

무선 통신 환경에서 효율적 데이터 전송과 데이터 품질을 보장하는 지능형 AGENT의 설계

○ 최 영준, 김 명철, 마 중수
한국 정보 통신 대학원 대학교

The Design of Intelligence Agent to Guarantee Efficient Data Transmission & Quality In Wireless Environment

○ Young-Joon Choi, Myung-Chul Kim, Joong-Su Ma
Information and Communications University

요약

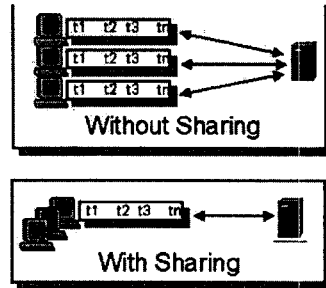
최근 들어 노트북 PC와 같은 이동 컴퓨터를 이용한 무선 환경에서의 작업이 많이 행해지고 있다. 그러나 무선 통신 환경은 데이터 손실이 빈번하고, 무선 통신 환경에서 사용되는 이동 컴퓨터의 대역폭이 작기 때문에 멀티미디어 서비스를 수신하는데 제약을 갖게 된다. 위와 같은 문제를 효율적으로 해결하기 위해 SPAND라는 툴킷(toolkit)의 기본 원리를 응용하여 무선 환경에서의 이동 컴퓨터의 사용을 효과적으로 제어하는 지능형 Agent의 개념을 도입하려 한다.

1. 개요

무선 통신 환경은 유선 환경에서처럼 안정된 전송 매체가 아닌 공중파를 이용하기 때문에 데이터 손실 위험이 높아서 송신자가 데이터를 수신자에게로 전송하고자 할 때, 데이터를 재전송할 확률을 높인다. 또한 이동 컴퓨터는 지원 가능한 디스플레이(display)의 크기와 대역폭이 일반적으로 작기 때문에 송신자 측의 대역폭과 수신자 측의 대역폭이 틀린 경우는 수신 데이터의 품질이 떨어질 수 있다.

여기에서 네트워크의 제한 시간(timeout)을 고려해야 하는데 그 이유는 제한 시간을 너무 짧게 잡아주면 저대역폭 네트워크 상에 전송을 하는 경우나 정체가 일어난 경우에는 재전송의 회수가 늘어날 수 있고, 너무 길게 제한 시간을 잡으면 ACK가 손실될 시 불필요하게 기다리는 시간이 길어져서 효율성이 떨어지는 현상이 발생할 수 있기 때문이다.

새로이 Agent를 도입하기 위해 '같은 도메인 안의 이동 컴퓨터들이 연결되는 다른 도메인의 네트워크 상태는 비슷할 것이다.'라는 가정을 둔다. 이 가정은 버클리 대학의 SPAND(Shared Passive Network Performance Discovery)에서 응용하였다. SPAND란 먼 거리의 인터넷 호스트와 통신 했을 때 성능을 기록하고, 이 정보를 서버에 기억시키는 소프트웨어 툴킷이다. 이 툴킷은 정보를 공유함으로써 네트워크의 특성을 반영한 통신을 해보자는 목적이 있다. 이 원리에 기초한 Agent는 공유 측정(Shared Measurement)과 수동 추론(Passive Probe)의 성격을 지닌다. 수동 추론은 같은 도메인 안의 이동 컴퓨터들의 과거 행적으로부터 얻은 정보를 참조하게 되므로 불필요한 네트워크 경로로의 전송이 생략될 수 있다.



<그림 1> 추론 공유와 비공유의 비교

<그림 1>은 추론한 정보를 공유했을 경우와 그렇지 않은 경우를 비교하고 있다. 공유 측정을 하면 중복된 네트워크 추론을 피할 수 있다. 같은 네트워크 도메인 안에 있는 근처의 이동 컴퓨터들은 같은 경로를 따를 것이기에 결과적으로 데이터를 전송하고자 하는 이동 컴퓨터가 같은 도메인 안에 있다면 이 이동 컴퓨터들이 측정하는 대역폭은 비슷하게 될 것이기 때문이다. 그러므로 각 이동 컴퓨터들이 독립적으로 이런 정보를 추론할 필요가 없게 되어 효율성을 높일 수 있다.

공유 수동 네트워크의 효율은 SPAND 프로젝트에서 이미 성능 평가가 이루어졌는데 공유 측정과 수동 추론을 하지 않은 것과의 네트워크 분석에서 뚜렷한 효율 향상이 있었음을 보여준다. 네트워크 변화가 자주 일어날 경우에는 정보를 공유하는 경우나 그렇지 않은 경우나 성능 면에서 큰 차이를 보이지 않았다. 하지만 네트워크 상태가 100초 이상 동안 변하지 않을 경우에는 정보를 공유 하지 않을 때 추론의 필요성

이 일정 수준으로 고정된 반면에 정보를 공유할 때는 그만큼 같은 도메인 안의 이동 컴퓨터들이 정보를 공유하는 시간이 길어지므로 추론의 필요성이 0.1 이하로 급격히 감소하는 그래프를 그리게 된다. 이 논문을 위해 SPAND 툴킷을 실제 설치해서 테스트해 본 결과 비슷한 효율을 얻을 수 있었다[1].

그러나 SPAND는 유선 통신 환경과는 다른 무선 환경의 특징과 이동 컴퓨터가 도메인 진입과 탈퇴가 자유롭다는 특성을 고려한 설계가 아니다. 또한 Agent가 데이터 재전송의 의무가 없고 시간 제한을 설정하는 기능없이 네트워크 통행의 감시자 역할만 한다. 그리고 도메인 단위로 관리하지 않기 때문에 관리해야 할 먼 거리의 호스트수가 많아지면 확장성 문제가 발생할 수 있다. 이에 SPAND의 단점을 보완하고 무선 통신 환경에 적합한 Agent를 설계하여 보고자 한다.

2. 재전송과 제한 시간 설정하는 알고리즘

여기에서는 재전송과 제한 시간을 설정하는 알고리즘을 Pseudo code로 제시하여 보았다. 제한 시간의 초기값 지정과 재전송시 제한 시간의 증가량은 실제 구현을 통한 시뮬레이션으로 정해져야 한다.

```

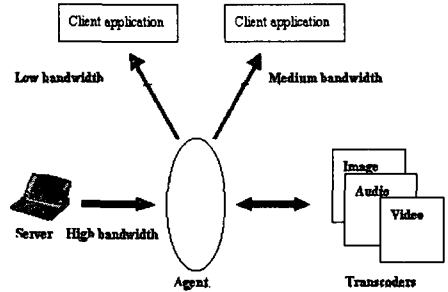
if(이동 컴퓨터로부터 데이터를 받으면)
{
    목적지 주소를 분석하여 도메인 이름을 알아낸다.;
    도메인 이름을 키 값으로 하여 Agent 내의 테이블의 참조
    회수를 1 증가시키고 테이블에서 제한 시간을 얻어낸다.
    만일 테이블에 없다면 도메인 이름과 함께 제한 시
    간의 초기값을 지정해주고 참조 회수를 1로 표시한다.;
}
while(제한 시간시간이 끝나기 전까지 ACK를 기다린다.)
{
    데이터를 전송한다.
    Set_timer(); //시간을 재기 시작한다.
    if(ACK가 도착하였다)
        ACK를 송신자에게 전송하고 프로그램을 완전히
        빠져나간다.
}
제한 시간이 지나도록 ACK가 도착하지 않은 경우는 제한
시간을 늘린다.
Set_timer();
while(제한 시간 안에 ACK가 도착하지 않았다면)
    데이터를 재전송하고 제한 시간을 늘린다.
time=get_time();
제한 시간=(time + n * 제한 시간) / (n+1)을 계산하여 다시 테
이블에 저장한다.;
ACK를 송신자에게 전송;
END
    
```

위의 알고리즘에 사용되는 테이블은 도메인 이름을 키(Key) 값으로 하여 해당 테이블의 위치를 찾아간다. 테이블은 기억 장소를 저장하기 위해 동적 체인 방식(Dynamic Linked List)을 사용한다. 새로운 엔트리가 추가되면 도메인 이름과 함께 제한 시간의 초기값을 지정하여준다. 그리고 참조 회수에 1을 증가시킨다.

위에서 재전송 후 제한 시간을 새로이 구하는 공식인 '제한

시간=(time + 제한 시간 * n) / (n+1)'은 재전송 전에 해당 도메인 을 참조했던 이동 컴퓨터의 수가 n번이었을 경우 새로이 평균을 구하는 방법이다.

3. 멀티미디어 데이터 변환



<그림 2> 데이터 변환 과정

<그림 2>는 수신자들보다 고대역폭을 갖는 송신자가 Agent를 통하여 전송할 때의 과정을 보여주고 있다. 송신자의 멀티미디어 데이터를 수신측 Agent가 받으면 Agent는 멀티미디어 데이터를 전송해야 할 이동 컴퓨터 주소를 알아내어 그것을 키 값으로 자신의 관리하에 있는 도메인의 이동 컴퓨터들의 자원 정보를 담고있는 테이블에서 이동 컴퓨터의 대역폭을 확인한다. 만일 멀티미디어 데이터를 받을 수 없는 대역폭이라면 Agent는 데이터 변환을 하여 이동 컴퓨터에 알맞은 형태로 이동 컴퓨터에게 전송한다. 데이터 변환 과정이 없이 원래의 이미지를 임의 대로 변환시켜서 이동 컴퓨터를 사용하는 사용자에게 보여준다면 디스플레이의 성질을 제대로 반영하지 못하는 오류를 발생시킬 수 있다. 데이터 변환 모듈에서 최적의 형태로 변환시켜서 전송하기 때문에 데이터 변환과정 시 이미지의 손상을 줄일 수 있다[3].

여기에서 사용되는 테이블은 자기 도메인 내의 이동 컴퓨터만을 관리한다. 이동 컴퓨터가 새로이 진입하는 경우에는 테이블에 이동 컴퓨터의 주소를 키 값으로 하여 테이블에 저장하고, 도메인을 벗어나는 경우 테이블에서 삭제한다. 추가와 삭제를 용이하게 하기 위해 동적 체인 저장 방식을 택한다.

이동 컴퓨터가 도메인을 벗어나서 새로 진입함을 인식하는 방식은 크게 두 가지를 들 수 있다. 첫번째는 이동 컴퓨터들이 자신이 속한 도메인의 Agent에게 주기적으로 광고 메시지(Advertisement Message)를 보냄으로써 Agent에게 자신의 존재를 확인시키는 방법이다. 두 번째는 Agent가 도메인 내의 이동 컴퓨터에게 데이터를 전송할 경우에는 Agent의 주소를 전송해야 할 데이터에 적재화(Payload) 시킴으로써 이동 컴퓨터가 기존에 알고있던 Agent의 주소와 비교하는 방법이다[2]. 그러나 위의 두가지 중 한가지만을 고집하는 것은 단점이 있다. 첫 번째 방식인 광고 메시지를 이용하는 경우는 이동 컴퓨터의 전원을 낭비하는 결과를 초래할 수 있고, 두 번째 방식을 사용한 경우는 도메인 내에서 오랫동안 데이터가 전송되어 지지 않을 때 실제로는 이동하지 않았는데도 자신이 무선 환경을 지원하지 않는 곳으로 이동했다는 판단을 할 수가 있다. 그러므로 제한 시간을 설정하고 제한 시간이 되면 이동 컴퓨터가 광고 메시지를 Agent에게 전송하는 방법을 채택하기로 한다. 그렇게 되면 잦은 광고 메시지 전송에 의한

전원 낭비를 줄일 수 있고 데이터의 흐름이 없을 경우 광고 메시지를 Agent에게 전송한 뒤 Agent의 응답을 통하여 자신이 다른 곳으로 이동했다는 오판을 하지 않을 것이므로 효율적일 것이다.

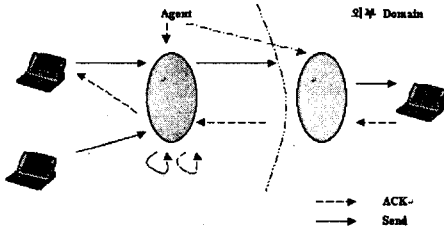
4. Agent의 동작 원리

Agent의 동작 원리를 송수신 측으로 나누어서 설명한다. 송신측 Agent의 동작 원리는 다음과 같다.

- 1) 송신자가 보낸 데이터를 ACK가 도착할 때까지 버퍼에 담고 있다.
- 2) 데이터의 주소를 분석하여 어떤 도메인으로 향하는지를 찾아내고 그 도메인의 주소를 키 값으로 하여 Agent에서 관리하고 있는 제한 시간 보관 테이블에서 제한 시간을 얻어내어 제한 시간을 설정한 뒤 데이터를 수신자에게 보낸다.
- 3) ACK가 제한 시간 안에 도착하였으면 제한 시간을 변경하지 않고 참조 회수만 증가 시키면서 ACK를 송신자에게 보내는 역할만 한다. 그러나 제한 시간 안에 ACK가 도착하지 않았다면 제한 시간을 늘려가며 ACK를 받을 때까지 데이터를 재전송한다.

반면에 수신측 Agent의 동작 원리는 다음과 같다.

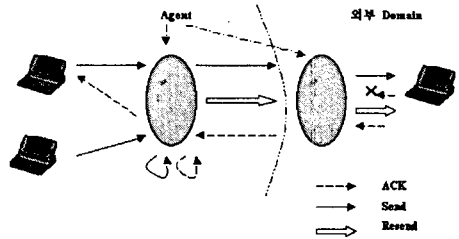
- 1) 송신자가 보낸 데이터를 받으면 어떤 이동 컴퓨터에게 전송해야 하는지 목적지 주소를 알아낸다.
- 2) Agent가 관리하고 있는 도메인의 각 이동 컴퓨터에 대한 자원 정보를 유지하고 있는 테이블에서 이동 컴퓨터의 대역폭을 찾아서 송신자가 보낸 데이터를 최적으로 데이터 변환을 한 뒤 이동 컴퓨터에게 전송한다.



<그림 3> 정상적으로 데이터 전송이 일어난 경우

그림 3은 ACK가 정상적으로 송신자에게 전달되는 과정을 도시화한 것이다. 즉, Agent를 경유하여 데이터를 전송할 때, Agent는 그 데이터의 목적 도메인을 얻어내어 제한 시간 테이블에서 그 도메인의 평균 제한 시간을 알아내서 제한 시간을 설정하고 ACK를 기다린다. 데이터를 받은 수신자측 Agent는 수신자의 대역폭에 맞게 데이터를 변환하여 송신자에게 전송하여 준다. 위의 경우는 제한 시간 안에 ACK가 도착하였으므로 수신자측 Agent는 데이터를 재전송할 필요없이 ACK를 송신자에게 전달한다.

같은 도메인 안의 또 다른 이동 컴퓨터가 외부 도메인으로 데이터를 전송하고자 한다면 Agent를 통하여 앞의 과정과 마찬가지로 제한 시간 값을 알아내어 전송을 하게된다. 이 과정에서 비디오나 오디오 같은 멀티미디어 데이터를 전송할 경우에는 데이터 변환이 필요하지 수신자측 Agent에서 데이터 변환 테이블을 참조하여 이동 컴퓨터에게 적당한 대역폭으로 데이터를 변환시켜서 목적지로 전송한다.



<그림 4> Agent가 재전송을 한 경우

<그림 4>는 ACK가 중간에 소실되거나 제한 시간 안에 ACK가 도착하지 않은 경우이다. 이 경우에는 송신측 Agent에서 ACK가 올 때까지 저장하고 있던 데이터를 다시 전송한다. Agent는 기존의 제한 시간과 재전송한 데이터에 대한 ACK가 도착한 시간을 이용하여 테이블을 갱신한다. 이때는 수신 도메인 측의 네트워크에 정체가 일어났음을 예상할 수 있다.

위의 모델의 장점은 다음과 같다.

- 1) 이동 컴퓨터인 송신자가 재전송에 대해 책임이 없기 때문에 이동 컴퓨터의 전원 절약의 효과를 가져온다
- 2) 네트워크 상황을 반영할 수 있고, 그 정보를 같은 도메인 안의 이동 컴퓨터들이 공유할 수 있다.
- 3) 새로운 이동 컴퓨터가 도메인에 진입하는 경우에도 이동 컴퓨터들이 얻어낸 제한 시간의 평균값을 사용하고, 단지 주소와 대역폭을 자신이 속한 도메인의 Agent에 추가하면 되기 때문에 확장성이 뛰어나다.

5. 결론

본 논문은 데이터 손실이 잦고 저대역폭의 이동 컴퓨터를 사용하는 무선 환경의 특성을 고려하여 새로운 데이터 전송 메커니즘을 사용하는 Agent를 제안 하였다. 즉, Agent가 송신자 대신 재전송을 담당하여 송신자의 부담을 줄이고 제한 시간을 제어하여 재전송의 효율을 증가시켜 보려 하였다. 또한 이미지, 음성 등의 무선 멀티미디어 데이터를 전송받은 경우 이동 컴퓨터에 최적인 상태로 변환시켜주는 기능을 Agent에 추가하였다. 기본적인 아이디어는 SPAND라는 툴킷의 동작 원리로부터 획득했지만 SPAND의 개념을 무선 통신 환경에의 효율적인 전송과 품질을 보장해주기 위한 방향으로 개선시켜서 지능형 Agent를 설계하였다.

앞으로의 실제 Agent를 구현한 후 성능 평가를 해서 무선 환경에 이 Agent를 실용적으로 사용할 수 있도록 개발 예정이다.

참고 문헌

- [1] Mark Stemm, Randy H. Katz, Srinivasan Seshan, 'SPAND: Shared Passive Network Discovery', USITS '97, Dec., 1997.
- [2] Chares E. Perkins, 'Mobile IP, Design Principles and Practices', ADDISON-WESLEY, 1997.
- [3] Eric A. Brewer, Randy H. Katz 외, 'A Network Architecture for Heterogeneous Mobile Computing', IEEE Personal Communications, October, 1998.