

## Protocol TCP (transmission control protocol) และ UDP (user datagram protocol)

### TCP

- Reliable

ไว้วางใจได้ว่าข้อมูลที่ส่งไปจะถึงผู้รับอย่างแน่นอน ซึ่ง TCP จะมีการตรวจสอบว่าข้อมูลที่ส่งไปนั้นถึงผู้รับจริง ๆ หรือไม่ ถ้าไม่ถึง TCP ก็ทำการส่งข้อมูลนั้นไปให้ใหม่อีกครั้ง

- Connection-oriented

มีการเชื่อมต่อช่องทางการรับส่งข้อมูลก่อนที่จะเริ่มส่ง เป็นการเตรียมทรัพยากรต่าง ๆ ระหว่างเครื่องผู้รับและผู้ส่ง เช่น socket (ดูเพิ่มเติมเรื่อง socket) หน่วยความจำ และตัวแปรที่ใช้เก็บข้อมูลต่าง ๆ เพื่อให้การรับส่งข้อมูลระหว่างต้นทางและปลายทางเป็นไปอย่างราบรื่นและมีประสิทธิภาพ

- Flow control

มีการควบคุมปริมาณข้อมูลที่รับส่งระหว่างต้นทางและปลายทาง เพื่อป้องกันไม่ให้ฝั่งผู้ส่ง ส่งข้อมูลมากจนเกินกว่าที่ buffer ของฝั่งผู้รับจะรับได้

- Congestion control

เป็นการควบคุมปริมาณการส่งข้อมูลเช่นกัน แต่เพื่อป้องกันไม่ให้ส่งข้อมูลเข้าไปในเครือข่าย ที่ ณ ขณะนั้นมีความหนาแน่นของข้อมูลสูงมาก ซึ่งมีความเสี่ยงที่ข้อมูลที่ส่งเข้าไปจะไปไม่ถึงผู้รับ

### UDP

- Unreliable

ไม่รับประกันว่าข้อมูลจะถึงผู้รับหรือไม่

- Connectionless

ไม่มีการสร้างช่องทางการรับส่งข้อมูลก่อนเริ่มส่ง

- No flow control

ไม่มีการควบคุมปริมาณการรับส่งข้อมูลระหว่างต้นทางและปลายทาง

- No congestion control

ไม่มีการควบคุมปริมาณการส่งข้อมูลระหว่างที่เครือข่ายมีความหนาแน่นสูง

## ข้อดีของ UDP

- เริ่มต้นส่งข้อมูลได้เร็วกว่าเพราะไม่ต้องรอการสร้าง connection
- ส่งข้อมูลได้เร็วกว่าเพราะไม่ต้องรอการตรวจสอบ
- ส่งข้อมูลได้ปริมาณมากกว่าเพราะไม่มี flow control และ congestion control

TCP จะใช้ในการรับส่งข้อมูลของ protocol บน application layer ที่ต้องการความน่าเชื่อถือสูง ดังต่อไปนี้

- SMTP ( Simple Mail Transfer Protocol)
- Telnet
- HTTP (Hypertext Transfer Protocol)
- FTP (File Transfer Protocol)

UDP จะเหมาะกับการรับส่งข้อมูลของ protocol บน application layer ยินยอมให้ข้อมูลบางส่วนสูญหายได้โดยไม่กระทบต่อคุณภาพของข้อมูลในภาพรวม ดังต่อไปนี้

- NFS (Network File System)
- Streaming multimedia
- Internet telephony
- SNMP (Simple Network Management Protocol)
- RIP (Routing Information Protocol)
- DNS (Domain Name System)

## 2 Tier

คือ ยุค 1980 คอมพิวเตอร์ได้รับความนิยม ระบบ EIS มักเป็นแบบ 2-tier systems จะประกอบด้วย 2 ส่วนคือ Presentation logic ส่วนกำหนดรูปแบบการติดต่อระหว่างผู้กับแอปพลิเคชัน และ Business logic จะกำหนดว่าข้อมูลจะถูกจัดการอย่างไรในธุรกิจนั้น ๆ โปรแกรม 2 ส่วนนี้จะถูกติดตั้งและทำงานใน Client ที่เป็น PC ต่อกับ DB ทำหน้าที่เก็บและควบคุมข้อมูลขององค์กร

### ข้อดี คือ

1. ไม่ต้องยุ่งเกี่ยวกับการจัดเก็บข้อมูลโดยตรง
2. ประหยัดหน่วยความจำ เพราะข้อมูลไม่ได้เก็บไว้ในเครื่อง Client
3. เครื่อง Client สามารถใช้ข้อมูลร่วมกันได้ ทำให้ไม่มีปัญหาข้อมูลไม่
4. เหมือนกัน (ทั้ง 3 ข้อที่กล่าวมาคือการลดภาระการทำงานของคอมพิวเตอร์)
5. เหมาะกับระบบงานขนาดกลางและไม่ซับซ้อน

### ข้อเสีย คือ

1. ไม่ปลอดภัย เพราะ Client สามารถเข้าถึงฐานข้อมูลได้โดยตรง
2. กรณีเป็นระบบใหญ่ๆจะทำให้ Client ทำงานหนัก
3. หาก app. มีการเปลี่ยนแปลงจะต้องเสียเวลาในการติดตั้ง app. เพิ่มเติมเนื่องจากต้องติดตั้งให้กับ Client ทุกเครื่อง

### 3 Tier

Concept พื้นฐานคือการแบ่งแยกหน้าที่ความรับผิดชอบของแต่ละ Tier ให้เด็ดขาดจากกัน ไม่ว่าจะเป็น Presentation logic, Business logic, Database

#### ข้อดี คือ

1. หากมีการเปลี่ยนแปลงในบาง layer จะส่งผลกระทบต่อ layer อื่นน้อยมาก
2. สามารถนำกลับมาใช้ใหม่ได้เช่น Business logic
3. คอมพิวเตอร์แต่ละตัวรับภาระน้อยลง รองรับการทำงานปริมาณมากๆ

#### ข้อเสีย คือ

1. ออกแบบและพัฒนาระบบค่อนข้างยาก

### สถาปัตยกรรมแบบ Thin และ Fat Clients

Thin-client model: เป็นระบบที่ให้ client มีการประมวลผลน้อยที่สุด โดยการใช้เพียงแค่ Browser ซึ่งทำหน้าที่แสดงผล (Presentation) เท่านั้น การประมวลผลส่วนใหญ่จะอยู่ที่เครื่องแม่ข่าย (Server) รวมทั้ง Code ของซอฟต์แวร์ด้วย ทั้ง Application processing และ การจัดการข้อมูล (Data management)

#### ข้อดี คือ

1. ระบบนี้จะมีความสะดวกในการบริหารจัดการ เพราะสามารถทำได้จากส่วนกลาง ตัวอย่างของระบบนี้เช่น การใช้ <http://www> ในระบบอินเทอร์เน็ต เป็นต้น

#### ข้อเสีย คือ

1. ระบบนี้คือเครื่องแม่ข่ายจะทำงานหนักมาก

Fat-client model: เป็นระบบที่ให้เครื่อง Client ทำงานเป็นส่วนใหญ่ ทั้งการแสดงผล (Presentation) และการประมวลผลการทำงาน (Application processing) Fat-client: จะทำหน้าที่ติดตั้งซอฟต์แวร์หรือ Code ซึ่งจะเป็นการกระจายการประมวลผล ไปทั้งที่เครื่อง Client และเครื่อง Server แต่จะหนักไปที่เครื่อง Client

#### ข้อดี คือ

1. เครื่องแม่ข่ายไม่จำเป็นต้องมีขนาดใหญ่มาก แต่เครื่อง client จะต้องมีประสิทธิภาพที่ดี เพราะจะต้องทำการประมวลผลที่เครื่อง client

#### ข้อเสีย คือ

1. การบริหารจัดการค่อนข้างยาก เพราะจะต้องติดตั้งซอฟต์แวร์ทั้งที่เครื่องแม่ข่าย และเครื่องลูกข่ายทั้งหมด

## Distributed programming

เป็นการเขียนโปรแกรมเพื่อรองรับการคำนวณแบบกระจายตัวเช่น บิลเทอเรล ระบบการซื้อขายผ่าน  
ไปรษณีย์ ระบบธนาคารแบบกระจายตัว หรือการบริการขนส่งข้ามคืน

ข้อดีคือใช้เวลาตอบสนองได้เร็วขึ้น, ใช้ต้นทุนน้อยกว่า, ปรับปรุงความถูกต้องของข้อมูล, การใช้ทรัพยากร  
ร่วมกัน, ลดต้นทุนตัวประมวลผลหลัก, เพิ่มความน่าเชื่อถือ

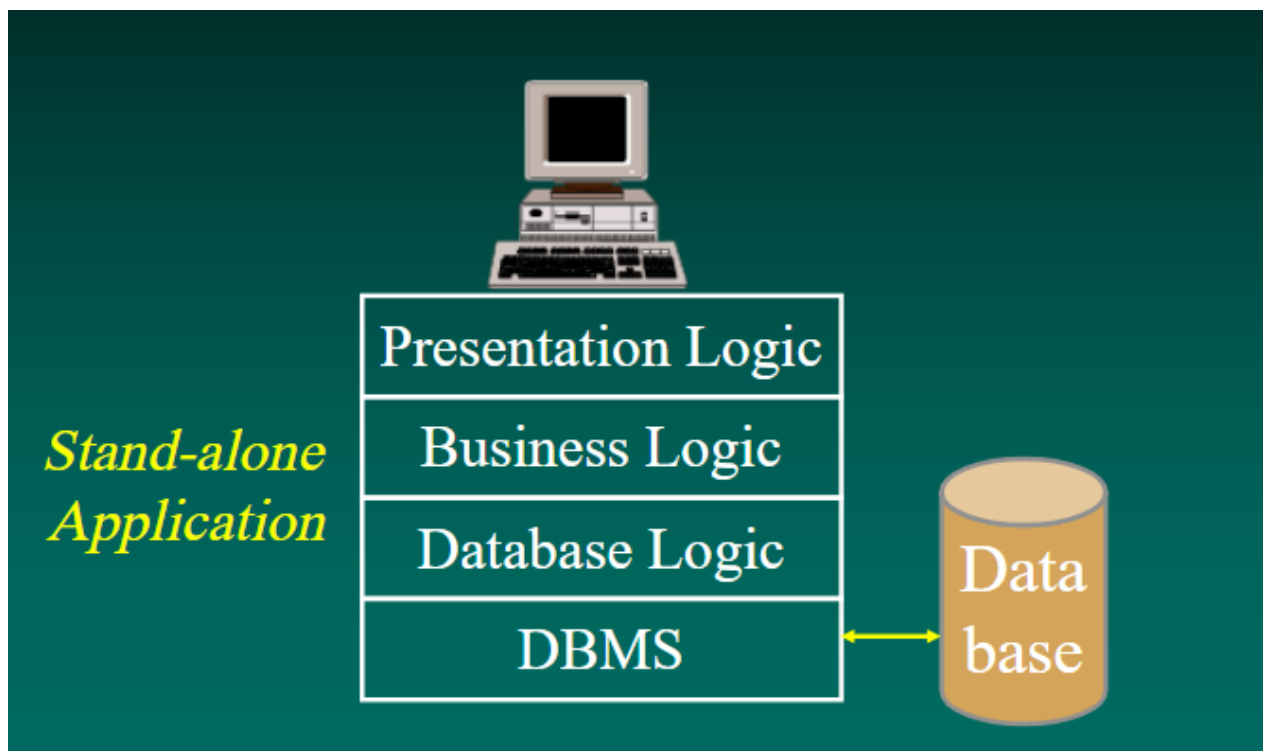
ข้อเสียคือการขาดแคลนผู้เชี่ยวชาญด้าน MIS, มาตรฐานของระบบ, ความถูกต้องของข้อมูล

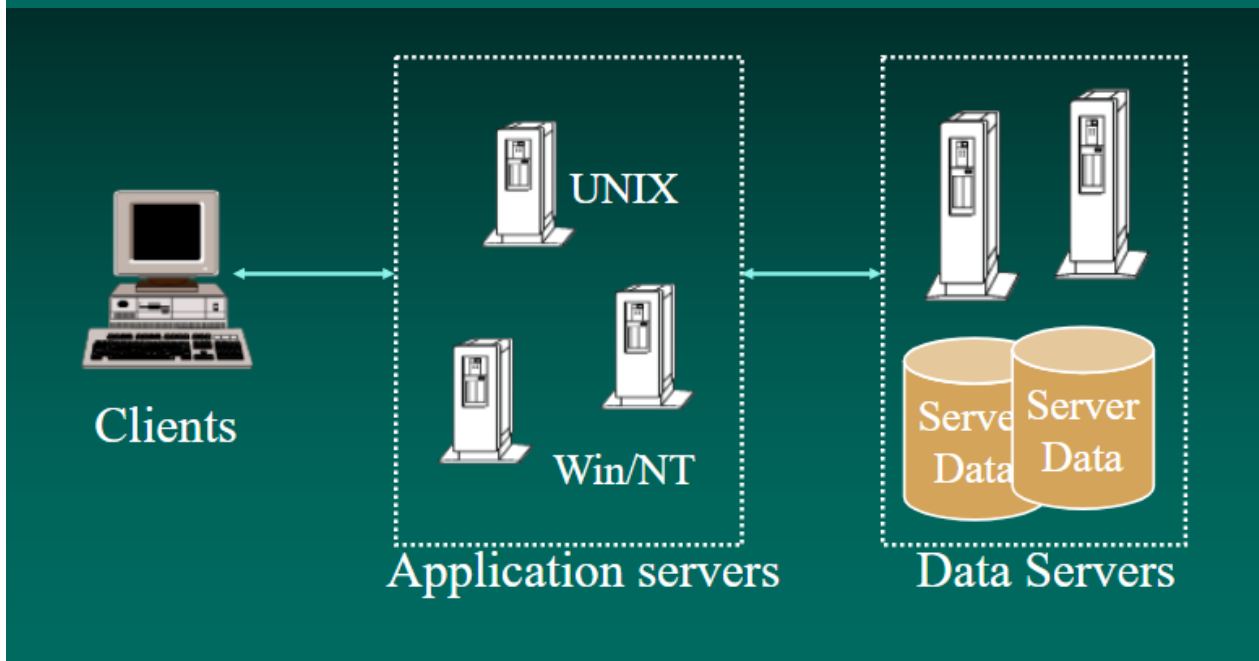
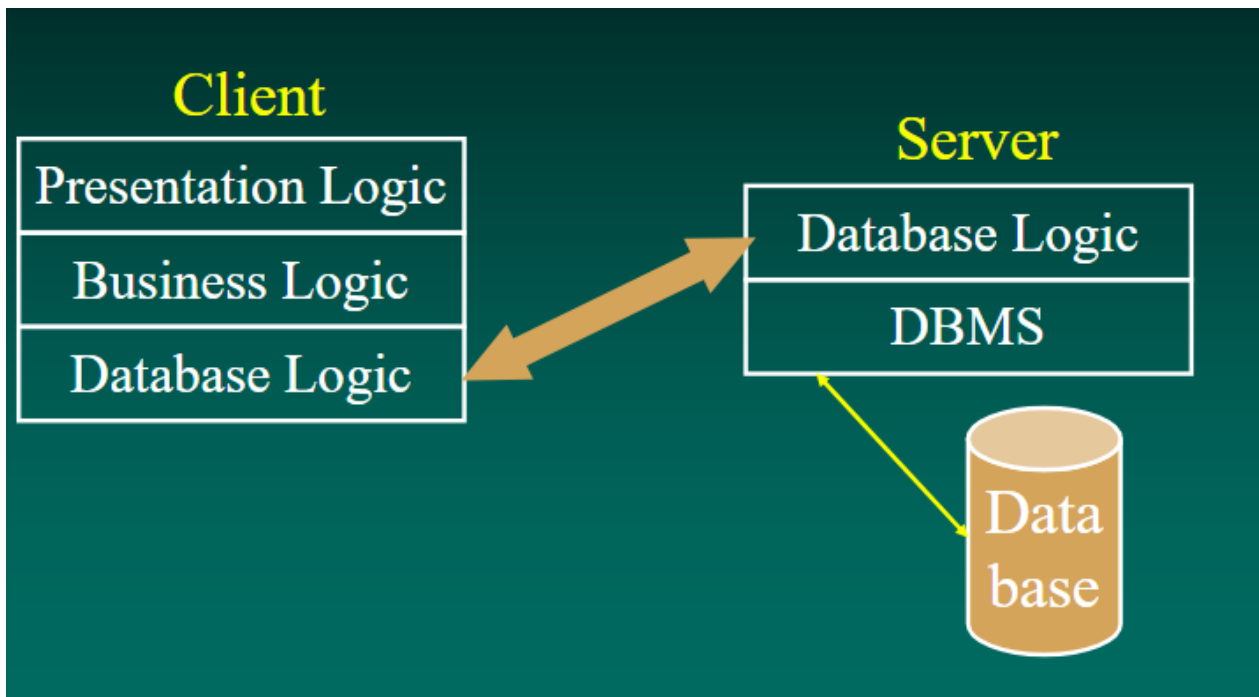
## Peer to peer มีทั้งหมด 2 แบบ

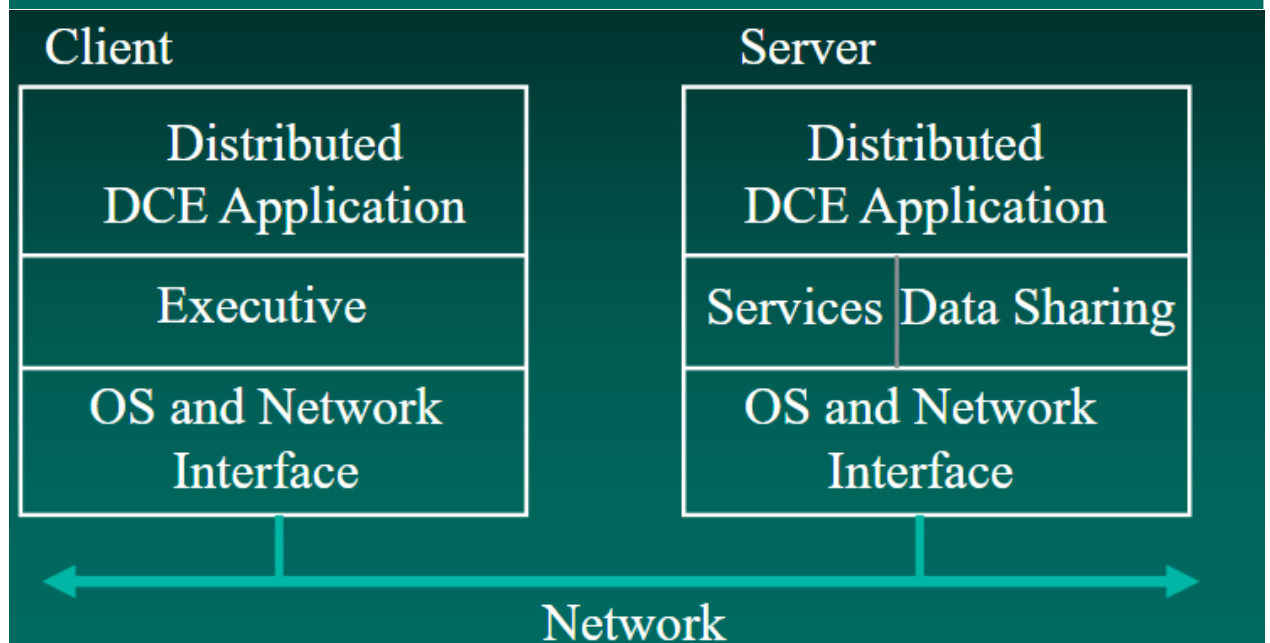
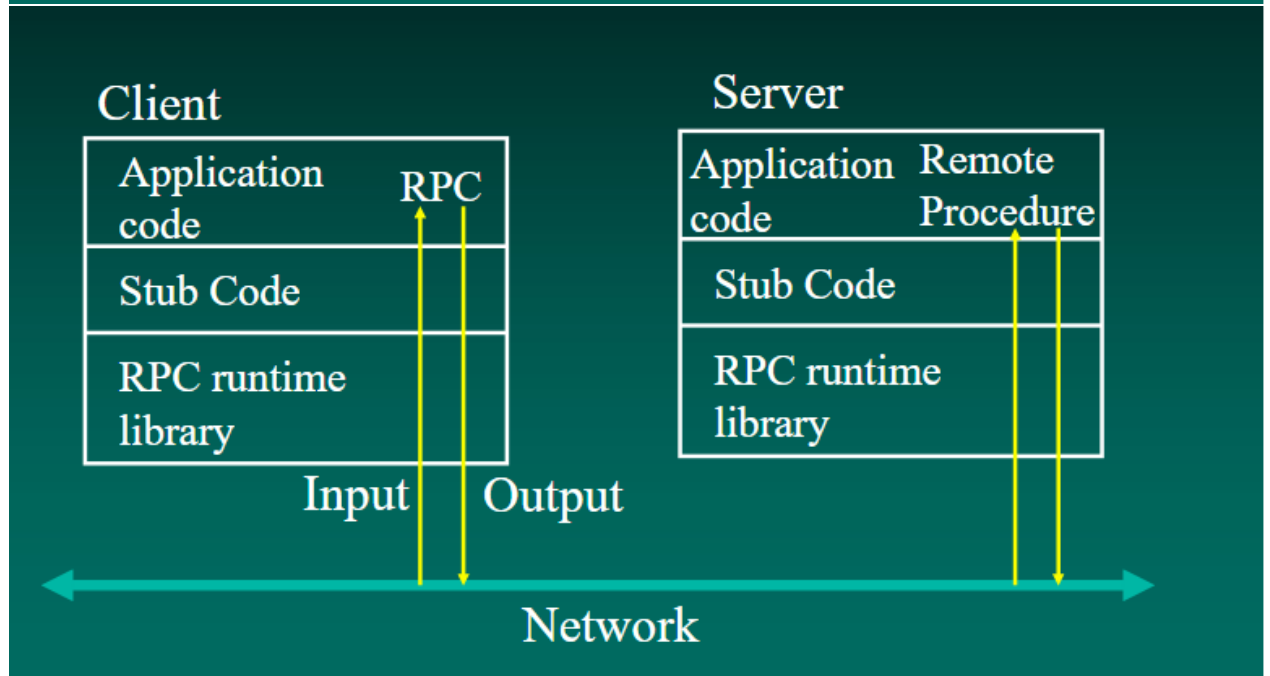
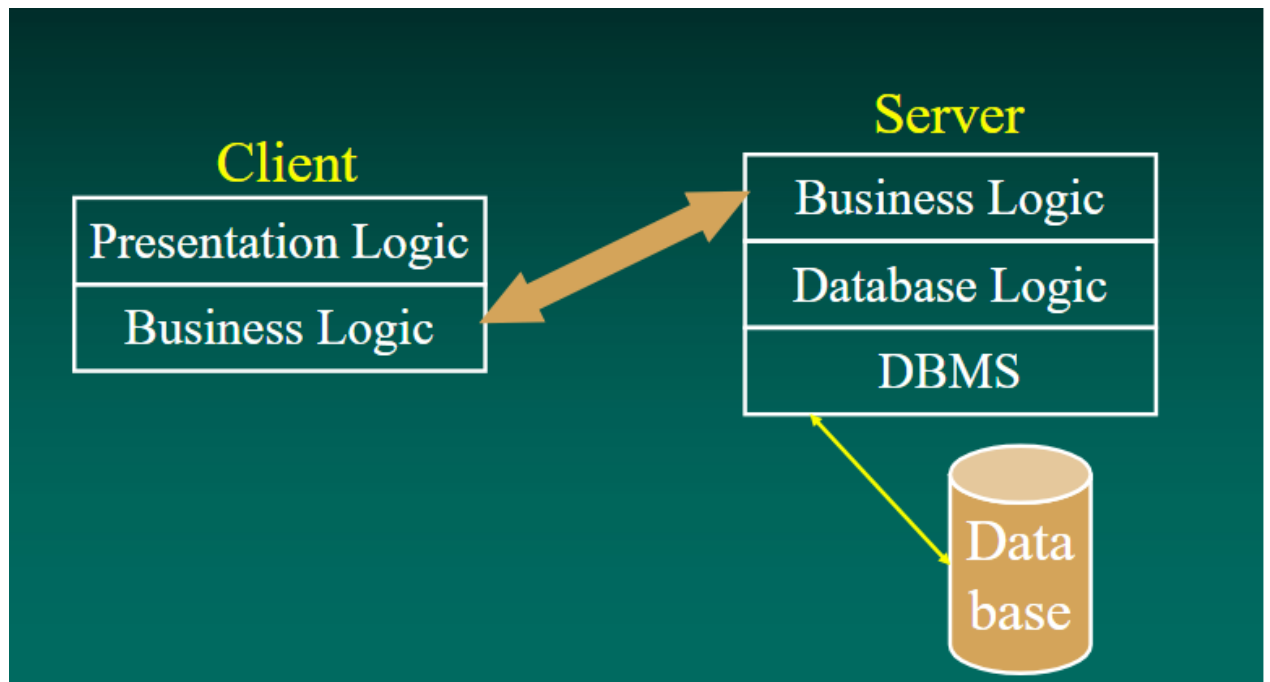
Reader centric: เข้าถึงข้อมูลโดยใครก็ได้

Publisher centric: กำหนดสิทธิในการเข้าถึงข้อมูล คล้ายๆการตีพิมพ์หนังสือ

คือไม่จำเป็นต้องผู้ดูแลหรือจัดการระบบ หรือเรียกอีกชื่อว่า Woke group เหมาะสำหรับการทำงานกับคอมไม่เกิน  
10 เครื่อง มีทรัพยากรที่แชร์กันไม่มากเช่นไฟล์ เครื่องพิมพ์ ไม่จำเป็นต้องรักษาความปลอดภัยของข้อมูล การ  
ขยายตัวของเครือข่ายในอนาคตไม่มาก เหมาะกับองค์กรขนาดเล็ก เป็นได้ทั้ง Client และ Server คนใช้ต้อง  
ฝึกอบรม แต่อาจจะเป็นการยากเนื่องจากผู้ใช้แต่ละคนอาจมีงานอื่นที่ต้องทำ







## Hybrid Peer-to-Peer

โมเดลแบบ Hybrid P2P นี้จะมีเครื่อง server ที่ทำหน้าที่ควบคุมรายละเอียดของข้อมูลที่อยู่ภายในเครือข่ายแต่การส่ง ข้อมูลจะเป็นแบบเดียวกับโมเดล Pure P2P (ส่งถึงกันโดยตรง) โมเดลแบบนี้จะช่วยลดปัญหาเรื่องการจัดการข้อมูลที่ทำให้ยากในโมเดลแบบ Pure P2P โดยเครื่อง server จะทำหน้าที่คอยตรวจสอบสถานะของทุก ๆ เพียร์ และควบคุมการไหลของข้อมูลในเครือข่ายแต่เพราะยังต้องใช้เครื่อง server กลางอยู่ดังนั้นถ้าเครื่อง server กลางเสียไปก็จะเสียการควบคุมข้อมูลไปแต่ละเพียร์ ก็จะสามารถแลกเปลี่ยนข้อมูลกันได้อยู่เนื่องจากมีการควบคุมข้อมูลที่ติดตั้งโมเดล นี้จึงมีความสามารถในการขยายขนาดเครือข่ายได้ดีกว่าโมเดล Pure P2P แต่ก็ยังมีขีดจำกัดของการขยายอยู่ที่จำนวนเครื่องลูกของเครื่อง server ที่จะรับได้ โมเดลแบบนี้มี ประสิทธิภาพที่จะนำไปใช้ กับแอปพลิเคชันต่าง ๆ แต่ไม่สามารถนำไปใช้กับแอปพลิเคชันที่มีขนาดของปัญหาใหญ่ ๆ ได้ เช่น Napster



## Unstructured peer-to-peer

เครือข่าย Unstructured peer-to-peer ที่ไม่ได้จัดการให้เป็นรูปร่าง หรือ โครงสร้างเฉพาะเจาะจง ที่แต่ละโหนดในเครือข่ายจะไม่ถูกชั้นตอนวิธีใด ๆ มาจัดการให้ปฏิบัติงานให้เกิดการเชื่อมต่อที่เกิดประโยชน์สูงสุด โดยแต่ละเพียร์ หรือ โหนด ถูกปล่อยให้เป็นอิสระในการเชื่อมโยงเพื่อนบ้าน

โดยทั่วไป สามารถแบ่งสถาปัตยกรรมของระบบเพียร์ทูเพียร์ แบบไม่เป็นโครงสร้าง หรือ unstructured ได้ออกเป็น สามประเภท หลักด้วยกันคือ

- ระบบ เพียร์ทูเพียร์ บริสุทธิ์ แบบดั้งเดิม pure peer-to-peer ระบบจะประกอบไปด้วย ลักษณะของการเชื่อมต่อในรูปแบบเพื่อนเพียร์ แต่ละโหนดจะเชื่อมต่อเฉพาะเพื่อนที่ตนรู้จักไม่ว่าจะใกล้หรือไกล equipotent ในระบบนี้จะมีชั้นค้นหาเส้นทางหรือ routing layer เพียงแค่หนึ่งชั้นเท่านั้น และ ภายในเครือข่ายจะไม่โหนดใดถูกเลือกให้ปฏิบัติหน้าที่ หรือ มี ฟังก์ชันพิเศษไปกว่าโหนดอื่น ๆ

เช่น Gnutella, Morpheus (software), Kazaa(KaZaA)

## structured peer-to-peer

เครือข่าย structured peer-to-peer แบบมีโครงสร้าง

เช่น Chord, CAN(Content addressable network), Pastry(DHT), Tornado



## Bit Torrent (บิททอร์เรนท์)

เป็นมาตรฐานชนิดหนึ่งที่สามารถทำให้เราดาวน์โหลดไฟล์ขนาดใหญ่ ๆ เช่น โปรแกรม หรือ ภาพยนต์ ได้ในความเร็วสูง โดยมีพื้นฐานอยู่บนการแบ่งปันกันและกัน บิททอร์เรนท์ ต่างจากการดาวน์โหลดธรรมดาตรงที่มันจะใช้แบบวิท (ขนาดช่องสัญญาณอินเทอร์เน็ต) อย่างเต็มประสิทธิภาพโดยการแบ่งไฟล์ใหญ่นั้นออกเป็นชิ้น ๆ ขนาดเล็ก แล้วจึงดาวน์โหลดไฟล์เหล่านั้นพร้อม ๆ กันซึ่งวิธีการนี้เป็นที่นิยมอย่างมากในการโหลดไฟล์ขนาดใหญ่ ๆ

Bit Torrent เป็นการแบ่งปันไฟล์,ข้อมูล ที่มีในเครื่องคอมพิวเตอร์ (PC) เชื่อมต่อกันในกลุ่มสมาชิกบิทส่งต่อข้อมูลระหว่างเครื่อง PC กับเครื่อง PC ด้วยกัน ผ่านเครือข่ายอินเทอร์เน็ตโดยใช้อัตราเร็วในการเชื่อมต่ออินเทอร์เน็ตเป็นตัวกำหนดความเร็วในการรับส่งข้อมูล สมาชิกแต่ละรายจะเป็นผู้กำหนดว่าจะเปิดเครื่องให้ส่งข้อมูลหรือปิดเครื่องไม่ให้ส่งข้อมูล (บ่งบอกถึงความมีน้ำใจ) ยิ่งสมาชิกจำนวนมาก ๆ เปิดเครื่องเพื่อช่วยกันส่งข้อมูล ยิ่งทำให้อัตราเร็วในการส่งข้อมูลมีมากขึ้น(ทำให้โหลดเสร็จเร็ว) และขึ้นอยู่กับว่าคู่สายที่เชื่อมต่ออินเทอร์เน็ตของเรามีความเร็วเพียงพอที่จะดาวน์โหลดได้ตามความเร็วรวมของกลุ่มหรือไม่และในขณะที่เราดาวน์โหลดไฟล์(แม้ยังโหลดไม่เสร็จสมบูรณ์) เราจะเป็นผู้ส่งต่อไฟล์ (อัปโหลด) ไปให้เครื่องอื่น ๆ ในกลุ่มของบิทเดียวกันด้วย เรียกได้ว่าเป็นทั้งผู้รับและผู้ให้ (จึงเรียกว่าการแบ่งปัน)

ประสิทธิภาพของ บิททอร์เรนท์ ที่ดีกว่า เพียร์ทูเพียร์

- 1.บิททอร์เรนท์ใช้ในการเชื่อมโยงข้อมูลเครื่องแม่ข่ายและเครื่องลูกข่ายผ่านอินเทอร์เน็ตได้สะดวกกว่า
- 2.ในขณะที่บิททอร์เรนท์กำลังดาวน์โหลดไฟล์ ยังสามารถส่งต่อไฟล์ไปได้ด้วยในขณะเดียวกันถึงแม้ยังโหลดไม่เสร็จ
- 3.ในบิททอร์เรนท์จะมีเครื่องลูกข่ายช่วยกันปล่อยไฟล์หลายๆเครื่อง ถึงแม้แต่ละเครื่องมีความเร็วต่ำแต่เมื่อรวมกันจะได้ความเร็วมากยิ่งขึ้น
- 4.บิททอร์เรนท์ มีไฟล์ Torrent เป็นตัวเก็บชิ้นส่วนขนาดไฟล์ตัวจริงไว้ แยกซอยเป็นชิ้นเล็ก ๆ ทำให้ส่งถ่ายข้อมูลได้แม่นยำ ตรวจสอบได้ตลอดเวลา

เมื่อไฟล์เริ่มต้นเผยแพร่มาจากคอมพิวเตอร์เครื่องหนึ่ง เครื่องอื่น ๆ ที่ต้องการไฟล์ (หรือผู้ที่รอโหลดอยู่นั่นเอง) ก็จะค่อยๆ ได้รับชิ้นส่วนไฟล์ไปทีละชิ้นทีละชิ้นแบบสุ่ม เหมือนภาพต่อจิ๊กซอว์ ทันทีที่ได้รับชิ้นส่วนไฟล์มา คอมพิวเตอร์เครื่องนั้นก็สามารถส่งต่อชิ้นส่วนไฟล์ที่ได้รับมาแล้วให้เครื่องอื่นที่ยังไม่มีได้ทันที ไม่ต้องรอให้ตัวเองได้ชิ้นส่วนไฟล์จนครบ 100% เสียก่อน เป็นลักษณะของการเติมเต็มให้กัน ชิ้นส่วนไฟล์ตรงไหนที่ขาดไป สุดท้ายแล้วก็จะได้รับมาจากคอมพิวเตอร์เครื่องใดเครื่องหนึ่งในเครือข่ายในที่สุด ด้วยสาเหตุนี้เอง โปรแกรมบิททอร์เรนท์จึงสามารถทำให้การส่งผ่านข้อมูลสามารถอำนวยความสะดวกได้ทั้งขาเข้าและขาออก

Tracker Server (แทรคเกอร์) เครือข่ายของบิททอร์เรนท์นั้นมีลักษณะโยงใยถึงกันหมด มีแทรคเกอร์ (Tracker) ที่ทำหน้าที่เป็นศูนย์กลางเก็บข้อมูลและสถิติต่าง ๆ ของผู้ร่วมใช้งาน, เสมือนประธานผู้ควบคุมการประชุม อย่างไรก็ตาม โปรแกรมบิททอร์เรนท์บางแบบ ไม่จำเป็นต้องมีแทรคเกอร์เครือข่ายของการใช้โปรแกรมบิท

ทอร์เรนท์นั้นเป็นลักษณะโยงใยถึงกันหมด ทุกเครื่องคอมพิวเตอร์สามารถรับส่งไฟล์ถึงกันได้ตลอดเวลา ซึ่งทุกเครื่องจะเป็นทั้งผู้รับและผู้ให้ ดังนั้น Tracker จึงต้องมีเน็ตที่แรงเพราะไม่ได้รับส่งไฟล์เอง สิ่งที่ทำให้ Bit Torrent อยู่ได้ก็คือหลักการที่ผู้ใช้ควรส่งไฟล์ขณะเดียวกับที่รับไฟล์ หากมีผู้ใช้งานก็จะเร็วมาก การทำงานของ

Bit Torrent ก็คือการหั่นไฟล์หนึ่งเป็นหลายๆ ส่วน แล้วส่งคนละส่วนไปยังผู้รับหลายคน พอผู้รับเหล่านั้นได้รับส่วนเหล่านั้นก็จะสามารถรับส่งกันเองเพราะต่างคนต่างมีชิ้นส่วนที่คนอื่นไม่มี ทำให้ไม่ต้องพึ่งผู้ส่งผู้เดียว หลักการทำงานของโปรแกรม Bittorrent เครือข่ายของการใช้โปรแกรม Bit Torrent นั้นเป็นลักษณะโยงใยถึงกันหมด ทุกเครื่องคอมพิวเตอร์สามารถรับส่งไฟล์ถึงกันได้ตลอดเวลา โดยในทาง Bittorrent นั้นจะเรียกคอมพิวเตอร์ที่รับส่งข้อมูลว่า เพียร์ (Peer) ซึ่งทุกเครื่องจะเป็นทั้ง ผู้รับ (Leecher) และ ผู้ให้ (Seeder)

### **ข้อดีของ Bit Torrent Client**

คือระหว่างที่เราดาวน์โหลดไฟล์ลงเครื่องเรา เราสามารถที่จะอัปโหลดแจกจ่ายชิ้นส่วนไฟล์ให้กับคนอื่นที่ต้องการจะดาวน์โหลดไฟล์เหมือนกันกับเราได้เลยพร้อมๆกัน ไม่จำเป็นต้องให้เราดาวน์โหลดให้เสร็จก่อนถึงจะปล่อยได้ ผู้ส่งไฟล์ไม่ใช่ Server เพียงฝ่ายเดียว แต่เป็นลูกข่าย Client ที่ช่วยกันส่งชิ้นส่วนไฟล์ให้แก่กัน ทำให้เกิดความรวดเร็วในการส่งข้อมูล และช่วยเพิ่มอัตราการถ่ายโอนข้อมูลเพราะมีลูกข่ายที่เชื่อมต่อกันอยู่จำนวนมาก และไม่มีเครื่องใดในเครือข่ายที่ทำงานหนักจนเกินไป Torrent เป็นการทำงานแบบ Peer to Peer การเชื่อมต่อเครื่องคอมพิวเตอร์เข้าด้วยกัน เพื่อการแลกเปลี่ยนข้อมูลระหว่างกันไฟล์ที่นำมาแชร์กันมาจากไหน? มาจากเครื่องของแต่ละ User แต่ทำการโหลดกันโดยตรงซึ่งแน่นอนว่าเว็บแทรคเกอร์ Tracker (เว็บผู้ให้บริการทอร์เรนท์) ทำหน้าที่เหมือน ตำรวจจราจร เท่านั้นเองTorrent มีความแม่นยำในการดาวน์โหลดสูง กล่าวคือ เมื่อเราดาวน์โหลดไฟล์ประเภทเพลง หรือหนัง ฯลฯ โอกาสที่ไฟล์จะเสียนั้นน้อยมากๆ จนแทบไม่มีเลย เพราะ ไฟล์ torrent เป็นตัวเก็บชิ้นส่วนขนาดไฟล์ตัวจริงไว้ แยกซอยเป็นชิ้นเล็กๆ ทำให้ส่งถ่ายข้อมูลได้แม่นยำ

### **รูปแบบการทำงานของ Bit Torrent**

Bit Torrent ที่มีการรับ-ส่ง File จาก Client หนึ่งไป Client หนึ่ง โดยจะเป็นข้อดีกว่าในระบบ FTP (File Transfer Protocol) , HTTP (Hypertext Transfer Protocol) ที่ข้อมูลมาจาก Server กลางอย่างเดียว ถ้าเมื่อมีคนเข้ามาใช้ข้อมูลเยอะ ๆ จะทำให้การ Transfer ช้าลง แต่ถ้าเป็น โครงข่ายของ Bittorrent ถ้ามีคนเข้ามาใช้ร่วมโครงข่ายมากก็จะทำให้มีการ Transfer มากขึ้นตามไป

## เครือข่ายซ้อนทับ (overlay network)

หมายถึง เครือข่ายคอมพิวเตอร์ที่ถูกสร้างให้อยู่ชั้นบนถัดจากเครือข่ายอื่น ๆ ที่มีอยู่แล้ว เช่น อยู่ชั้นบนของเครือข่ายระดับเดียวกัน (peer-to-peer network) โหนดต่าง ๆ ที่อยู่ภายในเครือข่ายซ้อนทับสามารถถูกเชื่อมต่อกันได้ในลักษณะที่เรียกว่า เสมือนจริง หรือ virtual หรือ logical links ซึ่งแต่ละลิงก์ อาจจะเชื่อมโยงกันผ่านกับโหนดอื่น ๆ ที่ตั้งอยู่ห่างไกลในระยะทางของเครือข่ายจริง หรือ underlying network ตัวอย่างเช่น ในระบบเครือข่ายแบบกระจาย (distributed systems) เช่น cloud computing เครือข่ายระดับเดียวกัน ระบบ client-server applications ต่างทำงานบนพื้นฐานของเครือข่ายซ้อนทับ เพราะโหนดเหล่านี้ต้องปฏิบัติการอยู่ชั้นบนระดับอินเทอร์เน็ต ถูกสร้างขึ้นให้ซ้อนทับอยู่ชั้นบนของเครือข่ายที่เชื่อมโยงกันด้วยระบบโทรศัพท์

