

MSS를 사용한 무선 네트워크 환경을 위한 AVI-based 캐쉬 무효화 알고리즘의 설계 및 성능 평가

박술예⁰ 김길용
 부산대학교 컴퓨터공학과
 (seypark, gykim)@hyowon.pusan.ac.kr

AVI based cache invalidation algorithm using MSS for wireless environment and performance study

Seul-Yea Park⁰ Gil-Yong Kim
 Dept. of Computer Engineering, Pusan National University

요 약

휴대폰의 사용이 급증하고 스마트폰, PDA와 같은 무선 단말기 등의 개발이 점차적으로 확대됨에 따라 이런 휴대용 단말기를 이용하여 인터넷 환경이 아닌 무선 네트워크를 이용하여 데이터 통신을 하는 일이 급속도로 증가되고 있다. 이런 환경에서 무선 단말기에서 캐쉬를 사용한다고 하면 아직은 적은 메모리와 스토리지 용량을 가진 무선 단말기에서의 데이터 통신을 좀 더 효과적으로 이용할 수 있을 것이다.

특히 본 논문에서는 모바일 클라이언트(MC) 내의 캐쉬된 데이터가 무효화(invalidation)될 때 서버에서 보내야 하는 무효성 보고 패킷(IR)의 발생과 관련하여 그 패킷 수를 줄여서 대역폭이 낮은 무선 환경에 적합한 알고리즘을 논의한다. MSS(Mobile Switching Station)를 이용한 계층적 캐쉬 알고리즘을 제시하며 데이터의 수정 주기를 이용한 AVI_based 알고리즘을 제시한다. MSS는 MC의 캐쉬의 내용을 알고 있는 state server의 역할을 하게 된다. MC는 캐쉬의 데이터의 avi가 이전보다 짧아졌다는 IR을 받거나 현재 시간과 avi값에 의해 그 데이터가 무효하다는 것을 판단한다. 본 논문의 마지막은 제시된 알고리즘의 시뮬레이션 설계를 통해 AVI 값의 주기성에 따른 성능 평가를 보여준다.

1. 서 론

무선 단말기 사용의 급증으로 통신 환경이 기존의 유선 네트워크에서 무선 네트워크 환경으로 확대되고 있다. 휴대폰을 이용한 통신 뿐만 아니라 PDA 나 스마트폰과 같은 모바일 단말기에 CDMA모듈을 장착해 무선 통신을 지원하는 기기들이 계속해서 개발되어 나오고 있다. 이러한 무선 단말기를 통한 데이터 통신은 낮은 대역폭과 자주 끊기는 불안정한 환경으로 인해 기존의 인터넷 환경과 같이 데이터 통신을 하기에는 많은 무리가 따른다. 모바일 단말기의 저장 능력에는 한계가 있어 PC와 같이 고용량을 지원하지 못한다.

우리는 이런 모바일 단말기를 이용한 무선 환경에서의 데이터 통신의 성능을 높이기 위해 캐쉬(cache)의 사용을 효율적으로 이용해야 한다. 하지만 적은 메모리 용량을 지니는 모바일 단말기에서의 캐쉬 이용은 기존의 인터넷 환경과 다른 방법으로 접근해야 한다. 본 논문에서는 캐쉬 내용의 유효성(validity)를 보장하도록 하는 방법으로 사용되는 무효성 보고(Invalidation Reports) 패킷의 발생과 관련한 문제들을 논의한다. 되도록 적은 수의 무효성 보고 패킷의 발생으로 서버에서의 무효성 보고 패킷의 발송으로 서버와 모바일 클라이언트 사이의 데이터 일관성 유지가 가능하도록 하는 방법을 연구한다.

이 논문에서는 두 가지 정책으로 무효성 보고 패킷의 발생 수를 줄인다.

먼저 무선 단말기를 이용한 데이터 통신 환경의 3단계로

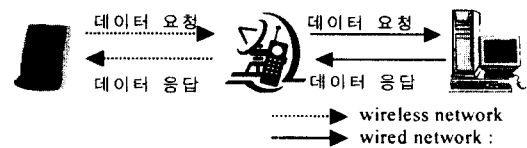


그림 1. 무선 단말기를 이용한 데이터 통신

구성한다. 현재 휴대폰을 이용해 데이터 통신을 하는 환경을 살펴보면 그림1과 같다.

모바일 클라이언트와 서버가 직접적인 무선 통신을 하는 환경이 아니라 기지국을 거쳐 데이터를 요청하고 기지국을 통해 데이터를 받게 된다. 여기서 모바일 클라이언트와 기지국 사이는 무선 통신이지만 기지국과 서버사이는 기존 인터넷 환경인 것이다. 이 기지국에서 모바일 클라이언트에 대한 정보를 가지고 있는 stateful 한 캐쉬 환경을 구축하여 무선 통신을 통한 패킷 수를 줄일 수 있다. 이 논문에서는 이러한 기지국을 MSS(Mobile Switching Station)으로 정의한다.

두번째로 데이터의 수정 주기(update interval)을 나타내는 AVI를 이용하여 서버에서 기지국으로 보내는 무효성 보고 패킷의 수를 줄인다. 이는 데이터의 수정 간격이 주기적인 경우 매우 효율적인 방법이다.

본 논문의 구성은 다음과 같다. 2장은 계층적 캐쉬 모델과 캐쉬 데이터 일관성 유지를 위한 서버와 MSS,

그리고 모바일 클라이언트 (MC) 각각에서의 알고리즘을 제시한다. 3장은 AVI 주기에 따른 앞서 제시한 알고리즘의 성능 평가를 한다. 4장에서는 결론 및 향후 연구를 제시한다.

2. AVI_based 무효성 보고 패킷의 발생과 캐시 데이터 일관성 유지

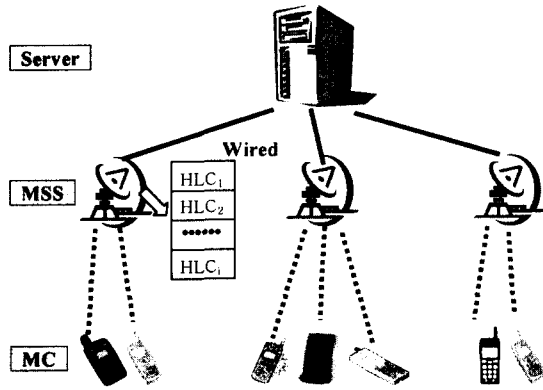


그림 2 모바일 단말기를 이용한 무선 데이터 통신 모델

본 논문에서 서버는 모바일 클라이언트의 캐시의 상태를 모르는 **stateless server**이다. 반면 MSS는 MC의 캐시 상태를 아는 **state server**로 구성한다.

실제 서버의 데이터베이스에 저장되어진 데이터들의 수정 패턴을 보면 주기적인 수정 패턴을 가진 경우가 많다. 예를 들어 뉴스라든지 날씨 정보 같은 경우 서버 측에서 주기성을 가지고 데이터를 수정하는 경우가 많다. 데이터의 수정 패턴을 파악해 서버에서 AVI(Absolute Validity Interval)를 측정할 수 있다. 즉 서버 데이터의 주기적인 수정 간격을 정해놓고 그 간격을 벗어날 경우에만 무효성 보고 패킷을 보낸다.

주기 L의 간격으로 IR을 보내는 **synchronous**한 방법을 채택한다. $T_i = iL$ 인 시간마다 IR을 보내도록 한다. 데이터베이스를 D라고 하고 각 데이터 j에 대한 AVI를 avi_j 라고 하면 Server는 list U_i 를 유지한다.

$$U_i = \{ [j, t_j, avi_j] \in D \mid avi \text{ of } j \text{ is smaller than before and } t_j \text{ is the timestamp of the last update of } j \text{ such that } T_i - L \leq t_j \leq T_i \text{ and } avi \text{ is the absolute validity interval of } j \}$$

여기서 MSS의 역할은 그림 서버와 클라이언트 중간에 위치하여 클라이언트의 캐시 정보를 유지하고 있어야 한다. 즉 **stateful**한 서버의 역할을 해 주는 것이다. 그리고 T_i 시간마다 서버에서 보내 온 IR을 받자마자 자신이 가지고 있는 클라이언트의 캐시 정보를 보고 클라이언트에게 필요한 정보만 보낸다.

$$M_i = \{ [j, t_j, avi_j] \in U_i \text{ and } j \in HLC_i \}$$

즉 MSS에서는 HLC를 참조하여 새로운 IR을 만든다.

이렇게 하면 서버로부터 온 모든 IR을 필요 없는 MC에게 보내지 않아도 되어서 IR의 패킷 수가 줄어들게 된다. 무선 네트워크를 사용하는 IR의 패킷 수가 줄어들게 되므로 **latency** 효과를 볼 수 있다.

클라이언트 측에서는 캐시의 데이터마다 AVI를 가지고 있어서 이를 보고 서버에 연결하지 않고도 그 데이터가 현재 유효한지 아닌지를 판단할 수 있다.

2.1 서버 알고리즘

```

• 데이터마다 수정 패턴을 이용해 AVI 측정
• AVI와 마지막 수정 시간을 나타내는 TS를 데이터와같이 저장
• while(1){
  if (  $T_{i+1} < \text{current time} < T_i$  ) {
    if ( data j is updated ) then
      { if ( (  $\text{current time} - TS_j$  ) < AVI ) then
          add [ j ,  $t_j$  ,  $avi_j$  ] in list  $U_i$  ;
        else
           $TS_j = \text{current time}$  ; }
    }
  else if (  $\text{current time} = T_i$  )
    broadcast  $U_i$  into MSS ;
  i++
}
    
```

2.2 MSS 알고리즘

```

• HLC ( Home Location Cache )
  각 MC의 cache에 어떤 데이터가 들어있는지에 대한 정보를 저장
• 서버에서 패킷 도착
  if ( IR 패킷 )
    HLC를 참조하여 새로운 IR을 생성 해당 MC에게만 IR패킷 발송
  else ( 데이터 패킷 )
    MC의 HLC에 id와 TS 저장
    MC로 데이터 패킷을 보낸다.
• MC에서 데이터 요구 패킷 도착
  서버에 데이터를 요구한다.
    
```

2.3 모바일 클라이언트 알고리즘

```

• transaction 발생하여 data 요구 시 서버로 데이터를 요구하기 전에 cache를 먼저 살펴본다.
• if ( cache에 원하는 데이터 id 존재 )
  { // cache안의 데이터의 유효성 판단
    if ( (  $TS_{id} + AVI$  ) <  $\text{current time}$  ) then
      cache 안의 데이터는 무효.
      MSS로 데이터 요구 패킷을 보낸다.
    Else// 주기적인 IR이 올때까지 기다린다.
      add [ j ] in list  $Q_i$  ;
  }
• MSS로부터 패킷 도착
  if ( IR 패킷 )
    IR 패킷 안의 data id에 해당하는 data를 cache에서 invalidation check
  else ( data 패킷 )
    transaction에 데이터 사용
    
```

3. 성능 평가

이번 장은 앞서 제시한 알고리즘의 성능 평가를 위한 시뮬레이션 모델을 제시하고 실험에 따른 결과를 보여준다. 이 논문에서 서버에서 IR (Invalidation Report) 을 발생하는 알고리즘은 데이터의 수정 주기인 AVI 가 중요한 인자가 된다. 수정 패턴이 주기성이 강한 데이터의 경우 본 논문에서 제시한 알고리즘을 이용할 경우 IR의 수를 많이 줄일 수 있다. 서버에서 데이터가 수정될 때마다 IR을 보내지 않아도 모바일 클라이언트에서 원하는 데이터의 AVI와 TS만을 가지고 그 데이터의 유효성을 판단할 수 있기 때문이다. 하지만 수정 패턴이 주기적이지 않은 경우 데이터의 AVI가 계속 바뀌게 되므로 서버에서 계속 IR을 보내야 한다.

3.1 환경 설정

Database size	1000 items
Item size	1000 bytes
Mean update interval	600 sec
Number of MSS	1
IR Broadcasting interval (in MSS)	120 sec
Number of client per one MSS	10
Disconnect probability	0.1
Mean disconnect interval	1000 sec
Query report size	64 byte
Hot spot	100 items of Database
Cold spot	900 items of Database
Hot spot access probability	0.8
Downlink bandwidth(MSS->MC)	1000 kbps
Uplink bandwidth(MC->MSS)	10 kbps
Downlink bandwidth(Server->MSS)	100 Mbps
Uplink bandwidth(MSS->Server)	100 Mbps

표 1. 시뮬레이션 모델을 위한 parameter 설정

표1은 시뮬레이션을 위한 parameter들이다.

하나의 서버와 하나의 MSS 에 10개의 클라이언트로 구성된 모델이다. 무선 네트워크의 속도는 downlink 가 1000 kbps, uplink는 10 kbps로 가정하고 실험을 하였다.

3.2 실험결과 및 분석

이 실험에서 변수로 둔 parameter는 수정 패턴의 주기성이다. 데이터의 AVI값이 그 이전 값보다 짧아져 IR을 보내게 되는 경우의 수를 변화 시켜 가며 실험을 하였다.

그 결과 데이터의 수정 패턴이 일정한 주기성이 없으면 없을수록 모바일 클라이언트에서 MSS로 데이터를 새로이 요구하는 수가 많이 발생했다. 즉 서버에서 데이터들의 AVI 값들이 변화가 심하여 IR을 많이 발생하게 되면 모바일 클라이언트의 캐쉬의 값이 무효화 되는 경우가 많이 발생하게 되어 up query 가 많이 발생하는 결과를 가져왔다.

또한 IR을 많이 발생하여 down link의 패킷 수도 많아져 그 만큼 평균 response time을 느리게 하였다.

4. 결론 및 향후 연구 방향

본 논문에서는 무선 네트워크에서 모바일 단말기를 이용한 데이터 통신의 성능을 향상시키기 위해 모바일 클라이언트의 캐쉬를 지원하도록 한다. 이 캐쉬의 데이터와 서버의 데이터 간의 coherency 유지를 위한 방법인 IR 발생을 위한 알고리즘을 제시하였다 데이터의 유효한 간격을 나타내는 AVI를 이용하여 서버에서 보내는 무효성 보고의 횟수를 줄이고, 서버와 클라이언트 중간에 있으면서 서버와는 유선 네트워크로 연결되어 있는 MSS통해 무선 네트워크를 통해 모바일 클라이언트로 보내지는 IR의 수를 줄인다.

이 MSS는 자신의 관리 영역에 있는 모바일 클라이언트의 캐쉬 데이터에 관한 정보를 알고 있기 때문에 서버로부터 받은 무효성 보고를 모든 클라이언트로 브로드캐스팅 하지않고 필요한 클라이언트만 보낼 수 있다. 서버에서의 IR발생의 원인이 데이터의 AVI값이 이전 값보다 적어졌을 때만 발생한다. 즉 AVI값의 일정할 경우 이 알고리즘은 매우 효과적이지만 그 값이 이전 값보다 적어지는 경우가 많이 발생하는 수정 패턴을 가진 데이터 베이스인 경우 좋은 성능을 나타내지 못한다.

향후 더 많은 parameter를 변화 시켜 가며 여러 상황에 따른 실험치를 얻어내고 실제 시스템에 더욱 적합한 알고리즘을 얻어내도록 할 것이다.

5. 참고 문헌

[1]D. Barbara and T. Imielinski. Sleepers and Workaholics : Caching Strategies in Mobile Environment. Very Large Databases Journal, December 1995
 [2]J. Jing, A. Elmagarmid, A. helal, and R. Alonso. Bit-sequences : An adaptive cache invalidation method in mobile client/server environments, Mobile Networks and Applications, 1997
 [3]Q. Hu and D. K. Lee. Cache algorithm based on adaptive invalidation reports for mobile environments. Cluster Computing, 1:39-50, 1998
 [4] S.Khurana and A. Kahol and S.K.S. Gupta and P.K.Srimani : An Efficient Cache Maintenance Scheme for Mobile Environment. Distributed Computing Systems, 2000. Proceedings. 20th International Conference on , 2000
 [5]Joe Chun-Hung Yuen and Edward Chan and Lam, K.-Y.and Leung, H.W. : An Adaptive AVI-based Cache Invalidation Scheme for Mobile Computing Systems. Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on , 2000
 [6]SungHun Nam , Uyoung Jung and ChongSun Hwang : An Efficient Cache Invalidation Scheme for Mobile Wireless Environments , IEEE 2001