모바일 애드혹 네트워크에서 로드 밸런싱을 위한 경로캐쉬 정보 관리 기법

김영덕[O1], 유홍석[2], 양연모[1]
대구경북과학기술연구원[1], 경북대학교 컴퓨터공학과[2]
ydkim@dgist.org, hsyoo@monet.knu.ac.kr, yangym@dgist.org

A Route Cache Management Scheme for Load Balancing in Mobile Ad Hoc Networks

Young-Duk Kim[O1], Hong-Seok Yoo[2], Yeon-Mo Yang[1]
Daegu Gyeongbuk Institute of Science and Technology[1]
Dept. of Computer Engineering Kyungpook National University[2]

Abstract - 최근 자가 구성이 가능한 모바일 애드 혹 네트워크상에서 통신을 지원하기 위하여 DSR, AODV등 많은 On Demand 라우팅 프로토콜들이 제안되었다. 이들 라우팅 프로토콜들은 경로탐색과정에서 Route Request(RREQ) 패킷을 브로드캐스트하여 목적 노드로부터 Route Reply (RREP) 패킷을 받음으로써 라우팅 경로를 설정한다. 이때 경로탐색의 지연을 최소로 줄이고자 과거 경로탐색 정보 및 라우팅 정보를 경로캐쉬에 저장하여, 중간노드에서 RREP를 대신 보낼 수 있다. 이러한 캐쉬 기법은 간단하면서 매우 효율적인 방법이지만, 캐쉬정보가 잘못된 경우 불필요한 지연을 일으킨다. 더욱이 캐쉬정보를 이용할 경우, 특정 노드에게 보내는 데이터가 집중되는 문제가 발생하여 혼잡의 원인이 된다. 본 논문에서는 이러한 혼잡을 해결하고자 캐쉬정보를 동적으로 관리하여, 혼잡의 원인이 되는 RREP를 억제하고자 한다. 이를 위하여 혼잡 임계값 및 파라미터를 정의한다. 마지막으로 실험을 통하여 네트워크 트래픽이 많을 경우 제안된 기법을 적용했을 때, 더 우수한 성능을 가짐을 보인다.

1. 서 론

모바일 애드 혹 네트워크는 모든 무선 노드들이 이동성을 가지고 Base Sation과 같은 Infrastructure가 없이 동적으로 자가 토폴로지 구성이 가능한 네트워크이다. 이러한 환경에서는 노드들의 이동 및 무선 채널의 특성에 따라 경로가 다양하게 바뀔 수 있으며, 노드들은 상황에 따라 경로를 재탐색하고 유지 및 복구를 할 수 있어야 한다. 이러한 동적인 네트워크를 지원하기 위하여 많은 라우팅 프로토콜들이 제시되었으며, 대표적인 예로 DSDV (Destination Sequence Distance Vector) [1], DSR (Dynamic Source Routing) [2], AODV (Ad-hoc On-demand Distance Vector) [3] 등이 있다. DSDV는 Table Driven방식의 프로토콜로써 주기적인 컨트롤메시지를 교환하여 라우팅 테이블을 유지한다. 이를 통하여 즉각적인 경로정보의 제공이 가능하지만 컨트롤메시지의 오버헤드가 상당히 큰 단점이 있다. DSR과 AODV는 Table Driven 방식의 문제점을 해결하고자 On-Demand 방식으로 동적인 경로탐색을 제안하였다. 데이터 전송이 요구될 때만 컨트롤메시지를 사용하도록 하여 오버헤드를 줄였으며, 특히 이미 라우팅을 완료하였던 경로에 대하여 해당 정보를 경로캐쉬에 저장함으로써 빠른 탐색을 지원한다. 이러한 경로 캐쉬 기법은 소스노드의 RREQ를 받았을 때, 중간노드가 목적노드가 아니더라도 자신의 캐쉬정보를 이용하여 RREP를 전송하도록 하는 방법이다. 그러나 캐쉬 기법은 특정 노드로 데이터를 계속 전송함으로써 해당 노드에 트래픽 집중 현상이 더욱 심각해지는 문제점이 있다. 이 때, 혼잡이 발생한 노드에서는 버퍼 오버플로우, 긴 패킷 전송시간, 낮은 패킷 전송률, 그리고 오버플로우 후 새롭게 경로탐색을 위한 컨트롤 패킷의 큰 오버헤드 문제가 발생하게 된다. 더욱이 혼잡이 발생한 노드는 많은 데이터를 전송하기 위하여 더 많은 에너지를 소비하게 되므로 네트워크 분할 문제도 초래할 수 있다. 이러한 문제들은 최근 무선 링크상에서 실시간 데이터 및 및 멀티미디어 데이터에 대한 수요 증가로 인하여 더욱 문제가 심화될 수 있다.

본 논문에서는 혼잡이 발생하였을 때, 동적으로 경로 캐쉬의 이용을 억제하여 RREP의 전송을 막는다. 이로써 특정 노드에 대한 트래픽 집중을 막을 수 있다. 또한 혼잡을 정의하는 기법들도 함께 제안한다. 본 논문의 순서는 다음과 같다. 2장에서 관련된 선행 연구로써 로드밸런싱을 고려한 라우팅 프로토콜 및 경로캐쉬 이용 프로토콜들을 살펴보고, 3장에서 제안된 프로토콜을 소개한다. 4장에서 실험을 통한 성능 분석을 설명하고 5장에서 결론을 맺는다.

2. 관련 연구

2.1 로드밸런싱을 고려한 라우팅 프로토콜

라우팅 기법으로 로드 밸런싱을 고려하는 논문으로는 DLAR (Dynamic Load Aware Routing Protocol) [4]이 있으며, 트래픽 로드를 버퍼링된 패킷의 개수로 정의한 뒤, 경로 탐색과정에서 최소의 로드값을 갖는 경로를 선택하도록 하였다. MCL (Routing Protocol with Minimum Contention Time and Load Balancing) [5]에서는 채널 획득을 위해 경쟁하는 주위의 노드들의 개수를 트래픽 로드로써 정의하였으며 CRP (Congestion-adaptive Routing Protocol) [6]에서는 혼잡의 상태를 3가지로 분류하여 각 상태에 따라 라우팅을 하였다. 상기의 프로토콜들은 혼잡이 발생한 경로를 회피하거나 대체경로를 만들어 라우팅을 하는 기법들로써 캐쉬정보를 사용하지 않음으로써 경로상의 모든 노드들을 모니터링하여 트래픽 정보를 수집한다. 그러나 정확한 트래픽 모니터링을 위하여 경로 캐쉬 메커니즘을 완전히 억제함으로써 혼잡이 발생하지 않은 경우에는 오히려 경로 탐색 시간이 지연될 우려가 있다. 이는 결국 네트워크 Throughput 성능을 감소시키는 결과를 초래한다.

2.2 경로 캐쉬 정보를 이용한 프로토콜

Workload Based Load Balancing [7] 은 특정 노드에 집중된 트래픽량을 Workload로 정의하고 특정 임계값을 선정하여 임계값을 넘는 값일때 RREQ 패킷을 drop함으로써 경로 탐색과정에 의도적으로 참여를 억제시킨다. 이를 통해 RREQ 패킷의 flooding 수를 줄일 수 있으며, 다른 노드들을 통하여 라우팅하게 됨으로써 로드 밸런싱을 이룰 수 있다. 그러나 이웃의 다른 노드들의 상황을 고려하지 않은 무절제한 RREQ 패킷의 drop은 경로탐색실패로 이어질 수 있으며, 이로 인하여 네트워크가 분리되는 결과를 초래할 수 있는 문제가 있다.

EAODV [8] 에서는 경로분산을 위한 선택적인 경로 캐쉬 기법을 제안하였다. 즉, 트래픽 경로의 분산을 위해 캐쉬의 소스/목적지 주소 정보를 이용하여 RREQ 패킷의 우회를 유도하였다. 그러나 우회를 통하여 불필요하게 경로가 길어지는 문제가 발생할 수 있으며, 이로 인한 전송지연이 나타나는 문제점을 가지고 있다.

3. 제안된 기법

3.1 기본 동작

일반적인 On-Demand 라우팅 프로토콜에서는 경로탐색을 위해 소스 노드는 RREQ 패킷을 브로트캐스트하고 목적지 노드로부터 RREP 패킷을 수신한다. 이 때, RREQ의 전송은 동적인 네트워크 토폴로지상에서 노드가 경로에 대한 정보를 전혀 모를 경우에 수행하는 최선의 방법이다. 한편 네트워크 대역폭 낭비를 줄이고 탐색 지연을 줄이고자 경로 캐쉬를 이용하고 있다. 이때, RREQ를 받은 중간 노드는 자신의 캐쉬 정보를 참고하여 캐쉬 테이블내에 목적주소와 일치하는 정보가 있을 경우, 즉시 소스에게 RREP을 응답한다.

그러나 제안하는 기법에서는 기본적으로 경로 캐쉬를 이용한 상기의 RREP 전송을 억제할 수 있음을 가정한다. 이를 위하여 네트워크내 모든 노드들은 자신의 버퍼길이(Q_{len})를 모니터링하고 있으며, 특정 버퍼 임계값($Q_{threshold}$)을 정의하여 현재 측정한 버퍼길이와의 차이를 비교한다. 동작의 시작은 경로 탐색 과정에서 RREQ를 받은 중간 노드는 먼저 RREQ의 목적지가 자신인지 판단하고, 자신이면 RREP를 소스 노드로 전송한다. 자신이 목적지가 아니면 경로 캐쉬를 읽어 일치하는 정보가 있는지 살펴본다. 일치하는 정보가 없으면 일반적인 방법으로 RREQ를 다시 브로트캐스트 한다. 만약, 캐쉬에 목적지 정보가 있으면 Q_{len} 값과 $Q_{threshold}$값을 비교하여 RREP을 전송할 지 판단한다. 즉, Q_{len}가 $Q_{threshold}$ 보다 크다면 현재 혼잡이 발생할 우려가 있다고 판단하여 RREP의 전송을 억제한다. 이로써 소스노드는 혼잡이 발생하는 노드를 피하여 라우팅을 할 수 있으며, 혼잡으로 인한 버퍼 오버플로우를 막을 수 있다. 이때 중간노드는 RREP를 억제하는 대신 RREQ를 다시 재전송하지만, 혼잡가능성으로 인한 Queueing 지연의 증가로 인하여 목적지로부터 RREP를 먼저 받을 확률은 줄어든다. 따라서 트래픽이 분산된 라우팅을 이룰 수 있다.

또한 부수적으로 RREP를 억제함으로써 부정확한 캐쉬정보의 이용을 막을 수 있다. 일반적으로 캐쉬정보는 최근에 전송이 성공한 경로정보를 저장하고 있지만, 이동성이 크고 무선링크의 변화가 큰 에드 혹 네트워크상에서는 토폴로지가 동적으로 변함에 따라 캐쉬정보도 정확성이 떨어지게 된다. 따라서 부정확한 캐쉬정보를 이용할 경우 패킷 손실이 발생하고, 경로를 재탐색함으로써 소요되는 지연이 더 증가된다. 특히, 혼잡이 발생했을 경우의 경로탐색 실패는 더 큰 오버헤드가 되어 네트워크의 성능을 감소시키는 원인이 되므로 RREQ를 통한 최신의 경로 정보를 획득함이 더 유리하다.

3.2 혼잡 모티터링 알고리즘

한 노드가 제안된 기법을 사용하여 RREP를 적절히 억제하기 위해서는 먼저 현재 상태가 혼잡인지 아닌지를 정확히 확인해야 한다. 따라서 제안된 기법에서는 혼잡 탐지를 위한 동적인 메커니즘을 필요로 한다. 즉, 적당한 RREP의 drop은 트래픽 분산에 도움을 주지만, 무절제한 RREP의 억제는 경로 캐쉬의 빠른 경로 탐색의 장점을 충분히 살리지 못하게 된다. 이러한 이유로 2개의 임계값을 정의한다. 첫 번째 임계값은 앞서 언급한 $Q_{threshold}$ 값으로써 현재 버퍼의 길이를 모니터링하면서 그 값을 비교한다. 예로 $Q_{threshold}$ 가 30 이고 노드의 버퍼링된 패킷 수가 30이 초과한다면 혼잡이 발생한 상태라고 판단한다.

그러나 단순히 $Q_{threshold}$ 값만을 사용하여 혼잡을 예상할 경우, 데이터 흐름의 Burst한 성격 혹은 재전송 등의 이유로 일시적인 버퍼 길이 증가를 야기할 수 있다. 이러한 노드의 내부적인 트래픽 특성을 고려하여 정의한 두 번째 임계값이 시간에 대한 임계값 $T_{threshold}$이다. 이 값은 $Q_{threshold}$ 가 만족된 상태에서 패킷이 큐에 저장되었던 시간적 기간을 나타낸다. 즉 패킷의 버퍼링 시간이 $T_{threshold}$ 보다 클 경우 진정한 혼잡상태라고 간주하게 되며, 해당 노드는 RREP 패킷을 억제하여 자신에게로 향하는 트래픽을 완화시키게 된다.

3.3 우선순위 패킷의 처리

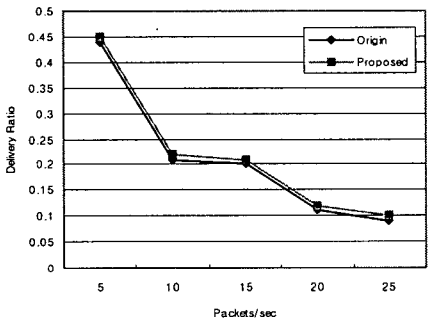
적절한 RREP 패킷의 억제로 혼잡을 예방하고 트래픽을 분산시키는 점에서 본 기법이 효율적이지만, 만약 해당 RREP를 억제한 노드를 지나는 경로가 유일하고 캐쉬 정보가 오류없이 정확하다면 오히려 성능을 감소시키는 원인이 된다. 특히, QoS를 지원하고 우선순위가 높은 패킷을 처리해야 할 경우는 캐쉬를 이용하여 빠른 응답을 요구한다. 이러한 경우에 RREQ 패킷의 option 필드에 Priority Flag을 추가한다. 즉, RREQ 패킷의 Flag가 1로 설정되어 있다면 노드는 캐쉬정보가 정확하다는 가정하에 RREP를 즉시 전송한다. 이때, 해당 패킷에 대해서는 임계값 적용을 하지 않도록 한다. 만약, 혼잡으로 인한 오버플로우가 발생할 경우 QoS의 우선순위에 따라 선택적으로 drop할 수 있도록 한다. 이것에 대한 구체적인 알고리즘은 향후 과제로 남겨두도록 한다.

4. 성능 평가

4.1 실험 환경

제안된 프로토콜의 성능을 평가하기 위하여 NS-2 시뮬레이터 [9]를 사용하였다. 적용된 라우팅 알고리즘은 경로 캐쉬를 사용하는 DSR이며, 1500m*300m 크기의 영역안에 50개의 모바일 노드를 랜덤하게 위치시켰다. 모든 노드들은 최고 10m/s의 속도로 자유롭게 이동하고 pause time은 50으로 설정하였으며 총 300초 동안 실험하였다. 각 노드의 데이터 전송 범위는 250m로 설정했으며, 소스와 목적지 노드 사이의 데이터 연결 수는 20개로 하였고, 트래픽 종류는 CBR(constant bit rate)을 사용하였다. 또한 네트워크 트래픽 로드를 달리 표현하기 위하여 초당 패킷 전송률을 5, 10, 15, 20, 25의 5 가지로 구분했으며, 패킷 크기는 512byte 크기로 실험하였다. 각 노드의 인터페이스 버퍼의 최대 크기는 50으로 설정하였고 $Q_{threshold}$는 45, 20, 10으로 나누어 실험했으며, $T_{threshold}$는 2, 1, 0.5의 경우로 나누어 실험 하였다.

4.1 실험 결과

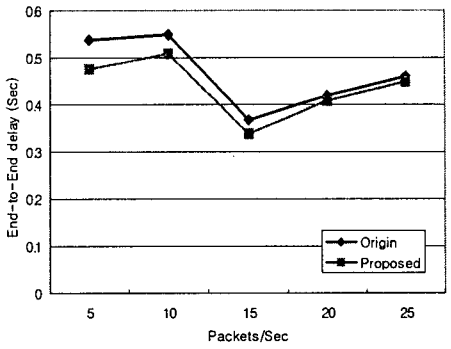


<그림 1> Delivery Ratio

그림1은 트래픽 부하에 따른 패킷 delivery ratio을 보여주는 그래프로써, 제안된 알고리즘이 불필요한 트래픽 집중을 방지하여 기존의 기법보다 더 적은 오버플로우 횟수 때문에 더 우수한 전송률을 가짐을 알 수 있다. 그러나 2가지 기법 모두 트래픽양이 많아짐에 따라 전송률이 떨어지는 경향은 비슷하며, 완전한 혼잡해결은 성취하기 어려움을 알 수 있다.

그림2는 트래픽 부하에 따른 패킷의 end-to-end 지연을 나타낸다. 주어진 알고리즘을 이용함으로써 특정 노드의 혼잡을 분산시키고 재전송을 위한 불필요한 경로 탐색 지연이 줄어듦을 알 수 있다. 그러나 혼잡이 가중될수록 end-to-end 지연은 모두 수렴되면서 증가함을 보인다.

표1은 트래픽 양에 따라 혼잡으로 인하여 버퍼에서 drop된 패킷의 개수를 나타낸다. 이는 그림 1의 전송률에 따른 영향을 반영한 결과라고 볼 수 있다.



<그림 2> End-to-End Delay

<표 1> 혼잡으로 인한 패킷 손실 개수

Packets/Sec	5	10	15	20	25
Origin	13238	37489	46552	85034	108448
Proposed	13077	37138	46036	84812	108151

표2는 실제 혼잡 상태 여부를 결정하기 위하여 각각 다른 버퍼 임계값을 두어 성능을 비교한 결과로써, $Q_{threshold}$ 와 $T_{threshold}$ 값이 각각 20, 1 일 경우 최적의 성능을 보였으며, 기존 방법에 비해 전체적으로 약 5%의 성능향상을 보였다.

<표 2> 임계값의 비교

임계값		15 packets/sec		
$Q_{threshold}$	$T_{threshold}$	Delivery Ratio	End-to-End Delay	Overflow Dropped
45	2	0.21	0.362	46550
20	1	0.20	0.337	46036
10	0.5	0.21	0.342	46258

5. 결 론

모바일 애드 혹 네트워크에서 특정 노드의 혼잡은 오버플로우로 인한 많은 패킷의 손실과 긴 패킷 전송 지연을 일으켜 성능을 크게 감소시킨다. 특히 경로 캐쉬를 사용할 경우 문제는 더욱 심각해진다. 그러나 기존의 로드 밸런싱 프로토콜들은 단지 대체경로를 찾는 기법이 대부분이며 효율적인 경로 캐쉬 관리 기법은 미흡하였다. 본 논문에서는 실제로 혼잡이 발생한 노드들 모니터링하여 파악하고 경로 탐색 요청을 받으면 현재의 혼잡 상태에 따라 동적으로 RREP를 억제하여 트래픽을 분산시켰다. 이를 위해 두 가지의 버퍼 임계값을 정의하여 실제 혼잡 상황을 정확히 판단하는 방법을 제시하였다. 마지막으로 실험을 통하여 네트워크 부하가 큰 환경에서 제안된 기법이 패킷 전송률, 패킷의 end-to-end 지연 등에서 기존 알고리즘보다 우수한 성능을 가짐을 보였다.

[참 고 문 헌]

[1] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance-vector Routing (DSDV) for Mobile Computers," Comp. Commun. Rev., pages: 234 - 44, October. 1994

[2] David B. Johnson, David A. Maltz, Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," Internet Draft, IETF Mobile Ad hoc Networks (MANET) Working Group.

[3] C. E. Perkins and E. Royer, "Ad-hoc on-demand Distance Vector Routing," Proc.2nd IEEE Wksp. Mobile Comp. Sys. App., 1999.

[4] Sung-Ju Lee and Mario Gerla, "Dynamic Load-Aware Routing in Ad hoc networks," Proceedings of IEEE ICC 2001

[5] Bong Chan Kim, Jae Young Lee, Hwang Soo Lee and Joong Soo Ma, "An Ad-hoc Routing Protocol with Minimum Contention Time and Load Balancing," IEEE Globecom, 2003.

[6] Tran, D.A. and Raghavendra, H., "Routing with congestion awareness and adaptivity in mobile ad hoc networks," IEEE WCNC, 2005

[7] Young J. Lee and George F. Riley, "A Workload-Based Adaptive Load-Balancing Technique for Mobile Ad Hoc Networks", IEEE WCNC 2005

[8] Bong Chan Kim, Hwang Soo Lee, and Joong Soo Ma, "Enhanced Ad Hoc On-demand Distance Vector(EAODV) Routing Protocol with Route Distribution", IEEE VTC 2005

[9] S.McCanne and S. Floyd, "NS network simulator"