AN EXPERT NETWORK ANALYZER

Nikitas J. Dimopoulos, Kin F. Li, Andrew Watkins,
Stephen Neville and Athanasios Rondogiannis

Department of Electrical and Computer Engineering
University of Victoria
Victoria, B.C.

ABSTRACT†

In this work, we shall present the framework of an expert system that will be capable of diagnosing and predicting faults in a cable television plant.

The structure of the diagnostic expert system under development comprises equipment specific and network specific parts. We shall present the integration of status monitoring data into our system and their use in determining failure points in the network. Once a failure point has been reliably determined, then the equipment part of the system is used to diagnose the failure. The cause of failure is determined at either the network level, the equipment level or both.

Finally, we shall present our efforts towards advanced prediction of faults based on the interpretation of the status monitoring data.

1. INTRODUCTION

Diagnostic expert systems have been developed for several areas of expertise. These include medical as well as engineering diagnosis.

Specifically, in the field of telecommunications and communication networks, there have been several examples of expert diagnostic systems which help interpret the status of the system and eventually produce a diagnosis.

Thus, Sugarawa [4] presented a distributed expert system capable of diagnosing local area network problems and has TCP/IP related diagnostic and troubleshooting knowledge. Miyazaki [2] describes techniques for updating switching networks operations and maintenance through expert systems. Nuckolls [3] describes an expert system that performs real time diagnosis of a large digital radio network. Yudkin [5] has introduced a methodology for building up a diagnostic system based on the structure behavior and functionality of the system and its components.

In this work, we shall present our approach of automating the diagnosis of a large Cable Television Network.

Thus in section 2 we shall introduce the structure of such a network and the diagnosis problem while in sections 3 and 4, we shall discuss our attempt to automate the diagnosis. Section 5 concludes this work and discusses future developments.

† This work is supported by the Canadian Cable Labs Fund

2. STRUCTURE OF THE NETWORK

A Cable Television Network incorporates a number of high frequency amplifiers forming (for conventional networks) a tree. In more advanced networks the structure incorporates a double ring of which subscriber loops emanate. In this work, we are focusing in conventionally structured tree networks. There are two categories of amplifiers the ones belonging to the main trunk and the ones forming subscriber loops. additionally, power supplies are located in throughout the network, each one powering a limited number (typically three)of amplifiers. The majority of the main trunk amplifiers are equiped with a status monitor which uses a reverse channel to report the status of the amplifier to the head office. Subscriber loops and power supplies are not normally monitored.

- Typical variables which are monitored include [1]
- Output pilot level
- Output data carrier level
- Raw DC voltage into the amplifier
- B+ voltage of amplifier power supply
- DC current into forward and reverse sections of the amplifier
- Temperature inside the trunk station
- Reverse Switch status
- Trunk lid status

The values of the monitored variables are allowed to vary within two intervals (warning and alarm) centered at typical values. If a value is outside these predefined intervals then a warning or an alarm is issued. Three consecutive alarms constitute a failure. Additionally, failure to raise a particular status monitor for three consecutive time intervals, also constitutes a failure.

Each status monitor apparatus has its own electronic address, which is used by the head office to poll the status monitor for a report.

Because of the number of amplifiers in the network, each amplifier is polled at fixed intervals (typically every few minutes).

The polling order may change if a particular section of the network needs closer attention.

A typical section of the main trunk is depicted in Fig. 1. Each amplifier in the network has a name, as well as a location, connectivity and functionality attributes.

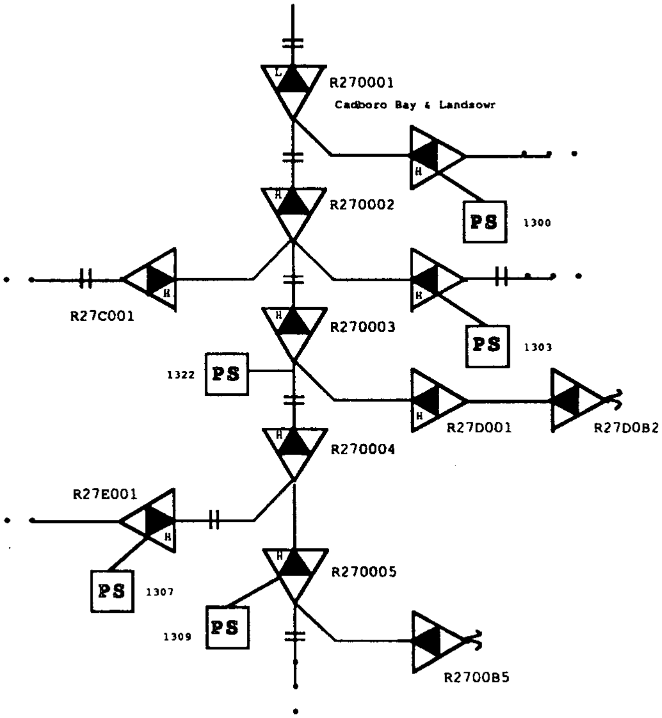

Figure 1. A typical Section of a Main Trunk (curtesy Roger's Victoria)

3. DIAGNOSIS PROBLEM

There are several modalities of failure. Some are discussed below.

A single amplifier may fail, whereupon the signal is fed through unamplifeid to the subsequent stage of the network. The subsequent amplifiers, equiped with automatic gain control, will boost the signal back to its normal level after two to three stages. Because of the failure, all amplifiers located between the failing amplifier and the stage at which the signal was boosted to its correct level, alarms or failures. Because of the tree structure of the network, the reporting amplifiers are not typically polled sequentially and the reports appear at seemingly random locations in the report.

A power grid failure, will affect a number of amplifiers. The affected amplifiers fail to communicate, while the deteriorated signal, cause downstream nodes to report failure until the signal is boosted again to its typical values. Again, because of the structure of the network, these reports appear at random locations and are seemingly unrelated.

Another category of failures are due to the unmonitored subscriber loops. These are typically reported by the subscribers themselves as deterioration of service which range from increased noise levels for some channels to complete interruption of reception. These reports are due to failures in the main trunk, subscriber loops or even interruption of service due to the customer being at arrears in his/her payments.

In this work, we shall concentrate in the diagnosis problem of the monitored main trunk.

4. DIAGNOSIS APPROACH.

We follow a model based approach in our reasoning[4]. Because the structure of the network is known a-priori, and since the behavior of the amplifiers in the network is well understood, and because there is a large umber of elements in the network, model-based reasoning is best suited for our domain. In model based reasoning, as opposed to classification reasoning, bases its decisions on an underlying model of the system being diagnosed.

In our case, the structure, and operation of the elements in the network, provide us with a general and powerful model on which we base our diagnosis.

As we discussed previously, the failure is localized to at most a small number of amplifiers, but because of their interconnection, deteriorated levels of the signal propagate and affect a large number of neighboring nodes which themselves report failure. One of the main objectives of our diagnostic system is the isolation of the *trully-failed* nodes from the ones which report failure because of their proximity to the failed nodes.

As it can be seen in Fig. 2., our system is divided into two parts.

The first part forms *clusters* of failure. A *failure cluster* is defined to be a connected set of nodes, at least one of which exhibits a failure while the remaining register a warning, alarm or failure within an observation window which spans approximately one hour.

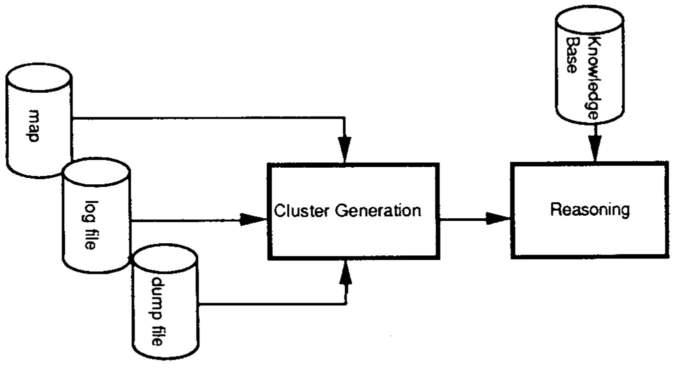

Figure 2. The structure of the Expert Network Analyzer

Failure clusters are important because they incorporate all the affected amplifiers within an observation window, and the reasoning engine can focus only within the cluster. Additionally, forming the failure clusters dynamically, we isolate the reasoning process from sources of information which cannot be made temporally constant. For example, the amplifier data-base together with the connectivity information changes in time as the cable-plant evolves.

Failure clusters, in addition to the amplifier names, incorporate connectivity information as well as observation information within the window of observation. A failure cluster is seeded with a single failing amplifier at a specific time and then the cluster forming algorithm includes in the cluster all connected amplifiers that exhibit warning, alarm or failure within an observation window centered around the failure time of the seed amplifier.

uploaded to our facilities. The alpha version is designed to analyze these data.

To this point, we have completed the clustering and the importation of these clusters into the NEXPERT OBJECT environment where the reasoning engine is currently under development. We have introduced some simple rules which recognize failures related to power grid problems as well as failures due to a single amplifier.

```
Cluster Forming Algorithm

Define the seed amplifier
Define the observation_window  as [seed_time -offset, seed_time +offset]
Put seed into suspect_queue
While the suspect_queue is not empty
A:   Remove first element from suspect_queue.; name it current_amplifier
     If current_amplifier is in the failure_set then discard it and goto A
     If current_amplifier does not register a fail or alarm or warning
                    within the observation_window then discard and goto A
         else
         find all amplifiers connected to the current_amplifier  and
                                   place them in the suspect_queue
         place current_amplifier  in failure_set
End when suspect_queue  is empty.
```

A failure cluster is translated into a set of NEXPERT objects which fully quantify the topology and behavior of the cluster within the window of observation. Each amplifier in the cluster comprises three objects, namely the topology_data object, the amp_data object and the log_data object. The topology data object incorporates static properties of the amplifier such as the the location, connectivity, power grid, status monitor number, termination etc. The log data object incorporates the fault data of the amplifier such as the fault message, the fault type (warning, alarm, failure) the actual and nominal values of the variable that has caused the failure and the fault times. Finally the amplifier data object incorporates the observed data for all the monitored variables of the said amplifier within the window of observation.

Typical examples of objects are given in Figs 3., 4., and 5.

```
{topol_R20000D
   Classes:     topology_data
   Properties:  location        (Haultain@foulbay)
                parent          (R20000C)
                pilot           ()
                power_grid      (1209)
                SMT_number      ()
                subs            ()
                termination     (False)
                type            ()
}
```

Figure 4. Example of a topology_data object

```
{data_R20000D
   Classes:     amp_data
   Properties:  B         (23.8,23.8,23.8)                /* B+ voltage */
                C         (1143.3,1143.3,1143.3)          /* current */
                P         (-100.0,-100.0,-100.0)          /* forward pilot */
                R         (-6.0,-6.0,-6.0)                /* reverse pilot*/
                RD        (54.9,54.9,54.9)                /* Raw DC voltage */
                T         (-10.0,-9.0,-10.0)              /* Temperature */
                times     (12140916,12140919,12140922)    /* Polling times */
}
```

Figure 3. Example of an amp_data object

5. STATUS AND CONCLUSIONS

We are currently implementing the alpha version of our expert network analyzer on a SPARC platform in C and NEXPERT OBJECT. The alpha version is designed to operate off-line in a batch mode. The status monitor data of the entire ROGER'S Victoria system are collected daily and

```
{log_R20000D
   Classes:     log_data
   Properties:  fault_message   (Communication)
                fault_nominal   ()
                fault_times     (12140931)
                fault_type      (Alarm)
                fault_value     (NO REPLY)
}
```

Figure 5. Example of a log_data object describing an amplifier in an alarm condition because of failure in replying to a polling request.

We are cooperating with domain experts to expand and refine our rule base.

Once the reasoning engine is refined, we plan to release the beta version for on line field testing.

The work described here is part of a larger undertaking in collaboration with the Canadian Cable Labs Fund to investigate the use of AI and Knowledge Engineering techniques in the automation of the diagnosis procedures in a large cable TV plant. The development has proceeded in two phases.

During the first phase, we have developed FLOWTOOL™ a graphical knowledge acquisition tool which is capable of capturing knowledge expressed in the form of flowcharts and translating it into rules for NEXPERT OBJECT™. In addition, the resulting knowledge base is linked to hypermedia documentation incorporating information that cannot be expressed in the form of rules within the knowledge base. Appropriate sections of this documentation is being made available automatically during the operation of the inferencing and this can be used to assist and/or train the user of the knowledge base in providing appropriate responses and comprehending the inferencing involved.

The second phase includes the development of the Expert Network Analyzer, which we discussed in this work, as well as the development of failure predictive models for the behavior of the network so that preventive maintenance procedures could be instituted. For this part of the work, we are collecting daily the status monitor data of the entire Roger's Victoria network, which we shall use in developing our predictive models.

REFERENCES

[1] *Instructin Manual and User Guide Status Monitor System*, C-COR, 5/84, rev 0, May, 1984.

[2] Miyazaki, T., Fujimoto, H., Kim, M.W., and Wakamo, M., "Improving Operation and Maintainance for switching network," in *Proceedings of GLOBECOM '89*, IEEE, Nov. 1989, pp. 1149-1153.

[3] Nuckolls, V., "Telecommunications diagnostic expert system," in *Proceedings of GLOBECOM '89*, IEEE, Nov. 1989, pp. 507-511.

[4] Sugawara, T., "A Cooperative LAN Diagnostic and Observation Expert System," in *Proceedings Ninth Annual International Phoenix Conference on Computers and Communications*, IEEE, March 1990, pp. 667-674.

[5] Yudkin, R.O., "On Testing Communications Networks," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 5, 805-812, 1988.