논문 21-46-10-11 The Journal of Korean Institute of Communications and Information Sciences '21-10 Vol.46 No.10
https://doi.org/10.7840/kics.2021.46.10.1632

트래픽 서비스 품질을 보장하기 위한 보안 자원 처리율 기반 매핑 알고리즘

서 기 원◆, 배 재 원*, 류 선 열**, 김 종 원***, 임 혁°

Security Resource Throughput-Based Mapping Algorithm for Guaranteeing Quality-of-Service of Traffic

Giwon Sur◆, Jaewon Bae*, Shun Yuel Ryu**, JongWon Kim***, Hyuk Lim°

요 약

데이터 네트워크는 악의적 사용자로부터 데이터의 탈취, 침투로 인한 시스템 마비 등의 위협을 받을 수 있다. 이러한 위협으로부터 네트워크를 보호하기 위해 방화벽, IDS 등의 보안 네트워크 기능이 활용되고 있다. 그러나 실시간 데이터 트래픽이 특정 네트워크 기능 자원에 집중되면, 해당 네트워크 기능 자원에서 시간지연이 과도하게 커져 트래픽의 서비스 품질 (quality-of-service, QoS) 요구조건이 만족되지 않을 수도 있다. 따라서, 네트워크 보안 자원이 트래픽에 대한 보안 서비스를 제공할 때, 트래픽의 다양한 QoS를 만족하도록 트래픽을 처리하는 것이 필요하다. 본 논문에서는 독립된 전용 보안 시스템과 가상화된 네트워크 기능(virtualized network function, VNF)을 생성하여 보안 자원(security resource, SR)을 구성하고, 이러한 보안 자원들이 각기 다른 전송속도와 시간지연 요구조건을 가진 실시간 데이터 트래픽에 보안 기능 서비스를 제공하는 경우를 고려한다. 본 논문은 SR에서 발생하는 시간지연을 M/D/1 큐 (queue)로 모델링하고, 각 트래픽을 SR에 배치하는 문제를 최적화 문제로 공식화한다. 이 최적화 문제를 multiple knapsack problem (MKP)으로 표현하고, 이의 풀이를 위한 탐욕 기반의 알고리즘을 제안한다. 트래픽의 QoS 요구조건과 SR의 개수가 변하는 다양한 실시간 데이터 네트워크 환경에 대한 시뮬레이션에서 제안된 알고리즘은 낮은 계산 복잡도로 높은 SR 처리용량이 가능한 배치를 찾을 수 있는 결과를 보여주었다.

Key Words : Quality of Service, Security Resource, Virtualized Network Function, M/D/1 Queueing Model, Multiple Knapsack Problem

ABSTRACT

Data networks could be threatened with data theft or service breakdown caused by intrusion from malicious users. To protect the networks from these threats, security network functions such as firewalls and intrusion detection systems are utilized. When the security network functions provide security services, it is needed to satisfy the quality-of-service (QoS) requirements of the traffic. If excessive real-time data traffic is configured

※ 본 연구는 국방과학연구소의 연구비 지원으로 수행되었습니다 (UD190020FD).
◆ First Author : Gwangju Institute of Science and Technology (GIST), School of Electrical Engineering and Computer Science, giwonsur@gm.gist.ac.kr, 학생회원
° Corresponding Author : Gwangju Institute of Science and Technology (GIST), AI Graduate School, hlim@gist.ac.kr, 정회원
* Gwangju Institute of Science and Technology (GIST), School of Electrical Engineering and Computer Science, jaewonbae@gm.gist.ac.kr, 학생회원
** Agency for Defense Development (ADD), Defense AI Technology Center, rsy10@add.re.kr
*** Gwangju Institute of Science and Technology (GIST), AI Graduate School, jongwon@gist.ac.kr, 종신회원
논문번호 : 202106-127-B-RU, Received June 7, 2021; Revised July 14, 2021; Accepted July 20, 2021

to be served by a specific network function resource, the processing time may grow exponentially and the system cannot satisfy the QoS requirements of the traffic. This paper considers a security resource pool that is composed of dedicated standalone security systems and virtualized network functions (VNFs) for providing security function services to traffic with different QoS requirements. We model the time delay in an SR as an M/D/1 queue model and formulate the allocation problem as an optimization problem. The optimization problem is represented as a multiple knapsack problem and is solved using a greed-based algorithm. The simulation results indicate that the proposed method achieves a high SR process capacity with low computational complexity in a real-time network environment.

Ⅰ. 서 론

코로나바이러스 시대에 접어들며 대면 접촉을 최소화하기 위해 일상생활, 경제활동 등이 온라인으로 이루어지고 있다. 멀리 떨어진 거리에 있는 사람들에게 화상통화를 통한 온라인 회의와 같은 실시간 소통을 지원하거나, 물리적으로 접근하기 힘든 장소의 영상정보를 실시간으로 제공하는 응용서비스들이 늘어나면서 실시간 데이터의 서비스 품질 (quality-of-service, QoS) 보장 요구가 많이 증가했다[1]. 영상, 사진, 음성 등의 실시간 데이터가 사용자에게 전달될 때, 데이터 사용의 목적에 따라 네트워크는 데이터 전송용량, 전송 시간 지연, 그리고 데이터 손실 등과 관련된 QoS를 보장하여야 하며, 사용자는 네트워크 서비스 제공자에게 일정 수준의 QoS 보장을 요구할 수 있다. 보통 네트워크 시스템에서 지나치게 많은 데이터가 일시에 전송되게 되면 네트워크 혼잡(congestion)이 발생하고, 이러한 혼잡으로 인해 네트워크의 트래픽 처리율이 저하되고 시간 지연이 늘어날 수 있다. 이러한 시간 지연은 특히 실시간 데이터 트래픽의 QoS를 손상할 수 있으므로, 네트워크 관리자는 실시간 데이터의 시간 지연을 최소화하기 위한 능동적인 망 관리 운영을 수행해야 한다.

대부분의 응용 애플리케이션 (application) 및 서비스에서 데이터 트래픽은 사용자에게 전달되는 과정 중 다양한 네트워크 기능 서비스를 받게 된다. 예를 들어, 데이터가 목적지에 도착했을 때 내부 네트워크의 정책에 따라 안전한 트래픽인지 검증해 접근을 막거나 허용하는 방화벽, 여러 컴퓨터 자원에 업무를 분산하여 데이터 서비스를 제공하는 로드 밸런서, 패킷을 자세히 감시해 악의적이거나 정책 위반을 하는 행동을 찾아내는 침입 탐지 시스템 (intrusion detection system, IDS), 악의적 패킷의 침입을 막는 침입 예방 시스템 (intrusion prevention system, IPS), 제한된 수의 공용 주소에 대해 사설망에서 목적지 주소를 찾아 트래픽을 전달하는 기능을 하는 network address translator (NAT) 가 있다. 이러한 다양한 네트워크 기능은 전용 하드웨어를 사용하거나 혹은 범용 서버 위의 가상화된 자원에서 구현되어 제공될 수 있다. 단일 기능만을 제공하거나 통합관리가 어려운 전용 하드웨어의 단점을 보완하기 위해 개발된 네트워크 기능 가상화 (network function virtualization, NFV)는 새로운 전용 장비의 설치 없이 소프트웨어적으로 가상화된 네트워크 기능 (virtualized network function, VNF)을 구현해 서비스를 제공할 수 있는 기술이다. 이 기술은 설비투자 비용, 운영 비용 절감을 가능하게 하며 유연한 네트워크 시스템을 만들 수 있게 한다[2].

네트워크 시스템에는 정상적인 트래픽 외에도, 취약점을 통해 데이터를 탈취하거나, 시스템의 마비를 일으키는 악성 트래픽이 인입될 수 있다[3]. 이를 예방하기 위해 네트워크 시스템은 방화벽, IDS, IPS 등으로 이루어진 보안 기능 자원을 필요로 한다. 실시간 데이터 트래픽이 특정 보안 기능 자원에 집중되면 해당 자원에서 과도한 시간지연이 발생하여 실시간 데이터 트래픽이 만족해야 하는 시간지연 QoS 요구조건을 만족시킬 수 없다. 따라서 네트워크 관리자는 실시간 데이터 트래픽의 시간지연과 QoS 요구조건을 모두 고려하여 서비스를 제공해야 한다. 보안 기능을 독립된 전용 보안 시스템과 VNF가 결합된 하이브리드 시스템으로 구성하면 기존의 자원을 활용하는 동시에 동적인 보안 기능의 전개가 가능하다. 또한, 가변적인 트래픽량에 대응하는 서비스를 제공하도록 조율이 가능하여, 트래픽의 적절한 보안 자원 배치를 통해 QoS 보장을 기대할 수 있다.

본 논문에서는 사이버 보안 기능을 제공하는 이종의 가용 보안 자원이 있는 네트워크에서 실시간 데이터 트래픽에 대한 보안 기능을 제공할 때, 실시간 데이터 트래픽의 전송지연 QoS 요구조건을 만족하도록 실시간 데이터 트래픽과 이를 처리할 보안 자원 간의 매핑을 결정하는 문제를 다룬다. 본 논문은 실시간 트

래픽과 보안 자원 간의 매핑을 실시간 트래픽의 QoS 요구조건을 만족하면서 트래픽 플로우의 전송속도 합을 최대화하는 최적화 문제로 제안한다. 이 매핑 문제는 다중 배낭 문제 (multiple knapsack problem, MKP)로 표현될 수 있으며, 본 논문은 이 문제의 해를 구하기 위해 그리디 휴리스틱한 알고리즘을 제안한다. 제안하는 알고리즘을 적용하면 주어진 보안 기능 자원에서 높은 처리율을 달성할 수 있으며, 최소의 자원으로 주어진 보안 기능 서비스를 제공할 수 있다.

이후 본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 살펴보고, 3장에서 사이버 보안 기능을 제공하는 네트워크 시스템 내 실시간 데이터의 시간지연을 모델링하고, 실시간 트래픽의 시간지연이 QoS 요구조건을 만족하면서 보안 자원의 처리율을 최대로 할 수 있는 보안 자원의 매핑 문제를 제시한다. 4장에서는 매핑 문제의 해를 구하는 그리디 휴리스틱 알고리즘을 상세히 기술한다. 5장에서는 다른 알고리즘과의 비교실험을 통해 제안하는 알고리즘의 성능을 검증하고, 6장에서 결론을 맺는다.

Ⅱ. 관련 연구

2.1 트래픽 시간지연에 관한 연구

트래픽의 시간지연은 전송망의 대역폭, 혼잡한 정도, 전송 거리에 따른 전파지연, 네트워크 장비의 처리속도 등에 영향을 받는다. 실시간 트래픽과 같이 트래픽의 시간지연 요구조건이 있는 경우는 시간지연 요구조건에 따라 트래픽의 우선순위를 정해 네트워크 서비스를 제공하는 등의 효율적인 시스템 관리로 트래픽을 처리하여야 한다.

네트워크 시스템에서 데이터 트래픽의 시간지연 모델링, 분석 및 이를 보장하기 위한 연구는 활발히 이루어져 왔다. Ye *et al.* 은 VNF 환경에서 플로우의 단대단 지연 모델링 수식을 도출하였다. 플로우가 NFV 노드에서 서비스를 받을 때 dominant resource generalized processor sharing (DR-GPS) 개념을 사용하여 플로우의 우선순위 및 필요한 자원의 비율에 따라 CPU, link bandwidth 자원을 할당하였다. 플로우 별로 고유한 처리율, 전송률을 가지게 한 후, 이 값을 M/D/1 queueing에 적용하여 NFV 노드 내에서 소요되는 지연 식을 도출했다. 시뮬레이션을 통해 플로우의 소요된 시간과 도출된 계산식의 값을 비교했고, 제안한 지연 식의 값이 시뮬레이션 결과 지연에 가까움을 증명하였다[4]. Xiu *et al.* 은 사용자의 QoS 조건을 기준으로 웹서비스를 선택할 수 있도록 패턴을 인지하여 QoS를 예측하는 latent factor analysis 기반 모델을 제안했다. 해당 모델은 사용자와 서비스에 따른 QoS 지표를 담는 tensor 매트릭스를 time point 별로 가지고 있어, 학습 시 사용자, 서비스, time point의 linear-bias 값을 사용하고, QoS 값이 non-negative임을 반영한다. 학습된 모델은 기존의 QoS 정보로 목표하는 대상의 사용자와 서비스에 대한 QoS를 동적으로 예측할 수 있다. 제안된 모델은 응답시간과 처리율(throughput)을 예측하는 실험 결과에서 다른 모델에 비해 예측 성능이 뛰어남을 보였다[5]. Karakus *et al.* 은 소프트웨어 정의 네트워킹 (software defined networking, SDN) 환경에서 트래픽을 분류하고 라우팅할 수 있음을 이용하여 트래픽의 QoS를 관리하는 방법을 소개했다. 시간지연에 민감한 트래픽에 대해 지연 제한의 목적을 가진 라우팅을 실현할 수 있으며, 트래픽 분류를 가능하게 해 플로우 기반 모니터링을 제공할 수 있음을 설명했다. 이 외에도 분산된 SDN의 제어기들이 서로 네트워크 정보를 공유하는 방식으로 도메인이 다른 네트워크에서 QoS를 관리하는 기법이 존재함을 보였다. SDN 환경의 트래픽 모니터링 시 증가하는 오버헤드를 주의해야 함을 강조하였고, 동적인 QoS 변화에 대응할 자동화된 QoS 관리메커니즘의 필요성을 주장하였다[6]. Park *et al.* 은 데이터의 전달망이 주요망과 예비망으로 나뉘는 이종 통신망 내에서 특정 망의 대역폭 낭비 및 데이터 트래픽의 서비스 품질 저하를 방지하기 위한 QoS 정책을 제안하였다. 제안된 정책은 트래픽을 중요 데이터 및 일반 데이터로 나누어 성능이 우수한 주요 망에 중요 데이터를 먼저 할당하도록 한다. 실험 결과 제안된 QoS 정책이 다른 환경의 서비스보다 낮은 지연과 개선된 품질의 서비스를 제공하여 주어진 통신망을 효율적으로 사용함을 확인하였다[7]. Joung과 Kwon은 Regulating Scheduler (RSC)를 deficit round robin에 적용하여 특정 큐가 링크를 독점하지 못하도록 해 플로우의 최대 지연시간을 보장하는 RSC를 제안하였다. 제안된 RSC는 가상 패킷이 담긴 큐가 처리될 때, 실제 패킷이 들어오면 해당 패킷의 큐를 먼저 처리하게끔 큐의 순서를 조율해 동작하며 패킷이 스케줄러에 인입 시 발생하는 대기시간을 줄이도록 한다. 실험을 통해 인입되는 플로우 전송속도 증가 시, 큐의 길이가 TDMA의 결과보다 완만하게 증가하여, 제안하는 스케줄러의 성능이 뛰어남을 보였다[8].

2.2 VNF 배치 및 할당에 관한 연구

최근 네트워크 시스템은 NFV 인프라를 활용하여

플로우에게 필요한 여러 네트워크 기능을 서비스 체이닝 형태의 VNF로 제공하고 있다. 플로우가 요구하는 VNF를 최적의 비용으로 제공하기 위한 VNF 배치 및 할당에 관한 연구가 활발히 이뤄져 왔다.

Ahvar *et al.* 은 출발지와 도착지점이 각기 다른 트래픽에게 네트워크 기능을 서비스해야 할 때 비용을 최소화하는 VNF 배치 알고리즘을 제안하였다. 알고리즘은 betweenness centrality 기반으로 동작하며, 트래픽 플로우의 최단 경로를 확인 후 높은 centrality, 즉 플로우의 최단 경로에 빈번히 포함되어 높은 비중을 지닌 노드에 VNF를 배치하고자 했다. VNF의 위치가 플로우의 최단 경로를 벗어나지 않을 경우 비용에 영향을 끼치는 추가적인 네트워크 자원이 필요하지 않게 된다. 알고리즘은 모든 플로우가 할당될 때까지 VNF 생성을 시도하며, 수행결과의 비용이 최적의 비용으로 여겨지는 integer linear programming (ILP)의 결과와 비슷함을 확인하였다[9]. Cappanera *et al.* 은 네트워크 서비스 제공자의 이익과 사용자의 이익의 합이 최대가 되도록 데이터센터에 VNF를 배치하는 알고리즘을 제안하였다. 서비스 제공자는 서비스를 받은 트래픽의 합을 최대화하려 하고, 이때 트래픽의 우선순위가 높을수록 합이 큰 것으로 간주하였다. 서비스 사용자는 환경에 영향을 최소로 끼치거나, 혹은 비용을 최소로 하는 등의 선호 정책을 시행하여, 전체 서비스받은 사용자의 만족된 선호가 최대가 되도록 데이터센터에 VNF를 배치하고자 했다. 이 과정에서 서비스 수용에 제한을 주는 트래픽의 QoS 요구조건은 물리적 거리가 있는 데이터센터의 전파지연이 고려되었다[10]. Jia *et al.* 은 VNF 서비스 체이닝이 데이터센터별로 분산되는 환경에서 VNF 배치 및 플로우 라우팅 알고리즘을 제안하였다. 제안된 알고리즘은 VNF 실행 및 전개 비용뿐 아니라, 플로우 전송 및 단대단 지연의 비용을 포함한 최종 비용을 최소화하는 방향으로 데이터센터별로 VNF를 몇 개 생성할 것인지, 플로우를 어느 VNF에 라우팅할 것인지 정하는 문제를 convex 최적화 문제로 정리하고 interior point method로 해결한다. 해당 알고리즘은 온라인으로 동작하여 동적으로 VNF를 전개 및 제거함으로써 급격하게 증감할 수 있는 트래픽에 유연하게 대처하는 장점이 있다[11]. Sur *et al.* 은 VNF의 시스템 지연을 고려하여 시간지연 요구조건이 만족된 실시간 트래픽량을 최대화하는 알고리즘을 제안하였다. VNF-플로우 매핑 문제를 다중 배낭 문제로 해결할 것을 제안하였다. 그러나, 트래픽을 처리하는 VNF의 처리율이 일정하고, 최적화 문제에서 VNF의 처리 지연만이 고려되었다[12].

2.3 보안 자원 매핑에 관한 연구

보안 자원이 트래픽을 처리할 때, 자원의 효율적인 사용 혹은 네트워크의 load balancing을 위해 보안 자원을 분산하여 위치시키거나 트래픽을 라우팅하여 보안 자원과 트래픽을 매핑하는 연구는 활발히 진행되어 왔다.

Jakaria *et al.* 은 트래픽이 가상화된 분산 방화벽에 전달될 때 load balancing을 수행하는 알고리즘을 제안하였다. 알고리즘은 해시테이블을 통해 트래픽을 방화벽에게 일차적으로 전달하고, 정해진 threshold를 기준으로 트래픽량이 많은 방화벽의 트래픽을 트래픽량이 적은 방화벽으로 옮기고, 트래픽을 옮길 방화벽이 없을 경우 새로 생성한다. 트래픽량이 적은 방화벽의 수가 많을 경우, 최소의 이용률을 가진 방화벽의 트래픽을 다른 방화벽으로 반복적으로 옮기고 트래픽을 서비스하지 않는 방화벽이 생길 경우 제거하여 전체 방화벽 수를 효율적으로 조절한다[13]. Jain *et al.* 은 IoT 환경에서 IDS가 IoT 기기의 데이터를 수집하는 과정에서 최적화를 위해 genetic 알고리즘을 사용할 수 있음을 제안하였다. 제안된 알고리즘은 데이터의 초기 population 생성 후 fitness value의 값을 계산하여 이후 genetic operator인 selection, crossover, mutation을 차례대로 수행하는 과정을 보인다[14]. Bagheri *et al.* 은 클라우드의 단일 방화벽의 규칙 lookup 시간을 줄이고 디도스 공격에 대해 감내할 수 있는 시스템을 위해 방화벽을 분산하는 알고리즘을 제안하였다. 알고리즘은 높은 비율의 트래픽에 매칭되는 규칙을 작은 방화벽으로 분산시켜 작은 방화벽을 먼저 거친 트래픽이 그 후 주 방화벽에 들어올 수 있도록 하며, 분산된 방화벽의 트래픽의 매칭률이 낮아지면 해당 방화벽을 다시 주 방화벽에 병합시킨다[15]. Yoon *et al.* 은 네트워크 모니터링 과정 중에서 발생할 수 있는 네트워크 과부하의 발생 방지를 위해, 효율적인 샘플링 위치 및 비율 선정 알고리즘을 제안하였다. 알고리즘은 betweenness centrality에 기반하여, 플로우의 최단 경로가 많이 겹치는 스위치 순으로 샘플링 위치를 정한다. SDN 실험에서 제안하는 알고리즘의 IDS 탐지율과 capture failure rate가 좋은 성능을 보임을 확인하였다[16]. L. Durante *et al.* 은 폭포 형태의 다중 방화벽에서 방화벽 규칙을 목적지 IP주소에 따라 분산하는 알고리즘을 제안하였다. 알고리즘은 방화벽 규칙 중 목적지 IP가 다른 방화벽이 존재하는 네트워크의 IP를 가질 경우, 해당 규칙을 목적지

주소의 네트워크에 위치하는 방화벽으로 옮기고, 이 과정에서 규칙에 대한 허용, 거부, 다른 규칙으로의 이동을 고려한다. 실험 결과를 통해 알고리즘이 적용되어 규칙이 변경된 방화벽의 평균 패킷 필터링 시간이 감소함을 확인하였다[17].

기존 발표된 연구 대비 본 연구는 실시간 트래픽의 단대단 시간 지연을 고려하여, 네트워크 시스템 내 트래픽 플로우의 시간지연을 모델링 후 각 플로우가 가진 다양한 전송속도와 시간지연 요구조건을 고려해 시간지연을 예측하여 QoS 만족여부를 판단 후 보안 자원에 서비스받도록 매핑한다. 기존에 시간지연을 모델링 하거나 예측하여 이에 대한 정보를 공유하는[4-6] 연구가 존재하지만 트래픽을 자원에 할당하는 연구는 대부분 시간지연 요구조건에 대한 고려가 없거나 발생 가능한 시간지연의 일부만을 QoS로 고려하여 자원의 활용도만을 높이기 위한 연구를 진행하였다[7-11]. 또한, 대부분의 기존 보안자원 매핑 알고리즘은 계산 복잡도가 높거나 최소비용 혹은 최대목적에 적합한 자원을 탐색하는 과정이 iteration의 제한이 없이 수행되어 이 과정에서 수행시간이 크게 증가할 위험이 있다[13-17]. 제안하는 알고리즘은 탐욕 기반으로 수행되어 다중 배낭 문제의 해를 pseudo-polynomial 한 계산 복잡도로 얻을 수 있다.

Ⅲ. 보안 자원이 활용된 네트워크 시스템

네트워크 시스템 내외부의 악의적 사용자는 네트워크에 연결된 정보시스템 및 네트워크의 취약점을 활용하여 주요한 정보의 탈취 및 시스템/서비스 마비 등의 공격을 시도할 수 있다. Fig. 1은 다양한 보안 위협으로부터 네트워크에 연결된 정보시스템을 보호하기 위해 보안 자원 (security resource, SR)이 도입된 네트워크 시스템의 토폴로지 (topology)를 보여준다. 네트워크에 설치된 보안 자원은 패킷 또는 플로우의 허용 여부를 제어하는 방화벽, 특정 기준에 따라 패킷을 분석하여 침입을 탐지하는 IDS, 침입을 막는 IPS 등을 포함한다. 보안 기능을 수행하는 보안 자원은 물리적 전용 시스템이나 가상화 (virtualization) 기술을 활용한 VNF로 구현될 수 있다. 보안 시스템이 처리해야 하는 트래픽량은 변화폭이 크기 때문에 최대 트래픽 처리용량을 고려해서 확장성이 부족한 물리적인 전용 시스템으로만 구축하기보다 가상화 기반의 VNF를 사용하여 하이브리드 방식으로 구현하면 유연한 시스템 구성이 가능하다.

본 논문은 네트워크상에서 운용 가능한 보안 자원

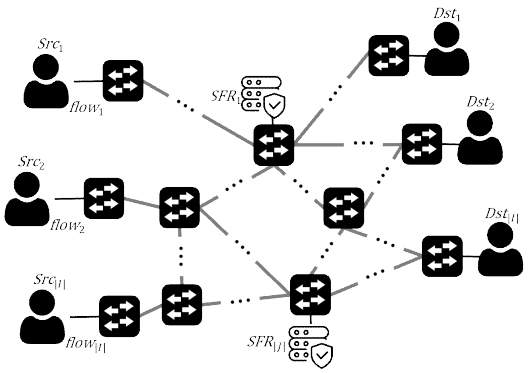


그림 1. 보안 자원이 존재하는 네트워크 토폴로지
Fig. 1. Network topology with security resources

의 최대 처리용량이 다르고, 보안 기능별 자원 요구량도 다른 경우에 보안 자원의 운용 효율성을 높이기 위한 알고리즘을 제안한다. 앞으로 물리적인 보안 전용 시스템과 가상화 보안 시스템으로 이뤄진 보안 자원의 셋을 보안 자원 Pool (security resource pool, SRP)로 명칭 한다. 여러 개의 데이터 플로우가 SRP의 특정 SR에 집중된 경우, 해당 SR이 처리할 수 있는 최대 용량 대비 자원 요구량이 증가하게 되어 처리를 기다리는 플로우는 서비스를 받기 위해 큐에 쌓이게 되고 시간지연이 늘어가게 된다. 시간지연 요구조건을 갖는 플로우에 과도한 시간 지연이 발생하지 않도록 SRP 내의 SR에 트래픽 플로우를 적절히 분배해 주어야 한다.

Fig. 1의 네트워크에서 $|I|$개의 트래픽 플로우, $I=\{1,\cdots,|I|\}$가 존재하고, 각 트래픽 플로우는 SR에서 서비스를 받고 목적지에 도달한다고 가정한다. 각 플로우는 전송속도 $rate_i$를 가지고 있으며, 시간지연 요구조건 $delay_i^{req}$ 시간 내에 목적지까지 도달해야 한다고 가정한다. SRP는 총 $|J|$개의 SR, $J=\{1,\cdots,|J|\}$로 구성되어 있으며, 트래픽이 지나갈 수 있는 스위치와 직접 연결되어 스위치를 지나는 트래픽 플로우를 서비스해줄 수 있다. 본 논문에서는 SR에 도착하는 트래픽은 memoryless process를 따르고 SR의 처리율이 최대 처리용량으로 고정되어있다고 가정하고, 각 SR에서 발생하는 시간지연을 M/D/1 queueing 모델로 구하였다[18]. 단, 일반적인 queueing 모델의 적용도 가능하다. 각 SR에서 소요되는 평균 시간지연 D_{SR}은 다음 식으로 주어진다.

$$D_{SR_j} = \frac{\frac{\lambda_j}{\mu_j}}{2\mu_j(1-\frac{\lambda_j}{\mu_j})} + \frac{1}{\mu_j},\ \lambda_j < \mu_j,\ \forall j \in J. \quad (1)$$

위 식에서 λ_j와 μ_j는 각각 SR_j에 유입되는 플로우들의 전송속도 합과 SR_j의 처리율을 나타내며, λ_j는 μ_j보다 작은 값을 가진다. 식 (1)에 따라 M/D/1 queueing 모델은 SR_j 내에 유입되는 플로우 전송속도가 증가할수록 지연이 급격하게 증가하는 특성을 갖게 된다. 플로우 i가 SR_j에 매핑되었을 때 해당 플로우에 대한 네트워크 시스템 내의 총 시간지연은 다음의 식 (2)와 같이 주어진다.

$$D(i|j) = d_{src(i)-SR_j} + D_{SR_j} + d_{SR_j-dst(i)}. \quad (2)$$

즉, 플로우의 총 시간지연은 출발 지점 $src(i)$에서 SR까지 소요되는 전파지연 $d_{src(i)-SR_j}$, SR 내에서의 지연 D_{SR_j}, SR에서 플로우의 목적지 $dst(i)$까지 전파지연 $d_{SR_j-dst(i)}$의 합으로 구성된다. 주요 용어는 Table 1에 정리되어 있다.

표 1. 시스템 매개변수
Table 1. System Parameters

I	Flow 의 집합
$rate_i$	Flow i 의 전송속도
$delay_i^{req}$	Flow i 의 시간지연 요구조건
Y_i	Flow i 가 매핑된 SR의 인덱스, 매핑되지 않을 경우 0
J	SR 의 집합
D_{SR_j}	SR j 내에서의 시스템 지연
$D(i\|j)$	Flow i 가 SR j 에 매핑되었을 때 플로우 i 의 전체 시스템에서의 총 시간지연

Ⅳ. SR-Flow 매핑을 위한 그리디 알고리즘

4.1 최적화 문제

본 논문은 설명된 네트워크 환경에서 플로우의 시간지연 요구조건을 고려하여 최대한 많은 플로우의 전송속도가 서비스받을 수 있도록 하는 목적을 지닌다. 다양한 경로의 플로우가 SRP를 통과하는 과정은 플로우가 서비스받을 SR을 선택하는 과정 혹은 SR이 서비스할 플로우를 선택하는 과정으로 볼 수 있다. 이 과정에서 주의할 점은 SR의 서비스를 받게 되는 플로우들의 수를 많게 하기보다 서비스를 받게 되는 플로우들의 총 전송속도의 합이 높도록 목표하는 것이다. 이 SR-Flow 매핑 문제를 아래와 같이 최대화 문제로 수식화한다.

$$\begin{aligned} &\max_{\boldsymbol{x}} \sum_{i=1}^{|I|}\sum_{j=1}^{|J|} rate_i \cdot x_{ij} \\ &s.t. \begin{cases} \sum_j x_{ij} \cdot D(i|j) \le delay_i^{req}, \\ \sum_j x_{ij} \le 1, \\ x_{ij} \in \{0,1\}, \end{cases} \quad \text{for } \begin{matrix} \forall i \in I, \\ \forall j \in J. \end{matrix} \end{aligned} \quad (3)$$

식 (3)에서 x_{ij}은 플로우 i가 SR_j에 매핑된 경우 1의 값을, 그렇지 않은 경우 0의 값을 갖는다. 따라서, 식 (3)은 플로우 집합에 속한 i와 SR 집합에 속한 j에 대해 SR_j에 매핑된 플로우 i의 전송속도의 합의 최대화를 나타낸다. 이때, 첫 번째 제약조건은 플로우 각각의 시간지연이 자신의 QoS 요구조건을 만족시키는 SR에 매핑이 가능할 수 있어야 함을 나타낸다. 두 번째 제약조건은 각 플로우가 최대 1개의 SR에만 할당되는 조건을 나타낸다. 만약, 플로우 i에 대해서 x_{ij}의 합이 0인 경우는 해당 플로우가 어떤 SR에도 할당되지 않았다는 것을 의미한다. 반면, 구해진 해에 대해 $\sum_{i \in I} x_{ij} = 0$ 인 경우는 SR_j에 할당된 플로우가 하나도 없는 것을 의미하며, 이런 경우 해당 SR_j는 서비스를 위해 준비될 필요가 없다.

4.2 다중 배낭 문제

본 논문은 SR-Flow 최적화 문제의 해를 구하기 위하여 탐욕 (greedy) 알고리즘을 제안한다. 제안된 탐욕 보안 자원 매핑 (greedy security resource mapping, GSRM)은 MKP 접근 방법에 기반하고 있다. MKP는 여러 개의 가방에 이익과 무게가 다른 물건을 담아 전체 이익을 최대화하는 방법을 찾는 알고리즘이다[19]. 본 SR-Flow 매핑 문제는 여러 가용 SR에 전송속도의 크기와 시간지연 요구조건이 다른 플로우들을 매핑해 전송속도의 합을 최대화하는 방법을 찾는 것이다. 기존 MKP 문제에서는 배낭 안에 있는 물건들의 무게의 합이 배낭의 용량을 넘지 않아야 하는데, 본 논문에서는 각 플로우가 어떤 SR에 매핑되는지에 따라 각 플로우의 시간지연이 다르게 계산되는 것을 고려하여 플로우의 시간지연 요구조건을 만족시켜야 한다. 즉, 제안하는 알고리즘은 식 (2)에서와

같이 출처에서 SR을 찾아가는 지연, SR 내에서의 존재하는 플로우의 전송속도 합에 따라 달라지는 시스템 지연, SR에서 목적지로 향하는 지연을 고려하여 각 플로우의 시간지연 요구조건을 충족시켜야 한다.

4.3 탐욕 알고리즘

1) Initial solution: 첫 단계에서는 플로우를 $rate_i \times delay_i^{req}$ 값이 큰 순서대로 정렬하고, 탐욕 방식으로 차례로 SR을 매핑하여 초기 solution을 찾는다. 식 (1)을 참고하면 SR 내 시간지연은 처리용량이 클수록 작은 값을 지니는 것을 알 수 있다. 따라서, 요구하는 시간지연이 짧을수록 처리용량에 해당하는 처리율을 크게 요구하는 것이고 무게가 무거운 것으로 볼 수 있다. 요구하는 시간지연이 짧지 않아, 큰 처리율을 필요로 하지 않는 플로우는 무게가 적은 것으로 볼 수 있다. 일반적인 Knapsack 문제에서 이익이 크고 무게가 적은 물건을 탐욕 방식에 따라 차례로 가방에 담는 것과 같이, 플로우의 전송속도 값이 크고 시간지연 요구조건 값이 큰, 즉 시간지연 요구조건이 짧지 않은 플로우 순서대로 SR에 매핑될 수 있도록 한다. SR 인덱스의 순서대로 플로우를 1번째부터 $|I|$번째까지 순차적으로 탐색하며 매핑되지 않은 플로우에 한해 시간지연 요구조건이 만족될 경우 매핑한다. 이미 다른 SR에 매핑된 플로우는 매핑대상에서 제외된다.

2) Rearrangement: 두 번째 단계는 정렬된 플로우의 역순으로 매핑된 플로우에 한해 먼저 매핑을 수행하는 단계로, SR을 차례로 탐색하며 매핑이 가능한 SR을 찾는다. 즉, 첫 단계에서 매핑되었던 플로우를 추출하여 새롭게 재매핑하는 것이다. 추출된 플로우는 첫 번째 SR부터 순회하며 요구조건을 만족시키는 SR을 찾아 매핑하며, 뒤이은 SR이 다음 플로우의 탐색 시작지점이 된다. Fig. 2에서 1, 3, $|I|$번째의 플로우가 첫 단계에서 매핑되었다 하자. 이때 $flow_{|I|}$부터 탐색을 시도하며 $flow_{|I|}$이 총 3개의 SR 중 2번째에 매핑되었다면, $flow_3$은 SR_3부터 탐색하며, 매핑이 되지 않으면 그 다음 SR_1, SR_2 순으로 탐색하는 것이다. 전 플로우가 매핑되지 못한 채로 한 바퀴를 마무리하면, 전 플로우가 시작한 순서와 동일하게 뒤이은 플로우도 매핑가능한 SR을 찾는다. Fig. 2의 ①에서 $flow_{|I|}$부터 시작하여 마지막 $flow_1$까지 다 탐색하였다면, ②에서 SR별로 $flow_1$부터 $flow_{|I|}$까지 순차적으로 탐색해 매핑되지 않은 플로우가 있을 경우 매핑 시도를 한다.

3) First improvement: 세 번째 단계는 매핑되지 못한 플로우의 매핑을 위해 매핑된 플로우들을 교환

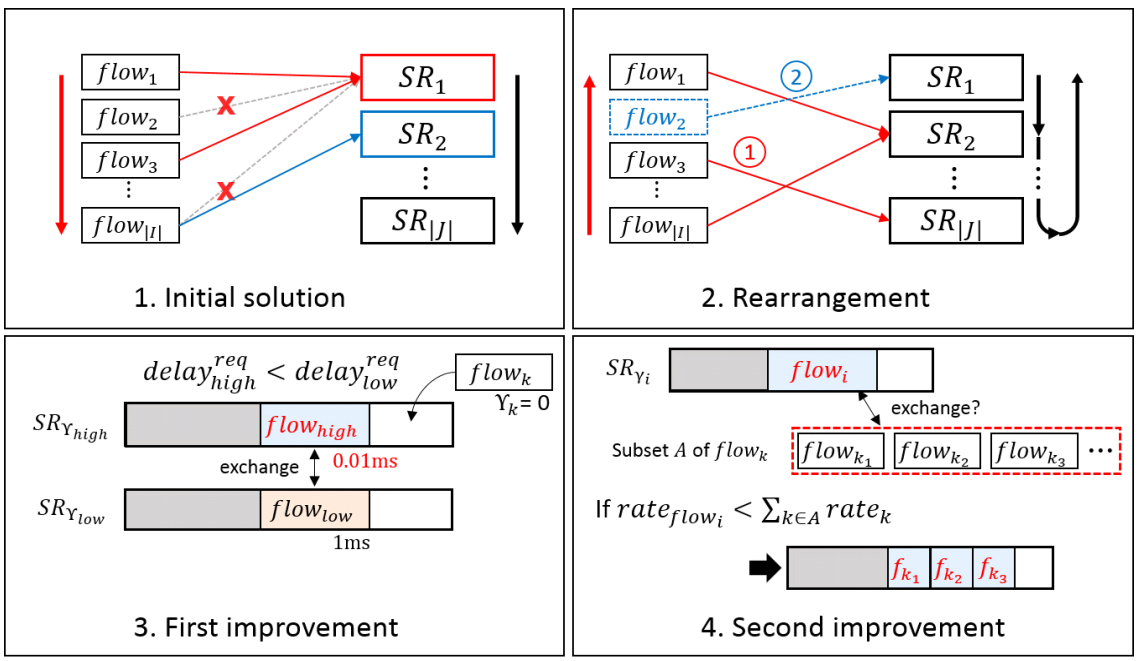


그림 2. GSRM의 매핑 과정
Fig. 2. Mapping process of GSRM

시도하는 단계이다. 서로 다른 SR에 매핑된 플로우를 1 대 1로 교환 시, 두 플로우 중 상대적으로 시간지연 요구조건이 짧아 큰 처리율을 요구했던 플로우가 속했던 SR에 교환된 상대적으로 시간지연 요구조건이 긴 플로우와 동시에 새로운 플로우가 매핑될 수 있게 하는 것이다. Fig. 2를 보면 상대적으로 시간지연 요구조건이 짧은 $flow_{high}$가 매핑에서 해제될 시, 해당 SR에 교환되는 시간지연 요구조건이 상대적으로 길어 $flow_{high}$보다 긴 시간지연을 수용할 수 있게 된 $flow_{low}$가 들어오게 되면서, 기존에 SR에 매핑되지 못했던 $flow_k$가 추가로 매핑될 수 있는 기회가 생기는 것이다. $flow_k$는 매핑 가능한 플로우 중 전송속도가 가장 큰 플로우로 선정한다.

4) Second improvement: 마지막 단계는 매핑된 플로우를 제외하고 기존에 매핑되지 못했던 플로우를 매핑하는 단계이다. 매핑된 한 플로우를 제외하였을 때 매핑되었던 SR에 기존에 매핑되지 못했던 플로우들 중 매핑 가능한 플로우 집합의 전송속도 합이 제외하기 전보다 크다면, 해당 플로우를 매핑에서 제외하고 새로운 플로우들을 매핑하는 과정이다. 단, 전송속도 합이 제외 전과 후가 같은 경우에는 재매핑하지 않는다. 정렬된 플로우의 역순으로 탐색하며 기존 매핑된 플로우를 대상으로 시도한다. 총 4단계가 모두 마무리되었을 때, 매핑된 결과대로 SR은 플로우를 서비스하게 된다.

제안하는 알고리즘의 진행순서는 Fig. 3과 같다. 먼저 플로우를 정렬 후, 초기화된 SR의 개수부터 시작하여 GSRM의 4단계를 순서대로 진행한다. 첫 단계로, SR이 플로우를 탐욕적으로 매핑 후, 매핑 정보를 가져오고 SR의 용량은 초기화한다. 플로우의 정렬은 initial solution의 앞 단계에서 진행하며, 한번 플로우가 정렬되면 반복 정렬하지 않는다. 두 번째 단계에서 매핑된 플로우의 역순으로 SR을 순회하며 재매핑 후, 남은 플로우에 대하여 SR이 플로우 순차적으로 다시 탐욕적 매핑을 수행한다. 이후 세 번째 단계에서 플로우 별로 플로우의 다음 인덱스에 해당하는 플로우부터 순차적으로 탐색하며 SR의 매핑을 교환하고, 새로운 매핑을 시도한다. 마지막 단계에서는 플로우 순차적으로 매핑을 제외 시 더 큰 이익을 가져오는 플로우들을 찾는다. 각 단계에서 수행한 매핑의 결과가 다음 단계로 전달되며, 4번째 단계가 끝날 경우 조건을 확인하여 QoS가 100% 만족되거나 최대 가용 개수로 진행할 때까지 SR의 개수를 한 개씩 늘려가며 알고리즘을 수행한다.

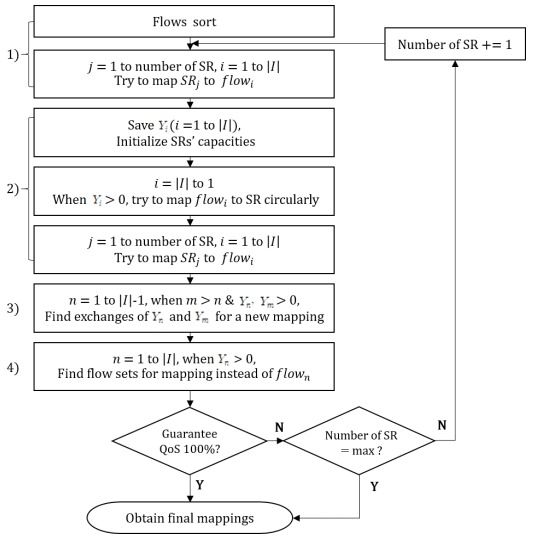


그림 3. GSRM 과정의 순서도
Fig. 3. The flow of GSRM process

V. 실험 및 결과

5.1 실험 환경

본 논문에서는 SR 매핑 시스템과 알고리즘의 성능 평가를 위해 CPU Intel Xeon Gold 6242R 4.1GHz, RAM 384GB 환경에서 알고리즘을 MATLAB으로 구현하고 실험을 진행하였다. Fig. 3.에 나타낸 순서도에 기반하여 알고리즘을 SR 개수에 따라 수행하였으며, Table 2의 여러 Case 별로 QoS가 만족된 트래픽의 비율(가, 나) 및 알고리즘이 수행된 시간(다)을 측정하는 실험을 각각 진행하였다. 마지막으로, QoS를 보장하기 위해 필요한 SR의 최소 개수를 알아내기 위해 SR 개수를 증가시키면서 매핑된 플로우의 비율(라)을 확인하는 실험까지 진행하였다. 제안된 GSRM의 성능을 분석하기 위하여, brute force 방식과 무작위 매핑 (random mapping)을 비교실험에 사용하였다. Brute force는 SR-플로우 매핑의 모든 경우의 수를 탐색하여 매핑된 전송속도 합이 가장 큰 경우를 선택하였다. Random mapping은 생성된 각 SR에 플로우를 무작위로 매핑하는 방식이다.

Table 2는 실험 별 플로우와 SR에 적용되는 인자값을 나타낸다. 플로우는 각 출처로부터 SR에 도달하는 전파지연, 전송속도, 시간지연 요구조건의 정보를 지니고, SR은 처리율, 플로우의 목적지별 전파지연 정보를 가진다. 플로우의 전송속도는 실험 별로 동일하게 30~80 Mb/s 의 constant bit rate를 지닌다. 플로우와 SR의 인자값에 따라 Case를 분류하였다. Case 1

표 2. 실험 Case들과 사용된 매개변수 값
Table 2. Experiment cases and used parameters

Case	플로우					SR		
	Source to SR 전파지연(ms)	전송속도 (Mb/s)	개수		시간지연 요구조건 (ms)	가용 개수	처리율 (Mb/s)	SR to Dest 전파지연(ms)
1	U(1, 20)	U(30, 80)	10	5	U(40, 50)	3	U(100, 200)	U(1, 20)
				5	U(120, 130)			
2			10	5	U(60, 70)	3		
				5	U(100, 110)			
3			40	20	U(40, 50)	10	U(100, 300)	
				20	U(120, 130)			
4			40	20	U(60, 70)	10		
				20	U(100, 110)			
5			40	20	U(60, 70)	20		
				20	U(100, 110)			

과 2는 플로우가 10개고 SR의 개수는 3이며 처리율은 100~200의 uniform 분포를 따른다. Case 3과 4는 플로우가 40개고 SR의 개수는 10이다. Case 5는 SR의 가용 개수가 20개이며 Case 3~5의 SR 처리율의 범위가 최대 300 Mb/s이다. 시간지연 요구조건 값은 uniform 분포를 따르며 전체 개수의 절반은 짧은 시간지연 요구조건을 갖고 나머지 반은 상대적으로 더 긴 시간지연 요구조건을 갖도록 설정했다. 예를 들어, Case 1은 시간지연 요구조건이 40~50 ms의 값인 플로우 5개, 120~130 ms의 값인 플로우 5개로 설정하였고, Case 2는 60~70 ms의 값인 플로우 5개, 100~110 ms 인 플로우 5개로 설정하였다.

5.2 실험 결과

5.2.1 SR 개수에 따른 트래픽 처리 성능 비교

Fig. 4의 그래프는 Table II의 Case 1~4에 대해서 가용 SR의 개수를 변화시키면서 전체 플로우 트래픽 총량 대비 QoS 요구조건이 만족된 플로우의 트래픽 량의 비율을 측정한 결과이다. Fig. 4(a)은 Case 1의 결과로 SR 개수가 1에서 3으로 증가할 때, brute force가 찾은 최적 매핑에서는 각 28%, 52%, 77%의 트래픽에 대해 QoS를 만족하도록 서비스가 가능하였다. 제안된 GSRM은 각각 27%, 47%, 71%의 성능을 보였으며 최적의 brute force 결과와 최대 6% 미만의 차이를 보였다. 이에 반하여 무작위 매핑은 brute force 결과와 최대 15% 이상 차이 나는 결과를 보였다. Fig. 4(b)는 Case 2의 결과로 GSRM은 18%, 43%, 69%의 성능을 보이며 19%, 46%, 71%의 성능을 보인 brute force 결과와 최대 2%의 미만의 차이를 보였다. 이러한 결과는 $O(|J|^2)$의 시간복잡도를 가지는 GSRM이 모든 경우를 탐색하여 최적의 해를 구하는 brute force와 비슷한 성능을 보여주고 있다. Brute force 방식은 $O(|J|^{|I|})$의 높은 시간복잡도를 가지며 SR의 개수가 3인 Case 1, 2에서만 적용 가능했다. Fig. 4(c)은 Case 3의 결과이며 GSRM은 SR의 개수가 1, 5, 10개 일 때 8%, 44%, 79%의 성능을 보여 9%, 35%, 60%의 성능을 보인 무작위 매핑과 SR 1개, 5개, 10개에서 0%, 9%, 19%의 매핑 비율의 차이를 보였다. Fig. 4(d)는 Case 4의 결과이며 SR의 개수가 1, 5, 10개 일 때 8%, 41%, 83%의 성능을 보인 GSRM과 무작위 매핑이 최대 16%의 성능 차이를 보였다.

5.2.2 시간지연 요구조건 특성에 따른 성능 비교

Fig. 4(a)의 매핑된 플로우의 전송속도 합 중 40~50 ms 요구조건을 가진 플로우의 처리 성능 결과는 다음과 같다. Brute force의 경우 SR 개수가 1, 2, 3일 때 전체 성공적으로 매핑된 전체 트래픽은 28%, 52%, 77% 이며, 이 중 40~50 ms 요구조건을 가진 플로우는 0%, 12%, 29%를 차지하며, 전체 매핑된 트래픽 대비 0%, 23%, 37%의 비율을 가진다. 전체 트래픽 중 40~50 ms 요구조건을 가진 플로우는 50%인 것을 고려하여 매핑된 결과를 보았을 때, brute force가 제약조건을 만족하는 트래픽량의 최대화를 위해 요구조건이 상대적으로 큰 120~130 ms 요구조건을 가진 플로우를 우선적으로 매핑한 것임을 알 수 있다. GSRM의 경우는 40~50 ms 요구조건을 가진 플

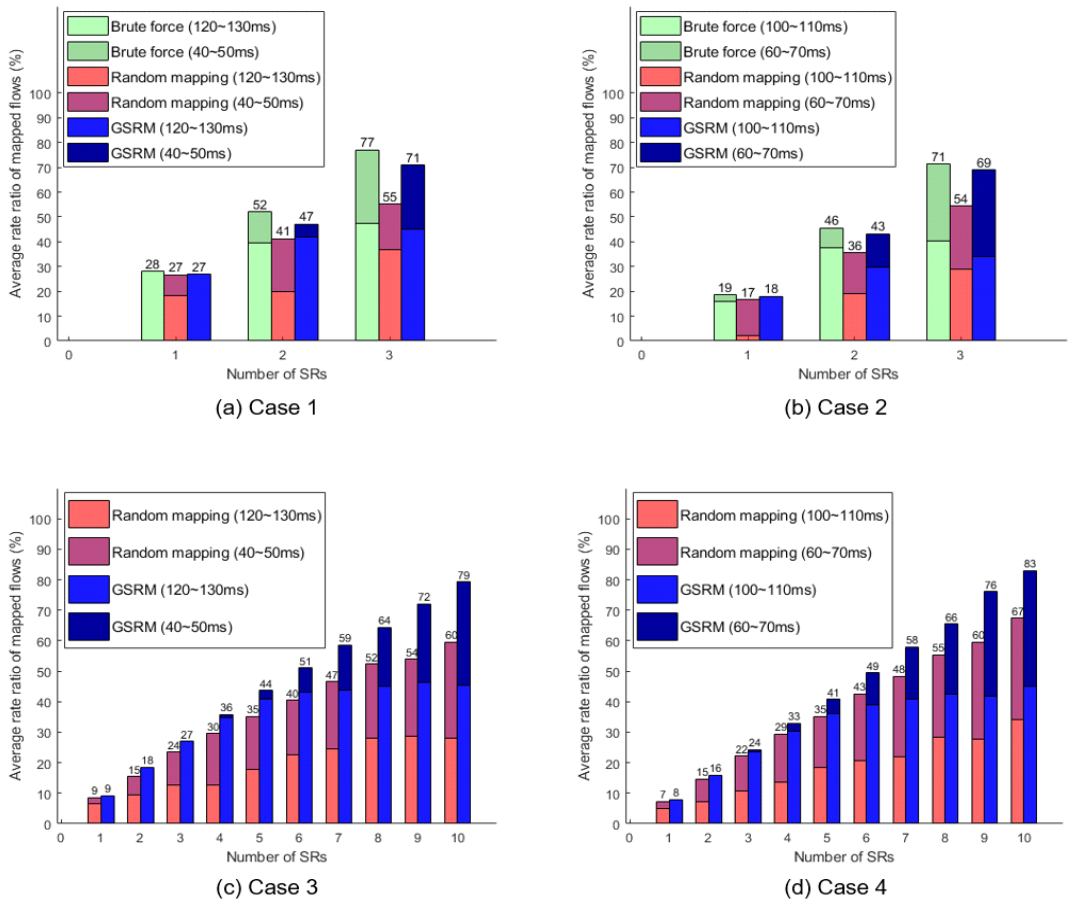


그림 4. Case 1-4의 QoS가 만족된 평균 트래픽 플로우의 전송속도 합 비율
Fig. 4. Average aggregate rate ratio of traffic flows with QoS satisfaction for Case 1-4

로우가 전체 매핑된 트래픽 대비 0%, 5%, 37%을 차지하였다. 그 결과로 GSRM도 최대화 목표를 위해 상대적으로 요구조건이 긴 플로우를 우선적으로 매핑하였음을 알 수 있다. 반면, 무작위 매핑의 경우 40~50 ms 요구조건을 가진 플로우가 30%, 51%, 33%을 차지하였다. Fig. 4(b)의 40~50 ms 요구조건을 가진 플로우의 매핑 성능은 다음과 같다. Brute force의 경우 전체 매핑된 트래픽 대비 16%, 17%, 42%의 비율을 가지며, GSRM의 경우는 0%, 30%, 51%의 비율을 가진다. 무작위 매핑은 각 82%, 47%, 48%의 비율을 가지며, 따라서 Fig. 4(b)의 성능은 Fig. 4(a)와 비슷한 경향을 지닌다. 이는 다른 알고리즘의 결과에 비해 무작위 매핑이 40~50 ms 요구조건을 가진 플로우를 더 매핑하였으며 이로 인해 전반적인 매핑 성능이 낮아진 것임을 알 수 있다. Fig. 4(c)에서 매핑된 플로우의 전송속도 합 중 40~50 ms 요구조건을 가진 플로우의 성능은 SR 개수가 각 1, 5, 10개일 때 다음과 같다. 무작위 매핑은 22%, 51%, 52%을 GSRM의 경우 각 0%, 7%, 43%을 차지한다. GSRM의 경우 SR의 개수가 커짐에 따라 매핑된 트래픽 대비 짧은 시간지연 요구조건을 갖는 트래픽의 매핑이 늘어나는 것을 확인할 수 있고, SR의 개수가 10인 경우에는 자원이 충분하여 50%에 근접하는 것을 확인할 수 있다. Fig. 4(d)는 Case 4의 결과이며 무작위 매핑의 경우 29%, 49%, 49%를 GSRM의 경우 0%, 12%, 46%를 차지하여 Case 3의 결과와 비슷한 경향을 지닌다.

자원이 부족한 상황에서 긴 시간지연 요구조건을 가진 플로우를 우선으로 매핑하고, 충분한 자원에서는 거의 동일한 비율로 플로우를 매핑한 것을 통해 GSRM은 주어진 각 자원의 처리율을 높여 총 트래픽 량을 최대화함을 알 수 있다. 따라서 GSRM은 자원의 효율적 사용으로 플로우의 QoS를 보장하는 매핑을

수행함을 보인다.

5.2.3 계산 복잡도에 따른 계산시간 결과

Fig. 5는 Case 1~4에 대해 SR 개수 별로 알고리즘 계산에 소요된 평균시간을 나타낸 것이다. Fig. 5(a)는 Case 1의 결과이며, SR 개수가 1, 2, 3일 때 brute force는 각각 1.43 시간, 3.17 시간, 7.28 시간이 소요되었고 무작위 매핑은 SR 개수 별 모두 5 ms 미만의 시간이, GSRM은 SR 개수 별 모두 10 ms 미만의 시간이 소요되었다. Fig. 5(b)는 Case 2의 결과이며, brute force는 1.45 시간, 3.19 시간, 7.31 시간이 소요되었고, 무작위 매핑은 SR 개수 별 모두 2ms 미만의 시간이, GSRM은 모두 5 ms 미만의 시간이 소요되었다. Fig. 5(c)는 Case 3의 결과로 SR 개수가 1, 5, 10개일 때, 무작위 매핑은 1.5 ms, 1.6 ms, 1.8 ms가 소요되며, GSRM은 7 ms, 56 ms, 82 ms로 소요되었고, Fig. 5(d)는 Case 4에서의 수행시간으로 무작위 매핑은 모든 개수에서 2 ms 미만으로, GSRM은 0.1초 미만으로 수행된다. 따라서, GSRM은 brute force에 준하는 성능을 훨씬 짧은 수행시간으로 도출하였다. 무작위 매핑과 비교했을 때는 GSRM이 비슷한 수준의 짧은 시간이 소요되면서 SR 개수가 증가함에 따라 더 큰 플로우 전송속도 합을 도출함을 알 수 있다. Fig. 5(c)에서 SR 개수가 8개일 때보다, SR 개수가 9개일 때 수행시간이 짧은 것을 볼 수 있는데, 이는 알고리즘의 각 단계의 탐색 및 비교 대상에 따라 iteration의 회수가 달라졌기 때문이다.

플로우를 한번 정렬 후 탐욕적으로 매핑하는

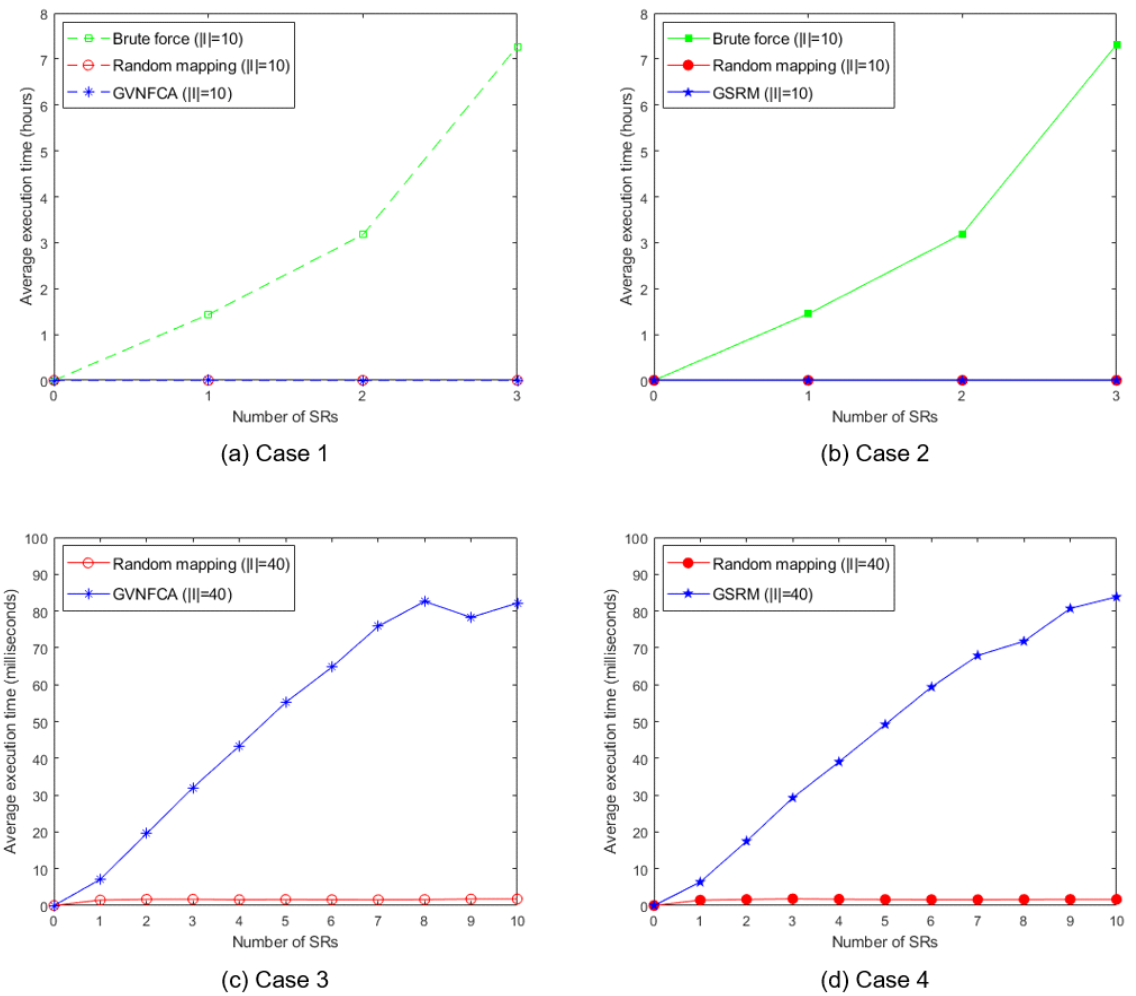


그림 5. 알고리즘의 평균 계산시간
Fig. 5. Average computation time of the algorithms

GSRM과 모든 경우의 수를 수행하고 최적의 매핑 결과를 도출하는 brute force의 매핑 방식의 차이에 따라 계산 복잡도가 낮은 GSRM이 우수한 수행시간 성능을 보임을 확인할 수 있다.

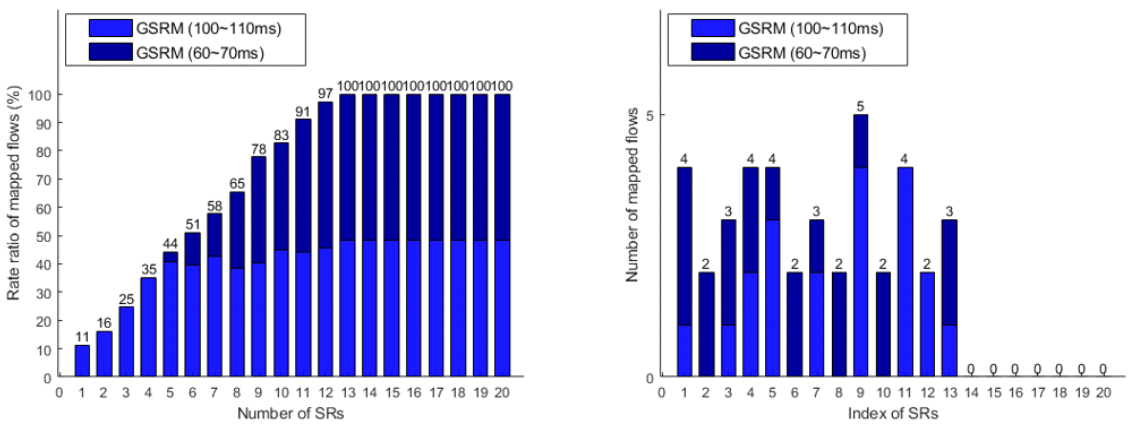
5.2.4 QoS 보장을 위한 SR의 개수

Fig. 6은 Case 5의 GSRM 실험 결과로, SR의 최대 개수는 20개이고, SR의 개수를 최대개수까지 늘려가며 매핑된 플로우의 비율을 보여준 결과이다. Fig. 6(a)에서 SR 개수가 13개일 때 플로우가 모두 매핑되어 성능이 100%로 나타났다. Fig. 6(b)는 SR의 최대 가용개수 20개 중 13개의 SR로 전체 플로우가 매핑되었을 때, 각 SR에서 매핑된 플로우의 수를 나타낸다. SR_1부터 SR_{13}까지는 플로우가 매핑되지만 SR_{14}부터 SR_{20}까지 매핑된 플로우의 개수가 0이다. 모든 플로우의 시간지연 요구조건이 만족되어 QoS가 보장된 상태에서는 추가적인 SR에 대해 더 이상 매핑을 시도하지 않음을 확인할 수 있다. GSRM은 QoS 보장을 위해 필요한 SR의 수를 도출할 수 있어 불필요한 자원의 낭비를 방지하고 효율적인 자원관리를 할 수 있다.

Ⅵ. 결 론

본 논문에서는 네트워크를 악의적 사용자로부터 보호함과 동시에 실시간 데이터 QoS 요구조건을 만족시키기 위해서, 물리적인 전용 보안 시스템과 VNF로 구성된 하이브리드 보안 자원 시스템 환경에서의 최적화 문제를 제시하고 이를 해결하는 알고리즘을 제안하였다. SRP 내에 존재하는 처리율이 각기 다른 분산된 SR이, 시간지연 요구조건과 전송속도가 다른 데이터 트래픽을 적절히 서비스하여 QoS가 향상되도록 하는 최적화 문제를 제시하였다. SR 내 시간지연은 M/D/1 queueing 모델로 구하였고, 각 플로우의 시간지연은 전체 시스템 내의 전파지연과 SR 시스템 지연의 합으로 구성되었다. 최적화 문제는 매핑 시 시간지연 요구조건이 만족된 플로우들의 전송속도 합이 최대화되는 것으로 정리되며, 해당 최적화 문제를 해결하기 위해 MKP를 적용한 탐욕 기반 알고리즘을 제안하였다. 계산 복잡도가 높거나 시간지연을 고려하지 않은 기존 연구들과 다르게, 제안하는 알고리즘은 시간지연 요구조건을 모델링된 시간지연과 비교하여 탐욕적으로 매핑하고 pseudo-polynomial 한 계산 복잡도로 최적화 문제를 해결한다.

MATLAB 환경에서 다른 알고리즘과의 비교실험을 통해, 제안된 알고리즘이 훨씬 짧은 시간으로도 brute force로 찾은 최적의 결과와 비슷한 성능을 낼 수 있음을 확인하였다. 제안하는 알고리즘은 자원의 용량이 부족한 경우에, 긴 시간지연 요구조건 값을 지닌 플로우를 우선 매핑하는 특징을 가지며 이로 인해 처리율을 높이는 결과를 가져옴을 확인하였다. 또한 제안된 알고리즘은 QoS 보장을 위해 필요한 자원의 개수를 예측할 수 있다. 사이버 보안이 필수적인 실제 네트워크 환경에 해당 알고리즘을 적용할 경우 데이터 트래픽의 QoS를 보장하며 최소한의 자원으로 보안 서비스를 제공할 수 있을 것으로 기대된다.



(a) Aggregate rate ratio of mapped flows (b) Number of mapped flows at each SR

그림 6. GSRM로 해결한 Case 5의 QoS가 만족된 트래픽 플로우의 전송속도 합 비율
Fig. 6. Aggregate rate ratio of traffic flows with QoS satisfaction for GSRM in Case 5

References

[1] V. Combs, *AT&T traffic report: Voice calls, IMs, and text messages are up, email and web traffic are down* (2020), Retrieved Dec. 22, 2020, from <https:// www.techrepublic.com/article/at-t-traffic-report-voice-calls-ims-and-text-messages-are-up-email-and-web-traffic-are-down/>

[2] R. Guerzoni, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action, introductory white paper," in *Proc. SDN and OpenFlow World Congress*, vol. 48, pp. 1-16, Darmstadt, Germany, Oct. 2012.

[3] R. Von Solms and J. Van Niekerk, "From information security to cyber security," *Elsevier Comput. & Secur.*, vol. 38, pp. 97-102, Oct. 2013.

[4] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet of Things J.*, vol. 6, no. 1, pp. 692-704, Feb. 2019.

[5] X. Luo, H. Wu, H. Yuan, and M. Zhou, "Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors," *IEEE Trans. Cybernetics*, vol. 50, no. 5, pp. 1798-1809, May 2020.

[6] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *Elsevier J. Netw. and Comput. Appl.*, vol. 80, pp. 200-218, Feb. 2017.

[7] G. Park, G. Lee, B. Roh, E. Kim, and D. Ryu, "A study on integrated structure and QoS policy for heterogeneous military transports," *J. KICS*, vol. 45, no. 2, pp. 409-417, Feb. 2020.

[8] J. Joung and J. Kwon, "Improved regulating scheduler for network delay guarantee," *J. KICS*, vol. 44, no. 6, pp. 1105-1112, Jun. 2019.

[9] S. Ahvar, H. P. Phyu, S. M. Buddhacharya, E. Ahvar, N. Crespi and R. Glitho, "CCVP: Cost-efficient centrality-based VNF placement and chaining algorithm for network service provisioning," in *Proc. IEEE Conf. NetSoft*, pp. 1-9, Bologna, Italy, Jul. 2017.

[10] P. Cappanera, F. Paganelli, and F. Paradiso, "VNF placement for service chaining in a distributed cloud environment with multiple stakeholders," *Elsevier Comput. Commun.*, vol. 133, pp. 24-40, Jan. 2019.

[11] Y. Jia, C. Wu, Z. Li, F. Le, and A. Liu, "Online scaling of NFV service chains across geo-distributed datacenters," *IEEE/ACM Trans. Networking*, vol. 26, no. 2, pp. 699-710, Apr. 2018.

[12] G. Sur, Y. Kim, S. Y. Ryu, and H. Lim, "Network function virtualization coordination for traffic flows with different quality of service requirement," in *Proc. KICS Summer Conf.*, pp. 969-970, Pyeongchang, Korea, Aug. 2020.

[13] A. H. M. Jakaria, B. Rashidi, M. A. Rahman, C. Fung, and W. Yang, "Dynamic ddos defense resource allocation using network function virtualization," in *Proc. ACM Int. Wrkshp. Secur. in Softw. Defined Netw. & Netw. Function Virtualization*, pp. 37-42, Scottsdale Arizona, USA, Mar. 2017.

[14] V. Jain and M. Agrawal, "Applying genetic algorithm in intrusion detection system of iot applications," in *Proc. IEEE Int. Conf. Trends in Electron. and Info.*, pp. 284-287, Tirunelveli, India, Jun. 2020.

[15] S. Bagheri and A. Shameli-Sendi, "Dynamic firewall decomposition and composition in the cloud," *IEEE Trans. Info. Forensics and Secur.*, vol. 15, pp. 3526-3539, 2020.

[16] S. Yoon, T. Ha, S. Kim, and H. Lim, "Scalable traffic sampling using centrality measure on software-defined networks," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 43-49, Jul. 2017.

[17] L. Durante, L. Seno, and A. Valenzano, "A formal model and technique to redistribute the

packet filtering load in multiple firewall networks," *IEEE Trans. Info. Forensics and Secur.*, vol. 16, pp. 2637-2651, 2021.

[18] R. Cahn, "*Wide area network design: Concepts and tools for optimization*," Morgan Kaufmann, p. 319, 1998.

[19] P. Toth and S. Martello, "*Knapsack problems: Algorithms and computer implementations*," Wiley, 1990.

서 기 원 (Giwon Sur)

2017년 2월 : 한동대학교 전산전자공학부 학사

2020년 3월~현재 : 광주과학기술원 전기전자컴퓨터공학부 석사과정

<관심분야> 컴퓨터 네트워크 자원 최적화, 사이버 보안, 강화학습

[ORCID:0000-0001-8085-3174]

배 재 원 (Jaewon Bae)

2017년 2월 : 충남대학교 전파공학과 학사

2019년 2월 : 광주과학기술원 전기전자컴퓨터공학부 석사

2019년 3월~현재 : 광주과학기술원 전기전자컴퓨터공학부 박사과정

<관심분야> 컴퓨터 네트워크, 사이버 보안, 블록체인, 프라이버시 보호

[ORCID:0000-0002-7527-3818]

류 선 열 (Shun Yuel Ryu)

2001년 2월 : 경상대학교 전자재료공학과 학사

2003년 2월 : 경북대학교 전자공학과 석사

2010년 9월~현재 : 국방과학연구소 선임연구원

<관심분야> 컴퓨터 네트워크, 사이버 보안

김 종 원 (JongWon Kim)

1987년 2월 : 서울대학교 제어계측 학사

1989년 2월 : 서울대학교 제어계측 석사

1994년 2월 : 서울대학교 영상통신 박사

2008년~현재 : 광주과학기술원 인공지능대학원 교수

<관심분야> Sustainable Orchestration of AI-inspired Cyber-Physical Services employing Shared Software-Defined Infrastructure and Common Cloud-native Platforms

임 혁 (Hyuk Lim)

1996년 2월 : 서울대학교 전기공학부 학사

1998년 2월 : 서울대학교 전기공학부 석사

2003년 8월 : 서울대학교 전기컴퓨터공학부 박사

2006년~현재 : 광주과학기술원 인공지능대학원 교수

<관심분야> 컴퓨터 네트워크, 사이버 보안, 인공지능

[ORCID:0000-0002-9926-3913]