A New Analysis of Hybrid Faults for Synchronous Convergent Voting Algorithms in Large Distributed Systems

M.H. Azadmanesh† and A.W. Krings‡

† Department of Computer Science and Center for Management of Information Technology, University of Nebraska at Omaha, Omaha, NE 68182–0459 USA.
‡ Department of Computer Science, University of Idaho, Moscow, ID 83844-1010 USA.

Abstract

Mean-Subsequence-Reduced (MSR) algorithms and other voting algorithms, based on Thambidurai and Park's fault-model, use a predefined value for values not received. Thus, they can not exploit omissive faults. This paper addresses a new fault-model, *Omission-MSR* (OMSR). Under OMSR, two new families of voting algorithms will be introduced, which are more fault-tolerant than existing voting algorithms. The results of MSR and OMSR voting algorithms will be transformed into a singular set of relations so that their convergence rate and fault tolerance can be compared. These relations provide the means to easily determine the convergent properties of different voting algorithms.

1 Introduction

The objective of reaching *Approximate Agreement* is to guarantee that the voted value for each non-faulty process is within the range of the initial correct values and that the diameter of the voted values for each of the non-faulty processes is within a prespecified small positive real value ϵ. The final voted value is obtained by employing multiple rounds of message exchange. In each round, each process sends its value to all processes. On receipt of a collection of values, each process executes a voting function F, to obtain its latest voted value, which it then broadcasts in the next round of message exchange. The objective is achieved if differences in values held by the non-faulty processes is reduced in each round [3, 6, 8]. This property, called single-step convergence, guarantees that the range of values will eventually be less than ϵ, given enough rounds.

Recent research [4] used the hybrid fault model of Thambidurai and Park [7] which partitions the total number of faults into a asymmetric, s symmetric, and b benign faults. A benign fault is a fault which is self-evident to *all* nodes. A symmetric fault is one whose value is perceived identically by all receiving non-faulty processes, whereas an asymmetric fault is capable of sending conflicting messages to different non-faulty processes. Using this fault model, a broad family of convergent voting algorithms was derived called *Mean-Subsequence-Reduced* (MSR) algorithms [4]. However, MSR and other existing voting algorithms are restrictive in that they *can not exploit omissive faults*. This paper exploits omissive faults which could be the dominant mode of failure in large distributed systems. The analysis of voting algorithms is based on *synchronous* [3] systems where the processing and the communication delays of non-faulty processes are bounded.

2 Background

Let $\mathbf{V}_i$ and $\mathbf{V}_j$ be multisets of values received by two non-faulty processes i and j respectively. Let $\mathbf{U}_{i\cup j}$ be the multiset of *correct* values received by the same processes, and $\rho(\mathbf{U}_{i\cup j})$ be the maximal interval of the correct values. Then for a completely connected system, a voting algorithm is single-step convergent if both of the following conditions are true for every voting round: 1) for each non-faulty process i, the voted value $F(\mathbf{V}_i) \in \rho(\mathbf{U}_{i\cup j})$, 2) for each pair of non-faulty processes (i, j), $|F(\mathbf{V}_i) - F(\mathbf{V}_j)| \leq C\,\delta(\mathbf{U}_{i\cup j})$, where $0 \leq C < 1$.

There are two types of convergence to be considered: local convergence over a specified subgraph, and global convergence over the entire system graph. For completely connected systems, local convergence implies global convergence. For partially connected systems, only local convergence is addressed. Several constraints are placed on these systems: 1) the system topology is a non-hierarchical, *d-regular, homogeneous*, undirected graph of N processing nodes, 2) messages may not be relayed between processes; thus, each non-faulty process i receives *only* the values of its immediate neighbors (including itself), 3) $N \gg d$ so that "wrap-around" effects can not assist the convergence process. Partially connected systems also use the following parameters: $\mathbf{P}_i$ is the set of processes adjacent to process i, $\mathbf{P}_{i\cap j}$ is the set of processes adjacent to both processes i and j, χ is the number of processes whose values are receivable by either i or j but *not* by both, f is the maximum number of faulty processes in either $\mathbf{P}_i \backslash \mathbf{P}_{i\cap j}$ or $\mathbf{P}_j \backslash \mathbf{P}_{i\cap j}$, and $\mathbf{U}_{i\cap j}$ is the multiset of correct values generated in $\mathbf{P}_{i\cap j}$.

In a partially connected system, $\mathbf{U}_{i\cap j} \subset \mathbf{U}_{i\cup j}$, yielding two distinct criteria for local convergence: Intersection and Union. The conditions for Intersection and Union convergence are the same as those of completely connected systems (above) except that Intersection convergence uses $\mathbf{U}_{i\cap j}$ rather than $\mathbf{U}_{i\cup j}$ [1, 5].

2.1 MSR Voting Algorithms

An MSR voting algorithm has the form [3, 4]: $F(\mathbf{V}_i) = \text{mean}\,[Sel_{\sigma_i}(Red^{\tau}(\mathbf{V}_i))]$. The "Reduction" function Red^{τ} removes the τ largest and τ smallest elements from the multiset $\mathbf{V}_i = \langle v_1 \ldots v_{V_i}\rangle$, where $V_i = |\mathbf{V}_i|$. The resulting submultiset is called *medial multiset* $\mathbf{M}_i = \langle m_1 \ldots m_{M_i}\rangle$, where $M_i = |\mathbf{M}_i|$. The "Selection" function Sel_{σ_i} then selects a subsequence $\mathbf{S}_i$ of σ_i elements from $\mathbf{M}_i$. The final voted value is the arithmetic mean of the selected multiset.

MSR algorithms assume that the multisets $\mathbf{V}$ are of the same size. Let this size be $V_\circ$. Then to ensure that $V_\circ = V_i = V_j$, if less than $V_\circ$ values are received, an arbitrary default value is chosen for each value not received. Consequently, medial multisets and σ for non-faulty processes will be of the same size as well.

Connectivity	$\tau_\circ$	α	β
Complete	$a + s$	a	$V_\circ + b$
Union	$a + s + f$	$\chi + a$	$V_\circ - \chi$
Intersection	$a + s + \chi$	$\chi + a$	$V_\circ - \chi$

Conditions for Convergence
$\tau \geq \tau_\circ$ $V_\circ \geq 2\tau + \max(\alpha + 1, \sigma)$ $\lvert\mathbf{P}_{i \cap j}\rvert \geq \beta$

Table 1: Summary of MSR Convergence Parameters

2.1.1 Fault-Tolerance

Benign faults can be ignored in completely connected systems, because *all* processes can delete the benign errors from their voting multisets [4], so that $V_\circ = V_i - b$. But, in partially connected systems without message relays, *no fault is self-evident to all non-faulty processes.* Thus, in partially connected systems, only symmetric and asymmetric faults are considered.

Malicious errors play an important role in determining the convergence rate and fault tolerance expressions. However, it is the *effective* number of asymmetric values, which includes both erroneous and non-erroneous values, that is actually the basis for these expressions. Let α represent the effective number of asymmetric values between two arbitrary non-faulty processes i and j. Let β represent the *fault burden* which is the minimum number of processes in $\mathbf{P}_{i \cap j}$ to guarantee convergence, and let $\tau_\circ$ be the *effective* number of malicious errors in a voting multiset $\mathbf{V}_i$, i.e. the number of malicious processes in $\mathbf{P}_{i \cap j}$ plus the number of processes in $\mathbf{P}_i \backslash \mathbf{P}_{i \cap j}$ that must be treated as malicious. Table 1 shows the conditions under which an MSR algorithm is convergent [4, 5].

2.1.2 Convergence Rate

Let g be an index into the selected multiset $\mathbf{S}$. Then, for each $g \in \{1, \ldots, \sigma\}$ there exists exactly one $k(g) \in \{1, \ldots, M\}$ which guarantees that $s_g = m_{k(g)}$ *for all possible* $\mathbf{M}$. Given the multisets $\mathbf{M}_i$ and $\mathbf{M}_j$, let $\mathbf{M}_\ell$ be any multiset of size $\min(M_i, M_j)$. Then, given subsequence $\mathbf{S}_\ell = Sel_{\sigma_\ell}(\mathbf{M}_\ell)$ and two indices g,$h \in \{1, \ldots, \sigma_\ell\}$ such that $g \leq h$, we define $\Delta k(g, h)$ as the number of elements in the submultiset $\langle m_{k(g)+1}, \ldots, m_{k(h)} \rangle$ of $\mathbf{M}_\ell$ spanned by elements $\langle s_g, \ldots, s_h \rangle$ in $\mathbf{S}_\ell$. So, $\Delta k(g, h) = k(h) - k(g)$. Then, for a given non-negative integer $\alpha < M_\ell$ and for each $g \leq \sigma_\ell$, define $\Delta(g)$ to be the minimum value of the difference $(h - g)$ such that $\Delta k(g, h) \geq \alpha$. Finally, define $\gamma_\alpha = \max(\Delta(g))$. It has been shown [4] that the convergence rate C of any synchronous MSR algorithm is $C_{MSR} = \gamma_\alpha / \sigma$.

3 Omission-MSR (OMSR) Model

Classes of symmetric and asymmetric faults are partitioned into *transmissive* and *omissive* modes. A transmissive fault delivers erroneous value(s) to one or more receiving processes, whereas an omissive fault fails to deliver any value to one or more receiving processes. The total number of faulty processes in the system is: $(\omega_{a1} + a') + (\omega_s + s') + b$, where: ω_{a1} is the number of *strictly omissive asymmetric* processes in $\mathbf{P}_{i \cap j}$ which send a single "correct" value to some processes and no value to other processes, a' is the number of transmissive asymmetric faults other than strictly omissive in $\mathbf{P}_{i \cap j}$, ω_s is the number of omissive symmetric processes in $\mathbf{P}_{i \cap j}$ whose values are not received by *any* non-faulty process, s' is the number of transmissive symmetric faulty processes in $\mathbf{P}_{i \cap j}$, and b is the number of benign faults in the system. The relationships between this new fault-model and the Thambidurai and Park's fault-model are specified by the relations: $a = (\omega_{a1} + a')$ and $s = (\omega_s + s')$.

OMSR algorithms discard omissive errors or self-evident transmissive errors. No defaults are substituted into $\mathbf{V}$. Thus, omissive errors remain omissive, and self-evident errors become omissive. As a result, in contrast to all previous studies in Approximate Agreement, $|\mathbf{V}_i| \neq |\mathbf{V}_j|$ is possible.

The OMSR model employs two subfamilies of *dynamic* and *fixed* selection functions. In a dynamic selection function, the number of selected elements depends on the size of $\mathbf{M}$, whereas the number of selected elements in a fixed selection function is fixed.

3.1 Dynamic-σ Selection Functions

Let $V_\circ = 2\tau + M_{max}$, where $\mathbf{M}_{max} = \langle m_1, \ldots, m_{M_{max}} \rangle$ is the medial multiset with the largest number of elements in the system. Also, define the enumerative selection set as a set of integers $E = \{e_1, \ldots, e_{\sigma_{max}}\}$, where $e_j < e_{j+1}\ \forall\ j \in \{1, \ldots, \sigma_{max} - 1\}$. σ_{max} is the number of elements selected from $\mathbf{M}_{max}$, such that $\mathbf{S}_{max} = Sel_{\sigma_{max}}(\mathbf{M}_{max}) = \langle m_{e_1}, m_{e_2}, \ldots, m_{e_{\sigma_{max}}} \rangle$. Then, let $\mathbf{S}_i$ be $Sel_{\sigma_i}(\mathbf{M}_i) = \langle m_{e_1}, \ldots, m_{e_{\sigma_i}} \rangle$, where σ_i is the largest value such that $e_{\sigma_i} \leq |\mathbf{M}_i|$. Thus, the number of elements selected depends on the size of $\mathbf{M}_i$.

3.2 Fixed-σ Selection Functions

A selection function is a mapping from the set $\{1, \ldots, \sigma\}$ to the codomain set $\{k(1), k(2), \ldots, k(\sigma)\}$. A selection function is non-enumerative if at least one of its codomain set values is a function of M. We adopt the name *fixed-σ* for one sub-family of non-enumerative selection functions, where $\sigma_i = \sigma_j$, regardless of the sizes of $\mathbf{M}_i$ and $\mathbf{M}_j$. However, unlike dynamic-σ selection functions, for two arbitrary processes i and j, the value $k(g)$, $g \in \{1, \ldots, \sigma\}$, for process i may not be the same as that for process j. Thus, to distinguish between $k(g)$ of different processes, define $k_i(g)$ the same as $k(g)$ except that $k(g)$ is associated with $\mathbf{M}_i$, and define $\Delta k_i(g, h) = k_i(h) - k_i(g)$. Other than $\sigma_i = \sigma_j$, fixed-σ selection functions have the properties that, if $M_i \geq M_j$, then: 1) $k_i(g) \geq k_j(g)$, where $g \leq \sigma$, and 2) $\Delta k_i(g, g+1) \geq \Delta k_j(g, g+1)$, where $g < \sigma$.

3.3 Convergence with OMSR Algorithms

In OMSR, the number of selected elements for two processes i and j may differ. In comparison to MSR, this affects the convergence rate by $\mu = (\sigma_{max} - \sigma_{min})/(\sigma_{max})$ [1, 2]. This fraction does not exist in MSR algorithms because $\sigma_{max} = \sigma_{min}$ and so $\mu = 0$. The convergence rate of an OMSR voting algorithm is: $C_{OMSR} = \mu + \gamma_\alpha/\sigma_{max}$. Table 2 shows the conditions under which OMSR algorithms are convergent [1, 2]. Note that, similar to MSR algorithms, Fixed-σ voting algorithms require equality in the number of selected elements. Thus, $\sigma = \sigma_{max} = \sigma_{min}$. Therefore, $\mu = 0$.

Connectivity	$\tau_\circ$	α	β	Conditions for Convergence
Complete	$a' + s'$	$a' + \omega_{a1}$	$V_\circ + \omega_s + b$	$\tau \geq \tau_\circ$
Union	$a' + s' + f$	$\chi + a' + \omega_{a1}$	$V_\circ + \omega_s - \chi$	$V_\circ \geq 2\tau + \max(\alpha + 1, \sigma_{max})$
Intersection	$a' + s' + \chi$	$\chi + a' + \omega_{a1}$	$V_\circ + \omega_s - \chi$	$\lvert \mathbf{P}_{i \cap j} \rvert \geq \beta$

Table 2: Summary of OMSR Convergence Parameters

4 Comparison with MSR Voting Algorithms

4.1 Convergence Rate

Since $a = (\omega_{a1} + a')$, values of α for both MSR and OMSR are the same. Also, in MSR, $\sigma = \sigma_{min} = \sigma_{max}$. Thus, C_{MSR} can be written as: $C_{MSR} = \gamma_\alpha / \sigma_{max}$. Since $C_{OMSR} \geq C_{MSR}$, the OMSR rate of convergence is worse than that of MSR. To improve C_{OMSR}, the following options can be employed: 1) use fixed-σ voting algorithms, or 2) use dynamic-σ but ensure that $e_i \leq M_{min}\ \forall\ e_i \in E$. In either case, since $\sigma_{max} = \sigma_{min}$, $\mu = 0$, which implies that $C_{OMSR} = C_{MSR}$.

4.2 Fault-Tolerance

In Table 2, to determine the minimum value of $V_\circ$ for which a convergent OMSR voting algorithm exists, consider the lower bounds on σ_{max} when $\alpha = 0$ and $\alpha > 0$. If the number of effective asymmetric values α is zero, then $\gamma_\alpha = 0$. To guarantee convergence, $\mathbf{S}_{max}$ must be non-empty, so any $\sigma_{max} \geq 1$ will suffice. If $\alpha > 0$, then σ_{max} must be at least two, to ensure that γ_α exists. In both cases the minimum value of $\max(\alpha + 1, \sigma_{max})$ is $(\alpha + 1)$. Thus, a convergent OMSR voting algorithm exists if: $V_\circ \geq 2\tau + \alpha + 1$. Using the same discussion, MSR voting algorithms have the same lower bound on $V_\circ$.

When an OMSR algorithm receives either an omissive or a self-evident error, it simply discards that value. The advantage of this approach can be illustrated by comparing the fault burden of MSR under completely connected systems with that of OMSR.

Under MSR, using Table 1 and relation $V_\circ \geq 2\tau + \alpha + 1$, the fault burden is: $\lvert \mathbf{P}_{i\cap j} \rvert \geq 2\tau + \alpha + b + 1 = (3a' + 2s' + b + 1) + 3\omega_{a1} + 2\omega_s$. However, under OMSR, using Table 2 and the same relation, the fault burden is: $\lvert \mathbf{P}_{i\cap j} \rvert \geq 2\tau + \alpha + \omega_s + b + 1 = (3a' + 2s' + b + 1) + \omega_{a1} + \omega_s$. Consequently, the value of $\lvert \mathbf{P}_{i\cap j} \rvert$ predicted under OMSR is $(2\omega_{a1} + \omega_s)$ processes less than that for MSR. By a similar observation, partially connected systems under OMSR show the same improvement. This result implies that for the same number of processes, OMSR voting algorithms are more fault-tolerant. The reason is that under MSR omissions are either presumed not to occur, or the voting algorithm employs a default value for those values not received. The later has the effect of transforming the omissive behavior into a more severe fault mode such as asymmetric or symmetric. In OMSR, however, a process, upon the detection of an error, simply drops the erroneous value from its voting multiset **V** and does not use it in the voting process. So, omissive faults are treated as omissive.

5 Summary and Conclusions

The OMSR fault-model has addressed five types of fault modes: strictly omissive asymmetric, transmissive asymmetric, transmissive symmetric, omissive symmetric, and benign. The model is complete in that it can degenerate to more restrictive fault models, by simply setting the appropriate parameters (ω_{a1}, ω_s, a, s, b) to zero.

In previous studies, a process can not simply ignore a "missing" value or a self-evident error unless the error is globally diagnosed by all other processes. This and the facts that Byzantine agreement algorithms are time consuming and require reliable network communication, may make the benign fault mode an impractical concept. By having the capability of locally ignoring erroneous data items, OMSR algorithms do not need to globally diagnose benign errors. In contrary to MSR, 1) if the error is detected by every process, the effect will be the same as a globally diagnosed benign error, and 2) if the error is detected by just a subset of processes, then diagnosis at the global level will not be needed.

The distinctive feature of OMSR is the ability to deal with different size multisets, thus exploiting omissive faults. In large point-to-point distributed systems, the majority of malicious faults could be omissive. Thus, they could be the dominant mode of failure. As a result, the amount of improvement shown can be very significant in designing fault tolerant systems.

References

[1] M.H. Azadmanesh, R.M. Kieckhafer, "A New Hybrid Fault Model for Omissive Faults", Rep. No. 95-1, Comp. Sc. Dept., University of Nebraska at Omaha, 1995.

[2] M.H. Azadmanesh, R.M. Kieckhafer, "Hybrid Fault Mode Analyses in Partially Connected Networks", Rep. No. 95-2, Comp. Sc. Dept., University of Nebraska at Omaha, 1995.

[3] D. Dolev, et al, "Reaching Approximate Agreement in the Presence of Faults," *JACM*, 33, 3, 499-516, Jul 1986.

[4] R.M. Kieckhafer, M.H. Azadmanesh, "Reaching Approximate Agreement With Mixed Mode Faults", *IEEE Trans. on Parallel Dist. Syst.*, 5, 1, 53-63, Jan 1994.

[5] R.M. Kieckhafer, M.H. Azadmanesh, "Low Cost Approximate Agreement in Partially Connected Networks", *J. of Computing and Information*, 3, 2, 53-85, 1993.

[6] L. Lamport, P.M. Melliar-Smith, "Synchronizing Clocks in the Presence of Faults", *JACM*, 32, 1, 52-78, Jan 1985.

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

[8] N. Vasanthavada, P.N., Marinos, "Synchronization of Fault-Tolerant Clocks in the Presence of Malicious Failures", *IEEE Trans. Comput.*, C-37, 4, 440-448, Apr 1988.