Software-Defined Network 에서의 Conflict Resolution 을 위한 정책엔진 구조 및 전략 분석

윤대건, 오상윤
아주대학교

kljp@ajou.ac.kr, syoh@ajou.ac.kr

Analysis on a Rule Engine and Strategy for Conflict Resolution of Software-Defined Network

Daegun Yoon, Sangyoon Oh
Ajou Univ.

요 약

본 논문은 Software-Defined Network 에서 추가되는 정책들이 기존의 정책들과 일으키는 충돌을 해결하기 위해 존재하는 정책엔진의 새로운 설계를 위해 분석한 기존 연구 내용들을 소개하고 분석한 내용들을 기반으로 정책엔진의 conflict resolution 전략을 제안한다. 본 논문에서는 SDN 에 새로운 정책이 추가되는 경우 발생할 수 있는 conflict 를 감지한 후 해결하기 위해, 정책엔진이 conflict detector 와 conflict handler 로 구성되는 구조를 가정한다. Conflict detector 는 새로 추가되는 정책이 기존의 정책들과 충돌을 일으키는지 감지하고 conflict handler 는 conflict resolution 을 통해 문제가 되는 정책을 삭제하는 역할을 한다. 본 논문에서는 conflict handler 가 Recency 중심 전략과 Priority 중심 전략을 사용하여 문제가 되는 정책을 삭제하는 방안에 대해서 분석한 결과를 소개한다.

Ⅰ. 서 론

최근 몇 년간 Software-Defined Network (SDN)에 관한 연구가 활발히 진행되었다. 기존의 네트워크에서는 벤더마다 다른 운영체제, 프로토콜, 인터페이스를 사용함에 따라 네트워크 장치나 사용하는 소프트웨어도 다르게 해야 한다. 그로 인해 운영체제가 변경되거나 새로운 프로토콜이나 인터페이스가 적용되어야 할 경우, 전체 네트워크의 구성을 다시 고려하고 재구성해야 하는 불편함이 존재한다. 하지만 SDN 에서는 이러한 네트워크 구성을 소프트웨어를 이용하여 관리할 수 있기 때문에 기존의 네트워크에 비해 더 유연하고 편리하게 네트워크를 관리할 수 있다.

Figure1 과 같이 SDN 에서의 네트워크는 Management Plane (MP), Control Plane (CP), Data Plane (DP)로 구성된다. MP 는 네트워크에서 구동되는 어플리케이션으로 구성되며 routing application, Quality of Service (QOS) management application, 방화벽과 같은 security application 등이 사용된다 [1]. CP 에서는 SDN controller 가 어떤 트래픽이 어떻게 어떤 네트워크 장치로 전달되도록 할 것인지를 제어한다. DP 는 기존의 네트워크와 같이 네트워크 장치들로 구성된다. 이러한 SDN 구조에서, MP 의 application 에서 방화벽의 설정과 같은 새로운 정책 (rule)을 추가하길 원할 경우, CP 의 SDN controller 가 DP 의 특정 스위치로 해당 정책을 포워딩한다. 이후 DP 에서는 새로운 정책을 전달받은 스위치가 기존의 정책들로 구성된 정책 테이블 (rule table)에 새로운 정책을 추가한다. 하지만 이 때 정책 테이블에 있던 정책들과 새로운 정책이 충돌할 가능성이 존재하고, 이러한 충돌을 감지하는 방법을 conflict detection 이라 한다.

또한 충돌이 감지되었을 경우, 정책들 간의 충돌을 해결하는 방법을 conflict resolution 또는 conflict handling [2]이라 한다. 본 논문에서는 conflict resolution 에 대해 분석한 내용을 소개하고, SDN 에서 추가되는 정책이 기존의 정책들에 일으키는 충돌을 해결하기 위해 conflict detection 과 conflict resolution 을 위한 방법들을 소개하고 conflict handler 의 Recency 중심 전략과 Priority 중심 전략에 대해 분석한 내용을 설명한다.

본론에서는 정책엔진의 정책 간의 충돌 감지와 이를 해결하는 방법에 대해 소개하고, 이를 수행하는 conflict detector 와 conflict handler 를 포함한 정책엔진의 구조를 제안한다. 결론에서는 conflict resolution 전략들에 대해 분석한 내용과 본 논문에서 진행한 연구의 한계점에 대해 서술한다.

Ⅱ. 본 론

SDN 은 MP, CP, DP 로 구성되며 본 논문에서 소개하는 정책엔진은 CP 에 배치된다. MP 의 어플리케이션에서 새로운 정책을 추가하면 CP 의 SDN controller 가 전달받아 해당 정책이 기존 정책들과 충돌을 일으키는지 확인하기 위해 정책엔진에 전달한다. 정책엔진은 추가되는 정책이 충돌을 일으키는지 감지하는 conflict detector 와 충돌을 해결하는 conflict handler 로 구성된다. 본 논문에서 제안하는 정책엔진의 구조는 Figure 1 에 표현되어 있는 바와 같다.

추가되는 정책이 정책엔진에 전달되면 conflict detector 의 conflict classifier 가 충돌 여부를 확인하고, 충돌이 감지되었을 경우 'Redundancy', 'Shadowing', 'Generalization', 'Correlation', 'Overlap', 'Imbrication' 중 한 가지 class 로 분류하고, 충돌이 감지되지 않았을 경우 'No conflict'로 분류한다. 본 논문에서의 conflict

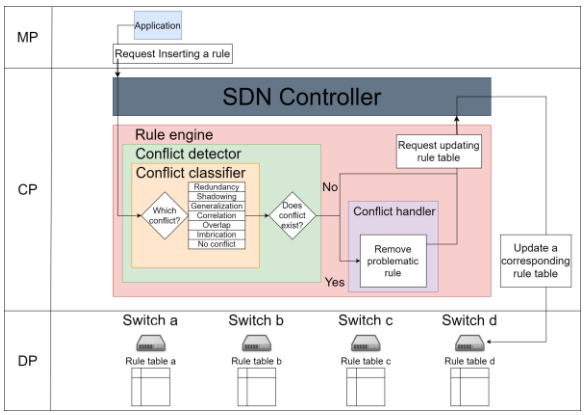

Figure 1 SDN 에서의 정책엔진의 구조와 새로운 정책이 추가되는 경우의 시스템 흐름

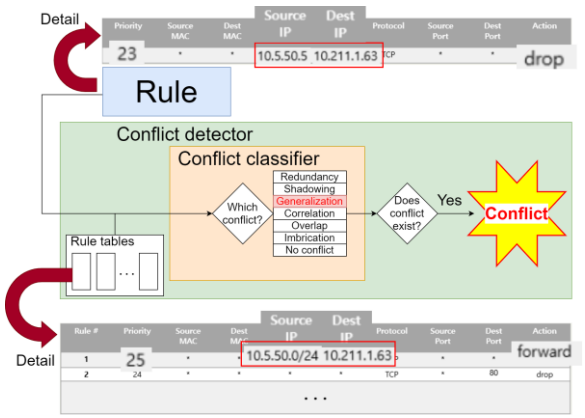

Figure 2 Conflict detection 의 예시

detector 는 Brew [3]의 classification 에서 사용되는 conflict class 들을 사용한다. Conflict detection 은 Figure 2 에 표현되어 있는 바와 같이 진행된다. Figure 2 의 예시는 기존의 정책 테이블에 있던 정책과 새로 추가되어야 할 정책이 충돌하는 것을 나타낸다. 이 때, 기존의 정책은 Source IP 10.5.50.0/24 에서 Destination IP 10.211.1.63 으로 보내는 모든 패킷을 forward 하도록 명시하지만, 새로 추가되는 정책은 앞의 정책과 같은 Source IP 와 Destination IP 를 명시하였음에도 패킷들을 drop 하는 action 을 정의하였다. 이 경우 6 개의 conflict class 들 중 generalization conflict 에 해당이 되어, 두 정책 중 하나를 conflict resolution 을 통해 삭제해야 한다.

서로 충돌하는 정책들 중 문제가 되는 정책을 결정하는 방법으로 Specificity 중심 전략, Recency 중심 전략, Priority 중심 전략, Arbitrary choice 중심 전략 등이 있으며, 본 논문에서는 정책엔진의 conflict handler 가 Recency 중심 전략과 Priority 중심 전략을 사용하는 것을 고려한다. Recency 중심 전략은 conflict 가 발생하는 정책들 중 가장 최근에 만들어진 정책을 적용하도록 하고, Priority 중심 전략은 각 정책에 우선순위를 부여하여 가장 높은 우선순위를 가진 정책을 적용하도록 한다. Recency 중심 전략을 사용하는 경우, 새로 추가되는 정책이 정책 테이블에 있던 기존의 정책보다 더 최신 정책이기 때문에 기존의 정책을 문제가 되는 정책으로 정하여 삭제하고 새로운 정책을 정책 테이블에 추가하도록 SDN controller 에 정책 테이블 업데이트를 요청한다. 반면, Priority 중심 전략을 사용하는 경우, Recency 중심 전략과는 달리 새로 추가되는 전략이 기존의 정책보다 더 최신임에도 불구하고 더 낮은 우선순위를 가지고 있으면 추가 요청된 정책을 문제가 되는 정책으로 정하여 drop 하고 SDN controller 에 정책 테이블 업데이트를 요청하지 않는다. 하지만 Conflict detector 가 충돌이 존재하지 않는다고 판단하였을 경우 conflict handler 를 거치지 않고 SDN controller 에게 새로운 정책을 추가하도록 요청한다.

Ⅲ. 고 찰

본 논문에서는 SDN 에서 발생하는 정책들 간의 충돌을 해결하기 위한 conflict detection 과 conflict resolution 에 대해 분석한 내용을 소개하고, 이를 지원하는 정책엔진의 구조를 제안하였다. Conflict handler 가 Recency 중심 전략을 사용할 경우 새로운 정책을 추가하고 오래된 정책을 삭제하므로 항상 최신 정책을 사용할 수 있다. 따라서 정책의 최신성을 중요시하는 네트워크로 구성된 SDN 에 사용하는 것이 적절하다. 하지만 각 정책의 중요도를 고려하지 않기 때문에 시스템이 의도하지 않은 대로 흘러갈 수 있다. 이러한 의도치 않은 시스템의 흐름은 SDN 의 패킷들이 당초 의도한 바와 다르게 전송되어 보안 상의 문제등을 일으킬 수 있다. 반면, Priority 중심 전략을 사용할 경우 중요도가 높은 정책을 유지할 수 있기 때문에 전술 네트워크와 같이 전장의 특수한 환경을 고려해서 만들어진 정책을 우선해야는 환경에서 효과적일 수 있다. 하지만 새로 추가하려는 정책의 우선 순위가 충돌이 발생하는 기존의 정책보다 낮을 경우 정책 테이블에 반영되지 않으므로 정책들의 최신성을 반영하기는 어렵다.

본 논문에서는 제안하는 정책엔진의 conflict resolution 을 위한 전략들에 대해 분석하였다. 그러나 사용하는 상황에 따라 사용할 전략을 선택하는 기준에 대해 향후 연구에서 정교하게 연구하고, 그 결과를 기반으로 conflict resolution 을 수행해야만 본 논문에서 설명한 전략과 구조가 주어진 목표를 달성할 수 있다.

ACKNOWLEDGMENT

본 연구는 방위사업청과 국방과학연구소가 지원하는 미래전투체계 네트워크기술 특화연구센터 사업의 일환으로 수행되었습니다.(UD190033ED)

참 고 문 헌

[1] Tran CN, Danciu V. On Conflict Handling in Software-Defined Networks. In: *International Conference on Advanced Computing and Applications (ACOMP)*; IEEE; 2018; 50-57.

[2] Khurshid A, Zou X, Zhou W, Caesar M, Godfrey PB. Veriflow: Verifying Network-Wide Invariants in Real Time. In: *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*; 2013; 15-27.

[3] Pisharody S, Natarajan J, Chowdhary A, Alshlan A, Huang D. Brew: A Security Policy Analysis Framework for Distributed SDN-based Cloud Environments. *IEEE Transactions on Dependable and Secure Computing* 2017; 16(6); 1011-1025.