확장 가능한 SFC 프로비저닝을 위한 Monitoring as a Service (MaaS)

이지수, 염상길, 추현승
성균관대학교 소프트웨어대학
e-mail:{jisoo49, sanggil12, choo}@skku.edu

Monitoring as a Service (MaaS) for scalable SFC provisioning

Jisoo Lee, Sanggil Yeom, Hyunseung Choo
College of Software, Sungkyunkwan University

요 약

오늘날 네트워크 서비스에 대한 수요가 증가함에 따라 네트워크 트래픽 또한 증가하고 있다. 이는 네트워크 서비스 구성을 복잡하게 만들어 최종 사용자에게 전달되는 네트워크 기능의 동적 구성이 요구된다. Service Function Chaining (SFC)은 일련의 Service Function (SF) 세트로 구성된 새로운 네트워크 서비스 배포 모델이다. SFC는 특정 서비스에 따라 플로우를 분류하여 네트워크 운영자의 효율적인 서비스 제공을 보장한다. SFC의 성능은 SFC를 구성하는 SF Instance (SFI)와 상관 관계를 가진다. 이때 다수의 SFC 환경에서 단일의 SFI 사용 시, 트래픽 오버플로우가 발생할 수 있다. 따라서, 본 논문에서는 확장 가능한 SFC 프로비저닝을 위한 Monitoring as a Service (MaaS)를 제안한다.

1. 서론

오늘날 인터넷 사용자에게 제공되는 네트워크 서비스는 네트워크 운영자의 서비스 요구 사항 및 정책을 충족시키기 위해 정적으로 구성된 네트워크 토폴로지이다. 서비스에 대한 수요가 커짐에 따라 네트워크 트래픽이 증가하여 복잡한 서비스 구성 문제를 야기한다. 따라서 네트워크 서비스에서 최종 사용자에게 전달되는 네트워크 기능의 동적 구성을 위한 유연성 있는 모델이 요구된다.

Service Function Chaining (SFC)은 일련의 Service Function (SF) 세트로 구성된 새로운 네트워크 서비스 배포 모델이다[1]. SFC는 특정 네트워크 서비스 및 응용 프로그램 트래픽의 유연한 관리가 가능하여 네트워크 동적성과 관리 비용 절감을 보장한다. SF 세트는 Service Function Path (SFP)라고 하는 트래픽 전달 그래프의 특정 위치에 배치된다. SFC는 네트워크 서비스에 따라 플로우를 분류하여 적절한 정책을 배포한다. 이는 네트워크 운영자에게 효율적인 서비스 제공을 보장할 수 있다.

Software Defined Networking (SDN)과 Network Function Virtualization (NFV)은 SFC에 중요한 역할을 하는 기술이다. SDN은 데이터 플레인과 제어 플레인을 분리하여 SFC 간의 네트워크 트래픽을 동적으로 제어한다. 또한 SDN 컨트롤러는 SFC를 구성하는 가상 또는 물리적인 일련의 SF Instance (SFI) 세트를 구성한다. NFV는 네트워크 운영자가 인프라에서 소프트웨어 응용 프로그램으로 배포하는 과정에서 더욱 유연하게 SF를 관리하고 조율하도록 돕는다[1].

유망한 SFC의 발전에도 불구하고 SDN 및 NFV 기술 기반 SFC에서는 해결해야 할 과제들이 있다. 주요 이슈 중 하나는 SFC의 성능 유지이다. SF는 네트워크에 다수의 분산된 인스턴스들과 함께 존재할 수 있기 때문에 SFC의 성능은 SFI의 성능과 상관 관계가 있다. 특히 단일의 SFI가 다수의 SFC에서 공유되는 경우 트래픽이 처리 용량을 초과하여 트래픽 오버플로우가 쉽게 발생할 수 있다. 그러므로 다수의 SFC 환경에서 SF 실패에 동적으로 대처 및 예방할 수 있는 모니터링 시스템이 요구된다.

따라서, 본 논문에서는 확장 가능한 SFC 프로비저닝을 위한 Monitoring as a Service (MaaS)를 제안한다. 제안 기법은 Zabbix 모니터링 솔루션을 이용한다. Zabbix에서 지원하는 모니터링 에이전트를 통해 각 SFI로 수신되는 트래픽 정보를 실시간으로 모니터링 한다. 트래픽 로드가 높은 SFI가 발견되면 해당 SFI를 포함하는 SFP는 새로 인스턴스화 된 SFI를 포함하는 SFP로 변경한다. SFI의 상태에 따라 네트워크 플로우를 분산시킴으로써 다수의 SFC는 안정된 SF를 공유하여 사용할 수 있다. 이는 SFC의 확장성 및 실패 관리의 효율성 향상에 기여할 것이다.

기존 연구인 분산 SF 실패 복구 메커니즘[2]은 실패한 SF를 복구하기 위해 이전 SF를 현재 SFP에 동적으로 통합한다. 그러나 이 연구에서는 SFI의 과부하를 고려하지 않았으며, 구현 결과를 제시하지 않았다. SDN 컨트롤러 오픈 소스에서도 몇 가지 SFC 관리 기법을 제안한다[3].

OpenDayLight (ODL)은 SFI를 선정하는 SFC 관리 기능을 통해 실제 SFP에서 미리 선정한 SFI로 변경 및 적용할 수 있다[4]. 하지만 ODL 컨트롤러도 SFI의 과부하 및 실패를 처리할 수 있는 메커니즘을 제공하지 않는다.

본 논문은 다음의 구성을 따른다. 먼저, 2장에서는 제안 기법에 바탕이 되는 관련 연구를 설명한다. 3장에서는 프레임워크와 MaaS의 동작 과정을 설명한다. 마지막으로 4장에서는 결론과 향후 진행할 연구에 대해 설명한다.

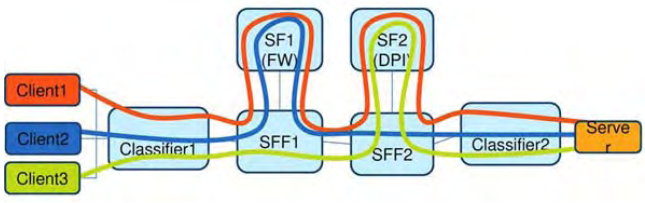
2. 관련 연구

2.1 NFV Management and Orchestration (MANO)

NFV MANO의 주요 기능 블록은 Virtualized Infrastructure Manager (VIM), VNF Manager (VNFM), NFV Orchestrator (NFVO) 이다[5]. VIM은 NFV의 물리적 자원과 가상화 자원을 관리하고 성능 측정을 위해 이를 수집한다. VNFM은 단일 또는 다수의 VNF 인스턴스에 배치되어 생명주기를 관리한다. NFVO는 다수의 VIM에 걸쳐 가상 자원을 통합적으로 조정하는 역할을 수행한다. 또한 네트워크 서비스의 인스턴스화, 스케일링, 초기화를 포함하는 생명주기를 관리한다. 이때 NFV MANO 환경에서 최종 사용자에게 신뢰성 있는 네트워크 서비스를 보장하기위해 VNF 스케일링 기술을 적용할 수 있다.

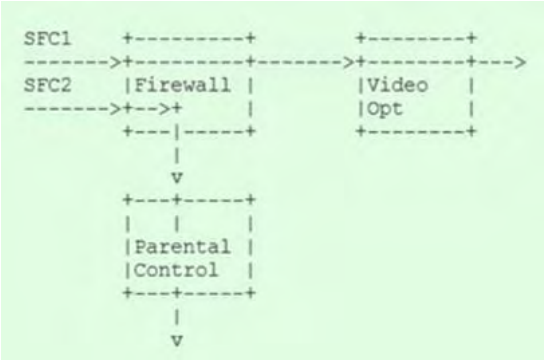
2.2 SFC 네트워크 모델

SFC는 네트워크 서비스 공급자가 특정 네트워크 서비스를 동적으로 구성하고 자동으로 관리할 수 있다. 그림 1은 SFC 구조이며[6], 크게 SFC 분류자, Service Function Forwarder (SFF), SF로 구성된다. SFC 분류자는 네트워크 트래픽이 들어오면 각 서비스를 식별하고 해당 SFF와 SFP에 정책을 할당한다. 이는 트래픽이 대상 서비스에 따라 사전 정의된 정책을 기반으로 분류되어 전송되도록 한다. SFF는 트래픽을 SF 또는 다른 SFF로 전달한다. SF는 Firewall, Deep Packet Inspection (DPI), Network Address Translation (NAT)와 같은 기능을 수행한 다음 다시 SFF로 트래픽을 반환한다[7].



(그림 1) SFC 구조

그림 2는 IETF SFC WG에서 제시한 SFC 트래픽 조정에 대한 유스케이스를 나타낸다[8]. 일부 SF은 다수의 SFC를 지원할 수 있다는 전제 하에 설명한다. SFC1은 Firewall, Video optimizer로 구성된 SFP를 가진다. SFC1의 네트워크 서비스 사용자는 안전한 웹 서핑과 비디오 최적화 서비스를 제공받는다. SFC2는 Firewall, Parental Control로 구성된 SFP를 가진다. SFC2의 네트워크 서비스 사용자는 자녀 보호 기능이 있는 안전한 웹 서핑 서비스을 제공받는다. 이때 SF 중 Firewall은 SFC1과 SFC2에 의해 두 사용자에게 공유되어 트래픽 오버플로우로 인한 네트워크 서비스에서의 문제가 발생할 수 있다.

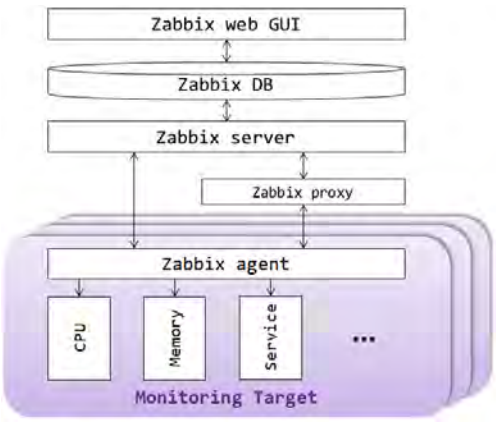


(그림 2) SFC 트래픽 조정 유스케이스

2.3 Zabbix 모니터링 솔루션

Zabbix는 엔터프라이즈 급 분산 모니터링 솔루션이다. 네트워크의 수많은 매개변수와 서버의 상태 모니터링을 제공한다[9]. Infrastructure-level, Cloud environment-level, VM-level, Service-level과 같은 계층에서의 모니터링을 지원한다. 또한, Zabbix는 이벤트 기반의 트리거를 구성할 수 있다. 트리거는 사용자 정의된 임계값을 초과할 경우 알람을 전송한다. 알람은 SMS 또는 이메일 등의 실시간 통지 메커니즘으로 제공한다. 따라서 네트워크 문제가 발생할 경우 신속하게 판단하고 대처할 수 있다.

그림 3은 Zabbix 구조이며, 크게 에이전트와 서버로 구성된다. 에이전트는 모니터링 대상에 배포되어 물리적 머신 및 가상 머신의 자원을 모니터링 한다. 서버는 에이전트와의 상호작용을 통해 정보를 수집하고 트리거 및 알람 기능을 수행한다. 수집된 정보는 Zabbix 웹 서버를 통해 통합적으로 모니터링 할 수 있다. SFC 환경에서는 SF를 모니터링 하여 결함 발생 시 즉각적인 조치를 취해야하므로 Zabbix 모니터 솔루션을 필요로 한다.



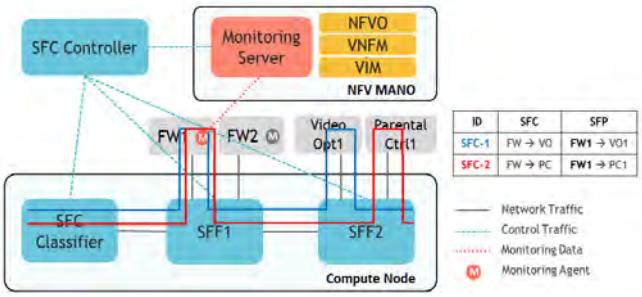
(그림 3) Zabbix 구조

3. Monitoring as a Service (MaaS)

3.1 시스템 프레임워크

그림 4는 MaaS 기능을 수행하는 제안 시스템의 프레임워크이다. 시나리오는 앞에 SFC 관련 연구에서 설명한 SFC 트래픽 조정 유스케이스를 적용한다. 하나의 SF이 다수의 SFC에 의해 공유되고 있다. SFF1에는 FW1, FW2가 연결되어 있고, SFF2에는 Video Opt1과 Parental Ctrl1가 연결되어 있다. 구성요소는 크게 SFC 컨트롤러, 모니터링 에이전트, 모니터링 서버, NFV MANO가 있다. SFC 컨트롤러는 ONOS 오픈 소스의 SDN 컨트롤러를 활용해 SFP 정책과 제어 메시지를 SFC 분류자와 SFF에게 전달한다. 모니터링 에이전트는 각 SFI에 배포되어 트래픽 로드를 모니터링 한다. 모니터링 서버는 모니터링 에이전트로부터 트래픽 로드 정보를 수집하고 이를 미리 정의된 임계값과 비교하여 트리거 발생 여부를 검토한다. NFV MANO는 SFI 스케일링 동작을 수행하여 유연성 있는 네트워크 서비스를 제공한다. NFV MANO에 실행 가능한 SFI 스케일링 동작은 다음과 같다.

SCALE_OUT: 인스턴스의 특정수를 확장
SCALE_IN: 인스턴스의 특정수를 축소
SCALE_OUT_TO: 인스턴스의 특정수로 확장
SCALE_IN_TO: 인스턴스의 특정수로 축소

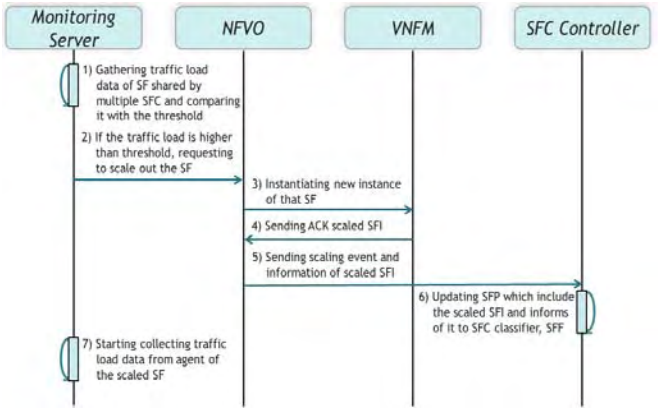


(그림 4) MaaS 프레임워크

3.2 동작 과정

그림 5에 나타난 MaaS의 동작 과정은 크게 4단계로 구분된다. 첫 번째는 모니터링 에이전트가 자동적으로 SFI에 배치되는 MaaS 배포 단계이다. 두 번째 단계는 네트워크 서비스가 시작되고 MaaS가 동작되어 SFI으로 들어오는 트래픽의 로드를 수집하는 모니터링 단계이다. 세 번째는 수집한 트래픽 로드와 임계값과의 비교를 통해 현재 SFI 실패 가능성을 검토하고 백업용 SFI의 존재 여부를 판단하는 결정 단계이다. 네 번째 단계는 결정 단계를 기반으로 SFP 변경 작업을 수행하는 실행 단계이다.

첫 번째 배포 단계는 특정 상황에 자동적으로 SFI에 모니터링 에이전트를 배치한다. 예를 들어, NFV MANO에 의해 SFI가 새로 인스턴스화 되면 Zabbix에서는 활성 모니터링 에이전트를 자동 등록할 수 있다. 자동 등록은 이전에 등록되지 않은 활성 에이전트가 확인을 요청하는 경우 이루어진다. 따라서 새로운 SFI가 생성되는 동시에 해당 인스턴스의 트래픽 로드 데이터를 수집한다.



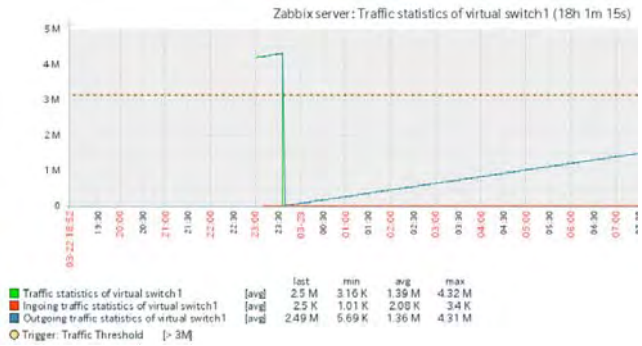
(그림 5) MaaS 동작 과정

두 번째 모니터링 단계에서 사용자 요구로 인한 네트워크 서비스의 수요가 들어오면 MaaS는 FW1으로 수신되는 트래픽 로드를 모니터링 한다. 처음에 두 SFC (SFC1, SFC2)는 모두 FW1을 사용한다. 이때 결정 단계에서 FW1으로 들어오는 트래픽 로드와 임계값을 비교하여 조건이 충족되면 알람을 발생시킨다. 각 SFI 상태에 따른 알람 조건을 표현할 때 고려해야 할 사항이 있다. 먼저, SFI의 특성을 고려하여 비교 메트릭을 선정해야 한다. 예를 들어, CPU와 연관성이 높은 SF라면 CPU 로드를 메트릭으로 선정한다. 제안 기법에서는 CPU/메모리 로드의 빈번한 변화로 결정 단계에서 오류가 발생할 수 있어 축적 데이터인 트래픽 로드를 메트릭으로 설정한다. 또한, 해당 메트릭의 임계값을 적절한 값으로 선정해야한다. 일반적인 SF의 트래픽 로드를 수집하여 평균값을 추정한다. 또한 백업 SFI 여부를 확인하는 임계값, 백업 SFI로 변경하는 임계값 등 단계별 임계값이 존재할 수 있다.

세 번째 결정 단계는 알람을 기반으로 SFI 스케일링 동작을 결정한다. MaaS를 통한 모니터링 단계에서 수집한 FW1의 트래픽 로드를 임계값과 비교한다. 임계값을 초과할 경우 FW1의 실패 가능성이 높다고 판단한다. 비교 결과로 트리거가 발생하면 NFV MANO는 어떤 조치를 취해야 할지 판단한다. 백업용 SFI의 존재 여부를 판단하고, 존재하지 않을 경우 새로운 백업용 SFI를 생성한다. 스케일링 동작을 결정할 뿐만 아니라 시스템에서 실현 가능한 동작인지를 판단한다. 예를 들어, scale-out 동작은 추가적인 인스턴스를 구성할 자원이 충분한지 고려해야한다.

네 번째 실행 단계는 결정 단계에서 요구하는 스케일링 동작 및 SFP 변경 작업을 수행한다. NFVO는 SFI 스케일링 동작 리스트를 포함하며 VNFM으로 SF에 대한 새로운 인스턴스 생성을 요청한다. VIM은 자원을 할당하고 VNFM은 스케일링 동작으로 생성된 백업용 SFI 스케일링 동작을 실행한다. NFVO는 백업용 SFI인 FW2 존재 여부를 확인하고 SFC Controller로 FW2에 대한 정보를 보낸다. SFC Controller는 FW2 정보를 기반으로 어떤 SFC의 SFI를 FW2로 사용할지 고려하여 SFP를 업데이트하고 관련 정책과 함께 SFC 분류자, SFF에게 알린다.

네트워크 서비스를 제공하는 SFI의 트래픽 로드가 매우 높아지면 SFI 실패로 인한 패킷 손실이 발생한다. 그림 6에서처럼 MaaS는 네트워크 트래픽 로드 상태에 대한 실시간 모니터링을 제공한다. 트래픽 로드가 임계값인 3M 바이트 이상이 되면 트리거가 발생한다. MaaS는 트리거 액션으로 위험 상황을 통지하는 알람을 보냄으로써 NFVO가 SFI scale-out 동작을 수행하도록 한다. NFVO는 백업용 SFI의 존재 여부를 확인하여 VNFM에게 새로운 SFI 생성을 요청한다. VIM과 VNFM은 SFI를 생성하고 NFVO에게 응답한다. 생성된 백업용 SFI가 기존의 네트워크 서비스를 제공하는 SFC에 사용된다.



(그림 6) 네트워크 트래픽 로드 모니터링

4. 결론 및 향후 연구

본 논문에서는 확장 가능한 SFC 프로비저닝을 위한 Monitoring as a Service (MaaS)를 제안하였다. 기존의 SDN/NFV 기반 SFC 환경에서 발생하는 SFI 성능 관리 난제를 해결한다. 이때 모든 시스템 관리에 있어 필수이자 기본적인 기능인 모니터링을 활용한다. 제안하는 MaaS는 활성화된 SFI에 서비스로써의 모니터링 기능을 배치하는 것이다. 트래픽 로드를 모니터링 하여 미리 정의된 조건에 따라 알림을 보낼지 판단한다. 알람에 따라 새로운 SFI를 할당하거나 제거하는 스케일링 동작을 수행한다. 이를 통해 네트워크 서비스를 구성하는 SFC 간의 안정적인 SF 공유가 가능하며 SFI의 오버플로우로 인한 패킷 손실 비율을 줄인다. 현재 Zabbix 모니터링 솔루션의 모니터링 동작과 Mininet을 이용한 네트워크 토폴로지 구성을 완료하였으며, 추후 IETF SFC WG에서 제시한 시나리오에 따라 필요한 SF를 설치하고 ONOS 애플리케이션을 개발할 것이다. 또한, 각 SF의 특성을 고려한 임계값 선정에 있어 심층적으로 연구하고자 한다.

ACKNOWLEDGEMENT

본 논문은 기초연구사업 (NRF-2010-0020210)과 과학기술정보통신부 및 정보통신기술진흥센터의 Grand ICT연구센터지원사업 (IITP-2018-2015-0-00742), 방송통신인프라 원천기술개발사업 (2014-3-00547, 자율 제어 네트워킹 및 자율 관리 핵심 기술 관리)의 연구결과로 수행되었음

참고문헌

[1] 안상현, “SFC(Service Function Chaining) 기술 소개 및 표준화 동향”, 2015.

[2] D. Suh et al., "Distributed Service Function Failover Mechanism in Service Function Chaining", Proc International Conference on Information Networking (ICOIN) 2017, January 2017.

[3] ONF White Paper. ONF TS-027: L4-L7 Service Function Chaining Solution Architecture [online] Available: https://www.opennetworking.org/, 2015.

[4] ODL, [online] Available: https://wiki.opendaylight.org/viewService_Function_Chaining:Main.

[5] ETSI, “Network Functions Virtualisation (NFV); Architectural Framework,” ETSI GS NPV 002 V1.2.1, December 2014.

[6] Yi Yang, "Only use FD.IO vpp to achieve high performance service function chaining", Inter, 2017.

[7] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", (http://www.rfc-editor.org/info/rfc7665), 2015.

[8] Liu, W., et al. "Service function chaining (sfc) general use cases." Work in progress, IETF Secretariat, Internet-Draft draft-liu-sfc-use-cases-08, 2014.

[9] https://en.wikipedia.org/wiki/Zabbix