

동기식 이더넷에서 Virtual Clock을 이용한 큐잉 방법

김준철^{○*} 김민준^{*} 엄종훈^{**} 권용식^{**} 김승호^{*}

^{*}경북대학교 컴퓨터공학과, ^{**}KT 미래기술연구소

{jckim[○], mjkim}@mmlab.knu.ac.kr, {jheom, yongsik}@kt.co.kr, shkim@knu.ac.kr

A Queuing Method Using Virtual Clock for Audio/Video Bridging

Junechul Kim^{○*} Minjun Kim^{*} Jonghoon Eom^{**} Yongsik Kwon^{**} Sungho Kim^{*}

^{*}Dept. of Computer Engineering, Kyungpook National University

^{**}Advanced Technology Laboratory, KT

1. 서론

IT기술의 급속한 발달과 초고속망의 보급이 확산됨에 따라, 공공기관이나 기업을 중심으로 구축되던 네트워크 환경이 가정 내의 디지털 전자기기로 확산되어 가면서 홈 네트워크 산업과 관련기기 시장에 대한 관심이 높아지고 있다. 그리고 가정 내의 통신을 위해서는 이더넷 기술이 각광받고 있다. 동기식 이더넷 (802.3ah Residential Ethernet 또는 802.1 Audio/Video Bridging)은 가정 내 혹은 사무실 내에서 최근 가전기기들 간의 정확한 동기를 맞추고 실시간성을 보장하면서 통신하는 기술을 말한다. 이를 위해 최우선적으로 시간 동기를 맞춘 데이터 중 실시간 데이터는 대역폭을 예약함으로써 QoS를 보장하고 비실시간 데이터에 대해서는 여러 가지 큐잉 방법을 통해서 QoS를 보장하려 한다. 하지만 우선순위가 높거나 같은 패킷들이나 많은 대역폭을 요구하는 트래픽들로 인한 정체상황에서는 연속적인 지연이 발생하고 결국 패킷이 폐기되어버린다는 단점이 있다. 그래서 본 논문에서는 동기식 이더넷에 AR(Average_arrival Rate)을 통한 Virtual Clock을 이용하여 위의 PQ, CQ, WFQ의 방법들보다 QoS를 향상시키는 큐잉 방법을 제안한다.

2. 동기식 이더넷에 고려될 수 있는 큐잉 방법

우선순위 큐잉은 FIFO 큐잉의 단점인 클래스의 구분이 없기 때문에 차등화된 서비스를 제공하지 못하는 문제를 해결하기 위해 만들어 졌다. FIFO 큐잉이 단일 FIFO 큐를 사용하는 것에 비해 3가지 FIFO 큐를 사용한다. 낮은 우선순위 큐에 저장되어 있는 패킷들은 높은 우선순위 큐에 저장되어 있는 패킷들이 모두 서비스 된 이후에 서비스가 된다. 우선순위 큐잉 방식의 장점은 우선순위가 높은 패킷을 우선적으로 서비스 해주는 방식으로 서비스를 제공할 수 있다는 것이다. 이 때문에 실시간 애플리케이션의 지원이 가능하다. 그러나 높은 우선순위 큐에 패킷이 계속해서 입력될 때, 낮은 우선순위 큐에 저장된 패킷이 서비스가 되지 못하는 아사 현상이 발생하는 문제가 발생한다. 또한, 동일한 우선순위의 패킷만 많이 들어오는 경우에는 FIFO 큐처럼 동작이 되는 문제도 생길 수 있다.

커스텀 큐잉은 우선순위 큐잉의 단점인 우선순위가 낮은 클래스의 패킷들이 우선순위가 높은 트래픽에 의해 아사되는 문제를 해결하기 위해 만들어졌다. 커스텀 큐잉은 최대 16개의 FIFO 큐를 사용한다. 관리자가 각 큐에 대해 클래스를 구분해서 각 클래스마다 각각의 큐를 지정해 사용할 수 있다. 우선순위 큐에서 클래스를 4개로 나눠 분리하는 것보다 더 세분화해 분리할 수 있다는 것이 장점이다. 모든 큐의 패킷이 공평하게 서비스되는 것이다. 그러나 이 방식은 트래픽의 특성을 고려하지 않고 서비스 차원에서의 공정성만을 감안하고 있기 때문에, 스케줄링 차원에서 차등화 된 서비스를 제공하는 것은 불가능하며, 관리자의 잘못된 설정에 의해 대역폭의 비효율적인 할당과 지연시간의 증가 등의 문제가 발생할 수 있다.

WFQ는 우선순위 큐잉의 단점인 우선순위가 낮은 클래스의 패킷들이 우선순위가 높은 트래픽에 의해 아사되는 문제를 해결하는 동시에 커스텀 큐잉 방식에서 차등화된 서비스를 제공하지 못하는 현상을 해소하기 위해 개발된 것이다. 최대 4096개의 큐를 사용할 수 있는 WFQ는 ping푼히 많은 큐를 제공하기 때문에 클래스를 플로우 단위로 나눠 사용한다. 대역폭 요구양이 다른 트래픽을 flow별로 구분하여서 QoS를 보장하려고 한다. 하지만 대역폭의 요구량이 많은 트래픽이 가중치에 따라 재할당 받은 대역폭으로도 누적된 서비스 지연이 계속된다면 결국 패킷이 폐기되어 버린다는 단점이 발생한다. 따라서, 본 논문에서는 기존의 큐잉 방법들이 가진 문제점을 해결하고, 동기식 이더넷에 적절한 큐잉 방법인 Virtual Clock을 이용한다.

3. Virtual Clock을 이용한 동기식 이더넷에서의 큐잉 방법

동기식 이더넷에서 패킷 전송 시 계속되는 지연으로 인해 큐에서 정체상황이 발생하게 된다면 패킷은 폐기되어 QoS를 보장하지 못하는 현상이 발생된다. 이를 개선하기 위한 큐잉 방법으로 본 논문에서는 동기식 이더넷에서 Virtual Clock을 이용한 큐잉 방법을 제안한다. Virtual Clock을 이용하면 일정시간 동안 도착하는 평균 패킷의 양에 따라 해당 서비스를 제공할 수가 있다. 또한 버스트한 상황에서 다른 패킷의 영향을 받지 않고 아사를 막는 조절을 할 수 있다. Virtual Clock 알고리즘은 TDM(Time Division Multiplexing)개념에서 출발했다. 그러나 Virtual Clock은 TDM처럼 여러 패킷들이 한 전송로로 시분할 다중화로 패킷을 전송되지만 일정한 타임슬롯 단위로 다중화 되는 것이 아니라 들어오는 패킷들의 간격에 따라 유동적인 타임슬롯으로 다중화시킨다. 진입하는 각 패킷마다 각각의 Virtual Clock에서 tick을 부여하고 타임스탬프를 찍게 된다. 그리고는 초당 패킷율인 AR(Average arrival rate, packets/sec)에 의해 분류되고 큐잉 스케줄러에 의해 AR별로 스케줄링 된다. IR(packet Inter-arrival time, sec)은 평균 패킷 도착 간격이고 AI(Average observation interval, sec)는 평균 관측 구간을 나타낸다. AI값은 도착한 총 패킷을 AR로 나눈 값이 된다. Virtual Clock 알고리즘에는 AR과 AI값을 관찰하고 패킷을 조절하기 위해 Virtual Clock(VC)과 보조 Virtual Clock(auxVC)를 사용한다. 보조VC를 사용함으로써 Virtual Clock 알고리즘은 패킷 도착율에 따른 큐잉을 보장할 수 있다. 일정하지 않은 패킷 갑자기 들어와도 VC와 보조VC의 사용으로 효율적인 큐잉을 할 수 있다.

4. 실험 결과 및 성능 비교

virtual Clock 알고리즘의 큐잉방법을 구현하기 위한 실험을 수행한다. 동기식 이더넷에서 Virtual Clock 이용한 큐잉방법을 우수성을 측정하기 위해 시나리오를 구성하고 시뮬레이션을 통해 실험 결과를 보인다. 실험은 OPNET Modeler를 이용하여 실험하였다. Virtual Clock 알고리즘이 버스트한 패킷이 들어와도 패킷이 폐기되지 않고 큐잉 되었다가 다시 처리하지 못한 패킷을 다시 처리해 준다고 볼 수 있다. Virtual Clock 알고리즘과 WFQ 알고리즘으로 스케줄링한 패킷을 비교해본 결과 VC가 WFQ보다 안정적으로 패킷을 처리해주는 모습을 볼 수 있다. 이는 Virtual Clock 알고리즘이 PQ, CQ의 단점을 보완한 WFQ 알고리즘보다 더 우수하다는 결과를 보여준다.

5. 결론 및 향후 연구과제

본 논문에서는 동기식 이더넷에서 비실시간성 데이터에 대해 여러 가지 다른 큐잉 방법들보다 QoS를 향상시킬 수 있는 Virtual Clock을 이용한 큐잉 방법을 제안했다. Virtual Clock을 통한 큐잉 방법은 많은 트래픽이 우선순위가 높거나 같은 트래픽이 많은 상황에서도 큐를 유동적으로 관리함으로써 효율적인 큐잉을 할 수 있다. 동기식 이더넷은 가정 내의 실시간 데이터를 지연 없이 전송하는 기술이다. 이러한 실시간 데이터 전송을 위해 시간을 동기화 하는 방법과 실시간 데이터 전송을 위한 예약방법과 전송 방법에 대해서 더 많은 연구가 이루어져야 한다.