폭우선생성트리 갱신문제를 위한 분산알고리즘

최형식*, 박정호*, 양해술**
*선문대학교 전자계산학과
**호서대학교
e-mail:comguy@omega.sunmoon.ac.kr

A Distributed Algorithms for Breadth-first spanning Tree Updating Problem

Hyung-Sik Choi*, Jung-Ho Park*, Hea-Sool Yang**
*Dept of Computer Science, Sun-Moon University
**Ho-Seo University

요약

폭우선생성트리 등의 문제를 해결하는데 필요한 정보가 네트워크상의 프로세서에 분산되어 있는 상황에서, 그들 정보를 교환하면서 그 문제를 해결하는 알고리즘을 분산알고리즘(Distributed Algorithm)이라고 한다.

폭우선생성트리가 이미 구성되어 있는 비동기식 네트워크상에서 네트워크 형상이 변할 경우, 이로인해 구성되어 있던 폭우선생성트리를 갱신해야 하는 경우가 발생한다. 본 논문에서는 이러한 경우 폭우선생성트리를 효율적으로 갱신하는 메시지 복잡도와 이상시간복잡도 모두 $O(p\sqrt{q} + q + a + n')$인 분산알고리즘을 제안한다. 여기서, a는 추가 링크의 수, n는 네트워크의 토폴로지가 변한후의 네트워크상에 존재하는 노드수를 각각 나타낸다. 그리고, p는 삭제 또는 추가 링크를 가진 이중연결성분에 속하는 전체 노드 수를 나타내며, q는 삭제 또는 추가 링크를 가진 이중연결성분에 속하는 전체 링크 수를 나타낸다.

1. 서론

네트워크상에서 프로세서와 링크의 추가 및 삭제가 빈번히 발생함으로 인해 네트워크 형상은 일정하지 않고 동적으로 변화하는 것이라고 생각할 수 있다. 동적으로 네트워크 형상이 변화하는 네트워크 환경에 있어서는 그 토폴로지(topology) 정보를 하나의 프로세서가 통괄해서 일괄적으로 관리하는 것보다 각 프로세서가 자기에 관한 토폴로지 정보만을 분담해서 관리하는 것이 좋다. 이와같이 어떤 문제를 해결하는데 필요한 정보가 네트워크상의 프로세서에 분산되어 있는 상황에서 그들 정보를 교환하면서 그 문제를 해결하는 알고리즘을 분산 알고리즘(Distributed Algorithm)이라고 한다[1]~[9].

분산 알고리즘이 실행되는 네트워크 환경에서 메시지수가 코스트의 대부분을 차지하고, 또 메시지의 전송시간이 분산 알고리즘이 처리를 종료하는데 걸리는 시간의 대부분을 차지한다. 따라서, 분산알고리즘의 효율은 주로 메시지 복잡도(message complexity)와 이상시간복잡도(ideal-time complexity)로 평가된다. 메시지 복잡도란 분산알고리즘 실행중에 네트워크에서 교환되는 메시지의 총수를 말한다. 대부분의 네트워크 모델에서는 메시지의 전송시간에 관해서 유한이라는 것이외에는 아무것도 가정하지 않는다(비동기식 네트워크). 그러므로 일반적으로 순차알고리즘의 평가에 적용되고 있는 시간복잡도(time complexity)의 개념은 적용할 수 없다. 따라서, 분산알고리즘의 실행시간을 평가하는데 이상시간복잡도라는 것을 채택하고 있는데, 이상

시간복잡도란 프로세서 내에서의 처리시간을 무시하고 메시지가 링크를 통과하는 전송시간을 1단위시간으로 했을 때의 분산알고리즘이 종료할 때까지의 단위시간수를 말한다.

네트워크상에서 임의의 프로세서를 중심으로 하여 다른 모든 프로세서와의 폭우선생성트리(Breadth-first spanning Tree)를 구하는 문제는 효율적으로 메시지를 전송하는 등의 경우에 있어서 중요한 문제이다.

네트워크의 토폴로지는 동적으로 변화하기 때문에, 폭우선생성트리를 포함하여 각종 문제를 해결했다고 하더라도 네트워크의 변화에 따라 이들 문제를 다시 해결해야 하는 경우가 발생한다. 이와같이 네트워크의 변화에 따라 다시 해결하는 문제를 갱신문제(Updating Problem)라고 한다. 즉, 임의의 네트워크상에서 임의의 프로세서에 대한 폭우선생성트리가 구성되어 있는 상황에서 네트워크가 변화했을 때, 폭우선생성트리를 다시 구성하는 문제를 폭우선생성트리 갱신문제(Breadth-first spanning Tree Updating Problem)이라고 한다.

갱신문제는 네트워크의 변화에 따라 문제를 다시 해결한다는 점을 고려하지 않고, 변화후의 새로운 네트워크에 대해 처음부터 그 문제를 해결하는 기존의 알고리즘을 적용시키더라도 해결할 수 있다. 즉, 문헌 [6]에 제안된 알고리즘을 새로운 네트워크에 적용할 경우, 폭우선생성트리 갱신문제를 해결할 수 있다. 그러나, 갱신문제의 경우, 각 프로세서는 토폴로지 변화전의 네트워크에 대한 답을 가지고 있으므로, 각 프로세서가 가지고 있는 여러 가지 정보를 이용하여 갱신문제를 효율적으로 해결할 수 있으며, 보조정보를 이용하여 이중연결성분 갱신문제를 효율적으로 해결하는 알고리즘이 제안되었다[9].

지금까지 보조정보를 이용하여 폭우선생성트리 갱신문제를 해결하는 분산알고리즘은 제안되지 않았으며, 문헌 [6]에 제안된 폭우선생성트리 문제를 해결하는 분산알고리즘을 토폴로지 변화후의 새로운 네트워크 $N'=(P', L')$에 적용시켰을 경우, 폭우선생성트리 갱신문제를 메시지 복잡도와 이상시간복잡도 모두 $O(n'\sqrt{e'})$에 해결할 수 있다. 단, $n'=|P'|$, $e'=|L'|$이다. 즉, 문헌 [6]의 알고리즘에서는 토폴로지 변화전의 네트워크에 대해 각 프로세서가 가지고 있는 폭우선생성트리 등에 대한 답을 보조정보로서 이용하지 않고, 완전히 처음부터 폭우선생성트리를 재구성한다.

본 논문에서는 보조정보를 효율적으로 이용함으로써 폭우선생성트리 갱신문제를 효율적으로 해결하는 분산알고리즘을 제안하는데, 보조정보로는 토폴로지가 변하기 전의 네트워크에 대한 폭우선생성트리와 이중연결성분에 대한 답을 이용한다. 즉, 폭우선생성트리에 대한 답으로서, 각 프로세서는 토폴로지 변화전의 네트워크상에서 어느 링크가 아버지와 연결된 링크이고, 어느 링크가 아들과 연결된 링크라는 것을 알고 있다. 또한, 어느 링크가 같은 이중연결성분에 속하는지를 알고 있다.

본 논문에서 제안하는 분산알고리즘의 메시지 복잡도와 이상시간복잡도 모두 $O(p\sqrt{q} + q + a + n')$이다. 여기서, a는 추가 링크의 수, p는 삭제 또는 추가 링크를 가진 이중연결성분에 속하는 전체 노드수를 나타내며, q는 삭제 또는 추가 링크를 가진 이중연결성분에 속하는 전체 링크수를 나타낸다.

일반적으로 $p \le n'$와 $q \le e'$ 그리고 $q \le n'\sqrt{e'}$와 $a \le n'\sqrt{e'}$가 성립한다. 따라서, 본 논문의 알고리즘은 메시지복잡도와 이상시간복잡도에 관해서 문헌 [6]의 알고리즘보다 뛰어나다. 즉, 보조정보를 이용하는 것이 보조정보를 이용하지 않는 경우보다 일반적으로 효율적이라고 할 수 있다.

2. 정의

본 논문에서 고려하는 네트워크 $N=(P, L)$는 프로세서의 집합 P와 링크의 집합 L로 구성되어 있으며, 네트워크 N은 무방향 연결 그래프(connected undirected graph) $G=(V, E)$와 같으므로, 본 논문에서는 무방향 연결 그래프에 관한 용어를 네트워크에 대해서도 동일하게 사용한다. 단, 모든 링크에 할당되어 있는 무게(weight)는 모두 동일한 네트워크, 즉 링크에는 무게가 할당되지 않은 것으로 가정하며, 다중링크는 없는 것으로 한다.

[정의 1] 이중연결성분(biconnected component) 문제 : 이중연결성분이란 네트워크 N상의 임의의 두 프로세서 사이에 두 개이상의 서로다른 경로가 존재하는 N의 극대부분 그래프로서, 이러한 이중연결성분을 구하는 문제를 이중연결성분 문제라고 한다. 이중연결성분 문제에 대한 답으로서 각 프로세서는 같은 이중연결성분에 속하는 인접링크에 대해서 같은 레이블을 가지며, 서로 다른 이중연결성분에 대해서는 다른 레이블을 갖는다.□

[정의 2] 최단경로 : 임의의 두 프로세서를 연결하는 경로(path)중에서 그 경로의 길이가 가장 짧은 경로

를 말한다. □

[정의 3] 폭우선생성트리(breadth-first spanning tree) 문제 : 루트(root)로 지정된 하나의 프로세서 r과 r이외의 모든 프로세서에 대해 최단경로를 구하는 문제를 말한다. □

네트워크와 분산 알고리즘에 관해 다음과 같은 가정을 둔다.

[가정 1] 네트워크상에 공유 메모리는 없고, 프로세서간의 통신은 링크를 통한 메시지 교환만으로 행한다. 프로세서 u가 인접프로세서 v에게 보낸 메시지는 보낸 순서대로 유한시간내에 도중에 없어지지 않고, v에게 반드시 보내진다. □

[정의 4] 폭우선생성트리 갱신문제(Breadth-first spanning Tree Updating Problem, 이후 BTUP라고 한다) : 다음의 초기상태에서 시작하여 최종상태에 이르는 문제를 말한다.

(초기상태) 토폴로지 변화전의 네트워크 N에 대한 폭우선생성트리가 이미 구성되어 있으며, 각 프로세서는 토폴로지 변화후의 네트워크 N'에서 어느 링크가 새로 추가되고 삭제되었는지를 알고 있다.

(최종상태) 토폴로지 변화후의 네트워크 N'에 대한 폭우선생성트리가 구성되어 있다. □

[정의 5] 이중연결성분 갱신문제 : 폭우선생성트리 갱신문제에 대한 정의에서와 같이, 이중연결성분이 이미 구성되어 있는 초기상황에서, 토폴로지 변화후에 새로운 네트워크에 대한 이중연결성분을 구하는 문제를 말한다. □

[정의 6] 분산 알고리즘 : 분산알고리즘(*Distributed Algorithm*)이란 각 프로세서가 실행하는 프로그램으로 구성한다. 자발적으로 프로그램을 개시하는 프로세서를 시작 프로세서(start processor)라고 한다. 시작 프로세서이외의 프로세서는 인접 프로세서로부터 메시지를 받으면 프로그램 실행을 새기하는데, 여기서는 시작 프로세서는 하나라고 가정한다. 메시지 전송 지연의 차이, 프로세서 동작 속도의 차이등에 의해 여러 가지 실행 과정이 발생할 수 있으나, 어느 실행과정에 있어서도 유한개의 명령 실행후에 모든 프로세서가 프로그램 실행을 끝냈을 때 분산알고리즘이 종료했다고 한다. 어떤 분산알고리즘 A가 BTUP를 해결한다는 것은 유한시간내에 A가 종료되고, 그때 각 프로세서 v는 프로세서 r을 루트로 하는 폭우선생성트리상에서의 v의 아버지와 아들을 알고 있다. □

[가정 2] 각 프로세서가 실행하는 프로그램은 동일하다. □

분산알고리즘의 평가는 일반적으로 메시지량과 시간을 이용해서 행한다. 시간평가는 메시지의 전송지연과 각 프로세서의 동작 속도에 차이가 있기 때문에 용이하지 않아서 일반적으로 이상시간복잡도라는 개념을 이용한다.

[정의 7] 메시지 복잡도는 분산알고리즘 실행도중에 네트워크의 모든 프로세서 사이에서 교환되는 메시지수이다. 또, 이상적인 시간복잡도는 프로세서내에서 처리 시간을 무시하고, 메시지가 링크에 전달되는 전송시간을 1단위 시간으로 했을 때 알고리즘이 종료할 때까지의 단위시간수이다. □

3. BTUP를 해결하는 분산알고리즘

본 논문에서는 네트워크 $N=(P, L)$과 프로세서 r을 루트로 하는 폭우선생성트리 $BT(N, r)=(P_{BT}, L_{BT})$가 구성되어 있는 상황에서, 네트워크의 토폴로지가 변해서 $N'=(P', L')$로 되었을 때, N'에 대한 폭우선생성트리 $BT(N', r)=(P'_{BT}, L'_{BT})$를 재구성하는 분산 알고리즘을 제안한다.

본 논문에서는 BTUP(*Breadth-first spanning Tree Updating Problem*)를 효율적으로 해결하기 위해, 두 종류의 보조정보(*auxiliary information*) 즉, 이중연결성분 갱신문제의 답과 폭우선생성트리 갱신문제의 답을 이용한다.

네트워크 N상에 있는 하나의 이중연결성분을 하나의 프로세서로 치환하고, 절단점(*articulation point*)에 의해 인접하는 두 개의 이중연결성분에 대해서는 하나의 새로운 링크로 연결시켜면 새로운 트리 $T'=(P_{T'}, L_{T'})$를 얻을 수 있다. 즉, $P_{T'}=\{v|$ v는 N에 있어서 하나의 이중연결성분$\}$, $L_{T'}=\{(u, v)|$ $u(\in P_{T'})$와 $v(\in P_{T'})$는 N에 있어서 인접한 이중연결성분$\}$이다.

네트워크 N의 토폴로지가 바뀌어서 N'로 되었다고 가정하자. 이때 N상에 있는 이중연결성분이 삭제링크 또는 추가링크를 포함했는지 여부에 따라 다음의 세 경우로 나누어서 생각할 수 있다.

(경우 1) 어떤 이중연결성분도 삭제링크와 추가링크를 포함하지 않은 경우

토폴로지가 바뀌었지만 삭제된 링크가 다시 복구가 되었거나, 추가된 링크가 다시 삭제되어 어떤 이중연결성분에도 아무런 영향을 미치지 않았기 때문에 N가 N'로 바뀌었지만, $N=N'$인 경우로서, 트리 T'의 토폴로지도 변하지 않고, 루트 r과 다른 프로

세서사이의 최단경로도 바뀌지 않는다. 따라서, BTUP를 해결하기 위한 추가 작업을 할 필요가 없으며, 토폴로지 변화전의 네트워크에 대한 폭우선생성트리 $BT(N, r)$이 그대로 토폴로지 변화후의 네트워크에 대한 $BT(N', r)$으로 된다.

(경우 2) 삭제링크가 존재하는 이중연결성분이 존재할 경우

삭제링크가 존재하는 이중연결성분을 B라고 하자. 이 경우에 B상에 있는 각 프로세서 u에 대해 경로 $r-u$가 바뀔지 모른다. 따라서, N'에 있어서 B에 대한 폭우선생성트리를 처음부터 다시 구성해야 한다.

그러나, T'상에 있어서 B의 후손인 다른 이중연결성분 B'에 대해서, B의 변화는 B'에 대한 토폴로지에 영향을 미치지 않으므로, B'에 대해 추가로 변경작업을 할 필요는 없다. 단, B'에 있어서 토폴로지 변화가 있을 경우에는 이들 B'에 대해서만 폭우선생성트리를 다시 구성하는 것으로 충분하다.

(경우 3) 추가링크를 포함한 이중연결성분이 존재할 경우

추가 링크에 의해 여러개의 이중연결성분(이후, BS로 표현)이 결합되어서 하나의 합병된 이중연결성분 M이 되는 경우로서, M상에 있는 각 프로세서 u에 대해 경로 $r-u$가 바뀔지 모른다. 따라서, N'에 있어서 BS에 대한 폭우선생성트리를 처음부터 다시 구성해야 한다.

합병된 이중연결성분 M을 하나의 프로세서로 대치시키면, N상에서의 T가 N'상에 있어서는 T''로 되는데, T'상에서 BS의 후손인 이중연결성분들은 그대로 T''상에서 M의 후손이 된다. 즉, N'에서 M으로 합병되더라도 후손들의 토폴로지에는 아무런 영향을 미치지 않으므로, 이들 후손들에 대해 추가로 변경작업을 할 필요는 없다. 단, B'에 있어서 토폴로지 변화가 있을 경우에는 이들 B'에 대해서만 폭우선생성트리를 다시 구성하는 것으로 충분하다.

4. 알고리즘의 평가

다음 정리에서 n'는 네트워크의 토폴로지가 변한 후의 네트워크상에 존재하는 노드수를 각각 나타내고, a는 추가 링크의 수를 나타낸다. 그리고, p는 삭제 또는 추가 링크를 가진 이중연결성분에 속하는 전체 노드 수를 나타내며, q는 삭제 또는 추가 링크를 가진 이중연결성분에 속하는 전체 링크수를 나타낸다.

[정리 1] 본 논문에서 제안하는 알고리즘의 메시지 복잡도와 이상시간 복잡도는 모두 $O(p\sqrt{q} + q + a + n')$이다.

(증명) 이중연결성분 갱신문제를 해결하는데 소요되는 메시지 복잡도와 이상시간 복잡도 모두 $O(q + a + n')$이 된다[9].

이중연결성분을 무결 이중연결성분 또는 손상된 이중연결성분으로 분류하는데 있어서, 토폴로지 변화전의 네트워크 N상에서의 이중연결성분에 할당된 레이블과 토폴로지 변화후의 네트워크 N'상에서의 이중연결성분에 할당된 레이블을 이용하여 간단히 판단할 수 있으므로, 이중연결성분의 분류하는데 소요되는 메시지 복잡도와 이상시간 복잡도는 $O(1)$이 된다.

무결 이중연결성분에 대한 폭우선생성트리를 구성하는데 있어서, N상에서의 폭우선생성트리가 그대로 N'에서의 폭우선생성트리로 된다. 트리에는 $n-1$ 이하의 링크가 있으므로, 무결 이중연결성분에 대한 폭우선생성트리를 구하는데 소요되는 메시지 복잡도와 이상시간 복잡도는 모두 $O(n)$이 된다.

문헌 [6]의 알고리즘은 프로세서와 링크수가 각각 n과 e인 네트워크에 대한 폭우선생성트리를 구하는데 메시지 복잡도와 이상시간 복잡도가 각각 $O(n\sqrt{e})$이다. 문헌 [6]의 알고리즘을 이용하여 손상된 이중연결성분에 대한 폭우선생성트리를 구하는데, 모든 손상된 이중연결성분에 속하는 프로세서와 링크수를 각각 p와 q라고 가정하면, 손상된 이중연결성분에 대해서는 메시지 복잡도와 이상시간 복잡도는 모두 $O(p\sqrt{q})$가 된다.

따라서, 본 논문에서 제안하는 알고리즘의 메시지 복잡도와 이상시간 복잡도는 모두 $O(p\sqrt{q} + q + a + n')$이 된다. □

참고문헌

[1] B. Awerbuch : "Complexity of network synchronization", *Journal of ACM*, Vol.32, No. 4, pp.804 ~ 823(Oct. 1985).

[2] B. Awerbuch and R.G.Gallager : "A new distributed algorithm to find breadth first search trees", *IEEE Trans. on Information Theory*, Vol.IT-33, No.3, pp.315 ~ 322(1987).

[3] B. Awerbuch : "Distributed shortest paths algorithm", *Proc. of 21st Symposium on Theory of Computing*, pp. 490 ~ 500(1980).

[4] M Ahuja and Y.Zhu : "An efficient distributed

algorithms for finding articulation points, bridges and biconnected components in asynchronous network", *In Porc. 9th Conference on Foundations of Software Technology and Theoretical Computer Science*(LNCS 405), pp. 99 ~ 108(1989).
[5] T.H. Cormen, C.E.Lerserson and R.L.Rivest : "Introduction to Algorithms", The MIT Press(1990).
[6] J. Park, T.Masuzawa, K.Hagihara and N.Tokura : "An efficient distributed algorithm for breadth first spanning tree problem", *Journal of IEICE*(D), Vol.J71-D, No.7, pp. 1576 ~ 1188(1988).
[7] B. Swaminathan and K.J.Goldman : "An incremental distributed algorithm for computing biconnected components", *Proc. 8th International Workshop on Distributed Algorithms*(LNCS 857), pp. 238 ~ 252(1994).
[8] T.Kameda and M.Yamashita " "Distributed Algorithms", Kindai-Kagaku-sya(1994).
[9] J. Park and C. Lee : "An Algorithm Solving the Biconnected-components Reconstruction Problem", *Journal of KIPS*, Vol. 5, No. 10, pp.2512 ~ 2520(1998).