(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2007년12월14일
(11) 등록번호 10-0784852
(24) 등록일자 2007년12월05일

(51) Int. Cl.
H04L 12/56 (2006.01) *H04B 7/005* (2006.01)
H04L 12/28 (2006.01)
(21) 출원번호 10-2006-0063521
(22) 출원일자 2006년07월06일
심사청구일자 2006년07월06일
(56) 선행기술조사문헌
KR1020050089756 A
(뒷면에 계속)

(73) 특허권자
인하대학교 산학협력단
인천 남구 용현동 253 인하대학교
(72) 발명자
유상조
인천 남구 용현4동 인하대학교
노권문
서울 금천구 가산동 두산아파트 116동 1302호
(74) 대리인
이은철

전체 청구항 수 : 총 10 항 심사관 : 이희봉

(54) 무선 에드혹 네트워크에서 시간 기반 공정성 패킷 스케줄링방법

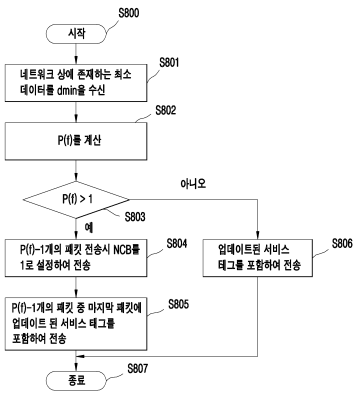
(57) 요 약

본 발명은 무선 에드혹 망에서의 분산 패킷 스케줄링 방법에 관한 것으로, 물리 계층의 다중 데이터률을 고려하여 분산된 방식으로 네트워크상에 각 플로우의 전송 순서를 결정하여 전체 네트워크의 처리량을 향상시키는 시간 기반 공정성을 제공하는 패킷 스케줄링 방법에 관한 것이다.

본 발명은 에드혹 네트워크의 분산 패킷 스케줄링 방법에 있어서,

각 플로우의 데이터률과 가장 낮은 데이터률을 갖는 플로우의 데이터률을 비교하는 단계; 상기 비교 결과를 이용하여 전송해야 하는 패킷 수를 계산하는 단계; 및 상기 패킷 수의 계산 결과에 따라 전송할 프레임에 플래그를 설정하여 전송하는 단계를 포함하는 것을 특징으로 한다.

대표도 - 도8



(56) 선행기술조사문헌
KR1020050071362 A
KR1020050035295 A
JP2006180139 A
JP2006074279 A

특허청구의 범위

청구항 1

에드혹 네트워크의 분산 패킷 스케줄링 방법에 있어서,

각 플로우의 데이터률과 가장 낮은 데이터률을 갖는 플로우의 데이터률을 비교하는 단계;

상기 비교 결과를 이용하여 전송해야 하는 패킷 수를 계산하는 단계; 및

상기 패킷 수의 계산 결과에 따라 전송할 프레임에 플래그를 설정하여 전송하는 단계;를 포함하는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 2

제 1항에 있어서, 상기 전송할 프레임에 플래그를 설정하여 전송하는 단계 후 모든 노드의 인접한 노드들의 플로우에 대한 데이터률 정보를 얻고 자신의 로컬 테이블에 저장하고, 자신의 로컬 테이블에 존재하는 플로우의 데이터률과 비교하여 가장 낮은 데이터률을 저장하여 데이터 전송 시 제어 프레임에 포함시켜 전송하는 단계를 더 포함하는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 3

제 2항에 있어서, 상기 저장된 가장 낮은 데이터률에 대한 정보를 패킷 전송시에 제어 프레임에 포함하여 전송하고, 인접 노드로부터 상기 저장된 데이터률보다 더 낮은 데이터률에 대한 정보를 얻는다면 그 데이터률로 상기 저장된 데이터률을 업데이트하는 단계를 더 포함하는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 4

제 1항에 있어서, 상기 패킷 수를 계산하는 단계는 플로우의 데이터률과 가장 낮은 데이터률을 갖는 플로우의 데이터률의 비에 의해 계산되는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 5

제 4항에 있어서, 상기 패킷 수가 한 개를 초과하는 경우에는 (패킷 수-1) 개의 패킷을 전송하고 전송시 제어 프레임에 플래그를 설정하여 전송하는 단계 및

전송 패킷 중 마지막 패킷에 업데이트된 서비스 태그를 포함하여 전송하는 단계를 포함하는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 6

제 4항에 있어서, 상기 패킷 수가 한 개를 초과하는 경우에는 패킷 전송 패킷 중 마지막 패킷 전송 시 DS와 ACK 프레임에 업데이트 태그를 붙여서 전송하는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 7

제 1항에 있어서, 상기 플래그가 1로 설정된 경우에는 수신 노드에서 RTS의 폐기 없이 CTS를 보내는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 8

에드혹 네트워크의 분산 패킷 스케줄링 방법에 있어서,

각 플로우의 데이터률과 가장 낮은 데이터률을 갖는 플로우의 데이터률을 비교하는 단계; 및

상기 비교 결과 데이터률이 가장 낮은 플로우가 하나의 패킷을 전송할 때, 사용하는 채널 점유 시간을 기준으로 각 플로우가 동일한 채널 점유 시간을 갖도록 하는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 9

제 8항에 있어서, 상기 각 플로우의 채널 점유시간(T(f))은 경쟁 관계에 있는 모든 플로우들의 집합에서 모든

플로우가 하나의 패킷을 서비스하는데 사용하는 채널 점유시간에서 플로우가 하나의 패킷을 서비스하는데 사용하는 채널 점유 시간이 차지하는 비율로 정의하는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

청구항 10

제 8항에 있어서, 각 플로우가 보장 받아야 하는 채널 점유 시간(GT(f))은

$$GT(f)=\frac{\frac{s_f}{d_{\min}}}{\sum_{i\in F}\frac{s_i}{d_{\min}}}$$

이고, 상기 수학식에서 F는 플로우 경쟁 그래프에 존재하는 모든 플로우들의 집합, s_f는 플로우 f의 패킷 사이즈, s_i는 플로우 i의 패킷 사이즈 및 $d_{\min}$은 플로우 경쟁 그래프에서 연결된 플로우들 중 가장 낮은 데이터률을 갖는 플로우의 데이터률을 나타내는 것을 특징으로 하는 시간 기반 공정성 패킷 스케줄링 방법.

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<9> 본 발명은 무선 에드혹 망에서의 분산 패킷 스케줄링 방법에 관한 것으로, 물리 계층의 다중 데이터률을 고려하여 분산된 방식으로 네트워크상에 각 플로우의 전송 순서를 결정하여 전체 네트워크의 처리량을 향상시키는 시간 기반 공정성을 제공하는 패킷 스케줄링 방법에 관한 것이다.

<10> 에드혹 네트워크는 고정된 기반 망의 도움 없이 이동 단말기들로만 손쉽게 자율적으로 망을 구성하여 긴급 구조나 전쟁터 등에서 무선 데이터 서비스를 제공할 수 있는 자율성과 융통성을 부여한 무선 네트워크이다. 에드혹 네트워크를 구성하는 노드들은 무선 인터페이스를 가지며, 이동 컴퓨팅 기능을 가진 호스트와 라우팅 기능을 가진 라우터를 동시에 만족하는 형상으로 흔히 이동 노드로 불려진다. 이러한 이동 노드는 저전력 구현 등의 이유로 도달 거리가 제한되므로 중간 노드로서 데이터 전달 기능을 가지며, 배터리를 사용하므로 에너지의 공급이 일정치 않은 특성을 갖는다.

<11> 상기의 에드혹 네트워크를 구성하는데 대중적으로 사용하는 기존 IEEE 802.11은 하나의 데이터률인 2Mbps를 제공하지만, 상업적인 성공을 거둔 IEEE 802.11b는 1, 2, 5.5, 11Mpbs의 데이터률을 제공하고 IEEE 802.11g는 6, 9, 12, 18, 24, 36, 54Mbps의 데이터률을 각 표준의 물리 계층에서 제공하고 있다. 이러한 다중 데이터률을 에드혹 네트워크에서 이용하고자 하는 노력은 라우팅 관련 연구에서부터 시작되었고, 최근에 다중 데이터률을 제공하는 에드혹 네트워크에서 성능 분석 연구가 이루어졌다. 그러나 에드혹 네트워크에서의 패킷 스케줄링 연구에 있어서 다중 데이터률을 고려하는 연구는 거의 이루어지지 않았다. 그 이유는, 처리량 기반 공정성(throughput-based fairness)을 제공하기 위해서 SFQ(Start-Time Fair Queueing), WFQ(Weighted Fair Queueing) 등과 같이 유선에서의 공정 큐잉 방식을 에드혹 패킷 스케줄링에 적용했기 때문이다. 이러한 패킷 스케줄링 기법의 기본 가정은 채널 용량이 고정되어 있는 것이다. 따라서, 다중 데이터률이 제공되는 환경에서 채널 용량을 고정시키기가 쉽지 않기 때문에, 에드혹 패킷 스케줄링 연구에서 다중 데이터률을 제공하는 환경을 고려하지 않았다.

<12> 기존의 에드혹 패킷 스케줄링 방법은 처리량에 기반한 공정성을 제공한다. 처리량 기반 공정성을 제공하는 스케줄링 방식은 TDMA 기반의 에드혹 네트워크를 위해 제안된 신용기반(credit-based) 공정 스케줄링 기법과 타임 스탬프에 기반한 공정 스케줄링 기법 두 가지로 나눌 수 있다. 타임 스탬프에 기반한 패킷 스케줄링 기법은 처리량 기반 고정성을 제공하기 위해 도착된 패킷들은 SFQ(Start-Time Fair Queueing)에 기반하여 두 개의 서비스 태그(start tag와 finish tag)를 갖고, 가장 작은 시작 태그(start tag)를 갖는 패킷을 서비스한다.

<13> 이하, 도 1 및 도 2를 참조하여 본 발명에 따른 에드혹 네트워크의 패킷 스케줄링을 설계할 때 고려해야 하는 쟁점을 설명한다.

<14> 도 1은 본 발명이 적용되는 일반적인 에드혹 네트워크의 노드 그래프를 나타내고, 도 2는 도 1의 플로우 경쟁관계를 나타내는 플로우 경쟁 그래프를 나타낸다.

<15> 에드혹 네트워크 환경에서는 패킷 스케줄링을 설계할 때, 다음과 같은 설계상의 쟁점을 고려해야 한다.

<16> 첫째, 무선 데이터 전송은 지역적으로 브로드캐스팅이기 때문에, 공유하는 채널에 대한 경쟁은 위치 종속적이다. 예를 들어 도 1의 노드 그래프와 같이 5개의 노드(Ni, i = 0,…,4)가 있고, 4개의 플로우(Fj, j = 0,…, 3)가 있을 때, 플로우간 경쟁관계를 도 2와 같이 플로우 경쟁 그래프로 변환할 수 있다. 도 2에서 각 꼭지점은 플로우를 나타내고, 각 간선은 플로우들 사이의 경쟁 관계를 나타낸다. F0(200)는 F1(201)과 F2(202)와 경쟁 관계에 있지만, F3(203)와는 직접적으로 경쟁관계에 있지 않다. 따라서 F0(200)와 F3(203)는 동시에 서비스 받을 수 있다. 이와 비슷하게 F1(201)은 F0(200), F2(202), F3(203)와 경쟁관계에 있다. 이처럼 플로우들 사이에서 경쟁관계에 있는 플로우들의 집합은 플로우의 위치에 따라서 다르다.

<17> 또한 F0(200)와 F3(203)와 같이 직접 경쟁관계에 있지 않은 플로우들은 동시에 서비스 받을 수 있지만, 이는 경쟁관계에 있는 플로우들 사이에서 엄격한 공정성을 제공하기 위해서는 한 순간에 하나의 플로우만 서비스를 받아야 하기 때문에 두 개의 플로우 F0(200), F3(203)가 직접적으로 경쟁 관계에 있지 않아서 동시에 서비스 된다면, 공정성을 훼손하는 문제가 발생한다. 하지만, 동시에 서비스 될 수 있는 플로우들 중 하나만 서비스 한다면, 네트워크의 처리량을 감소시키는 단점이 있다. 이러한 문제를 공간적 채널 재사용 문제라 한다.

<18> 2) 에드혹 스케줄링의 분산된 속성 - 유선 네트워크나 셀룰러 네트워크에서 각 라우터나 스위치, 기지국은 패킷 스케줄링을 결정할 때, 모든 플로우에 대한 정확한 정보를 가지고 있다. 하지만, 에드혹 네트워크에서는 경쟁관계에 있는 플로우가 각 다른 송신 노드에서 발생할 수 있고, 각 송신 노드에서는 다른 송신 노드의 플로우 정보를 직접적으로 접근할 수 없다. 이는 에드혹 네트워크에서 패킷 스케줄링을 설계할 때, 분산되어 있는 플로우 정보를 이용하여 계산해야 하는 문제를 포함하고 있음을 나타낸다.

<19> 상기의 설계상의 쟁점들을 고려하고 SFQ와 같은 유선의 패킷 스케줄링 방법을 에드혹 네트워크에 적용한 대표적인 스케줄링 방법이 EMLM-FQ(Enhanced- Maximize-Local-Minimum Fair Queueing)이다. EMLM-FQ는 네트워크의 각 플로우에게 <수학식 1>과 같은 서비스 양을 제공한다.

수학식 1

<20>
$$W_f(t_1,t_2) \geq C\frac{r_f}{k\sum_{g\in S} r_g}(t_2-t_1)-\alpha$$

<21> 상기 <수학식 1>에서 $W_f(t_1,t_2)$는 플로우 f가 시간 t_1부터 t_2까지 받은 서비스의 양이고, C 는 채널 용량이고, r_f와 r_g는 각각 플로우 f와 g의 가중치이며, S는 플로우 경쟁 그래프에서 플로우 f와 직접 연결되어 있는 플로우의 집합이다. k와 α는 위치 종속적 상수이다.

<22> 그러나 실제 에드혹 네트워크를 구성하는데 이용하는 IEEE 802.11x의 물리 계층과 같이 채널용량 C는 각 플로우별로 고정되어 있지 않다. 따라서, 물리계층에서 다중 데이터률을 이용하는 네트워크에서 적용하는데 적합하지 않는 문제점을 가진다.

발명이 이루고자 하는 기술적 과제

<23> 따라서, 본 발명의 목적은 상기한 문제점을 해결하기 위하여 에드혹 네트워크에서 다중 데이터률을 고려한 분산된 패킷 스케줄링 방법을 제공함에 있다.

<24> 또한, 본 발명의 목적은 에드혹 네트워크를 구성하는 물리 계층의 데이터률을 고려하여 새로운 공정성 모델인

시간 기반 공정성을 제공하는 패킷 스케줄링 방법을 제공함에 있다.

<25> 또한, 본 발명의 목적은 에드혹 네트워크에서 패킷을 스케줄링하는 방법에 있어서, 분산된 방식으로 각 플로우의 서비스 태그 정보를 교환하고 스케줄링 순서를 결정하기 위한 패킷 스케줄링 방법을 제공함에 있다.

발명의 구성 및 작용

<26> 상기한 목적을 달성하기 위한 본 발명은 에드혹 네트워크의 분산 패킷 스케줄링 방법에 있어서,

<27> 각 플로우의 데이터률과 가장 낮은 데이터률을 갖는 플로우의 데이터률을 비교하는 단계; 상기 비교 결과를 이용하여 전송해야 하는 패킷 수를 계산하는 단계; 및 상기 패킷 수의 계산 결과에 따라 전송할 프레임에 플래그를 설정하여 전송하는 단계를 포함하는 것을 특징으로 한다.

<28> 이하 본 발명의 바람직한 실시예에 따른 상세한 설명을 첨부된 도면들을 참조하여 설명한다. 하기에는 본 발명을 설명함에 있어 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다.

<29> 1. 네트워크 모델

<30> 본 발명에서 고려하는 네트워크 환경은 물리계층과 MAC 계층에서 다중 데이터률을 제공하는 환경과 모든 노드들이 동일한 채널을 공유하는 다중 홉의 무선 에드혹 네트워크를 고려한다. 또한, 전송 노드의 전송 범위 안에 있는 노드만이 패킷을 받을 수 있다. 단일 플로우는 단일 전송 노드로부터 단일 수신 노드까지 전송되는 일련의 패킷 스트림으로 정의한다.

<31> 본 발명에서 적용하는 MAC 계층으로 일반적인 CSMA/CA 패러다임과 다중 데이터률 적응 기법으로 RBAR(Receiver Based AutoRate) 프로토콜을 따른다. RBAR 프로토콜은 수신 노드가 RTS(Request-To-Send)와 CTS(Clear-To-Send) 프레임 교환 과정 동안에 데이터 프레임을 전송하기 위한 적절한 데이터률을 선택해서 송신 노드에게 공지하는 방식이다.

<32> 송신 노드가 하나의 패킷을 전송하기 위해서 짧은 제어 프레임인 RTS(Request-to-Send)를 수신 노드에게 전송하면, RTS를 받은 수신 노드는 자신의 주변 노드 중에서 패킷을 전송하고 있는 노드가 있는지 확인한다. 전송 중인 노드가 없다면 SNR(Signal-to-Noise Ratio) 값을 측정하여 최적의 데이터률을 결정하고 이에 대한 정보를 CTS(Clear-to-Send)에 포함하여 응답한다. CTS를 받은 송신 노드는 인접한 노드들에게 RTS-CTS 핸드쉐이크의 성공을 알리기 위한 목적으로 수신 노드에게 CTS의 응답으로 DS(Data-Sending) 프레임을 보내고, DATA 프레임을 전송한다. 마지막으로, 수신 노드는 성공적인 DATA 프레임 수신에 대한 응답으로 송신 노드에게 ACK 프레임을 전송한다. 송신 노드와 수신 노드의 한 홉 거리에 있는 모든 노드들은 위의 제어 메시지들을 모두 들을 수 있다.

<33> 이하 첨부된 도면을 참조하여 발명에 따른 시간 기반 공정성을 설명한다.

<34> 도 3은 본 발명이 적용되는 다중 데이터률을 사용하는 에드혹 네트워크의 노드 그래프를 나타내고, 도 4는 도 3의 플로우 경쟁관계를 나타낸다.

<35> 도 5는 본 발명의 실시예에 따른 처리량 기반 공정성을 제공하는 패킷 스케줄링과 시간 기반 공정성을 제공하는 패킷 스케줄링의 각 플로우 서비스 순서와 채널 점유시간의 비교를 나타내고, 도 6은 본 발명의 실시예에 따른 다중 데이터률을 제공하는 네트워크에서 시간 기반 공정성을 제공하는 패킷 스케줄링과 모든 플로우의 데이터률이 가장 낮은 데이터률일때의 EMLM-FQ의 각 플로우 서비스 순서와 채널 점유시간의 비교를 나타낸다.

<36> 2. 시간 기반 공정성

<37> 기존에 제안된 에드혹 패킷 스케줄링은 플로우들 간에 처리량 기반 공정성(throughput-based fairness)을 제공하기 위하여 SFQ(Start-Time Fair Queueing)과 같은 유선에서의 패킷 스케줄링을 기반으로 한다. 이러한 처리량 기반 공정성은 플로우별 가중치에 따라 각 플로우가 받는 전체 서비스의 양이 동일함을 의미한다. 반면, 시간 기반 공정성(time-based fairness)은 각 플로우가 사용하는 전체 채널 점유 시간이 동등함을 의미한다.

<38> 도 3의 노드 그래프에 대해 도 4와 같이 네트워크의 모든 플로우가 직접적인 경쟁 관계에 있을 때, 도 5는 처리량 기반 공정성을 제공하는 EMLM-FQ와 시간 기반 공정성을 제공하는 패킷 스케줄링 방법에 대한 각 플로우의 서비스 순서와 채널 점유 시간을 나타낸다. 각 플로우가 갖는 패킷 사이즈와 가중치가 동일하다고 가정하고, 각

플로우의 초기 서비스 태그는 $F0<F1<F2$ 이다.

<39> 도 4와 같이 모든 플로우가 직접적인 경쟁관계에 있으면, EMLM-FQ는 F0(400),F1(401), F2(402) 순서대로 각 플로우의 패킷을 서비스한다. 시간 기반 공정성을 제공하는 경우 각 플로우가 점유하는 채널 시간이 공정해야 하므로, 데이터률이 가장 낮은 F2(402)가 하나의 패킷을 전송할 때 사용하는 채널 점유 시간을 기준으로 각 플로우가 동일한 채널 점유 시간을 갖도록 한다. 그림 7과 같이 처리량 기반 공정성을 제공할 때, 6개의 패킷이 서비스되는 시간 동안에 시간 기반 공정성을 제공할 때는 10개의 패킷을 서비스 할 수 있다. 즉, 네트워크 처리량이 약 40% 향상된다.

<40> 각 플로우 f의 채널 점유시간 $T(f)$ 은 <수학식 2>와 같이 정의한다.

수학식 2

$$T(f)=\frac{\frac{s_f}{d_f}}{\sum_{i\in F}\frac{s_i}{d_i}}$$

<41>

<42> F는 플로우 경쟁 그래프에 존재하는 모든 플로우들의 집합이고, s_f 는 플로우 f의 패킷 사이즈, d_f 는 플로우 f의 데이터률이다. $T(f)$ 는 F의 모든 플로우가 하나의 패킷을 서비스 하는데 사용하는 채널 점유 시간에서 플로우 f가 하나의 패킷을 서비스 하는데 사용하는 채널 점유 시간이 차지하는 비율을 나타낸다.

<43> 채널이 항상 이용 중이라고 가정하면, 플로우의 전체 채널 점유 시간은<수학시 3> 같이 정의한다.

수학식 3

$$\sum_{f\in F}T(f)=1$$

$$(0\le T(f)\le 1)$$

<44>

<45> 경쟁하는 플로우들의 가중치가 동일하다고 가정하면, 시간 기반 공정성에 입각해서 각 플로우가 보장 받아야 하는 채널 점유 시간($GT(f)$, Guaranteed Time)은 <수학식 4>와 같이 정의한다.

수학식 4

$$GT(f)=\frac{\frac{s_f}{d_{\min}}}{\sum_{i\in F}\frac{s_i}{d_{\min}}}$$

<46>

<47> $d_{\min}$은 플로우 경쟁 그래프에서 연결된 플로우들 중 가장 낮은 데이터률을 갖는 플로우의 데이터률이다. 경쟁 그래프의 모든 플로우의 패킷 사이즈가 동일하다고 가정하면 GF(f)는<수학식 5>로 도식화 된다.

수학식 5

$$GT(f)=\frac{\frac{1}{d_{\min}}}{\frac{N}{d_{\min}}}=\frac{1}{N}$$

<48>

<49> N은 경쟁 그래프에서 연결되어 있는 전체 플로우의 수이다.

<50> $T(f)\geq GT(f)$ 이면, 플로우 f는 보상 받을 시간이 존재하지 않고, $T(f)<GT(f)$이면, 플로우 f는 보상 받을 시간이 존재한다. 경쟁 그래프에서 연결된 각 플로우가 보상 받아야 하는 시간은 경쟁 관계에 있는 플로우들 중 가장 낮은 데이터률($d_{\min}$)을 가진 플로우가 하나의 패킷을 서비스 할 때, 그 플로우가 보내야 되는 패킷의 수로 나타낼 수 있다. 보상 받아야 하는 시간이 존재하는 플로우 f는 시간 기반 공정성을 달성하기 위해서 f가 전송해야 하는 패킷 수(P(f))는<수학식 6>과 같이 정의된다.

수학식 6

$$P(f)=\frac{d_f}{d_{\min}}$$

<51>

<52> 따라서, 도 3의 F0는 dmin을 갖는 F2와 비교하여 시간 기반 공정성을 달성하기 위해서 5개의 패킷을 전송해야 한다. 시간 기반 공정성(TF, Time-Based Fairness)은<수학식 7>과 같이 정의한다.

수학식 7

$$TF=\left|\alpha_i^t(t_1,t_2)-\alpha_j^t(t_1,t_2)\right|$$

<53>

<54> 상기 수학식에서 $\alpha_i^t(t_1,t_2)$는 플로우 i가 시간 t_1부터 시간 t_2 까지 점유하는 채널 점유 시간이다. <수학식 7>의 값이 0에 근접할수록 이상적인 시간 기반 공정성을 제공한다.

<55> $d_{\min}$을 갖는 플로우 f_{dmin}이 받는 최소 서비스의 양은 EMLM-FQ의 <수학식 1>에서 정의한 각 플로우가 받는 최소 서비스의 양에서 유도된다. 그 이유는 도 3에서 모든 플로우의 데이터률이 $d_{\min}$이라 가정할 때, EMLM-FQ에서 제공하는 각 플로우의 최소 서비스 양을 도 6과 같이 제공하기 때문이다.

<56> 예를 들어 최소 데이터률을 갖는 F2가 받는 최소 서비스량은 F2의 데이터률이 모든 플로우의 데이터률이라고 가정했을 때, EMLM-FQ에서 F2가 받는 최소 서비스 양과 같다. 따라서 f_{dmin}이 받는 최소 서비스 양

$(W_{f_{dmin}}(t_1,t_2))$ 은 <수학식 8>과 같이 정의된다.

수학식 8

$$W_{f_d\min}(t_1,t_2) \geq d_{\min}\frac{r_f}{k\sum_{g\in S} r_g}(t_2-t_1)-\alpha$$

<57>

<58> 여기서 r_g와 r_f는 각각 플로우 g와 f의 가중치이며, S는 플로우 경쟁 그래프에서 플로우 f와 직접 연결되어 있는 플로우들의 집합이고 α와 k는 각 플로우의 경쟁관계에 따라 정의한 위치 종속적 상수이다. 플로우의 가중치가 동일하다고 가정하면 $W_{f_{dmin}}(t_1,t_2)$ 는 <수학식 9>과 같다. n은 플로우 경쟁 그래프에서 f와 직접 연결되어 있는 플로우의 수이다.

수학식 9

$$W_{f_d\min}(t_1,t_2) \geq d_{\min}\frac{1}{kn}(t_2-t_1)-\alpha$$

<59>

<60> d_{min}보다 큰 데이터률을 가진 플로우가 받는 최소 서비스의 양 $W_{f_d}(t_1,t_2)$ 은 <수학식 10>과 같이 정의된다.

수학식 10

$$W_{f_d}(t_1,t_2) \geq \frac{d_f}{d_{\min}}W_{f_d\min}(t_1,t_2)$$

<61>

<62> 3. 패킷 스케줄링

<63> 본 발명의 에드혹 패킷 스케줄링 방법의 각 플로우의 서비스 태그 할당 방법과 서비스 태그 정보에 대한 교환 방법은 EMLM-FQ를 따른다. 하지만, 시간 기반 공정성을 제공하기 위하여 각 플로우의 데이터률에 따라 전송해야 하는 패킷 수를 계산하기 위해서 모든 노드는 인접한 노드들의 플로우에 대한 데이터률 정보를 듣고 자신의 로컬 테이블에 저장하고, 자신의 로컬 테이블에 존재하는 플로우의 데이터률과 비교하여 가장 낮은 데이터률(d_{min})을 저장하여 데이터 전송 시 제어 프레임(RTS, CTS, DS, ACK)에 포함시켜 전송한다. 만약 로컬 테이블의 d_{min}보다 낮은 데이터률에 대한 정보를 듣는다면, 로컬 테이블의 d_{min}을 그 데이터률로 저장하고 제어 프레임에 삽입하여 전송한다. 따라서 경쟁 그래프에 연결된 플로우를 가진 모든 송신 노드는 경쟁하는 플로우 중 가장 낮은 데이터률에 대한 정보를 얻게 된다.

<64> 상기 로컬 테이블은 플로우 Id, 시작 태그, 종료 태그, 플로우 데이터 레이트(rate), 네트워크상에 존재하는 최소 데이터률(dmin)을 포함하는 테이블로 나타낼 수 있다.

<65> 또한, 스케줄링 과정 중에서 d_{min}과 송신 노드가 서비스 하려는 플로우의 데이터률과 비교하여 송신 노드가 전송해야 하는 패킷 수(P(f), <수학식 6>)를 계산하고, 만약 P(f)가 1보다 크고, 백로그 된 패킷이 존재한다면, P(f)개의 패킷은 동일한 서비스 태그를 갖고 즉시 전송을 시도한다. 즉, P(f)개의 패킷이 서비스 될 때까지 스

케줄링 과정에서 서비스 태그를 업데이트 하지 않는다. P(f)개의 패킷 중 첫 번째 패킷을 제외한 P(f)-1 개의 패킷들은 전송 시 수신 노드에서 RTS 폐기가 이루어 지지 않게 하기 위해서 RTS 프레임의 1bit 구간(NCB: Not Compare Bit)을 1로 설정하여 전송한다. RTS 프레임에 NCB 구간이 1로 세팅되어 있으면 수신 노드는 현재 채널이 사용 중이 아니면, CTS로 바로 응답하여 그 플로우가 서비스 되도록 한다.

<66>

<67> 본 발명에서의 각 패킷 전송은 RTS-CTS-DS-DATA-ACK 핸드쉐이크 과정을 따르고 패킷이 전송될 수 있는 데이터률을 결정하기 위하여 RBAR 프로토콜을 따른다. 만약 한 노드가 백오프 동안에 다른 노드에 의해서 어떤 전송이 이루어지고 있으면, 그 전송이 끝날 때까지 백오프 타이머를 멈추어 기다리고 전송 중인 플로우에 대한 서비스 태그 정보를 로컬 테이블에서 업데이트한다. 플로우 f의 백오프 타이머가 끝나고, 채널이 사용 중이지 않다면 $\widehat{B_f^R}$ (수신 노드의 로컬 테이블에 따른 플로우 f의 백오프 예측 값)를 포함하는 RTS를 전송한다. RTS를 받은 수신 노드는 자신의 로컬 테이블에서 플로우 f에 대한 백오프 값 B_f^R 이 $\widehat{B_f^R}$ 보다 작거나 같으면 CTS로 응답하고, 그렇지 않으면 CTS를 전송하지 않는다.

<68> 송신 노드와 수신 노드의 한 홉 거리에 있는 모든 이웃 노드들에게 플로우의 서비스 태그 정보, 데이터률과 네트워크에서 가장 낮은 데이터률(d_{min})에 대한 정보를 알리기 위해서 제어 프레임(RTS, CTS, DS, ACK)에 서비스 태그와 d_{min}을 포함하여 전송하고, CTS, DS, ACK 프레임에 플로우의 데이터률을 포함하여 전송한다.

<69> RTS에 플로우의 데이터률을 전송하지 않는 이유는 다중 데이터률 적응 기법으로 RBAR를 채택하고 있기 때문에, 수신 노드가 선택한 데이터률을 CTS에 포함하여 전송하기 전까지 송신 노드는 전송할 데이터률을 알지 못하기 때문이다. 또한, 플로우에 대한 업데이트 서비스 태그는 RTS, CTS 프레임에 포함하지 않는다. 그 이유는 RTS와 CTS는 채널 획득을 보장하지 않기 때문이다. RTS-CTS 과정이 성공한 이후에 DS와 ACK 프레임에 플로우에 대한 업데이트 태그를 붙여 주변 노드들에게 공지한다. 이로써, 스케줄링을 위한 정확한 플로우의 서비스 태그 정보가 전달된다.

<70> 경쟁 그래프에서 한 홉 거리에 있는 플로우에 대한 정보는 송신 노드와 수신 노드에 분산되어 있을 수 있다. 따라서 송신 노드의 플로우 f에 대한 백 오프 값 설정 방법은 송신 노드와 수신 노드에서 유지되는 테이블을 고려하여 백오프 값을 설정해야한다.

<71> 즉, $B_f = B_f^S + B_f^R$ 이다. B_f는 플로우 f의 백오프 값이고, B_f^S는 송신 노드의 로컬 테이블에 따른 플로우 f의 백오프 값, B_f^R은 수신 노드의 로컬 테이블에 따른 플로우 f의 백오프 값이다.

<72> 그러나 송신 노드에서는 B_f^S는 정확히 알지만, B_f^R은 알 수 없다. 따라서 송신 노드가 B_f^R을 예측할 수 있도록, 수신 노드는 ACK 프레임에 두 개의 파라미터(M_f,b_f)를 추가하여 전송한다. M_f는 수신 노드의 로컬 테이블에서 플로우 f가 서비스하기 전에, 서비스 되어야 하는 플로우들의 바이트 수이다. M_f는 <수학식 11>과 같이 계산된다.

수학식 11

<73>
$$M_f = \sum_{j \in B} (T_f - T_j) w_f$$

<74> 상기 수학식에서 w_f는 플로우 f의 가중치이고, T_j는 수신 노드의 로컬 테이블에서 플로우 f의 서비스 태그

(T_f)보다 작은 플로우의 서비스 태그이다. B 는 수신 노드의 로컬 테이블에서 서비스 태그보다 작은 서비스 태그를 갖는 플로우들의 집합이다. b_f 는 수신 노드의 테이블에서 플로우 f의 백오프 기간을 나타낸다. 따라서 송신 노드가 M_f, b_f 를 ACK 프레임을 통해 받으면 임의의 시간 t에서의 $\widehat{B_f^R}$ 은 <수학식 12>와 같이 계산할 수 있다.

수학식 12

$$B_f^R \approx b_f \cdot (M_f - d_{\min} \cdot (t - t_f)) / M_f$$

<75>

<76> 상기 수학식에서 t_f 는 이전에 ACK 프레임을 받은 시간이다.

<77> 본 발명에서의 패킷 스케줄링은 다음과 같은 절차를 따른다.

<78> 노드 n에서 채널이 사용 중이 아닌 것이 감지되면,

<79> a) 전송할 플로우 f가 로컬 테이블 E_n 에서 가장 작은 서비스 태그를 가지고 있다면 즉시 HOL(Head of Line) 패킷을 전송한다. 만약, f의 데이터률이 d_{min}보다 크다면, 시간 공정성을 달성하기 위해 전송해야 하는 패킷 수(P(f))를 구하고, P(f)개의 패킷들의 서비스 태그를 업데이트하지 않고 동일한 서비스 태그를 갖는다. P(f)개의 패킷들은 즉시 전송을 시도하고, P(f)개의 패킷들이 서비스 된 후 패킷의 서비스 태그를 업데이트한다.

<80> b) f보다 작은 서비스 태그를 갖는 플로우가 자신의 로컬 테이블(E_n)에 존재 하면, f의 백오프 값 B_f 을 $B_f^S + B_f^R$ 로 설정하고, 백오프 타이머가 동작을 시작한다.

<81> c) f의 백오프 타이머가 만료되고, 채널이 사용 중이 아니라면 f의 HOL 패킷을 전송한다.

<82> 이하, 도 7을 참조하여 본 발명에 따른 패킷 스케줄링 방법을 설명한다.

<83> 도 7은 본 발명의 실시예에 따른 각 플로우의 패킷의 서비스 순서와 채널 점유시간을 나타낸다.

<84> 그림 3과 같은 네트워크가 존재한다고 가정하자. 각 플로우의 패킷 사이즈와 가중치는 동일하다. 각 플로우 $F_i(i=0,1,2)$ 는 초기 서비스 태그 값으로 각각 $T_0=1, T_1=2, T_2=3$ 을 갖는다. 또한 $\frac{packetsize}{weight}=10$ 이라 가정하면, 도 7과 같이 F0는 가장 작은 서비스 태그를 갖기 때문에 즉시 pkt 0의 전송을 시작한다. 전송을 시작하기 전에 F0의 데이터률은 네크워크에 존재하는 가장 낮은 데이터률 보다 5배 높기 때문에, 시간 기반 공정성을 달성하기 위해서 전송해야 할 패킷 수를 계산하고 F0의 첫 번째 패킷 pkt 0을 전송할 때, DS와 ACK 프레임에 업데이트 태그를 붙이지 않는다. 그 이유는, 만약 노드 N1이 F0의 업데이트 태그를 수신하면 즉시 F1의 pkt 0(703)를 서비스하기 위해 채널에 접근을 시도할 것이기 때문이다. 따라서, 업데이트 태그를 붙이지 않음으로써, F0의 전송해야 할 나머지 패킷에게 전송할 기회를 줄 수 있다. pkt 0(703) 전송 후에, F0는 스케줄링 과정에서 전송해야 할 5개의 패킷 중 마지막 패킷 pkt 4가 서비스 되기 전까지 서비스 태그를 업데이트 하지 않기 때문에 F0는 가장 작은 서비스 태그를 가지므로 F0의 pkt 1(704) , pkt 2(705), pkt 3(706) 및 pkt 4(707)는 즉시 전송을 시도한다. 또한, pkt 1(704), pkt 2(705), pkt 3(706) 및 pkt 4(707)를 전송할 때, RTS 프레임의 NCB(Not Compare Bit)를 1로 설정하여 전송한다. 이러한 방법으로 수신 노드에서

는 값의 비교 없이 RTS 프레임에 대한 응답으로 CTS 프레임을 즉시 전송하도록 할 수 있다. F0의 전송해야 할 5개의 패킷 중 마지막 패킷인 pkt 4(707) 전송시 DS와 ACK 프레임에 업데이트 태그를 붙여서 주변 노드들에게 공지한다. 노드 N1이 F0의 pkt 4(707)의 업데이트 태그를 수신하면 F1의 패킷을 서비스하기 위해서 즉시 채널에 접근한다.

<85> F1의 첫 번째 패킷 pkt 0(708)을 전송하기 전에, F0와 같이 P(f)=2를 계산하고, F1의 pkt 0(708)의 전송 후에 pkt 1(709) 전송시 NCB를 1로 설정하고 DS와 ACK 프레임 전송 시 업데이트 태그를 붙인다. F2는 P(f)=1이기 때문에 추가로 전송할 패킷이 존재하지 않는다. 따라서 F2의 패킷 pkt 0(10)을 전송할 때, DS와 ACK 프레임 전송 시 업데이트 태그를 붙여서 전송한다.

<86> 도 8은 본 발명에 따른 패킷 스케줄링 방법을 나타낸 전체 흐름도이다.

<87> 상기에서 설명한 절차를 요약하여 나타내면 도 8과 같이 나타낼 수 있다.

<88> 노드는 패킷을 전송하기 전에 먼저 네트워크상에 존재하는 최소 데이터률(dmin)을 수신하고(S801) 패킷의 개수를 계산한다(S802). 패킷 수는 스케줄링 과정 중에서 d_{min}과 송신 노드가 서비스 하려는 플로우의 데이터률과 비교하여 수학식 6에 의해 구할 수 있다.

<89> 상기 계산 결과 패킷 수가 1보다 큰지 여부를 확인하고(S803) 크다면, 상기 결과보다 하나 적은 패킷을 전송하고 전송시 플래그-NCB(not compare bit)-를 설정하여 전송한다(S804).

<90> 상기 계산된 패킷의 개수에서 마지막 패킷에 업데이트된 서비스 태그를 포함하여 전송한다(S805).

<91> 상기 계산된 패킷의 수가 1 보다 적은 경우에는 업데이트된 서비스 태그를 포함하여 플래그의 설정 등과 같은 절차 없이 바로 전송한다(S806).

<92> 본 발명의 실시예에서는 구체적인 실시예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 안 되며 후술하는 특허 청구의 범위뿐만 아니라 이 특허 청구의 범위와 균등한 것들에 의해 정해져야 한다.

발명의 효과

<93> 본 발명에 따르면, 물리계층에서 다중 데이터률을 제공하는 네트워크 환경에서 각 플로우의 데이터률과 네트워크 상의 가장 낮은 데이터률과 비교하여 각 플로우가 전송해야하는 패킷 수를 계산하여 그 플로우가 전송해야 하는 패킷 수만큼 전송하게 함으로써, 전체 네트워크 처리량을 크게 향상시킨다.

<94> 아울러 본 발명을 이용하면, 무선 에드혹 네트워크와 같이 물리계층에서 각 플로우별 데이터률이 상이한 네트워크에서 각 데이터률에 기반한 시간 기반 공정성을 제공함으로써, 네트워크 처리량을 크게 향상 시킬 수 있다.

도면의 간단한 설명

<1> 도 1은 본 발명이 적용되는 일반적인 에드혹 네트워크의 노드 그래프

<2> 도 2는 본 발명이 적용되는 도 1의 플로우 경쟁관계를 나타내는 도면.

<3> 도 3은 본 발명이 적용되는 다중 데이터률을 사용하는 에드혹 네트워크의 노드 그래프

<4> 도 4는 본 발명이 적용되는 도 3의 플로우 경쟁관계를 나타내는 도면.

<5> 도 5는 본 발명의 실시예에 따른 처리량 기반 공정성을 제공하는 패킷 스케줄링과 시간 기반 공정성을 제공하는 패킷 스케줄링의 각 플로우 서비스 순서와 채널 점유시간을 비교하는 도면.

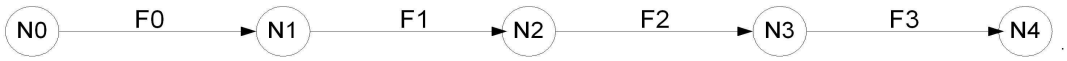
<6> 도 6은 본 발명의 실시예에 따른 다중 데이터률을 제공하는 네트워크에서 시간 기반 공정성을 제공하는 패킷 스케줄링과 모든 플로우의 데이터률이 가장 낮은 데이터률일 때의 EMLM-FQ의 각 플로우 서비스 순서와 채널 점유시간을 비교하는 도면.

<7> 도 7은 본 발명의 실시예에 따른 각 플로우의 패킷의 서비스 순서와 채널 점유시간을 나타내는 도면.

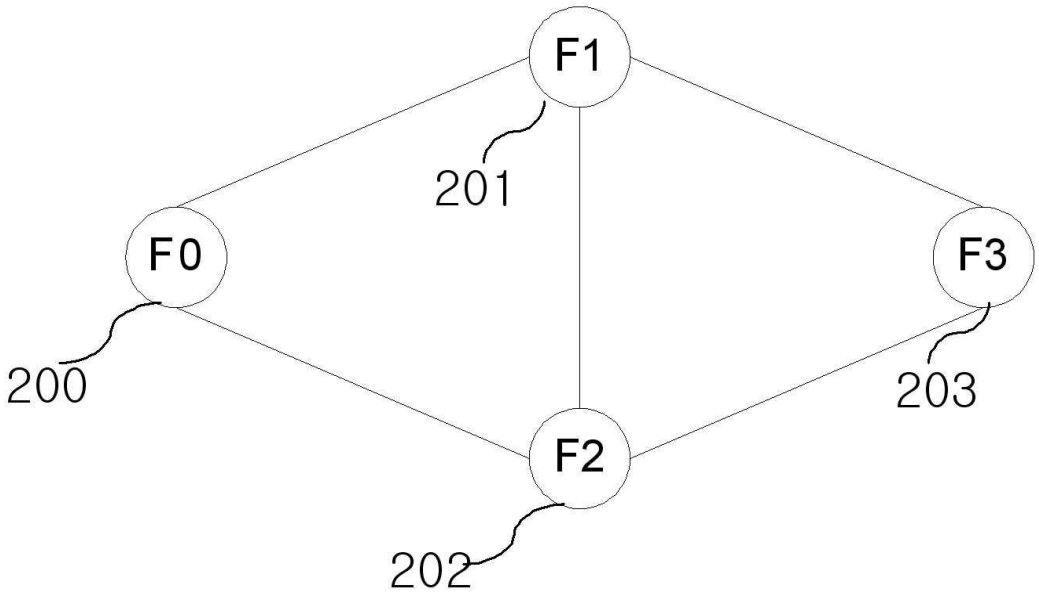
<8> 도 8은 본 발명의 실시예에 따른 패킷 스케줄링 방법의 전체 흐름도

도면

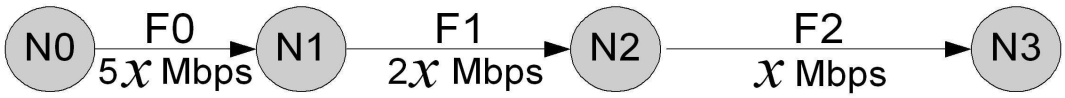
도면1



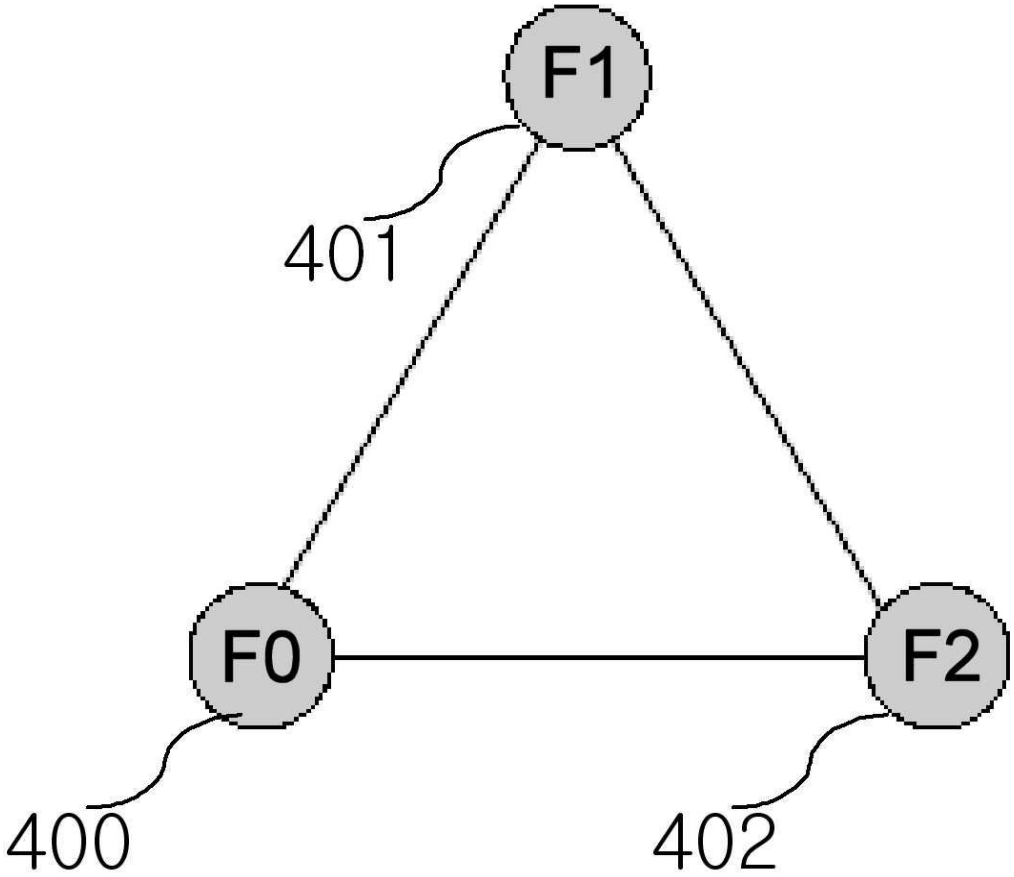
도면2



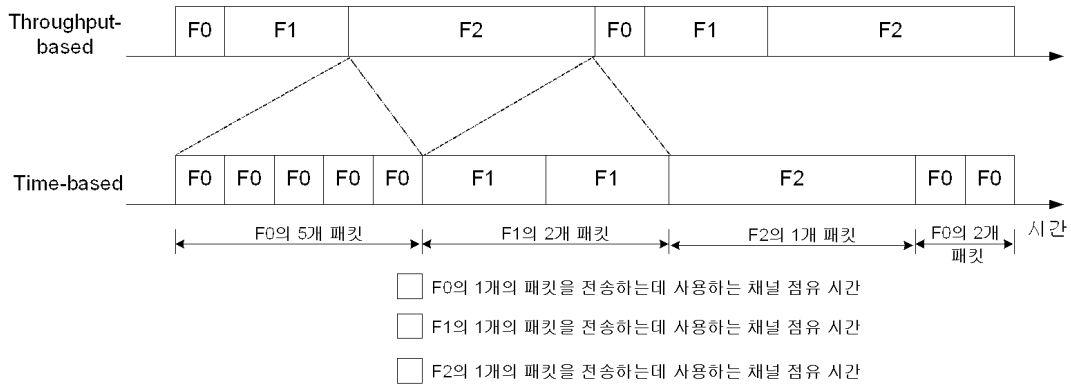
도면3



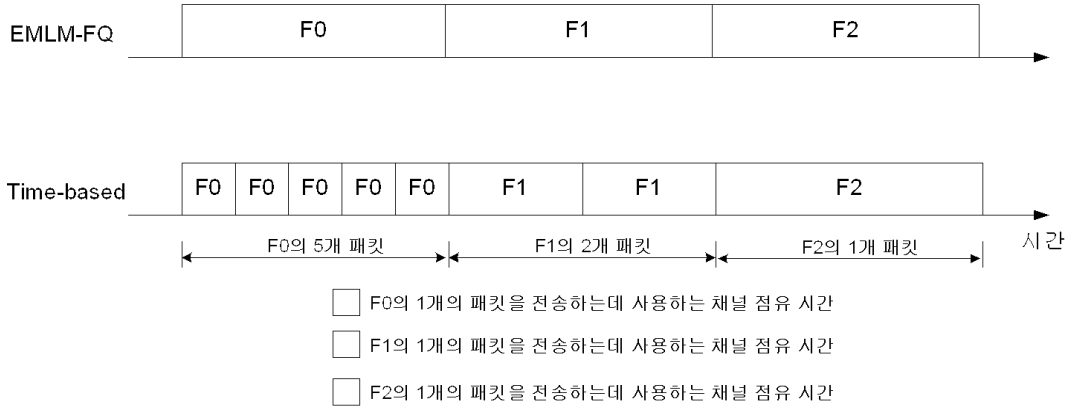
도면4


F1
401
F0
F2
400
402

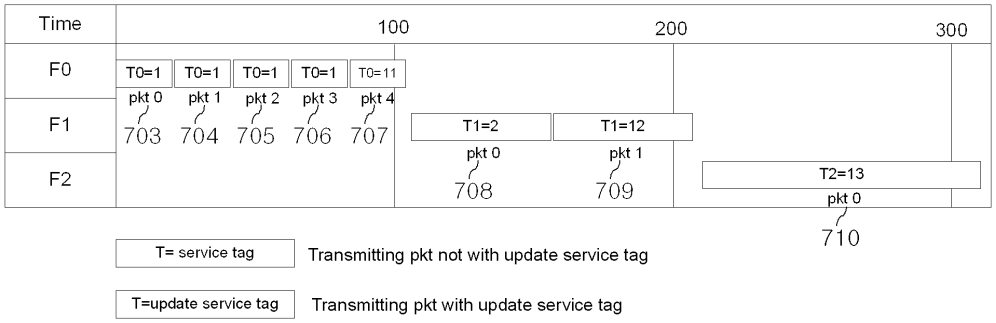
도면5


Throughput-based
F0
F1
F2
F0
F1
F2
Time-based
F0
F0
F0
F0
F0
F1
F1
F2
F0
F0
F0의 5개 패킷
F1의 2개 패킷
F2의 1개 패킷
F0의 2개 패킷
시간
F0의 1개의 패킷을 전송하는데 사용하는 채널 점유 시간
F1의 1개의 패킷을 전송하는데 사용하는 채널 점유 시간
F2의 1개의 패킷을 전송하는데 사용하는 채널 점유 시간

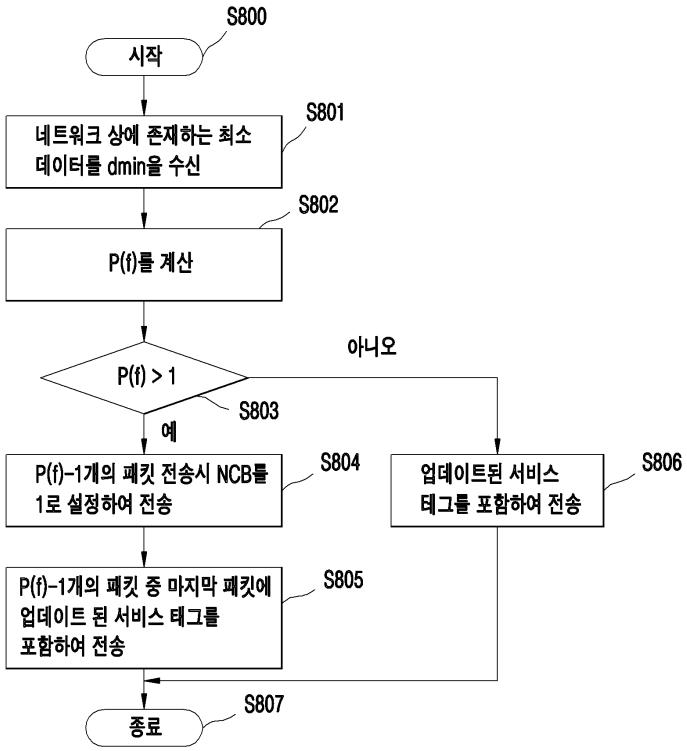
도면6


EMLM-FQ
F0
F1
F2
Time-based
F0
F0
F0
F0
F0
F1
F1
F2
F0의 5개 패킷
F1의 2개 패킷
F2의 1개 패킷
시간
F0의 1개의 패킷을 전송하는데 사용하는 채널 점유 시간
F1의 1개의 패킷을 전송하는데 사용하는 채널 점유 시간
F2의 1개의 패킷을 전송하는데 사용하는 채널 점유 시간

도면7


Time
100
200
300
F0
F1
F2
T0=1
T0=1
T0=1
T0=1
T0=11
pkt 0
pkt 1
pkt 2
pkt 3
pkt 4
703
704
705
706
707
T1=2
T1=12
pkt 0
pkt 1
708
709
T2=13
pkt 0
710
T= service tag
Transmitting pkt not with update service tag
T=update service tag
Transmitting pkt with update service tag

도면8


S800
시작
S801
네트워크 상에 존재하는 최소 데이터를 dmin을 수신
S802
P(f)를 계산
P(f) > 1
S803
아니오
예
S804
P(f)-1개의 패킷 전송시 NCB를 1로 설정하여 전송
S805
P(f)-1개의 패킷 중 마지막 패킷에 업데이트 된 서비스 태그를 포함하여 전송
S806
업데이트된 서비스 태그를 포함하여 전송
S807
종료