

가상 네트워크 망에서의 효율적인 테스트

시나리오 관리 기법

장준영*, 김재훈**, 김응구***

*아주대학교 지식정보보안학과

**아주대학교 정보통신전문대학원

*** (주)이루온 QA팀

e-mail:knock84@ajou.ac.kr

An Efficient Test Scenario Management on Virtual Network

Jun-Young Jang*, Jai-Hoon Kim**, Eung-Gu Kim***

*Department of Knowledge Information Security, Ajou University

**Graduate School of Information & Communication, Ajou University

*** QA, Eluon, Inc

요 약

여러 장치들과 연동하여서 동작하는 네트워크 시스템 개발 시, 회귀 테스트는 전체 시스템의 안정성을 보장하기 위한 가장 필요한 테스트중의 하나이다. 그러나, 서비스가 되고 있거나 연동을 위한 추가적인 장비가 필요할 경우 테스트의 제약을 받게 된다. 본 논문에서는 자동 회귀 테스트를 위한 가상 네트워크 망을 제공하는 시뮬레이터를 디자인 한다. 또한, 시뮬레이터를 효율적으로 관리 및 동작하기 위한 UI 환경을 제안한다. 오픈 소스를 기반으로 제작된 본 시나리오의 관리 툴은 회귀 테스트의 자동화와 시나리오의 생성 및 배포가 가능하며 여러 프로토콜을 통합하여 사용할 수 있는 환경과 검증 코드를 제공한다. 이로 인해 네트워크 시스템 상의 소프트웨어에 대해 기능 및 회귀 테스트가 가능해 질 것이고 결과적으로 프로그램의 개발 시간 단축과 유지보수에 탁월한 환경을 제공받게 될 것이다.

1. 서론

모바일 서비스 이래로 소프트웨어의 복잡도와 규모가 증가함에 따라 테스트의 중요성 또한 증대되고 있다. 특히, 여러 장치들과의 연동하여 동작하는 네트워크 시스템 개발 시, 회귀 테스트는 가장 중요한 테스트중의 하나이다. [Reg2010] 그러나, 아직 대다수의 테스트는 QA(Quality Assurance) 심사 과정에만 국한 되어 반복적이고 피동적인 매뉴얼 테스트 방법을 위주로 진행 되고 있다. 이것은 소프트웨어 수정 때마다 회귀 테스트를 수행할 수 없음을 의미하며 결과적으로 테스트 환경을 구축하기 위한 비용과 QA 기간이 급격하게 증가할 수 있는 문제점을 가지고 있다. [Na2010]

이러한 문제점을 해결하기 위해서는 개발 기간 중에 품질보증을 위한 1차적인 회귀 테스트가 이루어져야 한다. 하지만, 모바일 환경에서 동작하는 프로그램을 개발하고 연동할 때, 회귀 테스트를 위한 환경을 구축을 위해서는 연동 장비 대여와 장비를 대여 해주는 업체에 대한 소스 오픈 등 테스트 실행에 대한 여러 가지 비용이 소요되며 많은 제약들이 따르게 된다.

이러한 제약을 감소시킬 수 있도록 본 논문에서는 실제 연동장비 역할을 하는 시뮬레이터를 제안한다. 이 시뮬레이터는 연동하고자 하는 가상의 망을 만들어 피시험장

비가 충분한 회귀 테스트를 할 수 있도록 한다. 먼저 가상 네트워크 망을 제공하는 망의 범위를 설명하고 이에 관한 시뮬레이터의 전체 구조를 설계한다. 제안하는 시뮬레이터는 '시나리오 관리부', '엔진부', 'Interface Stack 부'로 나누어지며, 여기서는 '시나리오 관리부'에 대한 설계를 중점으로 다룬다.

시뮬레이터를 효율적으로 관리 및 동작을 하기 위한 컨트롤러인 '시나리오 관리부'는 오픈소스인 'Testlink Beta 1.9.3'을 기반으로 개발 되었다. 시나리오에 대한 테스트케이스를 생성하고 실행할 수 있으며 그에 대한 결과를 리포팅 할 수 있다. 이로 인해 테스트에 대한 자동화와 테스트 시나리오를 효율적으로 관리 할 수 있다.

2. 관련 연구

2.1 Testlink 의 구조

오픈소스로 구현된 Testlink 는 소프트웨어 또는 하드웨어등의 시험을 계획하고 그에 대한 결과를 입력하는 테스트 툴 소프트웨어이다. 결과를 입력한다는 건 Testlink 에서 자동으로 해당 테스트를 실행 한다는 것이 아니라 사용자가 직접 테스트한 것에 대한 결과를 저장하는 것을 말합니다. 본 논문에서 구현된 Testlink는 테스터가 입력한 테스트 대해 자동적으로 메시지를 날려 처리하고 그에

대한 결과를 확인 할 수 있는 기능을 가진다.

Testlink는 Smarty Template을 사용하여 UI와 내부함수코드가 분리되어 있다. 테스트 단위는 Test Project > Test Suite > Testcase > Version > Step 으로 되어 있다. 구현한 Testlink에서 Project 는 전체 회귀 테스트를 통칭하고 Suite 는 각 회귀 테스트를 실행 할 수 있는 최소 단위이다. Testcase 는 가상 장비중 하나를 나타내고 Version은 Testcase를 버전별로 관리 할 수 있는 단위를 나타낸다. 마지막으로 Step 은 실제 테스트의 메시지 단위이다. 본 논문에서는 Step 의 단위의 메시지를 Test Stub 메시지라고 생각해도 무방하다. [Martin2010]

3. 본론

3.1. 배경 및 시뮬레이터 요구사항

본 논문에서 제안하는 시뮬레이터의 구조는 아주대학교와 (주)이루온의 산학협력 프로젝트로 진행된 "Regression test tool for HLR"을 기반으로 한다.

제안하는 가상 네트워크 망의 범위는 이동통신망으로 한정하고 피시험 장비를 HLR (Home Location Register)로 가정한다. 그림 1은 시뮬레이터의 망 구조에 대해 나타내고 있다.

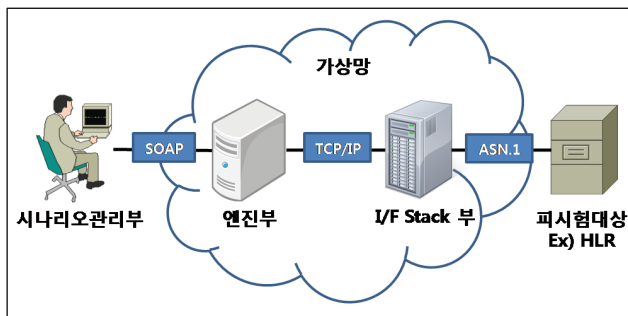


그림 1 시뮬레이터 망구조

실제 HLR 장비를 테스트하기 위한 물리적인 환경을 구성하는 대신 가상의 망을 구축하여 HLR 장비에 대한 회귀 테스트가 가능하도록 디자인 한다. 가상망은 엔진부와 I/F Stack 부로 구성되며 엔진은 시나리오에 대한 실행을 하고, I/F Stack은 다양한 프로토콜의 메시지를 만드는 역할을 한다.

제안하는 시뮬레이터의 요구사항으로는 One-PC에서 테스트가 가능한 환경을 구축해야 된다는 것에 있다. 그러기 위해서는 첫째, 서비스 지향 구조(SOA: service-oriented architecture) 이여야 한다. 즉, 웹기반 구조로 어디서든 접근하여 테스트가 가능하도록 설계 되어야 한다. 둘째, 다양한 프로토콜에 대한 시나리오 생성이 가능해야 한다. 셋째, 작성된 시나리오가 순차적으로 실행 되어야 한다. 넷째, 시나리오의 수행 결과를 저장하고 리포팅 할 수 있어야 한다. 다섯째, 다양한 프로토콜에 대한 가상메시지를 만들 수 있어야한다. 그림 2에 시뮬레이터의 세부적인 요구사항을 보여준다.

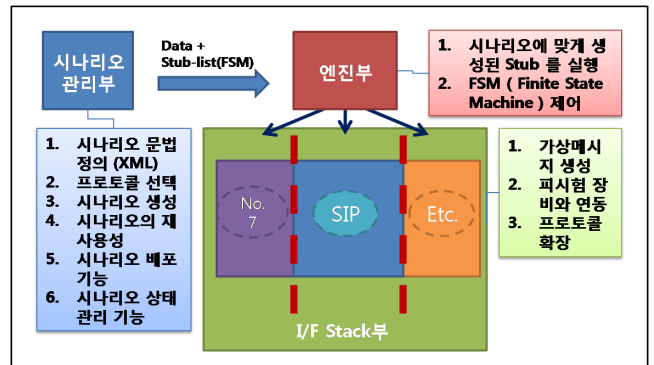


그림 2 시뮬레이터 요구사항

3.2 시뮬레이터의 전체 구조

이와 같은 요구사항으로 설계된 시뮬레이터의 상세 구조는 그림 3과 같다.

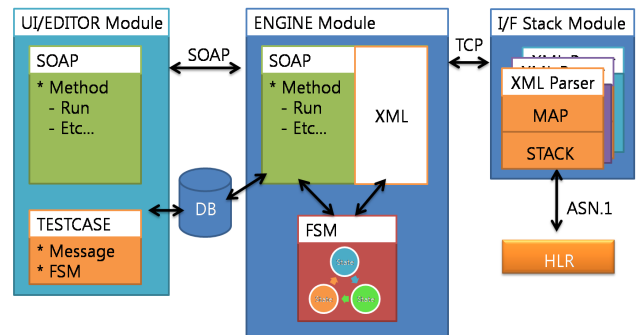


그림 3 시뮬레이터 상세구조

시나리오 관리부는 사용자에게 시나리오를 관리 할 수 있는 웹 어플리케이션을 제공하여 시나리오를 입력 및 테스트를 할 수 있다. User Interface 로 오픈소스인 Testlink (Test management tool) 를 사용한다. Testlink 는 Test Category를 제공하고 각 Test에 대한 테스트케이스, 요구사항, 테스트 계획, 빌드, 수행 및 결과 레포트를 산출 할 수 있어 시나리오를 시험하고 관리하는데 용이한 장점을 가지고 있다. 시나리오관리부는 엔진부와 SOAP을 이용한 RPC 통신을 하여 엔진에게 시나리오의 실행을 요청한다. RPC 동작으로 시나리오 관리부는 어디서든 엔진부에 접근하여 필요한 테스트를 수행할 수 있다.

엔진부는 시험할 Testcase 에 대해 FSM을 생성하여 가상의 에뮬레이터처럼 동작 한다. FSM을 제어하기 위한 Dispatcher를 구현하여 송/수신하는 메시지들을 확인한다.

I/F Stack 부는 시나리오관리부에서 작성된 Stub 메시지를 실제 장비의 프로토콜에 맞게 변환시켜 피시험장비에게 전송한다.

본 논문에서는 전체 시뮬레이터에 대한 설계 중 시나리오 관리부를 자세히 소개한다.

3.3 시나리오 관리부 설계

그림 4는 시나리오관리부와 엔진부에 대한 SW 구조를 나타낸다. User Interface인 Testlink 에 추가적으로 설계한 API는 아래와 같다.

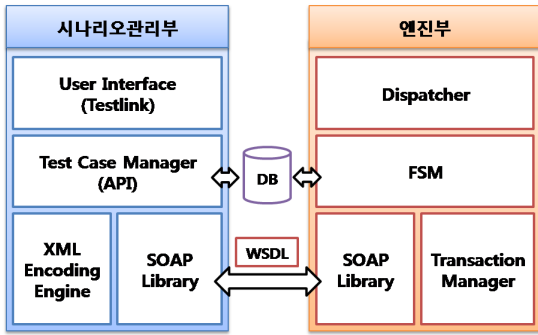


그림 4 시나리오 관리부와 엔진부의 SW 구조

- ① 테스트케이스에 대해 프로토콜을 선택할 수 있도록 화면 재구성
- ② 관련된 프로토콜에 대해 일반적인 메시지를 사용자가 선택하여 테스트 환경에 맞게 수정할 수 있도록 재구성
- ③ 각각의 프로토콜에 맞게 테스트케이스 메시지를 XML로 Encoding 하는 기능 추가
- ④ SOAP 라이브러리와 연동하여 엔진에게 테스트를 요청하는 기능 추가
- ⑤ 테스트 슈트(suite) 단위로 테스트에 대한 결과와 Log 를 확인할 수 있도록 재구성
- ⑥ 피시험장비로부터 수신된 메시지를 검증할 수 있는 알고리즘 구현

③의 Encoding 기능은 PHP 의 Extension_Functions (ext_skel 명령어) 을 사용해 C로 만든 라이브러리를 php 에 Import 시켜 각 프로토콜에 맞는 Encoding 룰을 선택하여 적용할 수 있도록 한다. 이로 인해 기존에 작업하던 각 프로토콜의 C 언어 기반의 라이브러리를 추가 작업 없이 Testlink 환경에서 효율적으로 사용할 수 있도록 해 준다. ④의 SOAP 기능 추가는 PHP 의 nusoap 라이브러리를 사용하며 현재 웹 서비스 WSDL (Web Service Description Language) 의 표준에 따라 구현한다.

DB 에 저장되는 테스트케이스 메시지는 XML 형태이다. 이는 Text 기반으로 사용에 용이하고 확장성 및 통합에 강점을 보이고 있는 Markup 언어로 추후에 메시지 검사와 동적 변수에 대한 자동처리가 용이하도록 하기 위해서이다.

시나리오 관리부는 시나리오 입력, 시나리오 생성, 시나리오 실행, 시나리오 결과 총 4개의 세부 모듈로 정의할 수 있다. 각 모듈에 대한 시스템의 처리 과정은 그림 5에 나타나 있다. 시나리오 입력은 DB 로부터 선택된 프로토콜에 대해 기본 메시지 정보들을 가져와 사용자에게 보여준다. 사용자가 입력한 시나리오에 대해 XML 데이터를 생성한다. 이 데이터는 엔진부에서 I/F Stack 부에게 전달할 Stub 메시지들이다. 사용자는 해당 시나리오를 실행하기 위해 시나리오를 Run 하게 되고 그에 대한 응답을 받게 된다. Stub 메시지들을 SOAP 메시지에 포함시키지 않고 따로 DB 로 설계한 이유는 송/수신 트래픽을 줄이기 위해서이다.

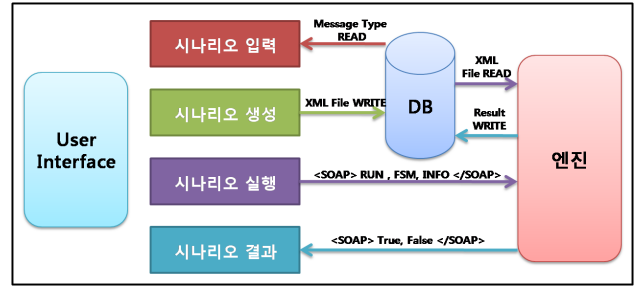


그림 5 시나리오관리부 처리흐름도

3.4 시나리오 관리부 구현

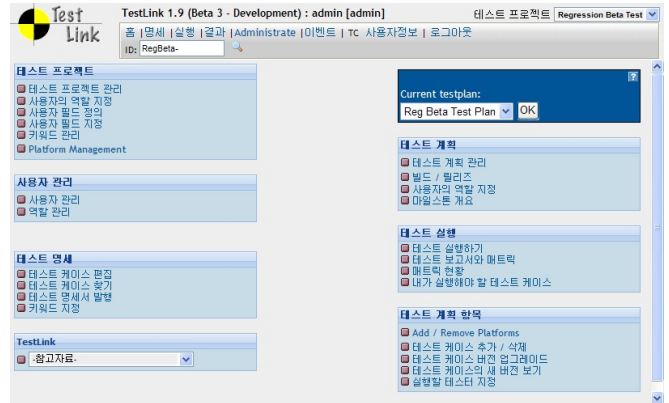


그림 6 테스트링크 메인화면

그림 6는 테스트링크의 메인화면이다. 이중에 “테스트 케이스 편집”과 “테스트 실행하기” 부분을 재구성한다.

테스트 편집 부분에 테스트케이스를 생성할 때, 생성하고자 하는 프로토콜을 선택하여 생성할 수 있다. 각 프로토콜마다 다른 UI와 Encoding 룰을 제공한다.

테스트케이스에 맞는 메시지들을 생성할 때, 기본적으로 제공하는 Message Type & Struct Type , FSM Type을 미리 정의 해 놓고 필요한 메시지를 DB에서 가져와 사용한다. “Insert” 버튼으로 생성되는 FSM_results 정보는 엔진부에서 FSM 생성하기 위한 정보가 된다. “DATA CREATE & STEP ENCODING” 버튼으로 각각의 프로토콜에 맞게 메시지를 XML 로 변환시킨다.

테스트를 실행하기 위해 엔진에게 SOAP 메시지를 요청하는 화면은 그림 7과 같다.



그림 7 테스트 실행 화면

전송하는 SOAP 의 메시지 형식은 아래와 같다.

```

<SOAP-ENV:Body>
  <ns1:GetStart xmlns:ns1="http://localhost/RegTest"
  ">
    <TestId>Testsuite 대한 정보</TestId>
    <TestInfo>
      <Item xmlns="">
        <CaseId xmlns="">Testcase에대한 정보</CaseId>
        <Fsm xmlns=""> 각 Step 메시지들에 대한
          프로토콜 정보 및 FSM 정보
        </Fsm>
        <Option xmlns="">추가옵션</Option>
      </ns1:GetStart>
    </SOAP-ENV:Body>
  
```

실행 결과 및 로그는 Result Loaded Table의 Suite Name 컬럼을 선택하면 실제 데이터 전송에 대한 상세 로 그를 확인 할 수 있다.

3.5 시나리오 관리부 시험 및 결과

시뮬레이터를 활용한 테스트 구현 예제로 실제 HLR 장비의 서킷위치등록 서비스에 대한 회귀 테스트를 진행 한다. 본 사례는 HLR 피시험장비에 대해 SGSN(Serving GPRS Support Node) A와 SGSN B 에 해당하는 가상 망 을 만들어 실제 HLR 서비스가 동작하고 있는지 테스트 한다.

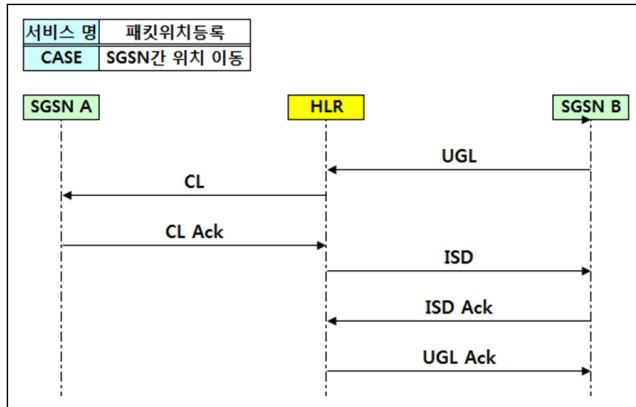


그림 8 시나리오 예제 흐름도

먼저 그림 8의 시나리오의 흐름을 토대로 SGSN A, SGSN B의 Testcase와 각각의 Step 메시지를 작성한다.

Step #	Step actions	예상결과	Execution
1	MAP OPERATION : MAP_Cancel_Location value CancelLocation-v3- MAP_Cancel_Location:Arg:Recv 수동 Arg ::= { identity imsi : 54001850004807F4H, cancellationType updateProcedure }		실패
2	MAP OPERATION : MAP_Cancel_Location value CancelLocation-v3- MAP_Cancel_Location:Arg:Send 수동 Arg ::= { identity imsi : 54001850004807F4H, cancellationType updateProcedure }		실패
1	MAP OPERATION : MAP_Update_Location value UpdateLocation-v3-Arg ::= { imsi 54001830008218F4H, msc-Number 91280192919905H }	MAP_Update_Location:Arg:Send	수동 실패
2	MAP OPERATION : MAP_Insert_Subscriber_Data value InsertSubscriberData-v3-Arg ::= { imsi 54001850004807F4H, msisdn 91280192121972H }	MAP_Insert_Subscriber_Data:Arg:Recv	수동 실패
3	MAP OPERATION : MAP_Insert_Subscriber_Data value InsertSubscriberData-v3-Res ::= { supportedCameIPhases { phase3 } }	MAP_Insert_Subscriber_Data:Res:Send	수동 실패
4	MAP OPERATION : MAP_Update_Location value UpdateLocation-v3-Res ::= { hlr-Number 91280192919984H }	MAP_Update_Location:Res:Recv	수동 실패

그림 9 SGSN A,B 의 Step 메시지 작성 예

작성된 메시지는 그림 9와 같고 현 시나리오의 수행 결과는 그림 7과 같다. SGSN A나 B가 HLR로부터 Step에 정의된 Recv 메시지 받으면 True를 그렇지 않다면 Fail을 반환한다.

4. 결론

현재 네트워크 시스템 개발 시 직접 연동이 필요한 장비들을 구매 또는 대여하여 개발한 네트워크 시스템에 대해 테스트를 한다. 이는 구매 시 많은 비용이 필요하고 또한 대여 시에도 비용뿐 만 아니라 대여 업체에게 자신의 개발 소스를 오픈해야 하는 어려움이 있다. 이를 해결하기 위해 One-PC 환경에서 테스트 가능한 네트워크 시뮬레이터를 필요로 한다. 또한, 이 시뮬레이터를 사용하기 위해서는 관리자가 용이하게 시나리오 작업을 할 수 있는 환경을 제공해야 한다. 본 연구는 가상 네트워크 망에서의 효율적인 테스트 시나리오 관리 기법을 위한 시나리오 틀을 개발한다. 이 틀은 시나리오를 생성 및 배포하고 관리하여 시나리오에 대한 재사용성, 시스템 형상과의 독립성, 멀티 동작 등을 보장 받을 수 있게 될 것이다. 이로 인해 기업 내의 비용 절감과 개발 기간 단축의 효과를 얻을 수 있을 것이다.

참고문헌

[Na2010] 나종채, 오영은, 유석문, "FitNesse와 STAF을 결합한 테스트 자동화 프레임워크의 구현", 한국정보과학회 추계 논문, 2009

[Reg2010] Wikipedia, 회귀테스트, http://ko.wikipedia.org/wiki/%ED%9A%8C%EA%B7%80_%ED%85%8C%EC%8A%A4%ED%8A%B8/.

[3GPP102] 3GPP TS 33.102, 3G Security (Release 8), <http://www.3gpp.org/ftp/Specs/>.

[3GPP922] 3GPP TS 23.922, Architecture for an All IP Network, <http://www.3gpp.org/ftp/Specs/>.

[WSDL2001] W3C Note 15 March 2001, WSDL 1.1, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315/>.

[Scott2004] Scott Nichol., Introduction to NuSOAP, <http://www.scottnichol.com/nusoapintro.htm/>.

[Mike2009] Mike Cohn, The forgotten layer of the test automation pyramid, <http://blog.mountaingoatsoftware.com/the-forgotten-layer-of-the-test-automation-pyramid/>.

[Martin2010] Martin Havlát, Andreas Morsing and Francisco Mancardi, TestLink Developer's Guide v1.16, <http://www.teamst.org/index.php/doc/>.