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

- \* ทำให้ระบบที่แทนข้อมูลข่าวสารด้วยรหัสที่ต่างกัน สามารถสื่อสารกันได้เช่นระบบหนึ่งแทนข้อมูลด้วยรหัส ASCII สามารถสื่อสารกับอีกระบบหนึ่งซึ่งแทนข้อมูลด้วยรหัส Unicode เป็นต้น
- \* ทำการบีบอัดข้อมูล (compress) และขยายข้อมูลที่ถูกลบีอัด (decompress) เพื่อเพิ่มประสิทธิภาพในการส่งข้อมูลผ่านเครือข่าย

# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

\* ทำการเข้ารหัส (encrypt) และถอดรหัส (decrypt) ข้อมูล  
ที่ส่งเพื่อประโยชน์ในแง่ของความปลอดภัย

**ข้อสังเกต:** เทคโนโลยีเครือข่ายในปัจจุบันได้ทำการ implement หน้าที่  
เหล่านี้ในเลเยอร์อื่นเช่น หน้าที่การเข้ารหัสและการถอดรหัสทำใน  
network layer และ application layer เป็นต้น

- **Session Layer:** ทำหน้าที่ควบคุมการติดต่อสื่อสารระหว่างผู้ใช้ ทำการ  
สร้างจุดเริ่มการเชื่อมต่อ ควบคุมการเชื่อมต่อระหว่างการสื่อสารให้  
ราบรื่น ทำการจัดลำดับการสื่อสารให้เหมาะสม และ

# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

ที่สำคัญคือทำการกำหนดจุดที่เรียกว่า **synchronization points** สำหรับการสำรองข้อมูลที่ทำ การส่งไปมา โดยมีวัตถุประสงค์เพื่อไม่ให้ข้อมูลสูญหายในระหว่างการส่ง ถ้ามีการสูญหายก็จะสามารถนำข้อมูลที่สำรองส่งต่อไปได้ ณ จุดดังกล่าว จะทำการแบ่งข้อมูลที่ยาวๆ ออกเป็นส่วนย่อยๆ ถ้ามีข้อมูลสูญหายเกิดขึ้น จะได้ไม่ต้องส่งข้อมูลใหม่ทั้งหมด ระบบจะย้อนมาที่จุด **synchronization point** ล่าสุดแล้วทำการส่งข้อมูลจากจุดนั้น

**ข้อสังเกต:** เทคโนโลยีเครือข่ายในปัจจุบันไม่ได้ใช้ **session layer** แต่งานที่ทำ จะกำหนดให้ทำใน **application layer** แทน



# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

- **Transport Layer:** รับผิดชอบการส่งข้อมูลข่าวสารทั้งหมดจาก **ต้นทาง** ถึง **ปลายทาง** (source-to-destination หรือ end-to-end)
  - \* ทำการแบ่งข้อมูลที่ต้องการส่งออกเป็นส่วนย่อยๆ แต่ละส่วนเรียกว่า packet แล้วส่งไปยัง Network Layer จากนั้น NL ก็จะส่ง packet เหล่านี้ออกไปสู่ปลายทางทีละ packet แยกออกจากกัน อาจมีบาง packet หายไประหว่างทาง บาง packet ก็ไปถึงปลายทางไม่เป็นไปตามลำดับที่ส่ง ปัญหาเหล่านี้เป็นหน้าที่ของ TL ที่จะต้องจัดการให้เรียบร้อย

# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

- **Network Layer:** มีหน้าที่รับผิดชอบการส่ง packet ระหว่างจุดต้นทาง (original source) และจุดปลายทาง (final destination)
  - \* NL จะทำการเพิ่มส่วนหัว (header) เข้าไปยังข้อมูลที่ส่งมาจาก SL ซึ่งประกอบด้วย ที่อยู่ต้นทาง (source address) ที่อยู่ปลายทาง (destination address) ที่อยู่ดังกล่าวเป็น logical address (IP address) ซึ่งแตกต่างจาก physical address

# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

## ข้อสังเกต:

- \* สำหรับการสื่อสารประเภท global communication แล้ว logical address จะต้องไม่ซ้ำกัน (unique)
- \* เมื่อ packet ถูกส่งจากต้นทางสู่ปลายทาง ค่า physical address (ถูกเพิ่มโดย Data-Link Layer) จะเปลี่ยนแปลงไปตามสถานี (station) ที่ packet ผ่านไป แต่ logical address จะไม่มีการเปลี่ยนแปลงแต่อย่างใดตลอดระยะทางการส่ง

# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

- **Data-Link Layer:** ทำหน้าที่จัดเรียงบิตของข้อมูลที่จะส่งให้เป็นหน่วยย่อยๆ เรียกว่าเฟรม (frame) แต่ละเฟรมประกอบด้วยข้อมูลจาก network layer
  - \* DLL จะทำการเพิ่ม**ส่วนหัว**(header)และ**ส่วนต่อท้าย** (trailer) เพื่อระบุรายละเอียดของเฟรมให้สถานีต่างๆที่เฟรมจะผ่านไปได้รับรู้ โดยเฉพาะข้อมูลเกี่ยวกับที่อยู่ (address) ของสองสถานีที่อยู่ติดกัน คือ **ที่อยู่ของสถานีส่งกับที่อยู่ของสถานีรับ**

# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

- \* DDL รับผิดชอบเฉพาะการรับ-ส่งเฟรมจากสถานีหนึ่งไปยังอีกสถานีหนึ่งที่อยู่ติดกัน (node-to-node) เมื่อสถานีที่ไม่ใช่สถานีปลายทางได้รับเฟรมหนึ่งๆ มันจะทำการเปลี่ยน source address เป็น address ของมันเอง และเปลี่ยน destination address เป็น address ของสถานีที่อยู่ถัดไป
- \* DDL ยังมีหน้าที่รับผิดชอบเกี่ยวกับความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูลระหว่างสองสถานีที่ติดกันด้วย ข้อมูลที่เพิ่มเข้าไปในส่วนที่เป็น trailer จะทำหน้าที่หาข้อผิดพลาดและแก้ไขข้อผิดพลาดที่เกิดขึ้น

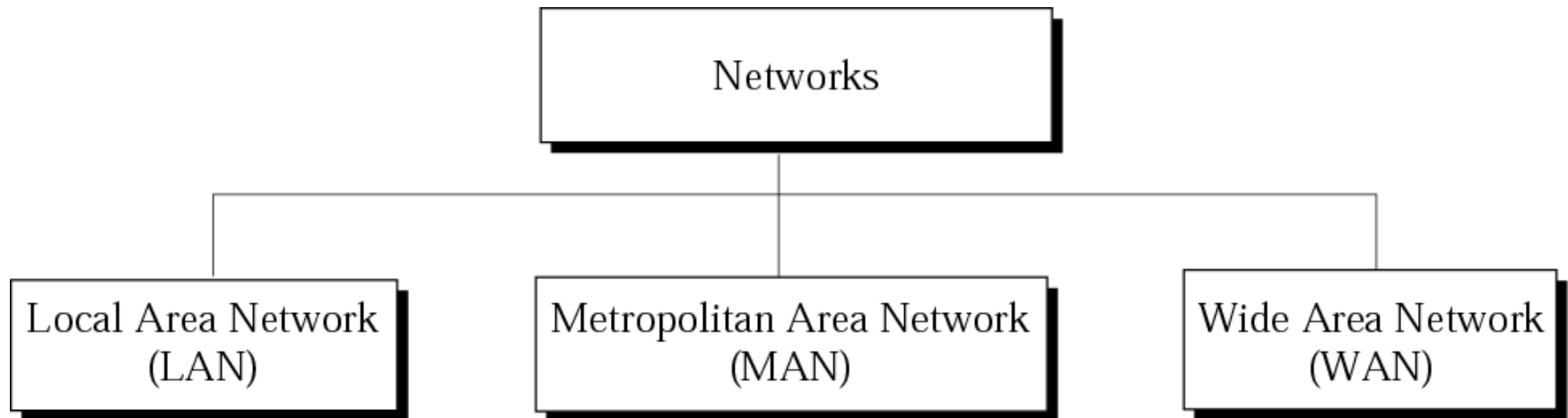
# หน้าที่ของทั้งเจ็ดเลเยอร์ (ต่อ)

- **Physical Layer:** ทำหน้าที่ส่ง**สายของบิต** (bit stream) ผ่านสื่อสายส่ง (physical medium)
  - \* PL เข้ารหัส (encode) และ ถอดรหัส (decode) บิตเป็น**กลุ่มของบิต** จากนั้น PL จะเปลี่ยน**สายของบิต**เป็นสัญญาณ (signal)
  - \* PL ทำการตรวจสอบคุณลักษณะเฉพาะทางกลศาสตร์และทางกายภาพของอุปกรณ์รับ-ส่ง (mechanical and physical specifications) เพื่อให้สามารถทำงานร่วมกันได้

**6.3**

# ประเภทของเครือข่าย

## ประเภทของเครือข่าย





# ประเภทของเครือข่าย

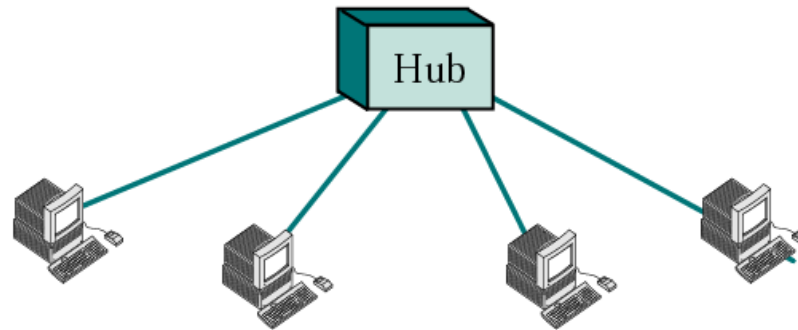
- เครือข่ายคอมพิวเตอร์แบ่งออกเป็น 3 ประเภทคือ
  - \* **LAN** : Local Area Network ครอบคลุมพื้นที่ขนาดเล็ก ใช้ภายในหน่วยงานที่อยู่ในอาคารเดียวกันหรือใกล้เคียงกัน
  - \* **MAN** : Metropolitan Area Network ครอบคลุมพื้นที่เมืองขนาดเล็ก ขนาดกลางหรือขนาดใหญ่ เช่น อำเภอ จังหวัด
  - \* **WAN** : Wide Area Network ครอบคลุมระดับประเทศ ทวีปหรือทั่วโลก

# Local Area Network

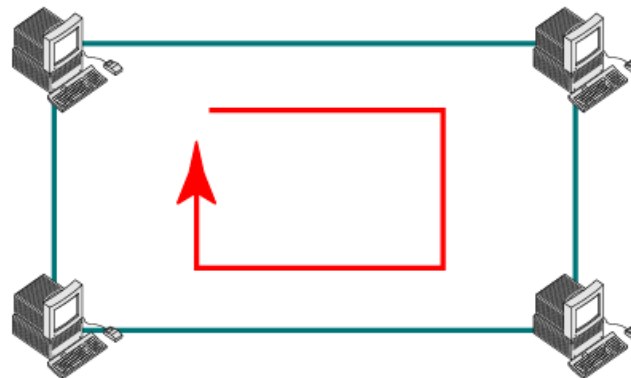
- LAN ออกแบบมาเพื่อให้องค์กรสามารถใช้ทรัพยากรร่วมกันได้ (ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูล)
- LAN ประกอบด้วยเครื่องคอมพิวเตอร์ และอุปกรณ์ต่อพ่วง ที่ต่อ เชื่อมกันโดยสายส่งสัญญาณ
- รูปแบบการเชื่อมต่ออุปกรณ์ต่างๆในระบบ LAN มี 3 แบบ แต่ละแบบมีจุดอ่อนและจุดแข็งที่แตกต่างกัน



a. Bus LAN



b. Star LAN



c. Ring LAN

# การเชื่อมต่อ LAN : Bus Topology

- **Bus Topology** เป็นการเชื่อมต่อคอมพิวเตอร์และอุปกรณ์ต่างๆ ผ่านสายส่งสัญญาณที่เรียกว่า bus ถ้าคอมพิวเตอร์เครื่องหนึ่งต้องการส่งเฟรมของข้อมูลไปยังคอมพิวเตอร์อีกเครื่องหนึ่ง เครื่องคอมพิวเตอร์ทั้งหมดที่เชื่อมต่อกับ bus จะรับเฟรมเหมือนกันหมด จากนั้นแต่ละเครื่องก็จะตรวจสอบว่า destination address ที่อยู่ใน header เป็นของตนเองหรือไม่? ถ้าใช่ก็รับไว้ แล้วนำข้อมูลที่ส่งมากับเฟรมไปประมวลผล แต่ถ้าไม่ใช่ก็ทิ้งเฟรมนั้นไป

# การเชื่อมต่อ LAN : Bus Topology (ต่อ)

- **ข้อสังเกต:** เราจะเห็นว่าในสถาปัตยกรรมการเชื่อมต่อ LAN แบบ bus topology นั้น ตรงหัวและท้ายของ bus จะมีอุปกรณ์ที่เรียกว่า bus terminator อยู่ด้านละ 1 ตัว อุปกรณ์นี้ทำหน้าที่กำจัดเฟรมข้อมูลเมื่อเฟรมมาถึง ถ้าไม่มีอุปกรณ์นี้หรืออุปกรณ์นี้ไม่ทำงาน สัญญาณที่ส่งผ่านในรูปแบบของเฟรมก็จะถูกส่งกลับไปที่ปลายทั้งสองจุดของสายส่งอย่างไม่มีที่สิ้นสุด

# การเชื่อมต่อ LAN : Star Topology

- **Star topology** : เครื่องคอมพิวเตอร์และอุปกรณ์ต่อพ่วงเชื่อมต่อกันผ่าน **Hub** หรือ **Switch** ที่ควบคุมการรับส่งเฟรม
- ถ้า LAN เชื่อมต่อผ่าน Hub ระบบ LAN ที่ได้จะมีลักษณะคล้าย **bus topology** โดยที่ hub จะส่งเฟรมที่ได้รับไปยังอุปกรณ์เชื่อมต่อทุกชนิดแล้วให้อุปกรณ์เหล่านั้นตรวจสอบ address เอง
- ถ้า LAN เชื่อมต่อด้วย switch ตัว switch จะทำการตรวจสอบ address เองแล้วจึงทำการส่งเฟรมไปยังอุปกรณ์ปลายทางที่เชื่อมต่อซึ่งมี physical address ตรงกัน

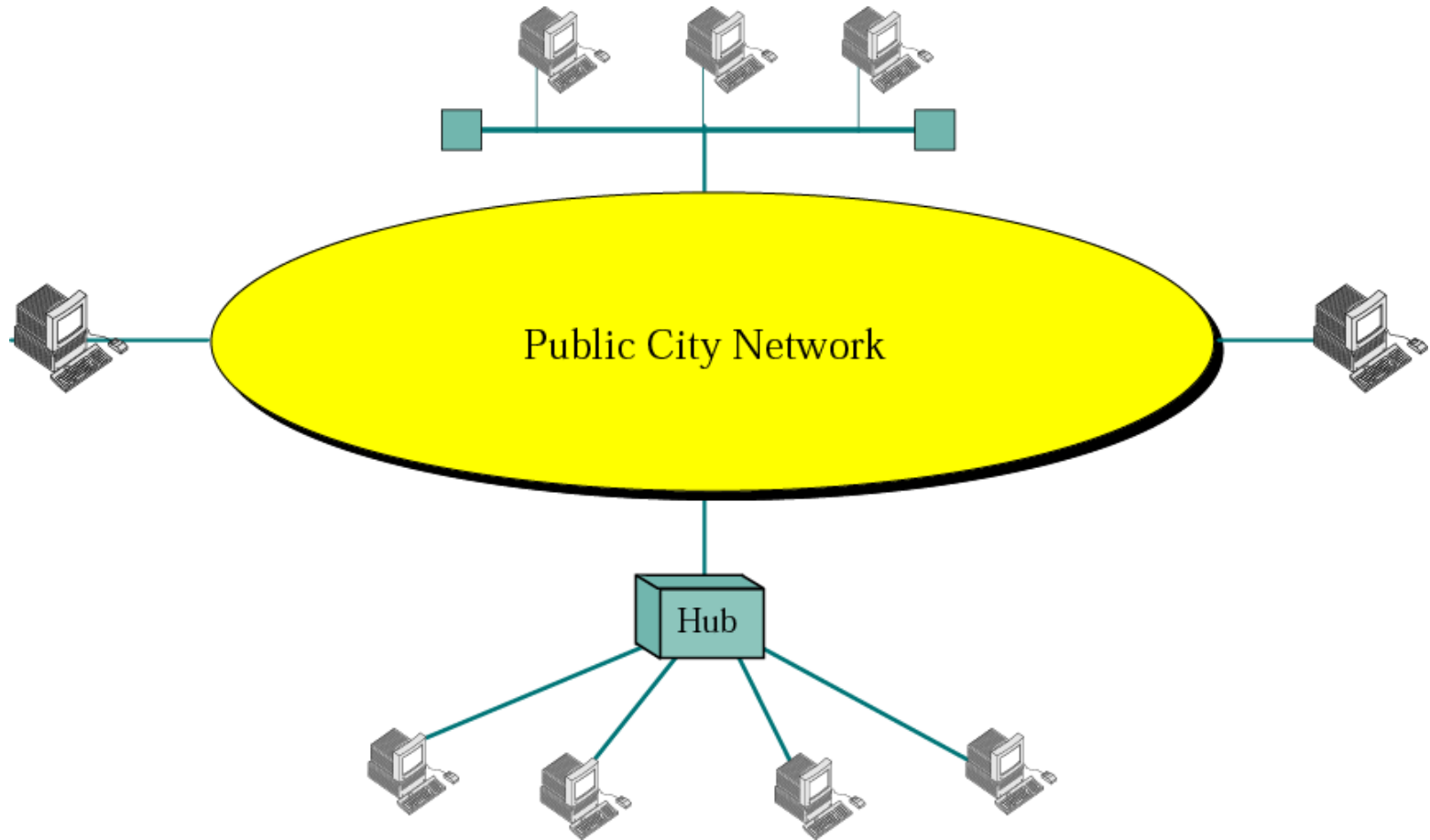
# การเชื่อมต่อ LAN : Ring Topology

- **Ring Topology** เป็นการเชื่อมต่อคอมพิวเตอร์และอุปกรณ์ต่อพ่วงที่มีลักษณะคล้ายวงแหวน ถ้าเครื่องคอมพิวเตอร์ต้นทางต้องการส่งเฟรมไปยังคอมพิวเตอร์ปลายทางที่อยู่ในวงแหวนเดียวกัน เครื่องต้นทางจะส่งเฟรมไปยังเครื่องที่**ใกล้เคียง (neighbor)** เครื่องที่ได้รับเฟรมก็จะแก้ไขเฟรมบางส่วนแล้วส่งต่อไปยังเครื่องใกล้เคียงต่อไปจนกว่าจะถึงเครื่องปลายทาง เมื่อเครื่องปลายทางได้รับ มันจะทำการเปิดเฟรม คัดลอกข้อมูลในเฟรม แล้วอาจจะทำลายเฟรมทิ้งหรือเพิ่มข้อความว่า**ได้รับข้อมูล**แล้ว จากนั้นจึงส่งเฟรมคืนไปยังผู้ส่งต้นทาง (**Acknowledgement**) ในกรณีนี้เครื่องต้นทางที่ส่งจะเป็นผู้ทำลายเฟรมเอง



## รูปที่ 6-5

# MAN

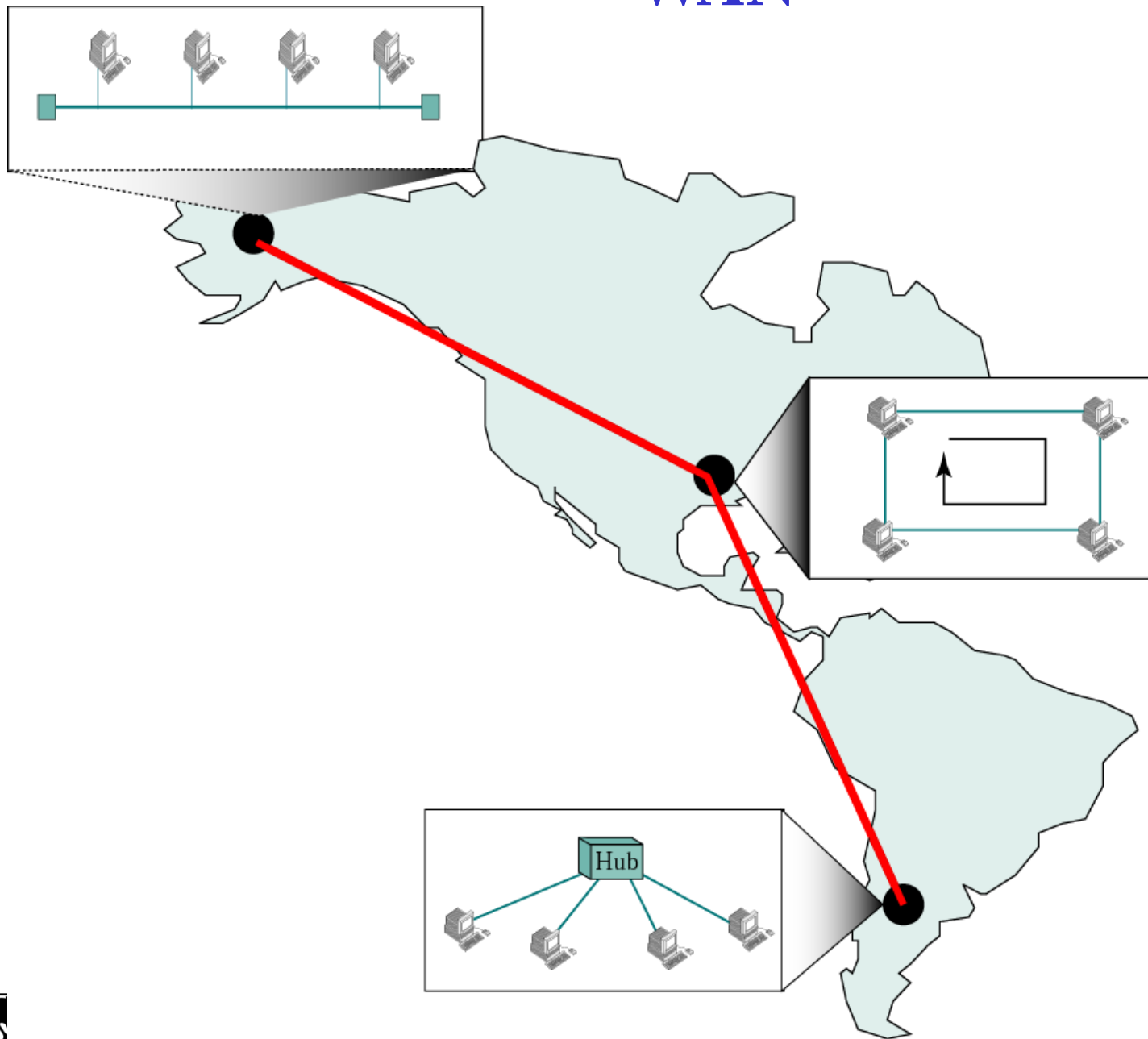




# Metropolitan Area Network : MAN

- **MAN** เป็นเครือข่ายที่ใช้บริการเชื่อมต่อจากสื่อสายส่งที่จัดเตรียมให้โดย **network service provider** เช่นหน่วยงานที่ให้บริการทางโทรศัพท์และการสื่อสาร การให้บริการจะครอบคลุมเขตบริเวณตัวเมืองโดยบริการมุ่งไปส่วนบุคคล ครัวเรือน หรือ องค์กรประเภทต่างๆ
- บุคคลหรือครัวเรือนสามารถเชื่อมต่อ PC ของตนเองเข้ากับเครือข่าย ส่วนองค์กรสามารถเชื่อมต่อเครือข่ายกับ LAN ของตนเอง

# WAN



# Wide Area Network : WAN

- **WAN** เป็นการเชื่อมต่อเครื่องคอมพิวเตอร์ส่วนบุคคล หรือเชื่อม LAN ขององค์กรเข้ากับเครือข่ายขนาดใหญ่ที่ครอบคลุมพื้นที่ ขนาดใหญ่เช่น จังหวัด ประเทศ หรือทั้งโลก
- **WAN** ต้องได้รับการติดตั้งและดำเนินการผ่านสื่อสายนำสัญญาณ ซึ่งมีลักษณะคล้ายกับ MAN
- ผู้ที่ใช้สายโทรศัพท์เชื่อมผ่าน ISP คือผู้ใช้ WAN ค่าใช้จ่ายในการเชื่อมต่อก็เป็นข้อตกลงระหว่าง ISP กับผู้ใช้

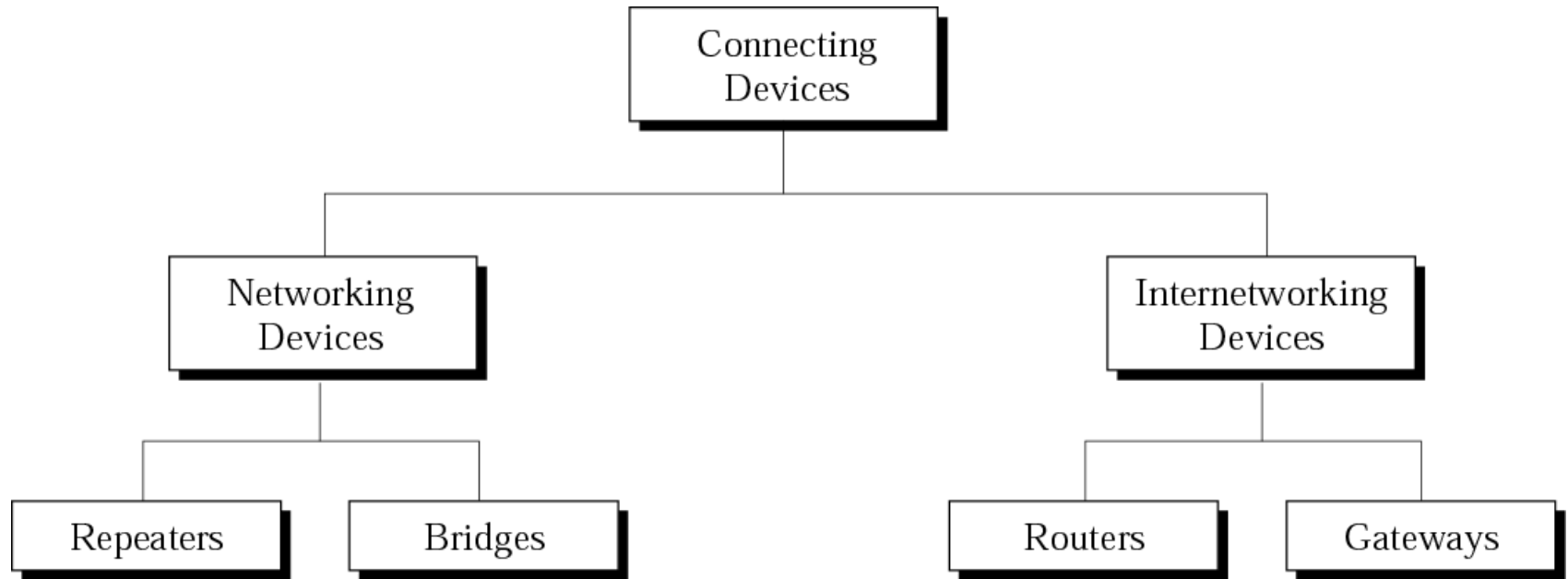
**6.4**

# อุปกรณ์เชื่อมต่อ

# อุปกรณ์เชื่อมต่อ

- เครือข่ายทั้งสามประเภทคือ LAN, MAN, และ WAN สามารถเชื่อมต่อกันด้วยอุปกรณ์เชื่อมต่อ การเชื่อมต่อระหว่างเครือข่ายทำให้เกิดการสื่อสารถึงกันทั่วโลก
- อุปกรณ์เชื่อมต่อแบ่งออกได้เป็น 4 ประเภทตามลักษณะการใช้งานที่สัมพันธ์กับเลเยอร์ของใน OSI Model ดังนี้: **Repeaters, Bridges, Routers, และ Gateways**
- Repeaters และ Bridges ปกติจะใช้เชื่อมต่ออุปกรณ์ภายในเครือข่าย ส่วน Routers และ Gateways ใช้เชื่อมต่อระหว่างเครือข่ายกับเครือข่าย

# อุปกรณ์เชื่อมต่อ



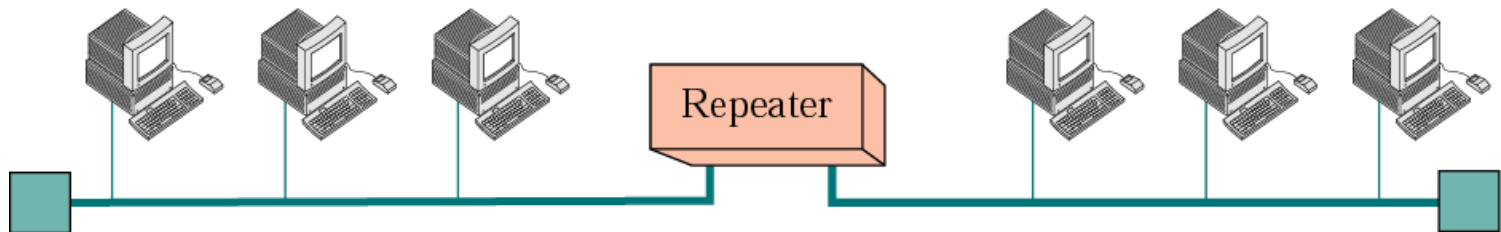
# Repeaters

- **Repeater** เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ใช้สำหรับการทำให้เกิดข้อมูลขึ้นมาใหม่ (**regenerate data**) เป็นอุปกรณ์ที่ทำให้การส่งสัญญาณสามารถไปได้ไกลขึ้นในเครือข่าย เมื่อสัญญาณส่งผ่านสื่อ ความแรงของสัญญาณจะอ่อนลง เมื่อถึงผู้รับปลายทางอาจทำให้ตีความผิดพลาดได้
- Repeaters จะใช้เฉพาะในระดับ physical layer ของโมเดล OSI เท่านั้น Repeaters ไม่รู้จัก physical address และไม่รู้จัก logical address มันจะ regenerate ทุกอย่างที่ได้รับ ใช้ได้ดีกับการเชื่อมต่อแบบ bus topology

# Repeater



a. Without Repeater



b. With Repeater





**Note:**

**Repeaters operate at the first layer (physical layer) of the OSI model.**



# Bridges

- เมื่อเครือข่ายเชื่อมกันด้วย bus technology สายส่งจะถูกใช้ร่วมกัน เมื่อผู้ใช้คนหนึ่งส่งข้อมูล คนอื่นจะส่งข้อมูลอีกไม่ได้ ต้องรอนกว่าคนแรกเสร็จ ทำให้ประสิทธิภาพเครือข่ายลดลง
- Bridges หน้าเหมือนผู้ควบคุมการจราจรข้อมูล มันจะทำการแบ่ง bus ที่ยาวออกเป็นส่วนย่อย แต่ละส่วนจะมีการจราจรที่เป็นอิสระต่อกัน
- Bridges ติดตั้งอยู่ระหว่าง bus สองส่วนที่ถูกแบ่ง มันสามารถส่ง หรือ สกัดกั้นเฟรมให้ไปสู่ที่อยู่ปลายทางที่กำหนดอยู่ในเฟรม

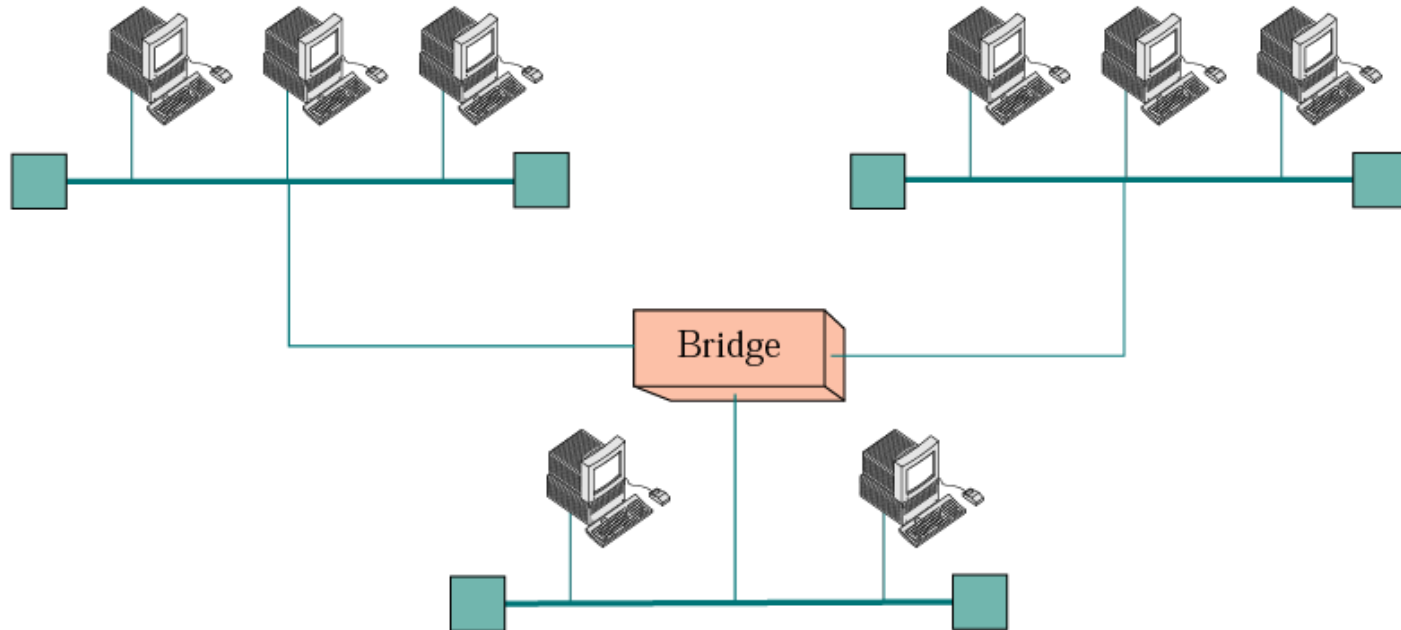
# Bridges (ต่อ)

- ถ้าเฟรมที่ต้องการส่งมีที่อยู่เริ่มต้นและที่อยู่ปลายทางอยู่ในส่วนเดียวกัน เฟรมนั้นก็ไม่ต้องส่งผ่าน bridges ด้วยเทคโนโลยี bridges ทำให้คู่ของสถานีตั้งแต่ 2 คู่ขึ้นไปสามารถส่งข้อมูลไปมาหากันภายในเวลาเดียวกัน
- นอกจากการควบคุมการส่งข้อมูลแล้ว bridges ยังอาจทำหน้าที่เป็น **repeater** ด้วยการ **regenerate** เฟรมได้อีกด้วย
- Bridges ติดตั้งและทำงานใน **physical layer** และเนื่องจาก bridges ต้องตีความหมายของที่อยู่ที่มากับเฟรมเพื่อการตัดสินใจส่งเฟรมไปในที่ที่ถูกต้อง Bridges จึงต้องทำงานใน **data-link layer** ด้วย

# Bridge



a. Without Bridge



b. With Bridge



Note:

**Bridges operate at the first two layers  
(physical layer and data-link layer) of the OSI model.**

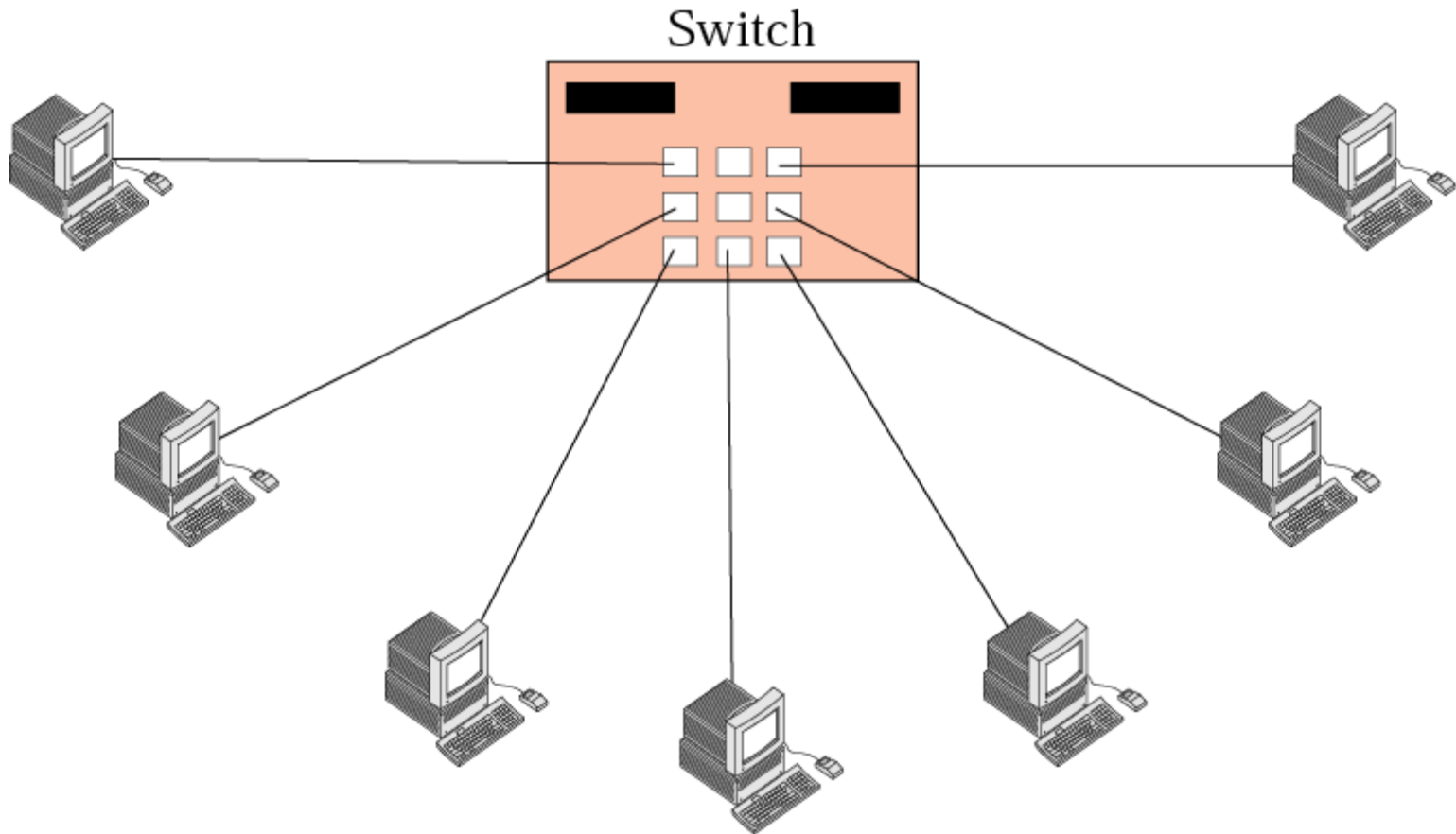


# Bridges (ต่อ)

- เพื่อให้เกิดประสิทธิภาพสูงสุด ได้มีการพัฒนาอุปกรณ์ชนิดใหม่ขึ้นมา เรียกตาม Data-Link Layer ว่า **สวิตช์ (switch)** ซึ่งเป็น bridge ที่ซับซ้อนและมีจุดเชื่อมต่อหลายจุด
- ตัวอย่างการทำงานเช่น เครือข่ายที่มี 20 สถานีอาจแบ่งออกเป็น 4 ส่วน เชื่อมต่อกับด้วย bridges 4 ตัวหรือเราอาจแบ่งออกเป็น 20 ส่วนๆละ 1 สถานี แล้วใช้ switch ที่มีจุดเชื่อมต่อ 20 จุดจำนวน 1 ชุด แต่ละสถานีจะต่อตรงกับ switch การส่งข้อมูลแต่ละสถานีจะส่งเฟรมตรงไปยัง switch ทำให้ประสิทธิภาพการส่งข้อมูลดีขึ้น

## รูปที่ 6-10

# Switch



# Routers

- Routers เป็นอุปกรณ์ที่เชื่อมต่อ LANs, MANs, และ WANs  
Router ทำงานที่ Network Layer
- ขณะที่ bridge ทำการกลั่นกรองเฟรมโดยดูจาก physical address ตัว router จะทำการจัดส่ง (route) packet ไปตาม logical address ของ packet สู่ปลายทาง
- เราอาจใช้ Bridge เชื่อมต่อขยาย LANs ให้ส่งข้อมูลได้ไกลขึ้น หรืออาจใช้เชื่อมต่อ LAN 2 วงที่อยู่ในองกรเดียวกัน แต่ router จะใช้เชื่อมต่อเครือข่าย 2 เครือข่ายที่เป็นอิสระต่อกันเช่น :



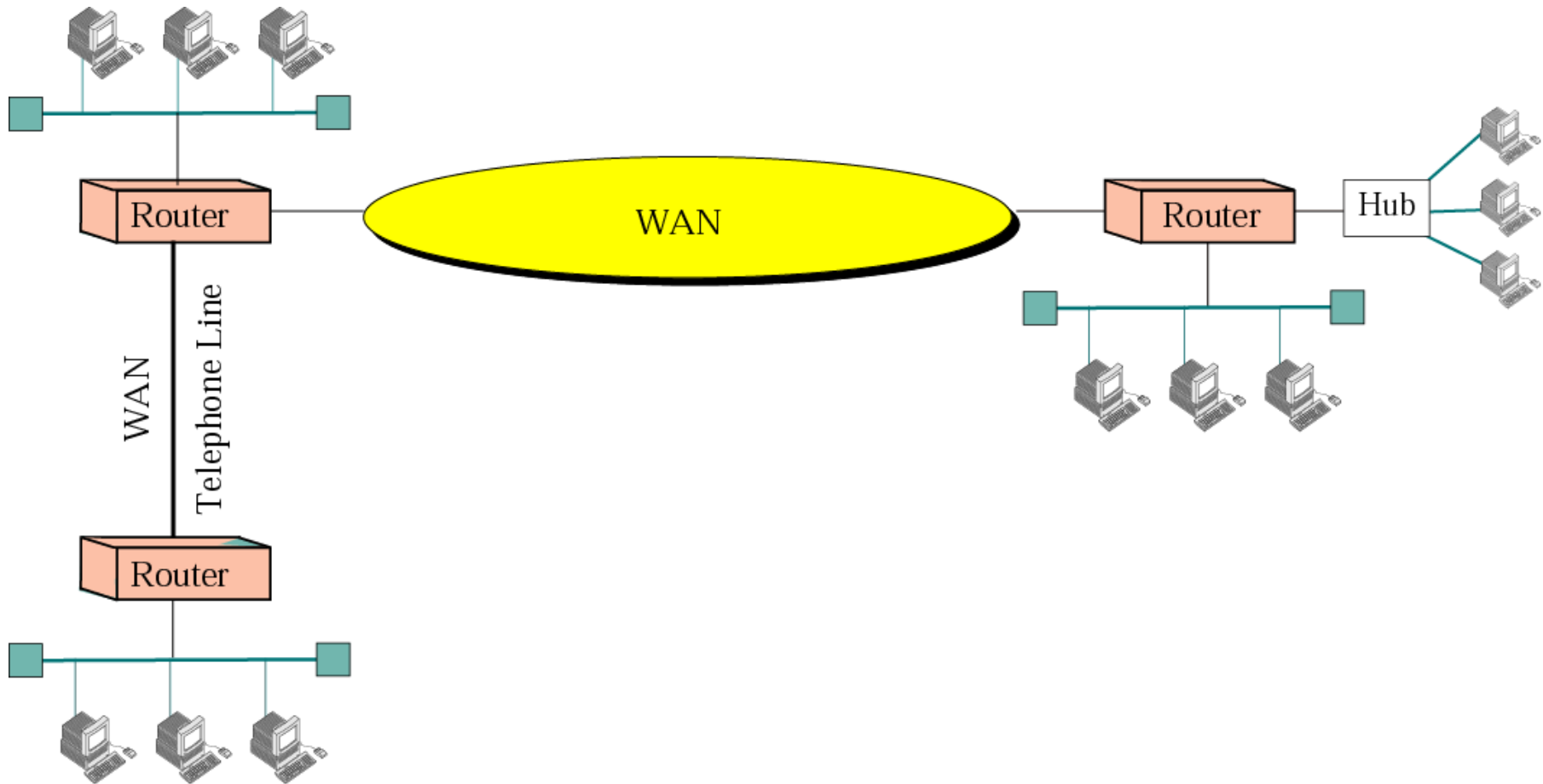
# Routers (ต่อ)

เชื่อมต่อ LAN กับ WAN, เชื่อมต่อ LAN กับ MAN, หรือ เชื่อม ต่อ WAN กับ WAN เป็นต้น ผลที่เกิดขึ้นจากการเชื่อมต่อนี้คือ อินเทอร์เน็ต (internet) นั่นเอง

- เครือข่ายอินเทอร์เน็ตที่เชื่อมต่อโลกใบนี้เข้าด้วยกันเป็นตัวอย่าง เครือข่ายที่เกิดขึ้นจากการเชื่อมต่อเครือข่ายย่อยๆเข้าด้วยกันโดยใช้ routers นั่นเอง ตัวอย่างดังรูปที่ 6.11

## รูปที่ 6-11

# Routers ใน internet



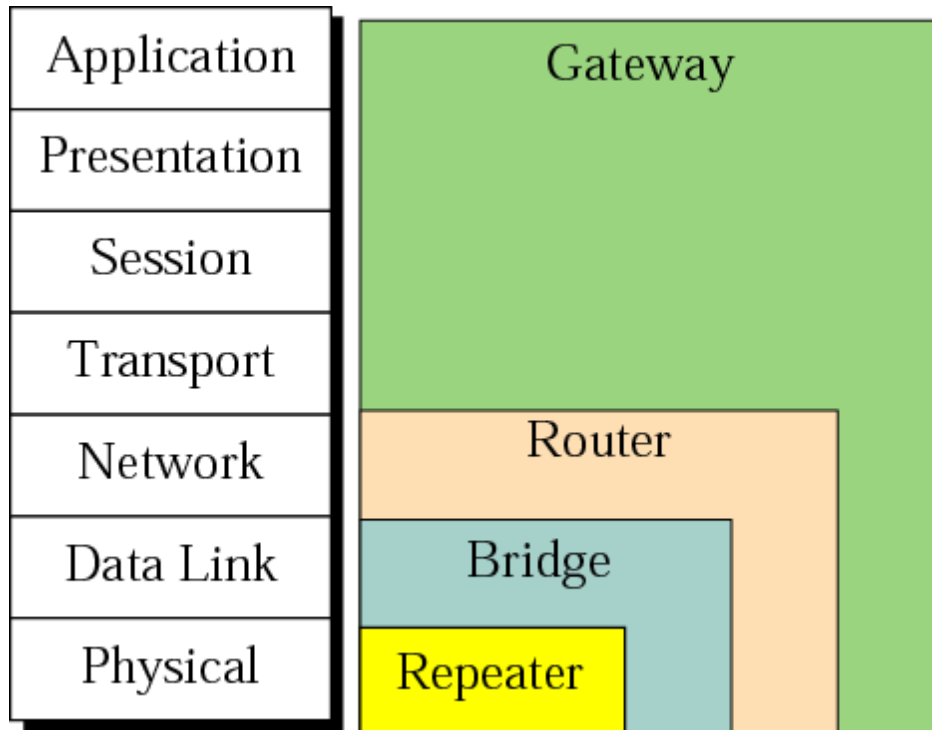
# Gateways

- **Gateway** เป็นอุปกรณ์เชื่อมต่อเครือข่ายที่ทำหน้าที่เป็น **protocol converter** โดยการทำให้เครือข่าย 2 เครือข่ายที่มี**ระเบียบวิธีในการดำเนินการ** (protocol) ในทั้ง 7 เลเยอร์ของโมเดล OSI ที่แตกต่างกันสามารถเชื่อมต่อและสื่อสารกันได้
- **Gateway** โดยปกติหมายถึงคอมพิวเตอร์ที่ติดตั้งด้วยซอฟต์แวร์ เฉพาะที่เข้าใจและสามารถแปลความหมายของ protocol ที่ใช้ในเครือข่ายหนึ่งไปเป็น protocol ที่ใช้ในอีกเครือข่ายหนึ่งที่เชื่อมต่อกันได้

# สรุป

- **ตัวอย่าง:** เราสามารถใช้ gateway เชื่อมต่อเครือข่ายที่ใช้โปรโตคอล AppleTalk กับเครือข่ายที่ใช้โปรโตคอล Novell Netware ได้
- ปัจจุบันมีผู้ใช้คำว่า **gateway** กับคำว่า **router** ในความหมายที่เหมือนกัน ความแตกต่างดูจะค่อยๆ เลือนหายไป
- OSI Model กับอุปกรณ์ต่อเชื่อมสามารถสรุปการเชื่อมต่อในเลเยอร์ต่างๆ ของ OSI Model ได้ดังรูป 6.12 ต่อไปนี้

## อุปกรณ์เชื่อมต่อกับแบบจำลอง OSI



**6.5**

# เครือข่ายอินเทอร์เน็ตและโปรโตคอล TCP/IP



# TCP/IP

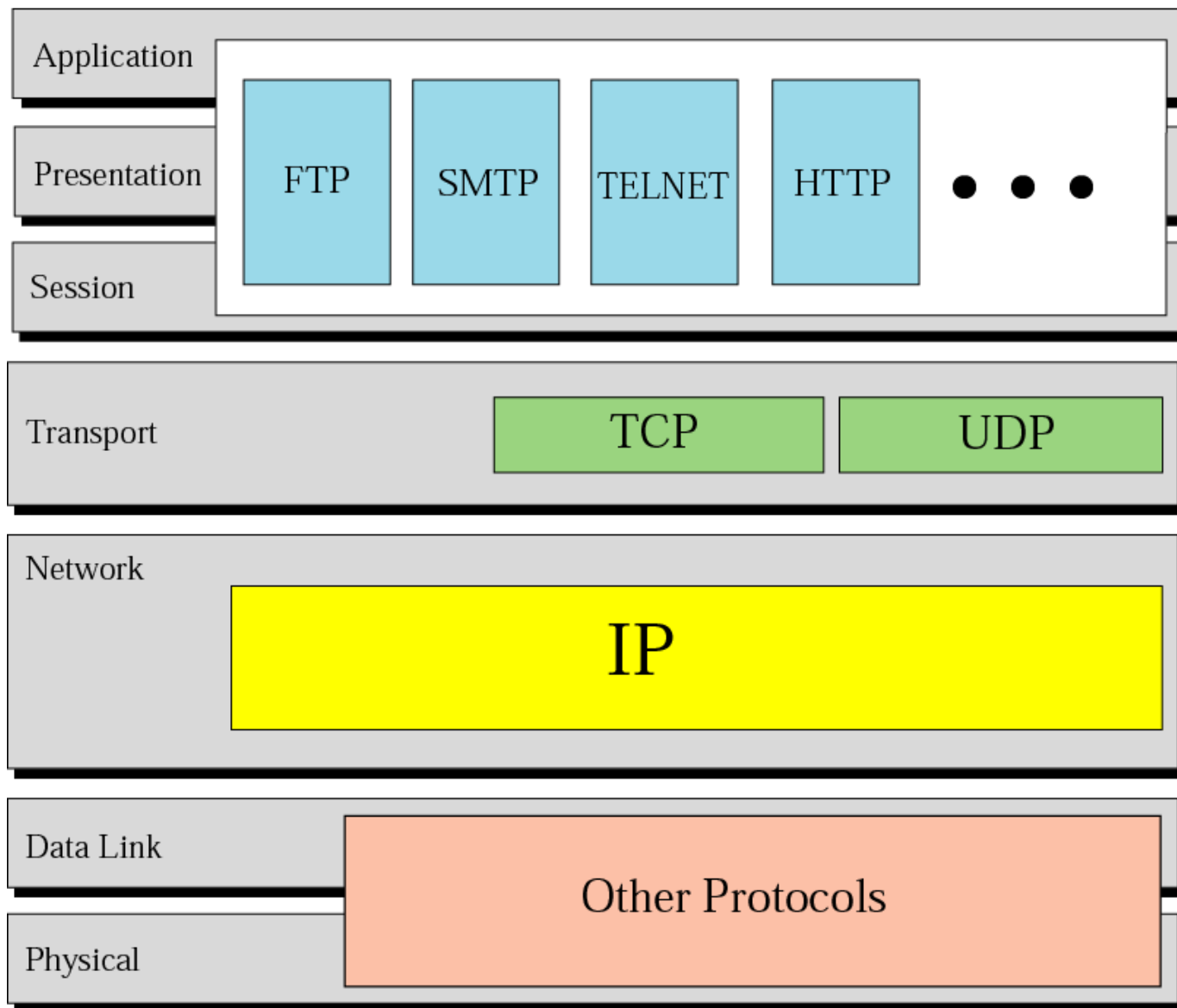
- ปัจจุบันมีเครือข่ายอินเทอร์เน็ตส่วนตัว (private internets) และเครือข่ายอินเทอร์เน็ตสาธารณะ (public internets) จำนวนมาก แต่อินเทอร์เน็ตที่เป็นที่นิยมกันมากคือ Internet
- Internet เป็นเครือข่ายที่เชื่อมต่อเครื่องคอมพิวเตอร์หลายพันล้านเครื่องทั่วโลกเข้าด้วยกัน เครื่องคอมพิวเตอร์แต่ละเครื่อง อาจมีคุณลักษณะที่แตกต่างกัน จำเป็นจะต้องมีข้อตกลงบางประการที่จะทำให้เครื่องคอมพิวเตอร์และอุปกรณ์ต่อพ่วงสามารถเชื่อมต่อและสื่อสารกันได้

# TCP/IP (ต่อ)

- **TCP/IP** ย่อมาจาก **T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol เป็นชุดของกฎเกณฑ์หรือกฎระเบียบที่เป็นทางการที่ใช้ควบคุมการสื่อสารทางอินเทอร์เน็ต
- TCP/IP พัฒนาขึ้นก่อน OSI Model ทำให้เลเยอร์ต่างๆที่อยู่ใน TCP/IP protocol จะไม่เหมือนกันเสียเลยที่เดียวกับเลเยอร์ใน OSI Model ดังรูปที่ 6.13



# โปรโตคอล TCP/IP กับแบบจำลอง OSI



# TCP/IP กับ OSI Model

- **Physical และ Data-Link Layer:** TCP/IP ไม่ได้กำหนด protocol ที่เฉพาะเจาะจงแต่อย่างใด TCP/IP สนับสนุน protocol ทุกชนิดทั้ง standard protocols และ proprietary protocols
- **Network Layer** (ที่ถูกต้องคือ **internetwork layer** หรือ **internet layer**) : TCP/IP สนับสนุน Internet Protocol (IP)
- **IP ไม่มีการตรวจสอบความผิดพลาดในการส่งข้อมูล และไม่ทำการตรวจสอบย้อนหลัง (tracking) ไม่รับประกันการสูญหาย**

# TCP/IP กับ OSI Model (ต่อ)

- หน่วยของข้อมูล (data unit) ใน IP layer เรียกว่า **IP datagram** ซึ่งเป็น packet อีกระยะที่ถูกส่งจากต้นทางสู่ปลายทาง
- **Datagram** ที่เป็นของ message เดียวกันหรือเป็นของ message ที่ต่างกัน อาจถูกส่งโดยใช้เส้นทางที่ต่างกัน และอาจถึงปลายทางด้วยลำดับที่ไม่เป็นไปตามลำดับหรืออาจมีการซ้ำกันได้ **IP ไม่ได้จำเส้นทางที่ส่งและไม่มีการจัดเรียง datagram** ให้อยู่ในลำดับที่ถูกต้องเมื่อ datagram ถึงปลายทาง

# TCP/IP กับ OSI Model : Addressing

- **Addressing:** TCP/IP ต้องการให้เครื่องคอมพิวเตอร์ทุกเครื่องที่เชื่อมต่อ กับ Internet จะต้องระบุ**ที่อยู่สากล** (international address) ที่ไม่ซ้ำกัน **ที่อยู่**ดังกล่าวบางครั้งเรียกว่า **Internet address** หรือ **IP address** นั้นเอง
- แต่ละ **IP address** ประกอบด้วย 4 ไบท์ (32 บิต) เพื่อให้อ่านและเข้าใจ ง่าย เลข IP address จะแทนด้วยเลขจำนวนเต็มฐาน 10 ที่มีจุดแบ่งแยก ไบท์ เรียกรูปแบบนี้ว่า **dotted-decimal notation** ดังแสดงในรูปที่ 6.14

## IP addresses in dotted-decimal notation

Bit Pattern

10000001 00001010 00000111 00011110

129.10.7.30

Dotted-Decimal Notation

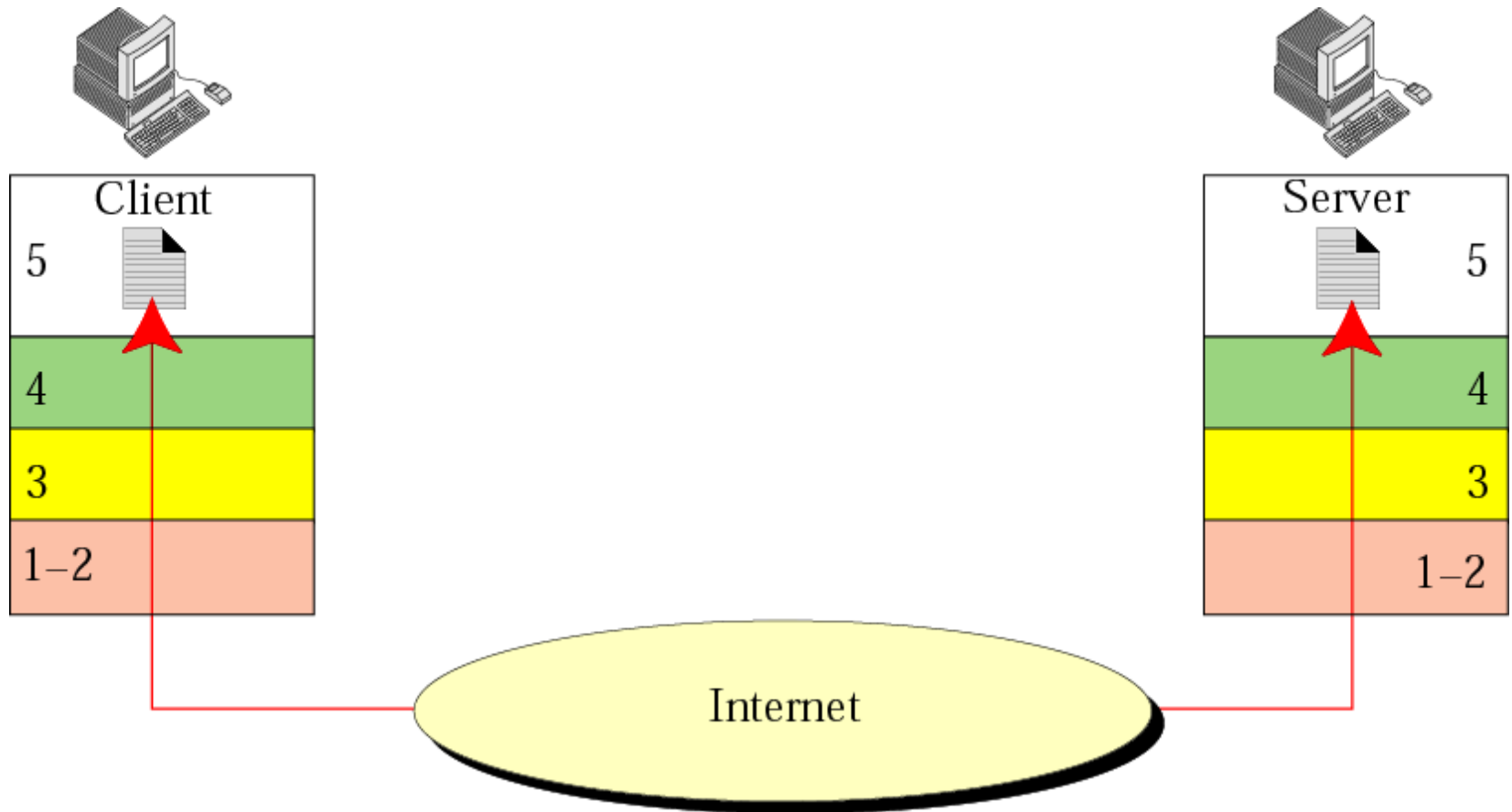
# TCP/IP กับ OSI Model : Transport Layer

- **Transport Layer:** ในเลเยอร์นี้ TCP/IP กำหนดให้มี 2 protocols คือ **Transmission Control Protocol (TCP)** และ **User Datagram Protocol (UDP)**
- UDP เป็น protocol ระดับการส่งแบบ end-to-end ซึ่งระบุเฉพาะกฎเกณฑ์เบื้องต้นที่จำเป็นเท่านั้น ส่วน TCP จะกำหนดกฎเกณฑ์อย่างละเอียดครอบคลุมการทำงานใน transport layer
- TCP จะแบ่งข้อมูลที่ต้องการส่งออกเป็นส่วนๆ แต่ละส่วนจะกำหนดหมายเลขกำกับ ถ้าส่วนใดส่วนหนึ่งเกิดการสูญหาย มันจะทำการส่งอีกถ้าส่วนที่ได้รับไม่เรียงลำดับ มันก็จะจัดการเรียงให้โดยใช้หมายเลข

# TCP/IP กับ OSI Model : Application Layer

- **TCP/IP Application Layer** จะทำงานได้เท่าเทียม (equivalent) กับการทำงานรวมกันของ 3 เลเยอร์ คือ Session, Presentation, และ Application Layers ใน OSI Model นั่นคืองานต่างๆที่ทำใน 3 เลเยอร์นั้นสามารถทำได้ในเลเยอร์เดียวใน TCP/IP คือ Application Layer
- การสื่อสารบน Internet ใช้รูปแบบ **Client-Server Model** โดยที่ **client** (โปรแกรมที่ใช้บนเครื่องท้องถิ่น) จะร้องขอใช้**บริการ**จาก **server** (โปรแกรมที่ใช้บนเครื่องที่อยู่ห่างไกล) ดังรูปที่ 6.15

# Client-server model





# File Transfer Protocol (FTP)

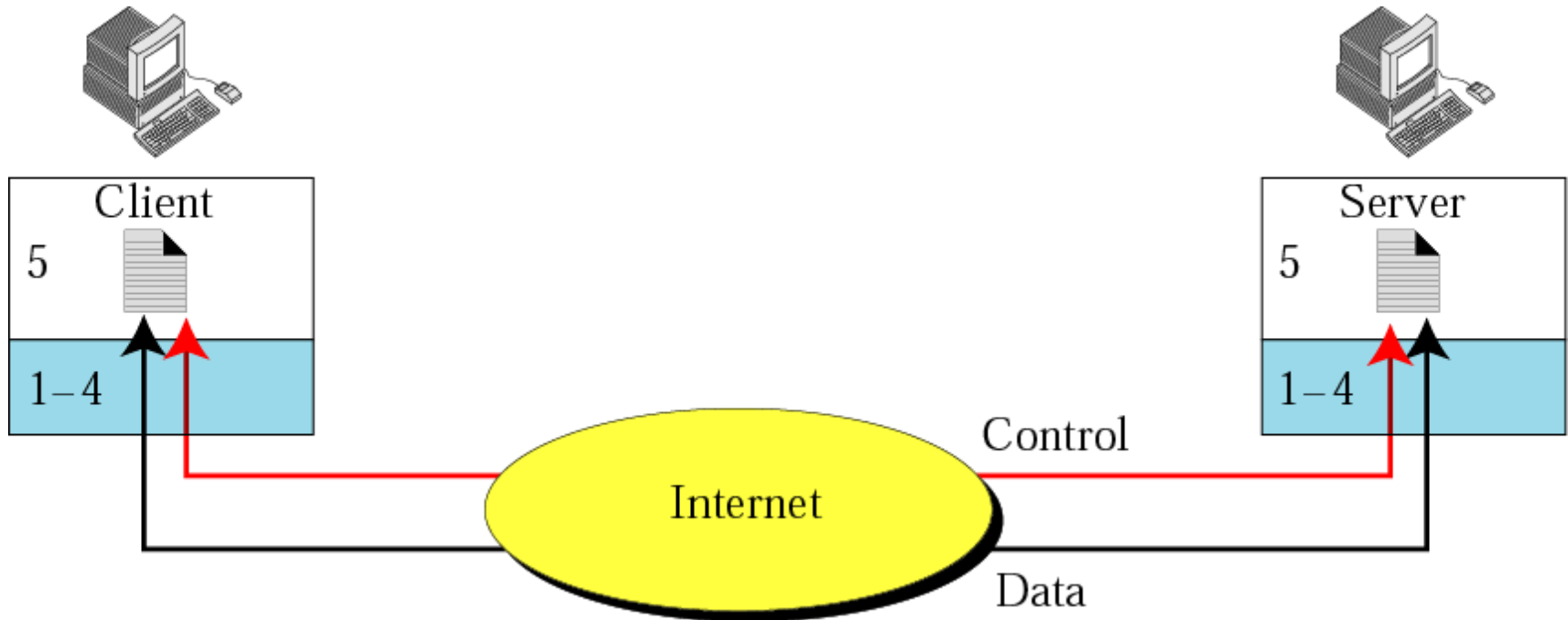
- **Protocol มาตรฐานบน Internet** ที่ใช้สำหรับการส่งแฟ้มข้อมูลจากคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่งมีชื่อว่า **File Transfer Protocol (FTP)**
- ก่อนที่จะมีการพัฒนา FTP การส่งแฟ้มข้อมูลระหว่างเครื่องคอมพิวเตอร์มีปัญหาหลัก 2 ประการคือ
  1. เครื่องคอมพิวเตอร์ 2 เครื่องใช้รหัสแทนข้อมูลต่างกัน
  2. แฟ้มข้อมูลบนเครื่องทั้งสองมีรูปแบบ(format)ต่างกัน

# File Transfer Protocol (FTP)

- FTP พัฒนาขึ้นมาเพื่อแก้ปัญหาทั้งสองประการดังกล่าว โดยกำหนดการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์เป็น 2 ส่วนคือ
  1. data transfer connection กับ
  2. control information connection
- ส่วน control connection จะทำงานตลอดช่วงเวลาของ FTP session ส่วน data connection จะทำงานเฉพาะเมื่อเวลาที่มีข้อมูลจะส่งเท่านั้น ดังแสดงในรูป 6.16

## รูปที่ 6-16

# FTP



# Simple Mail Transfer Protocol : SMTP

- งานหลักที่ใช้ Internet กันในทุกวันนี้คือการใช้ **electronic mail** โปรโตคอลที่สนับสนุนการใช้ email มีชื่อว่า **SMTP**
- การบริการ email โดยธรรมชาติก็จะแตกต่างจากการทำงานแบบอื่น การใช้บริการ email ผู้ใช้จะต้องติดตั้ง **SMTP software** ทั้งที่ **client และที่ server**
- SMTP จะต้องทำงานตลอดเวลาเพื่อพร้อมที่จะรับ mail เครื่องที่ปิดจะไม่สามารถรับ mail ได้ เพื่อแก้ปัญหานี้ SMTP จะใช้ร่วมกับโปรโตคอลอื่นเช่น **Post Office Protocol (POP)**

# Simple Mail Transfer Protocol : SMTP

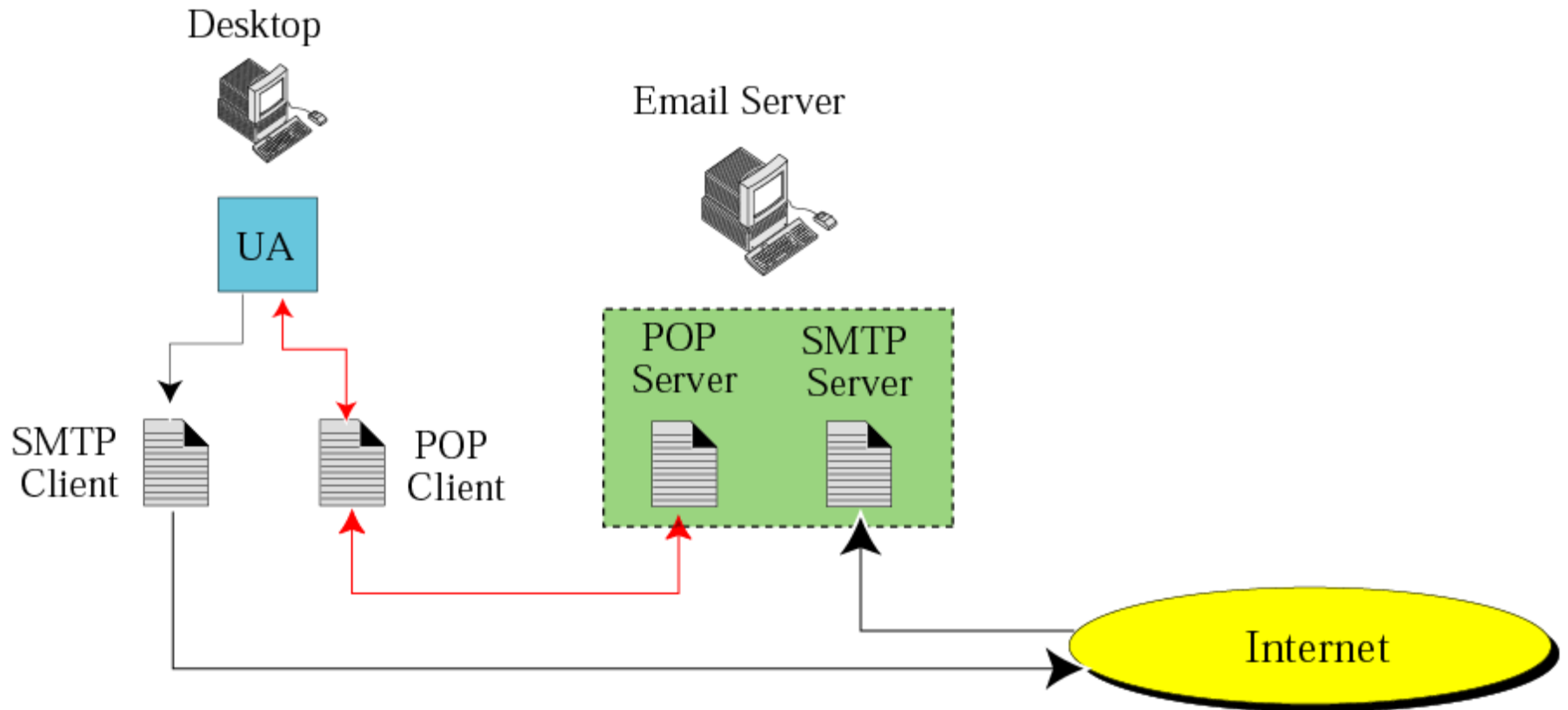
- ผู้ใช้ต้องใช้ SMTP จากฝั่ง client เพื่อการส่ง mail แต่ถ้าปิดเครื่อง ผู้ใช้ก็จะไม่ได้รับเมลที่มีผู้ส่งมาให้โดยตรง แต่เมลจะถูกเก็บไว้ที่ mail box ของผู้ใช้ซึ่งอยู่ใน server ที่ run โปรแกรม STMP ตลอดเวลา
- เมื่อผู้ใช้ต้องการอ่านเมล ก็จะต้องใช้ POP client เพื่อติดต่อกับ POP server แล้วทำการ download เมลที่ต้องการ
- โดยปกติก็จะมีส่วน user interface ที่เรียกว่า user agent ช่วยทำให้การรับและส่งเมลสะดวกขึ้น

# Simple Mail Transfer Protocol : SMTP

- **Addressing:** SMTP ใช้ ระบบการกำหนด address ที่ประกอบด้วย 2 ส่วนคือ **ส่วน logical part** กับ **ส่วน domain name** แยกกัน โดยเครื่องหมาย @ ดังรูปที่ 6.18
- ส่วน **local part** จะเป็นตัวกำหนดชื่อแฟ้มข้อมูลเฉพาะเรียกว่า **user mailbox** ซึ่งจะใช้เป็นที่ยกเก็บเมลทั้งหมดที่ส่งเข้ามา ผู้ใช้ก็จะเข้า ถึงเมลเหล่านี้โดยใช้โปรแกรม **user agent**
- ส่วน **domain name** จะกำหนดชื่อเครื่องคอมพิวเตอร์ที่ run SMTP เช่น **ouen@ru.ac.th**

## รูปที่ 6-17

# SMTP



รูปที่ 6-18

## Email address



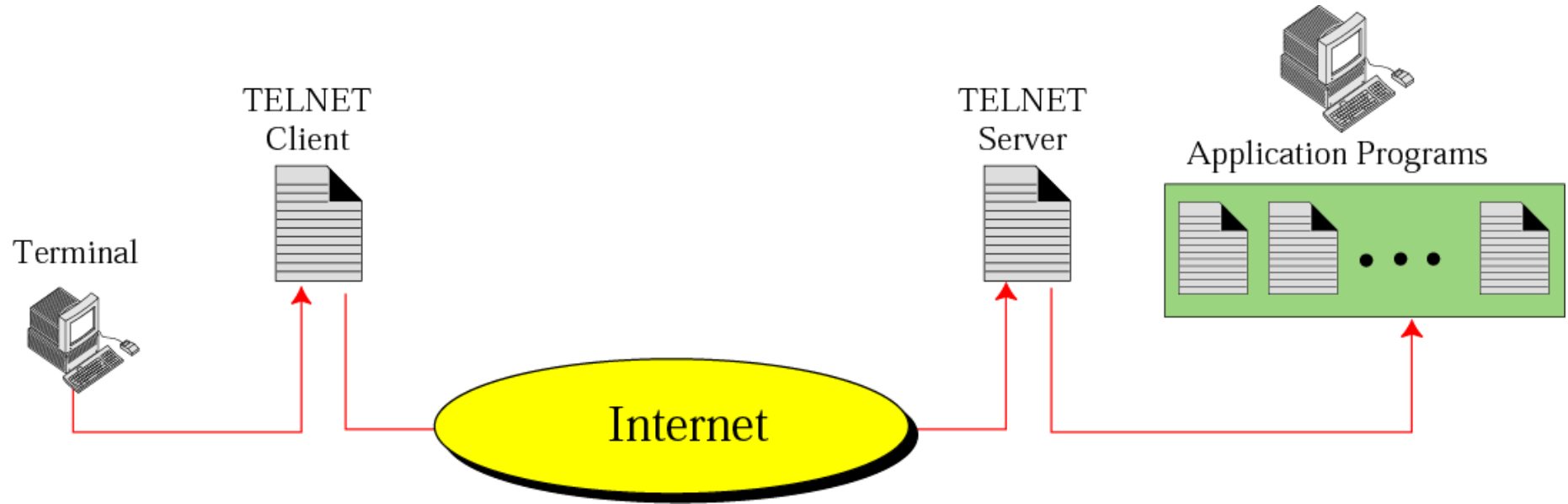


# TELNET

- **TELNET = Terminal NETwork** เป็น client-server program  
อเนกประสงค์บนอินเทอร์เน็ตที่ทำให้ผู้ใช้สามารถใช้บริการ  
**remote login**
- **TELNET** เป็นโปรแกรมที่สร้างการเชื่อมต่อจาก **local system**  
ไปยัง **remote system** โดยทำให้ **local terminal** สามารถทำงาน  
ในลักษณะเป็น **terminal** ของ **remote system** ดังรูปที่ 6.19

## รูปที่ 6-19

# TELNET



# Hypertext Transfer Protocol : HTTP

- **HTTP** เป็นโปรแกรม **client-server** ที่ใช้เข้าถึงและถ่ายโอนเอกสารบน **WWW (World Wide Web)** แม้ว่า HTTP จะถ่ายโอนข้อมูลในรูปแบบที่เป็น **ตัวอักษร (plain text)** เป็น **ลิงค์เชื่อมโยง (hypertext)** เป็น **เสียง (audio)** เป็น **ภาพเคลื่อนไหวและเสียง (video)** และอื่นๆ แต่วัตถุประสงค์ในการออกแบบ HTTP ก็เพื่อการถ่ายโอนเอกสาร **hypertext** โดยเฉพาะ
- ในการทำงาน **client HTTP** จะส่ง **คำขอ (request)** ไปยัง **server** ส่วน **server** ก็จะส่ง **คำตอบ (response)** มายัง **client** ทั้งคำขอและคำตอบ จะมีส่วนของ **แฟ้มเอกสารที่ฝังตัว (embedded)** เข้าไปด้วย

# Uniform Resource Locator : URL

- HTTP ใช้การกำหนดที่อยู่แบบพิเศษที่เรียกว่า **URL** ซึ่งเป็นรูปแบบมาตรฐานในการระบุประเภทของข้อมูลข่าวสารบนอินเทอร์เน็ต
- URL กำหนดรายละเอียดไว้ 4 ประเภทคือ **method, host computer, port, และ path** ซึ่งมีรูปแบบดังรูปที่ 6.20



## รูปที่ 6-20

# URL



# URL

- **Method:** เป็นโปรแกรม client-server ที่ใช้สำหรับถ่ายโอนข้อมูลในที่นี้คือเอกสาร HTTP
- **Host:** เป็นชื่อเครื่องคอมพิวเตอร์ที่เก็บข้อมูลที่ต้องการ โดยปกติชื่อจะขึ้นต้นด้วย www.
- **Port:** เป็นตัวกำหนดเลขที่ port ของ server (มีหรือไม่ก็ได้)
- **Path:** เป็นชื่อที่บอกเส้นทาง (pathname) ที่ข้อมูลเอกสารจัดเก็บใน pathname อาจมีเครื่องหมาย / เพื่อระบุแยก directory ได้



# World Wide Web : WWW

- **WWW หรือ Web** เป็นแหล่งที่เก็บของเอกสารประเภทสื่อผสม (multimedia documents) โดยมีวัตถุประสงค์เพื่อที่จะให้แต่ละองค์กรสามารถเก็บเอกสารไว้ที่เครื่องคอมพิวเตอร์ของตนเองที่เชื่อมโยงเข้าด้วยกัน ผู้ใช้คนอื่นๆสามารถใช้เอกสารเหล่านั้นร่วมกันได้
- **Hypertext:** WWW ใช้แนวคิดของเอกสาร hypertext ซึ่งเป็นเอกสารที่มี **ตัวอักษร คำ หรือวลีพิเศษ**ที่สามารถสร้างลิงค์ (link) ไปยังเอกสารอื่นที่มี text, images, audio, หรือ video

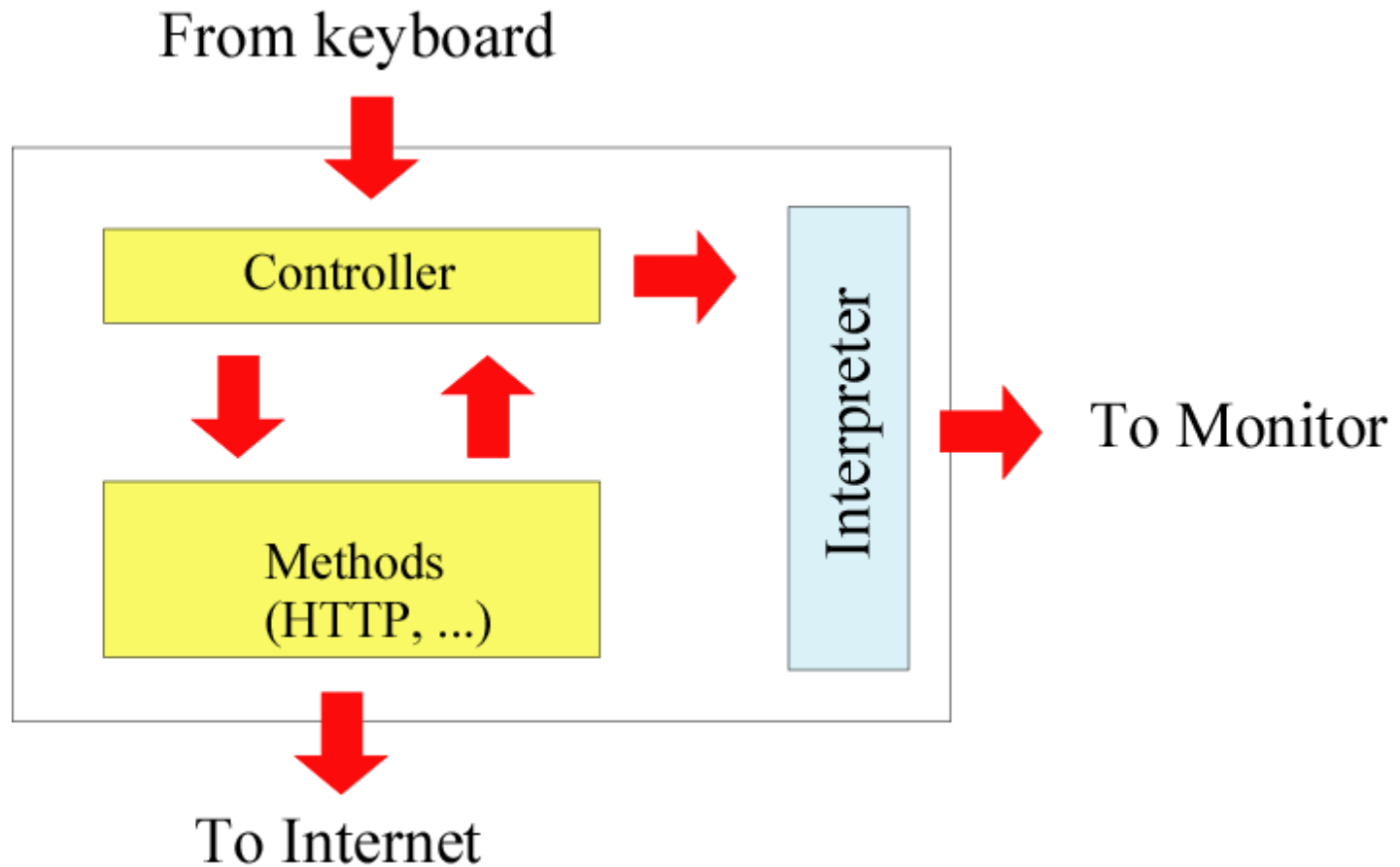
# World Wide Web : WWW

- เอกสาร **hypertext** ที่มีอยู่บน Web เรียกว่า **page** ส่วน page หลักขององค์กรหรือของส่วนบุคคลเราก็จะเรียกว่า **home page**
- **Browser:** การเข้าถึง page ใน WWW เราจะต้องมี **browser** ซึ่งประกอบด้วยองค์ประกอบ 3 ส่วนคือ **controller, method,** และ **interpreter** ดังรูปที่ 6.21
  - \* **controller** เป็นส่วนที่สำคัญที่สุด ประสานงานและควบคุมการทำงานของส่วนอื่นทั้งหมด



# World Wide Web : WWW

- \* **Method** เป็นโปรแกรม HTTP ประยุกต์ของ client ที่ใช้สำหรับ**รับ** (receive) เอกสารจาก WWW
- \* **Interpreter** เป็นส่วนที่ใช้สำหรับแสดงผลลัพธ์สู่หน้าจอ



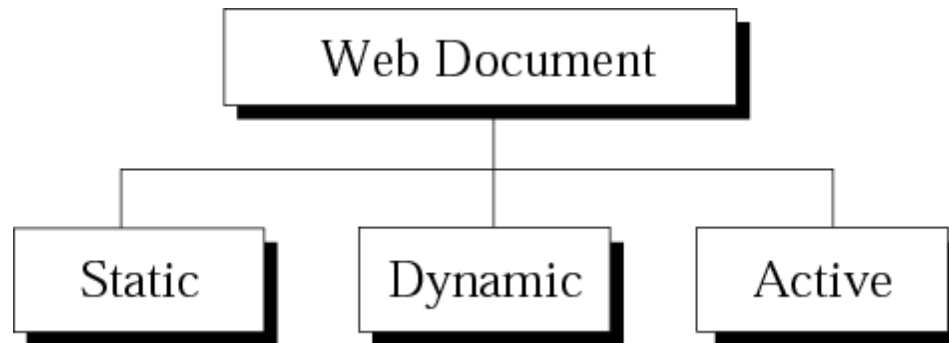
# ประเภทของเอกสารบน WWW

- เอกสารบน Web มี 3 ประเภทคือ **static, dynamic, และ active**
  - \* **Static documents:** เป็นเอกสารที่มีเนื้อหา (content) คงที่สร้างขึ้นที่ฝั่ง server มีไว้ให้คัดลอกไปเท่านั้น เอกสารประเภทนี้โดยทั่วไปจะภาษา HTML (Hypertext Markup Language) สำหรับจัดรูปแบบและลิงค์ภายในเอกสาร
  - \* **Dynamic documents:** เป็นโปรแกรมที่อยู่บนฝั่ง server เมื่อ browser ส่ง request เข้ามา server จะทำการ run โปรแกรมและส่งผลลัพธ์กลับไปยัง browser

# ประเภทของเอกสารบน WWW (ต่อ)

- \* Dynamic documents ใช้เทคโนโลยีที่มีชื่อว่า **Common Gateway Interface (CGI)** ที่รวมเอาโปรแกรมอย่าง Perl และ HTML เข้าไว้สำหรับการสร้างและแสดงผลเอกสาร
- **Active documents:** เป็นโปรแกรมที่ไม่สามารถ run บน server ได้ แต่จะเอาไว้ให้ browser ร้องขอให้ส่งไป run ที่ browser เช่น โปรแกรม animation จำเป็นต้อง run ที่ browser
- \* ปกติ active documents จะเขียนด้วยภาษา Java ดังนั้นที่ browser ต้องมี Java interpreter เพื่อ run active documents

## ประเภทของเอกสารบน Web



# คำที่สำคัญ

**LAN**

**MAN**

**WAN**

**FTP**

**HTML**

**HTTP**

**IP**

**OSI**

**STMP**

**TELNET**

**TCP**

**TCP/IP**

**URL**

**UDP**

**WWW**