

통신 품질 향상을 위한 최단 시간 경로 알고리즘

윤석환 · 김평중

한국전자통신연구소 멀티미디어연구부

김진수

대전산업대학교 산업공학과

Quickest Path Algorithm for Improving Quality of Services in Communication Networks

Seokhwan Yoon · Pyeongjung Kim

Dept. of Multimedia, Electronics and Telecommunications Research Institute

Jinsoo Kim

Dept. of Industrial Engineering, Taejon National University of Technology

Abstract

The quickest path problem is one of the important things for quality of services in communication networks. It is to find a path to send a given amount of data from the source to the sink with minimum transmission time, where the transmission time is dependent on both the capacities and the traversal times of the arcs in the network. This is found under the networks that the capacity and the lead time of each ring are predetermined. It is general to solve the quickest path problem using shortest path algorithms. The relevant algorithms proposed till now are based on the capacity of rings in distributed environments.

When the configuration of networks is changed, there can be two approaches to find the quickest paths. The one is to find new quickest paths, and the other is to update the current quickest paths. As one of the algorithms for the latter, the distributed quickest path update algorithm was proposed.

This paper aims to propose the distributed algorithm applicable to find the quickest path, when the configuration of networks is changed, using the quickest path tree update algorithm, and to verify its possibility of application by analyzing

the transmission amount of data from one node to another from the theoretical point of view.

1. 서론

네트워크에 대한 통신 품질의 중요한 문제의 하나로서 최단 시간 경로 문제(quickest path problem)가 있다[Chen, 1993]. 이 문제는 네트워크의 용량과 전송 시간이 정해졌을 경우, 일정량의 데이터를 최소의 전송 시간으로 보낼 수 있는 경로를 찾는 알고리즘이다.

일반적으로 최단 시간 경로 문제는 최단 경로 문제(shortest path problem)를 이용해서 해결하고 있다[Hung and Chen, 1991]. 서로 다른 링의 각 용량에 대해서 일정 수준 이상의 링만으로 구성되는 부분 네트워크를 고려할 때, 이들 각각의 전송 시간을 기준으로 최단 경로를 구한다. 이러한 경로들 중 전송 시간이 최소가 되는 경로가 최단 시간 경로이다. 이러한 개념을 적용하여 최단 시간 경로를 구하는 알고리즘이 제안되었다[1, 2, 3]. 이러한 분산 알고리즘에서는 링의 용량을 기준으로 최단 경로 트리를 구성하고 있다.

Y. Chen and Y. Chin은 정점 s, t 사이의 최단 경로를 찾기 위한 $O(m^2 + mn \log n)$ 순차적 알고리즘을 제안하였으며[Chen and Chin, 1990], Y. Hung and G. Chen은 모든 정점들 사이의 최단 경로를 찾기 위한 $O(mn^2)$ 알고리즘을 제안하고 이를 비동기식 메시지 패싱 모델에 적용한 분산 알고리즘을 제안하였다[Chen and Hung, 1994]. J.B. Rosen and G.L. Xue는 최악의 수행 시간을 고려하는 알고리즘을 제안하였으며[Rosen and Xue, 1991], 아울러 최단 경로 문제를 동적 기법에 의해 접근하는 알고리즘을 제안하기도 하였다[Rosen et al., 1991]. 增澤利光 등은 이미 구성되어 있는 최단 경로 트리에서 네트워크의 형상이 변할 때, 기존의 최단 경로 트리를 이용하여 새로운 최단 경로를 구할 수 있는 최단 경로 트리 갱신 알고리즘을 제안하였다[增澤利光 외, 1991]. D.A. Dunn et al.은 k 개의 최단 경로에 대한 비교 및 네트워크 장비의 복구를 위한 최대 흐름 라우팅에 관한 알고리즘을 제안하였으며[Dunn et al., 1994], A.V. Goldberg는 최단 경로 문제를 위한 스케이링 알고리즘을 제안하였다[Goldberg, 1995]. T. Shibuya et al.은 현실적 우회 최단 경로를 찾기 위해 인공지능 기법을 활용한 알고리즘을 제안하였으며[Shibuya et al., 1995], E. Ruppert는 가중치를 부여한 가지의 방향성 그래프에서 정점들 사이에 존재하는 k 개의 최단 경로를 찾아내는 병렬 알고리즘을 제안하였다[Ruppert, 1997].

본 논문에서는 최단 경로가 구해져 있는 상황에서 네트워크의 형상이 변할 때, 최단 경로 트리를 계속 새로 구성하는 것이 아니고, 이미 구성되어 있는 최단 경로 트리를 이용하여 새로운 최단 경로를 찾을 수 있는 기존의 최단 경로 트리 갱신 알고리즘[增澤利光 외, 1991]을 개선한 분산 알고리즘을 제안하고 이의 적용 가능성을 통신

전송량 분석을 통하여 확인하는 데에 그 목적이 있다.

2. 네트워크 정의

본 논문의 알고리즘을 전개하는 데에 필요한 네트워크에 대해서 다음의 가정을 한다.

- 네트워크에 공유 메모리는 존재하지 않고, 프로세스 사이의 통신은 링을 매개로 하는 메시지의 교환만으로 행해진다.
- 각 프로세스가 실행하는 프로그램은 동일하다.
- 알고리즘의 실행 중에 프로세스 및 링의 고장은 발생하지 않는다.

네트워크 N 은 4가지의 구성 요소를 포함하는 $N = (V, A, c, l)$ 으로 정의된다. 여기서 V 는 네트워크 내에서 프로세스의 집합을 의미하며, A 는 통신 링의 집합으로서, 서로 다른 프로세스의 순서에 따라 통신 링을 정의한다. 예를 들어, $(u, v) \in A$ 또는 $(v, u) \in A$ 의 경우, 프로세스 u, v 사이에는 양방향 통신이 가능한 통신 링이 존재한다. 비동기식 네트워크일 경우, 프로세스간 통신 지연 시간이 발생하지만 무시한다. c 는 링의 용량을 나타내는데, 이것은 1단위 시간에 그 링을 통해서 보낼 수 있는 최대의 데이터 양이다. 즉, $c(u, v)$ 는 링 $(u, v) \in A$ 의 용량을 나타낸다. 또한, l 은 링의 전송 시간을 표시하는데, 그 링에 데이터를 보낼 때 필요한 시간을 나타낸다. 즉, $l(u, v)$ 는 링 $(u, v) \in A$ 의 전송 시간을 나타낸다.

또한, 네트워크의 프로세스의 개수를 n , 링의 개수를 e 로 나타낸다. 통신 전송량은 알고리즘의 시작부터 종료할 때 까지 프로세스 사이에서 교환되는 메시지의 총 수로 나타낸다.

3. 최단 경로 트리 갱신 알고리즘

3.1 정의

본 논문의 내용 전개에 필요한 기본적인 수식 및 가정은 다음과 같다[1, 2, 7].

[정의 1] $P = (u_1, u_2, \dots, u_k)$ 를 $(u_1, -u_k)$ 의 경로를 나타내고, P 의 전송시간과 용량을 각각

$$l(P) = \sum_{i=1}^{k-1} l(u_i, u_{i+1}) \quad (3-1)$$

$$c(P) = \min_{1 \leq i \leq k-1} c(u_i, u_{i+1}) \quad (3-2)$$

로 정의한다.

이에 따라, 일정량 δ 의 데이터를 u_1 에서 u_k 로 보내는 데에 필요한 전송시간 $t(P, \delta)$ 는 다음과 같이 정리된다.

$$t(P, \delta) = l(P) + \delta/c(P) \quad (3-3)$$

[정의 2] 네트워크 N 에 대하여, N 에 존재하는 프로세스를 각각 s, t 라고 할 때, s 에서 t 로 일정량 δ 의 데이터를 보내는 전송시간 중 가장 짧은 경로가 최단 시간 경로이다. 최단 시간 경로는 일반적으로 복수 개가 존재한다. 그 중 임의의 하나를 $QP(s, t, \delta, N)$ 으로 표시하고, 이의 전송시간을 $QT(s, t, \delta, N)$ 라 하면,

$$\begin{aligned} QT(s, t, \delta, N) &= t(QP(s, t, \delta, N), \delta) \\ &= \min \{ t(P, \delta) \mid P \text{는 } N \text{에서의 } s-t \text{ 경로} \} \end{aligned} \quad (3-4)$$

이 성립한다.

[정의 3] 최단 시간 경로 문제를 다음과 같이 정의한다.

(초기 상태) 임의의 네트워크 N 에서, N 에 존재하는 프로세스 s, t , 그리고 전송할 데이터 전송량 δ 는 지정되어 있다.

(해의 상태) 임의의 네트워크 N 있어서, s 에서 t 로 δ 의 데이터를 보내는 최단 시간 경로 $QP(s, t, \delta, N)$ 를 알고 있다.

결국, $QP(s, t, \delta, N) = (s = u_1, u_2, \dots, u_k = t)$ 중의 각 프로세스 u_i 가 u_{i+1} 로 전송하는 링임을 알 수 있다.

[가정] 최단 시간 경로 문제에 있어서, 프로세스 s 만이 초기 프로세스가 된다.

이 가정은 논리 전개를 간단히 하기 위한 것이다. 임의의 프로세스의 부분 집합이 초기 프로세스가 되는 경우에도 통신 전송량 $O(e)$, 이상적 시간 전송량 $O(n)$ 을 통하여 s 에 알고리즘의 시작을 알려주는 것이 가능하기 때문에 본 논문에서 제안하는 알고리즘의 전송량의 순서에는 영향을 미치지 않는다.

3.2 최단 경로 트리 갱신 알고리즘

최단 경로가 구해져 있는 상황에서 네트워크의 형상이 변할 때, 최단 경로 문제는

다음의 정리와 같이 최단 경로 트리 문제의 알고리즘을 이용하여 해결하는 것이 가능하다.

[정리 1] P_j 를 네트워크 $N(c_j), j=1, 2, \dots, r$ 상의 최단 경로 $s-t$ 라 하고,

$$l(P_i) + \sigma/c(p_i) = \min_{1 \leq i \leq r} \{l(P_j) = \sigma/c(p_j)\} \text{라 하면 [Rosen et al, 1991],}$$

p_i 은 σ 단위의 데이터를 보내는 네트워크 N 에서 최속 경로 $s-t$ 가 된다.

[증명] p 를 σ 단위의 데이터를 보내는 네트워크 N 에서 최속 경로 $s-t$ 라 하면, $c(p) = c_{j_0}, j_0 \in \{1, 2, \dots, r\}$ 이 되며, [정의 1]에 의하여 p 와 p_{j_0} 는 네트워크 $N(c(p))$ 에서의 최단 경로 $s-t$ 가 된다. 그러므로, $l(p_{j_0}) = l(p)$, $c(p_{j_0}) \geq c(p)$ 이 성립하며, $t(\sigma, p_{j_0}) \leq t(\sigma, p)$ 이 성립한다. 전제에 의하여, $t(\sigma, p_i) \leq t(\sigma, p_{j_0})$ 이 되어 $t(\sigma, p_i) \leq t(\sigma, p)$ 이 성립한다. 따라서, p_i 은 σ 단위의 데이터를 보내는 네트워크 N 에서 최속 경로 $s-t$ 가 된다.

[정리 2] 네트워크 N 의 $s-t$ 경로 중 전송 시간이 최소인 것 중의 하나를 $SLP(s, t, N)$ 라 하고, 네트워크 N 을 구성하는 각 링의 서로 다른 용량을 $c_1 < c_2 < \dots < c_r$ 이라고 할 때, 각 $i(1 \leq i \leq r)$ 에 대하여 $N_i = (V, A_i, c, l)$ 이다. 여기서 $A_i = \{(u, v) \in A \mid c(u, v) \geq c_i\}$ 이 된다. 즉, N_i 는 N 에서 용량이 c_i 미만의 링을 제거한 것이다. 그러면,

$$l(SLP(s, t, N)) = \min \{l(P) \mid P \text{는 } N \text{에서의 } s-t \text{ 경로}\} \quad (3-5)$$

이 [정의 1]에 의해 성립하며,

$$QT(s, t, \sigma, N) = \min_{1 \leq i \leq r} \{t(SLP(s, t, N_i), \sigma)\} \quad (3-6)$$

이 성립한다.

[증명] [정의 1]의 식(3.3) 및 [정의 2]의 식(3.4)에 의해

$$\begin{aligned} QT(s, t, \delta, N) &= t(QP(s, t, \delta, N), \delta) \\ &= \min \{t(P, \delta) \mid P \text{는 } N \text{에서의 } s-t \text{ 경로}\} \end{aligned}$$

이 되며, 여기에 [정리 1]을 적용하면

$$\begin{aligned} &= l(P_i) + \delta/c(p_i) \\ &= \min_{1 \leq i \leq r} \{l(p_j) = \sigma/c(p_j)\} \end{aligned}$$

$$= \min_{1 \leq i \leq r} \{t(SLP(s, t, N_i), \sigma)\} \text{이 성립한다.}$$

이것을 이용하여 최단 경로 문제를 다음과 같이 풀 수 있다. 각 $i(1 \leq i \leq r)$ 에 대하여 N_i 의 전송 시간을 가중치(weight)로 간주 함으로써, s 를 정점으로 하는 최단 경로 트리 ST_i 를 구하고, 각 ST_i 에 대하여 $s-t$ 경로 P_i 의 전송 시간을 구한다. 이어서, $P_i(1 \leq i \leq r)$ 중에서 전송 시간을 최소로 하는 $s-t$ 경로를 최단 시간 경로로 정의함으로써 가능하다.

이와 같은 방법을 통하여, r 개의 최단 경로 트리를 만드는 부분이 알고리즘 전체의 성능을 결정한다. 이것은 이미 알려진 최단 경로 트리 구성 알고리즘을 r 회 반복해서 이용한다[Y. Hung & G. Chen, 1991]. 이와 같이, 각 최단 경로 트리를 하나씩 고치는 방법도 있지만, 각각의 최단 경로 트리가 크게 변하지 않을 경우, 즉, 정점에서부터 각 최단 경로 트리에 이르는 거리가 크게 변하지 않을 경우, 최단 경로 트리 갱신 알고리즘을 이용하면 효율적이다[增澤利光 外, 1991]. 이것은 네트워크에 형상 변화(프로세스와 링의 추가 및 제거)가 생긴 경우, 그 변화된 내용을 최단 경로 트리에 반영 함으로써 트리를 갱신하는 알고리즘이다. 이것은 트리를 새로 구성하는 것이 아니고, 이미 구성되어 있는 최단 경로 트리를 이용함으로써 최단 경로 트리 갱신 문제를 처리할 수 있다.

4. 최단 경로 트리 갱신 알고리즘을 이용한 분산 알고리즘

4.1 제안된 알고리즘

최단 경로 트리 갱신 알고리즘을 이용하여 최단 경로 문제를 풀이하는 분산 알고리즘은 다음의 3단계로 구성된다.

[단계 1] N_1 에 대하여 s 를 정점으로 하는 최단 경로 트리 ST_1 을 구한다.

[단계 2] $i(1 \leq i \leq r-1)$ 에 대하여 다음을 수행한다. 최단 경로 트리 갱신 알고리즘을 이용하여 ST_i 를 갱신한 ST_{i+1} 을 구한다. 이 때, N_{i+1} 은 N_i 에서 용량이 C_i 인 링을 제거한 것이므로 최단 경로 트리 갱신 알고리즘을 이용한다. ST_{i+1} 에 있어서, $s-t$ 경로 P_{i+1} 에 대하여 $t(P_{i+1}, \delta) = l(P_{i+1}) + \delta/c(P_{i+1})$ 를 구한다(프로세스 t 에 저장). 또한, 경로 P_{i+1} 에 나타난 각 프로세스는 ST_{i+1} 에 있어서 중심이 되는 경로를 기억한다.

[단계 3] 프로세서 t 가 기억하고 있는 모든 $t(P_i, \delta)$ ($1 \leq i \leq r$) 중에서 최소치를 부여하는 i 를 구하여 j 로 한다. 그 후 P_j 에 나타나는 각 프로세스에 대해

P_j 가 최단 시간 $s-t$ 경로인 것을 알린다. 마지막으로, s 가 모든 프로세스를 정지시키기 위한 메시지를 보낸다.

4.2 통신 전송량 분석

[정리 3] 최단 시간 경로 문제를 풀이하는 알고리즘의 통신 전송량은 $O(\sqrt{rnDe} + re)$ 이다. 여기에서 n, e 는 네트워크에서 각각 프로세스 수와 링의 개수를 나타내고 r 은 용량의 종류를 나타낸다. 또한, $0 \leq i \leq r-1$ 에 대하여 ST_i 와 ST_{i+1} 에 있어서 각 프로세서의 정점으로부터 거리 변화량의 최대치를 d_i 라 하면 (d_0 에 대해서는 ST_0 에 있어서 각 프로세서의 정점에서 거리 최대치로 함), $D = d_0 + d_1 + \dots + d_{r-1}$ 이다.

[증명] 최단 경로 트리 구성 알고리즘의 통신 전송량은 $O(\sqrt{nd_0e} + e)$ 이며, 최단 경로 트리 갱신 알고리즘의 통신 전송량은 $O(\sqrt{nde} + e)$ 이다(d : 각 프로세서의 갱신 전후의 최단 경로 트리에서 정점으로부터의 거리 변화량의 최대치)[增澤利光외, 1991].

이로부터, 각 네트워크에서의 프로세서 수에 따른 통신 전송량을 $O(n)$ 으로 나타내면, 알고리즘 전체의 통신 전송량은 $\sum_{i=0}^{r-1} O(\sqrt{nd_i e} + e)$ 이다. $D = d_0 + d_1 + \dots + d_{r-1}$ 이며, 이 값은 $d_0 = d_1 = \dots = d_{r-1} = D/r$ 일 때 최대가 되어,

$$\begin{aligned} \sum_{i=0}^{r-1} O(\sqrt{nd_i e} + e) &= O\left(\sum_{i=0}^{r-1} (\sqrt{nd_i e} + e)\right) \\ &\leq O\left(\sum_{i=0}^{r-1} (\sqrt{nDe} + e)\right) \\ &= O(\sqrt{rnDe} + re) \end{aligned} \quad (4-1)$$

이 성립한다.

본 논문에서 제안하는 알고리즘은 $D = O(rn^3/e)$ 의 경우 기존의 최단 경로 트리 문제의 알고리즘보다 통신 전송량에 있어서 효율이 좋다.

또한, 링 용량의 최대치를 정수로 할 때, 최단 경로 트리 갱신 알고리즘의 통신 전송량은

$$O(\min(\sqrt{nde}, d^{1/3} n^{4/3}) + e) \quad (4-2)$$

이 되며, 링 전송 시간의 최대치를 정수로 할 때, 제안하는 알고리즘의 통신 전송량은

$$O(\min(\sqrt{rnDe}, r^{2/3}D^{1/3}n^{4/3}) + re) \quad (4-3)$$

이 된다.

링의 전송 시간이 정수의 경우, 각 $d_i(0 \leq i \leq r-1)$ 는 아무리 커도 $O(n)$ 이다. 그러므로, $D = O(m) \leq O(m^3/e)$ 의 경우보다, 링의 전송 시간의 최대치가 정수일 때, 제안하는 알고리즘은 통신 전송량에 관하여 [2, 6, 8]에서 제시하는 알고리즘보다 효율이 좋다.

5. 결론

본 논문에서는 최단 경로가 구해져 있는 네트워크에서, 네트워크의 형상이 조금 변할 때에 최단 경로를 새로 구하는 것이 아니고, 기존의 최단 경로 트리를 갱신함으로써 새로운 최단 경로를 효율적으로 찾을 수 있는 알고리즘을 제안하였으며, 이의 적용 가능성을 통신 전송량 분석을 통한 이론적 관점에서 확인하였다. 이 알고리즘을 적용하면, 네트워크의 형상이 조금 변하는 경우에 기존의 최단 경로 트리를 이용함으로써 어떤 하나의 프로세스에서 다른 모든 프로세스로의 최단 시간 경로를 구하는 것이 가능하다. 다만, 설정한 최단 경로를 각 프로세스가 알게 하기 위해서는 메시지 수가 $O(n^2)$ 만큼 증가하는 단점이 있다.

향후에는 메시지 수를 감소할 수 있는 방안, 통신 전송량의 개량, 이상적 시간 전송량도 고려한 알고리즘이 개발되어야 한다고 생각하며, 이들에 대한 이론적 접근 뿐 아니라 실증적 자료 분석을 통한 시뮬레이션 연구도 수반되어야 할 것이다. 또한, 최단 시간 경로 트리의 갱신 문제를 고려하는 것도 의미가 있다고 하겠다.

참고문헌

- [1] Chen Y.L. and Chin Y.H.(1990), The quickest path problem, *Computer and Operations Research*, Vol. 17, pp. 153-161.
- [2] Hung Y.C. and Chen G.H.(1991), The quickest path problem in distributed computing systems, *Proceedings of 5th International Workshop on Distributed Algorithms(LNCS 579)*, pp. 181-192.
- [3] Hung Y.C. and Chen G.H.(1991), On the quickest path problem, *Proceedings of International Conference on Computing and Information(LNCS 497)*, pp. 44-46.
- [4] Rosen J.B., Sun S.Z. and Xue G.L.(1991), Algorithms for the quickest path

- problem and the enumeration of quickest path, *Computer and Operations Research*, Vol. 18, No. 6, pp. 579-584.
- [5] Rosen J.B. and Xue G.L.(1991), Sequential and distributed algorithms for the all pairs quickest path problem, *Proceedings of International Conference on Computing and Information(LNCS 497)*, pp. 471-473.
- [6] 増澤利光, 三浦康史, 朴政鎬, 都倉信樹(1991), ネットワークにおける最短経路木構成問題と幅優先生成木更新問題, 信學技報, COMP91-6(1991-04).
- [7] A. Billionnet and S. Elloumi(1995), An Algorithm for Finding the K-best Allocations of a Tree Structured Program, *Journal of Parallel and Distributed Computing*, Vol. 26, pp. 225-232.
- [8] Y.L. Chen(1993), An Algorithm for Finding the k Quickest Paths in a Network, *Computers and Operations Research*, Vol. 20, No. 1, pp. 59-65.
- [9] G.H. Chen and Y.C. Hung(1994), Algorithms for the Constrained Quickest Path Problem and the Enumeration of Quickest Paths, *Computers and Operations Research*, Vol. 21, No. 2, pp. 113-118.
- [10] D.A. Dunn, Wayne D. Grover and M.H. MacGregor(1994), Comparison of k -shortest Paths and Maximum Flow Routing for Network Facility Restoration, *IEEE Journal of Selected Areas in Communications*, Vol. 12, No. 1, pp. 88-99.
- [11] Andrew V. Goldberg(1995), Scaling Algorithms for the Shortest Paths Problem, *SIAM Journal of Computing*, Vol. 24, No. 3, pp. 494-504.
- [12] Eric Ruppert(1997), Finding the k Shortest Paths in Parallel, *Proceedings of 14th Symposium on Theoretical Aspects of Computer Science*, LNCS, Vol. 1200, Feb.
- [13] T. Shibuya, T. Ikeda and H. Imai(1995), Finding a Realistic Detour by AI Search Techniques, *Proceedings of 2nd Intelligent Transportation Systems*, Vol. 4, pp. 2037-2044.
- [14] E.I. Chong, S. Maddila and S. Morley(1995), On Finding Single-Source Single-Destination k Shortest Paths, *Proceedings of 7th International Conference of Computing and Information*, July 1995, pp. 40-47.