Network Convergence in the Presence of Omission Faults*

M. H. Azadmanesh
Computer Science Department
University of Nebraska at Omaha
Omaha, NE 68182-0459

A. W. Krings
Computer Science Department
University of Idaho
Moscow, ID 83844-1010

Abstract

Network Convergence in the presence of various fault modes has been studied for completely connected networks extensively. However, complete connectivity is impractical for large distributed systems. In addition, to attain network convergence for partially connected systems, most research presume message relays by intervening nodes. In other words, the network is logically converted to a completely connected network.

This research obtains the Global Network Convergence and the Fault Tolerance for a large octagonal torus in the presence of Omission faults without message relays. The analysis uses the Fault-Tolerant Midpoint voting algorithm.

This research will reveal the challenging task of expanding the results into a general case of any network topology, in combination with any voting algorithm, involving different fault modes.

1 Introduction

One of the fundamental problems in distributed systems is reaching agreement among the non-faulty computing modules[1]. The objective is for the non-faulty computing modules to reach a common decision. This problem arises frequently in practice. For example, when multiple aircraft control processors need to agree on flight data, or when copies of data in a distributed database system must be updated in such a way to ensure consistency among all the copies.

Some applications require the computing modules to agree on a single exact value. This problem, called Byzantine Agreement, has been studied extensively. In other applications, the agreement is approximated rather than exact. For instance, in clock synchronization or sensor data management [4], it is sufficient for each module to agree (vote) on a value different from other non-faulty modules by a predefined tolerance of ϵ.

The problem of *Approximate Agreement* (AA) has been investigated by many authors. Most authors assume complete connectivity among nodes [1, 4]. In practice, however, most network topologies are partially connected. To our knowledge, we are not aware of any research that solves the AA problem for partially connected networks. But, some progress in this direction has been made [2, 5]. In [5], Kieckhafer and Azadmanesh introduced a family of convergent voting algorithms capable of reaching agreement between two nodes. They gave the convergence rate and the conditions that must be met to guarantee convergence between the two nodes. Their findings are based on a 3-mode fault model consisting of *benign*, *symmetric*, and *asymmetric* faults. A fault is benign when it is self-incriminating or self-evident to all nodes. A symmetric fault occurs when a faulty node is capable of sending the same erroneous message to all receiving nodes. An asymmetrically faulty node behaves arbitrarily and thus is capable of transmitting conflicting values to various non-faulty nodes. In [2], Azadmanesh and Krings used a different approach, and extended the results of [5] to include *omission* faults. A node behaves omissively when its message is not received by at least one non-faulty node.

The recent progress has shed much light for better understanding of the AA problem in partially connected networks without message relays, and identified many of the parameters involved in reaching agreement. But, as indicated, these studies deal with agreement between two nodes, referred as *local* agreement. To reach agreement within the entire network, all non-faulty nodes in the network must converge. This is called *global* convergence.

This paper is concerned with global convergence, also called *network convergence*. Network convergence takes advantage of local convergence. Without local convergence, global convergence is not possible. We

*This work was supported by the College of Information Science & Technology (IS&T) Research Program.

[1]Herein, we refer to a computing module to be a process, a processor on which a process is executed, or a node in a network.

will show that global convergence without message relay is a very challenging problem and convergence depends heavily on the distribution of values within the network and the topology of the network.

The rest of the paper is organized as follows. Section 2 identifies some of the difficulties in reaching network convergence in comparison to completely connected networks. It also introduces the model of communication, and some definitions used throughout the paper. Section 3 discusses the general convergence process. The discussion is based on a fault free network which will give us good intuition on how to deal with network convergence in presence of faults. Section 4 finds the worst convergence rate and the number of rounds needed for convergence. It will also show that convergence occurs if each node experiences at most two omissions, otherwise convergence can not be guaranteed. Finally, Section 5 concludes the paper and provides some remarks for future research.

2 Partially Connected Networks

If the physical connectivity of the system is not complete, then messages are relayed by intervening nodes to achieve complete "logical" connectivity. As a system grows large, so does the traffic required for message relays. Thus, the assumption of complete connectivity restricts the application of convergent voting to relatively small systems. In this research, however, *the relay of messages is prohibited.* Each node receives only those messages initiated by its immediate neighbors. Global convergence must then occur with each node acting only on local information. This approach has the disadvantage that the system will converge more slowly than a completely connected system. However, it has the advantage that the overhead of messaging becomes independent of the number of nodes in the network.

In completely connected networks global convergence and local convergence are identical, since every node is connected to every other node. In such networks, convergence is "single-step" and follows a number of rounds until the conditions of AA are met. In each round, each node broadcasts its value to all nodes. Each node then collects the values broadcast to it, including its own, and then executes an approximation function F, to obtain its latest voted value. The voted value is then broadcast to be used in the next round of voting process. The conditions of AA are met if each round is convergent, i.e. the range of the correct values is reduced in each round. This property, called *single-step convergence*, guarantees that the range of correct values will eventually be less than ϵ.

The property of single-step convergence does not hold in partially connected networks. Two immediate nodes i and j may receive values from their respective neighbors that are not shared with each other. Since these values change with every round, nodes i and j may diverge with respect to the previous round. Hence, during the course of global convergence a cluster of local nodes may go through a period of convergence and divergence before they finally converge. As a result, it is very difficult to determine the network convergence rate.

Another difficulty that arises in partially connected networks is the number of faults that can be handled. In completely connected networks, the upper limit on the number of faulty nodes depends on the number of nodes in the network. This however is not true for partially connected networks. The upper limit is bound by the connectivity between each pair of nodes rather than the total number of nodes in the network. Otherwise, the network may become divergent.

The Communications Model – The communication model is an $N \times N$ two-dimensional, octagonal mesh with wraparound links, giving it regularity and symmetry. The communication links are bidirectional. Figure 1 shows a 5×5 model with all the connections. To reference our communication model, hereafter, we use the word "torus".

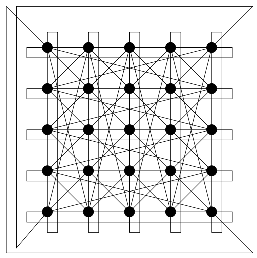

Figure 1: Example of a 5×5 octagonal torus.

In large distributed networks, omission faults may be the most common type of faults, due to local diagnosis of the message received, delay in communication, and link or node failure. By far, the most common type of omission faults are caused by *dropouts.* A dropout is a transient line outage lasting for fractions of a second to a few seconds. A node is said to "have experienced an omission" when it did not receive a message sent to it. This paper is mainly concerned with one common form of omissive faults called *Strictly Omissive Faults.* The word "strict" indicates that the receiving nodes either do not receive the message, or the message received is correct. We distinguish between two forms of strictly omissive faults:

a. **Strictly Omissive Asymmetric Fault:** It occurs when a message is received by at least one but not all the receiving nodes.

b. **Strictly Omissive Symmetric Fault:** It occurs when a message is not received by any of the receiving nodes.

In some studies link failures were treated as node failures [7]. This interpretation lowers the network fault tolerance. It may also prevent the non-faulty nodes from reaching agreement. In this research we treat link failures as link failures. As a result, the values held by non-faulty nodes connected to faulty links will be used as correct values, and thus incorporated into the list of correct values. However, one needs to be careful to ensure that the validity condition is satisfied, i.e. the voted values held by each non-faulty node is strictly within the range of the initial values held by the non-faulty nodes. Accordingly, we must guarantee that the network stays connected. To be more specific, consider the simple case where one non-faulty node holds the minimum value of the correct values, and the rest of the nodes hold the maximum correct value. If the node holding the minimum value can not send its value to any of its neighbors or receive values, but otherwise functions properly, then no convergence can occur, no matter how many rounds of voting takes place. Therefore, we assume the network model can experience only Strictly Omissive Asymmetric faults. For our communication model this assumption is not unrealistic because each node is connected to 8 neighbors and the chances for all links to fail is low.

Extreme-End Value – It is either the minimum or the maximum value of all correct values in the network. A *minimum extreme-end* value is the minimum of the correct values. Similarly, a *maximum extreme-end* value is the maximum of the correct values.

Virus – It is a continuous pattern of adjacent nodes where the nodes in the pattern either all hold the minimum extreme-end or all hold the maximum extreme-end value.

Zero-Virus – It is a virus with every node holding the minimum extreme-end value. Similarly, in a *one-virus*, every node holds the maximum extreme-end value.

Network Convergence – If the network is convergent, after a number of rounds the range of the correct values must finally become smaller. To formally define the criteria for network convergence, the following definitions are need:

$\mathbf{V}_i$ = The multiset of values received in a given round by node i from its immediate neighbors, including itself. For our model, $1 \leq |\mathbf{V}_i| \leq 9$, where $|\mathbf{V}_i|$ denotes the cardinality of $\mathbf{V}_i$.

$F(\mathbf{V}_i)$ = The approximation function used by node i. Node i applies F to $\mathbf{V}_i$ in order to obtain its voted value for the next round of voting.

$\mathbf{U}_{all}$ = The multiset of all correct values in the network before the voting process begins.

$\rho(\mathbf{U}_{all})$ = The *range* of $\mathbf{U}_{all}$, i.e. $[\min(\mathbf{U}_{all}), \max(\mathbf{U}_{all})]$.

$\delta(\mathbf{U}_{all})$ = The *diameter* of $\mathbf{U}_{all}$, i.e. $\max(\mathbf{U}_{all}) - \min(\mathbf{U}_{all})$.

A voting algorithm is convergent if, after a number of rounds, the following conditions are true:

VALIDITY – For each non-faulty node i, the voted value is within the range of correct values, i.e. $F(\mathbf{V}_i) \in \rho(\mathbf{U}_{all})$.

CONVERGENCE – For each pair of non-faulty nodes, i and j, the difference between their voted values is strictly less than the diameter of the submultiset of correct values initially, i.e. $|F(\mathbf{V}_i) - F(\mathbf{V}_j)| \leq C\,\delta(\mathbf{U}_{all})$, where $0 \leq C < 1$ is the *Convergence Rate*.

The Fault-Tolerant Midpoint function [1, 4] is used to obtain the next voting values, i.e.

$$F(\mathbf{V}_i) = \frac{\max(\mathbf{V}_i) + \min(\mathbf{V}_i)}{2}$$

Network convergence occurs when after a number of rounds, either:

a) each value in zero-viruses is changed to a value other than the minimum extreme-end value,

b) each value in one-viruses is changed to a value other than the maximum extreme-end value,

c) each value in zero-viruses and one-viruses is changed to a value other than minimum and maximum extreme-end values.

When an extreme-end value held by a node changes to a different value (or when a virus changes value), we say that the value has been “dissolved”, “invaded”, or “disappeared”.

Minority Extreme-End Value – Let the extreme-end value for the zero-viruses and the one-viruses be z and o respectively. Let Z and O be the total number of extreme-end values for the zero and one viruses respectively. If $Z \leq O$, then z is a minority extreme-end value. Otherwise o is the minority extreme-end value. We will show that for network convergence to occur, the viruses holding the minority extreme-end value must be invaded.

Dissolving-Distance of a Virus – It is the minimum number of rounds to dissolve the virus.

Innermost-Node – A node in a virus is an *innermost-node* if the number of rounds needed to dissolve its value is equal to the dissolving-distance of the virus. To find an innermost-node, the minimum number of rounds required to change the value of each node in the virus is obtained. A node having the maximum of the minimum number of rounds is an innermost-node. A virus might have multiple innermost-nodes.

Maximum Effective-Value – Assume a virus has the dissolving-distance r and the innermost-node p. Then, the *maximum effective-value* of the virus is the maximum value of all nodes at distance r from p. Similarly, the Minimum Effective-Value of the virus is the minimum value of all nodes at distance r from p.

Min & Max – These are the values held by each node in the zero and one viruses respectively.

3 Fault Free Analysis

This section discusses a fault free network informally. The discussion will give us an understanding of how to analyze a network with Strictly Omissive Asymmetric faults, which is the subject of the next section. Hereafter, these faults are called "omissions".

Consider the simulation results of a torus network which keeps track of changes as nodes collect values and decide on the next voted value. Figure 2 shows a 16×16 network containing two viruses. For simplicity in the graph, the link connections are not shown but it is assumed that it is a torus network. Also for simplicity, the circles representing the nodes are not shown. Instead the values in the network represent the locations of the nodes.

The nodes marked with 0 indicate that these nodes hold the minimum extreme-end value. Similarly, the nodes marked with 1 represent those nodes holding the maximum extreme-end value. The zero-virus is a 6×6 matrix. The area around the zero-virus is initialized to random values between the minimum and maximum extreme-end values. These values are indicated by "r". The r^+ represents the maximum of the random values. The rest of the network is filled with a one-virus .

After one round of voting, Figure 3 shows that the zero-virus has shrunk into a 4×4 matrix. This is because the nodes at the edge of the virus receive both the minimum extreme-end and non-extreme-end values. Thus, their values change into non-extreme values. But the nodes inside of the virus receive the same value Min from the adjacent nodes. Hence, those nodes stay at the same voted value Min.

The nodes marked with the non-extreme-end value 1^+ indicate that, in some round, they have had the maximum voted value among the nodes at the edge of the zero-virus . The nodes marked with 2 show the nodes which have changed their values to a non-extreme value.

In the second round, the next edge layer of the zero-virus is dissolved. After 3 rounds of voting, Figure 4 shows the network has converged, because the zero-virus has been dissolved. To this end a few remarks need to be made.

- Not all viruses have clean shapes, as in the previous figures. Viruses can be of any form.
- The number of rounds for a virus to disappear is the maximum of the shortest distance from any node of the virus to the edge of the virus plus one.
- In the Fault-Tolerant Midpoint function, the values other than the minimum and the maximum values in a multiset are ignored. Because of this property, if a zero-virus takes r rounds to disappear, the maximum value of all values at dissolving-distance r, i.e. the maximum effective-value, directly impacts the next extreme-end value. Similarly, for a one-virus, it is the minimum effective-value that directly impacts the next maximum extreme-end value. As a result, these two extreme-end values will determine the next network convergence rate.

  For example, look at the simulation results for the previous example. By looking at the 1^+ marks and the location of where r^+ appeared first, we can trace the maximum value from each layer of the virus to the next virus layer, all the way to an innermost node. Figure 5 shows the trace of the maximum values to the innermost node. Note that since a layer may have multiple nodes holding the same maximum value, and hence multiple traces may exist, only one of the paths is traced. The path is shown as a series of triangles pointing to the direction of the next change. The innermost node is shown as a $\star$.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	r	r	r	r	r	r	r	r	1	1	1	1	1
1	1	1	r	0	0	0	0	0	0	r	1	1	1	1	1
1	1	1	r	0	0	0	0	0	0	r	1	1	1	1	1
1	1	1	r	0	0	0	0	0	0	r^+	1	1	1	1	1
1	1	1	r	0	0	0	0	0	0	r	1	1	1	1	1
1	1	1	r	0	0	0	0	0	0	r	1	1	1	1	1
1	1	1	r	0	0	0	0	0	0	r	1	1	1	1	1
1	1	1	r	r	r	r	r	r	r	r	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 2: A network with two viruses (zero & one).

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
1	1	2	2	2	0	0	0	0	1^+	2	2	1	1	1	1
1	1	2	2	2	0	0	0	0	1^+	2	2	1	1	1	1
1	1	2	2	2	0	0	0	0	1^+	2	2	1	1	1	1
1	1	2	2	2	0	0	0	0	2	2	2	1	1	1	1
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3: The network after round 1.

- If there are more than one zero-virus with the same dissolving-distance r, the minimum of the maximum values of the nodes at distance r determines the next minimum extreme-end value. This is because the next minimum extreme-end value is the minimum voted value of the innermost nodes, and these nodes are associated with the minimum of the maximum values of all zero-viruses with the same dissolving-distance .

4 Convergence with Omission Faults

The following lemma determines the upper limit on the number of rounds before network convergence occurs. It will show that the worst number of rounds to converge is proportional to the diameter of the network. However, this upper limit is not necessarily an indication of how bad the voting algorithm performs. More specifically, the network might converge in one round and still have the worst convergence possible. As a result, the distribution of the values plays a major role in the number of rounds for convergence.

LEMMA 1 : *Given an $N \times N$ torus, if the network is convergent, the upper limit on the number of rounds before a convergence occurs is: $\lfloor \lceil N/2 \rceil /2 \rfloor$.*

PROOF: Due to space limitation, the proof is omitted. □

LEMMA 2 : *Assume a set of zero-viruses $\{v_1, \ldots, v_n\}$ with their maximum effective-values $\{x_1, \ldots, x_n\}$ respectively. Also assume the dissolving-distance of each zero-virus is $r_\circ$. Among these viruses, the next minimum extreme-end value is: $((2^{r_\circ} - 1)\mathrm{Min} + x_\circ)/2^{r_\circ}$, where $x_\circ = \min(x_1, \ldots, x_n)$.*

PROOF: Arbitrarily choose the virus v_i, $1 \leq i \leq n$. In the first round of voting, each node at distance $r_\circ - 1$ from the innermost node of v_i receives 9 values, including its own value. One such node must receive at least one none Min value, otherwise the $r_\circ$ is not the dissolving-distance for the v_i virus. Then, each node finds the next voted value by averaging the smallest and largest values in its multiset. The voted value becomes the new value to be used in the next round by its neighbors which are at distance $r_\circ - 2$. Note that in the first round the nodes at distances less than $r_\circ - 1$ do not change values because the values of their adjacent neighbors are all Min.

Among the nodes at distance $r_\circ - 2$ from the innermost-node, those nodes which are connected to the nodes at distance $r_\circ - 1$ holding the maximum value, will hold the maximum voted value in the next round. Therefore, the maximum voted value at dis-

2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	1^+	1^+	2	2	2	2	1	1
2	2	2	2	2	2	2	1^+	1^+	1^+	2	2	2	2	1	1
2	2	2	2	2	2	2	1^+	1^+	1^+	2	2	2	2	1	1
2	2	2	2	2	2	2	2	1^+	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 4: The network after round 3.

2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	1^+	1^+	2	2	2	2	1	1
2	2	2	2	2	2	2	$\star$	$\triangleleft$	$\triangleleft$	$\triangleleft$	2	2	2	1	1
2	2	2	2	2	2	2	1^+	1^+	1^+	2	2	2	2	1	1
2	2	2	2	2	2	2	2	1^+	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 5: The trace of the maximum voted value to the innermost node.

tance $r_\circ - 1$ at round 1 is: $(\text{Min} + x_i)/2$. In the second round of voting, the maximum voted value at distance $r_\circ - 2$ from the innermost-node node will be $(\text{Min}+(\text{Min}+x_i)/2)/2 = (3\text{Min}+x_i)/4$. The shortest distance from the innermost-node to x_i is $r_\circ$, and the maximum value at any particular node is the result of one of its neighbors holding the maximum value in the previous round. Therefore, after $r_\circ$ rounds, the voted value of the innermost-node will be:

$$\frac{(2^{r_\circ} - 1)\text{Min} + x_i}{2^{r_\circ}} \tag{1}$$

Since there are n zero-viruses, there are n different expressions of (1). Among all these expressions, the one with the minimum value will be the next minimum extreme-end value. According to the assumption, $x_\circ$ is the minimum in $\{x_1, \ldots, x_n\}$. Thus, the innermost-node of the virus with the effective-value $x_\circ$ will contain the next minimum extreme-end value. $\square$

LEMMA 3 : *Assume a set of one-viruses $\{v_1, \ldots, v_n\}$ with their minimum effective values $\{x_1, \ldots, x_n\}$ respectively. Also assume the dissolving-distance of each one-virus is r_1. Among these viruses, the next maximum extreme-end value is: $((2^{r_1} - 1)\text{Max} + x_1)/2^{r_1}$, where $x_1 = \max(x_1, \ldots, x_n)$.*

PROOF: The proof is similar to Lemma 2. $\square$

The theorem below uses the definitions:

$\varphi_\circ =$ A positive value indicating the difference between the minimum extreme-end and the next minimum non-extreme-end value. Thus, the minimum none extreme-end value is $(\text{Min} + \varphi_\circ)$.

$\varphi_1 =$ A positive value indicating the difference between the maximum extreme-end and the next maximum non-extreme-end value. Thus, the maximum none extreme-end value is $(\text{Max} - \varphi_1)$.

THEOREM 1 : *The convergence rate of a torus is:*

$$C = 1 - \frac{\min(\varphi_\circ, \varphi_1)}{2^{\lfloor \lceil N/2 \rceil /2 \rfloor} \delta(\mathbf{U}_{all})} \tag{4.1}$$

PROOF: Lemma 2 showed that among a group of zero-viruses with the same dissolving-distance, the next minimum extreme-end value is:

$$\frac{(2^{r_\circ} - 1)\text{Min} + x_\circ}{2^{r_\circ}} = \text{Min} + \frac{x_\circ - \text{Min}}{2^{r_\circ}} \tag{4.2}$$

The lower limit of (4.2) occurs when $r_\circ$ is maximized and $x_\circ$ is minimized. Thus, (4.2) becomes:

$$\text{Min} + \frac{(\text{Min} + \varphi_\circ) - \text{Min}}{2^{\lfloor \lceil N/2 \rceil /2 \rfloor}} = \text{Min} + \frac{\varphi_\circ}{2^{\lfloor \lceil N/2 \rceil /2 \rfloor}}$$

If all zero-viruses disappear before all one-viruses, then the maximum extreme-end value will still be Max and the minimum extreme-end values will be (4.2). Thus,

$$\begin{aligned}\max(|F(\mathbf{V}_i) - F(\mathbf{V}_j)|) &= \text{Max} - \text{Min} - \frac{\varphi_\circ}{2^{\lfloor \lceil N/2 \rceil /2 \rfloor}} \\ &= \delta(\mathbf{U}_{all}) - \frac{\varphi_\circ}{2^{\lfloor \lceil N/2 \rceil /2 \rfloor}}\end{aligned}$$

As a result, the convergence rate is:

$$C = 1 - \frac{\varphi_\circ}{2^{\lfloor \lceil N/2 \rceil /2 \rfloor}\delta(\mathbf{U}_{all})} \tag{4.3}$$

Following the same line of reasoning, if one-viruses disappear before the zero-viruses, then

$$C = 1 - \frac{\varphi_1}{2^{\lfloor \lceil N/2 \rceil /2 \rfloor}\delta(\mathbf{U}_{all})} \tag{4.4}$$

Comparing (4.3) and (4.4), the worst convergence rate is the minimum of the two expressions. □

Theorem 1 showed a worst condition for convergence. There could be other situations for worst convergence. In general, for convergence to become worse, none extreme-end nodes at the edge of the zero-viruses with values higher than the smallest none extreme-end values should behave omissively. This way, only the nodes with smaller none extreme-end values have the chance to invade the zero-viruses. As a result, the voted values of the nodes in the zero-viruses will become smaller. A similar situation exists for the one viruses, where the nodes holding low values may behave omissively. To make the convergence even worse, only one of the zero or one viruses should disappear, but not both. If both viruses disappear in the same round, the range of correct values will be smaller which implies better convergence.

In a torus, the best convergence occurs when every node has one neighbor holding the value Min and another holding the Max value. As a result, each node has in its multiset both the Min and Max values. Therefore, it takes one round for every node to converge. According to Lemma 2, the next minimum extreme-end value is

$$\frac{(2^{r_\circ} - 1)\text{Min} + x_\circ}{2^{r_\circ}} = \frac{(2-1)\text{Min} + \text{Max}}{2} = \frac{\text{Min} + \text{Max}}{2}$$

Similarly, by Lemma 3, the next maximum extreme-end value is

$$\frac{(2^{r_1} - 1)\text{Max} + x_1}{2^{r_1}} = \frac{(2-1)\text{Max} + \text{Min}}{2} = \frac{\text{Max} + \text{Min}}{2}$$

As a result, the convergence is perfect, i.e. $C = 0$.

With this discussion and from Theorem 1, the convergence rate is highly dependent on the distribution of the values, and thus the values of $x_\circ$ and x_1 at any point of time have direct impact on the convergence rate. This is in contrast to voting algorithms such as MSR or Egocentric algorithms [1, 6], used for completely connected networks, whose C expressions do not depend on the individual values, the distribution, or the range of values.

The next lemma and theorem show that a torus is two fault-tolerant.

LEMMA 4 : *If an $N \times N$ torus is convergent, then there is at least one 3×3 block containing at least 5 non-minority extreme-end values.*

PROOF: Consider a 3×3 block of the torus. Rewrite every such block in a column form, row after row, to get a 9×1 matrix. Placing all these columns next to each other forms a $9 \times N^2$ matrix. To contradict the assumption of the lemma, each column of the matrix must have at least 5 minority extreme-end values. Since there are N^2 columns in the matrix and each column represents a different block, there must be at least $5N^2$ minority extreme-end values.

On the other hand, since each row represents the entire network, the total number of minority extreme-end values in each row is the same, say X, for a total number of $9X$ in the entire matrix. Therefore, $5N^2 \leq 9X$ must hold, which implies $X \geq \frac{5}{9}N^2$. But since X is the total number of minority extreme-end values, its value can not be greater than $N^2/2$. Thus, by contradiction, the torus must contain one 3×3 block with at least 5 non-minority extreme-end values. □

THEOREM 2 : *A torus is 2-fault tolerant if it contains at least one zero-virus and one one-virus.*

PROOF: For the network to be possibly convergent, it must contain at least two different values, which implies that the network must contain at least one zero-virus and one one-virus.

Lemma 4 showed that there must be at least one 3×3 block with at least 5 non-minority extreme-end values. Thus, consider a block with 5 non-minority extreme-end values. By observation, no matter how these 5 values are distributed within the block, there is at least one node holding the minority extreme-end value that is adjacent to 3 nodes with non-minority extreme-end values. Thus, by the hypothesis that each node encounters at most two omissions, this node will become a non-minority extreme-end node in the next round of voting. As a result, after one round, the block will contain 6 non-minority extreme-end values. This will cause another minority node to be adjacent to three non-minority extreme-end values, and thus will change value. As a result, the block will be finally invaded with non-minority extreme-end values.

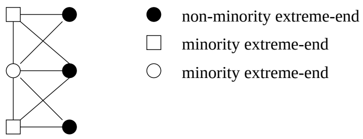

Figure 6: The connections between the left edge of a 3×3 invaded block and its left neighbors.

Now consider the left edge of a 3×3 invaded block, shown in Figure 6 as a column of $\bullet$. If there is a minority extreme-end value to the left-center of the edge, i.e. node $\circ$, it will be invaded in the next round of voting, which will in turn cause the nodes $\Box$ to be invaded in the follow-up round. If the $\circ$ node is not a minority extreme-end but the $\Box$ nodes are, all of these nodes will become non-minority in the next round. Thus, when we consider the same argument for each edge of the 3×3 block, it is obvious that the block will grow in all directions until the whole network becomes invaded. As a result, the network will be convergent, because, at least, the minority extreme-end values will be dissolved into non-extreme-end values. $\Box$

Corollary 2.1 : *A torus is convergent if each of its nodes can experience at most two omissive faults.*

PROOF: Due to space limitation, the proof is omitted. $\Box$

5 Summary

This paper has addressed the Approximate Agreement problem for an octagonal mesh in the form of a two dimensional torus. It was shown that network convergence, without the benefit of message relays, involves a number of challenging issues not inherent in completely connected networks, including:

- The network convergence is not single-step.
- The convergence rate is highly dependent on the distribution of initial values.
- The number of rounds from one convergence to another is not necessarily the same.
- Convergence in less number of rounds is not an indication of better convergence.
- Convergence is highly dependent on network topology.

Because of the difficulty of addressing these issues, this paper used a simple voting algorithm, Fault-Tolerant Midpoint. Using this voting algorithm, network convergence was obtained in the presence of omission faults. First, the worst number of rounds for convergence was analyzed. Based on this result, the worst network convergence was then attained. In general it was shown that network convergence becomes worse if the nodes holding values close to extreme-ends of correct values behave omissively. For future research, a number of questions are still open:

- For the same fault model and network topology, how can the analysis be extended to include other voting algorithms such as Fault-Tolerant Mean?
- How can the results be extended to include hybrid fault models [6]?
- How can the knowledge gained in this paper be extended to include other network topologies?

References

[1] M.H. Azadmanesh, A.W. Krings, "Egocentric Voting Algorithms", *IEEE Trans. Reliability*, Vol. 46, No. 4, pp. 494-502, Dec 1997.

[2] M.H. Azadmanesh, A.W. Krings, "A Step Toward Global Convergence in Partially Connected Networks", *10th Int'l Conf. on Parallel and Distributed Computing Systems*, pp. 234-241, 1997.

[3] F. Cristian, H. Aghili, R. Strong, "Clock Synchronization in the Presence of Omission and Performance Faults, and Processor Joins", *16th Int'l Conf. Fault-Tolerant Computing*, 1986.

[4] D. Dolev, N.A. Lynch, et al, "Reaching Approximate Agreement in the Presence of Faults", *JACM*, Vol. 33, No. 3, pp. 499-516, Jul 1986.

[5] R.M. Kieckhafer, M.H. Azadmanesh, "Low Cost Approximate Agreement in Partially Connected Networks", *Journal of Computing and Information*, Vol. 3, No. 2, pp. 53-85, 1993.

[6] R.M. Kieckhafer, M.H. Azadmanesh, "Reaching Approximate Agreement With Mixed Mode Faults", *IEEE Trans. Parallel Distributed Systems*, Vol. 5, No. 1, pp. 53-63, Jan 1994.

[7] P.M. Thambidurai, Y.K. Park, "Interactive Consistency with Multiple Failure Modes", *Proc. 7th Reliable Distributed Systems Symp.*, Oct 1988.