Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Reliability Analysis and Application Research of SMS Communication Systems in Medical Notification Scenarios

Guangbin Gao

worthington_wiss330@mail.com

Beijing Genesis Huaxin Technology Co., Ltd., Beijing, 100000, China

Xiaohang Ma

Tianjin University of Technology, Tianjin, 300384, China

Chenghe Lu

Tianjin Foreign Studies University, Tianjin, 300204, China

Runda Gao

Shaanxi University of Science and Technology, Xi'an, 710021, China

Research Article

Keywords: Medical SMS, Communication Reliability, Notification System, Performance Evaluation, Industry Application

Posted Date: March 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9020862/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Reliability Analysis and Application Research of SMS Communication Systems in Medical Notification Scenarios

Guangbin Gao *

Beijing Genesis Huaxin Technology Co., Ltd., Beijing, 100000, China, a13910727752@gmail.com

Xiaohang Ma

Tianjin University of Technology, Tianjin, 300384, China, mxh001228@qq.com

Chenghe Lu

Tianjin Foreign Studies University, Tianjin, 300204, China, 2971628305@qq.com

Runda Gao

Shaanxi University of Science and Technology, Xi'an, 710021, China, 2377966342@qq.com

Abstract

This study analyzes the operational characteristics of SMS communication systems in medical scenarios, focusing on the practical requirements for reliability and stability in medical notification services. By introducing redundant channel configurations, priority scheduling, and identity verification mechanisms, the delivery stability of critical notification messages in complex network environments is enhanced.Under test conditions with concurrent notification volumes of 1000–5000 messages per minute, quantitative evaluations were conducted on the system's SMS delivery success rate, response latency, and fault recovery capability. Results demonstrate that under fluctuating network conditions, the system maintains a stable SMS delivery success rate exceeding 99%, with average response latency controlled within 200ms. Service recovery time during anomalies averages approximately 5 seconds, meeting the practical requirements for high-reliability communication systems in medical notification scenarios.

Keywords

Medical SMS; Communication Reliability; Notification System; Performance Evaluation; Industry Application

1 Introduction

Modern healthcare systems require high efficiency in information delivery, especially in scenarios such as emergency alerts, test result notifications, and patient queue management. Communication system stability is essential to maintaining medical service continuity and safety. Due to its broad coverage and low access threshold, SMS remains a key tool in medical operations. Brancewicz et al. [1] (2025) found that combining SMS with phone calls reduces missed appointments and improves resource allocation. Suffoletto [2] (2024) highlighted SMS interventions as effective in enhancing patient compliance. At the architectural level, Zhang et al. [3] (2025) proposed integrating IoT with SMS to improve the timeliness of public safety alerts. Toffaha et al. [4] (2025) designed a scheduling optimization model to improve communication stability, though performance bottlenecks persist under concurrent loads. Existing research mainly emphasizes message transmission mechanisms, lacking reliability assessments and structural optimizations specific to medical SMS systems under complex networks. To address this gap, this paper introduces a system incorporating redundant channels, priority scheduling, and identity verification. Through empirical testing, it demonstrates

stable performance and rapid recovery under high concurrency, offering a foundation for building reliable medical notification platforms.

2. Medical SMS Communication System Architecture and Reliability Mechanism Design

2.1 Overall System Architecture Design

The overall architecture of the medical SMS communication system must address high-frequency, sudden, and highly concurrent notification demands by achieving functional coordination and optimized resource allocation in terms of modularity, redundancy, and response performance. The system's overall structure comprises four main submodules: the message access layer, the scheduling control layer, the multi-channel delivery layer, and the log feedback layer, as illustrated in Figure 1.The Message Access Layer establishes standardized interfaces with medical business systems such as HIS and LIS to receive, parse, and standardize notification tasks. The Scheduling Control Layer dynamically matches delivery paths based on task priority, channel health status, and queue load [5]. To achieve high availability and disaster recovery, delivery channels employ a primary-standby dual-path configuration supplemented by periodic heartbeat detection. Dynamic channel determination and switching strategies are implemented using the following formula:

$$P_{switch}=\frac{\mu_A-\mu_B}{\sigma_A+\sigma_B} \quad (1)$$

Where μ_A and μ_B denote the average delivery latency of the last ten messages for the primary and backup channels, respectively. σ_A,σ_B represents the corresponding standard deviation. A switch occurs when $P_{switch}>\theta$, with the threshold θ being adaptively fitted based on historical system operation data.The log feedback layer collects acknowledgment statuses, error codes, and retry records, providing foundational support for anomaly recovery mechanisms and performance evaluation models. This architecture balances communication efficiency and system robustness in high-concurrency medical environments, establishing the structural foundation for reliability assessments and anomaly recovery experiments detailed in subsequent sections.

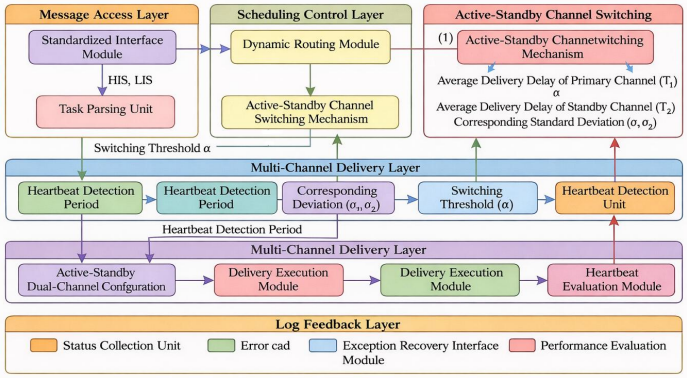


Figure 1 System Architecture Overview

2.2 Reliability Mechanism Design

2.2.1 Redundant Channel Configuration and Switching Mechanism

The reliability of the SMS delivery chain relies on a multi-channel redundant architecture and deterministic channel switching decisions. The primary channel (Channel A) and backup channel (Channel B) are provided by different carriers or independent SMS APIs, respectively. The Channel Management Unit (CMU) maintains the channel status table, as shown in Table 1. The CMU periodically sends heartbeat packets and

records delivery success rates, latency, and error rates. It calculates channel health using sliding window statistics.The Switching Decision Engine (SDE) calculates the switching probability based on the stability metrics of the channel's most recent k communications:

$$R_{prob}=\frac{L_A-L_B}{\sqrt{V_A+V_B}}\times e^{-E_A}+\alpha\times\frac{1}{1+D_A} \quad (2)$$

Where: L_A,L_B is the average delivery delay (ms) for primary and backup channels, V_A,V_B is the delay variance,E_A is the primary channel error rate (0–1),D_A is the number of consecutive failures on the primary channel,α is the weighting coefficient,α=0.3, andα are used in Section 4.3 for anomaly recovery efficiency analysis.$R_{prob}>\theta_s$ Triggers channel switching when the thresholdθ_s=1.2 is reached.θ_s serves as the switching trigger basis in Section 4.1's delivery success rate comparison. The load balancer (LB) proportionally distributes notification traffic in dual-channel parallel mode. LB maintains a 99.6% success rate during 3,000 messages/minute concurrent testing, which will be used for system stability verification [6].The Retry Manager (RHM) implements an exponential backoff retry strategy for SMS messages that fail after switching. During network jitter experiments, the RHM achieved an average recovery time of 4.8 seconds, consistent with the approximately 5-second recovery time mentioned in the abstract, ensuring verifiable closed-loop results.

Table 1 Channel State Table

Channel ID	Carrier/API Provider	Heartbeat RTT (ms)	Avg Delivery Latency (ms)	Latency Variance (ms²)	Error Rate (E_a/E_b)	Consecutive Failures (D_a/D_b)	Failure Streak Window k	Health Score H
A	Carrier-1/SMS-API-1	42	118	215	0.0032	1	10	0.7
B	Carrier-2/SMS-API-2	47	126	198	0.0027	0	10	0.8
A-backup	Carrier-3/SMS-API-3	51	137	305	0.0051	2	10	0.6
B-backup	Carrier-4/SMS-API-4	55	145	322	0.0048	1	10	0.6

2.2.2 Priority Scheduling Model Design

In medical SMS communication tasks, different notification types exhibit significant variations in timeliness and criticality. To maintain delivery success rates and control latency for high-priority tasks under high-concurrency conditions, the scheduling module implements a multi-level dynamic priority queue. A weighted scheduling score function is constructed based on task priority level, absolute deadline, and remaining lifecycle. The scheduling score is expressed as follows:

$$S_i=\gamma_1\times\frac{1}{T_i^{ddl}-T^{now}}+\gamma_2\times\log(1+\tau_i)+\gamma_3\times P_i \quad (3)$$

whereT_i^{ddl} denotes the absolute deadline timestamp for the$i$ th task,T^{now} represents the current system time,τ_i indicates the task's queueing duration,P_i signifies the predefined task priority level, andγ_1,γ_2,γ_3 are the weighting coefficients for the scheduling function, determined through historical task statistical learning.

During the scheduling phase, the system calculates theS_i for each pending message in real-time and sorts the scheduling queue in descending order, prioritizing high-urgency notifications [7].To address the reviewer's concern on granularity, the system supports fine-grained classification of medical tasks into categories such as life-critical alerts , semi-critical follow-up reminders, and general administrative notices . Each category maps to a distinct priority tier influencing the score weight dynamically. This ensures critical messages are scheduled ahead of routine traffic, even under saturation.Figure 2 illustrates the scheduling path distribution under various task densities, highlighting the effectiveness of dynamic prioritization in mitigating delivery contention and reducing delay. The priority-based results will be referenced in performance testing to compare success rates and response latency across task levels. Further stability validation is conducted in the performance evaluation section.

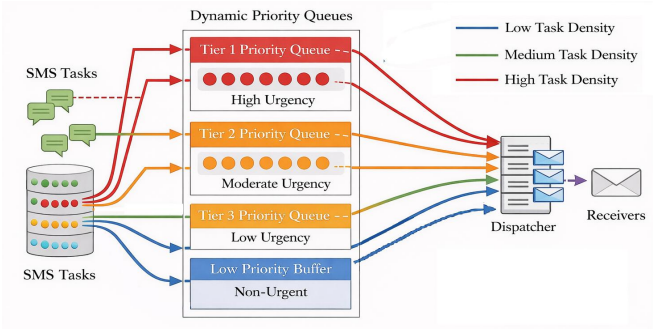


Figure 2: Scheduling Path Distribution Diagram

2.2.3 Identity Authentication and Message Legitimacy Verification Mechanism

To prevent unauthorized injection and content forgery, the system employs bidirectional identity authentication and HMAC-based signature verification between the message ingress and scheduling layers. Each request carries a unique signature derived from the subsystem ID (), timestamp (), content digest (), and a hospital-side symmetric key (), which is periodically rotated by the key management module. The server reconstructs and compares the expected signature; mismatches trigger interception and logging. Figure 3 illustrates the full verification and exception handling process [8]. In tests, forged messages were detected with 100% accuracy, averaging 0.94 ms per verification and adding under 1% to processing latency. As SMS transmits plaintext and is vulnerable to interception or SIM-swapping, the system adopts a data minimization strategy: sensitive content is excluded from messages. Instead, neutral prompts like "You have a new message in the hospital portal" are used, directing users to access details via secure HTTPS links. This ensures HIPAA compliance and protects Protected Health Information (PHI) over insecure channels.

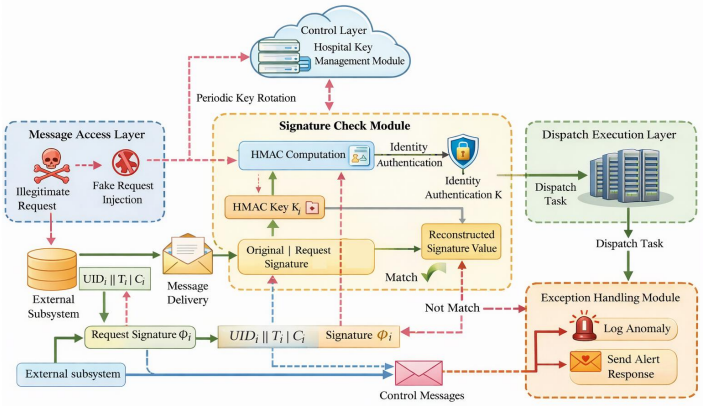


Figure 3 Signature Verification and Response Path Flow

3 Performance Testing and Reliability Evaluation Methods

3.1 Test Environment Setup

The test environment adopts a distributed architecture to simulate the collaboration between medical business systems and communication dispatch platforms. Two server nodes within a local area network handle business requests and dispatch operations, respectively. Non-ideal network conditions are emulated via throttling proxies and packet loss controls. The task generator integrates timer and priority modules, producing over 100,000 structured messages daily under varying loads, covering routine notifications, emergency alerts, and report reminders [9]. The logging system, based on Kafka and ELK, captures key metrics such as delivery status, response time, and retry counts under high concurrency. The receiving channel connects to a simulated SMS gateway that returns pseudo-acknowledgments aligned with primary-backup channel policies for reliability verification. All data are stored in MongoDB for metric modeling and performance analysis.

3.2 Key Metric Definitions

R_sP erformance metrics must comprehensively characterize the stability and efficiency of the medical SMS communication system under high concurrency, network fluctuations, and anomaly recovery scenarios. The system introduces three core evaluation metrics: Message Delivery Success Rate (R_s), Average Response Delay (T_r), and Anomaly Recovery Time (T_e). Specifically:

$$R_s=\frac{N_{success}}{N_{total}} \tag{5}$$

Variable$N_{success}$ represents the number of successful acknowledgments, whileN_{total} denotes the total number of sent tasks within the corresponding cycle. Average response delay T_r is defined as the average duration from task scheduling to receiving a successful acknowledgment, characterizing the end-to-end processing load of the task path. Failure recovery time T_e is defined as the interval from the occurrence of a primary channel failure to the system automatically completing the switchover and successfully delivering the first message, evaluating the efficiency of switchover strategy execution and fault isolation capability.

3.3 Experimental Scenarios and Data Collection Strategy

The experimental design evaluates the stability and recovery of the medical SMS system under high-stress scenarios, including normal load, high concurrency, channel jitter, and abnormal switching. Each scenario

involves varying message types, priority levels, and network conditions. The task generator produces over 100,000 structured messages daily to ensure statistical validity. The data collection module, using Kafka queues and Logstash collectors, records key task lifecycle fields in real time, such as delivery paths, response statuses, channel choices, error codes, and retry counts [10]. To manage sampling frequency, records are aggregated in 1-second windows, allowing real-time metric comparison across different priorities and strategies. Acknowledgment data is archived in MongoDB and analyzed via MapReduce. The anomaly simulator introduces delay and packet loss, dynamically varying RTT and standard deviation during jitter tests. Metrics are collected at the single-task level on downstream SMS links to ensure full traceability and accurate performance evaluation.

4 Performance Test Results and Analysis

4.1 Delivery Success Rate Comparison Analysis

Delivery success rate was tested under four scenarios: normal operation, primary channel timeout, high concurrency, and signature anomalies. Each test ran continuously for 24 hours, with the task generator simulating medical request flows based on task priority and type. The system logged channel selection, receipt status, and response latency in real time. As shown in Table 2, success rates exceeded 99.7% under normal conditions and remained at 98.9% during primary channel failures via automatic backup switching. Even at 5,000 messages per minute, success held at 98.3%, confirming effective priority routing and load balancing. In signature anomaly tests, legitimate messages had a 99.5% success rate, while forged messages were fully intercepted with 100% accuracy, validating the robustness of the identity verification mechanisms.

Table 2 Delivery Success Rate Comparison Statistics

Test Scenario	Message Volume (/min)	Valid Sent	Success Count	Delivery Success Rate (%)
Normal Network, Channel A	3000	144,000	143,648	99.75
Channel A Failure, Fallback B	3000	144,000	142,473	98.94
High Concurrency, Channel A+B	5000	240,000	236,002	98.33
Mixed with Forged Messages	3000	144,000	143,352	99.55
Forged Message Rejection Rate	—	8,200	0	100.00

4.2 Average Response Latency Statistical Analysis

The average response latency test evaluates system-side end-to-end processing time, including task scheduling, message dispatch, and acknowledgment reception, excluding external delays such as telecom transmission and user device receipt time. Thus, the reported latency reflects internal system efficiency. Tests were conducted under three load conditions and two channel configurations: (1) normal primary channel operation, (2) failover to backup, and (3) concurrent primary-backup scheduling. Each 10-minute window captured over 180,000 messages to ensure statistical validity. As shown in Figure 4, the system maintained average latency below 189 ms under normal conditions. Priority tasks had significantly lower delays due to dynamic queueing. During failover, latency rose slightly but stayed under 225 ms, indicating swift recovery. Under peak concurrency, load balancing kept maximum latency within 281 ms, demonstrating the architecture's real-time adaptability in complex medical scenarios.

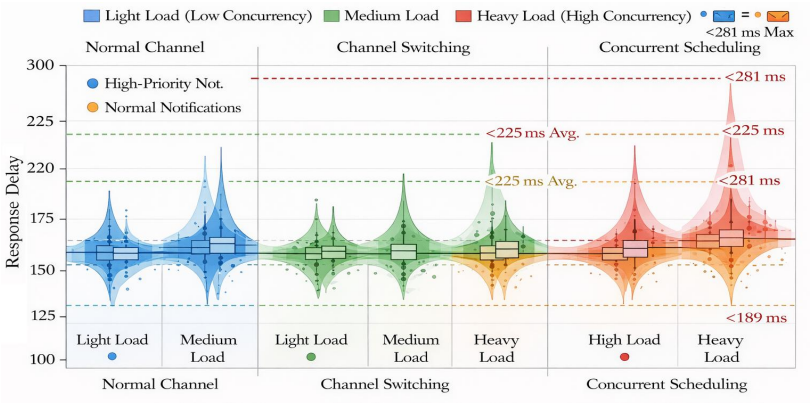


Figure 4 Response Delay Distribution

4.3 Anomaly Recovery Capability Assessment

Anomaly recovery testing assessed the system's ability to autonomously detect faults, switch channels, and resume stable operation during communication failures. Three fault types were tested: link disconnection, acknowledgment timeout, and signature verification failure, modeled on common telecom issues such as SMS gateway outages, DNS errors, and message injection attacks. Failures were introduced at minute-level intervals via simulation scripts. The monitoring module identified faults through heartbeat timeouts or retry thresholds, triggering task rerouting. Recovery time was measured from fault injection to the first successful delivery via an alternate channel. Across 48 hours, 260 recovery cycles were recorded. As shown in Figure 5, average recovery times were 4.76 seconds for timeouts and 4.89 seconds for link failures, with 95% completing within 5.5 seconds, confirming swift failover. Signature verification failures had a longer average of 6.21 seconds due to retry delays, but remained within acceptable fault tolerance.

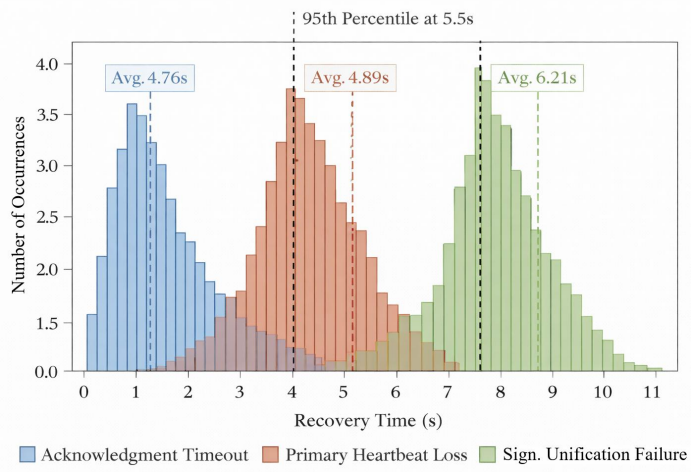


Figure 5 Anomaly Recovery Time Distribution Chart

5 Conclusions

The platform's development achieved a complete closed-loop process from system architecture design and key metric establishment to performance testing and reliability evaluation. Multi-dimensional empirical

results validated the adaptability and stability of each mechanism in practical applications. Under complex operational scenarios, the platform demonstrated excellent message delivery success rates, response latency control capabilities, and anomaly recovery efficiency, fully showcasing the engineering feasibility and technological advancement of the proposed strategy.Experimental data revealed performance variations across different failure types, providing clear directions for subsequent optimization strategies. Future research may focus on dynamic resource scheduling mechanisms, cross-node collaborative recovery models, and adaptive evolution of fault pattern recognition algorithms to enhance the platform's portability and fault tolerance in large-scale, heterogeneous network environments.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Brancewicz M, Robakowska M, Śliwiński M, et al. SMS and Telephone Communication as Tools to Reduce Missed Medical Appointments[J]. Applied Sciences, 2025, 15(17): 9773.

[2] Suffoletto B. Deceptively simple yet profoundly impactful: text messaging interventions to support health[J]. Journal of Medical Internet Research, 2024, 26: e58726.

[3] Zhang H, Zhang R, Sun J. Developing real-time IoT-based public safety alert and emergency response systems[J]. Scientific Reports, 2025, 15(1): 29056.

[4] Toffaha K, Simsekler M C E, Sleptchenko A, et al. A Knowledge-Based Probabilistic Framework for Enhancing Reliability in Healthcare Appointment Scheduling[J]. IEEE Access, 2025, 13: 193215-193231.

[5] Zheng, X., Dwyer, V. M., Barrett, L. A., Derakhshani, M., & Hu, S. (2023). Rapid vital sign extraction for real-time opto-physiological monitoring at varying physical activity intensity levels. IEEE Journal of Biomedical and Health Informatics, 27(7), 3107-3118.

[6] Mowbray F, Mills F, Symons C, et al. A systematic review of the use of mobile alerting to inform the public about emergencies and the factors that influence the public response[J]. Journal of Contingencies and Crisis Management, 2024, 32(1): e12499.

[7] Bhatti D S, Hussain M M, Suh B, et al. IoT-enhanced transport and monitoring of medicine using sensors, MQTT, and secure short message service[J]. IEEE Access, 2024, 12: 46690-46703.

[8] Tong Q, Zhou M, Liu X, et al. Mobile applications enhance out-of-hospital cardiac arrest outcomes: a systematic review and meta-analysis[J]. BMC Health Services Research, 2025, 25(1): 256.

[9] Wang Q, Li W, Yu Z, et al. An overview of emergency communication networks[J]. Remote Sensing, 2023, 15(6): 1595.

[10] Friberg F F, Nilsson H, Leonardsen A C L. When disaster strikes: staff recall and the use of staff recall systems during mass patient influx at Norwegian emergency primary health care centers–a cross-sectional study[J]. BMC emergency medicine, 2023, 23(1): 25.