**컴퓨터네트워킹 과제**

**HW#5**

**강좌 명: 컴퓨터네트워킹**

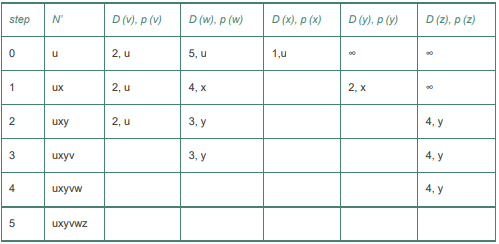
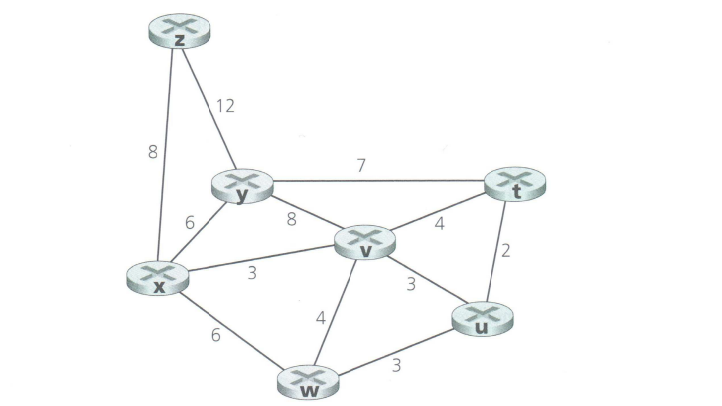
**교수님: 안종석 교수님**

**학과: 컴퓨터공학과**

**학번: 2017112138**

**이름: 정여준**

**P3. Consider the following network. With the indicated link costs, use Dijkstra’s shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1.**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Step** | **N’** | **D(t),p(t)** | **D(u),p(u)** | **D(v),p(v)** | **D(w),p(w)** | **D(y),p(y)** | **D(z),p(z)** |
| **0** | **x** | **∞** | **∞** | **3,x** | **6,x** | **6,x** | **8,x** |
| **1** | **xv** | **7,v** | **6,v** | **3,x** | **6,x** | **6,x** | **8,x** |
| **2** | **xvu** | **7,v** | **6,v** | **3,x** | **6,x** | **6,x** | **8,x** |
| **3** | **xvuw** | **7,v** | **6,v** | **3,x** | **6,x** | **6,x** | **8,x** |
| **4** | **xvuwy** | **7,v** | **6,v** | **3,x** | **6,x** | **6,x** | **8,x** |
| **5** | **xvuwyt** | **7,v** | **6,v** | **3,x** | **6,x** | **6,x** | **8,x** |
| **6** | **xvuwytz** | **7,v** | **6,v** | **3,x** | **6,x** | **6,x** | **8,x** |

xvt = 7

xvu = 6

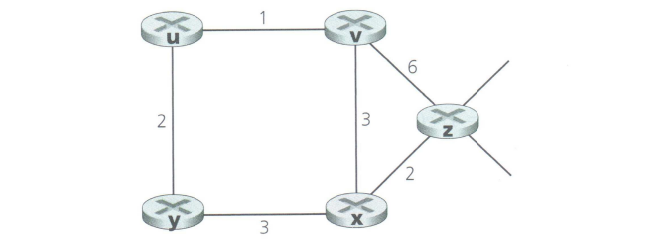
xv = 3

xw = 6

xy = 6

xz = 8

**P5. Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.**



Dm(m) = 0

Dm(n) = min{c(m, n) + Dn(n), c(m, n) + Do(n)}

Dm(o) = min{c(m, n) + Dn(o), c(m,o) + Do(o)}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cost to  from | u | v | x | y | z |
| v | ∞ | ∞ | ∞ | ∞ | ∞ |
| x | ∞ | ∞ | ∞ | ∞ | ∞ |
| z | ∞ | 6 | 2 | ∞ | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cost to  from | u | v | x | y | z |
| v | 1 | 0 | 3 | ∞ | 6 |
| x | ∞ | 3 | 0 | 3 | 2 |
| z | 7 | 6 | 2 | 5 | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cost to  from | u | v | x | y | z |
| v | 1 | 0 | 3 | 3 | 5 |
| x | 4 | 3 | 0 | 3 | 2 |
| z | 6 | 5 | 2 | 5 | 0 |

**P11. Consider Figure 5.7. Suppose there is another router w, connected to router y and z. The costs of all links are given as follows: c(x,y) = 4, c(x,z) = 50, c(y,w) = 1, c(z,w) = 1, c(y,z) = 3. Suppose that poisoned reverse is used in the distance-vector routing algorithm.**

1. **When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other?**
2. **Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again? Justify your answer.**
3. **How do you modify c(y,z) such that there is no count-to-infinity problem at all if c(y,x) changes from 4 to 60?**

(a)

|  |  |
| --- | --- |
| Router z | Informs w, Dz(x) = ∞ |
| Informs y, Dz(x) = 6 |
| Router w | Informs y, Dw(x) = ∞ |
| Informs z, Dw(x) = 5 |
| Router y | Informs w, Dy(x) = 4 |
| Informs z, Dy(x) = 4 |

(b)

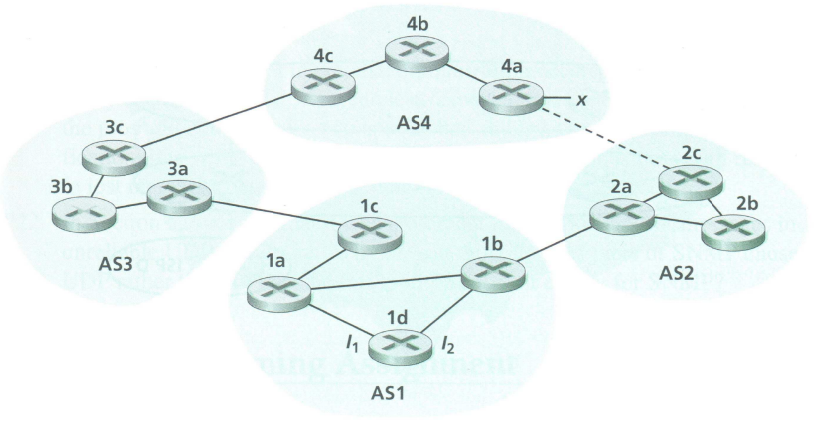
Count-to-infinity 문제가 발생할 것이다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| time | t0 | t1 | t2 | t3 | t4 |
| Z | ->w, Dz(x)=∞  ->y, Dz(x)=6 |  | 변화 없음 | ->w, Dz(x)=∞  ->y, Dz(x)=11 |  |
| W | ->y, Dw(x)=∞  ->z, Dw(x)=5 |  | ->y, Dw(x)=∞  ->z, Dw(x)=10 |  | 변화 없음 |
| Y | ->w, Dy(x)=4  ->z, Dy(x)=4 | ->w, Dy(x)=9  ->z, Dy(x)=∞ |  | 변화 없음 | ->w, Dy(x)=14  ->z, Dy(x)=∞ |

(c) 라우터간 링크가 제거되면 링크 비용이 4에서 60으로 변경돼도 Count-to-infinity 문제가 발생하지 않는다. 따라서 y와 z사이에 있는 링크를 제거한다.

**P14. Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is no physical link between AS2 and AS4.**

1. **Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, iBGP?**
2. **Router 3a learns about x from which routing protocol?**
3. **Router 1c learns about x from which routing protocol?**
4. **Router 1d learns about x from which routing protocol?**



(a) eBGP

(b) iBGP

(c) eBGP

(d) iBGP