(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년11월08일
(11) 등록번호 10-2033402
(24) 등록일자 2019년10월11일

(51) 국제특허분류(Int. Cl.)
H04L 12/851 (2013.01) *H04L 12/26* (2006.01)
(52) CPC특허분류
H04L 47/24 (2013.01)
H04L 12/4633 (2013.01)
(21) 출원번호 10-2017-0065312
(22) 출원일자 2017년05월26일
심사청구일자 2017년05월26일
(65) 공개번호 10-2018-0129376
(43) 공개일자 2018년12월05일
(56) 선행기술조사문헌
KR101686850 B1*
US20150143505 A1*
US20170111250 A1*
'다중큐잉 실시간 트래픽쉐이핑을 이용한 네트워크간 VPN 지원 유무선공유기 시스템', 한국정보통신학회논문지, Vol.19, No.5, pp1097-1103, 2015.05.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
배재대학교 산학협력단
대전광역시 서구 배재로 155-40 (도마동)
(72) 발명자
양승의
대전광역시 유성구 엑스포로 448, 305동 303호(전민동, 엑스포아파트)
정희경
대전광역시 서구 둔산로 155, 112동 1303호(둔산동, 크로바아파트)
(74) 대리인
유병욱, 한승범

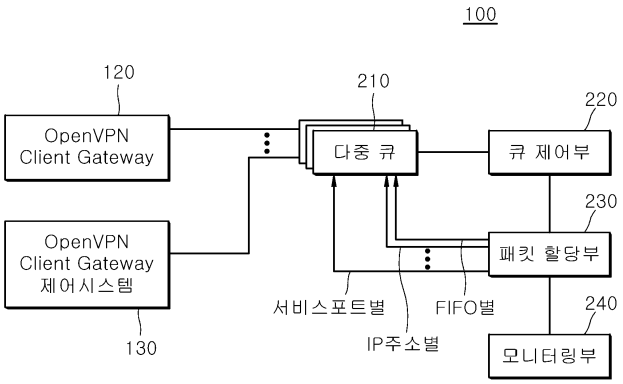
전체 청구항 수 : 총 15 항 심사관 : 김대성

(54) 발명의 명칭 **사물인터넷 지원 스마트 게이트웨이 및 그것의 VPN 터널링 실시간 속도 제어 방법**

(57) 요 약

본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이는 원격지의 네트워크를 가상 사설망(VPN)을 통해 연결하기 위한 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 잠시 버퍼링하였다가 사전에 정의된 우선순위에 따라 출력하는 다중 큐; 상기 VPN 터널링 구간에 걸리는 트래픽 부하 및 상기 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석하는 모니터링부; 및 상기 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐에 적응적으로 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 패킷 할당부를 포함한다.

대 표 도



(52) CPC특허분류
H04L 12/4641 (2013.01)
H04L 43/08 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1425102232
부처명 중소기업청
연구관리전문기관 중소기업기술정보진흥원
연구사업명 산학연협력기술개발사업(도약기술개발사업)
연구과제명 OpenWRT기반 사물인터넷(IoT)지원 스마트 게이트웨이 개발
기 여 율 1/1
주관기관 배재대학교 산학협력단
연구기간 2016.05.01 ~ 2017.04.30

명 세 서

청구범위

청구항 1

OpenVPN Client Gateway 및 상기 OpenVPN Client Gateway의 동작을 제어하는 OpenVPN Client Gateway 제어 시스템을 포함하는 사물인터넷 지원 스마트 게이트웨이에 있어서,

원격지의 네트워크를 가상 사설망(VPN)을 통해 연결하기 위한 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 잠시 버퍼링하였다가 사전에 정의된 우선순위에 따라 출력하는 다중 큐;

상기 VPN 터널링 구간에 걸리는 트래픽 부하 및 상기 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석하는 모니터링부; 및

상기 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐에 적응적으로 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 패킷 할당부

를 포함하고,

상기 OpenVPN Client Gateway 제어 시스템은

상기 OpenVPN Client Gateway가 유무선 공유기로 구현되는 경우 응용 프로그램 형태로 상기 유무선 공유기에 탑재되어 일체로 형성되고, 상기 VPN 터널링 구간에서 다중 큐잉을 통해 패킷을 전송할 때 네트워크 상태에 따라 적응적으로 패킷을 할당하여 트래픽 쉐이핑을 수행하며,

상기 모니터링부는

하기 수학식 1에 기초하여, 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 일정시간 간격으로 계산하고, 상기 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하며, 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

[수학식 1]

$$user\_util = 100 * (utime\_after - utime\_before) / (time\_total\_after - time\_total\_before);$$
$$sys\_util = 100 * (stime\_after - stime\_before) / (time\_total\_after - time\_total\_before);$$

여기서, utime은 유저 모드 지피스(user mode jiffies), utime_before 및 utime_after는 유저 모드 지피스가 발생하는 전/후 시간, user_util은 시스템 전체에 대한 CPU 사용 시간이고, stime은 커널 모드 지피스(kernel mode jiffies), stime_before 및 stime_after는 커널 모드 지피스가 발생하는 전/후 시간, sys_util은 VPN 터널링 데몬의 사용 시간임.

청구항 2

제1항에 있어서,

상기 모니터링부는

상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 상기 각 큐잉별 트래픽 부하량을 계산하고, 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 3

제2항에 있어서,

상기 패킷 할당부는

상기 각 큐잉별 트래픽 부하량에 기초하여 다중 큐잉 중 부하가 상대적으로 적은 큐잉을 찾아서 밴드폭을 줄이고, 상대적으로 부하가 큰 다른 큐잉에 밴드폭을 더 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 4

삭제

청구항 5

삭제

청구항 6

제1항에 있어서,

상기 패킷 할당부는

상기 CPU의 부하량에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 7

제3항 또는 제6항에 있어서,

상기 모니터링부는

VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고, 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 8

제7항에 있어서,

상기 모니터링부는

하기 수학식 2에 기초하여, 상기 VPN 터널링 디바이스의 패킷 정보를 일정시간 간격으로 계산하여 상기 전체 패킷 드롭율을 계산하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

[수학식 2]

$$rx\_drop\_rate=100*(rx\_drop\_after - rx\_drop\_before)/(rx\_pkt\_tot\_after - rx\_pkt\_tot\_before);$$
$$tx\_drop\_rate=100*(tx\_drop\_after - tx\_drop\_before)/(tx\_pkt\_tot\_after - tx\_pkt\_tot\_before);$$

여기서, rx_drop은 상기 VPN 터널링 디바이스의 송신 측에서 드롭된 패킷 수, tx_drop은 상기 VPN 터널링 디바이스의 수신 측에서 드롭된 패킷 수, rx_pkt_tot는 상기 VPN 터널링 디바이스의 송신 측 패킷 수, tx_pkt_tot는 상기 VPN 터널링 디바이스의 수신 측 패킷 수임.

청구항 9

제7항에 있어서,

상기 패킷 할당부는

상기 전체 패킷 드롭율에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 10

제1항에 있어서,

상기 패킷 할당부는

상기 패킷을 서비스 포트별, IP 주소별로 구분하고, 상기 구분된 패킷을 상기 네트워크 상태의 분석 결과에 기초하여 상기 다중 큐에 적응적으로 할당하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 11

제1항에 있어서,

상기 다중 큐의 개수, 우선순위 및 대역폭 중 적어도 하나를 조절 또는 변경하는 큐 제어부

를 더 포함하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 12

제1항에 있어서,

상기 다중 큐는

ToS(Type of Service)와 우선순위에 따라 패킷을 출력하는 SFQ(Stochastic Fair Queuing) 로직을 갖는 큐; 및

대역폭의 조절을 통해 패킷의 유효 속도 또는 최대 속도를 제한하는 HTB(Hierarchical Token Bucket) 로직을 갖는 큐

를 포함하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 13

제12항에 있어서,

상기 다중 큐는

상기 SFQ 로직 또는 상기 HTB 로직을 기반으로 상기 패킷을 다중 큐잉하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이.

청구항 14

OpenVPN Client Gateway 및 상기 OpenVPN Client Gateway의 동작을 제어하는 OpenVPN Client Gateway 제어 시스템을 포함하는 사물인터넷 지원 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법에 있어서,

원격지의 네트워크를 가상 사설망(VPN)을 통해 연결하기 위한 VPN 터널링 구간에 걸리는 트래픽 부하, 및 상기 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 잠시 버퍼링하였다가 사전에 정의된 우선순위에 따라 출력하는 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석하는 단계; 및

상기 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐에 적응적으로 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 단계

를 포함하고,

상기 OpenVPN Client Gateway 제어 시스템은

상기 OpenVPN Client Gateway가 유무선 공유기로 구현되는 경우 응용 프로그램 형태로 상기 유무선 공유기에 탑재되어 일체로 형성되고, 상기 VPN 터널링 구간에서 다중 큐잉을 통해 패킷을 전송할 때 네트워크 상태에 따라 적응적으로 패킷을 할당하여 트래픽 쉐이핑을 수행하며,

상기 네트워크 상태를 모니터링하고 분석하는 단계는

하기 수학식 1에 기초하여, 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 일정시간 간격으로 계산하는 단계;

상기 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하는 단계; 및

상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계

를 포함하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법.

[수학식 1]

```
user_util = 100 * (utime_after - utime_before) / (time_total_after - time_total_before);
sys_util = 100 * (stime_after - stime_before) /  (time_total_after - time_total_before);
```

여기서, utime은 유저 모드 지피스(user mode jiffies), utime_before 및 utime_after는 유저 모드 지피스가 발생하는 전/후 시간, user_util은 시스템 전체에 대한 CPU 사용 시간이고, stime은 커널 모드 지피스(kernel mode jiffies), stime_before 및 stime_after는 커널 모드 지피스가 발생하는 전/후 시간, sys_util은 VPN 터널링 데몬의 사용 시간임.

청구항 15

제14항에 있어서,

상기 네트워크 상태를 모니터링하고 분석하는 단계는

상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 상기 각 큐잉별 트래픽 부하량을 계산하는 단계; 및

상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계

를 포함하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법.

청구항 16

제15항에 있어서,

상기 VPN 터널링 구간의 트래픽을 제어하는 단계는

상기 각 큐잉별 트래픽 부하량에 기초하여 다중 큐잉 중 부하가 상대적으로 적은 큐잉을 찾아서 밴드폭을 줄이는 단계; 및

상대적으로 부하가 큰 다른 큐잉에 밴드폭을 더 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 단계

를 포함하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법.

청구항 17

삭제

청구항 18

제16항에 있어서,

상기 네트워크 상태를 모니터링하고 분석하는 단계는

VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하는 단계; 및

상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계

를 포함하는 것을 특징으로 하는 사물인터넷 지원 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시예들은 사물인터넷 지원 스마트 게이트웨이 및 사물인터넷 지원 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법에 관한 것이다.

배 경 기 술

[0003] 최근 클라우드, 스마트 디바이스, 사물인터넷 등 정보통신 기반의 다양한 서비스 기술이 급속히 발전하고 있으며, 여기에 필요한 네트워크 장비인 라우터, 스마트 게이트웨이, 스위치, 방화벽 등 플랫폼 기술도 더불어 발전하고 있다. 이러한 정보통신 기술의 발전은 스마트 디바이스의 활용이 시간과 공간의 제약을 극복하고, 사무실이나 집, 학교, 외부 등 어느 곳에서도 자신이 하던 일을 계속 이어서 할 수 있고, 물리적으로 떨어진 원격지 환경과 동일한 네트워크 환경을 제공하는 것이 가능하게 되었다.

[0004] 이렇게 다수의 원격지 네트워크를 하나의 로컬 네트워크로 구성할 수 있는 기술은 이미 잘 알려진 네트워크간 VPN(Virtual Private Network)장비로 구현이 가능하다. 상용 VPN장비의 경우 발전을 거듭하여 기본적인 라우터 기능, 터널링(Tunneling) 기능 외에 IDS(Intrusion Detection System), IPS(Intrusion Protection System) 등 지능형 방화벽, QoS(Quality of Service) 기술, 다중회선 지원, 로드밸런싱(Load Balancing)등 다양한 고급 기능을 적용할 수 있도록 제공하고 있다.

[0005] 그런데, 상용 VPN 장비의 경우 고성능의 제품으로 구축된 전용선 환경에서 전문가의 지원으로 활용되고 있으며, 고가의 전용 장비에 의해 기술의 특성상 기기간 호환성도 보장하기 힘든 문제를 가지고 있다. 또한, VPN 터널링 기술은 높은 도입 비용과 고가의 안정적인 전용회선이 필요하기 때문에 일반적인 가정이나 기업에서 쓰는 인터넷 환경에서는 운영하기 어려운 문제가 있다.

[0006] 이를 보안하고자 일반 인터넷 환경에 네트워크간 VPN을 지원하는 유무선 공유기가 개발되었으나, 대부분 단순한 VPN 서버의 기능만 제공하여 네트워크간 VPN 연동은 가능하지 않고 PC나 스마트폰에서 직접 연동하는 방법만 제공하기 때문에 그 활용성이 떨어지고, 지원하는 기기가 국한되어 있다. 또한 VPN 터널링의 속도가 5Mbps 수준에 불과하고, 상용 VPN 전용 장비의 경우 대중화 시키기에는 경제성이 떨어지며 전용선 등 회선 상황이 보장이 되어야 운영 이 가능한 문제를 가지고 있다.

[0007] 관련 선행기술로는 한국공개특허 제10-2004-0039909호(발명의 명칭: 전송계층 터널링을 이용한 가상사설망에서의 통신품질향상방법, 공개일자: 2004.05.12.)가 있다.

발명의 내용

해결하려는 과제

[0009] 본 발명의 일 실시예는 VPN 터널링의 실시간 트래픽 쉐이핑 알고리즘에 CPU 부하, 패킷 드롭(drop)율 그리고 큐

잉 우선순위를 적용하여 구현함으로써 저품질의 인터넷 상황에서도 안정적이고 원활한 사물인터넷 연동을 지원할 수 있는 스마트 게이트웨이 및 그것의 VPN 터널링 실시간 속도 제어 방법을 제공한다.

[0011] 본 발명이 해결하고자 하는 과제는 이상에서 언급한 과제(들)로 제한되지 않으며, 언급되지 않은 또 다른 과제(들)은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0013] 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이는 원격지의 네트워크를 가상 사설망(VPN)을 통해 연결하기 위한 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 잠시 버퍼링하였다가 사전에 정의된 우선순위에 따라 출력하는 다중 큐; 상기 VPN 터널링 구간에 걸리는 트래픽 부하 및 상기 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석하는 모니터링부; 및 상기 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐에 적응적으로 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 패킷 할당부를 포함한다.

[0014] 상기 모니터링부는 상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 상기 각 큐잉별 트래픽 부하량을 계산하고, 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0015] 상기 패킷 할당부는 상기 각 큐잉별 트래픽 부하량에 기초하여 다중 큐잉 중 부하가 상대적으로 적은 큐잉을 찾아서 밴드폭을 줄이고, 상대적으로 부하가 큰 다른 큐잉에 밴드폭을 더 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0016] 상기 모니터링부는 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하고, 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0017] 상기 모니터링부는 하기 수학식 1에 기초하여, 상기 시스템 전체에 대한 CPU 사용 시간과 상기 VPN 터널링 데몬의 사용 시간을 일정시간 간격으로 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산할 수 있다.

[0018] [수학식 1]

[0019]
```
user_util = 100 * (utime_after - utime_before) / (time_total_after - time_total_before);
sys_util = 100 * (stime_after - stime_before) /  (time_total_after - time_total_before);
```

[0020] 여기서, utime은 유저 모드 지피스(user mode jiffies), utime_before 및 utime_after는 유저 모드 지피스가 발생하는 전/후 시간, user_util은 시스템 전체에 대한 CPU 사용 시간이고, stime은 커널 모드 지피스(kernel mode jiffies), stime_before 및 stime_after는 커널 모드 지피스가 발생하는 전/후 시간, sys_util은 VPN 터널링 데몬의 사용 시간임.

[0021] 상기 패킷 할당부는 상기 CPU의 부하량에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0022] 상기 모니터링부는 VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고, 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0023] 상기 모니터링부는 하기 수학식 2에 기초하여, 상기 VPN 터널링 디바이스의 패킷 정보를 일정시간 간격으로 계산하여 상기 전체 패킷 드롭율을 계산할 수 있다.

[0024] [수학식 2]

[0025]
```
rx_drop_rate=100*(rx_drop_after - rx_drop_before)/(rx_pkt_tot_after - rx_pkt_tot_before);
tx_drop_rate=100*(tx_drop_after - tx_drop_before)/(tx_pkt_tot_after - tx_pkt_tot_before);
```

[0026] 여기서, rx_drop은 상기 VPN 터널링 디바이스의 송신 측에서 드롭된 패킷 수, tx_drop은 상기 VPN 터널링 디바이스의 수신 측에서 드롭된 패킷 수, rx_pkt_tot는 상기 VPN 터널링 디바이스의 송신 측 패킷 수, tx_pkt_tot는

상기 VPN 터널링 디바이스의 수신 측 패킷 수임.

[0027] 상기 패킷 할당부는 상기 전체 패킷 드롭율에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0028] 상기 패킷 할당부는 상기 패킷을 서비스 포트별, IP 주소별로 구분하고, 상기 구분된 패킷을 상기 네트워크 상태의 분석 결과에 기초하여 상기 다중 큐에 적응적으로 할당할 수 있다.

[0029] 본 발명의 일 실시예에 따른 사물인터넷을 지원하는 스마트 게이트웨이는 상기 다중 큐의 개수, 우선순위 및 대역폭 중 적어도 하나를 조절 또는 변경하는 큐 제어부를 더 포함할 수 있다.

[0030] 상기 다중 큐는 ToS(Type of Service)와 우선순위에 따라 패킷을 출력하는 SFQ(Stochastic Fair Queuing) 로직을 갖는 큐; 및 대역폭의 조절을 통해 패킷의 유효 속도 또는 최대 속도를 제한하는 HTB(Hierarchical Token Bucket) 로직을 갖는 큐를 포함할 수 있다.

[0031] 상기 다중 큐는 상기 SFQ 로직 또는 상기 HTB 로직을 기반으로 상기 패킷을 다중 큐잉할 수 있다.

[0032] 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법은 원격지의 네트워크를 가상 사설망(VPN)을 통해 연결하기 위한 VPN 터널링 구간에 걸리는 트래픽 부하, 및 상기 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 잠시 버퍼링하였다가 사전에 정의된 우선순위에 따라 출력하는 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석하는 단계; 및 상기 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐에 적응적으로 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 단계를 포함한다.

[0033] 상기 네트워크 상태를 모니터링하고 분석하는 단계는 상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 상기 각 큐잉별 트래픽 부하량을 계산하는 단계; 및 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계를 포함할 수 있다.

[0034] 상기 VPN 터널링 구간의 트래픽을 제어하는 단계는 상기 각 큐잉별 트래픽 부하량에 기초하여 다중 큐잉 중 부하가 상대적으로 적은 큐잉을 찾아서 밴드폭을 줄이는 단계; 및 상대적으로 부하가 큰 다른 큐잉에 밴드폭을 더 할당하여 상기 VPN 터널링 구간의 트래픽을 제어하는 단계를 포함할 수 있다.

[0035] 상기 네트워크 상태를 모니터링하고 분석하는 단계는 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하는 단계; 및 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계를 포함할 수 있다.

[0036] 상기 네트워크 상태를 모니터링하고 분석하는 단계는 VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하는 단계; 및 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계를 포함할 수 있다.

[0038] 기타 실시예들의 구체적인 사항들은 상세한 설명 및 첨부 도면들에 포함되어 있다.

발명의 효과

[0040] 본 발명의 일 실시예에 따르면, VPN 터널링의 실시간 트래픽 쉐이핑 알고리즘에 CPU 부하, 패킷 드롭(drop)율 그리고 큐잉 우선순위를 적용하여 구현함으로써 저품질의 인터넷 상황에서도 안정적이고 원활한 사물인터넷 연동을 지원할 수 있다.

[0041] 본 발명의 일 실시예에 따르면, 실시간에 다중 큐잉의 내부 트래픽을 세부적으로 계산하여 다중 큐잉 밴드폭을 효율적으로 실시간에 최적의 상태로 변화시켜 줄 수 있다. 즉, 미리 설계된 다중 큐잉 시나리오 없이 최적의 상태로 운영이 가능하다.

[0042] 본 발명의 일 실시예에 따르면, 미리 정해진 큐잉 알고리즘을 단순히 적용하여 발생하는 속도 제어 오류를 방지하고 실시간 해당되는 큐잉 및 관련 큐잉에 대해서 세부적으로 제어할 수 있다.

도면의 간단한 설명

[0044] 도 1은 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이의 네트워크 구성을 도시한 도면이다.

도 2는 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이를 설명하기 위해 도시한 블록도이다.

도 3은 원격 제어 모니터링의 일례를 도시한 도면이다.

도 4는 VPN 터널에 대한 다중 큐잉을 통한 트래픽 쉐이핑의 일례를 도시한 도면이다.

도 5는 본 발명의 일 실시예에 따른 트래픽 쉐이핑을 적용한 시스템과 기존 시스템 간 성능 분석 결과를 나타낸 비교표이다.

도 6 내지 도 9는 본 발명의 일 실시예에 따른 사물인터넷을 지원하는 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법을 설명하기 위해 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0045] 본 발명의 이점 및/또는 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 것이며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성요소를 지칭한다.

[0046] 또한, 이하 실시되는 본 발명의 바람직한 실시예는 본 발명을 이루는 기술적 구성요소를 효율적으로 설명하기 위해 각각의 시스템 기능구성에 기 구비되어 있거나, 또는 본 발명이 속하는 기술분야에서 통상적으로 구비되는 시스템 기능 구성은 가능한 생략하고, 본 발명을 위해 추가적으로 구비되어야 하는 기능 구성을 위주로 설명한다. 만약 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면, 하기에 도시하지 않고 생략된 기능 구성 중에서 종래에 기 사용되고 있는 구성요소의 기능을 용이하게 이해할 수 있을 것이며, 또한 상기와 같이 생략된 구성 요소와 본 발명을 위해 추가된 구성 요소 사이의 관계도 명백하게 이해할 수 있을 것이다.

[0047] 또한, 이하의 설명에 있어서, 신호 또는 정보의 "전송", "통신", "송신", "수신" 기타 이와 유사한 의미의 용어는 일 구성요소에서 다른 구성요소로 신호 또는 정보가 직접 전달되는 것뿐만이 아니라 다른 구성요소를 거쳐 전달되는 것도 포함한다. 특히 신호 또는 정보를 일 구성요소로 "전송" 또는 "송신"한다는 것은 그 신호 또는 정보의 최종 목적지를 지시하는 것이고 직접적인 목적지를 의미하는 것이 아니다. 이는 신호 또는 정보의 "수신"에 있어서도 동일하다.

[0049] 이하에서는 첨부된 도면을 참조하여 본 발명의 실시예들을 상세히 설명하기로 한다.

[0050] 도 1은 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이의 네트워크 구성을 도시한 도면이다.

[0051] 도 1을 참조하면, 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이(100)는 물리적으로 떨어져 있는 원격지의 네트워크(사무실, 원격지)를 가상의 네트워크를 이용하여 마치 하나의 네트워크처럼 보이게 구성하는 VPN 시스템에 구현하여 VPN 터널링 시 최적의 터널링 속도를 유지하여 우수한 인터넷 회선 품질을 제공할 수 있다.

[0052] 여기서, 상기 VPN 시스템은 두 개의 사설 네트워크(사무실, 원격지) 사이에 게이트웨이(110, 120)를 통해 연결될 수 있다. 두 개의 사설 네트워크(사무실, 원격지)는 인터넷을 두고 서로 떨어져 있고, 외부 접근이 어렵기 때문에 서로 트래픽을 전달할 수 없다.

[0053] 하지만, 사설 네트워크(사무실, 원격지)에는 인터넷으로 트래픽을 보내기 위한 게이트웨이(110, 120)가 연결될 수 있고, 각 게이트웨이(110, 120)들은 공인 주소(public ip)를 가지므로 두 개의 사설 네트워크(사무실, 원격지)가 게이트웨이(110, 120)를 통해 연결될 수 있다.

[0054] 이때, 상기 게이트웨이(110, 120)간 터널링(tunneling) 구간에서는 VPN을 통해 브릿지(bridge) 또는 라우팅(routing) 연결이 될 수 있다. 참고로, 상기 게이트웨이(110)는 오픈VPN 서버 액세스 게이트웨이(OpenVPN Server Access Gateway)이고, 상기 게이트웨이(120)는 오픈VPN 클라이언트 게이트웨이(OpenVPN Client Gateway)일 수 있다.

[0055] 상기 브릿지 방식으로 연결되는 경우, 상기 VPN 시스템은 두 개의 네트워크 간 라우팅이 필요없고, 주소 체계가 동일하며, 동일 네트워크처럼 동작하기 때문에 이더넷 패킷까지 교환되고 모든 장비에 접근이 가능하다. 따라서, DLNA(Digital Living Network Alliance), 인트라넷 등 동일 네트워크에서 동작하는 프로그램을 그대로 쓸

수 있다.

[0056] 상기 라우팅 방식으로 연결되는 경우, 상기 VPN 시스템은 모든 장비의 주소 체계가 다르고 다른 네트워크처럼 동작하기 때문에 명시적으로 라우팅 설정을 해서 사용할 수 있다.

[0057] 상기 VPN 시스템은 OpenVPN Client Gateway 제어 시스템(130)를 통해 상기 OpenVPN Client Gateway(120)의 동작을 제어할 수 있다. 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이(100)는 상기 OpenVPN Client Gateway(120) 및 상기 OpenVPN Client Gateway 제어 시스템(130)을 포함하여 구현될 수 있다.

[0058] 여기서, 상기 OpenVPN Client Gateway 제어 시스템(130)은 상기 OpenVPN Client Gateway(120)를 포함하여 구현될 수도 있고, 상기 OpenVPN Client Gateway(120)와는 별개로 구현될 수도 있다. 상기 OpenVPN Client Gateway(120)가 유무선 공유기로 구현되는 경우, 상기 OpenVPN Client Gateway 제어 시스템(130)은 응용 프로그램 형태로 유무선 공유기에 탑재되어 일체로 형성될 수도 있다.

[0059] 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이(100)는 VPN 터널링 구간을 다중 큐잉으로 구현하고, 상기 다중 큐잉을 통해 패킷을 전송할 때 상기 OpenVPN Client Gateway 제어 시스템(130)이 네트워크 상태에 따라 적응적으로 패킷을 할당하여 트래픽 쉐이핑(Traffic Shaping)을 수행할 수 있다.

[0060] 예를 들면, 본 발명의 일 실시예에 따르면 상기 VPN 터널링 구간에서 상기 다중 큐잉을 통해 DLNA(Digital Living Network Alliance) 동영상을 스트리밍(streaming)하면서, 이와 동시에 ftp(File Transfer Protocol) 송수신과 ssh(Secure Shell) 접속을 원활하게 함께 할 수 있다.

[0061] 참고로, 상기 트래픽 쉐이핑은 트래픽 흐름을 보다 매끄럽게 제한/정형화시키는 방법을 말하는 것으로, 네트워크 망 내로 유입 또는 유출되는 트래픽의 양 및 속도를 조절하여 트래픽 폭주를 방지하는 트래픽 제어 관리 기술이다.

[0062] 상기 OpenVPN Client Gateway 제어 시스템(130)은 도면에서와 같이 클라이언트 측의 OpenVPN Client Gateway(120)를 제어하는 것으로만 도시되어 있으나, 이에 한정하지 않고 서버 측의 OpenVPN Server Access Gateway(110) 측에도 구현될 수 있으며, 인바운드(inbound), 아웃바운드(outbound)의 패킷 전송에 모두 적용할 수 있다.

[0063] 한편, 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이(100)는 원격지에서 안전하게 VPN을 이용하여 사무실의 네트워크에 동일한 네트워크처럼 가상화 접속하여 센서 네트워크 장비들에 양방향으로 접속이 가능하도록 한다.

[0064] 이렇게 구현된 VPN 터널링은 도 3과 같이 원격지의 PC나 스마트 폰 화면을 통해서 사무실의 센서 네트워크 장치 등을 웹으로 접속하여 제어하고 모니터링할 수 있다. 카메라를 보며 센서 네트워크 장치들이 마치 바로 옆에 있는 것처럼 안전하게 제어가 가능하다. 참고로, 도 3은 원격 제어 모니터링의 일례를 도시한 도면이다.

[0066] 도 2는 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이를 설명하기 위해 도시한 블록도이다.

[0067] 도 2를 참조하면, 본 발명의 일 실시예에 따른 사물인터넷 지원 스마트 게이트웨이(100)는 다중 큐(210), 큐 제어부(220), 패킷 할당부(230), 및 모니터링부(240)를 포함할 수 있다.

[0068] 상기 다중 큐(210)는 원격지의 네트워크를 가상 사설망(VPN)을 통해 연결하기 위한 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 잠시 버퍼링하였다가 사전에 정의된 우선순위에 따라 출력할 수 있다. 다시 말해, 상기 다중 큐(210)는 패킷을 잠시 버퍼링하였다가 사전에 정의된 트래픽 제어 룰에 따라 패킷을 출력하는 저장소 역할을 한다.

[0069] 상기 다중 큐(210)는 실제 패킷을 저장하는 메모리 공간이 아니라, 패킷이 대기하고 있는 메모리의 위치를 알려주는 포인터 정보를 연속적으로 가지고 있는 것으로 볼 수 있다. 큐잉(queuing)은 이러한 다중 큐(310)를 이용하여 아웃바운드 또는 인바운드 인터페이스로 내보내기 위해 대기하고 있는 패킷들의 포인터 주소를 기 정의된 스케줄링에 따라 나열하는 작업을 말한다.

[0070] 상기 다중 큐(210)는 ToS(Type of Service)와 우선순위에 따라 패킷을 출력하는 SFQ(Stochastic Fair Queuing) 로직을 갖는 큐, 및 대역폭의 조절을 통해 패킷의 유효 속도 또는 최대 속도를 제한하는 HTB(Hierarchical Token Bucket) 로직을 갖는 큐를 포함할 수 있다.

[0071] 상기 다중 큐(210)는 상기 SFQ 로직 또는 상기 HTB 로직을 기반으로 상기 VPN 터널링 구간에서 인바운드 또는

아웃바운드하는 패킷을 다중 큐잉(Multi-Queuing)할 수 있다.

[0072] 상기 큐 제어부(220)는 상기 다중 큐(210)의 개수, 우선순위 및 대역폭 중 적어도 하나를 조절 또는 변경할 수 있다. 이를 위해, 상기 큐 제어부(220)는 상기 SFQ 로직 및/또는 상기 HTB 로직을 이용하여 상기 다중 큐(210)를 제어할 수 있다.

[0073] 즉, 상기 큐 제어부(220)는 네트워크 상태 또는 시스템 자원의 점유율에 따라 상기 다중 큐(210)의 개수, 우선순위, 대역폭 등을 제어할 수 있다. 여기서, 상기 다중 큐(210)의 개수는 많을수록 패킷 전송률이 우수하고 패킷 드롭(drop)율이 낮은 장점이 있지만, 버퍼링의 딜레이가 커지고 이는 곧 시스템 부하가 높아지게 되는 단점도 있게 된다. 그러므로, 상기 큐 제어부(220)는 네트워크 상태 또는 시스템 자원의 점유율에 따라 최적의 다중 큐잉이 가능하도록 제어할 수 있다.

[0074] 상기 큐 제어부(220)는 서비스 포트별, IP 주소별로 상기 다중 큐(210)를 개별 큐로 구분할 수 있으며, 상기 구분된 개별 큐의 우선순위 또는 대역폭을 조절할 수 있다. 여기서, 상기 서비스 포트는 ftp, htb, ssh, telnet, www, tcmp 프로토콜을 통해 접속되는 포트를 포함할 수 있다.

[0075] 상기와 같은 큐 제어부(220)의 동작을 통해, 상기 다중 큐(210)에는 ToS, QoS에 따른 우선순위가 지정될 수 있지만, IP 주소별, 서비스 포트별로 배치시킨 각 큐에 대해서도 우선순위가 할당되고 각 큐의 대역폭도 네트워크 상태, 시스템 자원 점유율 등에 따라 설정 및 변경될 수 있다.

[0076] 상기 패킷 할당부(230)는 후술하는 모니터링부(240)에 의한 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐(210)에 적응적으로 할당할 수 있으며, 이를 통해 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0077] 또한, 상기 패킷 할당부(230)는 상기 패킷을 서비스 포트별, IP 주소별 등 세부적으로 구분하여 할당할 수 있다. 즉, 상기 패킷 할당부(230)는 상기 패킷을 서비스 포트별, IP 주소별로 구분하고, 상기 구분된 패킷을 상기 네트워크 상태의 분석 결과에 기초하여 상기 다중 큐에 적응적으로 할당할 수 있다.

[0078] 이때, 상기 VPN 터널링 구간에 걸리는 네트워크 상태는 모니터링부(340)를 통해 분석될 수 있으며, 상기 패킷 할당부(230)는 상기 모니터링부(240)를 통해 분석된 네트워크 상태에 따라 최적화된 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘에 따라 패킷을 할당할 수 있다.

[0079] 상기 패킷 할당부(230)는 상기 트래픽 쉐이핑 알고리즘의 선택을 패킷 할당이 필요할 때마다 수행할 수 있지만, 패킷 할당 시점과는 상관없이 상기 모니터링부(240)에서 상기 네트워크 상태를 분석하는 일정 주기마다 수행하고 주기 동안에는 상기 선택된 트래픽 쉐이핑 알고리즘으로 동작하도록 할 수 있다.

[0080] 상기 트래픽 쉐이핑 알고리즘은 다중 큐잉에 따라 다양한 쉐이핑이 가능하다. 상기 트래픽 쉐이핑 알고리즘은 IDC(Internet Data Center)에서 사용하는 수준의 인바운드, 아웃바운드 트래픽 제어는 물론, 각 서비스 포트별, IP 주소별, IP 네트워크별, 또는 이더 패킷(ether packet) 수준의 트래픽 제어까지 구현할 수 있다. 또한, 상기 트래픽 쉐이핑 알고리즘은 패킷 큐잉을 통해서 TAP(Test Access Port) 장비에서 사용하는 패킷 인스펙션까지 구현할 수 있기 때문에 점점 지능화 되어가고 있는 해킹 공격에 대해서도 트래픽 제어가 가능하다.

[0081] 상기 모니터링부(240)는 상기 VPN 터널링 구간에 걸리는 트래픽 부하 및 상기 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석할 수 있다.

[0082] 이를 위해, 일 실시예로서, 상기 모니터링부(240)는 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하고, 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0083] 이때, 상기 모니터링부(240)는 하기 수학식 1에 기초하여, 상기 시스템 전체에 대한 CPU 사용 시간과 상기 VPN 터널링 데몬의 사용 시간을 일정시간(1초~10초) 간격으로 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산할 수 있다.

[0084] [수학식 1]

[0085]
$$user\_util = 100 * (utime\_after - utime\_before) / (time\_total\_after - time\_total\_before);$$
$$sys\_util = 100 * (stime\_after - stime\_before) / (time\_total\_after - time\_total\_before);$$

[0086] 여기서, utime은 유저 모드 지피스(user mode jiffies), utime_before 및 utime_after는 유저 모드 지피스가 발생하는 전/후 시간, user_util은 시스템 전체에 대한 CPU 사용 시간이고, stime은 커널 모드 지피스(kernel mode jiffies), stime_before 및 stime_after는 커널 모드 지피스가 발생하는 전/후 시간, sys_util은 VPN 터널링 데몬의 사용 시간임.

[0087] 이에 따라, 상기 패킷 할당부(230)는 상기 CPU의 부하량에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0088] 다른 실시예로서, 상기 모니터링부(240)는 VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고, 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0089] 이때, 상기 모니터링부(240)는 하기 수학식 2에 기초하여, 상기 VPN 터널링 디바이스의 패킷 정보를 일정시간(1초~10초) 간격으로 계산하여 상기 전체 패킷 드롭율을 계산할 수 있다.

[0090] [수학식 2]

[0091]
$$rx\_drop\_rate = 100 * (rx\_drop\_after - rx\_drop\_before) / (rx\_pkt\_tot\_after - rx\_pkt\_tot\_before);$$
$$tx\_drop\_rate = 100 * (tx\_drop\_after - tx\_drop\_before) / (tx\_pkt\_tot\_after - tx\_pkt\_tot\_before);$$

[0092] 여기서, rx_drop은 상기 VPN 터널링 디바이스의 송신 측에서 드롭된 패킷 수, tx_drop은 상기 VPN 터널링 디바이스의 수신 측에서 드롭된 패킷 수, rx_pkt_tot는 상기 VPN 터널링 디바이스의 송신 측 패킷 수, tx_pkt_tot는 상기 VPN 터널링 디바이스의 수신 측 패킷 수임.

[0093] 이에 따라, 상기 패킷 할당부(230)는 상기 전체 패킷 드롭율에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0094] 또 다른 실시예로서, 상기 모니터링부(240)는 상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 상기 각 큐잉별 트래픽 부하량을 계산하고, 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0095] 이에 따라, 상기 패킷 할당부(230)는 상기 각 큐잉별 트래픽 부하량에 기초하여 다중 큐잉 중 부하가 상대적으로 적은 큐잉을 찾아서 밴드폭을 줄이고, 상대적으로 부하가 큰 다른 큐잉에 밴드폭을 더 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0096] 즉, 상기 패킷 할당부(230)는 상기 모니터링부(240)를 통해 다중 큐잉에서 큐잉별로 패킷량, drop량이 계산되면, 그 계산 결과를 토대로 하여 트래픽 유발 큐잉을 찾아 해당 트래픽에 대하여 우선순위, 속도를 최적화할 수 있도록 제어할 수 있다.

[0097] 앞서 수학식 1, 2를 통해 설명한 CPU 부하량과 패킷 drop율은 리눅스 커널에서 실시간 시스템 정보를 가지고 계산하는 것이다. 즉, 추가 부하 없이 네트워크 트래픽 부하를 계산해내는 것이다.

[0098] 여기서, 패킷 drop율은 전체 모든 종류의 패킷에 대한 drop을 보는 것이다. 이때 문제점은 네트워크 패킷이 과부하 상태라 drop 하기는 하는데 어떤 서비스 때문에 drop을 하는지 모르는데 있다.

[0099] 이를 해결하고자 네트워크 트래픽을 세부적인 서비스별로 각각 다중 큐잉을 하여 각 큐잉별, 즉 서비스별로 패킷의 drop을 파악하여 어떤 서비스가 지정된 큐잉 폭보다 더 부하가 큰가를 파악하는 것이다. 즉, 큐잉별 패킷 drop을 계산하고 할당된 큐잉 밴드폭을 계산하면 각 서비스별로 세부적으로 부하량을 계산할 수 있다.

[0100] 이러한 계산은 큐잉별 실시간에 가능하고 이를 바탕으로 여러 큐잉 중 부하가 적은 큐잉을 찾아서 밴드폭을 줄이고, 이렇게 줄인 만큼 부하가 큰 큐잉에 밴드폭을 더 할당해 줄 수 있도록 하는 방식이다.

[0101] 이 방법은 전체 패킷 drop율만 가지고 네트워크 부하를 측정하는 경우 추가로 부하가 걸리지 않는 것과 동일한 장점을 그대로 가지고 있고, 여기에 어떤 종류의 서비스 패킷이 원인인가까지 찾아낼 수 있는 방법이다.

[0102] 따라서, 상기의 방법에 의하면 실시간에 다중 큐잉의 내부 트래픽을 세부적으로 계산하여 다중 큐잉 밴드폭을

효율적으로 실시간에 최적의 상태로 변화시켜 줄 수 있다. 즉, 미리 설계된 다중 큐잉 시나리오 없이 최적의 상태로 운영이 가능하다. 다시 말해, 미리 정해진 큐잉 알고리즘을 단순히 적용하여 발생하는 속도 제어 오류를 방지하고 실시간 해당되는 큐잉 및 관련 큐잉에 대해서 세부적으로 제어할 수 있다.

[0104] 도 4는 VPN 터널에 대한 다중 큐잉을 통한 트래픽 쉐이핑의 일례를 도시한 도면이다.

[0105] ToS, QoS를 지원하는 대부분의 유무선 게이트웨이는 단일 큐잉에서 동작한다. ToS를 기준으로 QoS를 제어해 주기 때문에 어느 정도 성능 향상은 기대할 수 있지만, 동일 ToS의 서비스 종류가 여러 개 있을 경우에는 문제가 발생한다. 이를 해결할 수 있는 방법이 다중 큐잉 지원 커널이다.

[0106] 다중 큐잉을 지원하기 위해서는 OS의 네트워크 커널에서 CBQ, SFQ 등을 지원하도록 커널 수정을 해야 하기 때문에 안정성 및 성능을 보장하기 위해서는 적절한 튜닝기술을 적용하는 것이 바람직하다.

[0107] 본 발명의 일 실시예에서는 센서 제어 및 VPN 접속에 최적화된 다중 큐잉과 터널링을 지원하도록 커널을 수정하고, 프로그램을 구현할 수 있다. 또한, 서로 다른 특성의 SFQ, HTB 큐를 구성하고 각 큐에 서비스 포트별, IP 주소별로 할당하여 동작하도록 할 수 있다.

[0108] 특히, ToS, QoS에 따른 우선순위도 있지만 센서 제어 및 VPN 터널링에 우수한 성능을 발휘하도록 IP, 서비스 포트별로 원하는 대로 배치시킨 각 큐에 대해서도 우선순위를 할당할 수 있고, 각 큐의 대역폭도 원하는 대로 지정하도록 할 수 있다.

[0109] 구체적으로 도 4를 참조하면, 먼저 tap0에 HTB qdisc를 핸들 1: 이름으로 할당하고, 디폴트 패킷은 모두 1:20으로 보낸다. 그러면 클래스 1:20은 SFQ qdisc 20: 큐를 통해 패킷을 전송하며, 클래스 1:1은 패킷을 서비스 포트별, IP 주소별로 구분하고, 서비스 포트별(ssh, ftp, telnet), IP 주소별(192.168.2.118, 192.168.2.248, 192.168.2.200, 192.162.2.110, 192.168.2.112,113,119,120,121)로 지정된 해당 클래스를 통해 각 클래스와 매칭되는 큐로 할당한다. 각 클래스는 서비스 포트별, IP 주소별로 정해진 우선순위에 따라 다중 큐잉을 수행한다.

[0110] 여기서, ssh 접속된 서비스 포트와 특정 IP 주소(192.168.2.112,113,119,120,121)에 대해서만 가장 높은 우선순위로 다중 큐잉을 수행하고, ftp 접속에 대해서는 가장 하위 우선순위로 다중 큐잉하는 예를 보여주고 있다.

[0111] 이는 트래픽 쉐이핑의 일례를 도시한 것으로, 본 발명의 일 실시예에 따른 트래픽 쉐이핑에 의하면, 시시각각 변하는 네트워크 상황에 따라 원하는 동작을 하는 트래픽 쉐이핑 알고리즘을 각각 구현해 두고, 네트워크 상황에 따라 적절한 알고리즘을 선택하여 실행할 수 있다.

[0113] 도 5는 본 발명의 일 실시예에 따른 트래픽 쉐이핑을 적용한 시스템과 기존 시스템 간 성능 분석 결과를 나타낸 비교표이다.

[0114] 도 5에 도시된 바와 같이, 본 연구결과는 동급 H/W장비에 비교해서는 전 분야 월등히 우수한 성능을 보여 주었고, 고가의 전용 VPN장비에 비교해도 지원방식, 호환성 그리고 특히 트래픽 쉐이핑 분야에서 우수한 성능을 보여주고 있다. 특히 다중큐잉 실시간 트래픽 쉐이핑 기술 적용으로 전용선이 아닌 일반 인터넷 회선상 에서도 우수한 성능을 기대할 수 있다.

[0115] 본 연구의 성능은 2가지로 생각해볼 수 있다.

[0116] 첫째, 정량적인 성능 향상으로 커널 튜닝 및 VPN 터널링 알고리즘 최적화를 통해서 동급 H/W에 비교하여 3배 이상의 성능 향상을 측정할 수 있다.

[0117] 둘째, 정성적인 성능 향상이다. 실시간 네트워크 상황에 따라 다중 큐잉에 적용되는 우선순위와 속도를 세분화된 큐잉별(서비스별)로 조절하여 전체적인 속도 총량은 동일하지만 우선순위에 따라 더 많이 할당하는 방식으로 하여 체감 속도를 높일 수 있는 속도 분배가 가능한데 있다.

[0118] 특히, 본 발명의 일 실시예에 따른 사물인터넷을 지원하는 스마트 게이트웨이의 경우, 다른 서비스보다 사물인터넷 관련 패킷 큐잉에 대해 우선순위를 높임으로써 체감할 정도로 성능이 향상되는 구현 방법이다.

[0120] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 예를 들어, 프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령

(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0121] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0123] 도 6 내지 도 9는 본 발명의 일 실시예에 따른 사물인터넷을 지원하는 스마트 게이트웨이의 VPN 터널링 실시간 속도 제어 방법을 설명하기 위해 도시한 흐름도이다.

[0124] 여기서 설명하는 VPN 터널링 실시간 속도 제어 방법은 본 발명의 하나의 실시예에 불과하며, 그 이외에 필요에 따라 다양한 단계들이 부가될 수 있고, 하기의 단계들도 순서를 변경하여 실시될 수 있으므로, 본 발명이 하기에 설명하는 각 단계 및 그 순서에 한정되는 것은 아니다.

[0125] 먼저 도 6을 참조하면, 단계(610)에서 상기 스마트 게이트웨이는 VPN 터널링 구간에 걸리는 트래픽 부하, 및 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석할 수 있다.

[0126] 이를 위해, 상기 스마트 게이트웨이는 도 7에 도시된 바와 같이, 상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭을 계산하고(710), 상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 상기 각 큐잉별 트래픽 부하량을 계산할 수 있다(720). 이어서, 상기 스마트 게이트웨이는 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다(730).

[0127] 다른 실시예로서, 상기 스마트 게이트웨이는 도 8에 도시된 바와 같이, 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하고(810), 상기 VPN 터널링 데몬의 부하량에 기초하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산할 수 있다(820). 이어서, 상기 스마트 게이트웨이는 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다(830).

[0128] 또 다른 실시예로서, 상기 스마트 게이트웨이는 도 9에 도시된 바와 같이, VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고(910), 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다(920).

[0129] 다시 도 6을 참조하면, 단계(620)에서 상기 스마트 게이트웨이는 상기 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐에 적응적으로 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0131] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CDROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은

기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0132] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0133] 그러므로, 다른 구현들, 다른 실시예들 및 청구범위와 균등한 것들도 후술하는 청구범위의 범위에 속한다.

부호의 설명

[0135] 110: OpenVPN Server Access Gateway

120: OpenVPN Client Gateway

130: OpenVPN Client Gateway 제어 시스템

210: 다중 큐

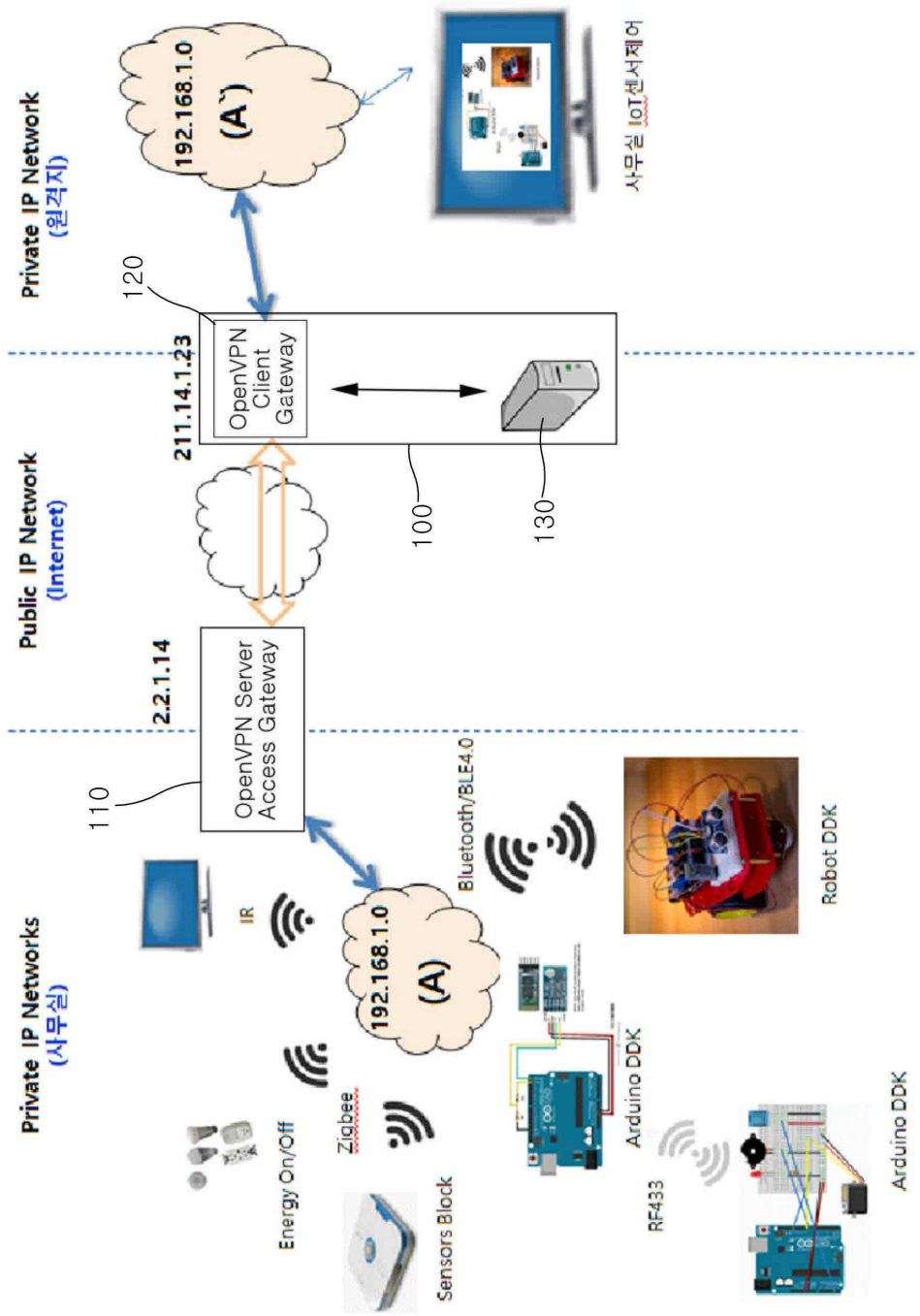
220: 큐 제어부

230: 패킷 할당부

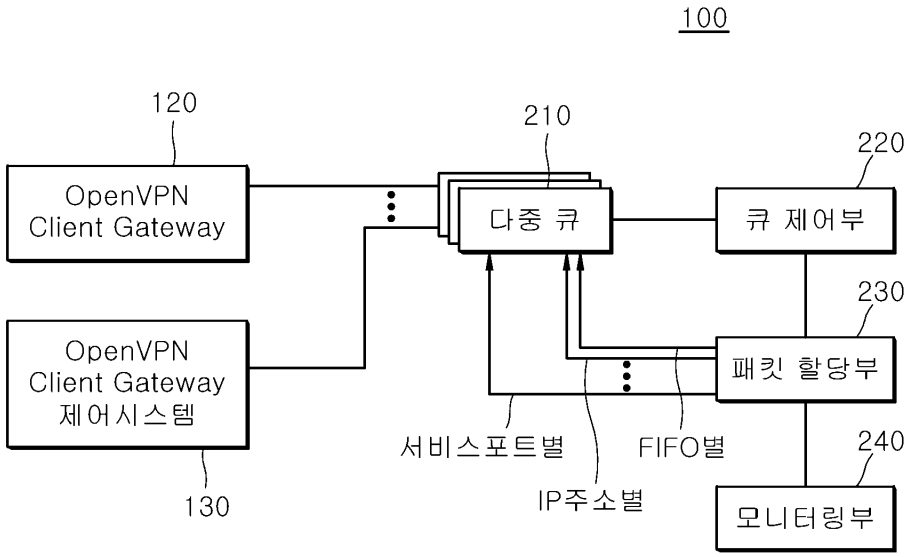
240: 모니터링부

도면

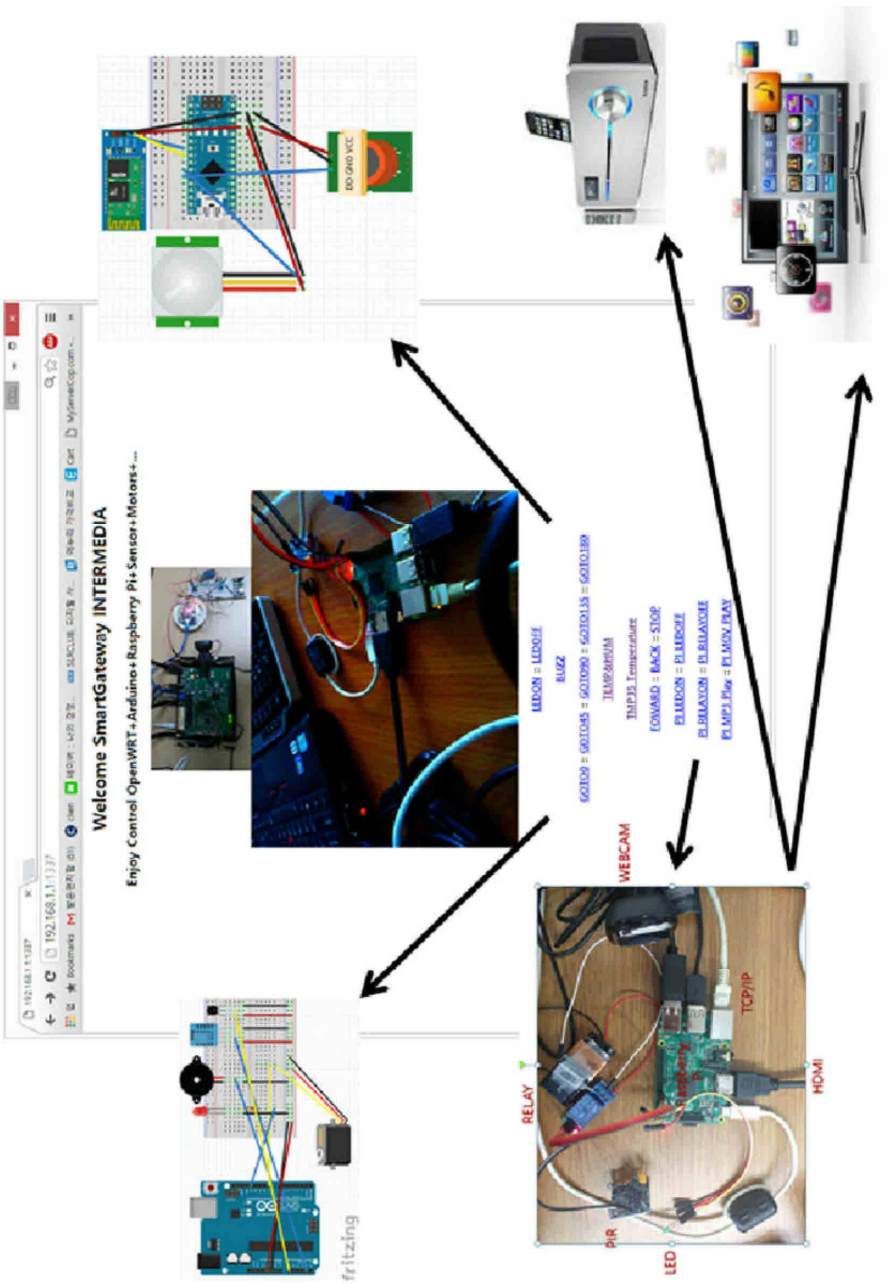
도면1


Private IP Network
(원격지)
192.168.1.0
(A`)
사무실 IoT센서제어
120
211.14.1.23
OpenVPN
Client
Gateway
Public IP Network
(Internet)
100
130
2.2.1.14
OpenVPN Server
Access Gateway
110
Private IP Networks
(사무실)
IR
192.168.1.0
(A)
Bluetooth/BLE4.0
Robot DDK
Arduino DDK
Zigbee
Energy On/Off
Sensors Block
RF433
Arduino DDK

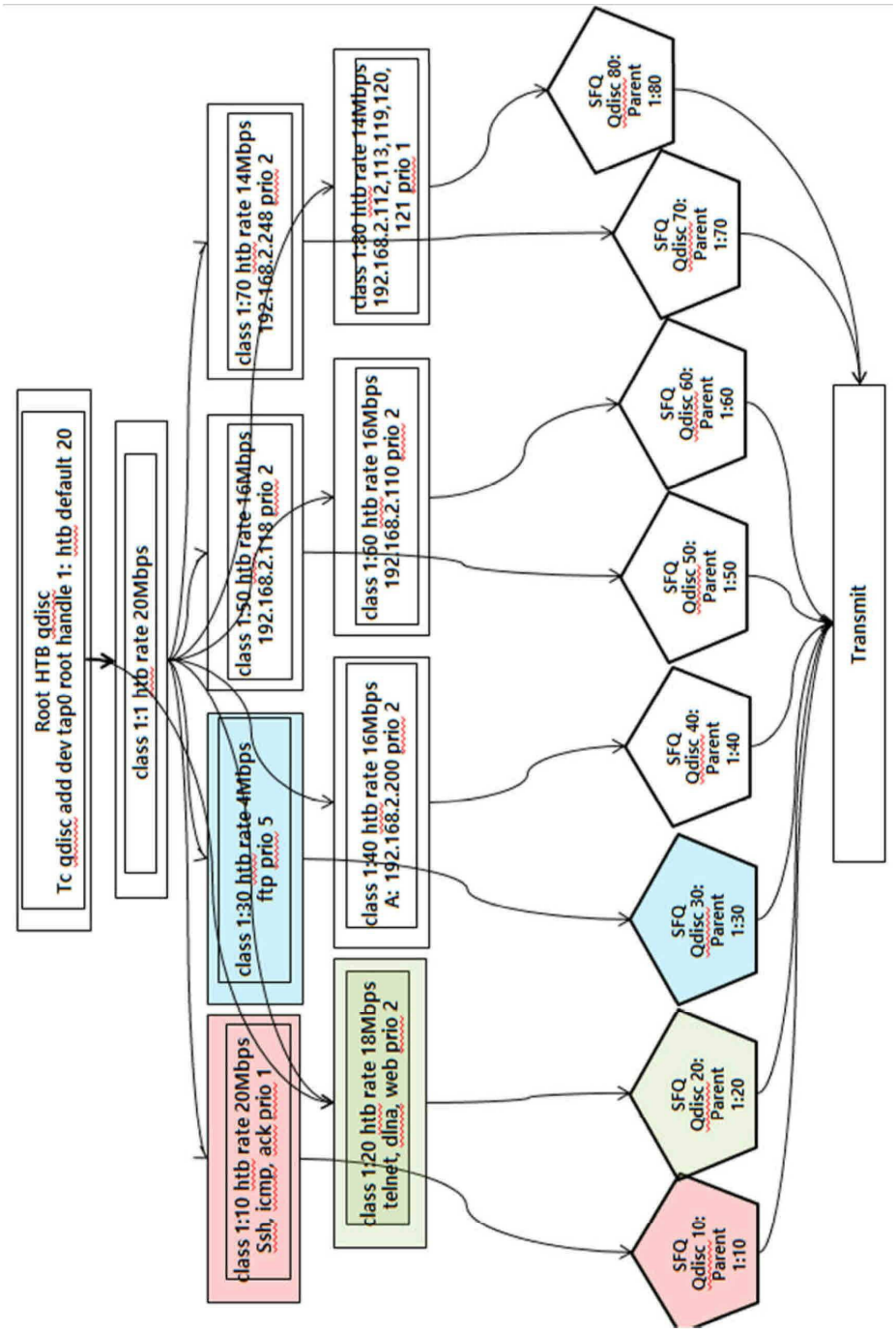
도면2


100
120
OpenVPN
Client Gateway
OpenVPN
Client Gateway
제어시스템
130
210
다중 큐
220
큐 제어부
230
패킷 할당부
240
모니터링부
서비스포트별
IP주소별
FIFO별

도면3


Welcome SmartGateway INTERMEDIA
Enjoy Control OpenWRT+Arduino+Raspberry Pi+Sensor+Motors+...
LEDON = LEDOFF
BUZZ
GOTO0 = GOTO45 = GOTO90 = GOTO135 = GOTO180
TEMP/HUM
TEMP35 Temperature
FOWARD = BACK = STOP
FLMEDIA = FLMEDIA
FLBRIGHTON = FLBRIGHTOFF
FLMP3 Play = FLMP3 PLAY
WEBCAM
TCP/IP
HDMI
RELAY
PIR
LED
fritzing

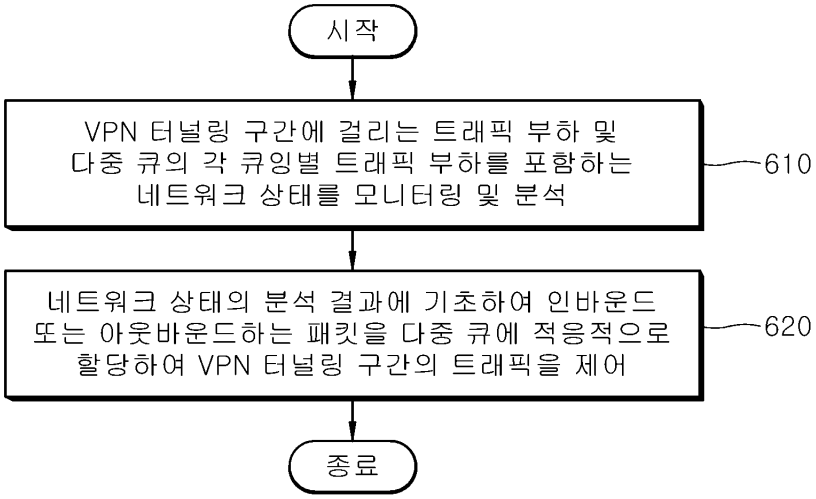
도면4


Root HTB qdisc
Tc qdisc add dev tap0 root handle 1: htb default 20
class 1:1 htb rate 20Mbps
class 1:10 htb rate 20Mbps
Ssh, icmp, ack prio 1
class 1:30 htb rate 4Mbps
ftp prio 5
class 1:50 htb rate 16Mbps
192.168.2.118 prio 2
class 1:70 htb rate 14Mbps
192.168.2.248 prio 2
class 1:20 htb rate 18Mbps
telnet, dlna, web prio 2
class 1:40 htb rate 16Mbps
A: 192.168.2.200 prio 2
class 1:60 htb rate 16Mbps
192.168.2.110 prio 2
class 1:80 htb rate 14Mbps
192.168.2.112,113,119,120,
121 prio 1
SFQ
Qdisc 10:
Parent
1:10
SFQ
Qdisc 20:
Parent
1:20
SFQ
Qdisc 30:
Parent
1:30
SFQ
Qdisc 40:
Parent
1:40
SFQ
Qdisc 50:
Parent
1:50
SFQ
Qdisc 60:
Parent
1:60
SFQ
Qdisc 70:
Parent
1:70
SFQ
Qdisc 80:
Parent
1:80
Transmit

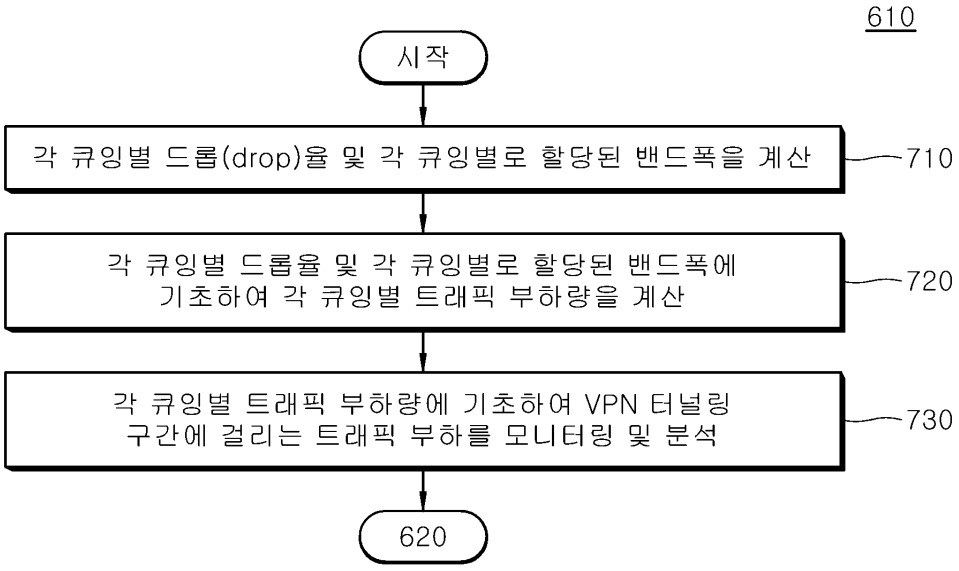
도면5

번호	항목	연구결과	동급H/W	전용 VPN
1	VPN터널링 속도	25~33Mb	5~10Mb	30-100Mb
2	VPN터널링 방법	SSL	SSL	IPSec
3	VPN지원방식	G2G, G2C	G2C	G2G
4	인터넷 회선	일반	일반	전용선
5	IDS, IPS	O	X	O
6	트래픽 쉐이핑	다중큐, 실시간	미지원(QoS)	단일큐
7	IoT지원	O	X	X
8	호환성	높음	낮음	낮음

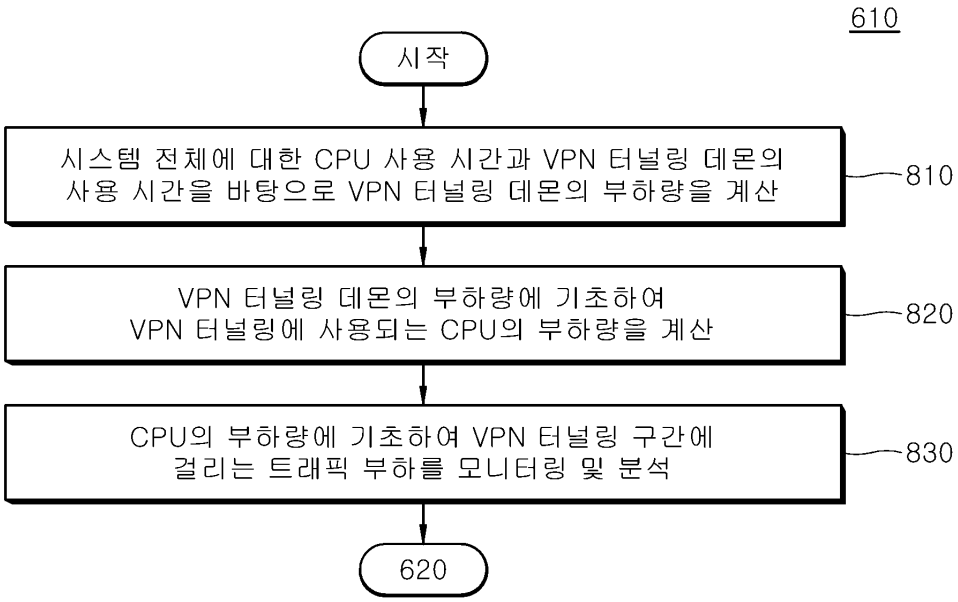
도면6



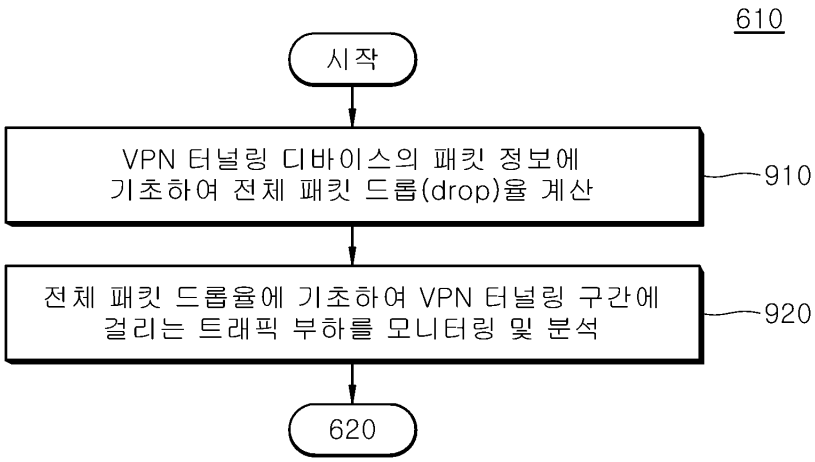
도면7



도면8



도면9



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 제11 내지 제13항

【변경전】

사물인터넷 기반 스마트 게이트웨이.

【변경후】

사물인터넷 지원 스마트 게이트웨이.