



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2007년11월02일
(11) 등록번호 10-0773036
(24) 등록일자 2007년10월29일

(51) Int. Cl.

H04L 12/28(2006.01)

(21) 출원번호 10-2006-0063520
(22) 출원일자 2006년07월06일
심사청구일자 2006년07월06일

(56) 선행기술조사문헌

US20040185841 A

US20040100936 A

Y.Q. Chen 외, "Service index based fairness scheduling in wireless ad-hoc networks", 제16회 통신정보 합동학술대회(2006.4.28)

(73) 특허권자

인하대학교 산학협력단

인천 남구 용현동 253 인하대학교

(72) 발명자

유상조

인천 남구 용현4동 인하대학교

진영천

인천 남구 용현4동 인하대학교

노권문

서울 금천구 가산동 두산아파트 116동 1302호

(74) 대리인

이은철

전체 청구항 수 : 총 2 항

심사관 : 김병성

(54) 무선 에드혹 네트워크에서 서비스 색인 기반 공정 패킷스케줄링 방법

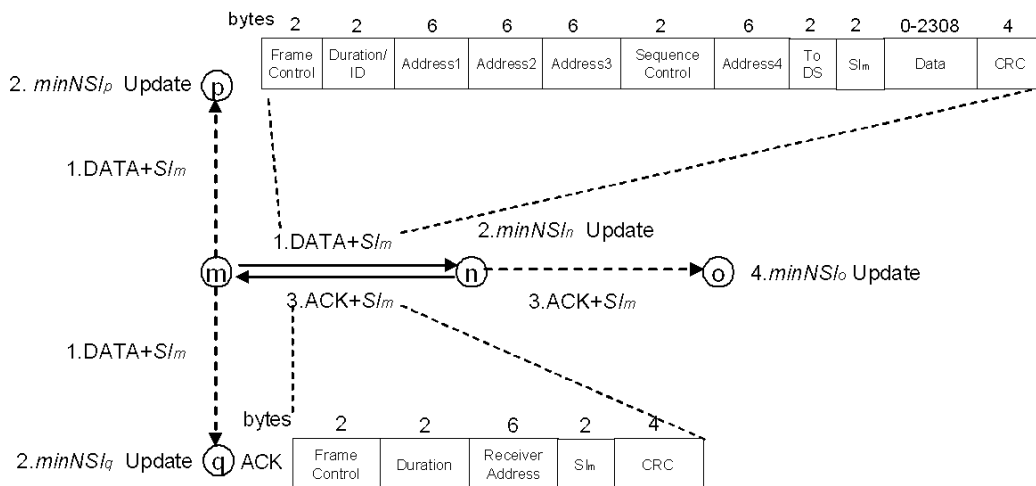
(57) 요약

본 발명은 무선 에드혹 망에서의 패킷 스케줄링 방법에 관한 것으로, 특히 노드에 존재하는 각 플로우가 받은 서비스 양을 계산하여 서비스 색인을 결정하고 이웃 노드들과 계산된 서비스 색인을 교환하여 MAC 계층의 백오프 값을 결정함으로써 전체 네트워크의 공정성을 향상시키는 에드혹 패킷 스케줄링 방법에 관한 것이다.

본 발명은 에드 혹 네트워크에서 패킷 스케줄링 방법에 있어서,

각 노드의 플로우가 받은 서비스 양을 계산하여 플로우 서비스 색인(SI_m^i)과 노드 서비스 색인(SI_m)을 설정하는 단계; 상기 계산된 노드 서비스 색인을 이웃 노드들과 교환하는 단계; 및 상기 교환된 노드 서비스 색인을 기반으로 MAC 계층의 백오프 값을 설정하는 단계; 를 포함하는 것을 특징으로 한다.

대표도 - 도4



특허청구의 범위

청구항 1

에드 혹 네트워크(Ad-hoc Networks)에서 패킷 스케줄링(Packet Scheduling) 방법에 있어서,

각 노드의 플로우가 받은 서비스 양을 계산하여 플로우 서비스 색인(SI_m^i)과 노드 서비스 색인(SI_m)을 설정하는 제 1 단계;

$$SI_m^i(t_1, t_2) = \frac{W_i(t_1-t_2)}{\phi_i \times (t_1-t_2)}$$

(상기 플로우 서비스 색인(SI_m^i)은 $W_i(t_1-t_2)$ 로 설정되고, $W_i(t_1-t_2)$ 는 플로우 i가 시간 t_1 에서 t_2 까지 받은 서비스 양을 나타내며, ϕ_i 는 플로우 i의 가중치를 나타내고,

$SI_m^i(t_1, t_2)$ 는 노드 m의 플로우 i가 시간 t_1 에서 t_2 까지 받은 총 서비스 양을 나타낸다.

$$SI_m = \frac{\sum_{i=0}^{f(m)} SI_m^i(t_1, t_2)}{f(m)}$$

또한, 상기 노드 서비스 색인(SI_m)은 $f(m)$ 로 설정되고, $f(m)$ 은 노드 m에 존재하는 플로우의 수를 나타내며, SI_m 은 노드 m에 존재하는 모든 플로우들에 대한 평균 서비스 색인을 나타낸다.)

상기 계산된 노드 서비스 색인(SI_m)을 이웃 노드들과 교환하는 제 2 단계; 및

상기 교환된 노드 서비스 색인을 기반으로 MAC 계층의 백오프 값을 설정하는 제 3 단계; 를 포함하되,

상기 제 1 단계는,

상기 플로우 서비스 색인(SI_m^i)은 주기적으로 계산되고,

$$SI_m^i(t + \Delta t) = \alpha \times SI_m^i(t) + (1 - \alpha) \times SI_m^i(t, t + \Delta t)$$

$$SI_m^i(t, t + \Delta t) = \frac{W_i(t, t + \Delta t)}{\phi_i \times \Delta t}$$

($SI_m^i(t, t+\Delta t)$ 는 주기적으로 계산되는 상기 플로우 서비스 색인으로서 Δt 동안의 플로우 서비스 색인 값을 나타내며, α 는 이전 SI_m^i 의 가중치의 값으로서 $0 \leq \alpha \leq 1$ 의 범위에서 결정되고, $W_i(t, t+\Delta t)$ 는 Δt 시간 동안에 플로우 i가 받은 서비스 양을 나타낸다.)에 의해 계산되는 단계; 를 더 포함하고,

상기 제 2 단계는,

계산된 노드 서비스 색인을 이웃 노드들과 교환할지 여부를 결정하기 위해 각 노드의 패킷 스케줄러가 최소 서비스 태그를 가진 패킷이 전송되어야만 하는지 여부를 주변 노드의 전송 상태에 따라 결정하는 단계, 즉 하나의 노드에 존재하는 각 플로우의 다른 종료 서비스 태그 값에 대한 임계값을 미리 정의하여 각 플로우의 종료 서비스 태그의 최대 차이값이 상기 임계값 이하가 되면 다음 플로우가 스케줄링 되도록 결정하는 단계;

데이터 프레임에 노드 서비스 색인을 붙여서 전송하거나 제어 프레임에 노드 서비스 색인을 포함하기 위한 2 바이트의 필드를 할당하며, 상기 노드 서비스 색인이 프레임에 포함되어 전송되는지 여부를 확인하기 위해서 제어 프레임의 네 번째 비트에 플래그를 설정하는 단계; 및

상기 노드 서비스 색인의 교환에 의해 수신된 노드 서비스 색인이 최소 이웃 서비스 색인

$$minNSI_m(t_1, t_2) = \min_{\forall n \in N_m} \{SI_n(t_1, t_2)\}$$

(이때, N_m 은 노드 m 의 이웃 노드들의 집합을 나타내며, $minNSI_m(t_1, t_2)$ 는 노드 m 의 모든 이웃 노드의 노드 서비스 색인값 중에서 가장 작은 값을 나타낸다.)보다 작은 경우, 수신한 상기 노드 서비스 색인값이 새로운 최소 이웃 서비스 색인이 되도록 업데이트하는 단계; 를 더 포함하며,

상기 제 3 단계는,

각 노드에서 플로우의 ID, SI_m^i 및 노드의 최소 이웃 서비스 색인을 저장하는 하나의 로컬 테이블을 유지하는 단계;

각 노드가 이웃 노드들에게 노드 서비스 색인을 전송한 후 자신의 로컬 테이블에서 차이색인(DI_m^i)을 계산하는 단계;

$$DI_m^i = \frac{SI_m^i - minNSI_m}{SI_m^i}$$

(상기 차이색인은 로 계산되고, SI_m^i 는 노드 m 의 플로우 i 가 받는 총 서비스의 양을 나타내고, $minNSI_m$ 는 최소 이웃 서비스 색인을 나타낸다.) 및

하나의 플로우에 대한 백오프 값을 계산하거나 충돌이 발생했을 때 새로운 백오프 값을 계산하는 단계;

(상기 백오프값은 $backoff = \{F(p_i^k) - S(p_i^k)\} \times sf \times \rho$ 로 계산되고, $F(p_i^k)$ 는 종료 태그, $S(p_i^k)$ 는 시작태그를 나타내며, sf 는 채널 유용성을 위해서 얼마간의 효율성의 정도를 보장하기 위하여 선택된 상수보다 작은 값으로 초기 백오프 값을 결정하는 스케일 요소를 나타내고, ρ 는 평균값으로 1을 가지는 랜덤 변수를 나타낸다.) 를 더 포함하는 패킷 스케줄링 방법.

청구항 2

제 1 항에 있어서,

상기 MAC 계층의 백오프 값을 설정하는 단계(제 3 단계)는 상기 차이색인(DI_m^i) 값이 1 보다 작거나 같으면 차이 색인에 1을 더한 값에 상기 계산된 백오프값을 곱한 값으로 백오프값을 설정하고 상기 차이색인(DI_m^i) 값이 1 보다 큰 경우에는 상기 계산된 백오프값에 2를 곱한 값으로 백오프값을 설정하는 것을 특징으로 하고,

상기 충돌이 발생했을 때 새로운 백오프값을 계산하는 단계는 상기 계산된 백오프값에 2를 곱한 값이 미리 설정되어 있는 충돌 임계값보다 큰 경우에는 상기 계산된 백오프값에 2를 곱한 값보다 작은 임의의 값을 선택하여 새로운 백오프값으로 하는 것을 특징으로 하는 패킷 스케줄링 방법.

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <7> 본 발명은 무선 에드혹 망에서의 패킷 스케줄링 방법에 관한 것으로, 특히 노드에 존재하는 각 플로우가 받은 서비스 양을 계산하여 서비스 색인을 결정하고 이웃 노드들과 계산된 서비스 색인을 교환하여 MAC 계층의 백오프 값을 결정함으로써 전체 네트워크의 공정성을 향상시키는 에드혹 패킷 스케줄링 방법에 관한 것이다.
- <8> 무선 에드혹 네트워크는 단말기와 라우터 역할을 동시에 하는 이동 노드들로 구성된 자기 조직적 네트워크이다. 최근에 멀티미디어 트래픽 요구의 급속한 증가로 인해 에드혹 네트워크에서 QoS(Quality of Service)를 제공하는 방법이 당면 과제로 급부상하고 있다. 이러한 QoS를 제공하는 방법 중 한 가지 방법이 패킷 스케줄링 기법이다. 패킷 스케줄링 기법은 공정한 대역폭 할당과 자원 유용성(resource utilization) 최대화라는 중요한 두 가

지 설계 목적을 갖는다. 공정성은 잘 행동하는 사용자가 대역폭을 초과 요구하는 사용자들 때문에 불이익을 받는 것을 방지하고, 자원 유용성 최대화는 부족한 무선 자원을 이용하여 웹 브라우징, 비디오 컨퍼런스, 큰 파일의 원격 전송과 같은 통신 집중적 어플리케이션(communication-intensive application)을 효율적으로 제공하는 데 중요하다.

<9> 무선 에드혹 네트워크에서 IEEE 802.11과 같은 경쟁 기반 MAC 프로토콜은 구현이 용이하기 때문에 대중적으로 사용된다. 그러나 무선 에드혹 네트워크에서 IEEE 802.11의 DCF(Distributed Coordination Function)는 공정성에 문제가 있는 것으로 널리 알려져 있다. 스케줄링 기법은 패킷들의 스케줄링 순서를 명백하게 결정하기 때문에 더 나은 공정성을 제공할 수 있어서 IEEE 802.11의 공정성 문제에 대한 하나의 대안이 될 수 있다. 기존에 연구된 패킷 스케줄링 기법에는 GPS(Generalized Processor Sharing), WFQ(Weighted Fair Queueing), CIFQ(Channel-Condition Independent Fair Queueing)등이 있다. 이러한 기법들은 각 노드에 존재하는 플로우들 사이에서 공정한 서비스를 제공하는 지역적 공정성(local fairness)을 제공한다. 그러나 지역적 공정성을 제공하는 스케줄링 알고리즘은 에드혹 네트워크의 위치 종속적 채널 경쟁과 불충분한 스케줄링 정보 때문에 전역적 공정성(global fairness) 달성을 위해 이용하는 것은 부적합하다. 왜냐하면, 에드혹 네트워크의 무선 전송은 지역적인 브로드캐스팅이고, 공유 채널에 대한 경쟁은 위치 종속적이기 때문에 유선 네트워크 또는 셀룰러 네트워크와 다르게 더 이상 패킷 스케줄링이 송신 노드의 지역적인 결정만으로 이루어질 수 없다. 즉, 다시 말하면 기존의 패킷 스케줄링 방법은 전역적 공정성(global fairness)을 제공할 수 없다. 여기서, 전역적 공정성이란 에드혹 네트워크에 존재하는 모든 플로우들이 네트워크 전반에 걸쳐 공정한 서비스를 받을 수 있는 것을 의미한다.

<10> 이와 같이, 에드혹 네트워크에서 공정성은 전역적 공정성과 지역적 공정성으로 나누어진다. 전역적 공정성이란 같은 공유 매체에 접근하는 모든 플로우가 공정한 서비스를 받는 것을 의미하고, 지역적 공정성이란 한 노드 안에 존재하는 각 플로우들이 공정한 서비스를 받는 것을 의미한다.

<11> 무선 에드혹 네트워크에서의 기존 패킷 스케줄링 알고리즘들은 지역적 공정 큐잉 알고리즘인 WFQ(Weighted Fair Queueing)와 WF2Q(Worst-Case Fair Weighted Fair Queueing)등이 있으며, 이들은 GPS(Generalized Processor Sharing)원리를 기반으로 한다. WFQ는 각 플로우의 패킷마다 두 개의 태그(start tag, finish tag)를 할당한다. 시작태그(Start tag, $S(p_i^k)$)와 종료태그(Finish tag, $F(p_i^k)$)는 <수학식 1> 같이 정의된다. 여기서 p_i^k 는 플로우 i의 k번째 패킷을 의미하고, l_i^k 는 플로우 i의 k번째 패킷의 길이를 나타낸다. $A(p_i^k)$ 는 플로우 i의 k번째 패킷이 도착한 시간을 나타낸다.

수학식 1

<12>
$$S(p_i^k) = \max \{ v[A(p_i^k)], F(p_i^{k-1}) \}$$

<13>
$$F(p_i^k) = S(p_i^k) + \frac{l_i^k}{\phi_i}$$

<14> 여기서 $F(p_i^1) = 0$ 이고, 가상 시간(virtual time) $v(t)$ 는 <수학식 2>와 같이 정의된다. 또한, C는 채널 용량이고, $B(t)$ 는 시간 t에서 서버에 백로그된 플로우의 집합이다.

수학식 2

<15>
$$\frac{dv(t)}{dt} = \frac{C}{\sum_{i \in B(t)} \phi_i}$$

<16> 위와 같은 식을 기본으로 하는 WFQ, WF2Q, CIF-Q와 같은 공정 큐잉 알고리즘을 무선 에드혹 네트워크의 패킷

스케줄링에 적용하면, 지역적 공정성만을 제공하고, 전역적 공정성은 제공하지 않는다. 또한, WFQ와 같은 time-stamp 방식을 에드혹 패킷 스케줄링에 적용하면 다음과 같은 두 가지 문제점을 갖는다. 첫째, 각 노드는 모든 이웃 노드의 플로우에 대한 서비스 태그 정보를 교환하고, 유지해야 한다. 이는 서비스 태그 정보를 전송하기 위해 추가적인 오버헤드를 발생시키기 때문에 네트워크의 처리량을 감소시킨다. 두 번째, 전송할 백로그 된 플로우가 없음에도 불구하고, 시스템의 가상시간은 초기화될 수 없다.

발명이 이루고자 하는 기술적 과제

- <17> 따라서, 본 발명의 목적은 상기한 문제점을 해결하기 위하여 에드혹 네트워크에서 구현이 용이하고 지역적 공정성과 전역적 공정성을 동시에 만족시키는 패킷 스케줄링을 제공함에 있다.
- <18> 또한, 본 발명의 목적은 전역적 공정성을 달성하기 위하여 각 플로우가 받은 서비스의 양을 계산하여 서비스 색인을 결정하고 이 서비스 색인을 이웃 노드들과 교환하는 방법과 수신한 이웃 노드들의 서비스 색인과 비교하여 MAC 계층의 백오프 시간을 조정하는 패킷 스케줄링 방법을 제공함에 있다.
- <19> 또한, 본 발명의 목적은 전역적 공정성을 달성하기 위하여 각 플로우의 스케줄링을 위해서 최소의 제어 정보만을 MAC 계층의 제어 프레임에 포함하여 전송함으로써 네트워크 처리량의 감소를 최소화하는 패킷 스케줄링 방법을 제공함에 있다.
- <20> 또한, 본 발명의 목적은 에드혹 네트워크에서 채널 유용성을 최대화 하는 방법을 포함하는 패킷 스케줄링 방법을 제공함에 있다.

발명의 구성 및 작용

- <21> 상기한 목적을 달성하기 위한 본 발명은 에드 혹 네트워크에서 패킷 스케줄링 방법에 있어서,
- <22> 각 노드의 플로우가 받은 서비스 양을 계산하여 플로우 서비스 색인(SI_m^i)과 노드 서비스 색인(SI_m)을 설정하는 단계;
- <23> 상기 계산된 노드 서비스 색인을 이웃 노드들과 교환하는 단계; 및
- <24> 상기 교환된 노드 서비스 색인을 기반으로 MAC 계층의 백오프 값을 설정하는 단계; 를 포함하는 것을 특징으로 한다.
- <25> 이하 본 발명의 바람직한 실시 예에 따른 상세한 설명을 첨부된 도면들을 참조하여 설명한다. 하기에 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.
- <26> 1. 지역적 공정성과 전역적 공정성
- <27> 본 발명은 에드혹 네트워크에서 지역적 공정성과 전역적 공정성을 달성하기 위하여, 각 노드는 지역적 공정 패킷 스케줄링 알고리즘과 전역적 공정 패킷 스케줄링 알고리즘을 결합하여 스케줄링한다. 따라서, 지역적 공정 패킷 스케줄링 알고리즘인 WFQ를 한 노드에 존재하는 각 플로우의 서비스 순서를 결정하는데 이용하고 전역적 공정성을 달성하기 위해서 MAC 계층의 백오프 시간을 조정한다.

<28> 먼저, 무선 에드혹 네트워크에서 각 플로우가 시간 간격 T 동안에 플로우 가중치에 따라 받는 서비스의 양을 나타내기 위해서 “플로우 서비스 색인” 개념을 도입한다. <수학식 1>과 <수학식 2>에 기반하여, <수학식 3> 같은 색인들이 정의된다. 서비스 색인(service index) $SI_m^i(t_1, t_2)$ 는 노드 m의 플로우 i가 시간 t1에서 t2동안

에 받는 총 서비스의 양을 나타낸다. 노드 서비스 색인(node service index) $SI_m(t_1, t_2)$ 는 노드 m에 존재하는 모든 플로우들에 대한 평균 서비스 색인을 나타내고, 최소 이웃 서비스 색인(minimum neighbor service index) $minNSI_m(t_1, t_2)$ 는 노드 m의 모든 이웃 노드들의 노드 서비스 색인 중 최소 노드 서비스 색인을 나타낸다. 이 세 개의 파라미터는 <수학식 3>와 같이 정의된다.

수학식 3

<29>
$$SI_m^i(t_1, t_2) = \frac{W_i(t_1 - t_2)}{\phi_i \times (t_1 - t_2)}$$

<30>
$$SI_m = \frac{\sum_{i=0}^{f(m)} SI_m^i(t_1, t_2)}{f(m)}$$

<31>
$$\min NSI_m(t_1, t_2) = \min_{\forall n \in N_m} \{SI_n(t_1, t_2)\}$$

<32> 상기 <수학식 3>에서 ϕ_i 는 플로우 i 의 가중치이고 $W_i(t_1 - t_2)$ 는 시간 t_1 에서 t_2 까지 플로우 i 가 받은 서비스의 양을 나타낸다. 또한 $f(m)$ 은 노드 m 에 존재하는 플로우의 수를 나타내고, N_m 은 노드 m 의 이웃 노드들의 집합을 나타낸다.

<33> 따라서 노드 m 의 지역적 공정성은 시간 t_1 에서 t_2 동안에 노드 m 의 두 개의 플로우 쌍에 대해서 $|SI_m^i - SI_m^j|$ 값이 0에 가까워야 함을 의미하고, 전역적 공정성은 무선 네트워크에 존재하는 모든 두 개의 플로우 쌍에 대해서 $|SI_m^i - SI_m^j|$ 값이 0에 가까워야 함을 의미한다.

<34> 이하, 도 1 및 도 2를 참조하여 본 발명에 따른 백오프 시간 결정 방법을 설명한다.

<35> 도 1은 본 발명에 적용되는 하나의 플로우에 대한 백오프 값을 계산하기 위한 알고리즘을 나타낸다.

<36> 도 2는 본 발명에서 충돌이 발생했을 때, 새로운 백오프 시간을 계산하는 알고리즘을 나타낸다.

<37> 2. 백오프 시간 결정 방법

<38> 노드 m 의 이웃 노드 n 의 서비스 인덱스(SI) 값을 SI_n 이라고 할 때, 플로우 i 가 다른 노드의 SI_n 보다 큰 SI_m^i 를 가지고 있다면, 플로우 i 를 가지는 노드는 전송 기회를 줄이기 위해서 백오프 타이머를 조정한다. 지역적 공정성은 WFQ를 이용하여 이미 적용되었기 때문에 한 노드의 각 플로우들은 비슷한 서비스 색인을 갖는다. 따라서 $|SI_m^i - SI_m^j|$ 는 0에 가깝다. 각 노드의 SI_m 은 무선 네트워크의 전역적 공정성의 정도를 보여줄 수 있다. 따라서, 본 발명에서는 플로우의 SI_m^i 는 이웃 노드의 SI_n 과 비교되고 적합한 백오프 값을 계산한다. 다음은 위 과정의 절차를 설명한다.

<39> 1) 각 노드는 각 플로우의 ID, SI_m^i 과 노드의 $\min NSI_m$ 를 저장하는 하나의 테이블을 유지한다.

<40> 2) 먼저, 하나의 플로우에 대한 백오프 시간은 <수학식 1>에 표현된 WFQ의 종료태그(finish tag) $F(P_i^k)$ 와 시작태그(start tag) $S(P_i^k)$ 의 차이에 의해서 결정된다. 상기 종료태그(finish tag) $F(P_i^k)$ 와 시작태그(start tag) $S(P_i^k)$ 는 기존의 패킷 스케줄링 기법인 WFQ(Weighted Fair Queuing)에서 사용되는 값이다. 결정된 플로우의 백오프 시간은 <수학식 4>와 같다.

수학식 4

<41>
$$backoff = \{F(p_i^k) - S(p_i^k)\} \times sf \times \rho$$

<42> 여기서, sf 는 채널 유용성을 위해서 얼마간의 효율성의 정도를 보장하기 위하여 선택된 상수보다 작은 값으로 초기 백오프 값을 결정하는 스케일 요소로서, 당업자라면 한 네트워크 안에서 패킷의 길이 또는 전체 노드의 수에 따라 유동적으로 변할 수 있음을 이해할 수 있을 것이다. ρ 는 평균값으로 1을 갖는 랜덤 변수이다. 보통,

ρ 는 [0.9,1.1]사이에 균등하게 분산되어 있다.

상기 sf 및 ρ 를 이용한 백오프 값 계산에 대하여는, 상술한 IEEE 802.11을 참조한다면 용이하게 이해할 수 있을 것이다. 상술한 IEEE 802.11에서는 플로우 간의 충돌 확률을 줄이기 위하여 랜덤 백오프(Random backoff)를 이용하고 있다. 즉, 전송 전에 임의의 백오프 시간을 정하고, 정한 시간을 채널이 아이들(idle)하다고 판단되는 시간 동안 감소시켜 가장 먼저 0에 도달하는 노드가 전송을 행하는 방식을 취한다. 따라서, sf 값은 유동적으로 변할 수 있는 초기 값을 결정하기 위한 것이며, ρ 값은 충돌을 줄이기 위한 것이다.

3) 각 노드는 이웃 노드들에게 SI_m (노드 서비스 색인)을 전송한다.

<43> 삭제

<44> 4) 노드의 모든 이웃 노드들은 위의 정보를 받고 $\min NSI_m$ 을 업데이트 한다.

<45> 5) 전송이 끝난 후에, 전송 노드는 전송한 플로우에 대해서 자신의 로컬 테이블에서 차이 색인(difference index) DI_m^i 를 계산한다. DI_m^i 는 <수학식 5>와 같이 정의된다.

수학식 5

$$DI_m^i = \frac{SI_m^i - \min NSI_m}{SI_m^i}$$

<46>

<47> 하나의 플로우에 대한 백오프 값을 계산하기 위해서, 아래 알고리즘 1의 식을 이용한다.

알고리즘 1 백오프 시간 계산

If ($DI_m^i \leq 1$)
 $backoff = backoff \times (1 + DI_m^i)$
Else ($DI_m^i > 1$)
 $backoff = backoff \times 2$

<48>

<49> 즉, 차이 색인이 1보다 작거나 같으면 차이색인 값에 1에 더하여 백오프 값에 곱하고 그렇지 않으면 백오프 값에 2를 곱한다.

<50> 6) 충돌에 대한 해결책으로 IEEE 802.11 표준에서 제안된 방법을 이용한다. 충돌이 발생했을 때, 새로운 백오프 시간은 아래 알고리즘 2의 식 같이 계산된다.

알고리즘 2 충돌 발생시 백오프 시간 계산

$temp = 2 \times backoff$
If ($temp > collision-threshold$)
 $new\ backoff = random\ value\ between\ [0, temp]$

<51>

<52> 상기 알고리즘 2에서 충돌 임계값(collision-threshold)은 다양하게 설정 가능하나 800이 바람직하다.

<53> 이하, 도 3 및 도 4를 참조하여 본 발명에 따른 채널 유용성 최대화 방법을 설명한다.

- <54> 도 3은 본 발명에 따른 채널 유용성 최대화 방법을 설명하기 위한 네트워크의 예시도이다.
- <55> 도 4는 본 발명에 따른 채널 유용성 최대화 방법을 나타내는 순서도이다.
- <56> 3. 채널 유용성 최대화
- <57> 에드혹 네트워크에서, 서로 간섭을 주지 않는 플로우들은 잠재적으로 동시에 패킷을 전송할 수 있다. 그러나 한 지역 안에서 한 플로우의 전송은 네트워크의 다른 지역에 존재하는 플로우들의 전송에 영향을 미친다. 따라서, 동시에 전송할 수 있는 노드의 선택은 중합 채널 이용률(aggregate channel utilization)을 결정한다. 에드혹 패킷 스케줄링의 전역적 속성은 네트워크에 존재하는 모든 플로우들 사이에서 공정성을 고려하여 동시에 전송 가능한 노드의 효율적인 선택을 요구한다.
- <58> 본 발명은 지역적 공정성과 전역적 공정성을 달성하기 위하여 WFQ를 이용하고 백오프 시간을 조정한다. 하지만, 이러한 방법은 채널 유용성을 감소시킨다. 예를 들어, 도 3과 같은 네트워크에서 노드들과 플로우들이 존재할 때, 시간 t 에서 노드 2(302)가 WFQ에 의해서 전송할 플로우로 $f1(301)$ 을 선택하면 백오프 시간이 지나고 $f1(301)$ 을 전송하기 위하여 채널에 접근한다. 시간 $t + \Delta t$ 에서 노드 4(306)는 $f3(305)$ 를 전송할 플로우로 결정한다. $f1(301)$ 과 $f3(305)$ 는 충돌 때문에 동시에 전송할 수 없다. 그러나 노드 4(306)가 전송할 플로우로 $f4(307)$ 를 선택하면 $f1(301)$ 과 $f4(307)$ 는 충돌없이 전송할 수 있다. 그러므로 이러한 상황에서 WFQ 알고리즘은 채널 자원을 낭비한다.
- <59> 이러한 문제에 대해서, 본 발명은 다음과 같은 방법을 제안한다. 각 노드의 스케줄러는 최소 서비스 태그를 가진 패킷이 전송되어야만 하는지 아닌지를 결정할 수 있다. 예를 들어, 도 3에서 노드 2(302)가 플로우 $f1(301)$ 을 전송중일 때, 노드 4(306)가 WFQ에 의해서 $f3(305)$ 가 전송할 플로우로 결정되더라도 노드 4(306)는 플로우 $f3(305)$ 의 패킷을 전송할 수 없다. 왜냐하면, 노드 4(306)가 $f3(305)$ 를 전송하기 위하여 RTS 패킷을 노드 3(304)에게 전송하면, 노드 3(304)에서는 충돌이 발생하기 때문에 노드 4(306)가 CTS 패킷을 받을 수 없다. 따라서 노드 4(304)의 스케줄러는 $f3(305)$ 의 패킷 전송을 미루고 다음 플로우인 $f4(307)$ 를 스케줄링한다. 따라서 네트워크 처리량은 개선된다. 그러나 이러한 접근은 최소 서비스 태그를 갖는 패킷이 먼저 전송되지 않을 수도 있고, 얼마간의 지연이 발생할 수 있는 문제를 야기한다. 따라서 지연과 처리량 사이에는 trade-off 관계가 성립된다.
- <60> 본 발명에서는 하나의 노드에 존재하는 각 플로우의 다른 종료 서비스 태그(finish service tag) 값에 대한 임계값을 미리 정의한다. 하나의 노드에서 각 플로우의 종료 서비스 태그의 최대 차이값이 임계값 이하가 되면 처리량을 최대화하기 위해서 가장 적합한 다음 플로우가 스케줄링 될 수 있고, 다른 방법으로 비록 채널 자원을 낭비할지라도 가장 작은 종료 서비스 태그(finish service tag)를 갖는 패킷이 전송될 수 있다. 그래서 <수학식 6>에 의해서 지연은 바운드로 된다.

수학식 6

$$\min F_m = \min_{\forall i} [F_m^i]$$

- <61>
- <62> F_m^i 은 노드 m의 플로우 i의 HOL(Head Of Line) 패킷의 가상 종료시간을 나타낸다.
- <63> 도 2는 채널 유용성을 최대화하기 위한 방법을 나타내는 순서도를 나타낸다.
- <64> 먼저, WFQdp 의해서 패킷 스케줄링을 시작하고(S200), 채널이 휴면(idle)상태인지 여부를 확인한다(S210).
- <65> 상기 확인 결과 채널이 휴면(idle)상태이면 임의의 시간 t 에서 RTS/CTS 간 데이터 교환이 가능하지 여부를 확인한다(S220).
- <66> 상기 확인 결과 교환이 가능하면 백오프 시간을 계산하고(S230) 교환이 불가능하면 서비스 태그의 차이값이 임계값을 넘어섰는지 여부를 확인한다(S221). 상기 확인 결과 임계값을 넘어섰으면 백오프 시간을 계산하고(S230), 임계값 보다 작으면 다른 플로우의 패킷을 스케줄링하고(S222) 다시 처음으로 피드백 하여 WFQ에 의해서 패킷 스케줄링을 수행한다(S200).
- <67> 백오프 시간 계산 후 백오프 타이머를 시작하고 전송한다(S240).

- <68> 도 3은 본 발명의 일실시예로서 시간에 따른 채널 할당을 나타낸 도면이다.
- <69> 세 개의 전송 노드가 있고, 각 플로우의 초기 시작 시간은 $f1=0$, $f2=0$, $f3=10$, $f4=10$, $f5=0$, $f6=10$ 이라 가정하고 네트워크 토폴로지는 그림 4와 같다. 6개의 플로우 모두 같은 패킷 크기와 가중치를 갖는다. 각 패킷의 전송 시간은 백오프 시간보다 훨씬 큰 100이라 가정한다. 제안된 기법은 다음과 같이 동작한다.
- <70> 1) 시간 0에서, 노드 2는 WFQ 스케줄링에 의해서 전송할 플로우로 백오프 값 0을 가지는 $f1$ 을 선택한다. 시간 1에서, 플로우 $f3$ 는 채널이 사용 중임을 감지하고, 채널이 휴면(idle) 상태가 될 때까지 기다린다.
- <71> 2) 시간 100에서, 플로우 $f1$ 과 $f5$ 는 전송을 끝마친다. 플로우 $f3$ 와 $f6$ 는 백오프 시간을 10으로 설정하고, 채널이 휴면(idle) 상태인 것이 감지되면, 백오프 타이머를 시작한다. 시간 110에서, $f3$ 와 $f6$ 는 전송을 시작한다.
- <72> 3) 시간 210에서, 플로우 $f3$ 와 $f6$ 는 전송을 끝마친다. 노드 6은 이미 두 개의 패킷을 전송했기 때문에, 플로우 $f4$ 는 백오프 시간을 10으로 설정하고, 플로우 $f5$ 는 20으로 설정한다. $f2$ 에 대한 백오프 시간은 단지 5가 남았기 때문에, $f2$ 와 $f5$ 는 백오프 타이머를 시작한다. $f2$ 는 시간 215에서 전송을 시작하고 시간 315에 전송을 끝마친다. $f5$ 는 시간 230에서 전송을 시작하고 시간 330에 전송을 끝마친다.
- <73> 4) 시간 315에서 $f1$ 은 이미 노드 2가 2개의 패킷을 전송했기 때문에, 20으로 백오프 시간을 설정한다. $f4$ 에 대한 백오프 시간은 5가 남았기 때문에 시간 335에서 $f1$ 과 $f4$ 가 전송을 시작한다.
- <74> 5) 시간 435에서, $f1$ 과 $f4$ 는 전송을 끝마친다. 그 후에 $f1$ 은 이미 3개의 패킷을 전송했고, $f3$ 는 단지 두 개의 패킷만 전송했기 때문에, $f1$ 은 백오프 시간으로 15로 설정하고 $f3$ 은 10으로 설정한다.
- <75> 6) 시간 445에서 $f3$ 는 전송을 시작하고, 시간 455에서 $f6$ 는 전송을 시작한다.
- <76> 이와 같은 유사한 절차를 통해서, 본 발명은 좋은 지역적 공정성과 전역적 공정성을 달성할 수 있다.

<77> 4. 패킷 스케줄링

<78> 각 노드는 SI_m , $minNSI_m$ 값과 {송신기, 수신기, Flow ID, , 수신시간}을 포함하는 테이블을 유지한다.

<79> 도 1을 예로 들어 설명하면, 노드 2는 $f1$, $f2$ 및 $f3$ 에 대한 정보를 얻을 수 있다. 왜냐하면, $f1$, $f2$ 는 노드 2가 전송하려는 플로우이기 때문에 정보를 가지고 있을 수 있고 $f3$ 는 노드 4가 $f3$ 을 전송할 때 노드 3에서 CTS 프레임 안에 $f3$ 에 대한 플로우에 대한 정보를 노드 4에게 전송하기 때문에 노드 2가 $f3$ 에 대한 정보를 엿들을 수 있다. 그래서 노드 2에서 $f3$ 에 대한 정보를 테이블에 다음과 같이 저장한다.

표 1

flow ID	SI_m	$minNSI_m$	송신측	수신측	수신된 시간
$f1$	노드2	노드1
$f1$	노드2	노드1
$f2$	노드2	노드3
$f2$	노드2	노드3
$f3$	노드4	노드3
$f3$	노드4	노드3

<81> 상기 표에서 하나의 플로우 ID에 대해서 2개의 항목들이 있는 이유는 이전에 받은 값과 현재 받은 값을 비교하기 위해서이다.

<82> 송신 노드가 SI_m 값의 전송이 필요하다고 결정하면, DATA 프레임의 비트를 1로 설정하고 SI_m 값을 DATA 프레임에 추가하여 전송한다. 모든 이웃 노드는 SI_m 을 수신하고, $minNSI_m$ 값을 업데이트한다. 또한, 수신 노드는 SI_m 를 ACK 프레임에 추가하고 비트를 1로 설정하여 전송한다. 따라서 언제든지 노드들은 패킷을 전송하기 전에 자신의 테이블에 저장된 SI_m 값과 $minNSI_m$ 값과의 차이 값을 계산한 후, 백오프 시간을 결정하고, 백오프가 만료되면 채널에 접근한다. 전체적인 스케줄링 알고리즘은 알고리즘 3과 같다.

Algorithm 3 패킷 스케줄링 알고리즘

Init:

For node m , let the set neighbors of m be \mathfrak{R} .
 $\mathfrak{R} = \{N_n \mid \text{neighbor nodes of node } m\}$

Procedure:

- (a) Node m uses algorithm WFQ to schedule outgoing packets.
- (b) Node m counts each flow service and calculate the flow's SI_m^{SI} .
- (c) Calculate the SI_m in periodic time t .
- (d) If $\left\{ \left| \frac{SI_m(t+\Delta t) - SI_m(t)}{SI_m(t)} \right| \geq \theta \right.$ or (last transmitted expired T)
 or (new node joins and content the medium) } then
 Sender transmits the SI_m by DATA MAC header.
 Receiver transmits the same SI_m by ACK MAC header.
- (e) If $\forall N_n \in \mathfrak{R}$, then update the $minNSI$ with received SI_m .

<83>

<84>

알고리즘 3에서 초기에 노드 m 의 이웃 노드들의 집합을 R 이라 하면, 노드 m 은 WFQ(Weighted Fair Queueing)에 의해서 자신에게 들어오는 패킷들의 서비스 순서를 스케줄링한다. 그리고 노드 m 은 전송한 플로우의 서비스 양을 계산하고, 각 플로우의 SI_m^i 를 계산한다. 또한, 주기적으로 SI_m 을 계산한다. 만약, 주기적으로 계산할

$$\left| \frac{SI_m(t+\Delta t) - SI_m(t)}{SI_m(t)} \right| \geq \theta$$

때, 이거나, 새로운 노드가 네트워크에 들어온 것을 확인하면, 그 때 노드 m 은 SI_m 을 DATA MAC 프레임의 헤더에 포함시켜 전송하고, 이 DATA MAC 프레임을 수신한 수신 노드는 ACK MAC 프레임의 헤더에 같은 SI_m 을 포함하여 전송한다. 따라서 노드 m 의 이웃 노드들의 집합 R 에 포함되어 있는 노드들은 SI_m 을 수신함으로써 $minNSI$ 를 업데이트한다.

<85>

5. 구현 시 고려사항

<86>

본 발명을 구현하기 위해서 다음과 같은 사항들을 고려해야만 한다.

<87>

먼저, 각 플로우의 백오프 타이머는 SI_m 과 $minNSI_m$ 사이에서 계산된 차이 값인 DI_m^i 에 따라 조정되어야 한다. 따라서, 각 플로우의 SI_m 은 주기적으로 <수학식 7>과 같이 계산된다.

수학식 7

<88>

$$SI_m^i(t + \Delta t) = \alpha \times SI_m^i(t) + (1 - \alpha) \times SI_m^i(t, t + \Delta t)$$

<89>

여기서, $SI_m^i(t+\Delta t)$ 는 시간 $(t+\Delta t)$ 에서 SI_m^i 값을 나타내고, α 는 새로 결정된 SI_m^i 안의 이전 값 SI_m^i 의 가중치를 나타낸다. Δt 는 SI_m^i 를 계산하는 동안의 시간 간격을 나타낸다. $SI_m^i(t, t+\Delta)$ 는 시간 $(t, t+\Delta)$ 동안의 SI_m^i 을 나타내고 <수학식 8>과 같이 계산된다.

수학식 8

$$SI_m^i(t, t + \Delta t) = \frac{W_i(t, t + \Delta t)}{\emptyset_i \times \Delta t}$$

<90>

상기 <수학식 8>에서 $W_i(t, t + \Delta t)$ 는 Δt 시간 동안에 플로우 i 가 받은 서비스 양을 나타낸다.

<91>

<수학식 7>과 <수학식 8>에 따라서 각 플로우의 SI_m^i 과 각 노드의 SI_m 는 얻어질 수 있다.

<92>

또한, 본 발명은 알고리즘의 오버헤드를 줄이기 위해서, 각 노드 m 은 프레임 헤더 대신에 데이터 패킷의 한 부분으로 SI_m 을 모든 이웃 노드들에게 전송한다. 다른 방법으로, 만약 프레임 헤더의 한 부분으로 SI_m 이 전송된다면, 각 프레임 헤더에 필드가 추가되어야 한다. 오버헤드를 최소화하기 위해서 송신 노드는 다음의 기준에 준하여 SI_m 의 전송 여부를 결정할 수 있다.

<93>

기준 1) 이전에 전송된 값과 현재 값과의 편차가 임계값 θ 를 초과하면, SI_m 은 이웃 노드들에게 전송된다. 예를 들어 편차가 5%이면 SI_m 은 전송된다.

<94>

기준 2) 만약, 새로운 노드가 매체에 접근한다면, 이웃 노드들에게 SI_m 을 전송한다.

<95>

기준 3) 만약, 시간 기간 T 동안 노드가 SI_m 을 전송하지 않았다면, SI_m 은 이웃 노드들에게 전송된다.

<96>

IEEE 802.11표준의 현재 프레임 형식에서 데이터 프레임의 프레임 제어 필드 중 하위 타입 필드는 4비트로 구성되어 있다. 본 발명은 네 번째 SI_m 비트를 이 프레임에 포함되어 있는지에 대한 여부를 나타내기 위해서 사용한다. 만약, 그 비트가 1로 설정되어 있으면, 프레임에 SI_m 이 포함되어 있는 것을 나타낸다. 수신된 SI_m 이 $\min NSI_m$ 보다 작다면 $\min NSI_m$ 를 수신한 SI_m 값으로 대체하여 업데이트한다. 반면에 그 비트가 0으로 설정되어 있으면, 수신 노드를 제외한 모든 이웃 노드들은 그 프레임을 무시한다. 값을 설정하기 위한 2 바이트가 데이터 프레임에 추가된다.

<97>

플로우의 수신 노드 주변에는 몇몇의 이웃 노드들이 존재할 수 있기 때문에, ACK 프레임 안에 한 비트가 할당되고, DATA 프레임과 유사하게, ACK 프레임 안에 SI_m 을 위한 2 바이트가 추가된다. 따라서 ACK 프레임은 업데이트된 정보를 전송할 수 있다.

<98>

도 1에서 노드 4(306)는 ACK 프레임을 통해서 노드 2(302)의 값을 얻어 올 수 있다. ACK 프레임 안에 존재하는 SI_m 은 이전에 수신한 DATA 프레임 안에 포함된 과 같다. 따라서, 각 노드의 SI_m 은 두 홉 거리에 있는 노드들에게 전달될 수 있다. ACK 프레임에 SI_m 을 추가하여 전송하는 목적은 전역적 공정성을 달성하기 위한 것이다.

<99>

도 4는 위 절차들을 나타낸다.

<100>

도 4에서 노드 m 이 노드 n 에게 데이터를 전송할 때, DATA 프레임에 SI_m 을 추가하여 전송한다. 그러면, 노드 n 과 노드 m 의 인접 노드 p 와 노드 q 는 노드 m 의 SI_m 을 수신한다. 그리고 노드 n, p, q 는 각각 $\min NSI_n$ 값, $\min NSI_p$ 값, $\min NSI_q$ 값을 업데이트한다. 노드 n 은 수신한 SI_m 을 ACK 프레임에 포함하여 전송한다. 따라서 노드 o 는 노드 m 의 SI_m 을 수신하여 $\min NSI_o$ 값을 업데이트한다.

<101>

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

발명의 효과

<102>

본 발명에 따르면, 에드혹 네트워크 환경에서 지역적 공정성과 전역적 공정성을 제공함으로써, 각 플로우의 공정성을 크게 향상시킨다.

<103>

또한, 본 발명에 따르면, time-stamp 방식의 WFQ와 같은 패킷 스케줄링을 에드혹 패킷 스케줄링에 적용했을 때,

교환해야 하는 서비스 태그 정보가 필요하지 않기 때문에, 네트워크 처리량의 감소를 최소화할 수 있다.

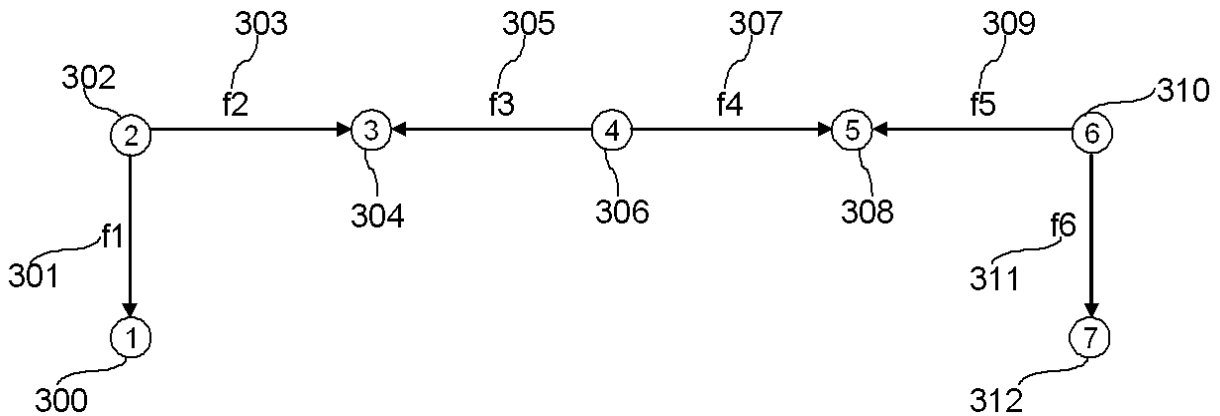
- <104> 또한, 본 발명에 따르면, time-stamp 방식의 WFQ와 같은 패킷 스케줄링을 에드혹 패킷 스케줄링에 적용했을 때, 가상 시간 초기화 같은 문제를 발생시키지 않는다.
- <105> 또한, 본 발명에 따르면, 전역적 공정성을 달성하기 위하여 노드들 간에 전송해야 하는 각 노드의 서비스 색인의 전송을 위하여, 기존의 IEEE 802.11 표준의 적은 수정만으로 전송할 수 있는 방법을 제공함으로써, 추가적인 오버헤드 발생을 최소화한다.
- <106> 아울러 본 발명을 이용하면, 무선 에드혹 네트워크의 채널 유용성을 최대화 하는 방법을 제공함으로써, 에드혹 네트워크의 전체 처리량을 향상할 수 있다.

도면의 간단한 설명

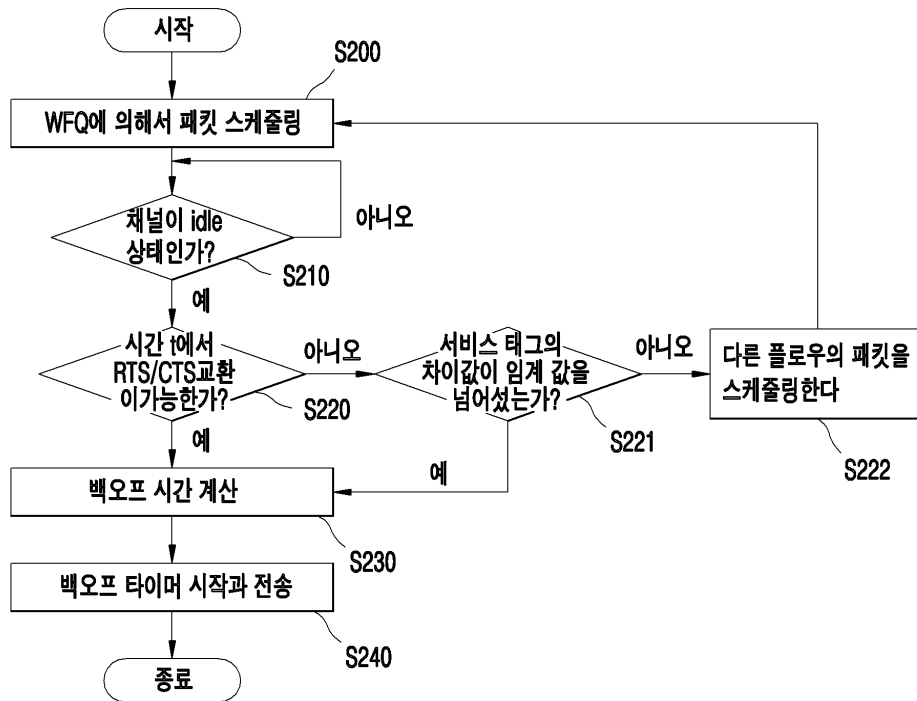
- <1> 도 1은 본 발명에 따른 채널 유용성 최대화 방법을 설명하기 위한 네트워크의 예시도.
- <2> 도 2는 본 발명에 따른 채널 유용성 최대화 방법을 나타내는 순서도.
- <3> 도 3은 본 발명에 따른 채널 유용성 최대화 방법을 적용하였을 때의 백오프타임 및 전송 방법에 관한 일예시도.
- <4> 도 4는 본 발명에 따른 서비스 색인의 교환과 업데이트 방법을 나타내는 도면.
- <5> 삭제
- <6> 삭제

도면

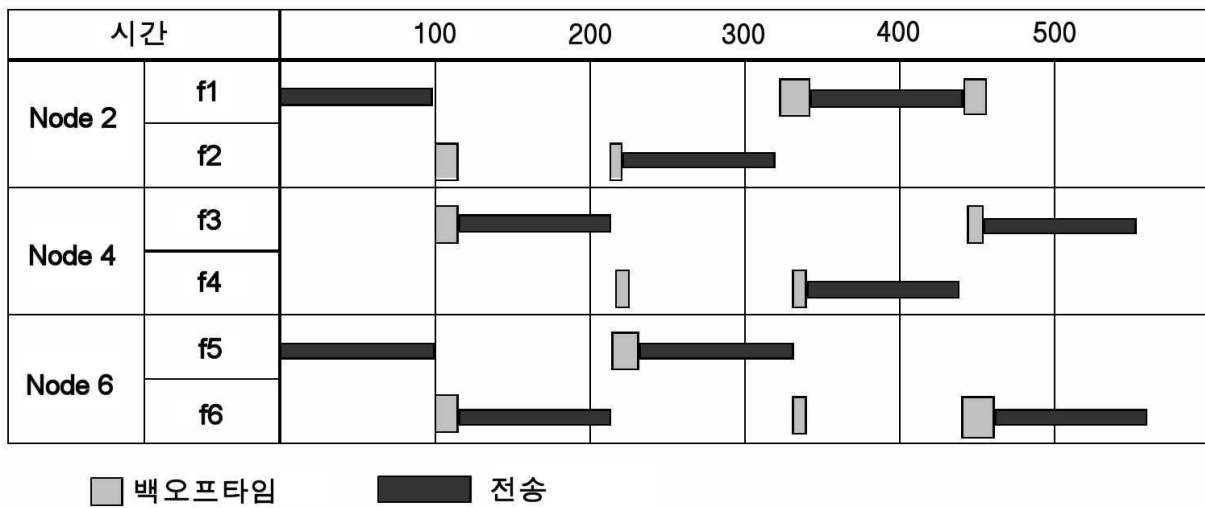
도면1



도면2



도면3



도면4

