



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년07월08일
(11) 등록번호 10-1415363
(24) 등록일자 2014년06월27일

(51) 국제특허분류(Int. Cl.)
H04L 12/26 (2006.01) H04L 12/24 (2006.01)
(21) 출원번호 10-2013-0016423
(22) 출원일자 2013년02월15일
심사청구일자 2013년02월15일
(56) 선행기술조사문헌
“Fit a Compact Spread Estimator in Small
High-Speed Memory”, Myungkeun Yoon ; Tao Li ;
Shigang Chen ; Jih-Kwon Peir

(73) 특허권자
인하대학교 산학협력단
인천광역시 남구 인하로 100, 인하대학교 (용현동)
(72) 발명자
양대현
서울 서초구 서초중앙로 200, 13동 1407호 (서초동, 삼풍아파트)
신동오
서울특별시 강서구 공항동 32-73 61-46번지 3층
(74) 대리인
양성보

전체 청구항 수 : 총 6 항

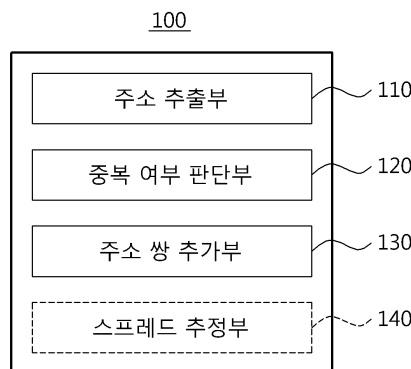
심사관 : 전용해

(54) 발명의 명칭 작은 메모리를 이용해서 스프레드를 근사하는 알고리즘

(57) 요약

본 발명의 실시예들은 네트워크 보안을 위한 스프레드 추정 시스템 및 방법에 관한 것이다. CSE(Compact Spread Estimator)를 이용하여 스프레드를 추정하는 방법에 있어서, 수신된 패킷으로부터 출발지 주소 및 목적지 주소를 추출하는 단계, 복수의 블룸필터(bloom filter)를 이용하여 상기 추출된 출발지 주소 및 목적지 주소로 구성된 주소 쌍의 중복 여부를 판단하는 단계 및 상기 주소 쌍이 중복되지 않은 것으로 판단된 경우, 상기 주소 쌍을 상기 CSE의 가상벡터에 추가하는 단계를 포함하고, 수신되는 복수의 패킷들로부터 추출되는 주소 쌍들 중 상기 출발지 주소에 대해 중복되지 않은 주소 쌍의 수에 기초하여 상기 가상벡터의 크기가 동적으로 변경되는 것을 특징으로 하는 스프레드 추정 방법이 제공될 수 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 46570

부처명 교육과학기술부

연구사업명 LINC사업 기술개발과제

연구과제명 개인 정보 유출 탐지 솔루션 개발

기 여 율 1/1

주관기관 인하대학교 산학협력단

연구기간 2012.09.01 ~ 2013.02.28

특허청구의 범위

청구항 1

CSE(Compact Spread Estimator)를 이용하여 스프레드를 추정하는 방법에 있어서,

수신된 패킷으로부터 출발지 주소 및 목적지 주소를 추출하는 단계;

복수의 블룸필터(bloom filter)를 이용하여 상기 추출된 출발지 주소 및 목적지 주소로 구성된 주소 쌍의 중복 여부를 판단하는 단계; 및

상기 주소 쌍이 중복되지 않은 것으로 판단된 경우, 상기 주소 쌍을 가상벡터에 추가하는 단계를 포함하고,

수신되는 복수의 패킷들로부터 추출되는 주소 쌍들 중 상기 출발지 주소에 대해 중복되지 않은 주소 쌍의 수에 기초하여 상기 가상벡터의 크기가 동적으로 변경되고,

상기 주소 쌍을 가상벡터에 추가하는 단계는,

상기 출발지 주소가 확장된 힛수에 기초하여 상기 가상벡터에 상기 주소 쌍을 추가하고,

상기 확장된 힛수는, 상기 가상벡터의 크기가 상기 출발지 주소와 연관하여 증가된 힛수를 포함하고,

상기 가상벡터의 크기는, 상기 가상벡터의 초기 크기에 기초하여 결정되는 확률에 따라 증가 여부가 결정되고,

상기 가상벡터의 크기가 증가한 경우, 상기 출발지 주소가 확장관리자에 등록되어, 상기 출발지 주소의 확장된 힛수가 갱신되고,

상기 출발지 주소가 확장된 힛수에 기초하여 상기 출발지 주소에 대한 스프레드를 추정하는 단계

를 더 포함하는 것을 특징으로 하는 스프레드 추정 방법.

청구항 2

제1항에 있어서,

상기 중복 여부를 판단하는 단계는,

상기 복수의 블룸필터에 상기 주소 쌍이 멤버로 등록되어 있는지 여부를 확인하여 상기 주소 쌍의 중복 여부를 판단하는 것을 특징으로 하는 스프레드 추정 방법.

청구항 3

제1항에 있어서,

상기 복수의 블룸필터는, 복수의 카운팅 블룸 필터(Counting Bloom Filter)를 포함하는 것을 특징으로 하는 스프레드 추정 방법.

청구항 4

제1항에 있어서,

상기 주소 쌍을 상기 복수의 블룸필터 중 최상위 블룸필터에 등록하는 단계

를 더 포함하고,

상기 최상위 블룸필터에 대해 기설정된 수의 주소 쌍이 등록되는 경우, 상기 복수의 블룸필터 중 최상위 블룸필터가 변경되는 것을 특징으로 하는 스프레드 추정 방법.

청구항 5

제4항에 있어서,

상기 최상위 Bloom필터에 등록하는 단계는,

상기 주소 쌍이 중복되지 않은 주소 쌍이거나 또는 상기 복수의 Bloom필터 중 마지막 Bloom필터에 등록된 멤버인 경우에 상기 주소 쌍을 상기 최상위 Bloom필터에 등록하는 것을 특징으로 하는 스프레드 추정 방법.

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

CSE(Compact Spread Estimator)를 이용하여 스프레드를 추정하는 시스템에 있어서,

수신된 패킷으로부터 출발지 주소 및 목적지 주소를 추출하는 주소 추출부;

복수의 Bloom필터(bloom filter)를 이용하여 상기 추출된 출발지 주소 및 목적지 주소로 구성된 주소 쌍의 중복 여부를 판단하는 중복 여부 판단부; 및

상기 주소 쌍이 중복되지 않은 것으로 판단된 경우, 상기 주소 쌍을 가상벡터에 추가하는 주소 쌍 추가부를 포함하고,

수신되는 복수의 패킷들로부터 추출되는 주소 쌍들 중 상기 출발지 주소에 대해 중복되지 않은 주소 쌍의 수에 기초하여 상기 가상벡터의 크기가 동적으로 변경되고,

상기 주소 쌍 추가부는,

상기 출발지 주소가 확장된 횟수에 기초하여 상기 가상벡터에 상기 주소 쌍을 추가하고,

상기 확장된 횟수는, 상기 가상벡터의 크기가 상기 출발지 주소와 연관하여 증가된 횟수를 포함하고,

상기 가상벡터의 크기는, 상기 가상벡터의 초기 크기에 기초하여 결정되는 확률에 따라 증가 여부가 결정되고,

상기 가상벡터의 크기가 증가한 경우, 상기 출발지 주소가 확장관리자에 등록되어, 상기 출발지 주소의 확장된 횟수가 갱신되고,

상기 출발지 주소가 확장된 횟수에 기초하여 상기 출발지 주소에 대한 스프레드를 추정하는 스프레드 추정부를 더 포함하는 것을 특징으로 하는 스프레드 추정 시스템.

명세서

기술분야

[0001] 본 발명의 실시예들은 네트워크 보안을 위한 스프레드 추정 시스템 및 방법에 관한 것이다.

배경기술

[0002] 네트워크 보안에서 '스프레드(spread)'는 네트워크를 관찰하는 특정 기간 동안 하나의 출발지 주소가 얼마나 다양한 목적지 주소로 네트워크 패킷을 발송했는지 측정하는 것을 의미한다. 스프레드 에스티메이터(spread estimator)는 라우터에서 동작하는 소프트웨어 또는 하드웨어 모듈로, 각 출발지 주소별로 스프레드의 수를 추정하는데 사용된다.

[0003] 최근에는 라우터가 1 메가바이트 정도의 메모리 공간을 스프레드 에스티메이터를 위해 할당할 수 있다고 가정한다. 이때 중요한 기술적 도전과제는 SRAM(Static Random Access Memory)같이 크기가 작고 고속으로 동작하는 메모리에 알맞은 스프레드 에스티메이터를 구현하는 것이다. 가장 최근에 연구된 알고리즘은 CSE(Compact

Spread Estimator)로, CSE는 SRAM에 알맞은 약 1MB~4MB 정도의 작은 메모리 공간을 사용하면서도 효율적으로 출발지-목적지 주소 쌍을 추정한다. CSE는 가상벡터(virtual vector)를 이용하여 공유되는 SRAM 메모리 비트맵 상에서 스프레드의 수를 추정한다. 예를 들어, 논문 "Fit a Compact Spread Estimator in Small High-Speed Memory"에는 종래기술에 따른 CSE에 대해 자세히 설명되어 있다.

- [0004] 그러나, CSE는 가상벡터의 크기에 따라 정확도와 셀 수 있는 한계치가 달라지는 문제점이 있다. 예를 들어, CSE에서, 가상벡터의 크기가 작으면 정확도가 높아지는 대신 큰 스프레드를 추정할 수 없게 된다. 반면에, 가상벡터가 크면 큰 스프레드를 추정할 수는 있으나, 정확도가 낮아지는 문제가 있다.

발명의 내용

해결하려는 과제

- [0005] 스프레드(spread)가 커질수록 가상벡터(virtual vector)의 개수를 동적으로 증가시켜 CSE(Compact Spread Estimator)의 한계를 극복할 수 있는 네트워크 보안을 위한 스프레드 추정 시스템 및 방법을 제공한다.

과제의 해결 수단

- [0006] CSE(Compact Spread Estimator)를 이용하여 스프레드를 추정하는 방법에 있어서, 수신된 패킷으로부터 출발지 주소 및 목적지 주소를 추출하는 단계, 복수의 bloom필터(bloom filter)를 이용하여 상기 추출된 출발지 주소 및 목적지 주소로 구성된 주소 쌍의 중복 여부를 판단하는 단계 및 상기 주소 쌍이 중복되지 않은 것으로 판단된 경우, 상기 주소 쌍을 가상벡터에 추가하는 단계를 포함하고, 수신되는 복수의 패킷들로부터 추출되는 주소 쌍들 중 상기 출발지 주소에 대해 중복되지 않은 주소 쌍의 수에 기초하여 상기 가상벡터의 크기가 동적으로 변경되는 것을 특징으로 하는 스프레드 추정 방법이 제공된다.
- [0007] 일측에 따르면, 상기 중복 여부를 판단하는 단계는, 상기 복수의 bloom필터에 상기 주소 쌍이 멤버로 등록되어 있는지 여부를 확인하여 상기 주소 쌍의 중복 여부를 판단하는 것을 특징으로 할 수 있다.
- [0008] 다른 측면에 따르면, 복수의 bloom필터는, 복수의 카운팅 bloom 필터(Counting Bloom Filter)를 포함하는 것을 특징으로 할 수 있다.
- [0009] 또 다른 측면에 따르면, 상기 스프레드 추정 방법은 상기 주소 쌍을 상기 복수의 bloom필터 중 최상위 bloom필터에 등록하는 단계를 더 포함할 수 있다. 이때, 상기 최상위 bloom필터에 대해 기설정된 수의 주소 쌍이 등록되는 경우, 상기 복수의 bloom필터 중 최상위 bloom필터가 변경될 수 있다.
- [0010] 또 다른 측면에 따르면, 상기 최상위 bloom필터에 등록하는 단계는, 상기 주소 쌍이 중복되지 않은 주소 쌍이거나 또는 상기 복수의 bloom필터 중 마지막 bloom필터에 등록된 멤버인 경우에 상기 주소 쌍을 상기 최상위 bloom필터에 등록하는 것을 특징으로 할 수 있다.
- [0011] 또 다른 측면에 따르면, 상기 주소 쌍을 가상벡터에 추가하는 단계는, 상기 출발지 주소가 확장된 힛수에 기초하여 상기 가상벡터에 상기 주소 쌍을 추가하고, 상기 확장된 힛수는, 상기 가상벡터의 크기가 상기 출발지 주소와 연관하여 증가된 힛수를 포함하는 것을 특징으로 할 수 있다.
- [0012] 또 다른 측면에 따르면, 상기 가상벡터의 크기는, 상기 가상벡터의 초기 크기에 기초하여 결정되는 확률에 따라 증가 여부가 결정되고, 상기 가상벡터의 크기가 증가한 경우, 상기 출발지 주소가 확장관리자에 등록되어, 상기 출발지 주소가 확장된 힛수가 갱신되는 것을 특징으로 할 수 있다.
- [0013] 또 다른 측면에 따르면, 상기 스프레드 추정 방법은 상기 출발지 주소가 확장된 힛수에 기초하여 상기 출발지 주소에 대한 스프레드를 추정하는 단계를 더 포함할 수 있다.
- [0014] CSE(Compact Spread Estimator)를 이용하여 스프레드를 추정하는 시스템에 있어서, 수신된 패킷으로부터 출발지 주소 및 목적지 주소를 추출하는 주소 추출부, 복수의 bloom필터(bloom filter)를 이용하여 상기 추출된 출발지 주소 및 목적지 주소로 구성된 주소 쌍의 중복 여부를 판단하는 중복 여부 판단부 및 상기 주소 쌍이 중복되지 않은 것으로 판단된 경우, 상기 주소 쌍을 가상벡터에 추가하는 주소 쌍 추가부를 포함하고, 수신되는 복수의 패킷들로부터 추출되는 주소 쌍들 중 상기 출발지 주소에 대해 중복되지 않은 주소 쌍의 수에 기초하여 상기 가상벡터의 크기가 동적으로 변경되는 것을 특징으로 하는 스프레드 추정 시스템이 제공된다.

발명의 효과

[0015] 스프레드(spread)가 커질수록 가상벡터(virtual vector)의 개수를 동적으로 증가시켜 CSE(Compact Spread Estimator)의 한계를 극복할 수 있다.

도면의 간단한 설명

[0016] 도 1은 본 발명의 일실시예에 있어서, 스프레드 추정 시스템의 내부 구성을 설명하기 위한 블록도이다.

도 2는 본 발명의 일실시예에 있어서, 스프레드 추정 방법을 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0017] 이하, 본 발명의 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다.

[0018] 본 발명의 실시예들에서는 CSE(Compact Spread Estimator)의 한계점을 극복하기 위해, 스프레드가 커질수록 가상벡터의 개수를 동적으로 증가시킬 수 있는 스프레드 추정 시스템 및 방법을 제공한다. 본 발명의 실시예들에 따른 스프레드 추정 시스템 및 방법에서는, 하나의 출발지 주소가 발생시키는 스프레드의 크기에 따라 여러 개의 가상벡터를 가질 수 있기 때문에, 출발지-목적지 주소 쌍이 중복해서 발생하는 경우 이를 효율적으로 걸러내야 한다. 따라서, 본 발명의 실시예들에 따른 스프레드 추정 시스템은 크게 두 개의 구성 요소로 이루어질 수 있다. 하나는 CSE와 같이 스프레드를 추정하기 위한 요소일 수 있고, 나머지 하나는 중복되는 주소 쌍을 효율적으로 거르기 위한 요소일 수 있다. 예를 들어, 스프레드를 추정하기 위한 메모리 공간과 중복되는 주소 쌍을 거르기 위한 필터가 이용될 수 있다. 일실시예로, 중복되는 주소 쌍을 거르기 위한 필터는 N 개의 블룸필터(bloom filter)들로 구성될 수 있다.

[0019] 이후에서는 일반 블룸필터를 이용하여 중복되는 주소 쌍을 필터링하는 방법(표 1 및 표 2의 알고리즘)과 카운팅 블룸필터를 이용하여 중복되는 주소 쌍을 필터링하는 방법(표 5 및 표 6의 알고리즘)에 대해 설명한다.

표 1

Algorithm 1: CSE with Incremental Virtual Vector (use Bloom Filter)

```

input: a packet
1  begin
2      src ← GetSourceAddress(packet)    // 패킷으로부터 출발지 IP를 얻는다.
3      dst ← GetDestinationAddress(packet) // 패킷으로부터 목적지 IP를 얻는다.
4      result ← MembershipTest(src||dst)  // 멤버십 테스트의 결과로 '0', '1', '2' 반환
5      if result is 0 then                // 새로운 주소 쌍이라면
6          Storing(src, dst)              // 가상벡터에 주소 쌍 저장
7      end if
8      if result < 2 then                  // 신규/마지막 블룸필터에 있던 주소 쌍은
9          AddTop(src||dst)                // 최상위 블룸필터에 추가한다
10         count ← count + 1               // 최상위 블룸필터에 계산된 limit 만큼
11         if 0 = count mod limit then     // 멤버를 등록하면
12             RotateBFQ()                 // 큐를 순환시킨다.
13         end if
14     end if
15 end
  
```

[0020]

- [0021] 표 1의 알고리즘 1은 스프레드 추정 방법의 전체적인 과정의 일례를 나타내고 있다. 스프레드 추정 시스템은 패킷을 수신하면, 우선 출발지 주소(함수 *GetSourceAddress()*의 리턴값 *src*)와 목적지 주소(*GetDestinationAddress()*의 리턴값 *dst*)를 추출한 뒤, 멤버십 테스트를 통해 추출한 출발지 주소와 목적지 주소의 쌍이 과거에 수신한 이력이 있는 주소 쌍인지 여부를 검사할 수 있다. 표 1에서는 멤버십 테스트의 수행 결과로, 결과값 '0', '1', '2' 중 하나(함수 *MembershipTest()*의 리턴값 *result*)가 반환될 수 있는 예를 설명하고 있다. 상기 예에서, 결과값 '0'은 수신된 패킷에서 추출된 주소 쌍이 새로운 주소 쌍임을 의미할 수 있고, 결과값 '1'과 '2'는 모두 추출된 주소 쌍이 과거에 수신한 주소 쌍임을 의미할 수 있다. 이때, 결과값 '1'은 최상위 Bloom필터에 주소 쌍을 갱신하라는 의미를 가질 수 있다.
- [0022] 예를 들어, 스프레드 추정 시스템은 멤버십 테스트의 수행 결과로 결과값 '0'이 반환되는 경우, CSE에 해당 주소 쌍을 기록한 후, 최상위 Bloom필터에 멤버로 등록할 수 있다. 또한, 스프레드 추정 시스템은 멤버십 테스트의 수행 결과로 결과값 '1'이 반환되는 경우, 해당 주소 쌍을 최상위 Bloom필터에 멤버로 등록하는 작업만을 수행할 수 있다. 또한, 스프레드 추정 시스템은 멤버십 테스트의 수행 결과로 결과값 '2'가 반환되는 경우, 특별한 작업을 하지 않고 다음 패킷을 기다릴 수 있다.
- [0023] Bloom 필터는 주어진 원소가 어떤 집합에 속하는지 여부를 검사하는데 사용할 수 있는 자료 구조일 수 있다. Bloom 필터에서 어떤 원소가 집합에 속한다고 판단한 경우엔 틀릴 수 있으나(실제로는 집합에 속해 있지 않을 수도 있다, 잘못된 양성판정(false positive)), 원소가 집합에 속하지 않는다고 판단했다면 그 원소는 절대 집합에 속하지 않았다는 것이 보장된다.
- [0024] 최상위 Bloom필터에 등록할 수 있는 멤버의 수는 잘못된 양성판정을 고려하여 결정될 수 있다. Bloom필터의 크기를 m , 등록할 멤버의 수를 n , 서로 다른 해시 함수의 개수를 k 라고 할 때, n 개의 멤버를 등록한 후 어느 특정 비트가 0으로 남아 있을 확률은 $\left(1 - \frac{1}{m}\right)^{kn}$ 와 같이 표현될 수 있다. 이 경우, 잘못된 양성판정을 내릴 확률은 $\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx (1 - e^{-kn/m})^k$ 와 같이 표현될 수 있고, m 과 n 이 주어졌을 때, 최적화된 k 의 값은 $k \approx (\ln 2) \left(\frac{m}{n}\right)$ 와 같이 표현될 수 있다.
- [0025] 예를 들어, Bloom필터의 크기 m 이 32 킬로바이트이고, 멤버의 수 n 이 3.2 킬로바이트라고 가정한다면, 최적화된 k 의 값을 7이 될 수 있다. 이 경우, 잘못된 양성판정을 내릴 확률은 0.00819%로 계산될 수 있다. n 의 크기는 m 과 k , 그리고 잘못된 양성판정을 내릴 확률을 종합적으로 고려하여 결정될 수 있다.
- [0026] 최상위 Bloom필터에 계산된 만큼의 멤버가 입력(등록)되면 가장 오래된 Bloom필터를 비우고, 비워진 Bloom필터를 최상위 Bloom필터로 지정할 수 있다(표 4의 알고리즘 4).

표 2

Algorithm 2: MembershipTest (use Bloom Filter)

input: a contact string

Result: 0 or 1 or 2

```

1  begin
2      if front > rear                //front가 rear보다 큰 경우
3          last ← rear + N          //last에 rear + N을 대입한다.
4      else
5          last ← rear              // 그 외의 경우 last에 rear를 대입한다.
6      end if
7      for i = front to last
8          result ← Queryi mod n(src||dst) // 각 블룸필터에 대해 멤버여부를 조회한다.
9          if result is 1 then        // 멤버라고 판명된 경우
10             if i = last then       // 그 위치가 큐의 마지막이라면
11                 return 1           // 1을 반환하고
12             else
13                 return 2           // 나머지의 경우 2를 반환한다.
14             end if
15         end if
16     end for
17     return 0                       // 멤버가 아니라면 0을 반환한다.
18 end

```

[0027]

[0028] 표 2의 알고리즘 2는 멤버십 테스트 알고리즘에 관한 것으로, 매개변수로 출발지-목적지 주소 쌍을 받을 수 있다. 멤버십 테스트는 사용 중인 블룸필터 모두에 대해 이루어질 수 있다. 변수 *front*는 사용 중인 블룸필터의 최상위 위치(최상위 블룸필터)를, 변수 *rear*는 최하위 위치(최하위 블룸필터)를 가리킬 수 있다. 멤버십 테스트는 최상위 블룸필터부터 시작하여 최하위 블룸필터까지 이루어질 수 있다. 단위 블룸필터의 수행결과 멤버십이 확인되면 '1' 또는 '2'가 반환될 수 있다. '1'이 반환되는 것은, 매개변수로 입력된 출발지-목적지 주소 쌍이 큐의 마지막 블룸필터의 멤버십이 확인되었으니, 최상위 블룸필터에 주소 쌍을 갱신하라는 의미를 포함할 수 있다. '2'가 반환되는 것은 단순히 매개변수로 입력된 출발지-목적지 주소 쌍이 블룸필터의 멤버십이 확인되었다는 의미를 포함할 수 있다.

[0029] 사용 중인 블룸필터에 대해 모두 테스트를 수행했음에도 불구하고 멤버가 아닌 경우로 판정이 되었다면, 매개변수로 입력된 출발지-목적지 주소 쌍이 처음 수신된 주소 쌍인 것으로 판단될 수 있다. 이 경우에는 '0'이 반환될 수 있다. 표 1의 알고리즘 1에서 멤버십 테스트 함수(MembershipTest())의 반환값이 '0'인 경우에는 함수 *storing()*이 수행될 수 있다. 여기서, 함수 *storing()*은 표 3의 알고리즘 3과 같이 표현될 수 있다.

표 3

Algorithm 3: Storing

```

input: src, dst

1  begin
2    expand_count ← GetExpandCount(src)           // src가 몇 번 확장되었는지 조회
3     $B[H_M(src || \text{expand\_count} || (H_M(dst) \bmod s))] \leftarrow 1$ 
4    r ← GetRandom()
5    if  $0 = r \bmod s/2$  then                          //  $2/s$  확률로 확장된다.
6      AddExpand(src)                               // 확장된 src만 확장관리자에 등록.
5    end if
8  end

```

[0030]

[0031] 표 3의 알고리즘 3은 동적으로 증가하는 가상벡터(Incremental Virtual Vector, IVV)를 적용한 CSE에서 주소 쌍을 저장하는 알고리즘으로, 매개변수로 출발지 주소(*src*)와 목적지 주소(*dst*)를 입력 받을 수 있다. 본 실시예에 따른 스프레드 추정 시스템은 *src*와 *dst*로 정해지는 가상벡터의 특정 위치에 값을 기재하기 전, 확장관리자로부터 출발지 주소가 몇 번 확장되었는지 얻어올 수 있다. 예를 들어, 스프레드 추정 시스템은 알고리즘 3에서 함수 *GetExpandCount()*를 통해 출발지 주소가 몇 번 확장되었는지를 확인할 수 있다. 이때, 출발지 주소가 등록되지 않은 출발지 주소인 경우, 함수 *GetExpandCount()*는 '0'을 반환할 수 있다. 확장된 수는 가상벡터의 색인으로서 이용될 수 있다. 종래기술에 따른 CSE는 고정된 크기의 가상벡터에서 목적지 주소만을 이용해 가상벡터의 어느 한 위치를 특정 짓지만, 본 실시예에서는 색인을 이용하여 가상벡터를 2차원으로 취급한다는 점에서 차별성을 보인다.

[0032] 알고리즘 3에서 라인 '3'은 추가하려는 원소(주소 쌍)에 대해 *k*가지의 해시값을 계산한 다음, 각 해시값에 대응하는 비트를 1로 설정함으로써 주소 쌍을 저장하기 위한 과정일 수 있다.

[0033] 공유되는 SRAM의 특정 비트를 결정하기 위한 요소로서, 출발지 주소, 확장 횟수 및 목적지 주소가 이용될 수 있다. 출발지 주소와 확장 횟수는 가상벡터를 결정짓는 요소이고, 목적지 주소는 앞의 두 요소에 의해 결정된 가상벡터의 특정 위치를 결정짓는 용도로 이용될 수 있다.

[0034] 모든 출발지 주소 별로 확장된 수를 관리하는 것은 상당히 많은 양의 메모리 공간을 요구하기 때문에, 확률적으로 가상벡터의 크기가 증가되도록 설계될 수 있다. 초기 가상벡터의 크기를 기준으로 절반 정도의 공간이 사용되었을 때를 기준으로 출발지 주소가 확장될 수 있으며, 네트워크 패킷 중 고유한 주소 쌍의 수를 *c*, 초기 가상

벡터의 크기를 *s*라고 할 때, 한 목적지 주소당 기대되는 확장 횟수는 $E = \frac{2}{s} \cdot c$ 와 같이 표현될 수 있다.

표 4

Algorithm 4: RotateBFQ

```

1  begin
2    if  $front - rear = 1$  then           // 비어있는 큐가 없다면
3      Clear( $rear$ )                     //  $rear$ 가 가리키는 블룸필터를 비우고
4       $front \leftarrow rear$              // 그 자리를 최상위 블룸필터로 지정한다.
5       $rear \leftarrow rear - 1 \bmod N$    //  $rear$ 의 위치를 재조정한다.
6    else
7       $front \leftarrow front - 1 \bmod N$  //  $front$ 의 위치만 재정리한다.
8    end if
9  end

```

[0035]

[0036]

표 4의 알고리즘 4는 중복된 주소 쌍을 가려내기 위해 필터로 이용하는 블룸필터 큐를 관리하는 알고리즘으로, 표 1의 알고리즘 1에 나타난 함수 *RotateBFQ()*를 설명하고 있다. 초기에 *front*와 *rear*가 모두 최상위 블룸필터를 가리킬 때는 *front*의 수를 하나씩 감소시키며 최상위 블룸필터를 변경할 수 있다. *N*개의 블룸필터가 모두 다 사용되는 상황에서는 최하위 블룸필터를 비우고 그 자리를 최상위 블룸필터로 지정할 수 있다. *rear*는 최상위 블룸필터의 바로 이전 위치를 가리키게 함으로써 표 2의 알고리즘 2에 의해 모든 큐가 검사될 수 있도록 구성될 수 있다.

표 5

Algorithm 5: CSE with IVV (use Counting Bloom Filter)

input: a packet

```

1  begin
2       $src \leftarrow GetSourceAddress(packet)$  // 패킷으로부터 출발지 IP를 얻는다.
3       $dst \leftarrow GetDestinationAddress(packet)$  // 패킷으로부터 목적지 IP를 얻는다.
4       $result \leftarrow MembershipTest(src||dst)$  // 멤버십 테스트의 결과로 '0' 또는 '1' 반환
5      if result is 0 then // 새로운 주소 쌍이라면
6           $Storing(src, dst)$  // 가상벡터에 주소 쌍 저장
7      end if
8       $AddTop(src||dst)$  // 최상위 블룸필터에 주소 쌍을 입력한다.
9       $count \leftarrow count + 1$  // 최상위 블룸필터에 계산된 limit 만큼
10     if  $0 = count \bmod limit$  then // 멤버를 등록하면
11          $RotateBFQ()$  // 큐를 순환시킨다.
12     end if
13 end

```

[0037]

표 6

Algorithm 6: MembershipTest (use Counting Bloom Filter)

input: a contact string

Result: 0 or 1

```

1  begin
2      if front > rear                                // front가 rear보다 큰 경우
3          last ← rear + N                            // last에 rear + N을 대입한다.
4      else
5          last ← rear                                // 그 외의 경우 last에 rear를 대입한다.
6      end if
7      for i = front to last
8          result ← Queryi mod N(src||dst)           // 각 블룸필터에 대해 멤버여부를 조회한다.
9          if result is 1 then                         // 멤버라고 판명된 경우
10             Deletei mod N(src||dst)              // 데이터를 삭제하고
11             return 1                               // 즉시 1을 반환하고
12         end if
13     end for
14     return 0                                        // 멤버가 아니면 0을 반환한다.
15 end

```

[0038]

[0039]

표 5 및 표 6의 알고리즘 5 및 알고리즘 6은 카운팅 블룸필터를 이용하고, 동적으로 증가하는 가상벡터 (Incremental Virtual Vector, IVV)가 적용된 CSE를 표현한 일레이다. 표 1 및 표 2의 알고리즘 1 및 알고리즘 2와의 차이점은 멤버십 테스트를 할 때, 일반 블룸필터 대신 카운팅 블룸필터를 이용한다는 점이다. 카운팅 블룸필터는 블룸필터에 원소 삭제 명령을 지원하도록 수정한 자료 구조일 수 있다. 이 구조는 블룸필터가 1비트의 배열을 사용했던 것을 확장하여, 배열의 각 원소가 n 비트를 가질 수 있다. 원소를 추가할 때는 각 배열의 원소값을 1 증가하며, 삭제할 때에는 원소값을 1 감소할 수 있다. 따라서, 카운팅 블룸필터의 이용으로 인해 알고리즘상의 차이가 발생한다. 알고리즘 6의 멤버십 테스트 함수(MembershipTest())는 주소 쌍(출발지 주소, 목적지 주소)이 멤버라면 '1'을, 멤버가 아니면 '0'을 반환할 수 있다. 또한, 알고리즘 6은 주소 쌍이 멤버로 확인되는 경우 해당 주소 쌍을 삭제하고, 그 즉시 '1'을 반환하도록 설계될 수 있다. 알고리즘 5는 멤버십 테스트의 결과가 '0'일 경우, 함수 Storing()을 수행할 수 있다. 그리고, 멤버십 테스트의 결과와 무관하게 주소 쌍을 최상위 블룸필터에 입력할 수 있다.

[0040]

CSE를 이용하여 일정 기간 동안 SRAM에 기록된 스프레드를 추정하는 방법은 다음 수학적 식 1과 같이 표현될 수 있다.

수학식 1

$$\hat{k} = s \cdot \ln(V_m) - s \cdot \ln(V_s)$$

여기서, V_m 은 SRAM에서의 0의 비율을, V_s 는 출발지 주소가 갖는 가상벡터에서의 0의 비율을 의미할 수 있다. 본 발명의 실시예들에서는 CSE와 동일한 방식으로 스프레드를 측정하지만 하나의 출발지 주소가 여러 개의 가상 벡터들을 가질 수 있으므로, 각각의 가상벡터들에 대해 $\hat{k}$ 을 구한 뒤 이를 모두 더해야 한다. 따라서, 수학식 1은 아래 수학식 2와 같이 본 발명의 실시예들에 맞게 변형될 수 있다.

수학식 2

$$\hat{k} = \sum_{i=0}^{cnt} \{s \cdot \ln(V_m) - s \cdot \ln(V_{s_i})\}$$

여기에서 cnt 는 출발지 주소가 확장된 횟수를 의미할 수 있고, 확장관리자를 조회하여 알아낼 수 있다. 확장관리자에 등록되지 않은 출발지 주소는 확장된 횟수를 0으로 처리할 수 있다. 이 경우는 CSE와 동일한 방식으로 $\hat{k}$ 를 구해야 한다.

도 1은 본 발명의 일실시예에 있어서, 스프레드 추정 시스템의 내부 구성을 설명하기 위한 블록도이고, 도 2는 본 발명의 일실시예에 있어서, 스프레드 추정 방법을 도시한 흐름도이다. 본 실시예에 따른 스프레드 추정 시스템(100)은 출발지와 목적지 사이에서 패킷을 라우팅하는 라우터에 포함된 시스템이거나 또는 상기 라우터와 연계된 시스템일 수 있다. 이러한 스프레드 추정 시스템(100)은 도 1에 도시된 바와 같이, 주소 추출부(110), 중복 여부 판단부(120) 및 주소 쌍 추가부(130)를 포함할 수 있으며, 필요에 따라 선택적으로 스프레드 추정부(140)를 더 포함할 수도 있다. 도 2의 실시예에 따른 스프레드 추정 방법은 도 1의 실시예에 따른 스프레드 추정 시스템(100)에 의해 수행될 수 있다.

단계(210)에서 스프레드 추정 시스템(100) 또는 주소 추출부(110)는 수신된 패킷으로부터 출발지 주소 및 목적지 주소를 추출할 수 있다. 예를 들어, 상기 라우터로 수신된 패킷이 스프레드 추정 시스템(100) 또는 주소 추출부(110)로 입력될 수 있고, 입력된 패킷에서 패킷의 출발지 IP와 목적지 IP가 추출될 수 있다.

단계(220)에서 스프레드 추정 시스템(100) 또는 중복 여부 판단부(120)는 복수의 블룸필터(bloom filter)를 이용하여 추출된 출발지 주소 및 목적지 주소로 구성된 주소 쌍의 중복 여부를 판단할 수 있다.

단계(230)에서 스프레드 추정 시스템(100) 또는 주소 쌍 추가부(130)는 주소 쌍이 중복되지 않은 것으로 판단된 경우, 주소 쌍을 가상벡터에 추가할 수 있다. 이때, 수신되는 복수의 패킷들로부터 추출되는 주소 쌍들 중 상기 출발지 주소에 대해 중복되지 않은 주소 쌍의 수에 기초하여 상기 가상벡터의 크기가 동적으로 변경될 수 있다.

다른 실시예에 따르면, 스프레드 추정 시스템(100) 또는 중복 여부 판단부(120)는 단계(220)에서 복수의 블룸필터에 주소 쌍이 멤버로 등록되어 있는지 여부를 확인하여 주소 쌍의 중복 여부를 판단할 수 있다.

또 다른 실시예에 따르면, 복수의 블룸필터는, 복수의 카운팅 블룸 필터(Counting Bloom Filter)를 포함할 수 있다.

또 다른 실시예에 따르면, 스프레드 추정 방법은 도 2에 도시된 단계들 이외에, 주소 쌍을 복수의 블룸필터 중 최상위 블룸필터에 등록하는 단계(미도시)를 포함할 수 있다. 이러한 미도시 단계는 도 2에서 설명한 단계(230) 이후에 실행될 수 있으며, 스프레드 추정 시스템(100)에 의해 수행될 수 있다. 이때, 최상위 블룸필터에 대해 기설정된 수의 주소 쌍이 등록되는 경우, 복수의 블룸필터 중 최상위 블룸필터가 변경될 수 있다.

또 다른 실시예에 따르면, 상기 미도시 단계는 주소 쌍이 중복되지 않은 주소 쌍이거나 또는 복수의 블룸필터

중 마지막 블룸필터에 등록된 멤버인 경우에 주소 쌍을 최상위 블룸필터에 등록하는 단계일 수 있다.

- [0053] 또 다른 실시예에 따르면, 스프레드 추정 시스템(100) 또는 주소 쌍 추가부(130)는 단계(230)에서 출발지 주소가 확장된 횟수에 기초하여 가상벡터에 주소 쌍을 추가할 수 있다. 여기서, 상기 확장된 횟수는, 가상벡터의 크기가 출발지 주소와 연관하여 증가된 횟수를 포함할 수 있다.
- [0054] 또 다른 실시예에 따르면, 가상벡터의 크기는, 가상벡터의 초기 크기에 기초하여 결정되는 확률에 따라 증가 여부가 결정될 수 있다. 이때, 가상벡터의 크기가 증가한 경우, 출발지 주소가 확장관리자에 등록되어, 출발지 주소의 확장된 횟수가 갱신될 수 있다.
- [0055] 또 다른 실시예에 따르면, 스프레드 추정 방법은 출발지 주소가 확장된 횟수에 기초하여 출발지 주소에 대한 스프레드를 추정하는 단계(240)를 더 포함할 수 있다. 이러한 단계(240)는 스프레드 추정 시스템(100) 또는 스프레드 추정부(140)에 의해 수행될 수 있다.
- [0056] 도 1 및 도 2에서 생략된 내용은 상술한 알고리즘 1 내지 알고리즘 6을 통해 설명한 내용을 참조할 수 있다.
- [0057] 이와 같이, 본 발명의 실시예들에 따르면, 스프레드(spread)가 커질수록 가상벡터(virtual vector)의 개수를 동적으로 증가시켜 CSE(Compact Spread Estimator)의 한계를 극복할 수 있다.
- [0058] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, 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 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0059] 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, 가상장치(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.
- [0060] 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), 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. 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.
- [0061] 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 형태

로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

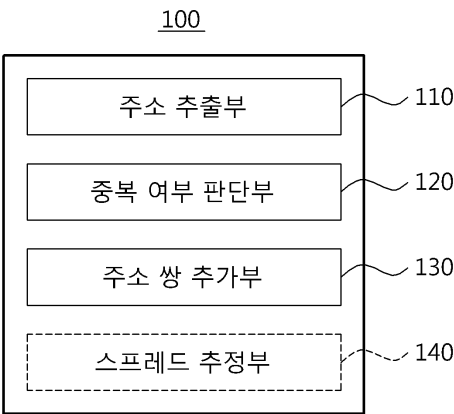
[0062] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

부호의 설명

- [0063] 100: 스프레드 추정 시스템
- 110: 주소 추출부
- 120: 중복 여부 판단부
- 130: 주소 쌍 추가부

도면

도면1



도면2

