In this paper, we propose an algorithm based on codebook
system to determine whether the connection of an alarmed
circuit actually breaks. The algorithm is capable of apply to
c人寿 large scale networks. The goal is to help network
operators and telecomm staffs identify actual broken links or
unavailable services from thousands of alarms emitted from
networks which should be solved with priority.

The paper is organized as follows: In section 2, we have a
brief discussion on communications network structure
nowadays. The introduction to event correlation and Codebook
system is in Section 3. Section 4 explains the algorithm in
detail, followed by Section 5’s Conclusion and future work.

II. COMMUNICATIONS NETWORK STRUCTURE

Modern telecommunication networks are designed and
built with large capacity to meet the never-ending demand for
bandwidth; with high versatility and flexibility to carry traffic
of various services; and reliably enough to achieve customer
satisfaction and quality requirements.

Network construction is an innovation process. Network
operators and telecomm companies should not only invest in
new equipment and adopt latest technology, but also make
good use of the existing infrastructure. As a result,
telecommunication broadband networks, like the one in Figure
1, are heterogeneous and in hybrid architecture; old and new
equipment co-exist, and various network topologies are mixed
and linked together.

The status of a network is monitored and tracked through
NMS (Network Management System) that tracks alarms sent
by managed objects when abnormal event occurs. In a complex
network, operators build a network resource repository
database to manage the settings of every service link and user
circuit. With the configuration data, when received a set of
alarms, one should easily find out which circuits are affected.
However, with the existence of protection schemes in the
network, determining whether the affected circuits’ connection
is actually broken is ironically difficult because backup routes
may take place to keep the link alive.

In telecomm scale networks composed of tens of thousands
various network equipment, alarms are sent all the time.
Operators should dispatch their limited service workforce to
severe link down circuits rather than those with alarms but still
connected. Fast alarm analysis and filtering is strongly needed
in daily network operation.
III. EVENT CORRELATION SYSTEMS

Telecom staffs’ daily chore of determining source of failures from hundreds of alarms can be logically regarded as correlate events with problems. Gupta and Subramanian [3] classified correlation schemes into the following categories: Rule-based, Model-based, Case-based, Generic reasoning, Probability networks and codebook approach. [3] claim the last technique is superior to others. The concept of the approach is to find out a matrix to serve the event-problem correlator. In the following subsection we survey some of the means to generate that matrix.

A. Network coding

Network coding, first proposed by Yeung and Cai [4], is a process that each node in a network performs linear operations over a finite field on received packets to generate forwarding packets. One of its particular applications is network tomography which aims at characterizing the internal properties of large networks through boundary nodes, like the work by Sharma et al. [5]. Firooz et al. [6] further utilizes this technique for network status monitoring. It demonstrated a coding and training process to generate a failure-link-mapping table capable of locating a broken or congested link in networks of any logical topology.

Network coding scheme require nodes within a network (for example: routers) to implement the encoding/decoding process; the codes used can be embedded and transmitted by existing protocols.

B. Event correlation codebook

While network coding schemes treat the target network as a ‘black-box’, operators exploit full information about every node and link within. With such information as a prerequisite, Yemini et al. [7] suggested an idea that the relationships between events (alarms, for instance) and the specific problem be described as a ‘code’ table. To find the optimal subset of the code table that is just enough to distinguish the problem of interest from another, [7] observed the causalities between events via ‘causality graph’ and remove redundant or unrelated ones. The shrunk causality graph, called ‘correlation graph’, gives the optimal subset named ‘codebook’ for event-problem correlation.

The codebook approach does not treat events separately. It groups the alarms into a vector. Then this vector is the matched to problem signatures in a codebook

IV. CODEBOOK ALGORITHM

The aim of our proposed algorithm is to figure out whether an alarmed circuit is still able to carry traffic. With that in mind, instead of caring for the whole network, we focus on user circuit matters. In other words, the problem domain of correlation scheme is narrowed down to circuit connectivity determination. The work by Jung et al. [8] which also targeted the similar goal is the basis of our method. In their previous work, all used ports are put into consideration. In circuits that utilize a lot of ports may result in a large modeling graph. Our work reduces the number of ports monitored in compare to [8] by applying notion from [7], that is, observe the causalities among nodes. The following subsections give detailed explanations and an illustrative example.

A. Event Causalities in a Route

Any end-to-end circuit can be treated as a combination of one or more links, or routes. Figure 2. draws this idea. A circuit between two ends in tree, mesh, star or bus topology networks is regarded as a single line route; whereas in ring topology networks has 2 distinct links between ends.

Figure 3. presents a typical user circuit. It is composed of five equipment, labeled A, B, C, D and E. The downlink/uplink ports are denoted as A1 and A2, respectively. A ring is built by nodes A, B, D, and C. Between node D and E set up a link aggregation (LAG). A circuit is unable to carry any traffic if and only if all distinct routes are disconnected. In this circuit, we have four distinct routes from A to E node. Each route passes through different set of ports, and it is in line topology. When one or more ports in a link are down, this link is not connected. The relation between ports’ status and the connectivity of the corresponding route can be expressed by a causal diagram. Figure 4. shows the causal diagram of the circuit in Figure 3. The directed edges means the link status of the source will be affected by that of the destination node.
Events in networks will propagate through node due to its underlying protocols or topology. When a port sends an alarm, its directly connected ports will also trigger a trap of the same type. By this causality, one of the events sent by 2 adjacent ports can be ignored without loss of information. Therefore, we can further reduce the nodes in the causality graph. Figure 5 is the correlation graph derived from Figure 4. Note that we have eliminated every one of the adjacent ports: B1, D1, E1, D2, E2 and C1 from the original causality graph. We define the route codebook $C$ from the 2-dimention matrix representation of the correlation graph whose rows denote every distinct path and columns indicate correlated ports. With the route codebook, we can now map the events and determine the connectivity of respective paths in a circuit.

**B. The correlation Algorithm**

Alarms generated by network equipment are various in kind. We classify alarms into two categories: major alarms that indicate plausible connection loss (ex: Loss Of Signal alarm), and minor alarms that indicate degradation of the connection (ex; Error Seconds). Since the purpose of the algorithm is to determine whether service is still available, we then specify the value according to the event’s severity. First, let the route codebook be matrix $C_{n \times m}$, we define the Simplified Link survivability Matrix (SLIM) $S_{n \times m}$ as follows:

$$s_{i,j} = \begin{cases} 1 & \text{port } j \text{ on route } i \text{ emits a major alarm} \\ 0.5 & \text{port } j \text{ on route } i \text{ emits a minor alarm} \\ 0 & \text{otherwise} \end{cases}$$

Next, we derive the Route Connectivity Vector (RCV) $P_{n \times 1}$ from $S$, where entry $p_i$ is the maximum element of $ith$ row in $S$:

$$p_i = \max\{s_{i,1}, s_{i,2}, \ldots, s_{i,m}\} \quad (2)$$

$\mathbf{if } p_i < 1 \mathbf{infers that the corresponding route } i \mathbf{is connected.}$

Finally, we multiply all the entries in $P$ together:

$$R = \prod_{k=1}^{n} P_k \quad (3)$$

If $R \neq 0$, we resolve that the service is no longer available due to all protection mechanisms, rings or link aggregations, in this network are overwhelmed and as a result, the connection is lost. On the contrast, if $R = 0$, infers that at least one of the many routes of this circuit is connected. Thus the circuit is capable of delivering services.

**Example:** Again we take circuit in Fig 4-1 as an example. The route codebook $S_{4 \times 8}$ of this circuit is shown in Figure 6.

At some time port D3 and B2 send ‘loss of signal’ major alarm, we use this codebook algorithm to generate the SLIM $S$ and compute the RCV $P$ vector respectively by (1) and (2):

$$S = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Multiply all the entries in $P$ together to obtain $R$:

$$R = 1 \times 1 \times 1 \times 0 = 0. \quad (5)$$

<table>
<thead>
<tr>
<th>Route</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B2</th>
<th>C2</th>
<th>D3</th>
<th>D4</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RouteR1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>RouteR2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RouteR3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>RouteR4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
We conclude that in this example, the connection survives the link loss between D3 and B2 and is still capable of delivering services.

V. CONCLUSION

In this paper, we propose an algorithm to explore the problem of service availability on user circuit. The algorithm observes the causalities among ports to reduce the size of the codebook in event correlation schemes, and the complexity is quite low. In telecomm networks, configuration and provision data is at operator’s hands. As a result, the codebook for every user circuit can be calculated and stored in advance. When events or alarms arise, not only affected circuits can be found, but also whether the circuit is still alive can be determined quickly. In addition, the built codebook can further be used as a filter for alarm receivers to ignore traps from certain ports. The suggested method works well on circuit with more than one route from source to destination, like those with Rings or provisioned with LAG. It compensates the protective mechanism’s negative effect on judging circuit connectivity.

In practice, maintenance and operation staff can quickly discover whether it will really affect customer service, and determine the need to send personnel to deal with this as a priority obstacle. The system makes the communications network more robust and their operation more reliable.

Broadband network nowadays provide various kinds of services. Some services require more network quality than others. If the quality of the still-connected circuit fails to comply, the service is regarded as unavailable. For example, video on-demand service that send streaming video data on the service line is one of the services demanding network quality: customers will not tolerate jitters or lags. When they happen, operators should also deal it ASAP. With that in mind, an improved codebook that not only considers add-on services of the user circuit but also take quality-related alarms into consideration is our future work so that this method meets practical operation requirements better.

REFERENCES