SPECTRUM by concord®

EVENT CORRELATION AND ROOT CAUSE ANALYSIS IN SPECTRUM®

MARCH 2004

concord®

Table of Contents

Table of Figures

Executive Summary

Action is the enemy of fault. The following paper is designed to explain how SPECTRUM software performs event correlation, impact analysis and root cause analysis for multiple vendors and technologies across network, system and application infrastructures. SPECTRUM uniquely offers three intelligent, automated and integrated approaches to problem solving — while also proactively communicating bi-directionally with infrastructure elements in a "trust but verify" methodology. SPECTRUM doesn't just listen for problems — it proactively talks with the infrastructure for health status and detailed diagnostic information. The end result is a cost-effective way to drive reliability, efficiency and effectiveness in managing IT infrastructure as a business service.

SPECTRUM's three approaches to problem solving discussed in this paper are:

- Model-based Inductive Modeling Technology (IMT)
- Rules-based Event Management System (EMS)
- Policy-based Condition Correlation Technology (CCT)

Our approach avoids the often technically religious arguments whether a model-based, rules-based or policy-based system is the best solution by leveraging all three. SPECTRUM is fundamentally a model-based system. ***Model-based*** systems are ***adaptable*** to changes that regularly happen in a real-time, on-demand IT infrastructure — however, these systems are criticized because development of new models and relationships may require programming skills. ***Rules-based*** systems are ***flexible*** in allowing customers to add their own intelligence without requiring programming skills — however, these systems are criticized since rules must be constantly re-written every time there is a move/add/change. SPECTRUM combines the best of both approaches, using models to keep up with changes while leveraging easy-to-create rules running against the models to avoid the need for constant rule editing. ***Policy-based*** systems are ***automated*** means of stitching together seemingly unrelated pieces of information to determine condition and state of physical devices and logical services. This condition correlation engine combines with SPECTRUM's modeling engine and rules engine to deliver a higher level of cross-silo service analysis.

Almost every service delivery infrastructure problem can be placed into one of three categories: *availability, performance or threshold exceeded*. Infrastructure faults occur when things break, whether they are related to LAN/WAN, server, storage, database, application or security. Infrastructure performance problems often result in brown-out conditions where services are available but are performing poorly. A slow infrastructure is a broken infrastructure from the user's perspective. The final category is abnormal behavior conditions where performance, utilization or capacity thresholds have been exceeded as demand/load factors fall significantly above or below observed baselines. SPECTRUM can pinpoint the cause of availability problems and proactively monitor response time to ensure performance, while also delivering thousands of out-of-the-box thresholds to predict and/or prevent problems.

Model-based, rules-based and policy-based analytics in SPECTRUM understand relationships between IT infrastructure elements and the customers or business processes they are designed to support. It is through this understanding of relationships that SPECTRUM has been shown to deliver 70% reductions in downtime while resolving 90% of availability or performance problems from a central location. SPECTRUM's root cause analysis has been able to reduce the number of alarms by several orders of magnitude while reducing Mean-Time-to-Repair (MTTR) from hours to minutes. SPECTRUM's distributed management architecture has also proven effective at performing root cause analysis for over 5 million devices (20+ million ports) in a single environment with fully meshed and redundant core and distribution network layers. Our integrated approach to fault and performance management has enabled enterprise, government and service provider organizations around the world to manage what matters through Service Level Intelligence™.

A Complex Problem in Need of A Solution

IT Infrastructure operations management is a difficult and resource intensive, yet necessary, undertaking. For those not experienced with the task, the difficulty and resource requirements are often beyond immediate comprehension. Many people just expect the infrastructure to work. They don't think of the infrastructure as a dynamic multi-vendor engine made of frequently changing components and technologies. In reality, the complexity and dynamics of today's real-time, on-demand IT architectures often leads to fragility. Inevitably, the infrastructure will fail or slow down and when it does, tools are required to quickly pinpoint the root cause, suppress all symptomatic faults, prioritize based on business impact, and aide in the troubleshooting and repair process to accelerate service restoration.

The Infrastructure is the Business

The IT infrastructure is a collection of interdependent components including computing systems, networking devices, databases and applications. Within the set of infrastructure components are multiple versions of many vendors' products connected over a multitude of networking technologies. To make it even more complex, each business environment is different from the next. There is no standard set of components configured in a standard way. There is constant change in devices, firmware versions, operating systems, networking technologies, development technologies and tools. But this dynamic and complex infrastructure is there for a purpose; ***the infrastructure IS the business***. Just as no organization could operate without electricity, telephones, water or gas — no modern business can profit and grow without an IT infrastructure. Either the infrastructure works and evolves or the organization is out of business. Systems, applications and networking vendors are continuously evolving their technologies and the customers of those vendors must keep pace. All companies must also evolve their people, processes and management tools for greater efficiency, or fall competitively behind.

To ensure the performance and availability of the infrastructure, most companies employ a dual approach of 1) highly available, fault-tolerant, load-balancing designs for infrastructure devices and communication paths; and 2) a management solution to ensure proper operation. In fact, the job of the management solution is further complicated by today's high-availability environments. The management solution must understand the load-balancing capacity; it must be able to track primary and fault-tolerant backup paths; and understand when redundant systems are active. The investment in the management solution is as important as the investment in the infrastructure itself. The solution must be broad, deep, flexible, adaptable, automated, integrated and intelligent to perform its intended function. The infrastructure is not static and the tool will need to embrace change while delivering an end-to-end integrated view of performance and availability across infrastructure silos. Unfortunately, many management tools are not adaptive and do not keep pace with the dynamics of real-time, on-demand IT.

Predicting and Preventing Problems

Many management vendors explain in great detail how their software helps IT operations staff AFTER a problem has already happened. We believe that the management software should help predict or prevent problems in the first place. Working for over a decade with thousands of customers in multiple vertical industries around the world, we have learned through experience how out-of-the-box utilization, performance and response time thresholds can be used to act as an early warning system when a problem is about to happen or when a service level guarantee is about to be violated. While these thresholds can obviously be tuned for a specific customer environment — there is tremendous value in having these out-of-the-box thresholds be relevant on day one using generic baselines. SPECTRUM enables its users to add their own thresholds and watches such that after a unique problem happens in the environment once, new watches can help predict or prevent it from happening again.

Knowing There is A Problem is Not Enough

To remain relevant and competitive in the marketplace, companies must minimize outages and performance degradations. In order to do this, the individual or groups responsible for the care of the infrastructure (e.g. IT, Engineering, Operations, etc) must be proactively notified of problems. There are many tools that claim to monitor the availability and performance of infrastructure components and the business applications which rely on them. Most of these tools simply identify that a problem exists and have some notification mechanism to alert operators or technicians of a problem. They often give visibility into only a small slice of the technologies being managed and have no ability to understand how the various technology components relate to each other. Before the true task of troubleshooting can begin, the troubleshooter has to isolate the problem. Simply knowing there is a problem and collecting all the problems on one screen is not enough. Troubleshooters need to know where the problem is (and where the problem is not) to effectively triage the issue. If multiple problems are happening simultaneously, issues should be automatically prioritized based on impacted customers, services or infrastructure devices. It is far too costly to rely on human intervention to determine the root cause of problems and it is also far too costly to sift through an unending stream of symptomatic problems. Every minute the troubleshooter spends isolating the problem is a minute lost to solving the problem.

For example, one service provider has experienced a situation where a service provider was receiving 500,000 daily problem notifications using their previous management tool.There was no doubt the service provider knew there were problems but no person, or team of people, could keep up with that many events. SPECTRUM's root cause analysis technology helped this service provider reduce their number of daily problem notifications to 200 actual alarm conditions while also automatically prioritizing issues based on impact. In this environment, 500,000 symptoms/effects had only 200 causes. Average time to find and fix a problem was reduced from over 4 hours to less than five minutes.

The Importance of Understanding Business Impact

The best management solutions will not only be able to identify problems, isolate them and suppress all symptomatic events, but also identify all impacted components, services and customers. For the business, understanding impact is as important as understanding the root cause. When outages or performance degradations occur, business services and the users of those business services are affected. When this happens, people typically cannot do their jobs effectively, resulting in lower productivity or efficiency and ultimately opportunity costs. Sometimes the services provided by the company to their customers are affected,

which results in lost revenue, SLA penalties, lost customers and even damaged brand reputation. Some organizations are tolerant of lost productivity but not many are tolerant of lost current or future revenue. Damage to the company's brand image or reputation requires years of marketing and advertising investments to repair.

For large organizations it is quite probable that there are too many problems occurring at the same time. Knowing the root cause allows an organization to efficiently get problems fixed without wasting time pursuing symptomatic problems. Knowing impact allows an organization to prioritize response efforts and effectively provide help desk services.

Event Correlation and RCA — A Three-Pronged Approach

The fundamentals of SPECTRUM's Root Cause Analysis (RCA) are as follows:

- The system must understand the relationship between information within the infrastructure and the systems/applications/services/customers that depend on that information.
- The system must be proactive in its monitoring and not just rely on event streams.
- The system must distinguish between a plethora of events and meaningful alarms.
- The system must scale and adapt to the requirements of growing and dynamic infrastructures.
- The solution must work across multiple-vendor and technology solutions.
- The system must allow for extensions and customization.

Management software applications efficiently performing root cause analysis should raise an alarm for the root condition and should prevent any symptom/effect resulting from the root condition from being presented as a unique or separate alarm. SPECTRUM relies on multiple techniques working cooperatively to deliver its event correlation and Root Cause Analysis capabilities.These include Inductive Modeling Technology (IMT), EventManagement System (EMS) and Condition Correlation Technology (CCT). Each of these techniques is employed to diagnose a diverse and often unpredictable setof problems. Just as a handyman needs multiple tools to have the right tool for the right task, SPECTRUM employs more than one tool for infrastructure fault and performance analysis.

Root Cause Analysis can be simply defined as the act of interpreting a set of symptoms/events and pinpointing the source that is causing those symptoms/events. Within the context of infrastructure management, events are occurrences of significant happenings presented from a source to other systems. Events are typically local to a source, and without proper context do not always help with RCA. Often correlation of events is required to determine if an actionable condition or problem exists but is almost always required to isolate problems, identify all impacted components and services, and suppress all symptomatic events. Many components provide events and events come in many forms; SNMP traps, syslog messages, application log file entries, TL1 events, ASCII streams, etc. More sophisticated management systems like SPECTRUM can also generate events based on proactive polling of component status, parameter based threshold violations, response time measurement threshold violations, etc.

Intelligent Service Assurance software solutions like SPECTRUM present RCA results in the form of alarms. SPECTRUM's RCA is the automated process of trouble-shooting the infrastructure and identifying the managed element(s) that have failed to perform their function. The goal of SPECTRUM's RCA is straightforward; ***identify a single source of failure, the Root Cause, and generate the appropriate actionable alarm for the failed managed element***. While simple in concept, this goal is difficult to achieve in real world practice.

Inductive Modeling Technology

The core of SPECTRUM's RCA solution is its patented Inductive Modeling Technology (IMT). IMT uses a powerful object oriented modeling paradigm with model-based reasoning analytics. In SPECTRUM, IMT is most often used for physical and logical topology analysis as SPECTRUM can automatically map topological relationships through its accurate and efficient auto discovery engine. Significant physical/logical entities are represented as software models in SPECTRUM. The models are a software representation of a real-world physical or logical device. These software models are in direct communication with their real-world counterparts enabling SPECTRUM to not only listen, but proactively query for health status or additional diagnostic information. Models are described by their attributes, behaviors, relationships to other models and algorithmic intelligence. Intelligent analysis is enabled through the collaboration of models in a system. Collaboration between models is one of the many value points of SPECTRUM's modeling system. Collaboration between models enables correlation of the symptoms, suppression of unnecessary alarms and impact analysis of affected users, customers and services. Collaboration includes the ability to exchange information and initiate processing between any models within the modeling system. A model making a request to another model may in turn trigger that model to make requests on other models, and so on. Relationships between models provide a context for collaboration. Collaboration between models enables:

- Correlation of the symptoms
- Suppression of unnecessary/symptomatic alarms
- Impact Analysis

In SPECTRUM, a "model" is the software representation of a real world managed element, or a component of that managed element. This representation allows SPECTRUM to not only investigate and query an individual element within the network, but also provides the means to establish relationships between elements in order to recognize them as part of a larger system. IMT's root cause analysis is built upon a sophisticated system of models, relationships and

behaviors that create a software representation of the infrastructure. Decisions as to which element is at fault are not determined by looking at a single element alone. Instead, the relationship between the elements is understood and the conditions of related managed elements are factored into the analysis.

A simple example of IMT in action can be demonstrated by a network router port transition from UP to DOWN. If a port model receives a LINK DOWN trap, it has intelligence to react by performing a status query to determine if the port is actually down. If it is in fact DOWN, then the system of models will be consulted to determine if the port has lower layer sub-interfaces. If any of the lower layer sub-interfaces are also DOWN, only the condition of the lower layer port will be raised as an alarm. An application of this example can be described by several Frame Relay Data Link Control Identifiers (DLCIs) transitioning to INACTIVE. If the Frame Relay port is DOWN, IMT will suppress the symptomatic DLCI INACTIVE conditions and raise an alarm on the Frame Relay port model. Additionally, when the port transitions to DOWN, IMT will query the status of the connected Network Elements (NEs) and if those are also DOWN, those conditions will be considered symptomatic of the port DOWN, will be suppressed, and will be identified as impacted by the port DOWN alarm. Root cause and impact are determined through IMT's ability to both listen and talk to the infrastructure.

As briefly discussed, IMT is a very powerful analytical system and can be applied to many problem domains. A more in depth discussion of IMT in action will be covered later in the paper.

Event Management System

There are times when the only source of management information is through event streams local to a specific source. There may be no way to talk to the managed element, but rather only a way to listen to it. Any one event may or may not be a significant occurrence — but in the context of other events, information or time, may be an actionable condition. Event Rules in SPECTRUM's Event Management System provide a more complex decision making system to indicate how events should be processed. Event Rules can be applied to look for a series of events to occur on a model in a certain pattern or within a specific time frame or with certain data value ranges. Event Rules can be used to generate other events or even alarms. If events occur such that the preconditions of a rule are met, another event may be generated allowing cascading events; or the event can be logged for later reporting/troubleshooting purposes; or it can be promoted into an actionable alarm. SPECTRUM provides six customizable Event Rule types that form the basis of the Event Management System rules-based engine.

These rule types are building blocks that can be used individually or cooperatively to effect an alarm on the most simple or sophisticated event-oriented scenarios. This Event Management System rules engine allows for the correlation of event frequency/duration, event sequence and event coincidence.We have discovered through extensive customer interviews that over 80% of rules writing in competing management systems fell into one of these three event types — frequency/duration, sequence or coincidence. Keep in mind Event Rules can be run against IMT models to avoid the need to constantly re-write rules to reflect infrastructure move/add/change activity. The Event Rule types are highlighted below followed by usage examples later in the paper:

- **Event Pair (Event Coincidence):** This rule is used when you expect certain events to happen in pairs. If the second event in a series does not occur, this may indicate a problem. The Event Pair rule type is used to create a more relevant event based on this scenario. Event rules created using the Event Pair rule type generate a new event when an event occurs without its paired event. It is possible for other events to happen between the specified event pair without affecting this event rule.
- **Event Rate Counter (Event Frequency):** This rule type is used to generate a new event based on events at a specified rate in a specified time span. A few events of a certain type can be tolerated, but once the number of these events reaches a certain threshold within a specified time period, notification is required. No additional events will be generated as long as the rate stays at or above the threshold. If the rate drops below the threshold and then subsequently rises above the threshold, another event will be generated.
- **Event Rate Window (Event Frequency):** This rule type is used to generate a new event when a number of the same events are generated in a specified time period. This rule type is similar to the Event Rate Counter. The Event Rate Counter type is best suited for detecting a long, sustained burst of events. The Event Rate Window type is best suited for accurately detecting shorter bursts of events. It monitors an event that is not significant if it happens occasionally, but is significant if it happens frequently. If an event happens a few times a day — no problem. If an event happens 5 times in one minute — there is a problem. If the event occurs above a certain rate, then another event is generated. No additional events will be generated as long as the rate stays at or above the threshold. If the rate drops below the threshold and then subsequently rises above the threshold, another event will be generated.
- **Event Sequence (Event Sequence):** This rule type is used to identify a particular order of sequence in events that might be significant in your IT infrastructure. This sequence can include any number and any type of events. When the sequence is detected in the given period of time, a new event is generated.
- **Event Combo (Event Coincidence):** This rule type is used to identify a certain combination of events happening in any order. The combination can include any number and type of events. When the combination is detected within a given time period, a new event is generated.
- **Event Condition (Event Coincidence):** This rule type is used to generate an event based on a conditional expression. Part of SPECTRUM's "trust but verify" methodology — a series of conditional expressions can be listed within the event rule and the first expression that is found to be TRUE will generate the event specified with the condition. This rule type is extremely powerful! Rules can be constructed to provide correlation through a combination of evaluating event data with IMT model data (including attributes which can be read directly from the remote managed element). For example, if a trap is received notifying the management system of memory buffer overload, to validate that an alarm condition has occurred, an Event Condition rule can initiate a request to the device to check actual memory utilization.

SPECTRUM implements a number of Event Rules out-of-the-box by applying one or more of the event rule types to event streams. Users can create or customize event rules using any of the rule types and apply these Event Rules on other event streams. Further implementation of Event Rules using the Event Management System will be discussed later in this paper.

Condition Correlation

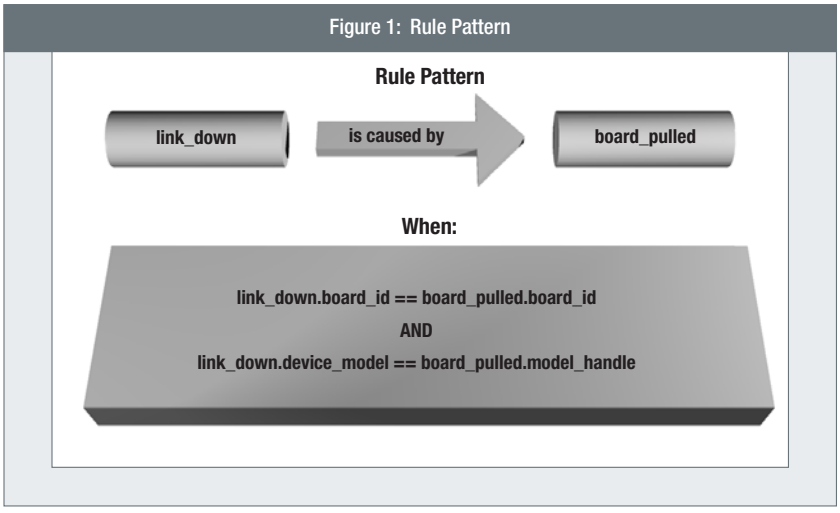
In order to perform more complex user-defined or user-controlled correlations, a broader set of capabilities is required. SPECTRUM's policy-based Condition Correlation Technology enables:

- Creation of correlation policies
- Creation of correlation domains
- Correlation of seemingly disparate event streams or conditions
- Correlation across sets of managed elements
- Correlation within managed domains
- Correlation across sets of managed domains
- Correlation of component conditions as they map to higher order concepts such as business services or customer access

In order to understand these capabilities, the terminology is described in this context:

- **Conditions:** A condition is similar to state. Condition can be set by an event/action and cleared by an event/action. It is also possible to have an event set a condition but require a user-based action to clear the condition. The condition exists from the time it is set until the time it is cleared. A very simple example of a condition is "port down" condition. The "***port down***" condition will exist for a particular interface from the time the LINK DOWN trap or set event (such as a failed status poll) is received until the time the LINK UP trap or clear event (such as a successful status poll) is received. A number of conditions that may be of use for establishing domain level correlations are defined out-of-the-box in SPECTRUM and more can be added by the user.
- **Seemingly Disparate Conditions:** Many devices in an IT infrastructure provide a specific function. The device level function is often without context as it relates to the functions of other devices/components. Most managed elements can emit event streams but those event streams are local to each component. A simple example is when a Response Time Management system identifies a condition of a test result exceeding a threshold. At the same time, an Element Management System may identify a condition of a router port exceeding a transmit bandwidth threshold. These conditions are seemingly disparate as they are created independently and without context or knowledge of each other. In reality the two are quite related.
- **Rule Patterns:** Rule Patterns are used to associate conditions when specific criteria are met. A simple example is a "port down" condition caused by a "board pulled" condition — but only if the port's slot number is equal to the slot number of the board that's been pulled. Figure 1 illustrates this rule pattern. The result of applying a rule pattern can be the creation of an actionable alarm or the suppression of symptomatic alarms.

Figure 1: Rule Pattern



- **Correlation Domains:** A Correlation Domain is used to both define and limit the scope of one or more Correlation Policies. A Correlation Domain can be applied to a specific Service. For example, in the Cable Broadband environment, a return path monitoring system may detect a return path failure in a certain geographic service area. This "return path failure" condition is causing subscriber's high-speed cable modems to become unreachable and Video on Demand (VoD) pay-per-view streams to fail. The knowledge that the return path failure, the modem problems and the failed video streams are all in the same correlation domain is essential to correlating the events and ultimately identifying the root cause. However, it is also important to have the ability to distinguish that a "return path failure" condition occurring in one correlation domain (Philadelphia, PA) not be correlated with VoD stream failure conditions occurring in a different correlation domain (Portsmouth, NH).
- **Correlation Policies:** Multiple Rule Patterns can be bundled/grouped into Correlation Policies. Correlation Policies can then be applied to a Service or Correlation Domain. For example, a bundle of a set of rule patterns applicable to OSPF correlation can be created and labeled the OSPF Correlation Policy. The OSPF Correlation Policy can be applied to each Correlation Domain where the Correlation Domain is defined by each autonomous OSPF region and the supporting routers in that region. Another example of a Correlation Policy could be defined by a set of rule patterns that operate within the confines of a MPLS/BGP VPN, labeled as the Intra-VPN Policy, and applied to all modeled VPNs. Whenever a rule is added or removed from a Correlation Policy, all related Correlation Domains are updated immediately and automatically. Multiple Correlation Policies can be applied to any Correlation Domain, and a Correlation Policy may be applied to many Correlation Domains.

Condition based correlations are very powerful and provide a mechanism to develop Correlation Policies and apply them to Correlation Domains. When applied to Service Level Management, Correlation Policies can be likened to metrics of an SLA; and Correlation Domains can be likened to service, customer or geographical groupings. There are times when the only way to infer a causal relationship between two or more seemingly disparate conditions is when those conditions occur in a common Correlation Domain. These mechanisms are necessary when causal relationships cannot be discovered through interrogations or receipt of events to/from the infrastructure components.

Fault Scenarios

Out of the box, SPECTRUM addresses a wide range of different scenarios where it can perform root cause analysis. This section provides specific scenarios where the techniques described in the previous section are employed to determine RCA and Impact Analysis. The detail will be limited to the basic processing for the sake of simplicity and brevity. Also for the purpose of the discussion and figures, the following table is provided showing the color of alarms that are associated with the icon status of SPECTRUM models at any given time.

Figure 2: Model Status Colors

Status of Model	Color of Alarm
Normal Operation	GREEN
Critical Fault	RED
Major Fault	ORANGE
Minor Fault	YELLOW
Unknown or Suppressed	GRAY
Down for Maintenance	BROWN
Initial Condition	BLUE

Inductive Modeling Technology

Communication outages are types of faults often described as "black-outs" or "hard faults." With these types of faults, one or more communication paths are degraded to the point that traffic can no longer pass. The cause of this fault could be many things including broken copper/fiber cables/connections, improperly configured routers/switches, hardware failures, severe performance problems, security attacks, etc. Often the difficulty with these hard communication failures is that there is limited information available to the management system as it is unable to exchange information with one or more managed element. It is SPECTRUM's sophisticated system of models, relationships and behaviors available through IMT that allows it to infer the fault and impact. IMT inference algorithms are also called ***Inference Handlers*** and a set of ***Inference Handlers*** designed for a purpose is labeled as an ***Intelligence Circuit*** or simply ***Intelligence***. This section will outline how SPECTRUM's intelligence is applied to isolate communication outages.

Building the Model — AutoDiscovery

The accurate representation of the infrastructure through the modeling system is the key to being able to determine the fault and the impact of the fault. A significant number of man-years of SPECTRUM's development have gone into creating a modeling system capable of representing not only a wide array of multi-vendor equipment, but also a wide range of technologies and connections that can exist between various infrastructure elements. SPECTRUM has specific solutions for discovering multi-path networks over a variety of technologies supporting many different architectures. SPECTRUM offers support for meshed and redundant, physical and logical topologies based on ATM, Ethernet, Frame Relay, HSRP, ISDN, ISL, MPLS, Multicast, PPP, VoIP, VPN, VLAN and 802.11 Wireless environments — even legacy technologies such as FDDI and Token Ring. SPECTRUM's modeling is extremely extensible and can be used to model OSI Layers 1-7 in a communication infrastructure.

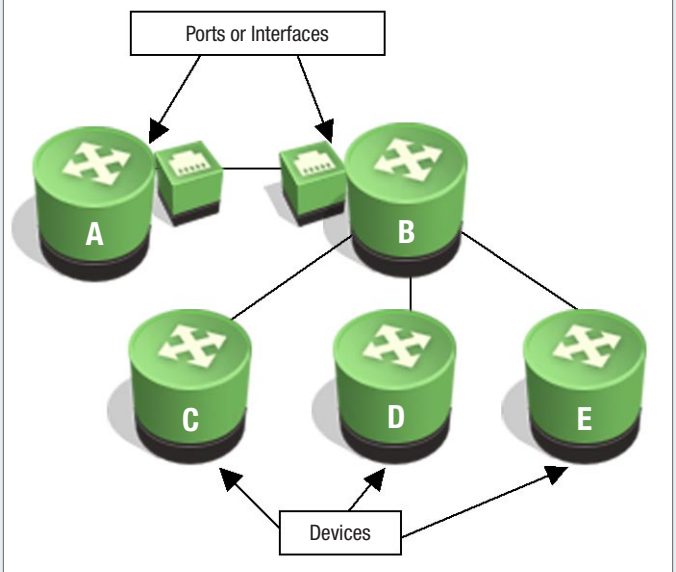
SPECTRUM provides four different methods for building the physical and logical topology connectivity model for any given infrastructure:

- SPECTRUM's ***AutoDiscovery*** application can be used to automatically and dynamically interrogate the managed infrastructure about its physical and logical relationships.We patented our approach to AutoDiscovery in 1996 and SPECTRUM was the industry's first product to discover Layer 2 switch connectivity. SPECTRUM's AutoDiscovery application works in two distinct phases (although there are many different stages within each phase that are not covered here). The first phase is ***Discovery***. When initiated, AutoDiscovery automatically discovers the elements that exist in the infrastructure. This provides SPECTRUM with an inventory of elements that could be managed. The second phase is ***Modeling***. AutoDiscovery uses management and discovery protocols to query the elements it has found to gain information that will be used to determine the Layer 2 and Layer 3 connectivity between managed elements. For example AutoDiscovery uses SNMP to examine route tables, bridge tables and interface tables, but also uses traffic analysis and vendor proprietary discovery protocols such as Cisco's CDP. AutoDiscovery is a very thorough, accurate and automated mechanism to build the infrastructure model.
- Alternately, the ***Modeling Gateway*** can be used to import a description of the entire infrastructure's components, as well as physical and logical connectivity information from external sources, such as Provisioning systems or Network Topology databases.
- The ***Command Line Interface*** or ***Programmatic APIs*** can also be used to build a custom integration or application to import information from external sources.
- ***Graphical User Interfaces*** can be used to quickly point, click and drag & drop to manually build the model.

SPECTRUM's modeling scheme allows a single managed element to be logically broken up into any number of sub-models. This collection of models and the relationships between them is often referred to as the ***semantic data model*** for that type of managed element. Thus a typical semantic data model for a networking device may include a chassis model with board models related to the chassis. Associated to the board models would be physical interface models. Each physical interface model may have a set of sub-interface models associated below them.

SPECTRUM has a set of well-defined associations that define how different semantic data model sets act with one another. When SPECTRUM represents the connectivity between two devices, a relationship is established not only between the two ports that form the link between them, but additionally relationships form between device models and to the corresponding interface and port models of other devices. This is depicted in Figure 3.

Figure 3: Modeling Device and Interface Level Connectivity

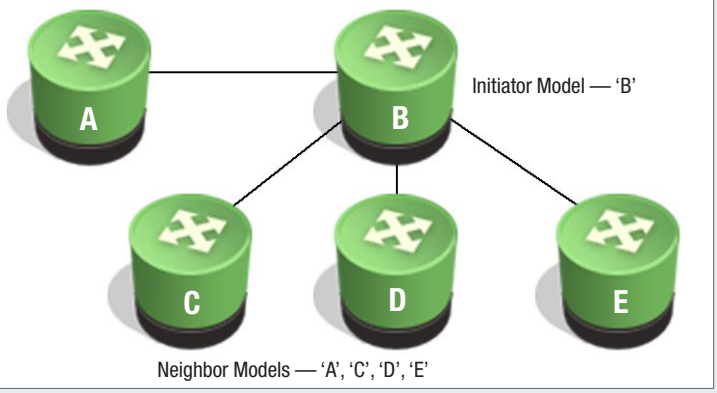


When Does the Analysis Begin?

SPECTRUM can begin to solve a problem proactively upon receipt of a single symptom. Many problems share the same set of symptoms and only through further analysis can the root cause be determined. For communication outages, the ***analysis is triggered*** when a model in SPECTRUM recognizes the communications failures. Failed polling, traps, events, performance threshold violations or lack of response can lead to this recognition. SPECTRUM has intelligence that validates the communication failures through retries, alternative protocols and alternative path checking as part of its "trust but verify" methodology. We will refer to the model which triggered the intelligence as the ***initiator*** model, although it should be noted that more than one model can trigger the intelligence.

The initiator model intelligence requests a list of other models that are directly connected to it. These connected models are referred to as the initiator model's ***neighbors***.

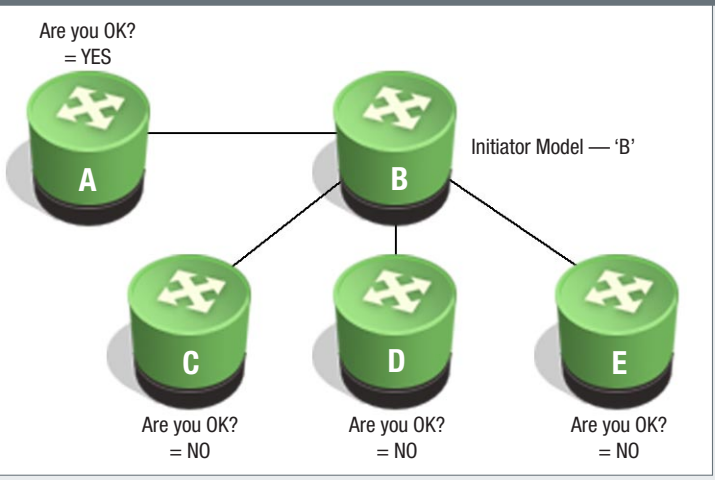
Figure 4: The Initiator Model and Neighbors



With a list of neighbors determined, the intelligence directs each neighbor model to check its current status. This check is referred to as the "***Are You OK?***" check. "***Are You OK***", is a relative term and a unique set of attributes related to performance and availability will vary from model to model based on the real-world capabilities of the device that the model is representing.

When a model is asked "Are You OK?", the model can initiate a variety of tests/checks to verify its current operational status. For example, with most SNMP managed elements the check is typically a combination of SNMP requests but could be more involved by interrogating an Element Management System or as simple as an ICMP Ping. A comprehensive check could include threshold performance calculations or execution of response time tests.

Figure 5: Determining the Health of Neighbors

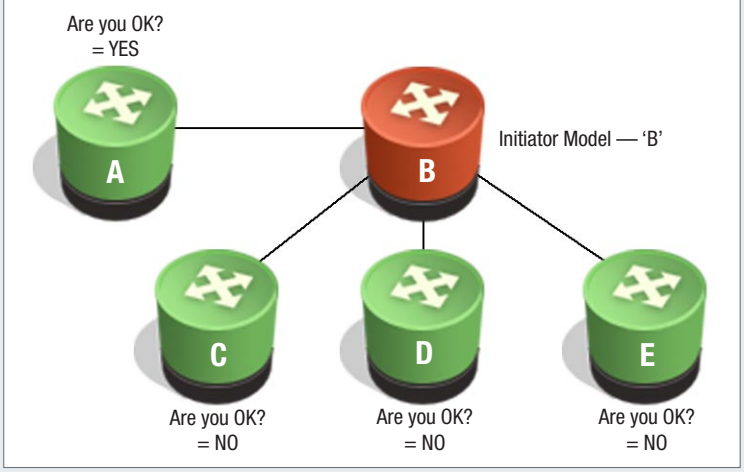


Each neighbor model returns an answer to "***Are you OK?***" and SPECTRUM then begins its analysis of the answers.

Fault Isolation

If the initiator model has a neighbor that responds that it is "***OK***", (Figure 6, Model A), then it can be inferred the problem lies between the unaffected neighbor and the affected initiator (Figure 6, Model B). It is important to note this fault isolation inference can be made without regard to the location of the SPECTRUM management server in the infrastructure. SPECTRUM inference handler algorithms need not understand the relative position of the polling engine to the managed element. In this case, the initiator model that triggered the intelligence is a likely culprit for this particular infrastructure failure. The result? A critical alarm will be asserted on the initiator model, and it is considered the "***Root Cause***" alarm.

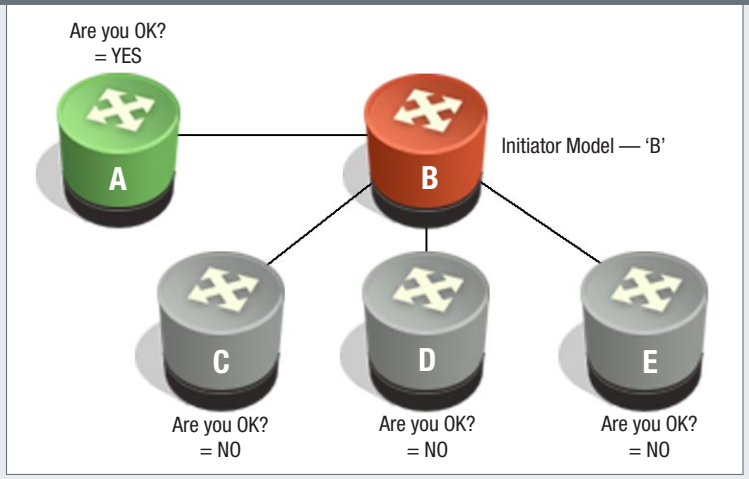
Figure 6: Fault Isolation in Progress



Alarm Suppression

As the analysis continues beyond isolating the device at fault (Figure 7, Model B), the next step is to analyze the effects of the fault; the goal of which is intelligent ***Alarm Suppression***. If a neighbor (Figure 7, Models C, D or E) of the initiator model also responds, "***No, I am not OK***", then this particular neighbor is considered to be affected by a failure that is occurring somewhere else in the infrastructure. As a result, SPECTRUM will place these models into a suppressed condition (Grey Color) because they are alarms symptomatic of a problem elsewhere.

Figure 7: Fault Isolation Complete



Impact Analysis

SPECTRUM continues to analyze the total impact of the fault because of its ability to understand that the individual models exist as part of a larger network of models representing the managed infrastructure.

As such, the intelligence will analyze each Fault Domain, a ***Fault Domain*** being the collection of models with suppressed alarms which are affected by the same failure. These impacted models are linked to the root fault for presentation and analysis. The intelligence provides a measurement of the impact that this fault is having by examining the models that are included within this Fault Domain and calculating a measurement that will serve as the ***Impact Severity***. The impact severity value is meant to supply a ranking system so Operators can quickly assess the relative impact of each particular infrastructure fault in order to prioritize corrective actions.

Event Management System

There are many applications of Event Rules that will allow higher order processing/correlation of event streams. Event Rule processing is required for situations when the event stream is the only source of management information (SPECTRUM can only listen, it cannot talk). For example, SPECTRUM's Southbound Gateway enables SPECTRUM to accept event streams from devices and applications not directly monitored by SPECTRUM. Event Rules can also be applied to perform intelligent processing of events within certain contexts; frequency, sequence, combination. As discussed earlier, SPECTRUM provides six event rule types that can be applied as event rules:

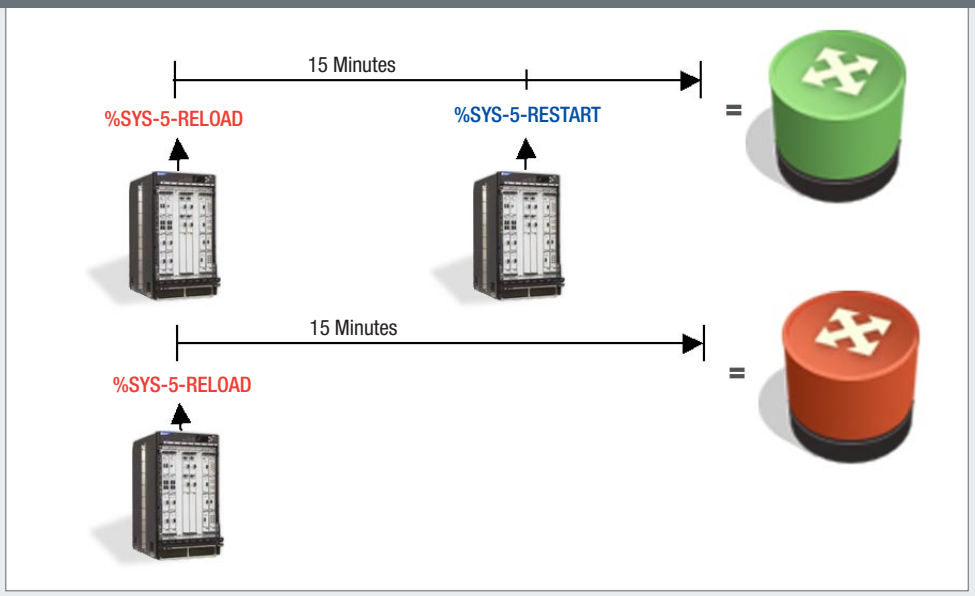
- **Event Pair:** Expected pair event or missing pair event in specified time span.
- **Event Rate Counter:** Events at specified rate in specified time span.
- **Event Rate Window:** Number of events in specified time span.
- **Event Sequence:** Ordered sequence of events in specified time span.
- **Event Combo:** Two or more events, any order in specified time span.
- **Event Condition:** Events parsed for specific data to allow creation of new events based on comparisons of variable bindings, attributes, constants, etc.

SPECTRUM provides many out-of-the-box event rules, but also provides easy-to-use methods for creating new rules using one or more of the event rule types. This section highlights a couple of out-of-the-box event rules and also a few customer examples of event rule applications.

An Out-of-the-Box Event Pair Rule

SPECTRUM has the ability to interpret Cisco syslog messages as event streams. Each syslog message is generated on behalf of a managed switch or router and is directed to the SPECTRUM model representing that managed element. One of the many Cisco syslog messages indicates a new configuration has been loaded into the router. The ***Reload*** message should always be followed by a ***Restart*** message, indicating the device has been restarted to adopt the newly loaded configuration. If not, a failure during reload is probable. An event rule based on the Event Pair rule type is used to raise an alarm with cause ***ERROR DURING ROUTER RELOAD*** if the ***Restart*** message is not received within 15 minutes of the ***Reload*** message. Figure 8 diagrams the events and timing.

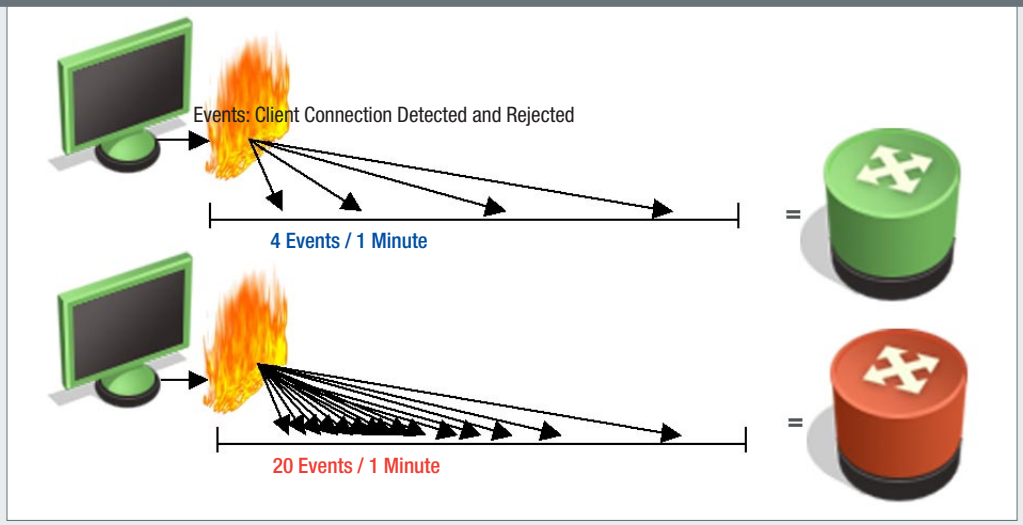
Figure 8: Event Pair Rule



Managing Security Events Using an Event Rate Counter Rule

SPECTRUM is often used to collect event feeds from many sources. Some customers send events from security devices such as Intrusion Detection Systems (IDSs) and Firewalls. These types of devices can generate millions of log file entries. One customer utilizes an Event Rate Counter rule to distinguish between sporadic client connection rejections and real security attacks. The rule was constructed to generate a critical alarm if 20 or more connection failures occurred in less than one minute. Figure 10 depicts this alarm scenario.

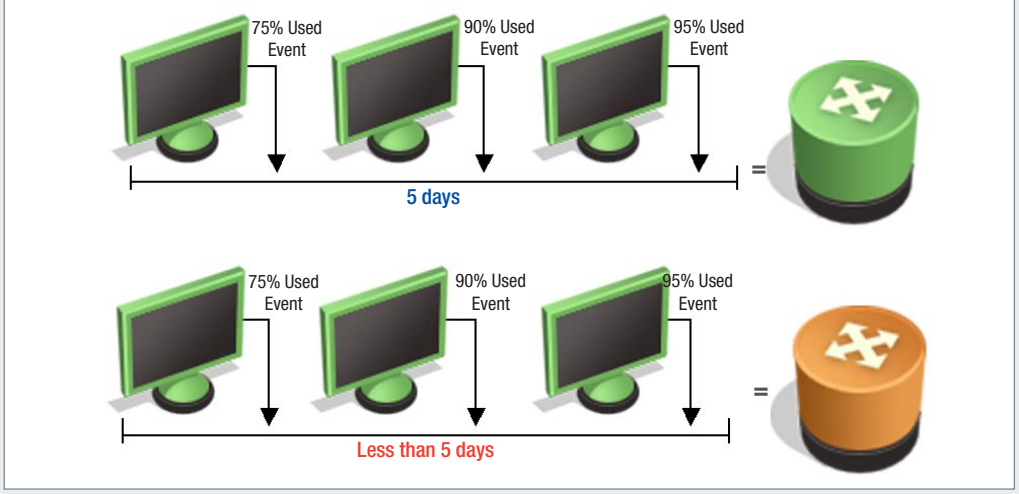
Figure 9: Event Rate Counter Rule



Managing Server Memory Growth Using an Event Sequence Rule

A common problem with some applications is the inability to manage memory usage. There are applications that will take system memory and never give it back for other applications to reuse. When the application does not return the memory, and also no longer requires the memory, it is called a "memory leak". The result is that performance on the host machine will degrade and eventually the "memory leaking" application will fail. At one customer environment this occurs on a Web Server application. The customer has a standard operating procedure to reboot the server once a week to compensate for the memory consumption. However, if the memory leak occurs too quickly, there is a deviation from normal behavior and the server needs to be rebooted before the scheduled maintenance window. The customer employs a combination of progressive SPECTRUM thresholds with an Event Sequence rule to monitor for abnormal behavior. Monitoring was set to create events as the memory usage passed threshold points of 50%, 75% and 90%. If those threshold points are reached in a period of less than one week, an alarm is generated to provide notification to reboot the server prior to the scheduled maintenance window. Figure 9 depicts the fault scenario.

Figure 10: Event Sequence Rule



An Out-of-the-Box Event Condition Rule Combined with an Event Pair Rule

RFC2668 (MIB for IEEE 802.3 Medium Attachment Units) provides management definitions for Ethernet hubs. Within the RFC is the definition of an SNMP trap used to notify a management system when the "jabber state" of an interface changes. Jabber occurs when a device experiencing circuitry or logic failure continuously sends random (garbage) data. The trap identifier simply indicates a change in condition and the variable data portion of the trap indicates whether "jabbering" has started or stopped. An Event Condition rule is applied to create distinct start/stop events by looking at the variable portion of the trap, and an Event Pair rule is used to create an alarm if the "jabbering start"is not closely followed by a "jabbering stop".

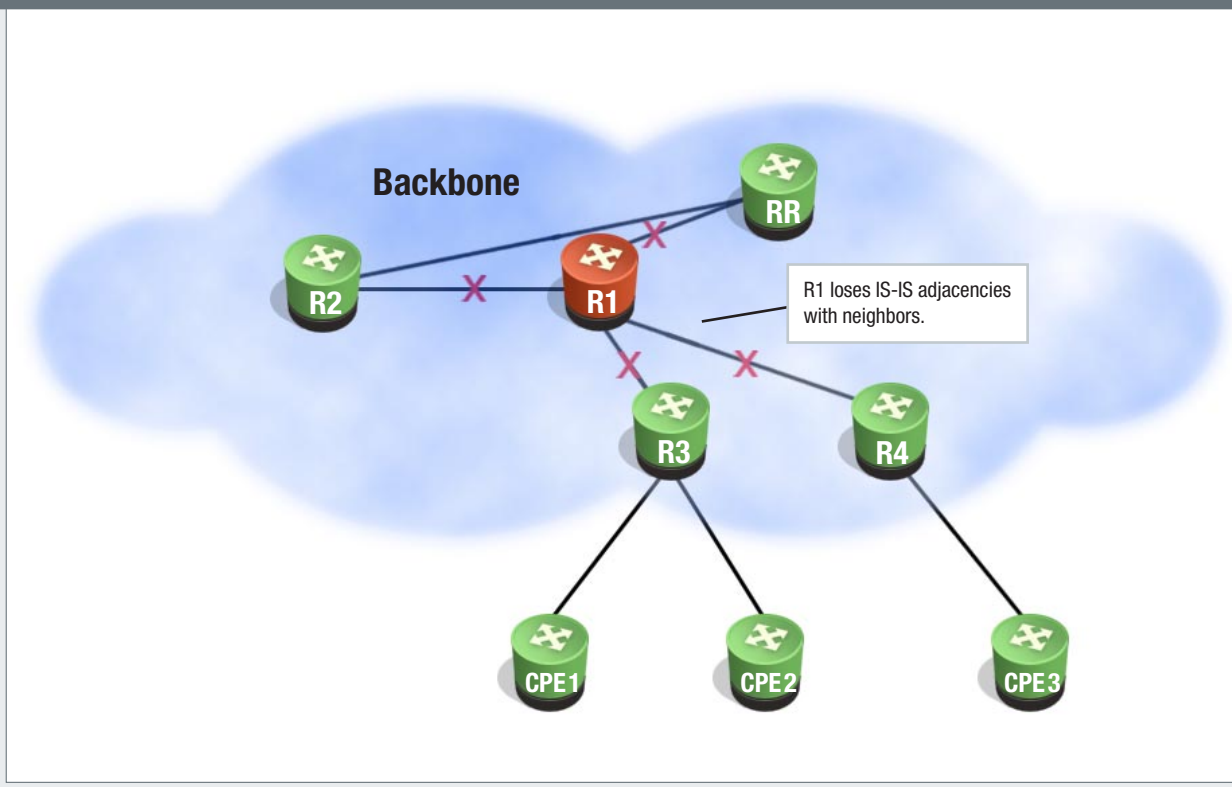
Condition Correlation Technology

There are many, many uses for policy-based Condition Correlation Technology (CCT). CCT is a whole new area of intellectual property innovation for SPECTRUM. For example, consider the complexities of managing an IP network that provides VPN connectivity across an MPLS backbone with intra-area routing maintained by IS-IS and inter-area routing maintained by BGP. Any physical link or protocol failure could cause dozens of events from multiple devices. Without sophisticated correlation capability applied carefully, the network troubleshooters will spend most of their time chasing after symptoms, rather than fixing the root cause.

An IS-IS Routing Failure Example

A specific example experienced by one of our customers can be used to describe the power of Condition Correlation. The failure scenario and link outages are illustrated in Figure 11. The situation occurs where a core router, labeled in the figure as R1, will lose IS-IS adjacencies to all neighbors (labeled in the figure as R2, R3, R4). This also results in the BGP session with the route reflector (labeled in the figure as RR) being lost. This condition, if it persists, will result in routes aging out of R1 and adjacent edge routers R3 and R4. Eventually, the customer VPN sites serviced by these edge routers will be unable to reach their peer sites (labeled in the figure as CPE1, CPE2, CPE3).

Figure 11: IS-IS Routing Failure



This failure causes a series of syslog error messages and Traps to be generated by the routers. The messages and traps that would be received by SPECTRUM are outlined in Figure 12.

Figure 12: Syslog Error Message and Trap Sequence for IS-IS Routing Failure

Source	Type	Message
R1	Syslog message	%CLNS-5-ADJCHANGE: ISIS: Adjacency to R2 (POS5/0/0) Down, hold time expired
R1	Syslog message	%CLNS-5-ADJCHANGE: ISIS: Adjacency to R3 (POS5/0/0) Down, hold time expired
R1	Syslog message	%CLNS-5-ADJCHANGE: ISIS: Adjacency to Rn (POS5/0/0) Down, hold time expired
RR	Syslog message	%BGP-5-ADJCHANGE: neighbor R1 Down BGP Notification sent
RR	Syslog message	%BGP-3-NOTIFICATION: sent to neighbor R1 4/0 (hold time expired) 0 bytes
RR	Trap	BGP Backwards Transition trap, neighbor = R1
R2	Syslog message	%CLNS-5-ADJCHANGE: ISIS: Adjacency to R1 (POS5/0/0) Down, hold time expired
R3	Syslog message	%CLNS-5-ADJCHANGE: ISIS: Adjacency to R1 (POS5/0/0) Down, hold time expired
Rn	Syslog message	%CLNS-5-ADJCHANGE: ISIS: Adjacency to R1 (POS5/0/0) Down, hold time expired

The root cause of all these messages is an IS-IS routing problem related to R1. For many management systems the operator or troubleshooter would see each of these messages and traps as seemingly disparate events on the event/alarm console. A trained operator or experienced troubleshooter may be able to deduce, after some careful thought, that an R1 routing problem is indicated. However in a large environment these events/alarms will likely be interspersed with other events/alarms cluttering the console. Even if the operator or troubleshooter is capable of making the correlation manually, there would be effort and time spent doing so. That time is directly related to costs, lower user satisfaction and lost revenue.

Without condition correlation, the alarm console users would receive notification of ten or more events. However, using a combination of an Event Rule and Condition Correlation, a set of rule patterns can be applied to a Correlation Domain consisting of all core (LSR) routers, enabling SPECTRUM to produce a single actionable alarm. This alarm will indicate that R1 has an IS-IS routing problem, and a network outage may result if this is not corrected. The seemingly disparate conditions that were correlated by SPECTRUM resulting in this alarm will be displayed in the "symptoms" panel of the alarm console as follows:

1. A local Event Rate Counter rule was used to define multiple 'IS-IS adjacency change' syslog messages reported by the same source as a routing problem for that source.
2. A rule pattern was used to make an IS-IS adjacency lost event "caused by" an IS-IS routing problem when the neighbor of the adjacency lost event is equal to the source of the routing problem event.
3. A rule pattern was used to make a BGP adjacency down event "caused by" an IS-IS routing problem when the neighbor of the adjacency down event is equal to the source of the routing problem event.
4. A rule pattern was used to make a BGP backward transition trap event "caused by" an IS-IS routing problem when the neighbor of the backward transition event is equal to the source of the routing problem event.

An HSRP/VRRP Routing Failure Example

Condition Correlation can also provide interesting and useful correlation of events when a link is lost to a router in a Hot Standby Routing Protocol (HSRP) or Virtual Router Redundancy Protocol (VRRP) environment. Consider the scenario of an important customer site that has two redundant routers providing access to the site via HSRP. There are multiple failure scenarios that have been accounted for, but one particular scenario is outlined as shown in Figures 13 and 14.

Figure 13: HSRP Routing Failover

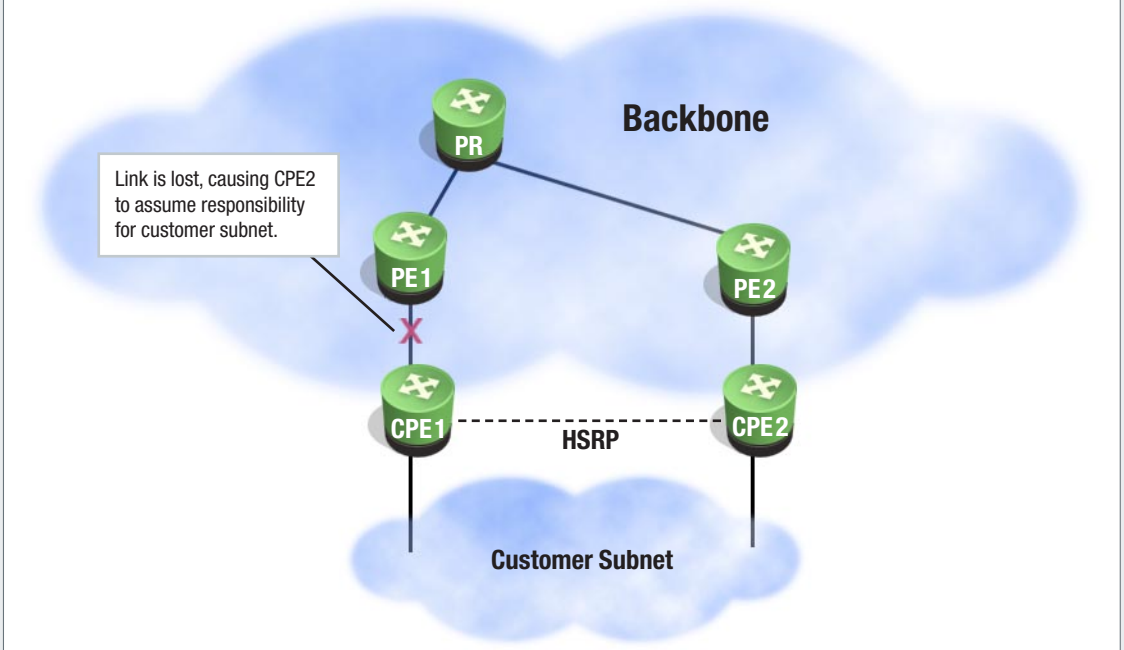


Figure 14: Syslog Error Message and Trap Sequence for HSRP Example

Source	Type	Message
PE1	Syslog message	%LINK-3-UPDOWN: Interface Serial0/1 changed state to down
PE1	Trap	Link down trap, ifIndex=5, ifOperStatus=Down
PE2	Trap	HSRPGrpStandByState, state=Active
NMS	Failed Poll	Device Contact Status Lost, Model=CPE1

For this case, the primary router experiences a failure and the customer's site is still being serviced by the redundant router. It is desirable to have an alarm notification of redundant fail-over and distinguish that from a total site outage. Knowledge from IMT, EMS and CCT are used to provide the root cause analysis.

The seemingly disparate conditions that were correlated by SPECTRUM resulting in this alarm will be displayed in the "symptoms" panel of the alarm console as follows:

1. A correlation domain was created that consists only of the two CPE HSRP routers and the PE router interfaces that connect to these sites.
2. A rule pattern was created correlating the coincidence of an HSRPGrpStandByState event with a state of active and a ***Device Contact Lost*** event to infer a ***Primary Connection Lost*** condition.
3. A rule pattern was created defining that a ***Bad Link*** event is caused by a Primary Connection Lost event.
4. Apply these rule patterns to the HSRP correlation domain(s) to prevent any correlations outside of that scope.

Without these rules, a critical alarm would have been raised on the lost CPE device, and on the connected port model. With these rules, a major (Orange) alarm is raised on the CPE device indicating that the primary connection to the customer is lost. The other conditions will show up in the symptoms table of this alarm.

Applying Condition Correlation to Service Correlation

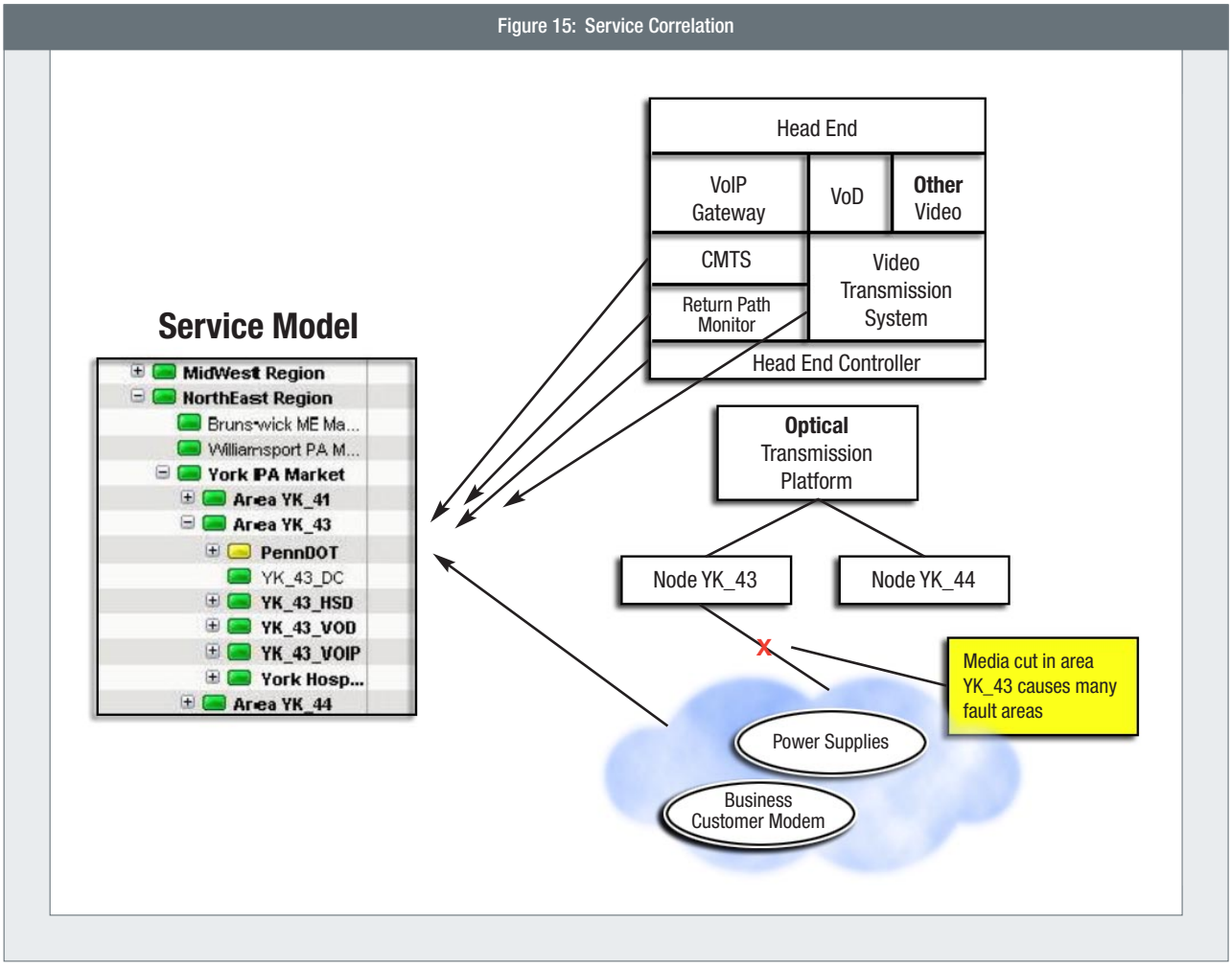
It is common to have several services running over the same network. As an example, in the cable industry, telephone service (VoIP), internet access (High Speed Data), Video on Demand (VoD) and Digital Cable are delivered over the same physical data network. Management of this network is quite a challenge. Inside the network (cable plant), the video transport equipment, video subscription services and the Cable Model Termination System (CMTS) all work together to put data on the cable network at the correct frequencies. Uncounted miles of cable along with thousands of amplifiers and power supplies must carry the signals to the homes of literally millions of subscribers.

With the flood of events and error messages that will be provided by the managed elements, the fact that there is a problem with the service will be obvious. The challenge is to translate all this data into root cause and service impact actionable information. Service impact relevance goes beyond understanding what is impacted; it is also important to understand what is not impacted. It's possible for the video subscription service to fail to deliver VoD content to a single service area, and yet all other services to that area could be fine. Or, a return path problem in one area could cause Internet, VoIP and VOD services to fail, digital cable to degrade, yet analog cable would still function normally.

In the case of a media cut in one area, the return path monitoring system and the head end controller would report return path and power problems in that area. The CMTS would provide the number of Cable Modems off-line for the node. The video transport system would generate tune errors for video subscriptions in that area. Lastly, any business customer modems that are being managed will become contact lost to the management system.

SPECTRUM can make sense of the resulting deluge of events by using the service area of the seemingly disparate events as a factor in the Condition Correlation. If the service areas and services are modeled in SPECTRUM, Condition Correlation can be used to determine which services in which areas are affected and the root cause or causes.

Figure 15: Service Correlation



Conclusion

Change is a constant, requiring any management system to be automated, adaptable, and extensible. The number of multi-vendor, multi-technology hardware and software elements (each with their own different versions) in a typical IT environment exponentially increases the complexity of managing a real-time, on-demand IT infrastructure. SPECTRUM currently supports several thousand distinct means to automate root cause analysis across over 1,000 multi-vendor network, system, application and security infrastructure components. A listing of SPECTRUM's supported vendors and devices can be found online at: http://www.aprisma.com/products/locator/.

Knowing about a problem is no longer enough. Predicting and preventing problems, pinpointing their root cause, and prioritizing issues based on impact are requirements for today's management solutions. The number and variety of possible fault, performance and threshold problems means no one approach to root cause analysis is suited for all scenarios. For this reason, SPECTRUM incorporates model-based IMT, rules-based EMS, and policy-based CCT to provide an integrated, intelligent approach to drive efficiency and effectiveness in managing IT infrastructure as a business service.

To learn more about how SPECTRUM can help you manage what matters, please call us directly, toll-free in the U.S. at (877) 437-0291 or worldwide at +1 603 334 2100. E-mail works too — sales@aprisma.com.

About Concord Communications

Concord Communications, Inc. (NASDAQ: CCRD) is a global provider of Business Service Management (BSM) software that reduces IT downtime, improves capacity planning, and optimizes service level management – thereby enabling customers to increase revenue and productivity. Built on more than 100 technology patents, Concord's family of world-class solutions addresses the needs of enterprise customers across 17 vertical markets, managed service providers, and both wireless and wireline telecommunications carriers. These solutions enable organizations of all sizes to map IT services to business needs, measure the actual end-user experience, and manage voice or data applications, systems, and networks.

More than 7,500 customers worldwide use Concord's software, including 23 of the world's 24 largest service providers, 14 of the world's 20 largest banks, and 11 of the world's 20 largest insurance companies. Founded in 1986 and headquartered in Marlboro, Massachusetts, USA, Concord maintains offices around the globe and can be found on the web at www.concord.com.