



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2015년09월03일  
(11) 등록번호 10-1549794  
(24) 등록일자 2015년08월27일

(51) 국제특허분류(Int. Cl.)  
H04L 12/861 (2013.01) H04L 12/811 (2013.01)  
(21) 출원번호 10-2014-0169856  
(22) 출원일자 2014년12월01일  
심사청구일자 2014년12월01일  
(56) 선행기술조사문헌  
US7215641 B1  
KR1020050076158 A  
KR100654429 B1  
KR100121428 B1

(73) 특허권자  
한국과학기술원  
대전광역시 유성구 대학로 291(구성동)  
(72) 발명자  
박홍식  
대전 유성구 대학로 291  
김명훈  
대전 유성구 대학로 291  
고남석  
대전광역시 유성구 장대로 한일유엔아이아파트  
101동 202호  
(74) 대리인  
양성보

전체 청구항 수 : 총 18 항

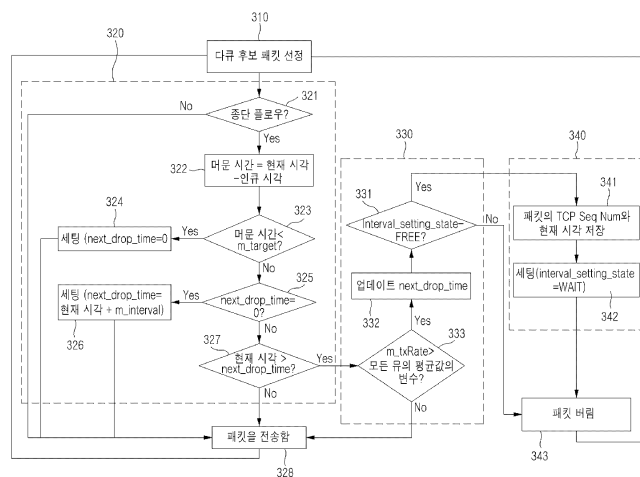
심사관 : 이형일

(54) 발명의 명칭 자동 버퍼 관리 방법 및 장치

(57) 요약

자동 버퍼 관리 방법 및 장치가 제시된다. 본 발명에서 제안하는 자동 버퍼 관리의 디큐 방법은 디큐 후보 패킷을 선택하는 단계, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교하는 단계, 현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인하는 단계, 상기 인터벌 설정 상태가 프리인 경우, 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스와 현재 시각을 저장하고 상기 인터벌 설정 상태를 대기 상태로 설정한 후, 상기 디큐 후보 패킷을 드롭하고 다음 디큐 후보 패킷을 선택하는 단계를 포함할 수 있다.

대표도



이 발명을 지원한 국가연구개발사업

과제고유번호 13-911-04-001

부처명 미래창조과학부

연구관리전문기관 방송통신전파진흥원

연구사업명 방송통신인프라원천기술개발사업

연구과제명 생체모방 알고리즘(Bio-inspired Algorithm)을 활용한 통신기술 연구

기 여 율 1/1

주관기관 한국과학기술원

연구기간 2013.04.01 ~ 2018.02.28

---

**특허청구의 범위**

**청구항 1**

자동 버퍼 관리의 인큐 방법에 있어서,

현재 시각으로 패킷에 태그 작업을 수행하고 상기 패킷을 헤시 기능으로 분류하여 큐를 할당하는 단계;

상기 할당된 큐가 중단 플로우인지 확인하고, 상기 큐가 중단 플로우일 경우 인터벌 설정 상태를 확인하는 단계;

상기 인터벌 설정 상태가 대기 상태인 경우, 현재 전송 제어 프로토콜 시퀀스와 저장된 전송 제어 프로토콜 시퀀스를 비교하는 단계; 및

상기 현재 전송 제어 프로토콜 시퀀스와 상기 저장된 전송 제어 프로토콜 시퀀스가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료하는 단계

를 포함하는 자동 버퍼 관리의 인큐 방법.

**청구항 2**

제1항에 있어서,

상기 현재 시각으로 패킷에 태그 작업을 수행하고 상기 패킷을 헤시 기능으로 분류하여 큐를 할당하는 단계는,

상기 패킷을 받은 시간을 저장하고, 헤시 기능으로 분류하여 새로운 플로우일 경우 새로운 큐를 할당하고 기존에 존재하던 플로우일 경우 기존에 할당되었던 큐를 할당하는 것을 특징으로 하는 자동 버퍼 관리의 인큐 방법.

**청구항 3**

제1항에 있어서,

상기 현재 전송 제어 프로토콜 시퀀스와 상기 저장된 전송 제어 프로토콜 시퀀스가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료하는 단계는,

상기 현재 전송 제어 프로토콜 시퀀스와 상기 저장된 전송 제어 프로토콜 시퀀스를 비교할 때까지 걸린 시간을 구하는 단계;

상기 걸린 시간의 변수값을 구하여 설정하는 단계; 및

상기 인터벌 설정 상태를 프리로 설정함으로써 상기 인큐 과정을 종료하는 단계

를 포함하는 자동 버퍼 관리의 인큐 방법.

**청구항 4**

제3항에 있어서,

상기 걸린 시간의 변수값을 구하여 설정하는 단계는,

상기 걸린 시간의 변수값이 미리 설정된 시간 이상일 경우 제거하거나, 또는 상기 걸린 시간들의 이동평균값을 계산하여 결정하는 것을 특징으로 하는 자동 버퍼 관리의 인큐 방법.

**청구항 5**

제1항에 있어서,

상기 할당된 큐가 중단 플로우가 아닌 경우 상기 인큐 과정을 종료하고,

상기 인터벌 설정 상태가 프리 상태인 경우 상기 인큐 과정을 종료하고,

상기 현재 전송 제어 프로토콜 시퀀스와 상기 저장된 전송 제어 프로토콜 시퀀스가 같지 않은 경우 상기 인큐 과정을 종료하는 것을 특징으로 하는 자동 버퍼 관리의 인큐 방법.

**청구항 6**

자동 버퍼 관리의 디큐 방법에 있어서,

디큐 후보 패킷을 선택하는 단계;

상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교하는 단계;

현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인하는 단계; 및

상기 인터벌 설정 상태가 프리인 경우, 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스와 현재 시각을 저장하고 상기 인터벌 설정 상태를 대기 상태로 설정한 후, 상기 디큐 후보 패킷을 드롭하고 다음 디큐 후보 패킷을 선택하는 단계

를 포함하는 자동 버퍼 관리의 디큐 방법.

**청구항 7**

제6항에 있어서,

상기 디큐 후보 패킷을 선택하는 단계는,

상기 디큐 후보 패킷의 큐에 패킷이 존재할 경우 상기 디큐 후보 패킷을 선택하는 단계를 반복적으로 수행하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 8**

제6항에 있어서,

상기 선택된 디큐 후보 패킷의 큐가 중단 플로우가 아닌 경우, 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 9**

제6항에 있어서,

상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교하는 단계는,

상기 머문 시간이 상기 미리 정해진 타겟 시간보다 크고, 상기 다음 드롭 시간이 0 이 아니고, 상기 현재 시각이 상기 다음 드롭 시간보다 큰 경우 상기 디큐 후보 패킷을 드롭하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 10**

제6항에 있어서,

상기 머문 시간이 상기 미리 정해진 타겟 시간보다 작으면 상기 다음 드롭 시간을 0으로 설정하고 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 11**

제6항에 있어서,

상기 머문 시간이 상기 미리 정해진 타겟 시간보다 크고, 상기 다음 드롭 시간이 0 인 경우, 상기 다음 드롭 시간을 업데이트한 후 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 12**

제6항에 있어서,

상기 머문 시간이 상기 미리 정해진 타겟 시간보다 크고, 상기 다음 드롭 시간이 0 이 아니고, 상기 현재 시각이 상기 다음 드롭 시간보다 작은 경우 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 13**

제6항에 있어서,

상기 현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인하는 단계는,

상기 큐의 전송 속도가 상기 모든 큐의 전송속도의 평균값의 변수보다 작은 경우 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 14**

제6항에 있어서,

상기 큐의 전송 속도가 상기 모든 큐의 전송속도의 평균값의 변수보다 큰 경우 상기 디큐 후보 패킷을 드롭하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 15**

제6항에 있어서,

상기 다음 드롭 시간은 상기 활성 인터벌과 상기 현재 시각을 이용하여 결정하고, 상기 다음 드롭 시간과 상기 활성 인터벌을 결정한 후 드롭 카운트를 1만큼 증가 시키는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 16**

제6항에 있어서,

상기 인터벌 설정 상태가 대기 상태일 경우, 상기 디큐 후보 패킷을 드롭하고 다음 디큐 후보 패킷을 선택하는 것을 특징으로 하는 자동 버퍼 관리의 디큐 방법.

**청구항 17**

자동 버퍼 관리의 인큐 장치에 있어서,

현재 시각으로 패킷에 태그 작업을 수행하고 상기 패킷을 헤시 기능으로 분류하여 큐를 할당하는 큐 할당부; 및 상기 큐가 중단 플로우이고, 인터벌 설정 상태가 대기 상태이고, 현재 전송 제어 프로토콜 시퀀스와 저장된 전송 제어 프로토콜 시퀀스가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료하는 인큐 수행부

를 포함하는 자동 버퍼 관리의 인큐 장치.

**청구항 18**

자동 버퍼 관리의 디큐 장치에 있어서,

디큐 후보 패킷을 선택하고, 상기 디큐 후보 패킷을 전송하고, 또는 드롭 하는 처리부;

상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교하는 패킷 비교부;

현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인하는 공평성 확인부; 및

상기 인터벌 설정 상태가 프리인 경우, 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스와 현재 시각을 저장하고 상기 인터벌 설정 상태를 대기 상태로 설정하는 인터벌 설정부

를 포함하는 자동 버퍼 관리의 디큐 장치.

**명세서**

**기술분야**

[0001] 본 발명은 지연 성능을 개선함과 동시에 플로우 간 전송속도 공평성을 향상시키기 위한 자동 버퍼 관리 방법 및 장치에 관한 것이다.

**배경기술**

[0002] 도 1은 일반적인 방식으로 버퍼를 관리하는 통신 시스템의 개략적인 구성도이다. 도 1을 참조하면, 통신 시스템은 사용자 단말(110)과, 기지국(120)과, 게이트 웨이 노드(130) 및 인터넷(140)을 포함한다. 일 예로서, 상기 통신 시스템이 차세대 무선 통신시스템인 진화된 패킷 시스템(EPC: Evolved Packet System)일 경우, 상기 기지국(120)은 진화된 노드비(eNB: Evolved Node B)에 대응하고, 상기 게이트 웨이 노드(130)는 진화된 패킷 코어(EPC: Evolved Packet Core)에 대응한다고 간주할 수 있다. 상기 진화된 패킷 코어 시스템은 회로 스위치된 모델(circuit-switched model)에 근간을 둔 셀룰라 시스템(cellular system)들과는 달리 패킷 스위치된 서비스(packet-switched service) 제공을 목표로 고안되었다. 또한, 상기 진화된 패킷 코어 시스템은 100 Mbps 이상의 고속의 다운링크 송신을 위한 전송율과, 수십 Mbps 에 이르는 업링크 송신을 위한 전송율을 제공한다.

[0003] 상기한 진화된 패킷 코어 시스템에서의 무선 구간 즉, 상기 사용자 단말(110)과 상기 기지국(120)간의 링크(link)는, 상기 사용자 단말(110)의 이동성과, 신호 세기 및 간섭 영향 등으로 인한 채널 상태에 따라 상기 링크의 전송율이 최소값과 최대값 사이에서 크게 그리고 자주 변화할 수 있다. 또한, 상기 채널 상태의 열화는 상기 기지국(120) 내의 버퍼 오버 플로우(overflow)로 인한 패킷 로스(packet loss)를 증가시킴으로써, 어플리케이션 성능(application performance)을 떨어뜨리는 문제점이 발생한다.

[0004] 상기한 버퍼 오버 플로우를 막기 위해서, 상기 기지국(120) 내의 버퍼를 관리하고 상기 기지국(120)과 상기 게이트 웨이 노드(130) 간의 플로우 제어를 포함하는 프레임워크(framework)가 종종 요구된다. 일반적인 프레임워크에서, 상기 기지국(120)은 상기 기지국(120) 내 버퍼의 큐 길이(queue length)를 주기적으로 모니터링하고, 상기 모니터링된 큐 길이와 미리 결정되어 있는 큐 길이의 임계값인 상위 바운드(upper bound)와 하위 바운드(lower bound) 각각을 비교한다. 상기 비교 결과 상기 모니터링된 큐 길이가 상기 상위 바운드 이상일 경우, 상기 기지국(120)은 상기 기지국(120) 내의 버퍼에서 오버 플로우가 발생하였음을 나타내는 온(ON) 메시지를 상기 게이트 웨이 노드(130)에게 송신한다. 상기 비교 결과 상기 모니터링된 큐 길이가 상기 하위 바운드 이하일 경우, 상기 기지국(120)은 상기 기지국(120) 내의 버퍼에서 오버 플로우가 발생하지 않았음을 나타내는 오프(OFF) 메시지를 상기 게이트 웨이 노드(130)에게 송신한다. 그러면, 상기 게이트 웨이 노드(130)는 상기 기지국(120)으로부터 수신된 온 또는 오프 메시지에 대한 응답 동작으로, 상기 기지국(120)으로 송신되는 트래픽의 양을 조절한다. 즉, 상기 게이트 웨이 노드(130)는 상기 기지국(120)으로의 트래픽 송신을 "중지(Stop)하거나" 또는 상기 기지국(120)과 중단된 트래픽 송신을 "재시작(Resume)"한다. 상기한 바와 같은 방식으로 수행되는 일반적인 프레임워크는 기지국의 계산적인 오버 헤드(overhead)를 크게 증가시키지 않으면서 최소한의 플로우 제어 효과를 가져온다.

[0005] 일반적으로, 기지국 내의 버퍼에 대한 입력 트래픽은, 네트워크 혼잡(network congestion) 상태를 검출하기 위해서 미리 결정되어 있는, 고정적인 값을 갖는 큐 길이의 임계값을 기반으로 제어된다. 이때, 상기 임계값은 상위 바운드와 하위 바운드를 포함하며, 상기 네트워크 혼잡은 상기 버퍼에 오버 플로우가 발생하였는 지 여부를 나타낸다. 그러나, 실제 통신 시스템에서는 네트워크 유동성 및 트래픽 상태가ダイナミック하게 변화하고, 그 특성을 예측할 수 없다. 따라서 실제 통신 시스템에서는 상기한 바와 같이 고정된 임계값만을 사용하여 네트워크 혼잡 상태를 검출하고, 플로우를 제어할 경우 다음과 같은 문제점이 발생한다.

[0006] 첫째, 상기 기지국 내의 버퍼에 대한 입력 및 출력 트래픽의 특성 예를 들어, 전송속도와, 상기 전송속도의 분포 등을 모르는 상황에서 상기 큐 길이의 임계값을 적정값으로 결정하는 것이 어렵다는 문제점이 있다.

[0007] 둘째, 네트워크 유동성들과 채널 및 트래픽 상태의 변화가 발생할 때마다 그 때 그때 대응할 수 없기 때문에, 네트워크 혼잡 상태의 검출 시 오류가 발생함으로써, 로스(loss)되는 패킷들이 증가하고, 버퍼의 언더-런

(under-run) 현상 발생 등으로 인하여 상기 버퍼의 처리량 저하를 가져오는 문제점이 있다. 예를 들어, 상기 상위 바운드가 상기 버퍼에서의 현재 트래픽 상태를 커버하기에 너무 작은 값으로 설정되어 있으면, 실제로는 네트워크 혼잡 상태가 아닌 경우를 네트워크 혼잡 상태로 검출하게 됨으로써, 불필요한 전송을 감소를 초래할 수 있다. 이로 인하여, 상기 버퍼의 언더-런 현상이 발생할 가능성이 높고, 상기 버퍼의 처리량 저하를 가져온다.

[0008] 반면에, 상기 상위 바운드가 너무 큰 값으로 설정되어 있으면, 실제 네트워크 혼잡 상태가 발생한 이후의 시점에서 네트워크 혼잡 상태를 검출하게 됨으로 인하여 적절한 시기에 플로우를 제어 할 수 없게 된다. 이로 인해서, 패킷 로스가 증가하고 어플리케이션의 성능 열화를 초래하는 문제점이 있다.

### 발명의 내용

#### 해결하려는 과제

[0009] 본 발명이 이루고자 하는 기술적 과제는 새로운 알고리즘인 ABM(Automatic buffer Management)을 통해 지연 성능을 개선함과 동시에 플로우 간 전송속도 공평성을 향상시키기 위한 자동 버퍼 관리 방법 및 장치를 제공하는 데 있다.

#### 과제의 해결 수단

[0010] 일 측면에 있어서, 본 발명에서 제안하는 자동 버퍼 관리의 인큐 방법은 현재 시각으로 패킷에 태그 작업을 수행하고 상기 패킷을 헤시 기능으로 분류하여 큐를 할당하는 단계, 상기 할당된 큐가 중단 플로우인지 확인하고, 상기 큐가 중단 플로우일 경우 인터벌 설정 상태를 확인하는 단계, 상기 인터벌 설정 상태가 대기 상태인 경우, 현재 전송 제어 프로토콜 시퀀스와 저장된 전송 제어 프로토콜 시퀀스를 비교하는 단계, 상기 현재 전송 제어 프로토콜 시퀀스와 상기 저장된 전송 제어 프로토콜 시퀀스가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료하는 단계를 포함할 수 있다.

[0011] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 자동 버퍼 관리의 디큐 방법은 디큐 후보 패킷을 선택하는 단계, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교하는 단계, 현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인하는 단계, 상기 인터벌 설정 상태가 프리인 경우, 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스와 현재 시각을 저장하고 상기 인터벌 설정 상태를 대기 상태로 설정한 후, 상기 디큐 후보 패킷을 드롭하고 다음 디큐 후보 패킷을 선택하는 단계를 포함할 수 있다.

[0012] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 자동 버퍼 관리의 인큐 장치 현재 시각으로 패킷에 태그 작업을 수행하고 상기 패킷을 헤시 기능으로 분류하여 큐를 할당하는 큐 할당부, 상기 큐가 중단 플로우이고, 인터벌 설정 상태가 대기 상태이고, 현재 전송 제어 프로토콜 시퀀스와 저장된 전송 제어 프로토콜 시퀀스가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료하는 인큐 수행부를 포함할 수 있다.

[0013] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 자동 버퍼 관리의 디큐 장치는 디큐 후보 패킷을 선택하고, 상기 디큐 후보 패킷을 전송하고, 또는 드롭 하는 처리부, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교하는 패킷 비교부, 현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인하는 공평성 확인부, 상기 인터벌 설정 상태가 프리인 경우, 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스와 현재 시각을 저장하고 상기 인터벌 설정 상태를 대기 상태로 설정하는 인터벌 설정부를 포함할 수 있다.

#### 발명의 효과

[0014] 본 발명의 실시예들에 따르면 트레이드오프(Tradeoff) 관계에 있는 딜레이 성능과 전송속도 공평성 성능을 높일 수 있고, 네트워크 중단간 딜레이를 감소 시킬 수 있다. 또한 네트워크 중단간 전송속도 공평성을 향상시킬 수 있고, 네트워크 장비의 이용률 (Utilization)을 높일 수 있다. 그리고, 보다 적은 비용으로 더 많은 사용자 트

래픽을 수용할 수 있고, 사용자가 별도의 파라미터 셋팅을 하지 않아도 된다.

**도면의 간단한 설명**

- [0015] 도 1은 종래 기술에 따른 일반적인 방식으로 버퍼를 관리하는 통신 시스템의 개략적인 구성도이다.
- 도 2는 본 발명의 일 실시예에 따른 자동 버퍼 관리의 인큐 방법을 설명하기 위한 흐름도이다.
- 도 3은 본 발명의 일 실시예에 따른 자동 버퍼 관리의 디큐 방법을 설명하기 위한 흐름도이다.
- 도 4는 본 발명의 일 실시예에 따른 자동 버퍼 관리의 인큐 장치의 구성을 나타내는 도면이다.
- 도 5는 본 발명의 일 실시예에 따른 자동 버퍼 관리의 디큐 장치의 구성을 나타내는 도면이다.

**발명을 실시하기 위한 구체적인 내용**

- [0016] 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0017] 도 2는 본 발명의 일 실시예에 따른 자동 버퍼 관리의 인큐 방법을 설명하기 위한 흐름도이다.
- [0018] 자동 버퍼 관리의 인큐(Enqueue) 과정은 QueueAllocation 블록과 CheckTCPSeqNForInterval 블록을 이용하여 수행될 수 있다. 그리고, 현재 시각으로 패킷에 태그 작업을 수행하고 상기 패킷을 해시 기능으로 분류하여 큐를 할당하는 단계(210), 상기 할당된 큐가 중단 플로우인지 확인하고, 상기 큐가 중단 플로우일 경우 인터벌 설정 상태를 확인하는 단계(220), 상기 인터벌 설정 상태가 대기 상태인 경우, 현재 전송 제어 프로토콜 시퀀스와 저장된 전송 제어 프로토콜 시퀀스를 비교하는 단계(230), 상기 현재 전송 제어 프로토콜 시퀀스와 상기 저장된 전송 제어 프로토콜 시퀀스가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료하는 단계(240)을 포함할 수 있다.
- [0019] 단계(210)에서, 현재 시각으로 패킷에 태그 작업을 수행(211)하고 상기 패킷을 해시 기능으로 분류(212)하여 큐를 할당(213)할 수 있다. 이때, 패킷을 받은 시간을 저장하고, 해시 기능으로 분류하여 새로운 플로우일 경우 새로운 큐를 할당하고 기존에 존재하던 플로우일 경우 기존에 할당되었던 큐를 할당할 수 있다. 다시 말해, 패킷을 받은 시간을 저장하고, 이를 인큐 타임(Enqueue Time)이라고 할 수 있다. 그리고, 5개 정보(IP Source Address, IP Destination Address, Source Port, Destination Port, Protocol)를 사용하여 해시 기능(Hash Function)을 돌린 후 플로우 큐(Flow Queue)를 할당할 수 있다. 이때 플로우가 새로운 플로우일 경우에는 새로운 큐를 할당하고, 기존에 존재하던 플로우일 경우에는 기존에 할당되었던 큐를 할당할 수 있다. 이로써 플로우별 큐 관리를 일관성 있게 수행할 수 있다.
- [0020] 단계(220)에서, 상기 할당된 큐가 중단 플로우(Edge Flow)인지 확인(221)하고, 상기 큐가 중단 플로우일 경우 인터벌 설정 상태(interval\_setting\_state)를 확인(222)할 수 있다.
- [0021] 자기 노드에서 트래픽(Traffic)이 시작된 경우 이 트래픽을 중단 플로우(Edge Flow)라 부른다. 자기 노드(node)에서 시작되지 않은 경우에 자기를 통과하는 트래픽(Traffic)을 통과 플로우(Through Flow)라고 한다. ABM(Automatic Buffer Management)은 중단 플로우(Edge Flow)에 대해서 동작하고, 통과 플로우(Through Flow)에 대해서 드롭테일(DropTail)과 같이 동작한다. 다시 말해, 통과 플로우(Through Flow)에 대해서는 드롭(Drop) 결정 과정, 공정성 확인(Fairness checking) 기능, 그리고 동적 인터벌(Dynamic Interval) 값 조정(디큐 과정에서의 SetIntervalByRtt 블록, 인큐 과정에서의 CheckTCPSeqNForInterval 블록)이 동작하지 않는다. 이는 ABM(Automatic Buffer Management)을 간단한 방식으로 구현 하면서도 성능 개선의 좋은 요소가 된다.
- [0022] 할당된 큐가 중단 플로우(Edge Flow)인지 확인하고, 상기 큐가 중단 플로우일 경우 동적 인터벌(Dynamic Interval) 값 조정의 CheckTCPSeqNForInterval()이 수행될 수 있다. 만약 할당된 큐가 중단 플로우가 아닌 경우에는 상기 인큐 과정을 종료할 수 있다.
- [0023] 만약 인터벌 설정 상태(interval\_setting\_state)가 프리 상태(FREE)인 경우에는 인큐 과정을 종료할 수 있다.
- [0024] 인터벌 설정 상태(interval\_setting\_state)가 대기 상태(WAIT)인 경우, 단계(230)에서 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)를 비교할 수 있다.



- [0025] 다시 말해, 인터벌 설정 상태(interval\_setting\_state) 대기 상태(WAIT)인 경우, 패킷의 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)를 확인하여 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 일치하는지 확인할 수 있다.
- [0026] 그리고, 디큐(Dequeue) 과정에서 이미 드롭(Drop)된 패킷이 존재하는 경우(도 3 참조, 343)에는 인터벌 설정 상태(interval\_setting\_state)가 대기 상태(WAIT)으로 설정(도 3 참조, 342)되어 있게 된다.
- [0027] 만약, 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)가 같지 않은 경우 상기 인큐 과정을 종료할 수 있다.
- [0028] 상기 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)가 같은 경우, 단계(240)에서 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료할 수 있다.
- [0029] 패킷의 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)를 확인하여 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 일치 하는지 확인 한 후, 그때까지의 걸린 시간(m\_Rtt)을 구할 수 있다(241). 걸린 시간(m\_Rtt)은 수학식1을 이용하여 구할 수 있다.
- [0030] 걸린 시간(m\_Rtt) = 현재시각 - 저장된 Drop Time                      수학식1
- [0031] 그리고, m\_Rtt 변수값을 구할 수 있다(242). 예를 들어, 걸린 시간의 변수값이 미리 설정된 시간 이상일 경우, 다시 말해 과도한 수치 인 경우 제거 할 수도 있다. 그리고, m\_Rtt 값들의 이동평균값을 계산하여 결정 할 수도 있다. 그리고 m\_interval을 m\_Rtt 변수값으로 설정할 수 있다.
- [0032] 마지막으로 인터벌 설정 상태(interval\_setting\_state) 상태를 프리 상태(FREE)로 설정하고 인큐(Enqueue) 과정을 종료할 수 있다(243).
- [0033] 도 3은 본 발명의 일 실시예에 따른 자동 버퍼 관리의 디큐 방법을 설명하기 위한 흐름도이다.
- [0034] 자동 버퍼 관리의 디큐(Dequeue) 과정은 ShouldDrop 블록, FairnessChecking 블록, Dynamic Interval 기능의 SetIntervalByRtt 블록, 처리 블록을 이용하여 수행될 수 있다. 그리고, 디큐 후보 패킷을 선택하는 단계(310), 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교하는 단계(320), 현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인하는 단계(330), 상기 인터벌 설정 상태가 프리인 경우, 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스와 현재 시각을 저장하고 상기 인터벌 설정 상태를 대기 상태로 설정한 후, 상기 디큐 후보 패킷을 드롭하고 다음 디큐 후보 패킷을 선택하는 단계(340)를 포함할 수 있다.
- [0035] 단계(310)에서, 디큐 후보 패킷을 선택할 수 있다. 디큐 후보 패킷의 큐에 패킷이 존재할 경우 디큐 후보 패킷을 선택하는 단계를 반복적으로 수행할 수 있다.
- [0036] 단계(320)에서, 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인(321)하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교할 수 있다.
- [0037] 만약, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우가 아닌 경우, 상기 디큐 후보 패킷을 디큐(328)하고 다음 디큐 후보 패킷을 선택(310)할 수 있다.
- [0038] 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 단계(322)에서 머문 시간을 계산할 수 있다. 머문 시간은 수학식2를 이용하여 구할 수 있다.

- [0039]            머문 시간(Sojourn time) = 현재 시각 - 인큐 시간(Enqueue time)       수학식2
  
- [0040]            다음으로, 머문 시간과 미리 정해진 타겟 시간(m\_target) 을 비교할 수 있다(323).
- [0041]            머문 시간이 미리 정해진 타겟 시간(m\_target)보다 작은 경우, 다음 드롭 시간(next\_drop\_time)을 0으로 설정(324)하고 디큐 후보 패킷을 디큐(328)하고 다음 디큐 후보 패킷을 선택(310)할 수 있다.
- [0042]            머문 시간이 미리 정해진 타겟 시간(m\_target)보다 큰 경우, 다음 드롭 시간(next\_drop\_time)을 확인할 수 있다(325).
- [0043]            머문 시간이 상기 미리 정해진 타겟 시간보다 크고, 상기 다음 드롭 시간이 0 인 경우, 상기 다음 드롭 시간(next\_drop\_time)을 m\_interval 값을 이용하여 업데이트한 후(326) 상기 디큐 후보 패킷을 디큐(328)하고 다음 디큐 후보 패킷을 선택(310)할 수 있다. 다음 드롭 시간(next\_drop\_time)은 수학식3을 이용하여 결정할 수 있다.
  
- [0044]            드롭 시간(next\_drop\_time) = 현재 시각 + m\_interval                    수학식3
  
- [0045]            다음 드롭 시간이 0이 아닌 인 경우, 현재 시각과 상기 다음 드롭 시간을 비교할 수 있다(327).
- [0046]            머문 시간이 상기 미리 정해진 타겟 시간보다 크고, 상기 다음 드롭 시간이 0 이 아니고, 상기 현재 시각이 상기 다음 드롭 시간보다 작은 경우 상기 디큐 후보 패킷을 디큐(328)하고 다음 디큐 후보 패킷을 선택(310)할 수 있다.
- [0047]            현재 시각이 상기 다음 드롭 시간(next\_drop\_time)보다 큰 경우, 디큐 후보 패킷을 드롭할 수 있다. 다시 말해, 머문 시간이 미리 정해진 타겟 시간보다 크고, 다음 드롭 시간이 0 이 아니고, 현재 시각이 상기 다음 드롭 시간(next\_drop\_time)보다 큰 경우 디큐 후보 패킷을 드롭할 수 있다.
- [0048]            현재 시각이 상기 다음 드롭 시간(next\_drop\_time)보다 큰 경우, 단계(330)에서 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간(next\_drop\_time)과 활성 인터벌(m\_acting\_interval)을 결정하고, 인터벌 설정 상태를 확인할 수 있다. 모든 큐의 전송 속도의 평균값의 변수는 평균값 그 자체가 될 수도 있고, 표준편차의 배수를 더해서 결정할 수도 있다.
- [0049]            큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교(333)하여 상기 큐의 전송 속도가 상기 모든 큐의 전송속도의 평균값의 변수보다 작은 경우 상기 디큐 후보 패킷을 디큐(328)하고 다음 디큐 후보 패킷을 선택(310)할 수 있다.
- [0050]            큐의 전송 속도가 모든 큐의 전송속도의 평균값의 변수보다 큰 경우 디큐 후보 패킷을 드롭할 수 있다. 다시 말해, 해당 큐의 전송 속도가 모든 큐의 전송속도의 평균값의 변수 보다 큰 경우, 활성 인터벌(m\_acting\_interval) 값과 다음 드롭 시간(next\_drop\_time)을 결정 할 수 있다(332). 활성 인터벌(m\_acting\_interval) 값은 다음과 같이 결정한다. next\_drop\_time이 0 인 경우, 활성 인터벌(m\_acting\_interval) = m\_interval로 한다. 다음 드롭 시간(next\_drop\_time)이 0가 아닌 경우, 활성 인터벌(m\_acting\_interval) =  $0.75 \wedge m\_drop\_count$ 로 한다. 이때 다음 드롭 시간은 수학식4와 같이 상기 활성 인터벌과 상기 현재 시각을 이용하여 결정할 수 있다.
  
- [0051]            다음 드롭 시간(next\_drop\_time) = 현재 시각 + 활성 인터벌(m\_acting\_interval) 수학식4
  
- [0052]            상기 다음 드롭 시간과 상기 활성 인터벌을 결정한 후 드롭 카운트를 1만큼 증가 시킬 수 있다. 그리고, 인터벌 설정 상태를 확인할 수 있다(331).
- [0053]            만약 인터벌 설정 상태가 대기 상태(WAIT)일 경우, 상기 디큐 후보 패킷을 드롭(343)하고 다음 디큐 후보 패킷을 선택(310)할 수 있다.

- [0054] 인터벌 설정 상태가 프리 상태(FREE)인 경우, 단계(340)에서 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 현재 시각(dropped time)을 저장(341)하고 상기 인터벌 설정 상태(interval\_setting\_stat)를 대기 상태(WAIT)로 설정(342)한 후, 상기 디큐 후보 패킷을 드롭(343)하고 다음 디큐 후보 패킷을 선택(343)할 수 있다.
- [0055] 도 4는 본 발명의 일 실시예에 따른 자동 버퍼 관리의 인큐 장치의 구성을 나타내는 도면이다.
- [0056] 자동 버퍼 관리의 인큐 장치(400)는 큐 할당부(410), 인큐 수행부(420)를 포함할 수 있다.
- [0057] 큐 할당부(410)는 현재 시각으로 패킷에 태그 작업을 수행하고 상기 패킷을 해시 기능으로 분류하여 큐를 할당할 수 있다. 큐 할당부(410)는 패킷을 받은 시간을 저장하고, 해시 기능으로 분류하여 새로운 플로우일 경우 새로운 큐를 할당하고 기존에 존재하던 플로우일 경우 기존에 할당되었던 큐를 할당할 수 있다. 다시 말해, 패킷을 받은 시간을 저장하고, 이를 인큐 타임(Enqueue Time)이라고 할 수 있다. 그리고, 5개 정보(IP Source Address, IP Destination Address, Source Port, Destination Port, Protocol)를 사용하여 해시 기능(Hash Function)을 돌린 후 플로우 큐(Flow Queue)를 할당할 수 있다. 이때 플로우가 새로운 플로우일 경우에는 새로운 큐를 할당하고, 기존에 존재하던 플로우일 경우에는 기존에 할당되었던 큐를 할당할 수 있다. 이로써 플로우별 큐 관리를 일관성 있게 수행할 수 있다.
- [0058] 인큐 수행부(420)는 상기 큐가 중단 플로우이고, 인터벌 설정 상태가 대기 상태이고, 현재 전송 제어 프로토콜 시퀀스와 저장된 전송 제어 프로토콜 시퀀스가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료할 수 있다.
- [0059] 할당된 큐가 중단 플로우(Edge Flow)인지 확인하고, 상기 큐가 중단 플로우일 경우 동적 인터벌(Dynamic Interval) 값 조정의 CheckTCPSeqNForInterval()이 수행될 수 있다. 만약 할당된 큐가 중단 플로우가 아닌 경우에는 상기 인큐 과정을 종료할 수 있다.
- [0060] 인터벌 설정 상태(interval\_setting\_state)가 대기 상태(WAIT)인지 프리 상태(FREE)인지를 확인할 수 있다. 만약 인터벌 설정 상태(interval\_setting\_state)가 프리 상태(FREE)인 경우에는 인큐 과정을 종료할 수 있다.
- [0061] 인터벌 설정 상태(interval\_setting\_state)가 대기 상태(WAIT)인 경우, 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)를 비교할 수 있다.
- [0062] 다시 말해, 인터벌 설정 상태(interval\_setting\_state) 대기 상태(WAIT)인 경우, 패킷의 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)를 확인하여 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 일치하는지 확인할 수 있다.
- [0063] 그리고, 디큐(Dequeue) 과정에서 이미 드롭(Drop)된 패킷이 존재하는 경우에는 인터벌 설정 상태(interval\_setting\_state)가 대기 상태(WAIT)으로 설정되어 있게 된다.
- [0064] 만약, 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)가 같지 않은 경우 상기 인큐 과정을 종료할 수 있다.
- [0065] 상기 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)가 같은 경우, 상기 인터벌 설정 상태에 따른 다이내믹 인터벌 값을 조정하여 인큐 과정을 종료할 수 있다.
- [0066] 패킷의 현재 전송 제어 프로토콜 시퀀스(TCP Sequence Number)를 확인하여 저장된 전송 제어 프로토콜 시퀀스(TCP Sequence Number)와 일치 하는지 확인 한 후, 그때까지의 걸린 시간 m\_Rtt을 구할 수 있다. 그리고, m\_Rtt 변수값을 구할 수 있다. 마지막으로 인터벌 설정 상태(interval\_setting\_state) 상태를 프리 상태(FREE)로 설정하고 인큐(Enqueue) 과정을 종료할 수 있다.
- [0067] 도 5는 본 발명의 일 실시예에 따른 자동 버퍼 관리의 디큐 장치의 구성을 나타내는 도면이다.
- [0068] 자동 버퍼 관리의 디큐 장치(500)는 처리부(510), 패킷 비교부(520), 공평성 확인부(530), 인터벌 설정부(540)를 포함할 수 있다.

- [0069] 처리부(510)는 디큐 후보 패킷을 선택하고, 상기 디큐 후보 패킷을 전송하고, 또는 드롭할 수 있다.
- [0070] 패킷 비교부(520)는 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우인지 확인하고, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 상기 디큐 후보 패킷의 머문 시간을 계산하여 미리 정해진 타겟 시간과 비교하고, 현재 시각과 다음 드롭 시간을 비교할 수 있다.
- [0071] 만약, 상기 선택된 디큐 후보 패킷의 큐가 중단 플로우가 아닌 경우, 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하도록 할 수 있다.
- [0072] 선택된 디큐 후보 패킷의 큐가 중단 플로우일 경우, 머문 시간을 계산할 수 있다. 다음으로, 머문 시간과 미리 정해진 타겟 시간  $m\_target$  을 비교할 수 있다.
- [0073] 머문 시간이 미리 정해진 타겟 시간  $m\_target$ 보다 작은 경우, 다음 드롭 시간(next\_drop\_time)을 0으로 설정하고 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하도록 할 수 있다.
- [0074] 머문 시간이 미리 정해진 타겟 시간  $m\_target$ 보다 큰 경우, 다음 드롭 시간(next\_drop\_time)을 확인할 수 있다.
- [0075] 머문 시간이 상기 미리 정해진 타겟 시간보다 크고, 상기 다음 드롭 시간이 0 인 경우, 상기 다음 드롭 시간(next\_drop\_time)을  $m\_interval$  값을 이용하여 업데이트한 후 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하도록 할 수 있다. 다음 드롭 시간이 0이 아닌 인 경우, 현재 시각과 상기 다음 드롭 시간을 비교할 수 있다.
- [0076] 머문 시간이 상기 미리 정해진 타겟 시간보다 크고, 상기 다음 드롭 시간이 0 이 아니고, 상기 현재 시각이 상기 다음 드롭 시간보다 작은 경우 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하도록 할 수 있다.
- [0077] 현재 시각이 상기 다음 드롭 시간(next\_drop\_time)보다 큰 경우, 디큐 후보 패킷을 드롭할 수 있다. 다시 말해, 머문 시간이 미리 정해진 타겟 시간보다 크고, 다음 드롭 시간이 0 이 아니고, 현재 시각이 상기 다음 드롭 시간(next\_drop\_time)보다 큰 경우 디큐 후보 패킷을 드롭할 수 있다.
- [0078] 공평성 확인부(530)는 현재 시각이 상기 다음 드롭 시간보다 큰 경우, 상기 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 다음 드롭 시간과 활성 인터벌을 결정하고, 인터벌 설정 상태를 확인할 수 있다.
- [0079] 큐의 전송 속도와 모든 큐의 전송속도의 평균값의 변수를 비교하여 상기 큐의 전송 속도가 상기 모든 큐의 전송속도의 평균값의 변수보다 작은 경우 상기 디큐 후보 패킷을 디큐하고 다음 디큐 후보 패킷을 선택하도록 할 수 있다.
- [0080] 해당 큐의 전송 속도가 모든 큐의 전송속도의 평균값의 변수 보다 큰 경우, 활성 인터벌( $m\_acting\_interval$ ) 값과 다음 드롭 시간(next\_drop\_time)을 결정 할 수 있다. 다음 드롭 시간과 상기 활성 인터벌을 결정한 후 드롭 카운트를 1만큼 증가 시킬 수 있다. 그리고, 인터벌 설정 상태를 확인할 수 있다. 만약 인터벌 설정 상태가 대기 상태(WAIT)일 경우, 상기 디큐 후보 패킷을 드롭하고 다음 디큐 후보 패킷을 선택하도록 할 수 있다.
- [0081] 인터벌 설정부(540)는 상기 인터벌 설정 상태가 프리인 경우, 상기 디큐 후보 패킷의 전송 제어 프로토콜 시퀀스와 현재 시각을 저장하고 상기 인터벌 설정 상태를 대기 상태로 설정할 수 있다. 그리고, 디큐 후보 패킷을 드롭하고 다음 디큐 후보 패킷을 선택하도록 할 수 있다.
- [0082] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치

는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0083] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

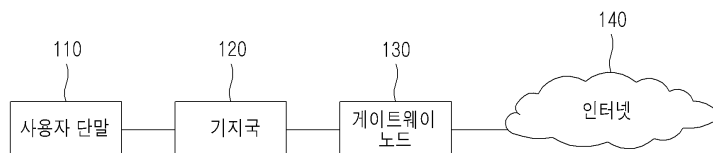
[0084] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0085] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

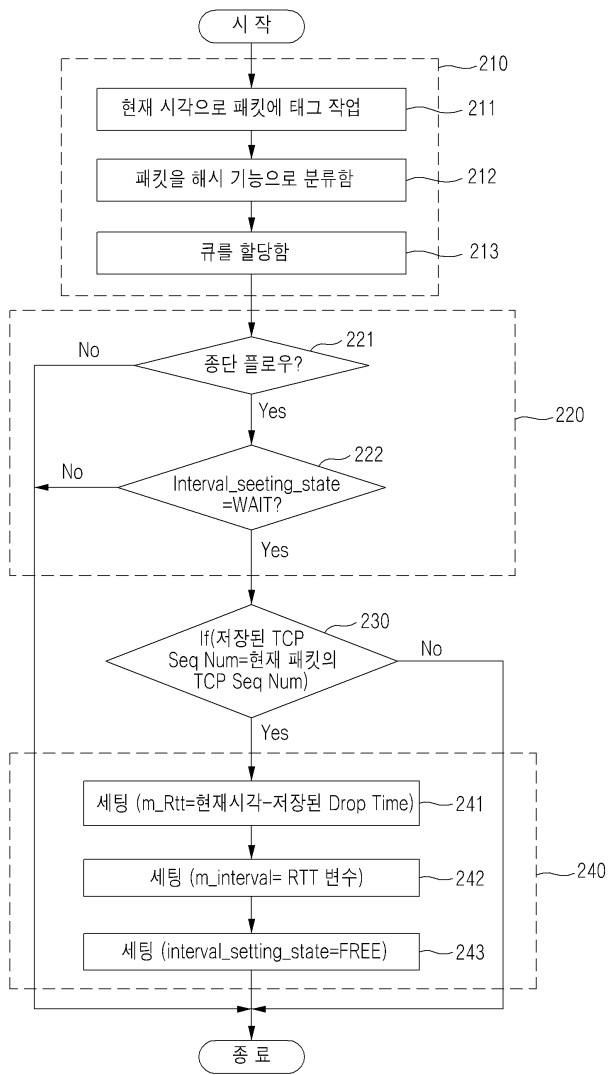
[0086] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

**도면**

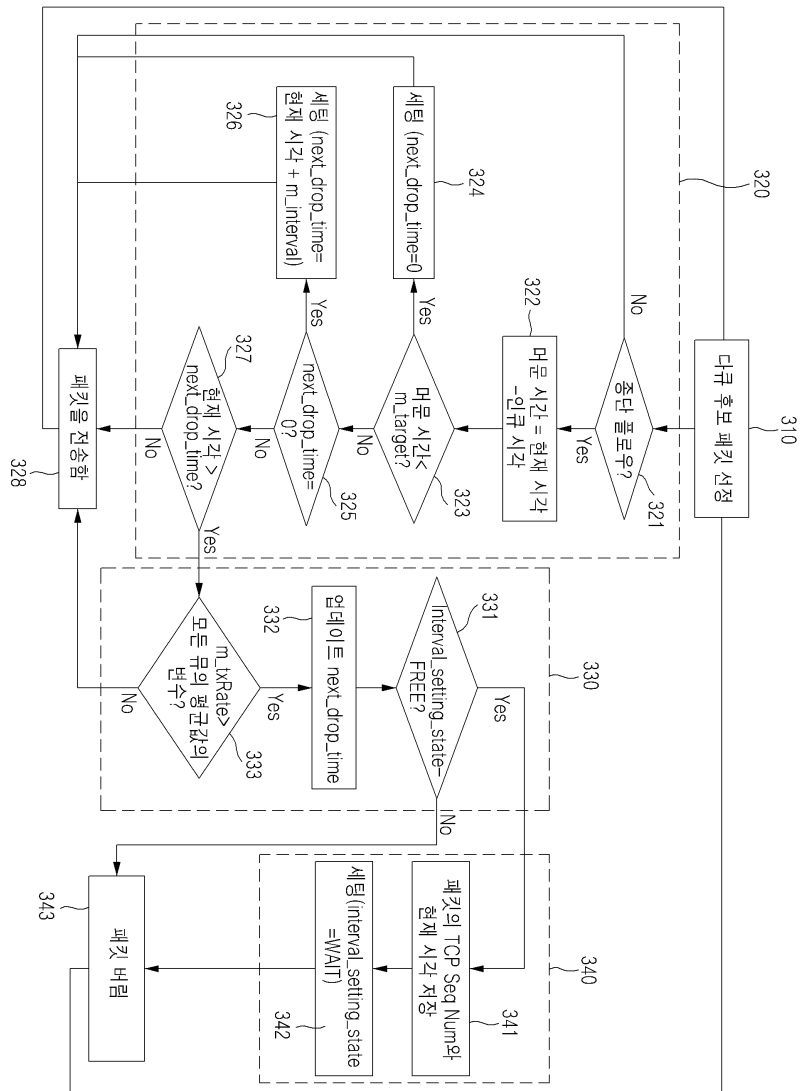
**도면1**



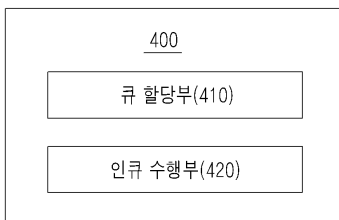
도면2



도면3



도면4



도면5

