

# Push 기술과 정보배포 기술의 분석 및 동향 연구

이윤정, 현은실, 김정범, 김태운  
고려대학교 컴퓨터학과

e-mail: genuine@netlab.korea.ac.kr

## A Perspective on Push Technology and Data Dissemination

Yoon-Jung Rhee, Eun-Sil Hyun, Jung-Bum Kim, Tai-Yoon Kim  
Dept of Computer Science & Engineering, Korea University

### 요약

최근, 인터넷, 무선 네트워크, 위성통신, 케이블 네트워크 등 서로 다른 네트워크 시스템들 간의 상호 연결 능력이 빠른 속도로 증가하고 있으며, 이는 데이터 배포에 기반을 둔 정보 중심 응용 분야를 새롭게 확산시키고 있다. 이러한 응용들은 데이터를 많은 수의 클라이언트로 전송하기 위해서 Push 기술을 사용하는데, Push 기술과 Broadcast와의 차이에 대해 최근에 많은 혼란과 논란이 일고 있다. 이러한 혼란은, Push 기술이 데이터 전달 메커니즘의 비중 있는 디자인의 한 면이라는 점과, 네트워크 정보 시스템이 정보 생산자와 소비자가 다른 집합 사이에서 서로 다른 데이터 전달체계를 이용할 수 있다는 점으로부터 발생된다. 본 논문에서는 배포 기반의 정보 시스템과 그 응용들을 위한 디자인 공간의 특징을 정의하고, 현재의 Push 기술의 구현 방법을 분석한다.

### 1. 서론

최근 많이 등장하고 있는 응용들은 많은 수의 클라이언트들에게의 데이터 배포와 연관되어 있는데, 주식이나 스포츠, 뉴스의 자동 수신기와 같은 정보 공급, 교통 정보 시스템, E-메일을 통한 뉴스 통지, 소프트웨어 배급 등을 이러한 배포 기반 응용들의 예로 들 수 있다. 이것은 광대역 데이터 전송 메커니즘 발전의 결과이다. 특히 인터넷, 무선 통신, 위성, 케이블, 전화망 등의 발전으로 인해 사용자들이 사무실, 가정, 심지어 도로 위에서 최신의 정보를 제공받을 수 있게 되었다.

#### 1.1 Broadcast

데이터 Broadcast 기술은 두 가지 이유로 배포 기반의 응용에서 중요한 역할을 한다. 첫째, 데이터 배포는 본래 1-to-N 혹은 M-to-N( $M \ll N$ ) 처리를 한다. 즉, 데이터는 적은 수의 서버로부터, 중복되는 관심사를 갖는 많은 수의 클라이언트로 전송된다. 따라서, 어떤 특정한 데이터 항목은 많은 클라이언

트들로 분배될 것이다. 두 번째, 대규모의 배포를 가능하게 했던 많은 통신 기술들이 Broadcast를 지원하고 있고, 어떤 경우에는, 의도적으로 Broadcast를 사용하고 있다. 예를 들어, Broadcast 위성 제공자들과 케이블 텔레비전 사업자들은 이제 수많은 가정과 사무실로 몇 Mbps의 전송 속도를 제공할 수 있게 되었다. 이동 통신 사용자를 위한 무선 네트워킹 기술은 Broadcast 구조를 필수적으로 이용하고 있고, Mbone의 기본인 IP 멀티캐스트와 같은 프로토콜들을 통해서, 제한적인 형태지만 Broadcast가 인터넷상에서 지원되고 있다.

#### 1.2 Push

Push 기술은 간단한 개념으로부터 출발한다. 사용자가 필요한 정보에 대한 요구를 명시적으로 요청하는 Pull 기술 대신, Push 기술은 특별한 요청 없이 데이터가 사용자에게 보내질 수 있게 하는 것이다. Push의 장점은 매우 직접적으로 나타난다. 전통적인 Pull 기술은 사용자들이 원하는 데이터에 대한

위치와 시기를 미리 알고 있어야하며, 관련 사이트를 위해 네트워크를 검색하고 소프트웨어의 업데이트를 위해 이미 알려진 사이트들을 조사하는데 많은 시간을 보내야한다. Push는 사용자의 이러한 부담들을 덜어준다. Push의 문제점 역시 뚜렷하게 나타나는데, 통제권이 사용자로부터 데이터 제공자에게 넘어가게 되므로, 사용자가 원하는 정보가 아닌 관계없는 정보를 받을 가능성이 커진다는 것이다. 이러한 잠재적인 문제점들은 사용자의 기호에 대한 잘못된 예측에서부터 "스페밍(Spamming)" 등에서 보여지듯 Push 기술의 오용들과 같은 문제들로부터 야기될 수 있다. 이렇듯, Push의 기술상 특징은 장점과 단점 모두에 대한 원인으로 작용한다.

Push 기술은 인간이 통신을 해온 세월동안 다양한 형태로 존재해 왔다. 신문에서부터 전화, 라디오, 텔레비전, E-mail에 이르기까지 다양하다. 데이터 Push를 위해 컴퓨터 네트워크를 사용한 초기의 작업은 1980년대 이루어졌다. MIT의 Boston Community Information System [1], 방송매체를 통한 데이터 분배를 위한 텔레텍스 시스템 [2,3], Datacycle의 데이터베이스 머신[4] 등은 Push 기술의 형태가 통합된 시스템의 예이다. 그러나, 최근에 인터넷과 Web이 Push 기술과 통합된 Webcasting으로 인해 개인적 관심사와 상업활동에서의 흥미뿐만 아니라 논쟁도 증폭되고 있다.

## 2. 정보 전달의 기본 특성

이 장에서는 데이터 전달 메커니즘을 위한 설계의 다른 측면에서 데이터 Push의 개념이 어떻게 적용되는가에 초점을 맞추어 데이터 전달에 대하여 일반적인 정의를 내리고, 복잡한 분배 시스템들을 단순히 Push 기반이나 Pull 기반으로 정의할 수 없는 이유를 설명한다.

### 2.1 데이터 전달의 여러 선택사항

다른 데이터 전달 형태들이 지원된다면, 분배 정보 시스템은 다양한 서버, 클라이언트, 네트워크, 데이터, 응용 등의 특성들에 대하여 최적화 될 수 있다. 데이터 전달 메커니즘을 비교할 수 있는 3가지 특성은 다음과 같다. 첫째, Push와 Pull. 둘째, 주기성과 비주기성. 셋째, Unicast와 1-to-N. 이 세 가지 특성들은 유명한 많은 연구들에 대한 논의를 위하여 좋은 초석을 제공한다. 특히, 이 특성들은 특별한 상황에 있는 전달 메커니즘에 대하여 최적의 선택을

하기 위하여 필수적으로 고려되어야 한다[5].

#### 2.1.1 클라이언트 Pull vs. 서버 Push

현재의 데이터베이스 서버와 개체 스토리지는 명시적으로 데이터를 요청하는 클라이언트를 위하여 해당 데이터를 관리한다. 요청이 서버에 도착하면, 서버는 해당 정보의 위치를 파악하고 클라이언트에게 그 정보를 전송한다. 이러한 요구-응답의 동작 방식이 Pull 기반이며, 서버로부터 클라이언트로의 정보 전송은 클라이언트의 Pull에 의해 시작된다. 반대로, Push 기반 데이터 전달 방식은 어떤 특별한 요구 없이, 많은 수의 클라이언트에게 정보를 전송한다. Push 기반의 전달 방식에서는 서버에 의하여 전송 시작된다.

#### 2.1.2 주기성 vs. 비주기성

Push와 Pull 방식 모두 주기적인 방식이나 비주기적인 방식으로 수행될 수 있다. 비주기적 전달은 이벤트 중심으로서, Pull 방식을 위한 데이터 요구나 Push 방식을 위한 전송이 사용자의 행동(Pull)이나 데이터 업데이트(Push) 등과 같은 이벤트에 의해서 촉발된다. 이에 반해, 주기적 전달 방식은 미리 정의된 스케줄에 의하여 수행된다. 이러한 스케줄은 고정될 수도 있고, 임의로 만들어질 수도 있다. 규칙적인 기반 위에서 주식 시세를 보내는 응용의 경우는 주기적 Push의 예이며, 어떠한 변동 사항이 있을 때만 주식 시세를 보내는 경우는 비 주기적인 Push의 예이다.

#### 2.1.3 Unicast vs. 1-to-N

데이터 전달 메커니즘의 세 번째 특성은 Unicast 혹은, 1-to-N 통신 어느 것에 기반을 두고 있느냐 하는 문제이다. Unicast 통신에서, 데이터 항목들은 보통 하나의 소스에서 하나의 다른 목적지로 보내지는 반면, 1-to-N 통신은 하나의 소스로부터 보내지는 데이터를 다수의 목적지가 수신할 수 있다.

1-to-N 데이터 전달 방식은 Multicast와 Broadcast의 2가지 형태로 구분될 수 있다. 멀티캐스트에서, 데이터는 클라이언트들 중에서 데이터 수신에 관심을 표시했던 일부 집합에게 보내진다. 수신자들은 이미 알려져 있기 때문에, 쌍방향 통신 매체를 이용한다면, 신뢰할 수 있는 Multicast 전송이 가능하다. Broadcast 전송은 모든 클라이언트들이

수신할 수 있는 매체를 통하여 정보를 전송한다.

## 2.2 전달 메커니즘 분류

위에서 설명했던 특성들을 이용하면, 현재 존재하는 많은 전달 메커니즘을 분류할 수 있다. 본 절에서는 메커니즘 몇 가지에 대하여 논의하고 있다.

비주기적인 Pull - 전통적인 요구/응답 메커니즘은 Unicast의 비주기적인 Pull 방식을 이용한다. 1-to-N을 사용할 경우, 클라이언트들은 다른 클라이언트들에 의한 요구를 간접적으로 사용하여 실제로는 요구하지 않은 데이터를 얻을 수도 있다[6].

주기적인 Pull - 원격 센서와 같은 응용에서는, 시스템은 주기적으로 변화된 값을 탐지하거나 상태 정보를 얻기 위해 다른 사이트로 요구를 보내기도 한다. 정보가 1-to-N 링크를 통하여 전송된다면, 클라이언트들은 다른 클라이언트의 요구/응답을 추적하여 데이터 아이템들을 얻어낼 수도 있다. 실제로 대부분의 Web과 인터넷 기반의 Push 시스템들은 클라이언트와 데이터 서버 사이의 주기적인 Pull방식을 이용하여 구현되고 있다.

비주기적인 Push - 현재, 정기구독과 관련된 프로토콜이 네트워크에서 정보를 배포하기 위한 방법으로 떠오르고 있다[7,8]. 이러한 시스템에서, 사용자들은 자신들이 수신하길 원하는 정보의 타입을 지시하는 프로파일 형태의 정보를 제공한다. 정기구독 방식은 Push 기반으로서, 수신되는 데이터에 대하여 정의된 스케줄이 없는 비주기성을 가질 경우에는 데이터 흐름이 데이터를 제공하는 소스에 의하여 시작된다. 정기구독 방식의 프로토콜들은 본래 1-to-N의 성격을 띠지만, 현재의 인터넷 기술의 한계로 인해, 다수의 클라이언트로 향하는 개별적인 Unicast 메시지를 이용하여 구현되기도 한다. 이러한 시스템에는 인터넷 E-mail 리스트와 인터넷상의 Push 시스템들이 포함된다. 진정한 1-to-N 전달은 IP-Multicast와 같은 기술들을 통해서만이 가능하다. 그러나 이런 기술들은 개별 인트라넷이나 LAN 등으로 제한된다.

주기적인 Push - 주기적인 Push 방식은 많은 시스템에서 데이터 배포를 위하여 사용되어 왔다. Unicast를 이용한 주기적 Push의 예로는 규정된 스케줄 상에서 요약정보를 배포하는 인트라넷 메일링 리스트를 들 수 있다. Majordomo 시스템에서 리스트 관리자는 요약정보를 전송하기 위하여 스케줄을 설정할 수 있다. 이러한 요약정보는 사용자가 지속

적으로 전송되어오는 개별적인 메시지에 의한 방해 없이 메일링 리스트를 따를 수 있게 한다. Broadcast나 Multicast 링크를 통한 주기적 Push를 사용하는 시스템은 많이 존재한다. TeleText[2,3], DataCycle[4], Broadcast Disks[6], Mobile databases 등이 예이다.

## 2.3 End-to-End에 대한 고려사항

Push 기술에 대한 두 번째 혼란은 네트워크 정보 시스템은 전형적으로 많은 연결 노드를 포함한다는 사실이다. 이러한 노드들은 다양한 구조들로 구성되어있고, 다른 종류의 데이터 전달 메커니즘이 서로 다른 노드 집합사이에서 이용될 수도 있다. 이종의 전달 메커니즘이 복합적인 시스템 내에 존재한다면, 데이터 소스와 그에 대한 사용자로 구성되는 end-to-end 시스템을 Push 기반이나 Pull 기반 어느 한쪽으로 정의한다는 것은 부적절하다.

대체로 분배 정보 시스템은 세 종류의 노드를 갖는다. 배포될 기본 데이터를 제공하는 데이터 소스와, 정보를 소비하는 클라이언트, 다른 소스들로부터 얻어진 정보에 다른 정보를 추가하고 소비의 주체인 클라이언트들에게 이 정보를 분배하는 중개자이다. 중개자 계층의 등장으로 인해, 정보 전달이 많은 다른 사용자들의 필요에 맞게 재단될 수 있게 된다.

중개자 계층은 위에서 언급했던 여러 데이터 전달 형태를 보여주는 접착제 역할을 한다. 많은 경우, 예상되는 브로커 사용 패턴은 이용될 전달 형태에 대한 선택을 조절할 수 있다. 예를 들어, 중개자가 많은 요구로 인해 부하가 심하다면 Push 기반의 전달 메커니즘을 사용을 고려할 수도 있다.

데이터 전달 체계에 있는 클라이언트의 입장에서 중개자가 데이터 소스가 될 것이다. 정보 수신자는 자신의 바로 상층 이상에 위치하는 대상을 탐지할 수 없다. 이러한 네트워크의 투명성 원칙으로 인해 데이터 전달 메커니즘이 전체적인 영향 없이 변할 수 있게 된다. 예를 들어, 노드 B가 노드 A로부터 필요한 데이터를 Pull 방식으로 수신하고 있고, 노드 C는 A로부터의 데이터를 포함하는 B로부터 주기적인 Broadcast 방식으로 데이터를 전송 받고 있다고 가정할 때, A가 B로의 전송을 Push 방식으로 전환한다고 해도, C는 자신의 데이터 수신 정책을 변화시킬 필요가 없다. 링크상의 변화는 직접 관련되어 있는 노드만이 관심을 갖는다. 이러한 투명성은 표면적으로 나타나는 데이터 전달 형태가 실제

네트워크 안에서 데이터가 앞서 전달되는 방식과 달라도 됨을 허용한다. 이러한 점들이 Push 기술을 둘러싼 많은 혼란을 야기하고 있다.

### 3. 현재의 Push 기술 재검토

Push 기술에 관련되어 논란의 소지가 있는 논란의 몇 가지 근원을 앞서 살펴보았다. 특히, 이 논란은 사용자가 보는 관점과 실제 시스템에 의해 사용되고 있는 데이터 전달 매커니즘 사이의 불일치로 인한 것이다. 더 나아가, 이러한 불일치 문제는 현재의 Push 기술과 관련되어 성능상의 관점에 많은 뿌리를 두고 있다. 성능 면에서의 불일치에서 발생하는 영향은 다음과 같다.

Push를 대신하는 Pull - 현재의 Webcasting 기술들은 본래 데이터 소스들로부터 정보를 얻기 위한 방법으로 Pull을 사용하고 있다. 이는 원칙적으로 Pull을 기반으로 하는 HTTP 프로토콜의 한계 때문이다. Push를 대신하는 Pull은 조사(Polling) 방식을 요구한다. 조사(Polling)는 많은 요구를 발생시키기 때문에 리소스 집중도가 매우 심하다. 이런 요구들은 클라이언트, 서버, 네트워크 등의 많은 리소스를 소비한다. 만일 모든 클라이언트들이 개별적으로 Poll을 사용한다면, 대량의 요구들로 인한 서버들의 극심한 부하라는 문제는 더욱 악화될 것이다.

비주기성을 대신하는 주기성 - Polling은 본래, 데이터의 전송을 요구하는 독립적인 이벤트인 주기적 방식 안에서 수행된다. 이러한 독립성은 다음과 같은 문제들을 야기한다. Polling이 너무 자주 일어난다면, 오버헤드가 심각해질 수 있고, 반대로 거의 일어나지 않는다면, 클라이언트들이 더 이상 유효하지 않은 데이터에 접근하게 될 것이다.

1-to-N을 대신하는 Unicast - Broadcast나 Multicast 장비가 없는 상황에서, 1-to-N 전송을 필요로 하는 시스템은 수신자의 수만큼의 여러 개의 동일한 메시지를 사용하여 이를 구현해야한다. 이러한 연구에서 대역폭 문제가 발생되리라는 것은 분명하다. 만일 n개의 클라이언트가 동일한 데이터 항목에 관심을 갖는다면, 그 항목은 네트워크를 통해 n번 전송되어야한다[5].

다행히, 네트워크의 투명성이라는 개념이 이런 상황들을 개선하기 위해 사용될 수 있다. 해결방법 중 하나가 방화벽 내에 지역서버를 두는 것이다. 모든 클라이언트들은 지역 네트워크와 시스템 구성에 가장 적합한 방식으로 지역 서버와 상호작용 한다.

지역서버는 그 지역을 대신하여 원거리의 데이터 소스로 Polling을 수행하게 되고, 이는 인터넷의 트래픽을 줄이는 효과를 가져온다. 데이터 소스는 각각의 데이터 항목에 대하여 단지 하나의 데이터만을 전송하게 되고, 필요한 장비가 구성되어 있다면, 지역서버는 자신의 클라이언트에게 해당 데이터를 Multicast하게 된다.

### 4. 결론

Push는 최신의 주제이지만, 과거에도 존재했던 개념으로서, 분산 정보 시스템에서 데이터 전달을 위한 한 방법이다. Push는 Broadcast와 동일하지 않으며, 많은 Push 기반의 산물들은 Unicast 연결을 이용한 주기적인 Pull 방식을 기본으로 하고 있다. 주의 깊은 데이터 관리 기술을 사용한다면 전체 시스템의 성능과 활용의 측면에서 중대한 영향을 끼칠 수 있기 때문에, Push는 더 많은 작업이 이루어져야 할 분야이다.

#### 참고문헌

- [1] D.Gifford, "Polychannel Systems for Mass Digital Communication" CACM, 33(2), 1990.2
- [2] M.Ammar, J.Wong, "The Design of Teletext Broadcast Cycles", Perf. Evaluation, 5, 1985
- [3] J.Wong, "Broadcast Delivery", Proceedings of the IEEE, 76(12), 1988.12
- [4] G.Herman, G.Gopal, K.Lee, A.Weinrib, "The Datacycle Architecture for Very High Throughput Database Systems", Proc. ACM SIGMOD Conf., 1987.4
- [5] M.Franklin, S.Zdonik, "Data In Your Face:Push Technology in Perspective", Conf., ACM SIGMOD, 1998.6
- [6] S.Acharya, R.Alonso, M.Fanklin, S.Zdonik, "Broadcast Disks : Data Management for Asymmetric communication Environments", ACM SIGMOD Conf., 1995.5
- [7] B.Oki, M.Pfluegl, A.Siegel, D.Skeen, "The Information Bus - An Architecture for Extensible Distributed Systems", Proc. 14th SOSP, 1993.12
- [8] D.Glance, "Multicast Support for Data Dissemination in OrbixTalk", IEEE Data Engineering Bulletin, 1996.8