



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0122523
(43) 공개일자 2012년11월07일

(51) 국제특허분류(Int. Cl.)
H04W 28/06 (2009.01) H04L 12/56 (2006.01)
H04L 29/06 (2006.01)
(21) 출원번호 10-2011-0040726
(22) 출원일자 2011년04월29일
심사청구일자 2011년04월29일

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
장주욱
서울특별시 마포구 백범로 35, 전자공학과 R807호 (신수동, 서강대학교)
현도원
경기도 고양시 덕양구 화신로 76, 108동 1104호 (행신동, 샘터마을)
(74) 대리인
이지연

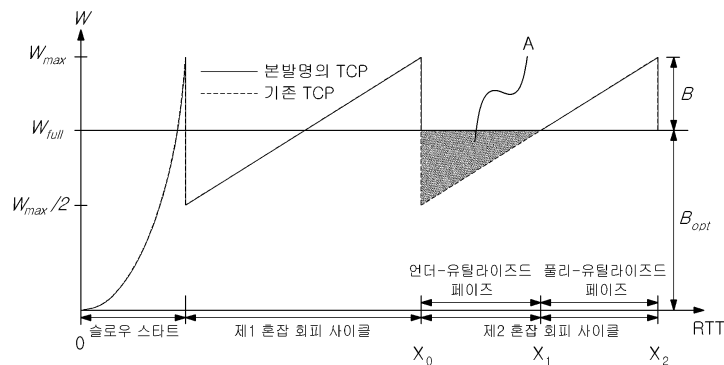
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 혼잡 윈도우 크기 조절 방법 및 그에 따른 TCP 시스템

(57) 요약

본 발명에 따르는 TCP를 위한 혼잡 윈도우 크기 조절 방법은, TCP 샌더가, 슬로우 스타트 페이즈에 따라 혼잡 윈도우 크기를 미리 정해둔 제1비율만큼씩 증가시키는 제1단계; 상기 TCP 샌더가, 상기 슬로우 스타트 페이즈에 따라 상기 혼잡 윈도우 크기를 증가하는 동안에, 미리 정해둔 수 이상의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 제2비율만큼 감소시켰다가, 다시 ACK 수신에 연동하여 상기 혼잡 윈도우를 미리 정해둔 제1 크기만큼씩 증가시키는 제2단계; 상기 TCP 샌더가, 상기 혼잡 윈도우 크기를 제2비율만큼 감소시킨 후에, 미리 정해둔 수의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 풀 혼잡 윈도우 크기로 설정하는 단계;를 구비함을 특징으로 한다.

대표도 - 도2



이 발명을 지원한 국가연구개발사업

과제고유번호 201032040

부처명 교육과학기술부

연구사업명 중견 연구자 지원사업

연구과제명 차세대 무선네트워크에서 IEEE 802.21기반 핸드오버시 TCP 성능

주관기관 서강대학교 산학협력단

연구기간 2010.05.01 ~ 2011.04.30

특허청구의 범위

청구항 1

TCP를 위한 혼잡 윈도우 크기 조절 방법에 있어서,

TCP 샌더가, 슬로우 스타트 페이지에 따라 혼잡 윈도우 크기를 미리 정해둔 제1비율만큼씩 증가시키는 제1단계;

상기 TCP 샌더가, 상기 슬로우 스타트 페이지에 따라 상기 혼잡 윈도우 크기를 증가하는 동안에, 미리 정해둔 수 이상의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 제2비율만큼 감소시켰다가, 다시 ACK 수신에 연동하여 상기 혼잡 윈도우를 미리 정해둔 제1크기만큼씩 증가시키는 제2단계;

상기 TCP 샌더가, 상기 혼잡 윈도우 크기를 제2비율만큼 감소시킨 후에, 미리 정해둔 수의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 풀 혼잡 윈도우 크기로 설정하는 제3단계;를 구비함을 특징으로 하는 TCP를 위한 혼잡 윈도우 크기 조절 방법.

청구항 2

제1항에 있어서,

상기 제2단계가,

상기 슬로우 스타트 페이지에 따라 상기 혼잡 윈도우 크기를 증가하는 동안에, 미리 정해둔 수 이상의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 제2비율만큼 감소시키는 제a1단계;

상기 제a1단계의 수행후에, 혼잡 윈도우 크기를 미리 정해둔 제1크기만큼 증가시키는 제a2단계;

상기 제a2단계의 수행후에, 현재 링크의 RTT가 미리 정해둔 RTTmin보다 크면 풀 혼잡 윈도우 크기를 미리 정해둔 풀 혼잡 윈도우 크기와 상기 혼잡 윈도우 크기 중 작은 값으로 재설정하는 제a3단계;

상기 제a3단계의 수행후에, ACK가 수신되면 상기 제a2단계로 복귀하여, 상기 혼잡 윈도우 크기를 미리 정해둔 제1크기만큼 증가되도록 하는 제a4단계;로 구성됨을 특징으로 하는 TCP를 위한 혼잡 윈도우 크기 조절 방법.

청구항 3

제1항에 있어서,

상기 제3단계가,

상기 혼잡 윈도우 크기를 제2비율만큼 감소시킨 후에, 미리 정해둔 수의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 제2비율로 감소시키는 제b1단계;

상기 제b1단계의 수행 후에, 현재의 RTT가 미리 정해둔 RTTmin보다 작거나 같은지 체크하는 제b2단계;

상기 제b2단계에서 현재의 RTT가 미리 정해둔 RTTmin보다 작거나 같으면, 상기 혼잡 윈도우 크기를 상기 풀 혼잡 윈도우 크기로 설정하는 제b3단계;

상기 제b2단계에서 현재의 RTT가 미리 정해둔 RTTmin보다 크면, 상기 혼잡 윈도우 크기를 미리 정해둔 크기만큼 증가시키는 제b4단계;

상기 제b2 및 제b3단계의 수행후에, ACK가 수신되면 상기 제b2단계로 복귀하는 제b5단계;

상기 제b2 및 제b3단계의 수행후에, 미리 정해둔 수의 ACK가 중복 수신되면, 상기 제1단계로 복귀하는 제b6단계;를 구비함을 특징으로 하는 TCP를 위한 혼잡 윈도우 크기 조절 방법.

청구항 4

TCP 시스템에 있어서,

TCP 샌더;와 TCP 리시버;로 구성되며,

상기 TCP 샌더가, 슬로우 스타트 페이지에 따라 혼잡 윈도우 크기를 미리 정해둔 제1비율만큼씩 증가시키고,

상기 슬로우 스타트 페이지에 따라 상기 혼잡 윈도우 크기를 증가하는 동안에, 미리 정해진 수 이상의 ACK가 상기 TCP 리시버로부터 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해진 제2비율만큼 감소시켰다가,

다시 ACK 수신에 연동하여 상기 혼잡 윈도우 크기를 미리 정해진 제1크기만큼씩 증가시키고, 상기 혼잡 윈도우 크기를 제2비율만큼 감소시킨 후에,

미리 정해진 수의 ACK가 상기 TCP 리시버로부터 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해진 풀 혼잡 윈도우 크기로 설정함을 특징으로 하는 TCP 시스템.

청구항 5

제4항에 있어서,

상기 TCP 샌더가,

상기 슬로우 스타트 페이지에 따라 상기 혼잡 윈도우 크기를 제1비율만큼씩 증가하는 동안에, 미리 정해진 수 이상의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해진 제2비율만큼 감소시키고, 혼잡 윈도우 크기를 미리 정해진 제1크기만큼 증가시키고,

상기 혼잡 윈도우 크기를 제1크기만큼 증가시킨 후에, 현재 링크의 RTT가 미리 정해진 최소 RTT보다 크면 풀 혼잡 윈도우 크기를 미리 정해진 풀 혼잡 윈도우 크기와 상기 혼잡 윈도우 크기 중 작은 값으로 재설정하고, 그 후에 ACK가 수신될 때마다 상기 혼잡 윈도우 크기를 미리 정해진 제3크기만큼 증가시킴을 특징으로 하는 TCP를 위한 혼잡 윈도우 크기 조절 방법.

청구항 6

제4항에 있어서,

상기 TCP 샌더가,

상기 혼잡 윈도우 크기를 제2비율만큼 감소시킨 후에, 미리 정해진 수의 ACK가 중복 수신될 때마다, 상기 혼잡 윈도우 크기를 미리 정해진 제2비율로 감소시키고,

상기 혼잡 윈도우 크기를 제2비율로 감소시킨 후에 ACK가 수신될 때마다 현재의 RTT가 미리 정해진 최소 RTT보다 작거나 같은지 체크하고, 현재의 RTT가 미리 정해진 최소 RTT보다 작거나 같으면, 상기 혼잡 윈도우 크기를 상기 풀 혼잡 윈도우 크기로 설정하고, 현재의 RTT가 미리 정해진 최소 RTT보다 크면, 상기 혼잡 윈도우 크기를 미리 정해진 제1크기만큼 증가시킴을 특징으로 하는 TCP 시스템.

명세서

기술분야

[0001] 본 발명은 TCP(Transmission Control Protocol)에 관한 것으로, 더욱 상세하게는 언더 버퍼드 링크에서 링크 사용과 처리량을 높이기 위한 혼잡 윈도우 크기 조절 방법 및 그에 따른 TCP 시스템에 관한 것이다.

배경기술

[0002] TCP Reno 등과 같은 종래 TCP는 혼잡(congestion)이 발생하면 네트워크 상황에 상관없이 현재 혼잡 윈도우(congestion window)의 크기를 절반으로 줄인다. 이와 같은 보수적인 혼잡 제어(congestion control)로 인해 언더 버퍼드 링크(under-buffered link)에서 링크 사용(link utilization)이 완전(100%)하게 이루어지지 않았다.

[0003] 소수의 롱 리브(long-lived) TCP 절차들에 대해, 링크를 완전하게 활용하기 위한 최적의 버퍼 크기(Bopt)는 룰 오브 씬(rule-of-thumb)에 의해 $B_{opt} = C \times RTT_{min}(\text{packets})$ 인 것으로 알려져 왔다. 여기서, 상기 C(packets/sec)는 링크 용량이고 상기 $RTT_{min}(\text{seconds})$ 은 RTT(Round Trip Time)의 최소값이다. 그리고 큐는 드롭 테일(drop-tail)을 사용한다.

[0004] 만일 현재의 버퍼 크기를 B라 할 때, 현재의 버퍼 크기(B)가 최적의 버퍼크기(Bopt)와 같거나 크다면($B \geq B_{opt}$), 링크는 완전(100%)하게 사용(fully utilize)되고 이때의 링크를 오버 버퍼드 링크(over-buffered link)라 한다.

- [0005] 그리고 현재의 버퍼 크기(B)가 최적의 버퍼크기(B_{opt})보다 작다면($B < B_{opt}$), 링크 사용은 완전(100%)하지 않으며, 이때의 링크를 언더 버퍼드 링크(under-buffered link)라 한다.
- [0006] 상기 링크 용량 C가 커질수록 최적의 버퍼크기(B_{opt})는 선형적으로 증가한다. 예를 들어 링크 용량 C가 1Tbps이고, RTT 최소값(RTT_{min})이 250ms인 경우에, 최적의 버퍼크기(B_{opt})는 250Gbits가 된다. 이에따라 고속의 SRAM을 가지고 링크 용량 증가에 따라 라우터의 버퍼를 구현하는데에는 어려움이 있었다. 또한 큐잉 딜레이(queueing delay)가 길어지거나 변화가 커지면, 혼잡 제어(congestion control) 알고리즘이 불안해 질 수 있었다.
- [0007] 도 1은 종래 기술에 따른 TCP에서의 혼잡 윈도우 크기의 변화와 링크 사용량, 버퍼 점유 상태를 예시한 것이다. 상기 도 1의 예에서 링크를 완전(100%)하게 사용할 때에 최적의 버퍼크기(B_{opt})는 1000(packets)이다. 도 1의 (a)에 도시한 점선은 풀 혼잡 윈도우 크기(W_{full})가 1000(packets)임을 나타낸다. 만일 혼잡 윈도우 크기(W)가 상기 점선 아래로 내려가면 도 1의 (b)의 링크 사용(link utilization) 역시 100% 아래로 내려간다. 이 언더 유틸라이즈드(under-utilization) 구간에서 버퍼는 도 1의 (c)에 도시한 바와 같이 비어있다. 그리고 혼잡 윈도우가 다시 증가하여 풀 혼잡 윈도우 크기(W_{full})를 넘게 되면 링크 사용(link utilization)은 100%가 되고 버퍼 크기도 커진다. 그리고 혼잡 윈도우 크기가 최대 혼잡 윈도우의 크기(W_{max}) 또는 풀 혼잡 윈도우 크기(W_{full})에 버퍼를 더한 크기, 즉 1250(packets)에 도달하게 되면, 링크와 버퍼는 모두 꽉 차게 되고 패킷 손실에 의해 3-dup ACKs가 발생하게 되며, 이 3-dup ACKs에 의해 혼잡 윈도우 크기가 최대 혼잡 윈도우 크기(W_{max})를 2로 나눈 값인 625(packets)로 줄어든다. 이때 상기 최대 혼잡 윈도우 크기(W_{max})를 2로 나눈 값은 풀 혼잡 윈도우 크기(W_{full})보다 작기 때문에 링크 사용(link utilization)은 다시 100% 아래로 내려가게 된다.
- [0008] 상술한 바와 같이 언더 버퍼드 링크에서 TCP Reno 등과 같은 기존 TCP는 링크를 완전(100%)하게 활용하지 못하는 구간이 있었으므로, 그 구간에서 링크를 완전하게 사용할 수 있게 하여, 링크 사용과 처리량을 높일 수 있는 새로운 혼잡 제어 기술의 개발이 절실하게 요망되었다.

발명의 내용

해결하려는 과제

- [0009] 본 발명은 TCP에서 최적 버퍼 크기보다 작은 링크인 언더 버퍼드 링크(under-buffered link)에서 링크 사용과 처리량을 높일 수 있는 TPC를 위한 혼잡 윈도우 크기 조절 방법 및 그에 따른 TCP 시스템을 제공하는 것을 그 목적으로 한다.

과제의 해결 수단

- [0010] 상기한 목적을 달성하기 위한 본 발명에 따르는 TCP를 위한 혼잡 윈도우 크기 조절 방법은, TCP 샌더가, 슬로우 스타트 페이지에 따라 혼잡 윈도우 크기를 미리 정해둔 제1비율만큼씩 증가시키는 제1단계; 상기 TCP 샌더가, 상기 슬로우 스타트 페이지에 따라 상기 혼잡 윈도우 크기를 증가하는 동안에, 미리 정해둔 수 이상의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 제2비율만큼 감소시켰다가, 다시 ACK 수신에 연동하여 상기 혼잡 윈도우를 미리 정해둔 제1크기만큼씩 증가시키는 제2단계; 상기 TCP 샌더가, 상기 혼잡 윈도우 크기를 제2비율만큼 감소시킨 후에, 미리 정해둔 수의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기를 미리 정해둔 풀 혼잡 윈도우 크기로 설정하는 단계;를 구비함을 특징으로 한다.

발명의 효과

- [0011] 상기한 본 발명은 TCP에서 최적 버퍼 크기보다 작은 링크인 언더 버퍼드 링크(under-buffered link)에서 링크 사용과 처리량을 높일 수 있는 효과가 있다.

도면의 간단한 설명

- [0012] 도 1은 종래의 TCP에 따른 혼잡 윈도우 크기 및 링크 사용, 버퍼 점유 변화를 예시한 도면.
 도 2는 본 발명의 바람직한 실시예에 따른 혼잡 윈도우 크기 변화를 예시한 도면.
 도 3은 본 발명의 바람직한 실시예에 따른 TCP 시스템의 구성도.
 도 4는 본 발명의 바람직한 실시예에 따른 TCP의 혼잡 윈도우 크기 조절 방법의 흐름도.
 도 5는 본 발명과 종래의 TCP의 성능을 비교한 도면.

발명을 실시하기 위한 구체적인 내용

- [0013] 본 발명의 바람직한 실시예에 따른 TCP를 도 2를 참조하여 설명한다.
- [0014] 본 발명의 TCP 샌더는 초기에 혼잡 윈도우 크기(W) 및 풀 혼잡 윈도우 크기(Wfull)를 미리 설정된 제1크기(1) 및 제2크기(a)로 설정한다.
- [0015] 이후 TCP 샌더는 슬로우 스타트(slow-start) 페이즈를 수행하여 혼잡 윈도우 크기(W)를 미리 설정된 제1비율($\times 2$)만큼씩 증가시키며, 상기 혼잡 윈도우 크기(W)를 슬로우 스타트 페이즈의 수행에 따라 증가하던 중에 미리 정해둔 수의 ACK가 중복수신되는 3-dup ACKs가 수신되면, 그때의 혼잡 윈도우 크기(Wmax)를 미리 정해둔 비율($\div 2$)로 감소시킨 후에 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제3크기(+1)만큼 증가시킨다.
- [0016] 또한 상기 TCP 샌더는 RTT가 증가하는 구간인지를 체크하고, 상기 RTT가 증가하는 구간이면 상기 풀 혼잡 윈도우 크기(Wfull)를 현재의 혼잡 윈도우 크기(W)와 풀 혼잡 윈도우 크기(Wfull) 중 작은 값으로 재설정한다. 이와 달리 상기 TCP 샌더는 RTT가 증가하는 구간이 아니면, 상기 풀 혼잡 윈도우 크기(Wfull)를 그대로 유지한다.
- [0017] 또한 상기 TCP 샌더는 풀 혼잡 윈도우 크기(Wfull)를 그대로 유지하거나 또는 재설정된 후에 다시 ACK 수신시마다 혼잡 윈도우 크기(W)를 증가시킨다. 이렇게 혼잡 윈도우 크기(W)를 증가하던 중에, 상기 TCP 샌더는 3-dup ACKs가 다시 수신되면, 혼잡 윈도우 크기(W)를 상기 풀 혼잡 윈도우 크기(Wfull)로 변경한다.
- [0018] 이와 같이 혼잡 윈도우 크기(W)를 풀 혼잡 윈도우 크기(Wfull)로 변경함으로써, 기존 TCP에는 언더 유틸라이즈드 페이즈였던 부분(A)에서도 링크가 100%로 사용될 수 있게 한다.
- [0019] 다만, 두번째 혼잡 회피 사이클의 첫 라운드에서는 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제2비율($\div 2$)로 감소시킬 수도 있으며, 이는 링크를 100% 사용하는 경우에 발생하는 버퍼 오버플로우(overflow) 문제를 해소하기 위한 것이다.
- [0020] 좀더 설명하면, 혼잡 윈도우 크기(W)가 풀 혼잡 윈도우 크기(Wfull)보다 커지면 버퍼 점유도 커진다. 그러나 RTT 역시 버퍼에 쌓인 패킷을 다 내보낼 수 있게 충분히 증가하기 때문에 다음 라운드에서 버퍼는 항상 비게 된다.
- [0021] 다시 말해, 첫 번째 혼잡 회피 사이클(congestion avoidance cycle) 마지막에서 혼잡 윈도우 크기(W)는 최대 혼잡 윈도우 크기(Wmax)이며, 버퍼 점유율을 B라고 둘 때에, 추가적인 패킷을 전송하기 전에 1 RTT를 기다린다면, 혼잡 윈도우 크기(W)는 최대 혼잡 윈도우 크기(Wmax)를 2로 나눈 값(Wmax/2)으로 남게 되며, 이 시간에 (Wfull-Wmax/2) 만큼의 패킷이 버퍼에서 내보내질 수 있다.
- [0022] 만약 (Wfull-Wmax/2)이 버퍼 크기(B)보다 크거나 같다면, 추가적인 패킷을 더 보내기 전에 버퍼는 비어져야 한다. 여기서 최대 혼잡 윈도우 크기(Wmax)는 풀 혼잡 윈도우 크기(Wfull)에 버퍼 크기(B)를 더한 것과 같고, 상기 풀 혼잡 윈도우 크기(Wfull)는 최적 버퍼 크기(Bopt)와 같으므로(Wmax=Wfull + B, Wfull = Bopt, 버퍼 크기(B)는 최적 버퍼 크기(Bopt)보다 작거나 같게(B $\leq$ Bopt) 된다. 이는 각 혼잡 회피 사이클(congestion avoidance cycle)의 시작에 있어 1 RTT면 버퍼를 비우는데 충분한 시간임을 나타낸다. 이러한 방식으로 버퍼 크기(B)가 최적 버퍼 크기(Bopt)를 2로 나눈 값보다 작거나 같다면(B $\leq$ Bopt/2), 2 RTT가 필요하다.
- [0023] 따라서 본 발명은 두번째 혼잡 회피 사이클의 첫번째 라운드에서는 혼잡 윈도우 크기(W)를 미리 정해둔 제2비율($\div 2$)로 감소시켰다가 그 다음 라운드에서부터는 혼잡 윈도우 크기(W)를 풀 혼잡 윈도우 크기(Wfull)로 설정한다.
- [0024] 또한 본 발명은 두번째 이후 혼잡 회피 사이클의 두번째 이후 라운드에서도 큐잉 딜레이에 의해 RTT가 커지면, 혼잡 윈도우 크기를 미리 정해둔 제3크기(+1)로 증가하여 링크 혼잡에 대응한다.
- [0025] <TCP 시스템>
- [0026] 본 발명의 바람직한 실시예에 따른 TCP 시스템의 구성을 도 2를 참조하여 설명한다.
- [0027] 상기 TCP 시스템은 TCP 샌더(100)와 TCP 리시버(102)와 중계기(104)로 구성된다. 상기 TCP 샌더(100)는 본 발명의 바람직한 실시예에 따라 언더 버퍼드 링크에서 링크 사용을 최대화하기 위해 혼잡 윈도우 크기(W)를 조절한다.

- [0028] 즉, 상기 TCP 샌더(100)는 슬로우 스타트 페이즈에 따라 혼잡 윈도우 크기(W)를 미리 정해둔 제1비율($\times 2$)만큼씩 증가시키고, 상기 슬로우 스타트 페이즈에 따라 상기 혼잡 윈도우 크기(W)를 증가하는 동안에, 미리 정해둔 수 이상의 ACK가 상기 TCP 리시버(102)로부터 중복 수신되면, 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제2비율($\div 2$)만큼 감소시켰다가, 다시 ACK 수신에 연동하여 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제3크기(+1)만큼씩 증가시키고, 상기 혼잡 윈도우 크기를 제2비율만큼 감소시킨 후에, 미리 정해둔 수의 ACK가 상기 TCP 리시버(102)로부터 중복 수신되면, 상기 혼잡 윈도우 크기(W)를 미리 정해둔 풀 혼잡 윈도우 크기(Wfull)로 설정한다.
- [0029] 좀더 상세히 설명하면, 상기 TCP 샌더(100)는 상기 슬로우 스타트 페이즈에 따라 상기 혼잡 윈도우 크기(W)를 제1비율($\times 2$)만큼씩 증가하는 동안에, 미리 정해둔 수 이상의 ACK가 중복 수신되면, 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제2비율($\div 2$)만큼 감소시키고, 혼잡 윈도우 크기(W)를 미리 정해둔 제3크기(+1)만큼 증가시키고, 상기 혼잡 윈도우 크기(W)를 제3크기(+1)만큼 증가시킨 후에, 현재 링크의 RTT가 미리 정해둔 최소 RTT보다 크면 풀 혼잡 윈도우 크기(Wfull)를 미리 정해둔 풀 혼잡 윈도우 크기(Wfull)와 상기 혼잡 윈도우 크기(W) 중 작은 값으로 재설정하고, 그후에 ACK가 수신될 때마다 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제3크기(+1)만큼 증가시킨다.
- [0030] 또한 상기 TCP 샌더(100)는, 상기 혼잡 윈도우 크기(W)를 제2비율($\div 2$)만큼 감소시킨 후에, 미리 정해둔 수의 ACK가 중복 수신될 때마다, 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제2비율($\div 2$)로 감소시킨다.
- [0031] 상기 혼잡 윈도우 크기(W)를 제2비율($\div 2$)로 감소시킨 후에, 상기 TCP 샌더(100)는 ACK가 수신될 때마다 현재의 RTT가 미리 정해둔 최소 RTT보다 작거나 같은지 체크하고, 현재의 RTT가 미리 정해둔 최소 RTT보다 작거나 같으면, 상기 혼잡 윈도우 크기(W)를 상기 풀 혼잡 윈도우 크기(Wfull)로 설정하고, 현재의 RTT가 미리 정해둔 최소 RTT보다 크면, 상기 혼잡 윈도우 크기(W)를 미리 정해둔 제3크기(+1)만큼 증가시킨다.
- [0032] <혼잡 윈도우 크기 조절 방법>
- [0033] 이제 상기한 TCP 시스템에 적용가능한 본 발명의 바람직한 실시예에 따른 TCP의 혼잡 윈도우 크기 조절 방법에도 4의 흐름도를 참조하여 상세히 설명한다.
- [0034] TCP 샌더(100)는 혼잡 윈도우 크기(W) 및 풀 혼잡 윈도우 크기(Wfull)을 미리 정해둔 제1크기 및 제2크기로 초기화한다(200단계). 여기서, 상기 제1크기는 1로, 상기 제2크기는 a일 수 있으며, 상기 a는 충분히 큰 값으로 정해진다.
- [0035] 이후 상기 TCP 샌더(100)는 TCP 리시버(102)로부터 ACK가 수신되면(202단계), 슬로우 스타트 페이즈를 수행하며, 그 슬로우 스타트 페이즈에 따라 상기 혼잡 윈도우 크기(W)를 미리 정해진 제1비율($\times 2$)로 증가한다(204단계).
- [0036] 이와 같이 TCP 리시버(102)로부터 ACK가 수신될 때마다 혼잡 윈도우 크기를 상기 제1비율로 증가하다가(202, 204 단계), 상기 TCP 리시버(102)로부터 3-dup ACKs가 수신되면(206단계), 즉 슬로우 스타트 페이즈 개시후 첫번째로 3-dup ACKs가 수신되면, 첫번째 혼잡 회피 사이클로 판단하여 페스트 리커버리(fast recovery)를 위해 상기 혼잡 윈도우 크기(100)를 미리 정해둔 제2비율($\div 2$)로 감소한다(208, 210단계).
- [0037] 이후 상기 TCP 샌더(100)는 혼잡 회피 절차를 수행하면서, 상기 혼잡 윈도우 크기를 미리 정해둔 제3크기(+1)만큼 증가시킨다(212단계). 여기서 상기 제3크기는 1로 정해질 수 있다.
- [0038] 이후 상기 TCP 샌더(100)는 현재 링크의 RTT를 검출하고, 그 RTT가 미리 정해진 최소 RTT(RTTmin)보다 큰지를 체크한다. 여기서 현재 링크의 RTT가 최소 RTT(RTTmin)보다 크면 RTT가 증가하는 구간으로 판단하여 상기 풀 혼잡 윈도우 크기(Wfull)를 현재의 혼잡 윈도우 크기(W)와 기설정된 풀 혼잡 윈도우 크기(Wfull) 중 작은 값으로 재설정한다.
- [0039] 이후 상기 TCP 샌더(100)는 ACK 수신시마다 상기 혼잡 윈도우(W)를 미리 정해둔 제3크기(+1)만큼 증가시킨다.
- [0040] 상기한 바와 같이 ACK 수신시마다 혼잡 윈도우(W)를 미리 정해둔 제3크기(+1)만큼 증가시키다가 3-dup ACKs가 수신되면(220단계), 즉 슬로우 스타트 페이즈 개시후 두번째로 3-dup ACKs가 수신되면, 상기 TCP 샌더(100)는 두번째 이후의 혼잡 회피 사이클에서 첫번째 라운드인지를 체크한다(208, 222단계). 즉 첫번째 혼잡 회피 사이클 이후 3-dup ACKs가 수신된 때인지를 체크한다.
- [0041] 상기 TCP 샌더(100)는 두번째 이후의 혼잡 회피 사이클에서 첫번째 라운드이면(222단계), 혼잡 링크의 버퍼를

다 비율 수 있도록 혼잡 윈도우 크기(W)를 미리 정해둔 제2비율($\div 2$)만큼 감소시킨다(224단계).

- [0042] 이와달리 상기 두번째 이후의 혼잡 회피 사이클에서 두번째 이후 라운드이면, 상기 TCP 샌더(100)는 현재 링크의 RTT가 미리 설정된 최소 RTT(RTTmin)보다 작거나 같은지를 체크한다(226단계).
- [0043] 상기 RTT가 최소 RTT(RTTmin)보다 작거나 같으면, 상기 TCP 샌더(100)는 혼잡 윈도우 크기(W)를 풀 혼잡 윈도우 크기(Wfull)로 변경하며(228단계), 그 이후 ACK 수신시마다(230단계) 다시 RTT가 RTTmin보다 작거나 같은지 체크하고, 여전히 상기 RTT가 RTTmin보다 작거나 같다면, 상기 TCP 샌더(100)는 혼잡 윈도우 크기(W)를 풀 혼잡 윈도우 크기(Wfull)로 유지한다.
- [0044] 그리고 상기 RTT가 최소 RTT(RTTmin)보다 크면(226단계), 상기 TCP 샌더(100)는 큐잉 딜레이로 인해 링크 혼잡이 발생한 것으로 판단하고, 혼잡 윈도우 크기(W)를 미리 정해둔 제3크기(+1)만큼 증가시키며(234단계), 그 이후 ACK 수신시마다(236단계) 다시 RTT가 최소 RTT(RTTmin)보다 작거나 같은지 체크하고, 여전히 상기 RTT가 최소 RTT(RTTmin)보다 크다면, 상기 TCP 샌더(100)는 혼잡 윈도우 크기(W)를 제3크기(+1)만큼 증가시키는 것을 반복한다.
- [0045] 또한 상기한 바와 같이 혼잡 윈도우 크기(W)를 풀 혼잡 윈도우 크기(Wfull)로 유지하거나 제3크기(+1)만큼 증가하다가 3-dup ACKs가 수신되면(238,232단계), 상기 TCP 샌더(100)는 다시 혼잡 윈도우 크기(W)를 미리 정해둔 제2비율($\div 2$)만큼 줄인 상태에서(222,224단계) 그 라운드를 보내고 다시 혼잡 윈도우 크기(W)를 증가시키는 과정을 반복한다(226~238단계).
- [0046] 상기한 실시예에서는 하나의 TCP 플로우에 대해서 설명하였으나, 여러 개의 플로우에 대해서도 동작할 수 있음은 본 발명에 의해 자명하다. 좀 더 설명하면, 일반적인 고속(high-speed) 네트워크에서는 동기화된 로스 모델(synchronized loss model)이 사용되므로, 플로우들이 하나의 병목 링크(bottleneck link)를 공유한다면, 이 플로우(flow)들은 거의 같은 시간에 패킷 손실을 겪게 되므로 플로우(flow)들의 톱니파(saw-tooth waves)는 동기화(synchronized)되고 인페이스(in-phase)된다. 이러한 경우에 동기화(synchronized)된 플로우(flow)들은 같은 버퍼가 요구되며, 플로우(flow)들은 링크와 버퍼를 공평하게 나누어 쓴다.
- [0047] 따라서 각각의 플로우가 가지고 있는 자원 안에서 본 발명의 TCP가 동작하기 때문에 여러 개의 플로우들에 대해서도 본 발명의 TCP는 동작이 가능하다.
- [0048] 상기한 바와 같은 본 발명의 성능을 평가하기 위해, 병목 링크의 용량 C=100 (Mbps), RTTmin = 80 (ms)인 토폴로지에 대해 성능을 평가하였다.
- [0049] TCP 샌더의 액세스 링크는 병목 링크의 용량 C보다 크고 RTTmin보다 빠르고, 링크를 풀리 유틸라이즈하는 풀 혼잡 윈도우 크기(Wfull)는 1000 (packets)으로 가정한다.
- [0050] 그리고 시뮬레이션에서는 버퍼 오버플로우를 피하기 위해 혼잡 제어에서 1 RTT를 기다리는 대신 풀 혼잡 윈도우 크기(Wfull)를 980(packets)으로 조금 낮게 설정하였다. 그리고 시뮬레이션은 ns-2를 사용하여 버퍼(B)가 B=250 및 k=1/4인 경우와 B=500, k=1/2인 경우에 대해서 시뮬레이션을 하였다.
- [0051] 기존의 TCP 1~3개의 플로우와 본 발명에 따른 TCP의 1~3개의 플로우에 대해서 링크 사용과 전송 시간에 대해서 성능을 평가하였으며, 그 결과, k=1/4와 k=1/2에 대해 본 발명에 따른 TCP가 10.77%와 3.74%의 성능 향상을 보였다. 또한 전송하고자 하는 데이터의 양을 5 Gbyte라고 두었을 경우에도 전송 시간에 있어 9.95%와 3.17%가 단축되었다.
- [0052] 따라서 언더 버퍼드 링크에서 본 발명에 따른 TCP는 많은 양의 데이터를 전송하는데 있어 유용하다.
- [0053] 도 5는 B= (B=500, k=1/2)에 대해 기존의 TCP와 본 발명에 따른 TCP간의 윈도우 크기, 링크 사용, 버퍼 점유 그리고 RTT의 변화를 도시한 것이다.
- [0054] 상기 도 5에서 두 번째 혼잡 회피 사이클부터 본 발명에 따른 TCP가 동작하여 도 5 (b)와 같이 언더 유틸라이즈드 페이즈에서 본 발명의 TCP가 링크를 거의 100% 사용한다.

부호의 설명

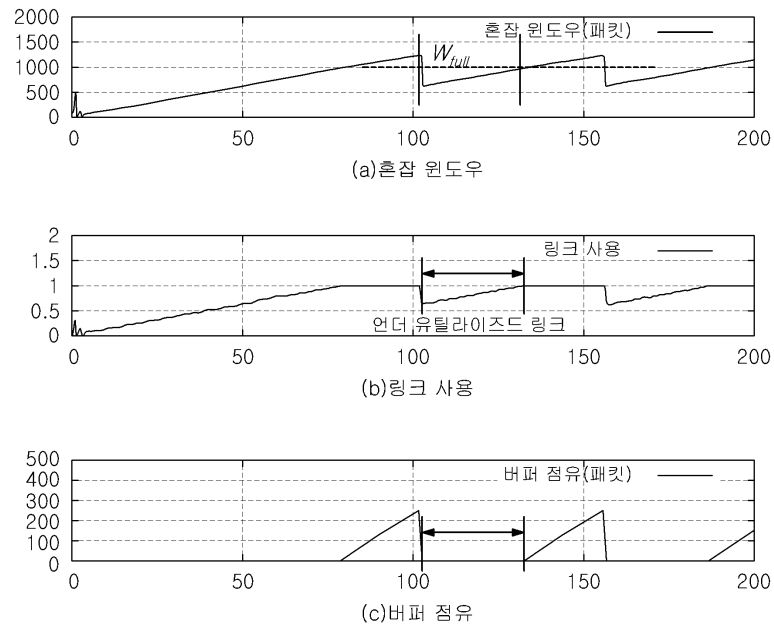
[0055]

100 : TCP 샌더

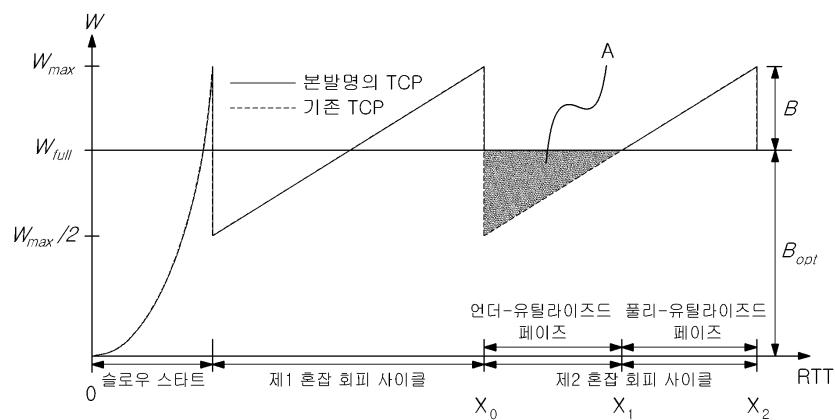
102 : TCP 리시버

도면

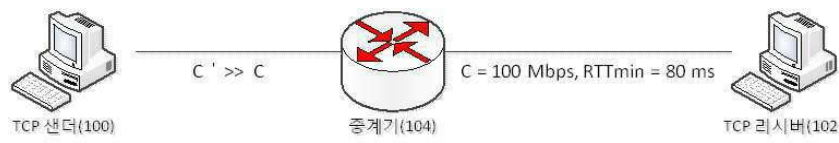
도면1



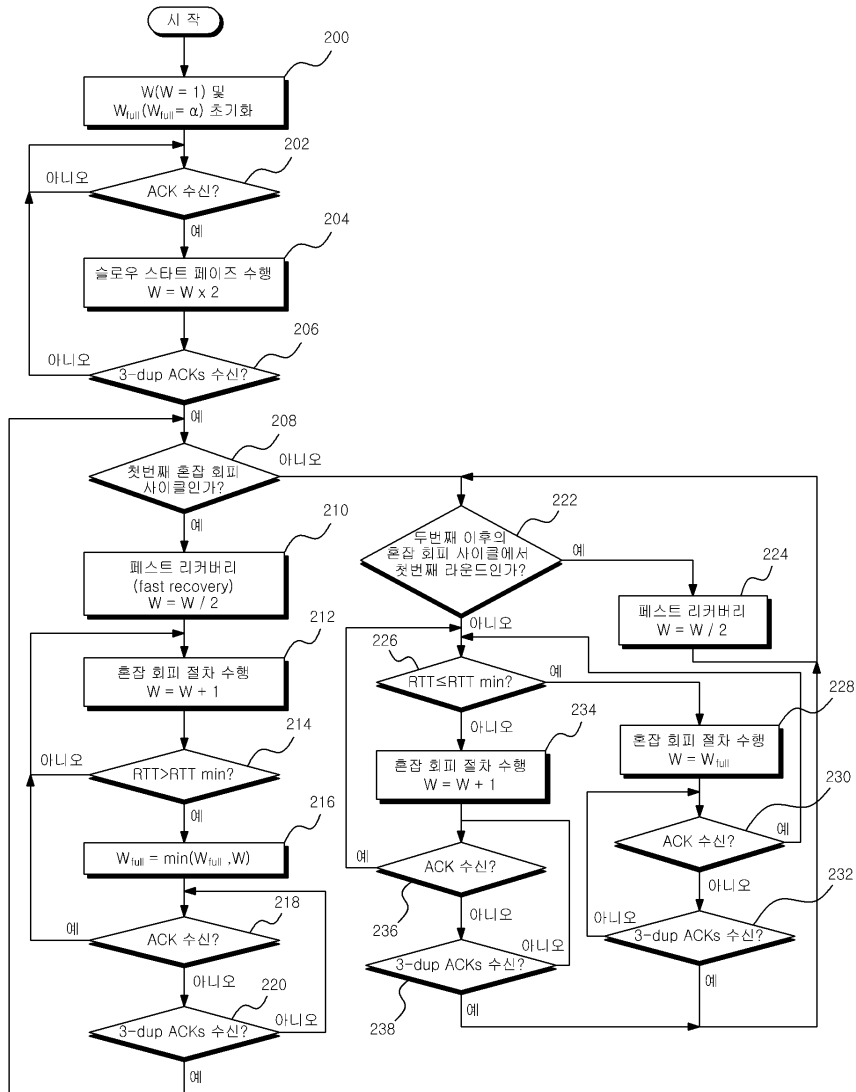
도면2



도면3



도면4



도면5

