SIENA의 이벤트 라우팅 알고리즘의 개선

정종윤[0] 류기열 이정태
아주대학교 정보 및 컴퓨터공학부
{jongyun, kryu, jungtae}@madang.ajou.ac.kr

Enhancement of SIENA Event Routing Algorithms

Jong-Yun Jeong[0] Ki-Yeol Ryu Jung-Tae Lee
College of Information and Computer, Ajou University

요 약

이벤트에 기반한 약결합의 분산 응용이 인터넷과 같은 광역네트워크 상에서의 주요 응용분야로 대두되고 있다. 분산응용의 컴포넌트 사이의 이벤트 전송을 구현하는 방법으로 내용기반 등록/발표 시스템이 최근 활발히 연구되고 있다. 본 논문에서는 효과적인 내용기반 등록/발표 시스템을 구현하기 위해 기존에 발표된 대표적인 시스템인 SIENA에 바탕을 두고, SIENA에서의 이벤트 라우팅 알고리즘의 문제점을 분석하고 이를 개선한 라우팅 알고리즘을 개발하였다.

1. 서론

기존의 이벤트 통지(event notification) 서비스를 구현하는 방식이었던 그룹기반 등록/발표(group-based subscribe/publish)의 대안으로 최근에 연구되고 있는 방식은 내용 기반(content-based) 등록/발표 시스템이다. 내용기반 등록/발표 시스템에서 이벤트 수신자는 자신이 받고자 하는 이벤트의 등록 정보를 이벤트 브로커(이벤트 서버)에게 전달하고 이벤트 브로커는 발표된 이벤트의 내용(content)을 주소로 사용하여 이벤트 수신자를 찾아 전달한다. 이 부류의 대표적인 시스템은 Elvin[6], Keryx[5], Gryphon[1], 과 SIENA[2,3]등이 있다. 특히, Gryphon과 SIENA는 peer-to-peer형식의 네트워크에 기반을 두고 있다.

본 논문은 인터넷 환경에서 효율적인 pere-to-peer에 한정하여 논의한다. SIENA는 이벤트를 등록할 때 이벤트 등록정보의 관계를 고려하여 등록 트래픽을 줄이고 이 관계 정보를 이벤트 서버에 유지함으로써 효율적으로 이벤트를 전송할 수 있는 방법을 제공한다. 그러나, SIENA의 라우팅 알고리즘은 다중 경로 문제와 불완전 등록 문제를 가지고 있다. 본 논문은 인터넷 환경에서의 이벤트 등록과 전송을 위한 효율적인 라우팅을 위해 SIENA 알고리즘의 문제를 해결한 개선된 알고리즘을 제안한다.

2. SIENA 이벤트 서비스

이벤트 통지 서비스는 일반적으로 이벤트 서비스와 클라이언트로 구분된다. 클라이언트는 이벤트 발표자와 이벤트 등록자로 나누어진다. 이벤트 등록자는 이벤트 서비스에 이벤트를 등록하거나 해지 또는 수신하고, 이벤트 발표자는 이벤트를 생성하여 이벤트 서비스에 발표한다. SIENA에서 이벤트는 속성들의 집합으로 표현되며 각각의 속성들은 (type, name, value)의 세 가지 요소로 구성된다. 이벤트를 등록하기 위한 데이터 구조를 이벤트 필터라고 하는데 이벤트 필터는 속성조건의 집합으로 이루어진다. 이벤트 필터 f의 모든 속성조건에 대하여 만족되는 속성이 이벤트 n에 존재하면 그 때 이벤트 필터 f는 n을 포함한다(cover)고 말한다. 또한 두 개의 필터 f와 f'에도 포함관계(covering relation, $<$)가 존재하는데 다음과 같이 표현할 수 있다.

$$f<f' \Leftrightarrow \forall n: n<f \Rightarrow n<f' \quad - (1)$$

SIENA는 이벤트를 등록하고 전송하기 위해 2가지의 자료구조를 이용한다. 하나는 이벤트 등록자로부터 잠재적인 이벤트 생성자들에게 이벤트 등록을 전송하기 위한 스패닝 트리(Spanning tree, 이하 ST라 줄여 부른다.)이다. 다른 하나는 필터의 등록과 이벤트 전송에 모두 사용되는 이벤트 필터의 poset구조이다.

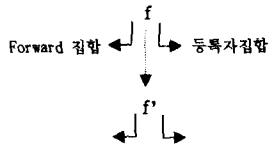


그림 1 poset 구조($f<f'$)

각 서버에는 주변 서버들이나 클라이언트들로(통틀어 이웃이라고 함)부터 도달한 이벤트 등록정보가 poset 구조로 하나씩 존재하게 된다. 등록자집합(subscriber set)은 이벤트 필터를 전달한 인접한 노드들의 집합으로 이후에 이벤트가 전송되면 원 등록자에게 전송하기 위한 경로로 사용된다. Forward 집합은 이벤트 등록이 전달되는 이웃 노드들을 나타낸다. SIENA에서 이벤트 등록을 전달하는 방식은 필터의 포함관계를 이용하여 보다 포괄적인 등록 f가 전달되었다면 f에 포함되는 f'은 이웃으로 전달되지 않는 원리를 따른다.

SIENA의 라우팅 알고리즘에는 2가지 문제가 내재되어 있다. 하나는 이벤트가 다중 경로를 통해 원등록자에게 전달될 수 있는 다중 경로 문제이고, 또 하나는 이벤트 등록이 ST를 통해 전달될 때 모든 노드에 전달되지 않아 발생하는 불완전 등록 문제이다. 이벤트 등록, 해지, 및 전송을 위한 SIENA의 라우팅 알고리즘은 [3]을 참조하라.

3. 이벤트 라우팅 알고리즘

3.1 등록 방법

서버 X가 이웃 U로부터 등록 메시지(subscribe(U,f,a))를 받

은 경우의 등록방법은 아래 알고리즘으로 기술된다.(a는 원등록자) 본 논문에서는 등록자외에 원등록자를 나타내는 기호를 등록자의 아래첨자로 붙인 확장 등록자(tagged subscriber)를 사용한다. 예를 들어, 2_a라면 2는 그 등록의 등록자이며 a는 그 등록의 원등록자를 의미한다. 확장 등록자의 집합에서 2_a,2_b와 같이 동일한 등록자를 사용하는 확장 등록자는 $2_{a,b}$로 표현할 수 있다.

algorithm subscribe
input U: 등록자
f: 필터
a: f의 원등록자
begin
(1) **if** $\exists f' \in$ Ps s.t. $f<f'$ and $U \in$ *subscribers(f')*
(2) **if** $U_a \in$ *tagged_subscribers(f')*
(3) **return**
(4) **else**
(5) U_a를 *tagged_subscribers(f')*에 추가한다. (merge)
(6) **elseif** $\exists f' \in$ Ps s.t. $f'=f$ and $U \notin$ *subscribers(f')*
(7) U_a를 *tagged_subscribers(f')*에 추가한다.
(8) **else**
(9) f의 등록정보를 포함관계에 따라 poset P_s의 정확한 위치에 삽입하고 U_a를 *tagged_subscribers(f)*에 추가한다.
(10) **endif**
(11) **if** $\exists f' \in$ Ps s.t. $f'<f$, $U_t \in$ *subscribers(f')*
(12) Ut를 *tagged_subscribers(f')*에서 제거하고 *tagged_subscribers(f)*에 넣는다. (merge)
(13) **if** *subscribers(f')* = ∅
(14) Ps로부터 f'를 제거한다.
(15) **endif**
(16) **endif**
(17) a의 ST에서 X의 자식노드에 속하는 모든 노드에 subscribe(X, f, a)를 보낸다
end

원등록자가 다르면 전달된 f를 포함하는 f'이 존재하더라도 이웃에 전달이 된다. 원등록자의 정보가 poset에 추가됨으로서 서로 다른 ST를 타고 전달된 필터들을 구분할 수 있다.

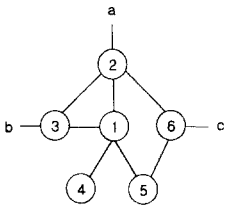
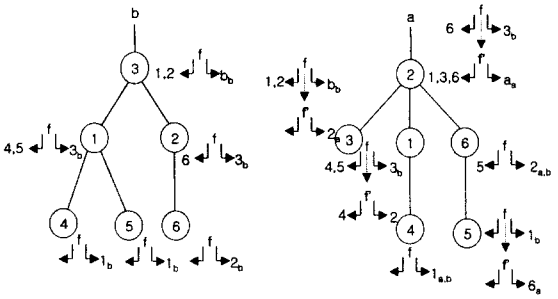


그림 2 예제 네트워크

그림3은 b에서 f를 등록한 후에 a에서 f'를 등록한 모습이다. SIENA에서는 노드 2에서 b가 f를 등록하는 과정에서 노드 6으로 등록을 전파했기 때문에 a가 f'를 등록할 때는 노드 6으로 f'을 보내지 않는다.

그러나, 본 논문에서는 원등록자가 다르면 각자의 ST를 따



a) b에서 f를 등록한 후 b) a에서 f'를 등록한 후
그림 3 f와 f'를 등록한 모습

라 등록 정보를 보내기 때문에 f'도 2->6을 지나 노드 5까지 전달된다. 한가지 주목 할 것은 노드 4와 6의 경우에 하나의 노드(e.g. 2)로부터 두 개의 포함관계를 가진 등록이 전달 된 경우 더 포괄적인 필터로 병합(merge)된다.

3.2 해지 방법

서버 X가 이웃노드 U로부터 이벤트 해지 메시지(unsubscribe(U,g,a))를 받았다고 하자. g에 의해 포함되고 U_a를 등록자 집합에 포함하는 모든 f에 대해서 다음 3가지 경우로 나누어 연산을 수행한다. 원등록자 a의 ST에서 노드 X의 자식노드를 *children(a, X)*라고 표기하기로 한다.

i) $f'<f$를 만족하는 가장 포괄적인 필터 $f' \in Ps$에 대하여, *children(a, X)*과 *children(b, X)*의 교집합이 공집합이 아닌 그런 b가 f'의 원등록자 중에 존재한다면, 그 교집합의 각 원소 노드 W에 split(f, f', X, b)을 전달한다. 그리고나서 X의 poset에서 f의 등록자 집합에서 U_a를 제거한다.

ii) $f<f'$를 만족하는 가장 포괄적인 필터 $f' \in Ps$에 대하여, *children(a, X)*과 *children(b, X)*의 교집합이 공집합이 아닌 그런 b가 f'의 원등록자 중에 존재한다면, 그 교집합의 각 원소 노드 W에 split(f', f, X, b)을 전달한다. 그리고나서 X의 poset에서 f의 등록자 집합에서 U_a를 제거한다.

iii) i)-ii)의 경우가 아니라면 X의 poset에서 f의 등록자 집합에서 U_a를 제거한다.

모든 경우에서 Ua를 제거한 후 등록자 집합이 ∅라면 필터 f를 poset에서 제거한다. 그리고 a의 ST에서의 자식 노드로 unsubscribe(X, g, a)를 보낸다.

3.3 이벤트 전송방법

이벤트가 생성자로부터 등록자에게 전달되는 경로는 등록이 전달된 경로의 역경로이다. 이 역경로를 계산하기 위해서 poset에 저장된 등록자 정보와 이벤트 생성자를 위한 ST의 정보를 이용한다. 이벤트 생성자가 노드 U라하고, $\Sigma = \{S_1, S_2, ..., S_n\}$을 각 노드를 위한 등록 ST의 집합이라고 하자. 각 트리(S_i)의 루트로부터 U로의 등록 경로(P_{iu})는 정확하게 하나씩 존재한다. S_i가 모두 최소깊이를 가지는 ST라면, 이들 경로의 집합은 노드 U를 루트로 하는 최소깊이 ST 트리가 되며 이는 U를 위한 이벤트 전송 ST로 활용한다. 제안된 방법을 적용하면 노드 4

를 위한 이벤트 전송 ST는 아래 그림과 같다.

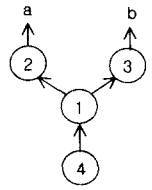


그림 4 노드 4의 이벤트 전송 ST

각 노드는 모든 다른 노드(이벤트 발표자)로부터 모든 다른 노드(필터 등록자)로 가는 경로 정보를 유지하고 있다. 하나의 노드에서 경로정보는 이벤트 발표자를 위한 ST에서의 자식노드의 집합으로 표현할 수 있다. 이벤트 발표자 노드를 P_1, P_2, ..., P_n이라고 하면 노드 X가 유지해야 하는 경로정보는 $children(P_i, X)$ (i=1,...,n)이다. 예를 들어, 그림 2에서 노드 2의 경로정보는 다음과 같다.

children(1, 2)={a, 6}, children(2, 2)={a, 3, 6},
children(3, 2)={a, 6}, children(4,2)={a, 6},
children(5,2)={a}, children(6,2)={a, 3}

예를 들어, 이벤트가 노드 4에서 이벤트가 발생하여 노드 2에 도착했다면 전파될 수 있는 다음 노드는 *{a, 6}*이 된다. 노드 P에서 발생한 이벤트 e가 어떤 노드 X에 도착했을 때 실제로 이동해야 할 다음 노드의 집합 $next(P, e, X)$는 X에서의 경로정보와 e가 매치되는 필터의 등록자 집합으로부터 쉽게 계산할 수 있다. 노드 X의 poset에서 e를 포함하는 모든 필터들의 등록자 집합을 $subscribers(X, e)$이라 하면 $next(P, e, X)$는 다음과 같다.

$$next(P, e, X) = children(P, X) \cap subscribers(X, e) \quad (2)$$

4. 분석

4.1 다중경로문제의 해결

SIENA의 다중경로 문제는 경로가 사이클을 이룰 수 있다는 것이다. 이는 이벤트를 전달할 이웃을 계산할 때 poset에서 유지하는 등록자 집합에만 의존하기 때문이다. 본 논문에서는 3장에서 설명한 것과 같이 이벤트 발표자를 위한 ST는 발표자로부터 모든 클라이언트에게 정확히 하나씩의 이벤트 전송 경로를 제공한다. 이벤트 등록의 역경로의 집합과의 교집합은 이벤트 발표자가 루트인 하나의 트리가 된다.

4.2 불완전 등록 문제의 해결

SIENA의 불완전 등록 문제는 어떤 등록자가 등록을 해지할 때, 다른 등록자의 등록에 대한 이벤트 경로가 없어질 수 있다는 것이다. 본 논문의 알고리즘에서는 한 클라이언트에 의한 이벤트 등록은 자신의 ST를 타고 모든 노드에 전달되기 때문에 이러한 문제가 발생하지 않는다.

4.3 관련 작업과의 비교

SIENA는 포함관계를 기반으로 하는 가장 포괄적 필터 등록 원칙을 이용하여 이벤트 등록의 전송량과 이벤트의 전송량을 줄인다. 그러나 앞의 2가지 문제점을 가지고 있다. 본 논문은 SIENA의 문제점은 해결했지만 등록정보가 ST를 따라 모든 노드에 브로드캐스트 되기 때문에 등록정보의 전송량은 SIENA에 비해 증가한다.

Gryphon은 등록정보의 저장을 위해 정교한 매칭트리(matching tree)를 가지고 이를 이벤트의 전달시에 라우팅 경로의 계산에 이용한다[2]. 매칭트리는 poset구조에 비해 매칭되는 필터를 찾는데 국지적으로 효율적인 방법을 제공하나 이벤트 등록의 포함관계를 고려하지 않고 각 서버가 동일한 매칭트리를 가지고 있기 때문에 각 노드가 가져야 할 정보의 양이 매우 크다. 전체적으로 라우팅 경로를 찾는 경우 poset구조에 비해 비효율적이다.

5. 결론 및 향후 연구

본 논문에서 제안한 방법은 기본적으로 SIENA의 이벤트 서비스 개념과 필터의 포함관계를 이용한 이벤트 등록의 전달 방식에 기반해서 SIENA 알고리즘에 내재한 두가지 문제점을 해결한 개선된 이벤트 라우팅 알고리즘을 제안한다. 본 논문에 제시된 알고리즘은 등록정보의 전송시에 SIENA에 비해 다소 전송량의 증가가 있지만 라우팅 경로를 찾는 알고리즘의 복잡도면에서나 이벤트의 전송량에서는 SIENA와 거의 동일하다.

앞으로 해야 할 일은 제시한 알고리즘을 시뮬레이션 환경에서 등록의 유사성, 유사한 등록의 지역성 등 다양한 특성을 고려하여 분석하는 일이다. 또한, 현재 네트워크 위상에 관계없이 최소깊이의 클라이언트의 ST를 가정하고 있는데 네트워크의 위상에 따른 클라이언트의 ST를 어떻게 만드는 것이 좋은가에 대한 분석이 필요하다. 마지막으로 poset에서의 필터와 이벤트와의 효율적인 매칭을 위한 방법에 대한 연구이다.

감사의 글

본 연구는 2000년 한국학술진흥재단의 연구비에 의하여 연구되었음(KRF-2000-EA0079)

참고 문헌

1. M. K. Aguilera, R.E. Strom, D.C. Struman, M. Astley, T.D. Chandra, "Matching Events in a Content-Based Subscription System," 18th ACM Symposium on Principles of Distributed Computing(PODC1999), Atlanta, GA May 1999, pp. 53-61.
2. A. Carzaniga, D.S. Rosenblum, and A.L. Wolf, "Design and Evaluation of a Wide-Area Event Notification Service," ACM Transactions on Computer Systems, 19(3), Aug. 2001, 332-383.
3. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Interfaces and Algorithms for a Wide-Area Event Notification Service". Technical Report CU-CS-888-99, Department of Computer Science, University of Colorado, October, 1999 (revised May 2000).
4. M. Hapner, et.al. Java Message Service, Version 1.0.2 Sunmicorsystems, Nov. 9. 1999.
5. Keryx WEB page. http://keryxsoft.hpl.hp.com, 1997.
6. B. Segall and D. Arnold. Elvin has left the building: A publish/subscribe notification service with quenching. In Proceedings of AUUG97, Brisbane, Queensland, Australia, Sept. 1997.