UOU Report Vol. 16. No. 2. pp. 191~198. 1985 연구논문집 제16권 제 2 호 pp. 191~198. 1985

트리 네트워크에서 정보의 출발점에 관한 연구

산 업 공 학 과
고 재 문

(1985. 4. 30. 접수)

〈요 약〉

본 논문은 일반적인 나무 구조 네트워크에서 각 edge에서의 정보 전달 시간이 주어졌을 때 로칼 브로드캐스팅 방식에 따라 전체 전달 시간을 최소화하는 문제를 다루었다. 정보의 최적 출발점과 각 vertex에서의 전달 순서를 결정하는 해법을 개발하고 이에 대한 타당성을 입증하였다.

A Broadcasting Center in a Tree-type Network

Dept. of Industrial Engineering
Koh, Jae-Moon

(Received April 30, 1985)

〈**Abstract**〉

This paper deals with the problem of minimizing the broadcasting time in a tree-type network given edge transmission times by local broadcasting.

An algorithm determining a broadcasting center and the call sequence at each vertex is suggested and its correctness is given.

Ⅰ. 서 론

Tree-type network에서 broadcasting이라 함은 어떤 정보를 하나의 vertex로부터 다른 모든 vertex에 전달하는 과정을 말한다. 전달하는 방법에는 여러 가지가 있으나 여기서는 다음과 같은 방법을 생각한다.

① 정보를 이미 전달받은 vertex만이 다른 vertex에 정보를 전달할 수 있다.

② 정보는 바로 이웃하는(직접 연결된) vertex에만 전달한다.

③ 한 vertex는 동시에 여러 곳으로 정보를 보낼 수 없고 또 동시에 여러 곳으로부터 정보를 받을 수도 없다. 즉 한 vertex는 한 번에 한 곳으로만 또는 한 곳으로부터만 정보를 주거나 받을 수 있다.

이와 같은 전달 방법을 local broadcasting이라고 하며[2], 대부분의 message switching network에서 볼 수 있다. 이와 관련된 여러 문제 중에서 본 논문에서는, 모든 vertex가 알아야 할 정보가 있을 때 어느 vertex로부터 정보를 전달할 것인가 하는 문제를 다룬다.

각 edge에서의 전달 시간이 1인 경우에 대해서는 Proskurowski[3]와 Slater[4] 등이 간단한 해법을 제시하였다. 이에 대한 확장으로서 각 edge에서의 전달 시간이 각기 다르고 정보의 출발점이 주어진 경우에 대해서 비슷한 해법이 개발되어 있다.[5] 본 논문에서는 이 두가지를 통합한 일반적인 경우에 대해 고찰한다.

Ⅱ. 모형의 정립

$T=(V, E)$를 vertex들의 집합 V와 edge들의 집합 E로 이루어진 free tree라 하고, T의 각 edge (i, j)에서의 정보 전달 시간 C_{ij}가 주어졌다

고 하자. 또한 어떤 정보가 있어서 앞에서 밝힌 local broadcasting 방식에 따라 가능한 한 빠른 시간 내에 T의 모든 vertex에 전달되어야 한다고 하자. 이때 전체 전달 시간에 영향을 주는 것은 정보의 출발점과 각 vertex에서의 전달 순서가 된다.

예를 들어 그림 1과 같은 tree-type network과 각 edge에서의 전달 시간이 주어졌다고 하자. 만일 정보의 출발점을 3으로 했을 때 최소 정보 전달 시간은 14(그림 2, 표 1)가 되며, 6으로 했을 때는 12(그림 3, 표 2), 7로 했을 때는 12(그림 4, 표 3)가 된다.[5] 다른 vertex에 대해서는 표 4에 정리되어 있다. 따라서 그림 1에서는 정보의 출발점을 6 또는 7로 하는 것이 전체 정보 전달 시간을 최소화 한다.

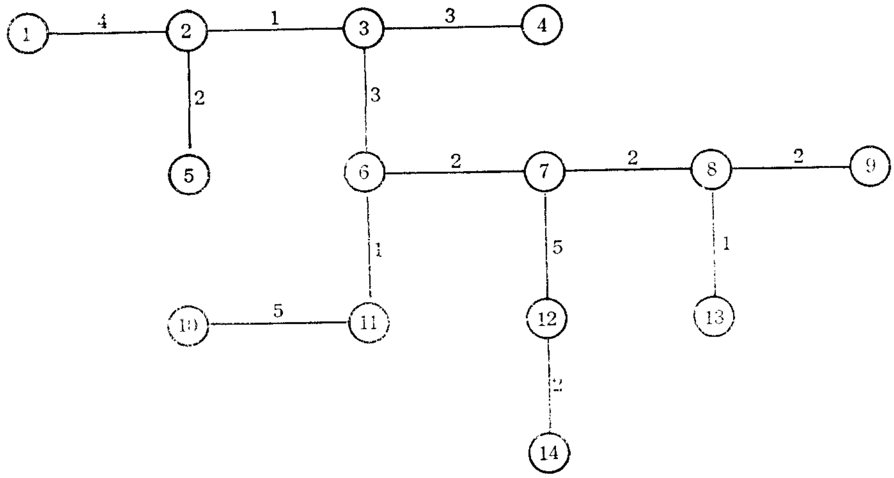

그림 1. tree-type network의 예
(각 edge 옆에 있는 숫자는 그 edge에서의 정보 전달 시간)

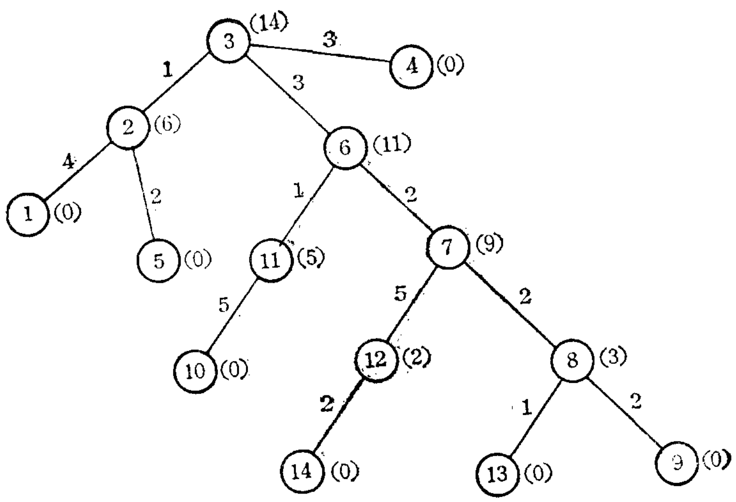

그림 2. root가 3일 때의 최소 정보 전달 시간
(각 vertex 옆의 ( ) 안의 숫자는 해당 subtree에서의 최소 정보 전달 시간임. 가장 밑에서부터 위쪽으로 차례로 구해 나간다).

표 1. root가 3일 때 각 vertex에서의 최적 전달 순서

단계	vertex	전 달 순 서
1	3	6, 2, 4
2	6	7, 11
	2	1, 5
	4	—
3	7	8, 12
	11	10
	1	—
	5	—
4	8	9, 13
	12	14
	10	—
5	9	—
	13	—
	14	—

표 2. root가 6일 때 각 vertex에서의 최적 전달 순서

단계	vertex	전 달 순 서
1	6	7, 3, 11
2	7	8, 12
	3	2, 4
	11	10
3	8	9, 13
	12	14
	2	1, 5
	4	—
	10	—
4	9	—
	13	—
	14	—
	1	—
	5	—

본 논문에서는 전체 정보 전달 시간을 최소화하는 정보의 출발점과 각 vertex에서의 전달순서를 결정하는 해법을 제시하고, 그 해법의 타당성을 입증하고자 한다. 이를 위해 편의상 다음과 같은 기호들을 도입한다.

$V(T)$: tree T에 속해 있는 모든 vertex들의 집합

$E(T)$: tree T에 속해 있는 모든 edge들의 집합

$T(u, v)$: tree T에서 edge (u, v)를 제거했을 때 생기는 2개의 subtree 중 u를 포함하고 있는 것

$b(v; T)$: v가 갖고 있는 정보를 T의 모든 vertex에 전달할 때까지 걸리는 최소 시간

$b(T)=\mathrm{Min}\{b(v; T) \mid v \in V(T)\}$

$b(u; T(u, v))$: u가 갖고 있는 정보를 $T(u, v)$의 모든 vertex에 전달할 때까지 걸리는 최소 시간

$BC(T)=\{v \in V(T) \mid b(v; T)=b(T)\}$: 정보 전달 시간을 최소화할 수 있는 정보 출발점들의 집합

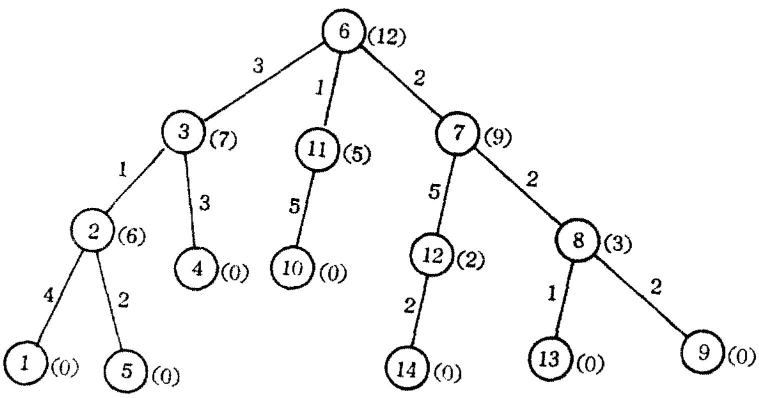

그림 3. root가 6일 때의 최소 정보 전달 시간

표 3. root 가 7일 때 각 vertex 에서의 최적 전달 순서

단계	vertex	전 달 순 서
1	7	6, 8, 12
2	6	3, 11
	8	9, 13
	12	14
3	3	2, 4
	11	10
	9	—
	13	—
	14	—
4	2	1, 5
	4	—
	10	—
5	1	—
	5	—

표 4. 정보의 출발점에 따른 최소 정보 전달 시간*

정보의 출발점 (v)	최소 정보 전달 시간 $b(v; T)$
1	19
2	15
3	14
4	17
5	17
6	12**
7	12**
8	14
9	16
10	18
11	13
12	17
13	16
14	19

*해법은 참고 문헌 (5) 참조

**최소값

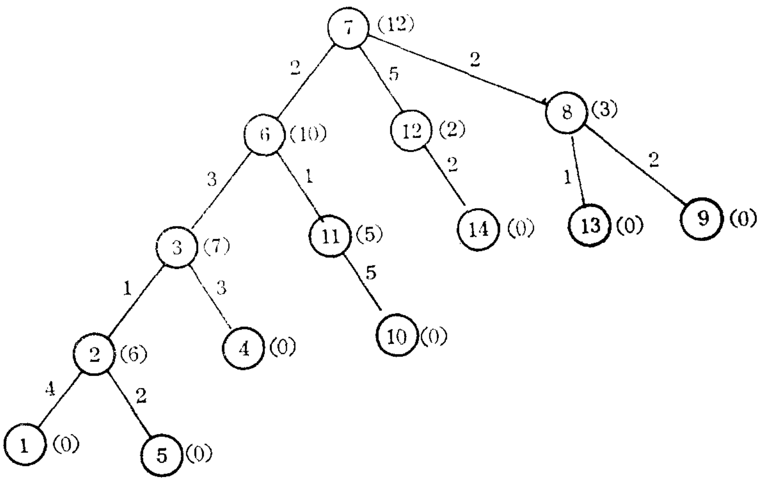

그림 4. root 가 7일 때의 최소 정보 전달 시간

Ⅲ. 해 법

본 논문에서 제시하고자 하는 해법은 다음과 같은 과정을 거친다. 우선 하나의 vertex 로 이루어진 subtree 안에서의 전달시간은 분명히 0이므로, tree T 의 모든 end vertex 에 대해서 0의 값을 지정한다. 이때 사용하는 변수를 t 라 하자. 즉 T 의 end vertex u 에 대해서 $t(u)=0$라 둔다. 다른 vertex 에 대해서는 end vertex 로부터 안쪽으로 차례로 구해 나간다.

본 해법의 각 단계에서는 T 의 vertex 들을 t 의 값이 결정된 것과 그렇지 않은 것으로 분류한다. t 의 값이 결정되지 않은 vertex 에 대해서는 다음과 같이 하나씩 차례대로 t 의 값을 구해 나간다. 즉 t 의 값이 결정되지 않은 어떤 vertex w 에 대해서 그 이웃하는 vertex 들이 하나(v)를 제외하고 전부 t 의 값들이 결정된 것들 $(u_1, u_2, \cdots, u_k)$일 때 w 를

input: free tree T consisting of n vertices with edge-transmission times c_{ij} ($n>2$).
output: call sequences at each vertex, minimum broadcasting time in each subtree, and broadcasting center

STEF (A0)
(A01) Let U be the set of endvertices in T.
(A02) For each $u\in U$, set $t(u)=0$.
(A03) Let T' be the tree yielded by deleting U in T.
(A04) Let W be the set of endvertices in T'.
(A05) For each $w\in W$, set $t(w)=\sum_{j=1}^{k} c_{wu_j}$ and $\text{call}(w)=u_1u_2\cdots u_k$, where $u_1, u_2, \cdots, u_k$ are adjacent vertices of w in U.

STEP (A1)
(A11) If $|V(T')|<3$, go to STEP (A4).
(A12) Select a vertex x in W such that $t(x)=\min\{t(w)\,|\,w\in W\}$.
(A13) Let y be the adjacent vertex of x in T'.
(A14) Define y to be the father of x.
(A15) $\text{call}(y)\longleftarrow x\|\text{call}\ (y)$.
(A16) $W\longleftarrow W-\{x\}$.
(A17) $V(T')\longleftarrow V(T')-\{x\}$.
(A18) $E(T')\longleftarrow E(T')-\{(x,y)\}$.
(A19) $U\longleftarrow U\cup\{x\}$.

STEP (A2)
(A21) Check whether y is a new endvertex of T'.
(A22) If so, go to STEP (A3).
(A23) Otherwise, go to STEP (A1).

STEP (A3)
(A31) Let call (y) be $u_1u_2\cdots u_k$.
(A32) Set $t(y)=\max\{\sum_{j=1}^{i} c_{yu_j}+t(u_i)\,|\,1\leq i\leq k\}$.
(A33) $W\longleftarrow W\cup\{y\}$.
(A34) Go to STEP (A1).

STEP (A4)
(A41) If $|V(T')|=1$, stop.
(A42) Let $x\in V(T')$ and $y\in V(T')-\{x\}$.
(A43) $\text{call}(x)\longleftarrow y\|\text{call}\ (x)$.
(A44) Set $b(T)\longleftarrow\max\{c_{xy}+t(x), c_{xy}+t(y)\}$. Stop.

그림 5. 해 법(A)

$u_1, u_2, \cdots, u_k$의 father vertex로 간주하고 $t(w)$의 값을 다음과 같이 결정한다.

$$t(w)=\text{Max}\{\sum_{j=1}^{i} c_{wu_j}+t(u_i)\,|\,1\leq i\leq k\}$$

단, $t(u_1)\geq t(u_2)\geq\cdots\geq t(u_k)$이다.

이런 과정을 모든 vertex에 대한 t의 값들이 결정될 때까지 반복한다. 그러면 임의의 vertex w에 대해서 $t(w)$의 값은 w와 그의 descendant들로 이루어진 subtree T_w에서의 최소 정보 전달 시간, 즉 $b(w;\ T_w)$가 되며, 최초의 tree T에서의 최소 정보 전달 시간도 쉽게 구할 수 있다. 이 과정을 정리하면 그림 5와 같다.

Ⅳ. 해법에 대한 타당성

본 절에서는 앞에서 제시한 해법 (A)에 따라 구한 t의 값들이 각 subtree에서의 최소 정보 전달 시간이 됨을 보이고, 최종 단계에서의 vertex가 정보의 최적 출발점임을 증명하고자 한다.

(보조 정리 1)

해법 (A)에 의하여 U에 포함되는 vertex들의 순서를 $v_1, v_2\cdots$라 하자. 즉, v_1을 처음으로 U에 들어 가는 vertex, v_2를 두번째로 U에 들어 가는 vertex, 일반적으로 v_k를 k번째로 U에 들어 가는 vertex라 하자. 그러면 $i\leq j$일 때 $t(u_i)\leq t(u_j)$이다.

(증 명)

U가 개정되는 경우는 (A19)뿐이며 U에 추가되는 vertex는 (A1)에 의하여 반드시 W에 속해 있는 vertex에서 고른다. 이제 어떤 단계에서 (A19)에 의하여 U에 $k+1$번째로 들어 가는 vertex를 생각하자. 그런데 v_k가 U에 추가될 때 v_{k+1}이 W에 포함되어 있으며 (A12)에 의하여 $t(v_k)\leq t(v_{k+1})$이 된다. v_k가 U에 추가될 때 v_{k+1}이 W에 포함되어 있지 않으면 v_{k+1}은 (A33)에 의하여 W에 들어 간다. 이때 (A15)에 의하여 v_k는 call(v_{k+1})에 포함되어 (A32)에 의하여 $t(v_k)\leq t(v_{k+1})$이다. 이것은 k의 가능한 모든 값에 대하여 성립하므로 $i\leq j$일 때 $t(v_i)\leq t(v_j)$이다. ■

보조 정리 1은 U에 나중에 추가되는 vertex의 t의 값이 먼저 들어 간 vertex보다 크거나 같음을 보여 준다.

(보조 정리 2)

해법 (A)의 어떤 단계에서, 임의의 어떤 vertex w에 대하여 call$(w)=u_1u_2\cdots u_k$라 하자. 그러면 $t(u_1)\geqslant t(u_2)\geqslant\cdots\geqslant t(u_k)$이다.

(증 명)

초기 단계인 (A05)에서는 명백히 $t(u_1)=t(u_2)=\cdots=t(u_k)=0$이다. (A15)에서 현재의 call$(w)=u_2\cdots u_k$에 새로운 vertex u_1이 추가된다고 하자. 그런데 call(w)가 개정되는 경우는 (A15)뿐이고 또한 call(w)에 포함되는 vertex는 (A19)에 의하여 즉시 U에 추가된다. 따라서 u_1이 call(w)에 추가될 때 $u_2,\cdots,u_k$ 들은 이미 U에 포함되어 있는 vertex들이다. 보조 정리 1에 의하여 $t(u_1)\geqslant t(u_2)$이다. 마찬가지로 이 관계는 u_2가 call(w)에 추가될 때도 성립한다. 즉, $t(u_2)\geqslant t(u_3)$이다. 같은 논리를 적용하면 $t(u_1)\geqslant t(u_2)\geqslant\cdots\geqslant t(u_k)$가 된다. ■

여기서 논리 전개의 편의를 위해 새로운 기호를 도입한다. 해법 (A)가 완료되었을 때에는 각 이웃하는 vertex 간에 부자 관계 (father-son)가 성립된다. 또한 자신의 father vertex가 없는 vertex는 root에 해당하며, 자신의 son vertex를 갖지 못하는 vertex는 end vertex에 해당한다. 이때 임의의 vertex w에 대하여 T의 subtree T_w를 다음과 같이 정의한다. 즉 T_w를 w와 그의 descendant들로 이루어지고 w가 그 root인 tree라고 하자. 또한 T_w에서의 정보 전달은 root인 w에서 시작한다고 하자.

(정리 1)

어떤 tree T_{wi}에서 root w의 son vertex들을 $u_1,u_2,\cdots,u_k$라 하고 subtree T_{u_i}, $i=1,2,\cdots,k$에서의 최소 정보 전달 시간을 $t(u_i)$라 하자. 그러면 $t(u_1)\geqslant t(u_2)\geqslant\cdots\geqslant t(u_k)$일 때

$$t(w)=\mathrm{Max}\{\sum_{j=1}^{i} c_{wu_j}+t(u_i)\mid 1\leq i\leq k\} \qquad ■$$

는 T_w에서의 최소 정보 전달 시간이다.

(증 명)

참고 문헌 (5)에 제시되어 있다.

정리 1은 각 subtree에서의 최소 정보 전달 시간 중 가장 큰 것에 해당하는 vertex에 가장 먼저 정보를 보내는 것이 최적이라는 것을 말해 준다.

(보조 정리 3)

해법 (A)가 완료되었을 때 임의의 vertex w에 대해서 call$(w)=u_1u_2\cdots u_k$라 하자. 만약 $t(u_i)=b(u_i;\ T_{u_i})\ \forall_i$이면 $t(w)=b(w;\ T_w)$가 된다.

(증 명)

(A14), (A15)에 의하여 $u_i,\ \forall_i$는 w의 son vertex이다. 또 보조 정리 2에 의하여

$$b(u_1;\ T_{u_1})\geqslant b(u_2;\ T_{u_2})\geqslant\cdots\geqslant b(u_k;\ T_{u_k})$$

가 된다. 그런데 $t(w)$는 (A32)에 의하여

$$t(w)=\mathrm{Max}\{\sum_{j=1}^{i} c_{wu_j}+b(u_i;T_{u_i})\mid 1\leq i\leq k\}$$

이다. 이 조건들은 정리 1의 조건들을 만족하므로 $t(w)=b(w;\ T_w)$이다. ■

이상의 결과를 종합하여 다음과 같은 중요한 정리를 얻을 수 있다.

(정리 2)

해법 (A)가 완료되었을 때 임의의 vertex w에 대하여 $t(w)=b(w;\ T_w)$이다.

(증 명)

수학적 귀납법으로 증명할 수 있다. 즉 초기 조건으로서 T의 모든 end vertex u에 대하여 명백히 $t(u)=b(u;\ T_u)=0$이다. 또한 보조 정리 3에 의하여 end vertex로부터 root까지 순차적으로 $t(w)=b(w;\ T_w)$가 됨을 알 수 있다. ■

정리 2는 각 vertex에서의 t의 값이 해당 subtree에서의 최소 정보 전달 시간임을 보여 준다.

(보조 정리 4)

u, v를 T의 이웃하는 vertex라 하자. $b(u;\ T(u,v))\geq b(v;\ T(v,u))$이면 다음이 성립한다

① $b(v;\ T)=c_{uv}+b(u;T(u,v))$

② $b(u;\ T)=c_{uv}+b(v;\ T(v,u))$

(증 명)

① 정리 1에 의하여 v로부터 가장 먼저 u에 정보를 보내는 것이 $b(v;\ T)$를 구하는데 최적이다. 따라서

$$b(v;\ T)=\mathrm{Max}\{c_{uv}+b(u;\ T(u,v)),\ c_{uv}+b(v;\ T(v,u))\}$$
$$=c_{uv}+b(u;\ T(u,v))$$

②u로부터 가장 먼저 v에 정보를 보내는 것이 $b(u;\ T)$를 구하는데 최적이 아닐 수도 있다. 왜냐 하면 $T(u,\ v)$ 속에 $b(v:\ T(v,u))$ 보다 더 큰 값을 갖는 subtree가 존재할 수도 있기 때문이다. 따라서

$$b(u;\ T)\leq\mathrm{Max}\{c_{vu}+b(v;\ T(v,u)),\ c_{vu}+b(u;\ T(u,v))\}$$
$$=c_{uv}+b(u;\ T(u,v))$$ ■

(보조 정리 5)

해법 (A)가 완료되었을 때 u를 v의 father vertex라 하자. 그러면 $b(u;\ T)\leq b(v;\ T)$이다.

(증 명)

정리 2에 의하여 $t(v)=b(v;\ T(v,\ u))$. (A12)에 의하여 $t(v)\leq b(u;\ T(u,v))$. 따라서 $b(v;\ T(v,u))\leq b(u;\ T(u,v))$. 보조 정리 4에 의하여

$b(u;\ T)\leq b(v;\ T)$ ■

이로부터 다음의 정리를 얻을 수 있다.

(정리 3)

해법 (A)가 완료되었을 때 (A4)에서 T'에 남아 있는 vertex를 $x,\ y$라 하자. 그러면 $b(x;\ T)=b(y;\ T)=b(T)$이다.

(증 명)

편의상 $t(x)\geqslant t(y)$라 하고 u를 최후로 U에 추가된 vertex라 하자. 그러면 보조 정리 1에 의하여 $t(y)\geqslant t(u)$가 되므로 $b(x;\ T)$를 구하기 위해서는 가장 먼저 y에 정보를 전달해야 한다. 따라서

$$b(x;\ T)=\mathrm{Max}\{c_{xy}+b(x;\ T(x,y)),\ c_{xy}+b(y;\ T(y,\ x))\}$$
$$=\mathrm{Max}\ \{c_{xy}+t(x),\ c_{xy}+t(y)\}$$
$$=c_{xy}+t(x)$$

또 $b(y;\ T)$를 구하기 위해서는 가장 먼저 x에 정보를 전달해야 한다. 따라서

$$b(y;\ T)=\mathrm{Max}\{c_{yx}+b(y,\ T(y,\ x)),\ c_{yx}+b(x;\ T(x,y))\}$$
$$=c_{xy}+t(x)$$

그러므로 $b(x;T)=b(y;\ T)$. 또한 x와 y는 각각 $T(x,y)$와 $T(y,x)$에서의 root에 해당하므로 보조 정리 5에 의하여

$$b(x,\ T)\leq b(v_1;\ T)\ \forall v_1\in V(T(x,y))$$
$$b(y;\ T)\leq b(v_2;\ T)\ \forall v_2\in V(T(y,x))$$

따라서 $b(x;\ T)=b(y;\ T)\leq b(v,\ T)\ \forall v\in V(T)$. 즉 $b(x;\ T)=b(y;\ T)=b(T)$. ■

정리 3은 해법 (A)가 완료되었을 때 최후에 남아 있는 vertex가 $BC(T)$에 속함을 보여 준다.

V. 적용 예제

그림 1에 주어진 tree network에서 앞에서는 각 vertex에 대한 최소 정보 전달 시간을 각각 구하여 그 중 최소값을 갖는 vertex를 정보의 최적 출발점으로 하였다. 여기서는 해법 (A)에 의하여 더 빨리 계산할 수 있음을 보인다. 그 과정을 살펴 보면 먼저 초기화 단계에서 $U=\{1,4,5,9,10,13,14\}$가 되고 T'는 그림 6a와 같다. 또 $W=\{2,8,11,12\}$가 $t(2)=6,\ \ t(8)=3,\ \ t(11)=5,\ \ t(12)=2$가 된다. (A1)에서 $x=12,\ y=7$이 되며 $W=\{2,8,11\}$이 된

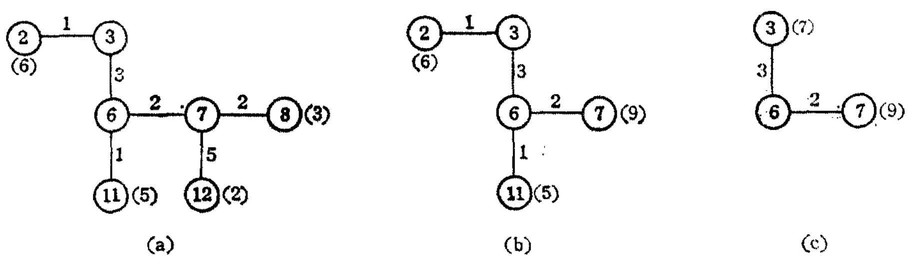

그림 6. 해법(A)의 그림 1에 대한 적용 과정
(각 vertex 옆의 ( ) 안의 숫자는 그 vertex의 t의 값을 나타냄)

다. 다음에 (A2)를 거쳐 (A1)으로 오면 $x=8$, $y=7$이 된다. 여기서 (A1)이 끝났을 때 7이 T'의 새로운 end vertex가 되므로 (A3)에서 $t(7)=9$가 되며 $W=\{2, 11, 7\}$. 되고 T'는 그림 6b와 같이 된다. 다시 (A1)에서 $x=11$, $y=6$이 되고 $W=\{2, 7\}$이 된다. (A2)를 거쳐서 (A1)으로 오면 $x=2$, $y=3$이 되고 3이 T'의 새로운 end vertex이므로 (A3)에서 $t(3)=7$이 된다. 이때 T'는 그림 7c와 같이 되며 다시 (A1)에서 $x=3$, $y=6$이 된다. 또한 (A2), (A3)를 거치면 $t(6)=10$이 된다. 그러면 $V(T')=\{6, 7\}$이므로 (A4)로 간다. 여기서 $b(T)=2+10=12$, 정보의 출발점은 6 또는 7이 된다.

Ⅵ. 결 론

본 논문에서는 일반적인 나무 구조 네트워크에서 각 edge에서의 전달 시간이 주어졌을 때 local broadcating 방식에 따라 전체 전달 시간을 최소화하기 위한 정보의 출발점과 각 vertex에서의 전달 순서를 결정하는 해법을 개발하였다. 그리고 그 해법의 타당성을 입증하였다. 이 연구는 edge에서의 전달 시간이 전부 1인 경우의 확장으로서 수행되었다.

본 연구에 대한 확장으로서 tree보다 좀 더 일반적인 network에 대해서도 고찰할 필요가 있으며, local broadcasting 방식을 약간 확대시켜 동시에 정보를 전달할 수 있는 vertex가 여러 개인 경우에 대해서도 생각할 수 있다. 또 edge에서의 전달 시간이 확률적인 경우에 대해서도 연구의 필요성을 느낀다.

참 고 문 헌

(1) A. M. Farley, "Minimal Broadcast Networks," Networks, Vol. 9, pp. 313—332, 1979.

(2) A. M. Farly, " Minimum-Time Line Broadcast Networks," Networks, Vol. 10, pp. 59—70, 1980.

(3) A. Proskurowski, "Minimum Broadcast Trees," IEEE Trans. Comput., Vol. C-30, No. 5, pp. 363—366, 1981.

(4) P. J. Slater, E. J. Cockayne, & S. T. Hedetniemi, "Information Dissemination in Tree," SIAM J. COMPUT., Vol. 10, No. 4, pp. 692—701, 1981.

(5) J. M. Koh, "Minimum Broadcasting Time in Tree Network," UIT Report, Vol. 15, No. 2, pp. 301—307, 1984.