쿠버네티스 기반 서비스 메시 환경에서의 mTLS 적용에 따른 네트워크 성능 분석

이희수 [1], 이재영 [2], 남재현 [3]
[1] 단국대학교 컴퓨터공학과 학부생
[2] 단국대학교 인공지능융합학과 석사과정
[3] 단국대학교 컴퓨터공학과 교수

heesu1117@dankook.ac.kr, leeja042499@dankook.ac.kr, namjh@dankook.ac.kr

Quantitative Analysis of Network Performance Impact of mTLS in Kubernetes-based Service Meshes

Heesu Lee[1], Jaeyoung Lee[2], Jaehyun Nam[3]
[1]Dept. of Computer Engineering, Dankook University (Undergraduate Student)
[2]Dept. of AI-based Convergence, Dankook University (Graduate Student)
[3]Dept. of Computer Engineering, Dankook University (Professor)

요 약

마이크로서비스 아키텍처의 확산에 따라 서비스 간 안전한 통신을 보장하는 기술의 중요성이 더욱 부각되고 있다. 서비스 메시는 이러한 환경에서 보안성과 관측 가능성을 제공하는 핵심 인프라이며, 그중 mTLS 는 상호 인증을 통해 통신의 기밀성과 무결성을 보장하는 주요 보안 메커니즘으로 활용된다. 본 논문은 Istio, Linkerd, Consul, Kuma 등 다양한 서비스 메시 환경에서 mTLS 적용이 네트워크 성능에 미치는 영향을 비교·분석하였다. 실험 결과, mTLS 를 활성화한 모든 환경에서 성능 저하가 관측되었으며, 이는 암호화 및 인증서 처리 과정에서 발생하는 연산 부담이 주요 원인으로 작용했을 가능성을 시사한다. 이러한 결과는 서비스 메시 도입 시 보안성과 성능 간의 트레이드오프를 고려한 mTLS 의 선택적 적용 전략이 필요함을 보여준다.

1. 서론

최근 클라우드 네이티브 환경에서는 쿠버네티스 기반의 마이크로서비스 아키텍처가 확산되고 있다. 마이크로서비스 아키텍처는 하나의 애플리케이션을 독립적인 서비스 단위로 분할하여 각각을 개별적으로 개발, 배포, 운영할 수 있도록 설계함으로써, 확장성, 장애 격리, 빠른 배포, 팀 간 병렬 개발 등 운영 효율성을 크게 향상시킨다 [1]. 그러나 서비스 간 통신이 빈번해지면서 네트워크 성능 저하와 보안 위협이 새로운 문제로 대두되고 있다 [2]. 특히, 민감한 데이터가 서비스 간에 전송되는 경우가 많아지면서 통신의 기밀성과 무결성을 보장하는 보안 메커니즘의 중요성이 부각되고 있다.

이러한 문제 해결을 위한 기술로 mTLS(mutual TLS)가 주목받고 있다. mTLS 는 통신하는 양측이 서로의 인증서를 검증함으로써, 중간자 공격을 방지하고, 데이터의 기밀성과 무결성을 확보할 수 있다. 그러나 암호화와 인증 과정에서 발생하는 연산 부담으로 인해 성능 저하가 발생할 수 있다. 본 논문에서는 주요 서비스 메시를 대상으로 mTLS 적용이 네트워크 성능에 미치는 영향을 정량적으로 분석하고자 한다.

2. 서비스 메시 개요

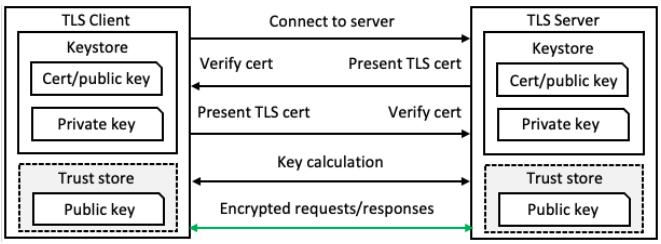
서비스 메시는 마이크로서비스 간 복잡한 네트워크 통신을 제어하고 관리하기 위한 인프라 기술로, 트래픽을 처리하는 데이터 플레인과 이를 제어하는 컨트롤 플레인으로 구성되며, 일반적으로 각 서비스 인스턴스에 사이드카 프록시를 배치하여 보안, 라우팅, 관측 등의 기능을 수행한다.

본 논문에서는 네 가지 대표적인 오픈소스 서비스 메시를 실험 대상으로 선정하였다. Istio 는 Envoy 프록시 기반으로 다양한 보안 및 관측 기능을 제공하며, 가장 널리 사용되는 서비스 메시 중 하나이다. Linkerd 는 자체 개발한 Rust 기반 경량 프록시를 통해 성능 효율성을 강조하며, Kuma 는 Envoy 기반이지만 설치와 운영이 간편하고 쿠버네티스 및 VM 환경 모두를 지원한다. Consul 은 서비스 디스커버리 중심의 프레임워크로 발전해왔으며, Envoy 프록시를 사용하여 다양한 인프라에 유연하게 적용 가능하다.

3. mTLS 개요 및 성능 영향

mTLS 는 TLS 기반 보안 통신에서 한쪽 인증만을 수행하는 일반적인 TLS 와 달리, (그림 1)과 같이 클

라이언트와 서버가 서로 인증서를 교환하고 신원을 검증하는 방식이다. 이로 인해 통신의 신뢰성을 양방향으로 보장할 수 있으며, 서비스 간 인증 및 권한 검증이 요구되는 환경에서 특히 중요하다. 인증서 기반의 신원 검증은 중간자 공격을 방지하며, 데이터 암호화를 통해 통신 내용을 보호한다.

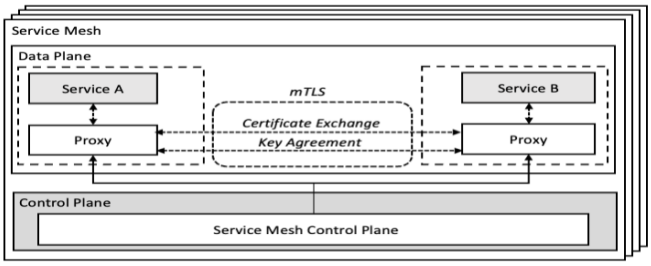


(그림 1) mTLS 동작 방식

그러나 이러한 보안 강화는 시스템에 연산 부담을 가중시킨다. 인증서 검증, 암호화 및 복호화 과정은 CPU 자원을 소모하며, 특히 고빈도 요청이 발생하는 마이크로서비스 환경에서는 이로 인한 성능 저하가 문제로 작용할 수 있다. 실제로 mTLS 적용 시 처리량 감소와 지연 시간 증가가 보고되고 있으며, 서비스 메시의 구현 방식과 프록시 구조에 따라 그 영향 정도는 달라질 수 있다.

4. 실험 환경 구성

실험은 Ubuntu 22.04 기반의 쿠버네티스 클러스터(v1.29.1)에서 수행되었으며, 서비스 메시로는 Istio (v1.25.1), Linkerd (v25.4.1), Consul (v1.21.0), Kuma (v2.9.0)를 각각 설정하였다.



(그림 2) mTLS 기반의 네트워크 성능 분석

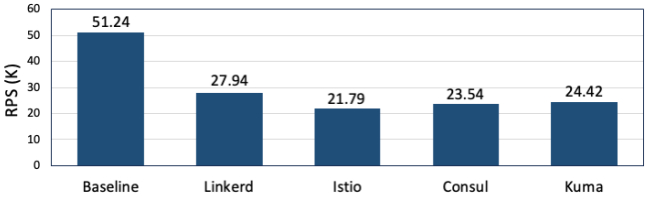
성능 측정은 mTLS 적용 여부에 따라 처리량(RPS)과 지연 시간(Latency)을 비교하는 방식으로 진행하였다. 실험 구조는 (그림 2)에 제시되었으며, wrk 도구를 통해 RPS 를, netperf 를 이용해 지연 시간을 측정하였다. 실험은 8 개의 스레드와 8 개의 동시 연결을 기반으로 수행되었다.

5. 실험 결과 분석

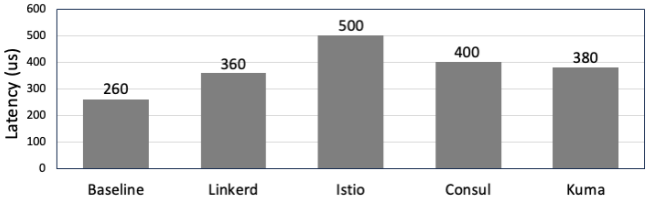
mTLS 적용 전후의 네트워크 성능을 비교한 결과 (그림 3, 그림 4), 전반적으로 처리량은 감소하고 지연 시간은 증가하는 경향이 나타났다. mTLS 가 적용되지 않은 베이스라인 환경에서는 처리량이 51.24K RPS, 지연 시간이 260 µs 로 가장 우수한 성능을 보였다.

mTLS 가 적용된 네 가지 서비스 메시 중에서는 Linkerd 가 27.94K RPS 와 360 µs 의 지연 시간을 기록하며 가장 우수한 성능을 나타냈다. 이는 Rust 기반 경량 프록시 구조의 연산 효율성이 mTLS 처리에 유리하게 작용했을 가능성을 시사한다. 반면 Istio 는 21.79K RPS 와 500 µs 로 가장 낮은 처리량과 가장 높은 지연 시간을 보여, Envoy 기반 프록시의 연산 부담이 성능에 부정적 영향을 미쳤음을 보여준다.

Consul 과 Kuma 는 각각 23.54K, 24.42K RPS 의 처리량과 400 µs, 380 µs 의 지연 시간을 기록하며 유사한 성능 양상을 나타냈다. 이들은 모두 Envoy 기반이지만 Istio 보다는 성능 저하가 상대적으로 덜하며, 설정 최적화 여부나 기능 경량화 수준의 차이가 영향을 미쳤을 수 있다. 전체적으로 서비스 메시의 구조와 프록시 구현 방식이 mTLS 적용에 따른 성능에 중요한 영향을 미치는 요인임이 확인되었다.



(그림 3) RPS 측정



(그림 4) Latency 측정

6. 결론

본 연구는 쿠버네티스 환경에서 다양한 서비스 메시를 대상으로 mTLS 적용이 네트워크 성능에 미치는 영향을 정량적으로 분석하였다. 그 결과, 모든 서비스 메시에서 mTLS 적용 시 성능 저하가 발생했으며, 이는 암호화 및 인증 절차에 따른 연산 부담이 주요 원인으로 작용했음을 보여준다. 특히 서비스 메시의 프록시 구조에 따라 성능 저하의 정도가 달랐으며, 경량 구조는 상대적으로 영향을 덜 받는 것으로 나타났다. 따라서 mTLS 의 보안적 이점을 효과적으로 활용하기 위해서는 서비스의 특성과 자원 제약을 고려한 유연한 적용 전략이 필요하다.

Acknowledgement

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학 ICT 연구센터(ITRC)의 지원을 받아 수행된 연구임(IITP-2025-RS-2023-00258649)

참고문헌

[1] K. M. Hasanth, et al., “Evaluating the Performance of Sidecar-based and Sidecarless Cloud-native service Mesh Solutions.”, ANTS, IEEE, 2024.

[2] K. Gunathilake, et al., “K8s Pro Sentinel: Extend Secret Security in Kubernetes Cluster.”, ICITR, IEEE, 2024.

[3] J. Hiller, et al., “The Case for Session Sharing: Relieving Clients from TLS Handshake Overheads.”, LCN Symposium, IEEE, 2019.