# Adaptive Circuit Breaker Thresholds in Istio: Comparative Analysis of Sidecar and Ambient Modes*

Seoyeon Kang[1], Jiwon Yoo[1], Kahyun kim[1], Hyeyeon Hyeon[1], and Seongmin Kim[1]

Sungshin Women's University, Seongbuk-gu, Seoul, Korea
{20211036, 220246055, 20211039, 20221143, sm.kim}@sungshin.ac.kr

**Abstract**

This study analyzes circuit breaker (CB) behavior in Istio's sidecar and ambient modes under identical configurations. Experimental results show that static configurations produce asymmetric blocking and latency patterns due to architectural differences between the two modes. These findings underscore the need for dynamically adjustable threshold values and adaptive, mode-aware policies optimized for stable and efficient traffic management in cloud-native environments.
***Keywords***— Service mesh, Sidecar, Ambient, Istio, and Circuit Breaker.

## 1 Introduction

In cloud-native environments, service meshes are widely used to manage inter-service communication, with the sidecar mode being the most prevalent architecture. However, it introduces operational overhead, as a proxy must be deployed alongside each workload. To mitigate this, Istio recently introduced the ambient mode, which decouples the lightweight Layer 4 (L4) proxy from the Layer 7 (L7) proxy to reduce proxy management complexity and enhance security.

While prior studies have primarily compared Istio sidecar and ambient modes in terms of architecture and performance [1, 2], few have examined the configuration challenges that arise during transitions. In particular, research remains limited on transition mechanisms and configuration parameters that should be considered for effective migration between the two modes. A static transition often fails to account for the ambient mode's structural and traffic-processing characteristics, leading to increased latency or even service level agreement (SLA) violations. Therefore, efficient transition and operation require a dynamic transition approach and performance evaluation under various configuration conditions.

This study adopts the circuit breaker as a representative case, applying identical configurations to both sidecar and ambient modes to conduct a cross-comparison of their traffic behaviors. Based on this analysis, we demonstrate the need for dynamic, mode-aware policy management, such as adjusting threshold values according to each mode's characteristics, and highlight the inefficiencies inherent in static transitions.

## 2 Showcasing Example: Circuit Breaker

We compare circuit breaker behavior between Istio sidecar and ambient modes under identical configurations. Experiments were conducted on a Kubernetes cluster (Istio 1.27.1) using the Bookinfo microservice application and the Fortio load generator to measure success

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. P-9, December 16-18, 2025, Sapporo, Japan. 

rate, latency, and throughput. Each mode was evaluated for 60 seconds under two conditions: 1) without circuit breaker configuration and 2) with identical threshold settings applied. For the latter, the circuit breaker parameters were intentionally set to low threshold values (`http1MaxPendingRequests=1`, `maxRequestsPerConnection=1`, and `maxConnections=4`) to clearly observe the blocking effect.

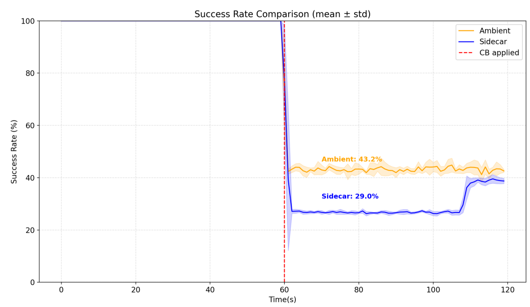


Figure 1: Comparison of success rate (ratio of successful HTTP 200 OK responses)

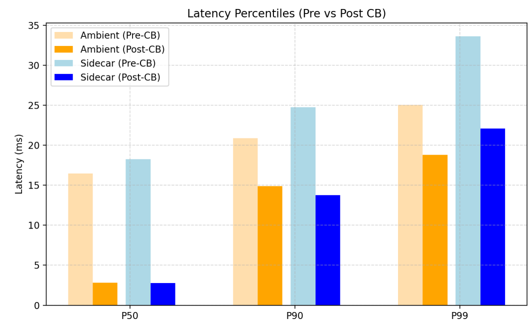


Figure 2: Latency comparison between sidecar and ambient mode

As shown in Figure 1, both modes exhibited a sharp decline in success rate immediately after circuit breaker activation. However, the sidecar mode showed a steeper drop and maintained a lower overall success rate (29%) compared to the ambient mode (43%), which sustained a more stable performance. These results indicate that the sidecar configuration enforces stricter blocking behavior, whereas the ambient mode permits slightly more requests to pass through.

Figure 2 illustrates the latency distribution before and after circuit-breaker activation. Both modes experienced an overall latency reduction after activation. The sidecar mode maintained lower latency at P50 and P90 but showed a slight increase at P99, suggesting longer L7 proxy queuing under high load. In contrast, the ambient mode exhibited a steadier yet slightly higher latency pattern, reflecting its lightweight L4-level blocking behavior.

Overall, static circuit-breaker configurations cause asymmetric behavior between the two modes: the sidecar mode over-restricts traffic, whereas the ambient mode provides smoother but less aggressive control.

## 3 Conclusion and Future Research Discussion

This study demonstrated that identical circuit breaker configurations produce divergent outcomes between the sidecar and ambient modes due to their architectural differences. The findings highlight the necessity of dynamically adaptive policies that respond to real-time service indicators—such as traffic rate, error ratio, and latency percentile—to maintain reliability in cloud-native environments. Future work will implement a feedback-based adaptive circuit breaker leveraging runtime telemetry from Prometheus and Grafana, aiming to minimize unnecessary blocking while sustaining throughput and latency stability under variable workloads.

## References

[1] ChoiHyun Kyeong et. al. Ambient mode and sidecar mode in istio: A quantitative analysis of operations and performance. *Annual Conference of KIPS*, pages 69–70, 2025.

[2] Mohammad R. S. Sedghpour et. al. An empirical study of service mesh traffic management policies for microservices. In *Proceedings of the International Conference on Performance Engineering (ICPE)*, 2022.