

Continuity of Operations Leadership Series for Government

Integrating Continuity of Operations (COOP) into the Enterprise Architecture

Integration Pillar



Table of Contents

Integration	1
Systems Integration	1
Information Systems Integration.....	1
Methods of Integration	2
Vertical Integration	2
Horizontal Integration	2
Mesh or Star Integration.....	3
Combined Integration Methods	3
Integration Process	4
Requirements Engineering	4
Data and Information Modeling	5
Functional, Physical and Logical Modeling	5
Functional Modeling.....	5
Physical Modeling.....	6
Logical Modeling.....	7
Functional, Physical and Logical Mapping.....	7
Interface Requirements and Specifications	8
Conclusion.....	8
About Juniper Networks	9

Table of Figures

Figure 1: Vertical Integration	2
Figure 2: Horizontal Integration.....	3
Figure 3: Integration Process.....	4
Figure 4: Generic Functional Decomposition	6
Figure 5: Functional, Physical and Logical Mapping	7





Integration

Integration is both a concept and an activity that is critical to the successful implementation and operations of an organization's information systems. Over the past decade, there has been a tremendous amount of effort within the IT community to develop integrated solutions within organizations - most notably, the concept of Enterprise Resource Planning (ERP). Integration is critical to successful COOP efforts if it is to become an integral part of an organization's operational profile rather than a bolt-on solution. Likewise, COOP must be integrated into the organization's culture and be practiced and exercised, regularly.

Systems Integration

Systems integration is the process of combining various component subsystems into a single system and ensuring that each of the subsystems works as a functional whole. For an organization's information system infrastructure, systems integration is the determination of data and information dependencies and the linking of the various information systems together so that organizational functions can be accomplished more efficiently. At a basic level, some forms of systems integration can be accomplished through common, dispersed technologies - for example, networks or techniques such as business process reengineering.

The objective of systems integration is not just to get systems to work together. The goal is to achieve increased capabilities by combining systems. To accomplish integration, it is necessary to identify interface points where the various systems can interact. If interfaces are compatible, systems can be directly connected. If interfaces are incompatible, then some form of conversion can be devised to translate one interface to another. In applications, these interface converters are often referred to as "wrappers." One of the responsibilities of integration engineering is the determination of the interfaces and the methods of approaches to permit the necessary interaction between systems.

From a COOP perspective, systems integration will include understanding how the interfaces or "glue" between systems remains functional during COOP operations. If elements of the organization are displaced and are performing essential functions remotely, systems integration should provide some assurance that the responsible systems are interconnected appropriately and that they have the necessary reachback for data and information, wherever they may be.

Information Systems Integration

The integration of information systems, from an organizational perspective, is actually a cultural approach to focusing on the organization's integral system of systems, rather than focusing on one system at a time in a "stovepipe" fashion. In this way, the organization can gain the greatest benefit from each system and avoid duplication of infrastructure and effort. The objective is to optimize, as much as possible, the total organizational system with the available resources. The starting point must be strategy and operations that are geared toward accomplishing the "essential functions" In this regard, integration is really focused on integrating all aspects, not just physical systems or software, but the strategy, policy, operations and systems into a working, cohesive unit that maximizes capability during normal operations and supports essential functions during COOP.

Integration must be a committed activity that continuously attempts to bring system elements together for greater capability and efficiency. In this sense, integration is an activity that combines and assimilates elements or systems into a larger system. This larger, collective entity will likely have multiple aspects. An organization's information systems should be integrated where possible and be capable of performing under various circumstances. In this sense, a Storage Area Network (SAN) may be developed for COOP but also be utilized for day-to-day operations, so that the system can be utilized, tested and stressed. This type of effort will ultimately improve the reliability and availability of the system.

Methods of Integration

There are several ways to approach the integration of systems and there are specific integration methods that must be used if systems are to work together. When systems are hierarchical in nature, they can be integrated vertically, thus creating a type of functional silo. Systems that are peer systems reside on the same level and can be integrated horizontally via some form of service platform. Systems that require complete interconnectivity with all other systems are integrated via a mesh or star approach.

Vertical Integration

Vertical integration is a process of identifying functionally related systems that are hierarchical in nature and aligning them into vertical support columns, often referred to as silos. The benefit of vertical integration is that it can be very effective in addressing short-term issues and provide very high performance. The drawback is that most silo approaches do not permit resource sharing or integration with other silos, resulting in duplication of effort and high cost of ownership. Using vertical integration also makes adding additional functionality difficult.

As shown in Figure 1, vertical integration ties functions together so that services can be provided to customers. Each silo has unique, dedicated functional elements that are not shared outside of the silo.

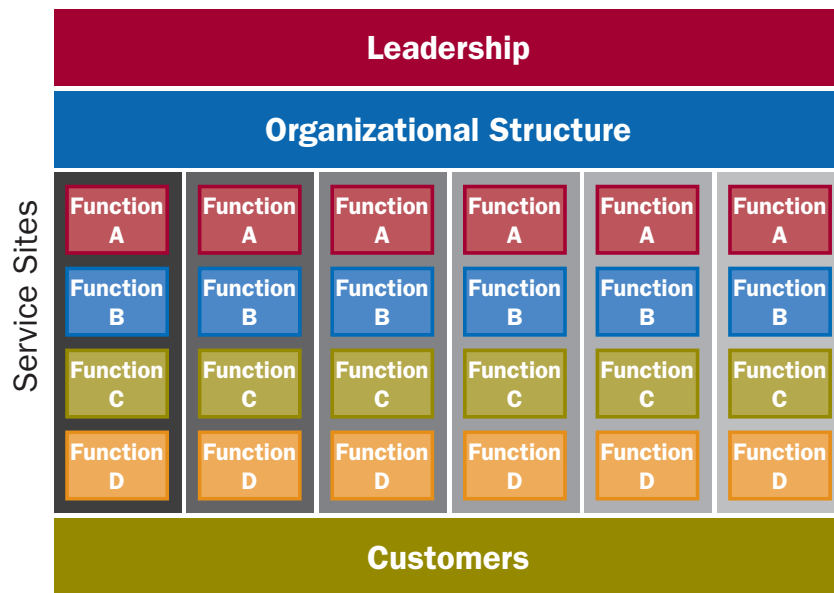


Figure 1: Vertical Integration

Horizontal Integration

Horizontal integration is the interconnectivity and interoperability of parallel functions in an effort to provide new services, improved capabilities and improved efficiency. There are several approaches to accomplishing this form of integration including establishing a bus platform or a service platform upon which various systems can interact. In either case, systems are more easily integrated when interconnections are addressed in the requirements phase of system development. Otherwise, integration will require new interfaces or wrappers that will permit systems to interact.

A bus is a dedicated subsystem that performs the job of integrating the various information systems that need to communicate and interact. The bus is responsible for performing all translations between the various interfaces, providing great flexibility and cutting the cost of integration by reducing the need to modify operational systems. The value of the bus approach to integration is that information systems can be upgraded or replaced with little impact to other systems by simply managing the interfaces between the bus and the upgrade or replacement. The key to successful bus implementation is complete transparency between systems.

A common view of a services platform for a large organization may include communication protocols. It is not uncommon to consider an intranet, using the IP as a services platform, since it provides a ubiquitous communications mechanism. Using client-server architectures for databases may imply that specific interface languages, such as Structured Query Language (SQL), can serve as a service platform. A more recent architectural approach to horizontal integration via a service platform is Service-Oriented Architecture (SOA).

Figure 2 shows horizontal integration, with each function being integrated by a service that is provided to customers. Each of these services can be different and augmented with additional functions that may not be provided to other services.

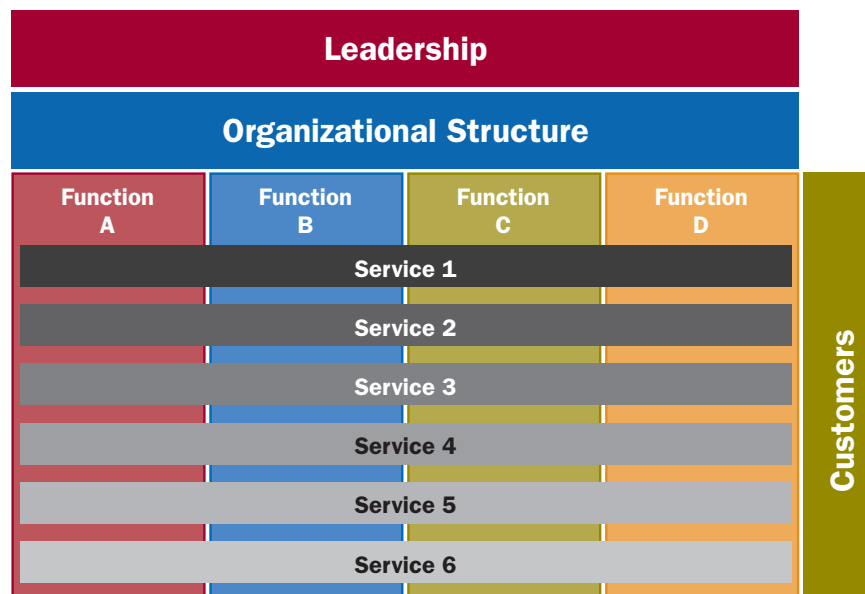


Figure 2: Horizontal Integration

Mesh or Star Integration

When complex integration or full interconnectivity between all systems is necessary, the mesh or star integration method is often used. Fundamentally, star integration is a concept where each system is directly interfaced with all other systems with which it must interact. From the perspective of a single system, the integration appears to be a star. From the perspective of the entire enterprise, the integrated systems appears to be a complete or partial mesh, often becoming so complex that it looks like spaghetti.

Star integration can be expensive and difficult to manage, but does provide considerable flexibility and can provide high performance since each interface can be optimized. The most troublesome part of star integration is that adding new functionality or systems can be very difficult since so many other systems can be impacted at the interface level.

Combined Integration Methods

Obviously, the integration approach selected should be based on organizational need. It is often necessary to use different integration methods within the same organization and to integrate different groups of integrated systems. For instance, some parts of an organization may be horizontally integrated and other parts may be vertically integrated. These various portions of the organization may then be interconnected via mesh integration.

Integration Process

Integration is a typical system design process. Rather than focusing on the development of a specific design solution, integration is focused on making different systems work together. The process for integration is not a new or standalone process, but uses standard design approaches. Therefore, there is no need to create a new process for integration. Fundamentally, the process involves requirements engineering; data and information modeling; functional, physical and logical modeling; and interface specification. This process is shown in Figure 3.

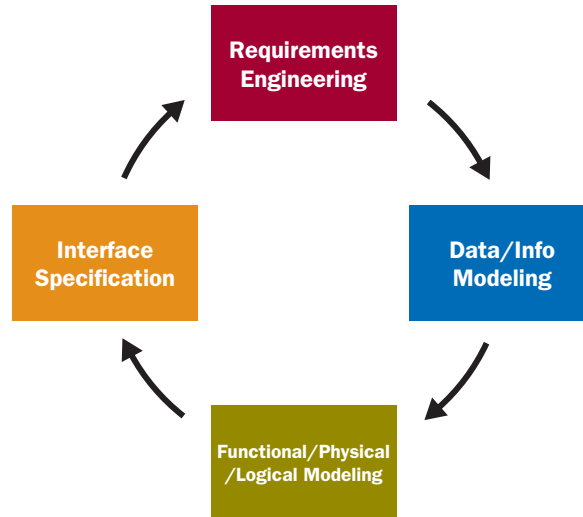


Figure 3: Integration Process

One aspect of integration that is often overlooked is human-machine integration. This focus is often referred to as the socio-technical aspect of information technology and addresses the integration of technology and humans and how they work together. Human-machine interfaces should be carefully considered, especially for COOP situations, since the COOP environment may be highly stressful and may impact human performance.

Requirements Engineering

The first aspect of integration is defining the necessary requirements for systems and their dependencies. If integration requirements are defined before systems are designed, procured or built, then those systems are more likely to have the capabilities and interfaces needed by the larger system.

The requirements effort needs to focus first on organizational objectives in normal operating environments and then expand upon those requirements for alternative situations that may be experienced during COOP. These new COOP requirements need to be folded in and integrated with the system requirements. The organization's available resources need to be balanced and allocated based on the integrated requirements set. The goal is to avoid defining the requirements for the status quo and then trying to add COOP requirements after the fact.

The objectives of requirements engineering is to find common solutions for multiple problems and to ensure that requirements are complete, not duplicative, and are not mutually exclusive. From a COOP perspective, the goal should be to find opportunities for dual-use functionality, where technologies can be utilized in day-to-day operations and during COOP events.

Some of the areas that should be focused on for dual-use capabilities during requirements engineering include data and information backup, storage and security; communications; telework and telecommuting; mobility, safety and so on. In determining the organizational needs for systems, effort should be focused on what systems need to work together. It is quite likely that systems that must interact during normal operations will likely need to interact during COOP.

Data and Information Modeling

A Data Flow Diagram (DFD) is a model of the movement of data and information in a system. The representation of data and information can be made at any layer of abstraction and for any type of system, be it an information system or an entire organization. DFDs help designers and process engineers to better understand the flow of data and information and thus identify the specific state of information in a system and the specific information interfaces that need to exist. DFDs are normally built top-down, such that a high-order context of the system is drawn first and then the system is decomposed into lower and lower levels until the desired level of understanding is reached.

DFDs can help an organization better understand how its information systems will operate and can help define the interfaces between them. A key advantage to DFDs is that models of information systems that need to interoperate can be compared such that the alignment or misalignment of interfaces can be clearly seen.

DFD modeling normally utilizes a top-down analysis approach starting with a context diagram that shows the “system” or organization from the perspective of all of its external interfaces. The flow of data and information from and to an organization’s exterior is critical to understanding how that organization interacts under normal situations and in identifying the information flow that needs to exist during COOP operations. Based on the top-level context diagram, system data models are decomposed into lower level processes, data stores and data flows. Processes are further decomposed into subprocesses and lower levels until the necessary level of detail is defined.

Functional, Physical and Logical Modeling

Proper systems integration goes beyond making hardware and software work together. The fundamental need for integration is that the organization’s functional capabilities need to operate together. Rather than performing these functions independently or in a duplicative way, they need to be properly aligned. The alignment can be physical or functional.

Physical integration is a more constrained form of integration. Its nature is one of making sure interfaces between physical things are properly matched. For instance, an electronic device that is built to handle 220 volts will not function if plugged into a 110-volt outlet. To function properly, the device and the outlet would have to be matched and the cable connecting them should be properly sized to handle 220 volts and the amperage rating of the circuit. For every form of physical interface, the two physical items must match at the points of connection.

A third form of integration is also necessary, especially when relating with information systems. Logical interfaces are needed to ensure that the correct data, information or knowledge is being passed across an interface. For information systems, the old adage of garbage in, garbage out is a truism. Regardless of how good the information system is, if it is fed the wrong data, its results will be wrong.

Functional Modeling

Functional modeling of a system or systems to be integrated is a well-known concept of systems management. It is a process of defining system functions, using a verb-noun pair, and then decomposing functions into lower-order functions. The functional decomposition does not need to represent every function that a system will be capable of performing but must define the functions that are necessary for the system solution being sought and the integration goal. Functional decomposition usually starts with primary system functions such as developing, verifying, deploying, operating, supporting and so on.

Figure 4 shows a basic, generic functional decomposition. As shown, functions are hierarchical and are often numbered to show their relationship.

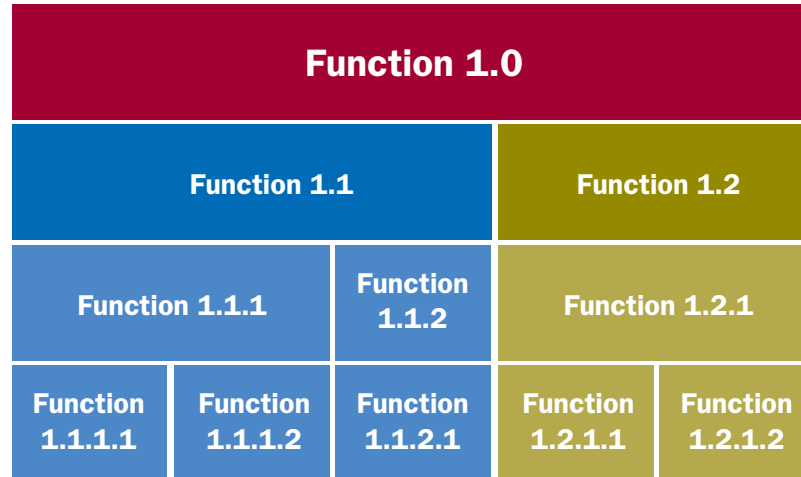


Figure 4: Generic Functional Decomposition

The system's functions are decomposed to a level that provides a clear understanding of system interactions. If multiple systems are to be integrated during development, then each system should be decomposed and the functions matched together. If two or more systems are to be integrated and one is a legacy system, the legacy system also needs to be decomposed so that integration points can be determined. In most cases, the systems under development will need to be matched to the legacy system functions and interfaces.

Once the system functions are fully defined, a design process will usually take place that identifies the physical elements of the system that are capable of accomplishing the functions. This process of design and synthesis involves integrating at the design level and is a matching of needs to capabilities.

Physical Modeling

From a physical perspective, integration is usually the job of design engineers. However, with information-related technologies, physical integration may be required when connecting physical systems together such as network devices. In such cases, it is necessary to fully delineate the physical aspects of the system interfaces. Depending on the level of design and integration, two possible forms of physical mapping may be needed. If specific systems are in the process of being designed, then there will be a need for internal physical interface definition. If systems are being assembled, then there will be a need for external interface definition.

An example of internal interfaces would be the definition of a particular type of daughter card that needs to be installed in a computer. A typical motherboard will have a specific number of expansion slots and each slot will be of a specific type. Integrating a graphics card into a computer is a rather trivial exercise if the type of interface is known. Systems that are being integrated into a larger system may need to have some of their internal elements changed or reconfigured.

External interfaces are typically easier to identify. Like internal interfaces, there will be specifications for various types of external interfaces. From an IT perspective, external interfaces will typically be cable connectors.

Physical interfaces are not confined to hardware. Software systems have similar types of interfaces. Software modules have internal interfaces that must be matched during development, such as software objects that have defined functional interfaces. Additionally, software applications will have specific processor requirements and will operate on one type of device or another. Naturally, there are software solutions that are portable and will operate on multiple platforms.

Logical Modeling

Logical modeling is accomplished at least twice during the system development process. First, logical modeling and analysis are accomplished to validate that system requirements are correct and complete. Finally, logical modeling is accomplished during design and development to verify system processes. The basic idea of logical modeling is to check that the system makes sense and will do what is needed, as defined by the requirements.

The essence of logic analysis, from a process standpoint, is determining the entrance and exit criteria for process tasks. In other words, what needs to happen or be complete in order to begin a specific task and what must the task accomplish in order to declare the task complete?

As COOP scenarios are contemplated, logic and logical flow, along with data and information flow, will be critical in order to ensure that essential organizational processes continue uninterrupted. It is very easy to make assumptions about what elements of a process are needed, but without some form of thorough analysis or modeling, there can never be an assurance that all pieces of a process are ready to transition into a COOP situation.

Functional, Physical and Logical Mapping

Based on the system-level requirement, and the derived requirements for accomplishing an organization's essential functions during COOP, a functional model needs to be created. As stated earlier, this functional model starts, at the highest order, with the primary system functions. Each of the primary system functions is decomposed to lower-order requirements. As the system is developed, the design process systematically maps functions to physical systems, including hardware, software and so on. During each of these steps, logical mapping is required in order to ensure that functions and physical solutions are valid and verified. Logical processing and modeling are used to make sure requirements are valid and that derived functions are logical representations of those requirements. During and after design, logical processes and modeling are used to verify that the physical systems developed actually fulfill the system's functions. Analysis, testing, demonstration or other means can represent this verification.

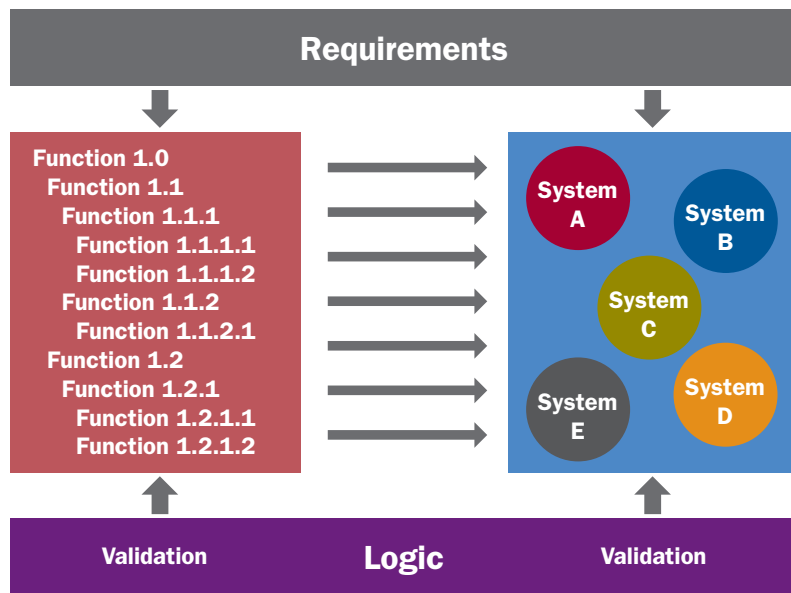


Figure 5: Functional, Physical and Logical Mapping

Interface Requirements and Specifications

During the development of system solutions, it is critical that specific requirements and specifications that identify the interfaces of the system are developed. These are necessary for the system itself and for other systems that may interact with it.

The questions to be answered by interface requirements and specifications are simple—if another system were to interact with the system under development, how would this be accomplished? The interface requirements must identify the functional interfaces of what is to be accomplished at the interface and how it is to be accomplished by either side of the interface. In other words, if a future system were to interact with the system being developed, what would the responsibilities be for each of the interacting parties involved?

Requirements and specifications are directly related to each other. Requirements define what the system must do functionally and physically and are usually expressed as “shall” statements. These requirement statements need to be clear and unambiguous. One measure of a good requirement is that it can be tested and verified. Requirements need to be as specific as possible.

At certain points in the development of systems, requirements are converted into specifications that delineate the “specific” criteria for systems elements. Typically, specifications contain details that go beyond requirements. Whereas requirements will provide designers with some flexibility to perform the art of engineering, where specifications are the outcomes and the details of that art.

Interface requirements and specifications fall into the same categories as other forms of requirements and specifications. Unfortunately, in many information systems being developed, interface requirements and specifications are not separated from the general requirements and specifications, making them appear to be less significant. In some cases, interfaces are forgotten altogether.

Interface requirements need to identify the functional and physical parameters inside and outside of the system being developed. Ultimately, the interface specifications need to clearly define each of the interior and exterior interfaces with measurable parameters and specific conditions for interactions. This information will make the system being developed more usable under various conditions. Considering COOP, a system that has clearly defined interface specifications will permit rapid and uneventful use of the system when time is of the essence.

Conclusion

Integration is a critical aspect of COOP. It is the engineering process that ensures that systems being developed are completely understood from the perspective of how they interact internally and how external systems interact with them. Without integration, each system would be required to be standalone or great pains would have to be taken to permit other systems to interact with them.

COOP situations will demand quick and responsive systems integration. As essential functions are reestablished, the systems that support those functions will be required to be operational quickly. New systems will likely need to be rapidly integrated with legacy systems.

Systems integration, as expressed in this paper, is a twofold issue. First, systems have to be designed for integration and their interfaces need to be carefully and fully described. Second, organizational structure will often determine how systems are to be integrated. Organizations that use silos will typically have dedicated information systems to support each of their service needs. Vertically integrated organizations, however, may share information technology resources to accomplish their services and missions.

To ensure that organizations are prepared to deal with COOP events, they need to thoroughly examine the organizational structure and the integration aspects of the systems they have. In this sense, systems are not narrowly defined as hardware and software, but all systems, including organizational processes that need to be utilized. Systems need to be fully understood and defined. To accomplish this, they need to be described

through functional decomposition, which describes what the system does. Next, they need to be described or modeled physically so there is a clear understanding of what element is responsible for each function. Finally, systems need to be analyzed from a logical perspective. Initially, logical analysis will be used to validate that a system's requirements are proper and eventually it will be used to ensure that the completed system is verified to do what it is meant to do.

The three elements of functional behavior physical components, and logic need to be mapped together to make sure the system is complete and capable of performing its intended mission.

Finally, because systems will reside in different places and may be used under different circumstances, it is necessary to fully describe system interactions as interface requirements and specifications. These defined system parameters will be used to make the system a part of larger, complex systems. In a situation where organizations will need to reform to accomplish their essential functions, the processes and information systems they use will need to be integrated rapidly into a functioning whole.

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 HEADQUARTERS
AND SALES HEADQUARTERS
FOR NORTH AND SOUTH AMERICA
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

EAST COAST OFFICE
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886-3146 USA
Phone: 978.589.5800
Fax: 978.589.0800

ASIA PACIFIC REGIONAL
SALES HEADQUARTERS
Juniper Networks (Hong Kong) Ltd.
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EUROPE, MIDDLE EAST, AFRICA
REGIONAL SALES HEADQUARTERS
Juniper Networks (UK) Limited
Building 1
Aviator Park
Station Road
Addlestone
Surrey, KT15 2PG, U.K.
Phone: 44.(0).1372.385500
Fax: 44.(0).1372.385501

Copyright © 2008, Juniper Networks, Inc.
All rights reserved. Juniper Networks, the
Juniper Networks logo, NetScreen, and
ScreenOS are registered trademarks of
Juniper Networks, Inc. in the United States
and other countries. All other trademarks,
service marks, registered trademarks, or
registered service marks in this document
are the property of Juniper Networks or their
respective owners. All specifications are
subject to change without notice. Juniper
Networks assumes no responsibility for
any inaccuracies in this document or for
any obligation to update information in this
document. Juniper Networks reserves the
right to change, modify, transfer, or otherwise
revise this publication without notice.