



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2020년01월09일

(11) 등록번호 10-2064679

(24) 등록일자 2020년01월03일

(51) 국제특허분류(Int. Cl.)

H04L 12/863 (2013.01) H04L 12/861 (2013.01)

H04L 12/935 (2013.01)

(52) CPC특허분류

H04L 47/6215 (2013.01)

H04L 49/3027 (2013.01)

(21) 출원번호 10-2018-0086807

(22) 출원일자 2018년07월25일

심사청구일자 2018년07월25일

(56) 선행기술조사문헌

JP05153122 A*

KR1020080004074 A*

KR1020170085906 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

이원준

서울특별시 강남구 선릉로 221, 104동 2005호(도곡동, 도곡렉슬아파트)

김규영

서울특별시 성북구 안암로 145 고려대학교 우정정보통신관 507A

(74) 대리인

송인호, 윤형근, 최영중, 최관락

전체 청구항 수 : 총 2 항

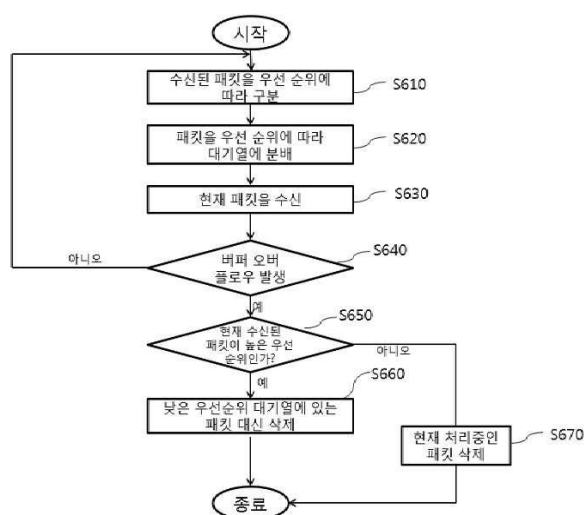
심사관 : 김대성

(54) 발명의 명칭 데이터 처리 방법

(57) 요약

본 발명은 네트워크 스위치 장치에서 버퍼 헤드를 없이 마이크로버스트 트래픽을 흡수할 수 있는 데이터 처리 방법에 관한 것이다.

본 발명의 일실시예에 의한 데이터 처리 방법은 네트워크 스위치 상에서 데이터를 처리하는 방법에 있어서, 수신된 패킷의 우선순위를 낮은 우선순위 또는 높은 우선순위로 구분하는 단계; 상기 구분된 우선순위에 따라 패킷을 출력 포트의 대기열에 분배하는 단계; 및 버퍼 오버 플로우 발생 시, 현재 수신된 패킷의 우선순위에 따라 대기열에 존재하는 패킷 또는 현재 처리 중인 패킷을 처리하는 단계를 포함할 수 있다.

대표도 - 도6

(52) CPC특허분류

H04L 49/9084 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2018023371

부처명 과학기술정보통신부

연구관리전문기관 한국연구재단

연구사업명 차세대정보·컴퓨팅기술개발

연구과제명 Web 응용을 위한 Seamless 자율 전송 네트워킹 제어 프레임워크

기 여 율 60/100

주관기관 고려대학교 산학협력단

연구기간 2018.03.01 ~ 2019.02.28

이 발명을 지원한 국가연구개발사업

과제고유번호 20170001950021001

부처명 과학정보통신부

연구관리전문기관 정보통신진흥센터

연구사업명 정보통신방송연구개발사업

연구과제명 멀티 서비스를 지원하는 프로그래머블 스위치 제어 기술 개발

기 여 율 40/100

주관기관 고려대학교산학협력단

연구기간 2018.01.01 ~ 2018.12.31

명세서

청구범위

청구항 1

네트워크 스위치 상에서 데이터를 처리하는 방법에 있어서,

수신된 패킷의 우선순위를 낮은 우선순위 또는 높은 우선순위로 구분하는 단계;

상기 구분된 우선순위에 따라 패킷을 출력 포트의 대기열에 분배하는 단계;

포트에 할당된 버퍼 크기를 현재 수신된 패킷의 크기와 모든 대기열 크기의 합과 비교하여 버퍼 크기를 초과했는지 여부에 의해 버퍼 오버 플로우가 발생했는지 여부를 확인하는 단계; 및

상기 버퍼 오버 플로우 발생 시, 현재 수신된 패킷의 우선순위에 따라 대기열에 존재하는 패킷 또는 현재 처리 중인 패킷을 처리하는 단계를 포함하되,

상기 패킷을 처리하는 단계는

상기 현재 수신된 패킷이 높은 우선순위의 패킷인 경우, 낮은 우선순위 대기열에 존재하는 패킷을 삭제하고, 현재 수신된 패킷은 상기 출력 포트의 대기열에 분배하는 단계; 및

상기 현재 수신된 패킷이 낮은 우선순위의 패킷인 경우, 현재 처리 중인 패킷을 삭제하는 단계를 포함하고,

상기 패킷을 처리하는 단계는 낮은 우선순위 대기열의 크기가 현재 수신된 패킷의 크기보다 큰 경우에만 수행되고,

상기 낮은 우선순위 대기열의 크기가 현재 수신된 패킷의 크기보다 작은 경우는 상기 현재 수신된 패킷이 드롭되고,

상기 버퍼 오버 플로우가 발생하지 않는 경우, 현재 수신된 패킷을 확인된 우선순위에 따라 대기열에 분배하는 단계를 포함하는 것을 특징으로 하는 데이터 처리 방법.

청구항 2

삭제

청구항 3

제1항에 있어서, 상기 삭제되는 패킷의 개수는

상기 현재 수신된 패킷 크기에 근거하여 결정되는 것을 특징으로 하는 데이터 처리 방법.

청구항 4

삭제

청구항 5

삭제

발명의 설명

기술 분야

[0001] 본 발명은 데이터 처리 방법에 관한 것으로, 보다 상세하게는 네트워크 스위치 장치에서 버퍼 헤드룸 없이 마이크로버스트 트래픽을 흡수할 수 있는 데이터 처리 방법에 관한 것이다.

배경 기술

[0002] 웹 검색, 분산 기계학습, 그리고 클라우드 스토리지와 같은 오늘날의 데이터센터 응용들은 크기가 작은 플로우

(정의된 임계값 미만 크기를 가진 플로우, 이하 스몰 플로우)를 위해 낮은 지연을, 크기가 큰 플로우(정의된 임계값 이상 크기를 가진 플로우, 이하 라지 플로우)를 위해 높은 처리량을 요구한다. 이러한 응용들은 마이크로버스트라고 불리는 버스티한 트래픽 패턴을 만들어내는데, 이는 현재 데이터센터에서 일반적이며 데이터센터 네트워크 혼잡의 대부분을 차지하고 있다. 마이크로버스트로 인한 패킷 손실은 플로우 완료 시간(이하 Flow Completion Time, FCT)을 악화시키기 때문에 마이크로버스트를 다루는 것은 중요한 문제이다. 마이크로버스트 트래픽은 응용이 생성하므로 일반적으로 플로우의 크기가 작다.

[0003] ECN(Explicit Congestion Notification)은 저지연을 달성하기 위해 최근 데이터 센터 전송 솔루션에 널리 사용되는 기술이다. ECN이 적용된 스위치는 ECN 마킹 임계 값을 넘은 패킷을 혼잡상태로 마킹하여 송신자가 전송물을 감소하게 한다. 따라서 대기열 길이는 ECN 마킹 임계 값 주변으로 낮게 유지되며, 이를 통해 스위치의 버퍼 헤드룸을 확보하여 마이크로버스트 트래픽을 흡수 할 수 있다. 관련된 선행문헌으로 대한민국 등록특허 제10-1732737호가 있다.

[0004] 불행히도, 버퍼 헤드 룸의 존재는 지연 시간과 처리량 사이의 근본적인 트레이드 오프를 야기한다. 스위치는 회선 속도 처리율(Line-rate throughput)을 제공하기 위해 적어도 $C * RTT$ (즉, BDP)의 버퍼 크기를 필요로 한다, 여기서 C는 링크 용량을 나타낸다. $K \geq C * RTT$ 를 사용하면 높은 처리량을 달성 할 수 있다. 그러나 이 경우 헤드 룸 크기가 작아지기 때문에 스위치는 마이크로버스트를 충분히 흡수 할 수 없으므로 과도한 패킷 손실이 발생할 수 있다. 반대로 $K < C * RTT$ 를 설정하면 스위치는 마이크로버스트 트래픽을 충분히 흡수 할 수 있지만 높은 처리량을 달성 할 수는 없다. 스위치가 충분한 버퍼 공간을 가지고 있음에도 송신자는 ECN 신호에 과도하게 반응하여 혼잡 윈도우 크기를 과도하게 줄이게 된다.

[0005] 따라서 높은 처리량과 낮은 지연을 동시에 만족하면서 마이크로버스트 트래픽을 흡수할 수 있는 데이터 처리 기술에 대한 연구가 필요한 실정이다.

발명의 내용

해결하려는 과제

[0006] 본 발명의 목적은 높은 처리량과 낮은 지연을 동시에 만족하면서 마이크로버스트 트래픽을 흡수할 수 있는 데이터 처리 방법을 제공하는데 있다.

과제의 해결 수단

[0007] 상기 목적을 달성하기 위해 본 발명의 일실시예에 의하면, 네트워크 스위치 상에서 데이터를 처리하는 방법에 있어서, 수신된 패킷의 우선순위를 낮은 우선순위 또는 높은 우선순위로 구분하는 단계; 상기 구분된 우선순위에 따라 패킷을 출력 포트의 대기열에 분배하는 단계; 및 버퍼 오버 플로우 발생 시, 현재 수신된 패킷의 우선순위에 따라 대기열에 존재하는 패킷 또는 현재 처리 중인 패킷을 처리하는 단계를 포함하는 데이터 처리 방법이 개시된다.

발명의 효과

[0008] 본 발명의 일실시예에 의한 데이터 처리 방법은 대기열 내 패킷을 떨어뜨림으로써 도착하는 패킷을 위한 공간을 확보할 수 있으므로 버퍼 헤드룸없이도 마이크로버스트 트래픽을 흡수 할 수 있다

[0009] 본 발명의 일실시예에 의하면, 포트 버퍼 전체가 마이크로버스트 트래픽을 흡수하기 위해 활용될 수 있다.

[0010] 본 발명의 일실시예에 의하면, 버퍼 헤드룸이 없기 때문에 대기열 길이를 제한하지 않는다. 따라서 스위치는 높은 처리량을 달성하기 위해 충분한 크기의 버퍼 공간을 활용할 수 있다.

도면의 간단한 설명

[0011] 도 1은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법을 설명하기 위한 도면이다.

도 2 내지 도 3은 본 발명의 일실시예와 관련된 마이크로버스트를 설명하기 위한 도면이다.

도 4는 본 발명의 일실시예와 관련된 ECN(Explicit Congestion Notification)을 적용할 때, ECN 임계값과 버퍼 헤드룸과의 관계를 설명하기 위한 도면이다.

도 5는 본 발명의 일실시예와 관련된 ECN 임계값에 따른 처리량과 지연시간과의 관계를 설명하기 위한 도면이다.

도 6은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법을 나타내는 흐름도이다.

도 7은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법에서 수신된 패킷의 우선순위에 따라 대기열에 존재하는 패킷을 처리하는 방법을 설명하기 위한 도면이다.

도 8은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법을 나타내는 의사코드(pseudocode)이다.

도 9 내지 도 10은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법과 다른 방법과의 성능을 비교한 그래프이다.

발명을 실시하기 위한 구체적인 내용

- [0012] 이하, 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법에 대해 도면을 참조하여 설명하도록 하겠다.
- [0013] 본 명세서에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "구성된다" 또는 "포함한다" 등의 용어는 명세서상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.
- [0014] 도 1은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법을 설명하기 위한 도면이다.
- [0015] 도시된 바와 같이, 어그리게이터(Aggregator)는 스위치 장치(100)를 통해 특정 데이터 처리(예: 키워드 검색)을 요청할 수 있다. 그러면 상기 스위치 장치(100)는 연결된 워커(worker 1~ worker 4)들은 분산 작업을 수행하고, 수행된 결과는 상기 스위치 장치(100)를 통해 상기 어그리게이터(Aggregator)로 리턴될 수 있다.
- [0016] 도 2 내지 도 3은 본 발명의 일실시예와 관련된 마이크로버스트를 설명하기 위한 도면이다.
- [0017] 보통 워커(worker 1~ worker 4)들은 성능의 거의 유사하기 때문에 처리 시간이 거의 균일하다. 이 경우 도 2에 도시된 바와 같이 거의 동일한 시간에 각 워커들에서 수행한 결과가 상기 스위치 장치(100)로 전달될 수 있다.
- [0018] 이와 같이, 짧은 시간동안 발생하는 트래픽 버스트를 마이크로버스트(microburst)라고 한다. 도 3에 도시된 피크는 마이크로버스트를 나타낸다. 마이크로버스트는 버퍼 오버플로우 및 이로 인한 패킷 손실을 발생시키고, 이로 인해 응용 성능이 저하된다.
- [0019] 한편, ECN(Explicit Congestion Notification)은 저지연을 달성하기 위해 최근 데이터 센터 전송 솔루션에 널리 사용되는 기술이다.
- [0020] 도 4는 본 발명의 일실시예와 관련된 ECN(Explicit Congestion Notification)을 적용할 때, ECN 임계값과 버퍼 헤드룸과의 관계를 설명하기 위한 도면이다.
- [0021] 도시된 바와 같이, ECN이 적용된 스위치는 ECN 마킹 임계 값을 넘은 패킷을 혼잡상태로 마킹하여 송신자가 전송률을 감소하게 한다. 따라서 대기열 길이는 ECN 마킹 임계 값(ECN Threshold K) 주변으로 낮게 유지되며, 이를 통해 스위치는 버퍼 헤드룸을 확보하여 마이크로버스트 트래픽을 흡수 할 수 있다. 여기서 버퍼 헤드룸의 크기는 B(포트 버퍼의 크기)-K가 된다. 따라서 K 값이 클수록 버퍼 헤드룸의 크기는 줄어들게 된다.
- [0022] 도 5는 본 발명의 일실시예와 관련된 ECN 임계값에 따른 처리량과 지연시간과의 관계를 설명하기 위한 도면이다.
- [0023] 도시된 바와 같이, ECN이 적용된 스위치를 사용할 경우, 버퍼 헤드 룸의 존재는 지연 시간과 처리량 사이의 근본적인 트레이드오프를 야기한다. 스위치는 회선 속도 처리율(Line-rate throughput)을 제공하기 위해 적어도 $C * RTT$ (즉, BDP)의 버퍼 크기를 필요로 한다. 여기서 C는 링크 용량을 나타낸다. 만약, $K \geq C * RTT$ 를 사용하면 높은 처리량을 달성 할 수 있다. 그러나 이 경우 버퍼 헤드 룸 크기가 작아지기 때문에 스위치는 마이크로버스트를 충분히 흡수할 수 없으므로 과도한 패킷 손실이 발생할 수 있다. 반대로 $K < C * RTT$ 를 설정하면 스위치는

마이크로버스트 트래픽을 충분히 흡수 할 수 있지만 높은 처리량을 달성 할 수는 없다. 스위치가 충분한 버퍼 공간을 가지고 있음에도 송신자는 ECN 신호에 과도하게 반응하여 혼잡 윈도우 크기를 과도하게 줄이게 된다.

- [0024] 도 5(a)는 K값에 따른 스위치 처리량을 나타내고, 도 5(b)는 K값에 따른 스몰 플로우의 평균 플로우 완료 시간 지연(Flow Latency)을 나타낸다.
- [0025] K가 높을수록 처리량은 증가하지만 지연 시간은 나빠지게 됨을 확인할 수 있다. 반대로, 마킹 임계 값 K가 낮아질수록 헤더 크기 커져서 지연 시간이 감소하지만, 처리량 역시 급격하게 감소함을 볼 수 있다.
- [0026] 도 6은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법을 나타내는 흐름도이다.
- [0027] 상기 스위치(100)는 패킷이 도착하면, 도착된 패킷의 우선순위를 확인하여 구분할 수 있다(S610). 이하에서 H는 높은 우선순위를, L는 낮은 우선순위를 의미한다. 상기 패킷의 우선순위는 패킷을 보내는 송신자가 패킷 전송 시 결정할 수 있다. 또는 상기 패킷의 우선순위는 스위치 장치(100)가 패킷 수신 후 전처리 과정에서 스몰 플로우 패킷에 우선순위 H를, 라지 플로우 패킷에 우선순위 L을 부여할 수도 있다. 여기서 스몰 플로우는 정의된 임계값 미만 크기를 가진 플로우를 말하고, 라지 플로우는 정의된 임계값 이상 크기를 가진 플로우를 의미할 수 있다.
- [0028] 상기 스위치 장치(100)는 확인된 우선순위에 따라 상기 패킷을 대기열에 분배한다(S620).
- [0029] 도 7은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법에서 수신된 패킷의 우선순위에 따라 대기열에 존재하는 패킷을 처리하는 방법을 설명하기 위한 도면이다.
- [0030] 도시된 바와 같이, 패킷 버스트(Burst)가 스위치에 입력 포트(Input Port)를 통해 들어오면 분배기(Classifier)는 우선순위에 따라 패킷을 출력 포트(Output Port) 대기열에 분배한다. 패킷의 우선순위가 H이면, 높은 우선순위 대기열(Q_H)에 분배하고, 패킷의 우선순위가 L이면, 낮은 우선순위 대기열(Q_L)에 분배한다.
- [0031] 그렇게 패킷이 대기열에 분배된 상태에서 상기 스위치 장치(100)는 현재 패킷을 수신할 수 있다(S630).
- [0032] 이 경우, 스위치 장치(100)는 현재 패킷 수신에 의해 버퍼 오버 플로우가 발생했는지 여부를 확인할 수 있다.
- [0033] 버퍼 오버 플로우 발생은 포트에 할당된 버퍼 크기 (PortBuffer)를 현재 패킷의 크기(P.bytes)와 모든 대기열 크기의 합($\sum Q_i$.bytes)의 합과 비교하여 버퍼 크기를 초과했는지 여부에 의해 확인될 수 있다.
- [0034] 만약, 버퍼 오버 플로우가 발생하지 않았으면, 현재 수신된 패킷을 확인된 우선순위에 따라 대기열에 분배하면 된다.
- [0035] 버퍼 오버 플로우가 발생한 경우, 현재 수신된 패킷의 따라 패킷을 처리하는 과정이 달라질 수 있다.
- [0036] 버퍼 오버 플로우 발생 시, 현재 수신된 패킷의 우선순위가 H인 경우, 낮은 우선순위 대기열 Q_L 의 꼬리 패킷 P_i 를 삭제함으로써 현재 수신된 패킷 P를 대기열에 넣을 수 있는 공간을 확보할 수 있다(Remove($Q_L P_i$))(S640, S650, S660). 이 때 현재 패킷 P의 크기에 대응되게 복수의 P_i 가 대기열로부터 삭제될 수 있다.
- [0037] 만약, 버퍼 오버 플로우 발생 시, 현재 수신된 패킷의 우선순위가 L인 경우, 현재 처리 중인 패킷 P를 삭제할 수 있다(Drop(P))(S640, S650, S670).
- [0038] 한편 상기 동작은 대기열 Q_L 의 크기가 패킷 P의 크기보다 커야만 작동하며 이 외의 경우는 대기열 Q_L 의 모든 패킷을 삭제해도 패킷 P가 들어갈 공간이 확보되지 않는다는 의미이므로 이 경우는, 현재 처리 중인 패킷 P를 드롭시킬 수 있다.
- [0039] 상기 패킷 P를 위한 공간이 버퍼에 존재한다면 패킷 P는 버퍼에 들어간다(Enqueue(P)).
- [0040] 도 8은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법을 나타내는 의사코드(pseudocode)이다. 즉, 도 8은 도 6의 의사코드라고 할 수 있다.
- [0041] 패킷 버스트(Burst)가 스위치에 입력 포트(Input Port)를 통해 들어오면 분배기(Classifier)는 우선순위에 따라 패킷을 출력 포트(Output Port) 대기열에 분배하는데, 포트 버퍼 공간이 부족한 경우 대기열 Q_L 에서 버스트와 같은 크기만큼의 패킷을 떨어뜨림을 확인할 수 있다. 출력 포트의 스케줄링은 우선순위 스케줄링(Strict

Priority Queueing, SPQ)를 이용한다.

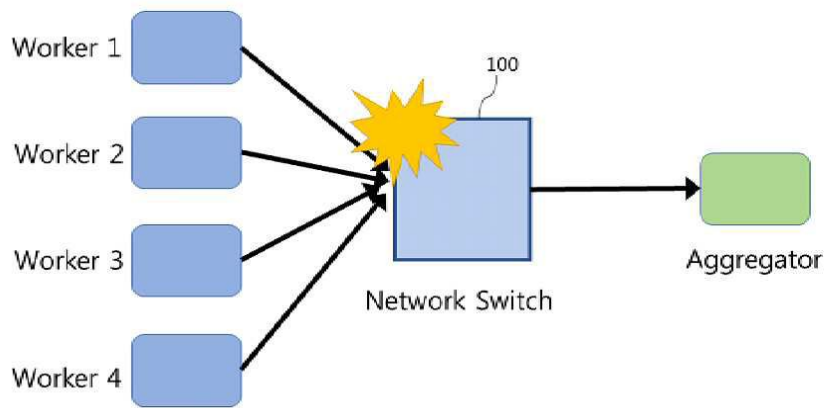
- [0042] 도 9 내지 도 10은 본 발명의 일실시예와 관련된 네트워크 스위치 장치에서 데이터를 처리하는 방법과 다른 방법과의 성능을 비교한 그래프이다.
- [0043] 도 9 내지 도 10은 ns-2 시뮬레이터를 이용한 본 발명의 일실시예 결과로써, 일반적으로 스위치 상에서 사용되는 기본 패킷 드랍테일 Droptail(드랍테일)과 ECN을 비교하였다.
- [0044] 도 9는 본 발명의 일실시예인 LossPass와 Droptail ECN의 처리량을 ECN 마킹 임계값 K를 10부터 90까지 증가시키며 측정한 결과이다. 본 결과 측정을 위해서 10Gbps 스위치에 연결된 두 개의 서버를 각각 송신자와 수신자로 나누어, 송신자가 하나의 영구적인 TCP 플로우를 수신자에게 전송토록 하였다. LossPass와 Droptail은 마킹 임계값을 사용하지 않기 때문에 하나의 결과값을 여러 마킹 임계값에 사용하여 ECN과 비교하였다. 임계값 K에 따라 처리량이 변하는 ECN과 달리 본 발명은 항상 높은 처리량을 달성함을 확인할 수 있다.
- [0045] 도 10은 LossPass FCT에 미치는 영향을 확인하기 위한 실시예의 결과로써, 총 192대의 서버, 8개의 리프 스위치, 2개의 스파인 스위치로 구성된 리프 스파인 토폴로지상에서 실험을 진행하였다. 각 리프 스위치는 24개의 10Gbps 다운링크 및 2개의 40Gbps 업링크를 가지고 있으며 베이스 RTT는 85.2us이다. 부하 분산으로는 ECMP를 이용하였다. 리프와 스파인 스위치는 각각 9MB와 16MB의 공유 버퍼를 가지며 버퍼 관리를 위해서는 동적 할당 알고리즘을 사용하였다. 워크로드로는 웹 검색 응용을 이용하였으며 이는 1.58MB의 평균 크기를 가지며 헤비테일 분포를 가지는 5,000개의 플로우를 192*191 통신쌍 사이에서 생성한다. 트래픽 로드는 50%에서 90%까지 변화를 주어 측정하였다. 결과는 본 발명의 일실시예인 LossPass의 값에 정규화되어 보다 분명하게 성능차이를 알 수 있도록 하였다. ECN(K=30)은 낮은 K값을 설정한 ECN을 대표하며 ECN(K=84)는 높은 K값을 설정한 ECN을 대표한다. 도 10(b)의 결과는 100KB 미만의 작은 플로우의 평균 FCT를 나타내는데 제안하는 발명인 LossPass가 비교 대상에 비해 최대 6배 가량 좋은 성능을 가짐을 알 수 있다. 또한 도 10(c)의 100KB 미만의 작은 플로우의 99백분위 FCT 결과를 살펴보면, LossPass가 비교대상들 중 가장 좋은 성능을 보임을 알 수 있다. 도 10(d)는 크기가 10MB이상인 큰 플로우의 평균 FCT에 대한 결과로, ECN만큼의 지연 시간을 보여주지는 못하지만 Droptail보다는 우수한 성능을 보임을 알 수 있다.
- [0046] 진술한 바와 같이, 본 발명의 일실시예에 의한 데이터 처리 방법은 대기열 내 패킷을 떨어뜨림으로써 도착하는 패킷을 위한 공간을 확보할 수 있으므로 버퍼 헤드룸없이도 마이크로버스트 트래픽을 흡수 할 수 있다
- [0047] 본 발명의 일실시예에 의하면, 포트 버퍼 전체가 마이크로버스트 트래픽을 흡수하기 위해 활용될 수 있다.
- [0048] 본 발명의 일실시예에 의하면, 버퍼 헤드룸이 없기 때문에 대기열 길이를 제한하지 않는다. 따라서 스위치는 높은 처리량을 달성하기 위해 충분한 크기의 버퍼 공간을 활용할 수 있다.
- [0049] 상술한 본 발명의 일실시예와 관련된 데이터 처리 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터로 판독 가능한 기록 매체에 기록될 수 있다. 이때, 컴퓨터로 판독 가능한 기록 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 한편, 기록매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.
- [0050] 컴퓨터로 판독 가능한 기록매체에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, 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.
- [0051] 한편, 이러한 기록매체는 프로그램 명령, 데이터 구조 등을 지정하는 신호를 전송하는 반송파를 포함하는 광 또는 금속선, 도파관 등의 전송 매체일 수도 있다.
- [0052] 또한, 프로그램 명령에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0053] 상기와 같이 설명된 데이터 처리 방법은 상기 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

부호의 설명

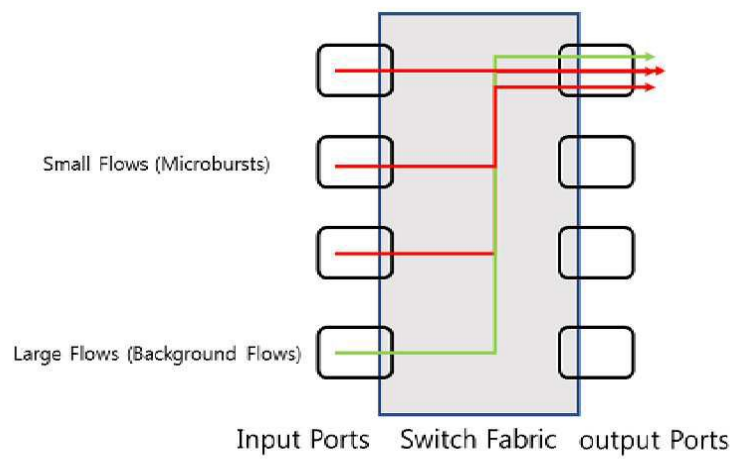
[0054] 100: 스위치 장치

도면

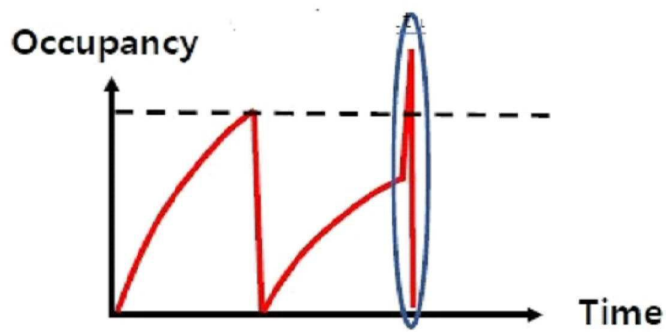
도면1



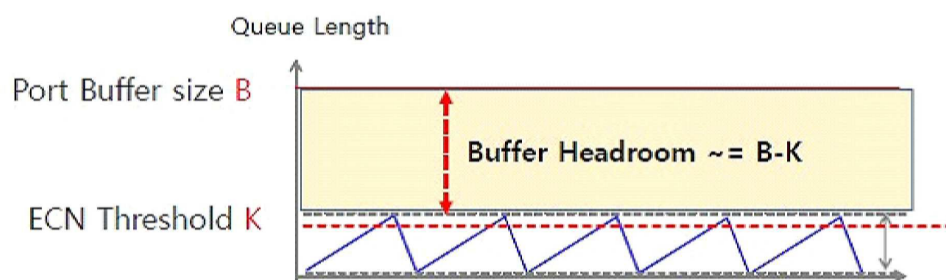
도면2



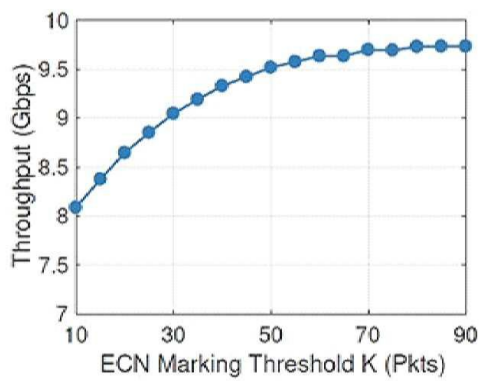
도면3



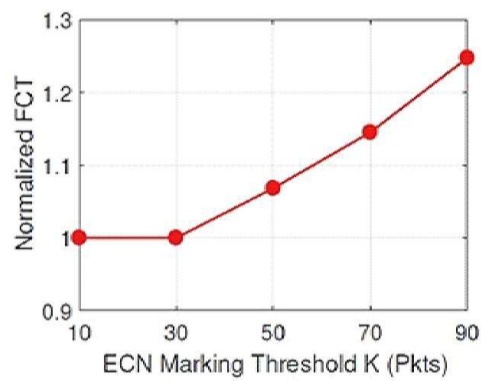
도면4



도면5

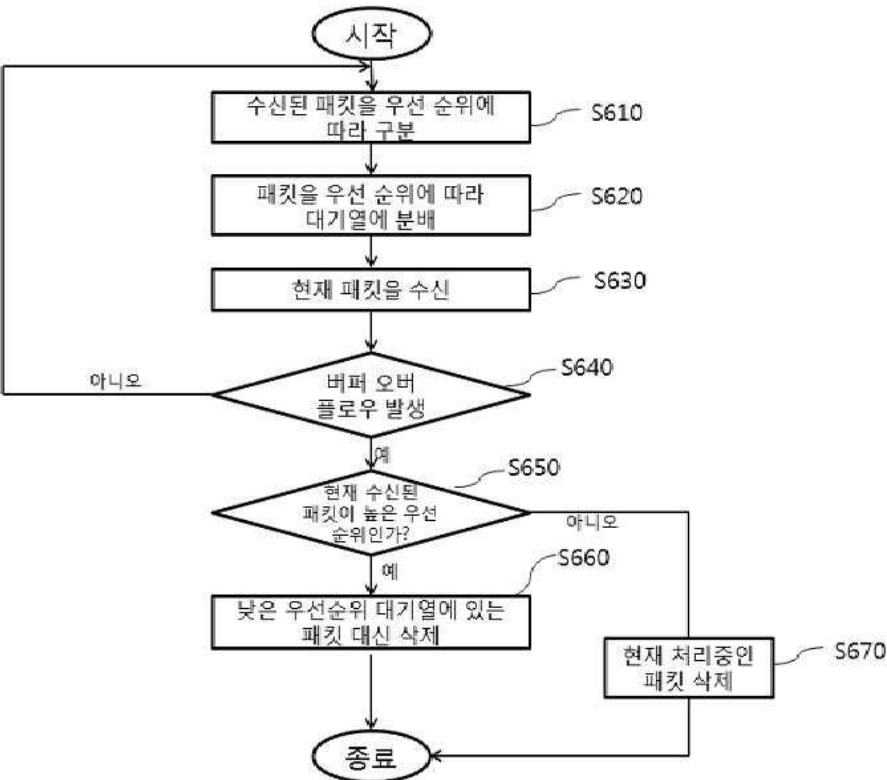


(a) Switch Throughput

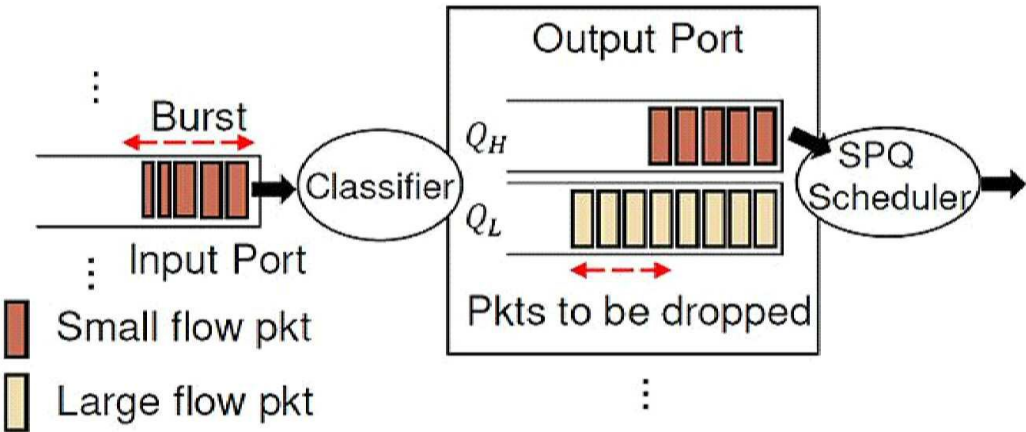


(b) Flow Latency

도면6



도면7

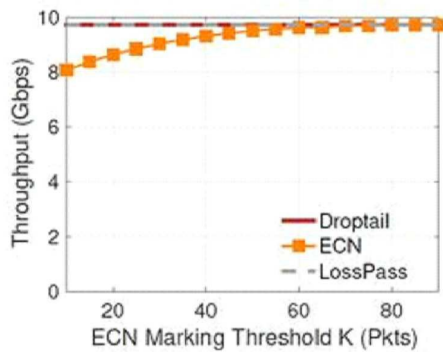


도면8

```

Require:  $\langle \text{Packet} \rangle P$                                 ▷ Active packet at a time
1:  $\langle \text{Queue} \rangle Q_H, Q_L$                                 ▷ Priority queues
2:  $\langle \text{Packet} \rangle P_t$                                     ▷ Tail packet in queue
3: if  $P.\text{Prio} = H$  then                                ▷ Small flow packet
4:   while  $(P.\text{bytes} + \sum Q_i.\text{bytes}) > \text{PortBuffer}$  do
5:     if  $P.\text{bytes} < Q_L.\text{bytes}$  then
6:        $\text{Remove}(Q_L.P_t)$                                 ▷ Drop enqueued packet
7:     else
8:       return  $\text{Drop}(P)$                                 ▷ No more packets in  $Q_L$ 
9:     end if
10:  end while
11: else                                                ▷ Large flow packet
12:  if  $(P.\text{bytes} + \sum Q_i.\text{bytes}) > \text{PortBuffer}$  then
13:    return  $\text{Drop}(P)$ 
14:  end if
15: end if
16: return  $\text{Enqueue}(P)$                                 ▷ Enqueue the packet
  
```

도면9



도면10

