토큰링과 이더넷에서의 인터넷 통합 서비스 지원

김소연, 이미정 (이화여자대학교 컴퓨터학과)

Internet Integrated Services Over TokenRing/Ethernet

Soyeon Kim, Meejeong Lee
(Dept. of Computer Science & Engineering, Ewha)

인터넷에서 통합서비스를 제공하기 위해서는 종단간 다양한 QoS를 지원해야하는데 이를 위해서는 인터넷 뿐 아니라 종단 사용자와 인터넷을 연결하는 랜에서의 QoS 지원이 필요하다. 이에, IETF(Internet Engineering Task Force)는 특정 링크 계층으로 구성되는 서브넷에서 QoS 보장을 지원하기 위한 기본 골격을 제안하였다. 그런데, 매체 접근 방식에 따라 서브넷이 제공할 수 있는 서비스의 정도가 크게 달라지기 때문에 이러한 IS802모델을 특정 서브넷에 적용하기 위해서는 이를 고려한 구현이 요구된다. 본 논문에서는 IS802모델을 이더넷과 토큰링에 적용한 모델인 이더넷_IS, 토큰링_IS의 구현 방식을 제안하였다. 특별히, 이더넷의 경우 매체 접근 방식의 특성상 IS802 모델의 적용만으로는 서브넷 전체에서 우선 순위에 따른 전송을 지원할 수 없기 때문에 이를 보완하기 위하여 페이즈 관리자를 제안하고 이더넷_IS 모델에 페이즈 관리자 모듈을 첨가한 페이즈 관리자 모델을 추가로 제안하였다. 시뮬레이션을 통해 기존의 전통적인 이더넷과 토큰링 및 새롭게 표준화된 IS802모델을 적용한 이더넷_IS와 토큰링_IS, 그리고 페이즈 모델간의 성능을 비교하였다. 시뮬레이션 결과에 의하여, 페이즈 모델의 경우 이더넷의 매체 접근 방식을 상당히 보완함으로써 QoS 보장을 요구하는 플로우에 대해서는 망의 여건에 대해 비교적 안정적인 서비스를 제공할 수 있음을 확인 할 수 있었다.

1. 개 요

IETF의 IS802그룹은 랜 상에서 인터넷 통합 서비스[1]를 지원하기 위해 서브넷에 대역폭 관리자[2]를 두어 통합 서비스를 위해 요구하는 기능적인 사항들을 지원하도록 하였다. 요구자 모듈, 대역폭 할당자, 통신 프로토콜로 구성된 대역폭 관리자는 응용/상위계층 프로토콜이 IEEE 802 형태 서브넷에 QoS를 요청하고 보장받는 메카니즘을 제공한다. 응용/상위 계층 프로토콜과 대역폭 관리자 사이의 인터페이스로 역할하는 요구자 모듈은 예약의 초기화를 위해 요구되는 파라미터들을 상위 계층으로부터 전달받아 입장 제어를 책임진 대역폭 관리자의 다른 요소들이 이해할 수 있는 적당한 형식으로 변환하여 전달하고 대역폭 할당자는 대역폭 할당, 기존 예약의 변경, 자원 이용 가능성에 대한 질의 등의 다양한 서비스 요구를 처리함으로써 서브넷의 자원 할당에 대한 상태를 유지하고 입장 제어를 책임진다.

응용이 요구하는 QoS 정보는 사용자 우선순위 값으로 표기되며 이 값은 모든 프레임의 전송과 수신에 연관된다. 각각의 다른 프레임 형식을 갖는 IEEE 802 네트워크에 사용자 우선순위 값을 공통적으로 사용하기 위해 IS802모델은 IEEE802.1p의 확장 프레임 포맷[3]을 사용하였다. IEEE802.1p는 다양한 매체 접근 방식을 사용하는 랜들로 구성된 브릿지드 랜 환경에서 일관된 방식으로 사용자 우선순위를 전달할 수 있도록 프레임 포맷을 확장하였다. 확장된 프레임의 우선 순위 필드는 3비트로 이루어져 8개의 우선 순위를 지원할 수 있고, 각 프레임은 트래픽 클래스 별로 큐잉되어 프레임에 명시된 사용자 우선순위 값을 기준으로 매체에 접근한다.

이더넷, 토큰링, FDDI 등의 서브넷에서 제공하는 서비스는 링크 계층의 하위 프로토콜 스택에 위치하는 매체 접근 방식에 의하여 크게 달라지기 때문에, 각 서브넷에서 실제적으로 IS802 모델을 구현하기 위해서는 서브넷의 고유한 전송 매체 접근 방법이 고려되어야 한다.

따라서 본 논문은 IEEE 802 형태 랜 중 가장 일반적으로 널리 사용되고 있는 이더넷과 토큰링에서 인터넷 통합 서비스를 지원하기 위한 방안을 제시하고자 한다.

2. 모 델

2.1 토큰링

토큰링은 우선순위 토큰을 사용하여 매체 접근을 제어하며 헤더에 우선 순위를 표기할 수 있는 형태의 프레임을 갖기 때문에 인터넷 통합 서비스 제공에 적합한 매체이다. 노드는 전송하려는 패킷의 우선순위 값이 토큰의 예약값 보다 큰 경우에만 토큰의 예약값을 갱신할 수 있고 자유 토큰의 우선순위 값 이상의 우선순위 패킷을 갖은 노드만이 자유 토큰을 잡아 THT(Token Holding Time)동안 패킷을 전송할 수 있기 때문에 서브넷에서 우선순위가 가장 높은 패킷이 먼저 서비스된다. 토큰링 프레임은 IEEE 802.1p 프레임과 마찬가지로 8개의 우선순위를 표기하기 위해서 3비트를 할당한다. 따라서 인터넷 통합 서비스 모델에서 정의된 서비스 종류인 보장 서비스, 제어 서비스, 최선 서비스 순으로 우선순위 값을 높게 할당한다면 토큰링 고유의 프레임과 매체 접근 방식을 사용해서도 서비스 클래스 별로 우선적인 서비스를 보장할 수 있다

토큰링_IS 모델에서 QoS 지원을 요구하는 응용은 실제 트래픽을 발생시키기 전에 요구자 모듈을 통해 실제 트래픽 생성의 주체인 세션 정보를 대역폭 관리자 노드에게 전달하고 대역폭 관리자는 네트워크의 안정된 상태를 유지하면서 새로운 요청이 만족 될 수 있는지 여부를 판단한다. 토큰링_IS 모델의 대역폭 관리자는 사용가능한 대역폭이 세션이 요구하는 평균 대역폭보다 큰 경우에 입장을 허가하는데 사용가능한 대역폭은 네트워크 구성(Network Configuratoin) 시간에 [서브넷 대역폭 값 * 목표 활용율(Target Utilization)]로 초기화한다. 이때 목표 활용율은 서브넷 대역폭 중 QoS 보장을 요구하는 세션에 최대로 할당될 수 있는 비율이다. 사용가능한 대역폭 값은 입장을 허가할 때 요구된 대역폭 값만큼 감소시키고 세션이 종료되면 세션에 할당했던 대역폭만큼 증가시킨다.

2.2. 이더넷

버스형 토폴로지로 구성된 이더넷의 노드는 전송매체를 관찰하여 전송매체가 사용중인지 여부를 판단한다. 패킷 전송을 원하는 노드는 전송매체가 휴지상태이면 전송을 시도하는데 두 개 이상의 노드가 동시에 패킷을 전송한 경우에는 충돌이 발생하여 패킷은 재전송되어야 한다. 이더넷의 노드는 패킷의 우선순위를 식별하지 못하고 하나의 큐에서 FIFO(First-In-Firs-Out)방식으로 서비스한다.

이더넷에 IS802 모델에서 정의한 기능을 구현하기 위해서 토큰링_IS 모델과 같이 대역폭 관리자 노드를 첨가하였고 이더넷 프레임을 IEEE 802.1p 프레임 형식으로 캡슐화를 하였다. 또, 프레임에 표기된 8개의 우선순위를 식별하기 위해 이더넷_IS 모델에 식별자 기능을 부가하였고, 이더넷의 FIFO 스케줄러를 우선순위가 높은 큐부터 FIFO 방식으로 서비스하는 우선순위 큐 FIFO 스케줄러로 수정하였다. 마지막으로 세션의 입장 제어와 관련된 관리 패킷에 대해서는 네트워크에서 충돌이 일어난 경우 성공적으로 전송될 때까지 재전송하여 네트워크에서 손실되지 않도록 하였다.

그러나 이더넷_IS 모델은 토큰과 같은 특별한 전송매체 점유 조절자 없이 전송 매체가 휴지 상태에 패킷 전송을 시도하는 이더넷의 매체 접근 방법을 사용함으로써 충돌을 예상할 수 없고 충돌의 가능성은 패킷을 보유한 노드가 많을수록 커지는 등 모델은 전송을 시도하는 노드 수 및 트래픽 발생 형태와 밀접한 관계를 갖는다. 다시말해 이더넷_IS 모델은 노드내에서만 우선순위 서비스를 제공할 뿐 노드간 우선순위 서비스를 제공하지 못한다. 그러나 서브넷 내에서 인터넷 통합 서비스를 지원하기 위해서는 각 노드는 더 높은 우선순위 플로우 패킷을 전송하고자 하는 노드가 있을 때 그 노드에게 전송할 기회를 양보하거나 최소한 기회를 더 많이 줄 수 있는 협조가 이루어져야 한다. 이에 본 연구는 이러한 협조를 이더넷 내의 노드들 간에 유도하기 위하여 다음과 같은 페이즈 개념을 정의하고 이더넷_IS 모델의 각 노드에 그림1과 같이 페이즈 관리자 모듈을 부가하였다.

- 상위 페이즈 : 보장 서비스와 제어 서비스 패킷과 관리 패킷만을 전송할 수 있는 상태
- 기본 페이즈 : 모든 종류 패킷을 전송할 수 있는 상태

각 노드의 페이즈 관리자는 페이즈 변이 정보를 생성하여 이를 알리고 페이즈 변이 정보를 수신하여 스케줄러에게 이를 알림으로써 현재 페이즈에 적합한 패킷만을 전송하도록 한다. 각 노드의 페이즈 변이 요구를 쉽게 관찰할 수 있고 또한 간단하게 수정될 수 있도록 하기 위해 다음과 같이 페이즈 전이 알고리즘을 설계하였다.

- 네트워크 구성 시간에 페이즈 값은 기본 페이즈로 주어진다.
- 기본 페이즈에서 노드내 보장 서비스와 제어 서비스 큐길이 합이 α이상인 경우 상위 페이즈로 전이하고 다른 페이즈 관리자에게 이를 브로드캐스트 한다.
- 상위 페이즈로 전이를 시작한 노드의 보장 서비스와 제어 서비스 큐길이 합이 β이상인 경우 기본 페이즈로 전이하고 이를 브로드캐스트한다.
- 페이즈 전이 패킷이 도착하면 수신한 값으로 페이즈를 전이한다.

4. 시뮬레이션 및 결과

제안하는 모델의 인터넷 통합 서비스 지원도를 알아보기 위해 일련의 시뮬레이션을 수행하였다. 패킷 손실율, 평균 패킷 전달 시간, 최대 패킷 전달 시간등을 측정하였다. 그러나 패킷 손실이 발생하지 않는 토큰링의 경우에는 손실율을 제외하였다. 시뮬레이션은 OPNetv4.0을 이용하여 SunOS 5.6에서 수행하였다.

4.1 시뮬레이션

토큰링 모델은 5개 노드를 100m간격으로 대역폭 10Mbps, 전파 지연 속도 5.1282E-09m/s의 링크로 연결하였고 THT는 0.01로 설정하였다. 토큰링_IS 모델은 토큰링 모델에 대역폭 관리자 노드를 추가하였다. 이더넷 모델은 5개 노드를 5.1282E-09m/s 전파 지연 속도

에 10Mbps대역폭을 가지는 500m 길이 링크로 연결하고, 이더넷_IS 모델과 페이즈 관리자 모델은 이더넷 모델에 대역폭 관리자 노드를 추가하였다.

이더넷의 성능은 동시에 전송 시도되는 패킷 수에 따라 달라지므로 결국 링크에 부착된 노드의 개수와 각 노드에서의 트래픽 발생 형태에 따라 성능이 달라지게 된다. 이러한 다양한 상황에서 각 모델의 성능을 실험하기 위해 본 논문은 토폴로지를 노드 수 5개, 10개로 변화시키고, 트래픽 발생 형태를 보장 서비스와 제어 서비스 패킷을 한 노드만이 발생시키는 집중형과 모든 노드들이 균일하게 발생시키는 분산형으로 변화시키면서 실험하였다. 모든 실험에서 보장 서비스와 제어 서비스 트래픽 양의 합은 50%로 고정하였고, 각 노드에서 발생하는 최선 서비스 트래픽 양은 균일하다고 가정하였다.

4.2 결과

네트워크 내 QoS 지원은 트래픽 양의 제어와 노드 내 우선순위 서비스 제공에 의해 토큰링_IS 모델과 이더넷_IS 모델이 토큰링 모델과 이더넷 모델에서 측정된 평균 및 최대 패킷 전달 시간, 손실율 면에서 성능을 향상시켰다. 이더넷_IS 모델은 각 유형의 실험마다 성능의 차이가 측정되어 토폴로지와 트래픽 발생 형태에 민감하게 반응함을 보여주었지만 페이즈 개념을 부가한 페이즈 관리자 모델은 이더넷_IS 모델을 적절하게 보완하였다.

그림2와 그림3은 각각 토큰링과 토큰링_IS의 경우 서비스 종류 별 평균 패킷 전달 시간을 보여준다. 두 그림에서 공통적으로 보장 서비스, 제어 서비스, 최선 서비스 순으로 평균 전달 시간이 작게 측정되었고 보장 서비스와 제어 서비스 패킷의 전달 시간은 최선 서비스 로드에 무관하게 작은 값이 측정되었다. 이는 토큰링에서는 노드내에서 뿐 아니라 노드간의 우선순위를 지원했기 때문이다. 그러나 최선 서비스 패킷의 경우 그림2의 토큰링 모델은 평균 전달 시간이 최고 32초로 큰 값이 측정되었고 그 증가율 역시 최선 서비스 로드 증가율에 비례적으로 증가하였지만 입장제어를 거쳐 QoS 지원을 요구하는 트래픽 양을 30%로 억제시킨 토큰링_IS 모델은 최고 2초 정도의 평균 전달 시간을 보이는 등 최선 서비스 로드 증가에 크게 영향을 받지 않았다

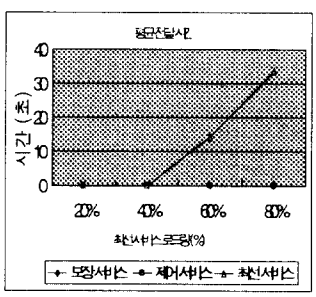

그림 2. 토큰링 모델

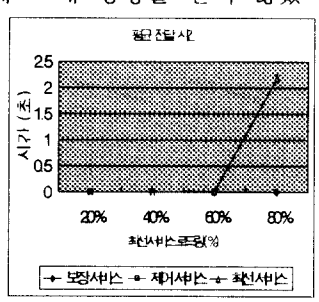

그림 3 . 토큰링_IS 모델

그림4-그림6은 각 모델의 패킷 손실율에 대한 결과이다. 이더넷의 각 노드와 네트워크는 서비스 종류를 구분하지 않고 서비스하므로 그림4와 같이 각 서비스 종류의 손실율 형태는 비슷하고, 입장제어과정이 없어 이더넷_IS나 페이즈 관리자 모델보다 발생하는 트래픽 로드가 더 크기 때문에 손실율은 급격하게 증가하였다. 반면에 그림 5와 그림6의 이더넷_IS 모델과 페이즈 관리자 모델은 보장 서비스와 제어 서비스의 패킷 손실이 최선 서비스에 비해 낮게 관찰되었는데, 입장 제어 절차를 통해 보장 서비스와 최선 서비스 량이 제어되었고 각 노드에서 보장 서비스 패킷과 제어 서비스 패킷을 우선적으로 서비스하였기 때문이다. 특히 그림 6의 페이즈 관리자는 네트워크에서 최선 서비스 패킷이 서비스될 수 없는 기간인 상위 페이즈를 두고 보장 서비스 패킷과 제어 서비스 패킷의 경쟁과 충돌 횟수를 줄임으로써, 최고 1%미만의 손실율을 보였다

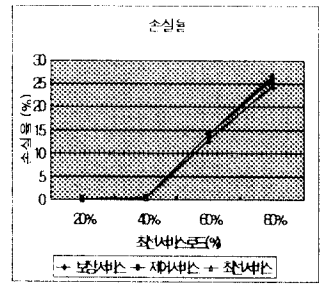

그림 4. 이더넷 모델

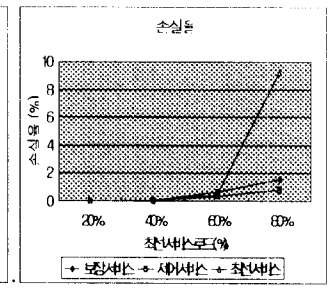

그림 5. 이더넷_IS 모델

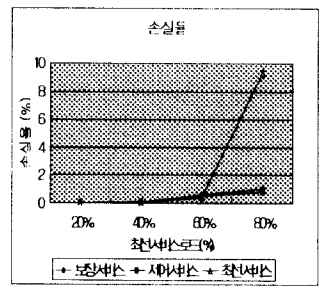

그림 6. 페이즈 관리자 모델

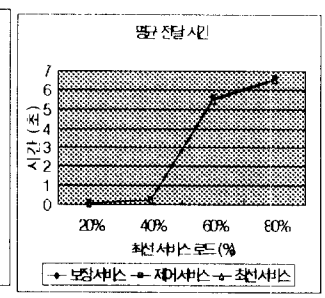

그림 7. 이더넷 모델

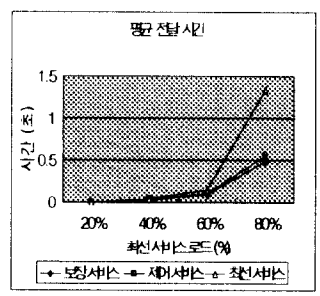

그림 8. 이더넷_IS 모델

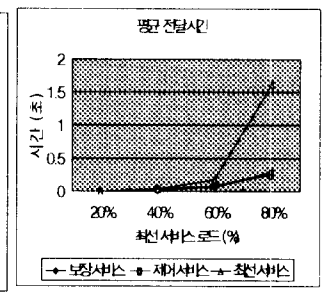

그림 9. 페이즈 관리자 모델

그림7 - 그림9는 평균 패킷 전달 시간을 보여준다. 이더넷 모델은 최선 서비스 로드가 커짐에 따라 패킷 전달 시간이 큰 폭으로 증가하였지만 이더넷_IS 모델과 페이즈 관리자 모델은 완만하게 증가하였다. 노드간에서 뿐 아니라 노드 내에서도 서비스 종류를 구분하지 않는 이더넷 모델은 모든 서비스 종류의 패킷 전달 시간이 같은 형태로 측정되었고, 노드내 우선순위 서비스를 제공하는 이더넷_IS 모델과 페이즈 관리자 모델은 보장 서비스, 제어 서비스, 최선 서비스 순으로 평균 패킷 전달 시간이 작게 측정되었다. 특히 보장 서비스와 제어 서비스 량이 많은 경우 상위 페이즈로 전이하여

네트워크에서 보장 서비스와 제어 서비스 트래픽 만을 독점적으로 서비스하는 페이즈 관리자 모델은 보장 서비스 패킷의 평균 패킷 전달 시간이 0.25초 이내 값으로 측정되었다.

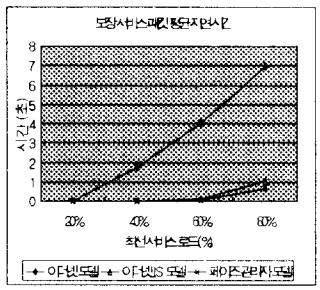

그림 10 . 노드 5개

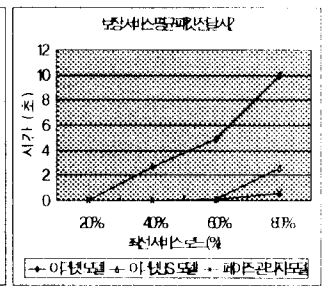

그림 11 . 노드 10개

그림 10과 그림11의 노드 수 5개 10개의 토폴로지에서 각 모델의 보장 서비스 패킷의 평균 전달 시간을 통해 이더넷_IS 모델을 페이즈 관리자 모델이 보완하였음을 알 수 있다. 이더넷_IS 모델은 5개 노드에서 약 1초, 10개 노드에서 약 2.2초로 측정되어 노드수에 따라 큰 변화를 보였지만 페이즈 관리자 모델은 공통적으로 약 0.5초 값을 보였다. 이더넷_IS모델에서는 각 노드가 보유한 패킷 중 가장 높은 우선 순위의 패킷을 전송하므로 노드 수가 커짐에 따라 경쟁하는 패킷의 수는 많아졌지만 페이즈 관리자 모델은 상위 페이즈에서는 최선 서비스 패킷만을 가진 노드에서의 전송을 방지했기 때문에 노드 수에 민감한 반응을 보이지 않았다.

5장. 결론

본 논문에서는 IS802모델을 이더넷과 토큰링에 적용한 모델인 이더넷_IS, 토큰링_IS의 구현 방식을 제안하였다. 특별히, 이더넷의 경우 매체 접근 방식의 특성상 IS802 모델의 적용만으로는 서브넷 전체에서 우선 순위에 따른 전송을 지원할 수 없기 때문에 이를 보완하기 위하여 페이즈 관리자를 제안하고 이더넷_IS 모델에 페이즈 관리자 모듈을 첨가한 페이즈 관리자 모델을 추가로 제안하였다. 시뮬레이션을 통해 기존의 전통적인 이더넷과 토큰링 및 새롭게 표준화된 IS802모델을 적용한 이더넷_IS와 토큰링_IS, 그리고 페이즈 모델간의 성능을 비교하였다. 시뮬레이션 결과에 의하여, 페이즈 모델의 경우 이더넷의 매체 접근 방식을 상당히 보완함으로써 QoS 보장을 요구하는 플로우에 대해서는 망의 여건에 대해 비교적 안정적인 서비스를 제공할 수 있음을 확인 할 수 있었다.

6. 참고문헌

[1] R.Braden, D. Clark, S.Shenker, "Integrated Services in the Internet Architecture : an Overview ", RFC 1633.

[2] Anoop Ghanwani, J.Wayne Pace , "A Framework for Providing Integrated Services over Shared and Switched IEEE 802 LAN Technologies", draft-ietf-issll-is802-framework-04.txt, March 98

[3] "P802.1p/D4 Standard for Local and Metropolitan Area Networks - Supplement to Media Access Control Brides : Traffic Class Expediting and Dynamic Multicast Filtering", IEEE standard, Sep. 1996