Computer Science and Application 计算机科学与应用, 2026, 16(11), 1-11
Published Online November 2026 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2026.1611321

Hans汉斯

云边协同收费系统多级容灾与状态同步机制研究

于　梦[1]，刘冠男[1]，杨　娜[2*]

[1]河北高速公路集团有限公司承德分公司，河北　承德
[2]哈尔滨工业大学(威海)汽车工程学院，山东　威海

收稿日期：2026年9月30日；录用日期：2026年10月28日；发布日期：2026年11月4日

摘　要

针对高速公路收费系统云边协同部署后链路抖动、服务不可用、节点异常和数据不同步等问题，提出一种多级容灾与状态同步机制。首先，从链路层、业务层、计算层和数据层划分系统故障模式，构建分层容灾控制模型，并建立链路保护、异常判定和业务恢复之间的时间尺度约束关系；其次，基于WebSocket双连接心跳探测机制，定义服务可用性向量和系统工作模式变量，构建业务层降级有限状态机，实现云站正常、疑似异常、降级判定、广场降级、恢复探测和人工确认回切等状态流转；最后，结合实站测试数据对模型进行验证。结果表明，主备切换动作最大触发时间为3 s，业务中断时间最大值为40 s，满足业务层异常判定与降级切换及恢复约束。该机制能够支撑云边协同收费系统在异常场景下实现快速降级、业务恢复和稳态回切。

关键词

云边协同，多级容灾，状态同步，WebSocket，有限状态机

Research on Multi-Level Disaster Recovery and State Synchronization Mechanism for Cloud-Edge Collaborative Toll Collection Systems

Meng Yu[1], Guannan Liu[1], Na Yang[2*]

[1]Chengde Branch of Hebei Expressway Group Co., Ltd., Chengde Hebei
[2]School of Automotive Engineering, Harbin Institute of Technology (Weihai), Weihai Shandong

[*]通讯作者。

文章引用：于梦，刘冠男，杨娜. 云边协同收费系统多级容灾与状态同步机制研究[J]. 计算机科学与应用, 2026, 16(11): 1-11. DOI: 10.12677/csa.2026.1611321

Received: September 30, 2026; accepted: October 28, 2026; published: November 4, 2026

Abstract

To address the problems of link jitter, service unavailability, node anomalies, and data inconsistency arising from the cloud-edge collaborative deployment of highway toll collection systems, a multi-level disaster recovery and state synchronization mechanism is proposed. First, system failure modes are classified into the link, business, computing, and data layers, based on which a hierarchical disaster recovery control model is established, together with the time-scale constraints among link protection, anomaly determination, and business recovery. Second, based on a WebSocket dual-connection heartbeat detection mechanism, a service availability vector and system operation mode variables are defined, and a finite-state machine for business-layer degradation is constructed to realize state transitions among normal cloud-station operation, suspected anomaly, degradation determination, plaza-level degradation, recovery detection, and manual confirmation for switchback. Finally, the model is validated using data collected from actual toll station tests. The results show that the maximum trigger time for the primary-backup switching action is 3 s, while the maximum business interruption time is 40 s, satisfying the constraints for anomaly determination and degradation switching and recovery at the business layer. The proposed mechanism supports rapid degradation, business recovery, and stable switchback of cloud-edge collaborative toll collection systems under abnormal conditions.

Keywords

Cloud-Edge Collaboration, Multi-Level Disaster Recovery, State Synchronization, WebSocket, Finite-State Machine

Copyright © 2026 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

随着分布式计算与边缘计算技术的发展，云边协同架构逐渐应用于对实时性、连续性和本地容错能力要求较高的业务系统，依托云端统筹、边缘算力就近调度和终端实时交互，可满足低延迟交易与本地化容错等业务场景的运行需求。高速公路收费系统正由传统单站集中式架构向云、边、端多级协同体系演进，国产化适配、云收费改造与站级联网优化持续推动系统向分布式和轻量化方向升级[1]-[4]。随着架构升级，车道终端承担核心交易与数据上报业务，站级服务、边缘云与广场服务之间的交互耦合度显著提升，系统跨层级调用、数据同步与状态联动的复杂度也随之增加。

分布式架构提升了系统部署灵活性与通行处理效率，但同时引入了多级联动的可靠性风险。传统集中式收费系统的故障影响通常集中于单站或局部节点；而在云边协同架构下，链路扰动、服务失效、节点故障和数据同步异常等问题可能通过跨层级依赖关系扩大故障影响范围，进而引发交易异常、业务中断和数据不一致等问题，制约收费业务连续性与系统运行稳定性。因此，面向云边协同架构构建兼顾故障感知、状态判定、业务降级与服务回切的容灾机制，成为保障高速公路收费业务连续运行的重要研究内容。

现有研究主要围绕收费系统应用优化、分布式系统可靠性保障以及状态感知与控制建模三个方面。

在收费系统优化方面，现有研究聚焦联网改造、云架构部署、智慧站点建设与国产化适配，为系统部署、车道运行效率与平台兼容等提供了技术支撑[1]-[4]。在分布式系统可靠性方面，冗余部署、主备切换、故障检测和异常诊断等技术已形成较为成熟的方法体系，可为收费系统容灾机制设计提供参考[5]-[7]。在状态感知与控制建模方面，WebSocket 等长连接技术可用于多节点状态感知与实时反馈[8]，边缘节点数据同步、时间对齐和异常检测可支撑降级场景下的数据连续性[9]；分布式故障检测与有限状态机等方法，则可分别为服务可用性判定和业务状态流转提供建模基础[10]-[15]。

综上，现有研究已在云收费系统架构部署、国产化适配及基础可靠性保障等方面形成了一定基础，但针对云边协同收费业务连续性的多级容灾协同研究仍有进一步深化空间，主要体现在以下三个方面：一是研究视角方面，交通收费领域研究多聚焦系统正向建设与功能优化，对故障场景下状态辨识、业务降级和恢复回切等动态容错过程关注不足；二是场景适配方面，通用分布式容错技术虽然较为成熟，但尚未充分结合收费系统交易实时性、通行连续性和数据一致性等业务约束，场景适配性仍有提升空间；三是体系协同方面，现有研究多针对链路自愈、服务切换、数据同步等单一容错模块展开，未构建多级容灾协同体系，缺乏对不同容灾层级时间尺度、触发边界与适配场景的统一分析，对复杂故障场景下的精细化容灾管控支撑仍显不足。

基于此，本文面向高速公路云边协同收费系统，围绕故障感知、状态判定、业务降级与服务回切等关键问题开展研究。首先，分析云边协同收费系统中的链路、服务、节点和数据等故障模式；其次，构建多级容灾控制模型，分析不同容灾层级之间的时间尺度约束关系；最后，以 WebSocket 双连接状态探测为基础，建立业务层降级有限状态机，描述云站正常、疑似异常、降级判定、广场降级、恢复探测和人工确认回切等状态流转过程，并结合测试指标对所提出机制进行可行性验证。

2. 云边协同收费系统故障模式与总体架构

云边协同收费系统由车道终端、站级交易服务、边缘云平台、广场降级服务以及数据同步链路共同构成。与传统单站本地部署模式相比，该架构将部分站级业务上移至边缘云侧，同时保留车道端设备控制和广场侧降级能力，形成“云侧集中处理、边侧本地保障、端侧实时执行”的协同运行模式。系统能力得到增强的同时，故障来源也由单一设备故障扩展为链路、服务、节点和数据等多类型故障。从业务连续性角度看，云边协同收费系统的故障可分为四类，如表 1 所示。

Table 1. Typical fault modes of a cloud-edge collaborative toll collection system
表 1. 云边协同收费系统典型故障模式

故障类型	典型表现	主要影响	容灾目标
链路故障	车道－广场、广场－站级或站级－云端链路抖动、中断、时延升高	心跳异常、交易服务访问不稳定	链路层：保障通信连通性
服务故障	站级交易服务不可用、业务接口无响应、服务异常退出	车道收费业务无法由云站承载	业务层：保障业务连续性
节点故障	超融合节点、虚拟机或业务主机异常	云侧业务恢复时间变长	计算层：保障计算资源恢复能力
数据故障	降级期间流水滞留、补传失败、状态不同步	交易数据完整性和一致性受影响	数据层：保障数据一致性

为适应上述故障特征，本文构建云边端协同的多级容灾体系，如图 1 所示。该体系分为四个层级：链路层通过 ERPS、VRRP 等机制保障通信链路可用；业务层通过 WebSocket 双连接与状态机控制实现云

站交易服务向广场降级服务的降级切换；计算层通过边缘云平台、超融合集群和虚拟机 HA 实现节点恢复；数据层通过降级期间本地暂存和恢复后增量同步保障交易数据一致性。

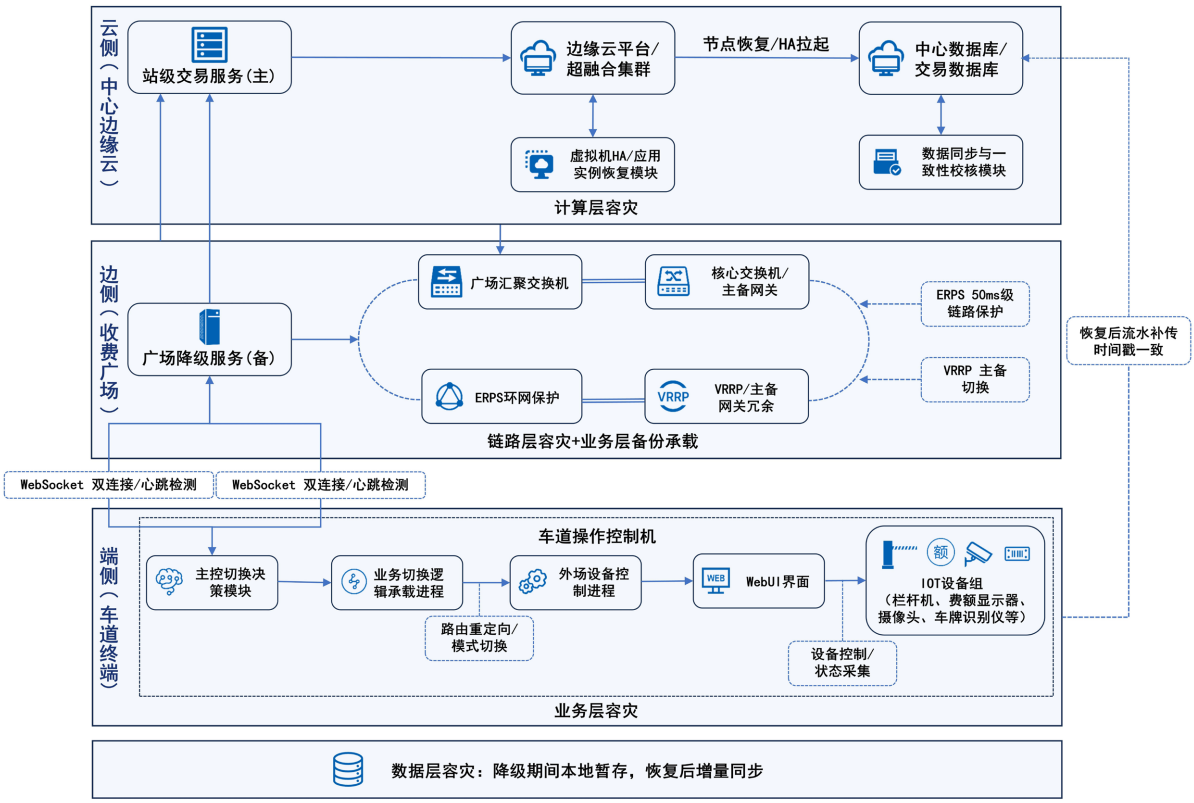


Figure 1. Overall architecture of multi-level disaster recovery and state synchronization in a cloud-edge collaborative toll collection system
图 1. 云边协同收费系统多级容灾与状态同步总体架构图

从总体架构来看，链路层负责吸收短时通信抖动；业务层负责在站级交易服务不可用系统才判定云站服务不可用，并触发降级切换。时实现收费业务的降级切换与连续接续；计算层负责云侧节点和应用实例恢复；数据层负责降级期间交易流水的补传与校核。四个层级并非同时触发，而是按照不同故障类型和时间尺度形成协同关系。

3. 多级容灾控制模型与时间尺度协调机制

为刻画云边协同收费系统的容灾行为，本文将系统容灾能力表示为：

$$R = \{R_\ell, R_b, R_c, R_d\} \tag{1}$$

式中：

R_ℓ ——链路层容灾能力，主要负责网络链路保护与通信恢复；

R_b ——业务层容灾能力，主要负责站级交易服务与广场降级服务之间的业务承载模式切换；

R_c ——计算层容灾能力，主要负责边缘云节点、虚拟机或业务实例恢复；

R_d ——数据层容灾能力，主要负责降级期间交易流水、状态记录和日志数据的补传校核。

对于一次故障恢复过程，其总体恢复时间可表示为：

$$T_{recover} = T_{detect} + T_{decision} + T_{switch} + T_{sync} \tag{2}$$

式中：

T_{detect} ——表示故障感知时间；

$T_{decision}$ ——表示故障判定与策略决策时间；

T_{switch} ——表示容灾切换执行时间；

T_{sync} ——表示恢复后的数据同步与状态校核时间。

不同容灾层级具有不同时间尺度，链路层强调快速收敛，业务层强调收费功能连续，计算层强调云侧资源恢复，数据层强调最终一致性。因此，多级容灾的关键不是各层级同时切换，而是形成有序的时间尺度协同。

为避免链路层短时抖动被业务层误判为站级交易服务失效，同时保证真实服务异常能够被及时识别，业务层异常判定时限应满足如下时间尺度约束：

$$T_{switch}^{link} < T_{loss} < T_{switch}^{business} \tag{3}$$

式中：

T_{switch}^{link} ——表示链路层保护收敛时间；

T_{loss} ——表示业务层完成云站异常判定与降级触发的最大允许时限；

$T_{switch}^{business}$ ——表示业务层降级切换与业务恢复的时间约束。

公式(3)表明，业务层异常判定时限应大于链路层保护收敛时间，使链路层短时抖动优先由网络自愈机制吸收；同时，该时限应小于业务层降级切换及恢复窗口，使真实服务异常能够在业务可接受时间内完成识别和处置。

一般来说，计算层恢复时间通常长于链路层和业务层切换时间。因此，当站级交易服务不可用时，业务层不应等待云侧节点完全恢复，而应先执行降级切换，将业务承载转移至广场降级服务；待云侧服务恢复并通过稳定性检测后，再执行回切，从而形成“链路层优先自愈、业务层连续接续、计算层渐进恢复、数据层最终校核”的分层协同控制逻辑。

4. 基于 WebSocket 的业务层降级状态机

业务层容灾控制依赖云、边、端多进程协同。为避免因具体部署差异影响模型普适性，如图 1 所示，本文将业务层控制过程分为三个功能对象：业务层切换逻辑承载进程负责站级交易服务与广场降级服务之间的连接状态维护、异常感知以及降级切换与回切控制；外场设备控制进程负责承接 IOT 设备连接并执行车道机电设备控制指令；主控切换决策模块位于车道操作控制机内，负责维护云站交易服务与广场降级服务的双向长连接，并下发路由变更指令。

传统 HTTP 轮询方式存在状态反馈不及时、请求开销较高等问题。本文采用 WebSocket 双连接探测机制，由主控切换决策模块分别与站级交易服务和广场降级服务建立持久连接，并按照固定周期发送心跳探测报文。设心跳探测周期为 Δt_{hb}，当服务端在规定时间内返回心跳应答时，认为该服务在当前探测周期内可用；当连续多个探测周期未收到应答时，则认为该服务进入异常状态。

为描述云站交易服务和广场降级服务的实时可用状态，定义服务可用性变量 $A_x(t)$：

$$A_x(t) = \begin{cases} 1, & x \in \{c, p\} \\ 0, & \end{cases} \tag{4}$$

式中：

$A_x(t)=1$——表示服务 x 在最近一次心跳探测周期 Δt_{hb} 内返回有效应答，系统判定该服务可用；

$A_x(t)=0$——表示服务 x 在连续 k 个心跳探测周期内均未返回有效应答，系统判定该服务不可用。其中：$x\in\{c,p\}$，c 表示云站交易服务，p 表示广场降级服务。$A_x(t)$ 为二值变量，用于表征 t 时刻服务 x 是否处于可用状态，由此可知，Δt_{hb} 决定单次服务状态探测的时间粒度，$k\Delta t_{hb}$ 则代表服务异常判定的时间阈值。

在单个服务可用性变量的基础上，进一步构造双服务可用性向量：

$$A(t)=\left[A_c(t),A_p(t)\right] \tag{5}$$

式中：

$A_c(t)$——表示 t 时刻云站交易服务的可用性；

$A_p(t)$——表示 t 时刻广场降级服务的可用性；

$A(t)$——描述系统外部服务环境状态，即“云站服务和广场降级服务是否可用”。

然而，服务可用并不等同于系统正在使用该服务。为描述车道业务当前由哪一类服务承载，定义系统工作模式变量：

$$m(t)\in\{C,P\} \tag{6}$$

其中：

$m(t)=C$——表示系统处于云站模式，车道业务由云站交易服务承载；

$m(t)=P$——表示系统处于广场降级模式，车道业务由广场降级服务承载。

需要注意的是，$A(t)$ 和 $m(t)$ 分别描述“服务可用状态”和“业务承载模式”，二者并不完全等价。例如，当 $A(t)=[1,1]$ 时，云站交易服务和广场降级服务均处于可用状态，但系统仍可保持 $m(t)=C$；只有当 $A(t)=0$ 且满足降级判定条件时，系统才会切换至 $m(t)=P$。

基于上述定义，系统可设置降级判定阈值和恢复稳定性观察窗口：

$$T_{loss}=k_1\Delta t_{hb},\quad T_{stable}=k_2\Delta t_{hb} \tag{7}$$

式中：

k_1——表示触发降级判定所需的连续异常心跳周期数；

k_2——表示云站服务恢复后需要持续稳定的心跳周期数，且要求 $k_2>k_1>1$。

据此，业务层有限状态机可根据 $A(t)$、$m(t)$、T_{loss} 和 T_{stable} 来描述云站正常、疑似异常、降级判定、广场降级、恢复探测和人工回切等状态流转过程，如图 2 所示。

图 2 给出了业务层降级状态机的整体流转逻辑。状态机以云站交易服务可用性 $A_c(t)$、广场降级服务可用性 $A_p(t)$ 和系统工作模式 $m(t)$ 为输入变量，将业务层容灾过程划分为正常运行、异常判定、广场降级、恢复探测和人工回切五个阶段。与单纯流程控制不同，该状态机并不在出现单次心跳异常时立即执行降级切换，而是先进入疑似异常状态，等待链路层短时自愈；只有当异常持续时间超过 T_{loss} 后，系统才判定云站服务不可用，并触发降级切换。这样可以避免短时链路抖动被误判为业务服务故障。

在降级运行阶段，车道业务由广场降级服务本地接续，主控切换决策模块仍持续探测云站服务状态。当重新收到云站心跳应答后，系统进入恢复探测阶段，并通过 T_{stable} 观察窗口判断云站服务是否持续稳定。若观察期间再次发生心跳异常，则状态回到广场降级运行态；若云站服务在 T_{stable} 内持续可用，则进入回切确认阶段。此时系统采用“人在回路”的非对称控制策略，由收费人员确认后再执行回切，最终恢复云站交易服务承载。该机制将“异常快速响应”和“恢复稳态确认”分离，有助于降低网络恢复初期反复切换造成的状态震荡风险。

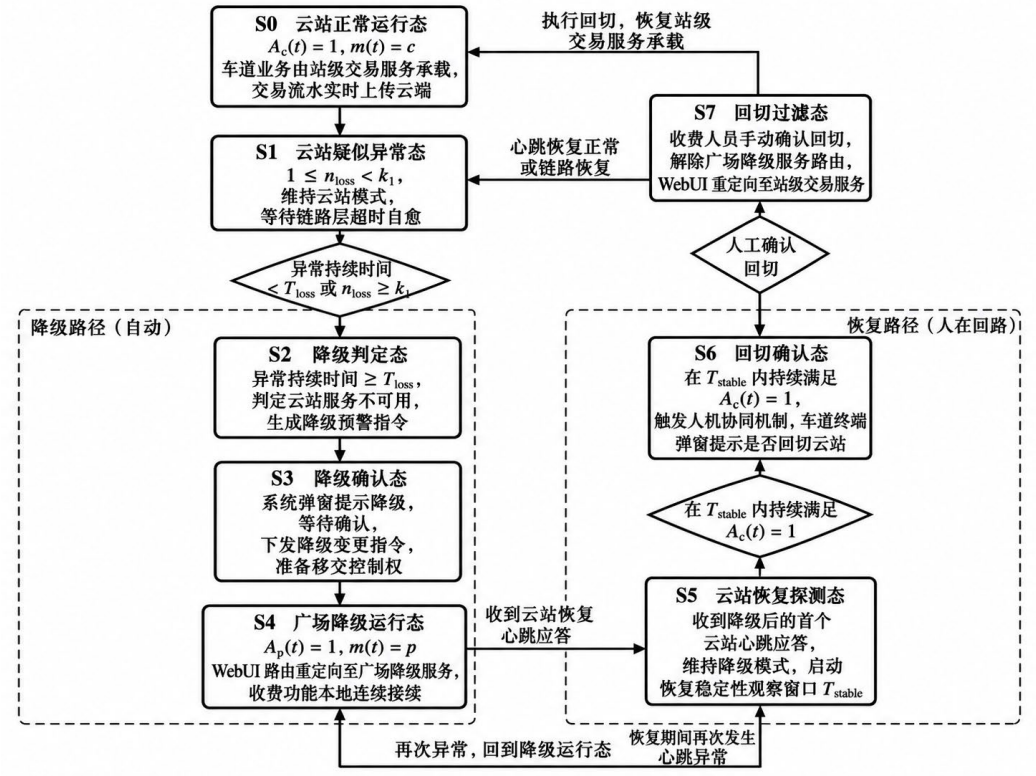


Figure 2. Business layer degradation state transition diagram
图 2. 业务层降级状态机流转图

5. 数据层容灾与状态同步机制

数据层容灾的目标是保障降级运行期间交易流水、设备状态和操作日志的完整性。在云站正常模式下，车道交易数据由站级交易服务统一处理并上传；系统进入广场降级模式后，相关数据由本地服务暂存；待云站服务稳定恢复并完成回切后，再执行增量补传和一致性校核。

5.1. 降级数据暂存与增量同步

设 t_0 为系统进入广场降级模式的时刻，t_r 为云站服务恢复并完成回切的时刻，则降级期间本地新增数据可表示为：

$$\Delta D_p\left(t_0,t_r\right)=\left\{d_i \mid t_0 \le t_i \le t_r\right\} \tag{8}$$

式中：

d_i ——表示降级期间产生的第 i 条交易流水、状态记录或日志数据。

系统恢复后仅对该时间窗口内新增数据进行补传，以减少重复传输，并为后续一致性校核提供数据基础。

5.2. 数据冲突检测与一致性校核

云站恢复后的中心侧数据状态可表示为：

$$D_c\left(t_r^+\right)=D_c\left(t_0\right)\oplus\Delta D_p\left(t_0,t_r\right) \tag{9}$$

其中，$\oplus$ 表示带有去重、排序和状态校核的数据合并操作。为避免重复补传或顺序错乱，以车道编号、

交易流水号作为数据唯一标识，并结合时间戳、交易状态和工作模式标识等关键字段进行一致性校核。若中心侧无对应记录，则判定为缺失数据，并按照原始流水产生顺序执行补传；若记录已存在且关键字段一致，则判定为重复数据，采用幂等方式进行过滤；若唯一标识相同但关键字段存在差异，则判定为冲突数据，结合交易状态演进关系和时间戳进行比对，对能够依据业务状态演进关系确定的记录保留有效业务终态，对无法自动判定的异常记录保留原始信息并进入人工对账流程，避免直接覆盖造成数据丢失。

通过上述“唯一标识匹配–关键字段校核–分类处理”机制，实现恢复同步过程中的数据冲突检测与一致性校核，并在业务恢复后逐步达到最终一致状态。

5.3. 分批补传与流量控制

降级持续时间不同会导致待补传数据规模存在差异。为避免云站服务恢复后大量历史数据集中回传，占用网络带宽和服务处理资源，本文采用分批补传方式，将待同步数据按照产生顺序划分为多个批次，并在上一批数据完成传输并确认后继续执行下一批补传。当补传过程中出现网络抖动、应答超时或传输失败时，暂缓后续数据发送，待链路恢复稳定后继续补传。同时，补传过程优先保障实时收费业务，降低历史数据同步对正常业务运行的影响。

随着待补传数据量增加，所需补传批次数和整体同步时间相应增加。通过限制单批传输规模，可避免大规模数据集中回传造成瞬时网络负载升高，使补传过程在数据恢复效率与在线业务连续性之间保持协调。待全部数据补传完成后，再执行一致性校核，完成降级期间数据的恢复同步。

需要强调的是，数据同步不参与业务层实时切换判定。业务层降级的首要目标是保障收费功能连续，数据层则负责恢复后的最终一致性。因此，数据补偿应在云站服务稳定恢复后执行，避免网络仍处于抖动状态时进行大规模回传，从而降低业务回切阶段的数据冲突风险，并为后续稽核、对账和运行追溯提供数据基础。

6. 典型容灾机制对比分析

前文围绕链路、业务、计算和数据等层级构建了多级容灾与状态同步机制。为进一步明确其在故障处置范围和恢复方式上的特点，本文选取双机热备和基于负载均衡的故障转移两类典型机制进行对比。三类容灾方式在处理对象、业务接续、恢复过程及数据同步等方面的主要特征如表 2 所示。

Table 2. Comparison of typical disaster recovery mechanisms
表 2. 典型容灾机制功能特征对比

对比维度	双机热备	基于负载均衡的故障转移	本文多级容灾机制
处理对象	主备节点	服务实例	链路、业务、计算和数据
故障检测方式	主备状态检测	服务健康检测	服务状态探测与分层判定
业务接续方式	备用节点接管	请求转移至可用实例	广场降级服务接续
恢复过程	按主备策略恢复	恢复实例重新加入服务	分层恢复与稳态回切
数据处理	根据部署方式配置数据同步	通常需配套数据同步机制	本地暂存、 增量补传与一致性校核
协同范围	侧重节点级故障处理	侧重服务级故障处理	不同容灾层级间协同处理

由表 2 可知，三类容灾机制在处理对象和适用范围上存在明显差异。双机热备主要解决主备节点之

间的故障接管问题，基于负载均衡的故障转移侧重服务实例可用性的维持，两者均可作为特定层级的容灾手段。本文机制并非对上述单一容灾方式的简单替代，而是针对云边协同收费系统中不同故障类型和恢复阶段，对各层级容灾手段进行统一组织与协调。因此，其主要特点体现在容灾范围由单一节点或服务扩展至多层级协同，并同时考虑业务接续、恢复控制和数据一致性等要求。

7. 工程验证与指标分析

为验证本文所提出的多级容灾控制模型和业务层状态机的有效性，基于 156 次实际测试记录，对主备切换触发、业务恢复以及原主节点恢复上线过程进行统计分析。

第 3 章中公式(2)给出了故障恢复时间的一般表达，在本文验证中，重点考察业务层容灾恢复效果，因此将业务层可见恢复时间记为 $T_{recover}^{business}$。该指标对应实测记录中的“业务中断时间”，表示故障发生后系统完成容灾切换并恢复正常业务承载所需的时间。数据层流水补传、对账和后续稽核校核属于恢复后的数据一致性过程，不纳入本节 $T_{recover}^{business}$ 的统计范围。

考虑到不同容灾方案在部署方式和运行条件上存在差异，本文不对其性能指标作直接量化比较，而是结合第 6 章的功能特征分析，从故障处理范围和恢复机制方面进行对比。本节主要针对所提机制的实际运行表现进行验证。

7.1. 状态机关键参数设置

根据云边协同收费系统业务连续性要求和业务层状态机控制目标，本文设定关键参数如表 3 所示。

Table 3. Key parameter settings of the state machine
表 3. 状态机关键参数设定表

参数	含义	设定值	控制作用
Δt_{hb}	心跳探测时间粒度	1 s	描述 WebSocket 心跳探测的基本时间粒度
k_1	降级判定连续异常周期数	3 个	判定云站交易服务是否持续不可用
T_{loss}	业务层异常判定最大时限	3 s	云站服务连续异常达到该阈值后触发降级判定
k_2	恢复稳态观察周期数	45 个	判断云站服务恢复后的持续稳定性
T_{stable}	恢复稳态观察窗口	45 s	云站服务持续稳定达到该窗口后进入人工回切确认
T_{switch}^{link}	链路层保护收敛时间	0.05 s	吸收链路层短时通信抖动
$T_{switch}^{business}$	业务层降级切换及恢复窗口约束	45 s	业务层完成降级切换及业务恢复的最大时间

7.2. 实测结果与验证分析

为验证所提模型在实站场景下的适用性，本文以某收费站站级交易服务主节点异常为典型故障场景开展测试。测试过程中，分别构造主节点网络中断和主节点上电恢复两类场景，观察运行监测平台主备状态切换、日志触发主备切换动作、收费业务恢复以及原主节点重新上线情况。其中，运行监测平台用于展示主备节点状态和业务恢复状态。测试共进行 156 次，为突出模型验证结果，本文选取各指标的平均值和最大值进行统计，结果如表 4 所示。

从主备切换触发结果看，运行监测平台界面切换完成时间最大值为 2 s，日志实际触发主备切换动作时间最大值为 3 s。其中，主备切换动作触发时间未超过业务层异常判定与降级触发最大允许时限 $T_{loss}=3$ s。结果表明，系统能够在设定时限内完成异常感知、主备状态切换和降级触发，具备秒级异常响应能力。

Table 4. Statistical results of multi-level disaster recovery tests
表 4. 多级容灾实测数据统计表

测试指标	观测量符号	平均值/s	最大值/s	验证作用
主节点异常至运行监测平台界面切换完成时间	T_{ui}	1.25	2	表征界面层主备状态切换响应能力
主节点异常至日志触发主备切换动作时间	$T_{trigger}$	1.30	3	表征异常感知与主备切换触发响应能力
业务中断时间	$T_{recover}^{business}$	30.5	40	表征业务层容灾恢复时间
原主节点恢复上线时间(恢复网络)	T_{online}^{net}	1.7	2	表征网络恢复场景下原主节点重新上线能力
原主节点恢复上线时间(恢复上电)	T_{online}^{power}	36.1	38	表征上电恢复场景下原主节点重新上线能力

从业务恢复结果看，业务中断时间平均值为 30.5 s，最大值为 40 s，小于业务层降级切换及恢复窗口约束 $T_{switch}^{business} = 45$ s。这说明系统能够在设定业务恢复窗口内恢复正常业务承载，满足收费业务连续运行要求。

从原主节点恢复结果看，恢复网络场景下原主节点上线时间最大值为 2 s，恢复上电场景下最大值为 38 s。该指标可作为计算层恢复能力的外部观测量，反映节点在网络或电源故障解除后，完成系统恢复、服务启动、心跳上报和平台状态注册的能力。测试结果说明，计算层能够在典型恢复场景下重新上线，为后续云站服务探测和回切控制提供支撑。

综上，实测结果表明：主备切换触发时间、业务层可见恢复时间均满足预设约束，计算层能够在典型网络恢复和上电恢复场景下完成重新上线。上述结果验证了本文多级容灾控制模型对异常感知、业务降级、业务恢复和计算层恢复等关键过程的覆盖能力，说明基于 WebSocket 的业务层状态机具有可行性与有效性。

8. 结论

本文面向高速公路云边协同收费系统，针对链路抖动、服务不可用、节点异常和数据不同步等问题，构建了多级容灾与状态同步机制，主要结论如下：

(1) 提出了链路层、业务层、计算层和数据层协同的多级容灾模型，明确了各层级在故障处理中的功能边界，实现通信自愈、业务接续、节点恢复和数据一致性保障的分层协同。

(2) 建立了多级容灾时间尺度约束关系，将链路层保护收敛时间、业务层异常判定时限和业务层降级切换及恢复窗口进行统一描述，可降低链路短时抖动导致的误降级风险。

(3) 构建了基于 WebSocket 双连接心跳探测的业务层降级有限状态机，实现云站正常、疑似异常、降级判定、广场降级、恢复探测和人工确认回切等状态流转，并通过“异常快速响应、恢复稳态确认”策略降低反复切换风险。

(4) 实际测试结果表明，主备切换动作最大触发时间为 3 s，业务中断时间最大值为 40 s，均满足模型设定的业务层控制约束；原主节点在恢复网络和恢复上电场景下均能完成重新上线。结果说明，所提机制具备工程可行性与有效性。

后续研究可进一步面向多收费站协同运行场景，完善降级期间交易流水补传、跨站数据一致性校核和回切阈值自适应优化方法。

基金项目

本文受 2025 年河北省省级重大科技支撑计划项目《全量鸿蒙高速公路智慧化应用场景》资助，项目

编号：252Q0801D。

参考文献

[1] 王志敏, 赖树坤. 高速公路联网收费系统国产化适配与平台兼容性验证研究[J]. 中国交通信息化, 2025(11): 67-71.

[2] 张天成. 四川省高速公路收费站级联网收费系统优化[J]. 交通科技与管理, 2023, 4(14): 17-19.

[3] 邵俊凯. 高速公路云收费系统方案[J]. 智能建筑与智慧城市, 2023(10): 178-180.

[4] 李成. 智慧高速公路云收费系统的应用[J]. 交通世界, 2025(24): 1-3.

[5] 赵建涛, 黄立松. 微服务故障诊断相关技术研究探讨[J]. 网络新媒体技术, 2020, 9(1): 1-8.

[6] 黄朝辉. 分布式系统故障检测与异常检测技术研究[D]: [硕士学位论文]. 广州: 华南理工大学, 2021.

[7] 孙燕. 面向冗余的分布式系统可靠性的设计与实现[D]: [硕士学位论文]. 保定: 河北大学, 2003.

[8] 李帅. 基于 WebSocket 的多 Agent 通信机制设计与应用[D]: [硕士学位论文]. 大连: 大连海事大学, 2019.

[9] 高志翔. 智能安全监控系统中边缘节点数据同步与异常检测算法研究[J]. 软件, 2026, 47(1): 108-110.

[10] Pimentel, V. and Nickerson, B.G. (2012) Communicating and Displaying Real-Time Data with Websocket. *IEEE Internet Computing*, **16**, 45-53. https://doi.org/10.1109/mic.2012.64

[11] Skvorc, D., Horvat, M. and Srbljic, S. (2014) Performance Evaluation of Websocket Protocol for Implementation of Full-Duplex Web Streams. 2014 37*th International Convention on Information and Communication Technology*, *Electronics and Microelectronics* (*MIPRO*), Opatija, 26-30 May 2014, 1003-1008. https://doi.org/10.1109/mipro.2014.6859715

[12] Rahmatulloh, A., Darmawan, I. and Gunawan, R. (2019) Performance Analysis of Data Transmission on Websocket for Real-Time Communication. 2019 16*th International Conference on Quality in Research*: *International Symposium on Electrical and Computer Engineering*, Padang, 22-24 July 2019, 1-5. https://doi.org/10.1109/qir.2019.8898135

[13] Chandra, T.D. and Toueg, S. (1996) Unreliable Failure Detectors for Reliable Distributed Systems. *Journal of the ACM*, **43**, 225-267. https://doi.org/10.1145/226643.226647

[14] Harel, D. (1987) Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, **8**, 231-274. https://doi.org/10.1016/0167-6423(87)90035-9

[15] Hassan, A.B. and Mansour, M. (2024) Performance Evaluation of Bidirectional Forwarding Detection over the Virtual Router Redundancy Protocol. *Procedia Computer Science*, **251**, 256-264. https://doi.org/10.1016/j.procs.2024.11.108