

망 관리 시스템을 위한 테스트베드 설계

송병권*, 김건웅**, 진명숙***

*서경대학교 정보통신공학과

**목포해양대학교 해양전자통신공학부

***경인여자대학 멀티미디어정보전산학부

e-mail: bksong@bukak.sekyeong.ac.kr, kgu@mmu.mmu.ac.kr,

jinms@dove.kyungin-c.ac.kr

Design of the Testbed for the Network Management System

Byungkwen Song*, Geonung Kim**, Myungsook Jin***

*Dept. of Information & Comm. Eng., Seokyeong University

**Faculty of Electronics & Comm. Eng., Mokpo National Maritime University

***School of Multimedia Information Computing, KyungIn Women's College

요약

일반적으로 망 관리 시스템(NMS: Network Management System)은 실제 자원과 이를 관리하기 위한 에이전트(agent) 개발이 완료된 후에 개발을 시작하거나, 실제 자원의 개발자로부터 관리 정보를 제공받아 병행해서 개발된다. 그러나, 전자의 경우는 실제 자원과 에이전트가 개발된 후에도 관리 시스템 개발까지의 공백이 초래되고, 후자 또한, 망 관리 시스템의 올바른 동작을 위하여 실제 자원이 완성된 이후 일정 기간 동안 정합시험이나 안정성 시험을 통해 문제점들을 보완해야 한다. 본 논문에서는 실제 자원의 개발전에도 망관리시스템의 개발 및 운용 테스트를 수행하도록 지원하는 테스트베드를 제안한다.

1. 서론

ITU-T에서는 개별 운용시스템의 한계성을 극복하고 표준화된 개방형 방식의 총체적이고 일원화된 통신망 운용관리체제를 구축하기 위하여 TMN (Telecommunication Management Network) 관리 개념을 권고했다[1]. TMN은 그림 1과 같이 망 관리 역할에 따라 관리자(manager)와 에이전트(agent)로 구분된다. 관리자는 에이전트에게 관리 요청을 전달하고 에이전트로부터 관리 요청에 대한 수행 결과를 반환 받거나, 특별한 사건이 발생했음을 알리는 통고(notification) 메시지를 수신한다. 또한 에이전트는 관리자로부터 관리요청을 수신한 다음 실제 자원(real resource)에 접근하여 해당 값을 가져온 후, 그것을 관리자에게 반환하거나 실제 자원에서 발생한 통고를 전달한다. 따라서 관리자는 에이전트를 통하여 실제 자원에 대한 관리를 종합적으로 수행한다[2].

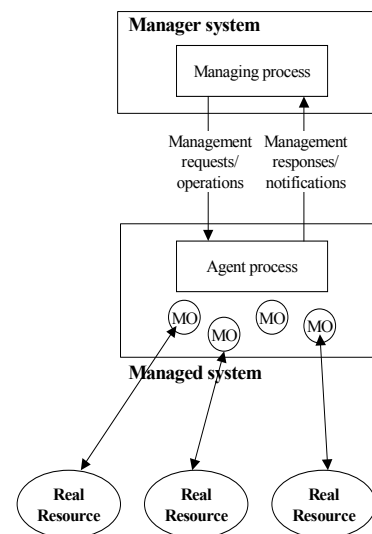


그림 1. TMN 관리 모델

망 관리 시스템에서 실제 자원은 전기통신망을 구성하는 교환기 및 각종 유무선 통신 장비 등 실제

관리하고자 하는 하드웨어 또는 소프트웨어적 요소다. 이러한 실제 자원을 관리하기 위한 망 관리 시스템은 실제 자원이 완성된 후에 개발을 시작하거나, 실제 자원 개발자로부터 사전에 관리 정보를 제공받아 실제 자원 개발과 병행해서 개발한다. 그러나 전자의 경우는 완성된 실제 자원을 기존 통신망에 설치 및 운용했을 때, 해당 장비에 대한 관리 시스템 부재로 일정 기간 동안 관리 공백이 초래되고, 후자 또한 실제 자원 개발자로부터 얻는 정보의 부족으로 인해, 양쪽이 완성된 이후에도 일정 기간 동안 정합시험이나 안정성 시험을 거치면서 문제점을 수정 및 보완해야 한다. 이러한 문제점들을 해결하기 위해선, 실제 자원 개발 전에도, 사용자가 지정하는 형태로 속성(attribute) 값들이나 사건(event)을 생성하여 에이전트에게 돌려주는 테스트베드가 필요하다.

본 논문에서는 먼저 2장에서 이러한 테스트베드의 역할과 전체 구성, 그리고 각 요소들에 대해 기술하고, 3장에서 RRS의 동작 시나리오를 소개한 후 결론을 맺는다.

2. 테스트베드의 전체 구성

본 논문에서 제안하는 테스트베드의 주 역할은 망 관리 시스템이 관심을 갖는, 실제 자원이 동작하면서 발생하는 속성 값들의 변화나 사건 등을 사용자가 원하는 방식으로 생성시켜, 관리시스템의 개발 및 운용 테스트를 하도록 하는 것이다. 그림 2는 관리자, 에이전트 그리고 본 연구에서 설계한 테스트베드, 즉 RRS(Real Resource Simulator)와의 관계를 나타낸다.

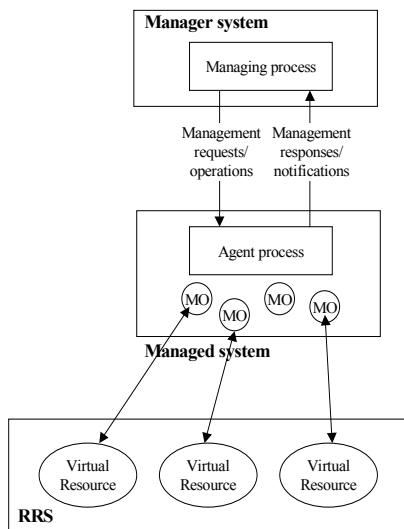


그림 2. 관리자, 에이전트, 그리고 RRS와의 관계

RRS는 다음 그림 3과 같이, 에이전트와의 다양한 통신 방식을 지원할 통신 모듈, 각 관리 객체 별로 사용자가 지정한 속성 값 및 통고의 생성 방식 등을 담고 있는 SDT(Simulation Data Table), 현재 생성된 관리 객체들을 담고 있는 테이블(MO Table), 이들을 바탕으로 속성 값을 변화시키거나 통고를 발생시킬 커널 메인(kernel main)과 이때 이용할 지원 함수(support function)코드, SDT 또는 객체 테이블 내의 값 변경, 통고의 발생 등을 직접 수행할 수 있도록 하는 GUI 인터페이스로 이루어져 있다.

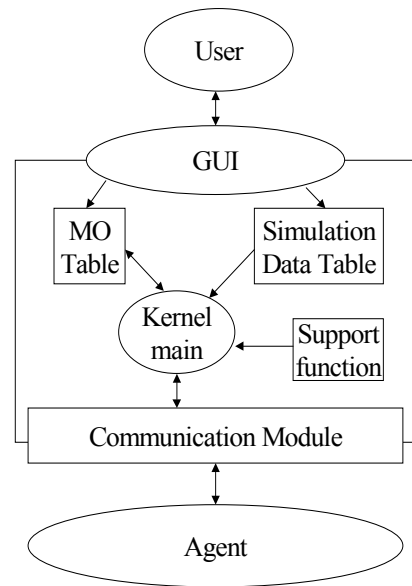


그림 3. RRS 전체 시스템 구조

여기서 커널 메인은 i) 에이전트로부터 요청 받은 속성 값을 사용자가 정한 지원함수의 도움을 받아 생성하여 반환하고, ii) 사용자가 지정한 랜덤 함수에 따라, 통고 메시지를 생성하여 에이전트에게 전달하는 기능을 수행한다. 따라서 커널 메인은 RRS내의 모든 객체들을 관리하게 되는데, 이들 관리 객체의 정보들이 관리 객체 테이블에 존재하게 된다. 이때 속성 선택스가 REAL 또는 INTEGER인 경우에 사용자가 지정한 지원 함수를 통해 랜덤값(random number)을 생성하고, 선택스가 STRING인 경우에는 사용자에게 의해 설정된 고유한 값을 이용한다.

SDT에는 각 관리 객체의 속성 값 및 통고들을 생성할 때 이용할 정보들을 담고 있다. 여기에는 랜덤 값을 생성하는데 이용할 지원함수, 통고 발생 간격을 결정할 지원 함수, 이들 지원 함수들의 관련 매개 변수들을 담고 있다.

GUI 인터페이스는 커널 동작 전에 SDT의 값들을 초기화하거나, 동작 중에도 사용자의 뜻에 따라 MO

테이블에 저장된 정보들을 변경할 수 있도록 한다. 그림 4는 SDT에 저장된 속성의 지원 함수 종류, 파라미터, 그리고 주기에 관한 정보를 설정하는 예다. 이러한 동작은 커널의 실행 전 또는 실행 중에도 이루어질 수 있다.

Set Attribute	
Attribute Name	cpuUtility
Syntax Name	Percentage
type	REAL
Value	
Random Number Type	Uniform
Parameter1	0
Parameter2	100
Period	
Parameter1	
Parameter2	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

그림 4. SDT 정보 설정 화면의 예

3. RRS의 동작 시나리오

3.1 GUI를 통한 SDT 정보 설정

그림 5는 사용자가 GUI를 통하여 SDT에 있는 값을 초기화 시키는 과정과 동작 중에 값을 변경하는 과정을 나타낸다.

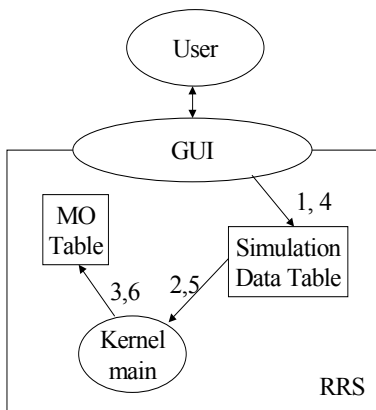


그림 5. GUI를 통한 SDT 값 설정 및 커널의 동작

- (1) 사용자는 GUI 인터페이스를 통하여 SDT에 저장된, 랜덤 값을 생성하기 위한 지원 함수와 생성 간격을 결정할 지원 함수, 각 함수들의 매개 변수들을 설정한다.
- (2) 시뮬레이터 커널은 사용자가 설정한 속성에 대한 초기값을 SDT로부터 읽어온다.
- (3) 시뮬레이터 커널은 SDT에서 읽어온 값을 이용하여 자신의 MO 테이블을 초기화한다.
- (4) 동작 중에도 사용자는 GUI 인터페이스를 통하여 SDT에 저장된 값들을 변경한다.
- (5) 변경된 경우, 이를 시뮬레이터 커널에게 알린다.
- (6) 시뮬레이터 커널은 변경된 값에 따라 MO테이블

의 관련 값들을 수정한다.

3.2 에이전트 요청에 따른 속성 값 생성 및 반환

그림 6은 에이전트로부터 요청을 받은 후, 커널에서 임의의 값을 생성하여 반환하는 예를 나타낸다. 속성 값은 현재 MO테이블에 저장된 값을 바로 돌려주는 방식도 가능하나, 여기서는 요청이 오면 새로 생성하는 방식을 보이고 있다.

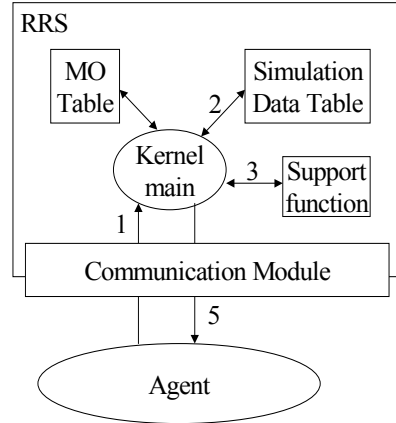


그림 6. 에이전트로부터 요청된 속성의 처리 과정

- (1) 에이전트가 특정 관리 객체에 대한 속성 값을 요청한다.
- (2) 커널은 SDT를 검색하여 관련 정보(지원함수의 타입 및 매개변수 값들)를 가져온다.
- (3) (2)의 과정에서 얻은 정보를 이용하여 지원 함수를 실행시키고, 그 결과 값을 얻는다.
- (4) 결과 값을 MO 테이블에 저장한다.
- (5) 결과 값을 에이전트에게 반환한다.

3.3 속성값 생성 및 저장

시뮬레이션 커널이 동작하게 되면, SDT에 정의된 값들에 의해 속성 값들을 생성하고 이것을 MO테이블에 저장해야 한다. 다음 그림 7은 이 과정을 보이고 있다.

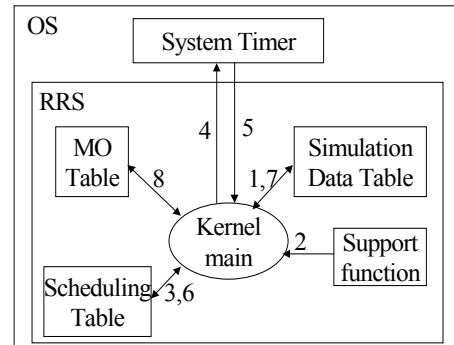


그림 7. 주기적인 속성값의 변경

- (1) SDT가 초기화되었거나 속성값을 변경한 발생한 경우, SDT로부터 속성값을 생성시키기 위한 정보들을 가져온다.

- (2) (1)의 정보에 따라 해당 지원 함수를 실행시켜 다음 발생 시간을 구한다.
- (3) 구해진 발생 시간을 스케줄링 테이블에 등록한다. 이러한 스케줄링 테이블은 시간에 따라 순서적으로 정렬되어 있고, 상대적 타이머(relative timer)를 이용한다.
- (4) 맨 처음에 변경해야 하는 속성 값인 경우, 남은 시간을 OS의 타이머로 알린다.
- (5) 주어진 시간이 경과한 후 OS의 타이머로부터 시그널(signal)이 들어온다.
- (6) 스케줄링 테이블을 검색하여 해당 속성을 찾는다.
- (7) SDT로부터 랜덤 값을 발생시킬 지원함수를 찾는다.
- (8) 지원함수를 이용, 속성값을 생성하여 MO테이블의 속성 값을 변경한 후 (1)로 돌아간다.

3.4 통고의 생성 및 전달

속성의 경우 발생 시간이 되면 지원 함수를 실행하여 그 결과 값을 MO 테이블에 저장하는 것에 반하여, 통고는 발생 시간이 되면 이를 능동적으로 에이전트에게 전달해야 한다. 이에 따라 통고를 처리하는 과정은 속성의 처리 과정과 다르게 되는데, 그림 8에서 이를 보이고 있다.

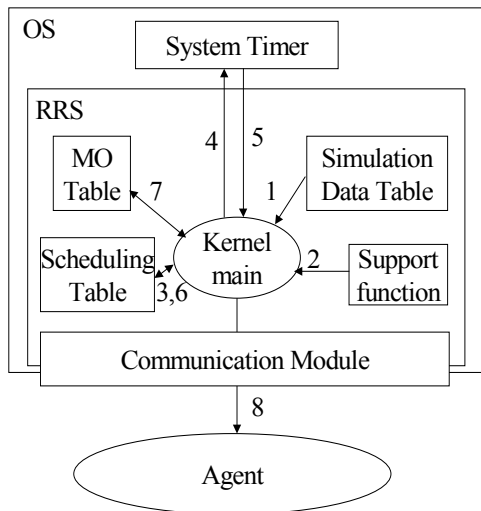


그림 8. 통고의 처리

- (1) SDT가 초기화되었거나 통고가 발생한 경우, SDT로부터 통고를 발생시키기 위한 정보들을 가져온다.
- (2) (1)의 정보에 따라 해당 지원 함수를 실행시켜 다음 발생 시간을 구한다.
- (3) 구해진 발생 시간을 스케줄링 테이블에 등록한다. 이러한 스케줄링 테이블은 시간에 따라 순서적으로 정렬되어 있고, 상대적 타이머(relative

timer)를 이용한다.

- (4) 맨 처음에 실행해야 하는 통고인 경우, 남은 시간을 OS의 타이머로 알린다.
- (5) 주어진 시간이 경과한 후 OS의 타이머로부터 시그널(signal)이 들어온다.
- (6) 스케줄링 테이블을 검색하여 해당 통고를 찾는다.
- (7) MO테이블로부터 통고와 관련된 정보를 찾는다.
- (8) 이를 에이전트에게 전송한 후 (1)로 돌아간다.

4. 결론

본 논문에서는 TMN 상에서 실제 자원이 없이도 사용자가 정의한 실제 자원의 동작 특성에 따라 그 행동을 대신할 수 있는 RRS에 대해 소개하였다. 제안한 RRS는 관리 객체의 속성과 통고를 고려하여 설계되었으며, 사용자가 설정한 시뮬레이션 관련 정보에 따라 속성 값을 변경시키고, 통고를 발생시키도록 되어 있다. 현재, RRS는 각 구성 요소별로 세부 설계가 진행중이며, UNIX의 사용자 영역에서 할 예정이다.

참고문헌

- [1] ITU-T M.3010,"Principles for a Telecommunication Management Network"
- [2] ISO7498-4/ITU-T X.700,"OSI Basic Reference Model Part 4: Management Framework"
- [3] ISO10040/ITU-T X.701,"Systems Management Overview"
- [4] ISO9595/ITU-T X.710,"Common Management Information Service Definition"
- [5] ISO9596-1/ITU-T X.711,"Common Management Information Protocol Specification"
- [6] ISO10165-1/ITU-T X.720,"Management Information Model"
- [7] ISO10165-2/ITU-T X.721,"Definition of Management Information"
- [8] ISO10165-4/ITU-T X.722,"Guidelines for the Definition of Managed Objects"
- [9] ISO10165-5/ITU-T X.723,"Generic Management Information"
- [10] ISO10164-5/ITU-T X.734,"Event Management Function"
- [11] John Larmouth, "ASN.1 Complete", Open Systems Solutions, 1999