Pipeline
Knowledge Is Power

www.pipelinepub.com Volume 4, Issue 4

Not for distribution or reproduction.

# Network, Heal Thyself!
# Technological Advances Put Building Blocks in Place to Create Self-Healing, Reactive Network Architecture

by Ken Ferderer

Since the beginning of network computing, networks have been designed as a collection of independent devices that have a limited awareness of each other, and no awareness of the collective whole outside of the basic routing fabric. As a result, networks have always had a limited ability to dynamically adjust to problems or events occurring within the network and are incapable of adjusting to any changes outside the network.

Traditionally, network engineers have attempted to build around these shortcomings with redundant routes, standby devices, and other mechanisms that introduce some basic resiliency into the network. These simple workarounds have always begged the larger questions:

- Is it possible for a network and its collection of independent devices to react appropriately to changes in the environment without manual intervention?
- Could a network ever recognize an event outside of its routing fabric that requires a change to its behavior or operation of the collective whole?
- And based on what happens, could it modify the behaviors of multiple independent devices to accommodate that event?

Take as an example a sophisticated emergency response network, which links together special sensors that detect fires and chemical, radiological or nuclear threats. Based on the 'type' of event recognized by the sensor arrays, the underlying logical network must be dynamically configured in any number of pre-defined configurations.

In one such scenario, the sensors may report that a fire has been detected and, based on this event, the network should immediately alter its current logical configuration to provide secure connectivity between first responders, including police and fire departments, local authorities, and local news agencies. If however, a chemical or nuclear event is detected, the underlying network should instantaneously reconfigure itself to securely connect all federal response agencies, command and control, and route around any unresponsive sites.

© 2006, All information contained herein is the sole property of Pipeline Publishing, LLC. Pipeline Publishing LLC reserves all rights and privileges regarding the use of this information. Any unauthorized use, such as distributing, copying, modifying, or reprinting, is not permitted. This document is not intended for reproduction or distribution outside of www.pipelinepub.com. To obtain permission to reproduce or distribute this document contact sales@pipelinepub.com for information about Reprint Services.

To date, this type of dynamic network-level reconfiguration – or self-healing – based on a non-network event has simply not been possible. Even the network management solutions that, by definition, provide a broader view of the deployed network environment, are only able to recognize limited network-level events, such as dropped routes, throughput issues, and device failures. These solutions have a very limited capability to automatically react to, or recover from, any type of network event. In fact, most network management solutions are only capable of pushing static configuration files out to deployed devices that have lost their configurations.

In most cases today, the only self-healing network alternative is to construct redundant routes so that if a path becomes unavailable, traffic will automatically re-route onto the secondary path. This solution is far from ideal, as it is impossible to build enough redundant routes and workarounds for every type of potential event, both internally and externally. Truly dynamic and automatic reconfiguration of the logical network is simply not possible with existing technologies no matter what device and management vendors would like us to believe. However, times are changing.


Technology advances have finally provided the building blocks required to create a sophisticated networking architecture that is truly dynamic and has the characteristics to self-heal – or react – to any number of definable events, even if they are occurring outside of the networked fabric. The essential technologies consist of:

- Device-level service oriented architectures (SOA): These SOA are able to expose underlying communication services and other capabilities on a typical network device to higher-level applications;
- Advanced policy-driven control and management solutions: These technologies control network characteristics and behavior based on configurable user defined policies and implement the behaviors onto selected devices;

Not for distribution or reproduction.

© 2006, All information contained herein is the sole property of Pipeline Publishing, LLC. Pipeline Publishing LLC reserves all rights and privileges regarding the use of this information. Any unauthorized use, such as distributing, copying, modifying, or reprinting, is not permitted. This document is not intended for reproduction or distribution outside of www.pipelinepub.com. To obtain permission to reproduce or distribute this document contact sales@pipelinepub.com for information about Reprint Services.

- Sophisticated rules engines: These engines are capable of defining events and the appropriate response criteria for scenarios both within the network fabric and external to the network.

All that is required is to carefully arrange these technologies into a cohesive and functional working system, and that begins with the actual network devices. Traditional networking devices have proprietary and closed architectures that expose to the end user very few services, except for basic transport services configured via the command line or mechanisms that enable the loading of whole configuration files from network management systems.

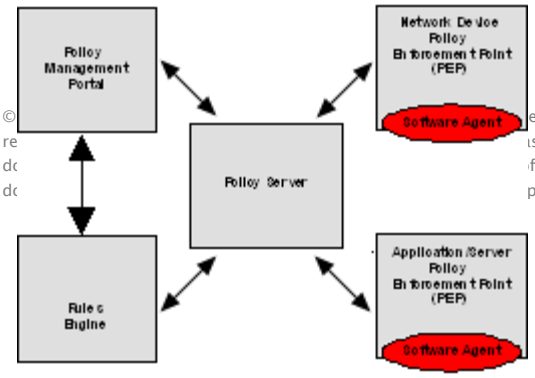
New networking devices being introduced into the market today are breaking that mold. Instead, they are shipping with fully open architectures that enable end-users to build and run applications directly on the device or on service blades within the device. In some cases, the device actually exposes the underlying kernel services and interface capabilities through a SOA layer where the external application or end user can toggle the behavior of the device through these exposed levers. The open nature of the device allows for robust application-level agents to be run on the device. These agents are capable of not only driving the device behavior, but also listening to the device and gleaning operational and service performance statistics in real time.

The ability to manipulate the device while simultaneously receiving real-time feedback is critical to the next building block required for the solution: the policy server. Although not a new concept, there have been steady advancements in policy-based networking technologies and these technologies are now capable of providing the control and automation functionality required for a truly dynamic, self-healing network.

Too frequently, the term 'policy server' has been used by the network management vendors to define template servers that manage device-level configuration files. In these solutions, network engineers build a static configuration file that meets their requirements then save the configuration as a template. Each subsequent device load is then achieved by simply pushing the proper configuration template down onto the device. This is not true policy-based networking.

The new generation of policy-based technologies now available in the market function at a higher level of abstraction from the network device and are far more powerful than these traditional template servers. These new policy servers are able to capture end-user requests through intuitive, non-technical interfaces and use these requests to define, implement, and govern the behaviors of the underlying network devices and services.

In this example (Figure 1), the policy server becomes the heart of a dynamic self-healing network as it provides the necessary intelligence that recognizes events within, and external to, the network and determines how the network should be reconfigured based on defined events.




© eline Publishing, LLC. Pipeline Publishing LLC reserves all rights and privileges re s distributing, copying, modifying, or reprinting, is not permitted. This do f www.pipelinepub.com. To obtain permission to reproduce or distribute this do print Services.

Not for distribution or reproduction.

**Fig. 1:** Policy server is central to the self-healing network

The policy server accomplishes this by first deconstructing the end user request into the specific service components that require action or changes to be made. This determination is made possible by a comprehensive repository of networked resources that includes all the deployed devices, available network services, and resources maintained by the policy server. From this repository, the policy server is able to determine all the deployed resources that require changes to satisfy the end-user request.

Because the network must be able to react to outside occurrences, the repository must be able to define and manage resources external to the network such as application servers, desktops, laptops, and users. These external resources are modeled in the repository in the same way as the network devices.

Once the resources that need to be re-configured are determined, the policy server sends directives down to agents running on the actual devices. These directives are constructed using a markup language, which is interpreted by the agent into the device-level command language necessary to initiate the services through the SOA layer on the device.

The combination of this new generation of policy servers – which can define, control and implement desired behaviors in underlying network devices – with the new open architectures and exposed SOA layers on these devices allow for dynamic and instantaneous changes to the logical network environment and behavior.

The final building block for our self-healing network solution consists of a sophisticated rules engine that defines events in the network, or external to the network, and the appropriate actions to take in such a form that the policy server can act upon them.

This rules engine must be capable of sophisticated logic patterns so that complex conditional decisions can be made. The ability to create complex rules, such as "*if A occurs, then do B, but if A* ***and*** *C occur, then do D,*" are necessary to effectively define a network capable of responding to real world events. In the real world, there are very few events and actions that consist only of a singular event-to-action relationship. Most of the scenarios network engineers deal with on a daily basis have multiple contributing events and multiple actions.

As with the policy server, the rules engine in our solution has to be capable of modeling events and actions outside of the network. As the policy server receives notice from an agent residing on a network device or non-network device (server, desktop, laptop, or even a handheld device) that a defined event has occurred, the policy server queries the rules engine for the appropriate response(s) to the event and begins to adjust the underlying network devices accordingly.

Obviously the complexity of the dynamic configuration or self-healing solution resides in the proper definition of events and actions. The mechanics of dynamically

Not for distribution or reproduction.

© 2006, All information contained herein is the sole property of Pipeline Publishing, LLC. Pipeline Publishing LLC reserves all rights and privileges regarding the use of this information. Any unauthorized use, such as distributing, copying, modifying, or reprinting, is not permitted. This document is not intended for reproduction or distribution outside of www.pipelinepub.com. To obtain permission to reproduce or distribute this document contact sales@pipelinepub.com for information about Reprint Services.

implementing change into the networked environment are relatively minor compared to correctly defining and controlling the change behaviors being implemented. More importantly, as every network engineer understands, any change introduced into a system can drastically impact the behavior of that system in any number of ways – good or bad.

To minimize the risk of building a house of cards that may collapse upon itself as a new change is being introduced based on an event, the policy server must maintain a persistent linkage at all times between deployed configurations and the policies that implemented those configurations within its repository. This critical functionality allows the network engineer to model potential event scenarios and see how the underlying logical network will behave as those incremental changes are being introduced, without disrupting the actual production networks.

Using these building blocks in conjunction with many other existing network technologies, we are finally making fundamental advances in network automation and self-healing. Companies working with these technologies are currently deploying dynamic network solutions such as the emergency response network described earlier. And, with further refinements from leading hardware and software companies, network engineers will soon have the ability to control and manage any network infrastructure using defined policy, and have that network be capable of recognizing events and implement appropriate actions automatically across the network fabric as needed. A network that can truly heal itself.

***If you have news you'd like to share with Pipeline, contact us at editor@pipelinepub.com.***

Not for distribution or reproduction.

© 2006, All information contained herein is the sole property of Pipeline Publishing, LLC. Pipeline Publishing LLC reserves all rights and privileges regarding the use of this information. Any unauthorized use, such as distributing, copying, modifying, or reprinting, is not permitted. This document is not intended for reproduction or distribution outside of www.pipelinepub.com. To obtain permission to reproduce or distribute this document contact sales@pipelinepub.com for information about Reprint Services.