

지연시간 한계의 만족과 효율적인 최소 지연변이 멀티캐스트 트리 생성 알고리즘

김 문 성[†] · 추 현 승^{**} · 이 영 로^{***}

요 약

멀티미디어 그룹 애플리케이션들이 증가함에 따라, QoS 요구사항을 만족하는 멀티캐스트 트리를 생성하는 것은 매우 중요한 문제로 부각되고 있다. 본 논문에서는 NP-complete 문제인 지연시간 제한과 지연변이 제한을 만족하는 멀티캐스트 트리(DVBMT : delay- and delay variation-bounded multicast tree) 문제를 다루겠다. 이 문제는 목적노드들을 포함하는 신장 트리를 생성하는 것으로, 이들 목적노드들은 최소화된 멀티캐스트 지연변이를 가지며, 시작노드에서 각 목적노드로의 경로상의 지연시간은 제한된 지연시간을 만족한다. 이러한 문제의 해법은 온라인 게임이나 쇼핑, 또는 원격 회의와 같은 실시간 통신 서비스를 제공하는데 필요하다. 지금까지 DVBMT 문제를 이상적으로 다루었다고 알려진 DDVCA보다 본 논문에서 제시한 알고리즘이 더욱 효율적이라는 것은 성능 평가를 통해 확인할 수 있다. 이를 통해 확인된 성능 향상은 DDVCA를 normalized surcharge로 계산 했을 때, 약 3.6%에서 11.1%에 이른다. 본 논문에서 제안한 알고리즘의 시간복잡도는 $O(mn^2)$ 이다.

Efficient Multicast Tree Algorithm for Acceptable Delay and Minimum Delay Variation

Moon-seong Kim[†] · Hyun-seung Choo^{**} · Young-ro Lee^{***}

ABSTRACT

With the proliferation of multimedia group applications, the construction of multicast trees satisfying QoS requirements is becoming a problem of prime importance. In this paper, we study the delay- and delay variation-bounded multicast tree (DVBMT) problem which is NP-complete. The problem is to construct a spanning tree for destination node, which has the minimized multicast delay variation, and the delay on the path from the source to each destination is bounded. A solution to this problem is required to provide decent real-time communication services such as on-line games, shopping, and teleconferencing. Performance comparison shows that the proposed scheme outperforms DDVCA which is known to be effective so far in any network topology. The enhancement is up to about 3.6%~11.1% in terms of normalized surcharge for DDVCA. The time complexity of our algorithm is $O(mn^2)$.

키워드 : DVBMT 문제(delay- and delay variation-bounded multicast tree Problem), 지연변이(delay variation), DDVCA(Delay and Delay Variation Constraint Algorithm)

1. 서 론

멀티캐스트 기술은 네트워크 계층의 라우팅 기술과 수송 계층의 신뢰성 제공 기술 및 멀티캐스트 응용서비스 기술로 분류할 수 있을 것이다. 이 중에서도 특히 멀티캐스트 라우팅 기술은 멀티캐스트 서비스 보급을 위한 핵심기반 기술이며 기술개발 측면에서 그동안 많은 진전을 이루었다. 멀티캐스트 통신에서는 동일한 멀티캐스트 그룹에 속하는

다수의 목적노드로 메시지들이 전송된다. 멀티캐스트 통신을 위한 효율적인 해답은 시작노드를 루트로 하여 모든 그룹 목적노드들을 신장하는 멀티캐스트 트리를 생성하는 것을 의미한다. 트리의 비용을 트리의 모든 링크 비용들의 총계라고 규정한다면, 이 트리의 비용을 최소화 하도록 멀티캐스트 트리를 디자인하여 네트워크 자원을 최소화 하여야 한다. 이러한 최소 비용 멀티캐스트 트리는 스타이너 트리(Steiner Tree)[2]라는 이름으로 잘 알려져 있다. 그리고 그와 같은 트리를 찾는 것은 NP-complete 문제이다.

실시간 통신에서 메시지들은 그들의 목적노드들로 제한된 시간 안에 전송되어야 한다. 만일 그렇지 않으면 메시지

※ 본 연구는 BK21과 정보통신부의 지원을 받았습니다. 책임저자 추현승.

[†] 준 회원 : 성균관대학교 대학원 정보통신공학부

^{**} 종신회원 : 성균관대학교 정보통신공학부 교수

^{***} 정 회원 : 한국전산원 BeN기획팀장

논문접수 : 2004년 8월 10일, 심사완료 : 2004년 10월 14일

들은 의미가 없을 것이다. 실시간 컴퓨터 네트워크의 시작 노드에서 각각의 목적노드로의 지연시간의 상한값을 보장해야 한다. 이것은 멀티캐스트 종단간(End-to-End) 지연시간 문제로 알려져 있다[6]. 원격 회의를 한다면, 현재 발표하고 있는 사람의 메시지를 모든 참가자가 동시에 들도록 하는 것은 매우 중요한 사항이다. 그렇지 않다면 실제 오프라인과 유사함을 원하는 원격회의의 의미를 잃게 될 것이다. 이는 멀티캐스트의 지연변이 문제와 관련이 있다[4]. 본 논문에서는 지연시간 제한과 지연변이 제한을 만족하는 멀티캐스트 트리(DVBMT: delay- and delay variation-bounded multicast tree) 문제를 다루었다. 현재까지 최고로 알려진 DDVCA[5]와 컴퓨터 시뮬레이션으로 비교하여 제안하는 알고리즘이 멀티캐스트 지연변이의 관점에서 효율적이라는 것을 보여주었다. 성능 향상은 DDVCA를 normalized surcharge로 비교했을 때, 약 3.6%에서 11.1%에 이른다.

본 논문의 구성은 다음과 같다. 2장에서는 멀티캐스팅을 위한 네트워크 모델에 대하여 설명하고, 3장에서는 우리가 제시하는 알고리즘을 상세히 기술한다. 그리고 4장에서는 시뮬레이션 모델을 이용하여 제안하는 알고리즘의 성능 평가 및 분석을 하고 5장에서 본 논문의 결론을 제시한다.

2. 관련 연구

2.1 네트워크 모델

n 개의 노드와 l 개의 링크(에지, 아크)를 지닌 그래프 $G=(V, E)$ 에 의해서 표현된 네트워크를 생각한다. V 는 노드들의 집합이며, E 는 링크들의 집합이다. 각 링크 $(i, j) \in E$ 는 지연시간 $d_{(i,j)}$ 를 갖는다. 하나의 링크가 갖는 지연시간은 그 링크의 대기지연, 전송지연, 그리고 전파지연의 합이다. 노드 u 에서 v 까지 경로를 $P(u, v) = \{(u, i), (i, j), \dots, (k, v)\}$ 로 나타내자. 경로 $P(u, v)$ 의 길이를 $n(P(u, v))$ 라고 나타내고, 이는 $P(u, v)$ 를 이루는 링크의 개수라고 정의한다. 관계(Relation) \leq 을 $P(u, v)$ 에서의 이진관계(Binary Relation)라 하고, $(a, b) \leq (c, d) \leftrightarrow n(P(u, b)) \leq n(P(u, d))$ 라 정의한다(여기서, $\forall (a, b), (c, d) \in P(u, v)$).

$(P(u, v), \leq)$ 는 전순서집합(Totally Ordered Set)이다. 주어진 시작노드 $s \in V$ 와 목적노드 $d \in V$ 에서, $(2^{s-d}, \infty)$ 는 s 에서 d 까지 가능한 모든 경로들의 집합이다.

$(2^{s-d}, \infty) = \{P_k(s, d) \mid s \text{에서 } d \text{까지 가능한 모든 경로들, } \forall s, d \in V, \forall k \in \Lambda\}$

여기서 Λ 는 첨자들의 집합(Index Set)이다. 어떠한 경로 P_k 의 경로 지연시간은 $(2^{s-d}, \infty)$ 에서 음이 아닌 실수 $R^+ \cup \{0\}$ 로의 함수로 나타낸다. (P_k, \leq) 은 전순서 집합이므로 P_k 와 $N_{n(P_k)}$ 는 동형이다. 따라서 전달사 함수 f_k 가 존재한다.

$$P_k = \{ (u, i), (i, j), \dots, (m, v) \}$$

$$\downarrow f_k$$

$$N_{n(P_k)} = \{ 1, 2, \dots, n(P_k) \}$$

우리는 경로 지연합수를 다음과 같이 정의한다.

$$\phi_D(P_k) = \sum_{i=1}^{n(P_k)} d_{f_k^{-1}(i)}, \quad \forall P_k \in (2^{s-d}, \infty)$$

$(2^{s-d}, \Delta)$ 는 s 에서 d 까지 종단간 지연시간이 Δ 지연 제한을 만족하는 경로들의 집합이다. 그러므로 $(2^{s-d}, \Delta) \subseteq (2^{s-d}, \infty)$.

멀티캐스트 통신에서 메시지는 멀티캐스트 그룹($M, |M| = m$)이라 하는 $M \subseteq V(s)$ 의 수신자들에게 전달되어진다. 시작노드 s 에서 멀티캐스트 수신자 m_i 로 메시지가 지나가는 경로는 $P(s, m_i)$ 로 주어진다. 따라서 멀티캐스트 라우팅 트리는 $T(s, M) = \bigcup_{m_i \in M} P(s, m_i)$ 로 정의되고 메시지는 $T(s, M)$ 을 사용하여 s 에서 목적노드들 M 으로 보내진다.

2.2 DVBMT 문제

멀티캐스트 종단간 지연시간 제한인 Δ 는 시작노드에서 목적노드로의 경로가 가지는 종단간 지연의 허용할 수 있는 상한선을 나타낸다. 이 측정 기준은 멀티캐스트 메시지로 다루어지는 정보는 시작노드에서 보내진 후 Δ 시간 후에는 효력을 상실함을 나타낸다.

멀티캐스트 지연변이 δ 는 시작노드에서 어떠한 목적노드들로 이어지는 두 개의 경로가 가지는 종단간 지연시간 간의 최대 차이이다.

$$\delta = \max \{ | \phi_D(P(s, m_i)) - \phi_D(P(s, m_j)) |, \forall m_i, m_j \in M, i \neq j \}$$

[4]에 처음으로 정의되고 토론된 문제인 종단간 지연시간 제한을 만족하면서, 멀티캐스트 지연 변이를 최소화하는 DVBMT 문제는 다음을 만족하는 트리를 찾는 것이다.

$$\min \{ \delta_a \mid \forall m_i \in M, \forall P(s, m_i) \in (2^{s-m_i}, \Delta), \forall P(s, m_i) \subseteq T_a, \forall a \in \Lambda \}$$

여기서 T_a 는 $M \cup \{s\}$ 를 포함하는 멀티캐스트 트리를 나타낸다. DVBMT 문제는 NP-complete로 알려져 있다[4].

2.3 선행 알고리즘

DVBMT 문제에 대하여 멀티캐스트 트리를 구성하는 유명한 두 개의 알고리즘들이 있다. 하나는 DVMA(Delay Variation Multicast Algorithm)[4]이며, 다른 하나는 DDVCA[5]이다. Pi-Rong Sheu와 Shan-Tai Chen이 제안한 DDVCA 알고리즘은 코어 기반 트리(CBT: Core based Tree)[1]에 바탕을 두고 있다.

DDVCA는 우선 $m_i \in M, v \in V$ 에 대해 각 (m_i, v) 쌍의 최소 지연시간들을 계산한다. 그리고 DDVCA는 각 노드에 대해서 그 각각의 노드와 각 수신자(목적노드) 사이의 지연변이 δ_a 들을 계산한다. 그런 후에, 최소 지연 변이 $\delta = \min \{ \delta_a | a \in A \}$ 를 갖는 노드들을 코어(Core)노드로 선택한다. 최종적으로, 각 목적노드는 최소 지연 경로를 통해 이 코어노드에 연결된다. 시작노드 역시 최소 지연 경로를 통해 이 코어노드에 연결된다. 만약 이 코어노드가 Δ 제한 지연을 만족하지 못한다면 DDVCA는 δ 다음으로 가능한 최소 지연변이 코어노드를 선택하고 위와 동일한 과정이 Δ 요구 조건을 만족할 때까지 반복한다.

생성된 트리의 지연변이의 관점에서, DDVCA가 DVMA와는 미세하나 우수하다는 것은 이미 알려져 있다. 그러나 시간 복잡도는 m 을 수신자(목적노드)의 수, n 을 네트워크의 총 노드 수로 보았을 때 DDVCA는 $O(mn^2)$ 이고 DVMA는 $O(klmn^4)$ 이다.

3. 제안하는 알고리즘

본 장에서 우리는 DDVCA보다 우수하여 멀티캐스트 지연변이를 최소화 시킬 수 있는 멀티캐스트 트리를 생성하는 새로운 알고리즘을 제안한다. 제안한 알고리즘의 기본적인 아이디어는 CBT에 바탕을 두고 있다.

3.1 제안하는 알고리즘 설명

본 논문의 목표는 최소의 멀티캐스트 지연변이를 갖는 멀티캐스트 트리를 생성하는 알고리즘을 제시하는 것이다. 제안한 알고리즘은 코어노드 선정과 멀티캐스트 트리를 생성하는 부분으로 이루어진다. 이제 코어노드 선정에 대해서 설명하겠다. 코어노드의 후보노드가 여러 개일 때, DDVCA는 후보노드 중에서 임의로 코어노드를 선택하지만 우리가 제안하는 알고리즘은 이러한 선택에 명확한 방향을 갖는다. 그 세부 적인 내용은 다음과 같다.

코어노드를 선택함에 있어, 우리는 최소 지연 경로 알고리즘을 사용한다. 3-4단계에서 제안한 알고리즘은 각 목적노드와 시작노드에서 네트워크의 다른 노드들의 최소 지연시간을 계산한다. 그 결과는 <표 2>에 나타나 있다. 5-23단계에서 제안한 알고리즘은 코어노드를 찾는다. 알고리즘이 진행되는 동안 8단계에서는 각각의 노드에 대해 그 노드와 목적노드 사이의 지연변이를 계산한다. 9-12단계에서는 시작노드와 각각의 노드를 잇는 경로가 목적노드를 지나는지 그렇지 않은지를 검사한다. 13-14단계와 15-17단계는 Δ 를 만족하는지 확인하고 최소 지연변이를 갖는 노드를 코어노드의 후보로 선택한다. 다음으로 19-23단계에서, 제안한 알고리즘은 $\min\{ \phi_D(P(s, c_i)) - \min\{ \text{pass}(s, c_i, m_j) \}$ 인 코어노드를 선정한다.

<표 1> 제안하는 알고리즘

Proposed Algorithm ($G(V, E), M, s, \Delta$)

Input	그래프 $G(V, E)$, M 은 멀티캐스트 그룹으로서 $ M = m$, 시작노드 s , 종단간 지연시간 제한 Δ
Output	최소의 멀티캐스트 지연변이를 갖으면서 $\phi_D(P(s, m_i)) \leq \Delta$ 을 만족하며 $\forall P(s, m_i) \subseteq T, \forall m_i \in M$ 인 멀티캐스트 트리 $T(s, M)$
01	Begin
02	$pass = Null; diff_{min} = \infty; candidate = \emptyset;$ $c = Null; compare = Null; T = \emptyset$ /* candidate: 코어노드의 후보들 */ /* compare: 코어노드와 방문한 목적노드들 간의 최대 차이 */
03	For $\forall m_k \in M \cup \{s\}$ Do
04	$Dij(m_k, v_i) = \text{Calculate the minimum delay between}$ $m_k \text{ and } v_i, \forall v_i \in V$
05	For $\forall v_i \in V$ Do
06	$max_i = \max\{Dij(m_k, v_i) \forall m_k \in M\}$
07	$min_i = \min\{Dij(m_k, v_i) \forall m_k \in M\}$
08	$diff_i = max_i - min_i$
09	For $\forall l$ in the minimum delay path from s to v_i Do
	/* l : the node */
10	If $l = m_k, \forall m_k \in M$
11	then $pass(s, v_i, m_k) = Dij(s, m_k)$
12	else $pass(s, v_i, m_k) = 0$
13	If $diff_i < diff_{min}$ and $Dij(s, v_i) + max_i \leq \Delta$
14	then $diff_{min} = diff_i; c = i$
15	For $\forall v_i \in V$ Do
16	If $diff_i = diff_c$ and $Dij(s, v_i) + max_i \leq \Delta$
17	then $candidate = candidate \cup v_i$
18	If $candidate = \emptyset$ then print "Tree construct fail!"
19	For $\forall c_i \in candidate$ Do
20	If $pass(s, c_i, m_k) = 0, \forall m_k \in M$
21	then $compare_i = 0$
22	else $compare_i = Dij(s, c_i) - \min\{$ $pass(s, c_i, m_j) \text{positive and } \forall m_j \in M\}$
23	$c = \min\{ i compare_i \}$
24	For $\forall m_k \in M$ Do
25	$T = T \cup \{ l l \in \text{the minimum delay path from } m_k$ $\text{to } v_c \}$
26	$T = T \cup \{ l l \in \text{the minimum delay path from } m_k$ $\text{to } v_c \}$
27	Return $T(s, M)$
28	End Algorithm.

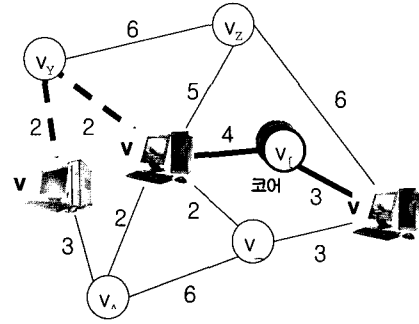
멀티캐스트 트리를 생성하는 과정의 24-25단계에서 각 목적노드는 이렇게 선정된 코어노드와 최소 지연 경로를 통해 연결된다. 26단계에서 시작노드 또한 코어노드와 최소 지연 경로를 통해 연결된다. 마침내 27단계에서 멀티캐스트 트리 $T(s, M)$ 이 생성된다.

제안한 알고리즘의 시간복잡도는 다음과 같이 계산된다. 3-4단계는 다익스트라(Dijkstra) 알고리즘을 사용하여 $O(mn^2)$ 시간에 수행된다. 여기서 n 은 $|V|$ 이다. 적절한 자료구조를 사용하여 $O(1)$ 시간에 10단계를 계산할 수 있기에 5-14단계는 $O(n^2)$ 시간에 이루어진다. 15-17단계는 $O(n)$ 에 수행될 수 있다. 19-22단계는 $O(m^2 |candidate|)$ 시간이 걸리는데 이는 최대 $O(m^2n)$ 이다. 25단계는 $O(n)$ 시간이 필요하므로 24-25단계의 총 시간은 $O(mn)$ 이다. 같은 방식으로 26단계에 필요한 시간은 $O(n)$ 이다. 결과적으로 제안한 알고리즘의 총 시간복잡도는 $O(mn^2)$ 이며 이는 DDVCA의 시간복잡도와 일치한다.

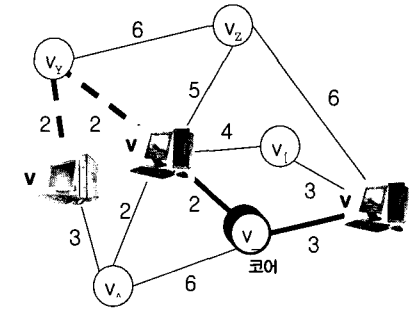
3.2 사례연구

이제 우리는 예제를 사용하여 제안한 알고리즘을 볼 것이다. (그림 1)의 (a)는 특정한 네트워크 토폴로지를 나타낸 것이고, 링크 지연시간은 각 링크 상에 표현하였다. 멀티캐스트 종단간 지연시간 제한 Δ 는 12이다. (그림 1)의 (b)는 DDVCA를 통해 얻어진 멀티캐스트 트리이다. (그림 1)의 (c)는 제안한 알고리즘에 의해 생성된 트리이다.

<표 2>로부터, 가장 작은 멀티캐스트 지연 변이를 갖는 노드가 v_3 와 v_4 그리고 v_8 임을 알 수 있다. 그러나 지연 제한 $\Delta=12$ 를 고려해야 하므로, 노드 v_3 는 제외된다. DDVCA는 임의로 노드 v_4 를 코어노드로 선택한다(물론 랜덤이므로 v_8 일 수 있으나 [5]의 의사코드는 v_4 를 선택한다). 그러나 제안한 알고리즘은 노드 v_8 을 코어노드로 선택한다. 이는 제안한 알고리즘이 $compare_{v_4} = 8-4=4$ 와 $compare_{v_8} = 6-4=2$ 중에서 작은 것을 19-23단계에서 계산하여, 노드 v_8 을 코어노드로 선택하기 때문이다.



(b)

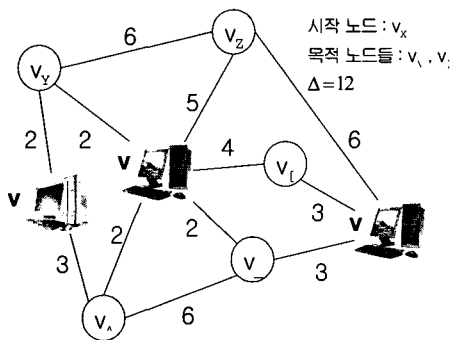


(c)

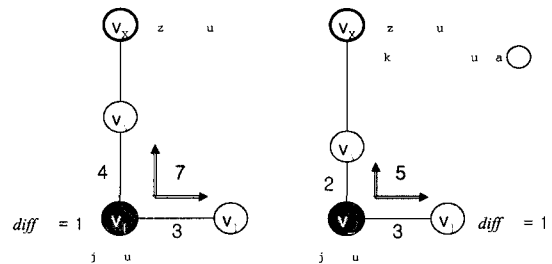
(그림 1) (a) 주어진 네트워크 $G(V, E)$, 링크에 표현된 값은 지연시간, (b) DDVCA, $\delta_{DDVCA}=7$, (c) 제안한 알고리즘, $\delta_{k_w}=5$

<표 2> 제안한 알고리즘에 의한 코어노드 선택

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	
시작노드	v_1	0	2	8	8	4	9	3	6
시작노드(v_1)에서 방문된 목적노드들까지 지연시간	v_5	0	0	0	4	4	4	0	4
	v_6	0	0	0	0	0	9	0	0
목적노드들	v_5	4	2	5	4	0	5	2	2
	v_6	9	7	6	3	5	0	7	3
max_i		9	7	6	4	5	5	7	3
min_i		4	2	5	3	0	0	2	2
$diff_i$		5	5	1	1	5	5	5	1



(a)



(a)

(b)

(그림 2) $compare_{v_4}$ 와 $compare_{v_8}$ 의 구조

결과적으로, DDVCA의 멀티캐스트 지연변이는 7이지만 제안한 알고리즘의 멀티캐스트 지연변이는 5이다. (그림 2) 코어노드를 선택하는 시점에서 동일한 $diff_i$ 값이 있을 때, 제안한 알고리즘이 (그림 2)의 (b)와 같이 현명하게 v_8 을 선택함을 보여준다.

4. 성능 평가

DVBMT 문제에서 최고의 해법으로 알려진 DDVCA와 본 논문에서 제안한 알고리즘을 멀티캐스트 지연변이의 관점에서 비교하였다.

4.1 랜덤 네트워크의 생성

랜덤 네트워크 토폴로지 생성에 대한 세부내용은 다음과 같다. 이 방법 [3]은 n (네트워크의 노드 수)와 P_e (어떠한 두 개의 노드 사이에 링크가 있을 확률)을 파라미터로 사용한다. 우리가 알고 있는 바와 같이, $n \geq 3$ 인 경우가 고려된다. 3개의 노드를 가진 트리는 유일하며, 따라서 우리는 이것을 초기의 트리로 사용한다. 그리고 n 개의 노드를 가진 신장트리로 확장한다. 확률 P_e 를 조정하면 후, 네트워크 토폴로지에 기반한 그래프 생성을 위해 트리에 속하지 않은 링크를 랜덤으로 생성한다. 조정된 확률 P_e 를 계산해보자. $Prob(event)$ 는 $event$ 의 확률을 나타낸다. e 가 두 개의 노드 사이에 가능한 링크라고 가정하면, 다음을 얻을 수 있다.

$$P_e = Prob(e \in \text{spanning tree}) + Prob(e \notin \text{spanning tree}) \cdot P_e^a$$

$$P_e = \frac{n-1}{n(n-1)/2} + (1 - \frac{n-1}{n(n-1)/2}) \cdot P_e^a$$

$$\therefore P_e^a = \frac{nP_e - 2}{n-1}$$

〈표 3〉 랜덤 그래프 생성 알고리즘

Graph Generation Algorithm

<p>A is an incident matrix, r is a simple variable, $random()$ is a function producing uniformly distributed random values between 0 and 1</p>
<p>01 Begin</p>
<p>02 $A_{1,2} = A_{2,1} = A_{2,3} = A_{3,2} = 1$</p>
<p>03 For $i=4$ to n Do</p>
<p>04 $r = (i-1) \times random() + 1$</p>
<p>05 $A_{r,i} = A_{i,r} = 1$</p>
<p>06 For $i=1$ to $n-1$ Do</p>
<p>07 For $j=(i+1)$ to n Do</p>
<p>08 If $P_e > random()$ then $A_{i,j} = A_{j,i} = 1$</p>
<p>09 End Algorithm.</p>

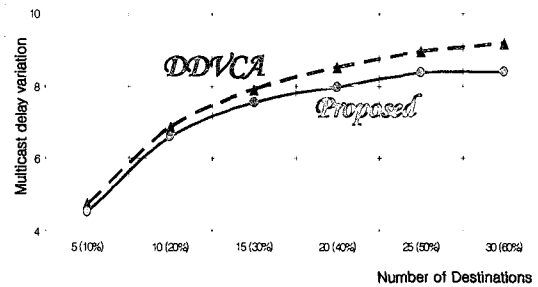
4.2 성능 평가와 분석

이제 우리는 제안된 알고리즘의 성능을 비교하는데 사용한 수치 결과를 설명하겠다. 제안한 알고리즘은 C++로 구현하였다. 네트워크 전체 노드 수 50, 100, 200에 대해 각각 10개의 서로 다른 네트워크를 생성하였다. 시작노드는 랜덤으로 선정하였으며, 목적노드들은 네트워크 토폴로지의 전체 노드 중에서 균일분포로 선택하였다. 또한, 멀티캐스트 그룹의 목적노드들은 전체 노드의 10, 20, 30, 40, 50, 그리고 60%를 차지하도록 하였다. Δ 는 랜덤으로 선정하였다. 각각의 $|V|$ 와 $P_e=0.3$ 에 대해서 1000번($10 \times 100 = 1000$)의 시뮬레이션이 이루어졌다.

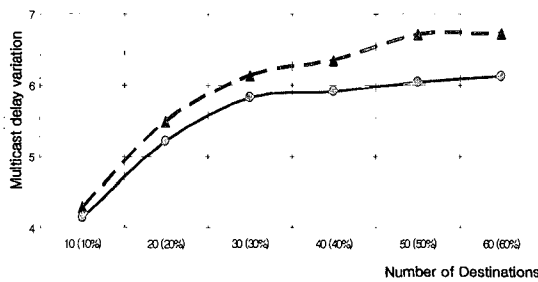
성능 비교를 위해, 우리는 동일한 시뮬레이션 조건상에서 DDVCA를 구현하였다. 그리고 다음과 같이 normalized surcharge를 사용하였다.

$$\bar{\delta} = \frac{\delta_{DDVCA} - \delta_{\text{제안한 알고리즘}}}{\delta_{\text{제안한 알고리즘}}}$$

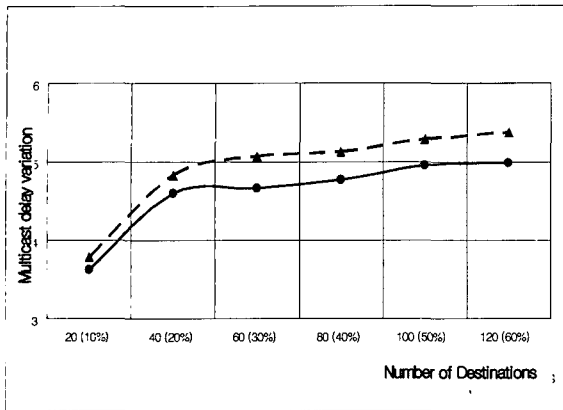
그래프로 나타내면서 이 값을 퍼센티지로 표현하였다(즉, $\bar{\delta} \times 100\%$). (그림 3)의 (a)와 (b) 그리고 (c)는 시뮬레이션 결과로 나타난 멀티캐스트 지연변이를 보여준다. (그림 3)의 (d)는 제안한 알고리즘이 DDVCA보다 항상 좋은 결과를 보임은 쉽게 알 수 있다. 성능 향상은 DDVCA와 normalized surcharge로 약 3.6%에서 11.1%에 이른다.



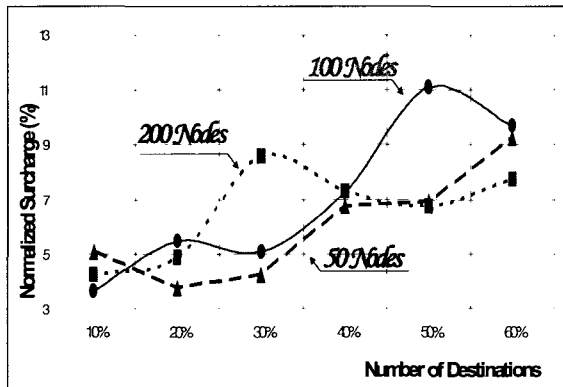
(a) |V| = 50



(b) |V| = 100



(c) $|V| = 200$



(d) $\delta \times 100\%$

(그림 3) 링크간 확률($P_e=0.3$)에서 각각의 $|V|$ 마다 δ 와 $\delta \times 100$ 의 값

5. 결론

본 논문에서는 종단간 지연시간 제한을 만족하며 멀티캐스트 지연변이 또한 최소화하여 메시지를 전송하는 방법에 대해서 생각하였다. NP-complete로 알려진 DVMT 문제의 멀티캐스트 트리 생성을 다루는 유명한 두 가지 접근이 있었으며, 우선 하나는 DVMA이다. DVMA는 멀티캐스트 트리가 갖는 지연 변이의 면에서 훌륭한 성능을 발휘하지만 $O(klmm^4)$ 의 높은 시간복잡도를 갖는다. 알려진 것처럼 높은 시간복잡도는 큰 규모의 고속 네트워크에서는 적당하지 않다. 다른 하나는 DDVCA이며, DDVCA가 생성한 트리의 멀티캐스트 지연변이의 면에서 DVMA보다 조금 우수하지만 시간복잡도는 $O(mn^2)$ 으로 현존하는 최고의 알고리즘이다.

한편, 제안한 알고리즘의 시간복잡도는 $O(mn^2)$ 으로 DDVCA의 시간복잡도와 동일하며, 더 나아가서 컴퓨터 시뮬레이션 결과는 제안한 알고리즘이 DDVCA보다 더 짧은 멀티캐스트 지연변이를 갖는 트리를 생성함을 보인다.

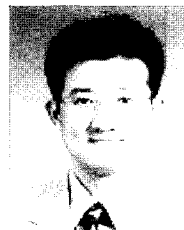
참고 문헌

- [1] Ballardie, T., Francis, P., and Crowcroft, J., "Core based tree(CBT) : An architecture for scalable inter-domain multicast routing," *Computer Commun. Rev.*, Vol.23, No.4, pp.85-95, 1993.
- [2] Hakimi, S. L., "Steiner's problem in graphs and its implication," *Networks*, Vol.1, pp.113-133, 1971.
- [3] Rodionov, A. S., and Choo, H., "On generating random network structures: Connected Graphs," Springer-Verlag Lecture Notes in Computer Science, Vol.3090, pp.483-491, August, 2004.
- [4] Rouskas, G. N., and Baldine, I., "Multicast routing with end-to-end delay and delay variation constraints," *IEEE JSAC*, Vol.15, No.3, pp.346-356, April, 1997.
- [5] Sheu, P.-R., and S.-T. Chen, "A fast and efficient heuristic algorithm for the delay- and delay variation bound multicast tree problem," *Computer Comm.*, Vol.25, pp.825-833, 2002.
- [6] Q. Zho, M. Parsa, and J. J. Garcia-Luna-Aceves, "A source-based algorithm for near-optimum delay-constrained multicasting," *IEEE INFOCOM'95 Proc.*, pp.377-385, March, 1995.



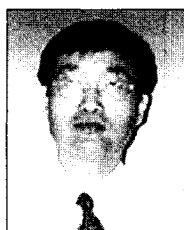
김문성

e-mail : moonseong@ece.skku.ac.kr
 2000년 건양대학교 수학과(학사)
 2002년 성균관대학교 수학과(이학석사)
 2004년 성균관대학교 정보통신공학부
 컴퓨터공학전공(박사수료)
 관심분야 : 라우팅 프로토콜, 이동컴퓨팅,
 텔레메틱스, NEMO, 정보보호



추현승

e-mail : choo@ece.skku.ac.kr
 1988년 성균관대학교 수학과(학사)
 1990년 University of Texas at Dallas,
 컴퓨터공학(석사)
 1996년 University of Texas at Arlington,
 컴퓨터공학(박사)
 1997년 특허청 심사관(사무관)
 1998년~현재 성균관대학교 정보통신공학부 부교수
 관심분야 : 광네트워크, 이동컴퓨팅, 라우팅 프로토콜, 그리드
 컴퓨팅



이영로

e-mail : lyr@nca.or.kr
 1986년 경북대학교 전자공학(학사)
 2002년 고려대학교 경영정보학(석사)
 2003년~현재 한국전산원 BeN기획팀장
 관심분야 : 품질보장 네트워크