인터넷 트래픽 분석을 위한 TCP 패킷 역전 및 중복 문제 실시간 처리 시스템

이수강, 안현민, 김명섭
고려대학교 컴퓨터정보학과

{sukanglee, queen26, tmskim}@korea.ac.kr

Realtime process system of Out-of-order and Retransmission problems for analysis Internet Traffic

Su-Kang Lee, Hyun-Min An, and Myung-Sup Kim
Dept. of Computer and Information Science, Korea Univ.

요 약

급격한 인터넷의 발전으로 효율적인 네트워크 관리를 위해 트래픽 데이터 분석의 중요성이 강조되고 있다. 발생된 트래픽 데이터를 분석하기 위해 해당 트래픽을 발생시킨 응용을 탐지할 수 있어야 한다. 응용을 탐지하기 위한 방법들 중 하나인 통계정보 트래픽 분류방법을 사용하여 트래픽을 분류할 수 있는데, 이러한 통계정보를 그대로 사용하여 분류하기에는 TCP 세션에서 발생하는 Out-of-order, Retransmission 과 같은 문제점들이 있다. 본 논문에서는 기존의 Out-of-order, Retransmission 해결 알고리즘의 문제점을 보완하고 이를 적용한 실시간 처리 시스템 모듈을 제안하고자 한다.

1. 서 론

초고속 인터넷의 보급과 다양한 인터넷 기반 응용의 등장으로 인해 트래픽이 복잡 다양해지고 있다. 이러한 상황 속에서 효과적인 네트워크의 관리를 위해 패킷을 수집하여 해당 패킷을 발생시킨 응용을 탐지할 수 있어야 한다. 트래픽 분석을 위한 응용을 탐지하기 위한 방법들 중 하나인 통계정보 트래픽 분류방법[1]은 패킷의 크기와 전송 방향, 전송 순서, 그리고 캡쳐 시간등을 사용하여 분류한다. 하지만 수집된 통계 정보를 사용하여 트래픽을 분류할 때 한계가 있는데 그것은 TCP 세션에서 발생하는 Out-of-order 와 Retransmission 문제이다.

기존 연구[2,3]을 바탕으로 학내망에서 발생된 트래픽을 분석하여 Out-of-order 와 Retransmission 문제들의 발생 빈도를 조사하고 이를 해결하기 위한 알고리즘을 제시하였다. 기존 연구에서 제시한 알고리즘은 Out-of-order 문제를 해결할 때 반대 방향 패킷을 무시하고 같은 방향 패킷의 SEQ 번호만을 기준으로 위치를 변경하였다. 이러한 방법으로는 옳겨지는 패킷과 반대 방향 패킷의 순서를 무시하고 문제를 해결하게 된다. 본 논문에서는 같은 방향의 패킷뿐만 아니라 반대 방향의 패킷또한 탐지하고 처리하는 부분을 추가하여 개선된 알고리즘을 실시간으로 처리하는 시스템에 적용시킨 모듈을 제안하고자 한다.

본 논문은 다음과 같은 순서로 구성된다. 2 장에서는 기존 연구의 문제점을 분석하고 개선시킨 Out-of-order 와 Retransmission 문제를 처리하는 알고리즘을 적용시켜 실시간으로 처리하는 모듈을 제안한다. 알고리즘과 모듈을 순서도, 의사코드와 함께 설명하고 끝으로 3 장에서는 결론 및 향후 연구에 대해 언급한다.

본 연구는 BK21 플러스 사업 및 2012 년 정부(교육과학기술부)의 재원으로 한국연구재단(2012R1A1A2007483)의 지원을 받아 수행된 결과임.

2. 본 론

본 장에서는 기존의 Out-of-order, Retransmission 문제 해결 알고리즘을 분석하고, 개선된 알고리즘을 적용시킨 실시간 처리 모듈을 제안한다.

기존 연구에서는 학내망에서 발생하는 트래픽을 수집하여 플로우가 끝나는 시점부터 해당 패킷들을 저장된 순서대로 검사하여 Out-of-order, Retransmission 문제를 해결하였다. 플로우가 끝나는 시점부터 문제를 분석하고 해결을 시작하기 때문에 실시간 처리 시스템에 적용시키기 어렵다. 또한 기존 연구의 알고리즘은 Out-of-order 문제를 해결할 때에 같은 방향의 패킷만 검사 하여 옮겨지는 패킷의 자리를 정하였다. 이러한 방법으로는 옮겨지는 패킷과 반대 방향 패킷의 순서를 무시하고 문제를 해결하게 된다. 따라서 같은 방향 뿐만 아니라 반대 방향 패킷과의 관계도 생각하여 처리해야 트래픽을 발생시킨 응용에서 보내고자 한 원래의 패킷 순서와 같아진다.

그림 1 은 Out-of-order 와 Retransmission 문제를 해결하기 위한 실시간 시스템 모듈의 순서도 이다. 입력 데이터는 수집된 1 분단위의 플로우와 패킷 데이터이다. 실시간 시스템 모듈은 플로우를 기준으로 해당 패킷들을 연결하여 만든 Flow with packet 형태의 데이터를 해시테이블에 저장한다. 해시테이블의 키값은 Flow 의 5-tuple(Source Ip, Source Port, Destination IP, Destination Port, Layer4-protocol)을 이용하여 만들어지며 해시테이블에 저장되어진 1 분 간의 트래픽 데이터들의 Out-of-order 와 Retransmission 문제를 순서대로 처리한다. 순서대로 처리된 데이터중에 세션이 종료된 플로우 데이터는 Flow with packet 파일형식으로 저장되고 해시테이블내에 남아있는 데이터를 지운다.

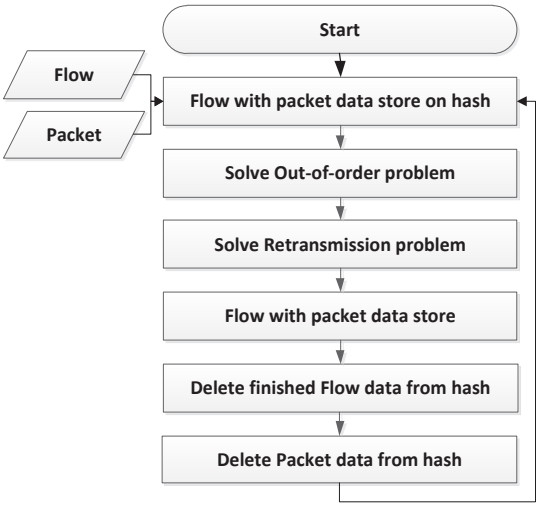

그림 1. 실시간 처리 모듈 순서도

여전히 해시테이블은 끝나지 않은 플로우의 데이터를 갖고 있으며 순서도와 같이 모듈은 1 분단위로 계속 동작하며 나머지 데이터를 실시간으로 처리하게 된다.

```
Remove all non-payload packets from the packet sequence
P(n) : n-th packet in a TCP flow
P(n).seq : n-th packet's sequence number
P(n).ack : n-th packet's acknowledge number
P(n).direction : n-th packet's direction
1 : procedure Solve Out-of-order problem
2 : Input : Packets in TCP Flow and Flow data
3 : if ( Is P(n), P(n+1) Out-of-order state? )
4 :     for ( i = n-1 to decremented by 1)
5 :       if ( P(n+1).direction == P(i).direction )
6 :         if ( P(n+1).seq >= P(i).seq )
7 :           if( P(n+1).direction != P(i+1).direction )
8 :             if( P(n+1).seq + payload length > P(i+1).ack )
9 :               then move P(n+1) back of P(i+1)
10:                 end for
11:             else move P(n+1) back of P(i)
12:   end for
13:end procedure
```

그림 2. Out-of-order 알고리즘 의사코드

그림 2 는 Out-of-order 문제를 탐지하고 해결하는 알고리즘의 의사코드 이다. 해당 알고리즘은 플로우 데이터와 페이로드가 있는 패킷만을 입력으로 받는다. P(n)과 P(n+1)이 Out-of-order state 이면 4~12 번째 줄에 의해 backtracking 을 시작한다. 6 번째 줄의 조건문에 의해 P(n+1)의 위치가 P(i) 뒤에 위치해야 한다면 7 번째 줄에 의해 P(i+1)의 방향과 비교를 한다. 만약 P(n+1)이 P(i+1)과 반대 방향일 경우 8 번째 줄의 조건문에 의해 P(i+1)의 Sequence 와 Acknowledge 번호를 비교하여 P(i+1) 뒤에 위치할지, P(i) 뒤에 위치할지를 결정하게 된다. P(n+1)이 P(i+1)뒤에 위치해야 한다면 9 번째 줄에 의해 패킷이 이동되고 프로시져는 종료된다. P(n+1)이 P(i)뒤에 위치해야 한다면 11 번째 줄에 의해 패킷이 이동되고 프로시져는 종료된다.

```
Remove all non-payload packets from the packet sequence
P(n) : n-th packet in a TCP flow
P(n).seq : n-th packet's sequence number
P(n).ack : n-th packet's acknowledge number
P(n).direction : n-th packet's direction
1 : procedure Solve Retransmission problem
2 :   Input : Ordered Packets in TCP Flow and Flow data
3 :     if ( Is P(n), P(n+1) Retransmission state? )
4 :       if ( P(n).payload len == P(n-1).payload len )
5 :         delete P(n-1)
6 :       else if ( P(n).payload len != P(n-1).payload len )
7 :         delete the Packet payload length is long
8 :     else
9 :       if ( P(n).direction == P(n+1).direction )
10 :          if ( P(n+1).seq != P(n).seq + P(n).payload
length )
11 :            delete P(n+1)
12 : end procedure
```

그림 3. Retransmission 알고리즘 의사코드

그림 3 은 Retransmission 문제를 탐지하고 해결하는 알고리즘의 의사코드 이다. P(n)과 P(n+1)이 Retransmission state 이고 두 패킷의 페이로드 길이가 같다면 4~5 번째 줄에 의해 보다 뒤에 위치한 패킷인 P(n+1)이 삭제된다. 만약 두 패킷의 페이로드 길이가 다르다면 7 번째 줄에 의해 보다 페이로드 길이가 긴 패킷이 삭제된다. 만약 P(n)과 P(n+1)번째 패킷이 Retransmission state 가 아니면서 두 패킷의 방향이 같다면 10~11 번째 줄에 의해 P(n+1)의 sequence 번호와 P(n)의 sequence 번호를 P(n)의 페이로드 길이와 합한 값을 비교하여 값이 서로 다르다면 P(n+1)이 삭제된다. 마지막 패킷까지 모두 검사를 마치면 해당 프로시져는 종료되며 입력으로 받은 1 분 플로우 내에 Out-of-order 문제와 Retransmssion 문제를 모두 해결하게 된다.

3. 결론 및 향후 연구

본 논문에서는 기존 연구에서 제시한 Out-of-order, Retransmission 해결 알고리즘을 보완하고 적용한 실시간 처리 모듈을 제안하였다.

향후 연구과제로는 트래픽 수집시 Drop 패킷의 발생원인에 대해 연구할 계획이다. 본 논문에서 제안한 실시간 처리시스템을 바탕으로 학내망에서 발생되는 패킷들의 Sequence, Acknowledge 번호를 조사하여 패킷을 수집할 때 Drop 패킷들의 발생 비율과 일어나는 원인을 조사하고 해당 문제를 해결하는 연구를 수행하고자 한다.

참 고 문 헌

[1] Ying-Dar Lina, Chun-Nan Lua, Yuan-Cheng Laib, Wei-Hao Penga and Po-Ching Lina, "Application classification using packet size distribution and port association" Proc. of the Journal of Network and Computer Applications, In Press, Corrected Proof, Available online, March, 20. 2009.

[2] 이수강, 안현민, 김명섭, "인터넷 트래픽 분석을 위한 TCP 패킷 역전 및 중복 문제 해결 알고리즘", 2013 년도 한국통신학회 추계종합학술발표회, 연세대학교, 서울, Nov. 30, 2013, pp.553-554.

[3] 안현민, 최지혁, 함재현, 김명섭, "TCP 세션의 이상동작으로 인한 트래픽 분석 방법론의 한계와 해결방안, KNOM Review, Vol. 15, No. 1, Dec. 2012, pp. 31-39.