스마트폰 기반 OAM을 위한 효율적인 메시지 전달에 관한 연구

김규완, 김강석, 예홍진
아주대학교 일반 대학원 지식정보보안학과
e-mail:potoit@ajou.ac.kr, kangskim@ajou.ac.kr, hiyeh@ajou.ac.kr

A Study for effective message transmission with smartphone in OAM system

Kyuwan Kim, Kangseok Kim, Hongjim Yeh
Dept of Knowledeg Information Security, Graduate School Of Ajou University

요 약

본문에서는 Pull/Push 서비스를 사용하여 스마트폰 기반의 OAM(Operation, Administration & Maintenance) 시스템에서 보다 빠르고 효율적인 메시지 전달 방법을 제안한다. OAM은 네트워크의 운영, 관리, 결함 표시, 성능 정보, 데이터와 진단기능을 제공하는 네트워크 관리 시스템이다. 최근 스마트폰의 사용이 활성화 되고 있고 이를 효율적 업무 환경 구축을 위해 활용하고자 하는 사회적 요구가 많이 늘어나고 있다. 현재 사용중인 OAM 시스템을 조사하고 기존의 Pull기술과 Push기술을 분석한 후 스마트폰 기반의 OAM 시스템을 위한 모델을 제시한다.

1. 서론

빠르게 변화 하는 통신 환경에서 운영상에 나타나는 결함(Defects), 이상(Anomalies), 장애(Failures), 고장(Fault) 등을 실시간으로 발견하고 관리자에게 알려주는 시스템의 도입이 필요하게 되었다.

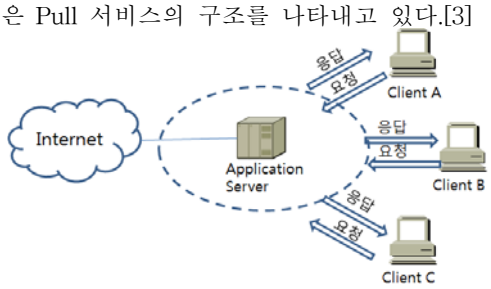
이러한 OAM(Operation, Administration & Maintenance)는 실시간 프로세스 상태 모니터링, 형상 모니터링, 이벤트/장애 모니터링, 로그 관리, 백업/Trace 관리, 명령어 제어, 통계 관리 등 여러 가지 기능을 지원하고 있다. 이 기능은 서비스 지향 구조로 개발되어 다양한 새로운 서비스들이 빠른 속도로 개발될 수 있게 설계 되었다[1].

현재 대부분의 OAM 시스템은 유선 인프라에 국한되어 있다. 최근에는 모바일 네트워크 고도화 및 스마트폰의 보급 확산으로 스마트폰의 사용이 활성화 되면서 이를 효율적 업무 환경 구축을 위해 활용하고자 하는 사회적 요구가 높아지고 있다. 본문은 스마트폰 기반의 통신망 운영 관리에서 관리자에게 보내지는 메시지를 Pull/Push 기술을 사용하여 효율적으로 전달하는 모델을 제시 한다.

본문의 구성은 2장 Pull/Push 기술에 대해서 고찰하고, 3장은 스마트폰 기반 OAM 시스템 모델을 제시하고, 마지막 결론에서는 향후 연구와 보완 사항을 제안한다.

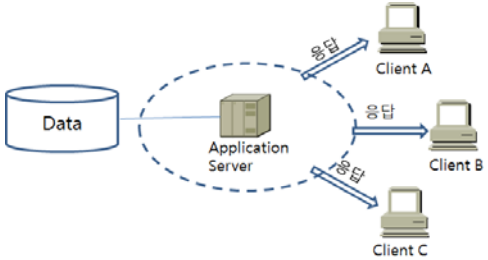
2. 관련 연구

Pull 기술은 정보를 얻기 위해 사용자는 매우 능동적인 정보 검색 행위를 필요로 한다. 서버/ 클라이언트 구조에서 클라이언트는 원하는 정보를 받기 위해서는 주기적으로 방문하여 갱신된 정보를 확인 하는 작업이 필요하다. 그림1은 Pull 서비스의 구조를 나타내고 있다.[3]



(그림1) 일반적인 Pull 기술의 구조

Push 기술은 기존의 Pull기술과는 달리 클라이언트의 요청이 없이 사용 된다. 기존에 클라이언트가 자신이 원하는 정보를 서버에게 등록하고 서버에서 등록한 정보가 준비 되었을 때 그것을 자동적으로 클라이언트에게 전달(Push) 함으로써 사용자가 원하는 정보를 받는 기법이다. 그림2에서는 Push 기술의 구조를 나타내고 있다.



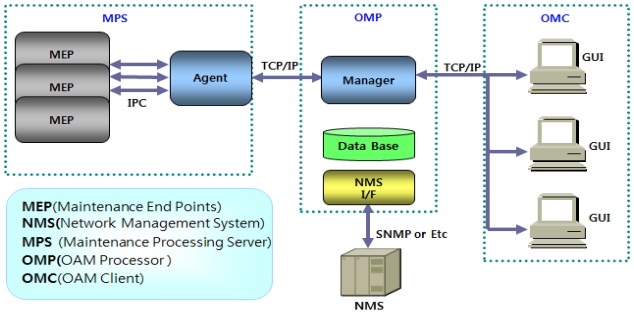
(그림2) 일반적인 Push 기술의 구조

이러한 Push 기술이 원활하게 구현되기 위해서는 seamless

접속과 여러 클라이언트에게 동시에 보낼 수 있는 멀티케스트 기능, 전달 받은 정보가 악의성이 없는 정보인지 출처를 보증하는 기능, 등록된 클라이언트에게만 정보를 수신하게 하는 능력이 필요하다.[2][4]

3. 제안된 스마트폰 기반 OAM 구조

기존의 OAM 시스템은 관리 대상 어플리케이션의 자원을 관리하고 정보를 수집하는 MPS(Maintenance Processing Server)와 다수의 클라이언트와 MPS를 연동 관리하는 OMP(OAM Processor), 원격에서 GUI환경으로 관리하는 OMC(OAM Client) 3가지로 그림3과 같이 구성된다.



(그림3) TCP/IP기반 OAM 구조

MPS의 Agent는 MEP(Maintenance End Point)에게 프로세스 상태, 자원, 이벤트, 통계, 레포트등의 메시지를 주기적으로 Manager에게 전달 한다. Manager는 수신된 메시지 종류별로 DB에 기록 및 로그를 남긴다. Manager는 연결된 클라이언트에게 수신된 메시지를 전달한다. OAM에서 주로 사용되는 메시지는 표1과 같다.

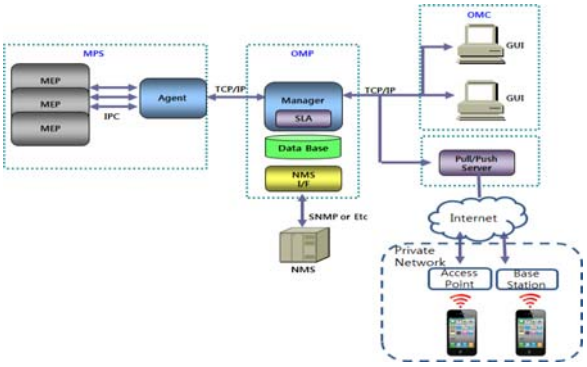
메시지 종류	설 명
Session Connect Message	세션 연결
Configuration Management	형상 정보
Event Management	장애 이벤트
Performance Management	성능 정보
MML Message	MML 명령어
Trace Message	Trace 설정

<표1> OAM 메시지 종류

OAM에서 수많은 메시지 들이 오고가며 각 메시지별 중요도 역시 상이할 것이다. 그중 OAM의 핵심이 되는 장비 이상 정보는 장애 이벤트 메시지로 전달되며 관리자가 즉시 반응할 수 있는 실시간성이 요구 된다.

관리자에게 실시간 메시지와 분석메시지를 효율적으로 제공하기 위해 다중계층 개념을 도입한다. 기존 메시지를 실시간과 분석 메시지로 구분한 후, n개의 MPS를 우선순위 계층으로 분류하여 각 메시지를 등급에 맞추어 분류한다. 실시간 메시지는 Push 기술로 그 외의 분석 메시지는 Pull 기술로 전달 하게한다. 그림 4는 본문에서 제안된 스마트폰 기반 OAM 구조이다.

스마트폰에 전달되는 메시지를 Pull/Push기술로 제안하는 이유는 3G망과 WiFi를 동시에 사용할 수 있는 인터넷 접근성 때문이다. 운영관리에서 주요한 이슈는 실시간 장애



(그림4) 제안된 스마트폰 기반 OAM 구조

메시지이다. 스마트폰에서 어플리케이션이 실행중이 아니더라도 실시간으로 장애 메시지를 받아야 한다. Pull 기술만 사용하여 구현할 경우 관리자가 항상 어플리케이션을 모니터링 하는 불편함과 제한된 화면 구성으로 많은 장비를 관리해야하는 어려움이 존재한다. 또한 Pull 기술은 요청과 응답이 존재하기 때문에 많은 패킷이 전송되며 이에 따라 베터리의 효율성도 떨어지게 된다. 이와 같이 실시간성 메시지는 Push 기술로, 분석 메시지를 Pull 기술로 전달할 경우 높은 효율이 예상된다.

4.결론 및 향후 연구

최근 스마트폰의 보급이 늘어나고 그에 따라 많은 사회적 관심이 높아지고 있다. 본 제안은 기존의 메시지를 관리자 운영 방침에 맞추어 다중 계층으로 분리하였고, 기존에 유선 인프라 기반의 OAM 시스템을 스마트폰 기반의 Pull/Push기술을 사용하여 제안 하였고, 장애 메시지에 대한 실시간을 보장한다.

스마트폰 기반의 운영 관리에 대한 연구는 아직 부족한 상태이다. 특히 Push 기술은 자동적으로 사용자에게 메시지를 전달하기 때문에 Push 서버과 스마트폰 사이에는 신뢰성이 요구된다. 또한 Pull/Push 서버는 많은 메시지를 전달 받게 된다. 스마트폰은 기존의 테스크탑에 비해 성능, 기능의 제약이 존재하기 때문에 이 메시지를 어떻게 처리할 것인지 추가적인 연구가 필요하다.

참고문헌

[1] Requirements for OAM functions in Ethernet-based networks and Ethernet services, ITU-T Y.1730, December 27, 2006

[2] Scheduling time critical requests for multiple data objects in on-demand broadcast, Victor C.S Lee, and Kai Liu, February 2010

[3] J. P Martin-Flatin, "Push vs Pull in Web-based Network Management", Proc 6th IFIP/IEEE International Symposium on Integrated Network Management (IM'99), Boston, MA, USA, pp.3-18, May 1999

[4] Drummond Reed, "Pushing push: Advancing the Feature of Channel Communications", W3C Push Workshop, September 8, 1997