

CONTINUOUS SYSTEMS, NONSTOP OPERATIONS WITH JUNOS SOFTWARE

Optimizing and Upgrading High Availability Systems

Table of Contents

Executive Summary	4
Introduction	4
A Scarcity of Resources	5
What Causes Network Downtime?	5
Preventing System Errors	6
Routing and Forwarding on Separate Planes	6
Modular Software	7
A Single Code Base	7
Building Nonstop Platforms	8
Graceful Routing Engine Switchover	8
Graceful Restart Protocol Extensions	8
Nonstop Active Routing	9
Support for Bidirectional Forwarding Detection	10
Additional Redundancy through VRRP	10
Virtual Chassis for EX Series Switches	10
J Series Chassis Clusters	11
Chassis Cluster Formation	11
Security and Performance Factors	11
Protection Against Attacks	11
Shortening and Eliminating Planned Outages	12
Hot Swappable Interfaces	12
Faster Upgrades, Reduced Risk	12
Unified In-Service Software Upgrades	13
Unified ISSU Methodology	13
System Requirements	13
Automated Compliance Checks	13
Unplanned Outages: The Human Factor	14
A Configuration Safety Net	14
A Scarcity of Tools	15
The Key to Prevention: JUNOScript Automation	15
Scripting and XML	16
Commit Scripts	16
Device-level Configuration Support	17
Comprehensive Macro Capabilities	17
Operation Scripts	18
Event Policies	18
Incremental Benefits of Scripting	18

Preventive Management	19
The JUNOS OAM Implementation	19
Built-In Diagnostics and Troubleshooting	19
Traceoptions for Protocol Debugging	19
Packet Capture	20
Real-Time Performance Monitoring	20
Advanced Insight Solutions	21
Embedding JTAC Experience into JUNOS	21
Proactive and Reactive Services	21
AIS Internals	21
The Customer Controls the Data	22
What About Security?	22
Intelligence Driven Analysis	22
Incident Driven Analysis	22
Pluses for the Customer	22
Conclusion	22
References	23
About Juniper Networks	24

Table of Figures

Figure 1: Causes of network downtime	5
Figure 2: Separate control and forwarding planes	6
Figure 4: One predictable release train	7
Figure 3: Modular OS components	7
Figure 5: The JUNOS commit model	14
Figure 6: Traceoptions provides a wide range of variables for observing network and system events	20

Executive Summary

The audiences for this paper are network designers and operators in Service Providers and Enterprises. High availability (HA) represents one of the most difficult challenges facing today's high-performance networks, and as a result, HA remains at the heart of the engineering philosophy at Juniper Networks®. Juniper calls its approach to HA "continuous systems", reflecting the company's overall goal to provide systems that do not disrupt or degrade service. This goal requires that Juniper developers and engineers consider all potential sources of downtime and find ways to provide fail-safe mechanisms. They also need to anticipate that downtime can never be completely eliminated, and thus also provide for rapid recovery, and, ideally, find ways to proactively prevent downtime altogether.

Continuous systems represents a significant milestone for Juniper Networks—a unique, holistic approach that offers one of a kind autoscripting abilities, significant improvements in network instrumentation, and exceptional remediation techniques for dealing with network problems. New capabilities introduced as part of Juniper Networks continuous systems include nonstop active routing (NSR), which relies on the modular, fault-tolerant Juniper Networks JUNOS® Software, a network operating system that provides:

- Uninterrupted routing and forwarding
- Unified in-service software upgrade (unified ISSU), which makes it possible to replace an entire operating system without an interruption in routing
- JUNOScript Automation, a powerful set of scripts for on-box problem detection and resolution
- Advanced Insight Solutions (AIS), a suite of tools that embeds Juniper engineering expertise directly into routers and other network devices

Continuous systems is a solution that scales extraordinarily well across an entire, multi-chassis network. By ensuring continuity of services, Juniper Networks raises the bar for the telecom industry and revolutionizes the process of detecting, diagnosing, and repairing network problems.

Introduction

IP networking has transformed the telecommunications industry over the last decade. Global populations now have the power to access, create, and distribute information in ways undreamed of when the public switched telephone network (PSTN) dominated interpersonal communication. Internet connectivity is now a given for businesses, and the future of e-commerce is assured. Yet with all of these changes, the telecom network still sets the standard for reliability in many people's minds.



With continuous systems, Juniper Networks meets this challenge head-on and introduces a new standard of availability for IP networking—self-aware network devices that are always alert to potential issues and always ready to initiate repair.

Considering the critical role that IP plays in today's networks, Juniper Networks had a significant advantage when the company developed and deployed its first systems: TCP/IP was already a global standard. In fact, when JUNOS was introduced in 1998, it was the first network operating system designed specifically for IP routing. Ever since those early days, Juniper has maintained an impeccable reputation for reliability and innovation. Stable service and support are the touchstones of Juniper engineering.

With the advent of continuous systems, Juniper raises the bar for IP/MPLS networking, and introduces software, equipment, and services that keep networks up and running 24 hours a day, seven days a week. Juniper has devoted extensive resources to developing continuous systems so that customers can be assured of nonstop network operations, even during Routing Engine (RE) restarts and system upgrades.

At Juniper Networks, continuous systems is a culture, not a single feature, protocol, or product. Continuous systems means addressing all areas that could potentially cause service degradation and outages. It's about a complete, all-around approach that looks at potential causes of problems and finds ways to mitigate them to maintain service-level agreements (SLAs). Rather than a set of features or a box, continuous systems considers a network device's redundancy, architecture, and operations with broad functions and capabilities contributing to service continuity.

Continuous systems is first and foremost nonstop active routing, a solution that allows network devices to transparently switch REs without disrupting routing or forwarding. Next, continuous systems is unified ISSU, a feature that makes it possible to replace an entire operating system, not just selected modules or features, with no interruption in routing. Continuous systems also provides a suite of on-box automation scripts that improve problem detection times, speed up diagnosis, and take the steps needed to restore normal operations.

This paper focuses on Juniper Networks continuity of service offerings—pioneering architecture, a streamlined code base, and, with AIS, an innovative system that embeds Juniper’s expertise directly into routers and other customer devices.

A Scarcity of Resources

IT companies everywhere are facing difficult economic and technical challenges. Reducing OpEx spending is a critical goal, yet networks are expected to steadily roll out new services, increase bandwidth, and serve more and more customers—often without a corresponding increase in staff. Without adequate staffing, many organizations operate mainly in reactive mode; in some cases entire divisions are relegated to firefighting.

Dedicating so many staff to urgent issues means that there is little time left to innovate, roll out new services, or find ways to cut costs, let alone look for ways to prevent outages in the first place. At the same time, an outage can be extraordinarily expensive in terms of SLA penalties and damaged customer confidence.

This is precisely where continuous systems comes in, introducing important advances that help networks detect and diagnose network problems, and, ultimately, prevent outages altogether.

What Causes Network Downtime?

Recent industry research shows that the primary causes of network downtime are:

- Planned maintenance events (hardware and software upgrades)
- Unplanned system failures (hardware failures and software defects)
- Human error, such as configuration changes that negatively impact network performance

Most network outages are the result of human error—changes made to the network that are incorrect, mistimed, or fail to follow the appropriate workflow procedures. Depending on the study, human error is held responsible for 50 to 80 percent of network downtime. Yet many network engineers tend to overlook the human contribution to downtime, focusing instead on hardware, link, or software failures.

Hardware failures tend to be few and far between compared to software problems and human error. Mean time between failures (MTBF) for most chassis and components are quite good; it is the software that emerges as the larger contributor to downtime. For JUNOS-based systems, some system errors are internal JUNOS issues that appear, for example, when a login fails or a peering process terminates unexpectedly. But many more system errors can be attributed to human error, for the most part mistakes in configurations.

With the convergence of high demand services onto IP infrastructures, network outages of any kind are no longer acceptable. Even relatively small packet losses can have a negative effect on users’ perception of service delivery, while a major node, link, or interface failure can have serious effects for the provider. Operating systems must therefore provide network operators with tools that minimize network failures whenever possible, and minimize the effects of failures that do occur.

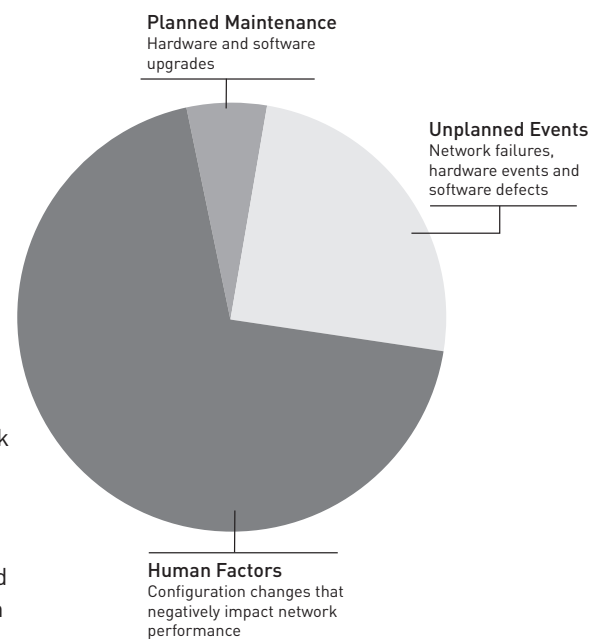


Figure 1: Causes of network downtime

Preventing System Errors

Two key features underlie JUNOS Software's unique resilience to internal failures—separation of the router's control plane and forwarding planes, and operating system modularity. Juniper Networks introduced both of these features to the routing community, and both are in widespread use today.

Routing and Forwarding on Separate Planes

Figure 2 depicts the architecture that enables continuous systems.

The need to forward packets and process routes simultaneously presents a major challenge for network routers. If the traffic load through a router becomes very heavy, most resources might be used in performing packet forwarding, causing delays in route processing and slow reactions to changes in the network topology. On the other hand, a significant change in the network topology might cause a flood of new information to the router, causing most resources to be used in performing route processing and slowing the router's packet forwarding performance.

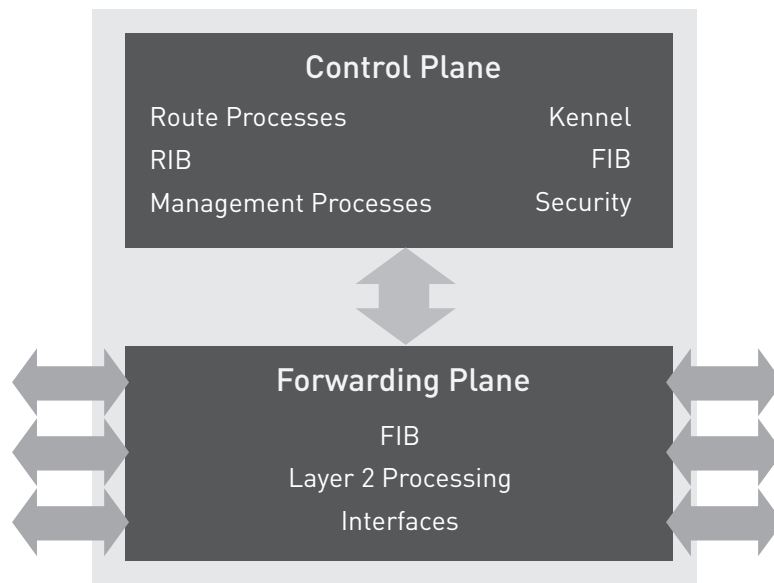


Figure 2: Separate control and forwarding planes

The key to the problem lies in internal resource allocation. If most resources are consumed by one of the two basic functions, the other function suffers and the router is destabilized. The solution is to perform these functions in separate physical entities, each with its own resources, as shown in Figure 2.

Juniper Networks pioneered modern router architecture by cleanly separating the control plane from the forwarding plane.²

The control plane is also known as the routing plane, and its primary component is the Routing Engine (RE), which is redundant in many Juniper platforms. The forwarding plane is also known as the data plane, and its primary component is the Packet Forwarding Engine (PFE). The control plane maintains peer relationships, runs routing protocols, builds the routing table, maps destination IP addresses with physical router interfaces, and builds the forwarding table, or FIB. The FIB is exported to the forwarding plane, which uses it to send packets out of the correct interface and on to the next hop router. Having a copy of the FIB in the forwarding plane makes it possible for the router to continue forwarding packets even if a software bug or routing issue causes problems in the control plane.

On Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers, the control plane runs on a separate physical RE, a host complex that is built into the router with its own processor, memory, and storage. On Juniper Networks J Series Services Routers, the control plane and forwarding plane share a single processor, memory, and storage. However, the control and forwarding planes are implemented as real-time threads, and the shared resources are allocated in such a way as to ensure that the forwarding plane is still able to forward traffic as quickly as possible.

On all platforms, the JUNOS control plane is based on a BSD kernel.

²Juniper continues leadership in this area by having taken this level of separation one step further with Juniper Networks JCS1200 Control System, an independent control plane scaling system.

Modular Software

The division of labor between control and forwarding planes has its parallel in the next essential architectural characteristic of JUNOS—its fundamental modularity. A key advantage of modularity is the inherent fault tolerance that it brings to the software. Each module of JUNOS runs in its own protected memory space and can restart independently, so one module cannot disrupt another by “scribbling” on its memory. If there is a software problem with JUNOS production code, the problem can be quickly identified, isolated, and fixed without an interruption in service. JUNOS automatically restarts failed modules without having to reboot the entire machine.

This modular design contrasts to a monolithic architecture that is built as one large set of code. In a monolithic architecture without isolation between processes, a malfunction may cause a full system crash, as one problem creates memory leaks and other issues that can impact many other processes. These problems may require a router or switch reboot to correct, putting the platform out of service during the restart.

The JUNOS architecture also yields important software engineering advantages. A reasonably small team of engineers manages the software included in each module, and the same team of engineers is responsible for the same module release after release. As a result, any addition or change to the module is very well understood in terms of how the change will affect the code.

A Single Code Base

Modularity is not the only feature that underlies JUNOS Software’s resistance to failure. The truly unique nature of JUNOS lies in its fundamental virtue—a single source base of code. Unlike other network operating systems that splinter into many different programs and images—and then just share the same name—JUNOS has remained a single, cohesive system throughout its life cycle.

Juniper Networks engineers develop each JUNOS feature only once, and then apply it to all routers, switches, and security platforms where it is required without requiring a complete overhaul of the code. As a result, each new version of JUNOS is a superset of the previous version. Customers don’t need to add separate packages when a feature is desired, but only need to enable it.

Juniper Networks methodologically enhances the single JUNOS source base through a highly disciplined development process that follows a single release train (shown in Figure 4). Developers ensure a single consistent code set for each feature, and the result is well understood, extensively tested code. The JUNOS testing process includes repeated testing with automated regression scripts. Developed over many years, these test scripts are key pieces of Juniper Networks intellectual property. Through the extensive testing of each JUNOS release, bugs and other problems are far more likely to be found and corrected by Juniper engineers before customers ever see the new version.

Because the same code runs across all Juniper Networks routers, each feature provides a common user experience on all devices. A BGP or OSPF configuration works the same way on a branch router as it does in the core of a service provider network, and also uses the same diagnostic and configuration tools. When a network rolls out on multiple Juniper platforms, a single operations team already has the knowledge required to configure and monitor all of the new devices. This kind of efficiency can significantly reduce a network’s operations expenses.

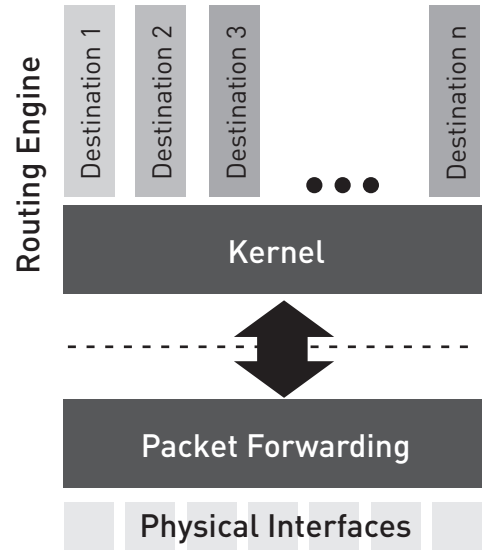


Figure 3: Modular OS components

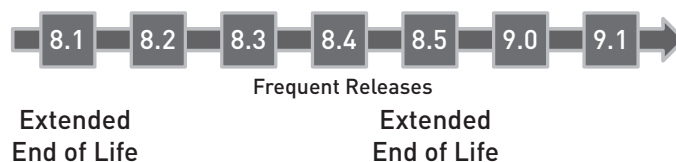
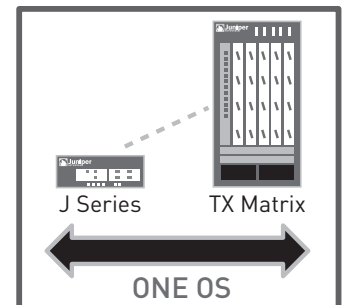


Figure 4: One predictable release train

Building Nonstop Platforms

Separation of the control and forwarding planes is as fundamental to modern router architecture as protocol layering is to the Internet. With the new architecture in place, Juniper's engineers began to work on the next milestone, ensuring that routing and forwarding could continue uninterrupted, even during control plane disruption.

Graceful Routing Engine Switchover

Most routers today make use of redundant control plane processors (Routing Engines in Juniper Networks terminology), so that if one processor fails, the other can take over router operations. Under this architecture, one RE serves as the master and the other as backup. The two REs exchange frequent "keepalive" messages to detect whether the other is alive and well. If the backup RE stops receiving keepalives after a specified interval, it takes over route processing for the master.

The limiting factor in this scenario lies in the fact that the data plane's Packet Forwarding Engine (PFE) is reinitialized during the switchover from master to backup RE. All data plane kernel and forwarding processes are restarted, and traffic is interrupted. To prevent such a disruption, control plane state information needs to be synchronized between the master and backup RE. This is where graceful Routing Engine switchover (GRES) comes in.

GRES provides stateful replication between the master and backup REs. Both REs maintain a copy of all important entities in the JUNOS kernel such as interfaces, routes, and next hops. In this way, the backup does not need to learn any new information before taking over for the master. The router's forwarding plane breaks its connection with the routing tables on the old master and connects to the new master. From the point of view of packet forwarding, switching the PFE connection from one RE to the other happens immediately, so no packet loss occurs.

This solution isn't perfect either, however, because under GRES the control plane routing protocol process restarts. Neighboring routers detect the restart and react to the event according to the specifications of each protocol. If there is an RE switchover in router X, for example, any neighboring router that has a peering session with router X sees the peering session fail. When router X's backup RE becomes active, it reestablishes the adjacency, but in the meantime the neighbor has advertised to its own neighbors that router X is no longer a valid next hop to any destination beyond it, and the neighbor routers start to look for an alternate path. When the backup RE comes online and reestablishes adjacencies, its neighbors advertise the information that router X is again available as a next hop and devices should again recalculate best paths. These events are called "routing flaps," which consume resources on the control planes of all affected routers and can be highly disruptive to network routing.

What, then, is the solution? In order to preserve routing during an RE failover, GRES must be combined with either graceful restart protocol extensions or with nonstop active routing, Juniper Networks' recommended solution.

Continuous Systems With JUNOS Software

The action: combine graceful RE switchover with graceful restart protocol extensions or nonstop active routing

The result: uninterrupted routing and forwarding

The advantage: no downtime for customers when routing problems hit the device

The benefits in customer loyalty: immeasurable

Graceful Restart Protocol Extensions

Graceful restart protocol extensions represent a solution to the flapping problem (but not necessarily the best solution.) Graceful restart is defined by the IETF in a series of RFCs, each specific to a particular routing protocol. Graceful restart specifies that if router X's control plane goes down, its neighbors don't immediately report to their own neighbors that router X is no longer available. Instead, they wait a certain amount of time (or grace period). If router X's control plane comes back up and reestablishes its peering sessions before the grace period expires, as would be the case during an instantaneous RE switchover, the temporarily broken peering sessions are not visible to the network beyond the neighbors.

With graceful restart, an RE failover is transparent to all nodes in the network with the exception of router X's peers. This is a great advantage, as there is no disruption to forwarding on router X, its peers, or any other routers across the network. There is no change in traffic patterns, and no impact on latency, packet ordering, or optimal route selection.

During the grace period, it is assumed that the node that is not routing is forwarding traffic and preserved state—often called nonstop forwarding. The graceful wait interval is configurable by the user and negotiated between the nodes. It is often several seconds long. During this graceful wait interval, the traffic is not supported by active routing, so the restarting nonstop forwarding node could potentially send traffic to a destination that is no longer valid—often called blackholing traffic.

Other issues connected with graceful restart include:

- Each neighbor is required to support the graceful restart protocol extensions.
- Graceful restart must stop if the network topology changes during the grace period.
- In some cases, it is not possible to distinguish between link failure and control plane failure.
- Routing reconvergence could exceed the grace period, for example if router X has hundreds of BGP peers or protocol interdependencies that complicate the reconvergence process.
- There is no widespread industry acceptance of graceful restart.

The requirement for all nodes to be running the graceful restart extensions is particularly bothersome in a multi-vendor, multi-chassis environment, and even more difficult when a different organization controls each peering router. In addition, during the graceful restart period, router X is not removed from the network topology, and the topology is therefore “frozen.” This means that graceful restart should only be used when the network is stable—a circumstance that is difficult to guarantee.

A better solution is required, one that is transparent to network peers, doesn’t require peer participation or allow adjacencies or sessions to drop, and has a minimal impact on convergence. The RE switchover should also be allowable at any point, no matter how much routing is in flux.

Nonstop Active Routing

Juniper’s solution to the problems presented by graceful restart is known as nonstop active routing (NSR). This phrasing may be familiar to users, as Juniper is not the only router vendor that has caught the “nonstop routing bug.” However, Juniper has implemented something radically new and innovative. Juniper engineers define “nonstop” as the integrity of control and forwarding planes in the event of failovers or system upgrades, including minor and major release changes. Routers running JUNOS are not going to miss or delay any routing updates when network problems occur. The goal of a nonstop operation is very ambitious, yet truly reflects Juniper’s innovation and expertise as it introduces these new concepts and this new vision for the industry.

With NSR, the responsibility for repairing a failed RE is placed entirely on the router itself. There is no need to modify or extend existing routing protocols or place any demands on peers. NSR uses the same infrastructure as GRES to preserve interface and kernel information. However, NSR also preserves routing information and protocol sessions by running the routing protocol process on both REs. In addition, nonstop active routing preserves TCP connections maintained in the kernel.

From a system architecture point of view, the principle difference between NSR and the graceful restart protocol extensions is that both REs are fully active in processing protocol sessions. Both REs are running the routing processes and receiving routing messages from network neighbors. Selection of the master is now a matter of selecting one of two running REs and connecting its outbound message queue to the network to communicate with neighbors. NSR is self-contained and does not rely on helper routers (as in graceful restart) to assist the routing platform in restoring routing protocol information.

Network-Level Resiliency

Juniper Networks supports path resiliency through robust implementations of switching and routing protocols, including Rapid Spanning Tree Protocol (RSTP), OSPF, BGP, and IS-IS. Standard Layer 3 protocols such as OSPF provide the fastest recovery from link failures and are more scalable than Layer 2 protocols. To improve further on Layer 3 protocol convergence times, Juniper supports bidirectional forwarding detection (BFD), which provides rapid detection of link, interface, tunnel, and peer failures, resulting in continuous network operations.

Nonstop bridging and routing mechanisms enhance the resiliency characteristics of network protocols by preventing service interruptions during the brief period when the backup Routing Engine takes over for a failed RE. Left to their own devices, the absence of the master RE would cause routing and switching protocols to begin the process of reconverging network paths to route around what they believe to be a failed device. The Juniper nonstop routing and nonstop bridging protocols prevent such a reconvergence from occurring, thus maintaining service continuity.

Nonstop bridging extends these benefits to the Layer 2 protocols implemented in Ethernet switching. Together these features enable RE switchover that is transparent to neighbors, maintaining Layer 2 and Layer 3 stability for supported platforms and protocols.

Because nonstop active routing does not disrupt protocol adjacencies, the RE switchover is transparent to neighbors. Even if the routing topology changes during the switchover, routing remains stable.

Implementing nonstop active routing was a non-trivial task at Juniper Networks. A project of such complexity required months of intensive work by engineers across all divisions of the company. The result is a contribution to the field of telecommunications that is unique and eminently scalable. With nonstop active routing replacing the graceful restart protocol extensions, Juniper sets yet another routing industry standard that will be in place for many years to come.

Nonstop active routing also makes it possible for Juniper to launch important advances such as unified in-service software upgrades. See the section titled "Shortening and Eliminating Planned Outages" for details.

Support for Bidirectional Forwarding Detection

The BFD protocol, which was developed at Juniper Networks, is a simple, high-speed "hello" protocol that verifies connectivity between pairs of systems. BFD neighbor systems negotiate a peer relationship, and each neighbor specifies how quickly it can receive BFD packets. BFD rates can be specified in sub-millisecond increments.

BFD is extraordinarily useful because it places a minimal load on network devices, markedly improves failure detection times, and reduces latency within the router and between the router and its neighbors. With nonstop active routing enabled, BFD session state is saved on both the master and backup Routing Engine. When an RE switchover occurs, BFD session state does not need to be restarted and peer routers continue to interact with the routing platform as if no change had occurred.

Additional Redundancy through VRRP

New redundancy features for Juniper Networks EX Series Ethernet Switches and J Series Services Routers greatly enhance continuous systems capabilities for both platforms. The advances are based on the Virtual Router Redundancy Protocol (VRRP), an IETF standard³ that prevents a router or switch from serving as a single point of failure on a network.

Virtual Chassis for EX Series Switches

Juniper Networks EX Series Ethernet Switches with Virtual Chassis technology deliver the high availability, high port density, and low power/cooling requirements of a modular chassis in a stackable, cost-effective, scalable platform. The innovative switches feature a 128 Gbps virtual backplane that allows up to 10 instances of the Juniper Networks EX4200 Ethernet Switch to be interconnected and managed as a single unit. A single virtual switch can span multiple wiring closets, floors, or even data center server racks, greatly reducing recurring management and maintenance costs.

Virtual Chassis technology is based on the NetScreen Redundancy Protocol (NSRP), which in turn is based on VRRP. NSRP uses redundant physical connections to integrate redundant security systems into a high availability network. At the device level, redundant components such as power supplies, fan trays, control modules, interface cards, and switch fabrics can eliminate the most common causes of hardware failure. In an EX Series switch, these physical components are field replaceable and hot swappable, so any component that goes down fails over to the backup component automatically and seamlessly.

EX Series switches were designed to support redundant REs. Each switch in a Virtual Chassis configuration has a Routing Engine. When two or more EX Series switches are deployed using Virtual Chassis technology, they offer the same RE redundancy features as any Juniper chassis-based switch or router, including GRES for hitless RE failover.

In a Virtual Chassis implementation with two or more EX4200s, JUNOS selects one switch's RE as the master and a second switch's RE as the backup in hot-standby mode. The remaining switches in the Virtual Chassis serve as line cards only, ready to be selected as the backup RE if the master RE should fail. Operators can selectively prioritize REs to assign master and backup status, as well as determine the order in which the remaining switches will ascend if the master and backup fail to ensure seamless and immediate failover.

³See RFC 3768 "Virtual Router Redundancy Protocol", www.ietf.org/rfc/rfc3768.txt

J Series Chassis Clusters

Juniper provides redundancy for J Series Services Routers by grouping two routers into a cluster. Both routers must be running JUNOS Software with enhanced services. The two nodes use VRRP to back each other up, ensuring stateful failover of processes and services in the event of system or hardware failure. If the master node fails, the backup takes over traffic processing. Nodes in a cluster are interconnected over Gigabit Ethernet links and synchronize configuration, kernel, and session state to facilitate high availability of interfaces and services. Chassis clustering provides:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple PFEs that presents a “single router” view of the cluster
- Synchronization of configuration and dynamic runtime states between nodes within a cluster
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold

Chassis Cluster Formation

To form a chassis cluster, two of the same kind of systems—either two instances of Juniper Networks J6350 Services Router, Juniper Networks J4350 Services Router, Juniper Networks J2350 Services Router, or Juniper Networks J2320 Services Router—combine to act as a single system that enforces the same overall security. Although the J Series router pairs must both be the same kind, they can contain different Physical Interface Modules (PIMs). When a J Series router joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration. Up to 15 chassis clusters can be deployed in each environment.

Security and Performance Factors

In order to provide continuity of service, networks require a foundation of stable hardware and software. JUNOS Software users capitalize on its exceptional stability, fault tolerance, and engineering discipline to set new standards for efficiency. With such a solid infrastructure in place, JUNOS-based platforms have the means to provide security and performance enhancements unmatched in today’s marketplace.

Ensuring the security of IP networks is a complex undertaking. Attacks are now a given across the Internet, but responding to vulnerabilities can sometimes create new problems, and adjustments can lead to networks with more holes than barriers. Too often, security and performance are seen as competing aspects of networking, with security improvements coming at the cost of degraded performance. In today’s environment, this is no longer an acceptable option.

Enterprises and service providers need infrastructure equipment with line rate packet filtering capable of tens of thousands of entries, and they must be able to dynamically augment and propagate these filters during an attack. Networks also need traffic shaping capabilities that include the ability to rate-limit traffic and place it into prioritized queues on both the control plane and data plane. IP/MPLS networks will find that JUNOS is extraordinarily effective in meeting all of these needs.

Protection Against Attacks

Most router attacks are directed against one of the routing protocols or the network OS itself. That means that attacks must enter at the forwarding plane and make their way up to the control plane. The link between these two entities, then, serves as a “choke point” at which malicious packets can be identified and stopped. JUNOS Software uses complex packet filtering to allow only specifically permitted packets to enter the control plane; all others are blocked. Discarding bad packets en route to the control processor helps prevent any degradation in performance. Using multiple prioritized queues to the control processor can make a significant difference here, as attack traffic can then be quarantined in a separate queue. This approach enables legitimate management traffic to get through, while blocking attack traffic. Rate-limiting capabilities ensure that essential traffic permitted through the filters, such as Internet Control Message Protocol (ICMP) packets, cannot be exploited for flooding attacks.⁴

If a distributed denial of service (DDoS) attack is in progress against a network node or transiting the network toward its target, JUNOS helps the operator trace the attack traffic to its entry points, where specific filters or rate limiters can be enabled to stop or alleviate the intrusion. Source address verification using unicast reverse path forwarding (unicast RPF) also helps reduce exposure to DDoS attacks.

⁴See “Best Practices for Securing Service Provider Networks” at www.juniper.net/us/en/local/pdf/whitepapers/2000180-en.pdf

Efficient Operations at the Philadelphia Stock Exchange

When the Philadelphia Stock Exchange (PHLX) decided to upgrade its network in 2007, the goal was to move to an infrastructure that captured increased trading volume quickly, reliably, and efficiently. Maximum transaction speed was vital. The PHLX wanted to be able to disseminate real-time market data and report quotes and trades faster to the national markets, while continually meeting the demands for increased capacity to keep up with the exponential growth of the industry's real-time quote feeds.

The upgrade's effect on operations at the exchange was remarkable. According to a study conducted by Lake Partners Strategy Consultants, after the upgrade the PHLX operations staff:

- Spent 75% less time upgrading
- Saw a 72% reduction in frequency of unplanned incidents (actual outages are rare)
- Saw a 75% reduction in time spent troubleshooting

In the words of a PHLX senior executive, "This is more than a new network. It's the foundation upon which we're meeting our customers' needs and building the Exchange of the future."

The ability to add filters under attack means that network managers can quickly modify policies to drop or rate-limit traffic that fits a particular profile. Dynamic filtering also preserves access to the command-line interface. Since the control and forwarding planes are separated, the CLI (control plane) doesn't lock up when interfaces (forwarding plane) are being hit by attack packets. Without this functionality, users would not be able to get into the CLI to make changes, and would need to reboot their systems.

Further, there are no open windows of vulnerability while configuration changes are processed line by line, as in older operating systems. In JUNOS, changes must be verified and committed before they are executed, and the changes are collected in a batch and executed as an atomic whole.

Thousands of complex filters work at line rate in routers running JUNOS. Nested filtering, which is the ability to configure a filter within a filter so that it can be reused in several places, saves substantial amounts of memory. Chained filtering makes it possible to implement specific filters one after another, in a technique known as forwarding table filtering. Both of these capabilities are unique to Juniper Networks routers.⁵

Shortening and Eliminating Planned Outages

The days when IP networks could schedule downtime for maintenance or upgrades without losing customers may very well be numbered. Modern service level guarantees and "five nines" uptime requirements preclude the traditional practice of taking router operations offline. In some cases, maintenance window provisions in RFPs have disappeared altogether. Globalization is also a significant factor—with multiple customers and teams working around the clock, there are no off-peak traffic periods for the always-on network. The bottom line is this. Modern network operating systems must enable in-service router changes and upgrades.

Hot Swappable Interfaces

The routing community took the first step towards facilitating unified in-service software upgrade (unified ISSU) when vendors introduced hot swappable interfaces for network devices. Many routers, including all of those in Juniper's product line, no longer need to be reset in order to insert or remove an interface card. Instead, the box dynamically recognizes the new interface and begins to communicate with it immediately. New components can thus be inserted and removed from the router without taking the system down.

Faster Upgrades, Reduced Risk

JUNOS is renowned in the telecom industry for ensuring reliable and predictable upgrades. The reason is twofold—the JUNOS code base, which uses a single build for all applications and all platforms, and Juniper's strict principles of engineering discipline. Juniper engineers adhere to high standards of development to maintain the single train model.

- New features can only be added to the software main line—never to bug fix releases—ensuring stability from one revision to the next.
- No back porting of features is allowed.
- There are no "customer specials." All features requested by all customers are developed and released in the mainline code.

⁵See "Efficient Scaling for Multiservice Networks" at www.juniper.net/us/en/local/pdf/whitepapers/2000207-en.pdf

Following these rules means that at all times Juniper developers are working with only a single source base of code at any release. The result is well understood code, with new features and changes carefully tested for correct integration. For Juniper's customers, this means superior reliability.

The discipline of the JUNOS development process also enables the delivery of dozens of new features in each release in a highly repeatable way, year after year. Juniper Networks customers can confidently plan the resources and activities required to upgrade to new versions, with most customers upgrading at least once a year. JUNOS users have confidence in the reliability and predictable behavior of the software and consider such upgrades a routine maintenance task rather than a high risk, time-consuming network project. In fact, customers routinely deploy the first shipping version of each new release, and every JUNOS release has been on time since the first distribution in 1998.

Unified In-Service Software Upgrades

The ability to provide unified ISSU—replacement of an entire operating system without a planned outage—is unique to devices running JUNOS. It is worth repeating that Juniper Networks customers upgrade a *complete operating system*, not just individual subsystems, without control plane disruption and with minimal disruption of traffic. This is a complex operation that requires extensive software changes, from the control plane code to microcode running on the forwarding cards.

An upgrade of this kind would be impossible for users of other systems, who are forced to juggle multiple release trains and software versions in planning each upgrade. Careful planning and testing are required to choose the right release—one that includes the new functions but does not forego any existing feature or hardware support. Also, only JUNOS provides an automatic configuration check before the upgrade. With other solutions, users are mostly notified of an inconsistent, unsupported configuration after the upgrade, when it's too late to abort.

Due to these risks, many IT groups avoid upgrading their software outside of feature or hardware additions and continue to run old versions of code, limiting their options when the network must support new requirements. Additionally, planning with these systems is very difficult if users are waiting for new features. Related uncertainty can create havoc with quarterly budgets and project resources.

Unified ISSU Methodology

Considering the immense variation among today's IP network topologies, equipment, and services, it's not surprising that the various router vendors have taken different approaches to unified ISSU. The right approach is one that addresses the practical problems in today's networks and has the flexibility to meet the needs of networks of the future. Though the unified ISSU process is complex and approaches to the problem vary, operators have two major goals for the procedure.

- Maintain protocol adjacencies—a broken adjacency makes it necessary to recalculate routing paths. Tens of thousands of protocol adjacencies must be reset and reinitiated, and as many as a million routes removed, reinstalled, and processed in order to establish network-wide forwarding paths.
- Meet SLA requirements—an upgrade mechanism shouldn't affect network topology or interrupt network services. Noticeable packet loss, delay, and jitter can be extraordinarily expensive in terms of SLA penalties and damaged customer confidence.

Unified ISSU accomplishes these goals by leveraging nonstop active routing, which eliminates routing disruptions so that Layer 2/3 adjacencies can stay alive, and minimizes packet loss to meet the requirements of SLAs.

System Requirements

Unified ISSU is available on supported, dual-RE router platforms, and upgrade paths are available from any supported release to another.⁶ For example, customers can use unified ISSU today to migrate their Juniper Networks T640 Core Router from JUNOS 9.0 to JUNOS 9.1. Unified ISSU uses NSR, thus replacing the graceful restart protocol extensions, and GRES is required for the upgrade process.

Automated Compliance Checks

Automated checking saves operators significant time and is more accurate than performing manual compliance checks. One of the first processes carried out by unified ISSU is to verify that all hardware and configured features are supported by the currently installed JUNOS release and are unified ISSU-compatible. This is followed by the same check for the new JUNOS release. Subsequent steps depend on the degree of compatibility.

⁶See the Juniper Networks *High Availability Configuration Guide*, at www.juniper.net/techpubs/software/junos/junos92/swconfig-high-availability/about-this-guide.html#preface

- System elements supported by unified ISSU are updated in-service with no or minimal disruption to traffic.
- System elements that are supported by both JUNOS releases but that do not support in-service upgrade are automatically managed by unified ISSU. Unsupported elements are taken offline by the unified ISSU process, updated to the new version while offline, and rebooted to bring them online when unified ISSU is complete. The procedure does not require operator intervention or support.
- System elements that are not supported by either the new or old version of JUNOS, such as new line cards, must be removed from the system before running unified ISSU, then reinstalled after the upgrade.

With these capabilities, unified ISSU is the fastest, most reliable in-service upgrade method available today—one that represents an important step forward for IP networking.

Unplanned Outages: The Human Factor

How much do network outages cost in today’s global environment? In a 2007 survey of Ziff-Davis enterprise customers, the typical respondent placed a value of \$3 million on each day of downtime.⁷ With stakes this high, networks must find ways to mitigate human error—the inadvertent operational mistakes that studies show cause most network downtime.

Networking vendors have historically left human error issues to their customers, offering only training and documentation to help users cope when a problem comes up in the network. Juniper Networks, by way of contrast, has maintained a longstanding focus on the human factors aspects of operations by simplifying and automating key processes that can be prone to human error.

A Configuration Safety Net

Juniper engineers have built configuration error mitigation into JUNOS since its first release. Even so, mistakes are inevitable when making changes to a network system. Up to 60% of device configuration does, after all, involve complex, repetitive, manual data entry, where typos can easily occur.

It is here that the difference between the JUNOS configuration model and competitive systems really stands out. JUNOS Software offers the ability to test policy configurations offline before they are applied to the live network. When operators begin to configure a router, they are actually viewing and changing a file called the candidate configuration. The router does not implement the changes added to the candidate configuration until the user commits them and activates the configuration on the router. This candidate “scratchpad” makes it possible to modify system settings without making operational changes to the current operating configuration and causing potential damage to current network operations.

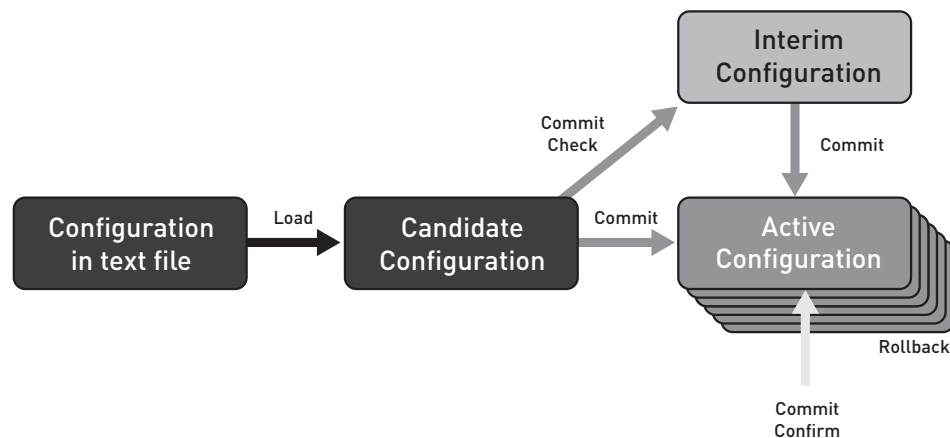


Figure 5: The JUNOS commit model

As an additional safety measure, an optional *confirm* function requires that the user enter confirmation of configuration changes within a defined time frame of activation, or the system will revert to the previous configuration. This prevents unintended or incomplete configuration changes from isolating remotely managed devices. Further, if an activated configuration degrades operations, the *rollback* command quickly restores any of the 50 previous configurations. Rolling back to a previous configuration is much faster than undoing many individual commands.

⁷Refer to Ziff-Davis webcast, “JUNOS in the Enterprise Network.” 2007

Juniper Networks J Series Services Routers have a hardware “rescue button” that onsite staff can engage to roll back to a known, working configuration—without a system reboot—eliminating the need for a truck roll when systems become isolated.

All of these features have been available in JUNOS for many years, and they remain for the most part exclusive to JUNOS. Similar features have only recently been introduced to the market, and for the largest vendor, those features are limited to an operating system unused by the great majority of its customer base.

A Scarcity of Tools

Today’s network providers are offering increasingly complex services with a limited set of financial and human resources. It is no wonder that operations teams are stretched to the maximum, with each individual feeling immense pressure to keep the system running smoothly. Under these circumstances, operators need access to tools and techniques that expedite everyday tasks, provide advance warning of potential problems, and remediate problems when they do occur.

Until now, very few tools have been available to help operators deal with configuration mistakes and other errors. Network management system (NMS) tools are widely used to perform configuration checks, but these tools tend to be reactive in nature. What is actually needed are proactive tools that prevent configuration errors from happening in the first place. Operations teams also need ways to avoid repeated mistakes typically the errors that a network experiences over the years made over and over again by new operations staff members.

Tools are also needed to improve response times for addressing system issues. A link failure should be detected immediately. When operators spot a particular set of system conditions that could lead to an error, they need to be able to step in immediately and take corrective action. These are still reactive processes, but with faster diagnosis, operators have the time to make informed decisions about resolving network problems.

Enterprises and carriers, in turn, need a way to enforce their own set of network best practices, rules, and policies, so that common operational standards are in place across the entire business infrastructure. Operational guidelines can guarantee that settings are uniform across media, interfaces are configured with the appropriate protocols, and appropriate routing parameters are in effect for the entire network. With across-the-board security procedures, IT organizations can ensure that each device has been equipped for filtering and attack mitigation. Each time a network prevents an outage through compliance with its standard rule set, the total cost of ownership is lowered substantially.

The Key to Prevention: JUNOScript Automation

With JUNOScript Automation, Juniper Networks introduces a remarkable advance for network operators: *the ability to provide native, device-level support for predefined operating procedures*. The new JUNOS scripting framework embeds intelligence directly into network devices by means of customized, on-box scripts developed by each network’s engineers. When detection, diagnosis, and repair are handled on board a network device itself, problem solving accelerates enormously.

JUNOScript Automation’s Commit Scripts, Operation Scripts (Op Scripts), and Event Policies are unique, in that they reflect each customer’s business needs and procedures. By using the scripts, IT organizations automate their best practices for finding and proactively resolving issues from the first, leading indicators of problems. And operations teams can nail down custom-defined design rules, ensuring that general business policies are enforced throughout the network.

A Commit Script, for example, could ensure that each router configuration limits the number of VLANs on customer-facing Ethernet interfaces to value X. An Operation Script could specify that no more than 1,000 permanent virtual connections (PVCs) should be connected to each ATM interface. Automated configuration and performance guarantees of this kind are invaluable, as they reduce the number of times that an operator needs to touch a network device and the likelihood of operator error. JUNOScript Automation is a feature set with no equivalent in any other router operating system.

Advantages of JUNOScript Automation

Commit Scripts: automated safeguards for error free configuration

Operation Scripts and Event Policies: fast, accurate problem diagnosis and repair

Read more about Op Scripts and Event Policies in the "*New Approach to Network*"

Scripting and XML

JUNOScript Automation owes its exceptional usefulness and agility to Juniper's aggressive stance towards adopting and promoting open standards. The new scripting environment relies heavily on XML (eXtensible Markup Language), an international standard for sharing structured data across information systems. XML is an ideal platform for network management, as it can encapsulate the complex, hierarchical, and volatile configuration of any networked device.

XML uses straightforward, HTML-like tags to represent management data. Data with a tag labeled <input-bytes> has an obvious meaning, for example. XML output from one application can easily be used as input to another. Obtaining datasets in a form that can be handed directly to other applications means freedom from writing and supporting custom code for multiple, diverse operations tasks.

XML's plain-text format and simple syntax greatly facilitate functions such as comparing configuration files, developing scripts, and accessing router state information. The ease of use and reliability of the XML programming interface facilitates the process of creating, testing, and rolling out revenue bearing services such as MPLS VPNs, tiered services, and dedicated access.

XML pervades JUNOS Software. All commands entered on a Juniper device are encapsulated in XML, and XML is used to encapsulate all information passed between different modules of the operating system. Whenever a network management tool issues a command or polls information from a device, it does so in an XML format. XML and SLAX, the XML style sheet optimized by Juniper Networks, provide an ideal environment for developing scripts that monitor devices, configurations, and network events.

Commit Scripts

With the introduction of JUNOScript Automation's Commit Scripts, Juniper takes another step forward towards the goal of ensuring error free network configurations. Commit Scripts are automated sequences of commands that inspect each committed candidate configuration to ensure that it complies with the network's standard operating procedures. If the Commit Script uncovers a noncompliant configuration, it can produce a warning message, prevent the configuration from becoming active, or even correct the configuration on the fly.

Commit Scripts are generally written by each organization's most experienced engineers—those who can flag potential errors in basic configuration elements such as interfaces, peering, and VPNs. By developing a library of custom scripts over time, IT organizations can ensure that configurations are error free and 100% in compliance with their own network rules and standards. For example, a Commit Script could safeguard routing by preventing an interior gateway protocol (IGP) from using an import policy that accidentally imports the full routing table, or it could maximize interface density by ensuring that the appropriate number of channels is configured on each channelized interface.

If the Commit Script uncovers an error and changes the system configuration, the changed configuration is automatically loaded, standard validation checks are performed, and the code is verified for correct syntax, as are statements already present in the configuration before the script was applied. A basic set of required variables can be extended to a full, complex configuration that is triggered and becomes the active, operational routing platform configuration.

Internet service providers (ISPs) often use Commit Scripts to prevent provisioning errors. For example, assume that an ISP runs OSPF on all network-facing interfaces but provisions static routes on all customer-facing interfaces. That ISP might want to deploy a Commit Script that issues a warning if the customer's activation staff accidentally deletes an interface that is running OSPF⁸. Providers can also use Commit Scripts to decrease the amount of manual configuration it takes to enable MPLS, add a default encapsulation type, or control a dual RE configuration.

⁸See Bonica and Krishnaswami "User Interface Intelligence for Routers" at <http://cabernet.cs.princeton.edu/presto07/agenda.html>

Power to the People

Much of a network's success depends on the degree of control that engineers have over their own network devices, the network's ability to respond to new business opportunities, and the cost-effectiveness of providing new and ongoing services. Networks can only meet today's economic challenges by leveraging the efficiency and ease of use of well designed control systems in network operations. Purpose-built tools are needed in order to efficiently provision and manage IP/MPLS services.

For this reason, Juniper Networks continues to invest in making available comprehensive router instrumentation and standards-based XML applications to make it easier for customers to generate and gather network management information and control their own routers. With JUNOScript Automation, Juniper provides access directly into the JUNOS Internet software. The JUNOScript interface allows customers to create solution-oriented applications quickly and cleanly. By providing standards-based access to the rich set of Juniper Networks Internet backbone router instrumentation and control mechanisms, Juniper is making development of advanced management software as simple and efficient as it can be.

By giving developers the tools for effective interaction with our routers, Juniper is cultivating the development of a set of focused, multi-vendor applications that integrate into the business models of the service provider community, thereby allowing customers to focus on delivering profitable, next-generation network services.

Device-level Configuration Support

Commit Scripts extend the expertise of top engineers to all operations personnel while bringing a greater degree of order and meaning to the network. Here are some examples of everyday tasks that can be automated using Commit Scripts:

- **Link scaling:** ensure that SONET/SDH interfaces are configured for the correct maximum transmission unit (MTU) size. This parameter is one of the most common causes of network problems and application performance issues, because the default MTU value varies among different types of physical interfaces and different vendors.
- **Port management:** ensure that ports are configured correctly, either for customer connections or core connectivity.
- **Cross-protocol checks:** make certain that every core-facing interface enabling ISO protocols also enables MPLS.
- **Implementation consistency:** ensure that all customer-facing interfaces are located on customer dedicated concentrators and slots rather than core-facing concentrators. This design rule guards against customer outages during system maintenance.
- **Security:** make certain that all public exchange point peers use Message Digest 5 (MD5) encryption.

All of these capabilities raise the level of intelligence of the router, allowing it to enforce crisp decisions regarding configuration.

Comprehensive Macro Capabilities

The true power of Commit Scripts lies in their macro capabilities. Users can create a macro that takes an input of simple configuration variables and outputs a complete configuration, guaranteeing full consistency among all network configurations. A single macro can condense a provider's specific choice of knobs, options, and default values to only a few configuration lines and variables. Such an abstracted configuration can abbreviate the code needed to set up a virtual private LAN service (VPLS) from dozens of lines to a short stanza.

Abstraction is a fundamental notion in building complex systems.⁹ Having a reduced (abstracted) view of a configuration greatly simplifies troubleshooting. When macro statements are combined with configuration scripting, configurations are easier to comprehend, easier to change, and easier to troubleshoot. There is much less chance of a configuration error as the individual elements of the configuration are only typed once. Support personnel do not have to spend valuable time investigating each individual value for a potential configuration error. Once technicians ensure that the correct stanza is applied, they can assume that the issue being experienced originates in another area of the configuration.

⁹See "MPLS Plug-and-Play for Metro Ethernet Networks" at www.juniper.net/us/en/local/pdf/whitepapers/2000241-en.pdf

The scripting/macro model lays the foundation for enabling the customer's configuration to be fully automated, potentially carving hours of diagnostic time from the support staff's day. Macros also represent an ideal way to standardize the integration of northbound interfaces with network management and operations support systems (OSS) applications.

Operation Scripts

Operation Scripts use the same software mechanisms as Commit Scripts, but focus on network monitoring and troubleshooting rather than device configuration. Op Scripts diagnose and fix problems in the network by building and running router commands, receiving and inspecting the command output, and determining the appropriate response. The process can be repeated until the source of the problem is identified and reported to the CLI.

If an Op Script detects a potential problem such as a routing failure or MTU violation, its response can range from sending notification messages and checking status indicators to shutting down low priority processes. An Operation Script can even change the router configuration to correct the problem. These are powerful capabilities.

Using an Operation Script to populate specific MIB variables for SNMP tracking, for example, enables the kind of fine grained monitoring that is not possible when generic thresholds are used throughout the entire system. If a circuit fails, an Op Script can generate a custom log with additional information about the specific problem, telling the network operations center (NOC) to check on a particular customer or link. Taking immediate corrective action gives engineers extra time to study the problem, troubleshoot multiple systems if necessary, and plan and deploy the right fix.

A typical Operation Script might initiate sequences such as:

- If CPU usage spikes above 75%, disable services x, y, and z to prevent failures.
- If VPN A goes down, run command XYZ to see if anything else is broken on the device.
- Monitor link utilization on customer facing interfaces and send an SNMP trap if latency exceeds 5 milliseconds.
- Check for label-switched paths (LSPs) to multiple destinations.
- Display Domain Name System (DNS) hostname information for router X.

Op Scripts capture the network's "brain trust"—the hands-on experience that multiple operators have gained over the years—and extends it to the entire network support staff. By pinpointing directly relevant information more quickly, scripts give operations teams more options for reacting to minor issues, rather than letting unchecked, escalating events lead to worst-case scenarios.

A key benefit of Op Scripts is their ability to iteratively narrow down the cause of network problems. Even if an Op Script doesn't immediately uncover the root cause of a problem, the script gives operators a running start that can be immensely valuable. Rapid problem diagnosis is crucial during an outage, when each minute of downtime can be extraordinarily expensive in terms of lost customer connections and transactions, SLA deductions and penalties, and damaged customer confidence.

Event Policies

Event Policies, also known as Event Scripts, come into play when multiple network events occur simultaneously. An Event Policy can correlate the events and call the appropriate Operations Script to pinpoint the issue. The scripting tools that are used to add new users to the system can also be used to automate fault detection. For example, a new interface linkup event could be used to trigger an Event Policy. The Event Policy would then call a Commit Script to ensure that the interface is properly configured.

This kind of procedure produces true plug-and-play operations, without the need to call on a manually populated OSS system or require an operator to manually log in to the device. Additionally, a successful script configuration can send automatic notifications to any number of groups within the organization to announce a new service.

Incremental Benefits of Scripting

One of the characteristics of complex systems is the cascade effect of errors, where small problems rapidly escalate into major ones. Instead of waiting for an outage that is significant enough to trip alarms and notify the network staff, JUNOScript Automation allows network engineers to set up early warning systems that not only detect emerging problems, but also take immediate steps to avert further outages and restore normal operations.

JUNOScript Automation yields incremental benefits for enterprises and providers. An initial deployment might not produce immediate results, but over time networks build a library of scripts that target problems typically seen by the staff. Once the scripts begin preventing problems, the returns increase exponentially, especially for recurring issues. IT organizations that can avoid one, two, or three outages over 5, 7, or 10 years yield high investment returns indeed.

Preventive Management

JUNOS Software provides complete visibility into the router's system health and performance. The JUNOS health monitor uses objects in the forwarding MIB to track CPU usage, wire transfer rates, and other critical system indicators. In addition to the standard Remote Network Monitoring (RMON) capabilities, JUNOS offers accounting, port mirroring, and active and passive flow monitoring. Real-time performance monitoring (RPM) probes provide information about round-trip times and delays across the network. Support for cflowd and a rich set of filtering capabilities allow passive or active flow captures based on detailed capture criteria. Port mirroring can then be used to direct the flow captures to a packet analyzer.

The JUNOS OAM Implementation

Operators across the globe rely heavily on Operation, Administration, and Maintenance (OAM) capabilities to provision and monitor their networks. Standardized OAM tools enable network operators to comply with quality-of-service (QoS) guarantees, detect anomalies before they escalate, and isolate and bypass network defects. With key OAM certifications in many areas, Juniper provides a robust OAM implementation that serves operators well. Juniper also offers multilayer OAM for the carrier edge, which brings all of the routing functionality required in core networks to edge Ethernet aggregation.

Combining the standardized OAM tools with the special capabilities of JUNOScript Automation brings operators benefits unmatched by other industry vendors.

Built-In Diagnostics and Troubleshooting

JUNOS Software's modular architecture offers a major advantage to operators debugging network problems. If an issue is uncovered in a specific module—say, for example, one of the daemons that handles routing, packet management, interface management, or security services—that module can be patched and restarted without affecting the rest of the operating system. New daemons and interfaces can be added to specific modules as needed, so partner technologies are much easier to integrate into the network.

JUNOS' extensive tracing and logging operations allow operators to monitor events that occur in the router—both normal operations and error conditions—and to track packets generated by or passed through the router. With on-box instrumentation and automated tools providing detailed state information for each modular process, system administrators can decide when to proactively restart software processes, preventing minor problems before they lead to major ones.

Traceoptions for Protocol Debugging

Traceoptions (tracing options) is the primary, and one of the most useful, troubleshooting tools available in JUNOS. *Traceoptions* allows users to fine-tune a configuration file to flag routing events such as:

- BGP state changes
- OSPF LSA (link-state advertisement) flooding, request packets, and update packets
- LDP state machine events
- IS-IS graceful restart events

When a flagged event occurs, a log entry is made to a user-specified log file on the router. The log entries can then be searched for relevant information. If a user is performing active troubleshooting, events can be displayed in real time on the console screen.

Flagging options vary according to the function under observation, and the user has the flexibility to be as general or as specific as needed. Generally, when trying to debug a problem with a routing protocol, tracing is turned on for that protocol only. Users can turn tracing on for BGP, Distance Vector Multicast Routing Protocol (DVMRP), Internet Group Management Protocol (IGMP), IS-IS, LDP, MPLS, Multicast Source Discovery Protocol (MSDP), OSPF, Pragmatic General Multicast (PGM) protocol, PIM, RIP, RIPng, RSVP, SNMP, and VPLS.

JUNOS traceoptions can be safely used without negatively affecting performance, and is well suited for production environments. Operators generally trace high-level protocol operations on an ongoing basis, so that if and when a problem arises, they can examine the resulting logs to obtain the necessary information about the source of the problem. More specific traceoptions flags are then set in the configuration file to pinpoint the problem's exact cause. Knowing how and when to use traceoptions commands is a key skill in good troubleshooting.

For the most efficient debugging, it's a good practice to set the appropriate parameters in the configuration ahead of time and leave them disabled. When the user turns on tracing, it's a simple matter to enable the trace commands needed. This is a great timesaver.

```
[edit]
ExampleUser@Juniper2# set protocols ospf traceoptions flag ?
Possible completions:
  all                          Trace everything
  database-description         Trace database description packets
  error                        Trace errored packets
  event                        Trace OSPF state machine events
  flooding                     Trace LSA flooding
  general                      Trace general events
  hello                       Trace hello packets
  ldp-synchronization         Trace synchronization between OSPF and LDP
  lsa-ack                      Trace LSA acknowledgment packets
  lsa-request                  Trace LSA request packets
  lsa-update                   Trace LSA update packets
  normal                       Trace normal events
  nsr-synchronization         Trace NSR synchronization events
  on-demand                   Trace demand circuit extensions
  packet-dump                  Dump the contents of selected packet types
  packets                      Trace all OSPF packets
  policy                       Trace policy processing
  route                       Trace routing information
  spf                          Trace SPF calculations
  state                       Trace state transitions
  task                         Trace routing protocol task processing
  timer                       Trace routing protocol timer processing
```

Figure 6: Traceoptions provides a wide range of variables for observing network and system events.

Packet Capture

Packet capture and analysis can be used to track interface accounting, determine why an application is running slowly, or why bandwidth utilization is running too high on a particular circuit. When packet capture is enabled on an interface, the entire packet is captured in the Routing Engine (RE) and stored on board in files formatted in libpcap, a system-independent interface for user-level packet capture. Packet capture files can be opened and analyzed offline with packet analyzers such as tcpdump and Ethereal.

With a Juniper Networks Services PIC (Physical Interface Card), real-time displays of traffic statistics show link status, input/output packet counts, and packet-per-second counts. Using JUNOS Software's filtering capabilities, users can specify in detail exactly what packet types to capture from which interfaces. The captured information is written to a file with a well understood format that can be read by many protocol analysis applications.

Transit traffic can be captured on systems with a services PIC, but not on systems without the PIC.

Real-Time Performance Monitoring

Real-time performance monitoring (RPM) is a necessity in today's converged IP networks, where troubleshooting relies on instant access to information about system status. When RPM is enabled, routers generate traffic between specific network elements and continuously measure network performance along those paths. If an alarm is generated for an MTU violation, for example, the problem can be isolated and addressed immediately. If the operator is taking advantage of JUNOScript Automation, an Op Script can trigger a configuration change when it receives the alarm.

JUNOS Software's RPM probes collect data on both a per-destination (end-to-end) and per-application basis. RPM probes can measure round-trip time minimums, averages, maximums, standard deviation, and jitter. The probes comprise ping (ICMP) packets as well as UDP, TCP, and HTTP packets. VPN tests are supported for all probe types. If the result of a probe or test exceeds the specified threshold, the router generates a system log message and sends any SNMP traps the user has configured. RPM probes can also discover and display common interface problems such as data framing errors. RPM records can be exported to J-Web and external NMS such as HP Performance Insight software.

Advanced Insight Solutions

Advanced Insight Solutions (AIS) is a new Juniper Networks offering that revolutionizes problem detection and resolution for IP networks. With AIS, Juniper Networks pledges to solve customers' network problems faster—or prevent them in the first place—through an advanced suite of automation tools and a tighter business partnership between customers and Juniper engineers. This new service capitalizes on the benefits of Operation Scripts and Event Policies by offering *scripts that have been customized by Juniper engineers* to detect, diagnose, and remediate router problems.

Embedding JTAC Experience into JUNOS

Most operators are all too familiar with the steps they need to take to solve a network problem: first, sort through an immense amount of data in their logs; next, determine what data should be reported to Juniper if the problem is serious; and finally, once the Juniper Technical Assistance Center (JTAC) is contacted, fill in the gaps for any information that might not have been collected initially. All of this needs to be done under tremendous pressure, as every minute of an outage costs time and money.

Imagine the benefits one would gain by skipping all of the intermediate steps to resolution—in other words, having precisely the right information about the problem on hand and ready for delivery to Juniper. This is exactly what happens with AIS. The logical steps a Juniper engineer would follow to solve a problem are encoded directly into custom scripts running on top of JUNOS. The scripts are written by those who understand Juniper's products best—Juniper engineers with extensive field experience and the ability to translate that experience into efficient, streamlined code. The AIS scripts know exactly what information to gather when specific issues arise, what steps the operator should take, and, if the customer desires, how to quickly transmit that information to Juniper for immediate attention.

AIS is optimized for problems that require JTAC involvement. The service is not triggered by routine network events such as link flaps or an adjacency loss.

Proactive and Reactive Services

AIS enables both reactive and proactive services from Juniper Networks Technical Services. With the preventive service, AIS is the software's early warning system. It gives IT organizations the complete "inside story," specifically prepared for them, about risks that could affect their network such as security vulnerabilities, hardware problems, or software bugs. Under the reactive service, AIS detects system problems in progress, analyzes them, notifies the user, and contacts JTAC for assistance if the customer desires.

AIS Internals

A full AIS deployment comprises three basic elements:

1. On-box **Advanced Insight Scripts** (AI-Scripts) are both intelligence driven for proactive analysis and services, and event driven for reactive analysis services. The scripts monitor router operations, collect the data needed to detect, diagnose, and resolve problems, and report details to the customer via SNMP traps or chassis alarms. The information is also stored locally on the router's hard drive. If the customer chooses to use the optional Advanced Insight Manager (AIM), the AI-Scripts send the AIM an encrypted Juniper Message Bundle (JMB) containing the collected data.
AI-Scripts require no fees or licensing.
2. **Advanced Insight Manager** is a standalone software application running on a Solaris or Linux server. AIM stores the JMBs in a database that is readily available for customer analysis. Advanced Insight Manager integrates readily with Juniper products such as JUNOScope and third-party network management systems.
3. If desired by the user, AIM can open a secure session with **Juniper Networks Support Systems** (JSS), located at JTAC. Through JSS, information that Juniper Advanced Services engineers use to create proactive analyses for customers is stored in an AI database. AIS incident information that demands direct attention opens a JTAC trouble case immediately.

Advanced Insight Manager is available with a free 60 day evaluation license. AIS is included in Juniper's J-Care Technical Services offerings, which are available for purchase under a service contract.

The Customer Controls the Data

One very important aspect of Advanced Insight Solutions is the extent of customer control. Users have the ability to adjust the level of information—if any—that AIM sends to Juniper. Using the information the customer is willing to share, Juniper is positioned to determine potential risks, perform migration analysis on customer requests, prepare detailed trend analyses, and make other proactive suggestions or recommendations. Having this kind of targeted information on hand empowers customers and enables them to manage their network for maximum uptime and performance.

What About Security?

The AIS system has been tightly architected to secure user data. All communication between the scripts, AIM, and JSS is encrypted and sent through secure tunnels. AIM must authenticate itself and the user before it transfers any data to Juniper, and it disconnects immediately after the data transfer. No ports are left open, so there is no entry point for attacks or compromise. Because communication is one-way and authentication is required at every step, AIS procedures may seem complicated from an operational perspective, but are simple and powerful from a security perspective.

Intelligence Driven Analysis

Under the terms of the intelligence driven (proactive) service, the AI-Scripts collect detailed information about the device's configuration parameters, status, and resource utilization. The collected data is then populated into a JMB and sent at regular intervals to AIM. Juniper Networks engineers analyze the data, proactively determine potential problem areas, and write a customer report that identifies possible risks and recommends steps towards mitigation.

Incident Driven Analysis

Under the terms of the incident driven service, if the customer experiences a problem, the AI-Scripts run through the troubleshooting logic targeted for that problem, collect all relevant information, and send it to AIM in a JMB. AIM notifies the user about any potential issues, presents the user with all relevant details about the issue, and asks for the user's approval to open a case with JSS.

Once Juniper Networks engineers start working on the case, *they have all the information they need in order to solve the problem; there is no need to contact the customer for additional details.* System administrators will receive Juniper's recommended solutions in their AIM console, so there is no need to search through lengthy log files on each network device to identify the problem and determine the proper solution. Through AIS, customer systems become smarter and more self-aware, and customers can resolve many issues encountered in their network before these turn into larger problems.

Pluses for the Customer

Advanced Insight Solutions is a unique service that scales well, and it is the natural choice for customers with several hundred devices on their network. Rather than establishing separate connections in the classical pull model, the devices only connect to the Advanced Insight Manager, and only the AIM talks to JSS. If multiple devices encounter problems, AIS allows users to display a view of all systems at once, rather than having to log into each device separately.

Conclusion

Service providers and enterprises face a challenging set of constraints in today's networking environment, with a commitment to provide new services and additional bandwidth to an increasing customer base but the need to do so with static or decreasing numbers of staff. An overextended staff increases the chance of operational errors and the likelihood of network outages that will impact services and customers.

The solution lies in the new Juniper Networks services that underlie continuous systems. JUNOScript Automation enables networks to develop custom scripts that track system problems and, if a problem occurs, take the appropriate action—that is, the action previous experience has shown to work in that particular network environment. Unified ISSU enables complete operating system installs without control plane disruption and with minimal disruption of traffic. Nonstop active routing assures users of continuous service in the face of routing problems, system errors, and control plane failures. With Advanced Insight Solutions, customers can count on the expertise of Juniper engineers not only at the end of the problem solving process, but also at the beginning, as specialized Juniper knowledge is integrated into network systems with powerful, onboard scripts.

Automation is the key to reducing the load on overburdened operations staff, and with the move to continuous systems, Juniper Networks has conceived, designed, and implemented a suite of automated tools and procedures that revolutionize operator response to system problems.

References

The following online guides provide further information about many of the topics discussed in this paper. All are linked from www.juniper.net/techpubs/software/junos/:

Advanced Insight Solutions User Guide

CLI User Guide

High Availability Configuration Guide

JUNOS Internet Software Configuration and Diagnostic Automation Guide

JUNOS Network Interfaces Configuration Guide

JUNOS Software with Enhanced Services Security Configuration Guide

JUNOS Feature Guide (advanced)

Policy Framework Configuration Guide

Routing Protocols Configuration Guide

System Basics Configuration Guide

Advanced Insight Solutions (AIS): Innovative Support Technology that Enables J-Care Technical Services.

Juniper Networks white paper, 2008.

www.juniper.net/us/en/local/pdf/whitepapers/2000270-en.pdf

Advancing the Economics of Networking: Juniper Networks Ethernet Switching Solutions Reduce Capital and Operational IT Expenses.

Juniper Networks white paper, 2008.

www.juniper.net/us/en/local/pdf/whitepapers/2000251-en.pdf

Bonica, Ron and Krishnaswami, Umesh. *Intelligent User Interfaces for Routers*. PRESTO: Workshop on Programmable Routers for the Extensible Services of Tomorrow. Princeton University, May 30-31, 2007.

Building a Highly Available Enterprise Network with Juniper Networks EX Series Switches.

Juniper Networks white paper, 2008.

www.juniper.net/us/en/local/pdf/whitepapers/2000257-en.pdf

Doyle, Jeff. *NSF, NSR, and GR*. *Network World*, "Cisco Subnet" series, June 28, 2007.

www.networkworld.com/community/doyle

Garrett, Aviva. *JUNOS Cookbook*. O'Reilly Media, Inc., 2006.

Garrett, Aviva, Drenan, Gary and Morris, Cris, and Juniper Networks. *Juniper Networks Field Guide and Reference*. Addison-Wesley, 2002.

Healy, John. *Analysis of Network Outage Reports for NRSC Meeting*. Network Technology Division–Office of Engineering and Technology (PowerPoint presentation).

Hellberg, Chris, Greene, Dylan, and Boyes, Truman. *Broadband Network Architectures: Designing and Deploying Triple-Play Services*. Prentice Hall, 2007.

Hubbert, Evelyn, Whiteley, Robert, and Batiancila, Rachel. *Who Has Changed My Network? The Discipline Of Network Change And Configuration Management*. Forrester Research, February 2007.

Increasing Network Availability with Automated Scripting.

Juniper Networks white paper, 2007.

www.juniper.net/us/en/local/pdf/whitepapers/2000252-en.pdf

The Costs of Downtime: North American Medium Businesses 2006. Infonetics Research

The Costs of Enterprise Downtime: North American Vertical Markets 2005. Infonetics Research

JUNOS Configuration Guides. www.juniper.net/techpubs/software/junos/junos84/. Titles of volumes in the series, all linked off the Web page listed above, include:

Efficient Scaling for Multiservice Networks.

Juniper Networks white paper, 2007.

www.juniper.net/us/en/local/pdf/whitepapers/2000207-en.pdf

How Operating Systems Create Network Efficiency.

Lake Partners Strategy Consultants.

www.juniper.net/us/en/local/pdf/whitepapers/lake_partners_network_efficiency.pdf

JUNOS in the Enterprise Network. Ziff-Davis webcast.

Transcript available at www.juniper.net/products_and_services/junos/junos_podcast/juniper_teleconference090507.pdf

JUNOS product documentation index. www.juniper.net/techpubs/

Kerravala, Zeus. *As the Value of Enterprise Networks Escalates, So Does the Need for Configuration Management. Enterprise Computing & Networking*. November 2004. Yankee Group, 2004.

Marschke, Doug, and Reynolds, Harry. *JUNOS Enterprise Routing: A Practical Guide to JUNOS Software and Enterprise Certification*. O'Reilly Media, 2008.

MPLS Plug-and-Play for Metro Ethernet Networks.

Juniper Networks white paper, 2007.

www.juniper.net/us/en/local/pdf/whitepapers/2000241-en.pdf

Securing Service Provider Networks.

Juniper Networks white paper, 2006.

www.juniper.net/us/en/local/pdf/whitepapers/2000180-en.pdf

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

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
Fax: 35.31.8903.601

Copyright 2009 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. JUNOSe is a trademark of Juniper Networks, Inc. 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.

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

