

액티브 네트워크에서의 피드백을 이용한 효율적인 자원 할당 방안

정준호⁰ 이중수 이영희
한국정보통신대학원대학교
(jhjung, jslee, yhlee)@icu.ac.kr

Efficient Resource Allocation with Feedback in Active Networks

Joon-Ho Jung⁰ Joog-Soo Lee Young-Hee Lee
School of Engineering, Information and Communications University

요 약

액티브 네트워크에서는 패킷을 전송을 위한 계산과정에서 액티브 노드의 자원을 사용하며, 이렇게 사용되는 자원을 패킷의 특성에 맞게 효율적으로 할당하는 것은 중요한 문제이다. 만약 자원을 효율적으로 분배하지 못하면 특정 패킷에 의해서 액티브 노드의 모든 자원이 소비되어서 다른 패킷을 처리하기 위한 자원을 할당할 수 없게 된다. 또한 악의가 있는 패킷 또는 오류가 발생한 패킷들에 의해서 동일한 문제가 발생할 수 있다. 본 논문에서는 이러한 문제의 해결을 위해서 CPU 스케줄러, 전송 스케줄러, feedback을 핵심요소로 가지는 액티브 네트워크에서의 효율적인 자원 할당 방안을 제안한다.

1. 서 론

액티브 네트워크에서는 다양한 패킷들을 서로 다른 방식으로 전송할 수 있는 환경을 제공함으로써 패킷들의 다양한 서비스 요구사항을 만족할 수 있으며 [1], 네트워크상에 존재하는 라우터와 노드들이 패킷의 특성에 맞게 전송을 하기 위한 계산과정을 수행하고, 이러한 계산을 처리하는 프로그램들을 능동적으로 로딩할 수 있는 환경을 제공한다[2].

액티브 네트워크에서는 액티브 패킷이 계산을 처리하는 프로그램을 운반하거나, 프로그램에 대한 주소를 운반함으로써 패킷을 처리하기 위해 필요한 프로그램들을 전송 받게 된다[9]. 액티브 패킷은 오프라인을 통해서 네트워크에 진입하게 되며, 패킷이 액티브 노드에 도착하게 되면 패킷을 처리하기 위한 프로그램을 로딩하여 전송을 위한 계산과정을 거친 후 다음 노드로 전송된다 [3,4].

액티브 노드는 CPU와 대역폭을 자원으로 사용하여 전송에 필요한 계산을 하게 된다. 다수의 패킷이 노드에 전달 되었을 경우 이들 패킷들 사이에 이러한 자원들을 패킷의 특성에 맞게 효율적으로 할당하는 것은 아주 중요한 문제이다. 만약 액티브 노드가 다양한 특성을 가진 패킷들에게 골고루 자원을 할당하지 못한다면 e-mail과 같이 긴급한 전송이 필요하지 않은 다수의 패킷을 처리하기 위하여 모든 자원을 소비한 뒤 실시간 전송이 필요한 패킷을 처리할 수 없거나, 악의가 있는 사람에 의해 전송된 패킷 또는 오류가 발생한 패킷이 노드의 모든 자원을 소비함으로써 다른 패킷들에게 자원을 사용할

기회를 빼앗는 일이 [5] 발생하게 된다. 따라서 이러한 문제를 방지하기 위해서는 서로 다른 특성을 가진 패킷들에게 한정된 양의 CPU와 대역폭과 같은 자원을 효율적으로 나누어주는 일이 매우 중요하다.

본 논문은 이러한 문제점을 해결하면서 더욱 효율적으로 자원을 할당하기 위한 방안으로서 입력 큐의 상태정보와 피드백된 출력 큐의 상태 정보를 이용하여 CPU 스케줄러가 각 플로우의 상황에 맞게 효율적으로 자원을 할당하는 방안을 제안하고자 한다.

본 논문의 나머지 부분은 다음과 같이 구성되어 있다. 2장에서는 관련된 연구에 대해서 소개하고 3장에서는 효율적으로 자원을 할당하기 위한 새로운 방안을 제안하며 4장에서는 결론을 맺는다.

2. 관련 연구

다양한 특성을 가진 패킷을 처리하기 위한 효율적인 자원관리는 이미 그 중요성이 인식되었으며, 다양한 연구기관에서 연구되어지고 있다.

Ramachandran의 연구[6]는 CPU와 대역폭을 하나의 자원으로 보고 두 자원의 사용량의 총합을 일정하게 유지하는 방안을 제안하고 있다. 특히 DRR (Deficit Round-Robin) 스케줄링 알고리즘[7]을 이용하여 전송 스케줄러가 전송할 패킷의 수량 변화에 좀더 능동적으로 대응할 수 있도록 하고 있으며, 이러한 출력 큐에서의 정보를 다시 CPU 스케줄러에게 피드백 함으로써 효율적으로 자원을 할당하고자

한다. 하지만 대량의 패킷이 전송되었을 때 발생할 수 있는 오버플로우 문제를 해결할 방안을 제시하지 못하고 있다.

Williams의 연구[8]는 중단간의 대역폭을 여러 개의 패스를 통하여 예약하여 사용할 수 있게 함으로써 네트워크의 중단간을 단일 패스를 이용하여 충분한 리소스 할당을 받지 못하는 경우에도 효율적인 접속을 할 수 있는 기회를 제공한다. 이러한 방법은 새로운 패스를 찾고 자원을 할당해야 하는 부담이 있다.

RCANE [9]은 자원의 사용량을 모니터링하며, 자원을 안전하게 접근할 수 있게 함으로써 다양한 Denial-of-Service (DoS) 공격으로부터 안전할 수 있으며, 가상 프로세서의 개념을 이용하여 스케줄링 방식에 따라 보장되는 CPU 할당 시간을 효율적으로 이용하고 있다. 그러나 다양한 세션을 생성하고 관리하여야 하는 부담을 앓고 있다.

SNAP[10]은 계산을 위한 자원을 할당하는 데서 발생하는 문제를 해결하기 위해 새로운 접근을 보이고 있다. 패킷에 대한 처리를 수행하는 프로그램이 액티브 노드의 자원을 일정량 이상을 사용할 수 없도록 함으로써 새로운 자원할당을 위한 방안을 제시하고 있다. 특히 액티브 패킷이 프로그램을 포함하는 것은 물론이며 사용될 메모리 (힙, 스택) 영역을 포함하고 있어서 노드가 패킷의 처리에 필요한 리소스의 양을 쉽게 예측할 수 있다. 하지만 패킷이 메모리 영역을 포함하고 있어서 전송되는 패킷의 사이즈가 증가하는 부담을 가지고 있다.

Darwin [11,12]은 중단간의 패스를 구성하고, 그 사이에 존재하는 자원을 하나의 트리로 구성하여 관리하는데, Beagle이라는 신호 프로토콜을 이용하여 실제적인 네트워크 자원을 할당하며, 어플리케이션의 특성에 맞는 서비스를 제공하기 위해서 delegate를 사용한다. 하지만 트리 형태로 자원을 추상화하고 관리하는데 많은 부담이 발생하게 된다.

3. 피드백을 이용한 효율적인 자원 할당

3.1 고려 사항

패킷의 특성에 맞는 전송을 위해서는 그에 맞는 계산과정이 필요하며, 이러한 계산을 위해서 필요한 자원의 양 또한 서로 다르다. 효율적으로 자원을 할당하지 못하면 특정 패킷에만 자원이 집중되고, 나머지 패킷들은 충분한 자원을 할당 받지 못하게 된다. 충분한 자원을 할당 받기 위해서 패킷들은 큐에서 대기해야 하며, 이것은 패킷의 전송 속도가 늦어지는 큰 장애 요인이 된다. 따라서 원활한 패킷의 전송을 위해서는 모든 패킷의 특성을 고려하여 상황에 맞게 효율적으로 자원을 할당할 수 있는 새로운 자원할당 방안이 필요하다.

본 논문은 중단간 전송속도에 지장을 주지않으면서 액티브 노드의 자원을 충분히 사용할 수 있는 효율적인 자원 할당을 위해서 CPU 스케줄러, 전송 스케줄러, feedback을 핵심요소로 가지는 새로운 패킷 스케줄링 기법을 소개하는데 그 목적을 둔다.

3.2 시스템 구조

본 논문에서 참조하는 액티브 노드의 구조는 CPU 스케줄러, 전송 스케줄러, 피드백의 세 가지 중요 부분으로 이루어져있다. 그림 1에서 보는 바와 같이 각각의 플로우-출발지와 목적지가 같고 동일한 라우터를 통과하며 요구되는 서비스 요구가 같은 패킷의 흐름-마다 큐를 할당하여 플로우를

분리시켰으며, 두개의 스케줄러에 의해서 패킷의 입출력이 조절되는 구조이다. 특히 피드백은 전송을 위한 큐의 상태를 CPU 스케줄러에게 전송함으로써 효율적으로 CPU 스케줄러가 상황에 맞게 효율적으로 자원을 할당하도록 한다.

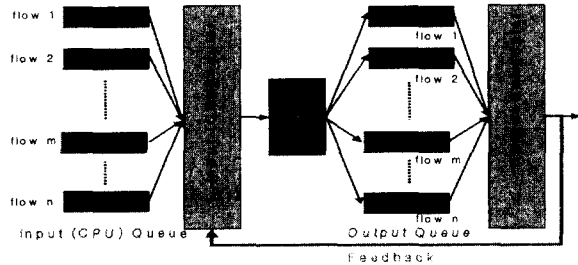


그림 1. 액티브 노드의 구조

스케줄러는 빈 큐를 검사하는 부담을 줄이기 위해서 큐에 하나 이상의 패킷을 가지고 있는 큐의 리스트인 Active_list를 유지하며, 선점 가능한 DRR (Deficit Round-Robin) [7] 스케줄링 알고리즘을 사용함으로써 과도한 패킷의 전송으로 인해 발생할 수 있는 오버플로우 문제를 처리할 수 있게 된다.

3.3 Operational model

네트워크 카드를 통해서 전송 받은 패킷은 CPU에서의 계산을 위해서 패킷이 속한 플로우의 입력 큐에 넣어진다. CPU 스케줄러는 계산할 패킷을 입력 큐로부터 선택하고, 계산이 끝난 다음 패킷은 출력 큐에 넣어지고 전송 스케줄러는 전송을 위해서 출력 큐로부터 패킷을 전송하게 된다. 본 논문에서 모든 액티브 패킷은 CPU와 대역폭을 사용하며 전송을 위한 계산과정이 필요하다고 가정한다.

3.3.1 패킷 스케줄링

액티브 노드에서는 CPU 스케줄링과 전송을 위한 패킷 스케줄링의 두 가지 알고리즘이 노드 운영체제에 의해서 실행되어진다. CPU 스케줄링 알고리즘은 피드백된 출력 큐의 상태와 입력 큐의 상태 정보를 이용하여 각 플로우의 특성과 상황에 맞는 효율적인 자원할당을 한다.

CPU 스케줄링

스케줄러는 입력 큐로부터 하나의 패킷을 선정하고, CPU 사용시간을 할당하고, 계산이 끝나면 출력 큐로 내보내어 전송 스케줄러가 패킷을 전송할 수 있도록 한다.

```

Loop
  For each backlogged flow(i)
    Check Overflow then select flow(i)
    Get round Quantum(i)
    Set Deficit value [DC(i) = Quantum(i) + DC(i)]
    While [DC(i) > consumed CPU time for flow(i)]
      Processing packet(p)
      Calculate consumed CPU time for flow(i)
      [Consumed CPU time for flow(i) = CPU requirement
      for packet(p) + Consummed CPU time for flow(i)]
      Save the Deficit value [DC(i) = DC(i) - Consumed CPU
      time for flow(i)]
    Increase round
  End
    
```

그림 2. CPU 스케줄링 알고리즘

여기서는 각각의 플로우가 각자의 큐를 이용하여 전송되는 패킷을 보관하기 때문에 큐로부터 패킷을 선정하는 것이 아주

중요하게 작용하게 된다. 스케줄러는 DRR [7]을 사용함으로써 이전 라운드에서 사용하지 못한 라운드 퀀텀을 축적하여 사용할 수 있기 때문에 처음에 설정된 범위 내에서 능동적으로 자원을 사용할 수 있다. 그림 2는 스케줄링 알고리즘을 기술하고 있다.

스케줄링 알고리즘에서 보듯이 라운드 퀀텀 값은 전체적인 계산 시간을 좌우한다. 따라서 그림 3에서처럼 플로우의 특성에 맞게 퀀텀 값을 충분하게 설정함으로써 DRR의 초기 적응 속도를 향상 할 수 있게 된다. N은 현재 처리해야 할 패킷을 저장하고 있는 큐의 개수이다.

```

Get round Quantum(i)
  Presource = Pre-specified amount of CPU time
  Aresource = Available amount of CPU time
  Cresource = Total Allocated CPU time / N
  Resource = MIN(Presource, Aresource)
  Quantum = MIN(Resource, Cresource)
Return Quantum
End
    
```

그림 3. 퀀텀 값을 계산하는 알고리즘

그림 4에서처럼 과도한 패킷 전송은 출력 큐가 threshold를 넘어서 오버플로우가 발생할 가능성이 있는데 이런 경우 CPU 스케줄러는 출력 큐에서의 상황을 피드백을 통해서 알게 되고, 이와 동시에 입력 큐의 상태를 확인하여 입출력 큐 모두가 패킷을 수용할 충분한 여유가 있는 플로우의 라운드 퀀텀을 선점하여 오버플로우를 미리 극복할 수 있게 된다.

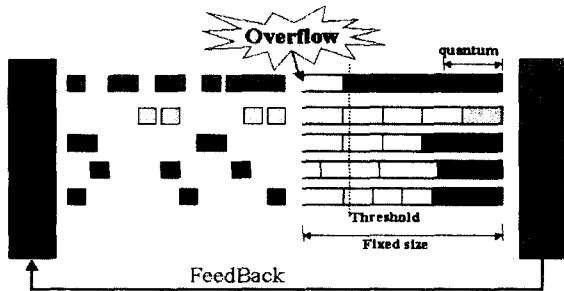


그림 4. 과도한 패킷 전송에 의한 큐의 상태변화

3.3.2 피드백

액티브 네트워크에서는 CPU에서의 계산에 얼마만큼의 시간이 소요되는지 알 수 없기 때문에 전송 큐에서는 패킷의 도착속도를 예측할 수 없게 된다. 따라서 전송 큐에서는 모든 패킷들의 CPU 사용량이 적어서 전송 큐의 보관 능력을 넘어서게 되거나 CPU 사용량이 많은 경우 또는 그렇지 않은 경우와 같이 다양한 특성을 가진 패킷들이 동시에 존재하게 될 것이다. 이때 CPU 스케줄러가 계산을 천천히 처리함으로써 전송 큐의 보관 능력 내에서 패킷을 효율적으로 처리할 수 있도록 하거나, 계산을 빨리 처리함으로써 전송 큐의 공간을 충분히 사용하게 될 것이다. 피드백 기능은 이와 같이 다양한 정보를 정기적으로 보고 함으로써 CPU 스케줄러가 상황에 맞게 동작하도록 한다.

4. 결론

본 논문에서는 액티브 네트워크에서 문제가 되고 있는 자원 할당에 대한 기존의 몇몇 연구를 살펴보았으며, 좀 더 플로우의 특성에 맞는 자원할당을 하기 위해서 플로우마다 큐를 가지는 구조와 출력 큐의 상황을 전송하기 위한 피드백,

전송할 패킷을 선택하는 전송 스케줄러, 그리고 피드백 정보와 입력 큐의 상태를 분석하여 효율적으로 자원을 할당하는 CPU 스케줄러를 이용하는 방안을 제안하였다. 이러한 효율적인 자원 관리는 액티브 네트워크의 계산 능력이 주는 부담을 줄여줌과 동시에 시스템의 자원 이용률을 한층 높일 수 있을 것으로 기대된다.

참고 문헌

- [1] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A Survey of Active Network Research", *IEEE Communications Magazine*, Vol. 35, No. 1, pp80-86. January 1997.
- [2] D.J. Wetherall, J.V. Guttag, D.L. Tennenhouse, "ANTS: a toolkit for building and dynamically deploying network protocols," *Open Architectures and Network Programming*, 1998 IEEE Page(s): 117-129
- [3] David Wetherall, "Active Networks vision and reality: lessons from a capsule-based system," 17th ACM Symposium on Operating System Principles (SOSP'99), Kiawah Island, SC, December 1999.
- [4] Active Network Working Group, "Architectural Framework for Active Networks," RFC, July 1999
- [5] Michael Hicks, Pankaj Kakkar, Jonathan T. Moore, Carl A. Gunter and Scott Nettles, "PLAN: a Packet language for Active Networks," *Proceedings of the third ACM SIGPLAN international conference on Functional programming*, 1998, Page 86
- [6] V. Ramachandran, R. Pandey, Chan, S.-H.G., "Fair resource allocation in Active Networks," *Computer Communications and Networks*, 2000. Proceedings. Ninth International Conference on , 2000 Page(s): 468-475
- [7] M. Shreedhar, G. Varghese, "Efficient fair queuing using deficit round-robin," *Networking, IEEE/ACM Transactions on* , Volume: 4 Issue: 3 , June 1996 Page(s): 375-385
- [8] D. Williams, Chi Nguyen; A. Fekete, M. Hitchens, B. Kummerfeld, "An Active Network bandwidth reservation scheme," *Open Architectures and Network Programming Proceedings*, 1999. OPENARCH '99. 1999 IEEE Second Conference on , 1999 Page(s): 59-66
- [9] Paul Menage, "RCANE: A Resource Controlled Framework for Active Network Services (BiTeX)," *In the Proceedings of the First International Working Conference on Active Networks (IWAN'99)*, July 1999
- [10] J.T. Moore, M. Hicks, S. Nettles, "Practical programmable packets," *INFOCOM 2001. Proceedings. IEEE* , Volume: 1, 2001 Page(s): 41-50
- [11] P. Chandra, A. Fisher, C. Kosak, Ng, T.S.E.; P. Steenkiste, E. Takahashi, Hui Zhang, "Darwin: customizable resource management for value-added network services," *Network Protocols*, 1998. Proceedings. Sixth International Conference on , 1998 Page(s): 177-188
- [12] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A Survey of Active Network Research", *IEEE Communications Magazine*, Vol. 35, No. 1, pp80-86. January 1997