Executive Summary

Network software upgrades have long represented one of the most challenging operational tasks in enterprise environments. Traditional upgrade processes require complete system reloads that can result in 3-5 minutes of network downtime—an eternity for mission-critical applications, real-time services, and SLA-bound operations.

Extended Fast Software Upgrade (xFSU) fundamentally alters this equation. By utilizing the separation of Control and Data planes with Non-Stop Forwarding (NSF) and the Graceful Restart architecture, xFSU ensures hitless traffic for supported protocols.

Customers do not have to worry about usability; the code performs an automated self-check and provides clear feedback. The underlying separation of the control and data planes is entirely transparent, so you are not required to understand or manage those architectural details.

This document examines the technical challenges that xFSU addresses, explains how the technology works, and provides guidance on deployment considerations.

The Challenge: Network Upgrades in Production Environments

Before understanding xFSU's value proposition, it's essential to understand why traditional network upgrades are so disruptive and what happens during a conventional reload operation.

The Anatomy of a Traditional Reload

When a network switch undergoes a traditional reload, the following sequence occurs:

1. **Complete System Shutdown:** The operating system initiates a full shutdown, terminating all processes and clearing all volatile memory.
2. **Hardware Reset:** The ASIC forwarding tables are cleared. All programmed MAC addresses, ARP entries, and routing information are purged from the hardware.
3. **Boot Sequence:** The system goes through Power-On Self Test (POST), ROMMON initialization, and IOS-XE boot — a process that typically takes 2-3 minutes.
4. **Protocol Convergence:** After the booting process, the routing protocols must re-establish adjacencies, spanning tree must reconverge, and the switch must relearn all MAC addresses and ARP bindings.
5. **Traffic Recovery:** After completing all the above steps, the traffic forwarding resumes at full capacity.

The Real-World Impact

A 3 to 5 minutes downtime window of a traditional reload creates cascading effects throughout the network.

Routing Protocol Disruption

OSPF and IS-IS routers will detect the loss of adjacency and begin reconvergence. Depending on the timer configurations and network topology, this can trigger route recalculations across the entire routing domain. BGP sessions will drop, potentially causing route withdrawals that propagate to upstream providers.

Spanning Tree Reconvergence

When a switch reloads, spanning tree topology changes occur. Ports transition through listening and learning states (30 seconds with legacy STP), and the network must recalculate the optimal forwarding topology. If the reloading switch is the root bridge or is in the forwarding path, traffic blackholes can occur until convergence completes.

State Loss and Flooding

All dynamically learned information is erased and must be reacquired: MAC address tables (requiring flooding until relearned), ARP caches (causing ARP storms), DHCP snooping bindings (potentially blocking legitimate clients), and port security learned addresses. The flooding during MAC relearning can saturate links and impact traffic across the network.

Application Impact

Modern applications are increasingly intolerant of network disruption: VoIP calls drop after seconds of packet loss, video conferencing quality degrades significantly, database replication may fail and require resynchronization, and eal-time trading systems can miss market opportunities or fail to execute critical transactions.

xFSU: A Fundamentally Different Approach

xFSU addresses these challenges by fundamentally changing what happens during a software upgrade or reload. Rather than treating an upgrade as a " cold start" that discards all state, xFSU preserves critical information across the reload boundary.

The Core Innovation: Control and Data Plane Separation

xFSU leverages the architectural separation between the Control Plane and Data Plane that exists in modern network switches. Understanding this separation is key to understanding how xFSU achieves its dramatic reduction in downtime.

The Control Plane: RIB

The Routing Information Base (RIB) resides in the control plane and manages the routing decisions and topology changes. It runs the routing protocols (OSPF, BGP, IS-IS), processes the route updates and determines the best paths to network destinations. The RIB is maintained in software on the CPU.

The Data Plane: FIB

The Forwarding Information Base (FIB) resides in the data plane and enables high-speed, hardware-based packet forwarding. It is programmed into the ASIC's TCAM and registers, allowing wire-speed forwarding decisions without CPU involvement. The FIB is derived from the RIB but operates independently once programmed.

Graceful Reload: The Foundation of xFSU

Graceful reload is the core mechanism that enables xFSU to maintain network connectivity during a software upgrade. It coordinates the preservation and restoration of the forwarding state while ensuring that routing protocol neighbors maintain their adjacencies.

How Graceful Reload Works

When a graceful reload is initiated, the switch performs a coordinated sequence of operations:

1. **Pre-Reload Notification:** The control plane notifies all routing protocol neighbors that a graceful restart is about to occur. This is done using protocol-specific mechanisms (described below).
2. **State Checkpoint:** Critical forwarding state—including the FIB, interface configurations, and protocol state—is checkpointed to persistent storage.
3. **Control Plane Restart:** The control plane software restarts while the data plane continues forwarding traffic using the preserved FIB entries.
4. **State Restoration:** After restart, the control plane restores its state from the checkpoint and re-synchronizes with neighbors who have maintained their adjacencies throughout the process.

Non-Stop Forwarding (NSF)

Non-Stop Forwarding (NSF) is the capability that allows the data plane to continue forwarding packets while the control plane is restarting. NSF works in conjunction with graceful restart to provide hitless or near-hitless upgrades.

The NSF Principle

The fundamental principle of NSF is that **the forwarding plane can operate independently of the control plane** for a limited period. During this window, packets continue to be forwarded based on the existing FIB entries while the control plane restarts and recovers. As long as the network topology doesn't change during this window, traffic continues to flow correctly.

NSF Requirements

For NSF to work effectively, several conditions must be met:

- **Stable Topology:** The network topology must remain stable during the restart window. If links fail or new routes are announced, the stale FIB entries may cause traffic blackholes.
- **NSF-Aware Neighbors:** Neighboring routers must support and be configured for graceful restart to maintain their adjacencies during a restart.
- **Timely Recovery:** The control plane must restart and resynchronize before protocol timers expire on the neighbors.

Protocol-Specific Graceful Restart Mechanisms

Each routing protocol implements graceful restart differently, but they all share the same goal: allowing neighbors to maintain their routing state and adjacencies while the restarting router recovers.

BGP Graceful Restart

BGP graceful restart as defined by RFC 4724 allows BGP sessions to be preserved across a restart, preventing route withdrawals and the resulting traffic disruption.

How it Works

1. **Capability Advertisement:** During session establishment, BGP peers exchange graceful restart capabilities in their OPEN messages. This indicates their ability to preserve forwarding state during a restart.
2. **Restart Notification:** When the switch initiates xFSU, the BGP process sends End-of-RIB markers and sets the restart bit in its capability advertisement.

3. **Route Preservation:** Neighbors mark all routes learned from the restarting router as "stale" but continue to use them for forwarding. They do NOT withdraw these routes from their own neighbors.
4. **Session Re-establishment:** After restart, the BGP process re-establishes sessions with its neighbors and exchanges routes.
5. **Stale Route Cleanup:** After the full routing table is exchanged, stale routes that were not refreshed are removed.

Key Timers

The restart time (default 120 seconds) defines how long neighbors will wait for the restarting router to reestablish the session. The Stalepath Time (default 360 seconds) defines how long stale routes are preserved. xFSU completes well within these timers.

IS-IS Graceful Restart

IS-IS graceful restart as defined in RFC 5306 allows IS-IS adjacencies to be maintained during a router restart, preventing SPF recalculation across the network.

How it works

1. **Restart TLV:** The restarting router includes a restart TLV in its Hello PDUs indicating it is capable of graceful Restart.
2. **Restart Request:** When xFSU begins, IS-IS sends Hello PDUs with the Restart Request (RR) bit set, indicating a restart is in progress.
3. **Neighbor Suppression:** Neighbors that receive the RR bit suppress their normal reaction to the restart. They do NOT remove the restarting router from their LSP database and do NOT trigger SPF recalculation.
4. **Adjacency Preservation:** Neighbors maintain the adjacency in the "UP" state even though Hello PDUs may be missed during the restart window.
5. **Database Synchronization:** After restart, IS-IS re-synchronizes its link-state database with neighbors using normal flooding procedures.

Key Benefit

By suppressing SPF recalculation, IS-IS graceful restart prevents the ripple effect of route changes across the entire IS-IS domain. The network continues to forward traffic along established paths.

OSPF Graceful Restart

OSPF Graceful Restart defined as per RFC 3623 for OSPFv2, and RFC 5187 for OSPFv3 enables OSPF routers to maintain their adjacencies and prevent network-wide reconvergence during a restart.

How it works

1. **Grace LSA:** Before restarting, OSPF floods a grace LSA (Link State Advertisement) to all neighbors. This LSA contains the grace period (how long neighbors should wait) and the restart reason.
2. **Helper Mode:** Neighbors that receive the grace LSA enter "helper mode." In this mode, they maintain the adjacency with the restarting router and continue to advertise it as reachable in their own LSAs.

3. **SPF Suppression:** Helper routers do not run SPF or update their routing tables in response to the restart. Traffic continues to be forwarded along existing paths.
4. **Adjacency Re-establishment:** After restart, OSPF re-establishes adjacencies with neighbors, synchronizes the LSDB, and exits graceful restart mode.
5. **Normal Operation:** Neighbors exit helper mode when they see the restarting router's new router LSA, indicating successful recovery.

OSPF Timer Consideration

For optimal xFSU performance with OSPF, consider tuning the retransmit interval. The default OSPF retransmit interval is 5 seconds—if the control plane restart takes longer than this, LSA retransmissions may occur. xFSU's minimal downtime is designed to complete within this window.

Protocol Graceful Restart Summary

Protocol	Notification Mechanism	Neighbor Behavior
BGP	Graceful restart capability in OPEN message; End-of-RIB markers	Mark routes as stale; continue forwarding; do not withdraw routes
IS-IS	Restart TLV in Hello PDUs; Restart Request (RR) bit	Suppress SPF; maintain adjacency UP; preserve LSP database
OSPF	Grace LSA flooded before restart	Enter helper mode; suppress SPF; continue advertising restarting router

What xFSU Preserves

- **Hardware Forwarding State:** ASIC TCAM and register entries are cached in the memory before the upgrade and repopulated afterward, preserving Layer 2 and Layer 3 forwarding paths.
- **Protocol Adjacencies:** Routing protocol neighbors are notified of the graceful restart, so they maintain their adjacencies rather than declaring that the switch is down.
- **Routing Tables:** The RIB and FIB persist across the reload due to graceful reload, eliminating the routing convergence phase.
- **Interface State:** Physical port states are maintained, preventing link flaps that trigger spanning tree reconvergence.

The Result: A "Hiccup" Instead of an Outage

With xFSU, the network experiences a brief interruption rather than 3 to 5 minutes. Control plane processing pauses momentarily while the software restarts, but the data plane continues forwarding based on the preserved hardware state. To the rest of the network, this appears as a brief hiccup—routing neighbors may see a few missed hello packets but will not declare the adjacency down.

How xFSU Works: The Upgrade Flow

Understanding xFSU's operation flow helps network operators appreciate both its capabilities and the source of its minimal downtime. The process involves careful coordination between control plane and data plane components.

Phase 1: Initiation and Notification

When the xFSU command is issued:

```
C9300# install add file <image> activate xfsu commit
```

The control plane sends protocol-specific messages to notify neighbors of the impending Graceful Restart:

- **BGP:** Sends End-of-RIB markers and advertises Graceful Restart capability
- **IS-IS:** Sets the Restart Request (RR) bit in Hello PDUs
- **OSPF:** Floods Grace LSA to all neighbors

Neighbors will now expect a brief interruption and will maintain their adjacencies rather than declaring the switch down.

Phase 2: Control Plane Upgrade

The control plane begins its upgrade from IOS XE V1 to V2. During this phase, **the data plane continues forwarding traffic** based on the existing V1 FIB entries. No traffic disruption occurs yet—packets continue to be forwarded at wire speed using the ASIC's programmed forwarding tables.

Phase 3: State Caching

Before the data plane upgrade, all ASIC TCAM and register entries are cached in memory. This cached state represents the complete forwarding configuration—every route, every MAC address entry, every ACL—that will be needed to restore forwarding after the ASIC reset.

Phase 4: Data Plane Upgrade (Downtime Window)

This is the only phase where traffic is impacted. The data plane upgrades from V1 to V2:

- Network interfaces are disabled
- L2/L3 traffic stops forwarding
- The ASIC is reset by flushing all TCAM/Register entries

This is the minimal downtime window.

Phase 5: State Restoration and Traffic Resume

The ASIC is repopulated with the cached TCAM/Register entries. Network interfaces are enabled. With the data plane now upgraded to V2 and the forwarding state restored, L2/L3 traffic resumes. The entire upgrade is complete.

Timeline Summary

Phase	What Happens	Traffic Status
Initiation	Notify neighbors (GR messages)	✓ Forwarding
Control Plane	IOS XE V1 → V2 upgrade	✓ Forwarding (V1 FIB)
State Caching	Cache TCAM/Registers to memory	✓ Forwarding
Data Plane	ASIC reset, interfaces disabled	✗ DOWN (a few seconds)
Restoration	Repopulate TCAM, enable interfaces	✓ Forwarding (V2)

Protocol and Feature Support

xFSU provides hitless behavior for specific protocols and features. Understanding what is and isn't supported is critical for planning maintenance windows.

Fully Supported

Layer 2 Features

- Layer 2 switching and VLAN forwarding
- Per-VLAN Spanning Tree (PVST+)
- Rapid Spanning Tree (RSTP) and Multiple Spanning Tree (MSTP)
- Static port-channels (mode: on)
- LACP (Link Aggregation Control Protocol)
- UDLD (UniDirectional Link Detection)
- VXLAN

Layer 3 Routing Protocols

- BGP (IPv4 and IPv6) with Graceful Restart
- OSPF, OSPFv2, and OSPFv3
- IS-IS

Features Requiring Additional Consideration

Protocols and features not listed above may experience longer downtime. This includes features that require full reinitialization, have external dependencies that cannot be preserved, or have not yet been integrated with the xFSU client framework. Always validate your specific feature set against the latest release notes. xFSU does block upgrade when certain non-compliant features are detected, like BFD, MACSEC.

Platform Support and Requirements

Supported Hardware

Cisco Catalyst 9300 Series – All models

Software Requirements

IOS-XE Version	CLI Syntax	Notes
17.3.2+	`activate reloadfast`	Initial support
17.6.x	`activate reloadfast`	Full support
17.8.x+	`activate xfsu`	Recommended (new CLI)
17.15.2+	`activate xfsu`	Less than a few seconds downtime

Operational Considerations

When to Use xFSU

xFSU is designed for environments where minimizing downtime is critical. Ideal deployment scenarios include:

- **Production Network Upgrades:** Upgrading access switches during business hours when extended outages are unacceptable.
- **24/7 Operations:** Environments like hospitals, financial trading floors, or manufacturing facilities where no good maintenance window exists.
- **SLA-Critical Services:** Networks supporting services with contractual uptime requirements.
- **Data Center Environments:** ToR switch upgrades, leaf-spine fabric maintenance, and environments sensitive to east-west traffic disruption.

When NOT to Use xFSU

xFSU adds complexity to the upgrade process. In some scenarios, a traditional reload is more appropriate:

- **Lab/Non-Production Environments:** Where downtime is acceptable and simplicity is preferred.
- **Major Version Migrations:** Some major version upgrades; for example, from 16.x to 17.x, may require configuration migration that benefits from a clean restart.
- **Unsupported Configurations:** If the eligibility check fails, address the underlying issues before attempting xFSU.
- **Critical Unsupported Features:** If your deployment relies heavily on features not supported by xFSU, plan for extended downtime accordingly. Some of the features not supported by xFSU are IPSec, MACsec, LISP, BFD, etc.

xFSU Commands and Verification

Checking Eligibility

Before attempting any xFSU operation, verify that the system meets all prerequisites:

```
show xfsu eligibility
```

Software Upgrade with xFSU

To upgrade to a new software version with minimal downtime:

```
install add file image-url activate xfsu commit
```

Fast Reload (Same Version)

To reload the current version using xFSU technology:

```
reload fast
```

Post-Upgrade Verification

After xFSU completes, verify successful operation:

```
show version | include reason
```

Expected output includes " Image Install with Reloadfast" or " Reload Fast Command"

```
show log | include FAST
```

Look for: %FED_IPC_MSG-5-FAST_RELOAD_COMPLETE

```
show xfsu status
```

Caveats and Known Issues

Prerequisites

xFSU operations require specific prerequisites to be met. The eligibility check verifies these conditions, but operators should be aware of the key requirements:

- **Install Mode:** The switch must be running in install mode, not bundle mode.
- **Autoboot Configuration:** For standalone switches, autoboot must be disabled. For stacks, SSO mode must be ready.
- **FPGA Compatibility:** The FPGA version must be supported for xFSU operation.
- **Stack Topology:** Stack configurations require a full ring topology—partial rings are not supported.
- **Licensing:** Stack operations require Network Advantage license and standalone operations require Network Essentials license.
- **MACsec:** MACsec is not eligible for xFSU.
- **Spanning Tree:** The root STP node or non-root node with multiple forwarding links must be eligible for xFSU.

Known Issues

Bug ID	Description	Resolution
CSCwk44644	FSU failure on stacks with SMU installed. Affects 17.12.2.	Fixed in 17.12.3, 17.15.1
CSCwr07980	PGA upgrade triggers during fast reload. Affects 17.12.3, 17.15.2.	Fixed in 17.12.4, 17.15.3

Important: Always consult the Cisco Bug Search Tool and release notes for your specific software version before planning xFSU operations in production environments.

Conclusion

xFSU represents a significant advancement in network upgrade technology. By leveraging the separation between control plane and data plane, combined with NSF and Graceful Restart capabilities, it transforms software upgrades from multi-minute outages into hitless upgrade. In newer architectures, ongoing work aims to achieve sub-seconds to zero downtime.

The protocol-specific Graceful Restart mechanisms—BGP's route preservation, IS-IS's SPF suppression, and OSPF's helper mode—work together to ensure that routing adjacencies remain intact while the control plane restarts. Combined with NSF's ability to continue forwarding on preserved FIB entries, xFSU delivers truly hitless upgrades for mission-critical networks.

For organizations operating mission-critical networks where downtime directly impacts business operations, xFSU enables maintenance activities that were previously relegated to overnight or weekend windows to be performed during normal business hours with minimal user impact.

Success with xFSU requires understanding its prerequisites, ensuring protocol timer configurations are optimized, and validating eligibility before each operation. When properly deployed on IOS XE 17.15.2 or later, xFSU delivers on its promise of hitless upgrade without compromising network reliability.