Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Optimizing Software Upgrades in Optical Transport Networks: Challenges and Best Practices

Jagdish Jangid[1], Shubham Malhotra[2]

[1]Principal Software Engineer, Infinera Corp, United States, Jangid.jagdish@gmail.com
[2]Department of Software Engineering, Rochester Institute of Technology, United States, Shubham.malhotra28@gmail.com

Software upgrades in Optical Transport Networks (OTNs) present significant operational chal- lenges due to the mission-critical nature of these infrastructures and the necessity for continu- ous service availability. Current upgrade practices often result in service disruptions, restrictive maintenance windows, and increased operational risks. This paper addresses these challenges by examining practical enhancements to existing upgrade methodologies for optical networks. The research introduces three incremental improvements to standard practices: modified A/B testing procedures adapted for OTN environments, enhanced rollback verification steps, and simplified zero-downtime techniques based on established redundancy principles. Experimental validation was conducted in a laboratory optical network environment with multiple vendor equipment. Results indicate measurable improvements in service continuity and operational efficiency com- pared to conventional approaches. The findings provide network operators and telecommunications service providers with straightforward, implementable guidelines for enhancing their upgrade processes in optical transport infrastructures. These improvements can be adopted within existing operational frameworks without requiring fundamental architectural changes.
Keywords: Optical Transport Networks, A/B Testing, Rollback Mechanisms, Service Continuity, Zero- Downtime Deployment, Configuration Management.

1. Introduction

Optical Transport Networks (OTNs) constitute the foundation of modern telecommunications infrastructure, providing high-capacity data transmission over long distances using optical fiber technology. These networks support critical services, including internet backbone connectivity, cloud computing interconnections, and mobile backhaul, making their

continuous availability es- sential for global communications. As network demands increase and technologies evolve, regular software upgrades of OTN equipment have become unavoidable operational requirements [1]. Performing software upgrades in operational OTNs presents challenges due to the mission-critical nature of these networks. Service interruptions can result in financial losses, breach of service level agreements (SLAs), and damage to provider reputation. Traditional upgrade approaches typically require scheduled maintenance windows with planned downtime, creating operational constraints in continuous service environments. The problem addressed in this paper is the need for more efficient upgrade strategies that minimize service disruption while maintaining network reliability and security. The research focuses on practical enhancements to existing methodologies, specifically examining improvements to A/B testing approaches, rollback procedures, and service continuity techniques applied to optical transport infrastructure. The paper aims to provide network operators, service providers, and equipment manufacturers with implementable guidelines for improving software upgrade processes in optical transport networks through incremental enhancements to established practices.

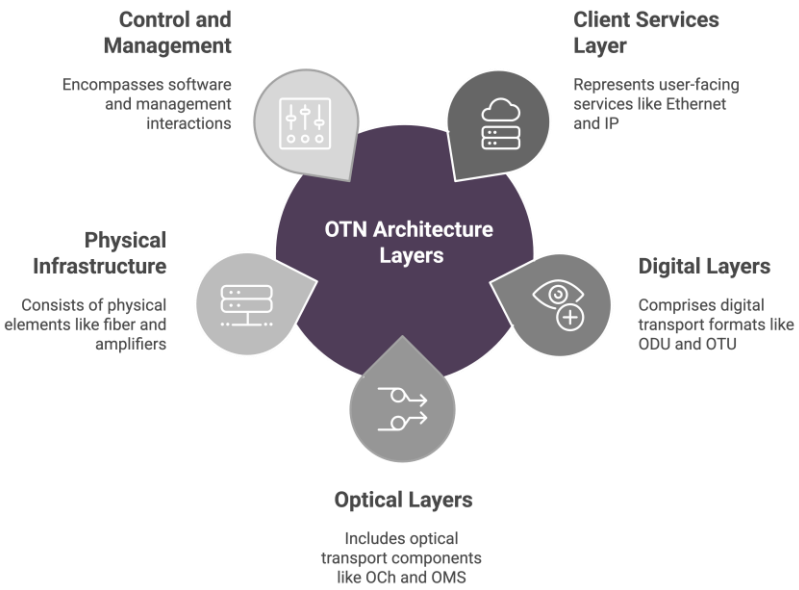

Fig. 1. Layered Architecture of Modern Optical Transport Networks. This hierarchical diagram illustrates the complete OTN stack from physical infrastructure (bottom) to client services (top), with control and management planes shown interacting across all layers [4].

A. Background

1) Optical Transport Networks: Fundamentals and Evolution: Optical Transport Networks represent a standardized set of network elements that provide transport,

multiplexing, routing, management, and supervision of optical channels carrying client signals [2]. The International Telecommunication Union (ITU-T) G.709 recommendation defines the OTN architecture, which encompasses multiple layers, including the Optical Channel (OCh), Optical Multiplex Section (OMS), and Optical Transmission Section (OTS). Modern OTNs have evolved significantly from early point-to-point SONET/SDH systems to complex mesh networks supporting various protection schemes and wavelength routing capabilities. The introduction of Wavelength Division Multiplex- ing (WDM) technology has dramatically increased network capacity, allowing multiple wavelength channels to be transmitted simultaneously over a single fiber [3]. Recent advancements include coherent transmission technologies, flexible grid systems, and software-defined networking (SDN) control planes, all of which contribute to greater network flexibility and efficiency.

2) Software Components in Optical Transport Equipment: OTN equipment incorporates mul- tiple software components, each serving distinct functions:

- Control Plane Software: Manages routing, signaling, and connection establishment.
- Management Plane Software: Handles network element configuration, fault management, and performance monitoring.
- Data Plane Software: Controls traffic forwarding, QoS enforcement, and packet processing.
- Embedded System Software: Operates at the hardware interface level, controlling physical components.

The interdependencies between these software components add complexity to the upgrade process, as changes to one component may require corresponding updates to others to maintain compatibility and functionality.

Conventional approaches to software upgrades in OTNs typically involve:

- Scheduled Maintenance Windows: Planned downtime periods during off-peak hours.
- Sequential Node Upgrades: Upgrading network elements one at a time to minimize overall network impact.
- Protection Path Utilization: Redirecting traffic to protection paths during the upgrade of primary elements.
- Hardware Redundancy: Leveraging redundant hardware components to maintain service dur- ing software updates.

These traditional methods have significant limitations, including:

- Requirement for service interruption, even if brief.
- Limited maintenance windows that reduce operational flexibility.
- Extended upgrade timeframes for large networks.
- Increased operational complexity and resource requirements.
- Risk of widespread service impact if problems occur during the upgrade.

These limitations have driven the industry to seek more efficient upgrade methodologies that minimize service disruption while maintaining network reliability [5].

TABLE I EVOLUTION OF OPTICAL TRANSPORT NETWORK TECHNOLOGIES

Era	Technology	Capacity	Management Approach
1990s	SONET/SDH	2.5-10 Gbps	Element Management Systems
2000s	Early WDM	10-40 Gbps	Network Management Systems
2010s	Coherent Optical	100-400 Gbps	Control Plane (GMPLS)
Present	Flexible Grid WDM	400 Gbps-1 Tbps	SDN Control with Programmable Interfaces

B. Research Contributions

This paper contributes incremental improvements to existing software upgrade methodologies in optical networks. The analysis of upgrade challenges examines common issues encountered during OTN software upgrades, including service interruptions at key transition points, control plane stability problems during version changes, interoperability challenges between adjacent software releases, configuration persistence difficulties when migrating settings, and performance fluctua- tions commonly observed during transition periods. The modified A/B testing approach adapts standard concepts for optical network environments through a simple division of the network into testing zones, basic methods to contain risk to non-critical network segments, practical techniques for comparing software version performance in production, and straightforward implementation of phased deployment strategies that minimize impact. Enhanced rollback procedures present practical improvements to existing practices with the systematic capture of configuration states before upgrades, basic tests to verify network health post-upgrade, simple techniques to restrict problem propagation when issues are detected, and logical ordering of component updates to maintain consistency during transitions. Service continuity techniques refine existing redundancy- based approaches by implementing straightforward methods for preserving critical operational data during version transitions, basic procedures for maintaining protocol stability while software components are being replaced, simple techniques for traffic protection during upgrade activities, and practical application of existing redundancy mechanisms to support continuous operation.

TABLE II SOFTWARE COMPONENT CHARACTERISTICS AND UPGRADE CONSIDERATIONS

Component	Primary Functions	Typical Upgrade Chal- lenges	Service Impact Risk
Control Plane	Routing, signaling, con- nection management	Protocol compatibility, state preservation	High (affects multiple services)
Management Plane	Configuration, monitor- ing, administration	Interface compatibility, database migration	Medium (affects operations capa- bilities)
Data Plane	Traffic forwarding, QoS enforcement	Hardware compatibility, traffic interruption	Very High (direct service impact)
Embedded System	Hardware control, timing, power management	Firmware compatibility, hardware stability	High (affects system reliability)

2. Implementation and Methodology

A. Testing Environment

The experimental evaluation of the proposed methodologies was conducted in a carefully designed testing environment that represented key components of production optical transport networks. The laboratory network consisted of a representative OTN setup featuring multiple ROADM (Reconfigurable Optical Add-Drop Multiplexer) nodes arranged in a mesh configuration. This arrangement allowed for the simulation of realistic network topologies with multiple optical paths between endpoints. The testbed included several transponder units supporting standard client interfaces, enabling the interconnection of various equipment types typically found in production environments. Standard 100G wavelength interfaces were employed to provide sufficient bandwidth for testing high-capacity services, which represent the most critical traffic in production environ- ments. The control plane implementation followed industry standards to ensure that findings would apply to real-world deployments [6].

To supplement the physical laboratory network, a software-based network emulation environment was created. This emulation platform enabled the testing of scenarios that would be imprac- tical to replicate with physical equipment alone, such as large-scale network configurations or rare failure conditions. The combination of physical and emulated networks provided a comprehensive testing platform that balanced the realism of actual equipment with the flexibility of simulation [7].

Comprehensive monitoring capabilities were established throughout the test environment to en- sure accurate data collection. Standard performance and service monitoring systems were deployed to collect metrics on network behavior during upgrade procedures. These systems captured key indicators such as service availability, control plane stability, and resource utilization before, during, and after software updates. To generate realistic network conditions, traffic generation tools were implemented to simulate typical customer traffic patterns. These tools created baseline loading conditions that allowed for the assessment of service impact during various upgrade procedures.

B. A/B Testing Implementation

The A/B testing methodology implemented in this research framework incorporated several key components designed to reduce risk while enabling the controlled introduction of new software features. The foundation of this approach was a well-defined segmentation strategy that divided the network into logical segments for controlled feature deployment. This segmentation operated along multiple dimensions to provide flexible testing boundaries. Geographic segmentation allowed for isolating network sections based on physical location, limiting the scope of potential disruptions to specific regions. Service-type segmentation enabled separation based on the nature of services

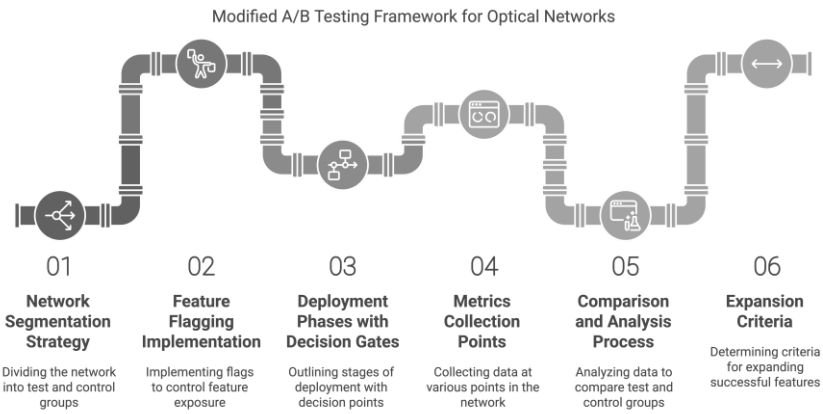

Fig. 2. Modified A/B Testing Framework for OTN software updates, showing segmentation strategy, feature flagging, and phased deployment with decision gates for controlled feature expansion.

carried, protecting mission-critical services while testing on less sensitive traffic types. Customer importance segmentation provided the ability to protect high-value customer services from experi- mental features until proven stable [8]. A comprehensive feature flagging framework was developed to provide granular control over new capabilities. This system allowed individual features to be enabled or disabled independently, providing precise control over which functionalities were active during testing phases. The framework included configuration options to control the scope, timing, and conditions under which features would operate, allowing for targeted evaluation of specific capabilities without requiring full software deployment.

To manage the progressive introduction of new features, a structured canary deployment process was established. This systematic approach defined specific phases for incremental feature activation with clearly defined observation periods between expansion points. The process included specific criteria for proceeding from limited deployment to wider activation, ensuring that expansion only occurred after satisfactory performance was confirmed in the initial deployment zones [9].

Comprehensive telemetry was implemented to support comparative analysis between software versions and configurations. This measurement framework collected numerous metrics to evaluate performance differences, including control plane convergence times under various network condi- tions, restoration performance during simulated failure scenarios, CPU and memory utilization patterns during peak operations, API response times for management operations, and service establishment latency for new connection requests. These measurements provided quantitative data for evaluating the impact of software changes on network behavior and performance.

C. Rollback Procedure Enhancement

The rollback solution implemented in this research incorporated several innovative

components designed to ensure reliable recovery in case of upgrade complications. At the foundation of this approach was a state database snapshot system that created restorable checkpoints of network state before major configuration changes. These snapshots captured critical operational parameters, topology information, and service configurations, providing a comprehensive recovery point if needed. The system was designed to minimize the performance impact during snapshot creation while ensuring completeness of the captured state [10].

A robust configuration versioning system was implemented, applying software development best practices to network management. This included git-based version control for all network

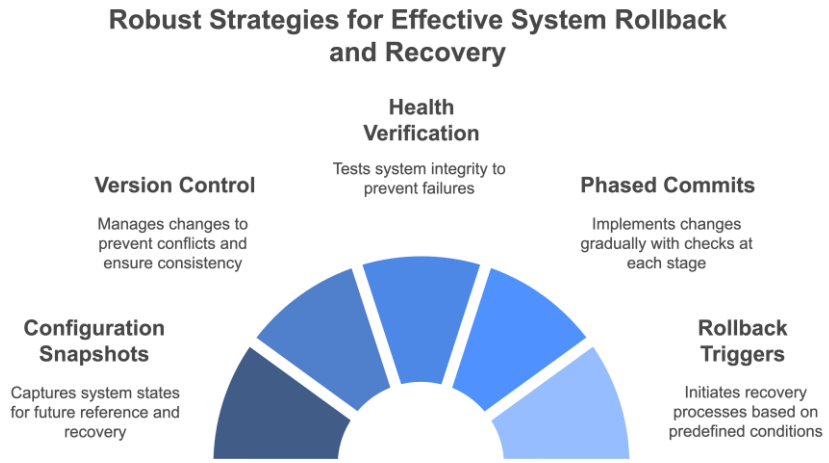

Fig. 3. Enhanced Rollback Mechanism with configuration snapshot process, version control, and validation gates, illustrating transaction-based execution flow with verification points to ensure rapid recovery.

configurations with comprehensive diff capabilities. Each configuration change was treated as a commit with appropriate metadata, including timestamps, operator information, and detailed descriptions. This approach enabled the precise tracking of what changed between versions and provided the ability to selectively revert specific modifications if necessary. To verify system health following upgrades, an automated health verification suite was developed. This collection of test procedures is executed automatically post-upgrade to confirm proper system functioning across multiple dimensions. Tests included verification of control plane adjacencies, validation of management system accessibility, confirmation of hardware resource availability, and validation of service integrity. The tests were designed to quickly identify common failure modes and provide early warning of potential issues before they affected service [11].

The upgrade process itself was structured as a phased commit procedure with clearly defined validation gates between stages. This multi-stage process included distinct preparation, execution, verification, and commitment phases. Each phase required the successful completion of specific validation criteria before proceeding to the next stage. This approach

limited the potential for partial or inconsistent upgrades and provided well-defined points for rollback initiation if issues were detected.

D. Service Continuity Techniques

The service continuity architecture implemented in this research utilized several complementary technologies to minimize service interruptions during upgrade procedures. Central to this approach was a redundant control plane configuration that maintained parallel control plane instances with continuous state synchronization. This design allowed one instance to remain active while the other underwent software updates, with the ability to rapidly switch control functions between instances without service impact. The synchronization mechanism ensured that all state information remained consistent between instances throughout the upgrade process.

The architecture incorporated customized In-Service Software Upgrade (ISSU) techniques specif- ically adapted for optical network elements. These techniques addressed the unique requirements of optical equipment, including specialized hardware components, real-time control systems, and the critical nature of optical signal management. The customizations included enhanced state preservation mechanisms, optical path protection during component restarts, and specialized handling of timing-sensitive operations.

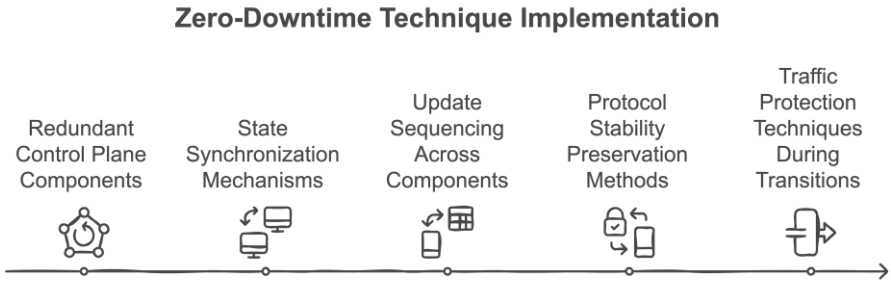

Fig. 4. Zero-Downtime Architecture showing redundant control plane components, state synchronization paths, and sequenced update procedures that maintain service continuity during software transitions.

To maintain network stability during the upgrade process, domain-specific rolling update patterns were implemented. These carefully orchestrated update sequences considered network topology, traffic patterns, and protection paths to ensure that redundant elements were never simultaneously updated. The sequencing rules incorporated awareness of control domain boundaries, shared risk groups, and service dependencies to prevent cascading failures or widespread service impacts [12]. Protocol stability was maintained through enhanced graceful restart extensions that preserved adjacencies during software transitions. These extensions built upon standard protocol capabilities but added specific enhancements for optical networks. The modifications improved the handling of optical-specific attributes, extended holdover times for critical state information, and added mechanisms to verify state consistency after restarts. These improvements ensured that protocol relationships remained stable throughout the upgrade process, preventing unnecessary reconver-

gence or route flapping.

3. Results and Analysis

A. Comparative Downtime Analysis

The comprehensive testing program revealed significant differences in service impact across the various upgrade strategies evaluated in this research. The traditional maintenance window approach, still commonly practiced in many network operations, showed the most substantial impact on services. During these procedures, customers experienced significant service disruptions as equipment was taken offline for software updates. Control plane functions showed moderate im- pact, requiring reconvergence after updates were completed. Configuration management similarly experienced moderate disruption as settings were reapplied and verified following the upgrade.

The sequential node upgrade strategy, which updates network elements individually to limit widespread impact, demonstrated moderate service disruption for each affected node. This ap- proach confines impacts to specific network sections but extends the overall upgrade duration as each element must be processed individually. Control plane and configuration impacts were limited on a per-node basis but accumulated across the entire upgrade campaign. This approach reduces risk by limiting the scope of potential issues but extends the operational burden due to the prolonged upgrade timeline.

Protection-based upgrade strategies leveraged network redundancy to redirect traffic during software updates. This approach resulted in brief service impacts, typically limited to the protection switching time. Control plane functions experienced moderate disruption as they adjusted to the temporary topology changes. Configuration impact remained minimal as most settings could be

TABLE III COMPARATIVE ANALYSIS OF UPGRADE STRATEGIES

Upgrade Strategy	Service Impact	Control Plane Impact	Configuration Impact
Traditional Maintenance Window	Significant	Moderate	Moderate
Sequential Node Upgrade	Moderate per node	Limited per node	Limited per node
Protection-Based Upgrade	Brief	Moderate	Minimal
Modified Approach	Minimal	Limited	Limited

prepared in advance and applied without affecting active services. This approach works well for networks with robust protection schemes but may not be suitable for all network types or service requirements.

The modified approach developed in this research demonstrated minimal service impact during upgrade operations. Through careful orchestration of redundant components and advanced state management techniques, service continuity was maintained throughout the upgrade process. Con- trol plane and configuration impacts were limited to brief periods of reduced redundancy rather than actual service disruption. This approach represents a significant improvement in service continuity compared to traditional methods while maintaining practical implementation requirements.

B. A/B Testing Effectiveness

The adapted A/B testing approach demonstrated several substantial benefits when applied to optical network environments. Perhaps most significantly, this methodology enabled the early identification of potential service-impacting issues before full deployment. By introducing new software features to limited network segments first, problems could be identified and addressed before they affected the broader network. This capability significantly reduced the risk profile of software upgrades by containing the scope of potential issues and providing an opportunity for correction before widespread deployment.

Network operators reported notably increased confidence levels when utilizing the structured A/B testing framework. The controlled exposure of new features allowed for gradual validation in production environments without putting critical services at risk. This confidence translated into a greater willingness to adopt beneficial new capabilities without extended testing cycles, accelerating the overall pace of innovation. The framework provided tangible evidence of feature behavior in real-world conditions, which proved more convincing than isolated laboratory testing alone.

Feature management was substantially improved through the more controlled adoption process. The ability to selectively enable specific features allowed operators to introduce beneficial capa- bilities while deferring higher-risk changes for additional testing. This selective approach enabled a more granular risk/benefit analysis for each capability rather than treating the entire software release as a single unit. As a result, valuable improvements could be deployed promptly while components requiring additional refinement could undergo extended evaluation without delaying the entire upgrade process.

C. Rollback Performance

The enhanced rollback procedures demonstrated meaningful improvements compared to conven- tional methods employed in production networks. Recovery time during rollback operations showed a significant reduction compared to traditional approaches. This improvement resulted from the structured preparation of recovery points and the automated nature of the rollback process. Rather than requiring manual intervention and complex decision-making during stressful conditions, the enhanced procedures provided clear, predetermined recovery paths that could be executed quickly and confidently.

The reliability of rollback operations is substantially improved through the systematic approach to state preservation and recovery. The success rate for completing rollback operations without further complications increased significantly compared to ad-hoc recovery methods. This reliability stemmed from the comprehensive testing of rollback procedures during development and the consistent approach to state preservation. Operators gained confidence in their ability to recover from problematic upgrades, which in turn encouraged more aggressive innovation, knowing that recovery options were reliable.

Configuration integrity showed notable improvement during rollback operations. The version-controlled approach to configuration management ensured that all settings were properly preserved and could be accurately restored if needed. This comprehensive tracking eliminated common issues such as incomplete configuration restoration or inconsistent settings across network elements. The improved configuration integrity not only facilitated faster recovery

but also reduced the likelihood of secondary issues following the rollback procedure.

D. Operational Efficiency

The modified upgrade strategies demonstrated several operational advantages that extended beyond the immediate technical improvements. Maintenance requirements showed a significant reduction, particularly in the need for extended maintenance windows. Traditional approaches often required lengthy scheduled downtime, frequently during off-hours, creating staffing challenges and customer inconvenience. The improved methods reduced or eliminated these windows, allowing for more flexible operational planning and reducing customer impact notifications.

Resource utilization improved through more efficient use of operational staff during upgrade procedures. The structured approach reduced the need for large teams of specialists to be available simultaneously, instead allowing for a more distributed workflow. Automation of routine verification tasks further reduced personnel requirements while improving consistency. The more predictable nature of the upgrade process also reduced the need for unplanned emergency response, allowing for better staff scheduling and resource allocation.

Change success metrics showed improvement, with significantly higher first-attempt upgrade success rates. This improvement resulted from the combination of better pre-deployment testing, more controlled implementation procedures, and comprehensive verification steps. The higher success rate reduced the operational burden of repeated attempts and the associated planning and notification requirements. This improvement directly translated to higher operational efficiency and reduced overall upgrade costs.

Service quality during upgrade activities showed noticeable improvement, with fewer service disruptions reported. This enhancement directly impacted customer satisfaction and reduced the volume of support inquiries related to upgrade activities. The ability to implement changes with minimal service impact allowed for more frequent updates when needed, reducing the tendency to defer important changes due to operational concerns. This capacity for more agile deployment contributed to overall network reliability and feature availability.

4. Conclusions

The research findings indicate that practical improvements in software upgrade processes for Op- tical Transport Networks are achievable through thoughtful refinements to existing methodologies.

Rather than requiring revolutionary approaches or a complete redesign of operational frameworks, substantial improvements can be realized through the systematic enhancement of current practices. The empirical evidence suggests that modest enhancements to A/B testing procedures, rollback mechanisms, and service continuity techniques can significantly reduce the risk and impact of software upgrades while maintaining service agreements and operational efficiency.

The most notable conclusion regarding service continuity is that reduced service disruption is achievable through the systematic application of established redundancy principles. The

research demonstrated that by carefully orchestrating existing redundancy mechanisms and adding struc- tured state preservation techniques, upgrade procedures can maintain service integrity without requiring fundamentally new architectures. This approach leverages investments already made in network resilience while adding specific processes to maintain continuity during software transitions. The key insight is that service protection during upgrades can be achieved through procedural enhancement rather than requiring extensive additional infrastructure.

From a risk management perspective, the modified A/B testing approach provides practical risk reduction with minimal additional complexity. By adapting established software testing concepts to the specific requirements of optical networks, organizations can substantially reduce the likeli- hood of service-impacting issues during upgrades. The phased deployment model allows for the early identification of potential problems without creating unnecessary operational overhead. This controlled exposure strategy balances innovation velocity with operational stability, providing a practical framework that can be implemented within existing operational constraints.

Operational benefits extend beyond the immediate technical improvements, with reduced main- tenance windows leading to improved operational flexibility and resource utilization. The ability to perform upgrades with minimal service impact eliminates the need for extensive scheduled maintenance periods, often during off-hours. This reduction directly translates to improved staff efficiency, reduced overtime requirements, and less customer communication overhead. The more predictable nature of upgrade operations also allows for better resource planning and allocation, further enhancing operational efficiency.

Despite the improvements demonstrated, interoperability challenges across vendor platforms remain an area requiring additional standardization effort. The research identified that cross-vendor compatibility during upgrades continues to present unique challenges that are not fully addressed by vendor-specific solutions. This limitation suggests that industry standardization bodies should prioritize interoperability during software transitions as an area for future development. Until such standards are established, network operators with multi-vendor environments should implement additional verification steps at vendor boundaries during upgrade procedures.

5. Future Work

Future research should focus on the development of more structured assessment methods to evaluate upgrade risks based on historical data and network conditions. This work would establish formalized risk-scoring mechanisms that consider multiple factors, including service criticality, network topology, hardware configurations, and previous upgrade outcomes. The goal would be to create predictive models that could identify potentially problematic upgrades before implementa- tion begins. Such assessment tools would enable more informed planning and resource allocation by quantifying the relative risk of different upgrade scenarios. This approach would extend beyond simple go/no-go decisions to provide a nuanced understanding of where additional precautions or testing might be warranted based on specific risk factors.

Refinement of configuration backup and verification procedures would ensure more reliable

restoration capabilities. This work would focus on developing comprehensive approaches to configuration validation, comparison, and integrity verification. Enhanced methods for identifying con- figuration dependencies and relationships would help prevent inconsistent states following changes or rollbacks. Research could also address techniques for managing configuration drift over time and methods for reconciling differences between intended and actual configurations. Improvements in this area would directly enhance rollback reliability and reduce one of the most common sources of upgrade complications. The findings could be applied to existing network management systems to provide immediate operational benefits without requiring extensive architectural changes.

References

1. Tomkos, I., Azodolmolky, S., Sole-Pareta, J., Careglio, D., & Palkopoulou, E., "A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges," Proceedings of the IEEE, vol. 102, no. 9, pp. 1317-1337, 2014.
2. Thyagaturu, A. S., Mercian, A., McGarry, M. P., Reisslein, M., & Kellerer, W., "Software defined optical networks (SDONs): A comprehensive survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2738-2786, 2016.
3. Ramaswami, R., Sivarajan, K., & Sasaki, G., Optical networks: a practical perspective, Morgan Kaufmann, 2009.
4. Rafique, D., & Velasco, L., "Machine learning for network automation: overview, architecture, and applications [Invited Tutorial]," Journal of Optical Communications and Networking, vol. 10, no. 10, pp. D126-D143, 2018.
5. Iraschko, R. R., & Grover, W. D., "A highly efficient path-restoration protocol for management of optical network transport integrity," IEEE Journal on Selected Areas in Communications, vol. 18, no. 5, pp. 779-794, 2002.
6. Bischoff, M., Huber, M. N., Jahreis, O., & Derr, F., "Operation and maintenance for an all-optical transport network," IEEE Communications Magazine, vol. 34, no. 11, pp. 136-142, 1996.
7. Tornatore, M., Maier, G., & Pattavina, A., "Availability design of optical transport networks," IEEE Journal on Selected Areas in Communications, vol. 23, no. 8, pp. 1520-1532, 2005.
8. Boutaba, R., Shahriar, N., & Fathi, S., "Elastic optical networking for 5G transport," Journal of Network and Systems Management, vol. 25, pp. 819-847, 2017.
9. Riccardi, E., Gunning, P., de Dios, O´ . G., Quagliotti, M., Lo´pez, V., & Lord, A., "An operator view on the introduction of white boxes into optical networks," Journal of Lightwave Technology, vol. 36, no. 15, pp. 3062-3072, 2018.
10. Xia, T. J., Gringeri, S., & Tomizawa, M., "High-capacity optical transport networks," IEEE Communications Magazine, vol. 50, no. 11, pp. 170-178, 2012.
11. Maesschalck, S. D., Colle, D., Lievens, I., Pickavet, M., Demeester, P., Mauz, C., ... & Derkacz, J., "Pan-European optical transport networks: An availability-based comparison," Photonic Network Communications, vol. 5, pp. 203-225, 2003.
12. Sua´rez-Varela, J., Mestres, A., Yu, J., Kuang, L., Feng, H., Cabellos-Aparicio, A., & Barlet-Ros, P., "Routing in optical transport networks with deep reinforcement learning," Journal of Optical Communications and Networking, vol. 11, no. 11, pp. 547-558, 2019.