

데이터 엔지니어링 방법론을 기반으로한 네트워크 트래픽 분석 시스템

한영신¹ · 김태규² · 정재은^{1†} · 정찬기³ · 이철기⁴

Network Traffic Analysis System Based on Data Engineering Methodology

Young-Shin Han · Tae-kyu Kim · Jason J. Jung · Chan-Ki Jung · Chil-Gee Lee

ABSTRACT

Currently network users, especially the number of internet users, increase rapidly. Also, high quality of service is required and this requirement results a sudden network traffic increment. As a result, an efficient management system for huge network traffic becomes an important issue. Ontology/data engineering based context awareness using the System Entity Structure (SES) concepts enables network administrators to access traffic data easily and efficiently. The network traffic analysis system, which is studied in this paper, is designed and implemented based on a model and simulation using data engineering methodology to be available in evaluating large network traffic data. Extensible Markup Language (XML) is used for metadata language in this system. The information which is extracted from the network traffic analysis system could be modeled and simulated in Discrete Event Simulation (DEVS) methodology for further works such as post simulation evaluation, web services, and etc.

Key words : Ontology, Data engineering, System Entity Structure(SES), Network traffic analysis, and Discrete Event Simulation(DEVS)

요 약

현재 네트워크 사용자 특히 인터넷 사용자 증가 속도는 가히 기하급수적이라 할 수 있다. 더불어 질 높은 서비스에 대한 요구가 제기되고 있는데 이것은 필연적으로 트래픽의 폭증을 가져오고 있다. 따라서 네트워크 트래픽의 효율적 분석 관리는 과거에 비해 더욱 중요한 사항으로 대두되고 있다. SES를 이용한 네트워크 분석은 네트워크 관리자들에 쉽고, 효율적으로 트래픽데이터를 접근이 가능하다. 따라서 본 연구의 목적은 대용량의 네트워크 트래픽 데이터 효과적으로 처리하기 위해 최적화된 네트워크 트래픽 시스템을 구현하기 위하여 데이터 엔지니어링 방법론인 SES를 이용하여 네트워크 패킷의 정보 구조를 설계한후 DEVS를 이용하여 분석하는 시스템을 구현하고자 한다.

주요어 : 온톨로지, 데이터 엔지니어링, System Entity Structure(SES), 네트워크 트래픽 분석, Discrete Event Simulation (DEVS)

1. 서 론

지난 몇 십 년간 지속적인 전산화를 해오고 있는 기업과 공공기관은 방대한 데이터를 가지고 있지만 대부분 단편적이고 부서별 정보에 불과하다. 따라서 해당 기관의 정책 결정자에게 정작 필요한 정보는 여러 단계의 정보 가

공을 거치거나 아니면 어떤 경우에는 아예 얻지 못할 수도 있다. 그러나 급격히 변화하는 기업 환경 속에서 정책 결정자들은 더 빠르고 풍부한 정보를 필요로 하게 되었다. 현재 네트워크 사용자 특히 인터넷 사용자 증가 속도는 가히 기하급수적이라 할 수 있다. 더불어 질 높은 서비스에 대한 요구가 제기되고 있는데 이것은 필연적으로 트래픽의 폭증을 가져오고 있다 따라서 네트워크 트래픽의 효율적 분석 관리는 과거에 비해 더욱 중요한 사항으로 대두되고 있다. 네트워크 분석은 네트워크 내의 모든 통신 요소를 관찰, 제어하고 나아가 네트워크 변화에 응용함으로써 안전하고 효율적으로 네트워크가 운용될 수 있도록

2008년 11월 24일 접수, 2009년 3월 22일 채택

¹⁾ 영남대학교 전자정보공학부

주 저 자 : 한영신

교신저자 : 정재은

E-mail: j2jung@gmail.com

하는 모든 행위를 일컫는다. 네트워크 관리자는 일관된 트래픽 분석 관리 시스템의 부재로 인해, 트래픽 관리에 필요한 정보를 얻으려 할 때 원시 데이터를 여러 도구를 거쳐 가공해야만 한다. 그리고 체계적으로 오랜 시간에 걸쳐 관리된 트래픽 정보의 부족은 장단기 네트워크 용량 설계를 위한 정보의 빈곤을 야기 시킨다^[1].

복잡한 시스템 명세사항의 디자인과 시뮬레이션을 위한 모델링 연구방법에는 개체 관계모델(ER)^[2]과 표준화 모델링 언어(UML)^[3] 모델링방법이 있다. 개체 관계 모델(ER)은 데이터 개체들과 그들의 관계들과 같은 시스템의 구조를 나타내는 연구방법이고 표준화 모델링 언어(UML)는 객체와 그들의 관계들을 나타내기 위해 디자인한다. 비록 ER과 UML은 시스템의 명세사항의 모델링과 여러 종류의 논리적 모델의 묘사가 가능하다고 할지라도, 각 연구방법은 명확한 모델링에 제약을 받는다. 즉, 개체 관계 모델(ER)은 논리적 모델 표현에 유용하지만 시각적 모델링에서는 실용적이지 못하다. 또한 UML은 논리적 모델과 시각적 모델에서 알맞았지만 모델 지속성이 부족한 문제가 있었다^[4]. SES(System Entity Structure)는 데이터 처리 개념과 온톨로지를 기반으로 다중레벨(상위의 application과 하위 레벨 데이터 명세사항) 모델링과 시뮬레이션에 유용한 XML 모델링 구조에 의해 표현된다. 따라서 본 연구에서 제시하고 있는 SES 데이터 엔지니어링 방법론은 ER이나 UML보다 모델 지속성이 좋고 시각적, 논리적으로 표현이 가능한 계층적인 트리 구조이기 때문에 쉽고 빠른 정보 수집, 빠른 응답시간, 사용자 친화적 스키마이다. 이 연구에서는 데이터 처리기반의 모델과 시뮬레이션이 데이터 모델을 사용하여 탄력성 있게 적용할 수 있는 데이터 교환을 적용한다. 이 논문에서는 대용량의 네트워크 트래픽 데이터를 효과적으로 처리하기 위해 최적화된 네트워크 트래픽 시스템을 구현하기 위하여 데이터 엔지니어링 방법론인 System Entity Structure(SES)를 이용하여 네트워크 패킷의 정보 구조를 설계한 후 DEVS를 이용하여 분석하는 시스템을 구현하고자 한다.

2. 관련연구

2.1 Discrete Event System Specification (DEVS)

본 연구에서 활용될 모델링 및 시뮬레이션 이론은 Discrete Event System Specification(DEVS)형식론이다. DEVS 형식론^[5]은 본래 이산 사건 시뮬레이션을 위한 방법론으로 출발하였으나, 최근 들어 DESS(Differential Equa-

tion Specified Systems)와 연결된 방법론인 DEV&DESS 형식론이 등장하였다. DESS 형식론은 주로 동일시간 간격으로 샘플링 한 정보를 기반으로 시뮬레이션을 수행하는 형식론으로 주로 자연 현상을 모델링할 때 많이 사용된다. 본 자료에서는 이산 사건 시뮬레이션을 위한 고전적인 DEVS 형식론을 주로 설명하겠다. DEVS 형식론에서는 다음 2가지 모델을 통해서 이산사건 계의 시스템을 표현하게 된다.

첫째는 basic 모델로서 시스템의 행위적 특성을 표현하는데 사용되는 모델 프레임이다. Basic 모델은 다음과 같이 정의된다.

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

단,

- X : 입력 사건의 집합
- S : 순차적 상태의 집합
- Y : 출력 사건의 집합
- $\delta_{int} : S \rightarrow S$: 내부 전이 함수
- $\delta_{ext} : Q \times X \rightarrow S$: 외부 전이 함수
- $\lambda : S \rightarrow Y$: 출력 함수
- $t_a : S \rightarrow R+0 \rightarrow \infty$: 시간 진행 함수

단, $Q = \{(s,e) \mid s \in S, 0 \leq e \leq t_a(s)\}$
 et_a : 최근의 상태 전이 이후로 흐른 시간.

둘째는 coupled 모델로서 시스템을 구성하는 서브컴포넌트를 표현하고 이들 사이의 관계를 정의하는 모델이다. Coupled 모델은 다음과 같이 정의된다.

$$DN = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle$$

단,

- D : coupled 모델의 구성요소 모델 i에 대한 이름의 집합
- M_i : 구성요소가 되는 basic 모델
- I_i : 모델 i의 영향을 받는 모델들의 집합
- $Z_{i,j} : I_i$ 의 원소 각 j에 대해서 i에서 j로의 출력 번역 함수
- Select : 타이 브레이킹 함수

Basic 모델과 Coupled 모델을 통해서 대부분의 시스템의 특성의 표현이 가능하다. 그러나, 시스템의 구조적인 특성을 표현하는데 있어서 Coupled 모델이 갖는 한계가 존재한다. 이것은 시뮬레이션 환경이기 때문에 존재하는 문제로서, 시스템을 구성하는 컴포넌트가 될 수 있는 후

보 모델에 대한 정의와 이들 중 시뮬레이션을 수행하기 위한 선택이 가능한 구성이 필요하게 된다. 이를 위해서 정의된 모델링 이론이 System Entity Structure(SES)^[6]이다.

2.2 System Entity Structure(SES)/Pruned Entity Structure(PES)

System Entity Structure(SES)는 시스템의 구조를 나타내는 지식을 특정 형식으로 표현한 것으로서 Zeigler가 제안한 개념이다. 모델들의 분할(decomposition), 분류(taxonomy)와 모델간의 연결 관계에 대한 정보를 가지고 있다. System Entity Structure(SES)는 트리 형태로 계층적인 모델들을 정의하고, 트리의 말단은 모델 베이스의 모델로 나타낸다. 그림 1에서 System Entity Structure(SES) 안에는 시스템의 구조를 표현하기 위한 지식으로 entity, aspect, specialization의 3가지 형태의 노드가 있다. Entity 노드는 실세계의 한 객체와 대응되며 aspect 또는 specialization을 자식으로 가질 수 있다.

일반적으로 시스템은 System Entity Structure(SES)의 서브-구조(sub-structure)로 표현된다. 이 서브-구조를 Pruned Entity Structure (PES)라고 하며 System Entity Structure (SES)에서 PES를 얻어내는 작업을 전정(pruning)이라고 한다. PES에는 오직 entity와 aspect만이 존재한다. 즉, specialization 노드의 자식 노드는 전정 과정에서 하나의 노드만 선택되고 나머지 specialization 노드는 사라진다. 즉 PES는 합성트리로 변환할 수 있으며, 결국 시뮬레이션 모델로 합성할 수 있다.

그림 1은 SES와 XML로 적용의 개념적인 관계에 대한 SES 기본 방법론을 나타낸다. 우선 SES는 원본 데이터 내부의 컴포넌트를 묘사할 수 있도록 개발되었고, SES

구조는 DTD나 스키마, 요소, 양상, 다중양상(Multi-Aspect), 한정화를 만들기 위한 중요한 정보가 되며, DTD나 스키마의 주요 컴퍼넌트로 만들어진다.

온톨로지 레벨에서 Modeler는 하나 또는 그 이상의 모델에 의존하는 SES를 개발할 수 있고, SES는 주어진 어플리케이션 도메인 내 관심사의 실용적인 프레임을 만족하는 온톨로지를 생성하기 위해 결합한다. SES 다양한 방법으로 열거를 할 수 있고, 이를 적용 레벨의 XML 스키마 또는 XML DTD(XSD or DTD)로 변환할 수 있고, 가지치기 연산을 통해서 SES로 PES를 생성하고, 시뮬레이션 모델로 변환한다.

2.3 Protégé

Protégé는 Stanford 대학에서 개발한 자바기반의 오픈 소스 온톨로지 편집기로서 클래스와 속성의 정의 및 관계 설정, 속성 제약, 인스턴스 생성, 질의 기능을 제공하며 시각화나 추론과 관련된 다양한 종류의 플러그인을 제공한다. 또한, 개발자, 대학, 정부, 기업 등으로 이루어진 강력한 커뮤니티에 의해 지원을 받고 있으므로 꾸준한 기능 업데이트를 보장받을 수 있다. 1999년 Protégé-2000이 발표되었고, 그 후 Protégé 3.1(2005), Protégé 3.2 beta (2006)가 지원되고 있으며, 2006년 현재 등록된 사용자만 도 약 6만여 명에 달한다^[8].

3. 네트워크 트래픽 분석

본 연구에서 제시하고 있는 네트워크 트래픽 분석의 목표는 네트워크 관리자가 복잡한 네트워크 토폴로지를 관리할 수 있게 도와주며, 보안과 효과적인 데이터 전송에 대한 효율을 증가시키는 것이다. 네트워크 사용, 특히 인터넷 사용자의 수는 빠르게 증가하고 있으며, 고품질의 서비스를 요구함에 따라 네트워크 트래픽의 갑작스런 증가를 가져온다. 따라서 거대 네트워크 트래픽을 위한 효과적인 관리 시스템이 중요한 이슈가 되고 있다.

데이터 엔지니어링방법론은 네트워크 트래픽을 다루기 위한 효과적인 시스템을 만들 수 있다. SES와 XML 모델링은 이러한 시스템이 거대한 데이터를 다루기 쉽게 하고 모델링과 시뮬레이션에 도움을 주는 방향으로 접근이 가능한데, 그 이유는 SES의 구조가 계층적 트리 구조이며, 이러한 계층적 구조는 빠른 접근이 가능한 구조이기 때문이다. 게다가 XML의 확장성과 이식성과 같은 특징이 메타데이터를 관리하는 데 유용하기 때문이다. 본 논문에서 제시하고 있는 데이터 엔지니어링 방법론을 기반으로 SES

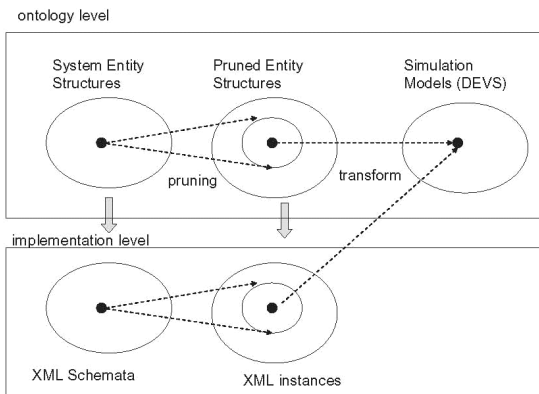


그림 1. 데이터 엔지니어링 방법론에 기반한 모델 구조^[7]

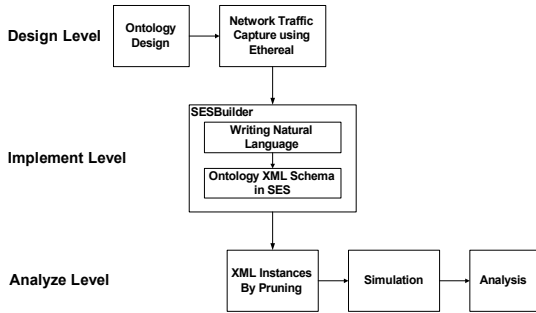


그림 2. 네트워크 트래픽 분석 시스템 모델 플로 차트

를 활용한 네트워크 트래픽 분석 시스템 모델 플로차트는 그림 2와 같이 표현 될 수 있다.

그림 2는 데이터 엔지니어링 방법론을 기반으로 온톨로지 System Entity Structure(SES)를 활용한 모델링 및 시뮬레이션 단계를 보여준다. 디자인단계, 구현단계, 분석 단계의 3단계로 분류한다. 디자인단계에서는 네트워크 트래픽 활동을 온톨로지 디자인하고 트래픽데이터를 캡처한다. 구현단계에서는 SES빌더를 사용하여 시스템에서 필요한 기본적인 트래픽 데이터를 텍스트 파일로 받고 Ontology XML Schema in SES를 생성한다. 분석 단계에서는 네트워크 트래픽 데이터를 XML in PES로 이식하고 어플리케이션 이식을 이행(Implement)하여 완전한 XML 메타데이터 파일을 생성한다. 분석단계에서는 PES를 이용하여 DEVS 모델링 과 시뮬레이션을 통해 시뮬레이션 결과를 분석한다. 본 연구에서 제안된 방법론은 DEVS-Java 시뮬레이션 환경에서 테스트 및 검증을 하였다. DEVS-Java는 DEVS이론을 Java 언어로 개발한 환경으로 현재까지의 DEVS 기반의 언어 중 가장 많이 사용되고 있는 환경이다. DEVS 이론에서 제공하고 있는 실험 프레임워크 개념을 사용하여, 다중 시스템공격과 침입의 패턴을 찾아내고 예산을 줄이며 안정적인 네트워크보안을 유지시키는 빠르고 효율적인 평가를 제시했다.

개발되는 모델의 특성은 계층적인 구조를 갖는다는 것이다. 이러한 특성을 고려할 때 네트워크 트래픽 모델은 그림 3과 같은 형태를 갖게 될 것이다.

네트워크 트래픽을 분석하기 위한 네트워크 패킷에 포함된 정보를 계층적으로 분류 및 구조화 시킨 SES로서 최상위 컴포넌트인 NetworkTrafficAnalysis는 호스트 컴퓨터에서의 분석을 위한 HostBaseAnalysis와 네트워크 망에서의 분석을 위한 NetworkTrafficAnalysis로 구성 된다. 본 논문에서는 호스트에서의 분석에 목적을 두었기 때문에 NetworkBaseAnalysis는 세부적으로 발전시키지 않았

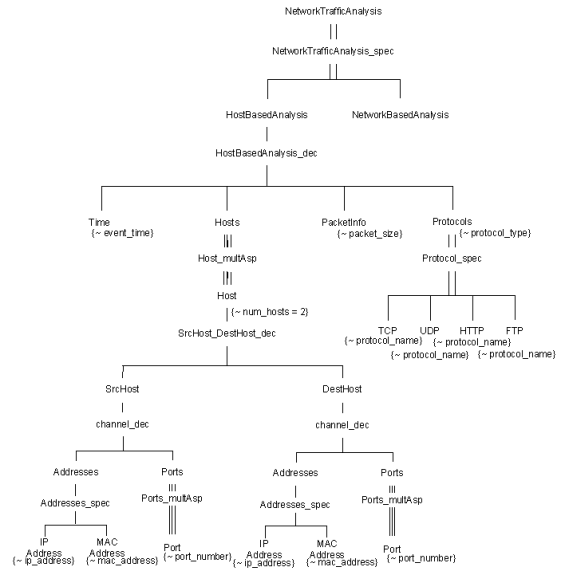


그림 3. System Entity Structure(SES) for Network Traffic Behavior

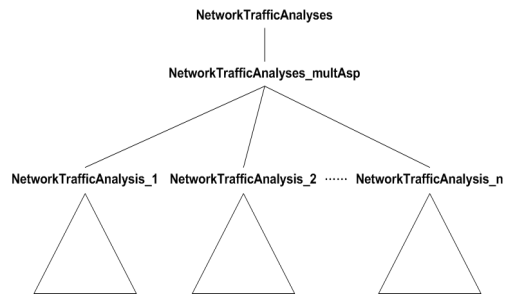


그림 4. MultiAspect for pruning of copies of its entity

다. HostBaseAnalysis는 Hosts, Time, Protocols, PacketInfo로 구성된다. Hosts는 복수개 Host인 SrcHost(출발 호스트)와 DestHost(도착 호스트)로 구성되며 각각은 Addresses와 Ports로 구성되는 똑같은 구조로 이루어져있다. Addresses는 IP_Address와 MAC_Address로 구성된다. 또한, Protocols는 TCP, UDP, HTTP, HTTP 등의 프로토콜 종류를 포함한다. 본 논문에서는 패킷에 포함된 정보 중에서 네트워크 트래픽을 분석하기 위해 가장 기본에 되는 정보에 대해서만 SES로 구조화 하였으나, 그림 3에서 보인 SES를 발전 시켜 향후에 정확성과 신뢰성을 높인 분석이 가능하다. 계층적 모델링 기법이 갖는 특징은 SES(System Entity Structure)에서 설명된 것과 같이 시스템을 구성하는 구성요소(Component) 및 특화(Specialization) 된 형태를 고려하여, 표현 할 수 있게 된다. 그림 4에서 볼 수 있듯이,

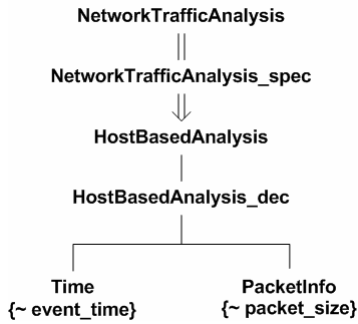


그림 5. PES for a case Study

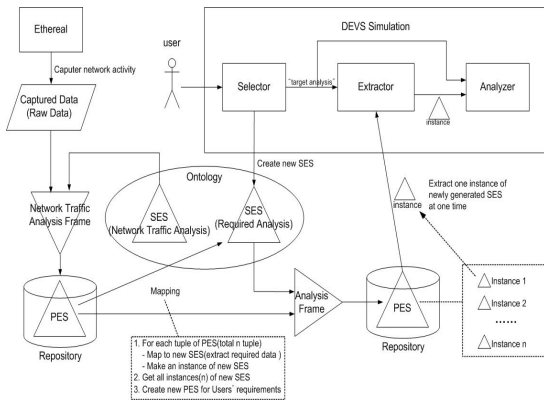


그림 6. Ontology based Automated Modeling and Simulation

상위 요소 NetworkTraffic에 대하여 다양한 관점을 지니고 있어, 네트워크 평가를 위한 PES의 다양한 보기가 사용자의 다양한 요구를 감안하여 생성할 수 있다는 의미이다. 그 요소의 복사본에 대한 가지치기(Pruning)를 위한 다양한 관점을 나타낸다. 예를 들어 만약 사용자가 얼마나 많은 TCP 패킷을 수신하고 있는가에 대한 측정을 원한다면, TCP 패킷과 호스트 IP 주소 데이터를 요구하며, PES의 하나의 보기는 이러한 요구에 따라 생성할 수 있다. 그림 5는 시간과 PacketInfo에 대한 관점을 담고 있는 PES를 보여준다.

그림 6은 온톨로지를 개념적으로 표현하는 SES를 생성하고, 가지치기(pruning)과 매핑(mapping)을 통해 실제 계에서 사용가능하기 위한 PES를 자동으로 생성하는 과정을 보여준다. 네트워크 분석기(Ethereal)로부터 캡처한 데이터와 일반적인 네트워크의 행동을 표현하는 SES의 조합으로 실제 사용 가능한 XML파일(PES)을 생성후 데이터베이스에 저장한다. 그 후, 사용자로부터 분석하고자 하는 대상(pragmatic frame)을 입력받고 이에 따른 새로운 온톨로지를 SES의 객체로 생성한다. XML형태로 존

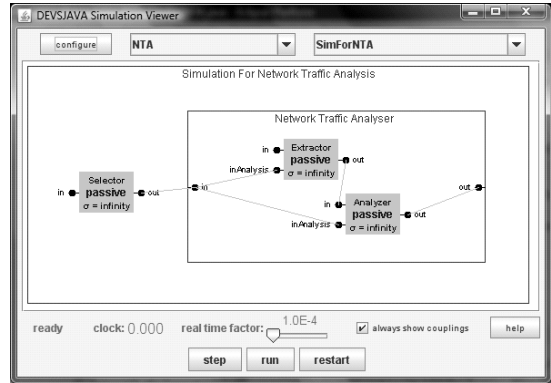


그림 7. DEVSJAVA Simulation Screen

재하는 일반적인 네트워크의 행동에 대한 attributes를 가지고 있는 PES로부터 새로 생성된 SES와의 매핑을 통해 사용자가 원하는 분석에 필요한 정보만을 포함하고 있는 PES를 XML 객체의 형태로 생성시켜 준다. 마지막으로, DEVS 시뮬레이션을 통해, 새로 생성한 PES를 분석하고 결과 값을 출력한다.

4. 네트워크 트래픽 분석에 대한 DEVS 시뮬레이션

그림 7은 본 연구를 DEVSJAVA 모델링하고 시뮬레이션 한 그림이다. Selector를 통해 사용자의 목적 분석(target analysis)을 입력받고, 그에 따른 SES 생성, 매핑, PES 생성과정을 거친 후 Extractor모델과 Analyzer모델을 통해, 결과를 분석한다.

본 연구에서 사용자의 요구를 받아들이는 입력 모델과 분석 모델을 분리함으로써, 성능 향상을 위한 분산 시뮬레이션에 적용될 수 있도록 모델링 하였다.

Selector 모델과 NetworkTrafficAnalyzer 모델을 서로 다른 컴퓨터에 위치시켜 한곳으로 집중되는 작업량을 분산시켜 CPU와 메모리에 생기는 과부하를 피함으로 성능 향상을 얻을 수 있었다.

표 1에서 대표적인 네트워크 패킷 분석 시스템인 Ethereal^[9]와 본 연구에서 구현한 시스템의 특성을 비교한다. 또한, 본 연구의 실험(프로토콜 사용량분석)을 통해 얻은 메모리의 사용량과 수행시간(execution time)을 Ethereal의 성능과 비교분석한 결과를 표 2에서 보여준다. 실험을 위해 반나절, 하루, 이틀 동안의 네트워크의 행동(activities)을 수집하였다. 또한 일반 데스크탑 컴퓨터를 사용하였다. Loading time이란 데이터베이스에 저장되어있는 데이터

표 1. Analyzing Manners of Ethereal and Our System

	Ethereal	Our System
Functionality	Protocol Analysis Throughput Analysis Service Analysis And more	Protocol analysis Throughput analysis User-selected and combined analysis
Analyzing Methods	Graphical charts Text analyses	Graphical charts Text analyses
Complexity	Complicate • Hard to learn • Hard to evaluate	Simple • Easy to learn • Easy to understand
Locality	Local machine only • Local machine : Monitor, capture, and analysis	Potentially distributed Environment • Local machine: monitor, capture • Remote machine: analysis
Data Size	Big whole data for every analyses	Specified small data for each analysis
Modularity	One process	Several individual processes
Scalability	Complete system, Not good for flexibility	Interoperability with XML-related systems

표 2. Memory Usages and Execution Times

	Ethereal			SES/NZER		
	Half day	One day	Two days	Half day	One day	Two days
Loading time	1min 18sec	2min 28sec	N/A	5min 28sec	10min 44sec	20min 59sec
Num of Events	1,063,803	2,045,700	N/A	1,063,803	2,045,700	4,091,400
Memory Usage	706 MB	1323 MB	N/A	98 MB	98 MB	98MB
Analyzing time	25sec	50sec	N/A	5min 29sec	10min 58sec	22min 59sec

를 사용하기 위해 불러오는 시간을 말하며 만나질, 하루 동안의 데이터에서는 Ethereal이 본 연구보다 좋은 성능을 보여준다.

또한, 데이터 분석시간(Analyzing Time)에서도 Ethereal의 성능이 더 좋다. 하지만, 메모리의 사용량에서 Ethereal은 이벤트의 수에 따라 선형(linearly)으로 증가 하는데 반하여 본 연구는 98MB를 유지하는 것을 볼 수 있다. Ethereal을 방대한 양의 데이터를 한 번에 불러와 시스템 메모리(RAM)에 저장하고 쓰는 반면, 본 연구는 불필요한 데이터는 제거했을 뿐만 아니라, 한 번의 분석시간에 작은 양의 필요한 데이터만을 불러와서 사용함에서 비롯된다. 실험을 통해, 이틀간의 데이터를 Ethereal은 처리할 수 없음을 보여줬다. 하지만, 본 연구는 분석하고자 하는 데이터의 크기에 영향을 받지 않으므로 이틀간의 데이터를 분석하는데 문제가 없었다. 하지만, Loading과 Analyzing을 위해 오랜 시간이 필요했다. 이 문제는, 분산 시뮬레이션을 통한 작업량의 분할로 해결 할 수 있을 것으로 보인다.

5. 결 론

본 논문은 네트워크 트래픽 분석 시스템에서 생성된 유용한 데이터를 네트워크 관리자에게 제공하여 네트워크 트래픽 조작의 어려움에 대한 최적의 해법을 제시하였다. 기존 네트워크 트래픽 관리 시스템은 분석 메소드의 부재, 데이터의 제한, 지나친 자원 소모와 같은 여러 문제를 가지고 있다. 이러한 문제는 느린 분석 시간과 많은 유지보수비용이란 문제를 야기한다.

SES 데이터 엔지니어링을 이용한 시스템은 네트워크 트래픽 데이터를 분석하기 위한 효과적인 데이터 관리 시스템이며 또한 네트워크 트래픽을 생성하는 복잡한 시스템을 모델링과 시뮬레이션에 사용될 수 있다. 예를 들어 침입감지시스템과 이상행동분석기와 같은 사이버 보안 시스템이 그 중 하나가 될 수 있다. 또한 복잡한 웹 서비스와 같은 분야에도 사용될 수 있다. 좀 더 효과적이고 효율적인 원격모델을 사용하기 위해 데이터 엔지니어링 방

법론을 통한 네트워크 트래픽 분석 시스템은 장래성 있는 접근으로 보이며, 이 시스템에 적용된 모델과 시뮬레이션 기반 데이터 엔지니어링 기법 역시 웹서비스를 위해 복잡하게 설계된 시스템에 장래성 있는 방법론이다.

참 고 문 헌

1. 윤상의, 유희렬, “네트워크 트래픽 관리를 위한 데이터웨어하우스시스템구현” 한국통신 무선통신연구소.
2. Teorey, T.J., Yang, D., and Fry, J.P., “A logical design methodology for relational databases using the extended entity-relationship model,” ACM Computing Surveys (CSUR), Vol. 18 No. 2, p. 197-222, June 1986.
3. OMG. Unified Modeling Language (UML), Version 2.1.1, 2007, <http://www.omg.org/technology/documents/formal>
4. Sarjoughian, H., and Flasher, R., “System Modeling with Mixed Object and Data Models,” DEVS Symposium, Spring Simulation Multiconference, Norfolk, Virginia, pp. 199-206, March 2007.
5. Jordan, D. and Russell, C., “Java Data Objects,” O’Reilly, 2003.
6. Zeigler, B.P., Kim, T.G., and Praehofer, H., “Theory of Modeling and Simulation,” 2nd ed., Academic Press, New York, 2000.
7. Zeigler, B.P. and Zhang, G., The System Entity Structure: Knowledge Representation for Simulation Modeling and Design, in Artificial Intelligence, Simulation and Modeling, L. E. Widman, K. A. Loparo, and N. R. Nielsen, Eds. Wiley, p. 47-73, New York, 1989.
8. B.P. and Hammonds, P.E., “Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange,” Elsevier, 2007.
9. StanfordUniversity,ProtegeOntologyEditorandKnowledge-baseFramework,<http://protege.stanford.edu/>
10. Ethereal, network protocol analyzer, 2007, <http://www.ethereal.com/>
11. SESBuilder, An Integrated Tool to utilize System Entity Structure, 2007, <http://www.sesbuilder.com/>



한 영 신 (yshan95@ewhain.net)

1997 이화여자대학교 전산정보학과(석사)
 2004 성균관대학교 전기전자컴퓨터공학과(박사)
 2004 이화여자대학교 컴퓨터그래픽스/가상현실 연구센터 박사후연구원
 2005 성결대학교 멀티미디어학부 전임강사
 2007 인하대학교 컴퓨터공학부 BK 계약교수
 2007 아리조나대학교 ACIMS 센터 Visiting Scholar

관심분야 : 컴퓨터시뮬레이션, 공장 자동화, 그리드 컴퓨팅, 데이터엔지니어링



김 태 규 (taekyu@gmail.com)

2000 중앙대학교 컴퓨터공학과 학사
 2006 아리조나대학교 Electrical and Computer Engineering 석사
 2008 아리조나대학교 Electrical and Computer Engineering 박사
 2009 안보경영연구원 M&S연구센터 선임연구원

관심분야 : 컴퓨터시뮬레이션, 그리드 컴퓨팅, 데이터엔지니어링, 네트워크보안시스템



정재은 (j2jung@gmail.com)

1999 인하대학교 컴퓨터공학과 학사
2001 인하대학교 컴퓨터공학과 석사
2005 인하대학교 컴퓨터공학과 박사
2006 INRIA Rhone-Alpes, 박사후 연구원
2006 인하대학교 컴퓨터공학부 BK 계약교수
2007 영남대학교 전자정보공학부 전임강사

관심분야 : Semantic Web, Ontology, Description Logic, Web 2.0



정찬기 (ckjung34@yahoo.co.kr)

1986 공군사관학교 전자공학 학사
1994 플로리다공대 전산공학 석사
2001 플로리다공대 전산공학 박사
2008 국방대학교 전산정보학과 조교수

관심분야 : ITA/EA, SOA, 분산처리, LVC체계, 체계통합



이철기 (cslee@skku.ac.kr)

1980 성균관대학교 전자공학과 졸업
1985 Arizona state University 전기 및 컴퓨터 공학석사
1990 University of Arizona 전기 및 컴퓨터 공학박사
1995 삼성 정보통신 본부 컴퓨터 응용 개발실 수석 연구원
1995~현재 성균관 대학교 정보통신공학부 교수

관심분야 : 컴퓨터 시뮬레이션, 객체지향 모델링, 공장자동화, 전문가 시스템