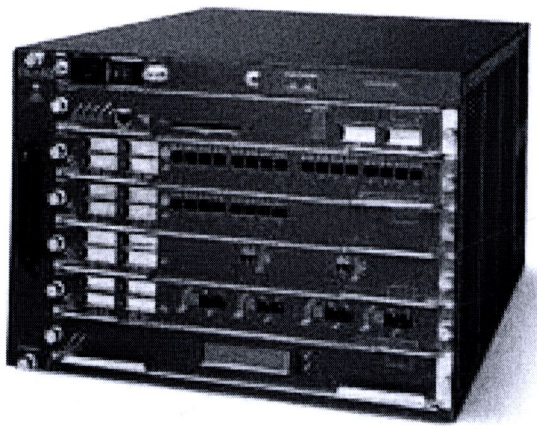
บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การทำงานของเราท์เตอร์

2.1.1 เราท์เตอร์(Router)

เราท์เตอร์เป็นอุปกรณ์ที่ซับซ้อนกว่าสวิตซ์ โดยทำงานเสมือนเป็นเครื่องหรือ Node หนึ่งใน LAN ซึ่งจะทำหน้าที่รับข้อมูลเข้ามาแล้วส่งต่อไปยังปลายทาง โดยอาจส่งในรูปแบบของแพคเกจที่ต่างออกไป เพื่อไปผ่านสายสัญญาณแบบอื่นๆ เช่น สายโทรศัพท์ที่ต่อผ่านโมเด็มก็ได้ ดังนั้นจึงอาจใช้เราท์เตอร์ในการเชื่อมต่อ LAN หลายแบบเข้าด้วยกันผ่าน WAN ได้ และจากการที่เราท์เตอร์ทำตัวเสมือนเป็น Node หนึ่งใน LAN ยังทำให้สามารถทำงานอื่นๆ ได้อีก เช่น รวบรวมข้อมูลเพื่อหาเส้นทางที่ดีที่สุดในการส่งข้อมูลต่อ หรือตรวจสอบข้อมูลที่เข้ามานั้นมาจากไหน ควรจะให้ผ่านหรือไม่ เพื่อช่วยในเรื่องการรักษาความปลอดภัยด้วย ตัวอย่างของเราท์เตอร์ดูได้จากรูป 2.1



รูป 2.1 เราท์เตอร์

2.1.2 ข้อแตกต่างระหว่างเราท์เตอร์กับสวิตซ์

ข้อที่แตกต่างกันระหว่างสวิตซ์กับเราท์เตอร์คือ สวิตซ์จะทำงานในระดับชั้น Data Link Layer คือจะใช้ข้อมูล MAC Address ในการทำงานส่งข้อมูลไปยังที่ใดๆ ซึ่งหมายเลข MAC Address นี้มีการกำหนดมาจากฮาร์ดแวร์หรือที่ส่วนของ Network Interface Card (NIC) และถูกกำหนดมาเฉพาะตัวจากโรงงานไม่ให้ซ้ำกัน ถ้ามีการเปลี่ยน Network

Interface Card (NIC) นี้ไป ก็จะทำให้ MAC Address เปลี่ยนไปด้วย ส่วน Network Layer Address ในการส่งผ่านข้อมูลโปรโตคอลของเครือข่ายชนิดต่างๆ ไม่ว่าจะเป็น IPX, TCP/IP หรือ AppleTalk ซึ่งจะเป็นโปรโตคอลที่ทำงานใน Network Layer การกำหนด Network address ทำได้โดยผู้ดูแลระบบเครือข่ายนั้น ทำให้สามารถแก้ไขเปลี่ยนแปลงได้ง่าย และสามารถใช้อุปกรณ์เราท์เตอร์เชื่อมโยงเครือข่ายที่แยกจากกันให้สามารถส่งผ่านข้อมูลร่วมกันได้และทำให้เครือข่ายขยายออกไปได้เรื่อยๆ

2.1.3 หน้าที่หลักของเราท์เตอร์

เราท์เตอร์เป็นอุปกรณ์ที่ถูกนำมาใช้เพื่อการเชื่อมต่อระหว่างเครือข่าย เช่น การเชื่อมต่อระหว่าง เครือข่าย ที่มีไอพีแอดเดรส 192.168.20.0 กับเครือข่ายที่มีไอพีแอดเดรส 192.168.30.0 เป็นต้น รวมทั้งการเชื่อมต่อเครือข่ายย่อย (เครือข่ายเดียวกันแต่ถูกแบ่งออกเป็นส่วนย่อยๆ เช่น แบ่งเครือข่ายที่มีหมายเลขไอพีแอดเดรส 192.168.30.0 ออกเป็นเครือข่ายย่อยๆ (Subnet) จำนวน 6 เครือข่าย) จากนั้นนำมาเชื่อมต่อกัน เพื่อการสื่อสารกันด้วยเราท์เตอร์

เราท์เตอร์เป็นอุปกรณ์เชื่อมต่อเครือข่ายทำงานในระดับชั้น Network ตามมาตรฐานของ OSI Model หน้าที่หลักของ เราท์เตอร์ได้แก่ การอ้างอิงไอพีแอดเดรสระหว่างเครื่องลูกข่ายที่อยู่กันคนละเครือข่าย รวมทั้งการเลือกและจัดเส้นทางที่ดีที่สุด เพื่อนำข้อมูลข่าวสาร ในรูปแบบของแพ็กเกจจากเครื่องลูกข่ายต้นทางบนเครือข่ายที่ตนดูแลอยู่ไปยังเครื่องลูกข่ายที่อยู่กันคนละเครือข่ายเราท์เตอร์ ที่ใช้เพื่อการเชื่อมต่อระหว่างเครือข่ายผ่านทาง WAN หรือโครงข่ายสาธารณะ อย่างเช่นผ่านทางเฟรมรีเลย์ หรือ ISDN เราเรียกว่า WAN Router ส่วนเราท์เตอร์ที่ใช้เชื่อมต่อระหว่างเครือข่ายชนิดติดตั้งบนแลนและเชื่อมต่อระหว่างเครือข่ายด้วยสายสัญญาณของระบบแลน เรียกว่า Local Router หรือบางครั้งจะถูกเรียกว่า Internal Router ซึ่งเราท์เตอร์ประเภทนี้อาจเป็นเราท์เตอร์ในรูปแบบผลิตภัณฑ์ของเราท์เตอร์เต็มตัว หรือแบบที่มีการติดตั้งการ์ดแลนหลายชุดบนเครื่องคอมพิวเตอร์ที่ติดตั้งเซิร์ฟเวอร์เป็นต้น

หน้าที่หลักของเราท์เตอร์คือการหาเส้นทางในการส่งผ่านข้อมูลที่ดีที่สุด และเป็นตัวกลางในการส่งต่อข้อมูลไปยังเครือข่ายอื่น ทั้งนี้เราท์เตอร์สามารถเชื่อมโยงเครือข่ายที่ใช้สื่อสัญญาณหลายแบบแตกต่างกันได้ไม่ว่าจะเป็น Ethernet, Token Rink หรือ FDDI ซึ่งในแต่ละระบบจะมีแพคเกจเป็นรูปแบบแตกต่างกัน โดยโปรโตคอลที่ทำงานในระดับบนหรือ Layer 3 ขึ้นไปเช่น IP, IPX หรือ AppleTalk เมื่อมีการส่งข้อมูลก็จะบรรจุข้อมูลนั้นเป็นแพคเกจในรูปแบบของ Layer 2 คือ Data Link Layer เมื่อเราท์เตอร์ได้รับข้อมูลมาก็จะตรวจดูในแพคเกจ เพื่อจะทราบว่าใช้โปรโตคอลแบบใด จากนั้นก็จะตรวจดูเส้นทางส่ง

ข้อมูลจากตาราง Routing Table ว่าจะต้องส่งข้อมูลนี้ไปยังเครือข่ายใดจึงจะต่อไปถึงปลายทางได้ แล้วจึงบรรจุข้อมูลลงเป็นแพคเกจของ Data Link Layer ที่ถูกต้องอีกครั้ง เพื่อส่งต่อไปยังเครือข่ายปลายทาง

2.2 คำสั่งที่ใช้ในการทำงานของเราท์เตอร์

เมื่อทำการติดต่อกับเราท์เตอร์ผ่านพอร์ตอนุกรมแล้ว ผู้ใช้จะทำการส่งคำสั่งไปทำการกำหนดค่าการทำงานให้กับเราท์เตอร์ สำหรับคำสั่งในการกำหนดค่าการทำงานเข้าไปมีดังนี้

2.2.1 คำสั่งกำหนดค่าการทำงานของเราท์เตอร์

2.2.1.1 คำสั่งที่ใช้ในการเปลี่ยนชื่อเราท์เตอร์

สามารถทำการเปลี่ยนชื่อเราท์เตอร์แต่ละตัวได้ด้วยคำสั่งดังตัวอย่าง 2.1

ตัวอย่าง 2.1 คำสั่งที่ใช้ในการเปลี่ยนชื่อเราท์เตอร์

```
Router> enable
Router# configure terminal
Router(config)# hostname Cisco
```

จะเห็นว่าชื่อของเราท์เตอร์จะเปลี่ยนจาก Router เป็น Cisco

2.2.1.2 คำสั่งที่ใช้ในการกำหนดค่ารหัสผ่านของเราท์เตอร์

สามารถทำการกำหนดค่า Console Password บนเราท์เตอร์ได้ด้วยคำสั่งดังตัวอย่าง 2.2

ตัวอย่าง 2.2 คำสั่งที่ใช้ในการกำหนดค่ารหัสผ่านของเราท์เตอร์

```
Router(config)# line console 0
Router(config-line)# password ccna
Router(config-line)# login
```

จากคำสั่งด้านบนจะทำให้การเข้าใช้งานผ่าน Console จะต้องใส่รหัสผ่านคำว่า ccna ก่อนถึงเข้าใช้งานได้ดังตัวอย่าง 2.3

ตัวอย่าง 2.3 การเข้าใช้งาน Console เมื่อมีการตั้งรหัสผ่าน

```
Router>
Router>enable
Password:
```

2.2.1.3 คำสั่งที่ใช้ในการกำหนดค่ารหัสผ่านในส่วนของ Virtual Terminal Lines ของเราท์เตอร์

สามารถทำการกำหนดค่ารหัสผ่านในส่วนของ Virtual Terminal Lines บนเราท์เตอร์ได้ด้วยคำสั่งดังตัวอย่าง 2.4

ตัวอย่าง 2.4 คำสั่งที่ใช้ในการกำหนดค่ารหัสผ่านในส่วนของ Virtual Terminal Lines ของเราท์เตอร์

```
Router(config)# line vty 0
Router(config-line)# password ccna
Router(config-line)# login
```

จากคำสั่งด้านบนจะทำให้การเข้าใช้งานผ่านการ Telnet จะต้องใส่รหัสผ่านคำว่า ccna ก่อนถึงเข้าใช้งานได้ดังตัวอย่าง 2.5

ตัวอย่าง 2.5 การเข้าใช้งาน Telnet เข้าไปที่ Router_first เมื่อมีการตั้งรหัสผ่าน

```
Router>
Router>telnet 192.168.1.1
```

2.2.1.4 คำสั่งที่ใช้กำหนดค่ารหัสผ่านในการเข้าใช้ Privileged Exec Mode ของเราท์เตอร์

สามารถทำการกำหนดค่ารหัสผ่านสำหรับการเข้าใช้งาน Privileged Exec Mode ได้ด้วยคำสั่งดังตัวอย่าง 2.6

ตัวอย่าง 2.6 คำสั่งที่ใช้กำหนดค่ารหัสผ่านในการเข้าใช้ Privileged Exec Mode

```
Router(config)# enable password ccna
Router(config)# exit
```

จากคำสั่งด้านบนจะทำให้การเข้าใช้งาน Privileged Exec Mode จะต้องใส่รหัสผ่านคำว่า ccna ก่อนถึงเข้าใช้งานได้แต่ถ้าต้องการให้รหัสผ่านนั้นมีการ Encryption ด้วยจะต้องใช้คำสั่งดังตัวอย่าง 2.7 แทน

ตัวอย่าง 2.7 การกำหนดให้รหัสผ่านมีการ Encryption

```
Router(config)# enable secret ccna
Router(config)# exit
```

2.2.1.5 คำสั่งที่ใช้ในการกำหนดค่าให้กับอินเตอร์เฟส

สามารถทำการกำหนดค่า IP Address, Subnet Mark, Clock rate, และการเปิดใช้งานอินเตอร์เฟสให้แต่ละอินเตอร์เฟสได้ด้วยคำสั่งดังตัวอย่าง 2.8

ตัวอย่าง 2.8 คำสั่งที่ใช้ในการกำหนดค่าให้กับอินเตอร์เฟส

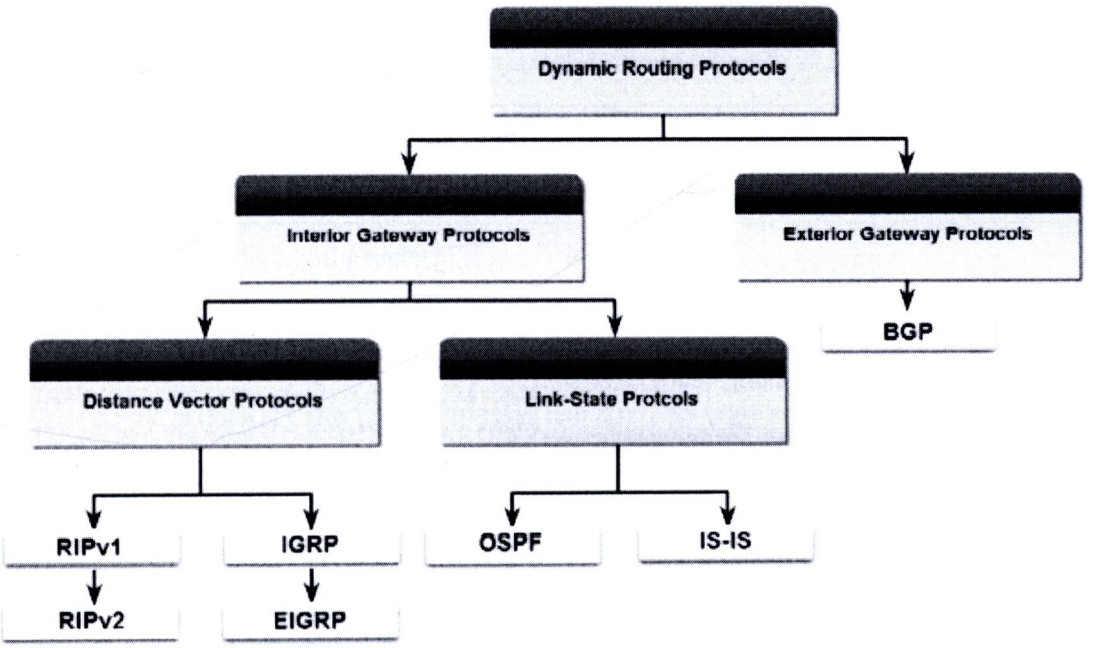
```
Router(config)# interface serial 0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# clock rate 56000
//only DCE Interface Type
```

DCE Interface Type

จากคำสั่งด้านบนเป็นการกำหนดค่าให้กับอินเตอร์เฟส serial 0/0 ซึ่งกำหนดให้ IP Address เท่ากับ 192.168.1.1, Subnet Mark เท่ากับ 255.255.255.0 และ Clock rate เท่ากับ 56000 (สำหรับอินเตอร์เฟสที่มีชนิดเป็น DCE เท่านั้น) จากนั้นก็ทำการเปิดใช้งานอินเตอร์เฟสนี้ด้วยคำสั่ง no shutdown

2.2.1.6 คำสั่งที่ใช้ในการกำหนดค่าการส่งต่อแพคเกจของเราท์เตอร์

เนื่องจากเราท์เตอร์เป็นอุปกรณ์เครือข่ายที่ทำงานที่ลำดับชั้นที่สาม ซึ่งมีความสามารถในการส่งผ่านแพคเกจข้ามเนตเวิร์กที่ต่างกันได้ โดยจะมีรูปแบบในการส่งผ่านแพคเกจอยู่สองรูปแบบคือ Static Routing และ Dynamic Routing โดย Dynamic Routing จะอาศัยโปรโตคอลต่าง ๆ ซึ่งแยกตามประเภทได้ดังรูป 2.3



รูป 2.2 การแบ่งแยกประเภทของ Dynamic Routing Protocols

การกำหนดค่าการส่งผ่านแพคเกจในระบบเครือข่ายสามารถทำได้โดยใช้คำสั่งต่างๆ แยกตามประเภทดังตัวอย่างต่อไปนี้

ตัวอย่าง 2.9 คำสั่งที่ใช้ในการกำหนด Static Route

```
//Static Routing
Router(config)# ip route 192.168.2.0 255.255.255.0 serial 0/0
OR
Router(config)# ip route 192.168.2.0 255.255.255.0
```

ตัวอย่าง 2.10 คำสั่งที่ใช้ในการกำหนด RIPv1

```
//RIPv1
Router(config)# router rip
Router(config-router)# network 192.168.1.0
Router(config-router)# network 192.168.2.0
.
```

ตัวอย่าง 2.11 คำสั่งที่ใช้ในการกำหนด RIPv2

```
//RIPv2
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# network 192.168.1.0
Router(config-router)# network 192.168.2.0
```

ตัวอย่าง 2.12 คำสั่งที่ใช้ในการกำหนด EIGRP

```
//EIGRP
Router(config)# router eigrp 123
Router(config-router)# network 192.168.1.0
Router(config-router)# network 192.168.2.0
.
.
```

ตัวอย่าง 2.13 คำสั่งที่ใช้ในการกำหนด OSPF

```
//OSPF
Router(config)# ip routing
Router(config)# router ospf 456
Router(config-router)# network 192.168.1.1 0.0.0.255 area 0
Router(config-router)# network 192.168.1.2 0.0.0.255 area 0


Router(config-router)# log-adjacency-changes
Router(config-router)# exit
Router(config)# exit
```

คำสั่งต่างๆ เหล่านี้เป็นคำสั่งที่ใช้ในการกำหนดค่าการทำงานให้เราท์เตอร์สามารถส่งผ่านแพคเกจข้ามเนตเวิร์กที่ต่างกันได้ โดยแต่ละโปรโตคอลจะมีรูปแบบการกำหนดค่าที่ต่างกันไป

2.3 หลักการและโปรโตคอลที่เกี่ยวข้อง

2.3.1 โหมดในการใช้งาน CLI

โดยปกติเมื่อเปิดโปรแกรมจำลองเทอร์มินอลอย่างเช่น HyperTerminal หรือ SecureCRT และต่อพอร์ตคอนโซลเข้าไปยังเราท์เตอร์ หลังจากกด enter แล้ว โดยปกติโหมดแรกที่จะพบก็คือ User EXEC Mode (ในการเข้าถึง CLI จะมี 3 วิธี ได้แก่ การต่อพอร์ตคอนโซล, การเทลเนตผ่านเนตเวิร์ก และการต่อพอร์ต AUX) ปกติแล้วโหมดของการใช้งาน CLI โหมดแรกที่จะพบก็คือโหมด User EXEC เสมอ โดยที่โหมด User EXEC เป็นโหมดที่เปิดโอกาสให้มีการใช้คำสั่งพื้นฐานที่จำเป็นต่อการดูสถานะของการทำงานบางอย่างของเราท์เตอร์ โดยจำกัดคำสั่งที่สามารถคีย์ได้ไว้ ข้อสังเกตก็คือ ภายในโหมดนี้จะมีเครื่องหมายพรอมต์ (prompt) รอรับคำสั่งที่ได้รับจะเป็นเครื่องหมาย ">"

หากต้องการเรียกใช้คำสั่งอื่นๆทั้งหมด เราจำเป็นต้องเปลี่ยนจากโหมด User EXEC เข้าสู่อีกโหมดหนึ่งที่เรียกว่า Privileged EXEC Mode ด้วยการใช้คำสั่ง enable (โหมดนี้นิยมเรียกอีกอย่างว่า ENABLE Mode) โหมดนี้เปิดโอกาสให้เราใช้งานได้ทุกคำสั่งที่เกี่ยวข้องกับการสำรวจดูสถานะของการทำงานและความเป็นไปต่างๆของเราท์เตอร์ ข้อสังเกตก็คือ ภายในโหมดนี้จะมีเครื่องหมายพรอมต์(prompt) รอรับคำสั่งที่ได้รับจะเป็น

เครื่องหมาย "#" ภายใต้โหมด Privileged EXEC นี้ เราทำได้แค่เพียงการดูสถานะของการทำงานต่างๆของเราท์เตอร์ทั้งหมดเท่านั้น แต่ยังไม่สามารถทำการกำหนดตั้งค่าต่างๆให้กับเราท์เตอร์ได้ จนกว่าจะได้เปลี่ยนจากโหมด Privileged EXEC (หรือที่เรียกว่า Enable Mode) เข้าสู่โหมดที่เรียกว่า Global Configuration Mode(โกลบอลคอนฟิกูเรชันโหมด) ด้วยคำสั่ง configure terminal ข้อสังเกตคือ ภายในโหมดนี้ เครื่องหมายพรอมต์(prompt) รอรับคำสั่งที่ได้รับจะเป็น (config)#

หลังจากเข้าสู่โหมด Global Configuration Mode แล้ว จะสามารถกำหนดตั้งค่าต่างๆได้ทั้งหมด ภายใต้โหมด Global Configuration นี้ ผู้ใช้สามารถที่จะตั้งค่าคำสั่งต่างๆลงไปได้เพื่อกำหนดตั้งค่าต่างๆ และเปิดฟีเจอร์ต่างๆของเราท์เตอร์ขึ้นมา คำสั่งต่างๆที่คีย์ได้ภายใต้โหมดนี้โดยตรงจะถูกเรียกว่าเป็นคำสั่งที่ทำงานภายใต้ Global Configuration Mode คำสั่งที่ตั้งค่าได้ภายใต้ Global Configuration Mode มักเป็นคำสั่งที่มีผลต่อการทำงานของเราท์เตอร์ทั้งตัว ตัวอย่างเช่น คำสั่ง hostname, คำสั่ง ip domain-lookup เป็นต้น ภายใต้ Global Configuration Mode นี้ยังมีการแบ่งย่อยออกเป็นโหมด Sub-Configuration สำหรับการกำหนดตั้งค่าให้กับองค์ประกอบหรือฟีเจอร์ต่างๆของเราท์เตอร์อีก โดยคำสั่งที่คีย์ลงไปเพื่อกำหนดตั้งค่าต่างๆของเราท์เตอร์จะถูกนำไปสร้างเป็นสิ่งที่เรียกว่า "running-configuration" ซึ่ง "แอกทีฟ (active)" อยู่ในหน่วยความจำ (RAM) ของเราท์เตอร์และ IOS ของเราท์เตอร์ ซึ่งเป็นระบบปฏิบัติการที่มีหน้าที่ควบคุมเราท์เตอร์ก็จะทำงานต่างๆ ตาม running-configuration ที่แอกทีฟอยู่ในหน่วยความจำ RAM หากต้องการดูว่าขณะนี้ running-configuration ที่แอกทีฟอยู่มีอะไรบ้าง คำสั่งที่ใช้คือ show running-configuration หรือย่อๆว่า show run

2.3.2 การทำให้สามารถเทลเนต (Telnet) มายังเราท์เตอร์ได้

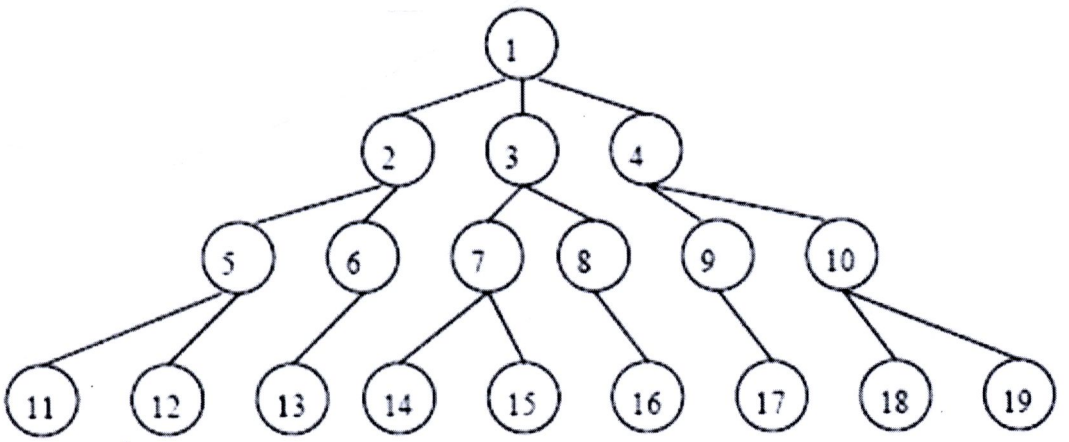
โดยปกติเราท์เตอร์จะยังไม่เปิดโอกาสให้เราได้เทลเนตผ่านเนตเวิร์กเข้ามายังตัวมันได้ จนกว่าเราจะได้ทำการต่อพอร์ตคอนโซลเข้ามาเพื่อสั่งอีนาเบิลอินเตอร์เฟสที่เป็น LAN และตั้งค่าหมายเลข IP Address ให้กับมัน และทำการตั้งค่ารหัสผ่านบน LINE VTY ด้วยคำสั่ง password และกำหนดให้มีการล็อกอินด้วยคำสั่ง login หรือไม่ก็อีกทางเลือกหนึ่งคือ การใส่คำสั่ง no login เข้าไปภายใต้ Line VTY คำสั่ง no login จะเป็นการสั่งให้เราท์เตอร์ไม่ต้องตรวจสอบการล็อกอินของผู้ใช้ที่เข้ามาทาง Line VTY ผู้ใช้ที่เทลเนตเข้ามาทาง Line VTY สามารถเข้าถึง CLI ในโหมด User EXEC ได้ทันทีโดยไม่ต้องระบุรหัสผ่านใดๆ (แต่แบบนี้ไม่ปลอดภัย) การกำหนดรหัสผ่านก่อนเข้าสู่ Privileged Mode (หรือที่เรียกว่า Enable Mode) ด้วยคำสั่ง enable secret หรือ enable password ทั้งนี้ขึ้นกับว่าต้องการให้ผู้ที่เข้าถึงผ่านทางเทลเนตสามารถเข้าสู่โหมด Privileged EXEC ได้หรือไม่ หาก

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

ต้องการ เราจำเป็นต้องตั้งค่ารหัสผ่านด้วยคำสั่ง enable secret หรือ enable password (ควรใช้ enable secret)

2.4 การค้นหาแบบกว้างก่อน (Breadth First Search)

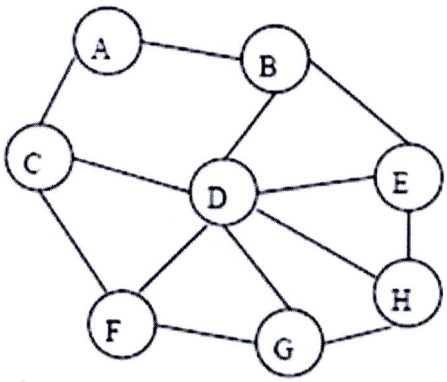
การค้นหาแบบกว้างก่อนเป็นการค้นหาที่กำหนดทิศทางการค้นหาแบบทีละระดับของโครงสร้างต้นไม้โดยเริ่มจากโหนดราก (ระดับที่ 0) แล้วลงมาระดับที่ 1 จากซ้ายไปขวา เมื่อเสร็จระดับที่ 1 ไประดับที่ 2 จากซ้ายไปขวา เช่นกัน ทำเช่นนี้ไปเรื่อยๆ จนพบโหนดที่ต้องการตามรูป 2.6 โดยลำดับการเดินทางของโหนดเป็นไปตามหมายเลขที่กำกับไว้บนโหนด



รูป 2.3 ลำดับการเดินทางบนโหนดของการค้นหาแบบกว้างก่อนบนโครงสร้างต้นไม้

สำหรับการค้นหาแบบกว้างก่อน (Breadth First Search) บนโครงสร้างต้นไม้ จะอาศัยโครงสร้างข้อมูลแบบคิว (Queue) มาช่วย และด้วยวิธีการเช่นเดียวกับการค้นหาแบบลึกก่อนคือ ให้เริ่มต้นสำรวจที่โหนดเริ่มต้น แล้วนำโหนดข้างเคียงเก็บไว้ในคิว เมื่อสำรวจโหนดเริ่มต้นเสร็จ ให้นำข้อมูลในคิวออกมาสำรวจ แล้วนำโหนดข้างเคียงที่ยังไม่ได้สำรวจและไม่ได้อยู่ในคิวใส่คิวไว้ ทำเช่นนี้ไปเรื่อยๆ จนพบโหนดที่ต้องการ หรือเมื่อสำรวจครบทุกโหนด

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย
วันที่ 7 ธ.ค. 2550
เลขทะเบียน 190938
เลขเรียกหนังสือ



รูป 2.4 โครงสร้างข้อมูลแบบกราฟ

การสำรวจเริ่มต้นที่ A นำโหนดข้างเคียง B C ไว้ในคิว เมื่อสำรวจ A เสร็จ นำข้อมูลในคิว คือ B ออกมาสำรวจ แล้วนำข้อมูลข้างเคียงคือ D E ใส่คิว ตอนนี้คิวจะมี B D E อยู่ แล้วนำ B ออกมาสำรวจทำเช่นนี้เรื่อยๆ จะได้ลำดับการสำรวจข้อมูลคือ (A B C D E F G H) ตามรูป 2.8

โหนดที่สำรวจ	คิว
A	B C
B	C D E
C	D E F
D	E F G H
E	F G H
F	G H
G	H
H	

รูป 2.5 ลำดับการค้นหาแบบกว้างก่อน

ในการค้นหาข้อมูลแบบนี้บนโครงสร้างของกราฟ มีข้อน่าสังเกตคือ โหนดที่เริ่มต้นการสำรวจจะต้องมีการกำหนดมาให้ก่อนว่าโหนดใดที่จะเป็นโหนดเริ่มต้น และข้อสังเกตุอีกประการหนึ่งคือวิธีการค้นหาแบบกว้างก่อนที่ใช้สำหรับโครงสร้างข้อมูลแบบกราฟ สามารถใช้กับโครงสร้างข้อมูลแบบต้นไม้ได้ด้วย