제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

최단경로 재구성 분산알고리즘

김인환, 박정호
*선문대학교 컴퓨터정보학부

Inhwan, KIM, Jungho, PARK
*Division of Information and Computer Science, SUNMOON Univ.
e-mail:jhpark@sunmoon.ac.kr

요 약

본 논문에서는 최단경로 재구성문제를 해결하는 분산알고리즘을 제안한다. 최단경로 재구성문제란 최단경로가 이미 구성되어 있는 네트워크상에서 네트워크의 토폴로지 변화가 생겼을 때 토폴로지 변화에 따라 최단경로를 재구성하는 문제이다. 본 논문에서는 여러 개의 프로세서와 링크의 추가와 삭제에 대해 최단경로 재구성문제를 효율적인 분산알고리즘을 제안한다.

1. 서론

컴퓨터네트워크상에서는 프로세서가 갑자기 정자하거나 정지상태에 있던 프로세서가 회복되는 경우가 있다. 또, 프로세서간을 연결하는 링크도 사용을 못하게 되거나 두 개의 프로세서 사이에 새로 링크를 설치하는 경우가 있으므로 네트워크 형상은 일정하지 않고 동적으로 변화하는 것이라고 생각할 수 있다. 동적으로 네트워크 형상이 변화하는 네트워크 환경에 있어서는 그 토폴로지 정보를 하나의 프로세서가 통괄해서 관리하는 것보다 각 프로세서가 자기에 관한 토폴로지 정보만을 분담해서 관리하는 것이 바람직하다. 이와같이 어떤 문제를 해결하는데 필요한 정보가 네트워크상의 프로세서에 분산되어 있는 상황에서 그들 정보를 교환하면서 그 문제를 해결하는 알고리즘을 분산알고리즘이라고 한다.

네트워크상에서 임의의 프로세서를 중심으로 하여 다른 모든 프로세서와의 최단경로(shortest path)를 구하는 문제는 효율적인 메시지 전송을 위해서 중요한 문제이다. 네트워크의 토폴로지는 동적으로 변화하기 때문에 최단경로문제를 포함하여 각종 문제를 해결했다고 하더라도 네트워크의 변화에 따라 이들 문제를 다시 해결해야 하는 경우가 발생한다. 이와 같이 네트워크의 변화에 따라 다시 해결하는 문제를 재구성문제(Updating Problem)라고 한다. 즉, 임의의 네트워크상에서 임의의 프로세서에 대한 최단경로가 구성되어 있는 상황에서 네트워크가 변했을 때, 최단경로를 다시 구성하는 문제를 최단경로 재구성문제라고 한다.

최단경로 재구성문제에 대한 분산알고리즘은 아직까지 연구・발표되지 않았는데, Dijkstra가 제안한 순차알고리즘을 분산화하면 $O(n^2)$의 메시지복잡도와 시간복잡도로 최단경로 구성문제에 대한 분산알고리즘을 구현할 수 있다. n은 네트워크상의 프로세서수이다. 네트워크의 토폴로지가 바뀌어서 최단경로를 재구성해야 할 때, Dijkstra의 알고리즘을 이용

하면 $O(n'^2)$의 메시지복잡도로 최단경로 재구성문제를 해결할 수 있다. 네트워크 토폴로지 변화후의 새로운 네트워크에 있어서 이중연결성분의 수가 k개(각 연결성분에 속하는 프로세서의 수가 B_1, B_2, …, B_k)라고 가정할 때, Dijkstra 알고리즘의 메시지복잡도는 $O((B_1+B_2+\cdots+B_k)^2)$가 된다. 본 논문에서는 이중연결성분을 이용하여 메시지복잡도 $O(B_1^2+B_2^2+\cdots+B_k^2)$에 최단경로 재구성문제를 해결할 수 있는 분산알고리즘을 제안한다. 따라서, 본 논문에서 제안하는 알고리즘은 이중연결성분의 수가 많을수록 효율적인 알고리즘이 된다.

2. 정의

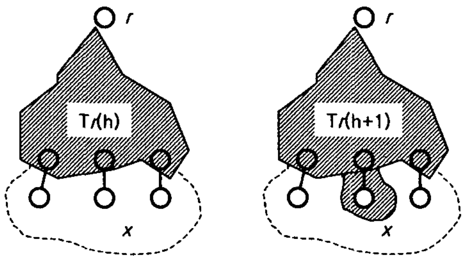
3. 최단경로 재구성 분산알고리즘

여기서는 네트워크 토포롤지가 변화했을 때 최단경로를 재구성하는 분산알고리즘에 대해 설명하기로 한다.

3.1 Dijkstra알고리즘의 분산화

Dijkstra알고리즘을 분산 네트워크에 적용한 경우의 기본 아이디어는 최단경로트리 WSPT T에 포함될 프로세서 중에서 최단거리가 작은 순으로 하나씩 최단 경로를 확정해 가는 것이다[5]. 즉, 처음에는 루트 프로세서 r만으로 구성된 서브트리 $T_r(0)$ = (V", E") (V" = {r}, E" = ϕ)에서 시작하여 매번 하나의 프로세서를 서브트리에 추가해 감으로써, 이 알고리즘에서는 n번의 반복을 통해 최단경로트리 WSPT T를 완성하게 되는데, 각 i(1≤i≤n)단계를 시작하는 상황에서는 WSPT T의 서브트리인 $T_r(h)$에서 시작해서 하나의 프로세서를 추가하여 $T_r(h+1)$로 확장하게 된다. 즉, i = h + 1이 성립한다. 이후, 프로세서 s를 루트로 하는 크기가 m인 서브트리를 Ts(m)로 나타내기로 한다. 단, 서브트리의 크기에 루트는 포함하지 않는 것으로 한다.

$T_r(h)$가 다음 (그림 1)의 (a)와 같이 구성되었다고 하고, 이 때 최단경로 r-x가 가장 작다고 할 때, i번째 단계에서는 프로세서 x를 찾아서 $T_r(h)$에 추가하여 (b)와 같이 $T_r(h+1)$로 확장한다.



(a) 제i번째 단계의 시작시 (b) 제i번째 단계의 종료시

그림 1 분산 Dijkstra 알고리즘의 개략

3.2 알고리즘 SUA

본 논문에서 제안하는 최단경로 재구성분산알고리즘(이하, SUA(Shortest-path Updating Algorithm)라고 한다)은 두 단계로 구성된다.

[1단계] 이중연결성분 재구성

문헌 3에 제안되어 있는 이중연결성분 재구성 알고리즘을 이용하여 토폴로지 변화후의 네트워크에 대한 이중연결성분을 구성한다.

[2단계] 최단경로 재구성

1단계가 종료하게 되면 토폴로지 변화후의 네트워크에 있어서, 이중연결성분과 절단점이 판정되는데, 이때, 트리 T=(V, E)가 구성된다. V={u | 노드 u는 이중연결성분}, E={(u,v) | u와 v는 인접한 이중연결

성분}. 각 이중연결성분은 여러 개의 절단점을 포함하는데, 트리 T에서 루트에 가까운 절단점을 그 이중연결성분의 대표로 한다. 다음 그림에서 빗금친 노드가 이중연결성분의 대표를 나타낸다.

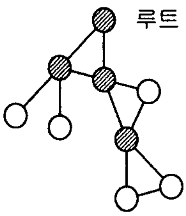


그림 2 이중연결성분과 대표

SUA에서는 루트가 알고리즘을 시작하는데, 루트는 루트가 대표로 되어 있는 이중연결성분 B_r에 대해 Dijkstra의 방법을 이용하여 최단경로를 구성한다. 즉, 루트에서 도달할 수 있는 경로중에서 가장 적은 경로를 하나씩 확정해 나간다. 이들 이중연결성분 B_r에 대한 최단경로 재구성이 완료되면 T상에서 루트가 대표로 되어 있는 이중연결성분 B_r의 아들에 해당하는 인접 이중연결성분 B_{r1}에 대한 최단경로를 재구성하기 위해 이들 이중연결성분 B_{r1}의 대표에게 메시지를 보냄으로써, 이들 대표노드가 Dijkstra알고리즘을 이용하여 이중연결성분 B_{r1}에 대한 최단경로를 재구성한다. 그리고나서, T상에서 거리 2의 위치에 있는 이중연결성분 B_{r2}에 대한 최단경로를 재구성한다. 따라서, T상에서 루트로부터 최장경로의 길이가 s라고 할 때, 상기와 같은 작업을 s번 수행하면, T상에서 가장 멀리 있는 잎 노드(leaf node)에 대한 최단경로 재구성을 하게 되어 토폴로지 변화후의 네트워크에 대한 최단경로를 재구성하게 된다.

[정리] SUA의 메시지복잡도는 $O(B_1^2+B_2^2+\cdots+B_k^2)$이다.

(약증) SUA는 두단계로 구성되어 있으며, 이중연결성분을 재구성하는 1단계에서는 $O(n'+a+b)$의 메시지복잡도를 갖는다[3]. n'는 토폴로지 변화후의 네트워크의 프로세서수, a는 추가 링크수를 나타내고, b는 삭제링크를 포함하는 (토폴로지 변화전의) 이중연결성분에 포함된 링크수의 합계를 나타낸다.

최단경로를 재구성하는 2단계에서는 각 이중연결성분에 있는 절단점이 중심이 되어 그 이중연결성분에 대한 최단경로 재구성작업을 하게 되는데, 이때 Dijkstra의 방법을 이용하므로, B_i개의 프로세서를 갖는 이중연결성분에 있어서 $O(B_i^2)$의 메시지 복잡도로 그 이중연결성분에 대한 최단경로를 재구성하게 된다. 따라서, k개의 이중연결성분에 대한 최단경로를 재구성하는데는 $O(B_1^2+B_2^2+\cdots+B_k^2)$의 메시지 복잡도가 필요하게 된다. □

4. 결론

네트워크 토폴로지 변화후의 새로운 네트워크에 있어서 이중연결성분의 수가 k개(각 연결성분에 속하는 프로세서의 수가 $B_1, B_2, \cdots, B_k$)라고 가정할 때, Dijkstra의 알고리즘을 이용해서 최단경로문제를 해결할 경우, 메시지복잡도는 $O((B_1+B_2+\cdots+B_k)^2)$가 되는데, 본 논문에서는 이중연결성분을 이용하여 메시지복잡도 $O(B_1^2+B_2^2+\cdots+B_k^2)$에 최단경로 재구성문제를 해결할 수 있는 효율적인 분산알고리즘을 제안한다.

참고문헌

[1] B.Awerbuch : "Complexity of network synchronization", Journal of ACM, Vol.32, No.4, 1985.

[2] B. Swaminathan and K.J.Goldman : "An incremental distributed algorithm for computing biconnected components", Proc. 8th International

Workshop on Distributed Algorithm(LNCS 857), 1994.

[3] 박정호, 이창석 : "An Algorithm Solving the Biconnected-components Reconstruction Problem", 한국정보처리학회 논문지 제5권제10호, 1998.10.

[4] 박정호, 박윤용 : "A Distributed Algorithm for Weighted Shortest Path Problem", 한국정보처리학회 논문지 제6권제1호, 1999.1.