JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 20, No. 9 : 1710~1716 Sep. 2016

무중단 네트워킹 서비스 제공을 위한 서비스 중 소프트웨어 업그레이드 기술 설계 및 구현

윤호선* · 류호용

A design and implementation of an in-service software upgrade technology to provide a seamless networking services

Ho-sun Yoon* · **Ho-yong Ryu**

Hyper-connected Communication Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon 34129, Korea

요 약

네트워크 장비에서 동작하는 소프트웨어의 버그 수정이나 새로운 기능 추가를 위해서 소프트웨어를 업그레이드 할 필요가 있다. 하지만 서비스 중인 소프트웨어를 업그레이드하기 위해서는 네트워크 서비스를 종료한 후에 소프트웨어를 업그레이드해야만 하는 문제가 있다. 이러한 문제를 해결하기 위해서 서비스 중 소프트웨어 업그레이드(ISSU : In-Service Software Upgrade) 기술이 사용된다. ISSU는 네트워크 장비를 오프라인 시키거나 네트워크 서비스를 중단하지 않고 소프트웨어를 업그레이드하는 기술이다. 본 논문에서는 무중단 네트워킹 서비스를 제공하기 위해서 ISSU 기술을 네트워크 OS에 적용하는 방법을 제안하고 구현한다. 본 논문에서는 고가용성 기능을 가지고 있는 한국전자통신연구원에서 개발한 N2OS를 이용하였다. 또한 ISSU 기능이 정상적으로 동작함을 검증하기 위해서 가상 머신 기반의 시험 환경을 만들고 시험을 진행하였다.

ABSTRACT

In general, software upgrade technique is needed to add new features or fix bug of software on a network devices. However, the problem is that the software must be upgraded after the termination of networking service to replace new package. An ISSU(In-Service Software Upgrade) technique is used to solve such the problem. ISSU is a technology to upgrade the software without interrupting the network service or an offline network equipment. In this paper, to provide a seamless networking service, we design and implement an architecture to apply ISSU technique to a network operating system. In this paper, we use high-availability feature in N2OS which has been developed by ETRI. In addition, in order to verify that the implemented ISSU function is operation properly, we proceed to test using a test environment based on a virtual machine.

키워드 : 서비스 중 소프트웨어 업그레이드, 네트워크 OS, 고가용성, 소프트웨어 업그레이드, 스위치오버

Key word : High Availability, ISSU, Network OS, Software Upgrade, Switchover

Received 04 August 2016, **Revised** 08 August 2016, **Accepted** 23 August 2016

* **Corresponding Author** Ho-sun Yoon(E-mail:yhs@etri.re.kr, Tel:+82-42-860-5329)
Hyper-connected Communication Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon 34129, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2016.20.9.1710 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

소프트웨어 및 장비를 개발하는 업체들은 개발된 제품의 안정성과 성능 향상을 위해서 지속적인 소프트웨어 업그레이드를 진행한다. 소프트웨어 업그레이드는 소프트웨어 버그 수정, 필요 없는 기능의 삭제, 새로운 기능의 추가 등과 같은 기능을 수행한다. 일반적으로 소프트웨어 업그레이드를 진행할 때는 서비스를 종료하고 소프트웨어를 업그레이드한 후에 다시 서비스를 실행하는 절차로 수행된다. 하지만 이러한 경우에는 서비스가 중단되는 문제를 야기할 수 있다. 이러한 문제 해결을 위해 서비스 중단 없이 소프트웨어를 업그레이드하는 다양한 기술들에 대한 연구가 진행되고 있다.

서비스 중단 없이 소프트웨어를 업그레이드하기 위한 ISSU(In-Service Software Upgrade) 기술을 적용하기 위해서는 액티브 노드와 스탠바이 노드 상에서 스위치오버를 수행할 수 있어야 한다. 즉, 스텐바이 노드에 새로운 버전의 소프트웨어 패키지를 업그레이드하고, 액티브 노드와 스텐바이 노드를 스위치오버한 후에, 액티브에서 스텐바이로 변경된 노드에 새로운 버전의 소프트웨어 패키지를 업그레이드함으로써 서비스 중단 없이 소프트웨어 업그레이드를 완성한다.

시스코나 쥬니퍼와 같은 주요 네트워크 장비 업체들은 고가용성을 위해서 ISSU 기능을 제공하고 있다. 네트워크 장비 업체에서 제공하는 ISSU 기능은 단순한 소프트웨어에 대한 업그레이드뿐만 아니라 하드웨어를 비롯한 시스템 전체를 고려해서 소프트웨어 업그레이드를 수행한다[1-4].

소프트웨어 수준에서의 ISSU 기능은 SA(Service Availability) 포럼에서 공개하고 있는 OpenSAF에서 제공하고 있다. OpenSAF는 액티브 노드와 스탠바이 노드 사이에서 스위치 오버를 수행하면서 ISSU 기능을 수행할 수 있는 프레임워크를 제공하고 있다[5,6]. 이외에도 SDN(Software Defined Network) 컨트롤러를 위한 소프트웨어 업그레이드 방법뿐만 아니라 다양한 업그레이드 방법들이 연구되고 있다[7-9].

본 논문에서는 한국전자통신연구원에서 개발해서 공개하고 있는 네트워크 OS인 N2OS(Neutralized Network Operating System)에 ISSU 기능을 적용하기 위한 방법을 제안하고 구현한다. 제안하는 방법에는 ISSU 기능을 제어하기 위한 소프트웨어 아키텍처, 업그레이드 시나리오 기술 방법, 그리고 업그레이드 상태에 따른 수행 절차 등을 포함한다. 또한 구현된 ISSU 기능이 정상적으로 동작함을 증명하기 위해서 가상 머신 기반의 시험 환경을 구축하고 테스트 프로그램을 이용해서 ISSU 기능을 검증한다.

이 논문은 2장에서 공개 네트워크 OS인 N2OS에 대해서 설명하고, 3장에서 ISSU 기능 구현을 위한 구조 설계 및 구현 방안을 제안하고, 4장에서는 구현된 ISSU 기능이 정상적으로 동작함을 검증하고 결론을 맺는다.

Ⅱ. N2OS 개요

네트워크 OS는 스위치, 라우터, 전송 장비 등 네트워크 장비에 탑재되어 하드웨어 자원을 관리하고 다양한 네트워크 프로토콜들이 동작할 수 있도록 관리하는 시스템 소프트웨어이며, 한국전자통신연구원은 N2OS라는 네트워크 OS를 공개 소프트웨어로 개발하고 있다. N2OS는 모두 일곱 개의 suite로 구성되어있다.

- Management Suite : 네트워크 장비 및 전체 네트워크를 관리하기 위해서 자동으로 패키지 인스톨 및 장비 설정 등을 수행하며, 인증 및 모니터링 기능을 수행한다.
- Application Adoption Suite : VLAN(Virtual Local Area Network), LACP(Link Aggregation Control Protocol) 및 mSTP(Multiple Spanning Tree Protocol)등과 같은 Layer 2 프로토콜, RIP(Routing Information Protocol), ISIS(Intermediate System to Intermediate System), OSPF(Open Shortest Path First), BGP(Border Gateway Protocol) 등과 같은 Layer 3 프로토콜, 고가용성을 위한 VRRP(Virtual Router Redundancy Protocol), 그리고 Open Flow 기능 등이 구현되어있다.
- Networking Service Management Suite : 프로세스 관리를 위한 Process Manager, CLI(Command Line Interface) 제공을 위한 Command Manager, IPC(Interprocess Communication) 관리를 위한 IPC Manager, 포트 및 인터페이스 관리를 위한 PIF Manager, 라우팅 정보 관리를 위한 RIB Manager, MPLS(Multi-Protocol Label Switch) Label 관리를 위한 Label Manager, Check Point 관리를 위한 Check Point Manager, GRE(Generic

Routing Encapsulation) 등과 같은 다양한 터널을 관리하기 위한 Tunnel Manager, 플로우 관리를 위한 Flow Manager 등이 구현 되어있다.

- HA Suite : 고가용성 지원을 위해서 라이프 사이클 관리 기능, 상태 복제 기능, 스위치오버 기능, DLU (Dynamic Live Upgrade) 기능 등을 제공하고 있다. DLU 기능은 단일 노드에서 동작 중인 소프트웨어를 중단하지 않고 업그레이드할 수 있는 기술로서 버그 수정 등과 같은 간단한 소프트웨어 업그레이드 기능을 수행할 수 있다.
- Network Silicon HAL : 다양한 종류의 네트워크 칩을 수용하기 위한 HAL(Hardware Abstraction Layer) 기능을 수행한다.
- Customized Utility Acceleration Suite : IPC, 이벤트 및 메모리 관리 기능, 공통 라이브러리 등을 제공한다.
- Dedicated Kernel Space Suite : 커널에 존재하는 VLAN 기능, L2 스위칭 기능, Layer 3 포워딩 기능, MPLS 포워딩 기능, 터널링 기능 등을 제공한다.

2014년부터 소스 릴리즈를 시작해서 현재 버전 0.03.00까지 릴리즈된 상태이며, 공개된 소스는 GPLv2 라이센스를 준수하고 있다. 구체적인 N2OS의 소프트웨어 아키텍처 및 개발자 가이드 등이 N2OS 사이트에서 제공되고 있다[10].

Ⅲ. ISSU 기능 설계 및 구현

이 장에서는 Software Upgrade Manager 구조, 업그레이드 시나리오 정보 및 절차를 기술하는 업그레이드 시나리오 파일, 그리고 로드, 스위치오버, 수용 및 중단 절차 등에 대해서 기술한다.

3.1. 소프트웨어 업그레이드 매니저

N2OS에서 ISSU 기능을 수행하기 위해서 소프트웨어 업그레이드 매니저(SUM : Software Upgrade Manager) 블록을 새롭게 추가했다.

그림 1에서 보듯이 ISSU 기능을 수행하기 위한 SUM은 명령어 입력을 위한 Command Manager와의 인터페이스, 스위치오버를 위한 Check Point Manager와의 인터페이스, 소프트웨어 업그레이드 동안에 프로세스를 관리하기 위한 Process Manager와의 인터페이스, 그리고 DLU 기능을 수행하기 위한 프로세스들과의 인터페이스를 갖는다.

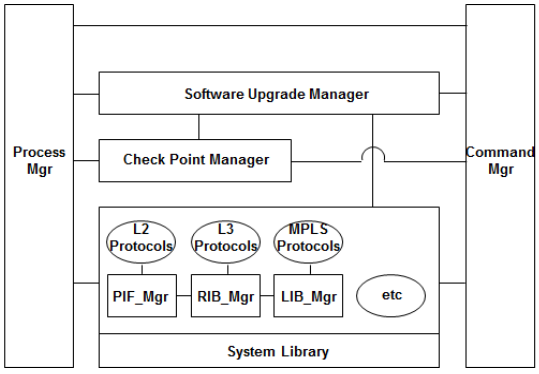

Fig. 1 N2OS Architecture including SUM

그림 2는 ISSU 상태도를 나타낸 것이다.

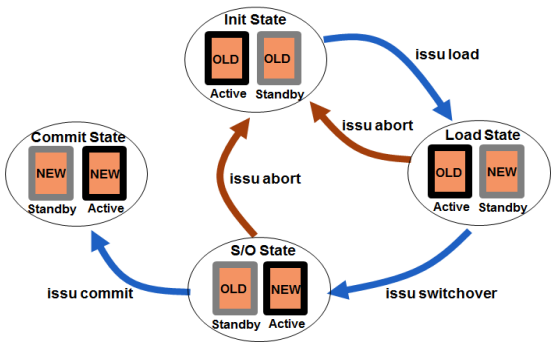

Fig. 2 ISSU Status Diagram

각 상태는 아래와 같은 의미를 갖는다.

- 초기 상태(Init State)는 이전 소프트웨어 버전으로 액티브와 스탠바이 노드가 동작하는 상태이다. ISSU는 동작하고 있는 액티브 노드와 대기 중인 스탠바이 노드로 구성된 redundancy 환경에서 수행된다.
- 로드 상태(Load State)는 액티브 노드가 정상적으로 동작하는 중에 스탠바이 노드에 새로운 소프트웨어 패키지가 로딩된 상태를 나타낸다.
- 스위치오버 상태(S/O State)는 액티브 노드와 스탠바이 노드를 스위치오버한 상태를 나타낸다. 스위치오버 상태는 새로운 소프트웨어 패키지가 액티브 노드

에서 동작하고 있으며, 스탠바이 노드에는 이전 소프트웨어 패키지가 로딩된 상태이다. 스위치오버는 N2OS에서 제공하는 방식을 이용하며, N2OS는 두 노드가 서로의 상태를 실시간으로 공유하는 방식을 제공하고 있다.

- 수용 상태(Commit State)는 스탠바이 노드에 새로운 소프트웨어 패키지를 로딩함으로써 모든 노드에 새로운 소프트웨어 패키지가 로딩된 상태를 나타낸다.

초기 상태에서 로딩 상태로, 로딩 상태에서 스위치오버 상태로, 그리고 스위치 오버 상태에서 수용 상태로 상태를 변경하기 위해서는 사용자가 반드시 개입하도록 설계하였다. 또한 각 상태에서 초기 상태로 돌아가기 위해서 사용자가 업그레이드 중단(abort)을 수행할 수 있도록 하였다. 사용자가 수용을 완료한 상태에서 이전 버전으로 돌아가기 원하는 경우에는 소프트웨어 다운그레이드를 수행한다. 위와 같은 모든 절차는 소프트웨어 업그레이드 매니저가 관리한다.

3.2. 업그레이드 시나리오 기술 파일

소프트웨어 업그레이드 절차를 수행하기 위해서는 업그레이드에 필요한 다양한 정보 및 업그레이드 절차에 대한 정보를 체계적으로 관리할 필요가 있다. 본 논문에서는 이러한 정보를 관리하기 위해서 XML 파일 형태의 업그레이드 시나리오 기술 파일(USD : Upgrade Scenario Description)을 이용한다. USD 파일에 포함되는 내용은 아래와 같다.

- 소프트웨어 패키지 버전, 예상 업그레이드 소요 시간, 스위치오버 후 사용자가 수용 상태로 상태를 변경할 때까지의 최대 대기 시간
- 업그레이드 그룹 이름 및 그룹 업그레이드 예상 소요 시간. 업그레이드 그룹은 업그레이드 최소 단위인 업그레이드 스탭들이 모인 업그레이드 단위임
- 업그레이드 스탭 이름, 예상 소요 시간, 실패 시 최대 재시도 횟수, 재시도하기 위해서 대기하는 시간
- 활성화 및 비활성화 시킬 엔터티 이름, 실행 취소를 위해서 대기하는 시간, 최대 실행 취소 횟수
- 커널에 인스톨 및 언인스톨할 모듈 이름, 실행 취소를 위해서 대기하는 시간, 최대 실행 취소 횟수
- DLU를 수행할 엔터티 이름, 실행 취소를 위해서 대기하는 시간, 최대 실행 취소 횟수
- 현재 수행 중인 소프트웨어 패키지가 저장된 저장소로 특정 소프트웨어를 복사하거나 저장소로부터 삭제할 엔터티 이름, 실행 취소를 위해서 대기하는 시간, 최대 실행 취소 횟수

예상되는 전체 소프트웨어 업그레이드 소요 시간 또는 예상되는 업그레이드 그룹 및 업그레이드 스탭 소요 시간을 초과하는 경우에는 롤백(Rollback)을 수행한다. 롤백은 지금까지 수행한 모든 업그레이드 절차를 역으로 단계별로 취소하는 것을 의미한다.

3.3. ISSU 로드 절차

ISSU 로드 절차는 스탠바이 노드에 설치된 이전 소프트웨어 패키지를 새로운 소프트웨어 패키지로 변경한다. 그림 3은 ISSU 로드 수행 절차를 나타낸 것이다.

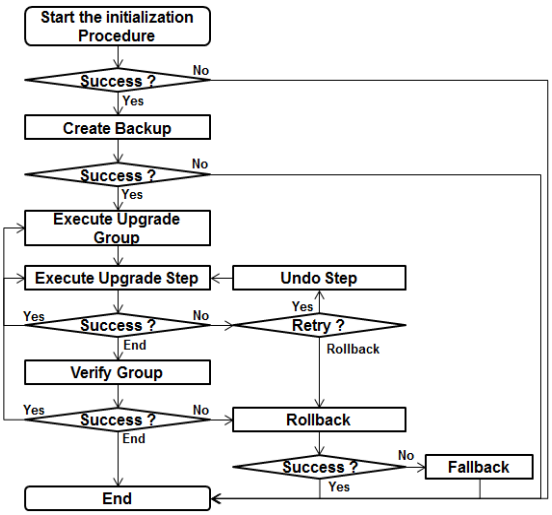

Fig. 3 ISSU Load Procedure

- ISSU 기능을 위한 초기화 절차를 수행한다. 초기화 절차에는 소프트웨어 패키지를 저장소로부터 가져오는 기능을 포함한다.
- 백업 절차를 수행한다. 백업 절차는 업그레이드가 실패한 경우, 백업 상태로 돌아가기 위해서 현재 상태를 저장하는 절차이다.
- 업그레이드 그룹을 수행한다. 업그레이드 그룹은 업그레이드 절차를 그룹핑하는 용도로만 사용된다.

- 업그레이드 스탭을 수행한다. 만약 실패하면 재시도할지 여부를 결정한다. 재시도를 수행하는 경우에는 스탭 수행을 취소(Undo)하고 다시 스탭을 수행한다. 만약 최대 재시도 횟수를 초과하거나 재시도를 하지 않는 경우에는 롤백을 수행한다. 롤백이 실패하면 폴백(Fallback)을 수행한다. 폴백은 실행 중인 모든 소프트웨어를 이전 소프트웨어 버전으로 한 번에 변경한다.
- 모든 업그레이드 그룹 절차가 정상적으로 수행되면 업그레이드 절차를 완료한다.

3.4. ISSU 스위치오버 절차

ISSU 스위치오버는 N2OS에서 제공하는 방식을 활용하며, 스위치오버를 완료한 후에 새로운 소프트웨어 패키지는 액티브 노드에서 구동되며 이전 소프트웨어 버전은 스탠바이 노드에서 구동된다. 그림 4는 ISSU 스위치오버 절차를 나타낸 것이다.

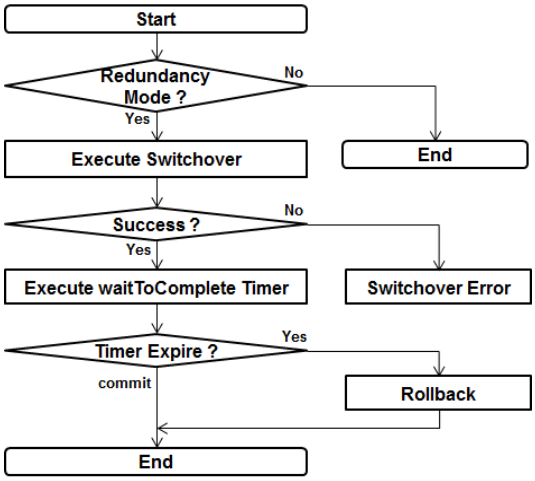

Fig. 4 ISSU Switchover Procedure

- 스위치오버를 수행할 때 redundancy 모드인지 simplex 모드인지 여부를 확인한다. redundancy 모드는 액티브 노드와 스탠바이 노드가 상태를 공유하는 방식을 의미하며, simplex 모드는 단일 노드에서 수행되는 것을 의미하며, 이 경우에는 스위치오버를 수행할 수 없다.
- 스위치오버가 실패한 경우에는 N2OS에서 에러 절차를 수행하며, ISSU 절차는 롤백을 수행한다.
- 스위치오버가 완료된 후에 사용자가 정해진 시간 내에 수용 절차를 수행하지 않으면 자동으로 롤백을 수행한다.

3.5. ISSU 수용 절차

ISSU 수용 절차는 두 노드 모두에서 새로운 버전의 소프트웨어 패키지가 구동되도록 한다. simplex 모드인 경우에는 로드 및 스위치오버 절차 없이 수용 절차만 수행하면 된다. 그림 5는 ISSU 수용 절차를 나타낸 것이다.

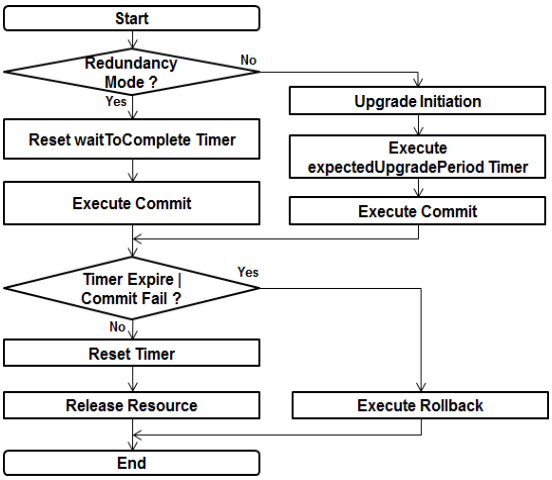

Fig. 5 ISSU Commit Procedure

- redundancy 모드인 경우에는 타이머를 리셋하고 수용 절차를 진행한다. 수용 절차는 로드 절차와 동일하다. simplex 모드인 경우에는 시나리오에 따라서 업그레이드를 진행한다. 이 모드에서는 DLU가 사용될 수 있으며, 경우에 따라서는 서비스가 중단될 수 있다.
- 성공한 경우에는 모든 타이머나 자원을 릴리즈하며, 실패한 경우에는 롤백을 수행한다.

수용 절차를 완료하면 모든 소프트웨어 업그레이드 절차가 완료된다.

3.6. ISSU 중단 절차

사용자는 ISSU를 중단할 필요가 있는 경우에 중단 절차를 수행할 수 있다. 중단 절차는 롤백 절차와 동일하며 롤백이 실패하는 경우에는 폴백을 수행한다. 폴백이 실패하는 경우에는 백업 상태로 돌아가며, 이 경우에는 시스템 리부팅을 수행한다.

Ⅳ. 시험 환경 및 결과

ISSU 기능이 정상적으로 동작하는지 여부를 판단하기 위해서 가상 머신을 이용해서 두 노드에 ISSU 기능을 포함하는 N2OS를 구동시킨 시험 환경을 구축하였다. 구축된 시험 환경에서 ISSU 기능을 시험하기 위한 테스트 프로그램을 만들었다. 테스트 프로그램은 이전 버전에서 3초마다 1씩 더하고 새로운 버전에서 3초마다 2씩 더하는 기능을 수행한다.

구체적인 시험 절차는 그림 6과 같다. 그림 6에서 보듯이 로드, 스위치오버, 수용 단계를 수행하는 중간에 중단 절차들을 수행함으로써 모든 단계에서 기능이 정상적으로 동작하는지 여부를 확인하도록 하였다.

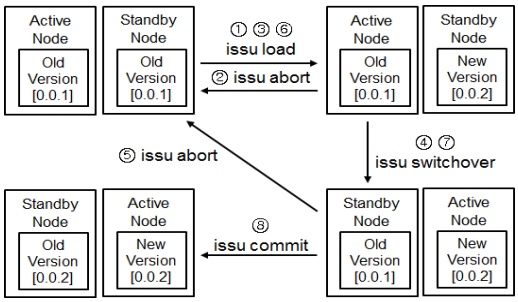

Fig. 6 ISSU Test Procedure

```
12:01:06.259][LOG_DEBUG] Auto Increment [11] Data1 [727] Data2 [Checkpoint_Test]
12:01:09.255][LOG_INFO] Ver[0.0.1] - My Name[UPDMGR_TEST], Process Type[25], Ipc
12:01:09.259][LOG_DEBUG] Auto Increment [12] Data1 [727] Data2 [Checkpoint_Test]
12:01:12.255][LOG_INFO] Ver[0.0.1] - My Name[UPDMGR_TEST], Process Type[25], Ipc
12:01:12.259][LOG_DEBUG] Auto Increment [13] Data1 [727] Data2 [Checkpoint_Test]

12:01:13.254][LOG_DEBUG] gNosCompName : 25, gBeforeRole : 4, gAfterRole : 6
12:01:13.254][LOG_DEBUG] before lcsSetRole.haState : 4
12:01:13.254][LOG_DEBUG] after lcsSetRole.haState : 6
                    ⋮
12:01:13.301][LOG_DEBUG] UPDMGR_TEST Set Role() Start
12:01:13.301][LOG_DEBUG] UPDMGR_TEST Set Role() End

12:01:13.355][LOG_DEBUG] ------------------------------
12:01:13.355][LOG_DEBUG] Checkpoint Read Data
12:01:13.355][LOG_DEBUG] ------------------------------
12:01:13.355][LOG_DEBUG] Auto Increment [15] Data1 [727] Data2 [Checkpoint_Test]
12:01:13.355][LOG_DEBUG] ------------------------------
12:01:15.256][LOG_INFO] Ver[0.0.1] - My Name[UPDMGR_TEST], Process Type[25], Ipc
12:01:16.357][LOG_DEBUG] ------------------------------
12:01:16.357][LOG_DEBUG] Checkpoint Read Data
12:01:16.357][LOG_DEBUG] ------------------------------
12:01:16.357][LOG_DEBUG] Auto Increment [17] Data1 [727] Data2 [Checkpoint_Test]
12:01:16.357][LOG_DEBUG] ------------------------------
```

Fig. 7 Log of test program

그림 7은 그림 6에서 4번 절차를 수행한 후에 액티브 노드에서 출력한 테스트 프로그램의 로그를 나타낸 것이다. 그림 7에서 보듯이 이전 버전이 동작하는 경우에 3초마다 1씩 증가하다가 스위치오버가 발생해서 새로운 버전인 0.0.2 버전이 액티브 노드에서 동작하는 경우에 3초마다 2씩 증가하는 것을 볼 수 있다. 그림 7에서 알 수 있듯이 본 논문의 시험에서는 N2OS에서 제공하는 Check Point 기능을 이용하기 때문에 증가되는 값이 계속 유지되고 있다.

그림 8은 그림 6의 절차에 따라서 ISSU 절차를 수행한 후에 액티브 노드에서 ISSU 동작 이력을 출력한 것이다. 그림 8에서 보듯이 그림 6의 시험 절차와 동일하게 동작하였음을 확인할 수 있다. 또한 각 절차가 수행되는 시간은 1초미만으로 확인되고 있다. 다만, ISSU 기능 수행을 위해서 소요되는 시간은 스위치오버를 비롯한 많은 경우의 수가 존재하기 때문에 큰 의미를 두기는 어려울 것으로 판단된다.

```
lcw-sso_standby(exec)#lcw-sso_stanstory
Upgrade Date    Old Version     New Version     Last State
20160727 11:36:20               0.0.1           0.0.1   ISSU_LOAD_DOING
20160727 11:36:21               0.0.1           0.0.1   ISSU_LOAD_DONE
20160727 11:36:23               0.0.1           0.0.1   ISSU_ABORT_DOING
20160727 11:36:23               0.0.1           0.0.1   ISSU_ABORT_DONE
20160727 11:36:28               0.0.1           0.0.1   ISSU_LOAD_DOING
20160727 11:36:28               0.0.1           0.0.1   ISSU_LOAD_DONE
20160727 11:36:31               0.0.1           0.0.1   ISSU_SWITCHOVER_DOING
20160727 11:36:31               0.0.1           0.0.1   ISSU_SWITCHOVER_DONE
20160727 11:36:38               0.0.1           0.0.1   ISSU_ABORT_DOING
20160727 11:36:41               0.0.1           0.0.1   ISSU_ABORT_DONE
20160727 11:36:45               0.0.1           0.0.1   ISSU_LOAD_DOING
20160727 11:36:46               0.0.1           0.0.1   ISSU_LOAD_DONE
20160727 11:36:53               0.0.1           0.0.1   ISSU_SWITCHOVER_DOING
20160727 11:36:53               0.0.1           0.0.1   ISSU_SWITCHOVER_DONE
20160727 11:36:58               0.0.1           0.0.1   ISSU_COMMIT_DOING
20160727 11:36:59               0.0.1           0.0.1   ISSU_COMMIT_DONE
20160727 11:36:59               0.0.2           0.0.2   ISSU_DONE
lcw-sso_standby(exec)#
```

Fig. 8 ISSU Procedure History

Ⅴ. 결 론

본 논문에서는 서비스 중단 없이 소프트웨어를 업그레이드할 수 있는 ISSU 기능을 공개 네트워크 OS인 N2OS에 추가하기 위한 방안을 제안하였다. 또한 제안된 방법을 이용해서 실제로 ISSU 기능을 구현한 후에 간단한 시험을 통해서 ISSU 기능이 정상적으로 동작함을 보였다. 본 논문에서 제안한 방법은 소프트웨어에 대한 업그레이드만을 고려한 것이며 시스템 차원에서의 업그레이드는 고려하지 않았다. 추후에는 소프트웨어뿐만 아니라 시스템 차원에서의 업그레이드를 제공하는 방안에 대한 연구가 필요하다. 또한 본 논문에서

개발된 기술을 네트워크 자동화 시험 기술 등과 연계하는 방안에 대해서 연구를 진행할 필요가 있다.

ACKNOWLEDGMENTS

This work was supported by the ICT R&D program of MSIP/IITP. [R0101-15-0070, Development of The High Availability Network Operating System for Supporting Non-Stop Active Routing]

REFERENCES

[1] CISCO, Cisco IOS Software: Guide to Performing In-Service Software Upgrades [Internet]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/high-availability/prod_white_paper0900aecd80456d57.pdf.

[2] CISCO, Cisco IOS High Availability (HA) - In-Service Software Upgrade (ISSU) Technical Overview [Internet]. Available: http://www.cisco.com/c/dam/en/us/products/ollateral/ios-nx-os-software/high-availability/prod_presentation0900aecd80456cb8.pdf.

[3] CISCO, UNIFIED ISSU: A COMPLETE APPROACH TO IN-SERVICE SOFTWARE UPGRADES [Internet]. Available: http://junipeer.ru/image/t-series/Unified%20SSU%20A%20Complete%20Approach%20to%20In-Service%20Software%20Upgrades.pdf.

[4] JUNIPER, Junos OS High Availability Configuration Guide [Internet]. Available: https://www.juniper.net/techpubsen_US/junos12.2/information-products/topic-collections/config-guide-high-availability/config-guide-high-availability.pdf.

[5] SA Forum, Service Availability Forum Application Interface Specification SAI-AIS-SMF-A.01.02 [Internet]. Available: http://www.saforum.org/hoa/assn16627/images/ai-ais-smf-a.01.02.pdf.

[6] SA Forum, Service Availability Forum Application Interface Specification SAI-AIS-AMF-B.04.01 [Internet]. Available: http://www.saforum.org/HOA/assn16627/mages/SAI-AIS-AMF-B.04.01.pdf.

[7] K. Saur, J. Collard, N. Foster, A. Guha, L. Vanbever and M. Hicks. (2016, March). Safe and Flexible Controller Upgrades for SDNs. *Symposium on SDN Research* [Online]. pp. 1-13. Available: http://www.cs.umd.edu/mwh/papers/sdnupdate-submitted.pdf.

[8] P. Hosek and C. Cadar. (2013, May). Safe Software Updates via Multi-version Execution. *International Conference on software Engineering* [Online]. pp. 612-621. Available: http://srg.doc.ic.ac.uk/files/papers/mx-icse-13.pdf.

[9] K. P. Bhat and M. D. Hall, Interception proxy-based approach for in-service software upgrade, US Patent 0,295,088, to Samsung Electronics CO., LTD, CONLEY ROSE PC, Dallas, 2008.

[10] Electronics and Telecommunications Research Institute. Open N2OS(Neutralized Network Operating System) [Internet]. Available: http://openn2os.etri.re.kr.

윤호선(Ho-Sun Yoon)

2011년2월 순천향대학교 공학박사
현재 한국전자통신연구원 선임연구원
※관심분야 : 네트워크 OS, SDN/NFV, 네트워크 보안

류호용(Ho-Yong Ryu)

1999년2월 광운대학교 공학박사
현재 한국전자통신연구원 네트워크SW플랫폼연구실 실장
※관심분야 : 네트워크 OS, SDN/NFV, OCP