Database Analysis of CATV Network Interconnectivity

by
Robert V. Moel
Time Warner Cable, Houston Division
William Spies
Time Warner Cable, Cincinnati Division

Abstract

Current outage detection systems ignore the interconnectivity between transmission elements. A failed element is analyzed in isolation without considering its impact on devices downstream from it. A simple database algorithm that describes this interconnectivity would improve outage detection speed and accuracy. This paper will describe such a program written in a common fourth generation language (4GL) called Clipper dBase.

Introduction

Cable television systems can be modeled as "weakest" link systems. Any device failure causes subsequent failure of all devices downstream from it. The probability of a point in the system failing is related to the probability of all previous devices failing[1]. Being able to know which other devices have failed would aid in customer communication about outages and would minimize falsely detecting another outage that is part of the first (and thereby not wasting manpower on chasing phantom outages).

Elements of an Interconnectivity Database Program

The questions that need to be answered are: (1) What relationship exists between devices in the network that uniquely define their relationship; in effect how do you encode which device is dependent on which. (2) The database and program must be able to model the weakest link attribute of the network. (3) The program that manipulates the network database must be fast and it must be real-time. There is no use for a program that tells you an outage occurred when the day is done. Finally, the program must be recursive. It must be able to function the same at any point in the system and recursively determine all other failed devices.

One of the key items in creating the database is determining what fields create links between database records that are unique. Consider figure 1. The three amplifiers are named 1, 2 and 3 (any convenient naming scheme will work). Amplifier 1 feeds amplifiers 2 and 3 and has them as successors. Similarly,

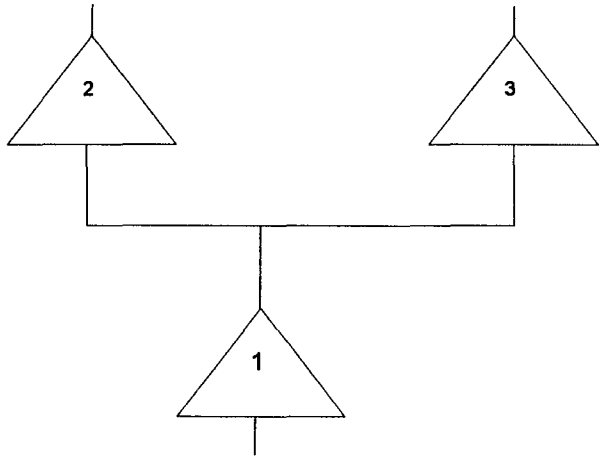
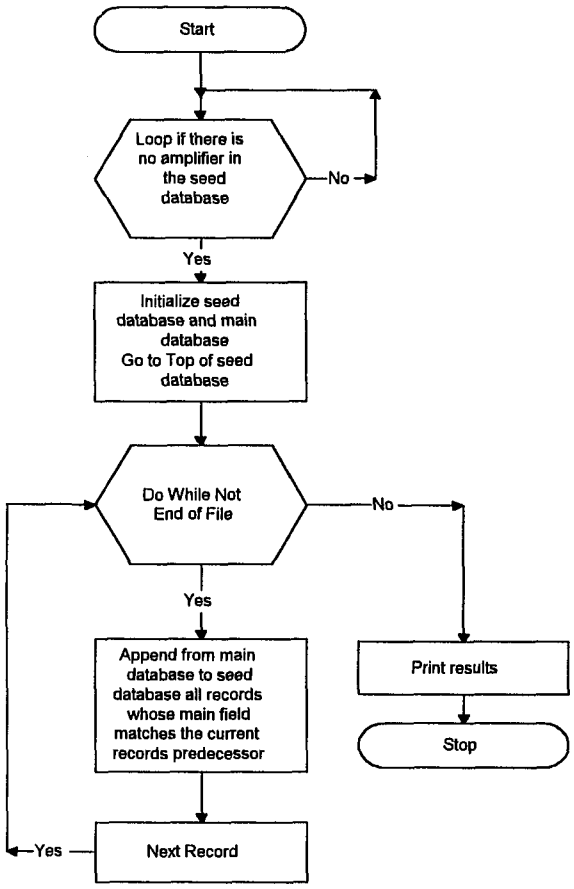


Figure 1

Amplifier 1 is fed by no other amplifier. Amplifier 1 has a null predecessor. Amplifiers 2 and 3 are fed by Amplifier 1 and hence Amplifier 2 has Amplifier 1 as its immediate predecessor and Amplifier 3 has Amplifier 1 as its immediate predecessor. Amplifier 1 can have either 2 or 3 as its immediate successor. Using the successor is, therefore, not a unique way to relate amplifiers in a database network. However, Amplifier 2 has only one predecessor, that of Amplifier 1. Similarly, Amplifier 2 has only one predecessor, again Amplifier 1. Devices in cascade as described above can be determined uniquely if the predecessor is specified as the linking field in the database record.

By stringing database records together by using the predecessor as the linking field, a database would be created that models the weakest link behavior of a real cable television system.

Figure 2



The code that was written to determine the effect that the weak link has on the rest of the cable system operates on a PC. The program is compiled to execute in real-time. As the flow chart will demonstrate, amplifiers are entered in real time and the list of amplifiers that have failed are produced instantly. Table 1 contains the program listing. Figure 2 is the flow-chart of this program.

The program works by starting with a seed amplifier. A pointer is set initially to this first amplifier. Then the program appends all the amplifiers whose predecessor fields match the amplifier currently pointed to. After all the amplifiers are copied to the end of the database, the pointer moves to the next record and the process starts again. When the pointer moves past the last record, the program stops looking for more amplifiers, prints the results and terminates.

Amplifiers are not the only devices that can be added to the database. Any device can be a part of the system as long as one knows how the devices interconnect with each other from a reliability sense. For example, while a main power supply is installed approximately in the middle of a group of amplifiers, from a reliability view point it is at the beginning of a cascade.

Because the program uses matching and copying, it is very fast. A database with 1,000 amplifier records and an average trunk run of twenty amplifiers will take approximately two seconds to respond to an inquiry when run on a "486" machine. These characteristics would make it feasible for this program to be integrated with a billing host's outage detection system either as a part of the host's program or off line a as separate processor with which the host would interconnect via a wire link. Any single amplifier outage would be used to

determine what other amplifiers were also out. The additional amplifiers determined to be out would be up loaded to the host for dissemination to customer service agents.

Outage Detection In Cincinnati

Our system in Cincinnati has been using a combination of two-way monitoring of the plant and the database program to improve outage detection and correction, determine the underlying outage problem, develop a plan to improve the performance, and finally, measure the results.

The first step was to use the two-way response information from converters in the field to identify the location of outages. Part of the difficulty in using these converters is that there is always a group of converters that do not answer back. These "no-answer" converters may not be experiencing an outage but may have some other localized problem such as being on a switched outlet. This no-answer rate determines the "noise" threshold. Converters must be non-responding in excess of this rate to indicate an outage.

Once the exact location of the outage is determined, the next step is to feed the information into the database and get a list of all other amplifiers that were affected. We now have a true location of all the subsequent amplifiers affected and a better understanding of how a particular outage disrupts the system.

These disruptions were further categorized by outage fix code. In some instances the fix codes were changed to provide better information of cause. For example, the repair code labeled "intermittent" was not really a repair code because it did not tell us anything about what was done.

Once an analysis of the fix codes and outages was completed, two areas emerged as being key to resolving the problem. The first deals with the placement of power supplies. The analysis changed the way we placed supplies. Before we used to place supplies in anticipation of future growth. These under loaded supplies added to the number of supplies in cascade and increased the probability that an area would be out. Sometimes these areas were affluent and influential. Currently we minimize supplies and maximize loading. If an area grows, we will return to it and redesign the powering if needed.

The second deals with repeat outage offenders. What we discovered by using the database was that we were experiencing outages from devices that had failed and had recently been repaired by our in-house facility. The facility, it turned out, was ill equiped to perform the repair. They were merely bandaging the problem instead of thoroughly testing and repairing the unit. The few dollars we were saving were costing us dearly.

We arranged to have the original manufacturer repair the equipment by establishing an "extended warranty" agreement with them. This approach gave us a cost effective alternative to in-house repair that guaranteed the problem would not return to soon. Ultimately, this change gave the technicians more confidence in the refurbished equipment and, instead of "box" swapping, they are more likely to look for fundamental problems with the plant.

Conclusion

A database and program can be constructed to simulate a cable television system. Using this system one can detect outages faster, find and repair fundamental plant reliability problems, and better assess powering needs and power supply location.

Table 1 -- Program Listing

```
* Program -- OUTDET.PRG
* Writer -- Robert V. Moel
* All rights reserved
* This resets all files and initializes several criteria
close all
set safety off
set talk off
set confirm on
set bell off
* Clears screen
@ 0,0 clear
@ 23,0 say "OUTDET written by Robert V. Moel, all rights
reserved"
@ 24,0 say "Press Enter to continue"
read
@ 0,0 clear
* Opens amplifier database as the source database
use amp alias source
sele 2
* Opens seed database
use out alias sink
* Sets up do while loop
repeat=.t.
do while repeat
 sele sink
* Clears seed database
 zap
 mampnum=space(8)
* enter the amplifier that failed
 @ 24,0 clear
 @ 24,0 say "Enter first failing amp" get mampnum
 read
 append blank
 mampnum=ltrim(rtrim(mampnum))
 replace amplifier with mampnum
 go top
* this is the iterative do loop that appends from the main database
to
* the seed database each amplifier that points to another amplifier
 do while .not. eof()
* Set the pointer
  point=recno()
* Look for all amplifiers for which the current amplifier pointed to
* is a predecessor
  temp_amp=amplifier
  append from amp for pred_amp=temp_amp
* because appending moves the pointer, reset the pointer and
* then skip to the next record
  goto point
  skip
 enddo
 answ="N"
 @ 24,0 clear
 @ 24,0 say "Would you like to print affected amplifiers?" get
answ
 read
 if upper(answ)="N"
  return
 else
* print or display amplifiers that were affected by the outage
  if upper(answ)="Y"
   answ1="S"
   @ 24,0 clear
   @ 24,0 say "(S)creen or (P)rint " get answ1
   read
   if upper(answ1)="S"
    display all
    @ 24,0 clear
    @ 24,0 say "Press Enter to Continue"
    read
   else
    if upper(answ1)="P"
     list all to print
    else
     quit
    endif
   endif
  endif
 endif
* do another outage?
 answ2="Y"
 @ 24,0 clear
 @ 24,0 say "Another Outage (Y/N)" get answ2
 read
 if upper(answ2)="N"
* if no other outages, leave the loop
  exit
 else
  @ 0,0 clear
  loop
 endif
enddo
* reset all and quit
close all
return
```

1. Moel, Robert and Spies, William Reliability Modeling of Cable TV Systems, 11-4 to II-5, CableLabs, September 1, 1992.