

이종 라우팅 메커니즘을 위한 quickest path 기반 통합 라우팅 알고리즘

Quickest Path Based Integrated Routing Algorithms for Different Network Router Mechanisms

방영철*
Young-Cheol Bang
정성택**
SungTaek Chung

요약

링크상에 대역폭과 지연 제약들이 존재하는 네트워크 상에서 Quickest Path는 소스에서 목적지 까지 크기가 σ 인 메시지를 전송하는데 최소지연을 보장한다. 본 논문은 Circuit Switching, Stop and Forward, 그리고 이것들의 조합으로 이루어진 메커니즘을 반영하는 노드들에서의 메시지 전달을 위한 6가지의 전송 모드들을 고려한다. 처음 5개의 각 모드에 대해, 크기가 σ 인 메시지의 전송을 위한 Quickest Path를 계산하는 $O(m^2 + mn\log n)$ 시간 알고리즘을 소개한다. 그리고 마지막 모드에 대해 $O(m+n\log n)$ 시간 안에 Quickest path를 계산하는 알고리즘을 소개한다.

Abstract

The quickest path problem deals with the transmission of a message of size σ from a source to a destination with the minimum end-to-end delay over a network with bandwidth and delay constraints on the links. We consider two basic modes and four variations for the message delivery at the nodes reflecting the mechanisms such as circuit switching, Internet protocol, and their combinations. For each of first five modes, we present $O(m^2 + mn\log n)$ algorithm to compute the quickest path for a given message size σ . For the last mode, the quickest path can be computed in $O(m+n\log n)$ time.

☞ Keyword : Quickest Path, Router, 전송모드, 대역폭, 흐름율

1. 서론

n 개의 노드와 m 개의 간선 또는 링크를 갖는 그래프 $G=(V, E)$ 에 의해 표현되는 네트워크를 고려하자. 각 링크 $e=(i, j) \in E$ 는 대역폭 $B(e) \geq 0$ 과 링크지연 $D(e) \geq 0$ 을 가진다. 메시지는 $f_e \leq B(e)$ 로 표현되는 일정한 흐름율(flow rate)로 간선 e 를 따라 연속적인 스트림 형태로 전송 된다. 크기가 σ 인 메시지는 $\sigma/f_e + D(e)$ 시간

으로 흐름율이 f_e 인 간선 e 를 통해 보내질 수 있다. 흐름율은 다른 간선에서 달라질 수 있으며 메시지는 노드에서 지연될 수 있다.

$(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$ 로서 주어지는 $s = v_0$ 에서 $d = v_k$ 까지의 단순 경로 P 를 고려하자. 여기서, $(v_j, v_{j+1}) \in E, j = 0, 1, \dots, (k-1)$ 이며, v_0, v_1, \dots, v_k 는 서로 유일하다. P 를 통해 전송되는 메시지가 경험하는 지연은 중간에 있는 노드들에서 사용되는 메시지 전송 메커니즘에 의존한다. 경로 P 의 노드 v 에서 $B_{in}(v)$ 와 $B_{out}(v)$ 는 각각 incoming과 outgoing 간선의 대역폭이고, $f_{in}(v)$ 와 $f_{out}(v)$ 는 각각 Incoming과 Outgoing 간선의 흐름율이라 가정하고 아래 4개의 기본 모드와 2개의 변형을 고려하자. 그림 1

* 정회원 : 한국산업기술대학교 컴퓨터공학과 교수
ybang@kpu.ac.kr (제1저자)

** 정회원 : 한국산업기술대학교 컴퓨터공학과 교수
unitaek@kpu.ac.kr

[2005/06/20 투고 - 2005/07/18 심사 - 2005/09/16 심사완료]

에서 모든 노드들에 중간에 있는 노드 v 에 있는 메시지의 도착 프로세스와 출발 프로세스의 타이밍 다이어그램을 보여주고 있다. P 의 전달지연(propagation delay)은 $D(P) = \sum_{j=0}^{k-1} D(e_j)$ 이며, $e_j = (v_j, v_{j+1})$ 이다.

I. Circuit Switching: 메시지는 중간 노드에서 버퍼링 없이 s 에서 t 까지 일정한 비율로 전송된다. 모드 I에서 P 의 대역폭은 $B^I(P) = \min_{j=0,1,\dots,k-1} B(e_j)$ 이며 크기가 σ 인 메시지를 전송하기 위한 경로 P 의 종단간지연은 $T^I(P) = \sigma/B^I(P) + D(P)$ 이다. 따라서, P 의 모든 e 에 대해 $f_e = B^I(P)$ 이며 소스와 목적지를 제외하고 P 의 모든 v 에 대해 $f_{in}(v) = f_{out}(v) = B^I(P)$ 이다.

II. Earliest Departure: 중간노드 v 에 수신된 메시지는 Incoming 흐름율과 Outgoing 대역폭 중 작은 비율로 내보내진다. 즉, $f_{out}(v) = \min f_{in}(v)$, $B_{out}(v)$ 이다. 따라서, 만약 Outgoing 대역폭이 Incoming 흐름율보다 작으면, 메시지는 적당하게 버퍼링되고 어떠한 지연 없이 outgoing 대역폭으로 보내진다. 모드 IIa에서 더 낮은 대역폭으로의 재전송은 전체 메시지가 $f_{in}(v) > B_{out}(v)$ 인 조건 하에, v 에 완전히 수신된 후에만 시작한다. 그러나, incoming으로의 재전송은 $f_{in}(v) \leq B_{out}(v)$ 인 조건에서 지연이 없다.

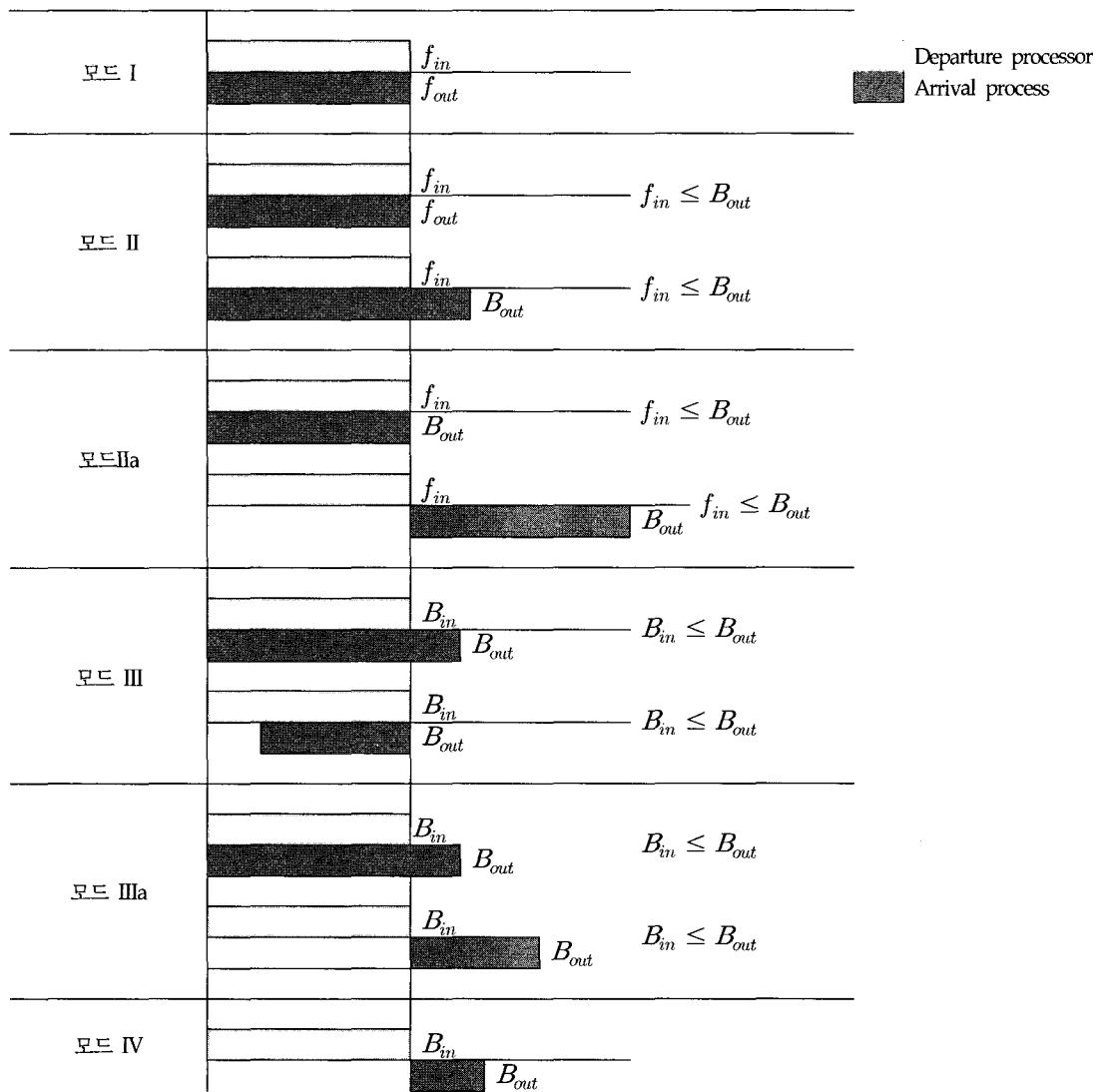
III. Full Outgoing 대역폭: 이 모드에서 각 간선에서의 흐름은 그 간선의 대역폭과 같다. 즉, P 에 있는 모든 e 에 대하여 $f_e = B(e)$ 이다. 중간노드 v 에 수신된 메시지는 다음과 같이 재전송 된다; (a) 만약 $f_{in}(v) = B_{in}(v) \geq B_{out}(v)$ 이면, 메시지는 $B_{out}(v)$ 의 흐름율로 즉시 전송된다. (b) 만약 $f_{in}(v) = B_{in}(v) < B_{out}(v)$ 이면,

메시지는 가능한 곧 $B_{out}(v)$ 율로 전송 될 수 있도록 버퍼링 되며 적당한 지연이 발생한다(흐름율은 늘어날 수 있다.). 모든 IIIa는 outgoing 대역폭이 Incoming 흐름율보다 클 경우를 제외하고는 III과 같다. 이 경우 메시지는 $B_{out}(v)$ 율로 전송되기 전에 v 에 완전히 버퍼링 된다.

IV. Store-and-Forward: 간선 (u, v) 를 따라 전송되는 메시지는 v 로부터 전송되기 전에 v 에 메시지 전체가 수신될 것이다. 이 경우 P 에서 모든 e 에 대해 $f_e = B(e)$ 이다. 타입 IV에서 P 의 대역폭은 $B^{IV}(P) = \frac{1}{\sum_{j=0}^{k-1} \frac{1}{B(e_j)}}$ 이며 크기가 σ 인 메시지를 경로 P 로 전송 할 때 모드IV에서의 종단간지연은 $T^{IV}(P) = \sigma/B^{IV}(P) + D(P)$ 이다.

$i = I, II, IIa, III, IIIa, IV$ 에 대하여, 모드 i 의 경로는 i -path라 하고, 모든 i -path 중에서 종단간지연이 가장 작은 경로를 i -quickest path라 하자. 이때, P_σ^i 는 크기 σ 의 메시지 대해 모드 I에서의 i -quickest path라 한다.

이러한 전송 모드들은 데이터 네트워크에서 사용되는 다양한 라우팅 기법을 반영한다. 또한, 각 모드에서 링크는 고정된 흐름율로 전송을 수행한다. 모드 I은 버퍼링을 포함하지 않는 전형적인 Circuit Switching에 대응 되며, 모드 IV는 메시지가 모든 중간 노드마다 완전하게 버퍼링 된다는 점에서 다른 극단적인 방법을 나타낸다. 전화 네트워크는 모드 I에 속하며 IP(인터넷 프로토콜) 컴퓨터 네트워크는 모드 IV에 속한다. ATM (Asynchronous Transmission Mode)과 액티브 네트워크, 오버레이 네트워크 기술의 출현과 더불어 이 모드들의 조합이 사용되어 다양한 모드들이 나오게 된 계기가 됐다. 모드II에서 메시지는 충분한 Outgoing 대역폭이 있으면 Circuit Switching 되고, 그렇지 않으면 $f_{in}(v)$ 의



〈그림 1〉 다양한 모드의 중간노드에 arrival과 departure timing 다이어그램

감소된 흐름율로 전송 된다. 이러한 Circuit switching에서의 흐름율의 감소는 매우 어려운 작업이며, 모드 IIa의 경우처럼, 최소한 재전송되기 전 메시지가 완전히 수신되어야 할 때까지의 지연이 발생한다. 모드 III과 모드 IIIa의 주된 초점은 간선에서 대역폭의 Fragmentation을 피하는 것이고, 그렇게 함으로서 메시지가 간선을 따라 전송되는 동안 각 간선의 전체 대역폭이

활용된다.

II-path는 최소대역폭을 갖는 간선에 대응되는 흐름율을 활용하여 I-path로 변경 될 수 있다. I-path는 더 낮은 대역폭을 가진 간선에 도착된 후 흐름율이 감소되기 전까지 s부터 시작되는 흐름율의 적절한 증가와 P를 따라 되풀이하여 이동함으로써 II-path로 변경될 수 있다. 흐름율이 축소되는 동안 경로 P의 종단간 지연은 동일

Path	Mode I	Mode II	Mode IIa	Mode III	Mode IIIa	Mode IV
P1	$\frac{\sigma}{1} + 5$	$\frac{\sigma}{1} + 5$	$\frac{\sigma}{4/5} + 5$	$\frac{\sigma}{1} + 5$	$\frac{\sigma}{4/5} + 5$	$\frac{\sigma}{2/3} + 5$
P2	$\frac{\sigma}{4} + 6$	$\frac{\sigma}{4} + 6$	$\frac{\sigma}{20/7} + 6$	$\frac{\sigma}{20/6} + 6$	$\frac{\sigma}{20/7} + 6$	$\frac{\sigma}{5/2} + 6$

하게 유지된다. 일반적으로 경로 P 에서 다양한 모드의 종단간 지연에 대해 다음과 같은 부등식을 갖는다.

$$T^I(P) = T^{II}(P) \leq T^{III}(P) \leq T^{IIIa}(P) \leq T^{IV}(P) \text{ 과}$$

$$T^{II}(P) \leq T^{IIIa}(P) \leq T^{IV}(P)$$

예 1.1: s 에서 t 까지 두개의 경로 P_1 과 P_2 를 갖는 그림 2(a)의 네트워크를 고려하자. 그림 2에서 링크 상에 있는 숫자 중 첫번째 숫자는 링크의 대역폭이고, 두 번째 숫자는 링크 지연이다. 다양한 모드 하에 경로의 종단간 지연은 아래의 테이블에 나타나 있다.

이 경우 $P=P_1$, P_2 에 대해 $T^{III}(P) \leq T^{IIIa}(P) = T^{IIIa}(P)$ 을 가진다. 하지만, 일반적으로 모드 IIa와 III(또는 IIIa)에서 종단간 지연은 아래에서 설명된 것처럼 순서를 따르지 않는다. 그림 2(b)의

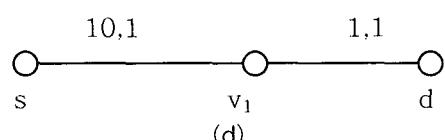
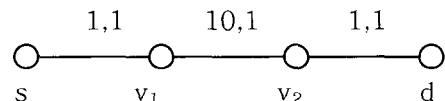
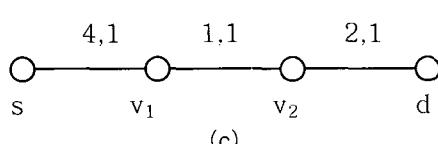
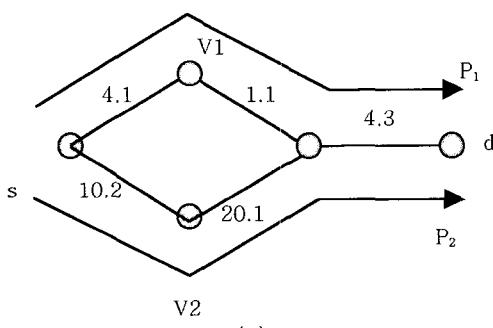
네트워크에서 $T^{IIIa}(P) = \sigma + 3$, $T^{III}(P) = \frac{\sigma}{10/19} + 3$, $T^{IIIa}(P) = \frac{\sigma}{1/2} + 3$ 과 같다. 따라서,

$$T^{IIIa}(P) < T^{III}(P) < T^{IIIa}(P) \text{ 이다. } \square$$

그림 2(c)의 네트워크에서 $T^{IIIa}(P) = \frac{\sigma}{4/5} + 3$, $T^{III}(P) = \frac{\sigma}{1} + 3$, $T^{IIIa}(P) = \frac{\sigma}{2/3} + 3$ 이며, 따라서

$$T^{III}(P) < T^{IIIa}(P) < T^{IIIa}(P) \text{ 이다.}$$

그림 2(d)의 네트워크에서 $T^{IIIa}(P) = \frac{\sigma}{10/11} + 2$, $T^{III}(P) = \frac{\sigma}{1} + 2$, $T^{IIIa}(P) = \frac{\sigma}{1} + 2$ 이다. 따라서, $T^{III}(P) < T^{IIIa}(P) < T^{IIIa}(P)$ 이다. \square



〈그림 2〉 Example 1.1에 대한 네트워크 실례

모드 I은 Chen과 Chin [1], Rosen et al [5], 그리고 Rao와 Batsell [3]에 의해 Quickest Path Problem 이라는 제목으로 연구되어 오고 있다. 어떤한 메시지 사이즈 σ 에 대하여, I-quickest path는 $O(m^2 + mn \log n)$ 시간 안에 계산될 수 있다. 하지만 다른 모드들을 다루기 위해 I-quickest path를 확장하는 직접적인 방법이 없는 것처럼 보인다. 본 논문에서 II-, IIa-, IIa-quickest path는 I-quickest path와 같은 시간복잡도로, IV-quickest path는 간선 e 의 비용으로 $r/B(e) + D(e)$ 를 활용하는 다익스트라 Shortest Path 알고리즘을 사용하여 $O(m + n \log n)$ 시간에 계산될 수 있음을 보여준다.

2. Quickest Path의 계산

본 절에서는 주어진 메시지 크기 σ 에 대한 II-quickest path의 종단간 지연을 계산하기 위한 알고리즘 Quick II(σ)를 설명한다. 경로 P_σ^{II} 자체는 다익스트라 알고리즘의 경우에서와 같이 선행자 포인터를 적절하게 관리함으로써 구성될 수 있다. 다른 모드(모드 IV제외)를 위한 Quickest Path는 이 알고리즘을 조금 변형시켜 계산될 수 있다.

b_1, b_2, \dots, b_c 는 $e \in E$ 인 대역폭 $B(e)$ 의 각각의 값을 나타낸다고 하자. 각 노드 v 는 $TE[v][b]$ 와 같은 배열 $TE[v][\cdot]$ 로 나타내며, 여기서 $b \in SBW$ 이고, $SBW = b_1, b_2, \dots, b_c$ 는 항상 유한 집합이다. $TE[v][b]$ 는 메시지 스트림의 끝 부분이 s 부터 v 까지 어떤 경로를 따라 흐름을 b 로 v 에 도착한 시간이다. (s 와 v 사이에 있는 노드들의 흐름률은 이 모드에서 적어도 b 가 되어야 한다.) 메시지를 인도하는 간선은 중간 노드에서 지연되지 않기 때문에 한번 감소된 흐름률은 이 값을 유지하거나 그 후에 좀 더 축소될 것이다. 결과적으로 메시지가 흐름률 b 로 목적지에 수신되면 $T^{II}(P) = \sigma/b + D(P)$ 이다.

```

algorithm Quick - II( $\sigma$ )
1.  $A \leftarrow 0$ ;
2. for each vertex  $v$  in  $V - \{s\}$  do
3.   for  $b \in SBW = \{b_1, b_2, \dots, b_c\}$  do
4.      $TE[v][b] \leftarrow \infty$ ;
5.      $A \leftarrow A \cup \{(v, b)\}$ ;
6.     if  $v \in Adj(s)$  then  $TE[v][B(s, v)] \leftarrow \sigma / B(s, v) + D(s, v)$ ;
7.   while  $A \neq \emptyset$  do
8.     choose  $(v, b) \in A$  such that  $TE[v]/b$  is minimum;
9.     for each  $w$  in  $Adj(v)$  do
10.       if  $b \leq B(v, w)$  then
11.          $TE[w][b] \leftarrow \min\{TE[w][b], TE[v][b] + D(v, w)\}$ ;
12.       else
13.          $TE[w][B(v, w)]$ 
            $\leftarrow \min\{TE[w][B(v, w)],$ 
            $TE[v][b] - \sigma / b + D(v, w) + \sigma / B(v, w)\}$ ;
14.      $A \leftarrow A - \{(v, b)\}$ ;
15. return  $\min_{b \in SBW}\{TE[d][bj]\}$ ;

```

알고리즘 II. II-quickest path 계산을 위한 알고리즘

알고리즘의 개요는 다음과 같다. Main Loop (8-14라인)으로의 초기 엔트리에 대해 집합 A 는 모든 $v \in V - s$ 와 $b \in SBW$ 에 대한 모든 쌍(v, b)를 포함한다. 알고리즘은 메인 루프의 (n-1) c 번 반복을 포함하며, 각 반복에서 노드 v 와 대역폭 b 가 선택되고 $T[v][b]$ 에 의해 표현되는 s 에서 v 까지 경로 P_v 는 $(v, b) \in A$ (라인8)에서 대응하는 어떠한 경로보다 작은 종단간 지연을 가진다. $Adj(v)$ 는 모든 w 의 집합을 나타내며, $(v, w) \in E$ 이다. v 가 일단 선택되면 모든 정점 $w \in Adj(v)$ 은 P_v 가 간선 (v, w) 를 추가 함으로서 노드 w 까지 확장되었는지 아닌지를 알아보기 위해 검사되며, $T[w][b]$ (라인 9-13)의 현재 값보다 더 작은 종단간 지연을 얻을 수 있다. 특히, (v, w) 의 대역폭이 b 보다 크거나 같으면, b 의 흐름율이 간선 (v, w) (라인 11)에 사용되며, 그렇지 않다면 $B(v, w)$ 의 흐름율이 사용된다(라인 13). 어떠한 경우든 v 에서 w 까지 경로의 확장은 b 보다 크지 않은 w 로의 흐름율을 야기시킨다. 모든 $w \in Adj(v)$ 가 검사되면, (v, b) 는 A 로부터 제거되고 라인 8에서 더 이상 고려되지 않는다.

Quick-II의 정확한 증명에 사용될 몇몇의 준비 과정을 살펴보자. 경로 $P = s, v_1, v_2, \dots, v_p, v_{p+1}$ 에 대해 $T_{p,\sigma}[v_i]$ 는 메시지 크기 σ 에 대한 s 에서 v_i 까지 부경로(subpath)의 종단간 지연을 나타내며 $\widehat{TE}[v][b]$ 는 (v, b) 가 라인 8에서 선택될 때 $TE[v][b]$ 의 값을 나타낸다고 하자. 어떤 경로 $P = s, v_1, v_2, \dots, v_p, v_{p+1}$ 에서, $i < j$ 인 경우, $T_{p,\sigma}[v_i] \leq T_{p,\sigma}[v_j]$ 를 갖는다. 라인 8에서 (v, b) 의 다양한 선택에 의해 생성되는 $\widehat{TE}[v][b]$ 값들의 순서는 내림차순이 아니며, 라인 4, 6, 11, 13에서 생성되는 모든 (v, b) 에 대한 $TE[v][b]$ 의 순서는 오름차순이 아님을 쉽게 알 수 있다. 각 유한한 $\widehat{TE}[v][b]$ 는 s 에서 v 까지 어떤 II-path의 종단간 지연에 대응되며 Quick-II에 의해 계산되며 이러한 관찰은 다음과 같은 결과를 제공한다.

Lemma 2.1 $P^* = s, v_1, v_2, \dots, v_p, v_{p+1}$ 가 v_i 로의 흐름율이 f_i 인 s 에서 v_{p+1} 까지의 II-quickest path라면, Quick-II가 실행되는 동안 모든 $i=1, 2, \dots, p+1$ 에 대해 $TE[v_i][f_i] \geq T_{p,\sigma}[v_i]$ 이다.

Theorem 2.1 알고리즘 Quick-II(σ)는 s 에서부터 다른 모든 노드까지 크기 σ 의 메시지를 전송하는 II-quickest path의 종단간 지연을 $O(m^2 + mn \log n)$ 시간 안에 계산한다.

Proof: s 에서 t 까지의 경로가 없다면, $TE[d][b]$ 는 어떤 b 에 대해서 ∞ 로부터 축소되지 않을 것이다. 경로가 있다면, $P^* = s, v_1, v_2, \dots, v_p, v_{p+1}$ 가 $s = v_0$ 이고, $v_{p+1} = d$ 인 경우 메시지 크기 σ 에 대한 II-quickest path라고 가정하자. f_i 는 P 에 부합하는 v_i 에서의 Incoming 흐름율이라 하자. 모든 $i=1, 2, \dots, p+1$ 에 대해 $\widehat{TE}[v_i][f_i] = T_{p,\sigma}[v_i]$ 임을 보일 것이다. 이 조건은 알고리즘의 실행 동안 $\widehat{TE}[v_i][f_i] > T_{p,\sigma}[v_i]$ 이기 때문에

알고리즘의 종료에서 $TE[v_i][f_i] = T_{p,\sigma}[v_i]$ 를 의미한다.

위의 관측과 Lemma 2.1로부터, 어떤 $1 \leq i < j \leq p+1$ 에 대해 $\widehat{TE}[v_i][f_i] \geq T_{p,\sigma}[v_j] \geq T_{p,\sigma}[v_i]$ 이다.

(v_i, f_i) 전에 (v_i, f_i) 가 선택되면, \widehat{TE} 값들의 단조성에 의해 우리는 $\widehat{TE}[v_i][f_i] \leq \widehat{TE}[v_i][f_i]$ 이다. 게다가 조건 $\widehat{TE}[v_i][f_i] \leq T_{p,\sigma}[v_i]$ 가 만족한다면, $TE[v_i][f_i] \leq T_{p,\sigma}[v_i]$ 이다.

귀납의 기준으로서 라인 6에서 초기화 된 것처럼 $\widehat{TE} = T_{p,\sigma}[v_i]$ 에 주의하라. 귀납적 가설을 위하여 $\widehat{TE}[v_i][f_i] = T_{p,\sigma}[v_i]$ 이라 가정하자. 만약 (v_{i+1}, f_{i+1}) 이 (v_i, f_i) 전에 라인 8에서 선택되면, 앞의 단락으로부터 $(v_{i+1}, f_{i+1}) = T_{p,\sigma}[v_{i+1}]$ 임을 알 수 있다. 만약 (v_{i+1}, f_{i+1}) 이 (v_i, f_i) 이후에 선택되면, $TE[v_{i+1}][f_{i+1}]$ 의 값은 (v_{i+1}, f_{i+1}) 이 라인 8에서 계속하여 선택될 때 라인 11-13에서 $T_{p,\sigma}[v_{i+1}]$ 의 값으로 될 것이다. 귀납법에 의하여 $\widehat{TE}[v_{p+1}][f_{p+1}] = T_{p,\sigma}[v_{p+1}]$ 이며, 이는 II-quickest path의 지연 시간이다.

이 알고리즘은 본질적으로 기껏해야 다익스트라 알고리즘의 $c \leq m$ 인스턴스를 포함하며, 인스턴스는 인터리브 되는 단계들과 함께 $b \in SBW$ 알고리즘에 각 대응한다. 시간 복잡도는 다익스트라 알고리즘의 시간복잡도를 정확히 따른다. ■

모드IIa를 위한 알고리즘은 아래의 명령문으로 Quick-II의 라인 13을 교체함으로써 얻어진다.

13'. $TE[w][B(v, w)] \leftarrow \min \{ TE[w][B(v, w)], TE[v][b_v] + D(v, w) + \sigma / B(v, w) \}$

$T^I(P) = T^{II}(P)$ 인 어떤 경로 P 에 대하여 위에서처럼, Quick-II는 I-quickest 경로를 결정하기 위해 사용될 수 있다. 그러나, 이런 I-paths에 따른 중간 노드들로의 도착 시간은 Quick-II에

의해서 계산되는 $TE[v][b]$ 의하여 주어지지 않는다. 모드 I에 대한 알고리즘인 Quick-I은 라인 6을 6'으로 교체하여 라인 3에서 시작하는 반복문안으로 이동시키고, Quick-II의 라인 10-13을 아래의 명령문으로 교체함으로써 얻는다.

if $b \leq B(v, w)$ then

$TE[w][b] \leftarrow \min\{TE[w][b], TE[v][b] + D(v, w)\};$

6'. if $v \in adj(v)$ and $b \leq B(s, v)$ then
 $TE[v][b] \leftarrow \sigma/b + D(s, v);$

알고리즘 Quick-I는 [5]의 알고리즘의 변형으로 볼 수 있다. 이는 다익스트라 알고리즘의 c개의 인스턴스를 기반으로 하며, 별도로 실행 된다. $G(b)=(V, E(b))$ 는 $B(e) \geq b$ 와 $\exists e \in E(b)$ 가 필요충분조건을 만족할 경우 서브네트워크를 나타낸다. $G(b)$ 에서 $s - t$ 경로는 링크지연만을 기반으로 하는 shortest delay path를 나타낸다고 하자. [5]에서 $s - t$ 경로는 각각의 $b = b_1, b_2, \dots, b_c$ 에 대한 $G(b)$ 에서 독립적으로 계산된다. 그러면, I-quickest path는 c 경로들 중에서 가장 낮은 종단간 지연을 갖는 경로가 선택된다. 알고리즘 Quick-I는 단지 $B(v, w) \geq b$ 인 경우에만 v 에서 w 까지 경로를 확장 함으로써 $G(b)$ 의 경로 계산을 섞는다.

모드III에 대한 알고리즘은 Quick-II의 라인 11을 아래의 명령문으로 교체함으로써 얻어지며,

11'. $TE[w][B(v, w)] \leftarrow \min\{TE[w][B(v, w)], TE[v][b] + D(v, w);\}$

모드 IIIa에 대한 알고리즘은 Quick-III의 라인 11을 다음에 나오는 명령문으로 교체함으로써 얻어질 수 있다.

11' $TE[w][B(v, w)] \leftarrow \min\{TE[w][B(v, w)], TE[v][b] + \sigma/B(v, w) + D(v, w)\}$

$i = I, IIa, III, IIIa$ 에 대해 알고리즘 Quick-i에 대응하는 알고리즘의 증명과 시간 복잡도는 Theorem 2.1의 약간의 수정을 통해 얻어 질 수 있다.

3. 결 론

본 논문은 Circuit Switching, 인터넷 프로토콜, 그리고 그것들의 조합으로 형성된 메커니즘을 반영하는 6종류의 Quickest Path Problem에 대한 설명 뿐만 아니라 기본 알고리즘과 그것의 변형인 첫 번째 5개 모드에서의 Quickest Path 계산방법과 다익스트라 알고리즘이 응용된 마지막 모드에서의 Quickest Path 계산방법이 소개되었다.

앞으로 연구 방향은 본 논문에서 소개된 다양한 전송모드에 대한 path-table의 계산을 Multiple Paths (모드I에서 [4, 8]과 같은), Multicast Trees [7], 그리고 신뢰성문제를 포함할 것이다.

참 고 문 헌

- [1] Y. L. Chen and Y. H. Chin. The quickest path problem. *Computers and Operations Research*, 17(2): 153-161, 1990
- [2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Co., New York, 1990.
- [3] N. S. V. Rao and S. G. Batsell. Algorithm for minimum end to end delay paths. *IEEE Communications Letters*, 1(5): 152 - 154, 1997.
- [4] N. S. V. Rao and S. G. Batsell. QoS routing via multiple paths using bandwidth reservation. In *IEEE INFOCOM98: The Conference on Computer Communications*, volume 1, pages 11 - 18. 1998.
- [5] J. B. Rosen, S. Z. Sun, and G. L. Xue.

- Algorithms for the quickest path problem and the enumeration of quickest paths. *Computers and Operations Research*, 18(6): 579~584, 1991.
- [6] G. Xue. End to end data paths: Quickest or most reliable? *IEEE Communications Letters*, 2(6):156~158, 1998.
- [7] G. Xue and S. Sun. Optimal multicast trees in communication systems with channel capabilities and channel reliabilities. *IEEE Transactions on Communications*, 47(5): 662~663, 1999.
- [8] G. Xue, S. Sun, and J. B. Rosen. Fast data transmission and maximal dynamic flow. *Information Processing Letters*, 66:127~132, 1998.

● 저자 소개 ●



방영철 (Young-Cheol Bang)

1994년 University of Oklahoma 전산학과 졸업(학사)

1997년 University of Oklahoma 대학원 전산학과 졸업(석사)

2000년 University of Oklahoma 대학원 전산학과 졸업(박사)

2000~2002년 한국전자통신연구원 선임연구원

2002~현재 한국산업기술대학교 컴퓨터공학과 교수

관심분야 : 유.무선 라우팅 프로토콜, 홈네트워크, 멀티미디어 전송, 차세대 영상시스템 등

E-mail : ybang@kpu.ac.kr



정성택 (Sung-Taek Chung)

1992년 한국과학기술원 전기 및 전자공학과 졸업(학사)

1995년 한국과학기술원 정보 및 통신공학과 졸업(석사)

2000년 한국과학기술원 전기 및 전자공학과 졸업(박사)

2000년~2003년 메디너스 MRI 연구소장

2004년~현재 한국산업기술대학교 컴퓨터공학과 교수

관심분야 : 디지털 영상처리, 멀티미디어, 컴퓨터 그래픽스

E-mail : unitaek@kpu.ac.kr