A Survey of Event Correlation Techniques and Related Topics

Michael Tiffany
3 May 2002

1. Introduction

As the world relies on computers to do increasingly important and complicated tasks, the field of network management becomes ever more important. Whether ensuring that the e-commerce servers are constantly up and accessible or simply making sure that company executives can send and receive e-mail, the network engineers are heavily relied upon to ensure consistency. Whenever there is a problem, the engineers in charge of maintaining the network need to be able to quickly pinpoint the source of the problem, whether it is as simple as a stopped mail daemon or as complicated as a fiber cut between satellite offices. The specialization of event correlation aids in this endeavor, by attempting to consolidate the information received into a concise, clear package that can be quickly deciphered.

2. Terms and Definitions

Before discussing some of the various methodologies being employed today to correlate events, I will first briefly define and discuss events and event correlation.

2.1 Events

An event in network management is typically defined as a piece of information dealing with a happening in the network, and may also be referred to as an alarm, due to its nature usually being something causing problems. The network management system can be programmed to methodically obtain these events by polling devices, the devices can send events to the management system, or, as is commonly the case, a hybrid combination of the two is used. Examples of events are hardware or software failures, security violations, and performance issues. Liu et al [1] make further distinctions, defining and making a distinction between primitive events and composite events. A primitive event is very simple; it may be a link down alarm or a failed login attempt. Composite events, as the name suggests, combine primitive events, often integrating a timing aspect. This can be useful to detect multiple failed logins, which could indicate a break-in attempt, or a flapping interface which is repeatedly going down and coming back up, and may be less of a concern if it is not consistently down for long periods of time.

2.2 Event Correlation

Event correlation is defined in many different ways, but in its barest essence, an event correlator attempts to do exactly as the name suggests: associate events with one another in useful ways. Sometimes the sheer number of events which come in can be enough to overwhelm a engineer who cannot possibly treat each symptom separately. The object of event correlation is to attempt to pinpoint larger problems which could be causing many different symptoms to emerge. According to Guerer et al [2] there are several subcategories of event correlation, including compression (deduplication), count,

suppression, and generalization. Compression reduces multiple occurrences of the same event into a single event, likely with some kind of counter. This allows an engineer to see that an event is recurring without having to see every instance individually, especially if the problem is already known, but events keep being received. Count is defined to be somewhat similar to compression: it is the substitution of a specified number of similar alarms with a single alarm. It is important to note that these need not necessarily be the same event, and also that there is a threshold associated with such a relation. Suppression associates a priority with alarms, and may choose to hide a lower priority alarm if a higher priority alarm exists. Finally, in the practice of generalization, alarms are associated with some sort of a superclass which is reported rather than the specific alarm. This could be seen to be useful to correlate events referring to multiple ports on the same switch or router if it has completely failed; it is unnecessary to see each particular failure if it can be determined that the entire unit is having problems.

There also exist other broad categories of event correlation, as defined by Hasan et al [3]. A seemingly obvious method of event correlation is attempting to determine some type of causal relationship (i.e., event A causes event B). An instance of this type of correlation is seen when a connectivity problem is traced to a particular failing piece of hardware. Another type of correlation has to do with the previously discussed composite events, and is a temporal correlation. In essence, there is a time period associated with each possible correlation, and if the proper events occur during a particular time period, they may be correlated. Hasan claims that any causal system can be represented as a temporal system, and that there are, in fact, instances of relations which can be

determined through temporal correlation and not event correlation, making temporal correlation the more powerful technique. The practice of event correlation is useful and necessary not only to reduce the number of alarms but also to do some processing of the likely causes to take some of the workload off of the network engineer.

3. Event Correlation Systems

Many factors come into play in the process of event correlation. The two most important aspects of an event correlation system are the speed with which correlation occurs, and the accuracy of the data returned by a correlation. Obviously, even an infinitely fast system that returns incorrect information is just as bad as a system which produces a perfect diagnosis every time, but only does so long enough after the fact so as to be of no use in diagnosing and fixing the problem. A system must have an appropriate combination of these two characteristics in order to be considered effective.

3.1 Rule-based Systems

A somewhat traditional approach to event correlation is that of rule-based analysis. In this approach, sets of rules are matched to events when they come in. Based on the results of each test, and the combination of events in the system, the rule-processing engine analyzes data until it reaches a final state. It will then report a diagnosis, which could include, unfortunately, none at all, depending on the depth and capability of the ruleset. Unfortunately, this approach does not necessarily perform well in respect to either of our criteria. For the results to be very accurate, an excessive amount of expert knowledge is typically needed to input the correct rules and keep them updated in case of any changes or new data, though this will be discussed in more depth later. In addition,

the rigidity of the path through the rule sets makes it so that events are may always be compared with an inordinate amount of test cases, slowing the system down and making correlation even more difficult.

3.2 Codebook Systems

The codebook approach is somewhat similar to the rule-based approach, but rather than treating events separately, they are grouped into an alarm vector which represents all of the events. This alarm vector is then matched to problem signatures in a so-called codebook. Kliger et al [4] describe a codebook system, explaining that the codebook consists of an optimal set of alarms which must have distinguishable characteristics. In other words, no two alarms can have the same signature, as there would be no way to correctly choose one over the other. In order to optimize performance, a design should use small sets of symptoms while still providing the best guess of the cause. This can be done in the creation of the codebook by forcing a minimum Hamming distance, a measure of differing symptoms, between different causes and by reducing otherwise indistinguishable cases into one which encompasses all of them. The codebook approach always produces a diagnosis, as opposed to the rule-based scheme, though some diagnoses will be more likely to be correct than others. Unfortunately, the codebook technology needs the same expert knowledge as a rule-based system in order to accurately populate the codebook, though Yemini et al [5] claim that problem signatures can be created automatically. Unfortunately, many of the details of this system appear hidden behind corporate patents, so many such claims have little researchable evidence to back them up. Gupta et al [6] proposes an algorithm to reduce the size of the codebook.

This algorithm will find and eliminate events which will not help distinguish between problems, and will eliminate complementary signatures, among other things. With a smaller, more effective codebook, the speed of the system is likely to increase. In fact, it is likely that a codebook system will run faster than a corresponding rule-based system simply because there are less comparisons for each event, but empirical data has proven difficult to come by.

3.3 Artificial Intelligence Systems

A approach that is radically different from the rule-based and codebook approaches uses various forms of artificial intelligence (AI). There are many different types of artificial intelligence, and event correlation techniques have been proposed which utilize various combinations of them, including Bayesian belief networks and expert systems. AI systems have an advantage in that, if well-programmed, they have the capability to be somewhat self-learning, helping to eliminate the continuous need for the expert knowledge of the previous systems. They also have the capability to sift through data at least as fast as the other systems to produce their results. Sheppard et al [7] present a comparison between the Dempster-Shafer method (a probabilistic approach) and the nearest neighbor approach, a form of which is used in the codebook scheme. In specific, the paper deals with the possibility of bit error, and notes that the Dempster-Shafer method is not only more likely to produce the correct diagnosis, but is very likely to be able to correctly diagnose problems within two guesses. A benefit of a probabilistic method such as this is the capability to produce probabilities for each problem so they can be easily ranked.

In a similar manner, Wietgrefe et al [8] propose a Cascade Correlation Alarm Correlator, which is based on a neural network, another AI structure. They claim that when they incorporate a technique called inverse learning into their scheme, their system will never return a wrong answer as noise in the system increases. The paper compares this approach to a codebook approach, and determines that without inverse learning, the CCAC returns correct diagnoses more often than a codebook, and with inverse learning, diagnoses are returned less often, but are guaranteed to be correct. This could help to resolve a sticking point which appears to be preventing a more widespread use of artificial intelligence systems, since many people seem to be uncomfortable with the fact that an AI system typically introduces some uncertainty into the system. It is difficult, if not impossible, to exactly trace the workings of the system on any given event, and this concerns many engineers, since they are used to determinism in their work. People are generally willing to receive an answer a smaller percentage of the time with the benefit of being guaranteed the right answer, since they seem to be innately untrusting of machines unless they can follow the same steps as the machine. Even the option of always receiving feedback is not enough to change peoples beliefs, especially when it has a possibility, albeit usually small, of being incorrect. There is hope that an approach may be devised which uses the best characteristics of different AI techniques to produce the best results. Lin [9] proposes a system which uses a hybrid combination of case-based reasoning, rule-based reasoning, model-based reasoning, Bayesian networks, and neural networks to correlate events. Similarly, Guerer et al [2] propose a technique which uses an expert system for filtering, a neural network for correlation, and case-based reasoning

for fault identification and correcting and updating the fault library.

Huard [10] uses fault management on XUNET as the basis of his theories centering around artificial intelligence. He proposed the construction of a belief network for each particular fault, working through such faults as link down, no connection, and no logfile mail received. Huard proposes that once these smaller belief networks, they can be combined to provide a complete belief network for all systems. A difficulty with this process lies in the probability assignment step, since each relationship in a belief network has some probability associated with it. If the probabilities are too different from the correct values, the network representation will be, at best, misguided. With Lazar, Huard [11] furthers this proposal, presenting a method to pinpoint the cause of a fault or set of faults at a minimum cost. The cost may be defined in time, network resources, or any other valid, simple measurement, and involves the assignment of probabilities to each symptom relating it to the various possible faults. Such a strategy could prove useful for network engineers who are under pressure to produce correct results quickly. Provided that artificial intelligence can overcome the stigma associated with probabilistic methods, it may become a useful technology for event correlation for such reasons.

4. Rule Creation

As we have seen, a major part of most event correlation systems is creating the initial relationships, or rules. If these relationships are incorrect, any information received from them will be just as incorrect, and if they are incomplete, so are any possible results. Traditionally, these rules are input by experts during the creation of any such system. With the highly dynamic nature of networks today, however, this becomes exceedingly

difficult, if not impossible to maintain. Therefore, any automation of this process is welcomed by engineers. Oates et al [12] suggest that potential relationships can be extracted from properly formatted system logs. Unfortunately, many log entries are not detailed enough to extract such information, and again the dynamism of the network may overwhelm the ability of this scheme to keep up with changes. Sterritt [13] postulates that artificial intelligence combined with human knowledge is more useful than either technique alone, but lacks many details and support for his ideas. A useful technique for dealing with a dynamic network comes from Breitbart et al [14]. Topology discovery is typically regarded as a very useful tool for event correlation, since knowledge of the network topology allows a network management system or an engineer to trace multiple problems to find the (hopefully) single cause. This paper acknowledges that layer-3 topology is usually fairly easily obtained from routing information, but that in any particular subnet, it is likely that there are layer-2 elements such as switches and bridges which need to be considered. An algorithm is presented to establish links between switches and routers, and also those between different switches. The algorithm works not only over single subnet switched domains, but also on multiple subnets and even subnets which utilize VLANs. The authors mention that in testing a prototype using their algorithm, they discovered several links which were not present in the network administrator's topology maps. Tools such as these discussed here can be useful and may even be necessary in the creation of a useful, accurate event correlation system.

5. Conclusion

There is a distinct lack of data by which event correlation systems can be compared, and

so we must examine the theory behind them rather than empirical evidence. Most systems can be shown to work better than most others under specialized circumstances, and, as such, any data present will likely be skewed toward the author's preferred system. Every network is different in topology, symptoms, and their causes, so there cannot be a single methodology which is the best choice for every single network. Systems which utilize multiple methodologies may be capable of handling more situations, but will likely be more complicated to make up for it.

Bibliography

1. *Composite Events for Network Event Correlation*
G. Liu, A. K. Mok, E. J. Yang
May 1999, IEEE/IFIP International Symposium on Integrated Network Management

2. *An Artificial Intelligence Approach to Network Fault Management*
Denise W. Guerer, Irfan Khan, Richard Ogler, Renee Keffer
1996, SRI International

3. *A Conceptual Framework for Network Management Event Correlation and Filtering Systems*
Masum Hasan, Binay Sugla, Ramesh Viswanathan
May 1999, IEEE/IFIP Intonation Conference on Integrated Network Management

4. S. Kliger, S. Yemini, Y. Yemini, D. Ohsie, S. Stolfo
1995, Integrated Network Management

5. *High Speed & Robust Event Correlation*
Yechiam Yemini, Shaula Alexander Yemini, Eyal Mozes, David Ohsie
1996, IEEE

6. *Preprocessor Algorithm for Network Management Codebook*
Minaxi Gupta, Mani Subramanian
April 1999, 1st USENIX Workshop on Intrusion Detection and Network Monitoring

7. *Improving the Accuracy of Diagnostics Provided by Fault Dictionaries*
John W. Sheppard, William R. Simpson
1996, Proceedings of the 14th IEEE VLSI Test Symposium

8. *Using Neural Networks for Alarm Correlation in Cellular Phone Networks*
Hermann Wietgrefe, Klaus-Dieter Tuchs, Klaus Jobmann, Guido Carls, Peter Froelich, Wolfgang Nejdl, Sebastian Steinfeld
May 1997, International Workshop on Applications of Neural Networks to Telecommunications (IWANNT)

9. *A Hybrid Approach to Fault Diagnosis in Network and System Management*
Along Lin
1998, Hewlett-Packard Laboratories

10. *Probabilistic Reasoning for Fault Management on XUNET*
Jean-Francois Huard
1994, AT&T Bell Labs

11. *Fault Isolation based on Decision-Theoretic Troubleshooting*
Jean-Francois Huard, Aurel A. Lazar
1996, Columbia University

12. *Automatically Acquiring Rules for Event Correlation From Event Logs*
Tim Oates, David Jensen, Paul R. Cohen
1997, Computer Science Technical Report 97-14 (Umass)

13. *Discovering Rules for Fault Management*
Roy Sterritt
April 2001, 8th Annual IEEE International Conference on the Engineering of Computer Based Systems (ECBS)

14. *Topology Discovery in Heterogeneous IP Networks*
Yuri Breitbart, Mios Garofalakis, Cliff Martin, Rajeev Rastogi, S. Seshadri, Avi Silberschatz
2000, Proceedings of IEEE INFOCOM