

무선 XML 스트리밍을 위한 시뮬레이션 시스템의 설계 및 구현

박정현 김충수 정연준*
동국대학교 컴퓨터공학과
{dreameboy, rona1st, ydchung}@dongguk.edu

Design and Implementation of a Simulation System for Wireless XML Streaming

Jung Hyun Park, Chung Soo Kim, Yon Dohn Chung
Dept. of Computer Engineering, Dongguk University

요 약

XML은 데이터 교환을 위한 표준으로서 데이터를 효과적으로 조직하고 표현 할 수 있다. 본 논문에서는 XML 데이터를 무선방송하기위한 스트림을 생성하는 시뮬레이션 시스템을 설계하고 구현한다. 구현한 시스템은 서버 모듈과 클라이언트 모듈로 구성되는데 서버 모듈은 XML 데이터에 대하여 스트림을 생성하는 역할을 하고 클라이언트 모듈은 생성된 스트림에 대해서 질의를 수행하고 생성된 스트림으로부터 XML 데이터를 복구하는 역할을 한다. 질의 수행에 대한 결과로 결과 XML 데이터가 출력되고 성능 평가요소로 질의를 수행하는 동안 걸린 수행시간을 측정할 수 있다.

1. 서 론

최근 무선 통신 및 정보기술의 발달로 인한 무선 네트워크 환경의 급속한 확산으로 무선 스트리밍 서비스에 대한 관심이 높아지고 있다. 스트리밍 서비스는 무선정보시스템(Wireless Information System) 분야 및 선택정보급(SDI: Selective Dissemination of Information) 기술 분야에 널리 사용되고 있다[3].

XML은 인터넷 상 에 존재하는 데이터를 표현하고 교환하기 위한 표준으로 각광 받기 시작하면서 점차 그 응용범위가 확대되고 있다. 현재 인터넷과 같은 유선 환경에서의 XML 활용 연구는 활발히 이루어지고 있지만 무선 환경에서의 XML 활용에 대한 연구는 아직 까지 많이 이루어지지 않고 있는 현실이다.

본 논문에서는 XML을 데이터로 사용하는 무선 스트리밍 서비스를 위한 시뮬레이션 시스템을 설계 및 구현하였다. XML 데이터로부터 XML 스트림을 생성하고 생성된 스트림에서 사용자의 질의를 입력받아 그 결과를 보여주도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 시스템 구현을 위한 관련지식을 살펴보고 3장에서는 시스템의 구조 및 기능에 대해서 설명을 하고 4장에서는 구현된 시스템에 대한 상세설명과 결과화면에 대해 설명한다. 마지막으로 5장에서는 결론 및 향후 연구과제에 대하여 언급한다.

2. 배경

2.1 XML & XPath

W3C에 의해서 XML(eXtensible Markup Language)이 표준으

* 교신저자

1) 본 연구는 “정보통신 기초기술연구지원 사업”의 부분적인 지원을 받아 수행되었음.

로 제정된 이후로 많은 관련기술들이 발달하고 있다. 요즘은 단순한 문서 포맷을 넘어서서 데이터를 교환하고 표현하는 표준으로 자리 잡고 있다[2]. 또한 XML은 준 구조적인(semi-structured) 데이터의 표현이 가능하기 때문에 평면적 데이터 뿐 만아니라 구조를 가지고 있는 데이터 또한 표현할 수 있는 장점을 갖는다. 현재 XML은 문서 교환을 위한 용도 뿐 만이 아니라 전자 상거래, 전자 도서관 등 다양한 분야에 활용되고 있다[1].

Xpath는 XML 문서의 일부분을 지시하기위해 사용된다. Xpath는 URL 경로표기법을 사용하여 XML 문서의 계층적 구조를 논리적으로 탐색하며 노드를 사용하여 트리구조로 형성된다.

2.2 무선 방송 스트리밍

무선 방송(Wireless broadcast)은 같은 신호 채널에 존재하는 다수의 사용자들에게 무선 채널을 이용해서 데이터를 전달하는 방식을 말한다. 또한 멀티캐스트(Multi-cast) 방식이 특정 다수를 대상으로 데이터를 전송 하는 반면에 방송은 불특정 다수를 대상으로 한다는 점에서 차이가 있다.

이런 방송에 있어서 가장 중요한 성능 평가 요소로는 접근 시간(Access Time)과 튜닝 시간(Tuning Time)이 있다. 무선 접근시간은 사용자가 방송을 수신하기 시작하는 시점부터 필요로 하는 데이터를 모두 읽을 때 까지 걸리는 시간을 의미하며, 튜닝시간은 사용자가 원하는 데이터를 읽는데 소비한 실질적인 시간을 의미한다. 방송 환경에서 사용자는 자신이 필요로 하지 않는 데이터가 방송되는 시간동안 에너지를 적게 소모하는 휴지상태(Doze mode)로 있게 된다. 그리고 원하는 데이터를 수신하는 동안은 활성상태(Active mode)로 있게 된다. 다시 말해서 튜닝시간은 접근 시간에서 휴지상태(Doze mode)동안의 시간을 제외한 시간을 말한다. 따라서 사용자 단말기의 에너지 제약성으로 인해 튜닝시간을 줄이는 문제가 중요하게 연구되고 있다.

스트리밍(Streaming)은 전송되는 데이터를 끊임없고 지속적인

물의 흐름처럼 처리하는 기술을 말한다. 스트리밍 기술을 이용하면 우선 방송 환경에서 에너지 효율성 및 대역폭의 효율성 그리고 확장성까지 얻을 수 있다.

3. 시스템 설계

시스템의 전체 구조는 그림 1과 같다. XML 문서를 입력받아 서버에서 그 XML 문서에 대한 스트림을 생성하고 생성된 스트림을 방송하게 되면 클라이언트에서 질의를 수행 하고 수행 결과로 결과 XML 데이터를 얻게 된다.

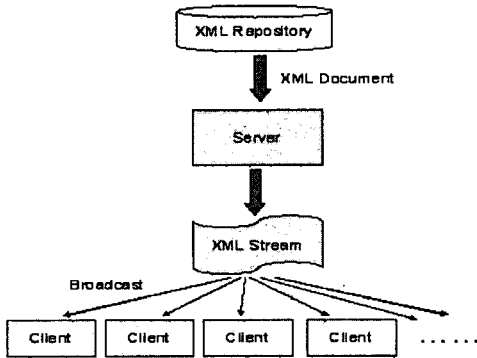


그림 1. 시스템 전체 구조도

3.1 스트림 생성

서버는 XML 문서를 이용해서 그에 해당하는 바이트 스트림을 생성한다. 데이터는 버킷(bucket)단위로 전송되기 때문에 스트림을 다시 버킷스트림으로 생성을 하게 된다. 그림 2와 같은 구조의 XML 문서가 서버에 들어오면 그림3과 같은 XML 문서를 기반으로 한 버킷스트림과 바이트스트림이 생성된다.

```

    <? Xml version="1.0" encoding="UTF-8"?>
    <root>
    <person>
    <name>
    <first>Griff</first>
    <last>Fred</last>
    </name>
    <age>22</age>
    <gender>M</gender>
    </person>
    </root>
    
```

그림 2. XML 문서

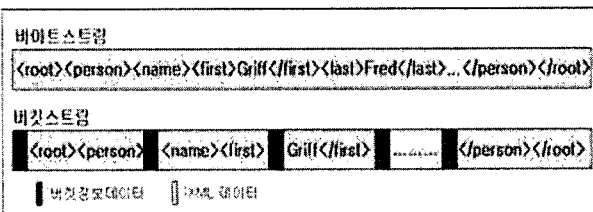


그림 3. XML 바이트스트림과 버킷 스트림

그림 3과 같이 생성된 버킷 스트림은 인덱스가 존재하지 않기 때문에 방송 스트림 상에서 원하는 데이터를 얻을 때 까지 계속해서 방송을 듣고 있어야 하는 문제점이 발생한다. 그래서 이를 해결하기 위해 방송 시에는 다양한 색인 방법을

이용하여 스트림을 방송한다. 본 시스템에서도 우선 스트리밍 방송을 위한 두 가지 색인기법을 사용한다.

3.1.1 단순(Naive) 기법

단순 색인기법은 모든 경로(FULL-PATH)에 대한 테이블을 작성하고 각 경로의 시작주소, 끝 주소를 저장한다. 그리고 작성된 테이블정보를 색인정보로 이용해서 함께 스트리밍 방송을 하게된다. 그림 4는 단순 색인 기법을 이용해서 버킷 스트림을 생성하고 질의를 수행하는 과정이다.

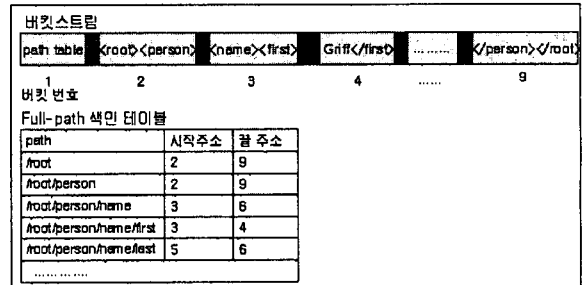


그림 4. 단순 색인기법을 이용한 버킷 스트림

단순기법은 그림4에서처럼 데이터 스트림과 모든 경로 색인 테이블에 기반 한 색인을 같이 방송하기 때문에 방송을 계속 듣고 있지 않아도 색인정보를 이용해서 필요한 데이터가 방송될 때만 방송을 듣고 데이터를 얻을 수 있다. 예를 들어 "root/person/name" 과 같은 질의를 수행 한다면 색인 테이블의 정보를 이용하여 3번째 버킷부터 듣기 시작하여 6번째 버킷까지 읽은 데이터를 가져오게 된다.

3.1.2 체인 기법

단순 색인기법에서는 완전 경로 테이블을 색인 정보로 이용해서 스트림을 방송했던 것과 달리 XML 문서에 대한 트리를 생성한 후에 생성된 트리를 색인정보로 이용하는 기법이다. 그림 5는 체인기법을 이용한 질의 수행을 보여주고 있다.

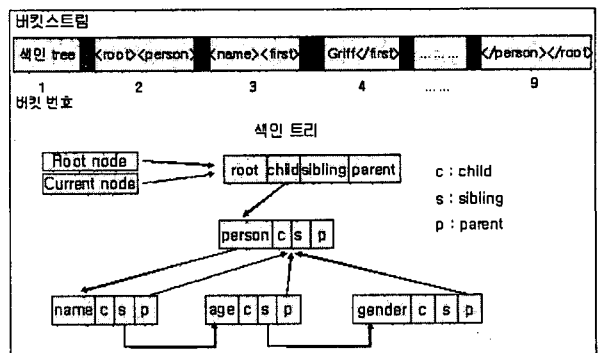


그림 5. 체인색인기법을 이용한 질의수행

그림 5에서처럼 체인기법은 트리를 색인정보로 사용한다. 색인 트리의 구조는 일반적인 형태와 달리 형제(sibling)노드 간의 이동이 가능하다. 구조가 복잡한 XML 문서의 경우 전체 검색(Full-Scan)을 할 경우 많은 시간이 소요되는데 체인 트리 구조를 색인정보로 이용하면 질의 수행 시간을 줄이는 데 효과를 볼 수 있다.

3.2 클라이언트의 스트림 접근

클라이언트는 생성된 XML 버킷스트림에 접근해서 임의의 위치와 XPath 형식의 질의를 입력 받아 그것을 수행하는 역할을

한다. 색인 정보를 가지고 있지 않은 버킷스트림에 대해서는 계속 방송을 듣다가 질의에 해당하는 데이터가 방송되면 데이터를 얻게 된다. 단순 색인기법을 사용한 버킷스트림의 경우에는 데이터와 함께 방송되는 완전경로 테이블 형태의 색인정보를 이용해서 질의에 해당하는 데이터 버킷의 주소를 확인한 후에 그 주소에 해당하는 버킷들이 방송될 때 데이터를 가져오게 된다. 채인가법을 사용한 버킷스트림에서는 데이터와 함께 방송되는 색인정보로 트리구조를 사용한다. 원하는 데이터의 버킷주소정보를 색인트리를 이용해서 얻은 후에 원하는 데이터가 방송되면 데이터를 받아온다.

4. 시스템의 구현

본 논문에서 구현된 시스템은 Microsoft VisualStudio .net 환경에서 개발언어로 C#을 사용하였다. 구현된 시스템은 서버에서 색인방식 및 전송되는 버킷 크기를 입력 받아서 버킷스트림을 생성하게 되고 클라이언트에서 경로(path) 형식의 질의를 입력받아서 질의 수행시간과 결과 데이터에 대한 XML 문서를 나타내게 된다.

4.1 서버 실행부

그림 6은 서버에서 XML 문서를 기반으로 버킷스트림을 생성한 모습을 보여주고 있다. 샘플 데이터 집합으로는 yahoo.xml 문서를 사용하였다[4]. 버킷스트림은 단순색인 기법을 사용하여 생성하였다. 그림6은 서버에서 입력되는 XML 문서를 보여주고 있고 그림7은 입력된 XML 문서를 버킷 크기가 512byte 인 버킷스트림으로 생성한 모습을 보여주고 있다.

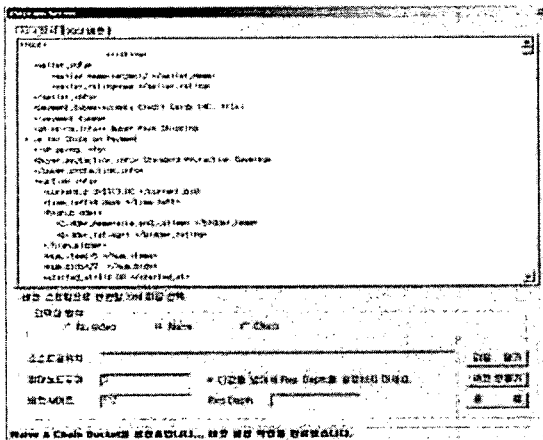


그림 6. 서버의 입력 XML 문서 화면

4.2 클라이언트 실행부

그림 8은 생성된 버킷스트림을 방송하면서 사용자 질의를 수행하는 모습을 보여주고 있다. 왼쪽에는 방송되고 있는 버킷스트림과 스트림의 색인정보 등이 표시되고 오른쪽 하단에는 질의 입력 하는 부분이 있고 상단에는 질의 수행결과에 대한 접근시간 및 튜닝시간이 표시되고 또한 결과 XML 데이터를 보여준다.

5. 결론 및 향후 연구과제

본 논문에서는 무선 XML 스트리밍 서비스를 위한 스트리밍 시뮬레이션 시스템을 구현하였다. 현재 XML은 인터넷 상에서

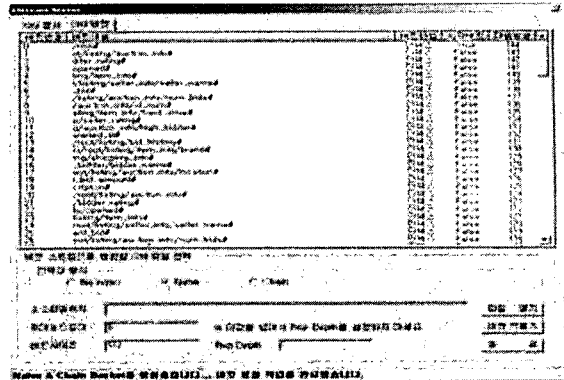


그림 7. 서버의 XML 버킷스트림 생성 화면

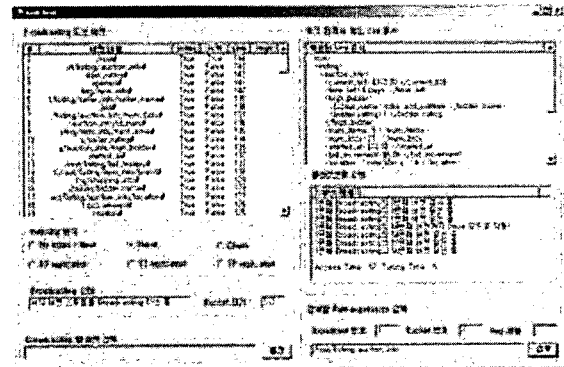


그림 8. 클라이언트의 질의 수행 결과 화면

데이터의 표현 및 교환의 표준으로 자리 잡고 있으며 많은 데이터들이 XML을 기반으로 표현되어 있다. 따라서 XML 데이터를 이용한 무선 스트리밍 서비스 연구를 위해 본 시스템을 구현하게 되었다. XML 스트리밍 시뮬레이션 시스템은 XML 문서를 서버에서 다양한 색인 방식을 사용하여 스트림을 생성하게 되고 생성된 스트림을 기반으로 클라이언트에서 질의 수행을 하게 된다. 본 시스템 구현의 의의는 논문에서 제안된 색인기법 외에 무선 스트리밍 방송을 위한 다양한 색인기법을 실험해 볼 수 있을 뿐만 아니라 다양한 무선 스트리밍 방송 실험을 위한 환경을 구현했다는 데에 있다. 향후 연구로는 질의 수행 시에 XPath의 *, // 등을 수행 가능하게 하는 연구를 생각할 수 있다.

6. 참고문헌

[1] Bray, T., Paoli, J., Sperberg-McQueen, C., "Extensible markup language(XML) 1.0", 1998.
 [2] D.Olteanu, T. Furche, F. Bry, "An Efficient Single-Pass Query Evaluator for XML Data Streams", Proc. of ACM Symposium on Applied Computing(SAC), 2004.
 [3] Y. D. Chung, M. H. Kim, "Effective Data Placement for wireless broadcast", Distributed and Parallel Databases, 2001.
 [4] <http://www.cs.washington.edu/research/xmldatasets>