(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0010531
(43) 공개일자 2018년01월31일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 12/801* (2013.01)
H04L 12/805 (2013.01) *H04L 12/841* (2013.01)
(52) CPC특허분류
H04L 69/16 (2013.01)
H04L 47/283 (2013.01)
(21) 출원번호 10-2016-0092784
(22) 출원일자 2016년07월21일
심사청구일자 없음

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
더 리전츠 오브 더 유니버시티 오브 콜로라도, 어 바디 코퍼레이트
미국 콜로라도주 80203 덴버 8번 플로어 그랜트 스트리트 1800
(72) 발명자
이진성
경기도 수원시 영통구 삼성로 11 래미안 영통마크원 2단지아파트 204동 2303호
임영빈
미국, 콜로라도주 80301, 볼더, 5119 윌리암스 포크 트레일 에이피티 118
(뒷면에 계속)
(74) 대리인
이건주, 김정훈

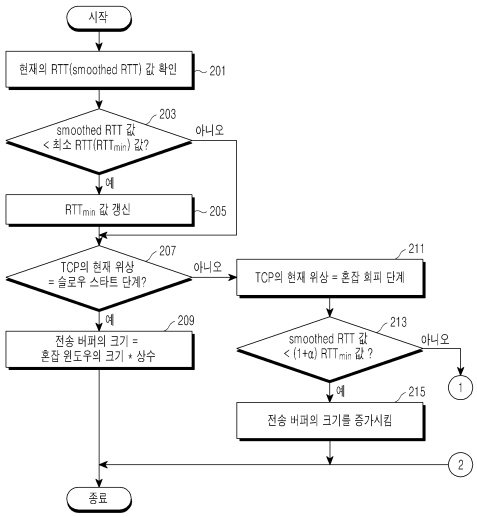
전체 청구항 수 : 총 26 항

(54) 발명의 명칭 **통신 시스템에서 전송 제어 프로토콜의 전송 버퍼 제어 방법 및 장치**

(57) 요 약

본 개시는 통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하기 위한 방법 및 장치에 관한 것이다. 본 개시의 일 실시 예에서 제안하는 방법은; 통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 방법에 있어서, 상기 TCP에서 현재 라운드 트립 타임(round trip time : RTT) 값 및 최소 RTT 값을 확인하는 과정; 및 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 전송 버퍼의 크기를 조정하는 과정을 포함한다.

대 표 도 - 도2a



(52) CPC특허분류
H04L 47/29 (2013.01)
H04L 47/36 (2013.01)
(72) 발명자
하상태
미국, 콜로라도주 80027, 루이스빌, 861 세인트 앤드루스 레인
문상준
서울특별시 동작구 상도로 407 삼호아파트 102동 1402호
박중신
서울특별시 영등포구 도림로64가길 15-1
이지철
경기도 수원시 영통구 효원로 363 신매탄위브하늘채아파트 130동 906호
이주형
경기도 과천시 부림로 2 주공아파트 9단지 917동 503호

명 세 서

청구범위

청구항 1

통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 방법에 있어서,

상기 TCP에서 현재 라운드 트립 타임(round trip time : RTT) 값 및 최소 RTT 값을 확인하는 과정; 및

상기 TCP에 따른 혼잡 제어 방식에서 현재 단계, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 전송 버퍼의 크기를 조정하는 과정을 포함하는 TCP의 전송 버퍼 제어 방법.

청구항 2

제 1 항에 있어서, 상기 전송 버퍼의 크기를 조정하는 과정은,

상기 현재 단계가 슬로우 스타트(slow start) 단계이면, 혼잡 윈도우의 크기에 일정 상수를 곱하여 상기 전송 버퍼의 크기를 조정하는 과정임을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 3

제 1 항에 있어서, 상기 전송 버퍼의 크기를 조정하는 과정은,

상기 현재 단계가 혼잡 회피(congestion avoidance) 단계이면, 상기 현재 RTT 값과 상기 최소 RTT 값의 차이 값에 따라 상기 전송 버퍼의 크기를 조정하는 과정임을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 4

제 3 항에 있어서, 상기 전송 버퍼의 크기를 조정하는 과정은,

상기 현재 RTT 값이 제1 임계 값보다 작은 경우, 상기 전송 버퍼의 크기를 일정 상수만큼(additive) 증가시키는 과정;

상기 현재 RTT 값이 제1 임계 값과 제2 임계 값의 사이 값인 경우, 상기 전송 버퍼의 크기를 유지시키는 과정; 및

상기 현재 RTT 값이 상기 제2 임계 값보다 큰 경우, 전송 버퍼의 크기를 배수(multiplicative)로 감소시키는 과정을 포함하는 TCP의 전송 버퍼 제어 방법.

청구항 5

제 4 항에 있어서, 상기 제1 임계 값 및 상기 제2 임계 값은 상기 RTT 최소 값을 기반으로 결정되고, 상기 제2 임계 값은 상기 제1 임계 값보다 큰 값임을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 6

통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 방법에 있어서,

상대 장치로 정상 응답 신호(TCP acknowledgement)를 전송하는 경우의 라운드 트립 타임(round trip time :

RTT) 값을 기반으로 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계를 예측하는 과정;

상기 상대 장치로부터 수신된 데이터 패킷을 기반으로 혼잡 윈도우의 크기를 예측하는 과정;

상기 TCP에서 현재 RTT 값 및 최소 RTT 값을 확인하는 과정; 및

상기 현재 단계, 상기 혼잡 윈도우의 크기, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 상기 상대 장치로 전송할 윈도우의 크기를 조정하는 과정을 포함하는 TCP의 전송 버퍼 제어 방법.

청구항 7

제 6 항에 있어서, 상기 현재 단계를 예측하는 과정은,

상기 상대 장치로 상기 정상 응답 신호를 전송하는 경우, 상기 상대 장치로부터 하나의 라운드(round)에서 수신된 패킷 트레인의 길이와 상기 최소 RTT 값의 절반 값의 비교 결과에 따라 상기 현재 단계를 예측하는 과정임을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 8

제 6 항에 있어서, 상기 현재 단계를 예측하는 과정은,

상기 상대 장치로부터 하나의 라운드(round)에서 수신된 데이터 패킷에 포함된 소정 개수의 정상 응답 신호들을 이용하여 계산된 RTT 값과, 상기 라운드의 이전 라운드에서 계산된 RTT 값의 차이 값에 따라 상기 현재 단계를 예측하는 과정임을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 9

제 6 항에 있어서, 상기 혼잡 윈도우의 크기를 예측하는 과정은,

상기 상대 장치로부터 수신된 데이터 패킷의 총 바이트 크기를 미리 알려진 최대 세그먼트 사이즈(maximum segment size)로 나눈 값을 이용하여 상기 상대 장치에서 결정된 혼잡 윈도우의 크기를 예측하는 과정이며,

상기 혼잡 윈도우의 크기는 패킷 개수의 단위임을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 10

제 6 항에 있어서, 상기 상대 장치로 전송할 윈도우의 크기를 조정하는 과정은,

상기 현재 단계가 슬로우 스타트(slow start) 단계이면, 혼잡 윈도우의 크기에 일정 상수를 곱하여 상기 상대 장치로 전송할 윈도우의 크기를 결정하는 과정을 포함함을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 11

제 6 항에 있어서, 상기 상대 장치로 전송할 윈도우의 크기를 조정하는 과정은,

상기 현재 단계가 혼잡 회피(congestion avoidance) 단계이면, 상기 현재 RTT 값과 상기 최소 RTT 값의 차이 값에 따라 상기 상대 장치로 전송할 윈도우의 크기를 결정하는 과정을 포함함을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 12

제 11 항에 있어서, 상기 상대 장치로 전송할 윈도우의 크기를 조정하는 과정은,

상기 현재 RTT 값이 제1 임계 값보다 작은 경우, 상기 상대 장치로 전송할 윈도우의 크기를 일정 상수만큼(additive) 증가시키는 과정;

상기 현재 RTT 값이 제1 임계 값과 제2 임계 값의 사이 값인 경우, 상기 상대 장치로 전송할 윈도우의 크기를 유지시키는 과정; 및

상기 현재 RTT 값이 상기 제2 임계 값보다 큰 경우, 상기 상대 장치로 전송할 윈도우의 크기를 배수(multiplicative)로 감소시키는 단계를 포함하는 TCP의 전송 버퍼 제어 방법.

청구항 13

제 12 항에 있어서, 상기 제1 임계 값 및 상기 제2 임계 값은 상기 RTT 최소 값을 기반으로 결정되고, 상기 제2 임계 값은 상기 제1 임계 값보다 큰 값임을 특징으로 하는 TCP의 전송 버퍼 제어 방법.

청구항 14

통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 장치에 있어서,

데이터 패킷을 송수신하는 송수신부; 및

상기 TCP에서 현재 라운드 트립 타임(round trip time : RTT) 값 및 최소 RTT 값을 확인하고, 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 전송 버퍼의 크기를 조정하는 제어부를 포함하는 TCP의 전송 버퍼 제어 장치.

청구항 15

제 14 항에 있어서, 상기 제어부는,

상기 현재 단계가 슬로우 스타트(slow start) 단계이면, 혼잡 윈도우의 크기에 일정 상수를 곱하여 상기 전송 버퍼의 크기를 조정함을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 16

제 14 항에 있어서, 상기 제어부는,

상기 현재 단계가 혼잡 회피(congestion avoidance) 단계이면, 상기 현재 RTT 값과 상기 최소 RTT 값의 차이 값에 따라 상기 전송 버퍼의 크기를 조정함을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 17

제 16 항에 있어서, 상기 제어부는,

상기 현재 RTT 값이 제1 임계 값보다 작은 경우, 상기 전송 버퍼의 크기를 일정 상수만큼(additive) 증가시키고, 상기 현재 RTT 값이 제1 임계 값과 제2 임계 값의 사이 값인 경우, 상기 전송 버퍼의 크기를 유지시키고, 상기 현재 RTT 값이 상기 제2 임계 값보다 큰 경우, 전송 버퍼의 크기를 배수(multiplicative)로 감소시킴을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 18

제 17 항에 있어서, 상기 제1 임계 값 및 상기 제2 임계 값은 상기 RTT 최소 값을 기반으로 결정되고, 상기 제2

임계 값은 상기 제1 임계 값보다 큰 값임을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 19

통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 장치에 있어서,

데이터 패킷을 송수신하는 송수신부; 및

상대 장치로 정상 응답 신호(TCP acknowledgement)를 전송하는 경우의 라운드 트립 타임(round trip time : RTT) 값을 기반으로 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계를 예측하고, 상기 송수신부를 통해 수신된 데이터 패킷을 기반으로 혼잡 윈도우의 크기를 예측하고, 상기 TCP에서 현재 RTT 값 및 최소 RTT 값을 확인하여, 상기 현재 단계, 상기 혼잡 윈도우의 크기, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 상기 상대 장치로 전송할 윈도우의 크기를 조정하는 제어부를 포함하는 TCP의 전송 버퍼 제어 장치.

청구항 20

제 19 항에 있어서, 상기 제어부는,

상기 상대 장치로 상기 정상 응답 신호를 전송하는 경우, 상기 송수신부를 통해 상기 상대 장치로부터 하나의 라운드(round)에서 수신된 패킷 트레인의 길이와 상기 최소 RTT 값의 절반 값의 비교 결과에 따라 상기 현재 단계를 예측함을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 21

제 19 항에 있어서, 상기 제어부는,

상기 송수신부를 통해 상기 상대 장치로부터 하나의 라운드(round)에서 수신된 데이터 패킷에 포함된 소정 개수의 정상 응답 신호들을 이용하여 계산된 RRT 값과, 상기 라운드의 이전 라운드에서 계산된 RTT 값의 차이 값에 따라 상기 현재 단계를 예측함을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 22

제 19 항에 있어서, 상기 제어부는,

상기 송수신부를 통해 수신된 데이터 패킷의 총 바이트 크기를 미리 알려진 최대 세그먼트 사이즈(maximum segment size)로 나눈 값을 이용하여 상기 상대 장치에서 결정된 혼잡 윈도우의 크기를 예측하며,

상기 혼잡 윈도우의 크기는 패킷 개수의 단위임을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 23

제 19 항에 있어서, 상기 제어부는,

상기 현재 단계가 슬로우 스타트(slow start) 단계이면, 혼잡 윈도우의 크기에 일정 상수를 곱하여 상기 상대 장치로 전송할 윈도우의 크기를 결정함을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 24

제 19 항에 있어서, 상기 제어부는,

상기 현재 단계가 혼잡 회피(congestion avoidance) 단계이면, 상기 현재 RTT 값과 상기 최소 RTT 값의 차이 값

에 따라 상기 상대 장치로 전송할 윈도우의 크기를 결정함을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 25

제 24 항에 있어서, 상기 제어부는,

상기 현재 RTT 값이 제1 임계 값보다 작은 경우, 상기 상대 장치로 전송할 윈도우의 크기를 일정 상수만큼(additive) 증가시키고, 상기 현재 RTT 값이 제1 임계 값과 제2 임계 값의 사이 값인 경우, 상기 상대 장치로 전송할 윈도우의 크기를 유지시키고, 상기 현재 RTT 값이 상기 제2 임계 값보다 큰 경우, 상기 상대 장치로 전송할 윈도우의 크기를 배수(multiplicative)로 감소시킴을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

청구항 26

제 25 항에 있어서, 상기 제1 임계 값 및 상기 제2 임계 값은 상기 RTT 최소 값을 기반으로 결정되고, 상기 제2 임계 값은 상기 제1 임계 값보다 큰 값임을 특징으로 하는 TCP의 전송 버퍼 제어 장치.

발명의 설명

기 술 분 야

[0001] 본 개시는 통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하기 위한 방법 및 장치에 관한 것이다.

배 경 기 술

[0003] 통신 시스템에서 TCP는 인터넷의 핵심적인 프로토콜 중의 하나로서, 인터넷 프로토콜(internet protocol : IP) 네트워크 상의 호스트(host)에서 동작하는 어플리케이션 간의 신뢰성 있는 데이터 패킷 전송하기 위한 프로토콜이다. 상기 TCP는 처리율 성능의 최대화, 플로우(flow) 간의 공정성(fairness) 보장 등을 목표로 다양한 프로토콜이 개발되었다. 일 예로, 상기 TCP는 무선 네트워크에 특화된 다양한 프로토콜로, TCP 베이거스(vegas), 웨스트우드(Westwood) 및 베노(Veno) 등과 같은 프로토콜이 제안되었다. 그리고 상기 TCP는 대역폭 지연 시간 곱(bandwidth delay product)이 큰 네트워크를 위한 프로토콜로, 고속(high-speed) TCP, 큐빅(cubic) 및 패스트(fast) 등과 같은 프로토콜이 제안되었다. 또한 상기 TCP는 데이터 패킷 센터에 특화된 데이터 패킷 센터 TCP(data center TCP : DCTCP), 홀(HULL) 등과 같은 프로토콜과 여러 전송 경로를 동시에 활용하는 다중 경로(multi-path) 프로토콜들과 같은 프로토콜이 제안되었다.

[0004] 종래 제안된 TCP는 지연 시간을 고려할 때 TCP 상에서 측정된 라운드 트립 타임(round trip time : RTT)에서 발생되는 지연 시간을 최소화하기 위한 방안을 고려하였다. 그러나, 실시간 스트리밍 등과 같이 지연 시간이 중요한 어플리케이션에서는 어플리케이션 간에 발생되는 지연 시간을 최소화하는 것이 중요하다. 즉 데이터 패킷을 전송하는 어플리케이션에서 TCP의 전송 장치(sender)로 데이터 패킷을 내려 보내는 시점(예컨대, 소켓에 데이터 패킷을 쓰는 시점)으로부터 데이터 패킷을 수신하는 어플리케이션에서 TCP의 수신 장치(receiver)로부터 데이터 패킷을 전달 받는 시점(예컨대, 소켓으로부터 데이터 패킷을 읽는 시점)까지 걸리는 시간이 실제 어플리케이션의 성능에 크게 영향을 주는 것이다. 이러한 어플리케이션 간 지연 시간에 크게 영향을 미치는 것이 TCP의 전송 장치에서 관리하는 전송 버퍼(send buffer)의 크기이다. 기존의 TCP에서는 데이터 패킷의 처리율 성능을 최대화하기 위해 전송 장치에서 관리하는 전송 버퍼의 크기를 가능한 크게 유지하였다. 일 예로, 리눅스의 경우, 혼잡 윈도우(congestion window)의 크기를 고려하여, 혼잡 윈도우의 크기에 하나의 데이터 패킷 당 차지하는 메모리 용량을 곱하고, 좀 더 여유로운 크기를 갖도록 2를 곱한 값을, 전송 버퍼의 크기로 결정한다. 따라서 리눅스의 경우 혼잡 윈도우의 크기가 커지면 전송 버퍼의 크기도 급격하게 커지게 된다. 상기 전송 장치에서 관리하는 전송 버퍼의 크기가 커지면, 전송 장치에서 전송 기회가 생길 때마다 보낼 수 있는 데이터 패킷의 수를 충분히 확보하게 되므로 최대 처리율을 유지하는데 도움이 될 수 있다. 그러나 전송 버퍼의 크기가 지나치게 커지면 데이터 패킷이 실제 네트워크 상에서 전송되는 데 소모되는 시간보다 전송 장치에서 전송 기회를 기다리면서 소모되는 시간이 매우 커지게 된다. 이로 인해 어플리케이션 간 지연 시간의 증가는 저 지연 시간 전송이 중요한 실시

간 스트리밍 및 화상회의 등과 같은 어플리케이션에서 서비스 품질을 크게 저하시킬 수 있다. 이를 위해, TCP에서 데이터 패킷의 처리율에 대한 성능을 저하 시키지 않으면서, 어플리케이션 간 지연 시간을 최소화할 수 있는 방안이 요구된다.

발명의 내용

해결하려는 과제

[0006] 본 개시의 일 실시 예는 통신 시스템에서 어플리케이션 간의 지연 시간을 최소화하도록 전송 버퍼를 제어하는 방법 및 장치를 제공한다.

[0007] 또한 본 개시의 일 실시 예는 통신 시스템에서 이용되는 TCP의 전송 버퍼의 크기를 네트워크 상황에 따라 제어하는 방법 및 장치를 제공한다.

과제의 해결 수단

[0009] 본 개시의 일 실시 예에서 제안하는 방법은; 통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 방법에 있어서, 상기 TCP에서 현재 라운드 트립 타임(round trip time : RTT) 값 및 최소 RTT 값을 확인하는 과정; 및 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 전송 버퍼의 크기를 조정하는 과정을 포함한다.

[0010] 또한 본 개시의 일 실시 예에서 제안하는 방법은; 통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 방법에 있어서, 상대 장치로 정상 응답 신호(TCP acknowledgement)를 전송하는 경우의 라운드 트립 타임(round trip time : RTT) 값을 기반으로 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계를 예측하는 과정; 상기 상대 장치로부터 수신된 데이터 패킷을 기반으로 혼잡 윈도우의 크기를 예측하는 과정; 상기 TCP에서 현재 RTT 값 및 최소 RTT 값을 확인하는 과정; 및 상기 현재 단계, 상기 혼잡 윈도우의 크기, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 상기 상대 장치로 전송할 윈도우의 크기를 조정하는 과정을 포함한다.

[0011] 또한 본 개시의 일 실시 예에서 제안하는 장치는; 통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 장치에 있어서, 데이터 패킷을 송수신하는 송수신부; 및 상기 TCP에서 현재 라운드 트립 타임(round trip time : RTT) 값 및 최소 RTT 값을 확인하고, 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 전송 버퍼의 크기를 조정하는 제어부를 포함한다.

[0012] 또한 본 개시의 일 실시 예에서 제안하는 장치는; 통신 시스템에서 전송 제어 프로토콜(transport control protocol : TCP)의 전송 버퍼를 제어하는 장치에 있어서, 데이터 패킷을 송수신하는 송수신부; 및 상대 장치로 정상 응답 신호(TCP acknowledgement)를 전송하는 경우의 라운드 트립 타임(round trip time : RTT) 값을 기반으로 상기 TCP에 따른 혼잡 제어 방식에서 현재 단계를 예측하고, 상기 송수신부를 통해 수신된 데이터 패킷을 기반으로 혼잡 윈도우의 크기를 예측하고, 상기 TCP에서 현재 RTT 값 및 최소 RTT 값을 확인하여, 상기 현재 단계, 상기 혼잡 윈도우의 크기, 상기 현재 RTT 값 및 상기 최소 RTT 값 중 적어도 하나를 기반으로 상기 상대 장치로 전송할 윈도우의 크기를 조정하는 제어부를 포함한다.

[0014] 본 개시의 다른 측면들과, 이득들 및 핵심적인 특징들은 부가 도면들과 함께 처리되고, 본 개시의 바람직한 실시 예들을 개시하는, 하기의 구체적인 설명으로부터 해당 기술 분야의 당업자에게 자명할 것이다.

[0015] 하기의 본 개시의 구체적인 설명 부분을 처리하기 전에, 이 특허 문서를 통해 사용되는 특정 단어들 및 구문들에 대한 정의들을 설정하는 것이 효과적일 수 있다: 상기 용어들 "포함하다(include)" 및 "포함하다(comprise)"와 그 파생어들은 한정없는 포함을 의미하며; 상기 용어 "혹은(or)"은 포괄적이고, "및/또는"을 의미하고; 상기 구문들 "~와 연관되는(associated with)" 및 "~와 연관되는(associated therewith)"과 그 파생어들은 포함하고(include), ~내에 포함되고(be included within), ~와 서로 연결되고(interconnect with), 포함하고(contain), ~내에 포함되고(be contained within), ~에 연결하거나 혹은 ~와 연결하고(connect to or with), ~

에 연결하거나 혹은 ~와 연결하고(couple to or with), ~와 통신 가능하고(be communicable with), ~와 협조하고(cooperate with), 인터리빙하고(interleave), 병치하고(juxtapose), ~로 가장 근접하고(be proximate to), ~로 ~할 가능성이 크거나 혹은 ~와 ~할 가능성이 크고(be bound to or with), 가지고(have), 소유하고(have a property of) 등과 같은 내용을 의미하고; 상기 용어 "제어기"는 적어도 하나의 동작을 제어하는 임의의 디바이스, 시스템, 혹은 그 부분을 의미하고, 상기와 같은 디바이스는 하드웨어, 펌웨어 혹은 소프트웨어, 혹은 상기 하드웨어, 펌웨어 혹은 소프트웨어 중 적어도 2개의 몇몇 조합에서 구현될 수 있다. 어떤 특정 제어기와 연관되는 기능성이라도 집중화되거나 혹은 분산될 수 있으며, 국부적이거나 원격적일 수도 있다는 것에 주의해야만 할 것이다. 특정 단어들 및 구문들에 대한 정의들은 이 특허 문서에 걸쳐 제공되고, 해당 기술 분야의 당업자는 많은 경우, 대부분의 경우가 아니라고 해도, 상기와 같은 정의들이 종래 뿐만 아니라 상기와 같이 정의된 단어들 및 구문들의 미래의 사용들에도 적용된다는 것을 이해해야만 할 것이다.

도면의 간단한 설명

[0017] 도 1은 본 개시의 실시 예에 따른 통신 시스템의 일 예를 나타낸 도면,

도 2a 및 도 2b는 본 개시의 실시 예에 따른 통신 시스템에서 전송 장치(110)에 의해 전송 버퍼를 제어하는 방법을 나타낸 도면,

도 3은 본 개시의 실시 예에서 RTT 값에 따른 TCP의 전송 버퍼의 크기 변화의 일 예를 나타낸 도면,

도 4는 본 개시의 실시 예에 따른 통신 시스템에서 수신 장치(150)에 의해 전송 버퍼를 제어하는 방법을 나타낸 도면,

도 5는 본 개시의 실시 예에 따른 통신 시스템에서 중계 장치(130)에 의해 전송 버퍼를 제어하는 방법을 나타낸 도면,

도 6은 본 개시의 실시 예에 따른 통신 시스템에서 TCP의 전송 버퍼를 제어하는 전송 장치(110)의 내부 구성을 나타낸 도면,

도 7은 본 개시의 실시 예에 따른 통신 시스템에서 TCP의 전송 버퍼를 제어하는 수신 장치(150)의 내부 구성을 나타낸 도면,

도 8은 본 개시의 실시 예에 따른 통신 시스템에서 TCP의 전송 버퍼를 제어하는 중계 장치(130)의 내부 구성을 나타낸 도면,

도 9는 현재 통신 시스템의 TCP에서 지연 시간이 발생되는 장치를 확인한 결과를 나타낸 도면,

도 10은 통신 시스템에서 송신 장치(110) 및 수신 장치(150)에 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과를 나타낸 도면,

도 11은 통신 시스템에서 송신 장치(110), 중계 장치(130) 및 수신 장치(150)에 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과를 나타낸 도면.

상기 도면들을 통해, 유사 참조 번호들은 동일한 혹은 유사한 엘리먼트들과, 특징들 및 구조들을 도시하기 위해 사용된다는 것에 유의해야만 한다.

발명을 실시하기 위한 구체적인 내용

[0018] 첨부되는 도면들을 참조하는 하기의 상세한 설명은 청구항들 및 청구항들의 균등들로 정의되는 본 개시의 다양한 실시 예들을 포괄적으로 이해하는데 있어 도움을 줄 것이다. 하기의 상세한 설명은 그 이해를 위해 다양한 특정 구체 사항들을 포함하지만, 이는 단순히 예로서만 간주될 것이다. 따라서, 해당 기술 분야의 당업자는 여기에서 설명되는 다양한 실시 예들의 다양한 변경들 및 수정들이 본 개시의 범위 및 사상으로부터 벗어남이 없이 이루어질 수 있다는 것을 인식할 것이다. 또한, 공지의 기능들 및 구성들에 대한 설명은 명료성 및 간결성을 위해 생략될 수 있다.

[0019] 하기의 상세한 설명 및 청구항들에서 사용되는 용어들 및 단어들은 문헌적 의미로 한정되는 것이 아니라, 단순히 발명자에 의한 본 개시의 명료하고 일관적인 이해를 가능하게 하도록 하기 위해 사용될 뿐이다. 따라서, 해당 기술 분야의 당업자들에게는 본 개시의 다양한 실시 예들에 대한 하기의 상세한 설명은 단지 예시 목적만을

위해 제공되는 것이며, 첨부되는 청구항들 및 상기 청구항들의 균등들에 의해 정의되는 본 개시를 한정하기 위해 제공되는 것은 아니라는 것이 명백해야만 할 것이다.

[0020] 또한, 본 명세서에서 명백하게 다른 내용을 지시하지 않는 "한"과, "상기"와 같은 단수 표현들은 복수 표현들을 포함한다는 것이 이해될 수 있을 것이다. 따라서, 일 예로, "컴포넌트 표면(component surface)"은 하나 혹은 그 이상의 컴포넌트 표현들을 포함한다.

[0021] 또한, 제1, 제2 등과 같이 서수를 포함하는 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되지는 않는다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 개시의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0022] 또한, 본 명세서에서 사용한 용어는 단지 특정한 실시 예를 설명하기 위해 사용된 것으로, 본 개시를 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0023] 또한, 별도로 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 이해되어야만 한다.

[0024] 먼저, 본 개시의 실시 예는 인터넷을 구성하는 중요한 프로토콜 중의 하나인 TCP를 개선하여 통신 시스템 상에서 데이터 패킷의 처리율에 대한 성능을 저하시키지 않으면서 어플리케이션 간의 지연 시간을 최소화하기 위한 방법을 제공하기 위한 것이다.

[0025] 이를 위하여 본 개시의 실시 예에서는 TCP의 데이터 패킷을 관리하는 전송 버퍼를 제어하는 방법 및 장치를 제공한다. 이하에서는 도면을 참조하여 통신 시스템에 포함된 각 장치에서 TCP의 전송 버퍼를 제어하는 방법에 대하여 상세하게 설명하기로 한다.

[0026] 도 1은 본 개시의 실시 예에 따른 통신 시스템의 일 예를 보이고 있다.

[0027] 도 1을 참조하면, 통신 시스템은 전송 장치(110), 중계 장치(130) 및 수신 장치(150)를 포함한다.

[0028] 도 1을 참조하면, 상기 전송 장치(110)는 데이터 패킷을 전송하는 무선 통신 장치이다. 그리고 상기 수신 장치(150)는 상기 전송 장치(110)와 무선으로 접속되어 전송 장치(110)로부터 데이터 패킷을 수신하는 무선 통신 장치이다. 또한 상기 중계 장치(130)는 동일한 전송 프로토콜을 사용하는 분리된 네트워크를 연결하는 장치로서, 상기 전송 장치(110)와 수신 장치(150)를 서로 연결하는 장치이다. 일 예로, 상기 중계 장치(130)는 프록시(proxy) 서버일 수 있다.

[0029] 상기 전송 장치(110)는 어플리케이션 계층으로부터 수신된 데이터 패킷을 수신 장치(150)로 전송하기 위하여 전송 버퍼를 관리한다. 그리고 상기 전송 장치(110)는 혼잡 윈도우(congestion window)를 이용하여 전송 버퍼에 버퍼링된 데이터 패킷을 상기 수신 장치(150)로 전송한다. 이때, 본 개시의 실시 예에서는 전송 장치(110)에서 관리하는 전송 버퍼를 네트워크 상황에 따라 적응적으로 제어하여, 전송 장치(110)에서 데이터 패킷을 처리하는 처리율에 대한 성능을 저하시키지 않으면서 어플리케이션 간 지연 시간을 최소화하는 방안을 제안한다.

[0030] 본 개시의 실시 예에서 전송 장치(110)에서 관리하는 전송 버퍼를 네트워크 상황에 따라 적응적으로 제어하는 방법은, 전송 장치(110), 중계 장치(130) 및 수신 장치(150) 중 적어도 하나의 장치에 의해 수행될 수 있다.

[0031] 먼저, 도 2a 및 도 2b를 기반으로 본 개시의 실시 예에 따른 전송 버퍼를 제어하는 방법이 상기 전송 장치(110)에 의해 수행되는 경우에 대하여 설명하기로 한다.

[0032] 도 2a 및 도 2b는 본 개시의 실시 예에 따른 통신 시스템에서 전송 장치(110)에 의해 전송 버퍼를 제어하는 방법을 보이고 있다. 이때, 본 개시의 실시 예에 따라 전송 장치(110)에서 전송 버퍼를 제어하는 경우, 상기 전송 버퍼는 전송 장치(110)에 포함되어 있으므로 상기 전송 장치(110)는 직접적으로 전송 버퍼의 크기를 제어할 수

있다.

[0033] 도 2a 및 도 2b를 참조하면, 상기 전송 장치(110)는 네트워크 환경을 확인하기 위하여, 현재의 RTT 값을 확인한다(201). 상기 현재의 RTT 값은 TCP에서 가장 최근에 측정된 RTT 값으로, 상기 현재의 RTT 값에 높은 가중치를 부여하여 나타낼 수 있다. 이와 같이 상기 현재의 RTT 값에 높은 가중치를 부여하여 나타내는 값은 스무스드 RTT(smoothed RTT) 값으로 칭해질 수 있다.

[0034] 그리고 상기 전송 장치(110)는 상기 확인한 smoothed RTT 값이 현재까지 확인된 최소 RTT(RTT_{min}) 값보다 작은 값을 갖는지 판단하여(203), 상기 확인한 smoothed RTT 값이 RTT_{min} 값보다 작은 값을 갖는 경우 RTT_{min} 값을 상기 확인한 smoothed RTT 값으로 갱신한다(205). 반면, 상기 전송 장치(110)는 상기 확인한 smoothed RTT 값이 RTT_{min} 값보다 큰 값을 갖는 경우 RTT_{min} 값을 갱신하지 않고 단계 207로 진행한다.

[0035] 이후, 상기 전송 장치(110)는 TCP의 현재 단계 및 smoothed RTT 값을 기반으로 전송 버퍼의 크기를 제어한다(207~215). 이때, TCP에 따른 혼잡 제어 방식에서 혼잡 상태에 따라 수행되는 단계들은 구분된 단계들은 슬로우 스타트(slow start) 단계 및 혼잡 회피(congestion avoidance) 단계를 포함한다. 잘 알려진 것처럼 상기 슬로우 스타트 단계는 혼잡 감지 시까지 송신 데이터량을 지수적으로 증가시키는(즉 혼잡 윈도우의 크기를 지수적으로 증가시키는) 단계이다. 상기 혼잡 회피 단계는 상기 혼잡 감지 후 송신 데이터량를 선형적으로 증가시키는(즉 혼합 윈도우의 크기를 선형적으로 증가시키는) 단계이다.

[0036] 상세하게 상기 전송 장치(110)는 TCP의 현재 단계가 슬로우 스타트 단계인지에 판단한다(207). 이때, 상기 전송 장치(110)는 TCP의 현재 단계가 슬로우 스타트 단계인지를 판단하는 방법으로, 종래 다양한 방법들 중 하나의 방법을 이용할 수 있다. 일 예로, 상기 전송 장치(110)는 수신 장치(150)로부터 3회 이상 중복적인 정상 응답 신호(TCP acknowledgement)를 수신하기 이전까지를 슬로우 스타트 단계로 판단할 수 있다.

[0037] 상기 전송 장치(110)는 TCP의 현재 단계가 슬로우 스타트 단계인 경우, 전송 버퍼의 크기를 혼잡 윈도우의 크기에 일정 상수를 곱한 값으로 설정한다(209).

[0038] 반면, 상기 전송 장치(110)는 TCP의 현재 단계가 슬로우 스타트 단계가 아닌 경우로 판단한 경우, TCP의 현재 단계가 혼잡 회피 단계임을 확인한다(211). 상기 전송 장치(110)는 TCP의 현재 단계가 혼잡 회피(congestion avoidance) 단계임을 확인한 경우, 도 3을 기반으로 전송 버퍼의 크기를 제어할 수 있다. 도 3은 본 개시의 실시 예에서 RTT 값에 따른 TCP의 전송 버퍼의 크기 변화의 일 예를 보이고 있다. 즉, 상기 전송 장치(110)는 상기 smoothed RTT 값과 RTT_{min}의 차이 값을 고려하여 전송 버퍼의 크기를 조정한다. 이때, 상기 smoothed RTT 값과 RTT_{min}의 차이 값은 네트워크 상황에 따라 결정되는 변수 값, 미리 정해진 상수 값 또는 미리 정해진 정수 값 중 하나의 값이 될 수 있다.

[0039] 상세하게, 상기 전송 장치(110)는 상기 smoothed RTT 값이 RTT_{min}의 $1+\alpha$배보다 작은 경우(213), 도 3과 같이 전송 버퍼의 크기를 점차적으로(additive) 증가시킨다(215). 일 예로, 상기 전송 장치(110)는 상기 전송 버퍼의 크기를 1씩 증가시킬 수 있다. 그리고 상기 전송 장치(110)는 smoothed RTT 값이 $(1+\alpha)RTT_{\min}$와 $(1+\beta)RTT_{\min}$사이에 있는 경우(217), 도 3과 같이 기존의 전송 버퍼의 크기를 유지시킨다(219). 또한 상기 전송 장치(110)는 smoothed RTT 값이 $(1+\beta)RTT_{\min}$ 보다 큰 경우(221), 도 3과 같이 전송버퍼의 크기를 급격하게(multiplicative) 감소시킨다(223). 여기서, α는 β보다 작은 값을 가지며, 상기 α 및 β는 네트워크 상황에 따라 결정되는 변수 값, 미리 정해진 상수 값 또는 미리 정해진 정수 값 중 하나의 값이 될 수 있다.

[0040] 이때, Smoothed RTT 값의 증가는 해당 플로우의 전송 경로 상의 버퍼량의 증가, 플로우 간 경쟁 심화, 패킷 손실률의 증가에 따른 지연 시간 증가 등의 결과로 볼 수 있다. 따라서 Smoothed RTT 값이 증가하는 네트워크 환경에서는 전송 버퍼의 크기를 줄임으로써 어플리케이션이 감지하는 지연 시간을 줄여줄 필요가 있다. 그리고 smoothed RTT 값의 감소는 해당 플로우의 전송 경로 상의 버퍼량의 감소, 플로우 간 경쟁 감소, 패킷 손실률의 감소에 따른 지연 시간 감소 등의 결과로 볼 수 있다. 따라서 Smoothed RTT 값이 감소하는 네트워크 환경에서는 전송 버퍼의 크기를 늘림으로써 TCP의 전송 기회 발생 시 최대한의 데이터 패킷을 전송하도록 할 필요가 있다. 또한 전송 버퍼의 크기를 늘릴 때에는 점차적인 증가를 통해 갑작스런 증가로 인한 어플리케이션 지연 시간의 급격한 증가를 방지하고, 전송 버퍼를 감소 시킬 때에는 급격한 감소를 통해 악화된 네트워크 상황에 신속하게

대응하도록 한다.

[0041] 상기에서는 도 2a 및 도 2b를 참조하여, 전송 장치(110)에 의해 전송 버퍼의 크기를 직접적으로 제어하는 방법에 대하여 설명하였으며, 하기에서는 도 4를 기반으로 본 개시의 실시 예에 따른 전송 버퍼를 제어하는 방법이 수신 장치(150)에 의해 수행되는 경우에 대하여 설명하기로 한다.

[0042] 도 4는 본 개시의 실시 예에 따른 통신 시스템에서 수신 장치(150)에 의해 전송 버퍼를 제어하는 방법을 보이고 있다. 이때, 본 개시의 실시 예에 따라 수신 장치(130)에서 전송 버퍼를 제어하는 경우, 상기 전송 버퍼는 전송 장치(110)에 포함되어 있으므로 상기 수신 장치(150)는 간접적으로 전송 버퍼를 제어할 수 있다.

[0043] 도 4를 참조하면, 수신 장치(150)는 송신 장치(110)로 정상 응답 신호를 전송하는 경우의 RTT를 기반으로, TCP의 현재 단계를 예측한다(401).

[0044] 상기 TCP의 현재 단계를 예측하는 방법은, 일 예로 상기 수신 장치(150)가 송신 장치(110)로 정상 응답 신호를 전송한 경우, 상기 송신 장치(110)로부터 수신되는 패킷 트레인(packet train)의 길이가 단방향 지연 시간(일 예로, 최소 RTT 값의 절반 값)보다 큰 지를 확인한다. 여기서, 상기 패킷 트레인은 수신 장치(150)가 송신 장치(110)로부터 하나의 라운드(round)에서 수신하는 복수개의 데이터 패킷들의 묶음을 의미한다. 그리고 상기 수신 장치(150)는 상기 패킷 트레인 길이가 단방향 지연 시간보다 작거나 같으면 전송 장치(110)의 TCP가 슬로우 스타트 단계임으로 예측하고, 상기 패킷 트레인 길이가 단방향 지연 시간보다 크면 전송 장치(110)의 TCP가 혼잡 회피 단계임으로 예측한다. 이때, 상기 수신 장치(150)는 마지막으로 정상 응답 신호를 전송한 시점과 현재 시점 사이의 시간 간격이 일정 값보다 크면 새로운 라운드(즉, 새로운 패킷 트레인)으로 판단한다. 수신 장치(150)는 전송 장치(110)와 달리 정확히 패킷 트레인의 변경을 알 수 없기 때문에 이와 같은 방법을 사용한다.

[0045] 상기 TCP의 현재 단계를 예측하는 방법은, 다른 예로 상기 수신 장치(150)는 새로운 패킷 트레인을 수신할 때마다, 패킷 트레인의 처음 미리 정해진 개수의 정상 응답 신호들을 이용하여 RTT 값을 계산한다. 그리고 상기 수신 장치(150)는 k 번째 패킷 트레인에서 계산된 RTT 값이 k-1번째 패킷 트레인에서 계산된 RTT 값에 일정 상수를 더한 값보다 크면, 대역폭 지연 시간 프로덕트(bandwidth delay product : BDP)에 도달하여 전송 장치(110)의 TCP가 혼잡 회피 단계임으로 예측한다. 여기서, 상기 BDP는 대역폭과 지연 시간의 곱으로, 특정 경로를 통해 전송할 수 있는 데이터 패킷의 양의 최대 값을 나타낸다.

[0046] 반면, 상기 상기 수신 장치(150)는 k 번째 패킷 트레인에서 계산된 RTT 값이 k-1번째 패킷 트레인에서 계산된 RTT 값에 일정 상수를 더한 값보다 작으면 전송 장치(110)의 TCP가 슬로우 스타트임으로 예측한다. 한편, 상기 수신 장치(150)는 k 번째 패킷 트레인에서 계산된 RTT 값이 k-1번째 패킷 트레인에서 계산된 RTT 값에 일정 상수를 더한 값과 같은 경우, 내부 설정에 따라서 전송 장치(110)의 TCP가 혼잡 회피 단계 또는 슬로우 스타트 단계임으로 예측할 수 있다.

[0047] 다음으로, 상기 수신 장치(150)는 TCP의 데이터 패킷이 수신될 때마다, 수신된 데이터 패킷의 길이를 알려진(advertise) MSS(maximum segment size)로 나눈 값을 더하여 전송 장치(110)에서 결정된 혼잡 윈도우의 크기를 예측한다(403).

[0048] 그리고 수신 장치(150)는 새로운 패킷 트레인이 수신될 때마다, 현재의 RTT(smoothed RTT) 값을 확인한다(405). 상기 수신 장치(150)는 상기 확인한 smoothed RTT 값이 현재까지 확인된 최소 RTT(RTT_{min}) 값보다 작은 값을 갖는지 판단하여(407), 상기 확인한 smoothed RTT 값이 RTT_{min} 값보다 작은 값을 갖는 경우 RTT_{min} 값을 상기 확인한 smoothed RTT 값으로 갱신한다(409). 반면, 상기 수신 장치(150)는 상기 확인한 smoothed RTT 값이 RTT_{min} 값보다 큰 값을 갖는 경우 RTT_{min} 값을 갱신하지 않고 단계 411로 진행한다.

[0049] 상기 수신 장치(150)는 상기 예측된 TCP의 현재 단계(즉, 슬로우 스타트 단계 또는 혼잡 회피 단계), 상기 예측된 혼잡 윈도우의 크기, 상기 확인한 smoothed RTT 값 및 RTT_{min} 값을 기반으로, 송신 장치(110)로 전송할(advertisement) 윈도우의 크기를 결정한다. 이때, 상기 수신 장치(150)는 상기 송신 장치(110)에서 전송 버퍼의 크기를 결정하는 방법과 유사한 방법으로 advertisement 윈도우의 크기를 결정할 수 있다. 즉, 상기 수신 장치(150)는 상기 예측된 TCP의 현재 단계가 슬로우 스타트 단계인 경우, 상기 예측된 혼잡 윈도우의 크기에 일정 상수를 곱하여 advertisement 윈도우의 크기를 결정한다. 그리고 상기 수신 장치(150)는 상기 예측된 TCP의 현재 단계가 혼잡 회피 단계인 경우, 도 2의 단계 211 내지 223과 동일한 방법으로, 도 3을 기반으로 상기 예측된 혼잡 윈도우의 크기, 상기 확인한 smoothed RTT 값 및 RTT_{min} 값을 이용하여 advertisement 윈도우의 크기를 결

정할 수 있다.

[0050] 그리고 상기 수신 장치(150)는 상기 결정된 advertisement 윈도우의 크기에 미리 정해진 윈도우 스케일(scale)을 적용하여 최종 advertisement 윈도우의 크기를 결정한다(413). 따라서 상기 수신 장치(150)는 패킷 헤더의 advertisement 윈도우 필드에 상기 최종 advertisement 윈도우의 크기를 삽입하여 상기 송신 장치(110)로 전송한다(415).

[0051] 이로부터 상기 수신 장치(150)는 전송 버퍼로부터 출력되는 윈도우의 크기를 결정하여 간접적으로 전송 버퍼를 제어할 수 있다.

[0052] 상기에서는 도 4를 기반으로 상기 수신 장치(150)에 의해 전송 버퍼를 간접적으로 제어하는 방법에 대하여 설명하였으며, 이하에서는 도 5를 기반으로 본 개시의 실시 예에 따른 전송 버퍼를 제어하는 방법이 중계 장치(130)에 의해 수행되는 경우에 대하여 설명하기로 한다. 본 개시의 실시 예에 따른 통신 시스템에서는 송신 장치(110) 및 수신 장치(150)의 내부 구성 및 동작을 변경하지 않고, 중계 장치(130)에 의해 전송 버퍼를 제어할 수 있다

[0053] 도 5는 본 개시의 실시 예에 따른 통신 시스템에서 중계 장치(130)에 의해 전송 버퍼를 제어하는 방법을 보이고 있다.

[0054] 도 5를 참조하면, 중계 장치(130)는 전송 장치(110)와 통신을 수행하는 경우 수신 장치로 동작하고, 수신 장치(150)와 통신을 수행하는 경우 송신 장치로 동작할 수 있다. 따라서, 상기 중계 장치(130)는 전송 장치(110)와 통신을 수행하는 경우 도 4와 같이 수신 장치(150)와 동일한 방법(즉, 제1 전송 버퍼 제어 프로토콜, 531)으로 간접적으로 전송 장치(110)에 포함된 전송 버퍼(501)를 제어한다. 그리고 중계 장치(130)는 수신 장치(150)와 통신을 수행하는 경우 도 2a 및 도 2b와 같이 전송 장치(110)와 동일한 방법(즉, 제2 전송 버퍼 제어 프로토콜, 533)으로 직접적으로 중계 장치(130)에 포함된 전송 버퍼(503)를 제어한다. 이때, 상기 중계 장치(130)는 제1 전송 버퍼 제어 프로토콜과 제2 전송 버퍼 제어 프로토콜을 모두 수행함에 따라, 이에 필요한 추가적인 동작을 수행할 수도 있다. 상기 추가적인 동작의 예로는 중계 장치(130)의 내부에서 제1 전송 버퍼 제어 프로토콜에 의해 관리되는 수신 버퍼와 제2 전송 버퍼 제어 프로토콜에 의해 관리되는 송신 버퍼 간의 패킷 전달에서 오버플로우(overflow)로 인한 패킷 손실을 방지하기 위한 내부 버퍼를 제어하는 동작 등이 있다.

[0055] 따라서, 본 개시의 실시 예는 전송 장치(110)에 포함된 전송 버퍼(501) 또는 중계 장치(130)에 포함된 전송 버퍼(503)를 네트워크 상황에 따라 적응적으로 제어하여 데이터 패킷에 대한 처리율 성능을 저하시키지 않으면서 어플리케이션 간 지연 시간을 최소화할 수 있다. 그리고 상기 본 개시의 실시 예에서 어플리케이션 간 지연 시간을 최소화할 수 있음은 지연 시간이 중요한 다양한 어플리케이션의 서비스 품질을 향상시키는데 활용될 수 있다. 또한 본 개시의 실시 예는 전송 장치(110), 수신 장치(130) 및 중계 장치(150) 중 하나의 장치에 의해 수행될 수 있으므로, 다양한 통신 시스템의 환경에서 적용될 수 있다.

[0056] 상기에서는 도 5를 기반으로 상기 중계 장치(130)에 의해 전송 버퍼를 직접적 및 간접적으로 제어하는 방법에 대하여 설명하였으며, 이하에서는 도 6 내지 도 8을 기반으로 본 개시의 실시 예를 수행하는 송신 장치(110), 수신 장치(150) 및 중계 장치(130) 각각의 내부 구성을 간략히 설명하기로 한다.

[0057] 도 6은 본 개시의 실시 예에 따른 통신 시스템에서 TCP의 전송 버퍼를 제어하는 전송 장치(110)의 내부 구성을 보이고 있다.

[0058] 도 6을 참조하면, 전송 장치(110)는, 제어부(601), 송신부(603), 수신부(605) 및 저장부(607)를 포함한다.

[0059] 상기 제어부(601)는 전송 장치(110)의 전반적인 동작을 제어하며, 특히 본 개시의 실시 예들에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작을 제어한다. 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작은 상기 도 1 내지 도 3에서 설명한 바와 동일하므로 여기서 그 상세한 설명을 생략하기로 한다. 그리고 상기 제어부(601)는 상기 도 1 내지 도 3 각각에서 설명한 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작을 각각 또는 적어도 둘 이상의 실시 예들을 조합하여 TCP의 전송 버퍼를 제어할 수 있다.

[0060] 상기 송신부(603)는 상기 제어부(601)의 제어에 따라 통신 시스템에 포함되는 다른 엔터티들로부터 각종 신호 및 각종 메시지들을 수신한다. 여기서, 상기 송신부(603)가 수신하는 각종 신호 및 각종 메시지들은 상기 도 2a 및 도 2b에서 설명한 바와 동일하므로 여기서 그 상세한 설명을 생략하기로 한다.

[0061] 또한 상기 수신부(605)는 제어부(601)의 제어에 따라 통신 시스템에 포함되는 다른 엔터티들로부터 각종 신호 및 각종 메시지들을 수신한다. 여기서, 상기 수신부(605)가 수신하는 각종 신호 및 각종 메시지들은 상기 도 2a

및 도 2b에서 설명한 바와 동일하므로 여기서는 그 상세한 설명을 생략하기로 한다.

[0062] 상기 저장부(607)는 상기 제어기(601)의 제어에 따라 상기 송신 장치(110)가 수행하는 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작과 관련된 프로그램과 각종 데이터 패킷 등을 저장한다. 또한, 상기 저장부(607)는 상기 수신부(605)가 상기 다른 엔터티들로부터 수신한 각종 신호 및 각종 메시지들을 저장한다.

[0063] 한편, 도 6에는 상기 송신 장치(110)가 상기 제어부(601), 송신부(603), 수신부(605) 및 저장부(607)와 같이 별도의 유닛들로 구현된 경우가 도시되어 있으나, 상기 송신 장치(110)는 상기 제어부(601), 송신부(603), 수신부(605) 및 저장부(607) 중 적어도 두 개가 통합된 형태로 구현 가능함은 물론이다. 또한, 상기 송신 장치(110)는 1개의 프로세서로 구현될 수도 있음은 물론이다.

[0064] 도 7은 본 개시의 실시 예에 따른 통신 시스템에서 TCP의 전송 버퍼를 제어하는 수신 장치(150)의 내부 구성을 보이고 있다.

[0065] 도 7을 참조하면, 수신 장치(150)는, 제어부(701), 송신부(703), 수신부(705) 및 저장부(707)를 포함한다.

[0066] 상기 제어부(701)는 수신 장치(150)의 전반적인 동작을 제어하며, 특히 본 개시의 실시 예들에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작을 제어한다. 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작은 상기 도 4에서 설명한 바와 동일하므로 여기서 그 상세한 설명을 생략하기로 한다.

[0067] 상기 송신부(703)은 상기 제어부(701)의 제어에 따라 통신 시스템에 포함되는 다른 엔터티들로부터 각종 신호 및 각종 메시지들을 수신한다. 여기서, 상기 송신부(703)가 수신하는 각종 신호 및 각종 메시지들은 상기 도 1, 도 3 및 도 4에서 설명한 바와 동일하므로 여기서 그 상세한 설명을 생략하기로 한다. 그리고 상기 제어부(601)는 상기 도 1, 도 3 및 도 4 각각에서 설명한 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작을 각각 또는 적어도 둘 이상의 실시 예들을 조합하여 TCP의 전송 버퍼를 제어할 수 있다.

[0068] 또한 상기 수신부(705)는 제어부(701)의 제어에 따라 통신 시스템에 포함되는 다른 엔터티들로부터 각종 신호 및 각종 메시지들을 수신한다. 여기서, 상기 수신부(705)가 수신하는 각종 신호 및 각종 메시지들은 상기 도 4에서 설명한 바와 동일하므로 여기서는 그 상세한 설명을 생략하기로 한다.

[0069] 상기 저장부(707)는 상기 제어기(701)의 제어에 따라 상기 수신 장치(150)가 수행하는 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작과 관련된 프로그램과 각종 데이터 패킷 등을 저장한다. 또한, 상기 저장부(707)는 상기 수신부(705)가 상기 다른 엔터티들로부터 수신한 각종 신호 및 각종 메시지들을 저장한다.

[0070] 한편, 도 7에는 상기 수신 장치(150)가 상기 제어부(701), 송신부(703), 수신부(705) 및 저장부(707)와 같이 별도의 유닛들로 구현된 경우가 도시되어 있으나, 상기 수신 장치(150)는 상기 제어부(701), 송신부(703), 수신부(705) 및 저장부(707) 중 적어도 두 개가 통합된 형태로 구현 가능함은 물론이다. 또한, 상기 수신 장치(10)는 1개의 프로세서로 구현될 수도 있음은 물론이다.

[0071] 도 8은 본 개시의 실시 예에 따른 통신 시스템에서 TCP의 전송 버퍼를 제어하는 중계 장치(130)의 내부 구성을 보이고 있다.

[0072] 도 8을 참조하면, 중계 장치(130)는, 제어부(801), 송신부(803), 수신부(805) 및 저장부(807)를 포함한다.

[0073] 상기 제어부(801)는 중계 장치(130)의 전반적인 동작을 제어하며, 특히 본 개시의 실시 예들에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작을 제어한다. 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작은 상기 1 내지 도 5에서 설명한 바와 동일하므로 여기서 그 상세한 설명을 생략하기로 한다. 상기 제어부(601)는 상기 도 1 내지 도 5 각각에서 설명한 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작을 각각 또는 적어도 둘 이상의 실시 예들을 조합하여 TCP의 전송 버퍼를 제어할 수 있다.

[0074] 상기 송신부(803)은 상기 제어부(801)의 제어에 따라 통신 시스템에 포함되는 다른 엔터티들로부터 각종 신호 및 각종 메시지들을 수신한다. 여기서, 상기 송신부(803)가 수신하는 각종 신호 및 각종 메시지들은 상기 도 2a 및 도 2b 및 도 4에서 설명한 바와 동일하므로 여기서 그 상세한 설명을 생략하기로 한다.

[0075] 또한 상기 수신부(805)는 제어부(801)의 제어에 따라 통신 시스템에 포함되는 다른 엔터티들로부터 각종 신호 및 각종 메시지들을 수신한다. 여기서, 상기 수신부(805)가 수신하는 각종 신호 및 각종 메시지들은 상기 도 2a 및 도 2b 및 도 4에서 설명한 바와 동일하므로 여기서는 그 상세한 설명을 생략하기로 한다.

[0076] 상기 저장부(807)는 상기 제어기(801)의 제어에 따라 상기 중계 장치(130)가 수행하는 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 동작에 관련된 동작과 관련된 프로그램과 각종 데이터 패킷 등을 저장한다. 또한, 상기 저장부(807)는 상기 수신부(805)가 상기 다른 엔터티들로부터 수신한 각종 신호 및 각종 메시지들을 저장한다.

[0077] 한편, 도 8에는 상기 중계 장치(130)가 상기 제어부(801), 송신부(803), 수신부(805) 및 저장부(807)와 같이 별도의 유닛들로 구현된 경우가 도시되어 있으나, 상기 중계 장치(130)는 상기 제어부(801), 송신부(803), 수신부(805) 및 저장부(807) 중 적어도 두 개가 통합된 형태로 구현 가능함은 물론이다. 또한, 상기 중계 장치(130)는 1개의 프로세서로 구현될 수도 있음은 물론이다.

[0078] 상기에서는 도 6 내지 도 8을 기반으로 본 개시의 실시 예를 수행하는 송신 장치(110), 수신 장치(150) 및 중계 장치(130) 각각의 내부 구성에 대하여 설명하였으며, 하기에서는 도 9 내지 도 11을 기반으로 실제 통신 시스템에서 본 개시의 실시 예에 따라 통신 시스템에서 TCP의 송신 버퍼를 제어하는 방법을 적용한 결과에 대하여 설명하기로 한다.

[0079] 도 9는 현재 통신 시스템의 TCP에서 지연 시간이 발생되는 장치를 확인한 결과를 보이고 있다.

[0080] 도 9를 참조하면, (a)는 큐빅(cubic)에서 지연 시간이 발생되는 장치를 확인한 결과이고, (b)는 리노(reno)에서 지연 시간이 발생되는 장치를 확인한 결과이다. 도 9의 (a) 및 (b)에서는 모두 대부분의 지연 시간이 전송 장치(110)의 TCP 전송 버퍼에서 발생함을 확인할 수 있다. 그리고 도 9의 (a)에서와 같이, 큐빅의 경우 중계 장치(130)(일 예로, 프록시)의 TCP 전송 버퍼에서도 큰 지연 시간이 발생함을 확인할 수 있다. 이러한 통신 환경에서 송신 장치(110), 중계 장치(130) 및 수신 장치(150) 각각에 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용하면, 도 10 및 도 11과 같이 TCP에서의 지연 시간 및 처리율 성능을 최대화할 수 있다.

[0081] 도 10은 통신 시스템에서 송신 장치(110) 및 수신 장치(150)에 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과를 보이고 있다. 여기서, 도 10의 결과는 실제로 현재 통신 사업자에 의해 운용되고 있는 이동통신 네트워크들 중 한 네트워크에서 실제로 측정한 것이다.

[0082] 도 10을 참조하면, (a)는 기존 큐빅에서의 지연 시간, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간을 비교한 것이다. 도 10의 (a)에서는 기존 큐빅에서의 지연 시간에 비하여, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간이 확연히 감소하였음을 확인할 수 있다.

[0083] 그리고 (b)는 기존 큐빅에서의 평균 처리율, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율을 비교한 것이다. 도 10의 (a)에서는 기존 큐빅에서의 평균 처리율과, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율이 거의 유사함을 확인할 수 있다.

[0084] 이로부터, 본 개시의 실시 예에 따른 전송 장치(110)에서 TCP의 전송 버퍼를 제어하는 방법 및 본 개시의 실시 예에 따른 수신 장치(150)에서 TCP의 전송 버퍼를 제어하는 방법 중 하나의 방법을 이용하면, 통신 시스템에서 기존 TCP에 비하여 지연 시간을 최소화하고 평균 처리율을 유지시킬 수 있다.

[0085] 도 11은 통신 시스템에서 송신 장치(110), 중계 장치(130) 및 수신 장치(150)에 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과를 보이고 있다. 여기서, 도 11의 결과는 실제로 한국의 이동통신망 중 하나인 SK 텔레콤 네트워크에서 측정한 것이다.

[0086] 도 11을 참조하면, (a)는 기존 큐빅에서의 지연 시간, 중계 장치(130)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간을 비교한 것이다. 도 11의 (a)에서는 기존 큐빅에서의 지연 시간에 비하여, 중계 장치(130)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한

결과의 지연 시간 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 지연 시간이 확연히 감소하였음을 확인할 수 있다.

[0087] 그리고 (b)는 기존 큐빅에서의 평균 처리율, 중계 장치(130)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율을 비교한 것이다. 도 10의 (a)에서는 기존 큐빅에서의 평균 처리율과, 중계 장치(130)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율, 전송 장치(110)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율 및 수신 장치(150)에서 본 개시의 실시 예에 따른 TCP의 전송 버퍼를 제어하는 방법을 적용한 결과의 평균 처리율이 거의 유사함을 확인할 수 있다.

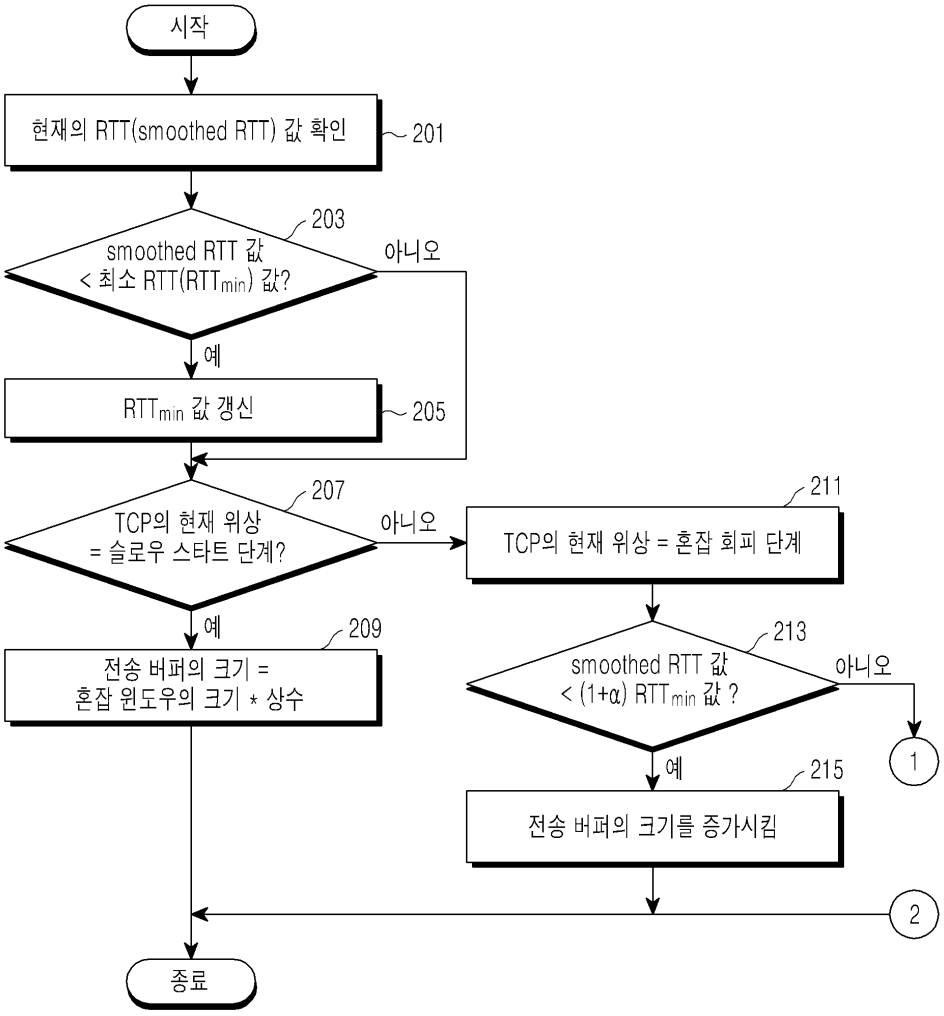
[0088] 이로부터, 본 개시의 실시 예에 따른 전송 장치(110)에서 TCP의 전송 버퍼를 제어하는 방법, 중계 장치(130)에서 TCP의 전송 버퍼를 제어하는 방법 및 본 개시의 실시 예에 따른 수신 장치(150)에서 TCP의 전송 버퍼를 제어하는 방법 중 하나의 방법을 이용하면, 통신 시스템에서 기존 TCP에 비하여 지연 시간을 최소화하고 평균 처리율을 유지시킬 수 있다.

[0089] 한편 본 개시의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 개시의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 개시의 범위는 설명된 실시 예에 국한되어 정해져서는 안되며 후술하는 특허 청구의 범위뿐만 아니라 이 특허 청구의 범위와 균등한 것들에 의해 정해져야 한다.

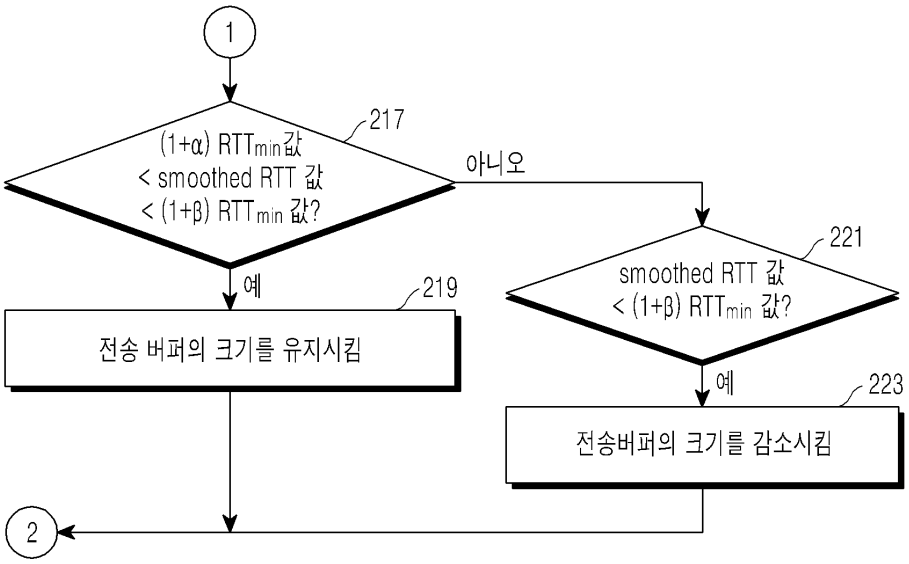
도면

도면1

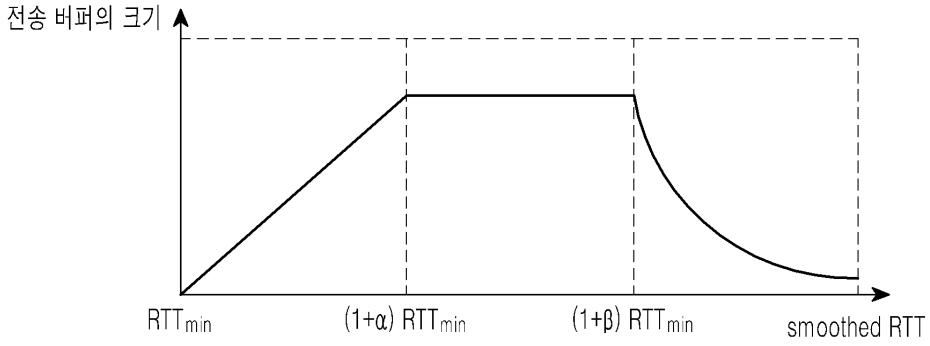
도면2a


시작
현재의 RTT(smoothed RTT) 값 확인
201
203
smoothed RTT 값
< 최소 RTT(RTTmin) 값?
아니오
예
RTTmin 값 갱신
205
207
TCP의 현재 위상
= 슬로우 스타트 단계?
아니오
211
TCP의 현재 위상 = 혼잡 회피 단계
예
209
전송 버퍼의 크기 =
혼잡 윈도우의 크기 * 상수
213
smoothed RTT 값
< (1+α) RTTmin 값 ?
아니오
1
예
215
전송 버퍼의 크기를 증가시킴
2
종료

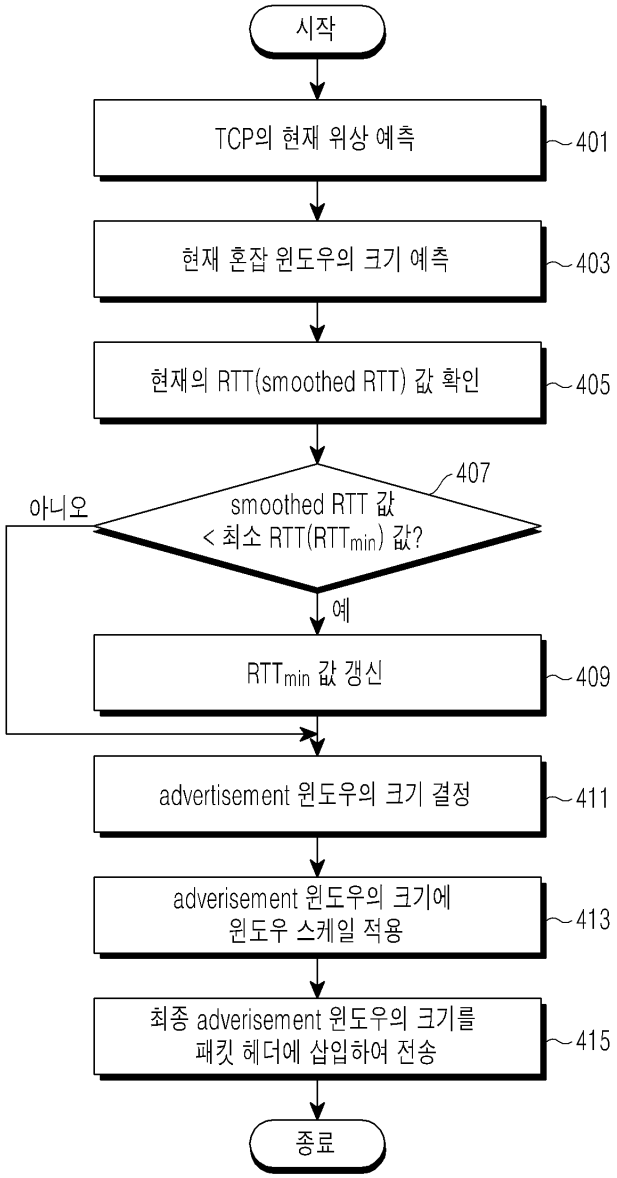
도면2b


1
217
(1+α) RTTmin값
< smoothed RTT 값
< (1+β) RTTmin 값?
아니오
221
smoothed RTT 값
< (1+β) RTTmin 값?
예
219
전송 버퍼의 크기를 유지시킴
예
223
전송버퍼의 크기를 감소시킴
2

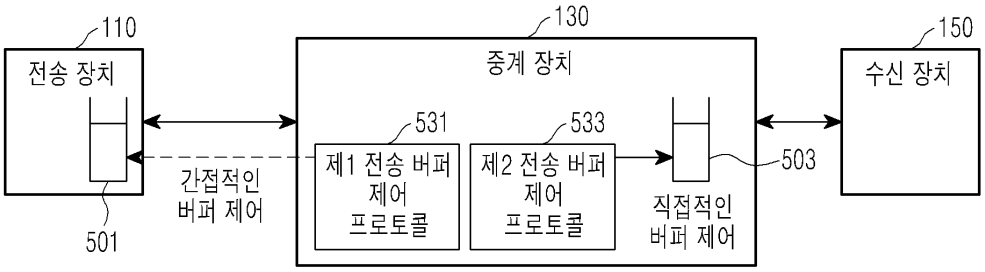
도면3


전송 버퍼의 크기
RTTmin
(1+α) RTTmin
(1+β) RTTmin
smoothed RTT

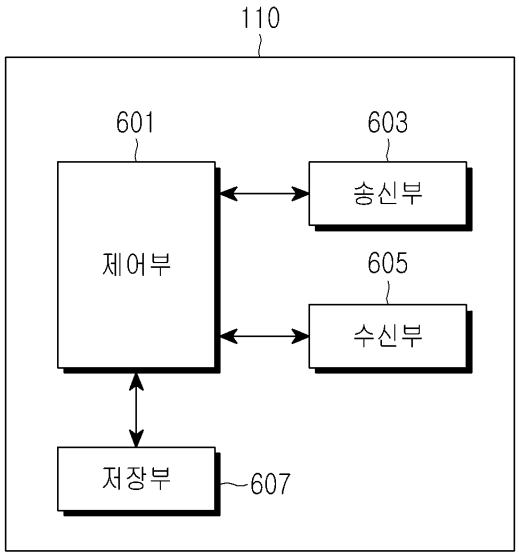
도면4


시작
TCP의 현재 위상 예측
401
현재 혼잡 윈도우의 크기 예측
403
현재의 RTT(smoothed RTT) 값 확인
405
407
아니오
smoothed RTT 값
< 최소 RTT(RTTmin) 값?
예
RTTmin 값 갱신
409
advertisement 윈도우의 크기 결정
411
adverisement 윈도우의 크기에
윈도우 스케일 적용
413
최종 adverisement 윈도우의 크기를
패킷 헤더에 삽입하여 전송
415
종료

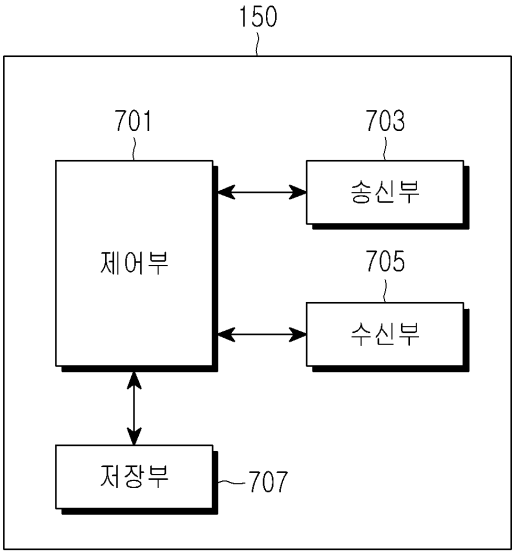
도면5


110
전송 장치
501
간접적인
버퍼 제어
130
중계 장치
531
제1 전송 버퍼
제어
프로토콜
533
제2 전송 버퍼
제어
프로토콜
503
직접적인
버퍼 제어
150
수신 장치

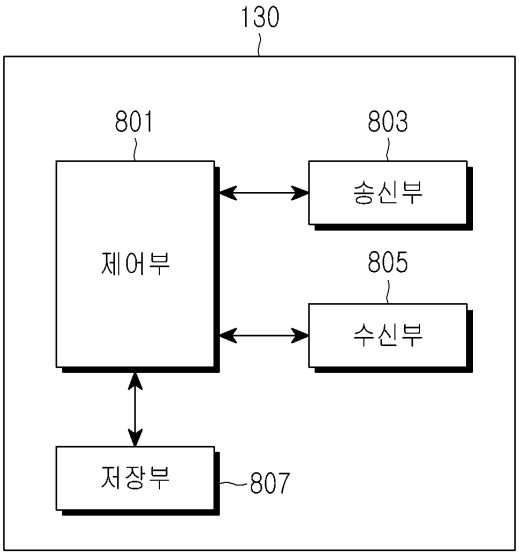
도면6


110
601
603
송신부
제어부
605
수신부
저장부
607

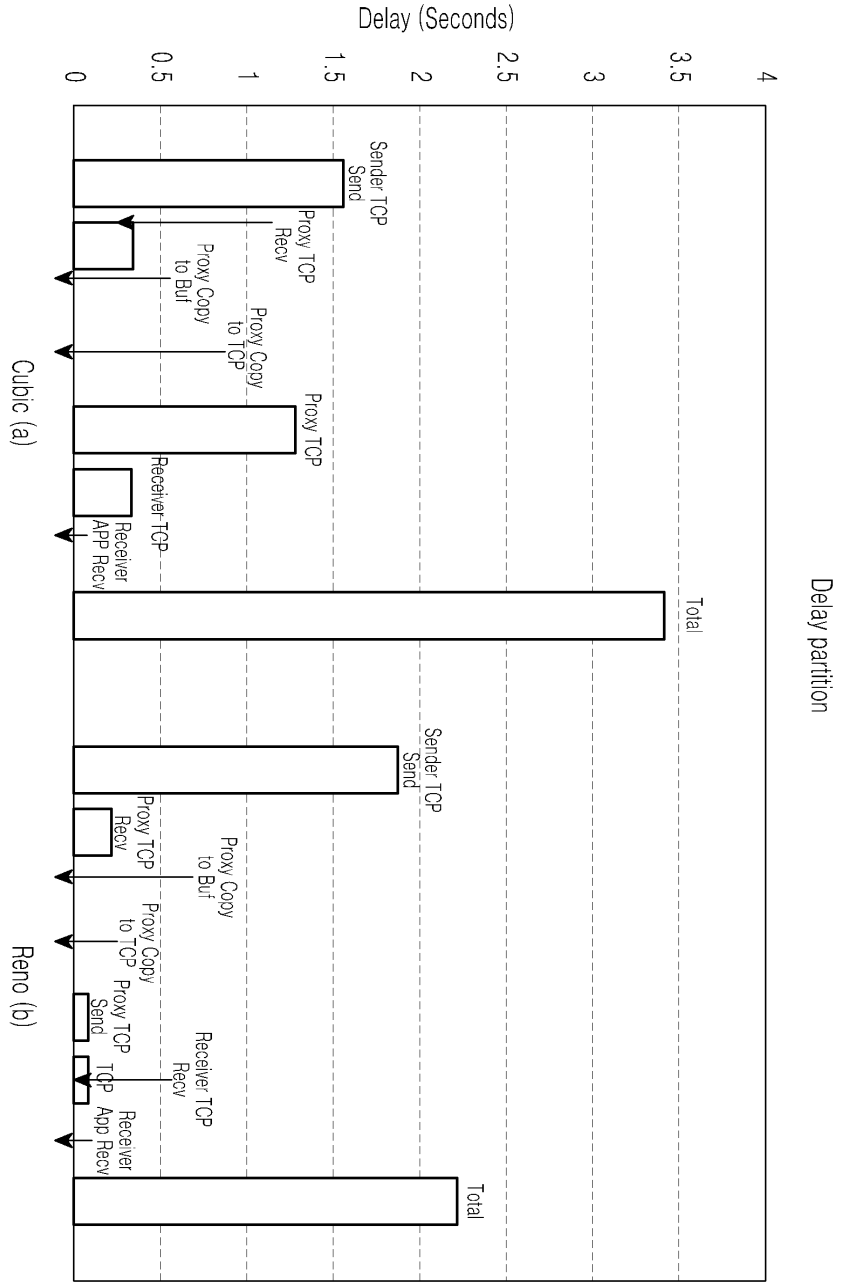
도면7


150
701
703
송신부
제어부
705
수신부
저장부
707

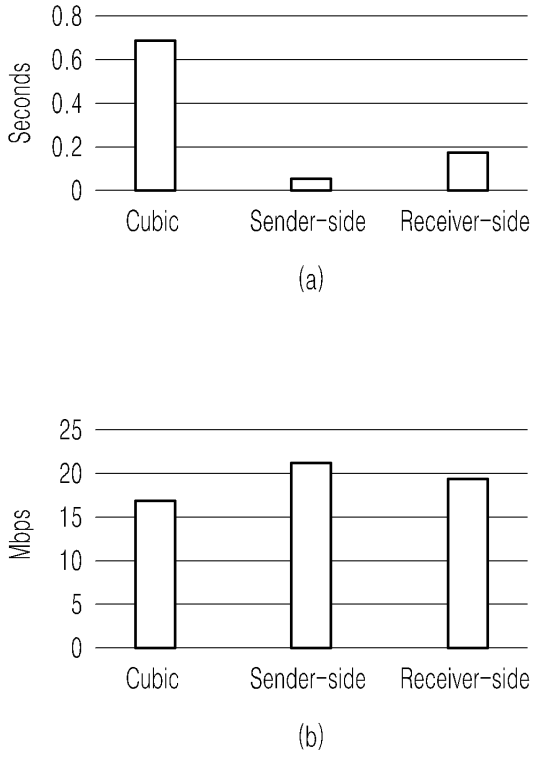
도면8



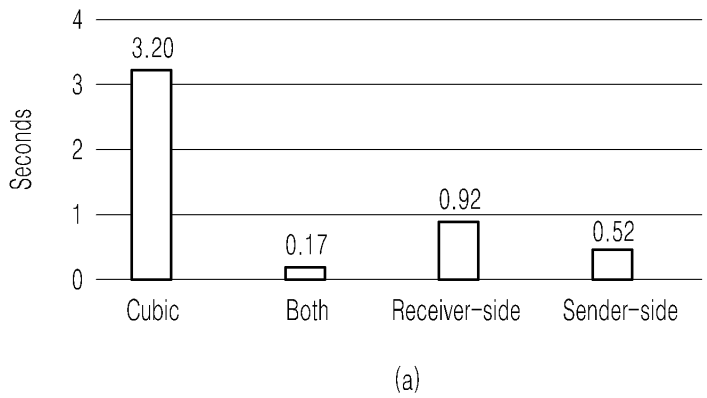
도면9


Delay (Seconds)
0
0.5
1
1.5
2
2.5
3
3.5
4
Delay partition
Cubic (a)
Sender TCP Send
Proxy TCP Recv
Proxy Copy to Buf
Proxy Copy to TCP
Proxy TCP
Receiver TCP
Receiver APP Recv
Total
Reno (b)
Sender TCP Send
Proxy TCP Recv
Proxy Copy to Buf
Proxy Copy to TCP
Proxy TCP Send
TCP
Receiver TCP Recv
Receiver App Recv
Total

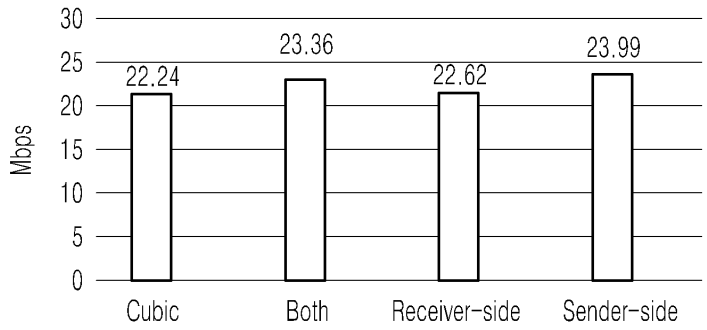
도면10


0.8
0.6
0.4
0.2
0
Seconds
Cubic
Sender-side
Receiver-side
(a)
25
20
15
10
5
0
Mbps
Cubic
Sender-side
Receiver-side
(b)

도면11


Seconds
4
3
2
1
0
3.20
0.17
0.92
0.52
Cubic
Both
Receiver-side
Sender-side

(a)


Mbps
30
25
20
15
10
5
0
22.24
23.36
22.62
23.99
Cubic
Both
Receiver-side
Sender-side

(b)