CPE 426 Computer Networks

Chapter 9:

Text Chapter 18&27: Internet Routing Part I:

TOPICS

- Chapter 27: Internet Routing and Routing Protocols
  - 27.1 Introduction
  - 27.2 Static vs Dynamic Routing
  - Extra: Router Configuration in Network
  - 27.3 Static Routing and Default Route
  - Extra: Examples of Static Routing
- BREAK
  - 27.4 Dynamic Routing and Router
  - 27.5 Routing in Global Internet
  - 27.6 Autonomous System Concept
  - 27.7 Two Types of Routing Protocol
  - 27.8 Routes and Data Traffic
  - Extra: Bellman-Ford Algorithm Review
  - Extra: Dijkstra Algorithm Review

Chapter 27:Internet Routing and Routing Protocol

- ลักษณะการส่ง Packet ใน IP Network จะส่งทีละ Hop จาก Network หนึ่ง ไปยังอีก Network หนึ่ง
- Router จะทำหน้าที่ดังกล่าว เนื่องจาก Router เป็นตัวเชื่อมระหว่าง Network
- การส่งจะดูที่ส่วน Prefix ของ IP Address ดังนั้น Router จะต้อง Run IP Protocol คือ ทำงานในระดับ Layer 3
- ที่ Router จะมีตารางชื่อ Routing Table ที่กำหนด IP Address ของ Next Hop

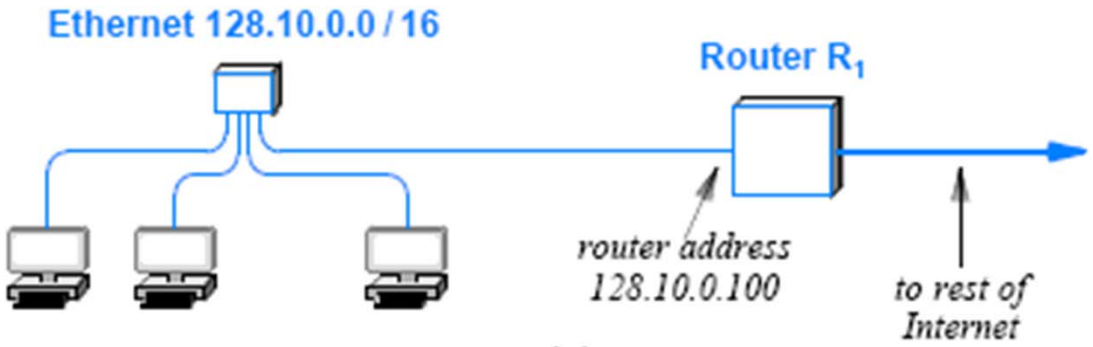
Chapter 27: 27.2 Static vs Dynamic Routing

- **การทำ Routing ทำได้สองแบบ**
  - Static Routing: หมายถึงตาราง Routing Table ของ Router แต่ละตัวจะไม่เปลี่ยน ปกติตารางนี้จะถูกกำหนดจาก Network Administration
    - คือตารางนี้จะได้จากการทำ Configuration ของ Router
  - Dynamic Routing: หมายถึงตาราง Routing Table สามารถเปลี่ยนได้ ตามสภาวะความคับคั่งของ Network ขณะนั้น โดยมันจะมีการ Update ตลอดเวลา
    - ตารางนี้ได้จากการกำหนดให้ Router ทำการ Run Routing Protocol
      - ในแต่ละ Routing Protocol ตัว Router จะทำการแลกเปลี่ยนข้อมูลกันเอง โดยผู้ดูแล Network ไม่ต้องมาเกี่ยวข้อง จากนั้น Router แต่ละตัวจะสร้างตาราง Routing Table จากข้อมูลที่มันรวบรวมได้ เมื่อข้อมูลที่ได้รับเปลี่ยน เนื่องจาก Network เปลี่ยน มันจะคำนวณตารางใหม่ และปรับให้เข้ากับสภาพของ Network ที่เปลี่ยนไป

Chapter 27: 27.3 Static Routing in Host and Default Route

- ข้อดีของ Static Routing คือง่าย และไม่ต้องใช้ Routing Software(Router จะทำงานน้อยลง) นอกจากนี้จะไม่ใช้ Resource ของ Network เพราะ Router ไม่ต้องแลกเปลี่ยนข้อมูลกัน
- แต่ข้อเสียคือ ตารางเปลี่ยนไม่ได้ ถ้ามี Link Down หรือ Router Down เส้นทางนั้นจะใช้ไม่ได้ ทำให้การส่งข้อมูลที่กำหนดเส้นทางนั้นหยุดชะงัก
  - นอกจากนี้แล้ว Static Routing จะจำกัดที่ Network ขนาดเล็ก เช่น ระหว่าง LAN ขององค์กร การเชื่อมต่อกับ Internet จะไม่ใช้ Static Routing
  - หรือใช้กำหนดสำหรับ Host ที่เชื่อมต่อกับ Network ที่มีทางออกผ่าน Router ตัวเดียว(คือค่า Gateway)
- การใช้ Static Route ควรมีการกำหนด Default Route เสมอ
- การทำ Static Route ต้องตรวจสอบให้ดีว่าจะไม่มี Route Loop

Chapter 27: 27.3 Static Routing in Host and Default Route



(a)

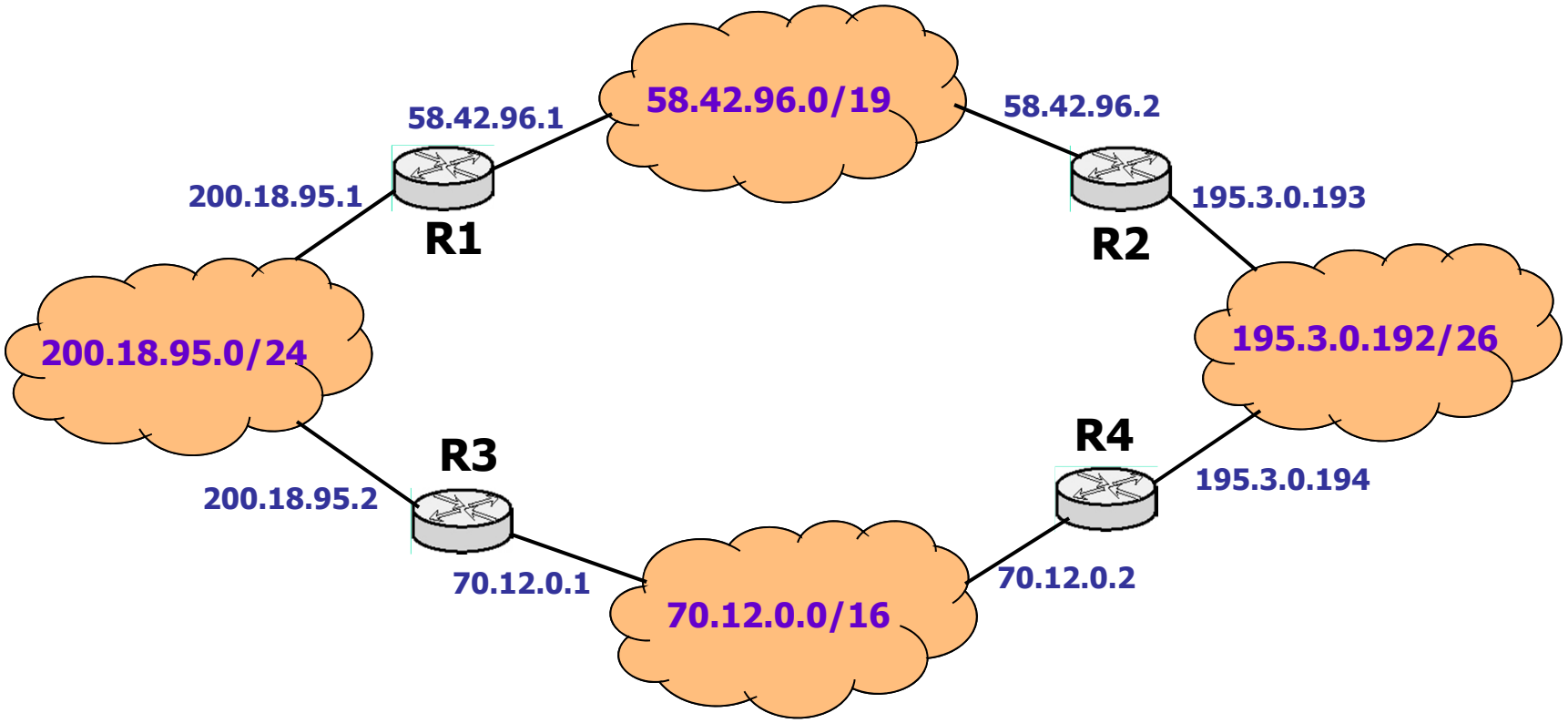
Net	Mask	Next hop
128.10.0.0	255.255.0.0	direct
default	0.0.0.0	128.10.0.100

(b)

Figure 27.1 (a) A typical connection to the Internet, and (b) the static forwarding table used in each host.

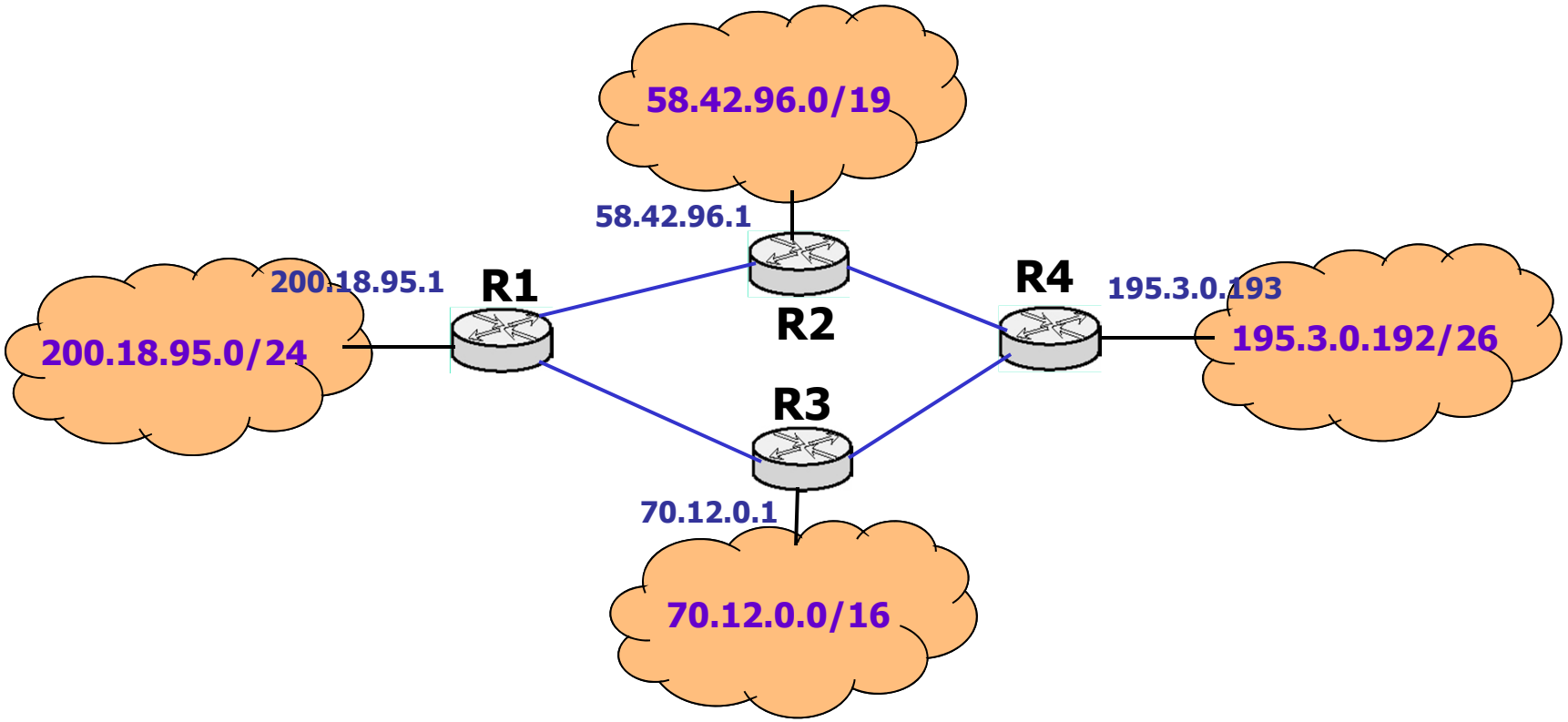
การกำหนด Interface สำหรับ Router

- แต่ละ Interface ของ Router จะต้องมี IP Address ที่ตรงกับแต่ละ Network ที่ Interface เชื่อมต่อ (Prefix เหมือนกัน แต่ Suffix ต่างกัน)



การกำหนด Interface สำหรับ Router

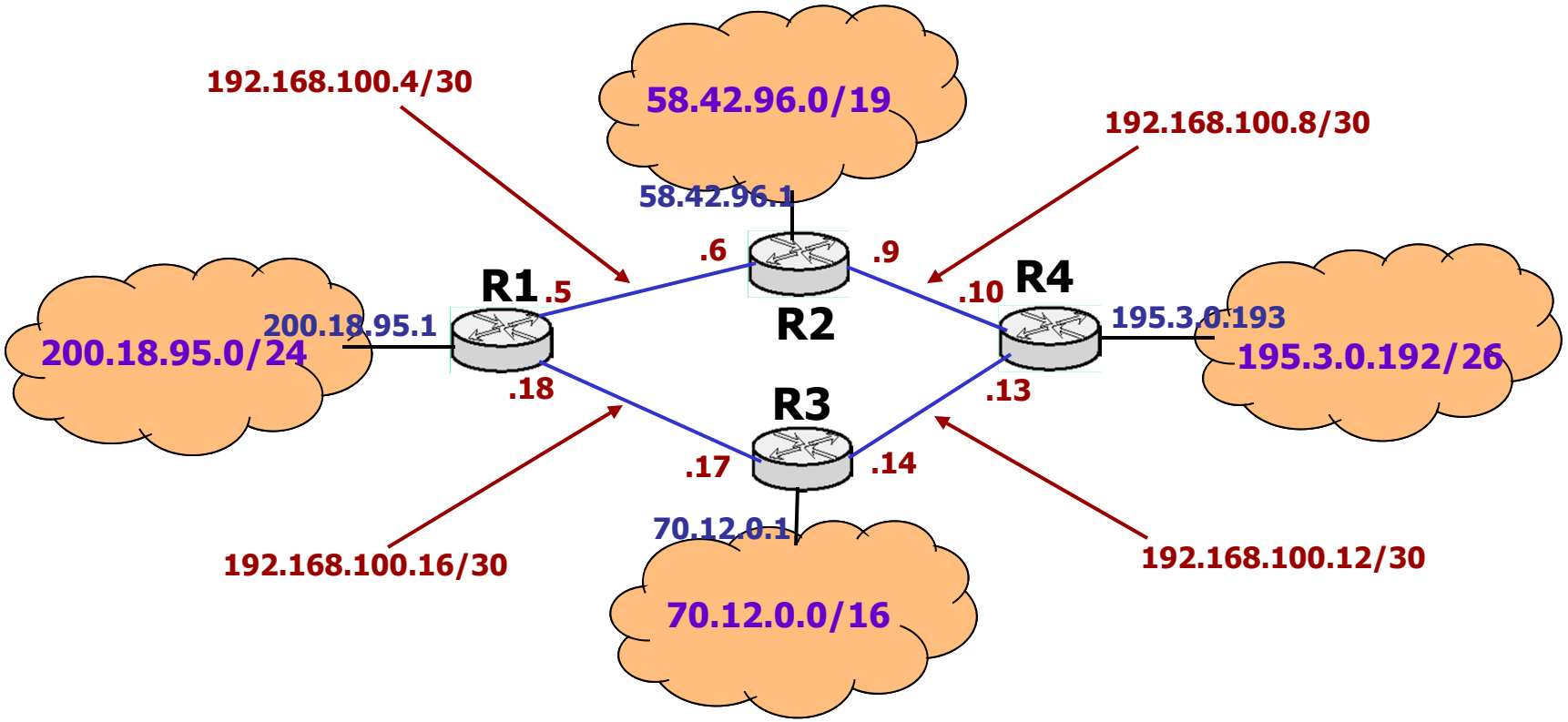
- สายที่เชื่อมต่อโดยตรงระหว่าง Router จะต้องถือเป็นหนึ่ง Network เช่นกัน แต่ Network นี้เป็นแค่ทางผ่านของข้อมูล ไม่จำเป็นต้องกำหนดใน Routing Table



การกำหนด Interface สำหรับ Router

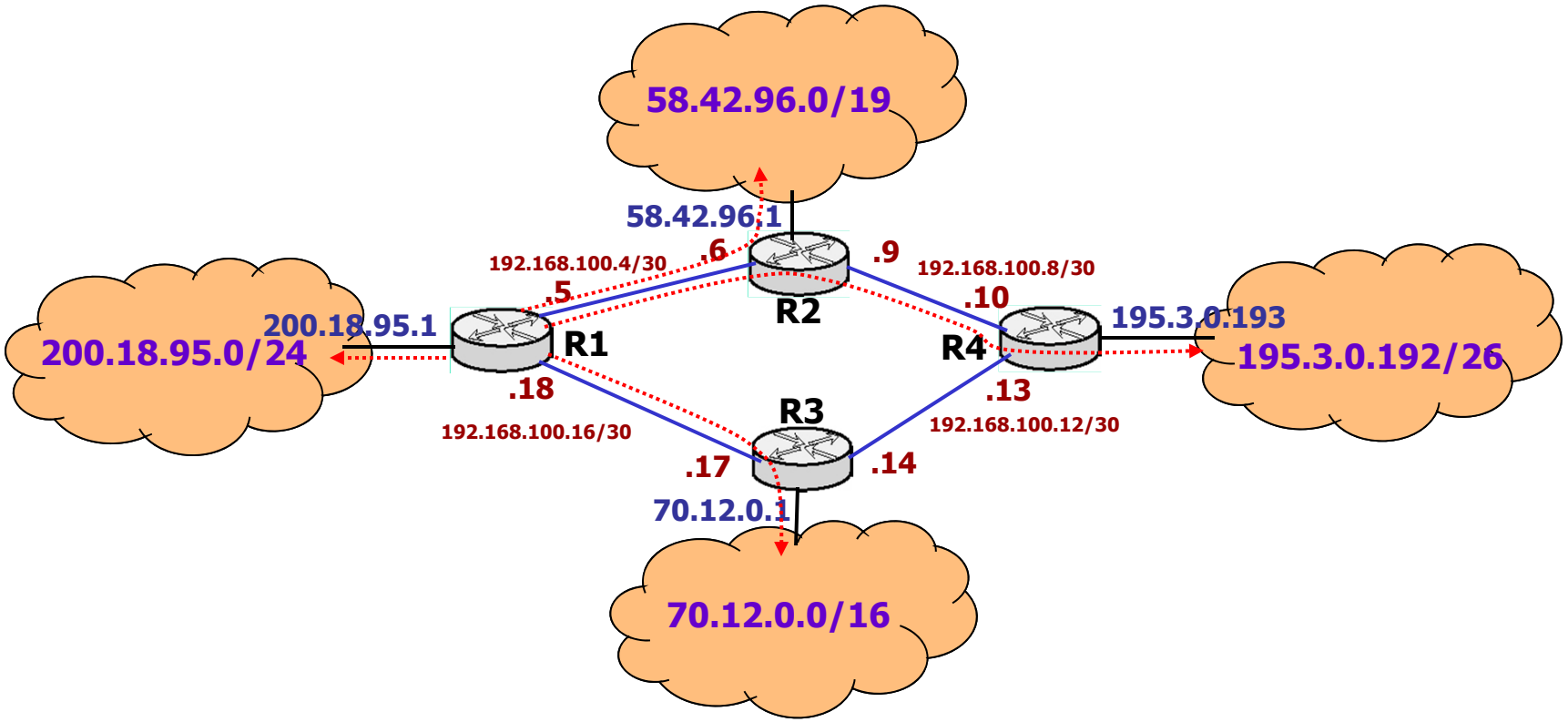
- สายระหว่าง Router สามารถ Subnet จาก Private IP ได้ และต้องการเพียงสอง Host Address ซึ่งปกติจะใช้ /30 ก็พอ เช่น Sub จาก 192.168.100.0/24 จะได้ 64 Subnet

Subnet 192.168.100.4/30 : Host Range 192.168.100.5-192.168.100.6; Broadcast 192.168.100.7



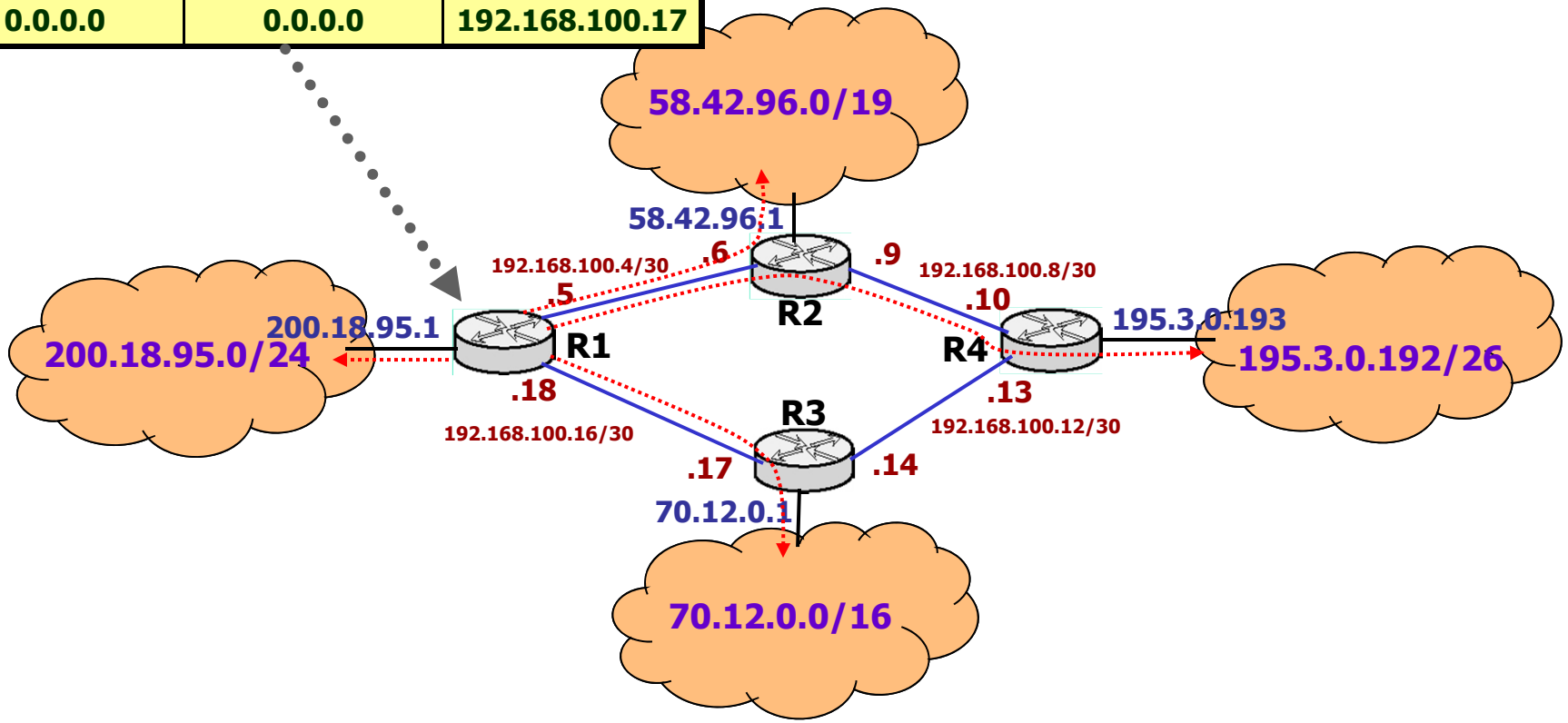
การกำหนด Interface สำหรับ Router

- การกำหนด Static Routing Table ให้กับ Router แต่ละตัวทำได้จากการกำหนดเส้นทางส่งข้อมูลจาก Router ไปยังทุกๆ Network (Network ที่เป็น Transit ไม่ต้องกำหนด เพราะไม่มี Host ปลายทาง) จากนั้นกำหนดเส้นทางหนึ่งให้เป็น Default Route เช่น R1



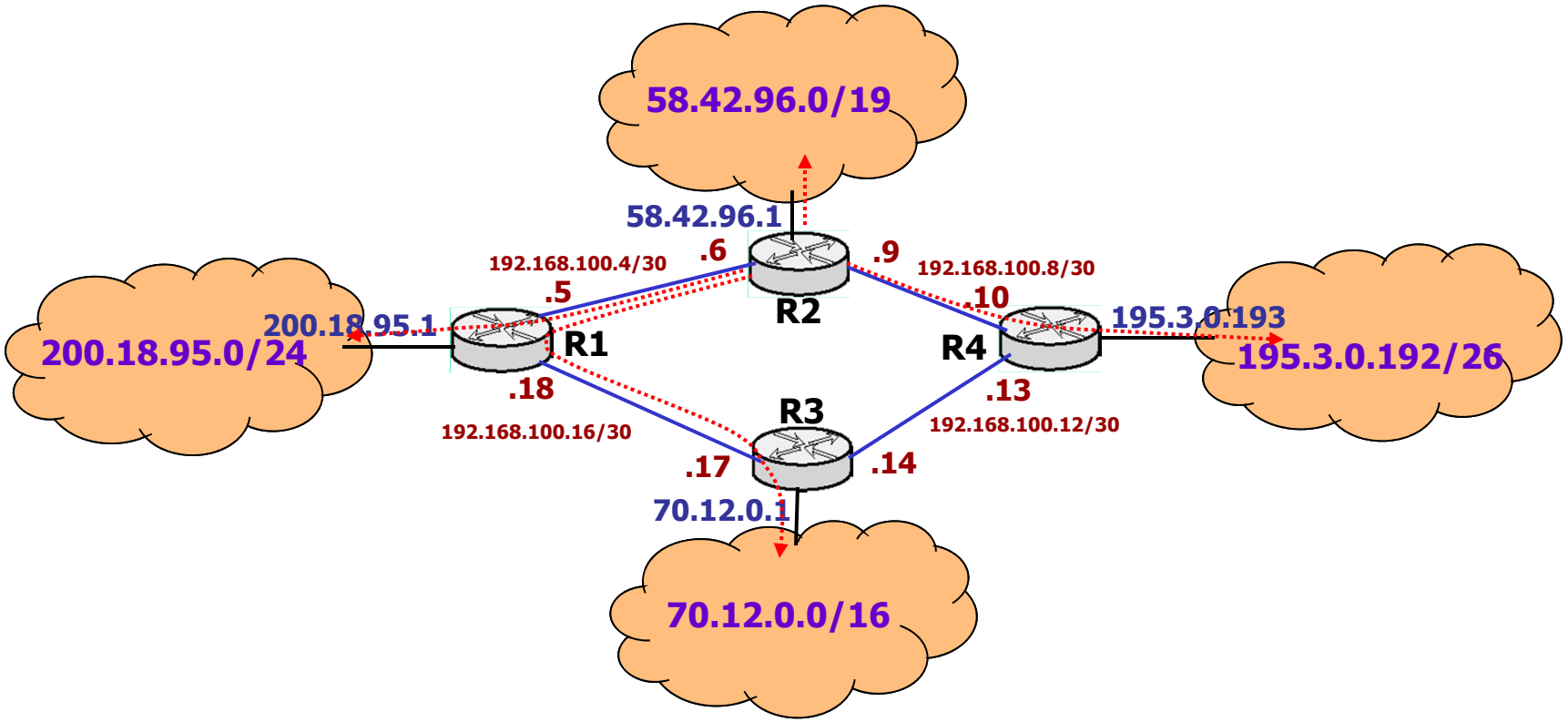
การกำหนด Interface สำหรับ Router

NW	Mask	Next Hop
200.18.95.0	255.255.255.0	Direct
70.12.0.0	255.255.0.0	192.168.100.17
58.42.96.0	255.255.224.0	192.168.100.6
195.3.0.192	255.255.255.192	192.168.100.6
0.0.0.0	0.0.0.0	192.168.100.17



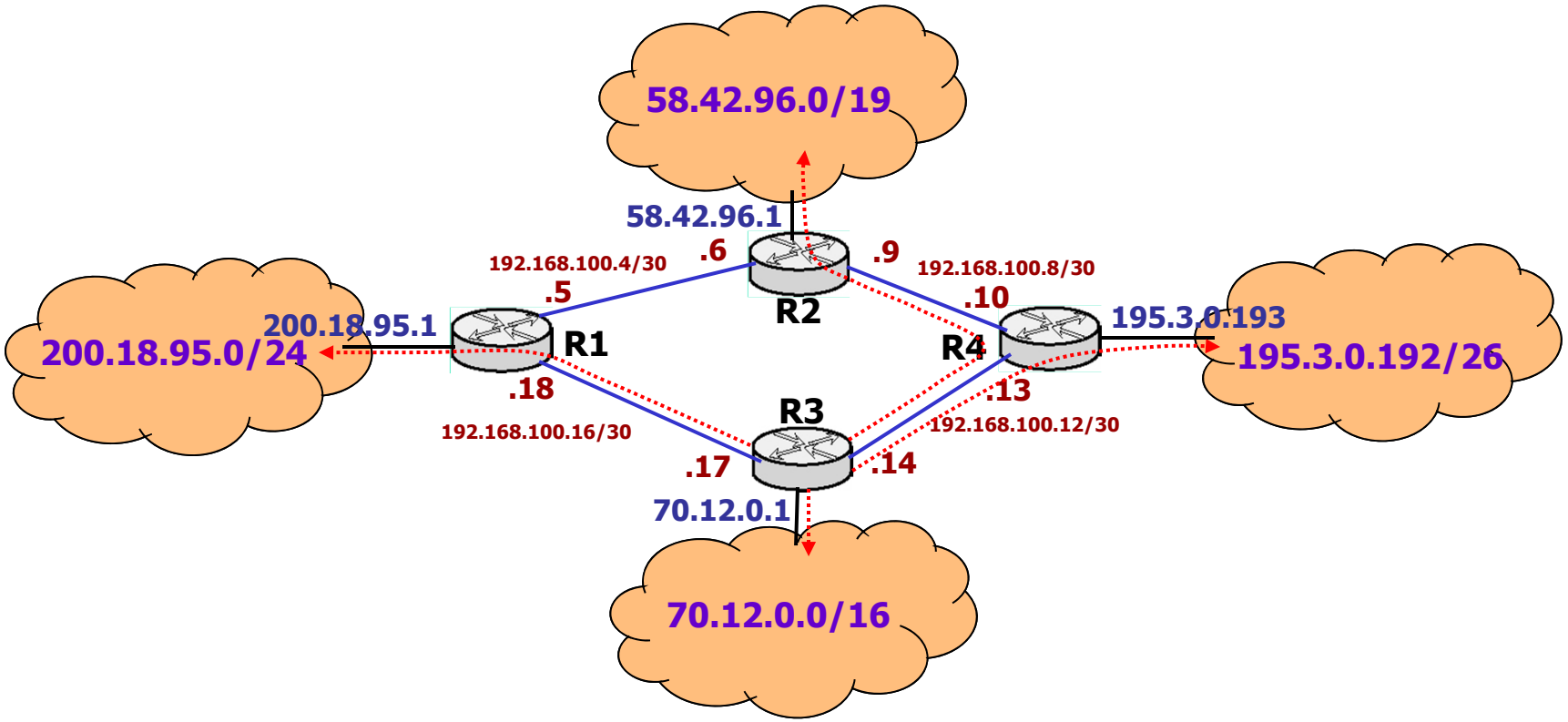
การกำหนด Interface สำหรับ Router

R2 Routing Table



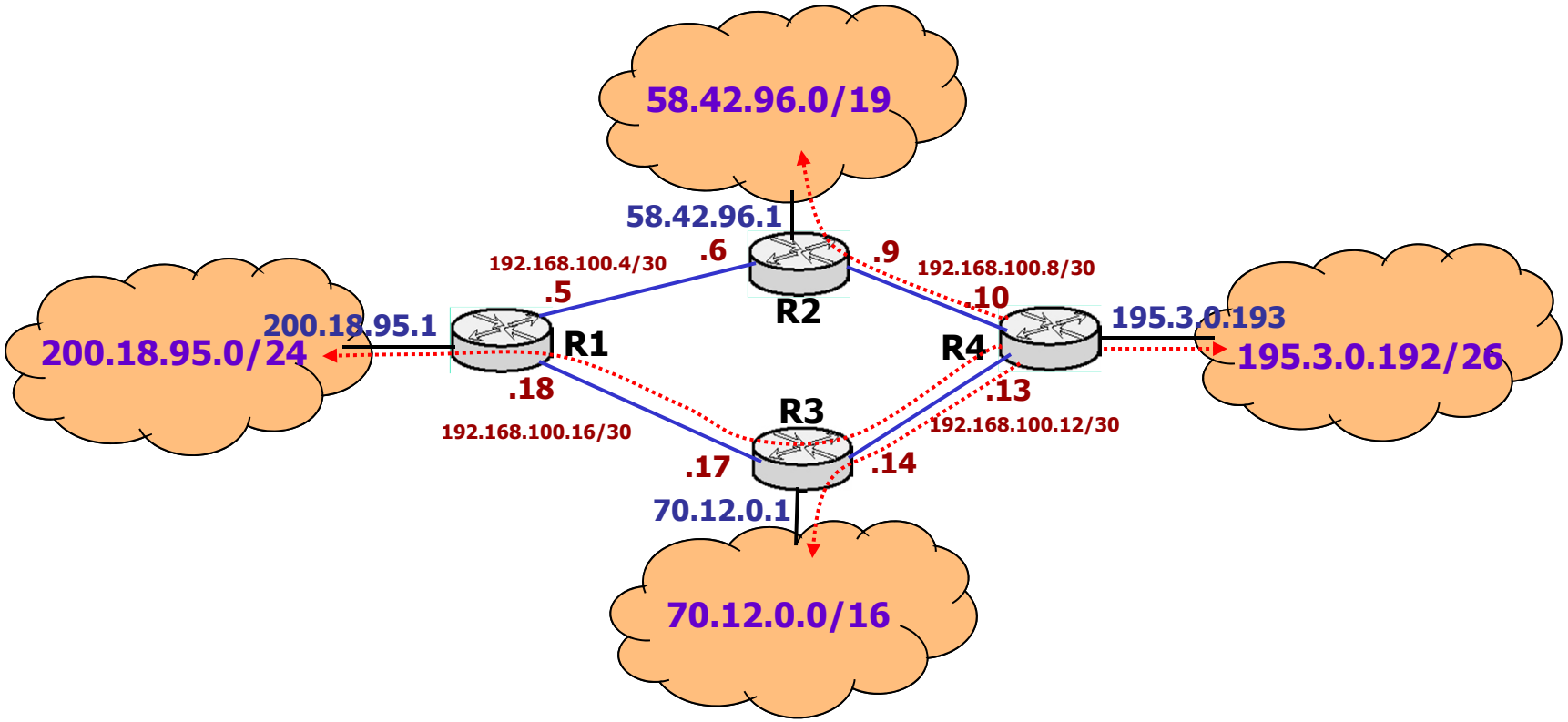
การกำหนด Interface สำหรับ Router

R3 Routing Table



การกำหนด Interface สำหรับ Router

R4 Routing Table



Chapter 27: 27.4 Dynamic Routing and Router

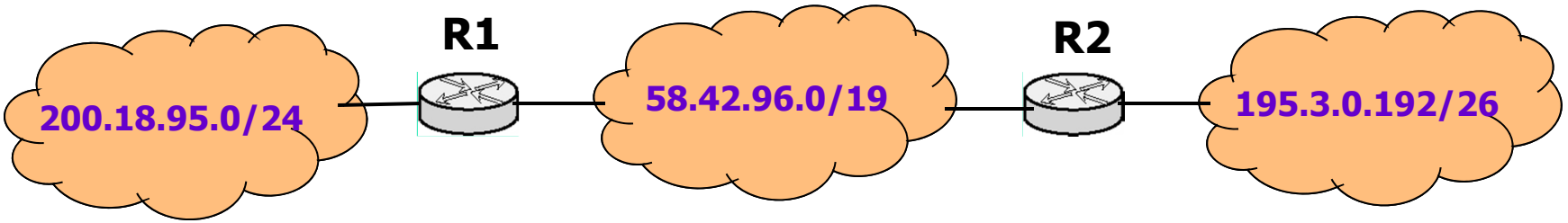
- ปกติ Router ใน Internet จะใช้ Dynamic Routing
  - เพื่อให้มีการแลกเปลี่ยน Routing Information ระหว่างกัน
- Static Routing อาจจะถูกใช้ในกรณีที่ Customer เชื่อมต่อกับ ISP ผ่าน Router ซึ่งในกรณีนี้มีทางออก Internet เพียงทางเดียว และไม่จำเป็นต้องใช้ Dynamic Routing
- หรือ Static Routing อาจจะใช้ภายในองค์กรเพื่อเชื่อมต่อ LAN หลายๆวงภายในอาคาร

Chapter 27: 27.4 Dynamic Routing and Router

- Router จะรู้จัก Network ที่เป็น Direct Connect เท่านั้น
- การที่ Router จะรู้จัก Network อื่น มันจะต้องเรียนรู้มาจาก Router ตัวอื่นที่ต่อโดยตรงกับ Network นั้น
- ใน Static Routing จะไม่มีวิธีที่ Router จะเรียนรู้ได้
- ดังนั้นการเรียนรู้ต้องมีการกำหนดจาก Software ใน Routing Protocol ใน Dynamic Routing
- ด้วยการเรียนรู้นี้เอง ทำให้ Router สามารถปรับตารางของตนได้อย่างเหมาะสม ตามสภาพ Network เป็นผลให้ตาราง Routing เป็น Dynamic

Chapter 27: 27.4 Dynamic Routing and Router

- จากรูป R1 รู้จัก 200.18.95.0/24 และ 58.42.96.0/19 แต่ไม่รู้จัก 195.3.0.192/26
- R2 รู้จัก 58.42.96.0/19 และ 195.3.0.192/26 แต่ไม่รู้จัก 200.18.95.0/24
- R1 และ R2 แลกเปลี่ยนข้อมูลกันผ่าน 58.42.96.0/19 ทำให้ Router แต่ละตัวรู้จัก Network อื่นๆ
- ถ้า 195.3.0.192 เกิดล่ม R2 จะรู้ และสามารถบอกต่อไปยัง R1 ได้ ว่า 195.3.0.192/26 Unreachable
- ถ้า R2 ล่ม ทำให้ R1 ไม่สามารถติดต่อได้ ดังนั้น R1 จะ Mark ว่า 195.3.0.192/26 เป็น Unreachable เช่นกัน



Chapter 27: 27.5 Routing in Global Internet

- **Internet ประกอบด้วย Router มากมาย ถ้าจะให้ Router ทุกตัวแลกเปลี่ยนข้อมูลกับ Router ทุกตัว จะทำให้ Routing Traffic มีมหาศาล**
- **เพื่อจำกัดจำนวน Traffic ใน Internet จะใช้การทำ Routing แบบเป็นลำดับชั้น (Hierarchy) โดยมีการแบ่งกลุ่มของ Router และมีการแลกเปลี่ยนข้อมูลภายในกลุ่ม จากนั้นจะมี Router ที่เป็นตัวแทนของกลุ่มทำการแลกเปลี่ยนข้อมูลกับภายนอก**
- **Routing Protocol ภายในกลุ่ม จะแตกต่างจาก Routing Protocol ระหว่างกลุ่ม**
  - ขนาดของกลุ่มจะไม่จำกัด ขึ้นอยู่กับขนาดขององค์กร
  - แต่ละองค์กรที่เป็นเจ้าของกลุ่ม มีสิทธิจะเลือก Routing Protocol อย่างไรก็ได้ ที่อยู่ภายในกลุ่ม
  - แต่ Routing Protocol ที่ใช้แลกเปลี่ยนข้อมูลระหว่างกลุ่มจะต้องเป็น Protocol เดียวกัน

Chapter 27: 27.6 Autonomous System Concept

- **แต่ละกลุ่มของ Router ที่ดูแลจัดการโดยองค์กรเดียว เราเรียกว่า Autonomous System (AS)**
  - อาจจะเป็นหนึ่งองค์กร หรือ หนึ่ง ISP
  - องค์กรหนึ่งอาจจะแบ่งกลุ่มของ Router เป็นหลาย AS ก็ได้
  - Router ภายใน AS จะมีการแลกเปลี่ยน Routing Information กัน

Chapter 27: 27.7 Two Type of Internet Routing Protocol

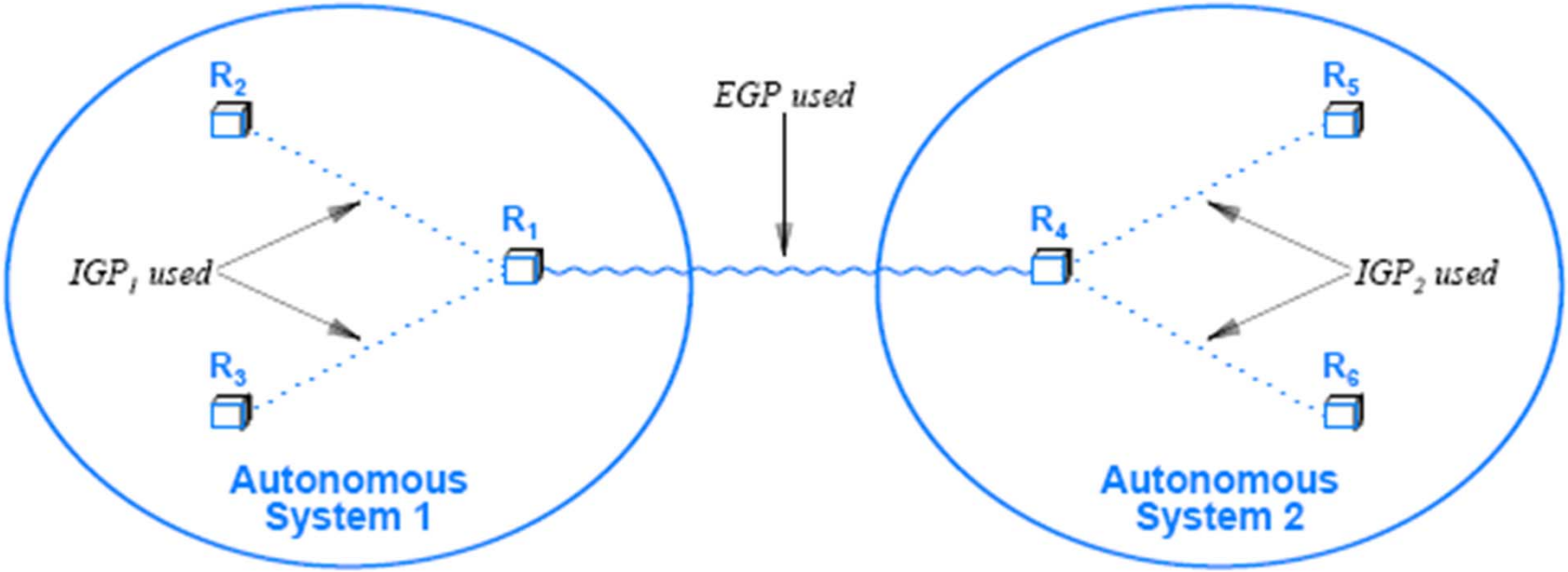
- **27.7.1 Interior Gateway Protocols(IGP)**
  - Router ภายใน AS จะใช้ Interior Gateway Protocol (IGP) ในการแลกเปลี่ยน Routing Information
  - แต่ละ AS มีสิทธิ์ที่จะเลือก IGP อันใดก็ได้
    - RIP
    - OSPF
    - IGRP (Cisco)
    - EIGRP (Cisco)

Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.2 Exterior Gateway Protocols(EGP)**
  - Router ในแต่ละ AS จะใช้ EGP ในการแลกเปลี่ยน Routing Information กับ Router ในอีก AS หนึ่ง
  - EGP ปกติจะซับซ้อนกว่า IGP แต่ว่าการใช้งานจะยืดหยุ่นกว่า และมี Overhead ต่ำกว่า
  - EGP จะสรุป Routing Information ในแต่ละ AS ก่อนที่จะส่งไปให้อีก AS หนึ่ง
    - การส่ง Routing Information ออกนอก AS สามารถกำหนดว่าข้อมูลใดให้ส่ง หรือไม่ให้ส่งได้

Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.3 ตัวอย่างการใช้งาน IGP และ EGP**
  - Router R1 จะ Run ทั้ง IGP1 และ EGP
  - Router R4 จะ Run ทั้ง IGP2 และ EGP

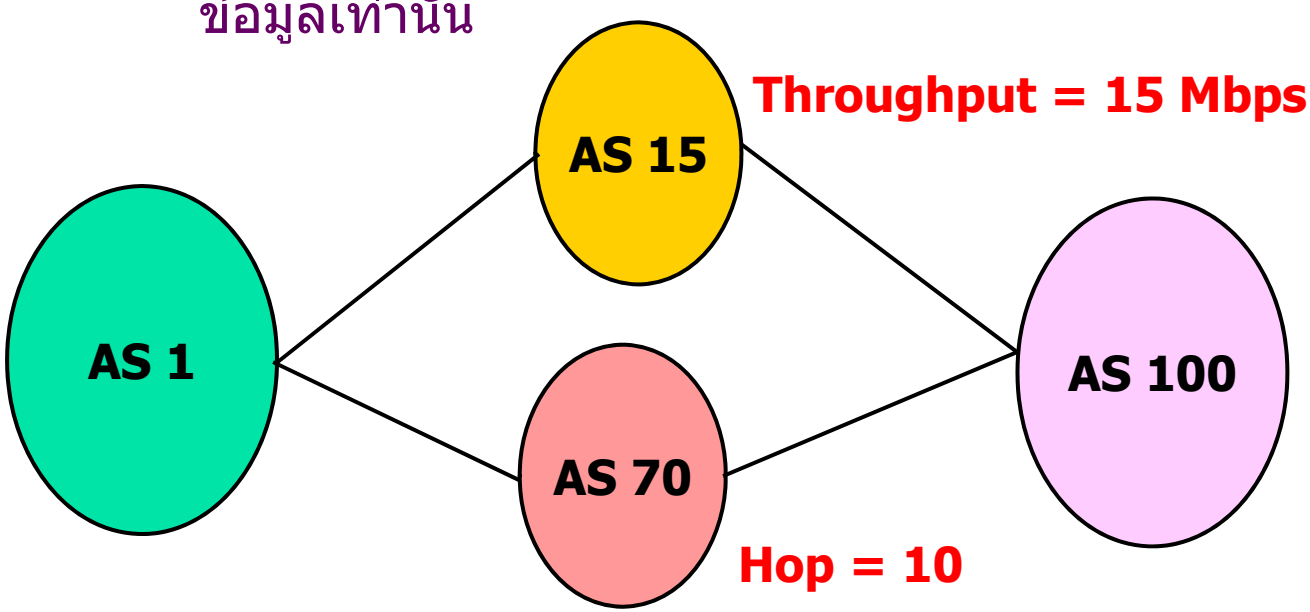


Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.4 Optimum Routes, Routing Metrics and IGP**
  - ปกติเส้นทางส่งข้อมูลใน Internet จะมีหลายเส้นทาง
  - Router จะเลือกเส้นทางที่ดีที่สุด (Optimal Routes)
    - Remote Application อาจจะเป็นเส้นทางที่ Delay ต่ำสุด
    - Web Application อาจเป็นเส้นทางที่ Throughput สูงสุด
    - Webcast หรือ Real-Time อาจเป็นเส้นทางที่ Jitter ต่ำสุด
  - เราใช้คำว่า Routing Metric เป็นตัววัดราคาของการส่งในแต่ละเส้นทาง
    - อาจจะวัดจาก Delay, Throughput หรือ Jitter หรือผสมกัน
  - แต่ปกติใน Internet จะใช้ Metric สองตัวร่วมกัน
    - Hop Count (จำนวน Network หรือ Router ที่ผ่าน)
    - Administrative Cost (กำหนดเองจาก Administrator) เพื่อควบคุมให้เส้นทางส่งข้อมูลเป็นไปตามต้องการ
      - เช่นกำหนดเส้นทาง 4 Hop ให้มี Administrative Cost ต่ำกว่าเส้นทางสอง Hop เพื่อบังคับให้ข้อมูลส่งไปทางนี้

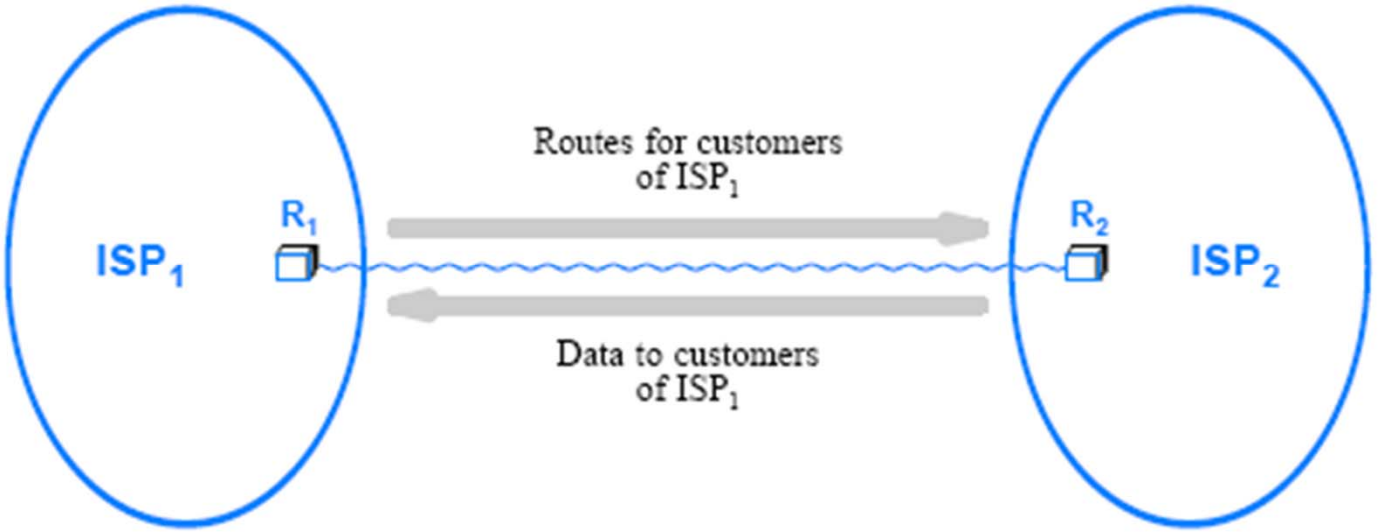
Chapter 27: 27.7 Two Type of Internet Routing Protocol

- **27.7.4 Optimum Routes, Routing Metrics and IGP**
  - การหาเส้นทางของ IGP จะใช้ Routing Metric
  - EGP จะไม่ใช้
    - เนื่องจากแต่ละ AS ใช้ IGP ต่างกัน และใช้ Metric ต่างกัน ไม่สามารถเปรียบเทียบได้
  - ดังนั้น EGP จะไม่พยายามหา Optimal Path มันเพียงหาเส้นทางส่งข้อมูลเท่านั้น



Chapter 27: 27.8 Routes and Data Traffic

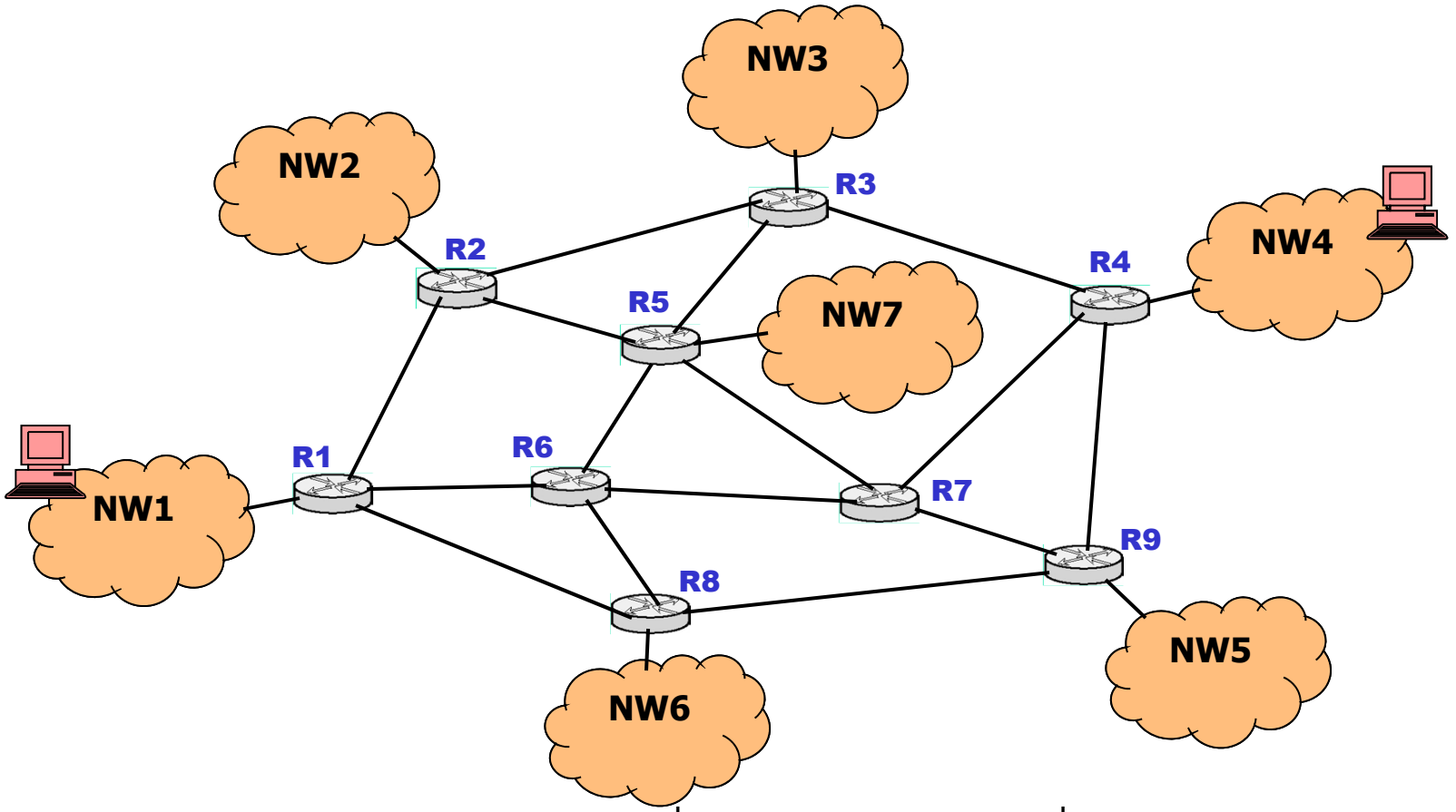
- **Data Traffic จะมีทิศทางการไหลสวนทางกับ Routing Traffic**



Least Cost Path Algorithms

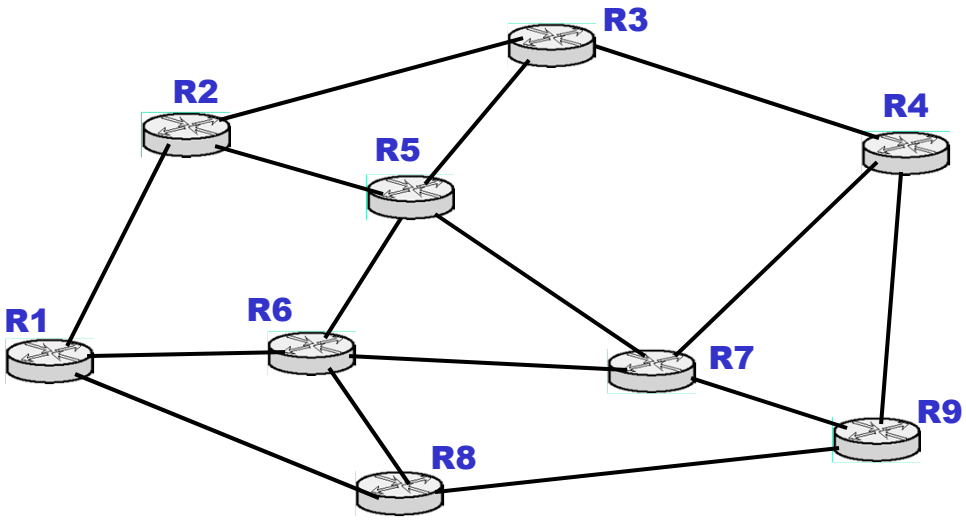
- ดูรายละเอียดบทที่ 18 18.12-18.13
- ดูจาก Course Notes วิชา CPE 326
- ดูจาก Course Notes วิชา CPE 231
  - เรื่อง Graph
- มีสอง Algorithm ที่ให้คำตอบเหมือนกัน แต่ใช้วิธีคำนวณต่างกัน
  - ดังนั้นจะใช้ข้อมูลจาก Routing Information ต่างกัน
  - กำเนิดเป็นวิธีการ Routing สองแบบ
    - Distance Vector Routing จะใช้ Bellman-Ford Algorithm
      - ในการนี้ Router จะแลกเปลี่ยนตาราง Routing Table เฉพาะกับเพื่อนบ้านเท่านั้น ข้อมูลจะ Propagate ทีละ Router เมื่อถึงเวลาแลกตาราง เช่น RIP(Routing Information Protocol)
    - Link-State Routing จะใช้ Dijkstra Algorithm
      - ในการนี้ Router จะส่ง Link State ของตนเอง (เฉพาะ Direct Connect) ไปให้กับทุกๆ Router แต่ละ Router จะรวบรวม Link State Database และสร้าง Topology ของทั้ง Network จากนั้นมันจะสร้าง Shortest Path First Tree โดยตัวมันเป็น Root ไปยัง Router ทุกๆตัว

Least Cost Path Algorithms



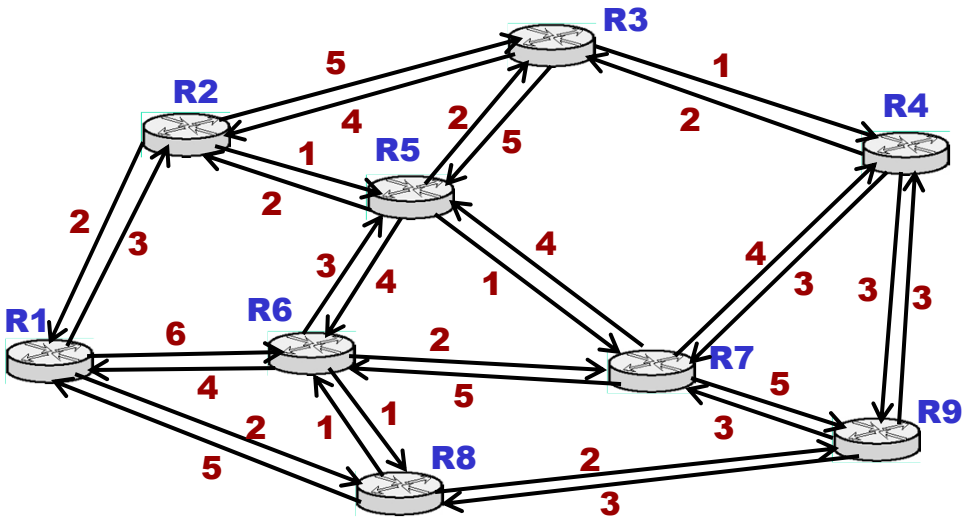
1. การส่งข้อมูลจาก Host หนึ่ง ไปยังอีก Host หนึ่ง กระทำผ่าน Router ถ้าอยู่คนละ Network การหาเส้นทางคือหาเส้นทางจาก Router หนึ่งไปยังอีก Router หนึ่ง ที่ Network นั้นเชื่อมต่ออยู่

Least Cost Path Algorithms



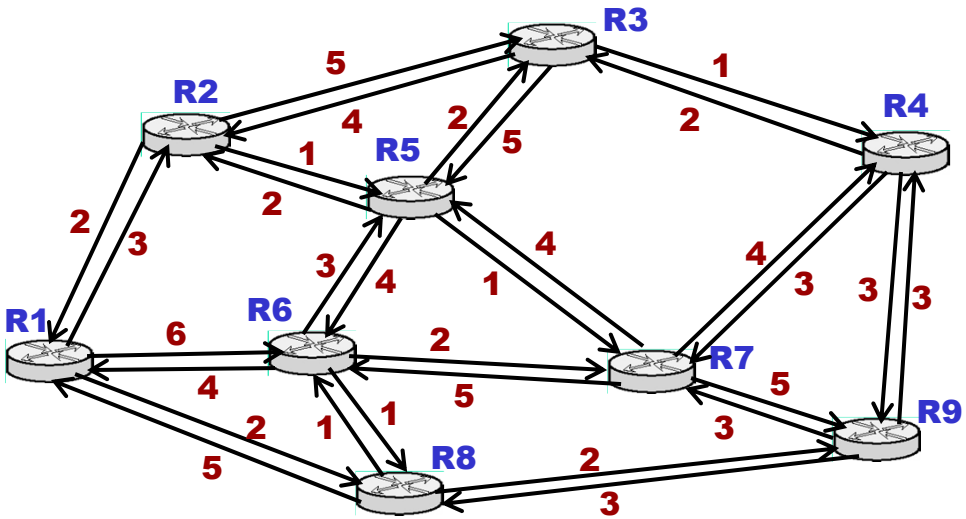
2. Cost ที่ส่งระหว่าง Router ไปและกลับไม่จำเป็นต้องเท่ากัน เพราะขึ้นอยู่กับ Queue ที่ Interface ของ Router ต้นทาง ไม่ใช่ปลายทาง เราแสดงค่า Cost ระหว่าง Router ที่มีเส้นเชื่อมต่อ โดยใช้ Cost Matrix

Least Cost Path Algorithms



2. Cost ที่ส่งระหว่าง Router ไปและกลับไม่จำเป็นต้องเท่ากัน เพราะขึ้นอยู่กับ Queue ที่ Interface ของ Router ต้นทาง ไม่ใช่ปลายทาง เราแสดงค่า Cost ระหว่าง Router ที่มีเส้นเชื่อมต่อ โดยใช้ Cost Matrix (W)

Least Cost Path Algorithms



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

2. Cost ที่ส่งระหว่าง Router ไปและกลับไม่จำเป็นต้องเท่ากัน เพราะขึ้นอยู่กับ Queue ที่ Interface ของ Router ต้นทาง ไม่ใช่ปลายทาง เราแสดงค่า Cost ระหว่าง Router ที่มีเส้นเชื่อมต่อ โดยใช้ Cost Matrix (W)

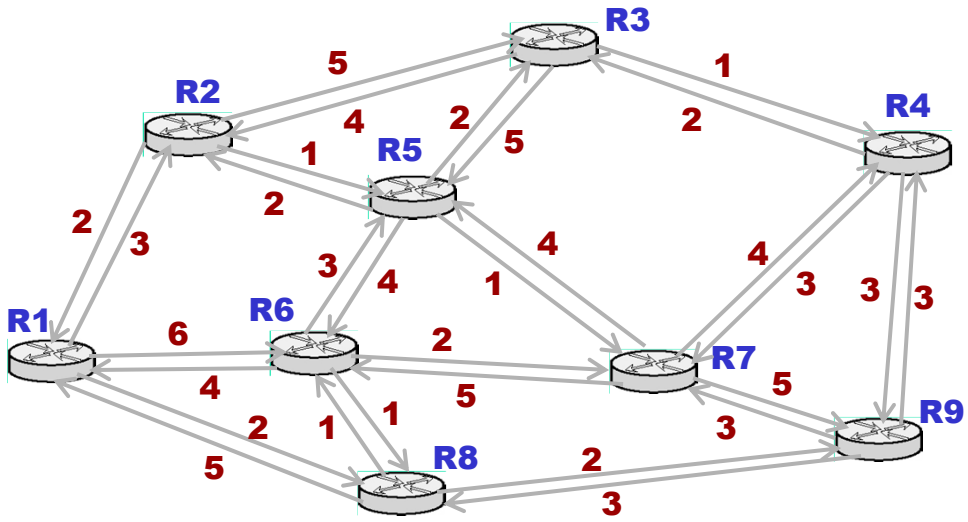
Bellman-Ford Algorithm Definitions

- **Find shortest paths from given node subject to constraint that paths contain at most one link**
- **Find the shortest paths with a constraint of paths of at most two links**
- **And so on**
- **s = source node**
- **w(i, j) = link cost from node i to node j**
  - w(i, i) = 0
  - w(i, j) = ∞ if the two nodes are not directly connected
  - w(i, j) ≥ 0 if the two nodes are directly connected
- **h = maximum number of links in path at current stage of the algorithm**
- **$L_h(n)$ = cost of least-cost path from s to n under constraint of no more than h links**

Bellman-Ford Algorithm Method

- **Step 1 [Initialization]**
  - $L_0(n) = \infty$, for all $n \neq s$
  - $L_h(s) = 0$, for all h
- **Step 2 [Update]**
- **For each successive $h \geq 0$**
  - For each $n \neq s$, compute
  - $L_{h+1}(n) = \min_j [L_h(j) + w(j,n)]$; $\forall$j
- **Connect n with predecessor node j that achieves minimum**
- **Eliminate any connection of n with different predecessor node formed during an earlier iteration**
- **Path from s to n terminates with link from j to n**

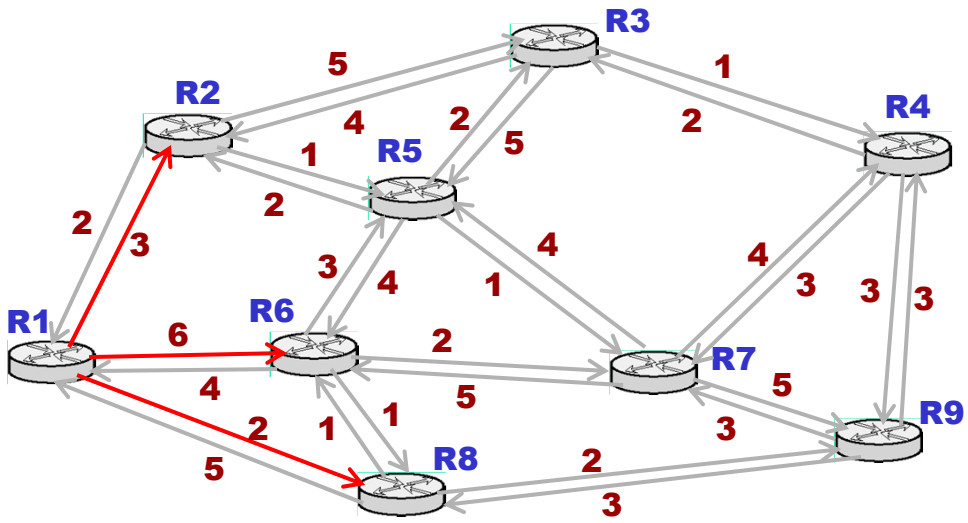
Example: Bellman Ford จาก Node 1 (h=0: Initialization: s=1)



Cost R1 ไปทุกๆ Node = Infinity
$L_0(n)=\infty$; n = 2,..,9: $L_0(1)=0$

	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Example: Bellman Ford จาก Node 1 (h=1: Initialization: s=1)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไปทุกๆ Node = Infinity

$L_0(n)=\infty$; n = 2,..,9: $L_0(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_1(2)=\min_j[L_0(j)+w(j,2)]$; j = 1,..,9

$=\min[L_0(1)+w(1,2),L_0(2)+w(2,2),L_0(3)+w(3,2),\ldots,L_0(9)+w(9,2)]$

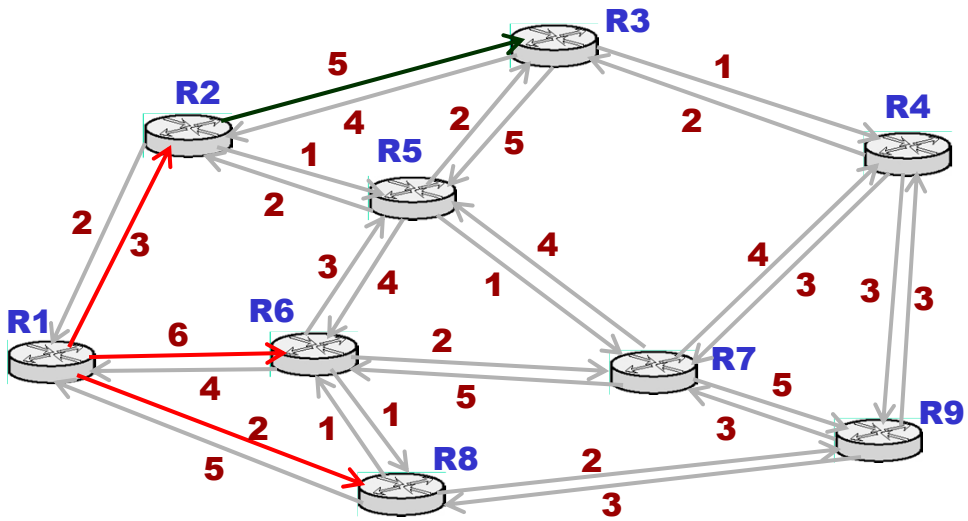
=min j = 1, path = 1-2, cost = 3

$L_1(3)=\min j[L_0(j)+w(j,3)]$; j = 1,..,9

$=\min[L_0(1)+w(1,3),L_0(2)+w(2,3),L_0(3)+w(3,3),\ldots,L_0(9)+w(9,3)]$

=all infinity

Example: Bellman Ford จาก Node 1 (h=2: n=2,3)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไป Node 3,4,5,7,9= Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_2(2)=\min_j[L_1(j)+w(j,2)]$; j = 1,..,9

$=\min[L_1(1)+w(1,2),L_1(2)+w(2,2),L_1(3)+w(3,2),\ldots,L_1(9)+w(9,2)]$

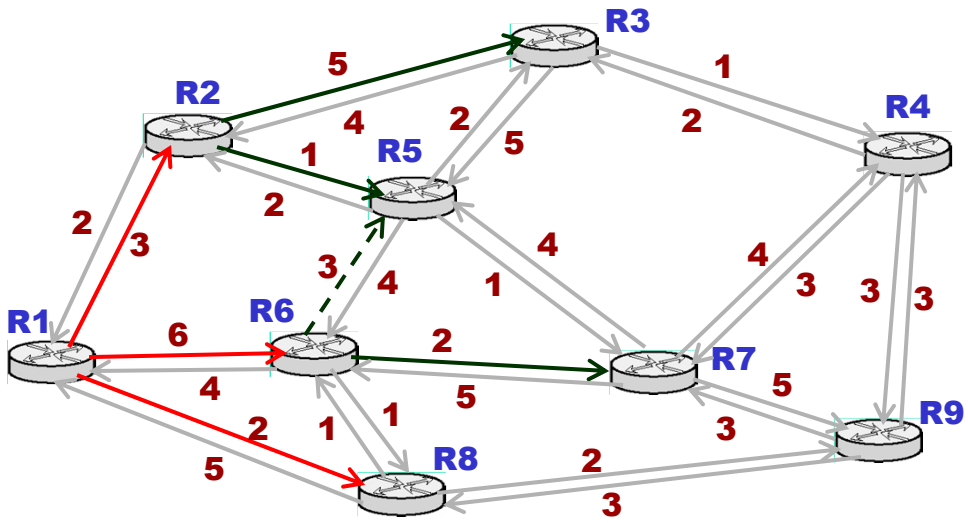
=min j = 1, path = 1-2, cost = 3

$L_2(3)=\min j[L_1(j)+w(j,3)]$; j = 1,..,9

$=\min[L_1(1)+w(1,3),L_1(2)+w(2,3),L_1(3)+w(3,3),\ldots,L_1(9)+w(9,3)]$

=min j=2, path = 1-2 plus 2-3 = 1-2-3, cost = 3+5=8

Example: Bellman Ford จาก Node 1 (h=2: n=5,7)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไป Node 3,4,5,7,9= Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_2(5)=\min_j[L_1(j)+w(j,5)]$; j = 1,..,9

$=\min[L_1(1)+w(1,5),L_1(2)+w(2,5),L_1(3)+w(3,5),\ldots,L_1(6)+w(6,5),..,L_1(9)+w(9,5)]$

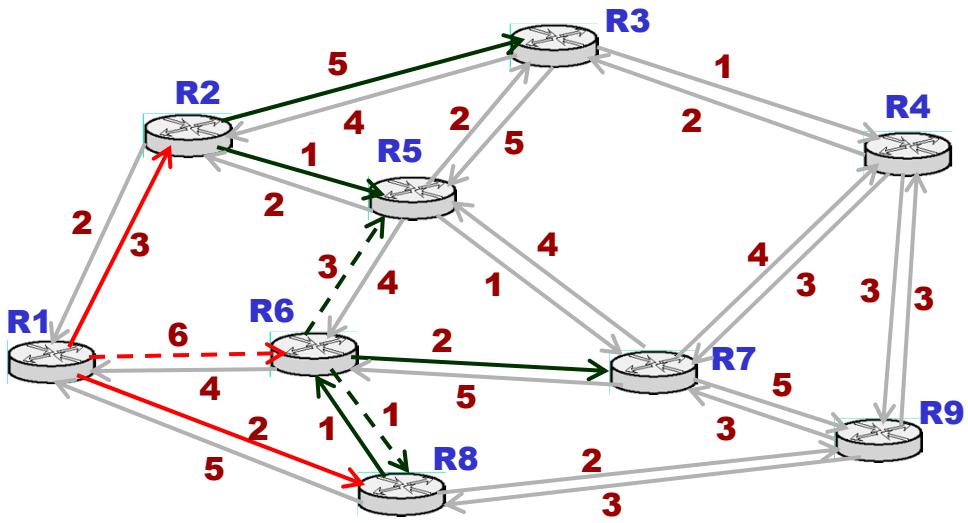
$=\min(3+1,6+3)=4$ (j=2, path = 1-2-5, cost = 4)

$L_2(7)=\min j[L_1(j)+w(j,7)]$; j = 1,..,9

$=\min[L_1(1)+w(1,7),L_1(2)+w(2,7),\ldots,L_1(6)+w(6,7),\ldots,L_1(9)+w(9,7)]$

=min j=6, path = 1-6 plus 6-7 = 1-6-7, cost = 6+2=8

Example: Bellman Ford จาก Node 1 (h=2: n=6)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไปทุกๆ Node = Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

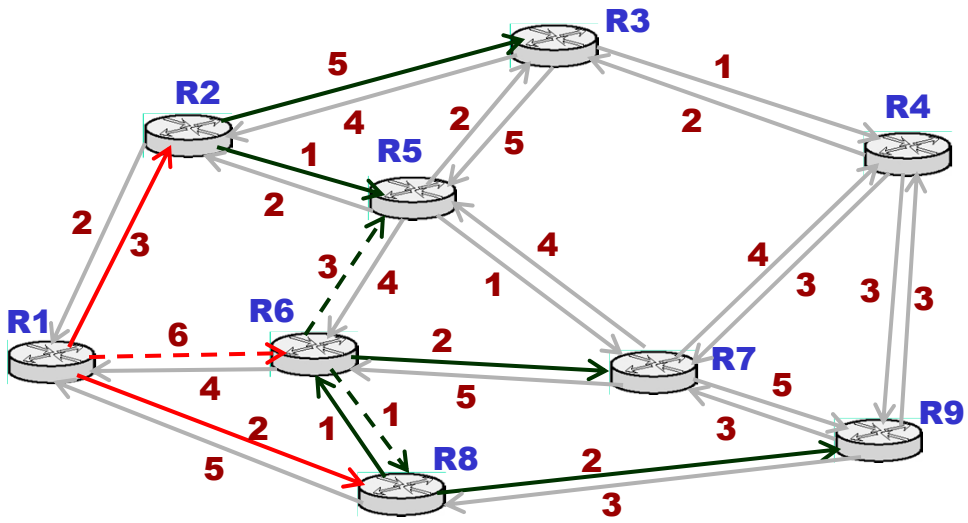
คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_2(6)=\min_j[L_1(j)+w(j,6)]$; j = 1,..,9

$=\min[L_1(1)+w(1,6),\ldots, L_1(8)+w(8,6),..,L_1(9)+w(9,6)]$

= j=8; Path 1-8+8-6 ถูกกว่า Path 1-6

Example: Bellman Ford จาก Node 1 (h=2: n=8)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไปทุกๆ Node = Infinity

$L_1(2)=3$; $L_1(6)=6$; $L_1(8)=2$; : $L_1(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

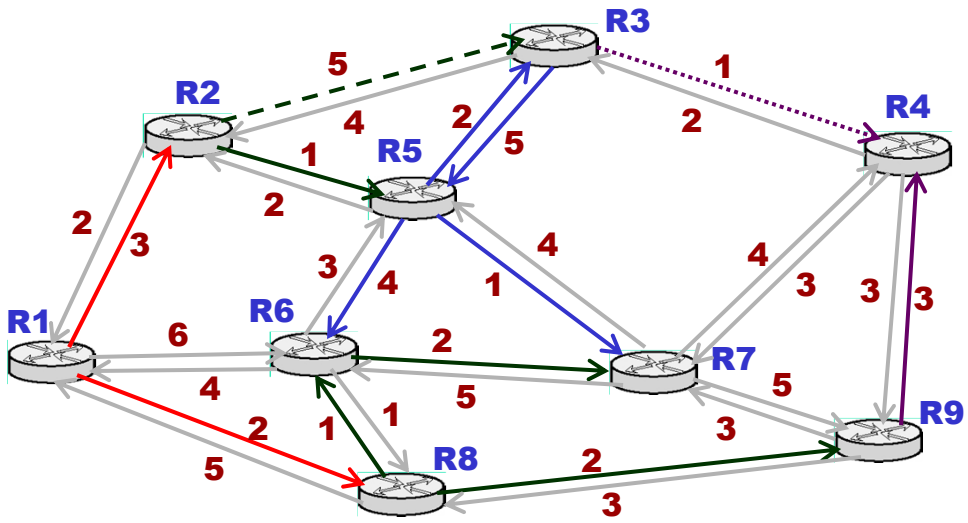
$L_2(8)=\min_j[L_1(j)+w(j,8)]$; j = 1,..,9

$=\min[L_1(1)+w(1,8),\ldots, L_1(6)+w(6,8),..,L_1(9)+w(9,8)]$

=Path ไม่เปลี่ยน

กรณี n=9 พบว่า Path 1-8-9 มีอันเดียวที่มี 2 Hop

Example: Bellman Ford จาก Node 1 (h=3)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Cost R1 ไป Node 4 = Infinity

$L_2(2)=3$; $L_2(3)=7$; $L_2(5)=4$; $L_2(6)=6$
$L_2(7)=8$; L2(8)=2; $L_2(9)=4$: $L_2(1)=0$

คำนวณ $L_{h+1}(n)=\min_j[L_h(j)+w(j,n)]$

$L_3(5)=\min_j[L_2(j)+w(j,5)]$; j = 1,..,9

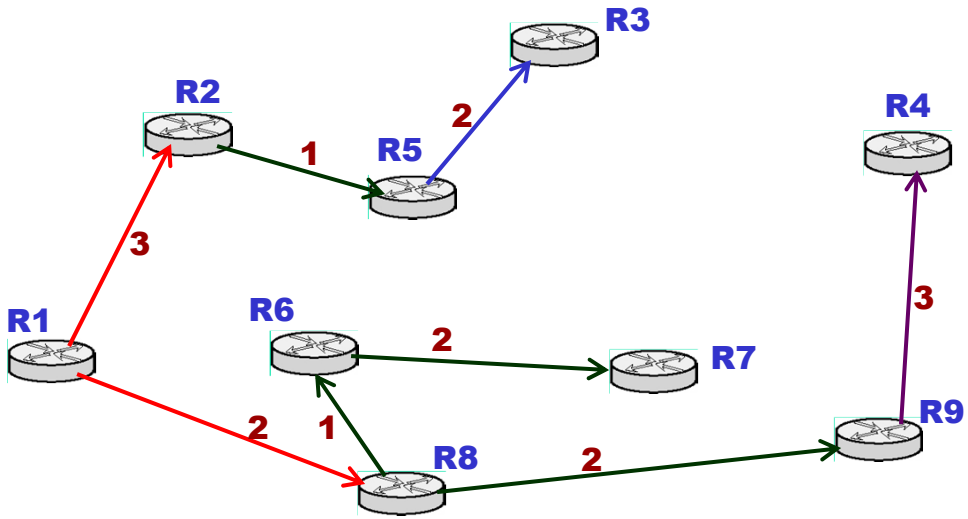
$=\min[L_2(1)+w(1,5),L_2(2)+w(2,5),L_2(3)+w(3,5),..,L_1(9)+w(9,5)]$

=Path ไม่เปลี่ยน

Path ไป R7 ไม่เปลี่ยน แม้ว่า Path 1-2-5-7 จะเท่ากัน

Path ไป R3 เปลี่ยน จาก 1-2-3 เป็น 1-2-5-3, Path R4 = min(1-2-3-4,1-8-9-4)

Example: Bellman Ford จาก Node 1 (h=4)



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Algorithm จะหยุดเมื่อถึงจำนวน Hop สูงสุดของ NW
Algorithm สามารถคำนวณได้แม้ว่าจะไม่รู้ Topology โดยการรับข้อมูลจาก Node ข้างเคียง แล้วมา Update ตารางของตัวเอง โดยใช้ค่า Cost ที่ต่ำกว่า

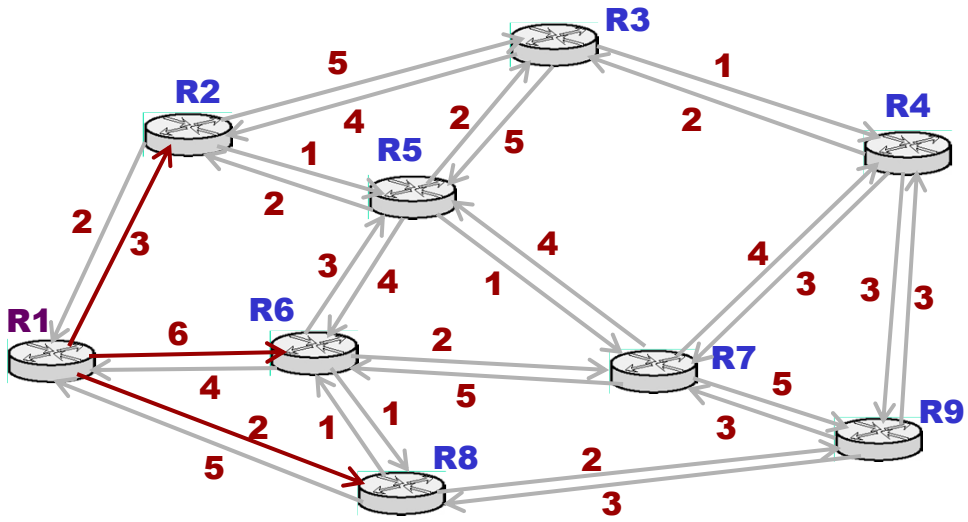
Dijkstra's Algorithm Definitions

- **Find shortest paths from given source node to all other nodes, by developing paths in order of increasing path length**
- **N = set of nodes in the network**
- **s = source node**
- **T = set of nodes so far incorporated by the algorithm**
- **w(i, j) = link cost from node i to node j**
  - w(i, i) = 0
  - w(i, j) = ∞ if the two nodes are not directly connected
  - w(i, j) ≥ 0 if the two nodes are directly connected
- **L(n) = cost of least-cost path from node s to node n currently known**
  - At termination, L(n) is cost of least-cost path from s to n

Dijkstra's Algorithm Method

- **Step 1 [Initialization]**
  - T = {s} Set of nodes so far incorporated consists of only source node
  - L(n) = w(s, n) for n ≠ s
  - Initial path costs to neighboring nodes are simply link costs
- **Step 2 [Get Next Node]**
  - Find neighboring node not in T with least-cost path from s
  - Incorporate node into T
  - Also incorporate the edge that is incident on that node and a node in T that contributes to the path
- **Step 3 [Update Least-Cost Paths]**
  - L(n) = min[L(n), L(x) + w(x, n)] for all n ∉ T
  - If latter term is minimum, path from s to n is path from s to x concatenated with edge from x to n
- **Algorithm terminates when all nodes have been added to T**

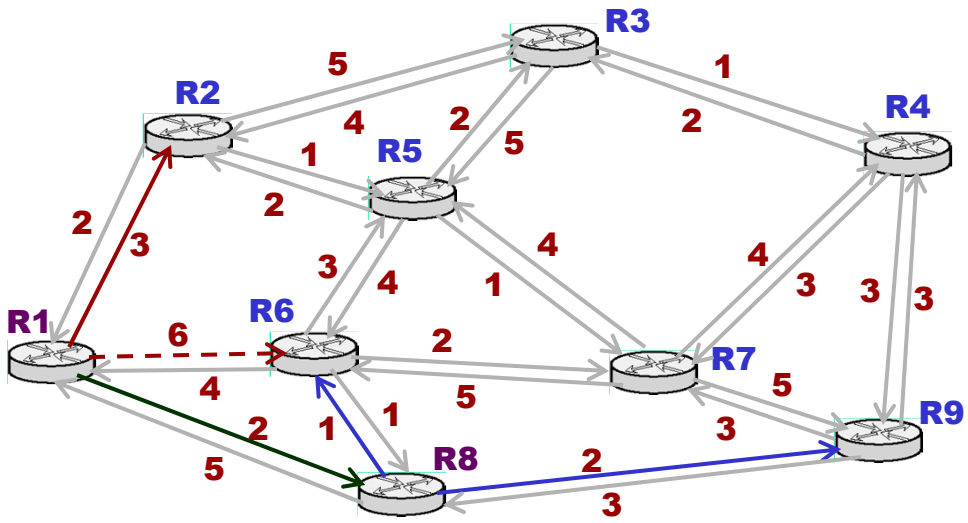
Example: Dijkstra จาก Node 1 (T={1})



T={1}, L(2)=3,L(3)=inf,L(4)=inf, L(5)=inf,L(6)=6,L(7)=inf,L(8)=2, L(9)=inf

	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

Example: Dijkstra จาก Node 1 (T={1})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

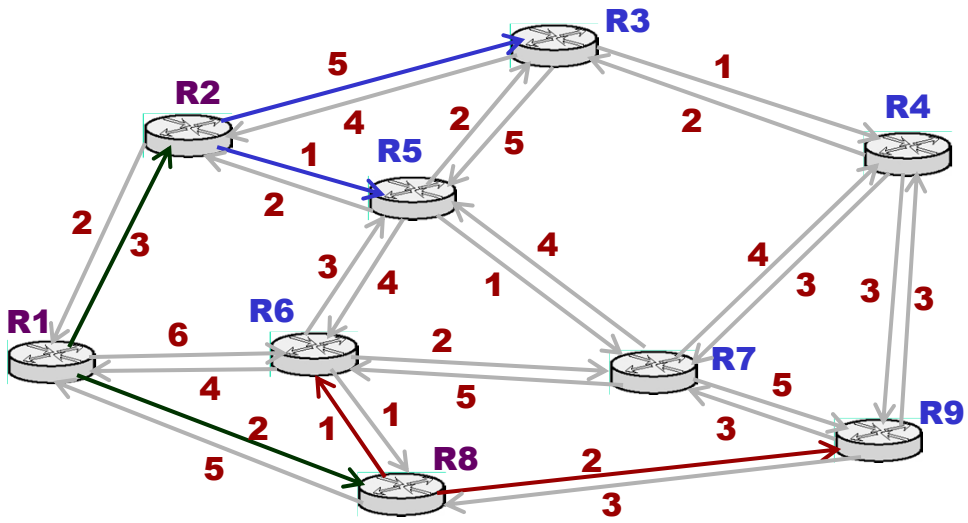
T={1}, L(2)=3,L(3)=inf,L(4)=inf, L(5)=inf,L(6)=6,L(7)=inf, L(8)=2, L(9)=inf

Min = L(8) ดังนั้นนำ Node 8 ใส่ใน T; T={1,8}

L(n) = min[L(n), L(8) + w(8, n)] for all n not in T

T={1,8}, L(2)=3,L(3)=inf,L(4)=inf, L(5)=inf, L(6)=3, L(7)=inf,L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

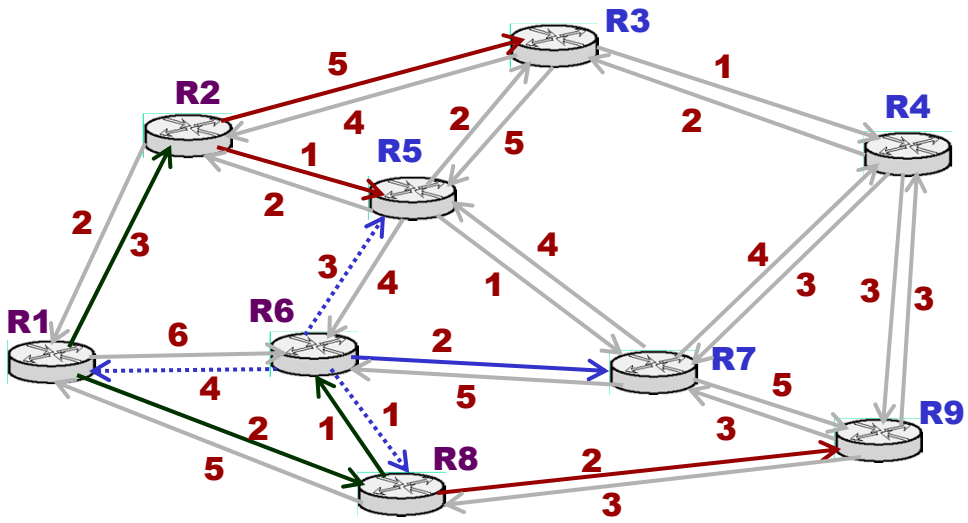
T={1,8}, L(2)=3, L(3)=inf,L(4)=inf, L(5)=inf,L(6)=3,L(7)=inf, L(8)=2, L(9)=4

Min = L(2) ดังนั้นนำ Node 2 ใส่ใน T; T={1,2,8}

L(n) = min[L(n), L(2) + w(2, n)] for all n not in T

T={1,2,8}, L(2)=3, L(3)=8, L(4)=inf, L(5)=4, L(6)=3, L(7)=inf, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

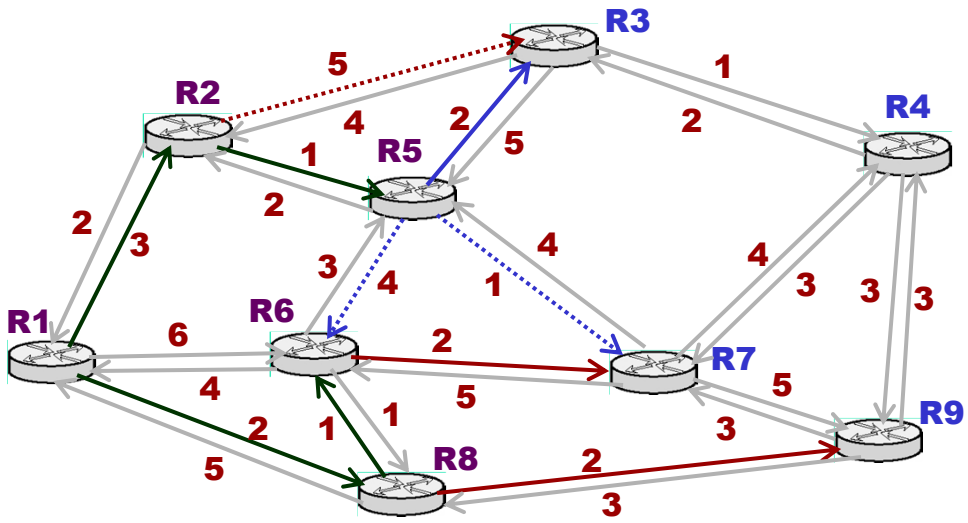
T={1,2,8}, L(2)=3, L(3)=8,L(4)=inf, L(5)=4, L(6)=3, L(7)=inf, L(8)=2, L(9)=4

Min = L(6) ดังนั้นนำ Node 6 ใส่ใน T; T={1,2,6,8}

L(n) = min[L(n), L(6) + w(6, n)] for all n not in T

T={1,2,6,8}, L(2)=3, L(3)=8, L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,6,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

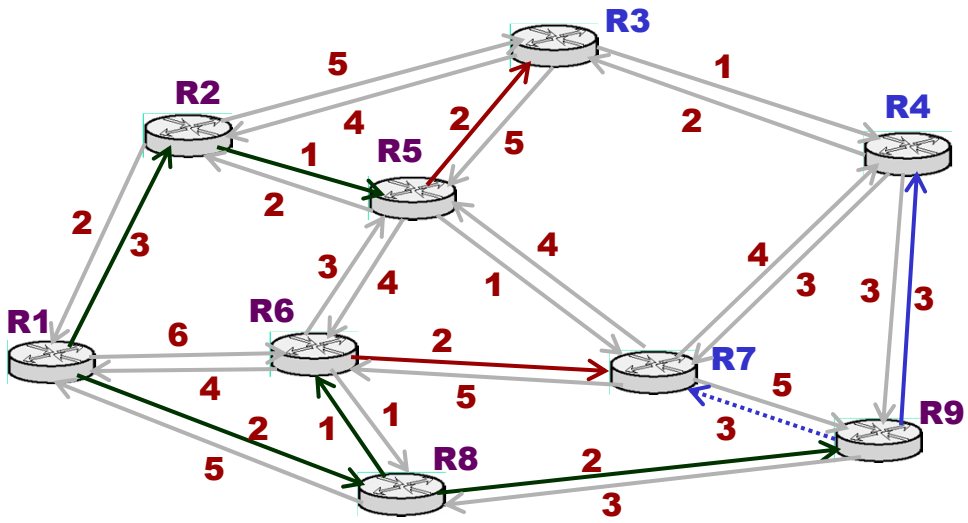
T={1,2,6,8}, L(2)=3, L(3)=8,L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(5) หรือ L(9) ก็ได้ ดังนั้นนำ Node 5 ใส่ใน T; T={1,2,5,6,8}

L(n) = min[L(n), L(5) + w(5, n)] for all n not in T

T={1,2,5,6,8}, L(2)=3, L(3)=6, L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,5,6,8})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

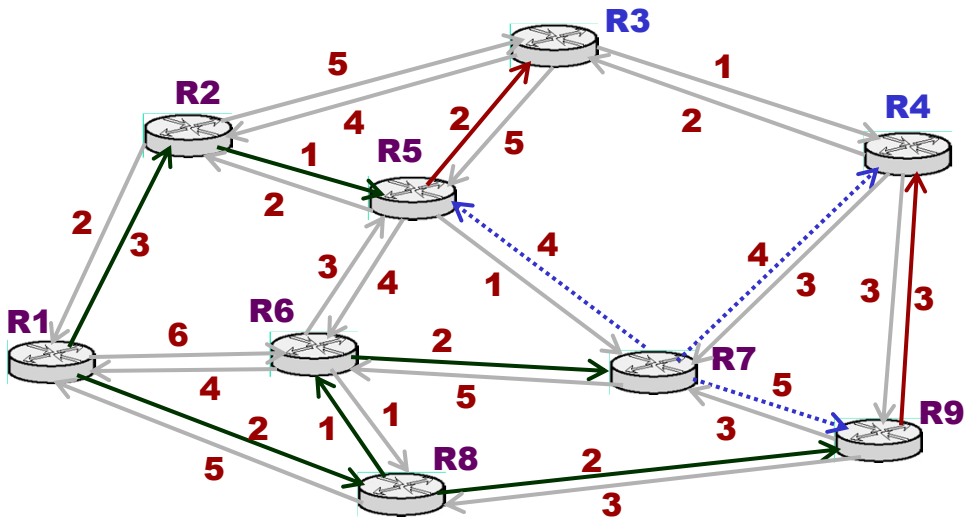
T={1,2,5,6,8}, L(2)=3, L(3)=6, L(4)=inf, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(9) ดังนั้นนำ Node 9 ใส่ใน T; T={1,2,5,6,8,9}

L(n) = min[L(n), L(9) + w(9, n)] for all n not in T

T={1,2,5,6,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,5,6,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

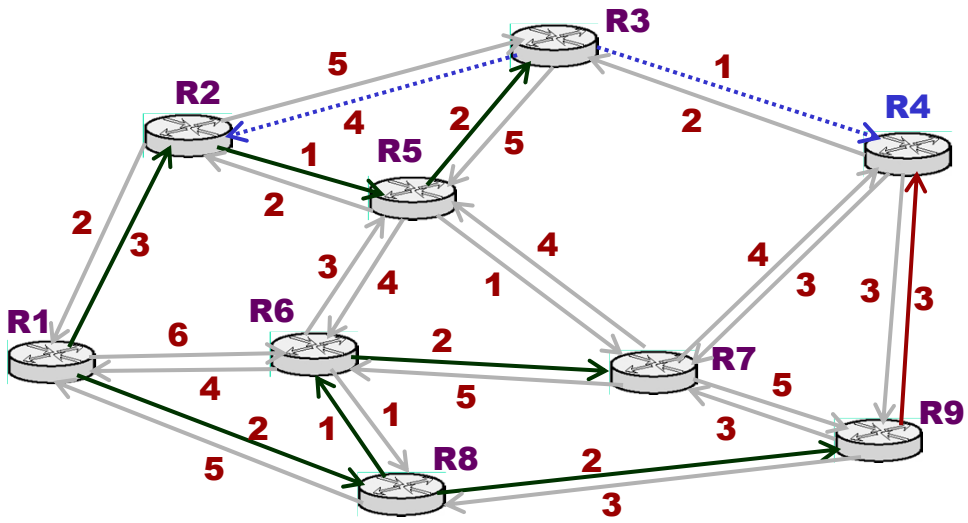
T={1,2,5,6,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(7) ดังนั้นนำ Node 7 ใส่ใน T; T={1,2,5,6,7,8,9}

L(n) = min[L(n), L(7) + w(7, n)] for all n not in T

T={1,2,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,5,6,7,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

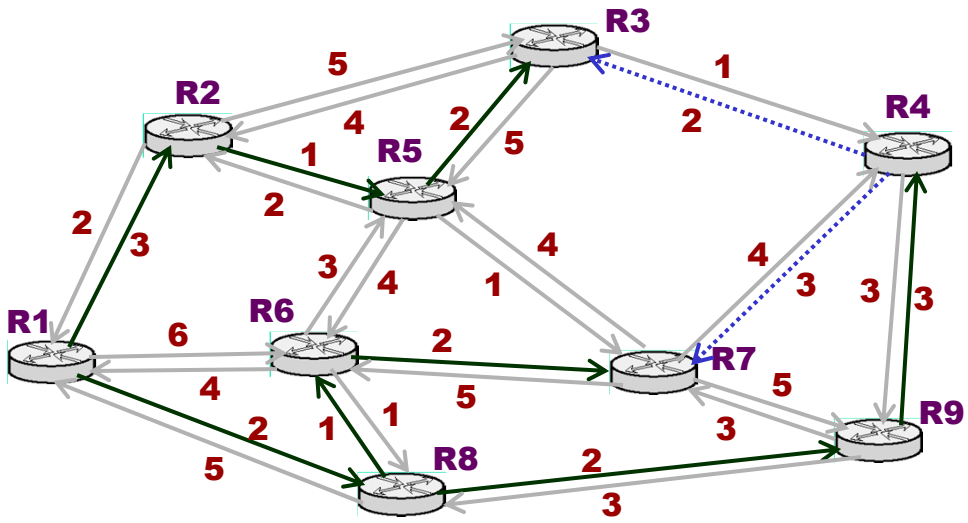
T={1,2,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(3) ดังนั้นนำ Node 3 ใส่ใน T; T={1,2,3,5,6,7,8,9}

L(n) = min[L(n), L(3) + w(3, n)] for all n not in T

T={1,2,3,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,3,5,6,7,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

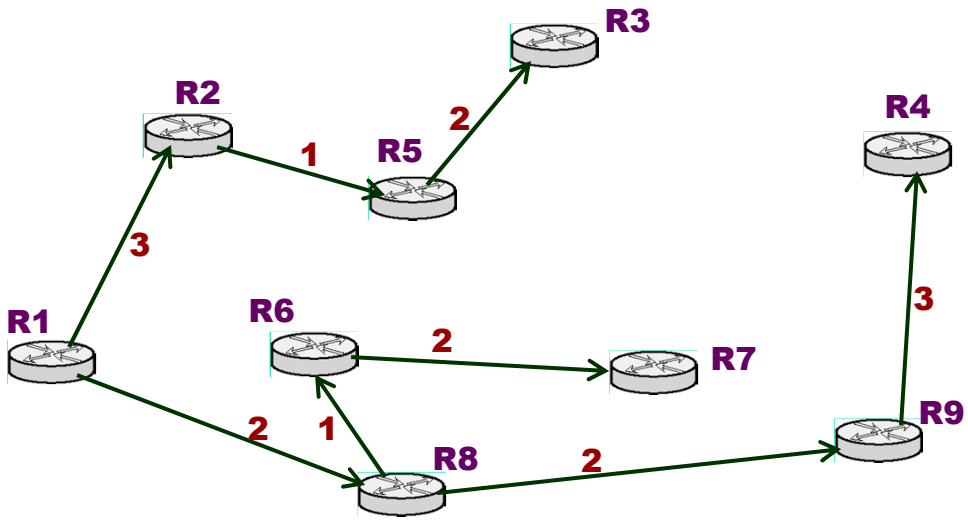
T={1,2,3,5,6,7,8,9}, L(2)=3, L(3)=6, **L(4)=7,** L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Min = L(4) ดังนั้นนำ Node 4 ใส่ใน T; T={1,2,3,4,5,6,7,8,9}

L(n) = min[L(n), L(4) + w(4, n)] for all n not in T

T={1,2,3,4,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Example: Dijkstra จาก Node 1 (T={1,2,3,4,5,6,7,8,9})



	1	2	3	4	5	6	7	8	9
1	-	3	∞	∞	∞	6	∞	2	∞
2	2	-	5	∞	1	∞	∞	∞	∞
3	∞	4	-	1	5	∞	∞	∞	∞
4	∞	∞	2	-	∞	∞	3	∞	3
5	∞	2	2	∞	-	4	1	∞	∞
6	4	∞	∞	∞	3	-	2	1	∞
7	∞	∞	∞	4	4	5	-	∞	5
8	5	∞	∞	∞	∞	1	∞	-	2
9	∞	∞	∞	3	∞	∞	3	3	-

T={1,2,3,4,5,6,7,8,9}, L(2)=3, L(3)=6, L(4)=7, L(5)=4, L(6)=3, L(7)=5, L(8)=2, L(9)=4

Algorithm Terminates

Dijkstra จะคำนวณได้ต่อเมื่อรู้ Topology ของ Network

HW # 7

- **Download และส่งสัปดาห์หน้า**

End of Week 12

- **Week 13 IP Routing II: BGP/RIP/OSPF and Multicast Protocols**
  - Chapter 27:27.9-27.16 + Extra Notes
    - BGP
    - RIP
    - OSPF
    - Subnet and VLAN
    - Switch Layer 3