



등록특허 10-2442083



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년09월13일
(11) 등록번호 10-2442083
(24) 등록일자 2022년09월05일

- (51) 국제특허분류(Int. Cl.)
H04L 9/40 (2022.01)
- (52) CPC특허분류
H04L 69/161 (2022.05)
- (21) 출원번호 10-2017-7028722
- (22) 출원일자(국제) 2016년04월08일
심사청구일자 2021년04월08일
- (85) 번역문제출일자 2017년10월11일
- (65) 공개번호 10-2017-0137088
- (43) 공개일자 2017년12월12일
- (86) 국제출원번호 PCT/EP2016/057795
- (87) 국제공개번호 WO 2016/162501
국제공개일자 2016년10월13일
- (30) 우선권주장
15163169.4 2015년04월10일
유럽특허청(EPO)(EP)
- (56) 선행기술조사문헌
KR1020130018858 A*
US20040190449 A1*
US20110170545 A1*

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 11 항

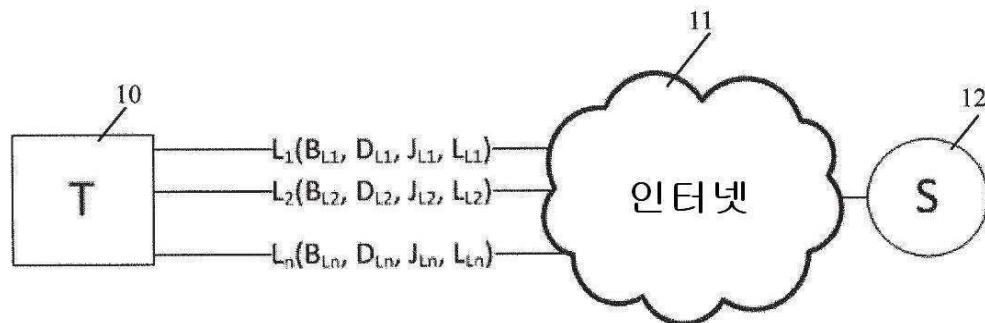
심사관 : 박보미

(54) 발명의 명칭 TCP 터널들 및 고유 TCP 정보를 기초로 하는 번들링 시나리오에서 패킷들의 스케줄링을 위한 방법 및 시스템

(57) 요 약

본 발명은 전송 제어 프로토콜(TCP) 세션들 또는 사용자 데이터그램 프로토콜(UDP) 세션들에 대해 패킷 기반 스케줄링을 사용하는 리소스 번들링을 위한 방법 및 시스템을 제공하며, 상기 시스템은 데이터를 송신 및 수신하도록 구성된 사용자 디바이스 및 데이터를 송신 및 수신하도록 구성된 서버를 포함한다. 상기 시스템은 TCP 터널들 및 적어도 하나의 전송 네트워크를 통해 데이터 패킷들을 스케줄링하고 분배하도록 구성된 제1 본딩 모듈 및 제2 본딩 모듈을 더 포함한다.

대 표 도 - 도1



(종래 기술)

명세서

청구범위

청구항 1

전송 제어 프로토콜(Transmission Control Protocol; TCP) 세션들 또는 사용자 데이터그램 프로토콜(User Datagram Protocol; UDP) 세션들에 대한 패킷 기반 스케줄링 시스템으로서,

제1 스케줄링 모듈 및 적어도 하나의 전송 네트워크에 접속된 적어도 2개의 액세스 인터페이스들을 포함하는 제1 본딩 모듈 - 상기 제1 본딩 모듈은 사용자 디바이스에 접속될 수 있고, TCP 터널은 제2 본딩 모듈에서 종단(terminate)하는 상기 액세스 인터페이스들 각각을 통해 구성되며, 상기 제1 스케줄링 모듈은 상기 TCP 터널들을 거쳐 상기 제2 본딩 모듈을 향하여 데이터 패킷들을 스케줄링 및 분배하도록 구성됨 - ; 또는 제2 스케줄링 모듈 및 상기 적어도 하나의 전송 네트워크 각각에 접속된 적어도 하나의 액세스 인터페이스를 포함하는 상기 제2 본딩 모듈 - 상기 제2 본딩 모듈은 서버에 접속될 수 있으며 그리고 상기 제2 스케줄링 모듈은 상기 TCP 터널들을 통해 상기 제1 본딩 모듈을 향하여 데이터 패킷들을 스케줄링하고 분배하도록 구성됨 - 과;

유선 또는 무선 접속을 통해 상기 제1 본딩 모듈에 접속된 제1 라우터와; 그리고

유선 또는 무선 접속을 통해 상기 제2 본딩 모듈에 접속된 제2 라우터를 포함하고,

상기 제1 스케줄링 모듈 또는 상기 제2 스케줄링 모듈은 상기 TCP 터널들로부터의 고유 정보(native information)에 기초하여 상기 제1 라우터와 상기 제2 라우터 사이에서 상기 TCP 터널들의 용량을 획득하도록 구성되며, 그리고

상기 TCP 터널들의 고유 정보는 혼잡 윈도우(C_{wnd})의 크기, TCP 슬로우 스타트 임계값(S_{thresh}), 평활(smoothed) 왕복 시간(T_{RTT}), 전송된 패킷들의 수(P_{out}), 확인된 패킷들의 수(P_{sacked}), 재전송된 패킷들의 수($P_{retrans}$) 및 손실된 것으로 간주되는 패킷들의 수(P_{lost}) 중 적어도 하나를 포함하는 것을 특징으로 하는

패킷 기반 스케줄링 시스템.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

제1항에 있어서,

상기 제1 스케줄링 모듈 또는 상기 제2 스케줄링 모듈은 상기 제1 라우터로부터 상기 제2 라우터로 현재 이동하고 있는 패킷들의 수, $P_{fly} = P_{out} - P_{sacked} + P_{retrans} - P_{lost}$ 를 획득하도록 구성되는 것을 특징으로 하는

패킷 기반 스케줄링 시스템.

청구항 7

제6항에 있어서,

상기 제1 스케줄링 모듈 또는 상기 제2 스케줄링 모듈은 잔여 용량(leftover capacity), $C_{left} = C_{ewnd} - P_{fly}$ 을 획득하도록 구성되는 것을 특징으로 하는

패킷 기반 스케줄링 시스템.

청구항 8

제7항에 있어서,

상기 제1 스케줄링 모듈 또는 상기 제2 스케줄링 모듈은 정의된 간격 내에서 상기 잔여 용량의 변화를 ΔC_{left} 을 획득하도록 구성되는 것을 특징으로 하는

패킷 기반 스케줄링 시스템.

청구항 9

제1항에 있어서,

상기 제1 스케줄링 모듈 또는 상기 제2 스케줄링 모듈은 상기 제1 라우터와 상기 제2 라우터 사이의 상기 TCP 터널들에 대한 스케줄링 가중치들을 획득하기 위해 스케줄링 알고리즘을 사용하도록 구성되는 것을 특징으로 하는

패킷 기반 스케줄링 시스템.

청구항 10

제9항에 있어서,

상기 스케줄링 가중치들은, 상기 터널들 각각을 통해 전송된 패킷들의 양(amount) 또는 로우(row)인 상기 터널들의 각각을 통해 전송된 패킷들의 양의 비율을 포함하는 것을 특징으로 하는

패킷 기반 스케줄링 시스템.

청구항 11

제10항에 있어서,

상기 제1 스케줄링 모듈 또는 상기 제2 스케줄링 모듈은 스케줄링 가중치들을 연속적으로 또는 타임슬롯들로 조정(adapt)하도록 구성되는 것을 특징으로 하는

패킷 기반 스케줄링 시스템.

청구항 12

전송 제어 프로토콜 세션들에 대한 패킷 기반 스케줄링 방법으로서,

- (a) 제1 스케줄링 모듈 및 적어도 하나의 전송 네트워크에 접속된 적어도 2개의 액세스 인터페이스들을 포함하는 제1 본딩 모듈을 접속하는 단계 - 상기 제1 본딩 모듈은 사용자 디바이스에 접속될 수 있음 - 와;
- (b) 제2 본딩 모듈에서 종단하는 상기 액세스 인터페이스들 각각을 통해 TCP 터널은 구성하는 단계와;
- (c) 상기 TCP 터널들 및 상기 적어도 하나의 전송 네트워크를 통해 상기 제1 본딩 모듈로부터 상기 제2 본딩 모듈을 향해 데이터 패킷들을 스케줄링하고 분배하는 단계와; 또는
- (d) 제2 스케줄링 모듈 및 적어도 하나의 액세스 인터페이스를 포함하는 상기 제2 본딩 모듈을 상기 적어도 하나의 전송 네트워크 각각에 접속시키는 단계 - 상기 제2 본딩 모듈은 서버에 접속될 수 있음 - 와;
- (e) 상기 제2 본딩 모듈로부터 상기 제1 본딩 모듈쪽으로 상기 TCP 터널들을 통해 데이터 패킷들을 스케줄링하고 분배하는 단계와;
- (f) 유선 또는 무선 접속을 통해 제1 라우터를 상기 제1 본딩 모듈에 접속시키는 단계와;

(g) 유선 또는 무선 접속을 통해 제2 라우터를 상기 제2 본딩 모듈에 접속시키는 단계와; 그리고
 (f) 상기 TCP 터널들로부터의 고유 정보에 기초하여 상기 제1 라우터와 상기 제2 라우터 사이에서 상기 TCP 터널들의 용량을, 상기 제1 스케줄링 모듈 또는 상기 제2 스케줄링 모듈에 의해 획득하는 단계를 포함하고,
 상기 TCP 터널들의 고유 정보는 혼잡 윈도우(C_{wnd})의 크기, TCP 슬로우 스타트 임계값(S_{thresh}), 평활(smoothed) 왕복 시간(T_{RTT}), 전송된 패킷들의 수(P_{out}), 확인된 패킷들의 수(P_{sacked}), 재전송된 패킷들의 수($P_{retrans}$) 및 손실된 것으로 간주되는 패킷들의 수(P_{lost}) 중 적어도 하나를 포함하는 것을 특징으로 하는
 패킷 기반 스케줄링 방법.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

제12항에 있어서,

상기 데이터 패킷들을 스케줄링하고 분배하는 단계는 상기 제1 라우터와 상기 제2 라우터 사이의 상기 TCP 터널들에 대한 스케줄링 가중치들을 획득하기 위한 스케줄링 알고리즘에 기초하는 것을 특징으로 하는

패킷 기반 스케줄링 방법.

청구항 17

제16항에 있어서,

상기 스케줄링 가중치들은, 상기 터널들 각각을 통해 전송된 패킷들의 양 또는 로우인 상기 터널들 각각을 통해 전송된 패킷들의 양의 비율을 포함하는 것을 특징으로 하는

패킷 기반 스케줄링 방법.

청구항 18

제17항에 있어서,

상기 스케줄링 가중치들은 연속적으로 또는 타임슬롯들로 조정되는 것을 특징으로 하는

패킷 기반 스케줄링 방법.

발명의 설명

기술 분야

[0001] 본 발명은 전송 제어 프로토콜(TCP) 세션들 또는 사용자 데이터그램 프로토콜(UDP) 세션들에 대해 패킷 기반 스케줄링을 사용하는 리소스 번들링을 위한 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 과거에, 세션/애플리케이션 접근 방식, 전송 접근 방식, 네트워크 접근 방식 및 링크 계층 접근 방식으로 크게 그룹화할 수 있는 리소스 번들링을 위한 접근 방식들에 대해 많은 노력들이 있어왔다. 다음에서는 이러한 그룹들 각각에 대한 가장 중요한 예들이 간단히 논의될 것이다.

[0003] HTTP 범위 요청은 리소스 번들링을 수행하는 데에도 사용될 수 있는 세션/애플리케이션-계층 피처이다. 이러한

피처는 콘텐트를 작은 부분들로 다운로드하기 위해 분할하며, 이후, 상이한 인터페이스들을 통해 별도의 TCP 세션들에 다운로드된다. 이러한 피처는, 삼성 갤럭시 S5 내에서 구현되며, 그리고 "네트워크 부스터(Network Booster)"라고 불린다(에스. 이. 코포레이션, "향상된 데이터 성능을 위한 네트워크 부스터("Network booster - for enhanced data performance," 기술 문서).

[0004] 전송 계층 솔루션의 가장 눈에 띄는 예로는 RFC(Request for Comment) 6824의 IETF(Internet Engineering Task Force)에서 지정한 다중경로 TCP(MPTCP)가 있다(A. Ford, C. Raiciu, M. Handley 및 O. Bonaventure, "다중 주소들로 다중 경로 작업을 위한 Tcp 확장들", RFC, 번호 6824, 2013년 1월). MPTCP는 애플리케이션들에 대해 투명한 피어들 간에 다중 경로들을 동시에 사용할 수 있게 해주는 TCP의 확장으로 볼 수 있다. MPTCP는 이미 아이폰과 같은 상용 제품들에 사용된다. 예를 들어, 시리(Siri) 서비스는 신뢰성을 향상시키기 위해 MPTCP를 사용한다. 아이폰이 셀룰러 및 Wi-Fi 접속들을 갖는 경우, 시리는 Wi-Fi와 셀룰러 네트워크를 통해 2개의 셱션들을 시작한다. 접속들 중 하나를 사용할 수 없거나 신뢰할 수 없게 되면, 백업 접속을 즉시 사용할 수 있다.

[0005] 또 다른 전송 계층 예는, 스트림 제어 전송 프로토콜(Stream Control Transmission Protocol: SCTP)이다(R. Stewart 외, "스트림 제어 전송 프로토콜" RFC 2960(제안된 표준), 인터넷 엔지니어링 태스크 포스, 2000년 10월, RFC 4960에 의해 폐기됨, RFC 3309에 의해 업데이트 됨). SCTP의 장점 중 하나는 원칙적으로 링크들의 번들링을 가능하게하는 다중-홈 피처이다.

[0006] 원칙적으로 세션/애플리케이션 계층 접근법들과 함께 전송 계층 접근법들은 네트워크의 특별한 개입 없이 최종 디바이스들에 의해 번들링이 수행되므로 종단 간(E2E) 시나리오들에 적합하다는 점을 말할 수 있다. E2E 접근법들의 단점은 인터넷 서버들뿐 아니라 사용자 디바이스들에 의해 지원되어야 하는 기술을 도입하려는 노력이 필요하다는 것이다.

[0007] 네트워크 계층 접근법들은 최종 디바이스들을 변경할 필요없이 용이하게 구현할 수 있으므로 네트워크 오퍼레이터들에게 선호되는 솔루션이다. 대신, 오퍼레이터가 제어하고 최종 디바이스들에 투명하게 번들링을 수행하는 프록시 서버들이 사용된다.

[0008] 링크 계층 번들링이란 동등한 기술의 다수의 채널들을 번들링하는 것을 말한다. 그러한 예들로는, IEEE 802.3ad(WO 2009/019258 A1)에 정의된 이더넷 링크의 번들링 및 아데로스 슈퍼워드 모드에서 구현된 Wi-Fi 채널들의 번들링이다. SmartAP 솔루션(E.G. Llairo와 D. Giustiniano, "Smartap : Practical wlan backhaul aggregation", Wireless Days. IEEE, 2013, pp. 1-7에서 제안됨)은 단일 라디오 다중 채널 시각화를 사용하는 무선 근거리 통신망(WLAN) 백홀의 집합을 다룬다. 이는, 상이한 채널들이 사용되는 경우에도 다수의 이웃하는 액세스 포인트(AP)들과의 접속들을 가능하게 한다. D. Giustiniano, E. Goma, A. Lopez Toledo, I. Dangerfield, J. Morillo 및 P. Rodriguez, "Fair wlan backhaul aggregation", 제16회 연례 국제 학술회의 모바일 컴퓨팅 및 네트워킹, MobiCom '10. New York, NY, USA:ACM, 2010, pp. 269-280은 상기 서술된 시스템의 공정성을 조사한다.

발명의 내용

해결하려는 과제

[0009] 본 발명의 목적은, 향상된 성능을 갖는 전송 제어 프로토콜(TCP) 세션들 또는 사용자 데이터그램 프로토콜(UDP) 세션들에 대해 패킷 기반 스케줄링을 사용하는 방법 및 시스템을 제공하는 것이다. 상기 목적은 독립항들의 피처들로 성취된다. 종속항들은 본 발명의 추가 양상들에 관한 것이다.

과제의 해결 수단

[0010] 본 발명은 바람직하게는 네트워크 오퍼레이터들에게 적합한 일반 번들링 아키텍처를 제안한다. 그러나, 본 발명은 오버-더-톱(OTT) 컨텐트 시나리오들에도 적합하다. 다른 접근 방법들과는 대조적으로 제시된 접근법은 시스템의 복잡성을 줄이고 높은 번들링 이득들을 제공하기 위해 TCP 터널들을 사용한다. 전통적인 번들 솔루션들과 같은 경로 모니터들은 필요하지 않다. 대신, TCP 터널들에서 오는 고유 정보는 가장 효율적인 패킷 스케줄링을 결정하기 위해 패킷 스케줄러들에 대한 입력으로 사용된다. 즉, 번들링 이득을 최대화하기 위해 스케줄링 접근법에 대한 가중치들을 결정할 수 있는 혁신적인 솔루션이 서술된다. 즉, TCP 터널들 및 특히 TCP 터널들로부터의 고유 정보는 각 터널을 통해 사용 가능한 대역폭을 추정하기 위해 사용될 수 있다. 이 정보는 사용 가능한 터널들에 대한 데이터 패킷들의 효율적인 매핑에 사용된다. 본 발명에 따르면, 상이한 터널들은 상이한 네트워크 인터페이스들, 예를 들어, WiFi 및 3GPP 통해 전너갈 수 있고, 각각의 터널을 통한 이용 가능한 대역폭은 상당히

상이할 수 있다. 본 발명은 터널들 각각을 통해 이용가능한 대역폭을 고려하여 효율적인 대역폭 집합을 제공한다.

[0011] 본 발명의 제1 양상에 따르면, 전송 제어 프로토콜(Transmission Control Protocol; TCP) 세션들 또는 사용자 데이터그램 프로토콜(User Datagram Protocol; UDP) 세션들에 대한 패킷 기반 스케줄링 시스템이 제공된다. 상기 시스템은 제1 스케줄링 모듈 및 적어도 하나의 전송 네트워크에 접속된 적어도 2개의 액세스 인터페이스들을 포함하는 제1 본딩 모듈을 포함하고, 상기 제1 본딩 모듈은 사용자 디바이스에 접속될 수 있고, TCP 터널은 제2 본딩 모듈에서 종단(terminate)하는 상기 액세스 인터페이스들 각각을 통해 구성되며, 그리고 상기 제1 스케줄링 모듈은 바람직하게는 인터넷을 통해 상기 TCP 터널들을 거쳐 상기 제2 본딩 모듈을 향하여 데이터 패킷들을 스케줄링 및 분배하도록 구성된다. 상기 시스템은, 제2 스케줄링 모듈 및 상기 적어도 하나의 전송 네트워크 각각에 접속된 적어도 하나의 액세스 인터페이스를 포함하는 상기 제2 본딩 모듈을, 홀로 또는 결합하여 더 포함하고, 상기 제2 본딩 모듈은 서버에 접속될 수 있으며, 그리고 상기 제2 스케줄링 모듈은 상기 TCP 터널들을 통해 상기 제1 본딩 모듈을 향하여 데이터 패킷들을 스케줄링하고 분배하도록 구성된다.

[0012] 데이터 패킷들은 네트워크 계층(계층 3)에 분산되는 것이 바람직하다. 이는 표준 TCP 프로토콜을 사용을 가능하게 한다. 따라서, 새로운 TCP 프로토콜이 필요하지 않을 수 있다.

[0013] 상기 시스템은, 유선 또는 무선 접속을 통해 상기 제1 본딩 모듈에 접속된 제1 라우터를 더 포함할 수 있고, 바람직하게는 상기 제1 라우터는 상기 사용자 디바이스로부터 데이터를 수신하고 상기 사용자 디바이스로 데이터를 송신하도록 구성된다.

[0014] 상기 시스템은, 유선 또는 무선 접속을 통해 상기 제2 본딩 모듈에 접속된 제2 라우터를 더 포함할 수 있고, 바람직하게는 상기 제2 라우터는 바람직하게 상기 서버로부터 데이터를 수신하고 상기 서버로 데이터를 송신하도록 구성된 네트워크 어드레스 변환 모듈을 포함한다.

[0015] 제1 라우터는 멀티-홈일 수 있는데, 이는 제1 라우터가 적어도 2개의 인터페이스들을 통해 인터넷에 접속됨을 의미한다. 2개의 라우터들(제1 및 제2 라우터) 모두 상이한 네트워크들에 속할 수 있으며 TCP 터널들을 통해 서로 접속된다. 2개의 라우터들 간의 TCP 터널들의 수는, 제1 라우터가 인터넷에 접속된 인터페이스들의 수에 기초한다. 이러한 인터페이스들 각각을 통해 하나의 터널이 사용된다.

[0016] "접속된(connected)"이라는 용어는, 특정 모듈이 접속된 각각의 모듈의 일부분, 즉 모듈들이 내부적으로 접속될 수 있음을 의미할 수도 있다. 예를 들어, 제1 본딩 모듈 및/또는 제2 본딩 모듈은 각각 사용자 디바이스 및/또는 서버의 필수 부분일 수 있다. 다른 예로서, 제1 본딩 모듈 및/또는 제2 본딩 모듈은 각각 제1 라우터 및/또는 제2 라우터의 필수 부분일 수 있다.

[0017] 바람직하게는, 상기 제1 스케줄링 모듈 및/또는 상기 제2 스케줄링 모듈은 상기 TCP 터널들로부터의 고유 정보에 기초하여 상기 제1 라우터와 상기 제2 라우터 사이에서 상기 TCP 터널들의 용량을 획득하도록 구성된다. 이러한 정보는 TCP 터널들 각각에 대한 스케줄링 가중치를 결정하는 데 사용된다.

[0018] 더욱이, 상기 TCP 터널들의 고유 정보는 혼잡 윈도우(C_{cwnd})의 크기, TCP 슬로우 스타트 임계값(S_{thresh}), 평활(smoothed) 왕복 시간(T_{RTT}), 전송된 패킷들의 수(P_{out}), 확인된 패킷들의 수(P_{sacked}), 재전송된 패킷들의 수($P_{retrans}$) 및 손실된 것으로 간주되는 패킷들의 수(P_{lost}) 중 적어도 하나를 포함할 수 있다. 대안으로 또는 부가적으로, TCP 터널 인터페이스들의 송신기 큐 크기는 추가적인 파라미터로서 고려될 수 있다.

[0019] 본 발명의 추가적인 양상에 따르면, 상기 제1 스케줄링 모듈 및/또는 상기 제2 스케줄링 모듈은 $P_{fly} = P_{out} - P_{sacked} + P_{retrans} - P_{lost}$ 로서 상기 제1 라우터로부터 상기 제2 라우터로 현재 이동하고 있는 패킷들의 수를 획득하도록 구성된다.

[0020] 보다 바람직하게, 상기 제1 스케줄링 모듈 및/또는 상기 제2 스케줄링 모듈은 $C_{left} = C_{cwnd} - P_{fly}$ 로서 잔여 용량(leftover capacity)을 획득하도록 구성된다.

[0021] 더욱 바람직하게, 상기 제1 스케줄링 모듈 및/또는 상기 제2 스케줄링 모듈은 정의된 간격 내에서 상기 잔여 용량의 변화를 ΔC_{left} 을 획득하도록 구성된다.

[0022] 하지만, 상기 파라미터들의 리스트는 본 발명의 동일한 효과들을 실현하기 위해 다른 파라미터들이 획득되고 사

용될 수 있다는 점에서 포괄적인 것은 아니며, 이는 당업자에 의해 이해될 것임을 알 수 있다.

[0023] 본 발명의 추가적인 양상에 따르면, 상기 제1 스케줄링 모듈 및/또는 상기 제2 스케줄링 모듈은 상기 제1 라우터와 상기 제2 라우터 사이의 상기 TCP 터널들에 대한 스케줄링 가중치들을 획득하기 위해 스케줄링 알고리즘을 사용하도록 구성된다.

[0024] 바람직하게, 상기 스케줄링 가중치들은, 상기 터널들 각각을 통해 전송된 패킷들의 양(amount) 및/또는 로우(row)인 상기 터널들의 각각을 통해 전송된 패킷들의 양의 비율을 포함한다.

[0025] 추가적으로, 상기 제1 스케줄링 모듈 및/또는 상기 제2 스케줄링 모듈은 스케줄링 가중치들을 연속적으로 또는 타임슬롯들로 조정(adapt)하도록 구성될 수 있다.

[0026] 바람직하게, 상기 제1 스케줄링 모듈 및/또는 상기 제2 스케줄링 모듈은 이전의 타임 슬롯 동안 수집된 파라미터들에 기초하여 후속(subsequent) 타임 슬롯의 스케줄링 가중치들을 조정하도록 구성될 수 있다.

[0027] 본 발명의 제2 양상에 따르면, 바람직하게는 상기에서 서술된 시스템을 사용하여, 전송 제어 프로토콜 세션들에 대한 패킷 기반 스케줄링 방법이 제공된다. 상기 방법은: (a) 제1 스케줄링 모듈 및 적어도 하나의 전송 네트워크에 접속된 적어도 2개의 액세스 인터페이스들을 포함하는 제1 본딩 모듈을 접속시키는 단계 - 상기 제1 본딩 모듈은 사용자 디바이스에 접속될 수 있으며 - 와; (b) 제2 본딩 모듈에서 종단하는 상기 액세스 인터페이스들 각각을 통해 TCP 터널은 구성하는 단계와; (c) 상기 TCP 터널들 및 상기 적어도 하나의 전송 네트워크를 통해 상기 제1 본딩 모듈로부터 상기 제2 본딩 모듈을 향해 데이터 패킷들을 스케줄링하고 분배하는 단계와; (d) 제2 스케줄링 모듈 및 적어도 하나의 액세스 인터페이스를 포함하는 상기 제2 본딩 모듈을 상기 적어도 하나의 전송 네트워크 각각에 접속시키는 단계 - 상기 제2 본딩 모듈은 서버에 접속될 수 있으며 - 와; 그리고 (e) 상기 제2 본딩 모듈로부터 상기 제1 본딩 모듈쪽으로 상기 TCP 터널들을 통해 데이터 패킷들을 스케줄링하고 분배하는 단계를 포함한다.

[0028] 상기 방법은, 유선 또는 무선 접속을 통해 제1 라우터를 상기 제1 본딩 모듈에 접속시키는 단계를 더 포함할 수 있고, 바람직하게는 상기 제1 라우터는 상기 사용자 디바이스로부터 데이터를 수신하고 상기 사용자 디바이스로 데이터를 송신하도록 구성된다.

[0029] 상기 방법은 유선 또는 무선 접속을 통해 제2 라우터를 상기 제2 본딩 모듈에 접속시키는 단계를 또한 포함할 수 있고, 바람직하게는 상기 제2 라우터는 바람직하게 상기 서버로부터 데이터를 수신하고 상기 서버로 데이터를 송신하도록 구성된 네트워크 어드레스 변환 모듈을 포함한다.

[0030] 다시 말하면, 접속하는 것은 2개의 개별적인 디바이스들/모듈들을 접속하는 것을 의미할 수 있거나 또는 제1 디바이스/모듈을 제2 디바이스/모듈에 내부적으로 접속하는 것을 의미할 수 있다; 즉, 제2 디바이스/모듈은 제1 디바이스/모듈에 통합될 수 있음이 명확하다.

[0031] 바람직하게, 상기 방법은 상기 TCP 터널들로부터의 고유 정보에 기초하여 상기 제1 라우터와 상기 제2 라우터 사이에서 적어도 2개의 상기 TCP 터널들 각각의 용량들을 획득하는 단계를 포함한다.

[0032] 추가적으로, 상기 데이터 패킷들을 스케줄링하고 분배하는 단계는 상기 제1 라우터와 상기 제2 라우터 사이의 상기 TCP 터널들에 대한 스케줄링 가중치들을 획득하기 위한 스케줄링 알고리즘에 기초할 수 있다.

[0033] 상기 스케줄링 가중치들은, 상기 터널들 각각을 통해 전송된 패킷들의 양 및/또는 로우인 상기 터널들의 각각을 통해 전송된 패킷들의 양의 비율을 포함할 수 있다.

[0034] 바람직하게, 상기 스케줄링 가중치들은 연속적으로 또는 타임슬롯들로 조정된다.

[0035] 더 바람직하게는, 후속 타임슬롯의 스케줄링 가중치들은 이전의 타임슬롯 동안 수집된 파라미터들에 기초할 수 있다.

[0036] 현재의 모든 솔루션들에 공통적인 것은 사용 가능한 링크들을 통해 패킷들을 스케줄링하는 것이다. 상이한 액세스 네트워크들의 리소스들을 번들링하는 것은 네트워크 장애에 대한 처리량과 복원력을 향상시키는 방법이다. 그러나 액세스 네트워크 리소스들의 번들링은 특히 TCP 트래픽의 경우 매우 까다로운 작업이다. 특히, 액세스 리소스 번들링을 수행할 때의 과제들은 다음과 같다.

[0037] * 첫 번째 과제는, 오늘날의 네트워크들에서 가장 많이 사용되는 전송 프로토콜들이 TCP(Transmission Control Protocol) 및 UDP(User Datagram Protocol)이며 이러한 프로토콜들이 여러 링크들에서 작동하도록 설계되지 않

았다는 사실에 기인한다. 이러한 프로토콜들은, 동일한 세션에 속한 패킷들이 하나의 인터넷 프로토콜(IP) 주소에서만 수신된다고 가정한다. 하지만, 다수의 액세스 링크들이 사용되면 각 인터페이스마다 상이한 IP 주소가 부여된다. 이 문제를 극복하기 위해, 터미널에 의해 사용되는 모든 액세스 링크를 대신하여 NAT(Network Address Translation)를 수행하는 추가 상자가 도입되었다.

[0038] • 두 번째 과제는, 사용 가능한 링크들을 통한 패킷들의 스케줄링과 관련된다. 우선 두 가지 기본 유형들의 스케줄링 방식: 세션 기초 및 패킷 기초를 구분할 수 있다. 세션 기초 스케줄링이 수행될 때 세션들이 사용 가능한 링크로 분배되는 동안 특정 세션의 패킷들은 항상 동일한 링크를 통해 전송될 것이다. 예를 들어, 세션 A의 패킷들은 링크 L_1 를 통해 전송되는 반면 세션 B의 패킷들은 링크 L_2 를 통해 전송된다. 세션 기초 접근법에는 몇 가지 단점들을 갖는다. 예를 들어, 단일 세션은 다수의 사용 가능한 링크들로부터 이익을 얻을 수 없다. 동일한 단말로/부터의 여러 세션들의 경우에도 상이한 링크들의 성능이 크게 다를 경우 번들링 이득을 보장될 수 없다. 본 발명의 초점은 단일 세션의 패킷들이 이용 가능한 링크들을 통해 분배되는 패킷 기반 스케줄링 접근법에 있다. 사용 가능한 대역폭(B), 지연(D), 지연 지터(J) 및 패킷 손실(L) 측면에서 상이한 링크들이 상이한 특성을 가질 수 있으므로, 패킷들이 사용 가능한 링크들을 통해 균등하게 분배되는 라운드 로빈 스케줄링을 수행하는 것으로는 충분하지 않을 수 있다. 대신, 가중치가 있는 스케줄링이 선호된다. 특히, TCP 세션들의 경우, 이러한 가중치들은 엔드-투-엔드 성능에 중요한 영향을 미치므로 이러한 가중치들은 신중하게 결정되는 것이 바람직하다.

[0039] • 마지막으로, 패킷 순서 재-지정의 과제들은 패킷 기반 스케줄링이 적용될 때 작용한다. 상이한 링크들의 상이한 특성을 및 패킷들이 인터넷을 통해 완전히 상이한 경로들을 통해 이동할 수 있는 사실로 인해, 패킷들이 잘못된 순서로 대상에 도착할 수 있다. 애플리케이션에 따라, 서비스 품질이 저하될 수 있다.

[0040] 몇몇 바람직한 실시예들은 이제 도면들을 참조하여 설명된다. 설명의 목적으로, 청구된 본 발명의 범위를 벗어나지 않는 한, 다양한 특정 세부 사항들이 제시된다.

도면의 간단한 설명

[0041] 도 1은 종래 기술에 따른 멀티 홈(multi-homed) 단말의 예를 도시한 것이다.

도 2a 내지 도 2d는 본 발명의 네 가지 바람직한 실시예들을 도시한다.

도 3은 본 발명의 일 실시예에 따른 슬롯들에서의 스케줄링 가중치의 적용을 도시한다.

도 4는 본 발명의 일 실시예에 따른 스케줄링 알고리즘의 일례를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0042] 원칙적으로, 인터넷(11)의 서버(12)에 접속된 단말들(10)인 소위 멀티 홈 단말들(10)에 대한 리소스 번들링은 적어도 2개의 별개의 또는 상이한 링크들(L_1, L_2, \dots, L_n)을 사용하여 이루어진다. 일 예가 도 1에서 도시된다.

[0043] 단말(10)은 다수의 라인들(L_1, L_2, \dots, L_n)로 인터넷(11)에 접속된다. 오늘날의 단말들(10)은 일반적으로 인터넷(11) 내의 서버(12)와의 통신을 위해 라인들 중 단지 하나의 라인만을 사용한다. 일반적으로, 사용되는 인터페이스들의 선택은 계층적 정책을 기초로 한다. 예를 들어, 스마트 폰들은 3GPP 인터페이스를 통해 Wi-Fi 인터페이스를 선호 할 수 있지만 랩톱들은 Wi-Fi 인터페이스를 통해 이더넷 인터페이스를 선호할 수 있다. 그 결과, 인터넷(11)을 향한 다른 링크들이 존재하더라도, 단말에 의해 인지되는 최대 대역폭은 선택된 네트워크 인터페이스에 의해 이용 가능한 대역폭으로 제한된다. 모든 이용 가능한 인터페이스들(N)의 동시 사용의 가능성을 가정하면, 단말(10) 당 감지된 대역폭은 상당히 증가할 것이다. 이상적인 경우, 터미널(10) 당 전체 대역폭(BW_{ideal})은 단순히 각 인터페이스(BW_i)에 의해 제공하는 대역폭의 합이다. 그러나, 실제 구현들에서는 좀 더 어려워 100% 번들링 효율(E)은 달성하기 거의 불가능하다. 따라서, 단말기(10)(BW)에 의해 인지되는 대역폭은 다음과 같이 표현될 수 있다:

수학식 1

$$BW = \left(\sum_{i=1}^N BW_i \right) E$$

[0044]

[0045] 이하에서, 상기에서 서술된 과제들 중 하나 이상에 대한 해결책을 제공하는 액세스 자원 번들링을 위한 아키텍처가 제시된다. 또한, 번들링 이득을 최대화하기 위해 스케줄링 접근법에 대한 가중치를 결정할 수 있는 혁신적인 솔루션이 제공된다. 먼저, 제안된 전체 아키텍처에 대해 설명하고, 이후 네이티브 TCP 정보에 기초한 패킷들의 가중치 스케줄링에 대한 주요 과제에 대해 자세히 설명한다.

[0046]

본 발명의 바람직한 실시예에 따른 해결책의 개관이 도 2(a)에 예시되어 있다. 좌측의 노드(C)는 사용자 디바이스(100), 예를 들어, 스마트폰 또는 램프를 나타내고, 우측의 디바이스(S)는 인터넷(I)(107)에서의 서버(108), 예를 들어, 애플리케이션 서버이다. 사용자 디바이스(100) 및 서버(108)는 표준 디바이스들이다. 본딩 인텔리전스는 제1 라우터(R₁)(101) 및 제2 라우터(R₂)(104)내의 상기 디바이스들에서 명백하게 구현된다. 제1 라우터(101)는 사용자 디바이스(100)(예를 들어, DSL 라우터)에 대해 인터넷(107)을 향한 게이트웨이이고 그리고 몇 개의 액세스 인터페이스들(여기에서는, 예시를 위해, 2개의 액세스 인터페이스들 L₁ 및 L₂로 도시됨)을 갖는다. 제2 라우터(104)는 네트워크 오퍼레이터의 하나 이상의 전송 네트워크들(103) 내의 라우터를 나타낸다. 2개의 라우터들(101, 104) 모두는 네트워크 오퍼레이터의 전송 네트워크(TR)(103)를 통해 서로 접속한다. 제1 라우터(101)에 의해 사용된 액세스 인터페이스들(L₁, L₂) 각각을 통해, 액세스 인터페이스들(L₁, L₂) 각각에 대해 제2 라우터(104)를 향한 각 TCP 터널(tun₁ 및 tun₂)을 구성한다.

[0047]

이들 터널들(tun₁, tun₂)의 목적은 사용자 디바이스(100)와 서버(108) 사이에서 데이터 패킷들을 전송하는 것이다. 라우터들(101, 104) 각각은 또한 구현되는 각각의 스케줄링 모듈(SC)(102, 105)을 갖는다. 스케줄링 모듈들(102, 105) 각각은 모든 관련 정보에 실시간으로 액세스하기 위해 라우터들(101, 104) 각각에 통합(내부적으로 접속)될 수 있음을 알아야 한다. 그러나, 스케줄링 모듈은 유선 또는 무선 접속을 통해 라우터에 간단히 부착(외부 연결)되는 외부 모듈로 구성될 수도 있다. 다음에서, "연결(connected)"이라는 용어는 내부 연결(또는 통합) 및/또는 외부 연결(또는 부착)에 대해 사용된다. 즉, 도 2(a)는, 스케줄링 모듈들(102, 105)이 각각 제1 및 제2 라우터(101, 104)에 통합되어 있음을 나타내지만, 이들 모듈들(102, 105)을 개별 디바이스들(도시하지 않음)로서 제공하는 것도 가능하다. 스케줄링 모듈들(102, 105)은 상이한 터널 인터페이스들(tun₁, tun₂)을 통한 패킷들의 스케줄링 및 분배를 담당한다. 최종적으로, 제2 라우터(104)는 인터넷(107)을 향한 인터페이스상에서 NAT-모듈(106)을 사용하고 있다. 스케줄링 모듈들(102, 105) 각각 및 대응하는 인터페이스들의 결합은 각각의 본딩 모듈이라 칭할 수 있다.

[0048]

도 2(b) 내지 (d)는 본 발명에 따른 또 다른 대안적인 해결책들을 도시한다. 컴포넌트들은 도 2(a)의 컴포넌트에 따라 대응하는 참조 부호들로 표시된다.

[0049]

여기서, 도 2(b)는 제1 및 제2 라우터(101, 104)가 생략된 다른 실시예를 도시한다. 즉, 스케줄링 모듈들(102, 105)은 각각 사용자 디바이스(100) 및 서버(108)에 통합된다(접속된다). 이러한 구성에서, 제1 번들링은 더 이상 제1 라우터(101)에서 실행되지 않고, 인터넷의 몇몇 경로들을 포함하는 사용자의 사용자 디바이스(100)에서 직접 실행된다. 또한, 제2 번들링은 제2 라우터(104)에서 더 이상 실행되지 않고 인터넷의 서버(108)에서 직접 실행된다.

[0050]

도 2(c) 및 도 2(d)는 도 2(a) 및 도 2(b)에 도시된 실시예들의 추가적인 변형들을 도시한다. 특히, 도 2(c)에서는 제2 라우터(104)만이 생략되고, 도 2(d)에서는 도 2(a)의 실시예와 비교하여 제1 라우터(101)만이 생략된다. 제1 및 제2 스케줄링 모듈(102, 105)은 각각 사용자 디바이스(100) 및 서버(108)에 통합될 수 있거나 또는 별개의 외부 디바이스들(도시되지 않음)로서 제공될 수 있음을 이해해야 한다.

[0051]

예를 들어, D. Kaspar의 "Multipath aggregation of heterogeneous access networks," SIGMultimedia Rec, vol. 4, no. 1, pp. 27- 28, Mar. 2012에서 제시된 것과 같은 알려진 접근 방식들과 비교할 때, 도 1의 실시예

에 따른 본 발명은 두 가지 중요한 단순화들을 갖는다. 먼저, 패킷들의 재정렬을 담당하는 디-지터 버퍼가 사용되지 않는다. 대신에, 패킷들의 재-순서화는 TCP의 네이티브 재-순서화 알고리즘으로 남겨진다. 즉, 재-순서화는 사용자 디바이스(100) 또는 인터넷 서버(108)에 의해 수행된다. 따라서, 시스템은, 추가적인 재-순서화 모듈을 필요로하지 않기 때문에, 사용자 디바이스(100) 또는 인터넷 서버(108)에 의해 행해진 재-순서화로부터 이득을 얻는다. 이는 다른 솔루션과 비교하여 시스템을 단순화한다. 둘째로, 제1 라우터(101)와 제2 라우터(104) 사이에서 TCP 터널들이 사용되는 사실로 인해, 링크 성능들의 결정을 담당하는 경로 모니터 모듈은 요구되지 않는다. 이를 터널들(tun_1, tun_2)로부터의 정보에 기초하여, 링크들(L_1, L_2)의 용량이 추정될 수 있다.

[0052] 따라서, 본 발명은 상기에서 서술된 과제들 중 하나 이상의 해결책을 제공할 수 있다. NAT는 제2 라우터(104)에서 수행되며, 이는, 패킷이 전송된 경로와 독립적인 클라이언트의 동일한 IP 주소를 서버(108)가 항상 보는 터널(tun_1, tun_2)의 사용과 함께 보장한다. 서버(108)에 의해 보여지는 사용자 디바이스(100)의 IP 어드레스는 제2 라우터(104)의 공용 IP 어드레스이다. 따라서, 서버(108)로부터 사용자 디바이스(100)로의 패킷들은 항상 제2 라우터(104)로 향할 것이다. TCP 터널들(tun_1, tun_2)의 사용은 경로 모니터 모듈의 요구 사항들을 피하면서 동시에 스케줄링 프로세스에 필요한 정보를 제공한다. TCP가 항상 혼잡 창(Congestion Window: C_{cwnd})의 적응을 기초로 매(every) 터널(tun_1, tun_2)에서 대역폭 사용을 최대화하려고 시도하려는 사실이 여기에서 활용된다.

[0053] TCP 터널들(tun_1, tun_2)에서 사용할 수 있는 정보의 예들은 다음과 같다:

[0054] • C_{cwnd} 의 크기 측면에서 링크의 용량,

[0055] • TCP 슬로우스타트 임계값(S_{thresh}),

[0056] • TCP에 의해 결정된/계산된 RTT인 평활한(smoothed) 왕복시간(T_{RTT}),

[0057] • 전송된 패킷들의 수인 P_{out} ,

[0058] • 확인된 패킷들의 수인 P_{sacked} ,

[0059] • 재전송된 패킷들의 수인 $P_{retrans}$,

[0060] • 손실된 것으로 간주되는 패킷들의 수인 P_{lost} .

[0061] 이러한 기본적인 파라미터들에 기초하여, 추가적인 파라미터들이 도출될 수 있다. 예를 들어:

[0062] • 발신자에 의해 전송되지만 아직 확인되지 않은 패킷들인 온 더 플라이 패킷들(P_{fly}). 따라서, 이들은 현재 발신자에서 수신자로 이동하는 패킷들이다.

$$P_{fly} = P_{out} - P_{sacked} + P_{retrans} - P_{lost}$$

[0063] .

[0064] • 단순히 C_{cwnd} - P_{fly} 이며 용량에 도달할 때까지 다른 확인을 기다리지 않고 링크를 통해 전송될 수 있는 패킷들의 수를 나타내는 잔여 용량(C_{left})

[0065] • 정의된 간격 내 잔여 용량(ΔC_{left})의 변경들.

[0066] 스케줄링 가중치들(SW)은 일반적으로 관계식들, 예를 들어, (A:B:C:D)로 표현된다. 4개의 사용가능한 링크들이 있는 다음 예제에서, 제1 링크는 A의 가중치를 가지며, 이는, A 패킷들이 다음 링크의 패킷들을 보내기 전에 이 링크를 통해 전송되고, B 패킷들이 제2 링크를 통해 전송되고, C 패킷들이 제3 링크를 통해 전송되며, 그리고 최종적으로 D 패킷들이 제4 링크를 통해 전송된다. 그 후, 제1 링크가 다시 스케줄링된다. 스케줄링 가중치들(SW)은 2개의 특성들을 갖는다.

[0067] • 링크들 간에 전송된 패킷들의 비율을 나타내는 가중치 비율(W_R). 예를 들어, 2개의 링크들의 경우, 비율 2는,

2배 더 많은 패킷들이 링크 2에 비해 링크 1을 통해 전송된다는 것을 의미한다.

- 로우의 한 링크를 통해 전송된 패킷들의 양을 나타내는 가중치(WH). 높이 10은, 로우 내의 10개의 패킷들이 링크 1을 통해 전송됨을 의미한다.

예: $W_R=2$, $W_H=10 \rightarrow SW=(10:5)$.

[0070] 성능상의 이유들로 인해, 스케줄링 가중치들의 적응들은 연속적으로 행해지지는 않지만, 도 3에 도시된 바와 같이 타임 슬롯들(예를 들어, 슬롯 n, 슬롯 n+1, 슬롯 n+2)에서 수행될 수 있다. 이를 타임 슬롯들의 지속 기간은 파라미터(D_T)이다. 타임 슬롯(X_n) 동안 수집된 파라미터들의 세트는 다음 타임 슬롯 X_{n+1} 에 대한 스케줄링 알고리즘($SW_{n+1}=f(X_n)$)에 대한 입력으로서 사용된다. 즉, 이러한 파라미터들(타임 슬롯 동안 수집된 파라미터들)과 함께 선택적인 입력 또는 구성 파라미터들에 기초하여, 스케줄링 알고리즘은 다음 슬롯에 대한 스케줄링 가중치들을 계산한다.

[0071] 본 발명에 따른 예시적인 스케줄링 알고리즘은 도 4에 도시된다. 여기서, 설명을 위해 2개의 터널들(터널 a 및 터널 b)이 사용될 수 있다. 스케줄링 알고리즘은 슬립 D_T , 즉, 슬롯 시간 D_T 를 기다리는 단계 S10에서 시작한다.

패킷이 전송되는 경우, 스케줄링 알고리즘은 단계(S11)에서 터널 a의 잔여 용량 $C_{left,a0}$ 미리결정된 임계값(THR) 이상인지 여부를 검사한다. 터널 a의 잔여 용량 $C_{left,a}$ 가 미리결정된 임계값 이상이면, 스케줄링 알고리즘은 단계 S12에서 터널 a를 통해 패킷을 전송할 것이고, 이 예에서는 SW(1:0)의 스케줄링 가중치를 갖고, 그리고 시작점으로 복귀하며(단계 S10), 그리고 절차가 다른 슬롯에 대해 반복된다. 여기서, SW(1:0)는 하나의 패킷이 터널 a를 통해 전송되고 그리고 터널 b를 통해 전송되는 패킷이 없음을 의미한다. 즉, 터널 a를 통한 패킷들만 전송되며, 이 예에서 W_R 은 무한이고 W_H 는 1이다. 하지만, 터널 a의 잔여 용량 $C_{left,a0}$ 미리결정된 임계값 이상이지 않으면, 스케줄링 알고리즘은 단계 S13에서 터널 b의 잔여 용량 $C_{left,b0}$ 미리결정된 임계값(THR) 이상이 아닌지를 결정할 것이다. 터널 a 및 터널 b에 대한 임계값들은 반드시 동일할 필요는 없지만 상이한 임계값들일 수 있음을 이해해야 한다.

[0072] 터널 b의 잔여 용량 $C_{left,b}$ 이 미리결정된 임계값 이상인 경우, 스케줄링 알고리즘은, 이 예에서 SW(0:1)의 스케줄링 가중치를 갖는 단계 S14에서 터널 b를 통해 패킷을 송신하고, 시작점으로 복귀하며(단계 S10), 그리고 절차가 다른 슬롯에 대해 반복될 것이다. 하지만, 터널 b의 잔여 용량 $C_{left,b0}$ 이 미리결정된 임계값 이상이 아니면, 스케줄링 알고리즘은 터널들 중 어느 것도 잔여 용량을 갖지 않는다고 결정할 것이고, 스케줄링 알고리즘은, 이 예에서 SW(1:1)의 스케줄링 가중치를 갖는 패킷을 단계 15에서 전송할 것이고, 단계 S10에서 시작점으로 복귀할 것이며, 그리고 절차가 다른 슬롯에 대해 반복된다.

[0073] 즉, 터널 a는 우선순위 터널이며, 사용 가능한 용량이 있는 경우 패킷은 이 터널을 통해서만 전송된다. 터널 a에 사용 가능한 용량이 없는 경우, 패킷들은, 터널 b에 사용 가능한 용량이 있으면 터널 b를 통해 전송된다. 두 링크들 모두에 사용 가능한 용량이 없는 경우 라운드 로빈 스케줄링이 수행된다.

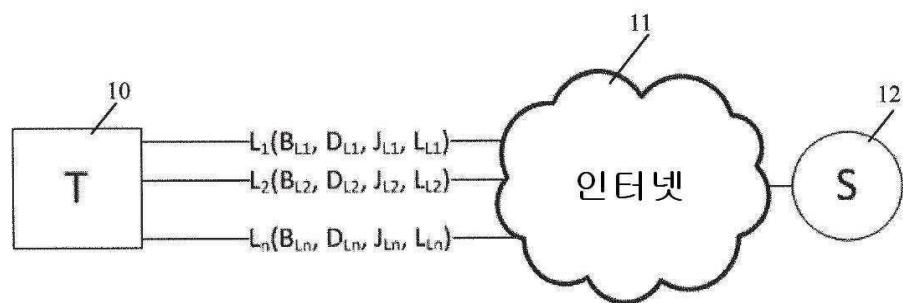
[0074] 비록 네트워크 계층 접근법의 맥락에서 기술된다고 할지라도, 통상의 기술자는 본 발명이 다른 벤들링 접근법들, 예를 들어, MPTCP에 또한 적용될 수 있음을 이해할 것이다. 하지만, 서술된 네트워크 계층 접근법(계층 3)을 사용하면, 새로운 TCP 프로토콜이 필요하지 않을 수 있다. 따라서, 표준 TCP 프로토콜들이 사용될 수 있다.

[0075] 본 발명은 그 범위 또는 본질적인 특성을 벗어나지 않고 몇몇 형태로 구체화 될 수 있으므로, 상술한 실시예들은 달리 명시하지 않는 한 상기의 상세한 설명 중 어느 것에 의해서도 제한되지 않고, 첨부된 청구 범위에서 정의된 범위 내에서 광범위하게 해석될 수 있으며, 따라서 본 발명에 속하는 모든 변경들 및 수정들은 첨부된 청구 범위에 포함되도록 의도된다.

[0076] 더욱이, 청구 범위에서 "포함하는"이라는 단어는 다른 요소 또는 단계를 배제하지 않으며, 하나의 의미를 갖는 단어는 복수를 배제하지 않는다. 단일 유닛은 청구 범위에 언급된 여러 피처들의 기능들을 수행할 수 있다. 속성 또는 값과 관련하여 "본질적으로", "약", "대략" 등의 용어는 또한 특히 속성 또는 값을 각각 정확하게 정의한다. 청구 범위 내의 임의의 참조 부호들은 범위를 제한하는 것으로 해석되어서는 안된다.

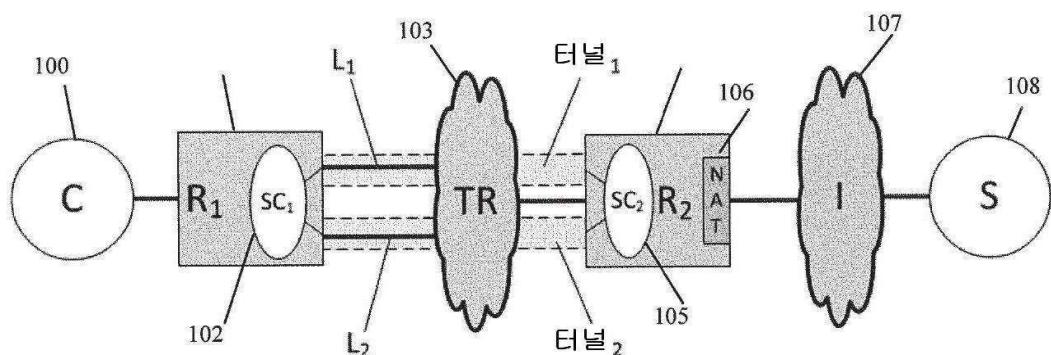
도면

도면1

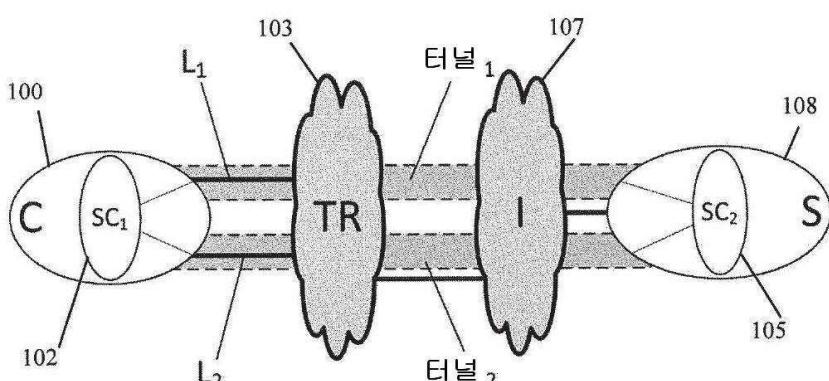


(종래 기술)

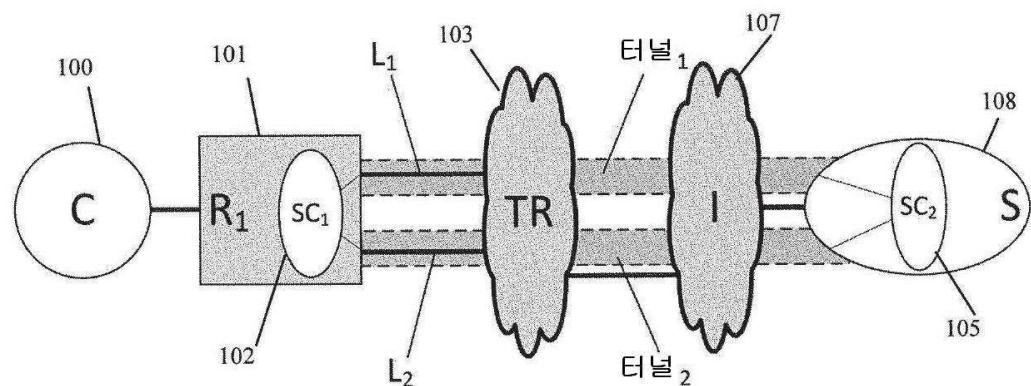
도면2a



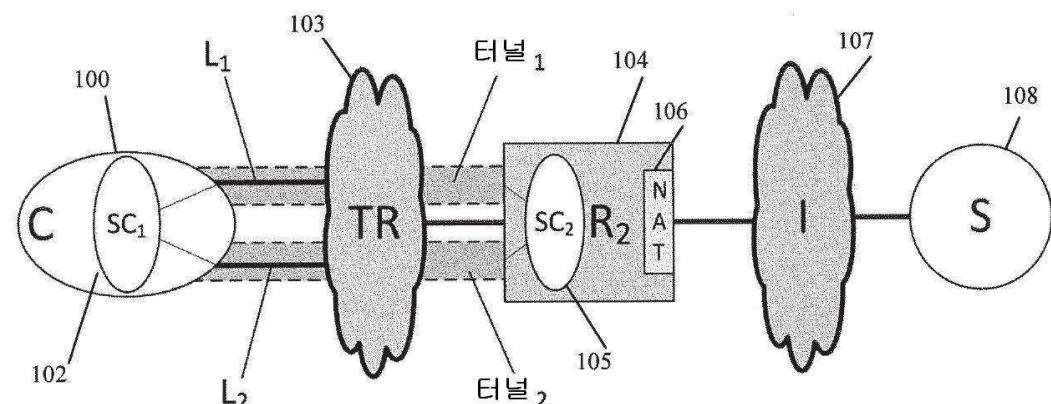
도면2b



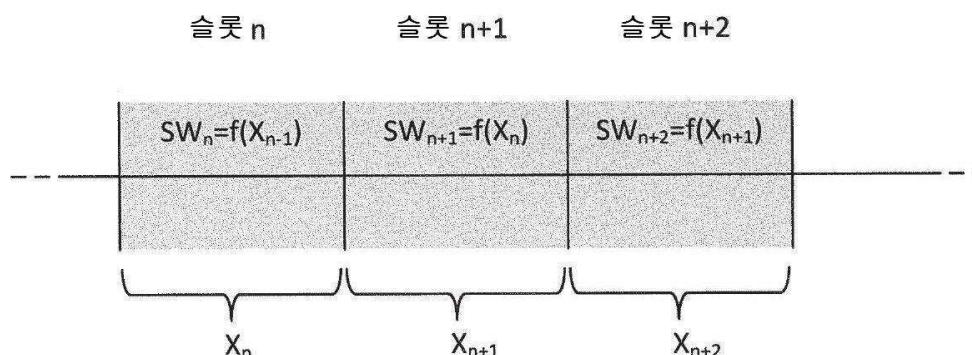
도면2c



도면2d



도면3



도면4

