(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0030649
(43) 공개일자 2019년03월22일

(51) 국제특허분류(Int. Cl.)
H04L 12/709 (2013.01) *H04L 1/16* (2006.01)
H04L 12/26 (2006.01) *H04L 12/807* (2013.01)
H04L 12/891 (2013.01) *H04L 12/911* (2013.01)
H04L 29/08 (2006.01)
(52) CPC특허분류
H04L 45/245 (2013.01)
H04L 1/16 (2013.01)
(21) 출원번호 10-2018-7036079
(22) 출원일자(국제) 2017년05월19일
심사청구일자 없음
(85) 번역문제출일자 2018년12월12일
(86) 국제출원번호 PCT/US2017/033619
(87) 국제공개번호 WO 2017/209995
국제공개일자 2017년12월07일
(30) 우선권주장
62/343,697 2016년05월31일 미국(US)
(뒷면에 계속)

(71) 출원인
앵커프리 인코포레이티드
미국 94063 캘리포니아 레드우드 시티 시포트 블바드 1800
(72) 발명자
라피더스, 유진
미국 94025 캘리포니아 멘로 파크 컨스티튜션 드라이브 155
몰카노브, 맥심
미국 94025 캘리포니아 멘로 파크 컨스티튜션 드라이브 155
(74) 대리인
특허법인 정안

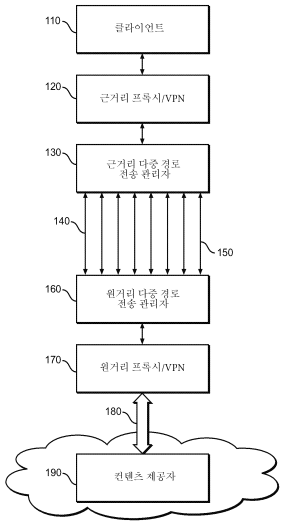
전체 청구항 수 : 총 27 항

(54) 발명의 명칭 **동시 연결의 통합된 처리량을 향상시키기 위한 시스템 및 방법**

(57) 요 약

클라이언트와 컨텐츠 제공자는 복수의 동시 전송 연결에 의해 연결된다. 데이터를 전송하는데에 사용되는 상기 전송 연결의 수는 전송될 데이터의 크기에 기초하여 선택되고, 상기 수는 데이터의 전송이 시작된 후에 남아 있는 데이터의 양과 상기 전송 연결의 속성에 기초하여 변경할 수 있다. 다른 양태에서, 전송 연결을 통해 전송될 데이터는 프레임들로 구성되고 각 프레임은 단 하나의 데이터 스트림으로부터 데이터를 포함한다. 상기 프레임은 프레임을 전송하는 전송 연결의 제어 윈도우 이하의 크기를 갖는다. 각 프레임은 라운드 로빈 방식으로 또는 상기 프레임의 크기 및 상기 전송 연결의 제어 윈도우의 크기에 기초하여 전송 연결에 할당될 수 있다.

대 표 도 - 도1



(52) CPC특허분류
H04L 43/0835 (2013.01)
H04L 43/0864 (2013.01)
H04L 43/0888 (2013.01)
H04L 47/27 (2013.01)
H04L 47/41 (2013.01)
H04L 47/827 (2013.01)
H04L 67/02 (2013.01)
H04L 67/28 (2013.01)
H04L 67/325 (2013.01)

(30) 우선권주장
15/254,732 2016년09월01일 미국(US)
15/254,583 2016년09월01일 미국(US)

명 세 서

청구범위

청구항 1

통합된 처리량(aggregated throughput)을 증가시키기 위한 시스템으로서, 상기 시스템은 하나 이상의 처리 디바이스를 포함하는 제1 컴퓨터를 포함하고, 상기 하나 이상의 처리 디바이스는,

제2 컴퓨터와 복수의 동시 전송 연결(simultaneous transport connection)을 수립하는 동작;

상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결에 대한 바람직한 데이터 그룹 크기를 결정하는 동작;

제2 컴퓨터로부터 데이터를 요청하는 하나 이상의 요청을 수신하는 동작;

상기 하나 이상의 요청에 대응하는 요청된 데이터를 획득하는 동작; 및

상기 제2 컴퓨터로부터 적어도 하나의 다른 요청을 수신하기 전에 상기 요청된 데이터의 전송이 완료되어야 하도록 상기 제1 컴퓨터와 상기 제2 컴퓨터 간에 상기 복수의 동시 전송 연결을 통해 상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 상기 요청된 데이터를 전송하는 동작을 수행하도록 프로그래밍되고;

상기 하나 이상의 처리 디바이스는,

상기 하나 이상의 요청 중 동일한 요청에 응답하여 수신된 상기 요청된 데이터의 일부를 2개 이상의 그룹으로 분할하는 동작으로서, 상기 2개 이상의 그룹의 각 그룹은 상기 복수의 동시 전송 연결의 대응하는 단일 연결에 할당되고, 대응하는 단일 연결의 바람직한 데이터 그룹 크기에 대응하는 데이터 크기를 갖는, 상기 분할하는 동작; 및

상기 대응하는 단일 연결을 통해 상기 2개 이상의 그룹의 각 그룹을 전송하는 동작을 수행하는 것에 의해,

상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 상기 요청된 데이터를 전송하는 동작을 수행하도록 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 2

제1항에 있어서, 상기 하나 이상의 처리 디바이스는,

상기 복수의 동시 전송 연결 중 이용 가능한 전송 연결의 최대 수를 결정하는 동작;

상기 복수의 동시 전송 연결 중 하나 이상의 연결의 속성에 따라 상기 복수의 동시 전송 연결의 각 연결의 바람직한 데이터 그룹 크기를 결정하는 동작; 및

상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 상기 요청된 데이터를 전송하는 동작으로서,

상기 요청된 데이터의 크기가 상기 복수의 동시 전송 연결의 바람직한 데이터 그룹 크기의 합보다 더 작을 때, 사용되는 상기 복수의 동시 전송 연결의 수가 상기 최대 수보다 더 작도록, (a) 사용되는 상기 복수의 동시 전송 연결의 수가 상기 최대 수에 도달하는 것과 (b) 상기 요청된 데이터의 전달이 완료되는 것 중 적어도 하나에 해당할 때까지, 상기 2개 이상의 그룹의 각 데이터 그룹을 상기 복수의 동시 전송 연결 중 상이한 연결들로 포워딩(forward)하는 동작을 수행하는 것에 의해

상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 상기 요청된 데이터를 전송하는 동작을 수행하도록 더 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 3

제2항에 있어서, 상기 하나 이상의 처리 디바이스는,

상기 이용 가능한 전송 연결의 최대 수를, 상기 제1 컴퓨터와 제2 컴퓨터 간에 현재 열려 있는 동시 전송 연결의 수로 설정하는 동작; 및

상기 요청된 데이터를 전달하기 위해 현재 열려 있는 전송 연결 중 적어도 하나의 연결을 사용하지 않는 동작을

수행하도록 더 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 4

제3항에 있어서, 상기 하나 이상의 처리 디바이스는 상기 요청된 데이터의 크기가 증가함에 따라 상기 요청된 데이터를 전달하는데 사용되는 상기 복수의 동시 전송 연결의 수를 증가시키도록 더 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 5

제1항에 있어서, 상기 하나 이상의 처리 디바이스는,

상기 적어도 하나의 그룹이 상기 적어도 2개의 상이한 요청 중 하나의 요청에 응답하여 획득된 데이터만을 포함하도록 상기 2개 이상의 그룹 중 적어도 하나의 데이터 그룹을 형성하는 동작; 및

상기 복수의 전송 연결 중 하나의 전송 연결에 상기 적어도 하나의 데이터 그룹을 분배하는 동작을 수행하도록 더 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 6

제5항에 있어서, 상기 하나 이상의 처리 디바이스는, 상기 적어도 하나의 데이터 그룹의 데이터를 순서화된 시퀀스로 제공함으로써 상기 복수의 전송 연결 중 상기 전송 연결에 상기 적어도 하나의 데이터 그룹을 분배하도록 더 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 7

제5항에 있어서, 상기 하나 이상의 처리 디바이스는 상기 복수의 동시 전송 연결 중 상기 전송 연결에 분배된 상기 적어도 하나의 데이터 그룹에 적어도 하나의 헤더를 부가하도록 더 프로그래밍되고, 상기 적어도 하나의 헤더는 상기 적어도 하나의 데이터 그룹의 크기 및 상기 하나의 요청에 응답하여 획득된 상기 데이터의 식별자를 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 8

제7항에 있어서, 상기 2개 이상의 그룹은 HTTP(Hyper Text Transport Protocol) 버전 2 이상에 따라 포맷되는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 9

제1항에 있어서, 상기 하나 이상의 처리 디바이스는 상기 2개 이상의 그룹의 각 데이터 그룹을 상기 복수의 동시 전송 연결 중 상이한 연결로 순차적으로 포워딩하도록 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 10

제1항에 있어서, 상기 하나 이상의 처리 디바이스는 상기 복수의 동시 전송 연결 중 적어도 2개에 대한 바람직한 데이터 그룹 크기가 상이하도록 각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하도록 더 프로그래밍된 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 11

제1항에 있어서, 상기 하나 이상의 처리 디바이스는,

상기 복수의 동시 전송 연결을 위한 전송 프로토콜을 구현하도록 더 프로그래밍되고, 상기 전송 프로토콜은 트래픽의 적어도 일부에 대한 수신확인 응답을 수신하기 전에 송신될 수 있는 상기 트래픽의 양을 제한하는 제어 윈도우를 부과하하고;

상기 하나 이상의 처리 디바이스는,

각 연결의 바람직한 데이터 그룹 크기의 범위 내에서 각 연결의 바람직한 데이터 그룹 크기를 선택하는 것에 의

해,

각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하도록 더 프로그래밍되고,

각 연결의 바람직한 데이터 그룹 크기의 범위는 각 연결에 대한 상기 제어 윈도우의 크기가 증가함에 따라 증가하는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 12

제11항에 있어서, 상기 전송 프로토콜은 전송 제어 프로토콜(TCP)이고, 상기 제어 윈도우는 수신자 윈도우와 혼잡 윈도우 중 더 작은 것인 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 13

제11항에 있어서, 각 연결의 바람직한 데이터 그룹 크기의 범위의 최대값은 각 연결에 대한 상기 제어 윈도우의 크기보다 더 크지 않도록 설정되는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 14

제11항에 있어서, 각 연결의 바람직한 데이터 그룹 크기의 범위의 최소값은 각 연결에 대한 상기 제어 윈도우의 크기의 절반보다 더 작지 않도록 설정되는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 15

제1항에 있어서, 상기 하나 이상의 처리 디바이스는,

상기 복수의 동시 전송 연결을 위한 전송 프로토콜을 구현하는 동작으로서, 상기 전송 프로토콜은 트래픽의 적어도 일부에 대한 수신확인 응답을 수신하기 전에 각 연결을 통해 송신될 수 있는 상기 트래픽의 양을 제한하는 각 연결에 대한 제어 윈도우를 부과하는, 상기 전송 프로토콜을 구현하는 동작; 및

상기 데이터 그룹 크기를 갖는 전체 그룹이 수신확인 응답을 수신함이 없이 각 연결을 통해 송신될 수 있도록 각 연결의 바람직한 데이터 그룹 크기를 설정함으로써 각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하는 동작을 수행하도록 더 프로그래밍되는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 16

제1항에 있어서, 상기 하나 이상의 처리 디바이스는,

각 연결의 바람직한 데이터 그룹 크기를 각 연결의 왕복(round trip) 시간에 각 연결을 통해 송신될 수 있는 데이터의 양으로 설정하는 것에 의해 각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하도록 더 프로그래밍되는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 17

제1항에 있어서, 상기 하나 이상의 처리 디바이스는 동일한 물리적 데이터 링크를 통해 상기 복수의 동시 전송 연결 중 적어도 2개를 수립하도록 더 프로그래밍되는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 18

제17항에 있어서, 상기 복수의 동시 전송 연결 중 적어도 2개는 동일한 출발지 및 목적지 네트워크 어드레스들을 갖는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 19

제1항에 있어서, 상기 제1 컴퓨터는 프록시와 가상 사설망(VPN) 서버 중 적어도 하나를 실행하는 것을 특징으로 하는 통합된 처리량을 증가시키기 위한 시스템.

청구항 20

통합된 처리량을 증가시키는 방법으로서,

제2 컴퓨터에 의해, 컴퓨터 네트워크를 통해 제1 컴퓨터와 상기 제2 컴퓨터간에 복수의 동시 전송 연결을 수립하는 단계;

상기 제1 컴퓨터에 의해, 상기 제2 컴퓨터로부터 2개 이상의 데이터 전송 요청을 수신하는 단계;

상기 제1 컴퓨터에 의해, 상기 2개 이상의 요청의 각 요청에 응답하여 요청된 데이터를 획득하는 단계;

상기 제1 컴퓨터에 의해 2개 이상의 데이터 프레임을 형성하는 단계로서, 상기 2개 이상의 데이터 프레임의 각 프레임은 상기 2개 이상의 요청 중 단 하나의 요청에만 대응하는 상기 요청된 데이터를 포함하고, 상기 2개 이상의 데이터 프레임은 상기 2개 이상의 요청의 2개의 상이한 요청에 대응하는 데이터를 포함하는, 상기 2개 이상의 데이터 프레임을 형성하는 단계;

상기 2개 이상의 데이터 프레임의 각 프레임의 모든 데이터가 상기 복수의 동시 전송 연결의 동일한 전송 연결로 포워딩되도록, 상기 제1 컴퓨터에 의해 상기 복수의 동시 전송 연결 중 2개 이상의 상이한 전송 연결로 상기 2개 이상의 데이터 프레임을 포워딩하는 단계; 및

상기 제1 컴퓨터에 의해, 상기 2개 이상의 상이한 전송 연결 중 하나의 전송 연결을 통해 상기 2개 이상의 요청 중 동일한 요청에 대응하는 데이터를 포함하는 상기 2개 이상의 데이터 프레임 중 적어도 하나의 데이터 프레임을 전송하는 단계를 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

청구항 21

제20항에 있어서,

상기 제1 컴퓨터에 의해, 동일한 임시 버퍼 내에 상기 2개 이상의 요청 중 2개의 상이한 요청에 대응하는 데이터를 포함하는 상기 2개 이상의 데이터 프레임을 저장하는 단계; 및

상기 복수의 동시 전송 연결 중 상이한 전송 연결로 각 프레임을 포워딩하기 전에 상기 제1 컴퓨터에 의해 상기 동일한 임시 버퍼로부터 상기 2개 이상의 데이터 프레임의 각 프레임을 검색하는 단계를 더 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

청구항 22

제20항에 있어서, 상기 요청된 데이터의 양이 증가함에 따라 상기 제1 컴퓨터에 의해 상기 2개 이상의 데이터 프레임을 전송하는데 사용되는 상기 2개 이상의 상이한 전송 연결의 수를 증가시키는 단계를 더 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

청구항 23

제20항에 있어서,

상기 2개 이상의 제2 요청 중 하나 이상의 제2 요청에 응답하여 상기 요청된 데이터를 전달하기 전에 상기 복수의 동시 전송 연결을 통해 상기 2개 이상의 요청 중 2개 이상의 제1 요청의 각 요청에 응답하여 상기 제1 컴퓨터에 의해 상기 요청된 데이터를 전달하는 단계로서, 상기 하나 이상의 제2 요청은 상기 2개 이상의 제1 요청 후에 수신된, 상기 요청된 데이터를 전달하는 단계; 및

상기 제1 요청에 응답하여 상기 요청된 데이터 중 전달되어야 하는 데이터의 잔량이 감소함에 따라 상기 제1 컴퓨터에 의해 상기 2개 이상의 데이터 프레임의 크기를 증가시키는 단계를 더 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

청구항 24

제20항에 있어서, 상기 복수의 동시 전송 연결 중 하나 이상의 전송 연결의 처리량이 증가함에 따라 상기 제1 컴퓨터에 의해 상기 2개 이상의 데이터 프레임의 크기를 증가시키는 단계를 더 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

청구항 25

제24항에 있어서, 상기 복수의 동시 전송 연결 중 상기 하나 이상의 전송 연결에 대한 트래픽의 적어도 일부에 대한 수신확인 응답을 수신하기 전에 송신이 허용된 패킷 손실률이 감소하는 것과 상기 트래픽의 양이 증가하는 것 중 적어도 하나를 검출함으로써 상기 처리량이 증가하는 것을 상기 제1 컴퓨터에 의해 검출하는 단계를 더 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

청구항 26

제20항에 있어서, 상기 제1 컴퓨터에 의해 프록시와 가상 사설망(VPN) 서버 중 적어도 하나를 실행하는 단계를 더 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

청구항 27

제20항에 있어서, 상기 제1 컴퓨터에 의해, 하이퍼텍스트 전송 프로토콜 버전 2 이상의 사양에 따라 상기 2개 이상의 데이터 프레임을 형성하는 단계를 더 포함하는 것을 특징으로 하는 통합된 처리량을 증가시키는 방법.

발명의 설명

기 술 분 야

[0001] 관련 출원

[0002] 본 출원은, 2016년 5월 31일에 출원되고 전체 내용이 본 명세서에 병합된 발명의 명칭이 "SYSTEM AND METHOD FOR IMPROVING AN AGGREGATED THROUGHPUT OF SIMULTANEOUS CONNECTIONS"인 미국 가출원 번호 62/343,697의 이익을 주장한다.

[0003] 본 출원은 또한, 2016년 9월 1일에 출원되고 발명의 명칭이 "SYSTEM AND METHOD FOR IMPROVING AN AGGREGATED THROUGHPUT OF SIMULTANEOUS CONNECTIONS"(대리인 관리 번호 ANCH-00501)인 미국 출원 번호 15/254,583의 전체 내용을 병합한다.

배 경 기 술

[0004] 컴퓨터 네트워크를 통해 다운로드되는 온라인 컨텐츠의 양은 온라인 비디오 스트리밍 및 사용자 생성 컨텐츠의 인기, 소셜 네트워크 및 풍부한 미디어 메시징의 확산, 클라우드 기반 저장 등 다수의 요인으로 인해 시간에 따라 급격히 증가하고 있다.

[0005] 요청된 컨텐츠는 종종 장거리에 걸쳐 전달되어야 한다. 가장 인기 있는 비디오만이 사용자 근처에서 캐싱될 수 있다. HTTPS와 같은 보안 프로토콜을 사용하는 컨텐츠는 적어도 일부 암호화 키를 제3자에게 개시하지 않고는 캐싱될 수 없다. 스포츠 및 뉴스와 같이 실시간으로 스트리밍되는 이벤트를 원격에서 보는 것은 캐싱을 사용할 수 없다. 로컬 컨텐츠는 종종 발신(origin) 국가의 외부에서는 캐싱되지 않는다.

[0006] 데이터 이동 거리가 증가하면 종종 패킷 손실률이 증가된다. 연결이 전송 제어 프로토콜(Transport Control Protocol: TCP)과 같은 전달-보장 프로토콜을 사용한다면, 이러한 손실은 보통 마지막 마일 혼잡(last-mile congestion)으로 해석되어, 연결 파이프의 각 세그먼트가 포화 지점에서 멀리 떨어져 있더라도 처리량이 상당히 감소하게 된다.

[0007] 송신자와 수신자 간에 다수의 동시 전달-보장 전송 연결(transport connection)을 통해 데이터 전송을 확산시키면 일반적으로 총 처리량이 증가하는데, 즉 각 새로운 연결을 추가하면 수신자 윈도우(receiver window)와 혼잡 윈도우의 통합된 크기가 증가하여, 각 왕복(round trip) 구간 동안 더 많은 데이터를 전달할 수 있게 된다. 장거리에 걸쳐 큰 데이터 파일을 전송하는 속도는 연결 파이프의 적어도 하나의 세그먼트가 혼잡해질 때까지 또는 서버와 클라이언트 컴퓨터의 자원(메모리, CPU 부하, I/O 용량)이 무리하게 쓰일 때까지 동시 연결의 수에 따라 증가한다.

[0008] 그리하여, 제한된 크기의 데이터 파일의 버스트를 포함하는, 실제 시나리오에서 다수의 동시 연결을 통해 데이터 전송 속도를 증가시킬 필요가 있다.

발명의 내용

[0009] 본 발명의 일 양태에서, 통합된 처리량(aggregated throughput)을 증가시키는 시스템으로서, 상기 시스템은 하나 이상의 처리 디바이스를 포함하는 제1 컴퓨터를 포함하고, 상기 하나 이상의 처리 디바이스는,

[0010] 제2 컴퓨터와 복수의 동시 전송 연결(simultaneous transport connection)을 수립하는 동작;

[0011] 상기 복수의 동시 전송 연결 중 적어도 일부의 연결의 각 연결에 대한 바람직한 데이터 그룹 크기를 결정하는 동작;

[0012] 제2 컴퓨터로부터 데이터를 요청하는 하나 이상의 요청을 수신하는 동작;

[0013] 상기 하나 이상의 요청에 대응하는 요청된 데이터를 획득하는 동작; 및

[0014] 상기 제2 컴퓨터로부터 적어도 하나의 다른 요청을 수신하기 전에 상기 요청된 데이터의 전송이 완료되어야 하도록 상기 제1 컴퓨터와 상기 제2 컴퓨터 간에 상기 복수의 동시 전송 연결을 통해 상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 상기 요청된 데이터를 전송하는 동작을 수행하도록 프로그래밍된다.

[0015] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는,

[0016] 상기 하나 이상의 요청 중 동일한 요청에 응답하여 수신된 상기 요청된 데이터의 일부를 2개 이상의 그룹으로 분할하는 동작으로서, 상기 2개 이상의 그룹의 각 그룹은 상기 복수의 동시 전송 연결의 대응하는 단일 연결에 할당되고, 대응하는 단일 연결의 바람직한 데이터 그룹 크기에 대응하는 데이터 크기를 갖는, 상기 분할하는 동작; 및

[0017] 상기 대응하는 단일 연결을 통해 상기 2개 이상의 그룹의 각 그룹을 전송하는 동작

[0018] 을 수행하는 것에 의해, 상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 상기 요청된 데이터를 전송하는 동작을 수행하도록 프로그래밍된다.

[0019] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는,

[0020] 상기 복수의 동시 전송 연결 중 이용 가능한 전송 연결의 최대 수를 결정하는 동작;

[0021] 상기 복수의 동시 전송 연결 중 하나 이상의 연결의 속성(attribute)에 따라 상기 복수의 동시 전송 연결의 각 연결의 바람직한 데이터 그룹 크기를 결정하는 동작; 및

[0022] 상기 요청된 데이터의 크기가 상기 복수의 동시 전송 연결의 바람직한 데이터 그룹 크기의 합보다 더 작을 때, 사용되는 상기 복수의 동시 전송 연결의 수가 상기 최대 수보다 더 작도록, (a) 사용되는 상기 복수의 동시 전송 연결의 수가 상기 최대 수에 이르는 것과 (b) 상기 요청된 데이터의 전달이 완료되는 것 중 적어도 하나에 해당할 때까지, 상기 2개 이상의 그룹의 각 데이터 그룹을 상기 복수의 동시 전송 연결 중 상이한 연결로 포워딩(forwarding)하는 것에 의해 상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 상기 요청된 데이터를 전송하는 동작을 수행하도록 더 프로그래밍된다.

[0023] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는, 상기 이용 가능한 전송 연결의 최대 수를 상기 제1 컴퓨터와 상기 제2 컴퓨터 간에 현재 열려 있는 동시 전송 연결의 수로 설정하는 동작; 및 상기 요청된 데이터를 전달하기 위해 현재 열려 있는 전송 연결 중 적어도 하나를 사용하지 않는 동작을 수행하도록 더 프로그래밍된다.

[0024] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 상기 요청된 데이터의 크기가 증가함에 따라 상기 요청된 데이터를 전달하는데 사용되는 상기 복수의 동시 전송 연결의 수를 증가시키도록 더 프로그래밍된다.

[0025] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는, 상기 적어도 하나의 그룹이 상기 적어도 2개의 상이한 요청 중 하나의 요청에 응답하여 획득된 데이터만을 포함하도록 상기 2개 이상의 그룹의 적어도 하나의 데이터 그룹을 형성하는 동작; 및 상기 복수의 전송 연결 중 전송 연결에 상기 적어도 하나의 데이터 그룹을 분배하는 동작을 수행하도록 더 프로그래밍된다.

[0026] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 상기 적어도 하나의 데이터 그룹의 데이터를 순서화된 시퀀스로 제공함으로써 상기 복수의 전송 연결 중 전송 연결에 상기 적어도 하나의 데이터 그룹을 분배하도록 더 프로그래밍된다.

[0027] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 상기 복수의 동시 전송 연결 중 전송 연결에 분배된 상기 적어도 하나의 데이터 그룹에 적어도 하나의 헤더를 부가하도록 더 프로그래밍되고, 상기 적어도 하나의 헤더는 상기 적어도 하나의 데이터 그룹의 크기 및 상기 하나의 요청에 응답하여 획득된 데이터의 식별자를 포함한다.

[0028] 일부 실시예에서, 상기 2개 이상의 그룹은 HTTP(Hyper Text Transport Protocol) 버전 2 이상에 따라 포맷된다.

[0029] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 상기 2개 이상의 그룹의 각 데이터 그룹을 상기 복수의 동시 전송 연결의 상이한 연결로 순차적으로 포워딩하도록 프로그래밍된다.

[0030] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 상기 복수의 동시 전송 연결 중 적어도 2개의 연결에 대한 바람직한 데이터 그룹 크기가 상이하도록 각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하도록 더 프로그래밍된다.

[0031] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 상기 복수의 동시 전송 연결을 위한 전송 프로토콜을 구현하도록 더 프로그래밍되고, 상기 전송 프로토콜은 트래픽의 적어도 일부에 대한 수신확인 응답(acknowledgement)을 수신하기 전에 송신될 수 있는 상기 트래픽의 양을 제한하는 제어 윈도우를 부과한다.

[0032] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 각 연결의 바람직한 데이터 그룹 크기의 범위 내에서 각 연결의 바람직한 데이터 그룹 크기를 선택하는 것에 의해 각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하도록 더 프로그래밍되고, 각 연결의 바람직한 데이터 그룹 크기의 범위는 각 연결에 대한 상기 제어 윈도우의 크기가 증가함에 따라 증가한다.

[0033] 일부 실시예에서, 상기 전송 프로토콜은 전송 제어 프로토콜(TCP)이고, 상기 제어 윈도우는 수신자 윈도우와 혼잡 윈도우 중 더 작은 것이다.

[0034] 일부 실시예에서, 각 연결의 바람직한 데이터 그룹 크기의 범위의 최대값은 각 연결에 대한 상기 제어 윈도우의 크기보다 더 크지 않도록 설정된다.

[0035] 일부 실시예에서, 각 연결의 바람직한 데이터 그룹 크기의 범위의 최소값은 각 연결에 대한 상기 제어 윈도우의 크기의 절반보다 더 작지 않도록 설정된다.

[0036] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는,

[0037] 상기 복수의 동시 전송 연결을 위한 전송 프로토콜을 구현하는 동작으로서, 상기 전송 프로토콜은 트래픽의 적어도 일부에 대한 수신확인 응답을 수신하기 전에 각 연결을 통해 송신될 수 있는 상기 트래픽의 양을 제한하는 각 연결에 대한 제어 윈도우를 부과하는, 상기 전송 프로토콜을 구현하는 동작; 및

[0038] 상기 데이터 그룹 크기를 갖는 전체 그룹이 수신확인 응답을 수신함이 없이 각 연결을 통해 송신될 수 있도록 각 연결의 바람직한 데이터 그룹 크기를 설정함으로써 각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하는 동작을 수행하도록 더 프로그래밍된다.

[0039] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 각 연결의 바람직한 데이터 그룹 크기를, 각 연결의 왕복 시간에 각 연결을 통해 송신될 수 있는 데이터의 양으로 설정하는 것에 의해 각 연결의 속성에 따라 상기 복수의 동시 전송 연결 중 적어도 일부의 각 연결의 바람직한 데이터 그룹 크기를 결정하도록 더 프로그래밍된다.

[0040] 일부 실시예에서, 상기 하나 이상의 처리 디바이스는 동일한 물리적 데이터 링크를 통해 상기 복수의 동시 전송 연결 중 적어도 2개를 수립하도록 더 프로그래밍된다.

[0041] 일부 실시예에서, 상기 복수의 동시 전송 연결 중 적어도 2개의 연결은 동일한 출발지 및 목적지 네트워크 어드레스들을 갖는다.

[0042] 일부 실시예에서, 상기 제1 컴퓨터는 프록시와 가상 사설망(VPN) 서버 중 적어도 하나를 실행한다.

[0043] 일부 실시예에서, 처리량을 증가시키는 방법은,

[0044] 제2 컴퓨터에 의해, 컴퓨터 네트워크를 통해 제1 컴퓨터와 상기 제2 컴퓨터간에 복수의 동시 전송 연결을 수립하는 단계;

[0045] 상기 제1 컴퓨터에 의해, 상기 제2 컴퓨터로부터 2개 이상의 데이터 전송 요청을 수신하는 단계;

[0046] 상기 제1 컴퓨터에 의해, 상기 2개 이상의 요청의 각 요청에 응답하여 요청된 데이터를 획득하는 단계;

[0047] 상기 제1 컴퓨터에 의해 2개 이상의 데이터 프레임을 형성하는 단계로서, 상기 2개 이상의 데이터 프레임의 각 프레임은 상기 2개 이상의 요청 중 단 하나의 요청에만 대응하는 상기 요청된 데이터를 포함하고, 상기 2개 이상의 데이터 프레임은 상기 2개 이상의 요청 중 2개의 상이한 요청에 대응하는 데이터를 포함하는, 상기 형성하

는 단계;

[0048] 상기 2개 이상의 데이터 프레임의 각 프레임의 모든 데이터가 상기 복수의 동시 전송 연결 중 동일한 전송 연결로 포워딩되도록, 상기 제1 컴퓨터에 의해 상기 복수의 동시 전송 연결 중 2개 이상의 상이한 전송 연결로 상기 2개 이상의 데이터 프레임을 포워딩하는 단계; 및

[0049] 상기 제1 컴퓨터에 의해, 상기 2개 이상의 상이한 전송 연결 중 하나의 전송 연결을 통해 상기 2개 이상의 요청 중 동일한 요청에 대응하는 데이터를 포함하는 상기 2개 이상의 데이터 프레임 중 적어도 하나의 데이터 프레임을 전송하는 단계를 포함한다.

[0050] 일부 실시예에서, 상기 방법은,

[0051] 상기 제1 컴퓨터에 의해, 동일한 임시 버퍼 내에 상기 2개 이상의 요청 중 2개의 상이한 요청에 대응하는 데이터를 포함하는 상기 2개 이상의 데이터 프레임을 저장하는 단계; 및

[0052] 상기 복수의 동시 전송 연결 중 상이한 전송 연결로 각 프레임을 포워딩하기 전에 상기 제1 컴퓨터에 의해 상기 동일한 임시 버퍼로부터 상기 2개 이상의 데이터 프레임의 각 프레임을 검색하는 단계를 더 포함한다.

[0053] 일부 실시예에서, 상기 방법은, 상기 요청된 데이터의 양이 증가함에 따라 상기 제1 컴퓨터에 의해 상기 2개 이상의 데이터 프레임을 전송하는데 사용되는 상기 2개 이상의 상이한 전송 연결의 수를 증가시키는 단계를 포함한다.

[0054] 일부 실시예에서, 상기 방법은,

[0055] 상기 2개 이상의 제2 요청 중 하나 이상의 제2 요청에 응답하여 상기 요청된 데이터를 전달하기 전에 상기 복수의 동시 전송 연결을 통해 상기 2개 이상의 요청 중 2개 이상의 제1 요청의 각 요청에 응답하여 상기 제1 컴퓨터에 의해 상기 요청된 데이터를 전달하는 단계로서, 상기 하나 이상의 제2 요청은 상기 2개 이상의 제1 요청 후에 수신된, 상기 요청된 데이터를 전달하는 단계; 및

[0056] 상기 제1 요청에 응답하여 상기 요청된 데이터 중 전달되어야 하는 데이터의 잔량이 감소함에 따라 상기 제1 컴퓨터에 의해 상기 2개 이상의 데이터 프레임의 크기를 증가시키는 단계를 더 포함한다.

[0057] 일부 실시예에서, 상기 방법은 상기 복수의 동시 전송 연결 중 하나 이상의 전송 연결의 처리량이 증가함에 따라 상기 제1 컴퓨터에 의해 상기 2개 이상의 데이터 프레임의 크기를 증가시키는 단계를 더 포함한다.

[0058] 일부 실시예에서, 상기 방법은 상기 복수의 동시 전송 연결 중 상기 하나 이상의 전송 연결에 대한 트래픽의 적어도 일부에 대한 수신확인 응답을 수신하기 전에 패킷 손실률이 감소하는 것과 송신이 허용된 트래픽의 양이 증가하는 것 중 적어도 하나를 검출함으로써 상기 처리량이 증가하는 것을 상기 제1 컴퓨터에 의해 검출하는 단계를 더 포함한다.

[0059] 일부 실시예에서, 상기 방법은 상기 제1 컴퓨터에 의해 프록시와 가상 사설망(VPN) 서버 중 적어도 하나를 실행하는 단계를 더 포함한다.

[0060] 일부 실시예에서, 상기 방법은, 상기 제1 컴퓨터에 의해, 하이퍼 텍스트 전송 프로토콜 버전 2 이상의 사양에 따라 상기 2개 이상의 데이터 프레임을 형성하는 단계를 더 포함한다.

도면의 간단한 설명

[0061] 본 발명의 장점을 용이하게 이해할 수 있도록 하기 위해, 위에서 간단히 기술된 본 발명의 보다 상세한 설명이 첨부된 도면에 도시된 특정 실시예를 참조하여 제공될 것이다. 이들 도면은 본 발명의 단지 예시적인 실시예를 도시하고 그 범위를 제한하는 것으로 간주되어서는 안 되는 것으로 이해하면서, 본 발명은 첨부된 도면을 사용하는 것을 통해 추가적인 특징과 함께 상세히 설명되고 기술될 것이다:

도 1은 본 발명의 일 실시예에 따라 다수의 동시 전송 연결을 통해 데이터를 전송함으로써 처리량을 증가시키는데 사용될 수 있는 제1 네트워크 환경의 개략적인 블록도이다;

도 2는 본 발명의 일 실시예에 따라 본 발명의 실시예에 따라 동시 전송 연결의 수를 증가시키는데 사용되는 제2 네트워크 환경의 개략적인 블록도이다;

도 3은 본 발명의 일 실시예에 따라 동시 전송 연결의 수를 변경하는데 사용되는 제3 네트워크 환경의 개략적인 블록도이다;

도 4a 및 도 4b는 본 발명의 실시예에 의해 달성될 수 있는 처리량이 향상된 것을 도시하는 그래프이다;

도 5는 본 발명의 일 실시예에 따라 동시 전송 연결의 수를 변경하는 방법의 프로세스 흐름도이다;

도 6은 본 발명의 일 실시예에 따라 동시 전송 연결을 통한 트래픽의 분배를 향상시키는데 사용되는 제4 네트워크 환경의 도면이다;

도 7은 본 발명의 일 실시예에 따라 동시 전송 연결을 통한 트래픽의 분배를 향상시키기 위한 방법의 프로세스 흐름도이다;

도 8은 본 발명의 일 실시예에 따라 동시 전송 연결을 통한 트래픽의 분배를 향상시키는데 사용되는 제5 네트워크 환경의 개략적인 블록도이다;

도 9는 본 발명의 일 실시예에 따라 동시 전송 연결을 통한 트래픽의 분배를 향상시키는데 사용되는 제6 네트워크 환경의 개략적인 블록도이다; 및

도 10은 본 명세서에 개시된 시스템 및 방법을 구현하기에 적합한 컴퓨터의 개략적인 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0062] 본 명세서에서 일반적으로 설명되고 도면에 도시된 바와 같이 본 발명의 구성 요소는 다양한 상이한 구성으로 배열되고 설계될 수 있다는 것이 쉽게 이해될 수 있을 것이다. 따라서, 도면에 도시된 바와 같이 본 발명의 실시예에 대한 다음의 보다 상세한 설명은 본 발명의 청구 범위를 제한하려고 의도된 것이 아니라 본 발명에 따라 현재 고려되는 실시예의 특정 예들을 단지 예시하려고 의도된 것이다. 본 명세서에 기술된 실시예는 도면을 참조하여 가장 잘 이해될 수 있을 것이며, 도면에서 동일한 부분은 명세서 전체에 걸쳐 동일한 도면 부호로 표시된다.

[0063] 본 발명에 따른 실시예는 장치, 방법 또는 컴퓨터 프로그램 제품으로서 구현될 수 있다. 따라서, 본 발명은 전적으로 하드웨어인 실시예, 전적으로 소프트웨어인 실시예(펌웨어, 상주 소프트웨어, 마이크로코드 등을 포함함), 또는 본 명세서에서 일반적으로 모두 "모듈" 또는 "시스템"이라고 언급될 수 있는 소프트웨어 및 하드웨어 양태를 결합한 실시예의 형태를 취할 수 있다. 나아가, 본 발명은 매체에 구현된 컴퓨터-사용 가능 프로그램 코드를 갖는 표현의 임의의 유형적인 매체에 구현된 컴퓨터 프로그램 제품의 형태를 취할 수 있다.

[0064] 하나 이상의 컴퓨터-사용 가능 또는 컴퓨터-판독 가능 매체의 임의의 조합이 이용될 수 있다. 예를 들어, 컴퓨터 판독 가능 매체는 휴대용 컴퓨터 디스켓, 하드 디스크, 랜덤 액세스 메모리(RAM) 디바이스, 판독 전용 메모리(ROM) 디바이스, 소거 가능 프로그래밍 가능 판독 전용 메모리(EPROM 또는 플래시 메모리) 디바이스, 휴대용 콤팩트 디스크 판독 전용 메모리(CDROM), 광학 저장 디바이스, 및 자기 저장 디바이스 중 하나 이상을 포함할 수 있다. 선택된 실시예에서, 컴퓨터 판독 가능 매체는 명령 실행 시스템, 장치 또는 디바이스에 의해 또는 이와 관련하여 사용하기 위해 프로그램을 포함, 저장, 통신, 전파 또는 전송할 수 있는 임의의 비 일시적인 매체를 포함할 수 있다.

[0065] 본 발명의 동작을 수행하기 위한 컴퓨터 프로그램 코드는 자바(Java), 스몰토크(Smalltalk), C++ 등과 같은 객체 지향 프로그래밍 언어 및 "C" 프로그래밍 언어 또는 이와 유사한 프로그래밍 언어와 같은 종래의 절차형 프로그래밍 언어를 포함하는 하나 이상의 프로그래밍 언어의 임의의 조합으로 기록될 수 있고, 또한 HTML, XML, JSON 등과 같은 설명 또는 마크업 언어를 사용할 수도 있다. 프로그램 코드는 컴퓨터 시스템 상에서 독립형 소프트웨어 패키지로서 완전히 실행되거나, 독립형 하드웨어 유닛 상에서 실행되거나, 컴퓨터로부터 어느 정도 떨어져 있는 원격 컴퓨터 상에서 부분적으로 실행되거나 또는 원격 컴퓨터 또는 서버 상에서 완전히 실행될 수 있다. 원격 컴퓨터에서 실행되는 시나리오에서, 원격 컴퓨터는 근거리 통신망(LAN) 또는 광역 통신망(WAN)을 포함하는 임의의 유형의 네트워크를 통해 컴퓨터에 연결될 수 있고, 또는 (예를 들어, 인터넷 서비스 제공자를 사용하여 인터넷을 통해) 외부 컴퓨터에 연결될 수 있다. 컴퓨터 네트워크는 인터넷 프로토콜 이외의 전송 프로토콜을 사용할 수 있다. 대응하여, 본 발명은 IP 어드레스 이외의 네트워크 어드레스의 유형에 대해 구현될 수 있다.

[0066] 본 발명은 본 발명의 실시예에 따른 방법, 장치(시스템) 및 컴퓨터 프로그램 제품의 흐름도 및/또는 블록도를 참조하여 아래에서 설명된다. 흐름도 및/또는 블록도의 각 블록, 및 흐름도 및/또는 블록도의 블록의 조합은 컴퓨터 프로그램 명령 또는 코드에 의해 구현될 수 있는 것으로 이해된다. 이들 컴퓨터 프로그램 명령은 일반 목적 컴퓨터, 특수 목적 컴퓨터 또는 다른 프로그래밍 가능 데이터 처리 장치의 프로세서에 제공되어, 컴퓨터 또

는 다른 프로그래밍 가능 데이터 처리 장치의 프로세서를 통해 실행되는 명령이 흐름도 및/또는 블록도의 블록 또는 블록들에 지정된 기능/동작을 구현하기 위한 수단을 생성하는 기계를 생성할 수 있다.

[0067] 이들 컴퓨터 프로그램 명령은 또한 비 일시적인 컴퓨터 판독 가능 매체에 저장되어, 컴퓨터 또는 다른 프로그래밍 가능 데이터 처리 장치가 특정 방식으로 기능하도록 지시하여, 컴퓨터 판독 가능 매체에 저장된 명령이 흐름도 및/또는 블록도의 블록 또는 블록들에서 지정된 기능/동작을 구현하는 명령 수단을 포함하는 제조 물품을 생성하도록 할 수 있다.

[0068] 컴퓨터 프로그램 명령은 또한 컴퓨터 또는 다른 프로그래밍 가능 데이터 처리 장치 상에 로딩(loaded)되어, 일련의 동작 단계들이 컴퓨터 또는 다른 프로그래밍 가능 장치 상에서 수행되어, 컴퓨터 또는 다른 프로그래밍 가능 장치에서 실행되는 명령이 흐름도 및/또는 블록도의 블록 또는 블록들에서 지정된 기능/동작을 구현하기 위한 프로세스를 제공하는 컴퓨터로 구현되는 프로세스를 생성하도록 할 수 있다.

[0069] 충분한 대역폭 용량(혼잡은 문제가 아님)이 있는 경우, 장거리에 걸친 동시 연결의 수는 일반적으로 서버 및 클라이언트 컴퓨터의 압도적인 자원 없이 유지될 수 있는 최대값으로 미리 설정된다. 그러나 일반적인 컴퓨터 네트워크를 통해 전송되는 대부분의 데이터 파일은 비교적 작은 크기를 갖는다. 이러한 전송은 종종 버스트로 발생하고(예를 들어, 웹사이트의 다운로드) 이후 비 활동 기간이 뒤따른다. 혼잡이 없는 경우에는 동시 연결의 수가 증가해도 이러한 제한된 데이터 전송에는 덜 효과적일 수 있다. 본 명세서에 개시된 실시예는 데이터를 전송하는데 사용되는 동시 연결의 수를 설정하는데 향상된 접근법을 제공한다.

[0070] 본 발명의 일 양태에서, 제1 컴퓨터는 상기 제1 컴퓨터와 복수의 동시 전송 연결을 통해 제2 컴퓨터로 데이터를 전달하기 위한 2개 이상의 요청 그룹을 수신하고, 여기서 상기 요청된 전달의 제1 그룹은 상기 제2 요청 그룹을 제출하기 전에 완료되어야 한다. 일부 실시예에서, 상기 제1 컴퓨터는 상기 요청된 데이터 전달이 완료된 것을 결정하는 수단 및 상기 요청된 데이터를 상기 제1 컴퓨터에 전달하는데 사용되는 동시 연결의 수를 설정하는 수단, 및 상기 요청된 데이터의 제1 그룹을 전달하는데 사용된 동시 연결의 수를 상기 요청된 데이터의 제2 그룹을 전달하는데 사용되는 동시 연결의 수와 상이하게 설정하는 수단을 포함한다.

[0071] 일부 실시예에서, 2개 이상의 동시 전송 연결 각각은 상기 데이터 전달을 확인(confirm)하기 위해 수신확인 응답을 생성하는 전송 프로토콜을 사용하고, 상기 요청된 데이터 전달은 상기 요청된 데이터가 전달되었다는 수신확인 응답을 수신한 후에 완료되도록 결정된다. 일부 경우에 이 프로토콜은 전송 제어 프로토콜(TCP)이다.

[0072] 본 발명의 일 실시예에서, 상기 제1 및 제2 컴퓨터는 이들 컴퓨터 간에 다수의 동시 연결을 수립하도록 구현된 트래픽 분배 모듈을 포함하고, 상기 데이터 전달 요청은 상기 클라이언트 컴퓨터에 의해 생성되고, 상기 클라이언트 컴퓨터는 상기 제2 컴퓨터와 상기 제1 컴퓨터 간에 동시 연결을 통해 데이터를 요청하고 수신하도록 하나 이상의 전송 연결을 개방한다. 이 실시예에 따르면, 상기 요청된 데이터의 전달은 상기 클라이언트 컴퓨터가 상기 데이터를 요청하고 수신하기 위해 개방한 연결을 닫은 후에 완료되도록 결정된다. 일부 경우에 상기 클라이언트 컴퓨터는 상기 제2 컴퓨터이다. 하나 이상의 경우에 상기 제2 컴퓨터는 상기 클라이언트 컴퓨터에 의해 개방된 전송 연결을 종료하도록 구현된 프록시를 포함한다.

[0073] 일부 실시예에서, 상기 동시 전송 연결 중 적어도 2개는 동일한 물리적 데이터 링크를 통해 개방된다. 일부 경우에 이러한 연결은 동일한 출발지 및 목적지 네트워크 어드레스들을 공유한다.

[0074] 일부 실시예에서, 제1 컴퓨터는 상기 제1 또는 제2 데이터 전달 중 적어도 하나의 데이터 전달의 크기의 추정값을 획득하고, 추정된 크기가 증가함에 따라 대응하는 전달을 위해 사용되는 동시 네트워크 연결의 수를 증가시킨다. 일부 경우에, 상기 데이터 전달의 크기는 상기 제1 요청 그룹으로부터 하나 이상의 네트워크 식별자를 획득함으로써 추정되고, 각각의 식별자는 도메인 명과 목적지 네트워크 어드레스 중 적어도 하나를 포함한다.

[0075] 다른 실시예에서, 하나 이상의 동시에 개방된 연결 각각이 적어도 일부 데이터가 전달되었다는 수신확인 응답을 수신하기 전에 송신될 수 있는 트래픽의 양을 제한하는 제어 윈도우와 관련된 경우, 상기 제1 컴퓨터는 하나 이상의 제어 윈도우의 추정된 크기를 획득하고, 이 크기가 증가함에 따라 데이터 전달에 사용되는 동시 네트워크 연결의 수를 감소시킨다. 일부 경우에, 상기 제어 윈도우는 전송 프로토콜에 의해 사용되는 혼잡 윈도우와 수신자 윈도우 중 가장 작은 것이다.

[0076] 일부 실시예에서, 상기 데이터 전달을 위해 사용되는 동시 연결의 수는 상기 요청된 데이터의 제1 그룹의 전달을 완료한 후 그리고 제2 요청 그룹이 제출되기 전에 변경된다. 일부 경우에, 제1 요청된 데이터의 전달 후에 데이터 전달에 사용되는 동시 연결의 수가 감소하고 나서 제2 그룹의 데이터 요청이 제출되기 전에 미리 한정된

시간 구간을 초과하는 비활동 기간이 뒤따른다.

[0077] 다른 실시예에서, 상기 데이터 전달에 사용되는 동시 연결의 수는 기존 연결을 닫거나 새로운 연결을 열지 않고도 하나 이상의 이전에 열린 동시 연결을 통해 적어도 일부 요청된 데이터의 전송을 일시 정지 또는 재개함으로써 변경된다. 일부 경우에 상기 제1 서버로부터 상기 제2 서버로 데이터를 전송하는데 사용된 동시 연결의 수는 상기 제2 서버로부터 상기 제1 서버로 데이터를 전송하는데 사용된 동시 연결의 수와 상이하게 변경된다. 일부 구현예에서, 상기 제1 서버로부터 상기 제2 서버로 데이터를 전송하는데 사용된 동시 연결의 수만이 변경되는 반면, 상기 제2 서버로부터 상기 제1 서버로 데이터를 전송하는데 사용된 동시 연결의 수는 동일하게 유지된다.

[0078] 일부 실시예에서, 상기 제1 컴퓨터는 프록시 서버와 VPN 서버 중 적어도 하나를 포함한다.

[0079] 본 발명의 일 양태에서, 제1 컴퓨터는 상기 복수의 동시 전송 연결을 통해 제2 컴퓨터로 데이터를 전달하기 위한 하나 이상의 요청 그룹을 수신하고, 상기 요청된 데이터를 전달하는 것은 적어도 하나의 다른 요청을 제출하기 전에 완료되어야 한다. 제1 컴퓨터는 요청된 데이터를 전달하기 시작하고 나서, 요청된 데이터의 전달이 완료되기 전에 데이터를 전달하는데 사용된 동시 연결의 수를 변경한다.

[0080] 일부 실시예에서, 상기 제1 컴퓨터는 전달될 데이터의 잔량 및 이 전달에 사용된 동시 전송 연결 중 적어도 하나의 전송 연결을 통해 데이터 전송의 적어도 하나의 파라미터 중 적어도 하나에 대한 추정값을 획득하고, 추정된 값이 변함에 따라 동시 연결의 수를 변경한다. 일부 경우에 제1 컴퓨터는 이 양이 감소함에 따라 데이터의 잔량을 전달하는 데 사용되는 동시 연결의 수를 감소시킨다. 일부 경우에 전달될 데이터의 잔량은 제2 컴퓨터로 송신되기 전에 제1 컴퓨터의 하나 이상의 버퍼에 저장된 요청된 데이터의 양으로부터 추정된다. 일부 구현예에서, 제1 컴퓨터는 프록시와 VPN 서버 중 적어도 하나를 포함하고, 상기 버퍼는 임시 데이터 저장을 위해 프록시 또는 VPN 서버에 의해 사용된다.

[0081] 다른 경우에, 전송 연결을 통해 데이터를 전송하는 파라미터들 중 적어도 하나의 파라미터는 제어 윈도우이고, 이 제어 윈도우는 적어도 일부 데이터가 전달되었다는 수신확인 응답을 수신하기 전에 송신될 수 있는 트래픽의 양을 제한하고, 잔여 데이터를 전달하는데 사용되는 동시 연결의 수는 이들 윈도우 중 하나 이상의 윈도우의 크기가 증가함에 따라 감소된다.

[0082] 일부 경우에, 전송 연결을 통해 데이터를 전송하는 파라미터들 중 적어도 하나의 파라미터는 이 연결을 통한 데이터 손실의 양이고, 잔여 데이터를 전달하는데 사용되는 동시 연결의 수는 하나 이상의 전송 연결을 통한 데이터 손실률에 따라 증가된다.

[0083] 일부 실시예에서, 데이터 전송에 사용되는 동시 연결의 수는 기존 연결을 닫거나 새로운 연결을 열지 않고 하나 이상의 이전에 열려 있는 동시 연결을 통한 데이터 전달을 일시 중지하거나 재개함으로써 요청 그룹이 제출된 후에 변경된다. 일부 경우에 제1 서버로부터 제2 서버로 데이터를 전송하는데 사용된 동시 연결의 수는 제2 서버로부터 제1 서버로 데이터를 전송하는데 사용된 동시 연결의 수와 다르게 변경된다. 일부 구현예에서, 제1 서버로부터 제2 서버로 데이터를 전송하는데 사용되는 동시 연결의 수만이 변경되는 반면, 제2 서버로부터 제1 서버로 데이터를 전송하는데 사용되는 동시 연결의 수는 동일하게 유지된다.

[0084] 일부 실시예에서, 동시 전송 연결들 중 적어도 2개는 동일한 물리적 데이터 링크를 통해 개방되고, 사용된 동시 연결의 수를 변경하는 것은 이 데이터 링크를 통해 데이터를 전송하는데 사용된 동시 연결의 수를 변경하는 것을 포함한다.

[0085] 본 발명의 다른 양태에서, 상기 제1 컴퓨터는 상기 제1 컴퓨터와 복수의 동시 전송 연결을 통해 제2 컴퓨터로 데이터를 전달하기 위한 2개 이상의 요청 그룹을 수신하고, 여기서 요청된 전달의 제1 그룹은 제2 요청 그룹을 제출하기 전에 완료되어야 한다. 상기 제1 컴퓨터는 또한 상기 제1 컴퓨터와 복수의 동시 전송 연결을 통해 제3 컴퓨터로 데이터를 전달하기 위한 2개 이상의 요청 그룹을 수신하고, 여기서 요청된 전달의 제1 그룹은 제2 요청 그룹을 제출하기 전에 완료되어야 한다. 이 양태에서, 상기 제1 컴퓨터로부터 상기 제2 및 제3 컴퓨터로 연결하는 하나 이상의 파라미터 간에 차이가 있다면, 상기 제1 컴퓨터는 상기 제3 컴퓨터에 사용하는 동시 연결의 수와는 다른 동시 연결의 수를 상기 제2 컴퓨터에 사용한다.

[0086] 일부 실시예에서, 이것은 제1 컴퓨터로부터의 거리, 왕복 시간 및 네트워크의 품질 중 적어도 하나의 차이이다. 일부 경우에 제1 컴퓨터까지의 거리가 증가함에 따라 또는 제1 컴퓨터까지의 왕복 시간이 증가함에 따라 동시에 사용되는 연결의 수가 증가된다. 다른 경우에, 제3 및 제1 컴퓨터 간의 네트워크의 품질에 비해 제2 컴퓨터와 제1 컴퓨터 간의 네트워크의 품질이 증가함에 따라 동시에 사용되는 연결의 수는 감소된다. 다른 경우에, 동시

에 사용되는 연결의 수는 하나 이상의 전송 연결에 걸친 데이터 손실률이 감소함에 따라 감소된다.

[0087] 일부 구현예에서, 하나 이상의 전송 연결은 송신된 데이터의 적어도 일부에 대한 수신확인 응답을 수신하기 전에 송신될 수 있는 데이터의 양을 제한하도록 제어 윈도우를 설정하는 전송 프로토콜을 사용하고, 동시에 사용되는 연결의 수는 하나 이상의 제어 윈도우의 크기가 증가함에 따라 감소한다. 일부 실시예에서, 전송 프로토콜은 전송 제어 프로토콜(TCP)이고, 제어 윈도우는 수신자 윈도우와 혼잡 윈도우 중 가장 작은 것이다.

[0088] 본 발명의 일 양태에서, 컴퓨터 네트워크를 통해 컴퓨터들 간의 다수의 동시 전송 연결의 통합된 처리량을 증가시키는 시스템은, 하나 이상의 요청을 수신하고, 제1 컴퓨터와 제2 컴퓨터 간의 복수의 동시 전송 연결을 통해 상기 제1 컴퓨터로부터 상기 제2 컴퓨터로 전송 요청된 데이터를 획득하도록 구현된 트래픽 관리 모듈로서, 상기 제2 컴퓨터로부터 적어도 하나의 다른 요청을 수신하기 전에 상기 요청된 데이터의 전송이 완료되어야만 하는, 상기 트래픽 관리 모듈, 상기 요청된 데이터를 2개 이상의 그룹으로 분할하고, 2개 이상의 데이터 그룹을 상이한 전송 연결들로 포워딩하도록 구현된 트래픽 분배 모듈, 및 하나 이상의 사용된 연결에 대한 데이터 그룹 크기의 바람직한 범위를 획득하고, 바람직한 범위 내에서 하나 이상의 사용된 연결에 대한 데이터 그룹 크기를 설정하도록 구현된 트래픽 튜닝 모듈을 포함한다.

[0089] 일부 실시예에서, 트래픽 튜닝 모듈은 이용 가능한 전송 연결의 최대 수를 획득하고, 각 사용된 연결에 대한 데이터 그룹 크기를 설정하는 반면에, 트래픽 분배 모듈은, 요청된 데이터의 전달이 완료될 때까지, 또는 상기 요청된 전달의 크기가 이용 가능한 연결에 대해 바람직한 그룹 크기의 합 미만으로 떨어질 때 이용 가능한 연결의 최대 수보다 더 적은 수의 전송 연결을 사용하여, 사용된 전송 연결의 수가 이용 가능한 연결의 최대 수에 도달할 때까지, 상이한 전송 연결들로 각 데이터 그룹을 포워딩한다. 일부 경우에 이용 가능한 전송 연결의 최대 수는 제1 컴퓨터와 제2 컴퓨터 간에 현재 열려 있는 동시 연결의 수로 설정되고, 현재 열려 있는 연결들 중 적어도 하나는 요청된 데이터를 전달하는 데 사용되지 않는다. 일부 구현예에서, 사용된 전송 연결의 수는 제2 컴퓨터로 전달되는 데이터의 양이 증가함에 따라 증가된다.

[0090] 다른 실시예에서, 상기 트래픽 분배 모듈은 하나 이상의 임시 버퍼로부터 적어도 2개의 다른 요청에 응답하여 수신된 데이터를 획득하고, 동일한 요청에 응답하여 획득된 데이터만을 포함하는 적어도 하나의 데이터 그룹을 형성하고, 이후 이 데이터 그룹을 단일 전송 연결로 분배한다. 일부 구현예에서, 단일 요청에 응답하여 단일 연결로 송신된 데이터 그룹은 순서화된 시퀀스로 제공된다. 일부 실시예에서, 상기 트래픽 분배 모듈은 상이한 전송 연결들로 송신된 데이터 그룹에 헤더를 부가하고, 적어도 하나의 헤더는 데이터 그룹의 크기 및 단일 요청에 대응하는 데이터의 식별자를 포함한다. 일부 경우에 하나 이상의 데이터 그룹은 HTTP(Hyper Text Transport Protocol) 버전 2 이상에 따라 포맷된다.

[0091] 다른 실시예에서, 모든 요청된 데이터가 동시 전송 연결로 포워딩될 때까지 다음 데이터 그룹이 다음 전송 연결로 포워딩되기 전에 각 데이터 그룹은 하나의 전송 연결로 포워딩된다.

[0092] 다른 실시예에서, 데이터 그룹 크기의 바람직한 범위는 적어도 2개의 상이한 동시 사용 연결에 대해 다르게 설정된다.

[0093] 또 다른 실시예에서, 하나 이상의 전송 연결은 송신된 데이터의 적어도 일부에 대한 수신확인 응답을 수신하기 전에 송신될 수 있는 데이터의 양을 제한하도록 제어 윈도우를 설정하는 전송 프로토콜을 사용하고, 바람직한 데이터 그룹 크기의 범위는 하나 이상의 제어 윈도우의 크기가 증가함에 따라 증가한다. 일부 경우에, 전송 프로토콜은 전송 제어 프로토콜(TCP)이고, 제어 윈도우는 수신자 윈도우와 혼잡 윈도우 중 가장 작은 것이다. 다른 경우에, 적어도 하나의 연결에 대한 바람직한 범위의 최대값은 이 연결에 대한 제어 윈도우의 크기 이하이도록 설정된다. 다른 경우, 적어도 하나의 연결에 대한 바람직한 범위의 최소값은 이 연결에 대한 제어 윈도우의 크기의 절반보다 더 크도록 설정된다.

[0094] 다른 실시예에서, 하나 이상의 전송 연결은 적어도 일부 이전에 송신된 데이터의 수신확인 응답을 수신하지 않고 연결을 통해 송신될 수 있는 데이터의 양을 제한하는 전송 프로토콜을 사용하고, 바람직한 데이터 크기의 범위는 이전에 송신된 데이터에 대한 수신확인 응답을 수신하기 전에 적어도 하나의 연결을 통해 데이터 그룹을 송신하도록 설정된다.

[0095] 다른 실시예에서, 바람직한 데이터 크기의 범위는 제1 컴퓨터와 제2 컴퓨터 간의 1 왕복 시간에 적어도 하나의 연결을 통해 데이터 그룹을 송신하도록 설정된다.

[0096] 다른 실시예에서, 동시 전송 연결들 중 적어도 2개는 동일한 물리적 데이터 링크를 통해 개방된다. 일부 경우에

동시 전송 연결들 중 적어도 2개는 동일한 출발지 및 목적지 네트워크 어드레스들을 가진다.

[0097] 일부 실시예에서, 트래픽 관리 모듈은 프록시와 VPN 중 적어도 하나를 포함한다.

[0098] 본 발명의 다른 양태에서, 컴퓨터 네트워크를 통해 컴퓨터들 간의 다수의 동시 전송 연결의 통합된 처리량을 증가시키는 시스템은 적어도 상기 트래픽 관리 모듈 및 상기 트래픽 분배 모듈을 포함한다. 상기 트래픽 관리 모듈은 복수의 동시 열려 있는 연결을 통해 제1 컴퓨터로부터 제2 컴퓨터로 데이터를 전송하기 위한 2개 이상의 요청을 수신하고, 2개의 상이한 요청에 응답하여 상기 데이터를 획득하고, 상이한 요청에 대한 응답을 포함하는 2개 이상의 데이터 프레임을 형성하고, 여기서 각 데이터 프레임은 단일 요청에 대응하는 데이터를 포함한다. 상기 트래픽 분배 모듈은 적어도 2개의 데이터 프레임 각각이 단일 전송 연결로 포워딩되도록 상기 데이터 프레임을 2개 이상의 상이한 전송 연결로 포워딩하여, 각 전송 연결을 통해 적어도 하나의 데이터 프레임을 전송한다.

[0099] 일부 실시예에서, 트래픽 관리 모듈은 2개 이상의 데이터 프레임을, 2개 이상의 데이터 프레임의 시퀀스를 저장하는 동일한 큐로 다중화하고, 이후 트래픽 분배 모듈은 적어도 2개의 데이터 프레임을 별개의 전송 연결로 포워딩하기 전에 상기 동일한 큐로부터 적어도 2개의 데이터 프레임 각각을 획득한다.

[0100] 다른 실시예에서, 사용된 전송 연결의 수는 제2 컴퓨터에 전달되는 요청된 데이터의 양이 증가함에 따라 증가한다.

[0101] 또 다른 실시예에서, 요청된 데이터는 적어도 하나의 다른 데이터 전달 요청을 수신하기 전에 동시 전송 연결을 통해 전달되어야 하고, 하나 이상의 데이터 프레임의 크기는 전달되어야 하는 데이터의 잔량이 감소함에 따라 증가된다.

[0102] 다른 실시예에서, 하나 이상의 데이터 프레임의 크기는 제1 컴퓨터와 제2 컴퓨터 간의 하나 이상의 전송 연결의 처리량이 증가함에 따라 증가된다. 일부 경우에, 처리량이 증가하는 것은 하나 이상의 사용된 전송 연결로부터 이전에 송신된 데이터에 대한 수신확인 응답을 수신하기 전에 패킷 손실률이 감소하는 것과 송신이 허용된 데이터의 양이 증가하는 것 중 적어도 하나를 검출하는 것에 의해 결정된다.

[0103] 도 1은 본 발명의 일 예시적인 구현예를 도시한다. 클라이언트 컴퓨터(110)는 근거리 프록시 또는 VPN(120)을 통해 원거리 프록시 또는 VPN(170)으로 데이터를 전달하기 위한 2개 이상의 요청 그룹을 발행하며, 원거리 프록시 또는 VPN은 고대역폭 파이프(180)를 통해 컨텐츠 제공자(190)에 요청을 전달한다. 제시된 예에서, 근거리 프록시 또는 VPN(120)은 원거리 프록시 또는 VPN(170)으로부터 상당히 멀리 떨어져 위치하여, 두 컴퓨터 간의 패킷 손실률을 증가시킨다. 이러한 손실이 처리율을 저하시키는 것을 방지하기 위해, 근거리 프록시 또는 VPN(120)은 근거리 다중 경로 전송 관리자(130)와 결합될 수 있다. 원거리 프록시 또는 VPN(170)은 또한 원거리 다중 경로 전송 관리자(160)와 결합될 수 있고, 근거리 전송 관리자와 원거리 전송 관리자 간의 트래픽은 다수의 동시 연결(140, 150)을 통해 분배된다.

[0104] 도시된 실시예에서, 동시 전송 연결(140)은 요청된 데이터를 전송하는데 사용되는 반면; 전송 연결(150)은 개방된 채로 있지만, 요청된 데이터를 전송하는데 사용되지는 않는다. 이것은 새로운 연결을 열거나 구 연결을 닫는 시간을 소비하지 않고 사용된 연결(140)의 수를 증가시키거나 감소시키는 변경을 할 수 있게 한다. 다른 실시예에서, 전송 연결은 데이터를 전송하는데 바로 사용되거나 또는 사용되지 않고 언제든지 닫히거나 열릴 수 있다.

[0105] 일부 구현예에서, 적어도 일부의 전송 연결(140, 150)은 근거리 컴퓨터(110)와 원거리 컴퓨터(190) 간의 라우트(route)의 적어도 일부를 커버하는 동일한 물리적 데이터 링크를 통해 개방된다. 예를 들어, 다수의 연결은 동일한 마지막 마일 ISP(internet service provider) 또는 무선 서비스 제공자를 사용함으로써 개방될 수 있다. 일부 실시예에서, 다수의 연결 간을 구별하기 위해 출발지 포트만을 사용하여 동일한 출발지 및 목적지 IP 어드레스들 간에 다수의 연결이 개방된다. 이러한 구현예는 다수의 연결이 동일한 중간 라우터(router)를 통과하면서 클라이언트와 컨텐츠 제공자 간에 동일한 라우트를 사용할 확률을 증가시킨다. 일부 경우에, 이 접근법은 이미 열려 있는 연결의 네트워크 특성으로부터 왕복 시간 또는 패킷 손실률과 같은 새로운 연결의 네트워크 특성을 외삽하는 데 사용되거나, 또는 기존 연결로부터 획득된 제어 윈도우로부터 새로운 연결의 제어 윈도우를 미리 설정하는데 사용된다.

[0106] 다른 구현예에서, 2개 이상의 동시 연결은 상이한 데이터 링크(예를 들어, Wi-Fi 및 셀룰러 연결 모두)를 통해 열리거나 또는 목적지 IP 어드레스들의 차이로 인해 상이한 라우트를 통과할 수 있다. 이것은 개별 파라미터에 따라 데이터를 전송하는데 향상된 연결 세트를 선택하는 것을 도와줄 수 있고; 다수의 연결이 동일한 마지막 마일 ISP를 통해 동일한 원거리 서버와 수립되더라도, 동일한 서버를 가리키는 상이한 목적지 IP 어드레스를 사용

하면 두 컴퓨터(110, 190) 간에 패킷이 취하는 다른 라우트로 인해 각 연결의 네트워크 파라미터를 변경할 수 있다.

[0107] 일부 구현예에서, 모듈(120)과 모듈(170)은 대응하여 프록시 클라이언트 및 프록시 서버이다. 클라이언트(110)에 의해 개방된 연결은 근거리 프록시 클라이언트(120)에 의해 종료되고, 이 근거리 프록시 클라이언트는 원거리 프록시 서버(170)와 전송 관리자(130)와 전송 관리자(160) 간에 수립된 다수의 전송 연결을 통해 요청 페이로드를 송신하고 응답 페이로드를 수신한다. 일부 실시예에서, 전송 연결(140, 150)은 전송 제어 프로토콜(TCP)과 같은 신뢰성 있는 전달 프로토콜을 사용하여, 페이로드의 전달을 보장하고 동시 전송 연결을 사용하는 것으로부터 통합된 제어 윈도우를 증가시키는 이익을 얻을 수 있다.

[0108] 다른 구현예에서, 모듈(120)과 모듈(170)은 대응하여 가상 사설망(VPN) 클라이언트 및 VPN 서버이다. 클라이언트(110)에 의해 생성된 데이터 패킷은 근거리 VPN 클라이언트(120)에 의해 사용자 공간으로 이동된다. 근거리 트래픽 관리자(120)는 아웃고잉되는 패킷을 동시 전송 연결(140)로 캡슐화한다. 원거리 트래픽 관리자(160)는 원래의 패킷을 추출하고, 원거리 VPN 서버(170)와 함께 이들 패킷을 컨텐츠 제공자(190)에 송신한다. 컨텐츠 제공자(190)에 의해 반환된 데이터는 역순으로 처리된다. 이 경우 동시 전송 연결은, 대개는 임의의 중개 라우터의 큐의 크기를 감소시키고 데이터 링크 세그먼트들 간의 대역폭 변경을 완화하는 것에 의해 처리량을 향상시키는데 도움을 준다. 특히 이러한 중개 라우터는 보통 연결당 하나의 큐를 유지하고, 각 큐의 크기를 감소시키면 오버플로우로 인한 대기 시간(latency)과 패킷 손실을 감소시키는데 도움을 준다. 클라이언트(110)가 신뢰성 있는 전달 프로토콜을 사용하여 다수의 연결을 개방하면, VPN 클라이언트(120)와 VPN 서버(170) 간에 다수의 동시 연결은 모든 클라이언트 연결을 단일 큐에 집중시킴으로써 야기되는 성능 저하를 방지하는데 도움을 준다.

[0109] 일부 구현예는 병렬로 사용되는 프록시 서버 및 VPN 서버(170)뿐만 아니라 프록시 클라이언트 및 VPN 클라이언트(120)를 모두 포함할 수 있다. 일부 실시예에서, 클라이언트(110)로부터 유래하는 모든 TCP 트래픽은 클라이언트 프록시(120)에 의해 종료되고, 그 페이로드는 다수의 TCP 전송 연결(140)을 통해 송신된다. 이러한 실시예에서, 클라이언트(110)로부터 유래하는 모든 UDP(user datagram protocol) 트래픽은 VPN 클라이언트(120)에 의해 캡슐화되고, 다수의 UDP 연결을 통해 송신된다.

[0110] 일부 구현예에서, 근거리 프록시 또는 VPN(120)은 다중 경로 전송 관리자(130)와 동일한 컴퓨터 상에 상주한다. 일부 실시예에서, 이들은 클라이언트(110)와 동일한 컴퓨터 상에 상주한다. 다른 구현예에서, 이들 구성 요소(110, 120, 130)들 중 임의의 하나 또는 두 개의 구성 요소는 다른 컴퓨터 상에 상주할 수 있다.

[0111] 일부 구현예에서, 원거리 프록시 또는 VPN(170)은 원거리 다중 경로 전송 관리자(160)와 동일한 컴퓨터 상에 상주하고; 또 다른 구현예에서 이들은 다른 컴퓨터 상에 상주할 수 있다.

[0112] 프록시 또는 VPN 모듈(120, 170)은 컨텐츠 서버(190)가 다수의 병렬 연결을 통해 동일한 데이터 파일의 전달을 지원하지 않을 때 유용하다.

[0113] 대안적인 실시예에서, 컨텐츠 서버(190)가 이러한 지원을 할 수 있는 경우, 클라이언트(110)는 프록시 또는 VPN 클라이언트(120)를 사용하지 않고 근거리 다중 경로 전송 관리자(130)에 직접 트래픽을 송신한다. 이러한 실시예에서, 컨텐츠 제공자(190)는 프록시 또는 VPN 서버(170)를 사용하지 않고 원거리 다중 경로 전송 관리자(160)로부터 요청을 수신하고 이 전송 관리자로 데이터를 송신한다. 이 실시예는, 예를 들어, 클라이언트(110)와 컨텐츠 제공자(190)가 다중 경로 TCP 프로토콜 확장을 지원하는 경우 실현될 수 있다.

[0114] 도 2는 근거리 다중 경로 전송 관리자(130)의 일 예시적인 구현예를 도시한다. 이 구현예에서, 근거리 다중 경로 전송 관리자(130)는 클라이언트(110)와 통신하는 근거리 프록시 또는 VPN(120)에 모두 연결된, 다중 경로 트래픽 분배기(220) 및 다중 경로 전송 튜너(230)를 포함한다. 근거리 다중 경로 전송 관리자(130)는 TCP 스택(210)과 데이터를 교환하며, TCP 스택은 원거리 다중 경로 전송 관리자(160)와의 동시 전송 연결을 유지한다. 도시된 실시예에서, 다중 경로 트래픽 분배기(220)는 다중 경로 전송 튜너(230)로부터 기존의 연결을 통해 커맨드를 수신하여 연결을 개방하거나 폐쇄하거나 또는 트래픽을 일시 정지하거나 또는 재개한다. 새로운 연결이 열리거나 닫혀야만 한다면, 다중 경로 트래픽 분배기(220)는 커맨드를 TCP 스택(210)으로 송신하고; 기존 연결의 사용이 일시 중지되거나 재개되어야 한다면, 다중 경로 트래픽 분배기는 TCP 스택(210)에 알릴 필요 없이 분배 알고리즘을 변경한다.

[0115] 설명된 구현예에서, 다중 경로 전송 튜너(230)는, 프록시 또는 VPN(120)으로부터 오는 데이터로부터 사용되는 동시 연결의 수를 변경하는 것을 도와주는 정보를 획득한다. 이 예에서, 튜너는 도메인(호스트 명), IP 어드레스, 포트 및 HTTP 헤더를 포함하지만 이에 한정되지 않는, 클라이언트(110)에 의해 발행된 요청에 관한 정보를

수신한다. 일부 실시예에서, 이 정보는 알려진 크기의 웹 사이트의 다운로드를 나타내는 요청 패턴을 검출하는 데 사용된다. 크기 추정은 아래에 설명된 바와 같이 연결의 수를 변경하는 데 사용된다. 추가적인 정보는 컨텐츠 요청에 응답하여 제공된 컨텐츠 크기, 또는 TCP 스택(210)과 같은 다른 출발지로부터 오는 정보(이는 근거리 다중 경로 전송 관리자가 충분한 시스템 권한을 갖는 경우에만 가능함)를 더 포함할 수 있다.

[0116] 대안적인 구현예에서, 다중 경로 전송 튜너(230)는 원거리 다중 경로 전송 관리자(160)까지의 왕복 시간을 관찰함으로써 또는 원거리 전송 관리자(160)의 API를 호출한 이후에 제1 바이트가 전달되는 시간을 관찰함으로써 사용된 연결(140)의 수를 설정하거나 변경하는 결정을 한다.

[0117] 또 다른 구현예에서, 다중 경로 전송 튜너(230)는, 예를 들어, 네트워크 혼잡의 알려진 패턴과 상관될 때, 연결 세션의 시작 이후의 경과 시간, 사용자 상태, 또는 하루 중 시간에 기초하여 사용된 연결의 수를 설정하거나 변경할 수 있다.

[0118] 사용된 동시 연결의 수를 증가시킬지 감소시킬지 여부를 결정하는데 필요한 데이터를 수신하는 것에 더하여, 다중 경로 전송 튜너(230)는 또한, 예를 들어, 원거리 다중 경로 전송 관리자(160)에 의해 노출된 API(application programming interface)와 지속적인 TCP 연결을 통해 피드백을 수신함으로써 원거리 다중 경로 전송 관리자(160)로부터 오는 제어 정보를 프록시 또는 VPN(120)으로부터 수신한다. 다중 경로 전송 튜너(230)는 또한 전송 연결을 열거나 닫을지 및 일시 중지하거나 재개할지에 관한 결정을 원거리 다중 경로 전송 관리자(160)에 알려줄 수 있다. 다시, 이것은 원거리 다중 경로 전송 관리자(160)를 참조하는 알려진 네트워크 어드레스에 API 호출을 발행하는 것을 포함할 수 있다. 일부 실시예에서, 연결은 근거리 다중 경로 전송 관리자(130)에 의해서만 개방될 수 있다. 그러나, 전형적인 실시예에서, 근거리 및 원거리 다중 경로 전송 관리자(130, 160)는 연결을 일시 중지하거나 재개할 수 있다.

[0119] 일부 구현예에서, 근거리 다중 경로 전송 관리자는 데이터를 송신하는데 원거리 다중 경로 전송 관리자와는 다른 상이한 연결을 사용한다. 이 구현예에서 일 방향으로 사용된 전송 연결의 수는 다른 방향으로 사용된 전송 연결의 수와는 다르고; 일부 연결은 양방향으로 트래픽을 교환하는 데 사용되는 반면, 다른 연결은 일 방향으로만 트래픽을 교환하는 데 사용된다.

[0120] 다른 실시예에서, 다중 경로 트래픽 분배기(220)는 다중 경로 전송 튜너(230)와 결합되거나 또는 TCP 스택을 변경함(일반적으로 루트 액세스 허가를 필요로 함)으로서 구현될 수 있다. 프록시 또는 VPN(120)을 통해 통신하는 대신에, 근거리 다중 경로 트래픽 관리자(130)는 클라이언트(110)와 통합될 수 있다.

[0121] 도 3은 원거리 다중 경로 전송 관리자(160)의 일 예시적인 구현예를 도시한다. 이 구현예에서, 원거리 다중 경로 전송 관리자(160)는 컨텐츠 제공자(190)와 통신하는 원거리 프록시 또는 VPN(170)에 모두 연결된, 다중 경로 트래픽 분배기(320) 및 다중 경로 전송 튜너(330)를 포함한다. 원거리 다중 경로 전송 관리자(160)는 TCP 스택(310)과 데이터를 교환하고, 이 TCP 스택은 근거리 다중 경로 전송 관리자(130)와 동시 전송 연결을 유지한다. 도시된 실시예에서, 다중 경로 트래픽 분배기(320)는 다중 경로 전송 튜너(330)로부터 이미 열려 있는 연결을 통해 커맨드를 수신하여 트래픽을 일시 중지하고 재개한다. 그러나, 근거리 다중 경로 트래픽 분배기(220)와는 달리, 다중 경로 트래픽 분배기(320)는 일부 실시예에서 전송 연결을 열거나 닫지 않고 TCP 스택(310)으로부터 열린 또는 닫힌 연결에 관한 정보만을 수신하고, 이 정보를 다중 경로 전송 튜너(330)로 전달한다. 이것은 새로운 전송 연결이 클라이언트(110)에 근접한 모듈(130)에 의해 개시되는 경우를 반영한다. 다중 경로 전송 튜너(330)가 하나 이상의 연결을 열거나 닫기로 결정하면, 튜너는 이 정보를 프록시 또는 VPN(170)을 통해 근거리 다중 경로 전송 관리자(130)로 전달한다.

[0122] 다중 경로 전송 튜너(330)는 또한 근거리 전송 관리자(130)에 의해 개방되거나 폐쇄되거나, 일시 정지되거나 또는 재개되는 연결에 대한 피드백을 수신하고, 사용된 전송 연결(140)의 수를 설정하거나 변경하는 것을 돕기 위해 요청된 도메인 또는 IP 어드레스의 패턴과 같은 정보를 사용한다. 일부 실시예에서, 다중 경로 전송 튜너(330)는 클라이언트(110)가 이전에 열었던 하나 이상의 전송 연결을 닫을 것을 호출하였다고 결정한 것에 응답하여 요청 그룹에 대응하는 요청된 데이터의 전달이 완료되었다고 결정한다.

[0123] 일부 실시예에서, 프록시 또는 VPN(170)은 원거리 및 근거리 트래픽 전송 관리자들 간에 적어도 단일 연결을 위한 파이프보다 더 큰 처리량을 가질 수 있는 파이프(180)를 통해 컨텐츠 제공자(190)에 연결된다. 이것은 예를 들어 원거리 프록시 또는 VPN(170)이 근거리 프록시 또는 VPN(120)에보다 컨텐츠 제공자(190)에 훨씬 더 가까이 위치된 경우일 수 있다. 설명된 실시예에서, 처리량의 차이는 수신 큐(340)에 의해 완화된다. 수신 큐(340)는 데이터가 클라이언트(110)로 전송되기 전에 컨텐츠 제공자(190)로부터 다운로드된 데이터를 저장하는 버퍼로서

구현될 수 있다. 다중 경로 트래픽 분배기(320)는 다수의 동시 연결(140) 간에 데이터를 분배하기 전에 수신 큐(340)로부터 데이터를 판독한다.

[0124] 설명된 실시예에서, 다중 경로 전송 튜너(330)는 수신 큐(340)에 저장된 데이터의 크기에 의해 클라이언트에 전달될 잔여 데이터의 크기를 추정한다. 예를 들어, 단일 파일이 컨텐츠 제공자(190)로부터 고처리량 파이프(180)를 통해 로딩되고, 수신 큐(340)에 저장된 데이터의 크기가 시간에 따라 감소하기 시작하면, 이것이 요청된 전달에 대해 마지막 잔여 데이터라는 지시자인 것으로 해석될 수 있다. 다중 경로 전송 튜너(330)는 이 정보를 사용하여 전달이 시작된 후에 사용된 동시 전송 연결(140)의 수를 변경할 수 있다.

[0125] 다른 경우에, 클라이언트(110)는 다수의 요청을 다른 컨텐츠 제공자(190)로 발행할 수 있고, 여기서 응답은 클라이언트(110)로 전송되기 전에 하나 이상의 수신 큐(340)에 저장된다. 이 경우, 다중 경로 트래픽 튜너(330)는 각 수신 큐(340)에 저장된 데이터에 관한 정보를 수신할 수 있고, 각 수신 큐(340)가 최대 버퍼 크기보다 더 적은 데이터를 갖고, 저장된 데이터의 총량이 시간에 따라 감소할 때, 컨텐츠 제공자(190)로부터의 데이터 전달이 완료된 것으로 간주할 수 있다.

[0126] 도시된 실시예의 또 다른 양태에서, 다중 경로 전송 튜너(330)는, TCP 스택(310)으로부터 이미 열려 있는 전송 연결의 하나 이상의 파라미터, 예를 들어, 제어 윈도우의 크기(W)(또는, 예를 들어, TCP 전송 프로토콜에 의해 사용되는, 수신자 윈도우와 혼잡 윈도우의 크기), 근거리 TCP 스택(210)까지의 왕복 시간, 및 패킷 손실률과 같은 파라미터를 수신한다. 이 정보는 하나 이상의 열려 있는 전송 연결에 대해 수집될 수 있다. 이 정보는 시스템 관리자가 보안 서버 환경에서 더 높은 레벨의 액세스 권한을 설정할 수 있기 때문에 일반적으로 서버 측에서 이용될 수 있다.

[0127] 일부 구현예에서, 동일한 원거리 다중 경로 전송 관리자(160)는 상이한 클라이언트(110)들에 연결된 2개 이상의 근거리 다중 경로 전송 관리자(130)에 의해 개방된 연결을 수신한다. 이 경우에, 다중 경로 트래픽 튜너(330)는 각 클라이언트(110)와의 연결과 관련된 파라미터, 예를 들어, 왕복 시간을 평가하고, 다른 수의 동시 연결을 열거나 상이한 클라이언트(110)와 이미 열려 있는 다른 수의 연결을 사용할 것을 다른 근거리 전송 관리자(130)에게 조언한다. 각 클라이언트(110)와 초기 연결의 수가 설정된 후, 이 수는 데이터 전달이 완료되기 전에 (예를 들어, 제어 윈도우의 변하는 크기(W) 또는 수신 큐(340)에 남아 있는 데이터의 크기에 기초하여) 변경될 수 있다.

[0128] 대안적인 구현예에서, 다중 경로 트래픽 분배기(320)는 원거리 프록시 또는 VPN(170)을 통해 컨텐츠 제공자의 서버에 연결되지 않고 이 컨텐츠 제공자의 서버(190)와 통합될 수 있다. 이 경우, 컨텐츠 제공자의 서버는, 다중 경로 트래픽 분배기(320)에 도달하는 데이터가 암호화되어 있더라도, 요청된 전달의 크기에 대한 보다 정확한 정보를 제공할 수 있다.

[0129] 또 다른 구현예에서, 원거리 다중 경로 트래픽 관리자(160)는 다중 경로 트래픽 분배기(320) 및 다중 경로 전송 튜너(330)를 별도로 구현하지 않고 TCP 스택(310)과 통합될 수 있다.

[0130] 설명된 실시예에 따라, 다른 수의 동시 전송 연결(140)이 상이한 요청 그룹을 위한 데이터를 전달하거나, 다른 클라이언트가 동일한 컨텐츠를 요청한 경우에도 다른 클라이언트(110)에 사용될 수 있다. 또한, 각 경우에 사용되는 연결(140)의 수는, 요청 그룹이 발행된 후, 그러나 요청 그룹에 대응하는 데이터 전달이 완료되기 전에 변할 수 있다.

[0131] 이들 실시예는 다수의 목적을 위해 사용될 수 있는데, 예를 들어, 상이한 사용자 및 상이한 컨텐츠 요청에 대해 상이한 연결 패턴을 생성함으로써 외부 검열자로부터 프록시 또는 VPN 트래픽을 난독화하는 수단으로서 사용될 수 있다.

[0132] 또 다른 경우에, 전송 연결(140)의 수를 변경하는 것은 공유된 대역폭의 병목 현상에 대해 대역폭 소비량을 공정하게 분배하는데 사용될 수 있고, 즉 연결의 수는 다른 사용자에 의해 이용되지 않는 대역폭 용량이 있는 경우에만 또는 대역폭이 다른 사용자와 공유하지 않는 전용 자원인 경우에는 증가된다.

[0133] 일부 실시예에서, 동일한 파일이 다수의 연결을 통해 전송되더라도, 사용된 연결(140)의 수는 요청된 파일의 수에 따라 변경된다. 이것은 다수의 파일이 단일 프록시 또는 VPN 연결을 통해 전송되는 경우에 비해 프록시 또는 VPN 사용자에게 더 공정하게 한다.

[0134] 개시된 실시예의 하나의 기능은 클라이언트(110)와 컨텐츠 제공자(190) 간의 장거리에 걸쳐 처리량을 튜닝하는 것이다. 특히 연결이 전송 제어 프로토콜(TCP)과 같이 전달을 보장하는 전송 프로토콜을 사용하는 경우, 통상

송신자와 수신자 간에 다수의 전달-보장 전송 연결(140)에 걸쳐 데이터 전송을 확산시키면 일반적으로 총 처리량을 증가시키지만, 이 전송 프로토콜은 송신자와 수신자 간에 진행 중인(in-flight) 데이터의 양을, 전형적으로 수신자 윈도우(수신자에 의해 수용될 수 있는 데이터의 양)와 (송신자와 수신자 간의 링크에 너무 많은 트래픽이 과부하로 걸리는 것을 막기 위해 송신자에 의해 계산된) 혼잡 윈도우 중 가장 작은 것인 제어 윈도우의 크기로 제한한다.

[0135] 이 경우에, 각각의 새로운 연결(140)을 추가하면 적어도 통합된 혼잡 윈도우가 공유 링크에서 패킷 손실이 증가된 양에 의해 제한될 때까지 수신자 윈도우와 혼잡 윈도우의 통합된 크기를 증가시킨다. 혼잡 윈도우들을 통합하면 또한 랜덤한 패킷 손실의 영향을 완화하는 추가적인 장점이 있다. 예를 들어 패킷 손실 후 하나의 연결에 대한 혼잡 윈도우가 50% 감소하면 10개의 병렬 연결에 대해 통합된 혼잡 윈도우는 단 5%만 감소한다.

[0136] 연결(140)이 데이터를 전송함으로써 완전히 점유되면, 그 평균 처리량은 T = W/RTT로 추정될 수 있고, 여기서 W는 제어 윈도우의 크기이고, RTT는 평균 왕복 시간이다.

[0137] 데이터의 양(S)이 제어 윈도우(W)와의 연결(140)을 통해 전송되고, 이 양이 N개의 패킷으로서 송신되고 여기서 각 패킷은 S = N*MSS이도록 최대 세그먼트 크기(MSS ~= 1500 B)를 갖는다고 가정하면, 이러한 연결을 통해 이들 데이터를 전달하는 시간(t1)은 다음 식으로 추정될 수 있다:

[0138] t1 =(N/W) * RTT (1)

[0139] 동일한 양의 데이터가 M개의 병렬 연결을 통해 전송되는 경우, 각 연결은 (W*M 제어 윈도우를 사용하여 통합된 연결과 동등한) N/M개의 패킷을 전송하기만 하면 되고, 이들 데이터를 전달하는데 드는 시간(tM)은 M배 더 작아진다:

[0140] tM =(1/M)*(N/*W)*RTT (2)

[0141] 통합된 혼잡 윈도우가 공유 링크에 의해 제한되지 않고 모든 병렬 연결이 안정된 처리량(각 병렬 연결에 대한 평균(W)은 단일 연결에 대한 것과 동일함)에 도달하는 한, 이 기술 분야에 통상의 지식을 가진 자라면 M개의 병렬 연결을 통한 데이터 전송이 임의의 M > 1에 대한 단일 연결을 통한 연결보다 항상 더 빠르다고 결론내릴 수 있을 것이다.

[0142] 그러나, 이 결론은 N개의 패킷이 (혼잡 제어 알고리즘이 안정된 처리량을 유지할 수 있도록) 파이프를 통해 송신된 후에 데이터 전송이 계속된다고 암시적으로 가정한다. 실제로 요청된 데이터의 전달은 종종 추가적인 데이터 전송으로 진행하기 전에 확인되어야 한다. 예를 들어,

[0143] (1) 요청된 파일은 다음 파일을 자동으로 요청하기 전에 완전히 다운로드되고 처리되어야 한다(예를 들어, 브라우저는 자바스크립트 파일을 수신하고, 그 코드를 실행하고, 이 코드에 의해 생성된 요청을 발행한다).

[0144] (2) 사용자는 이전에 요청된 컨텐츠를 검토한 후에 다음 파일 또는 파일 그룹에 대한 요청을 발행한다(예를 들어, 웹 사이트의 로딩 후에 사용자가 링크를 클릭할 때까지 비활동 기간이 이어진다).

[0145] (3) 다음 전송으로 진행하기 전에 확인되어야 하는 데이터의 양은 보통 요청된 컨텐츠의 유형에 의존한다. 예를 들어, 단일 자바스크립트 파일 ~10 KB 내지 100 KB; 뉴스 웹 사이트 ~ 0.5 MB 내지 5 MB; 비디오 ~= 20 MB 내지 2000 MB.

[0146] 다음 데이터 전송으로 진행하기 전에 N개의 패킷의 전달을 확인하기 위한 요구 사항은 두 가지 방식으로 전달 시간을 변경한다.

[0147] 첫째, 연결 파이프는 진행 중인 패킷의 최대 수(W)로 항상 채워지는 것은 아닐 수 있다. 예를 들어, W = 100이지만 하나의 연결을 통해 단 10개의 패킷만이 송신되는 경우에도, 100개의 패킷에 대해서와 같이 패킷을 전달하고 확인하는데 드는데 동일한 시간(RTT)이 걸린다. 병렬 연결의 수가 증가하면 각 연결을 통해 송신된 패킷의 수가 W 미만으로 떨어져서 그 처리량이 감소할 수 있다.

[0148] 둘째, 임의의 연결에서 마지막 패킷이 손실될 확률은 병렬 연결의 수에 따라 증가한다. M개의 연결 중 임의의 연결이 마지막 패킷을 손실하면 이 패킷이 복구될 때까지 N개의 패킷이 모두 전달되는 것을 확인하는 것이 지연된다.

[0149] TCP의 일부로서 구현되는 것들을 포함하여 대부분의 혼잡 제어 프로토콜은 전송 연결을 통한 마지막 패킷이 손실될 때 추가적인 지연을 도입한다. 혼잡 제어 프로토콜은 대개 이러한 손실된 패킷을 재전송하기 전에 추가적

인 시간 동안 대기한다(꼬리 손실 지연). 제1 재전송 후 마지막 패킷이 복원되면 마지막 패킷을 재전달하는데 드는 총 시간은 꼬리_손실_지연 + RTT = TLDF * RTT (TLDF는 1보다 더 큰 시간 손실 지연 계수임)로 추정될 수 있다.

[0150] 리눅스 운영 체제의 일부 구현예에서, 시간 손실 지연은 재전송 타임아웃(RTO)과 동일하며, 즉 제1 재전송의 경우, 시간 손실 지연 ~=SRTT + 4*RTTVAR이고, 여기서 SRTT와 RTTVAR는 다수의 ACK에 걸쳐 각각 평균된 RTT 값 및 그 변동값이다. 예를 들어, RTTVAR/SRTT = 0.5이면, TLDF ~=4이다.

[0151] 리눅스 운영 체제에서 구현된 하나의 알려진 개선에서, 송신자는 꼬리 손실 프로브(tail loss probe: TLP)로 알려진 마지막 패킷의 복제본(duplicate)을 발행하여, 제1 재전송에 대한 시간 손실 지연을 2*SRTT(TLDF ~=3)로 감소시킨다.

[0152] 이들 예에서, TLDF는 손실된 패킷이 제1 재전송 시도 후에 재전달된다는 가정에서 추정된다. 재전송된 패킷이 또한 손실되면, 이것은 다음 RTO를 증가시키거나 또는 TLP로부터 RTO로 전환하여, 다음 재전송 시도를 더 지연시키고 평균 TLDF를 증가시킬 수 있다.

[0153] 단일 패킷 손실의 확률(p)이 임의의 연결에서 임의의 패킷에 대해 동일하다고 가정하면, M개의 연결 중 적어도 하나의 연결이 그 마지막 패킷을 잃을 확률은 P = 1-(1-p)^M이다(p*M < 0.5의 경우에는 ~=p*M이다).

[0154] 불완전하게 채워진 연결 파이프 및 꼬리 손실 지연을 고려한 후에, 단일 연결을 통해 및 M개의 병렬 연결을 통해 N개의 패킷을 전송하고 전달을 확인하는데 드는 시간(t1, tM)은 각각 다음과 같다:

[0155] t1 = 천장값(N/W)*RTT + p*TLDF*RTT (3)

[0156] tM = 천장값(1/M)*(N*W))*RTT + (1-(1-p)^M)*TLDF*RTT (4)

[0157] 여기서, 천장값(X)은 X보다 크거나 같으면서 X에 가장 가까운 정수(K >= X)를 반환하는 함수이다.

[0158] 패킷 손실이 없다면(p = 0), M은 임의의 M > N/W(단일 패킷을 전달하고 확인하는 것은 W개의 패킷에 대해서와 동일한 시간을 가짐)에 대해 작은 N/W: tM = RTT에 대해서도 통합된 처리량을 감소시키지 않으면서 높은 수로 미리 설정될 수 있다.

[0159] 그러나, 패킷 손실이 존재하면(p>0) M이 추정된 최적 값(M_최적)보다 더 커질 때 M개의 병렬 연결의 통합된 처리량이 감소하는데, 즉 다수의 연결을 통해 데이터 전송이 확산되면 적어도 하나의 마지막 패킷이 손실되어 꼬리 손실 지연이 발생할 확률이 증가된다.

[0160] 도 4a 및 도 4b는 본 발명에 따라 처리량이 향상된 것을 나타내는 그래프이다.

[0161] 도 4a는 N/W = 10 및 p와 TLDF의 3개의 조합에 대해 식(3, 4)으로부터 계산한, 각 연결(N/(M*W))에 대한 채움 비율(fill rate)이 증가함에 따른 처리량 가속도(tM/t1)의 변화를 도시한다. 이것은, p 및 TLDF의 실제 값에 대해, 각 연결을 통해 송신되는 데이터의 양이 1 RTT(N/(M*W) ~= 1) 내에 전달될 수 있는 최대 양과 같을 때 최상의 처리량 가속도가 달성되는 것을 보여준다. 예를 들어, p = 0.01 및 TLDF = 3의 경우 M = 10에서 최상의 처리량 가속도가 달성되며 이는 더 큰 M 값에 대해서보다 최대 2배 더 우수하다.

[0162] p = 0.05 및 TLDF = 4의 경우 그래프는 N/(M*W) ~= 2에서 국부 최소값을 나타내고; 여기서 p와 TLDF가 더 증가하면 이것은 절대 최소값이 되는데, 즉 패킷 손실이 높을 때(예를 들어, 네트워크 품질이 낮은 경우), 2 RTT에서 전달될 수 있는 연결당 데이터의 최대 양을 송신하여 평균 시간을 느리게 하지만 꼬리 손실이 발생할 확률을 감소시킴으로써 최상의 성능을 얻을 수 있다.

[0163] 도 4b는 p = 0.002, TLDF = 3 및 3개의 값의 N/W에 대해 식(3, 4)으로부터 계산한, 각 연결(N/(M*W))에 대한 채움 비율이 증가함에 따른 처리량 가속도(tM/t1)의 변화를 도시한다. 이것은 N/W의 넓은 범위에 대해 N/(M*W) ~= 1일 때 최상의 처리량 가속도가 계속 달성된다는 것을 보여준다. W가 동일하게 유지된다면 이것은 적어도 N/W의 도시된 범위에 대해 M의 추정된 최적 값이 N에 비례하여 증가한다는 것을 의미한다.

[0164] 수신자 윈도우가 혼잡 윈도우보다 더 크다면, W는 매티스 식(Mathis equation)(C ~= sqrt(3/2))으로부터 W=C/sqrt(p)로서 추정될 수 있다. 예를 들어, p = 0.002는 W=27에 해당하고, N/W = 100은 파일 크기 ~4MB에 해당한다.

[0165] 추정된 최적의 연결 수를 설명하는 식은 다수의 요인에 따라 식(3 내지 4)과 다를 수 있다. 예를 들어, 다른 연결들 간에 W개의 값이 변하는 정도는, 다른 패킷 손실률, 별개의 연결을 통해 이전에 송신된 데이터의 상이한

양, 또는 (예를 들어, 마지막 마일 무선 링크 내 랜덤한 재전송에 의해 야기된) 다른 연결의 가변 왕복 시간에 의해 야기된다.

[0166] 또한, 식은 데이터 분배 알고리즘에 의해 야기된 상이한 병렬 연결을 통해 송신되는 데이터의 양이 가변하는 정도에 따라서도 달라질 수 있다. 이는 예를 들어, 각 연결을 통해 송신되는 데이터의 양을 라운드 로빈(round-robin) 분배를 사용하는 대신에 이전에 관찰된 처리량에 비례하게 하는 것에 의해 이루어질 수도 있다.

[0167] 본 명세서에 개시된 실시예는 식 (3-4)의 정확한 형태에 의존하지는 않는다. 이들 식은, 본 발명의 일부 실시예에 의해 사용된 가정을 검증하고, 이 실시예가 특히 효과적일 수 있는 조건을 수립하기 위해 포함된 것이다. 예를 들어, 다른 데이터 전송으로 진행하기 전에 이전에 요청된 데이터의 전달이 확인되어야 하는 경우, 패킷 손실이 존재하는 경우 전달을 완료하고 확인하는데 드는 시간을 감소시키는 최적의 수의 병렬 연결이 있다. 또한 최적의 병렬 연결의 수는 전달을 완료하고 확인하는 데 드는 데이터의 양에 따라 증가한다. 단일 왕복 사이클에서 또는 특히 패킷 손실률이 높은 경우에는 2회의 왕복 사이클에서 송신되어 확인될 수 있는 최대 데이터 양을 전달하도록 각 병렬 연결이 요청될 때 일반적으로 최적의 병렬 연결의 수가 달성된다.

[0168] 도 5는 예를 들어, 데이터 트래픽의 파라미터를 모니터링하여 다수의 동시 연결의 처리량을 최적화하는, 예를 들어, 현저히 향상시키고, 이러한 파라미터를 사용하여 추정된 최적의 연결의 수를 설정하는 방법의 프로세스 흐름도를 도시한다. 도 5의 방법은, 근거리 다중 경로 전송 관리자(130)와 원거리 다중 경로 전송 관리자(160) 간의 연결의 속성 및 검색되는 데이터의 크기에 따라 상이한 컴퓨터가 상이한 수의 연결(140)을 사용할 수 있도록 원거리 다중 경로 전송 관리자(160)에 결합된 각 근거리 다중 경로 전송 관리자(130)에 대해 실행될 수 있다.

[0169] 설명된 방법에서, 다중 경로 전송 관리자(130, 160)는 수신 큐 내의 데이터 크기(R) 및 기존 전송 연결의 평균 제어 윈도우의 크기(W)와 같은 데이터 전송의 파라미터를 획득한다(단계 510). W의 크기는 왕복 시간, 패킷 손실률, TCP를 구현하는 두 컴퓨터 간의 네트워크 연결의 거리와 같은 요인, 또는 네트워크 연결의 품질을 나타내는 다른 요인에 따라 TCP 프로토콜에 의해 결정된다. 다중 경로 전송 관리자(130, 160)는 이것이 미리 한정된 시간 기간, 예를 들어, 1s 동안 트래픽 볼륨의 시작, 일시 중지, 정지 또는 급격한 감소(예를 들어, 50% 내지 90% 감소) 이후의 제1 데이터 전송인지 여부를 더 결정한다(단계 520).

[0170] 단계(520)의 시작 또는 일시 중지가 검출된 후에 클라이언트(110)로부터의 새로운 요청 그룹이 수신되면(단계 540), 패턴 또는 요청된 도메인 및 새로운 요청 그룹의 요청 헤더로부터 전달 크기(N)가 추정된다(단계 560). 그 다음, 동시 전송 연결의 수는 R, S 및 W의 추정된 값에 대해 새로운 요청 그룹에 대응하여 요청된 전달 시간을 최적화, 예를 들어, 상당히 향상시키도록 설정된다(단계 570).

[0171] 요청된 전달의 데이터 전송이 계속되는 동안, 다중 경로 전송 관리자(130, 160)들 중 적어도 하나는 R 및 W와 같은 전송 파라미터를 계속 평가한다(단계 520). 이 관리자가 천장값(R/(M*W))으로 추정된 요청된 전달 데이터의 잔량을 전달하는 데 사용되는 왕복 사이클의 수가 증가하는 것을 검출하면(단계 530), 사용된 연결(140)의 수는 요청된 전달이 완료될 때까지의 시간을 감소시키기 위해 증가된다(단계 550). 각 다중 경로 전송 관리자(130, 160)가 독립적으로 동작하는 한, 클라이언트(110)로부터 컨텐츠 제공자(190)로 데이터를 전송하는데 사용되는 전송 연결(140)의 수는 컨텐츠 제공자로부터 클라이언트(110)로 데이터를 전송하는데 사용된 전송 연결(140)의 수와 다를 수 있다는 것이 주목된다. 또한, 클라이언트(110)로부터 컨텐츠 제공자(190)로 데이터를 전송하는데 사용된 전송 연결(140)의 수는 변경될 수 있는 반면, 컨텐츠 제공자로부터 클라이언트(110)로 데이터를 전송하는데 사용된 전송 연결(140)의 수는 일정하게 유지될 수 있고, 또 그 반대의 경우도 가능하다.

[0172] 전술한 바와 같이, 사용된 전송 연결(140)의 수는 패킷 손실, 처리량, 제어 윈도우, 혼잡 윈도우, 수신자 윈도우, 및 값(R/M*W)에 영향을 미치는 다른 파라미터를 포함하는, 전송 연결의 속성에 의존한다(단계 530). 특히, 제어 윈도우의 크기(W)는 이들 속성에 의해 영향을 받는다. 따라서, 다중 경로 전송 관리자(130, 160)에 의해 검출된, 클라이언트-컨텐츠 제공자 연결(140) 및 컨텐츠 제공자-클라이언트 연결(140)에 대한 이들 속성의 차이는 컨텐츠 제공자-클라이언트 연결(140)에 비해 사용된 클라이언트-컨텐츠 제공자 연결(140)의 수의 차이를 나타낼 수 있다.

[0173] 제어 윈도우의 현재 크기를 사용하는 대신에, 하나의 구현예는 추후에 (예를 들어, 제어 윈도우의 크기가 각 왕복에 따라 점진적으로 계속 증가한다고 가정하여) 제어 윈도우의 추정된 크기를 사용하거나, 또는 매티스(Mathis) 식을 사용하여 관찰된 패킷 손실률로부터 제어 윈도우의 평균 크기를 도출할 수 있다.

[0174] 사용된 연결의 수를 증가시키기 위해 데이터를 전송하는 하나 이상의 파라미터를 추정하는 방법의 다른 구현예

는 새로이 제한된 데이터 전달이 시작된 것을 검출하거나 그 크기를 추정하기 위해 상이한 단계 시퀀스, 상이한 방식할 수 있다.

[0175] 따라서, 이 구현예 및 다른 구현예는, 트래픽이 시작하거나 또는 일시 중지하는 것을 검출한 후에 트래픽 파라미터를 평가하고, 사용된 동시 연결의 수를 상이한 요청 그룹에 대해 또는 다른 사용자의 동일한 요청에 대해 다른 값으로 설정하는 공통 특징을 갖는다. 또한, 요청된 전달 동안 트래픽 파라미터를 재 평가하는 구현예는 요청된 전달이 완료되기 전에 사용된 동시 연결의 수를 변경하는 공통 특징을 갖는다.

[0176] 도 6은 본 발명의 다른 양태에 따라 동시 전송 연결을 통해 단일 파일을 전송하는 트래픽의 분배를 최적화하는, 예를 들어, 크게 향상시키는데 사용되는 네트워크 환경을 도시한 도면이다.

[0177] 이 양태에서, 원거리 다중 경로 전송 관리자(160)는 일련의 송신 버퍼(620)를 포함하고, 각 송신 버퍼는 TCP 스택(310)에 의해 수립된 별개의 전송 연결(140)에 데이터를 공급한다. 이들 송신 버퍼는 사용자 공간에 구현되거나, 또는 TCP 스택에 의해 제공되는 전송 연결의 송신 버퍼와 병합될 수 있다. 다중 경로 전송 관리자는 적어도 하나의 로더(loader)(630) 및 수신 큐(640)를 더 포함하고, 수신 큐는 컨텐츠 제공자(190)로부터 고대역폭(180) 파이프를 통해 원거리 프록시 또는 VPN(170)을 통해 송신되는 데이터를 저장한다.

[0178] 설명된 구현예에서, 로더(630)는 일련의 송신 버퍼(620)를 순차적으로 채운다. 각 연결을 위한 데이터를 송신하기 전에, 로더는 제어 윈도우의 크기(W)를 얻고, 수신 큐를 통해 더 진행하기 전에 수신 큐(640)로부터 W와 동일한 데이터 양을 추출하여 현재 송신 버퍼로 전송한다. 수신 큐(640)에서 비게 된 공간은, 예를 들어, 링 버퍼 내의 포인터를 이동시킴으로써 다음 데이터를 수신할 준비가 된 것으로 표시된다. 도 6은 제1 전송 연결을 위한 송신 버퍼가 그 제어 윈도우의 크기(W1)와 동일한 양의 데이터를 이미 수신하여서; 로더(630)가 제2 전송 연결을 위한 제어 윈도우의 크기(W2)를 획득하고 나서, 다음 송신 버퍼(도시된 예에서 W3)로 진행하기 전에 대응하는 양의 데이터를 수신 큐(640)로부터 제2 송신 버퍼로 전송할 준비가 된 경우를 도시한다. 수신 큐(640)에 남아 있는 데이터의 양이 (W3에 대해 도시된 예에서와 같이) 제어 윈도우의 크기보다 더 작은 경우, 로더(630)는 이용 가능한 양을 전송한다.

[0179] 일부 구현예에서, 다수의 로더(630)는 다수의 송신 버퍼(620)에 데이터를 공급하도록 병렬로 사용될 수 있다. 또한, 라운드 로빈 방식 대신에 상이한 분배 알고리즘이 사용될 수 있다(예를 들어, 제어 윈도우의 최고 크기와 연결이 먼저 어드레스될 수 있다). 로더(630)는 제어 윈도우의 크기 이외의 파라미터를 사용하여 각 송신 버퍼에 송신되어야 하는 양을 지정할 수 있다. 예를 들어, 최근 시간 구간(예를 들어, 0.1초 내지 10초) 동안 현재 연결의 평균 처리량에 비례하는 값을 지정할 수 있다.

[0180] 본 발명에 따르면, 전술된 바와 같이 도시된 구현예 및 그 변형예는 데이터 그룹을 상이한 전송 연결에 송신하는 공통 특징을 갖고, 여기서 각 데이터 그룹의 크기는 바람직한 범위 내로 설정되는데, 예를 들어, 대응하는 연결에 대한 제어 윈도우의 크기에 가능한 한 가깝게 설정된다.

[0181] 이것은 사용된 연결의 수를 전달될 데이터의 잔량으로 자동으로 조절할 수 있게 한다. 예를 들어, 수신 큐(640)가 제어 윈도우 크기(W1 내지 W3)에 대응하는 양을 3개의 송신 버퍼(620)에 채우기에 충분한 데이터만을 포함한다면, 다른 전송 연결은 더 이상 데이터를 수신하지 않을 것이다. 이러한 방식으로, 사용된 연결(140)의 수는 전달되어야 하는 데이터의 양이 증가함에 따라 또는 랜덤 패킷 손실로 인해 하나 이상의 제어 윈도우의 크기가 감소함에 따라 상승할 수 있다.

[0182] 일부 구현예에서, 바람직한 범위는 현재 제어 윈도우의 크기와 같은 최대값만을 포함할 수 있다. 다른 구현예에서, 이 바람직한 범위는 최소값을 더 포함할 수 있다. 예를 들어, 수신 큐(640)가 현재 제어 윈도우의 크기의 50% 미만을 포함하면, 로더(630)는 단일 왕복 시간에 최대 데이터 양이 송신될 확률을 증가시키기 위해 수신 큐(640)로부터 다음 데이터를 수신하기 위해 하나 이상의 왕복 시간을 기다릴 것이다. 50% 이외의 값은 40%에서 60%의 값과 같은 최소값을 결정하는 데 사용될 수 있다.

[0183] 수신 큐(640) 내 데이터의 잔량이 모든 제어 윈도우의 합(도시된 예에서 W1 내지 W4)보다 더 높은 경우, 동시 연결의 최대 수는 미리 한정된 한도를 초과하지 않도록 설정될 수 있다.

[0184] 도시된 실시예에서, 수신 큐(640)로부터의 데이터는 각 송신 버퍼(620)에 순차적으로 로딩되는데, 즉 제1 전송 연결(140)은 제2 연결(140) 등등보다 더 일찍 데이터를 수신한다. 현재 연결(140)의 제어 윈도우의 크기를 초과하지 않는 크기를 갖는 각 데이터 그룹은 현재 연결을 통한 단일 왕복 동안 전달될 수 있다. 연속적인 데이터 그룹을 하나의 시퀀스로 송신하면 수신 측에서 수행해야 하는 재순서화(reordering) 양을 감소시킬 수 있는 추

가적인 장점이 있다.

[0185] 도시된 구현예에서, 수신 측은 순서화 큐(ordering queue)(650)를 포함하고, 이 큐는 데이터가 올바른 순서로 프록시 또는 VPN(170)에 전달될 수 있을 때까지 TCP 스택(310)을 통해 다수의 전송 연결(140)로부터 수신된 데이터를 저장한다. 각 연결(140)이 각 왕복 동안 연속적인 데이터 세트를 전달하면, (예를 들어, 제3 연결이 제2 연결보다 더 빨리 데이터를 전달하는 경우) 데이터 그룹 간에만 또는 일부 데이터 패킷이 손실되어 재-전송되어야 할 때 재순서화 동작이 필요하다. 이것이 구현되지 않으면, 예를 들어, 순차적인 데이터 패킷이 다수의 연결들 간에 확산되는 경우, 모든 연결이 현재 사이클 동안 모든 데이터를 전달할 때까지 패킷이 순서화 큐(650)에 머무를 필요가 있어서, 이에 의해 대기 시간이 증가할 수 있다.

[0186] 도 7은 각 연결을 통해 바람직한 크기의 데이터 그룹을 송신하는 것에 기초하여 동시 전송 연결을 통해 트래픽의 분배를 향상시키기 위한 방법을 나타내는 프로세스 흐름도이다.

[0187] 수신 큐가 비어 있지 않으면(단계 710), 다음 전송 연결이 데이터 그룹을 수신하도록 선택된다(단계 720). 도시된 경우에, 다음 연결은 최대 연결의 수에 도달했을 때 제1 연결로의 랩 오버(wrap over)하게 순차적으로 선택된다. 다른 구현예에서, 예를 들어, 그 송신 버퍼(620)에 남아 있는 데이터의 잔량 및 그 제어 윈도우의 크기에 기초하여 다음 연결이 선택될 수 있다(720). 구체적으로, 빈 송신 버퍼(620) 및 큰 제어 윈도우를 갖는 연결(140)이 먼저 선택될 것이다. 선택된 연결에 대한 제어 윈도우의 크기를 획득한 후에(단계 730), 제어 윈도우의 최소 크기(W) 및 수신 큐(R)의 잔량이 선택된 연결의 송신 버퍼(620)로 이동된다(단계 740). 수신 큐(640)에 남아 있는 데이터의 크기는 대응하여 다음 데이터를 위한 공간을 자유롭게 하도록 조절된다.

[0188] 그리하여 제어 윈도우(W)가 처리량에 의존하고 데이터 프레임의 크기가 전술한 바와 같이 W의 크기에 의존하는 한, 프레임의 크기는 W가 증가함에 따라 증가할 것이다. 처리량과 제어 윈도우는 또한 패킷 손실률에 의존한다. 따라서, 패킷 손실률이 각각 감소하거나 증가하는 것에 기초하여 처리량이 증가하거나 감소하는 것이 추정될 수 있다.

[0189] 도 8은 본 발명의 실시예에 따라 동시 전송 연결을 통해 다수의 파일을 병렬로 전송하는 트래픽의 분배를 향상시키는데 사용되는 네트워크 환경의 다이어그램이다.

[0190] 이 양태에서, 클라이언트(110)는 다수의 컨텐츠 제공자(190)로부터 파일 그룹을 요청한다. 이들 파일은 프록시 또는 VPN 서버(170)를 통해 병렬로 다운로드된다. 원거리 다중 경로 전송 관리자(160)에 제출되기 전에, 설명된 구현예에서 다수의 다운로드 스트림은 프레임 어셈블러(frame assembler)(860)를 통해 송신되고, 이 프레임 어셈블러는 이들 데이터 스트림을 단일 수신 큐(640)로 다중화한다. 일부 경우에, 어셈블러는 프레임의 시퀀스를 생성하는 것을 포함하고, 각 프레임은 단일 데이터 스트림에 대응하는 페이로드 및 헤더를 갖는다. 헤더는 데이터 스트림의 식별자 및 페이로드의 크기를 포함할 수 있다. 헤더는 프레임 유형(제어 또는 데이터, 체크섬 또는 추가적인 옵션)과 같은 추가적인 파라미터를 더 포함할 수 있다. 일부 실시예에서, 각 프레임은 HTTP(Hyper Text Transport Protocol) 버전 2 이후의 사양에 따라 포맷될 수 있다.

[0191] 본 발명의 일 양태에 따르면, 로더(630)는 수신 큐(640)로부터 각 연속적인 프레임을 TCP 스택(310)에 의해 유지되는 상이한 전송 연결의 송신 버퍼(620)로 전송한다. 일부 예에서, 프레임 어셈블러(860)는 각 프레임에 대해 동일한 프레임 크기를 사용한다. 로더(630)는 각 프레임이 단 하나의 연결(140)의 송신 버퍼(620)로만 전송되는 것을 보장한다. 이러한 방식으로, 사용된 전송 연결의 수는 수신 큐(640)의 데이터 양에 따라 증가하고, 요청된 파일 그룹의 전달이 거의 완료될 때 떨어진다.

[0192] 일부 구현예에서, 프레임 어셈블러(860)는 예상된 전달 크기(예를 들어, 요청된 도메인에 대한 패턴으로부터 획득된 크기) 또는 (예를 들어, 대부분의 프레임이 단일 제어 윈도우에 맞는 것을 보장하도록) 전송 연결(140)에 의해 사용되는 제어 윈도우의 크기에 관한 정보에 기초하여 프레임 크기를 미리 설정한다. 일부 실시예에서, 미리 설정된 프레임 크기는 동일한 사용자로부터의 상이한 요청 그룹마다 또는 상이한 사용자가 동일한 요청 그룹을 송신하더라도 상이한 사용자마다 변한다. 또한 프레임 크기는 데이터 전송 동안에도 변할 수 있다. 예를 들어, 프레임 어셈블러(860)는 제어 윈도우의 평균 크기가 시간에 따라 감소하면 프레임 크기를 감소시킬 수 있다.

[0193] 일부 실시예에서, 프레임 어셈블러(860)는 전달될 데이터의 잔량이 감소함에 따라 하나 이상의 데이터 프레임의 크기를 증가시킨다. 이 접근법은 각 연결(140)이 상이한 연결로 전환되기 전에 최대 양 또는 거의 최대 양의 데이터를 전송하는 것을 보장하고, 여기서, 최대 데이터 양은 적어도 일부 데이터의 수신확인 응답을 수신하기 전에 전송이 허용된 데이터의 양이다. 다른 프레임이 다른 연결을 통해 송신되면 각 데이터 프레임의 크기를 증가

시킬 때 데이터를 전송하는 데 능동적으로 사용되는 동시 연결의 수가 감소한다. 위에서 언급한 바와 같이 작은 파일의 경우 이것은 전달-보장 프로토콜을 사용할 때 실제 처리량을 증가시킨다.

[0194] 일부 실시예에서, 프레임 어셈블러(860)는 패킷 손실률이 감소함에 따라 데이터 프레임의 크기를 증가시킨다. 다른 실시예에서, 프레임 어셈블러(860)는 적어도 일부 데이터의 수신확인 응답을 수신하기 전에 송신이 허용된 데이터의 양이 증가함에 따라 데이터 프레임의 크기를 증가시킨다. 이 접근법은 각 연결이 다른 연결로 전환되기 전에 최대 양 또는 거의 최대 양의 데이터를 송신하는 것을 보장하고, 여기서 최대 양은 적어도 일부 데이터에 대한 수신확인 응답을 기다리기 전에 송신이 허용된 데이터의 양이다. 데이터 손실률의 감소, 또는 수신확인 응답을 수신하기 전에 송신이 허용된 데이터 양의 증가(제어 윈도우의 증가)는 도 5의 방법에 따라 사용되는 동시 연결의 수를 감소시킨다. 각 데이터 프레임의 크기가 증가하면 유사한 결과를 달성하는데, 각 데이터 프레임의 크기가 증가하면, 특히 작은 파일의 경우, 데이터를 전송하는데 능동적으로 사용되는 동시 연결(140)의 수가 대응하여 감소하도록 상이한 프레임은 상이한 연결을 통해 송신된다.

[0195] 일부 경우에, 로더(630)는 수신 큐(640)의 다음 프레임에 대해 다음 연결(140)을 순차적으로 선택한다. 다른 경우에, 로더는, 단일 왕복 시간에 단일 프레임을 전송하는 연결의 수를 증가시키기 위해, 제어 윈도우의 크기에 기초하여 다음 연결(140)을 선택할 수 있다. 이것은, 프레임 어셈블러(860)가, 예를 들어, 우선 순위가 높은 데이터(예를 들어, 웹 사이트 렌더링을 멈출(stall) 수 있는 자바스크립트 파일)에 대해 더 작은 프레임을 사용하여 다른 크기의 프레임을 형성해서 전달 속도를 높이는 경우에 유용하다.

[0196] 전술한 구현예 및 그 변형예에 따르면, 각 데이터 프레임이 단일 데이터 스트림으로부터의 데이터만을 포함하는 데이터 프레임들을 형성하고 단 하나의 전송 연결(140)을 통해 각 데이터 프레임을 송신함으로써 다수의 전송 연결(140)을 통해 다수의 데이터 스트림의 데이터 전송을 순서화하는 공통 특징을 갖는다. 동일한 데이터 스트림으로부터의 데이터 프레임은 다수의 동시 전송 연결(140)을 통해 송신될 수 있으나, 각 데이터 프레임은 단일 전송 연결을 통해서만 송신될 수 있다.

[0197] 사용된 연결의 수를 자동으로 조절하는 것에 더하여, 이러한 구현예는 또한 수신자 측에서 데이터를 재순서화할 필요성도 감소시킨다. 도시된 실시예에서, 수신자 측은 순서화 큐(650)를 포함한다. 이 큐는 프레임의 시퀀스를 프레임 디스어셈블러(frame disassembler)(870)에 송신하며, 이 디스어셈블러는 이 프레임 시퀀스를 프록시 또는 VPN(170)을 통해 다수의 컨텐츠 서버(190)로 송신되는 개별 데이터 스트림으로 역다중화한다. 순서화 큐(650)는 각 프레임이 모든 각자의 데이터를 포함하는 것을 보장하기만 하면 된다. 프레임의 모든 패킷이 단일 전송 연결에 의해 전달되는 즉시, 이 프레임은 프레임 디스어셈블러(870)에 바로 송신될 수 있다. 이 경우에, 순서화 큐(650)는 개별 데이터 프레임을 전달하기 전에 2개 이상의 전송 연결(140)을 기다릴 필요가 없다.

[0198] 도 9는 본 발명의 다른 실시예에 따라 동시 전송 연결을 통해 다수의 파일을 병렬로 전송하는 트래픽의 분배를 최적화하는, 예를 들어, 크게 향상시키는데 사용되는 네트워크 환경을 도시한 도면이다.

[0199] 이 실시예에서, 다중 경로 전송 관리자(160)는 다수의 수신 큐(640)를 포함하고, 각 수신 큐는 프록시 또는 VPN(170)을 통해 대응하는 컨텐츠 제공자(190)로부터 로딩된 단일 데이터 스트림을 위한 데이터를 각각 저장하는데 사용된다. 이 실시예에서, 로더(630)는 또한 프레임 어셈블러의 기능을 수행한다. 구체적으로는, 로더(630)는 다음 수신 큐(640)로부터 취득되어야 하는 데이터의 양을 선택하고, 이를 프레임으로 포맷하고, 프레임을 다음 송신 버퍼(620)에 로딩하며, 이 프레임은 TCP 스택(310)에 의해 대응하는 전송 연결(140)을 통해 데이터를 전송하는데 사용된다. 이후 로더(630)는 다른 수신 큐(640) 및 전송 연결(140)에 대해 이 기능을 수행한다.

[0200] 설명된 실시예에서, 프레임 어셈블러 및 로더(630)는 각 전송 연결(140)에 대한 제어 윈도우의 크기(W)를 획득하고, 이 크기를 사용하여 프레임 크기를 최적화하는데, 예를 들어, 향상된 결과를 제공하는 프레임 크기를 선택한다. 예를 들어, 각 프레임 크기는 대응하는 제어 윈도우의 크기에 가능한 가깝게 설정되어, 프레임이 단일 왕복 시간에 전달될 확률을 최대화하거나 적어도 증가시킨다. 도시된 실시예에서는, 크기(W1)의 프레임 1은 이미 제1 수신 큐(640)로부터 제1 송신 버퍼(620)로 로딩되어 있다. 제1 수신 큐(640)의 이용 가능한 공간은 예를 들어 링 버퍼의 포인터를 변경하는 것에 의해 조절된다. 이후, 프레임 어셈블러 및 로더(630)는 제2 수신 큐(640)로부터 크기(W2)의 프레임을 어셈블링하고, 제3 수신 큐(640)로 진행할 준비가 된다. 전송 연결(140)의 수가 수신 큐(640)의 수보다 더 크면, 큐(640)는 처음부터 회전하기 시작할 수 있다. 도시된 예에서, 제4 송신 버퍼(620)는 제1 수신 큐(640) 등으로부터 잔여 데이터를 수신할 것이다.

[0201] 다른 구현예에서, 로더(630)는 송신 버퍼(620)에 대응하는 전송 연결(140)의 이전에 측정된 처리량에 기초하여

송신 버퍼(620)에 로딩된 데이터에 대해 프레임 크기를 설정한다. 다른 예에서, 로더(630)는 연결(140)의 유형(예를 들어, 유선 대 무선)에 기초하여 프레임 크기를 설정한다.

[0202] 또 다른 구현예에서, 다음 전송 연결(140)은 라운드 로빈 알고리즘과는 다른 알고리즘을 사용하여 다음 프레임을 수신하기 위해 로더(630)에 의해 선택될 수 있다. 예를 들어, 다음 연결(140)로서, 빈 송신 버퍼(620)를 갖는 연결(140)의 최근 시간 구간(예를 들어, 0.1초 내지 10초)에 걸쳐 제어 윈도우의 최대 크기 또는 최대 처리량을 갖는 연결(140)을 선택할 수 있다. 다음 수신 큐(640)는 또한 다른 알고리즘을 사용함으로써 선택될 수 있다. 예를 들어, 상이한 수신 큐(640)에는 상이한 우선 순위가 할당될 수 있고, 더 높은 우선 순위 데이터에는 더 높은 우선 순위 큐(640)가 할당되도록 큐에 데이터가 할당될 수 있다. 예를 들어, 웹 사이트의 로딩을 멈출 수 있는 자바스크립트 파일은 더 높은 우선 순위 큐(640)에 할당될 수 있다. 다른 예에서, 큰 배경 다운로드가 있는 작은 파일의 다운로드 속도를 높이기 위해 가장 적은 잔량 데이터를 갖는 큐(640)에는 더 높은 우선 순위가 할당될 수 있다.

[0203] 전술한 구현예 및 그 변형예는 각각이 단일 데이터 스트림으로부터의 데이터만을 포함하는 데이터 프레임들을 형성함으로써 다수의 전송 연결(140)을 통해 다수의 데이터 스트림의 데이터 전송을 재순서화하는 공통 특징을 갖는다. 각 데이터 프레임은 단 하나의 전송 연결(140)을 통해 송신되고, 그 크기는 이 연결이 다른 가능한 크기의 범위에 비해 (예를 들어, 단일 왕복 동안 각 프레임이 송신될 확률을 향상시킴으로써) 처리량을 증가시킬 수 있는 것으로 선택된다.

[0204] 상이한 연결을 통해 상이한 크기의 데이터 프레임을 송신하면 또한 수신자 측에서 재순서화 동작이 간단해지는 장점을 더 제공할 수 있다. 설명된 구현예에서, 각 전송 연결(140)로부터의 데이터는 별도의 재순서화 큐(650)에 저장된다. 도시된 예에서, 단 3개의 전송 연결만이 업스트림 데이터를 운반하고, 다른 연결은 사용되지 않은 상태로 남는다. 각 데이터 프레임은 단일 전송 연결(140)을 통해서만 송신될 수 있으나, 각 전송 연결(140)은 하나를 초과하는 데이터 스트림으로부터 오는 데이터 프레임을 운반할 수 있다.

[0205] 도시된 실시예는, 동일한 전송 연결(140)을 통해 송신된 데이터 프레임들이 동일하거나 유사한 크기를 가진다는 사실로부터, 연결 특정 재순서화 버퍼에 요구될 수 있는 메모리의 양을 보다 쉽게 예측할 수 있어서, 메모리 할당 효율을 증가시킬 수 있다는 장점을 얻을 수 있다. 이는 또한 다수의 연결로부터 프레임을 병렬로 재순서화하는 데 자원을 덜 소비하게 한다. 임의의 순서화 큐에서 임의의 프레임으로부터 모든 패킷이 전달되는 즉시, 이들 패킷은 프레임 디스어셈블러(870)로 송신되고, 이 디스어셈블러는 이들 패킷을 개별 데이터 스트림으로 역다중화하고 각 데이터 스트림을 프록시 또는 VPN(170)을 통해 대응하는 컨텐츠 제공자(190)로 포워딩한다.

[0206] 도시된 실시예는 다수의 동시 연결을 통해 데이터 전달을 보장하는 수단으로서 전송 제어 프로토콜(TCP) 및 TCP 스택(310)을 참조한다. 다른 실시예는 상이한 전달 프로토콜, 예를 들어, 추가적인 사용자 공간 흐름 제어 및 혼잡 제어 모듈을 갖는 사용자 데이터그램 프로토콜(User Datagram Protocol: UDP), 또는 스트림 제어 전송 프로토콜(Stream Control Transmission Protocol: SCTP)을 사용할 수 있다.

[0207] 다른 실시예에서, 다중 경로 트래픽 관리자(130, 160)는 클라이언트(110) 및 컨텐츠 제공자의 서버(190) 상에서 각각 실행되는 클라이언트 애플리케이션과 통합되는 것에 의해, 별도의 프록시 또는 VPN 모듈(120, 170)의 필요성을 제거할 수 있다. 예를 들어, 본 명세서에 개시된 실시예들은, 이것이 클라이언트(110) 및 컨텐츠 제공자(190)에 의해 지원된다면, 다중 경로 TCP 프로토콜 확장의 처리량을 가속화하는데 사용될 수 있다.

[0208] 일부 실시예에서, 상이한 데이터 프레임은 동일한 물리적 데이터 링크 또는 동일한 라우트 상의 모든 데이터 링크를 공유하는 상이한 전송 연결(140)을 통해 송신된다. 다른 실시예에서, 상이한 전송 연결은 유선 및 무선과 같은 상이한 데이터 링크를 사용하거나 또는 다수의 IP 어드레스를 갖는 동일한 서버로 향하는 상이한 라우트를 사용할 수 있다.

[0209] 다수의 동시 연결은 클라이언트(110)로부터 프록시 또는 VPN 서버(120)로 모든 트래픽 또는 이 트래픽의 일부만을 전송하는데 사용될 수 있다. 일부 실시예에서, 랜덤 데이터 손실의 영향을 완화하기 위해 장거리 컨텐츠만이 다수의 동시 연결(140)을 통해 액세스되는 반면, 다수의 연결(140)에 걸쳐 동일한 파일의 로딩을 확산시키지 않고 사용자에 가까운 컨텐츠가 직접 액세스된다.

[0210] 도 10은 본 명세서에 개시된 임의의 컴퓨터 및 서버를 구현할 수 있는 예시적인 컴퓨팅 디바이스(1000)를 나타내는 블록도이다. 컴퓨팅 디바이스(1000)는 본 명세서에서 논의된 것과 같은 다양한 절차를 수행하는데 사용될 수 있다. 컴퓨팅 디바이스(1000)는 서버, 클라이언트 또는 임의의 다른 컴퓨팅 엔티티로서 기능할 수 있다. 컴퓨팅 디바이스는 본 명세서에서 논의된 바와 같은 다양한 모니터링 기능을 수행할 수 있고, 본 명세서에 설명된

애플리케이션 프로그램과 같은 하나 이상의 애플리케이션 프로그램을 실행할 수 있다. 컴퓨팅 디바이스(1000)는 데스크탑 컴퓨터, 노트북 컴퓨터, 서버 컴퓨터, 핸드헬드 컴퓨터, 태블릿 컴퓨터 등과 같은 다양한 컴퓨팅 디바이스 중 임의의 것일 수 있다.

[0211] 컴퓨팅 디바이스(1000)는 하나 이상의 프로세서(들)(1002), 하나 이상의 메모리 디바이스(들)(1004), 하나 이상의 인터페이스(들)(1006), 하나 이상의 대용량 저장 디바이스(들)(1008), 하나 이상의 입력/출력(I/O) 디바이스(들)(1010), 및 디스플레이 디바이스(1030)를 포함하며, 이들 구성 요소는 모두 버스(1012)에 연결된다. 프로세서(들)(1002)는 메모리 디바이스(들)(1004) 및/또는 대용량 저장 디바이스(들)(1008)에 저장된 명령을 실행하는 하나 이상의 프로세서 또는 제어기를 포함한다. 프로세서(들)(1002)는 또한 캐시 메모리와 같은 다양한 유형의 컴퓨터 판독 가능 매체를 포함할 수 있다.

[0212] 메모리 디바이스(들)(1004)는 휘발성 메모리(예를 들어, 랜덤 액세스 메모리(RAM)(1014)) 및/또는 비 휘발성 메모리(예를 들어, 판독 전용 메모리(ROM)(1016))와 같은 다양한 컴퓨터 판독 가능 매체를 포함한다. 메모리 디바이스(들)(1004)는 또한 플래시 메모리와 같은 재기록 가능 ROM을 포함할 수 있다.

[0213] 대용량 저장 디바이스(들)(1008)는 자기 테이프, 자기 디스크, 광학 디스크, 솔리드-스테이트 메모리(예를 들어, 플래시 메모리) 등과 같은 다양한 컴퓨터 판독 가능 매체를 포함한다. 도 10에 도시된 바와 같이, 특정 대용량 저장 디바이스는 하드 디스크 드라이브(1024)이다. 다양한 컴퓨터 판독 가능 매체로부터 판독 및/또는 이 매체에 기록할 수 있기 위해 대용량 저장 디바이스(들)(1008)에는 다양한 드라이브가 또한 포함될 수 있다. 대용량 저장 디바이스(들)(1008)는 이동식 매체(1026) 및/또는 비 이동식 매체를 포함한다.

[0214] I/O 디바이스(들)(1010)는 데이터 및/또는 다른 정보를 컴퓨팅 디바이스(1000)에 입력하거나 이 컴퓨터 디바이스로부터 검색할 수 있는 다양한 디바이스를 포함한다. 예시적인 I/O 디바이스(들)(1010)는 커서 제어 디바이스, 키보드, 키패드, 마이크로폰, 모니터 또는 다른 디스플레이 디바이스, 스피커, 프린터, 네트워크 인터페이스 카드, 모뎀, 렌즈, CCD 또는 다른 이미지 캡처 디바이스 등을 포함한다.

[0215] 디스플레이 디바이스(1030)는 컴퓨팅 디바이스(1000)의 하나 이상의 사용자에게 정보를 디스플레이할 수 있는 임의의 유형의 디바이스를 포함한다. 디스플레이 디바이스(1030)의 예들은 모니터, 디스플레이 단말, 비디오 프로젝션 디바이스 등을 포함한다.

[0216] 인터페이스(들)(1006)는 컴퓨팅 디바이스(1000)가 다른 시스템, 디바이스 또는 컴퓨팅 환경과 상호 작용할 수 있게 하는 다양한 인터페이스를 포함한다. 예시적인 인터페이스(들)(1006)는 근거리 통신망(LAN), 광역 네트워크(WAN), 무선 네트워크 및 인터넷에 대한 인터페이스와 같은 임의의 수의 상이한 네트워크 인터페이스(1020)를 포함한다. 다른 인터페이스(들)는 사용자 인터페이스(1018) 및 주변 디바이스 인터페이스(1022)를 포함한다. 인터페이스(들)(1006)는 또한 하나 이상의 사용자 인터페이스 요소(1018)를 포함할 수 있다. 인터페이스(들)(1006)는 프린터, 포인팅 디바이스(마우스, 트랙 패드 등), 키보드 등을 위한 인터페이스와 같은 하나 이상의 주변 인터페이스를 더 포함할 수 있다.

[0217] 버스(1012)는 프로세서(들)(1002), 메모리 디바이스(들)(1004), 인터페이스(들)(1006), 대용량 저장 디바이스(들)(1008) 및 I/O 디바이스(들)(1010)가 서로 통신하게 할 뿐만 아니라 버스(1012)에 연결된 다른 디바이스 또는 구성 요소와 통신하게 한다. 버스(1012)는 시스템 버스, PCI 버스, IEEE 1394 버스, USB 버스 등과 같은 여러 유형의 버스 구조 중 하나 이상을 나타낸다.

[0218] 예시를 위한 목적으로, 프로그램 및 다른 실행 가능한 프로그램 구성 요소들은 본 명세서에서 이산 블록인 것으로 도시되어 있으나, 이러한 프로그램 및 구성 요소는 컴퓨팅 디바이스(1000)의 상이한 저장 구성 요소에 다양한 시간에 상주할 수 있고 프로세서(들)(1002)에 의해 실행될 수 있는 것으로 이해된다. 대안적으로, 본 명세서에 설명된 시스템 및 절차는 하드웨어로 구현되거나, 또는 하드웨어, 소프트웨어 및/또는 펌웨어의 조합으로 구현될 수 있다. 예를 들어, 하나 이상의 주문형 집적 회로(application specific integration circuit: ASIC)가 본 명세서에 설명된 시스템 및 절차 중 하나 이상을 수행하도록 프로그래밍될 수 있다.

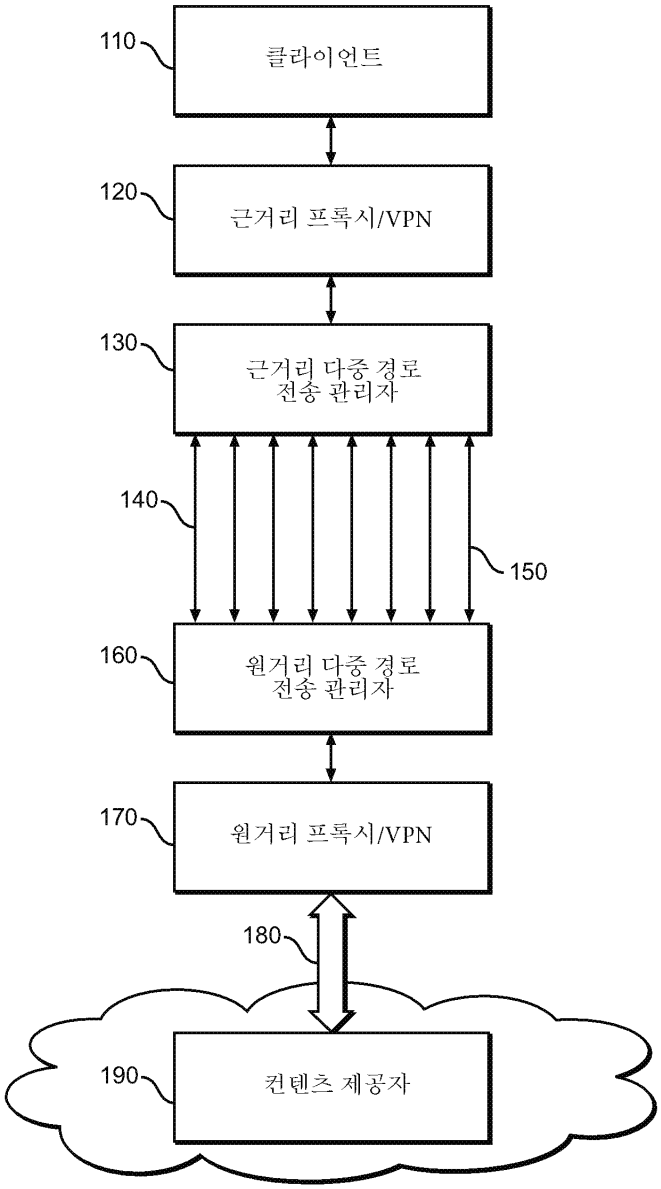
[0219] 본 명세서 전체에 걸쳐 "일 실시예", "실시예", "일례" 또는 "예"라는 언급은 실시예 또는 예와 관련하여 설명된 특정 특징, 구조 또는 특성이 본 발명의 적어도 하나의 실시예에 포함되는 것을 의미한다. 따라서, 본 명세서 전체에 걸쳐 다양한 곳에서 "일 실시예에서", "실시예에서", "일례" 또는 "예"라는 문구의 출현은 모두 반드시 동일한 실시예 또는 예를 언급하는 것은 아니다. 또한, 특정 특징, 구조 또는 특성은 하나 이상의 실시예 또는 예에서 임의의 적절한 조합 및/또는 서브 조합으로 결합될 수 있다. 또한, 본 명세서에 제공된 도면은 이 기술 분야에 통상의 지식을 가진 자에게 설명하기 위한 목적으로 제시된 것일 뿐, 도면이 반드시 축척에 맞게 그

려진 것은 아니라는 것을 이해해야 한다.

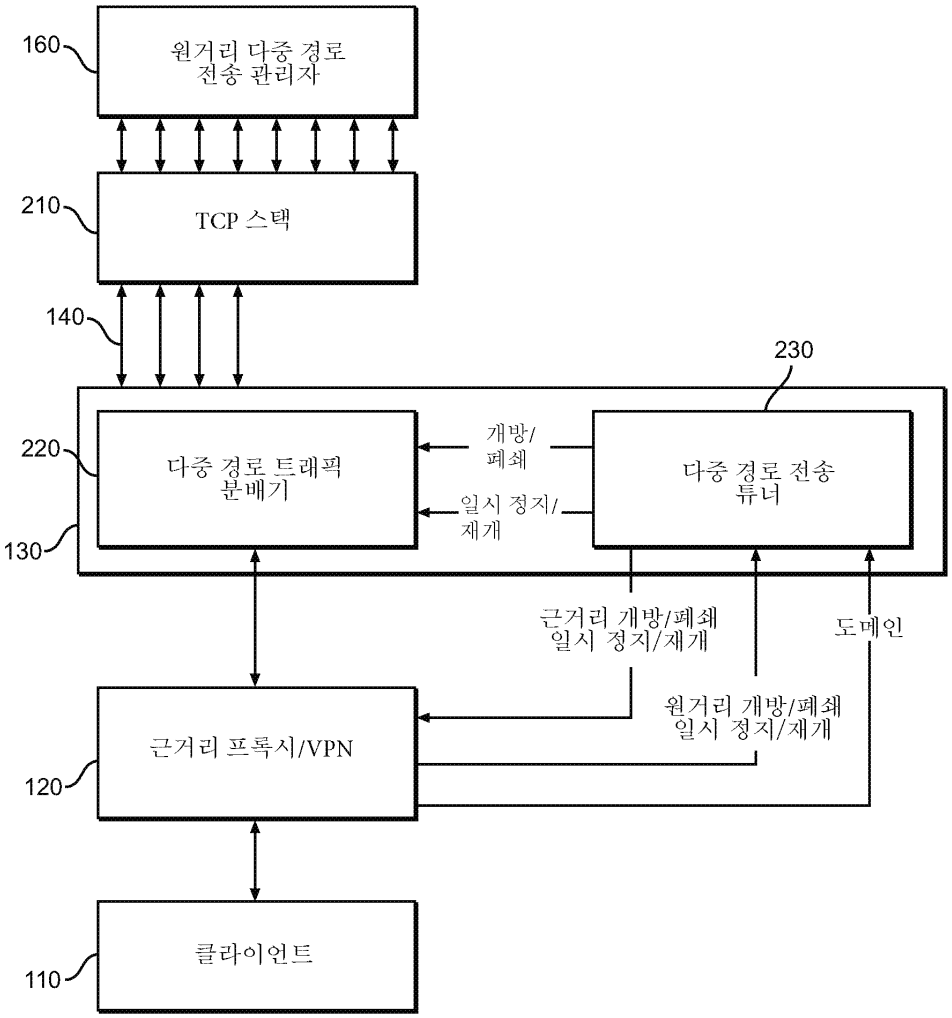
[0220] 본 발명은 그 사상 또는 본질적인 특징을 벗어나지 않고 다른 특정 형태로 구현될 수 있다. 설명된 실시예는 모든 점에서 단지 예시적인 것으로 고려되어야 하고, 본 발명을 제한하는 것으로 고려되어서는 안 된다. 그리하여, 본 발명의 범위는 전술한 설명에 의해서가 아니라 첨부된 청구 범위에 의해 한정된다. 청구 범위와 균등범위 내에 있는 모든 변경은 본 발명의 범위 내에 포함되는 것으로 이해되어야 한다.

도면

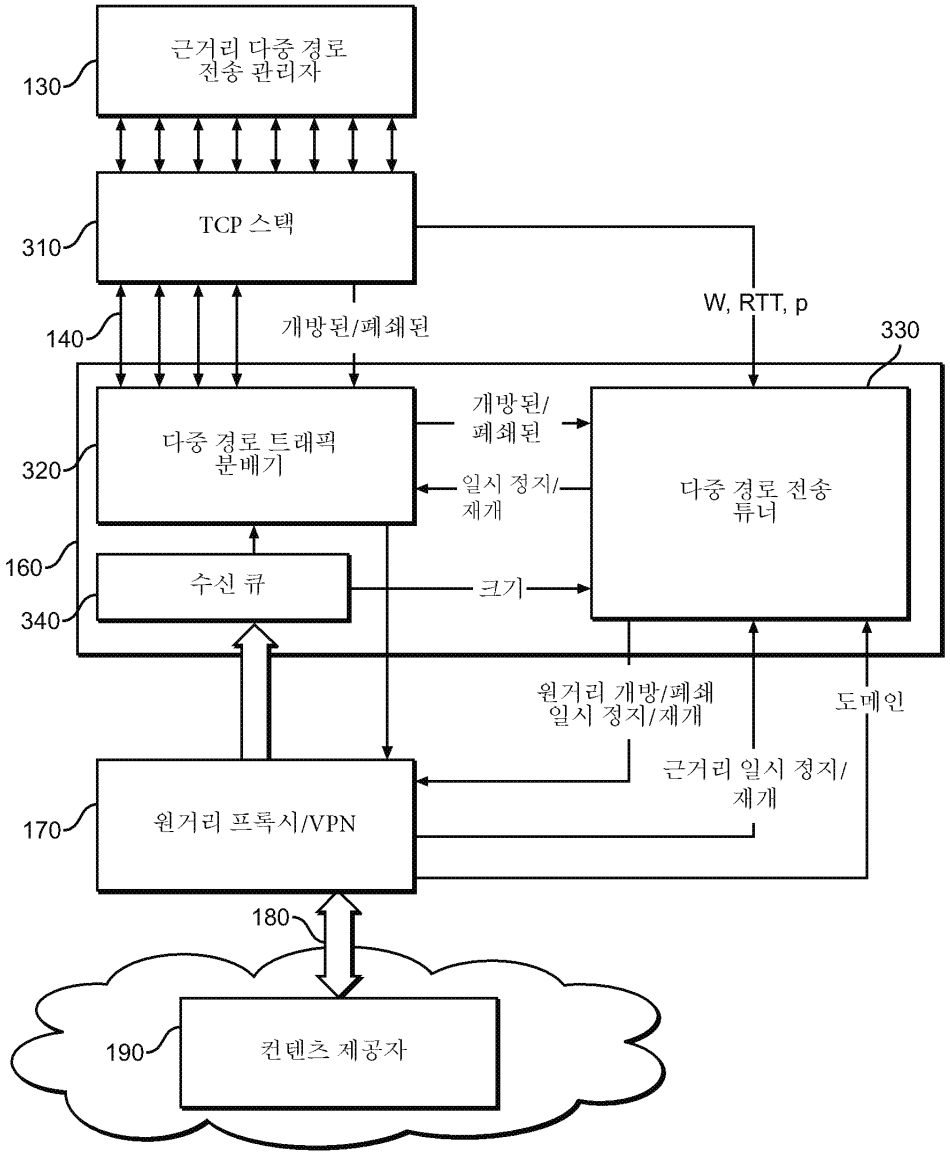
도면1



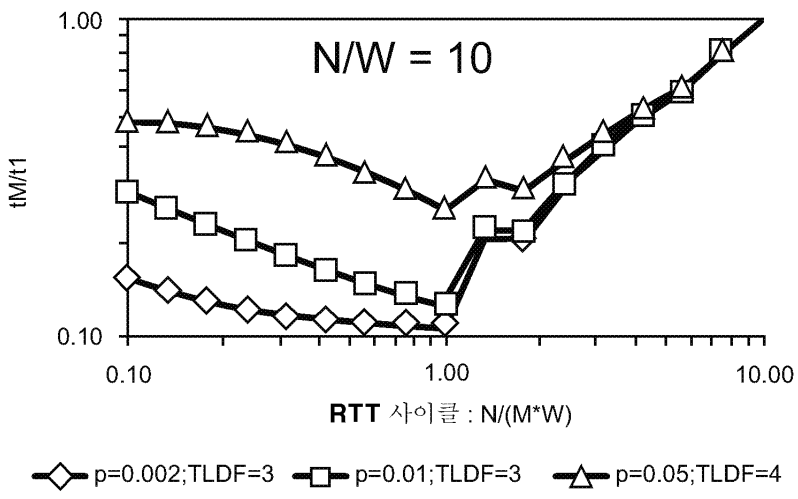
도면2


160
원거리 다중 경로
전송 관리자
210
TCP 스택
140
230
220
다중 경로 트래픽
분배기
개방/
폐쇄
일시 정지/
재개
다중 경로 전송
튜너
130
근거리 개방/폐쇄
일시 정지/재개
도메인
원거리 개방/폐쇄
일시 정지/재개
근거리 프록시/VPN
120
클라이언트
110

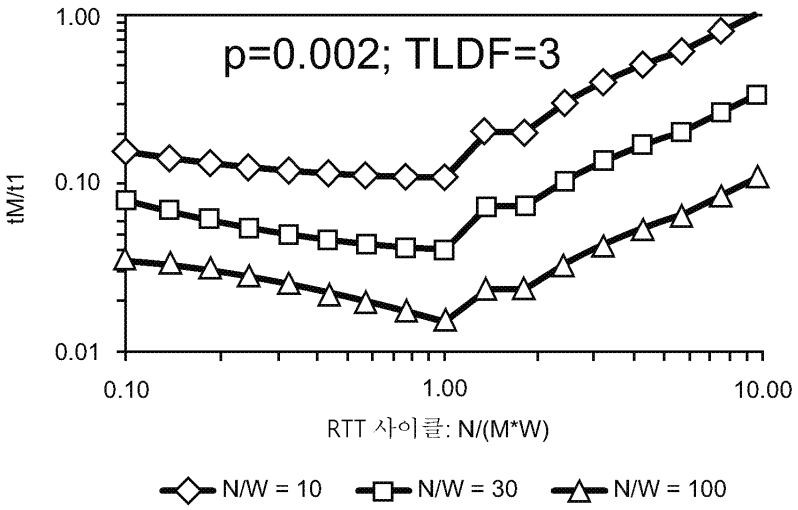
도면3


근거리 다중 경로
전송 관리자
130
TCP 스택
310
140
개방된/폐쇄된
W, RTT, p
330
다중 경로 트래픽
분배기
320
개방된/
폐쇄된
일시 정지/
재개
다중 경로 전송
튜너
160
수신 큐
340
크기
원거리 개방/폐쇄
일시 정지/재개
도메인
근거리 일시 정지/
재개
원거리 프록시/VPN
170
180
190
컨텐츠 제공자

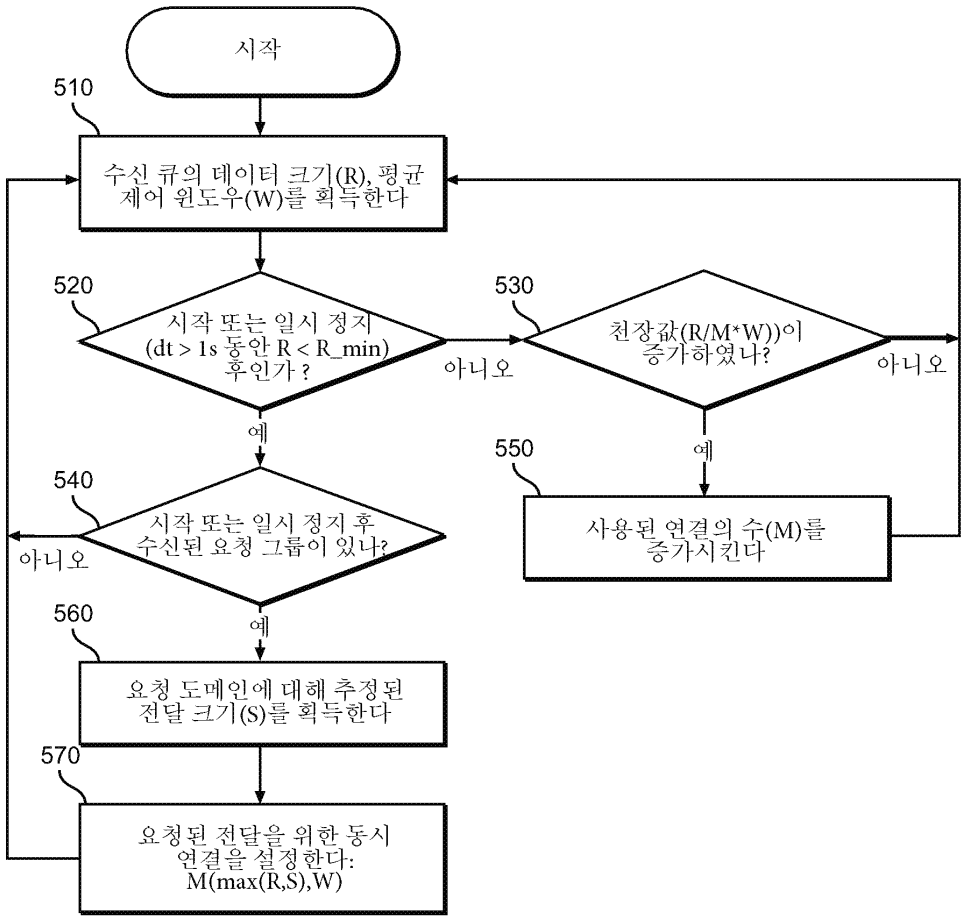
도면4a


N/W = 10
1.00
0.10
tM/t1
0.10
1.00
10.00
RTT 사이클 : N/(M*W)
p=0.002;TLDF=3
p=0.01;TLDF=3
p=0.05;TLDF=4

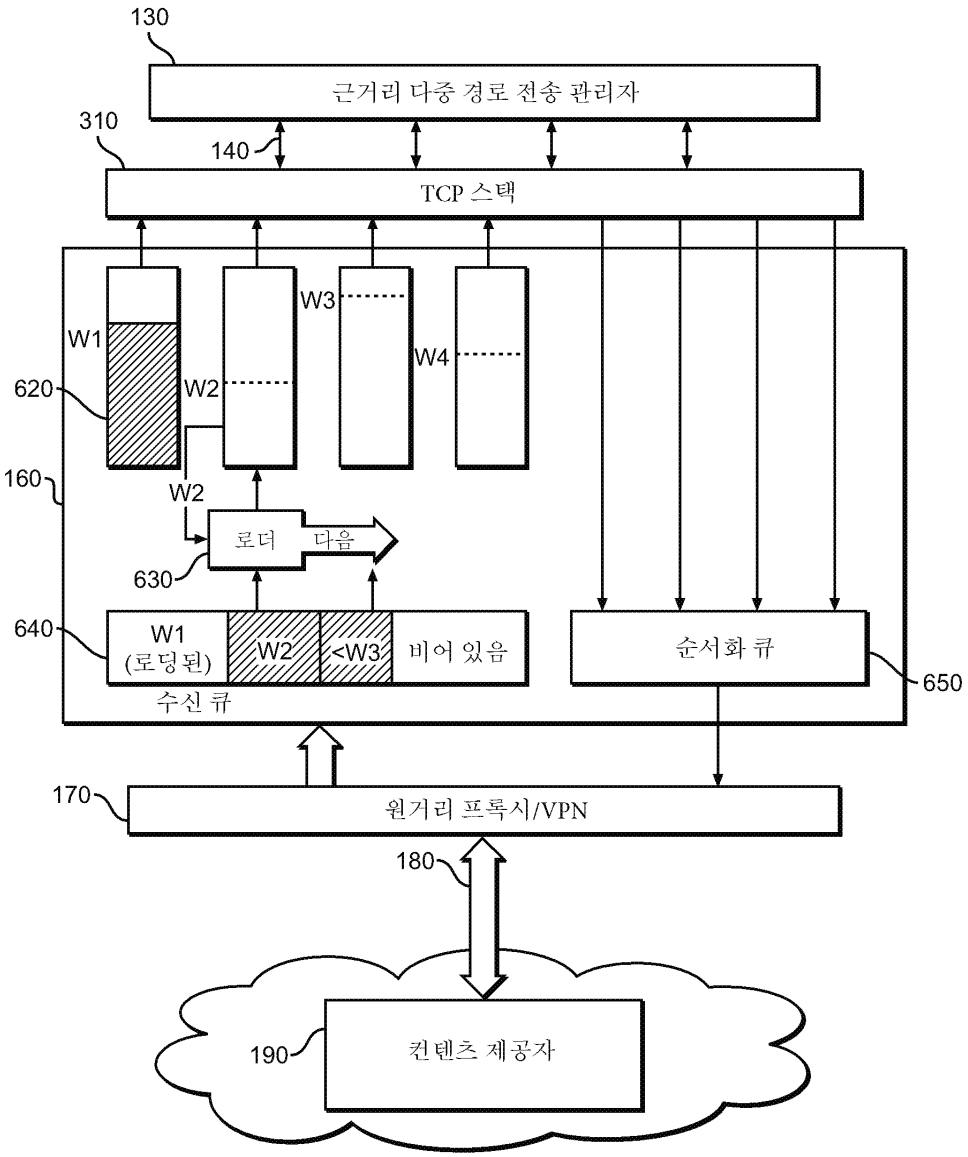
도면4b


p=0.002; TLDF=3
1.00
0.10
0.01
tM/t1
0.10
1.00
10.00
RTT 사이클: N/(M*W)
N/W = 10
N/W = 30
N/W = 100

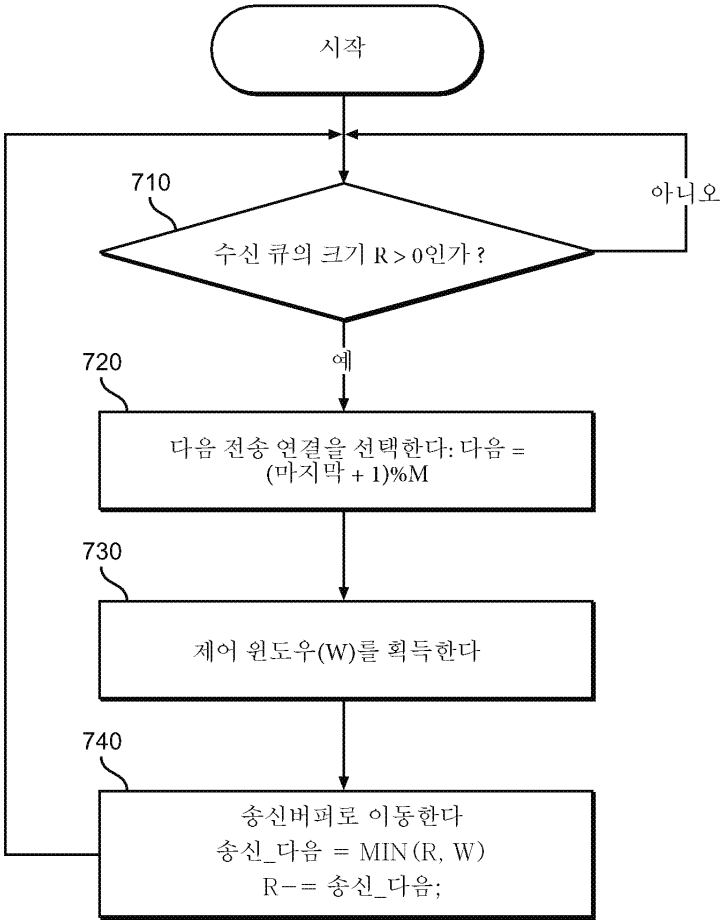
도면5


시작
510
수신 큐의 데이터 크기(R), 평균 제어 윈도우(W)를 획득한다
520
시작 또는 일시 정지 (dt > 1s 동안 R < R_min) 후인가 ?
아니오
530
천장값(R/M*W))이 증가하였나?
아니오
예
550
사용된 연결의 수(M)를 증가시킨다
540
시작 또는 일시 정지 후 수신된 요청 그룹이 있나?
아니오
예
560
요청 도메인에 대해 추정된 전달 크기(S)를 획득한다
570
요청된 전달을 위한 동시 연결을 설정한다: M(max(R,S),W)

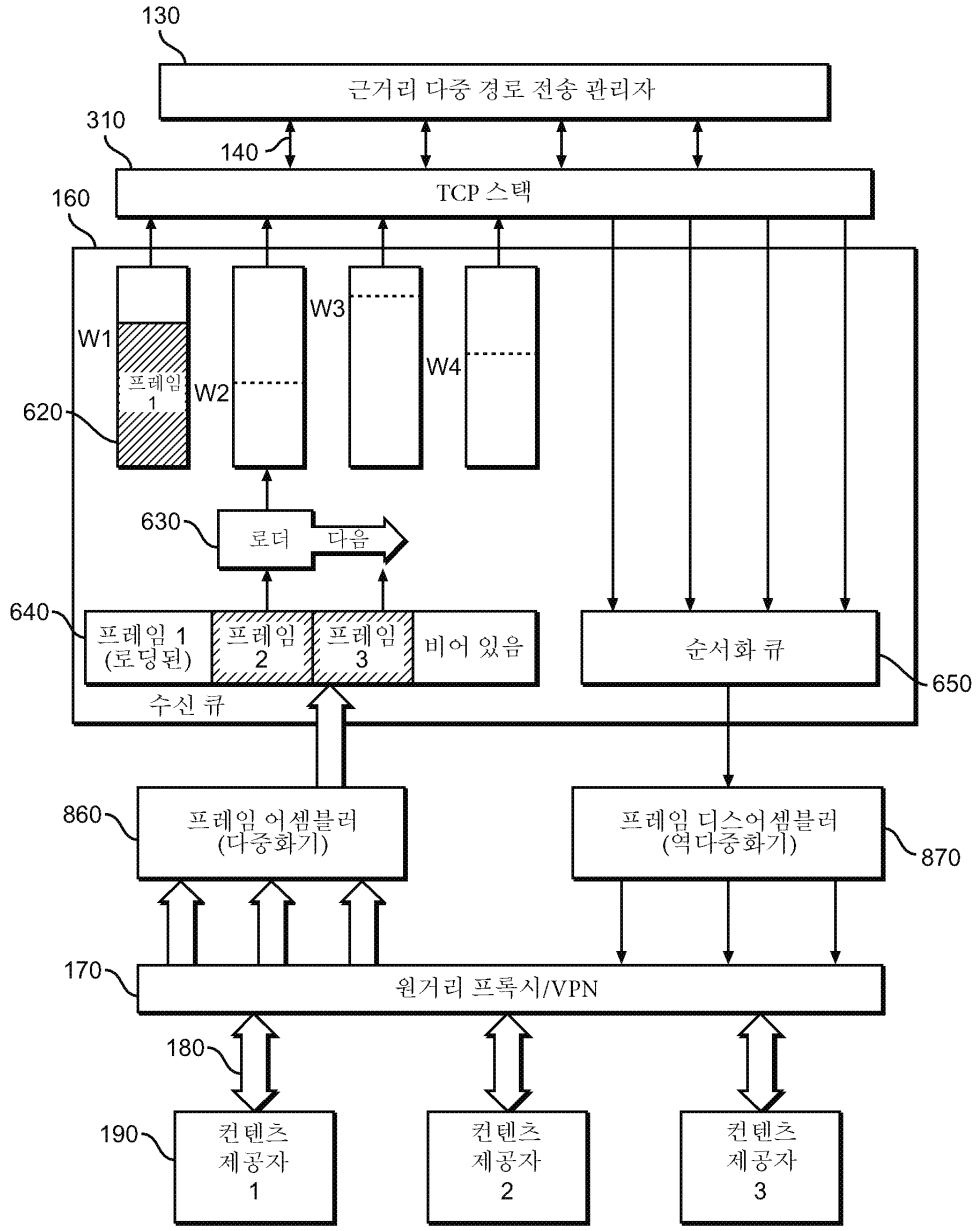
도면6


130
근거리 다중 경로 전송 관리자
310
140
TCP 스택
W1
W2
W3
W4
620
160
W2
로더
다음
630
640
W1
(로딩된)
W2
<W3
비어 있음
수신 큐
순서화 큐
650
170
원거리 프록시/VPN
180
190
컨텐츠 제공자

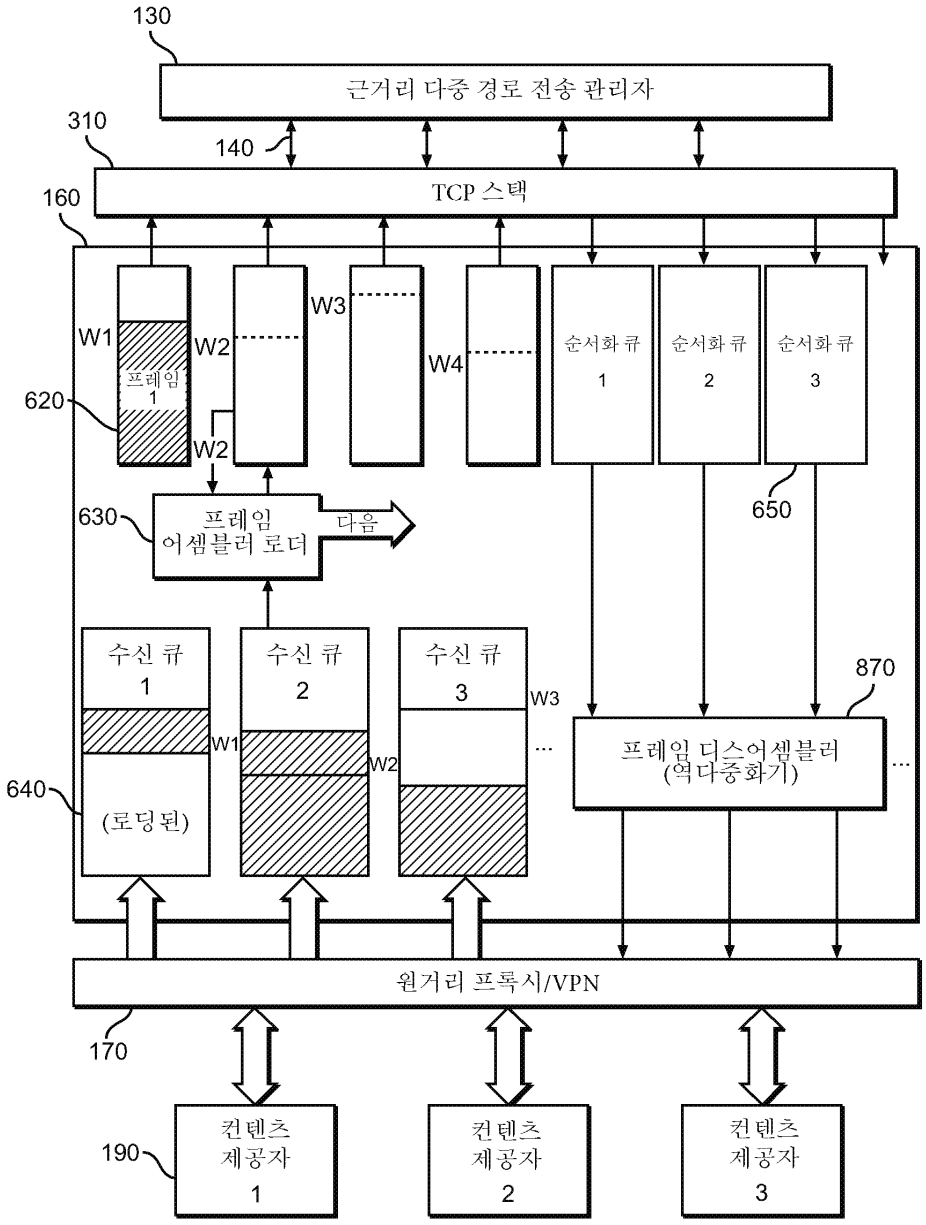
도면7


시작
710
수신 큐의 크기 R > 0인가 ?
아니오
예
720
다음 전송 연결을 선택한다: 다음 =
(마지막 + 1)%M
730
제어 윈도우(W)를 획득한다
740
송신버퍼로 이동한다
송신_다음 = MIN(R, W)
R-= 송신_다음;

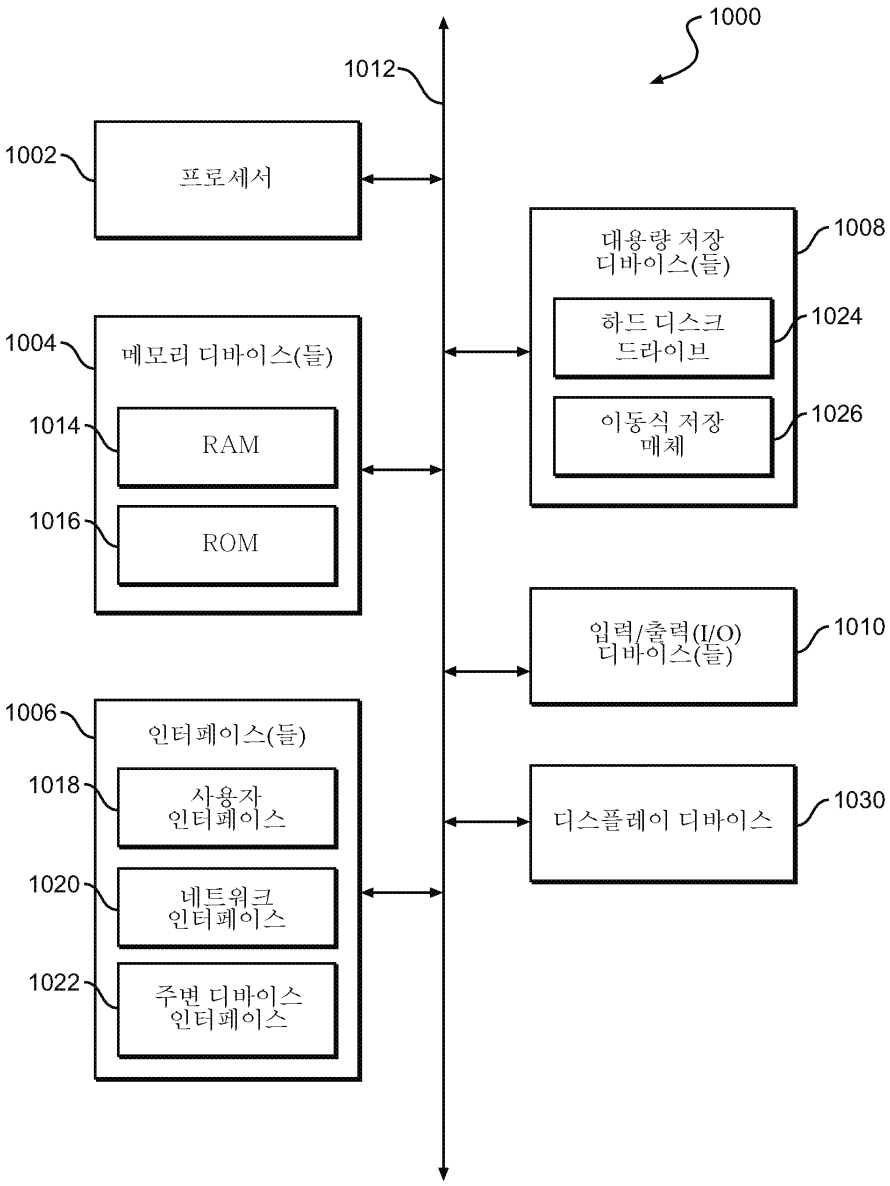
도면8


130
근거리 다중 경로 전송 관리자
310
140
TCP 스택
160
W1
프레임 1
W2
W3
W4
620
630
로더
다음
640
프레임 1 (로딩된)
프레임 2
프레임 3
비어 있음
순서화 큐
650
수신 큐
860
프레임 어셈블러 (다중화기)
프레임 디스어셈블러 (역다중화기)
870
170
원거리 프록시/VPN
180
190
컨텐츠 제공자 1
컨텐츠 제공자 2
컨텐츠 제공자 3

도면9


130
근거리 다중 경로 전송 관리자
310
140
TCP 스택
160
W1
W2
W3
W4
프레임 1
620
W2
630
프레임 어셈블리 로더
다음
순서화 큐 1
순서화 큐 2
순서화 큐 3
650
수신 큐 1
수신 큐 2
수신 큐 3
W1
W2
W3
640
(로딩된)
870
프레임 디스어셈블러 (역다중화기)
원거리 프록시/VPN
170
190
컨텐츠 제공자 1
컨텐츠 제공자 2
컨텐츠 제공자 3

도면10


1000
1012
1002
프로세서
1004
메모리 디바이스(들)
1014
RAM
1016
ROM
1006
인터페이스(들)
1018
사용자 인터페이스
1020
네트워크 인터페이스
1022
주변 디바이스 인터페이스
대용량 저장 디바이스(들)
1008
하드 디스크 드라이브
1024
이동식 저장 매체
1026
입력/출력(I/O) 디바이스(들)
1010
디스플레이 디바이스
1030