

UNIFIED ISSU: A COMPLETE APPROACH TO IN-SERVICE SOFTWARE UPGRADES

Comparing the Unified ISSU Capability in Junos OS to Other Approaches for In-Service Software Upgrades

Table of Contents

Executive Summary	3
Introduction	3
Background on ISSU	4
A Closer Look at ISSU	4
Juniper's Unified ISSU	7
Conclusion	8
References	8
Appendix A: Test Case: Verifying Performance of Juniper's Unified ISSU	8
Configuration	8
One-Line Execution	9
Results	9
About Juniper Networks	9

Table of Figures

Figure 1: High-level architecture view and software upgrade approaches	5
Figure 2: Lab test topology	8

List of Tables

Table 1: Comparisons of Different Unified ISSU Approaches	6
Table 2: Summary of Junos OS Unified ISSU Benefits and Features	7
Table 3: Test Results of Unified ISSU Versus Non-ISSU (Upgrade Path: Junos OS 9.4R1.8 to Junos OS 9.5R2.7) ...	9

Executive Summary

The problem of non-disruptive software upgrades is one that network equipment vendors have spent many research and development resources trying to address. Ultimately, the objective has long been the ability to keep equipment continuously in service while the software is being upgraded—networking vendors refer to this as in-service software upgrades (ISSU). In the pursuit of ISSU, many vendors have claimed victory by adopting their own meanings of the term. As a result not all claimed implementations of ISSU are created equal and there has been some muddying of the waters. To some, ISSU can mean the ability to upgrade a portion of the software image, or perhaps only a maintenance (bug fixes) release upgrade.

Juniper Networks® calls its ISSU implementation unified ISSU to distinguish from these other forms of ISSU. Unified ISSU is a true, comprehensive ISSU solution that fully delivers the promise of ISSU. It is the only implementation capable of performing major software upgrades (from one version to the next) that include new features for supporting services in a streamlined, automatic fashion. Furthermore, only unified ISSU is capable of performing maintenance upgrades without requiring reloads of forwarding line cards. Even as it is more complete than other solutions, unified ISSU is still faster. Service disruption during upgrades is several orders of magnitude quicker with unified ISSU than with any competitive solution—seconds as opposed to minutes.

This paper examines the benefits of implementing ISSU and explores the different approaches to ISSU taken by other industry IP/MPLS operating systems. Readers will learn how these approaches compare to the holistic solution delivered by Juniper's unified ISSU. Network planners and operators will find this paper compares and contrasts ISSU implementations, and provides critical information for designing "multi-9" high availability (HA) networks.

Introduction

With the explosion of video, mobile services, and other real-time applications, network users now impose more stringent requirements and expectations on what networks can deliver. Central to this, network uptime—or availability—is a critical element for ensuring a high-quality user experience. As a result, network equipment vendors have designed many features to minimize both planned and unplanned network downtime. However, the task of upgrading network OS software without network downtime has remained a daunting challenge.

Historically, software upgrades have required a reboot of hardware systems for the change to take effect. Thus, network equipment needs to be taken out of operation during the upgrade. In the case of routers, this means not only that revenue-generating bandwidth is temporarily unavailable, but it also creates a cascade of other effects—extra network capacity needs to be allocated to reroute application traffic, route tables need to be rebuilt, and any incompatibility with the new software version can be disastrous.

At first glance, software that sufficiently meets business needs and delivers value in a stable and cost-effective way might appear to incur little need for upgrade. However, software upgrades might be needed for a variety of reasons. Operators might want to upgrade even the most stable software as new features become available. For example, the evolution of services and business models often requires new software features or feature enhancements from vendors who support them; even legal and regulatory compliance can dictate the addition of new features or capabilities. In short, network operators who want to keep pace with innovation and the market often need to upgrade even the most stable OS.

Typically because a software upgrade requires the rebooting of hardware in order for the change to take effect, it is service impacting. The duration of the service impact depends on many factors, including the types of hardware, the complexity of the system, the efficiency of the software, and many others. Sometimes the actual hardware reboot can occur quickly, but the rebuilding of state information such as route tables can take much longer. On a router, for example, a reboot involves rebuilding routing adjacencies, exchanging routing information with neighbors, reconstructing the forwarding table, and downloading the forwarding information to line cards.

Lab tests have shown that a 16-slot core router can have a service disruption of up to 13 minutes during a traditional software upgrade. In a service provider network, the router can transmit up to 124 terabytes in this time. At the edge of the network, the service disruption can cause services to be temporarily unavailable. In the core of the network, although traffic often reroutes through a redundant path, service interruption can still be experienced and the network would run at a reduced capacity.

To minimize the service impact to the users, carriers traditionally would try to schedule maintenance windows during off-peak hours. However, today many critical services are automated to take place during these same "off-peak" hours (data backup; large file transfers)—this factor combined with the global nature of networks means there is no longer a clear-cut off-peak period.

Instead, network operators must now find ways to reduce the time and risk of planned maintenance events by ensuring continuous systems availability even during maintenance, changes, and upgrades. By eliminating the need for hardware reboots during the upgrade process, ISSU—if implemented correctly and completely—promises to virtually eliminate the risk associated with planned maintenance events.

Background on ISSU

ISSU is a mechanism that enables network devices to perform software upgrades without being taken out of operations. Normal business operations can continue while the upgrade is taking place. Service providers, enterprise networks, or data center operations can leverage ISSU to ensure that they do not lose valuable bandwidth and capacity during the course of a software upgrade.

Before delving into ISSU, it is helpful to take a step back and look at other capabilities that must be in place before it can become a reality. A major driving force of ISSU support is the “multi-9’s” network HA requirement. To reduce network downtime during an unexpected control plane failure, network operators are often encouraged to have redundant Routing Engines in place to enable HA. Features such as GRES, Nonstop Forwarding (NSF)/Graceful Restart (GR), and Nonstop Routing (NSR) provide protections against routing engine or process failures, and these features form the building blocks of unified ISSU. For more architectural details about unified ISSU design on a redundant system, please refer to the Juniper Networks white paper, ISSU: A Planned Upgrade Tool.

Graceful switchover, or sometimes referred to as stateful switchover, takes advantage of dual Routing Engines and provides stateful replication between the master and backup Routing Engines. When a failure occurs, the standby Routing Engine takes mastership and continues to provide services. But in order to preserve routing, graceful Routing Engine switchover must be combined with either NSF/GR or NSR during the failover. GR protocol extensions represent a solution to prevent adjacency and route flapping, but the solution requires each neighbor to support the GR protocol extensions, and there is a grace period within which the reconvergence needs to be completed. With NSR, the responsibility for repairing a failed Routing Engine is placed entirely on the router itself—there is no need to modify or extend existing routing protocols or place any demands on peers. Therefore, NSR is Juniper Networks preferred solution.

However, traditional implementations of these HA features were not designed with unified ISSU in mind and were only intended to provide redundancy when both Routing Engines were running the same software version. To upgrade from one complete software version to another, the Routing Engines must—by definition—be running different versions, so this further complicates the process for achieving unified ISSU. Unified ISSU requires much more than incremental, local code changes, so it cannot be added as an afterthought. Extra engineering effort has to be put in, and therefore more risk is involved.

A Closer Look at ISSU

In today’s advanced networking equipment, a software upgrade isn’t merely a single point of software replacement. Rather, it’s a package that consists of software files that run on various parts of the system—Routing Engines, forwarding engines, interface cards, etc. Hence, when looking at ISSU support, it’s important to view the network element as a system and understand the operating system’s impact and benefit to the system as a whole.

To understand the impact of different levels of ISSU on a system, it’s helpful to examine the various definitions that others have used for ISSU. In general, vendors typically mean one of three things when they claim ISSU support:

- **Software patch**—Bug fixes are applied to a specific software issue.
- **“Hitless” control plane upgrade**—Only control plane modules are upgraded.
- **Minor or maintenance upgrade**—The entire software image is upgraded, but only for bug fixes or minor feature enhancements.

These are distinguished from Juniper Networks definition of unified ISSU, where the entire software image is upgraded for adding new features (often referred to as a major release upgrade), or for bug fixes and minor feature enhancements (often referred to as a maintenance release).

When ISSU is a key factor in selecting network equipment, network designers should be aware of the full extent of unified ISSU possibilities and the benefit each brings, and choose a unified solution that treats the system as a whole. Figure 1 shows a high-level architecture view and illustrates different approaches in the market that are claimed as ISSU support.

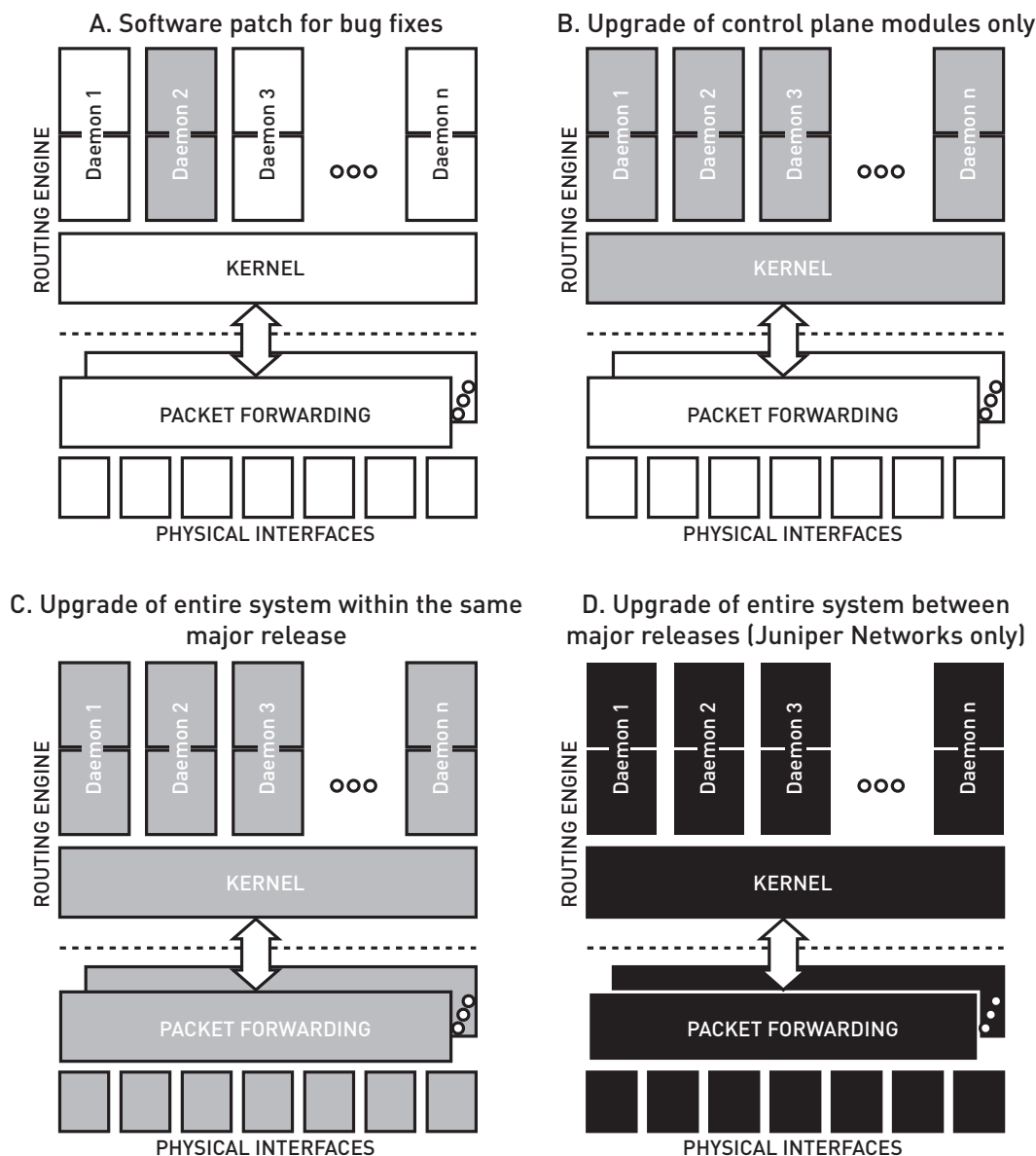


Figure 1: High-level architecture view and software upgrade approaches

Option A applies a software patch to fix specific problems. It is often marketed as ISSU, but in fact, it does not upgrade the entire OS and therefore cannot take advantage of new features or feature enhancements. It is touted as a quick way to address known software bugs without an entire software replacement, and can provide a fix to one particular software issue. Software patching provides a quick solution to existing problems, potentially with uninterrupted service, but it only meets a small subset of business needs and network requirements. Furthermore, this method does not work with every problem or bug, and customers report that they often cannot patch their most important issues.

The extent of the service disruption during patching is determined by the nature of the fix and the software architecture. In many cases, a software fix that addresses a system-wide issue requires rebooting all or many hardware components of the system, and can be just as service disruptive as a traditional software upgrade. The unpredictability makes it difficult to manage the expectation. Poorly designed patches can introduce new issues, which often require a subsequent patch. The quality control of different patch combinations imposes a significant engineering effort on implementers. For network operators, patch management becomes complex as the networks grow and have inconsistent patch versions throughout the network.

Another variation of option A can upgrade an individual software module to improve the code within that specific module. For example, this might include installing a newer version of a routing software package over an existing version of base code. Though the possibility of upgrading individual modules as needed sounds appealing, the complexity of qualifying and managing all permutations of software combinations can quickly become an operation and quality assurance burden. Therefore, this approach has not been seen on the market.

Option B is what some call a “hitless” control plane upgrade, and upgrades control plane modules only. Not only does it create an inconsistent state between the control plane and the forwarding plane, but also line card resets are eventually required for the forwarding plane to take advantage of the new features and functionality. At first glance, this method might appear to be not very service disruptive. But line card resets bring down interfaces, and as a result, routing and forwarding information needs to be relearned. Some implementers enforce a hard time limit within which the line card resets need to happen.

Now take a step back and look at how a true hitless upgrade could be achieved. Theoretically, a hitless upgrade would be feasible if both the control plane and the forwarding plane in a system had physical or virtual redundant components. A non-redundant component would cause some packet loss since it would most likely need to be reset to accommodate a software upgrade. But a fully redundant system is rather expensive to implement and rarely seen on the market¹. Thus, most ISSU implementations today run on partially redundant systems. For more information, please refer to the white paper, *ISSU: A Planned Upgrade Tool*.

Therefore, having a hitless control plane (that is, routing adjacencies stay up) does not equate to an undisrupted forwarding plane. On a partially redundant system, especially one whose redundancy level resides only within the control plane, how quickly the forwarding information can be refreshed to minimize the service disruption is an important part of ISSU design.

However, implementers in this category do not address the forwarding plane disruption and simply enforce line card resets.

Option C upgrades the entire system within the same major release. Therefore, only limited software enhancements and minor bug fixes can be exploited after the upgrade. Additionally, some implementers use the less preferred NSF/GR method to enable HA, which requires adjacent neighbors to have the capability to go into helper mode.

Options A, B, and C—though touted as ISSU by other vendors—each have their own disadvantages and are incomplete in their ability to provide true continuous systems availability.

Juniper Networks unified ISSU approach, option D, overcomes the limitations from the aforementioned approaches and provides a consistent, predictable, and manageable ISSU solution across releases and platforms. It allows the upgrade of the entire system between major releases and minor releases within a fraction of second to a few seconds. During the process, graceful Routing Engine switchover and NSR keep the control plane hitless while forwarding information gets quickly refreshed to minimize the service disruption.

Table 1 summarizes the benefits of each software upgrade approach and the levels of service disruptions during the execution of them.

Table 1: Comparisons of Different Unified ISSU Approaches

	A. Software Patch	B. Upgrade of Control Plane Modules Only	C. Upgrade of Entire System Within the Same Major Release	D. Juniper Networks Approach: Upgrade of Entire System Between Major Releases
Cons: Service disruptions	Low-High (depends on the patch)	High	Low-High (depends on implementation)	Low
Pros: Benefits from upgrade	Low	Low-High (depends on the level of service disruptions)	Low-Medium	High

¹Juniper Networks M120 Multiservice Edge Router delivers a unique solution with Forwarding Engine Board (FEB) redundancy that allows the switchover of forwarding path in just a fraction of a second.

Juniper's Unified ISSU

The engineers and developers at Juniper Networks have extended the concept of HA to develop what Juniper calls **continuous systems**. Continuous systems take a broad look at how to avert disruption and degradation of services—ensuring that the network is available, always. Continuous systems are much more than individual features, protocols, or products. During the product development cycle, Juniper engineers consider each device's redundancy, failover mechanisms, and operations to develop functions that support seamless service continuity.

As service expectations rise for IP networks, Juniper Networks has focused on developing the continuous systems that operators need to keep their networks running 24 hours a day, seven days a week. A continuous systems approach means that the Juniper engineers explore the different causes of service degradation and outages—whether resulting from planned maintenance, unexpected failures, or human factors—and find ways to avert and mitigate them for delivery of high uptime.

As part of Juniper's overall continuous systems initiative, unified ISSU takes into account availability of every hardware and software element, and approaches unified ISSU as an overarching system. It delivers unified ISSU across both major and maintenance releases without control plane disruption and with a maximum of a few seconds of traffic disruption to the system.

With Juniper's unified ISSU, the execution is a simple one-line command and Juniper Networks Junos® operating system takes care of the rest automatically — compatibility is automatically checked, route adjacencies are preserved, forwarding information is retained (and updated if necessary), and traffic is forwarded accordingly. Because of graceful Routing Engine switchover and Nonstop Routing (NSR) capabilities, neighbors do not detect the upgrade and continue to forward traffic based on local routing information.

The automatic operation streamlines Juniper's unified ISSU. For example, one of the first steps done by unified ISSU is to verify that all hardware installed in the system are supported by both the current and the new Junos OS release, and the current configuration is compatible with the new Junos OS release. These functions provide robust messaging to the operator and enable the operator to abort the operation and correct discrepancies.

Furthermore, the Juniper routing and switching platforms and Junos OS are specially engineered and designed with unified ISSU in mind. The same easy and smooth unified ISSU upgrade procedure applies to major upgrades as well as maintenance release upgrades, consistently across different Junos OS platforms.

Junos OS is the only network OS in the service provider market that provides unified ISSU support across major releases, maintenance releases, and product families. Table 2 summarizes the benefits and features of Junos OS unified ISSU and a high-level comparison against other types of network operating systems existing in the market today.

Table 2: Comparison of Junos OS Unified ISSU Against Other IP/MPLS Operating Systems

Type of Software Upgrade	Reasons for Software Upgrade	Junos OS Behavior	Platforms with Modular Operating Systems	Platforms with Monolithic Operating Systems
Major release upgrade	Adding new revenue-generating features: feature enhancements, bug fix, etc.	Yes (for example, 9.3R1 to 9.5R1)	No	No
Maintenance release upgrade	Adding feature enhancements, bug fix, etc.	Yes (for example, 9.5R1 to 9.5R2)	No (line card reset required)	No
Software patch	Bug fix	Treated as maintenance release upgrades	Yes	No
Other ISSU features				
Line card reset required		No	Yes	Yes
Service disruption during a full ISSU upgrade (across major releases) ²		A fraction of a second to a few seconds	More than 10 minutes	More than 10 minutes
Unified solution with minimal configuration and one-command execution		Yes	No	No

²Actual length of service disruption varies according to system parameters.

For those interested in more detail, please see *Appendix A* for a test scenario performed by Juniper that quantifies the exact time of a unified ISSU upgrade compared to traditional upgrades.

Conclusion

ISSU has long promised to eliminate the downtime and risks associated with standard maintenance—the networking equivalent of changing the tires and oil in a race car without making a pit stop. However, ISSU can mean different things to different network vendors. Some ISSU implementations are limited in their scope, while others eliminate only a portion of the network downtime, or introduce new risks. When planning for a software upgrade, network operators should be aware of different approaches on the market and make informed decisions based on their needs.

As the only ISSU implementation capable of upgrading from one major release to the next, Juniper’s unified ISSU for Junos OS dramatically reduces the risks and downtime of software maintenance and upgrades. For network operators to whom network downtime is not an option, Juniper Networks unified ISSU implementation provides the answer.

References

ISSU: A Planned Upgrade Tool: www.juniper.net/us/en/local/pdf/whitepapers/2000280-en.pdf

Network Operating System Evolution: www.juniper.net/us/en/local/pdf/whitepapers/2000264-en.pdf

Appendix A: Test Case: Verifying Performance of Juniper’s Unified ISSU

Figure 2 illustrates a common service provider network. Unified ISSU is performed on PE1. PE1 forms routing adjacencies with P1, P3, P4, PE2, CE1, and there is traffic running across the network.

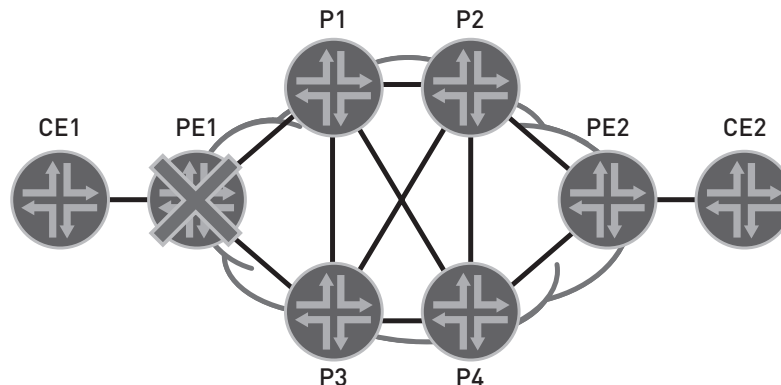


Figure 2: Lab test topology

Configuration

The following configurations are all that’s required for PE1 to be unified ISSU ready. Both graceful Routing Engine switchover and NSR need to be enabled before the upgrade takes place.

```

.....
//Enable GRES
chassis{
  redundancy{
    graceful-switchover;
  }
}
//Enable NSR
routing-options{
  nonstop-routing;
}
//Synchronizing configs
system{
  commit synchronise;
}
.....

```

One-Line Execution

Simply enter the following command to start the unified ISSU upgrade process on Junos OS:
Router>Request system software in-service-upgrade jinstall-9.5R2.7-domestic-signed.tgz<enter>

Results

Table 3 shows the result of a test for the previous scenario. In this test, there were approximately 200 VRFs and 400k routes. For comparison purposes, non-unified ISSU software upgrades were performed under the same system parameters. The comparison highlights a significant reduction of traffic disruptions during Juniper's unified ISSU upgrade.

Table 3: Test Results of Unified ISSU Versus Non-ISSU (Upgrade Path: Junos OS 9.4R1.8 to Junos OS 9.5R2.7)

	ISSU Upgrade	Traditional Software Upgrade
200 VRFs, 400k routes, 1,000 filter terms per IFL (approximately)	Less than 7 seconds	720 seconds (12 minutes)

For more lab results under different system parameters, please contact your Juniper account managers or system engineers.

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at www.juniper.net.

Corporate and Sales Headquarters

Juniper Networks, Inc.
 1194 North Mathilda Avenue
 Sunnyvale, CA 94089 USA
 Phone: 888.JUNIPER (888.586.4737)
 or 408.745.2000
 Fax: 408.745.2100
www.juniper.net

APAC Headquarters

Juniper Networks (Hong Kong)
 26/F, Cityplaza One
 1111 King's Road
 Taikoo Shing, Hong Kong
 Phone: 852.2332.3636
 Fax: 852.2574.7803

EMEA Headquarters

Juniper Networks Ireland
 Airside Business Park
 Swords, County Dublin, Ireland
 Phone: 35.31.8903.600
 EMEA Sales: 00800.4586.4737
 Fax: 35.31.8903.601

To purchase Juniper Networks solutions, please contact your Juniper Networks representative at 1-866-298-6428 or authorized reseller.

Copyright 2010 Juniper Networks, Inc. All rights reserved. Juniper Networks, the Juniper Networks logo, Junos, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.