IEEE Transactions on Reliability, Vol 46, No 4, pp. 494-502, Dec 1997.

Egocentric Voting Algorithms*

M.H. Azadmanesh, Member IEEE

University of Nebraska at Omaha

A.W. Krings, Member IEEE

University of Idaho, Moscow

Key Words – Approximate Agreement, Byzantine Agreement, Clock Synchronization, Convergent Voting Algorithms, Fault-Tolerant Multiprocessors, Hybrid Faults.

Summary and Conclusions – An important problem in distributed systems is distributed agreement. One form of distributed agreement is *Approximate Agreement* in which non-faulty processes need to agree on values within a predefined tolerance. In this paper, Approximate Agreement voting algorithms are partitioned into three broad categories called *Anonymous*, *Egophobic*, and *Egocentric*. Each category is further subdivided into families of algorithms. One such family of voting algorithms which belongs to the Egocentric category will be examined. Ad-hoc analyses of some members of this family of algorithms have been studied *individually* under an overly conservative fault-model in which all faults are presumed to behave in the worst case Byzantine manner. This paper develops a methodology to quickly determine the fault-tolerance and the convergence rate of any member of this family under a hybrid fault-model consisting of *asymmetric*, *symmetric*, and *benign* faults. The results will be weighted against those of several known voting algorithms. Finally, a sub-family of Egocentric algorithms with optimal performance will be identified.

1 Introduction

In distributed systems, it is often necessary for non-faulty processes to reach agreement on data values in the presence of faulty processes. The agreement is *exact* if the processes

*This work was supported by NASA Space Grant Consortium.

must arrive at a single consensus value. In many applications, however, the agreement is *approximate* [12, 18, 20]. The processes need to agree on values which are within a predefined tolerance of each other. The problem of agreement becomes considerably complex when faulty processes are capable of sending erroneous values to non-faulty processes.

Recently, a family of voting algorithms have been developed for the *Approximate Agreement* problem. These algorithms, called *Mean-Subsequence-Reduced* (MSR) [9], employ the hybrid fault model of Thambidurai and Park [19], which partitions faults into three modes: asymmetric (Byzantine), symmetric (single-valued) and benign (self-incriminating).

This paper employs the same hybrid fault-model but for a new family of voting algorithms called *Mean-Subsequence-Egocentric* (MSE). MSE algorithms in addition to the conditions required to satisfy the Approximate Agreement problem, require that the interval spanned between any pair of the initial correct values be within a prespecified positive value, say φ. Some members of this family of algorithms have been used in clock synchronization [11, 12, 16]. MSE algorithms can also be used in applications where a certain amount of discrepancy among processes, for instance, due to round-off errors, must be tolerated [1, 4, 5, 7, 15]. However, no general analysis exists in literature that can easily determine the performance of any of these algorithms under Byzantine or hybrid fault-models.

Section 2 presents some background material to understand the convergent voting process. Section 3 introduces some general categories of voting algorithms, where each category may contain more than one family of algorithms. Section 4 develops simple expressions to determine the fault-tolerance and to easily measure the performance of any MSE algorithm. Section 5 derives the optimal performance for MSE algorithms. In addition, it will present the criteria for a sub-family of algorithms that will produce optimal performance. Section 6 compares the performance of MSE algorithms with that of some known convergent voting algorithms. Finally, Section 7 concludes the paper with a summary and some remarks on future research.

1.1 Notation

$\mathbf{V}$	the Egocentric voting multiset; $V = \|\mathbf{V}\|$
$\rho(\mathbf{V})$	$[\min(\mathbf{V}), \max(\mathbf{V})] \;=\; [v_1, v_V]$: *range* of $\mathbf{V}$
$\delta(\mathbf{V})$	$\max(\mathbf{V}) - \min(\mathbf{V}) \;=\; v_V - v_1$: *diameter* of $\mathbf{V}$
$\mathbf{S}$	$Sel_\sigma(\mathbf{V})$: selected multiset containing σ elements from $\mathbf{V}$
σ	size of selected multiset $\mathbf{S}$
$F(\mathbf{V})$	mean $[Sel_\sigma(\mathbf{V})]$: Egocentric Approximation Function
C	$\omega_{\gamma_{(a+s)}}/\sigma$: convergence rate of Egocentric algorithms (section 4.1)
$\mathbf{U}_{i \cap j}$	multiset of correct values received by both non-faulty processes i and j
N	total number of physically independent processes in the system
$a,\ s,\ b$	number of asymmetric, symmetric, and benign faults in the system respectively
n	$N - b$: number of elements in $\mathbf{V}$ after the removal of benign errors
α, β	current voted value of non-faculty processes i and j respectively
φ	$\|\alpha - \beta\|$: maximum discrepancy amount between the current values of two arbitrary non-faulty processes i and j
$k(g)$	index position of the g^{th} selected element in $\mathbf{V}$
$\Delta k(g,h)$	$k(h) - k(g)$: number of elements in $\mathbf{V}$ spanned by elements $(s_g, \ldots, s_h)$ in $\mathbf{S}$
z	*effective* number of asymmetric errors in a voting multiset $\mathbf{V}$
γ_z	minimum value which ensures that $k(h) - k(g) \geq z$ for any values g and h, $g \leq h \leq \sigma$ and $z \geq 0$ (section 4.1)
$e_{i,g}$	weight of the g^{th} element in $\mathbf{S}_i$ for non-faulty process i (section 4.1)
$E_i\_Sel_\sigma$	$\langle e_{i,1}, \ldots, e_{i,\sigma} \rangle$: Enumerated Selected List for non-faulty process i (section 4.1)
ω_{γ_z}	$\sum_{g=1}^{\gamma_z} [e_{i,\sigma-g+1} - e_{j,g}]$: maximum weight-difference between the last γ_z elements of $\mathbf{S}_i$ and the first γ_z elements of $\mathbf{S}_j$ (section 4.1)

2 Background and Definitions

The objective of reaching Approximate Agreement is to guarantee that, at the termination of a voting algorithm, the voted value for each non-faulty process is within the range of the initial correct values and that the difference between any pair of voted values for non-faulty processes is strictly within a prespecified small positive real value ϵ. The final voted value, at the end of the voting algorithm, is obtained by employing multiple rounds of message exchange. In each round, each process sends its value to all receiving processes. Upon receipt of a collection of values, each process executes a function F, to obtain its latest voted value, which is used in the next round of message exchange. The objective of Approximate Agreement can be achieved by ensuring that each round is convergent, i.e. the range of the correct values is reduced in each round [6, 12, 21]. This property, called *single-step convergence*, guarantees that the range of values will eventually be less than ϵ, given enough rounds. A voting algorithm with this property is called *single-step convergent voting algorithm*. All voting algorithms considered herein have this property.

Systems can be *synchronous* or *asynchronous* [6]. In a synchronous distributed system, the processing and the communication delays of non-faulty processes are bounded. There is thus a point in time by which any process executing a convergent voting algorithm will have received all data from all non-faulty processes. Any data arriving after that time is considered to be from a faulty process. By contrast, asynchronous systems impose no bounds on process operation. The analysis of Approximate Agreement in this research is restricted to synchronous systems with complete connectivity.

2.1 Real–Valued Multisets

Approximate Agreement requires the manipulation of multisets of real values. A multiset is a collection of objects similar in concept to a set. However, it differs from a set in that all elements of a multiset are not necessarily distinct. A multiset of real numbers can be represented as a monotonically increasing sequence of the real values of its elements, i.e. $\mathbf{V} = \langle v_1, \ldots, v_V \rangle$ is ordered such that: $v_i \leq v_{i+1}\ \forall\ i \in \{1, \ldots, V-1\}$. The size of $\mathbf{V}$ is $V = |\mathbf{V}|$. Some of the useful operations on multisets are:

$\rho(\mathbf{V})$ = $[\min(\mathbf{V}), \max(\mathbf{V})] = [v_1, v_V]$; the real interval spanned by $\mathbf{V}$. $\rho(\mathbf{V})$ is called the *range* of $\mathbf{V}$.

$\delta(\mathbf{V})$ = $\max(\mathbf{V}) - \min(\mathbf{V}) = v_V - v_1$; the difference between the maximum and minimum values of $\mathbf{V}$. $\delta(\mathbf{V})$ is called the *diameter* of $\mathbf{V}$.

$\text{mean}(\mathbf{V})$ = The arithmetic mean of the real values of all elements of $\mathbf{V}$.

Subsequences – Consider two non-empty multisets $\mathbf{U}$ and $\mathbf{V}$, where $\mathbf{U} \subseteq \mathbf{V}$. $\mathbf{U}$ is a subsequence of $\mathbf{V}$ if there is an order-preserving one-to-one mapping k, from the *indices* of $\mathbf{U}$ to the *indices* of $\mathbf{V}$, i.e. $u_j = v_{k(j)}\ \forall\ j \in \{1, \ldots, U\}$ and $k(j) < k(j+1)\ \forall\ j \in \{1, \ldots, U-1\}$.

2.2 Single-Step Convergence

The property of single-step convergence is formally defined in terms of the following:

$\mathbf{V}_i$ = The multiset of values received in a given round by process i.

$\mathbf{U}_{i \cap j}$ = The multiset of correct values received by both non-faulty processes i and j, i.e. those values generated by non-faulty processes.

Each non-faulty process i executes a voting algorithm, producing a voted value $F(\mathbf{V}_i)$. The property of single-step convergence is guaranteed if *both* of the following conditions are true following every round of voting:

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

Convergence – For each pair of non-faulty processes, i and j, the difference between their voted values is *strictly less than* the diameter of the multiset of correct values received, i.e. $|F(\mathbf{V}_i) - F(\mathbf{V}_j)| < \delta(\mathbf{U}_{i \cap j})$.

Performance of a convergent voting algorithm is measured by its convergence rate. Assuming

that $\delta(\mathbf{U}_{i\cap j}) > 0$, the *convergence rate* C is the maximum possible value of the ratio:

$$C = \frac{|F(\mathbf{V}_i) - F(\mathbf{V}_j)|}{\delta(\mathbf{U}_{i\cap j})} \tag{2.1}$$

Fault-tolerance is the minimum number of processes N required to tolerate t faults. Under the Byzantine fault-model, it has been shown [6] that single-step convergence can be guaranteed for synchronous systems only if: $N \geq 3t + 1$.

2.3 Fault Classification

In many applications true Byzantine faults occur rarely and under complex conditions. This limitation leads to system designs which are more complex and require a greater number of processes than necessary to guarantee convergence. A more realistic approach to designing fault-tolerant distributed systems is to incorporate different types of faults and place a limit on the maximum number of faults in each class. Accordingly, Thambidurai and Park [19] partitioned faults into three types: *Benign*, *Symmetric*, and *Asymmetric* (Byzantine) faults. Benign faults, also called *manifest* faults [14], are defined as those which are self-incriminating or self-evident to *all* processes. A symmetric fault is defined as a fault whose value is perceived identically by all receiving non-faulty processes. An asymmetric fault is one which is capable of sending conflicting (arbitrary) messages to different non-faulty processes. If the number of asymmetric, symmetric, and benign faults are indicated by a, s, and b, respectively, then the total number of faults in the system is $t = a+s+b$. Thambidurai and Park used this partitioning to derive tighter bounds on the fault-tolerance of Byzantine Agreement algorithms [19][1]. By employing the same model, we will achieve a tighter bound on fault-tolerance and show that convergence can be determined more accurately and easily.

3 Categories of Voting Algorithms

A variety of convergent voting algorithms have been published which employ the iterative approach of successively receiving data elements and voting on the elements received [6, 9,

[1] The Interactive Consistency algorithm proposed by these authors is flawed. Lincoln and Rushby presented a correct algorithm under the same fault model [13].

12, 21]. Each iteration (round) follows a number of phases:

1. *Broadcast* – Each process broadcasts its current value to every process including itself.
2. *Collect* – Each process collects the values broadcast by other processes including itself.
3. *Sample* – Each process filters the values to produce the voting multiset $\mathbf{V}$. The mechanism for the production of the multiset depends on the fault-model employed.
4. *Execute* – The approximation-function $F(\mathbf{V})$ is applied to generate a single voted value.

Generally, voting algorithms fall into several broad categories, depending on the sampling method used. The sampling phase may perform tasks such as filtering out the recognized benign errors, substituting default values for missing values, etc. Within each category, the voting algorithms are grouped into families of algorithms. Each family is defined based on the format of the approximation-function. This function normally contains parameters that can vary from one voting algorithm to another, and hence produces different performance results for voting algorithms within the same family. Voting algorithms may fall into the following categories:

1. *Anonymous* – This category is so named because the value of the originating node has no bearing on how it is processed during the sampling phase. Therefore, no attempt is made to determine as to whether the value originated from a faulty or a non-faulty process [6, 9].
2. *Egophobic* – This category is so named because during the sampling phase, the voting algorithm favors values which most differ from its own value or even disregards its own value entirely [8, 10, 17].
3. *Egocentric* – In contrast to Egophobic, algorithms in this category tend to favor values closest to a process's own value. For example, a process may discard values which differ from its own value by more than a specified tolerance [11, 12, 16, 17].

One family of algorithms that belongs to the anonymous category is called *Mean-Subsequence-Reduced* or MSR. MSR algorithms use the approximation-function [9]:

$$F(\mathbf{V}) = \text{mean}\left[Sel_\sigma\left(Red^\tau(\mathbf{V})\right)\right]. \tag{3.1}$$

The "Reduction" function Red^τ removes the τ largest and τ smallest elements from multiset $\mathbf{V}$. The "Selection" function Sel_σ then produces the subsequence $\mathbf{S}$ containing σ elements of the reduced multiset. The final voted value is the arithmetic mean of the selected multiset.

4 MSE Voting Algorithms

The focus of this paper is on a particular family of Egocentric algorithms with the property, that for any two arbitrary non-faulty processes i and j with values α and β, $|\alpha - \beta| \leq \varphi$, where φ is a positive real value. The rationale for φ is that, in some applications, processes may not all produce exactly the same value. Thus, a certain amount of discrepancy among processes must be tolerated. This discrepancy amount depends on the application, and the upper bound on φ is known. The discrepancy might be, for instance, due to sensors reading the same input, clocks which must stay within a predefined known bound of each other, or multiple software or hardware versions where decision-algorithms are employed to determine the final vote from similar but not identical results [1, 4, 5, 12, 18, 20].

During the sampling phase, each process discards the globally diagnosed errors, and uses its own value as a default for each missing data item, or any value that differs from its own by an amount greater than the threshold φ. Specifically, we distinguish between *globally* diagnosed (benign) and *locally* diagnosed errors. When an error is recognized globally, *all* non-faulty processes can delete them from $\mathbf{V}$ and vote with a smaller sized multiset [9]. For instance, when a faulty process sends a value that, with respect to the value of all non-faulty processes, exceeds the threshold φ. Whereas, a locally diagnosed error is detected by a subset of non-faulty processes. For instance, when a process receives a value which differs from its own by more than φ, when a process misses a correct value that was received by some other processes, or when a process detects a value that was corrupted during transmission. For such errors, the non-faulty process replaces the missing values or the erroneous values with its own value.

Consequently, for two arbitrary non-faulty processes i and j, $|\mathbf{V}_i| = |\mathbf{V}_j|$. Let N be the total number of processes in the system, and let $n = (N - b)$ be the number of data elements after the globally diagnosed benign errors are removed, so that $|\mathbf{V}_i| = n$ for any arbitrary non-faulty process i. The multiset produced by the end of the sampling phase is named the *Egocentric voting-multiset* $\mathbf{V}$. During the execution phase, each process executes the following approximation-function, reducing the multiset $\mathbf{V}$ to a single voted value $F(\mathbf{V})$:

$$F(\mathbf{V}) = \text{mean}\left[Sel_\sigma\left(\mathbf{V}\right)\right] \tag{4.1}$$

$F(\mathbf{V})$ is the *Mean* of a *Subsequence* of an *Egocentric* (MSE) voting multiset. The varying parameters of $F(\mathbf{V})$ are the number of selected elements σ, and the distribution of these elements in $\mathbf{V}$. Thus, MSE algorithms differ from each other only in their definition of Sel_σ.

4.1 Convergence Rate of MSE Voting Algorithms

The following theorem shows that the convergence rate C of an MSE algorithm depends on parameters σ, γ_z, and ω_{γ_z}. The first parameter, σ, is the size of the selected multiset $\mathbf{S} = Sel_\sigma(\mathbf{V})$. The second parameter, γ_z, is a measure of how uniformly the elements of $\mathbf{S}$ are distributed within the Egocentric voting-multiset. The third parameter, ω_{γ_z}, shows the effect of γ_z on C. Specifically, it will be shown that a voting algorithm with a lower value of γ_z produces a smaller value for ω_{γ_z} which results into a better convergence rate.

For the remainder of this subsection, γ_z and ω_{γ_z} will be defined followed by the theorem showing the convergence rate for any member of MSE algorithms. The proof of the theorem is omitted for brevity and can be found in [3]. The subsection is concluded with a working example which demonstrates how to obtain the convergence rate for a particular voting algorithm, using the parameters γ_z and ω_{γ_z}.

Definition of γ – The selected multiset $\mathbf{S} = Sel_\sigma(\mathbf{V}) = \langle s_1, \ldots, s_\sigma \rangle$ is a *subsequence* of the Egocentric voting-multiset $\mathbf{V} = \langle v_1, \ldots, v_n \rangle$. Thus, each element of $\mathbf{S}$ corresponds to one unique element of $\mathbf{V}$. Now, let g be the index of any element of $\mathbf{S}$, and let $k(g)$ be the index of the corresponding element in $\mathbf{V}$. Then, for each $g \in \{1, \ldots, \sigma\}$ there exists exactly one $k(g) \in \{1, \ldots, V\}$. This guarantees that $s_g = v_{k(g)}$ *for all possible* $\mathbf{V}$.

Given two indices into $\mathbf{S}$, g and $h \in \{1, \ldots, \sigma\}$, where $g \leq h$, define $\Delta k(g,h) = k(h) - k(g)$ as the number of elements in $\mathbf{V}$ *spanned* by elements $(s_g, \ldots, s_h)$ in $\mathbf{S}$. Thus, $\Delta k(g,h)$ is the number of elements of $\mathbf{V}$ in the submultiset $\langle v_{k(g)+1}, \ldots, v_{k(h)} \rangle$. Finally, for any non-negative integer z, γ_z is defined as the *minimum* value of $(h-g)$ which *ensures* that z elements of $\mathbf{V}$ are spanned, *independent of g and h*, i.e. γ_z is the minimum value of $(h-g)$ which ensures that $k(h) - k(g) \geq z$, for any values g and h, $g \leq h \leq \sigma$. By this definition, γ_z exists only if $|\mathbf{V}| > z$. The following pseudo-code shows how γ_z is obtained:

$\gamma_z \leftarrow -1, \;\; I \leftarrow 0$
WHILE ($I \leq \sigma - 1$ AND $\gamma_z = -1$) {
    IF (*for every* $g \in \{1, \ldots, \sigma - I\}, \;\; \Delta k(g, g+I) \geq z$)
        $\gamma_z \leftarrow I$
    $I \leftarrow I + 1$
}

By definition, γ_z can not be negative. Thus, γ_z does not exist if it is negative at the end of the pseudo-code.

The parameter z is the effective number of asymmetric errors. Specifically, z is the number of asymmetrically faulty processes, a, plus the number of symmetrically faulty processes, s, whose values in the worst case error-scenario are processed differently by the two non-faulty processes i and j. This occurs, for instance, when each symmetrically faulty process generates a value $x = (\alpha + \varphi)$. Process i will accept this value because $|x - \varphi| = \varphi$ is within the acceptable range $[0, \varphi]$. Process j, however, will replace x with β when its own value is less than α, because $|x - \beta| > \varphi$. The pair (x, β) will then behave like an asymmetrically faulty process, as if a faulty process generated conflicting values to processes i and j. Hence, the effective number of asymmetric errors z is $(a + s)$ not a.

Definition of ω_γ – By referring to the definition of convergence rate, to obtain C, the maximum of $|F(\mathbf{V}_i) - F(\mathbf{V}_j)|$ must be determined. Ultimately, one needs to find the sum-difference of the selected elements in $\mathbf{S}_i$ and $\mathbf{S}_j$, because $F(\mathbf{V})$ is defined in terms of $Sel_\sigma(\mathbf{V})$.

Let ω_γ be a weight function that uses two lists that show the weight of each element in $\mathbf{V}_i$ and $\mathbf{V}_j$. The maximum difference in these weights will determine the value of ω_γ. These lists are called Enumerated Selected Lists E_i_Sel and E_j_Sel, and are associated with the processes i and j respectively. The g^{th} element of E_i_Sel represents the weight of the $k(g)^{\text{th}}$ element in $\mathbf{V}_i$ if it is in $Sel_\sigma(\mathbf{V}_i)$. The same is true for E_j_Sel. The justification for determining these weights are explained in the proof of Theorem 1 [3].

Define two Enumerated Selected Lists $E_i\_Sel_\sigma$ and $E_j\_Sel_\sigma$ with the following weights:

$$E_i\_Sel_\sigma = \langle e_{i,1}, \ldots, e_{i,\sigma} \rangle$$

where

$$e_{i,x} = \begin{cases} 1 & : \ k(x) = 1 \\ 2 & : \ 2 \leq k(x) \leq (n-a-s) \\ 3 & : \ otherwise \end{cases} \tag{4.2}$$

and

$$E_j\_Sel_\sigma = \langle e_{j,1}, \ldots, e_{j,\sigma} \rangle$$

where

$$e_{j,y} = \begin{cases} 0 & : \ 1 \leq k(y) \leq a \\ 1 & : \ otherwise \end{cases} \tag{4.3}$$

Now define the weight function ω_{γ_z} to be:

$$\omega_{\gamma_z} = \sum_{g=1}^{\gamma_z} [e_{i,\sigma-g+1} - e_{j,g}] \tag{4.4}$$

THEOREM 1 : *Given an MSE voting-algorithm* $F(\mathbf{V})$,

$$C = \frac{\omega_{\gamma_z}}{\sigma}, \quad z = (a+s) \qquad \square \tag{4.5}$$

Theorem 1 shows the convergence rate for one round of voting when the diameter of values of non-faulty processes is φ. However, to guarantee the property of single-step convergence, φ must be reduced in each round to ensure that the diameter of non-erroneous values will

be reduced in each round. Since $\delta(\mathbf{U}_{i\cap j}) = \varphi$, it can be seen from (2.1) that the diameter of non-erroneous values in the first round in the worst case is φC. This is the new value of φ that must be used in the second round of voting. In general, the new value of φ in the i^{th} round is φC^i. In addition, the number of rounds k needed to ensure that the diameter of non-erroneous values will be within ϵ of each other can be found by solving for k in $\varphi C^k \leq \epsilon$, which is $k \geq \lceil \log_C \epsilon/\varphi \rceil$.

A Working Example – This example shows how to use the definition of ω_γ in (4.4) to arrive at the convergence rate for a particular selection function. Assume the selection function selects all odd-numbered elements from $\mathbf{V}$. Furthermore, assume $N = 10$, $a = 1$, $s = 2$, and $b = 0$. Thus:

$\mathbf{V}$	=	v_1,	v_2,	v_3,	v_4,	v_5,	v_6,	v_7,	v_8	v_9	v_{10}
		↓		↓		↓		↓		↓	
$\mathbf{S}$	=	s_1		s_2		s_3		s_4		s_5	
g	=	1		2		3		4		5	
		↓		↓		↓		↓		↓	
k(g)	=	1		3		5		7		9	
h	=	1		2		3		4		5	
		↓		↓		↓		↓		↓	
k(h)	=	1		3		5		7		9	

The first two rows show the voting multiset $\mathbf{V}$ with 10 elements followed by its corresponding selected multiset $\mathbf{S}$. Between the two rows, the arrows show which elements of $\mathbf{V}$ are selected. Thus, $\sigma = 5$. The next two rows show the index values of the elements in $\mathbf{S}$ and their mapped index values in $\mathbf{V}$, respectively. Thus, $s_g = v_{k(g)}$. For example, if $g = 3$, then $s_3 = v_{k(3)} = v_5$. The last two rows, which are the same as g and $k(g)$, are given due to the definition of $\Delta k(g, h)$.

By referring to the expression in (4.5), $z = (a + s) = 3$. To obtain γ_z, based on its definition, we need to find the smallest value $\Delta k(g, h)$ such that $\Delta k(g, h) = k(h) - k(g) \geq z = 3$, for all valid values of g and h. By inspecting the last four rows, it can be seen that the smallest

value is 2. In other words, if $h = g + 2$, then $\Delta k(g, h) \geq z \ \ \forall\ g \in \{1, \ldots, \sigma - 2\}$. Therefore, $\gamma_z = \gamma_3 = 2$. The same minimum value of γ_3 could have also been obtained by following the pseudo-code given in the definition of γ_z.

On the other hand, the expressions in (4.2) and (4.3) along with using either $k(g)$ or $k(h)$ yield:

$$\begin{array}{ll} e_{i,1} = 1 & e_{j,1} = 0 \\ e_{i,2} = 2 & e_{j,2} = 1 \\ e_{i,3} = 2 & e_{j,3} = 1 \\ e_{i,4} = 2 & e_{j,4} = 1 \\ e_{i,5} = 3 & e_{j,5} = 1 \end{array}$$

Therefore, the convergence rate is:

$$C = \frac{\omega_{\gamma_3}}{\sigma} = \frac{\sum_{g=1}^{\gamma_3} [e_{i,\sigma-g+1} - e_{j,g}]}{\sigma} = \frac{\sum_{g=1}^{2} [e_{i,6-g} - e_{j,g}]}{5} = \frac{4}{5}$$

For the rest of the paper, when there is no room for ambiguity, the subscript of $\gamma_{(a+s)}$ will not be shown.

4.2 Fault-Tolerance

Recall that $N = n + b$. To determine the lower bound of N for which a voting algorithm exists, consider the case when all elements from $\mathbf{V}$ are selected, so that $\sigma = n$. Then:

$$C = \frac{\omega_\gamma}{\sigma} = \frac{\sum_{g=1}^{\gamma} [e_{i,n-g+1} - e_{j,g}]}{n} \tag{4.6}$$

The selection of every element implies $k(g) = g$, $g \leq n$. Therefore, using (4.2), $e_{i,n-g+1} = 3$ for $1 \leq g \leq (a + s)$, and using (4.3), $e_{j,g} = 0$ for $1 \leq g \leq a$, and $e_{j,g} = 1$ for $a < g \leq (a + s)$. Consequently,

$$[e_{i,n-g+1} - e_{j,g}] = \begin{cases} 3: & 1 \leq g \leq a \\ 2: & a < g \leq (a + s) \end{cases}$$

In addition, since $\Delta k(g, g + (a + s)) = (a + s), \ \ \forall\ g \leq \sigma - (a + s)$, it follows that $\gamma = (a + s)$. Therefore, (4.6) becomes:

$$C \ = \ \frac{\sum_{g=1}^{(a+s)} [e_{i,n-g+1} - e_{j,g}]}{n}$$

$$= \frac{\sum_{g=1}^{a} [e_{i,n-g+1} - e_{j,g}] + \sum_{g=a+1}^{(a+s)} [e_{i,n-g+1} - e_{j,g}]}{n}$$

$$= \frac{3a + 2s}{n}$$

Accordingly, $n \geq (3a+2s+1)$ ensures that $C < 1$. As a result, by incorporating the impact of benign faults, a convergent voting algorithm exists only if: $N \geq 3a + 2s + b + 1$. This is the same fault-tolerance obtained by the MSR fault-model [9].

5 The Optimal Convergence Rate

The effectiveness of an MSE algorithm depends on the selection function. The selection function affects the tightness of voted values. Hence, it affects the number of rounds needed to converge to within ϵ. Therefore, in this section, the property that a selection function must hold in order for a voting algorithm to generate the optimal convergence rate will be derived. A voting algorithm with the minimum value of C has the optimal convergence rate.

The following lemma will filter out those algorithms which are shown not to be optimal. The next theorem will then use the lemma to derive the optimal convergence rate. Finally, the corollary following the theorem will show the property that a selection function must have in order for an algorithm to yield the optimal convergence rate. The proofs, omitted due to space limitation, can be found in [3].

LEMMA 1 : *Consider the voting multiset* $\mathbf{V} = \langle v_1, \ldots, v_a, v_{a+1}, \ldots, v_{n-(a+s)}, \ldots, v_n \rangle$. *Also consider the three submultisets of* $\mathbf{V}$*:* $\langle v_1, \ldots, v_a \rangle$, $\langle v_{a+1}, \ldots, v_{n-(a+s)} \rangle$, *and* $\langle v_{n-(a+s+1)}, \ldots, v_n \rangle$ *with their associated selected multisets* $\mathbf{S}_1$, $\mathbf{S}_2$, *and* $\mathbf{S}_3$, *respectively. If* $\mathbf{S} = Sel_\sigma(\mathbf{V}) = \langle \mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3 \rangle$, *and* $|\mathbf{S}_1| + |\mathbf{S}_3| \neq 0$, *then an MSE voting algorithm with selected multiset* $\mathbf{S}_2$ *will have a better convergence rate than any algorithm with selected multiset* $\mathbf{S}$. □

Lemma 1 shows that algorithms which do not select elements from the extreme ends of the voting multiset, i.e. no elements are selected from the extreme right $(a+s)$ and the extreme left a elements, have better convergence rate than algorithms which select elements from

either or both extremes of their voting multisets. The worst convergence rate occurs when the elements are selected only from the first and last a elements of the voting multiset, because then $[e_{i,\sigma-g+1} - e_{j,g}] = 3$.

THEOREM 2 : *The optimal convergence rate for any Egocentric voting algorithm is:*

$$C = \frac{1}{\left\lfloor \frac{N-(2a+s)-b-1}{(a+s)} \right\rfloor + 1} \qquad \square$$

Corollary 2.1 : *Let a selection function select only from $\langle v_{a+1}, \ldots, v_{n-a-s} \rangle$. The selection function then yields optimal convergence rate if the following constraints are satisfied:*

$$\gamma = 1 \quad \text{and} \quad \sigma = \left\lfloor \frac{N-(2a+s)-b-1}{a+s} \right\rfloor + 1 \qquad \square$$

6 Comparison to Some Known Algorithms

Several known voting algorithms have been analyzed under both single-mode, where *all* faults are treated as asymmetric, and mixed-mode fault-models [6, 9, 12, 21]. This section applies the same voting algorithms to the Egocentric fault-model.

Fault-Tolerant Midpoint – The Fault-Tolerant Midpoint selects only the two extreme values of its multiset values, i.e. v_1 and v_n. Thus, $\sigma = 2$ and $\gamma = 1$. Also, the selection of v_1 and v_n implies that $k(\sigma) = k(2) = n$ and $k(1) = 1$, so that $e_{i,\sigma} = 3$ and $e_{j,1} = 0$. As a result:

$$\omega_\gamma = \sum_{g=1}^{\gamma} [e_{i,\sigma-g+1} - e_{j,g}] = e_{i,\sigma} - e_{j,1} = 3$$

Since $\sigma < \omega_\gamma$, we have $C > 1$. Hence, the Fault-Tolerant Midpoint is not convergent under the MSE model. But, instead of selecting v_1 and v_n, let us remove the largest and the smallest $(a+s)$ elements and then select the extreme values, i.e. $v_{(a+s+1)}$ and $v_{n-(a+s)}$. Since $k(\sigma) = k(2) = n - (a+s)$, according to (4.2), when $g = 1$, $e_{i,\sigma-g+1} = e_{i,2} = 2$. Similarly, since $k(1) = (a+s+1)$, according to (4.3), when $g = 1$, $e_{j,g} = 1$. Therefore,

$$\omega_\gamma = \sum_{g=1}^{\gamma} [e_{i,\sigma-g+1} - e_{j,g}] = e_{i,\sigma} - e_{j,1} = 1$$

As a result, $C = \omega_\gamma/\sigma = 1/2$. This is the convergence rate of Fault-Tolerant Midpoint previously obtained under Dolev's single-mode and Kieckhafer's MSR fault-models [6, 9].

Fault-Tolerant Mean – This algorithm selects all elements of $\mathbf{V}$. Thus, $\sigma = N - b$. Furthermore, since $\Delta k(g, g+1) = 1$, for $g \leq (\sigma - 1)$, $\gamma = (a + s)$. Thus:

$$C = \frac{\omega_\gamma}{\sigma} = \frac{\sum_{g=1}^{(a+s)} [e_{i,\sigma-g+1} - e_{j,g}]}{N-b} = \frac{\sum_{g=1}^{a} [e_{i,\sigma-g+1} - e_{j,g}] + \sum_{g'=a+1}^{(a+s)} [e_{i,\sigma-g'+1} - e_{j,g'}]}{N-b}$$

According to (4.2) and (4.3), $[e_{i,\sigma-g+1} - e_{j,g}] = 3$ and $[e_{i,\sigma-g'+1} - e_{j,g'}] = 2$. Recall that $t = a + s + b$. Therefore:

$$C = \frac{3a + 2s}{N-b} = \frac{3t - (s + 3b)}{N-b}$$

The single-mode fault model has the convergence rate [6]:

$$C = \frac{t}{N - 2t} \tag{6.1}$$

It is observed that the MSE convergence rate is worse than the single-mode fault-model if benigns are not the dominant mode of failure. However, if the extreme right $(a+s)$ elements and the extreme left a elements are not selected, then $[e_{i,\sigma-g+1} - e_{j,g}] = 1$. Thus:

$$C = \frac{\omega_\gamma}{\sigma} = \frac{\sum_{g=1}^{(a+s)} [e_{i,\sigma-g+1} - e_{j,g}]}{\sigma} = \frac{a+s}{(N-b) - (2a+s)} = \frac{t-b}{(N-2t+b)+s} \tag{6.2}$$

which shows much faster convergence than (6.1). For example, if $a = 1$, $s = 1$, $b = 2$, and $N = 13$, the convergence is more than three times faster than the rate predicted by the single-mode fault model.

The Fault-Tolerant Mean under the MSR fault model discards the extreme $(a+s)$ elements from both sides of the voting multiset. It has the convergence rate [9]:

$$C = \frac{a}{N - 2t + b} = \frac{t - b - s}{N - 2t + b} \tag{6.3}$$

By comparing (6.2) and (6.3), unless $s = 0$, MSR fault-model has better convergence rate.

Single-Mode Optimal – The optimal single-mode algorithm selects the first element of the voting multiset, and every t^{th} element thereafter. It has the convergence rate [6]:

$$C = \frac{1}{\left\lfloor \frac{N-2t-1}{t} \right\rfloor + 1}$$

Under MSE, $\Delta k(g, g+1) = t$. Thus, $\gamma_t = 1$. In addition, since the first and every t^{th} elements are selected, $e_{i,\sigma} = 3$ and $e_{j,1} = 0$. Thus:

$$\omega_{\gamma_t} = \sum_{g=1}^{\gamma_t} [e_{i,\sigma-g+1} - e_{j,g}] = e_{i,\sigma} - e_{j,1} = 3$$

Therefore,

$$C = \frac{\omega_{\gamma_t}}{\sigma} = \frac{3}{\left\lfloor \frac{N-b-1}{t} \right\rfloor + 1}$$

which shows much slower convergence than the optimal single-mode algorithm. However, if the extreme $(t-b)$ elements are not used in the selection process, then $\gamma_t = 1$, $e_{i,\sigma} = 2$ and $e_{j,1} = 1$. Thus:

$$C = \frac{\omega_{\gamma_t}}{\sigma} = \frac{1}{\left\lfloor \frac{n-2(t-b)-1}{t} \right\rfloor + 1} = \frac{1}{\left\lfloor \frac{N-2t+b-1}{t} \right\rfloor + 1}$$

which shows much faster convergence. This is also the same convergence rate predicted under the MSR fault-model [9].

Mixed-Mode Optimal – The optimal value of $C = \omega_\gamma/\sigma$ requires that ω_γ and σ be minimized and maximized respectively. According to Corollary 2.1, satisfying the constraint on σ when the selected elements are from within $\langle v_{a+1}, \ldots, v_{n-a-s} \rangle$ guarantees the existence of a selection function with $\gamma = 1$. This value of γ is obtained when the first element and every $(a+s)^{\text{th}}$ element thereafter are selected. The number of elements selected in this manner is consistent with the value of σ in the Corollary.

The selection of the first element gives $e_{j,1} = 1$. The selection of every $(a+s)^{\text{th}}$ element implies that $e_{i,\sigma} = 2$, because the σ^{th} element selected is within the last $(a+s)$ elements of $\langle v_{a+1}, \ldots, v_{n-a-s} \rangle$. As a result:

$$C = \frac{\omega_\gamma}{\sigma} = \frac{\sum_{g=1}^{\gamma} [e_{i,\sigma-g+1} - e_{j,g}]}{\sigma} = \frac{e_{i,\sigma} - e_{j,1}}{\left\lfloor \frac{N-(2a+s)-b-1}{a+s} \right\rfloor + 1} = \frac{1}{\left\lfloor \frac{N-2t+s+b-1}{a+s} \right\rfloor + 1}$$

The optimal algorithm under MSR has the following convergence [9]:

$$C = \frac{1}{\left\lfloor \frac{N-2t+b-1}{a} \right\rfloor + 1}$$

Therefore, MSE convergence rate is lower than MSR but close to it. Doing a similar analysis, however, it can be easily shown that the MSE convergence rate becomes three time worse than MSR algorithms when the extreme elements of $\mathbf{V}$ are included in the selection function.

Table 1 shows the convergence rates under different fault-models and voting algorithms. Depending on the span of the selected elements, three forms of MSE algorithms, differentiated by numeral subscripts, are considered. In the fourth column, the label $N - b$ indicates that MSE_1 algorithms select elements which expand over the entire voting multiset. In the fifth column, $N - 2(a + s) - b$ indicates that MSE_2 algorithms do not select elements from the extreme $(a+s)$ elements of the voting multiset. Similarly, $N-(2a+s)-b$ in the last column denotes that the rightmost $(a + s)$ and the leftmost a elements are not considered in the selection process. The following are observed from this table:

- The MSE voting algorithms which select elements from either extremes of the voting multiset are either non-convergent or their convergence rates are worse than that of the traditionally known voting algorithms. This is because, as Lemma 1 shows, in the worst case, the symmetric and asymmetric errors are at the extreme ends of the voting multiset. The traditional voting algorithms [6, 9, 12], before selecting elements, use a "reduction" function to discard the erroneous values from both ends of the voting multiset. This ensures the remaining errors after the reduction to be within the range of the correct values. Hence, $(a + s) + 1 \leq k(g) \leq n - (a + s)$, which implies that $[e_{i,\sigma-g+1} - e_{j,g}] = 1$. As a result, $C = \omega_\gamma/\sigma = \gamma/\sigma$. Whereas, when MSE algorithms select elements from both ends, $[e_{i,\sigma-g+1} - e_{j,g}] \geq 2$, which indicates that $\gamma < \omega_\gamma \leq 3\gamma$. Hence, $\gamma/\sigma < C \leq 3\gamma/\sigma$.

- The convergence rate under MSE_3 is better than MSE_1 and MSE_2. When comparing MSE_3 to MSE_2, the increase in convergence is due to the fact that MSE_3 has s more elements to select from. When MSE_1 is compared to MSE_3, MSE_1 has $(2a + s)$ more elements to select from, which suggests that convergence ought to be better. But MSE_1 selects from the extreme ends which worsens its convergence rate severely.

- Under any given selection function that can be applied to *both* MSE and MSR fault-models, MSR reveals either the same or better convergence rate than MSE (except the Optimal Single-Mode algorithm under MSE_3). In MSR, symmetric faults are treated as symmetric. Whereas, in MSE, symmetric faults, as described in Subsection 4.1, are treated like asymmetric faults. Consequently, MSR and MSE algorithms obtain γ with

respect to a and $(a+s)$ respectively. When the subscript of γ increases, i.e. $(a+s)$ versus a, γ might increase, but it never decreases. This is because more elements need to be skipped to ensure that $\Delta k(g, g+\gamma_{(a+s)}) \geq (a+s)$. Hence, the numerator of C might increase, which is an implication of worse convergence. Therefore, as the table shows, MSR and MSE_3 have the same convergence rate when $s = 0$.

7 Summary and Future Research

This paper has examined the problem of reaching Approximate Agreement for a new family of convergent voting algorithms (MSE) that belongs to the Egocentric category of algorithms. Traditionally, the study of these algorithms has been presented with ad-hoc proofs of their fault-tolerance and convergence rates. The analysis herein revealed simple expressions that can be used to easily determine the fault-tolerance and the convergence rate of any MSE algorithm. By knowing the facts that σ and γ, and in turn ω_γ, can be determined easily, the system designer can quickly devise a new MSE algorithm customized to a specific application.

Traditionally, Egocentric algorithms used the entire voting multiset, as in Fault-Tolerant Mean, to reach a single voted value. It was not known how the distribution of selected elements would affect the convergence rate. Here, it is shown that convergence is improved significantly if no elements of the largest $(a+s)$ and the smallest a data items are included in Sel_σ. In addition, it was not previously known how Egocentric algorithms would behave in a hybrid fault environment, or how they would perform against other known voting algorithms.

There are some directions to expand upon this work. This paper has addressed only Egocentric algorithms. Due to the egophobism during the sampling phase, it is not clear how Egophobic algorithms would perform under different constraints, but it appears that the same methodology can be applied to those algorithms.

Another avenue of study is that MSE along with other traditional voting algorithms [6, 9, 12, 19] can not exploit the presence of omission faults. As a result omission faults must be transformed to a more severe fault mode such as asymmetric. The exploitation of omission faults, although is inherently more complex, has shown significantly higher fault-tolerance,

and reducing the need to globally diagnose the benign faults [2]. It is conjectured that the same methodology used in [2] can be extrapolated to MSE.

References

[1] A. Avizienis, "The N-Version Approach to Fault-Tolerant Software", *IEEE Trans. Software Eng.*, Vol SE-11, No 12, 1985 Dec, pp 1491-1501.

[2] M.H. Azadmanesh, R.M. Kieckhafer, "New Hybrid Fault Models for Asynchronous Approximate Agreement", *IEEE Trans. Computers*, Vol 45, 1996 Apr, pp 439-449.

[3] M.H. Azadmanesh, A.W. Krings, "Performance Analysis of Approximate Voting Algorithms in the Absence of Reduction-Function", Technical Report, No TR-97-02, 1997; University of Nebraska at Omaha, Dept. Comp. Sci.

[4] L. Chen, A. Avizienis, "N-version Programming: A Fault Tolerant Approach to Reliability of Software Operation", *Proc. Int'l Symp. on Fault Tolerant Computing*, 1978, pp 3-9.

[5] D. Davies, J. Wakerly, "Synchronization and Matching in Redundant Systems," *IEEE Trans. Computers*, Vol C-27, 1978 Jun, pp. 531-539.

[6] D. Dolev, et al, "Reaching Approximate Agreement in the Presence of Faults," *JACM*, Vol 33, 1986 Jul, pp 499-516.

[7] D.E. Eckhardt, et al, "An Experimental Evaluation of Software Redundancy as a Strategy for Improving Reliability, *IEEE Trans. Software Eng.*, Vol 17, 1991 Jul, pp 692-702.

[8] J.L.W. Kessels, "Two Designs of a Fault-Tolerant Clocking System", *IEEE Trans. Computers*, Vol C-33, 1984 Oct, pp 912-919.

[9] R.M. Kieckhafer, M.H. Azadmanesh, "Reaching Approximate Agreement With Mixed Mode Faults", *IEEE Trans. Parallel Distributed Systems*, Vol 5, 1994 Jan, pp 53-63.

[10] C.M. Krishna, K.G. Shin, R.W. Butler, "Ensuring Fault Tolerance of Phase-Locked Clocks", *IEEE Trans. Computers*, Vol C-34, 1985 Aug, pp 752-756.

[11] C.M. Krishna, "Fault-Tolerant Synchronization Using Phase-Locked Clocks", *Microelectron. Reliab.*, Vol 30, No 2, 1990, pp 275-287.

[12] L. Lamport, P.M. Melliar-Smith, "Synchronizing Clocks in the Presence of Faults", *JACM*, Vol 32, 1985 Jan, pp 52-78.

[13] P. Lincoln, J. Rushby, "A Formally Verified Algorithm for Interactive Consistency Under a Hybrid Fault Model", *Proc. Fault-Tolerant Computing Symp.*, 1993 Jun, pp 402-411.

[14] P. Lincoln, J. Rushby, "Byzantine Agreement with Authentication Observation and Applications in Tolerating Hybrid and Link Faults", *Dependable Computing for Critical Applications-5*, 1995 Sep, pp 79-90.

[15] M.R. Lyu, A. Avizienis, "Assuring Design Diversity in N-Version Software: A Design Paradigm for N-Version Programming", *Dependable Computing and Fault-Tolerant System*, 1992, pp 197-218.

[16] P. Ramanathan, D.D. Kandlur, K.G. Shin, "Hardware-Assisted Software Clock Synchronization for Homogeneous Distributed Systems", *IEEE Trans. Computers*, Vol C-39, 1990 Apr, pp 514-524.

[17] P. Ramanathan, K.G. Shin, R.W. Butler, "Fault-Tolerant Clock Synchronization in Distributed Systems", *IEEE Computer*, Vol 23, 1990 Oct, pp 33-42.

[18] F.B. Schneider, *Understanding Protocols for Byzantine Clock Synchronization*, Technical Report, No 87–859, 1987 Aug; Cornell University, Dept. Comp. Sci.

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

[20] P.M. Thambidurai, et al, "Clock Synchronization in MAFT", *Proc. 19^{th} Fault-Tolerant Computing Symp.*, 1989 Jun, pp 142-151.

[21] N. Vasanthavada, P. Thambidurai, "Design of Fault-Tolerant Clocks with Realistic Assumptions", *Proc. 18^{th} Fault-Tolerant Computing Symp.*, 1989 Jun, pp 128-133.

	Fault-Model				
Voting Algorithm	Byzantine	MSR	MSE_1 $N-b$	MSE_2 $N-2(a+s)-b$	MSE_3 $N-(2a+s)-b$
Mid-Point	$\frac{1}{2}$	$\frac{1}{2}$	NC	$\frac{1}{2}$	$\frac{1}{2}$
FT-Mean	$\frac{t}{N-2t}$	$\frac{a}{N-2t+b}$	$\frac{3a+2s}{N-b}$	$\frac{t-b}{N-2t+b}$	$\frac{t-b}{N-2t+s+b}$
Single-Mode Opt.	$\frac{1}{\left\lfloor \frac{N-2t-1}{t} \right\rfloor + 1}$	$\frac{1}{\left\lfloor \frac{N-2t+b-1}{t} \right\rfloor + 1}$	$\frac{3}{\left\lfloor \frac{N-b-1}{t} \right\rfloor + 1}$	$\frac{1}{\left\lfloor \frac{N-2t+b-1}{t} \right\rfloor + 1}$	$\frac{1}{\left\lfloor \frac{N-2t+s+b-1}{t} \right\rfloor + 1}$
Mixed-Mode Opt.	$\frac{1}{\left\lfloor \frac{N-2t-1}{t} \right\rfloor + 1}$	$\frac{1}{\left\lfloor \frac{N-2t+b-1}{a} \right\rfloor + 1}$	$\frac{3}{\left\lfloor \frac{N-b-1}{t-b} \right\rfloor + 1}$	$\frac{1}{\left\lfloor \frac{N-2t+b-1}{t-b} \right\rfloor + 1}$	$\frac{1}{\left\lfloor \frac{N-2t+s+b-1}{t-b} \right\rfloor + 1}$
Mixed-Mode	$\frac{\gamma_t}{\sigma}$	$\frac{\gamma_a}{\sigma}$	$\frac{\omega_{\gamma_{(a+s)}}}{\sigma}$	$\frac{\omega_{\gamma_{(a+s)}}}{\sigma}$	$\frac{\omega_{\gamma_{(a+s)}}}{\sigma}$

Table 1: Comparison of Convergence Rates. (NC = Not Convergent)

Biographies of Authors

Dr. M.H. Azadmanesh, Assistant Professor; Department of Computer Science, University of Nebraska; Omaha, Nebraska 68182–0500 USA. Internet (e-mail): azad@ahvaz.unomaha.edu

M.H. Azadmanesh (Azad) received a BS (1976) in Cost Accounting, an MS (1982) and a PhD in Computer Science from Iowa State University and University of Nebraska-Lincoln respectively. He is an Assistant Professor in the Department of Computer Science at the University of Nebraska at Omaha. His research interests include Fault-Tolerant Systems, Distributed Agreement, Reliability Modeling, Real Time Systems, and Network Communication. Azad is a member of the ACM and the IEEE Computer and Reliability societies.

Dr. Axel W. Krings, Assistant Professor; Department of Computer Science, University of Idaho; Moscow, Idaho 83844-1010 USA. Internet (e-mail): krings@cs.uidaho.edu.

Axel W. Krings received the Dipl.Ing. in Electrical Engineering from the FH-Aachen, Germany, in 1982, and his MS and PhD degrees in Computer Science from the University of Nebraska - Lincoln, in 1991 and 1993, respectively. He is an Assistant Professor of Computer Science and Computer Engineering at the University of Idaho and a member of the Microelectronics Research and Communications Institute (MRC). Previous appointments include the Technical University of Clausthal, Germany. His research interests include: Fault-Tolerant Systems, Distributed Systems, Computer Networks, and Real-Time Scheduling. Dr. Krings is a member of the IEEE Computer Society.