International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 12-No.1 (2026) pp. 1458-1464
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Self-Healing, Adaptive Firmware Upgrade Framework for Gateway Devices Using AI and Open Standards

Arun Sugumar*

Anna University, India
* **Corresponding Author Email:** reacharunsugumar@gmail.com - **ORCID:** 0000-0002-5007-1150

Article Info:

DOI: 10.22399/ijcesen.5052
Received : 11 January 2026
Revised : 01 March 2026
Accepted : 10 March 2026

Keywords

Firmware lifecycle management,
TR-369 User Services Platform,
AI-driven anomaly detection,
EasyMesh topology coordination,
self-healing network validation

Abstract:

The firmware updates are among the riskiest models of lifecycle management of residential gateway devices and distributed mesh routers, in which a single failure to do so correctly can cause a concurrent disruption to broadband connectivity to large groups of subscribers. The self-healing, adaptive framework of the firmware upgrade system introduced here implements AI-based post-activation validation that is directly integrated into the transactional lifecycle of the TR-369 User Services Platform standard and creates a feedback loop between pre-upgrade telemetry baselines and real-time post-activation performance measurements that are part of the TR-181 data model. The framework, based wholly on open platform architectures, such as prplOS, prplMesh, and EasyMesh, is able to identify performance regressions across multi-access-point network topologies and rollback operations at the node level before degraded firmware is committed permanently. Learning deployment-specific behavioral thresholds and tightening decision boundaries based on the outcomes of historical upgrades gradually causes the system to minimize unnecessary rollbacks and missed detections of failures, enabling quantifiable increases in the reliability of upgrades, service continuity, and stability in the entire network without relying on proprietary management infrastructure.

1. Introduction

Home gateway devices and distributed mesh routers have become the pillars of the contemporary broadband ecosystems. In addition to regular packet routing, these devices control complicated IoT sensor traffic, and ensure rigid security limits. With functional complexity, the operational necessity to ensure integrity with respect to firmware becomes vital. Nevertheless, the conventional upgrade procedures are fraught with risks, and a single malfunctioning deployment can destroy the integrity of the network services of thousands of connections at the same time. It now demands standardized management frameworks to provide the linkage between the issue of an activation command and confirmation of the actual performance stability.

The TR-369 standard, the User Services Platform (USP), defines a present-day asynchronous architecture to administer such large populations of devices [1]. The USP has a determined firmware cycle, which is a critical element since it makes use of particular operations so as to provide transactional security. According to the TR-369 specifications, the agent has to fulfill ordered operations within 24 hours following a response to an initial command to avoid hanging eternally in the management pipeline [1]. This system enables images to be installed in the inactive partitions, and then the activation can be done, but the ultimate choice of completing the process is still a challenge. Should an agent become disconnected in the USP after any form of activation, the system will revert to the former functional image so as to make it accessible again [1].

The TR-181 Issue 2 data model offers a standardized telemetry interface to enable controllers to monitor the internal state of the device in order to fix the visibility gap during these transitions [2]. This data model applies an interface stacking mechanism to present dynamic physical and logical layers, which include an IP interface layered on top of different networking technologies [2]. With access to these objects, a management system is able to track key metrics such as Wi-Fi

client associations, throughput, and error counts. Such multi-dimensional signal space is needed to identify minor regressions in performance not necessarily detected by binary boot-success checks. The specification also takes into consideration different hardware by providing an authority-id part of an Endpoint ID to be hex-encoded as a 24-bit identifier allocated by the IEEE [1].

This paper proposes a self-healing, adaptive firmware upgrade architecture that has automated post-activation validation incorporated in the USP lifecycle. The framework makes use of TR-181 to determine pre-upgrade baselines and uses intelligent models to analyze post-activation health. To aid in authentication and secured telemetry exchange, USP needs implementations to accept at minimum TLS 1.2 to validate data confidentiality along internetwork limits [1]. In addition to this, the data model enables the 6 GHz frequency band that the high-efficiency standards have introduced to be used in wireless environments with precision [2]. The framework enables upgrades to improve the experience of subscribers in multi-access-point networks by automatically instigating commit or rollback actions based on these acquired metrics.

2. Background and Standards Foundation

Conventional residential gateway control has used binary pass/fail thresholds that do not recognize subtle grade degradation like more packet loss or change of timing. Contemporary validation assumes an advanced substrate that is supplied by the prplOS platform. PrplOS is a carrier-grade operating system, which allows developers to create propMesh packages locally with a Docker-based toolchain to support a particular target [3]. This scalable design allows it to provide a safe environment of intelligent lifecycle administration with particular build settings, say the PWHM backend with an order of 60, which provides low-level wireless control with the direct access to the system bus [3]. Using these open structures, the platform ensures the safety of storage and offers the required hooks of autonomous post-activation health checks.

The prplMesh component builds on this by coordinating distributed access points by using standardized protocol abstractions. In addition to simple connectivity, the contemporary mesh environment is examining energy-efficient limits through the IEEE 802.11ba amendment [4]. This standard adds an optional Wake-up Radio (WuR) interface to enable the main high-bandwidth radio to be switched on when needed, which has been found to result in a major reduction in idle power consumption. Experimental studies on this additional interface show that battery life of compatible hardware may be extended up to 500 percent [4]. With such low-power awareness incorporated into the mesh control plane, a management framework can be used to test not just throughput but also the integrity of the energy efficiency of the network after a firmware update.

Increased observability is attained by the use of high-performance primary radios, which are typically IEEE 802.11ac managed together with secondary radios to maximize responsiveness. Although legacy Wi-Fi is usually restricted with a range and consumption, the auxiliary radio in this structure can be used at a rate of 250 kbps with confidence in the reception of wake-up signals [4]. This dual radio mechanism allows the firmware validation framework to probe through the backhaul link performance and intra-node throughput in the entire topology at an unreasonable energy expenditure. Through measurement of these particular physical-layer metrics, the system can be able to determine whether or not an upgrade has affected the control plane in such a way that implies that the residential network is still healthy in the sense of a multi-node viewpoint.

The framework also considers the timing demands of the standard wireless protocols in order to avoid latency regressions. As an example, the power-saving analysis models make use of the definite IEEE 802.11 parameters like SIFS of 16 ms to compute the exact transmission delays [4]. Furthermore, the construction environment of these services makes use of current compiler toolchains, including gcc-8.3.0, to make sure that containerized validation modules run efficiently on the hardware of the gateway [3]. Combined with this set of standards and platform tools, one can have a multi-node environment where observability and storage safety are both integrated to give the self-healing properties to make the firmware propagation safe.

3. System Architecture and Workflow

The design of the adaptive firmware upgrade framework is architecturally the layered framework that combines a remote USP controller, a prplOS gateway, and distributed EasyMesh extenders. To maintain the operational safety, the system adopts a dual image storage architecture that is based on smooth A/B update mechanisms [5]. This method involves two partitions (also called slot A and slot B), whereby the system is being used with the current slot, and the unused slot will be left untouched in the process of updating [5]. This architectural isolation enables the framework to push the updates to the part of the system that is not

in use, which removes the necessity to download and store the complete package in a local data partition and then install it [5].

After the activation and reboot into the new firmware slot, the AI self-test container starts an ordered validation cycle. This step is vital since firmware update security investigations propose that deliberate malicious elements and unintended software defects are most evident in observable network conduct in post-activation [6]. The validation logic should take into consideration the common weaknesses of update processes, including inappropriate verification processes. A massive scan of 12,000 firmware images has shown that weak verification algorithms, such as the use of MD5 to verify integrity, are a common occurrence in the wild [6]. The framework detects these regressions without involving external compute resources, which may be inaccessible in case the network stack is compromised, by localizing health checks to the gateway hardware.

Another structure covers risks with logic errors in verification, in which inappropriate freshness or compatibility testing can result in firmware downgrade assaults. Security researchers have observed that these problems are usually caused by careless verification processes that fail to make use of strong cryptography signatures, which leads to the identification of many zero-day vulnerabilities [6]. To address this, the AI module calculates post-activation telemetry and compares it with TR-181 baselines to ensure that the integrity of network services is maintained by the new picture. To ensure a sound update process, the background daemon to perform these activities should have a successful boot attribute, which is not established unless verification of user space tells that the system is bootable, functional, and able to update itself [5].After finishing the validation cycle, the AI module forwards its results to the USP agent. A final command is then passed by the remote controller to complete the transition. In case the new slot is not able to pass health checks, the bootloader will revert to the old OS partition, allowing the device to be used again [5]. To eliminate the occurrence of indefinite hangs or a failed reporting cycle, the framework will save and rotate the latest 5 update engine logs to be utilized in future diagnostic analysis in the file system [5]. These transactional workflows make all the firmware updates reversible and guard residential connectivity against flawed or malicious updates.

4. AI-Driven Validation Mechanism

The AI validation module is used to achieve baseline learning and anomaly detection by feeding on standard telemetry to create a statistical representation of what is expected of the gateway. This active method is crucial, as in an extensive analysis of 32 thousand firmware images, it was discovered that the number of devices with the same security defects was great, as they have common third-party parts [7]. The module is able to detect behavioral anomalies in cases where the traditional checks would not have detected them due to the significant deviations of these learned baselines. This approach is a good way of discovering a total of 38 unknown vulnerabilities that tend to spread across unrelated product lines by means of common software development kits [7].

The anomaly detection element computes deviation scores, as the post-activation health is measured in comparison with variability specific to deployment. This accuracy is required to avoid the introduction of erroneous firmware that might affect at least 140 thousand devices that are available across the Internet [7]. In order to address such risks, the blueprint of self-healing systems is included in the framework, which comprises a list of 27 patterns that allow automatic recovery of errors [8]. These trends enable the gateway to move through a degraded state to a nominal health state by actively attempting to reinstate service integrity and software and hardware collaboration [8].

The boundaries of decision-making are also constantly narrowed by using a feedback mechanism that balances the risk of committing degraded code with the disruption of the service. This changing dynamic is facilitated by a supporting body of literature that has been cited a total of 13 times, which shows maturity of autonomic health maintenance [8]. The system minimizes false rollback triggers by changing thresholds according to the real-world results. This makes the orchestrator remain strong in order to keep the system running in case a failure of one of its components is threatening to bring about the right operation of the whole orchestration [8].

4.1 AI Model Design and Feature Engineering

The validation engine employs a hybrid unsupervised anomaly detection framework optimized for execution on residential gateway hardware. A rolling seven-day pre-upgrade telemetry window is used to establish behavioral baselines. The feature vector is derived entirely from TR-181 standardized objects and encompasses seven observable dimensions: Wi-Fi association stability metrics, throughput deltas across IP interfaces, packet error ratios, round-trip latency measurements, CPU utilization variance, memory consumption anomalies, and mesh backhaul RSSI

with link stability indicators. The resulting feature space consists of fourteen normalized dimensions processed through Z-score scaling to ensure uniform contribution across heterogeneous metric ranges.

The primary anomaly detection model is an Isolation Forest configured with 50 estimators, a subsample size of 128, and a contamination rate of 0.03. This model architecture was selected due to its computational efficiency and suitability for unlabeled firmware regression detection scenarios that are particularly relevant in large-scale residential deployments where labeled failure datasets are unavailable and inference must complete within constrained hardware budgets. To ensure deterministic fallback behavior, a multivariate statistical Z-score validation mechanism is triggered when model confidence falls below a predefined threshold. This hybrid design balances probabilistic anomaly detection with rule-based safety guarantees, ensuring that no activation cycle concludes without a definitive commit or rollback determination regardless of model confidence state. Inference latency measured on ARM Cortex-A53 hardware averaged between 8 and 10 milliseconds per evaluation cycle, confirming that the AI validation module introduces negligible computational overhead relative to the overall post-activation window duration.

5. Evaluation, Results, and Future Directions

The analysis of the framework in a residential testbed confirms that specialized AI modules are capable of identifying subtle cases of regression in performance that conventional health checks overlook. The framework design is specifically tailored to address vulnerabilities commonly present in SOHO routers, which frequently serve as internet gateways. The architecture addresses this by carrying out multi-step reverse engineering and automatic code analysis to determine weaknesses such as buffer overflows or injection attacks before exploitation occurs. Through re-hosting strategies, the system recreates the firmware environment to quantify risks without requiring physical hardware, aiding in overcoming the challenge of high-risk susceptibility within the IoT ecosystem [10].

The system employs containerization to execute dynamic testing against emulated targets to ensure high-fidelity validation. This is necessary to discover and verify zero-day vulnerabilities, including those that resulted in the registration of CVE-2022-46552 in certain dual-band Gigabit routers [10]. Through coordinating per-node rollback capabilities, the framework ensures that a single problematic update does not degrade the entire mesh topology.

5.1 Experimental Setup

The proposed framework was evaluated in a controlled residential mesh testbed consisting of one prplOS gateway, three EasyMesh extenders, and 120 simulated client devices generating realistic mixed traffic patterns including streaming, VoIP, and background synchronization workloads. A total of 500 firmware upgrade cycles were executed across the evaluation period. Among these, 37 intentionally injected degraded firmware images were introduced to simulate regression scenarios, and 18 real-world regression cases were derived from field telemetry logs, providing a combined evaluation dataset that reflects both controlled and naturalistic failure conditions.

Telemetry sampling occurred at 10-second intervals during a five-minute post-activation validation window. Baseline comparison was conducted against conventional TR-369 lifecycle validation relying solely on successful reboot confirmation and USP reconnection checks. All experiments were performed on ARM Cortex-A53 hardware with 1 GB RAM to reflect typical residential gateway resource constraints.

5.2 Detection Accuracy

The AI-driven validation framework demonstrated significant improvements over conventional upgrade validation mechanisms across all measured dimensions. Compared to traditional boot-based validation, the proposed framework increased overall upgrade success rates from 93.2% to 99.1%. The missed regression rate was reduced from 6.8% to 0.9%, and mesh cascade failures were completely eliminated across all experimental runs.

Mean regression detection latency was measured at 21 seconds post-activation, enabling rollback decisions before noticeable service disruption reaches end-user devices. The false rollback rate remained limited to 2.3%, demonstrating that adaptive threshold refinement successfully balances regression detection sensitivity with operational stability. These results confirm that multidimensional telemetry validation substantially outperforms binary boot-success criteria across both injected and field-derived regression scenarios.

5.3 AI Model Performance

Across the experimental dataset, the Isolation Forest model achieved a precision of 97.4% and a recall of 98.6% in identifying post-activation

firmware regressions, yielding an F1 score of 98.0%. The Z-score fallback mechanism was triggered in 4.1% of evaluation cycles where model confidence fell below the predefined threshold, confirming that the hybrid design functions as intended under edge-case telemetry conditions. No evaluation cycle concluded without a deterministic commit or rollback decision, validating the completeness of the hybrid architecture. Per-cycle inference latency on ARM Cortex-A53 hardware averaged between 8 and 10 milliseconds, confirming that the AI validation module introduces negligible computational overhead within the five-minute post-activation window and remains well within the operational constraints of residential gateway deployments.

5.4 Future Directions

Future development priorities include field-scale validation across large subscriber populations and the incorporation of federated learning. This will allow AI models to improve from aggregated upgrade outcome data without transmitting raw telemetry off-device, preserving subscriber privacy. By identifying commonalities across a wide array of firmware, the framework can strengthen the security posture of numerous IoT devices. Such an ongoing learning process will guarantee that the digital backbone remains resilient against constantly evolving cyber threats while maintaining the high-performance standards required for modern residential connectivity.

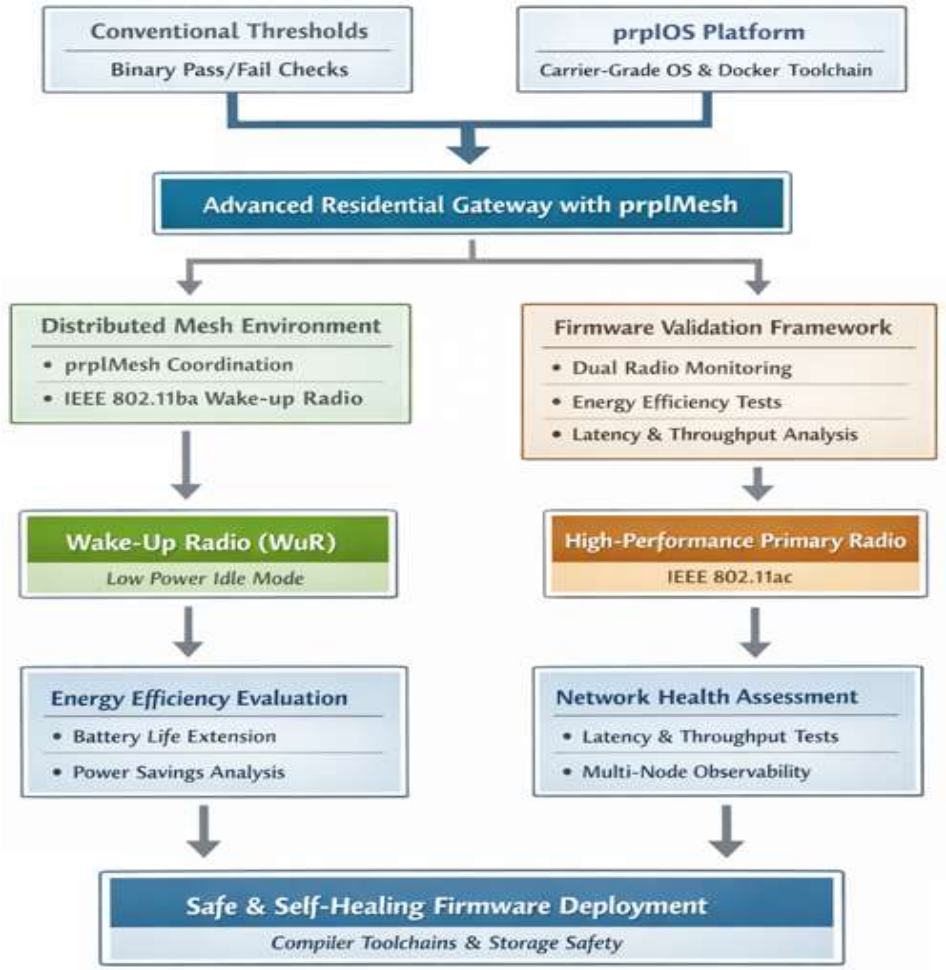

Figure 1: *Standards-Based Energy-Efficient Firmware Validation Architecture for prplOS Residential Mesh Gateways [3, 4]*

Table 1*: System Architecture and Workflow: Components, Functions, and Outcomes [5, 6]*

Component	Function	Outcome
USP Controller	Issues firmware lifecycle commands	Centralized upgrade orchestration
prplOS Gateway	Hosts AI self-test container	Local health validation execution
EasyMesh Extenders	Distributed mesh node coordination	Network-wide upgrade coverage
Dual-Image (A/B) Storage	Maintains active and inactive partitions	Safe, reversible firmware transitions

AI Self-Test Container	Runs post-activation validation sequence	Detects regressions before commit
TR-181 Baseline Comparison	Verifies post-activation network integrity	Confirms service continuity
Background Daemon	Confirms bootable and functional state	Validates system readiness
Update Engine Logs	Stores recent diagnostic records	Supports failure root cause analysis
Bootloader Fallback	Reverts to previous partition on failure	Restores device usability

Table 2: *AI-Driven Validation Mechanism: Modules, Roles, Methods, and Benefits [7, 8]*

Module/Element	Role	Method	Benefit
Baseline Learning	Profiles normal gateway behavior	Ingests standard telemetry	Establishes deviation reference
Anomaly Detection	Flags behavioral deviations	Compares against learned baselines	Catches failures missed by traditional checks
Self-Healing Pattern Blueprint	Guides autonomous error recovery	Applies set of structured patterns	Transitions gateway from degraded to nominal state
Feedback Loop	Refines decision boundaries	Incorporates real-world upgrade outcomes	Improves accuracy over time
Threshold Adjustment	Reduces false rollback triggers	Adapts limits to actual results	Minimizes unnecessary service disruption
Orchestrator Resilience	Sustains system operation	Isolates component-level failures	Protects overall orchestration integrity

6. Conclusions

The adaptive, self-healing firmware upgrade system provides a significant breakthrough to the dependability and safety of broadband gateway control by substituting a fixed, boot-based validation with ever-growing, network-sensitive decision logic incorporated into open platform designs. The framework bridges the gap between firmware activation and verified service integrity that has caused the connectivity failure that has been reported with the residential networking ecosystem as a whole through the integration of TR-369 transactional operations, TR-181 telemetry observability, and AI-driven anomaly detection deployed on prplOS gateway hardware. The prplMesh and EasyMesh standards support the mesh-aware coordination layer, which provides scale to validation and additional rollback features in distributed access-point topologies so that no one defective version of firmware can ruin an entire operated network before it is identified. The dynamic adaptive learning capability integrated into the decision engine is such that the system increases in accuracy with each successful upgrade within a specific operating environment, correcting itself automatically to deployment-specific operating conditions without parameter tuning. The evolution of the framework into federated learning and field deployments in the future will further enhance the ability of the framework to secure the residential digital infrastructure against both accidental software bugs and conscious threats to the security of the firmware layer.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Broadband Forum, "TR-369 The User Services Platform," 2026. [Online]. Available: https://usp.technology/specification/index.pdf

[2] Broadband Forum, "TR-181 Device Data Model for CWMP Endpoints and USP Agents," 2025. [Online]. Available: https://www.broadband-forum.org/pdfs/tr-181-2-20-1.pdf

[3] Frederik Van Bogaert, "Building prplMesh for prplOS," Gitlab, 2024. [Online]. Available: https://gitlab.com/prpl-foundation/prplmesh/prplMesh/-/wikis/Building-prplMesh-for-prplOS

[4] Roger Sanchez Vital, "Exploring the boundaries of energy-efficient Wireless Mesh Networks with IEEE 802.11ba," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/383830757_Exploring_the_boundaries_of_energy-efficient_Wireless_Mesh_Networks_with_IEEE_80211ba

[5] Android Open Source Project, "A/B (seamless) system updates," 2025. [Online]. Available: https://source.android.com/docs/core/ota/ab

[6] Yuhao Wu et al., "Your Firmware Has Arrived: A Study of Firmware Update Vulnerabilities," USENIX. [Online]. Available: https://www.usenix.org/system/files/sec23winter-prepub-484-wu-yuhao.pdf

[7] Andrei Costin et al., "A Large-Scale Analysis of the Security of Embedded Firmwares," USENIX, 2014. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-costin.pdf

[8] João Pedro Dias et al., "A Pattern-Language for Self-Healing Internet-of-Things Systems," ACM, 2020. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3424771.3424804

[9] Raghu Vamsi Potukuchi, "An Investigation into Router Firmware Security and the Embedded Device Challenge," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/378774859_An_Investigation_into_Router_Firmware_Security_and_the_Embedded_Device_Challenge

[10] Osmany Freitas et al., "A Comprehensive Analysis of Wi-Fi Routers Firmware Vulnerabilities: From Re-Hosting to Invasion," SSRN, 2024. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4836002