An Alarm Correlation Algorithm for Network Management Based on Root Cause Analysis

Dae Sun Kim*, Hiroyuki Shinbo*, Hidetoshi Yokota*

*KDDI R&D Laboratories, Inc. Fujimino, Saitama, Japan

da-kim@kddilabs.jp, shinbo@kddilabs.jp, yokota@kddilabs.jp

***Abstract*—The alarm correlation is an essential function of network management systems to provide detection, isolation and correlation of unusual operational behaviour of telecommunication network. However, existing alarm correlation approaches still rely on the manual processing, and depend on the knowledge of the network operators. Since, the telecommunication network produces a number of alarms which are so called the alarm floods, it could be very difficult for the network operators to detect the root cause problems in a short period of time. Therefore, we propose the alarm correlation algorithm which is able to isolate and correlate the root causes in a very short time. In addition, we show that the proposed algorithm performs well in terms of efficiency of analyzing alarms and accuracy of identifying root cause.**

***Keywords*— Network Management, Alarm Floods, Alarm Correlation, Root Cause Analysis**

1. INTRODUCTION

Considering the growing complexity of Today's networks, the operational efficiency and accuracy are required to handle the telecommunication network for network management systems. A network produces daily a number of data such as performance indicators and alarms. In particular, the area of fault management remains a key problem which is difficult to identify and predict the root cause problems in a short period of time for network operators. A root cause alarm[1][2] is the initiating cause problem which leads to an effect of other alarms. The incoming rate of network alarms often becomes too high for a network operator so that, the operator may not have enough time to recognize the network state without alarm correlation. The purpose of alarm correlation is to reduce a number of alarms and identify the root cause alarm from the number of alarms. A number of approaches[3][4] have been proposed for alarm correlation. Current alarm correlation approaches can be categorized as: Rule-Based approach[5][6], Codebook-Based approach[5][7], Case-Based approach[4][8], and Mining-Based approach[9][10][11]. The Rule-Based and Codebook based approaches have a drawback since the rule sets[6] and codebook[7] are specified based on experiences of the operators. Therefore, an excessive amount of expert knowledge is typically needed to specify the rule sets and the codebook. In the Case-based approach, if there is no matching case[8] for the current problem, it is unable to provide the solution for this problem in a short period time. Thus, it is difficult for the operators to make new cases that are not yet known. In the Mining-Based approach, many of the proposed algorithms[9][10][11] are not suitable to analyze the frequent event sets[11] in a short period time because of their long processing time. According to our investigations, most of the alarm correlation approaches still rely on the manual processing, and depend on the knowledge of the network operators which is a time consuming process. To address the above problems, an alarm correlation technique should diagnose the given problems in a very short time without expert knowledge. Moreover, the alarm correlation technique is expected to be lightweight and performed on any layered network architecture. Therefore, we propose a novel alarm correlation algorithm to identify the root cause alarm based on the TCP/IP model[12] so that, it can run at any types of environment, and topology on the Internet. In our approach, all alarms are classified according to an identifier of each TCP/IP protocol layer (e.g. Port number in TCP and UDP, IP address, Protocol type and so on)[12] and then clustered in order to indentify the root cause alarm according to the cause and effect relationship[13] between alarms without expert knowledge. The cause and effect relationship is that when alarm *A* happens, another alarm, alarm *B* is more likely to happen. The main idea of the proposed algorithm is that the fault occurring at each layer can be classified by the identifier, and the identifier is a key information to find the cause and effect relationship between the upper and lower TCP/IP layers. In general, the lower layer problems affect the higher layer problems as well as the neighbor nodes. In other words, the lower layer alarms appear as the root cause alarms more frequently than the higher layer alarms.

Our paper is organized as follows. In Section 2, we briefly introduce the current alarm correlation approaches. Detailed descriptions of the proposed alarm correlation algorithm are presented in Section 3. Then we show the experimental results in Section 4. Finally, we conclude in Section 5.

2. RELATED WORKS

In this section, we discuss four such alarm correlation approaches: *Rule-Based Alarm Correlation, Codebook-Based Alarm Correlation, Case-Based Alarm Correlation and Mining-Based Alarm Correlation.*

2.1 Rule-Based Alarm Correlation

The rule-based approach has been frequently used for alarm correlation. In this approach, the network operator identifies all of the events such as alarm, alert, error, threshold violation, and so on. After that, the network operator analyses the root causes and the symptoms[2] of the each root causes. Finally, the specific rule is created to process each of these events. However, the rule-based approach has a drawback since the rule set is specified based on human experiences. Therefore, an excessive amount of expert knowledge is typically needed to specify the rule sets.

2.2 Codebook based Alarm Correlation

The Codebook-Based approach is based on a representation of relationships between problems and symptoms as a matrix[5][7]. In this approach, the matrix is represented by the causality graph[5] so that there are only direct cause and effect relationships between problems and symptoms in the causality graph. The symptoms which are caused by a problem are represented by a code that identifies the problem. The code[5] is a sequence of values from *0* to *1*. The appearance of a particular symptom is denoted by *1*, and *0* means the symptom has not been observed. Therefore, the number of problems can be detected by the codebook. Since the codebook correlation is performed only once to detect a root cause problem, it is very efficient to detect the problems of network in terms of speed and accuracy. However, the expert knowledge is required to construct the codebook. Furthermore, a change in the network requires regenerating the codebook which is a time consuming process.

2.3 Case based Alarm Correlation

The case-based approach is a method to solve problems using the experience of the past cases. In this approach, problems are solved by the stored solutions. When a problem is solved, this solution is stored to be use in the future. However, if there is no matching case for the current problem, this approach is unable to provide the solution for this problem. Furthermore, it is impossible to make new cases that are not yet known in a short period time. Therefore, it also needs the expert knowledge to build the solution database[8], which is a time consuming process.

2.4 Mining based Alarm Correlation

In this approach, frequent event sets can be detected by several mining algorithms[9][10][11]. It is very useful to detect the frequent event sets, and identify a root cause alarm among the collected alarms. However, many of the proposed algorithms are not suitable to analyse the frequent event sets in a short period time because of their long processing time.

3. PROPOSED ALARM CORRELATION SCHEME

This section describes the proposed alarm correlation algorithm and illustrates it in detail with some cases.

3.1 Background Knowledge Representation

The Internet is a global system of interconnected networks that use the TCP/IP protocol suite[12]. Moreover, most of the telecommunication networks existing today are a part of the Internet. Therefore, our alarm correlation approach focuses on the cause and effect relationship between each layer of TCP/IP model to determine how a problem of network happened and to understand why it happened. In a layered network architecture, the services are grouped in a hierarchy of layers. An entity of layer *N* uses only services of layer *N-1* and it provides services only to layer *N+1* as shown in Figure 1. When talking about two adjacent layers within a node, the higher layer is a service user and the lower layer is a service provider. As a result, a relationship between adjacent layers can be defined as follows:

Definition 1. For any given two layers *N* and *N-1* within a node, *N* is a higher layer than *N-1*. Therefore, if *N-1* layer is not able to provide any service, *N* layer cannot provide any service to *N+1* layer. In other words, the lower layer problems always affect the higher layer problems.

In addition, each layer provides a protocol to communicate with its peer. Thus, N layer must be able to receive data from the peer N layer. Therefore, a relationship between identical peer layers can be defined as follows:

Definition 2. For any given two identical layers *N* and peer *N* between communicating nodes, if *N* layer is not able to provide any service, peer *N* layer cannot provide any service data unit which is generated by *N* layer to peer *N+1* layer. In other words, any layer problems always affect the its peer layer problems.

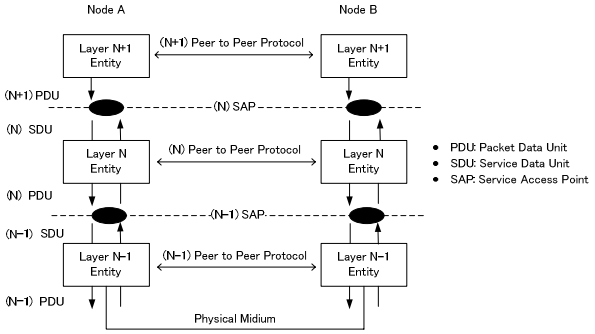


Figure 1. Layered architecture of TCP/IP reference model

The highest layer cannot be a service provider within a node but it can be the service provider for the highest layer of peer nodes. As a result, it can be defined as follows:

Definition 3. Highest layer problems within a node are either generated by itself or by lower layer problems. In the former case, the highest layer is not affected by the lower layer problems but the highest layer of peer node can be affected by it.

3.2 Root cause analysis

A root cause is the initial cause of other alarms and it should be identified and removed from the network. Otherwise, a number of alarms may be generated continuously. Table 1

shows a sample of alarm data set. ("Non-priv" denotes the set {1024,...,65535} of non-privileged ports.)

Table 1. A sample of alarm data set

Seq.	Src-IP	Src-Port	Dst-IP	Dst-Port	Description
1	IP1	80	-	-	Port Down
2	IP2	53	-	-	Port Down
3	IP3	Non-Priv	IP1	80	HTTP is unavailable
4	IP4	Non-Priv	IP1	80	HTTP is unavailable
5	IP4	Non-Priv	IP2	53	DNS is not reachable
6	IP1	-	-	-	IP1 Down
7	IP1	80	-	-	HTTP Down
8	IP2	53	-	-	DNS Down

As the Table 1 shows, *IP1* and *IP2* are the HTTP server and the DNS server respectively. In addition, *IP3* is the client of *IP1* and *IP4* is the client of both *IP1* and *IP2* since *IP3* and *IP4* alarms are targeted at *IP1* and *IP2*. In our alarm correlation approach, at first, all alarms are grouped according to source IP address and then sorted by each layer which range from application layer to the physical layer as shown in Figure 2. To sort by each layer, our algorithm needs to know which alarm belongs to which layer. For instance, since HTTP and DNS in Table 1 belong to application layer, those alarms are grouped into application layer alarm. In this manner, other alarms in Table 1 are grouped into their corresponding layers. The grouping of alarms makes it easier to manage a large number of alarms generated for the identical network problem. According to our definition 1 in Section 3.1, the lower layer problems affect the higher layer problems. Therefore, "*IP1 Down*", "*Port Down*", "*HTTP Unavailable*" and "*DNS Unreachable*" which are the lowest layer alarms in this example can be the root causes of each group in Figure 2.

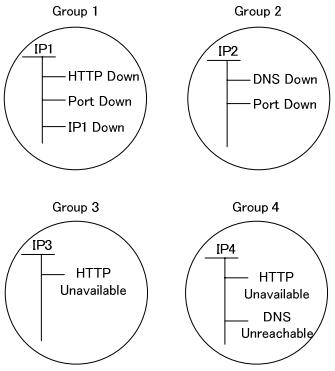


Figure 2. Alarm Grouping

Considering our definition 2, if an alarm is targeted at other alarm, they are merged into a group which we call the cluster[1]. Therefore, two "*HTTP Unavailable*" alarms in Group 3 and Group 4 are merged into Cluster 1. In addition, "*DNS Unreachable*" alarm is merged into Cluster 2 as shown in Figure 3 since the alarms in Group 3 and Group 4 are targeted at Group 1 and Group 2 respectively. The clustering of alarms also makes it easier to manage a huge number of alarms generated for problems in related areas. Finally, "*IP1 Down*" and "*Port Down*" are identified as the root causes of Cluster 1 and Cluster 2 respectively in Figure 3. In addition, two "HTTP Unavailable" alarms in Table 1 should be generalized to reduce multiple occurrences of identical alarms into a single alarm. And to conclude, a combination of grouping, clustering and generalizing is a key idea to determine the root cause of one or more alarms in this paper.

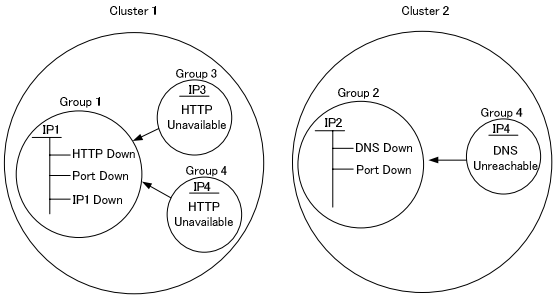


Figure 3. Alarm Clustering

3.3 Alarm correlation algorithm

The alarm correlation is not an easy task because the telecommunication networks usually generate a huge number of alarms every day. An interesting observation is that the most of alarms are noisy alarms which reduce the efficiency of analysing alarms and the accuracy of identifying the root cause. Thus, the alarm generalization process is required to reduce an amount of noisy alarms. To achieve this, Figure 4 shows the pseudo-code of the proposed alarm grouping algorithm which is the first step of alarm correlation.

```
Input: A set E of alarms (a1, a2, ··· an)
Output: A set G of groups (g1, g2, ··· gn)
Algorithm:
 1:  G = Group the set E by identical src_ip ;
 2:  let i = 1 ;
 3:  While G is not empty {
 4:     Sort the  gi in the order by layer ; // From application layer to physical layer
 5:     While identical alarms a, a` in gi exist do {
 6:        Remove a` from gi ;
 7:     }
 8:  i++;
 9:  }
10:  Output G
```

Figure 4. Alarm grouping algorithm

The main idea of the alarm grouping algorithm is to group alarms according to identical source IP address(Line 1). By doing so, the identical alarms can be detected at each group and it makes it easier to understand what happened at each node. After that, all alarms in each group are sorted in order by layer which range from application layer to the physical layer (Line 4). In addition, many identical alarms are removed in each group (Lines 5 and Line 6). Figure 5 shows an example of generalization process for the identical alarms. The difference between *Alarm 1* and *Alarm 2* in Figure 5 is only the time stamp value. Therefore, those alarms can be generalized as a single alarm in order to reduce a huge number of alarms.

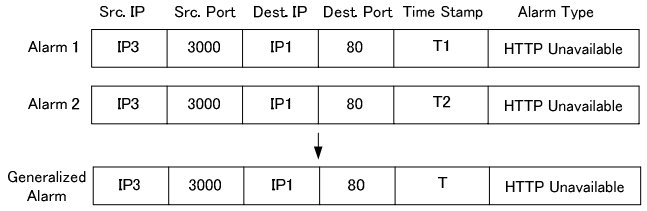


Figure 5. An example of generalization process for identical alarms

Finally, line 10 will output a set G of groups which include the generalized alarms.

```
Input: A set G of groups (g1, g2, ··· gn)
Output: A set C of clusters (c1, c2, ··· cn)
Algorithm:
 1:  n = Count(G) ;     // Count total number of groups in set G
 2:  For i = 1 to n {
 3:    k = Count( gi )   // Count total number of alarms in group gi
 4:    For j = 1 to k {
 5:      gi->aj->CK_ID = i ;     // Initialize Cluster Key ID(CK_ID)
 6:    }
 7:  }
 8:  T = G     //Store all alarms in table T
 9:  let j = 0 ;
10:  While T is not empty {
11:    For i = 1 to n {
12:      If aj is targeted at gi
13:        aj->CK_ID = i ;
14:        Break;
15:    }
16:    j ++ ;
17:  }
18:  C = Group the table T by identical CK_ID ;
19:  let i = 1;
20:  While C is not empty {
21:    Sort the ci in the order by layer ; // From application layer to physical layer
22:    While similar alarms a, a` in ci exist do {
23:      Remove a` from ci ;
24:    }
25:   i ++;
26:  }
27:  Output C
```

Figure 6. Alarm clustering algorithm

The main purpose of the alarm clustering algorithm is to find a cause and effect relationship between alarms. For example, *"IP2 Port down"* causes *"DNS down"* in Figure 3. Furthermore, *"DNS down"* causes *"DNS unreachable"*. Thus, in this example, *"IP2 Port down"* has a cause and effect relationship with *"DNS down"*. Also, *"DNS down"* has a cause and effect relationship with *"DNS unreachable"*. Therefore, all alarms which have an identical cause and effect relationship in this example can be merged into a single cluster.

To achieve this, Figure 6 shows the pseudo-code of the alarm clustering algorithm which is the second step of alarm correlation. In line 1 to 7, the algorithm counts total number of groups and alarms of each group to initialize the *Cluster Key IDs* which are used to cluster the alarms according to identical *CK_ID* as shown in Figure 7.

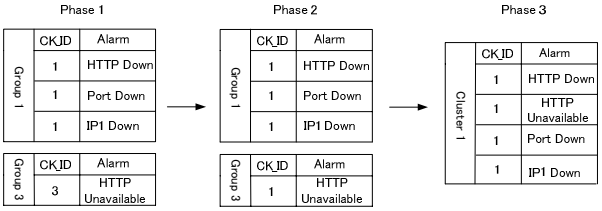


Figure 7. An example of alarm clustering

In Figure 7, at first, the *CK_ID* of *"HTTP Unavailable"* alarm is initialized to 3 in the phase 1 (Line 1 to 7 in Figure 6) and then it is set to 1 in the phase 2 since *"HTTP Down"* has a cause and effect relation with *"HTTP Unavailable"* (Line 8 to 17 in Figure 6). In the phase 3, all alarms are merged into a single cluster (Line 18 in Figure 6). In line 20 to 26, all alarms in each cluster are sorted in order by layer which range from application layer to the physical layer (Line 21). In addition, many similar alarms are removed in each cluster (lines 22 and 23). Figure 8 shows generalization process for the similar alarms.

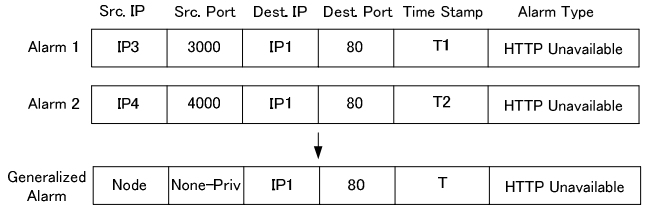


Figure 8. An example of generalization process for similar alarms

The differences between *Alarm 1* and *Alarm2* in Figure 8 are the source IP address, the source port number and the time stamp value. Therefore, those alarms can be generalized as a single alarm in order to reduce a huge number of alarms. Finally, line 27 will output a set *C* of clusters which include the root cause alarms.

The final step of alarm correlation is to identify the root cause alarms in each cluster. For instance, Figure 9 shows an example of output cluster after the alarm clustering algorithm processes the alarms in Table 1. It is very clear that "*IP1 Down*" and "*Port 53 Down*" are the root cause of Cluster 1 and 2 respectively. However, there is a possibility that multiple alarms can be appeared in the lowest layer of each cluster. In that case, since an alarm type can be divided into a critical alarm and a warning alarm generally, the critical alarm has a higher priority than the warning alarm. Therefore, any critical alarms can be the root cause alarms rather than the warning alarms.

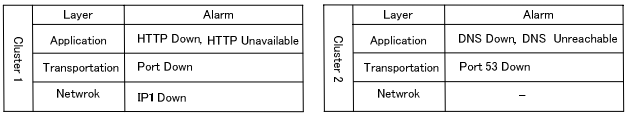


Figure 9. An example of output clusters in alarm correlation

By analysing a set of clusters, how a problem of network and why it happened can be realized. Moreover, it will be useful data to prevent the problems which will be happened in the near future.

4. EXPERIMENT AND EVALUATION

In this section, we describe the experiment and evaluation of our alarm correlation algorithm to confirm its efficiency of analyzing alarms and accuracy of identifying root cause alarm. The alarm correlation algorithm is coded in C++ language and performed on Linux machine. In addition, we used synthetic alarm datasets as shown in the Table 1 for the input data of the alarm correlation algorithm. Figure 10 shows the topology of twenty-two nodes with five critical alarms such as "*Link Down*", "*IP Down*" and so on. We assume that the critical alarms occur at five black nodes in Figure 10 at the same time and all node *N* are connected to all servers. For instance, *N2* is connected to all servers. Thus, *N2* generates several alarms such as "*WS1 HTTP Unavailable*" and "*FS2 FTP Unavailable*". Also, *R2* generates "*R3 Neighbor Loss*" alarm and "*R0 Neighbor Loss*" alarm simultaneously.

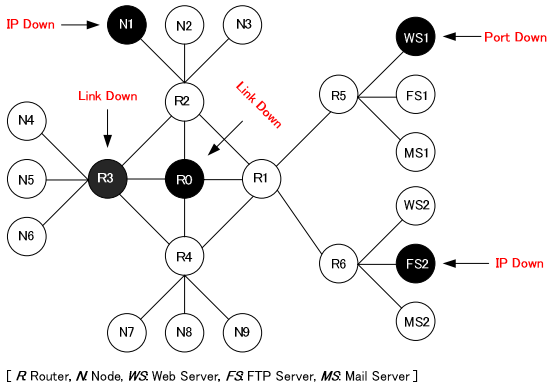
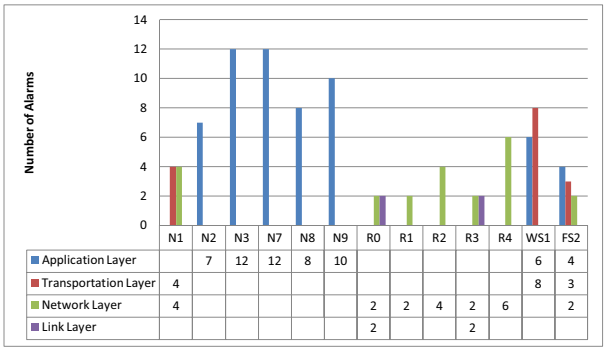


Figure 10. Topology of our experimental network

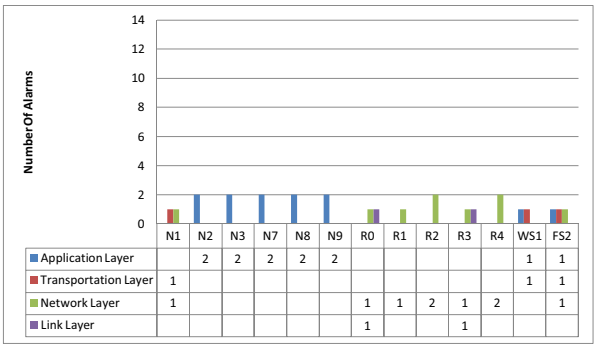
In this manner, we generate a hundred alarms randomly as shown in Figure 11.



	N1	N2	N3	N7	N8	N9	R0	R1	R2	R3	R4	WS1	FS2
Application Layer		7	12	12	8	10						6	4
Transportation Layer	4											8	3
Network Layer	4						2	2	4	2	6		2
Link Layer							2			2			

Figure 11. Synthetic alarm datasets of each node

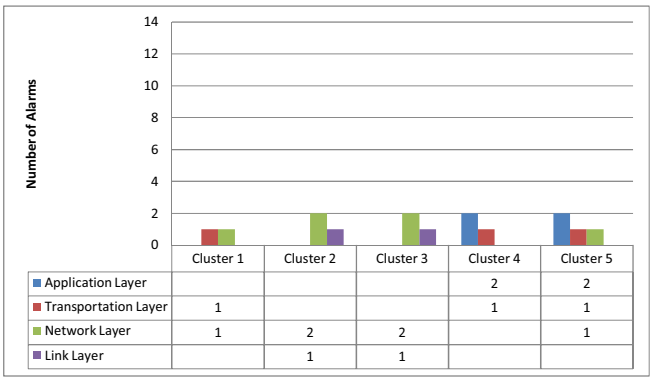
Since, *R3* link is down, there is no way to get the alarms from *N4* to *N6*. Thus, those alarms are not included in the alarm datasets.



	N1	N2	N3	N7	N8	N9	R0	R1	R2	R3	R4	WS1	FS2
Application Layer		2	2	2	2	2						1	1
Transportation Layer	1											1	1
Network Layer	1						1	1	2	1	2		1
Link Layer							1			1			

Figure 12. Output of grouping algorithm

Figure 12 shows the output of grouping algorithm. The number of original alarms is reduced from 100 alarms to 26 alarms since a number of identical alarms are generalized as a single alarm. For instance, since *N3* is connected to both *WS1* and *FS2*, it can generate the two types of alarm such as "*WS 1 HTTP Unavailable*" and "*FS2 FTP Unavailable*". In our experiment, *N3* generates six "*WS1 HTTP Unavailable*" alarms and six " *FS2 FTP Unavailable*" alarms and then, the grouping algorithm generalizes the two types of identical alarms as a single alarm respectively.



	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Application Layer				2	2
Transportation Layer	1			1	1
Network Layer	1	2	2		1
Link Layer		1	1		

Figure 13. Output of clustering algorithm

Figure 13 shows the output of clustering algorithm. The number of alarms after the grouping algorithm processes the alarms is reduced from 26 alarms to 15 alarms since a number of similar alarms are generalized as a single alarm. For instance, since *R3* generates "*Link Down*" alarm, it is the initial cause of "*R3 IP Down*" alarm which is detected at *R3* and "*R3 Neighbor Loss*" alarms which are generated by *R2* and *R4* respectively. Thus, those alarms have an identical cause and effect relationship so that they are merged into a single cluster. After that, "*R3 Neighbor Loss*" alarms are generalized as a single alarm by the clustering algorithm since they are similar alarms. Finally, our algorithm can identify a root cause alarm in each cluster as shown in Figure 14.

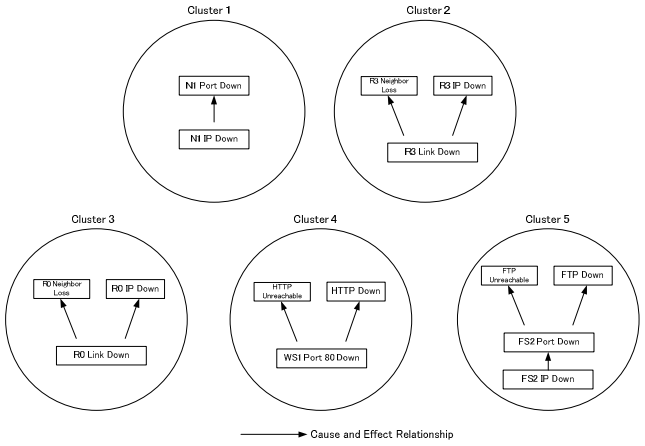


Figure 14. A root cause alarm of each clusters

The lowest alarms in each cluster such as "*N1 IP Down"* alarm, "*R3 Link Down"* alarm and so on in Figure 14 are identified as the root cause alarm. As a result, the number of original alarms is reduced from 100 alarms to 5 alarms.

ACKNOWLEDGMENT

This research was supported by the Ministry of Internal Affairs and Communications in Japan: "High Reliability Cloud Networking".

5. CONCLUSIONS

This paper proposed an alarm correlation algorithm leveraging TCP/IP model and focusing on root cause analysis without relying on expert knowledge to detect the root cause alarm. According to the cause and effect relationship, the alarms are merged into a cluster and then the root cause alarms are figured out exactly. In addition, a number of identical and similar alarms are reduced dramatically by alarm generalization. The results of experiment prove that this algorithm can analyse alarms efficiently and indentify the root cause alarm accurately.

REFERENCES

[1] AL-MAMORY Safaa O., HONGLI ZHANG, "Intrusion Detection Alarms Reduction Using Root Cause Analysis and Clustering," Journal of Computer Communications, February 2009, vol.32, no2, pp. 419-430.

[2] Banerjee D., Madduri V.R., Srivatsa M., "A Framework for Distributed Monitoring and Root Cause Analysis for Large IP Networks," 28th IEEE International Symposium on Reliable Distributed Systems, September 2009, pp.246-255.

[3] Hanemann, A.and Marcu, P., "Algorithm design and application of service-oriented event correlation," 3rd IEEE/IFIP, April 2008, pp. 61-70.

[4] M. Steinder and A. S. Sethi, "A Survey of Fault Localization Techniques in Computer Networks," Science of Computer Programming, Special Edition on Topics in System Administration, Vol. 53, November 2004, pp. 165-194

[5] White Paper, "Automating Root-Cause Analysis: EMC Ionix Codebook Correlation Technology vs. Rules-based Analysis," Nomber, 2009.

[6] Wu Jian and Li Xing ming, "A Novel Algorithm for Dynamic Mining of Association Rules," International Workshop on Knowledge Discovery and Data Mining, jan 2008 PP.94-99.

[7] Qiuhua Zheng and Yuntao Qian1, "An Event Correlation Approach Based on the Combination of IHU and Codebook," Lecture Notes in Computer Science, 2005, Vol.3802, pp.757-763

[8] L. Lewis, Managing Computer Networks - A Case-Based Reasoning Approach. Artech House, Inc., 1995.

[9] Jukic O. and Kunstic M., "Logical Inventory Database Integration into Network Problems Frequency Detection Process," ConTEL 2009, June 2009, pp.361-365.

[10] Risto Vaarandi, "A Data Clustering Algorithm for Mining Patterns from Event Logs," IP Operations and Management(IPOM 2003), Octerber 2003, pp.119-126.

[11] Risto, Vaarandi, "Tools and Techniques for Event Log Analysis," PhD thesis, Tallinn University of Technology, Department of Computer Engineering , Estonia, June 2005.

[12] Forouzan, Behrouz A, TCP/IP Protocol Suite (2th ed.). McGraw-Hill, 2009.

[13] A. Devitt, J. Duffin, and R. Moloney, "Topographical proximity for mining network alarm data," in ACM SIGCOMM workshop on Mining network data, Philadelphia, Pennsylvania, USA, Aug 22-26 2005.