Local
Fail-safe Network Reset Procedure *

(Extended Abstract)

Yehuda Afek
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

Baruch Awerbuch
Department of Mathematics and
Laboratory for Computer Science
MIT
Cambridge, MA 02139

Eli Gafni
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90024

Abstract

A simple, general and optimal procedure to adapt algorithms designed for fixed
topology networks to run on a network with dynamically changing topology is
presented. The communication and time complexities of the procedure, per topological
change, are independent of the number of topological changes and are linearly bounded
by the size of the subnetwork over which the algorithm is running.

1. Introduction

Many distributed algorithms for data communication networks have been
presented in recent years [AG-85, GHS-83, A-85]. Most of these algorithms are fault-
tolerant, that is, they operate correctly under the assumption that the network topology is fixed during the execution of the algorithm. Should the network undergo a

* This paper is an abstract of the paper “Applying Static Network Protocols to Dynamic Networks” Proc. 28th IEEE Symp. on Foundations of Computer Science, October 1987.
topological change in the midst of execution, the results are meaningless at best. Yet, real networks, such as the ARPANET, do undergo topological changes due to link failures and recoveries. This note presents a general and efficient technique by which fault-intolerant distributed algorithms can be adapted to run on dynamically changing networks.

It is necessary to make some assumptions about the frequency in which topological changes occur. If all links are always down, or if topological changes are so rapid that no messages are ever transmitted, then no meaningful distributed algorithms can be run on the network. Here we confine the discussion to topological changes which are due to link failures and recoveries and satisfy the following two conditions: (1) eventual detectability: the status of the link is eventually detectable by the incident nodes. (2) relative stability: eventually, topological changes stop for a sufficiently long period of time (the length of which depends on the application). The precise meaning of these conditions will be clear after we present our model of dynamic network in Section 2.

We adapt fault-intolerant algorithms to run on a dynamic network, in the following sense: If topological changes cease, yielding topology $G$, the algorithm terminates, producing a result (states of nodes when quiescence is reached) which is consistent with topology $G$. Any algorithm which operates correctly in this sense is called, fail-safe.

The starting point in the development of our technique is Finn’s approach [F-79, G-76] to the implementation of this adaptation. Finn proposed to abort the execution of the fault-intolerant algorithm whenever a topological change is encountered. A new execution is then started in the hope that the network will be stable for sufficiently long period of time to allow the new execution to run to completion without encountering new topological changes. To implement this idea one has to devise a mechanism by which messages of "old" executions will be recognized as such and subsequently purged so that they will not interfere with the most recent "new" execution. Such a mechanism is called a reset procedure.

In this note we present a new efficient reset procedure which is extremely simple. The procedure is based on the simple distributed snapshot algorithm [CL-85]. The reset procedure is performed only on nodes of the network which have participated in the recent execution of the fault-intolerant algorithm. If a topological change occurs