

멀티에이전트를 이용한 인터넷 트래픽 측정

김대은*, 이정환*, 홍충선*, 최태상**

*경희대학교 컴퓨터공학과

**한국전자통신연구원

e-mail:vamper@networking.kyunghee.ac.kr

Internet Traffic Measurement Using Multi-agent

Dae-Eun Kim*, Jung-Hwan Lee*, Chongng-Seon Hong*, Tae-Sang Choi**

*Dept of Computer Engineering, Kyung-Hee University

**Electronics and Telecommunications Research Institute

요 약

인터넷의 급속한 성장과 더불어 네트워크 관련 기술은 발달하였고 이에 따라 사용자는 시간과 공간적 제약을 벗어나 인터넷을 활용하고 있다. 급증하는 사용자와 함께 인터넷 트래픽도 급격히 증가하였고 이에 따라 국가 간, ISP 간 증설되는 회선의 비용과 관리의 비용 부담의 문제가 중요한 문제로 부각되었다. 공정한 요금의 부담을 위해서는 인터넷 사용량에 근거한 상호 과금 분담 체계를 이용해야 하지만 현재까지의 시스템으로는 트래픽의 정확한 측정과 평가를 위한 과금 시스템과 과금 분담에 대한 표준화 등이 존재하지 않는다.

따라서 본 논문에서는 이러한 불공정한 인터넷 사용환경을 개선할 수 있는 트래픽 측정 시스템에 대해서 연구하였으며 트래픽 측정 에이전트간의 역할 분담 및 에이전트가 수집하고 평가해야할 트래픽을 분류하였다. 그리고 장애가 발생하여 자신의 임무를 수행할 수 없는 에이전트를 대신하여 임무를 대행할 수 있는 임무 대체 기법에 대해서도 연구하였으며 효율적인 트래픽 수집 과정을 위하여 적응형 순환 큐를 적용시켜 트래픽 수집 에이전트의 효율적인 메모리 관리 방법을 제시함으로써 트래픽 측정 에이전트의 운용 효율성을 개선하였다.

1. 서론

인터넷의 사용량 및 사용인구는 급격히 증가하였으며 그 이용 범위 역시 기업, 대학과 연구소 중심에서 개인으로 확대되었다. 또한 인터넷 관련 네트워크기술의 발달로 시간과 공간의 제약 없이 인터넷을 사용하고 있으며, 그에 따라 발생하는 인터넷의 트래픽은 지수적으로 증가하고 있는 추세이다. 이에 따라 인터넷의 관리 및 국제회선의 증설에 들어가는 비용 역시 증가하고 있으며 이에 소요되는 비용에 대한 공정한 부담 체계가 각 국가 간, ISP 간의 해결해야 하는 중요한 문제로 대두되고 있다. 그러나 트래픽 측정 및 통계적 데이터 산출을 위한 시스템과 과금(課金) 체계가 존재하지 않기 때문에 비대칭적이며 불공정한 인터넷 국제회선 비용 산정 관행이 계속되고 있다. 이러한 문제를 해결하기 위하여 현재 각 국가들은 트래픽 측정 도구 및 과금 체계에 대한 기술적 해결방안을 계속 모색하고 있다.

또한 인터넷 트래픽을 측정하면 과금 뿐만 아니라

측정 결과를 통한 인터넷 성능을 직접적으로 파악하여 제어를 수행할 수 있는 수단을 제공하며, 측정된 자료의 통계적 분석을 통하여 네트워크 관리, 네트워크 설계 및 증설, 과금 정책 등에 사용될 수 있는 매우 핵심적인 기술이다.

본 논문에서는 MataMon(Multi-agent based Traffic Accounting Monitor)라 불리는 다수의 수동적 트래픽 측정 에이전트를 이용하여 비대칭적인 인터넷 사용환경 개선을 위해 효율적으로 트래픽을 측정하고 평가 할 수 있는 시스템을 설계 및 구현하였으며, 특정 에이전트의 장애(Fault) 발생 시 다른 에이전트가 임무를 대신하여 수행할 수 있는 대체 기법도 제안하였다.

2. 연구와 관련된 트래픽 측정도구

2.1 CoralReef

CoralReef[1]는 CAIDA(Cooperative Association for Internet Data Analysis)에서 개발된 수동적인

인터넷 트래픽 모니터로서 데이터를 실시간 혹은 트레이스 파일들을 이용해서 수집하고 평가하는 소프트웨어 패키지이다. CoralReef는 링크 레벨에서 분리기(Splitter)를 이용하여 트래픽을 복사하고 그 복사된 트래픽을 분석하는 구조를 가지고 있으며 ATM과 OC3MON, OC12MON을 지원한다. 또한 CoralReef는 지원할 수 있는 하드웨어 드라이버, 라이브러리, 다양한 유틸리티, 그리고 분석 어플리케이션을 소프트웨어 패키지로 제공한다. CoralReef는 디바이스 드라이버와 libcoral 라이브러리 C와 Perl로 이루어진 분석 모듈을 제공하며 보고서 생성은 HTML 생성 모듈을 이용하며 전체적인 컴포넌트 구성은 그림 1과 같다.

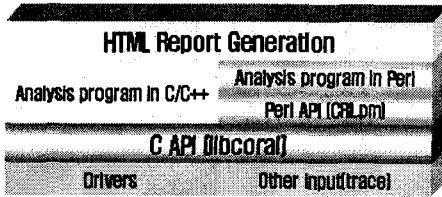


그림 1. CoralReef 소프트웨어 컴포넌트

2.2 NTraMet

NeTraMet[2]은 Network Traffic Meter의 합성어로 IETF RTFM (Realtime Traffic Flow Measurement) WG[3]에서 제안하고 오픈 소스로 구현한 소프트웨어 패키지이다.

RTFM WG에서 제안하고 구현한 NeTraMet은 그림 2와 같이 구성되며 Manager, Meter, Meter Reader, Analysis Application 이렇게 총 4개의 객체로 구성된다[4].

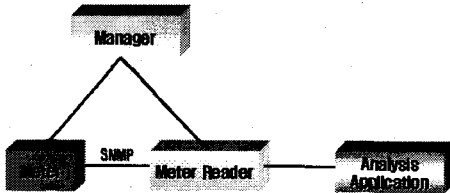


그림 2. NeTraMet의 구조

① Meter : 트래픽을 측정하고자 하는 대상, 즉 라우터나 호스트에 연결하여 트래픽에 대한 정보를 수집하는 역할을 한다.

② Meter Reader : Meter에서 수집된 패킷 정보를 SNMP를 이용하여 읽어 들인 후 통계적 데이터 산출을 위해 저장한다.

③ Manager : Meter와 Meter Reader의 운영 환경을 설정하고 조정하는 역할을 한다.

④ Analysis Application : Meter Reader로부터 수집된 패킷 정보를 가져와 분석하여 트래픽의 종류와 양에 따른 보고서를 작성한다.

3. MataMon의 설계와 구현

3.1 제안동기

앞장에서 설명한 CoralReef는 네트워크의 백본(Backbone) 라우터에 단지 하나의 시스템만이 적용되어 트래픽 수집과 분석하는 구조이며 NeTraMet의 경우에는 여러 개의 Meter가 네트워크에 적용될 수는 있으나 이들의 관계 및 임무설정에 대한 구조는 정의되어 있지 않아서 다수의 Meter가 적용되었다 할지라도 실상 CoralReef처럼 하나의 Meter가 트래픽 수집과 분석에 대한 모든 프로세스를 처리해야 하는 구조를 가지고 있다. 따라서 이들의 구조는 WAN과 같이 규모가 커서 다수의 라우터가 존재하고 이들의 트래픽을 모두 측정해야 하는 경우에는 트래픽 측정 에이전트의 적용에 있어 역할 분담과 각 에이전트의 관계 설정 및 에이전트의 관리에 있어 적합하지 않다고 할 수 있다.

따라서 본 논문에서는 다음과 같은 의문점을 제시해 보았다.

첫째, 하나의 트래픽 측정 에이전트가 전체 네트워크의 모든 트래픽을 처리하는 것이 가능한가?

둘째, 만약에 다수의 트래픽 측정 에이전트가 네트워크에 존재한다면 MAS(Multi-agent System)의 개념을 적용할 수 있지 않을까?

셋째, 다수의 트래픽 측정 에이전트가 존재한다면 이들을 어떻게 관리해야 하는가?

넷째, 만약에 트래픽 측정 에이전트에 장애(Fault)가 발생한다면 대처 방안은 무엇인가?

본 논문에서는 기본적으로 MAS(Multi-agent System)의 개념을 적용해 에이전트간의 상호관계 설정 및 임무 분담에 대한 규칙 그리고 특정 에이전트의 장애(Fault) 발생 시 다른 에이전트가 임무를 대신하여 수행할 수 있는 대체 기법에 대해 중점적으로 연구하였다.

3.2 MAS 개념을 적용시킨 트래픽 측정 에이전트 설계 및 구현

트래픽 측정 에이전트는 크게 두 개의 프로세스로 나뉘어진다. 첫째는 오직 패킷만을 수집하는 프로세스이고 다른 하나는 패킷을 수집하는 프로세스를 관리하고 트래픽 측정 에이전트가 구현된 시스템을 관리하는 프로세스이다.

본질적으로 MAS는 다수의 에이전트가 존재하는 시스템을 의미한다. 이들 에이전트는 수행해야 할 임무를 지닌 소프트웨어 혹은 하드웨어 기반의 자율적

인 시스템을 의미한다. 시스템의 특성은 표 1과 같이 크게 네 가지 정도를 들 수 있다.

표 1. MAS의 특성

| MAS(Multi-agent System) | |
|-------------------------|--|
| 항 목 | 첫 째, 다수가 아닌 하나의 에이전트만으로는 해결하고자하는 문제를 풀 수 없다. (에이전트 상호간의 협력이 필요하다.) |
| | 둘 째, 에이전트들을 통제할 수 있는 컨트롤 시스템은 따로 만들지 않는다. |
| | 셋 째, 시스템에서 관리하는 데이터는 한 에이전트에만 집중되어 있지 않다. 즉 모든 에이전트에게 분산되어 있다 |
| | 넷 째, 각 에이전트는 동일한 임무를 수행하지 않는다. 따라서 각 에이전트는 자신의 특성에 맞는 비동기적 연산을 수행한다. |

트래픽 측정 에이전트는 초기 실행 과정에서 데이터베이스에 있는 다른 에이전트의 정보를 읽어오며 읽어온 데이터는 시스템에서 리스트의 형태로 관리되며 시스템 사용자가 정한 시간 단위로 다른 인접 에이전트들과 통신하여 업데이트 한다. 그림 3에서는 본 논문에서 제안하는 MataMon이 사용하는 에이전트 관리 구조를 보여주고 있다.

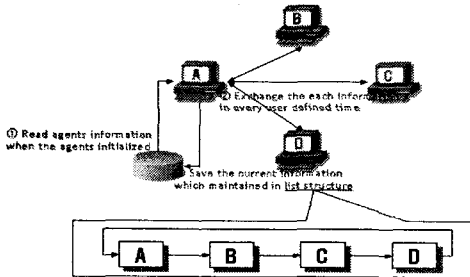


그림 3. MataMon에서 사용한 에이전트 관리 구조

각 에이전트는 자신의 리소스 사용현황을 주기적으로 다른 에이전트와 주고받으며 최종적으로 가장 리소스의 여유가 있는 에이전트가 로컬 네트워크의 트래픽이 아닌 전체적인 네트워크 트래픽에 대한 통계 데이터를 수집하고 다른 에이전트들이 출력해야 할 보고서를 통합적으로 출력함으로써 전체적인 시스템의 부하를 줄이도록 한다.

또한 주기적인 에이전트 리스트 업데이트를 통하여 현재 어떤 에이전트의 상태가 활성화되어 있는지 비활성화되어 있는지를 파악할 수 있다. 에이전트가 비활성화 되어 있는 경우라면 에이전트에 결함(Fault)이 발생하여 트래픽 측정이 불가능한 상태이므로 이를 인지한 다른 에이전트가 결함이 발생하여 비활성화된 에이전트를 대신해 패킷 수집 및 측정의 임무를 수행할 수 있다.

3.3 트래픽 수집기의 설계 및 구현

3.3.1 트래픽 수집기의 메모리(버퍼) 관리 기법

트래픽 수집기는 이더넷 카드를 통하여 패킷을 수집하는 라이브러리는 pcap 라이브러리[5]를 이용하였다. 수집된 패킷은 바로 데이터베이스에 저장되는 것이 아니라 잠시 버퍼에 저장되었다가 디코딩(Decoding) 과정을 거치게 되는데 이 과정에서 발생할 수 있는 패킷 수집의 오버헤드를 고려하여 패킷 수집 프로세스는 독립적인 프로세스로 구성하였고 네트워크의 상태에 따라 크기를 조절할 수 있는 적응형 순환 큐(Adaptive Circular Queue)의 구조를 적용하였다. 적응형 순환 큐란 수집되는 패킷의 증가율이 높을 경우에는 설정해 놓은 버퍼를 넘어서는 버퍼 오버플로 현상을 막을 수 있으며 또 지나치게 크게 설정해 놓은 큐의 크기에 의해 시스템 내에서 낭비되는 메모리 공간을 줄일 수 있다는 것이 장점이다. 적응형 순환 큐는 그림 4와 같은 구조이다.

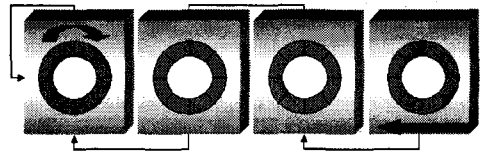


그림 4. 트래픽 수집기에 적용된 적응형 순환 큐 구조

3.3.2 트래픽 수집기의 플로우 분류 방법

CoralReef와 NeTraMet은 자신이 연결된 라우터나 호스트를 지나치는 모든 트래픽을 수집하고 평가한다. 그러나 과금은 모든 패킷에 대해서 적용가능하나 해당하는 네트워크의 근원지 주소와 목적지 주소가 해당할 경우에만 과금을 부과하는 것이 우선시 되어야 한다. 만일 모든 패킷을 패킷 수집기가 수집하여 디코딩 과정을 거친 후 데이터베이스에 정보를 유지하게 된다면 수집기가 평가하고 저장해야 하는 데이터의 양이 상당히 늘어날 것이다. 따라서 트래픽 수집기는 자신의 과금 정책에 맞는 패킷만을 수집하여 평가해야 할 것이다.

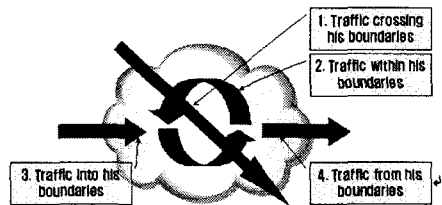


그림 5. 트래픽의 분류

본 논문에서는 그림 5와 같이 트래픽을 네 종류로 구분[6]하였고, 위에서 언급한 것과 같이 3번과 4번 타입의 트래픽에 과금을 부과하는 것을 우선하여 설계 및 구현하였다.

4. 성능평가

본 장에서는 제안한 실제 시스템을 구현하여 제안 사항에 대한 성능평가를 수행하도록 하였다.

첫 째, CoralReef와 NeTraMet이 트래픽을 수집하는 방법으로 프로그램을 작성하여 모든 패킷을 수집하고 디코딩 하여 데이터베이스에 저장하였다. 그리고 본 논문에서 제안한 적응형 순환 큐를 적용하고 트래픽 수집기가 지정된 주소들과 일치하는 패킷만을 수집하고 디코딩 하여 데이터베이스에 저장할 경우를 비교하여 시스템의 리소스를 비교하였다.

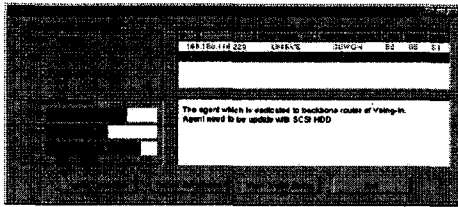


그림 5. 제안한 구조의 시스템 리소스 사용량

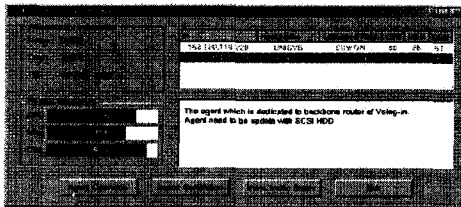


그림 6. 기존 구조의 시스템 리소스 사용량

성능평가는 동일한 환경에서 불필요한 프로세스들은 모두 종료하여 실행시킨 후 리소스를 측정된 결과이며 시스템의 리소스는 그림 5와 그림 6과 같이 현저하게 수치상 차이를 보였다.

둘 째, 네트워크와 각 트래픽 측정 에이전트의 위상을 구성하고 트래픽을 수집하다 하위 위상에 존재하는 트래픽 측정 에이전트의 프로세스를 강제로 종료시킨 후 상위 위상에 존재하는 트래픽 에이전트가 트래픽 수집을 위한 다른 프로세스를 생성하여 수집하는 시나리오를 적용시킨 후 실제 수집된 패킷과 그렇지 않았을 때의 패킷의 개수를 비교해 보았다.

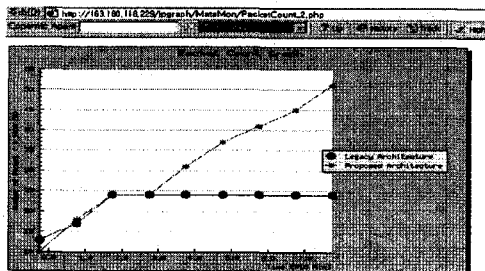


그림 7. 상위 에이전트에 의해 평가된 패킷

그림 7과 같이 기존의 시스템은 약 25,000개의 패킷을 수집한 후 종료하였으나 제안된 시스템은 에이전트 메시지 교환을 통하여 인접 에이전트가 결함이 발생하여 프로세스가 종료된 것을 인지하고 자신이 하위 트래픽 측정 에이전트의 연산을 대행하여 약 10분 후(에이전트 리소스 정보 교환 작업 수행 후)에는 다시 측정 데이터의 증가가 나타나게 된다.

5. 결론 및 향후 연구과제

본 논문에서는 비대칭적인 인터넷 사용 환경의 개선을 위해 정확하고 효율적인 트래픽 측정을 할 수 있는 시스템의 설계 및 구현에 대해서 연구하였다. 기존의 CoralReef와 NeTraMet과 같이 단순히 하나의 트래픽 측정 에이전트를 네트워크에 적용시킨 것과 같은 구조가 아니라 다수의 트래픽 측정 에이전트를 네트워크에 적용하여 서로 협력적인 문제해결 방법으로 보다 더 정확한 트래픽을 측정하는 MAS(Multi-agent System) 기반의 트래픽 측정 시스템을 설계 및 구현하였다. 또한 하위 위상에 위치하는 에이전트에 장애 발생시 상위 에이전트가 대신 패킷 수집 업무를 수행하게 하는 대체 에이전트의 작동 구조에 대해서 제안하였다. 또한 트래픽 수집기의 메모리 관리를 위해서는 적응형 순환 큐의 구조를 적용하였고, 그림 5에서 나타난 바와 같이 트래픽의 모델을 네 가지로 분류하여 트래픽 수집기가 모든 패킷을 수집 및 평가하여 발생할 수 있는 트래픽 측정 시스템의 오버헤드를 줄였다.

현재 MataMon은 시스템의 효율성 및 안정성의 테스트를 위해서 트래픽 생성기(SmartBit)[7]를 이용하여 시스템의 스트레스 테스트를 수행할 예정이며 실제 과금을 위해서는 어떠한 인자들을 고려해야 하는지 또 그들을 시스템에 어떻게 적용시켜야 하는지에 대해서 연구할 계획이다.

참 고 문 헌

- [1] CoralReef, <http://www.caida.org/tools/measurement/coralreef/>
- [2] NeTraMet Release Note, <http://www.knosof.co.uk/posix.html>
- [3] RTFM WG, <http://www.ietf.org/proceedings/96mar/charters/rtfm-charter.html>
- [4] N. Brownlee, "Using NeTraMet for Production Traffic Measurement", Integrated Management Strategies for the New Millenium, 2001
- [5] pcap library, <http://www.tcpdump.org/pcap.htm>
- [6] C. Mills, D. Hirsh, G.R. Ruth., "Internet Accounting : Background", RFC 1272, Nov 1991
- [7] SmartBits, <http://www.sprentcom.com/documents/611.pdf>