Low Cost Approximate Agreement
In Partially Connected Networks

R.M. KIECKHAFER and M.H. AZADMANESH
Department of Computer Science and Engineering
115 Ferguson Hall
University of Nebraska – Lincoln
Lincoln, NE 68588–0115

SUMMARY: In fault-tolerant multiprocessor systems, different non-faulty processes may arrive at different values for a given system parameter. To resolve this disagreement, processes must exchange and vote upon their respective local values. During voting, faulty processes may attempt to inhibit agreement by acting in a malicious or “Byzantine” manner. Approximate Agreement defines one form of agreement in which the voted values obtained by the non-faulty processes need not be identical. Rather, they need only agree to within a predefined tolerance. Approximate Agreement can be achieved by a sequence of convergent voting rounds, in which the range of values held by non-faulty processes is reduced in each round. Existing convergent voting algorithms assume complete connectivity between processes. Where the physical connectivity is incomplete, messages are relayed between processors to simulate complete connectivity. For large, sparsely connected systems, the message traffic associated with message relay could be prohibitive.

This paper addresses convergent voting in partially connected systems where message relay is prohibited. Simple expressions are developed for the convergence rates and robustness of a broad family of “locally” convergent voting algorithms, in the presence of two distinct fault modes. These expressions are then employed to determine the robustness of local convergence in some commonly used partially connected networks. Finally, issues affecting “global” convergence are addressed and a methodology for analyzing global convergence properties is given.

Keywords—Approximate Agreement, Byzantine Agreement, Clock Synchronization, Convergent Voting, Fault-Tolerant Multiprocessors, Mixed-Mode Faults.
1 INTRODUCTION

An important problem in the design of fault-tolerant distributed systems is ensuring that all non-faulty processes agree on the values of critical data items in the presence of faulty processes. This problem arises whenever non-faulty processes can legitimately form differing “opinions” regarding a specific value. They must then exchange and vote upon their local values to arrive at a single consensus value. If a faulty process is constrained to send the same erroneous value to all non-faulty processes, then simple majority voting is sufficient to provide immediate agreement. It is only necessary that the majority of the processes be non-faulty. The distributed agreement problem becomes significantly more complex if a faulty process is permitted to send conflicting values to different non-faulty processes. A faulty process with this property has been called malicious, two-faced, Byzantine, or asymmetric.

The classic form of distributed agreement, called Byzantine Agreement, demands that all non-faulty processes obtain identical voted values for any set of initial values. However, many applications do not require non-faulty processes to achieve exact agreement. Rather, they need only agree on a value to within a specified tolerance. This condition, called Approximate Agreement, is useful in areas such as sensor data management and fault-tolerant clock synchronization [Kie88, Lam85, Lun84, Sch87, Tha89]. Given an arbitrarily small non-negative real value $\epsilon$, Approximate Agreement is defined by two conditions [Dol83, Dol86]:

[A1] Agreement – The voting algorithms executed by all non-faulty processes eventually halt with voted values that are within $\epsilon$ of each other.

[A2] Validity – The voted value held by each non-faulty process is within the range of the initial values held by all non-faulty processes.

Several Approximate Agreement algorithms employ multiple rounds of message exchange and convergent voting algorithms which guarantee that the range of values held by the non-faulty processes is reduced in each round [Dol83, Dol86, Kie91, ...]
This property, called single-step convergence, guarantees that the range of values will eventually be less than $\epsilon$, given enough rounds. Recent research has produced simple expressions for the convergence rates and fault-tolerance in a broad family of convergent voting algorithms [Kie91]. The analysis accounts for the simultaneous presence of three distinct modes of faults: asymmetric (Byzantine), symmetric, and benign (self-incriminating).

To date, the Approximate Agreement problem has been solved only for systems which are completely connected. By contrast, most multiprocessors rely on partially connected networks [Fen81, Hwa84]. In systems with partial physical interconnections, processes must relay messages for the system connectivity to be logically complete. As $N$ becomes large, the process of globally exchanging local values can consume a great deal of time and communication resources. For example, in a regular graph of degree $d$ containing $N$ processors, at least $\lceil N/d \rceil$ message time slots are required for a node to receive $N$ messages. This cost could make Approximate Agreement impractical in large systems.

Two approaches have been taken to reduce the overhead of global message exchange in partially connected systems. One approach employs special purpose communication hardware to increase the efficiency of handling relayed messages [Ram90]. However, this approach does not reduce the order of complexity of message traffic. The other approach considers the system to be hierarchically composed of processor clusters [Shi87]. Within each cluster, all processors are completely connected. One processor in each cluster is also connected to one processor in another cluster such that the set of clusters is completely connected. It is then possible to set one tolerance on agreement within a cluster, and a looser tolerance on agreement between clusters.

This paper presents a different approach to reducing the overhead of Approximate Agreement. It is assumed that the system in question is a large, regular, sparsely connected network which prohibits the relay of messages. Each node receives only those messages initiated by its immediate neighbors. Global convergence must then occur with each process 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 number of messages passed by each link is $O(1)$, rather than $O(N/d)$. Thus, the elimination of message relays makes message passing overhead independent of $N$, achieving significant savings for large systems.

This paper examines the conditions under which fault-tolerant Approximate Agreement can be achieved without message relay for a broad family of convergent voting algorithms. Section 2 presents some necessary background and definitions, including a brief summary of recent advances with regard to completely connected systems [Kie91]. Section 3 presents the main results of this work, the derivation of simple expressions for the local convergence properties of partially connected systems. Section 4 presents a case study in global convergence. The results of this work, and directions for continuing research are summarized in Section 5.

2 BACKGROUND AND DEFINITIONS

The analyses in this paper are restricted to synchronous systems [Dol83]. In a synchronous system there are finite bounds on the processing and communication delays of non-faulty processes. Thus, a point in time exists when a convergent voting algorithm has received all data from all non-faulty processes. Any data arriving after that time is assumed to be from a faulty process. The synchronous system model is representative of real-time systems and applicable to both data voting and clock synchronization.

2.1 Real–Valued Multisets

Approximate Agreement requires the manipulation of non-disjoint 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. For example, a set of real numbers contains no more than one occurrence of any given
value, while a multiset of real numbers may contain multiple occurrences of a value. The number of times a particular object (value) appears in a multiset is called the *Multiplicity* of that object. A finite multiset $V$ of real values may be represented as a mapping $V : \mathbb{R} \to \mathbb{N}$. For each real value $r$, $V(r)$ is defined as the multiplicity of $r$ in $V$. The size of $V$ is $V = |V| = \sum_{r \in \mathbb{R}} V(r)$.

An alternative representation for a multiset of real numbers is a monotonically increasing sequence of the real values of its elements, i.e. $V = \langle v_1, \ldots, v_V \rangle$ ordered such that: $v_i \leq v_{i+1}$ $\forall$ $i \in \{1, \ldots, V-1\}$ [And63, Liu85]. Both representations of a multiset are equivalent, but for certain operations one form or the other is more convenient. To avoid confusion, we use upper-case symbols for elements in the real-to-integer mapping form, e.g. $V(r)$. Similarly, we use angle-braces and lower-case symbols for elements in the sequence form, e.g. $V = \langle v_1, \ldots, v_V \rangle = \langle v_i \rangle$ $\forall$ $i \in \{1, \ldots, V\}$.

**Real-Valued Parameters** — A multiset of real numbers has several useful real-valued parameters.

\[
\begin{align*}
\min(V) &= \min \{ r \in \mathbb{R} : V(r) > 0 \} = v_1; \text{ the minimum value of the elements in } V. \\
\max(V) &= \max \{ r \in \mathbb{R} : V(r) > 0 \} = v_V; \text{ the maximum value of the elements in } V. \\
\rho(V) &= [\min(V), \max(V)] = [v_1, v_V]; \text{ the real interval spanned by } V. \rho(V) \text{ is called the range of } V. \\
\delta(V) &= \max(V) - \min(V) = v_V - v_1; \text{ the difference between the maximum and minimum values of } V. \delta(V) \text{ is called the diameter of } V. \\
\text{mean}(V) &= \text{The arithmetic mean of the real values of all elements of } V; \\
\text{mean}(V) &= \frac{1}{V} \left( \sum_{r \in \mathbb{R}} V(r) \cdot r \right) = \frac{1}{V} \left( \sum_{i=1}^{V} v_i \right).
\end{align*}
\]
Multiset Relations – Two multisets U and V may be related to each other by Union, Intersection, Sum, or Difference.

Union: Let \( W = V \cup U \). Then \( W(r) = \max \{ V(r), U(r) \} \ \forall \ r \in \mathbb{R} \).

Intersection: Let \( W = V \cap U \). Then \( W(r) = \min \{ V(r), U(r) \} \ \forall \ r \in \mathbb{R} \).

Sum: Let \( W = V + U \). Then \( W(r) = V(r) + U(r) \ \forall \ r \in \mathbb{R} \).

Difference: Let \( W = V - U \). Then \( \forall \ r \in \mathbb{R}: \)

\[
W(r) = \begin{cases} 
V(r) - U(r) & \text{if } V(r) > U(r) \\
0 & \text{otherwise}
\end{cases}
\]

Reduction Functions – Three functions essential to many voting algorithms involve the removal of the largest and/or the smallest valued element from a multiset V [Dol83].

\( S(V) \) = the multiset remaining after one occurrence of the smallest element of V has been removed; \( S(V) = V - \{ \min(V) \} = \langle v_2, \ldots, v_V \rangle \).

\( L(V) \) = the multiset remaining after one occurrence of the largest element of V has been removed; \( L(V) = V - \{ \max(V) \} = \langle v_1, \ldots, v_{(V-1)} \rangle \).

\( \text{Red}(V) = S(L(V)) = L(S(V)) = \langle v_2, \ldots, v_{V-1} \rangle \).

Functions \( S(V) \), and \( L(V) \), may be applied recursively to remove more than one extremal element from V, where the number of applications of the function is indicated by a superscript. Given any non-negative integer \( t \leq V \), \( S^t(V) = \langle v_{(1+t)}, \ldots, v_V \rangle \), and \( L^t(V) = \langle v_1, \ldots, v_{(V-t)} \rangle \). Similarly, for any non-negative integer \( t \leq \lfloor V/2 \rfloor \), \( \text{Red}^t(V) = S^t(L^t(V)) = L^t(S^t(V)) = \langle v_{(1+t)}, \ldots, v_{V-t} \rangle \).

Subsequences – Intuitively, a subsequence of a multiset is a submultiset whose elements are determined solely by their relative positions in the sequence of the
original multiset. For a more formal definition consider two non-empty multisets $V = \langle v_i \rangle \forall i \in \{1, \ldots, V\}$ and $U = \langle u_j \rangle \forall j \in \{1, \ldots, U\}$, where $U \subseteq V$. $U$ is a subsequence of $V$ if there is an order-preserving one-to-one mapping $k$, from the indices of $U$ to the indices of $V$, i.e. $u_j = v_{k(j)} \forall j \in \{1, \ldots, U\}$ and $k(j) < k(j+1) \forall j \in \{1, \ldots, U-1\}$.

**Dominance** — Dominance describes the relative values of corresponding elements in two multisets. Given two equal-sized non-empty multisets $W = \langle w_1, \ldots, w_W \rangle$ and $W' = \langle w'_1, \ldots, w'_W \rangle$, $W$ dominates $W'$ if and only if $w_i \geq w'_i \forall i \in \{1, \ldots, W\}$. Dominance has the property that it is preserved under the subsequence operation, i.e. given two equal sized multisets $V$ and $V'$, let $U$ be a subsequence of $V$, and $U'$ be the same subsequence of $V'$. If $V$ dominates $V'$, then $U$ dominates $U'$.

### 2.2 Completely Connected Systems

Traditionally, each new convergent voting algorithm has been presented individually with an *ad-hoc* proof of its convergence rate. However, convergence in completely connected systems has recently been studied for a broad family of convergent voting algorithms in the presence of multiple fault modes. The results of that work are reviewed here as they form the basis of our study of incompletely connected systems.

#### 2.2.1 Single-Step Convergence

A *correct* value is defined as any value properly transmitted by a non-faulty process. Similarly, an *error* is any value transmitted by a faulty process, regardless of its actual numerical value. Single-step convergence is then formally defined in terms of the following.

$V_k = \text{The multiset of values received in one round by arbitrary non-faulty process } k.$
\[ V = |V_k| \]. If less than \( V \) values are received, then an arbitrary default value is chosen for each missing value so that \( V \) is identical for all non-faulty processes.

\( U_k \) = The submultiset of values in \( V_k \) generated by non-faulty processes.

\( U_{irj} = U_i \cap U_j \), the multiset of correct values received identically by two processes \( i \) and \( j \). With complete connectivity, \( U_{irj} \) is identical for all non-faulty processes. \( U_{irj} \) may thus be taken as the multiset of correct values received by all non-faulty processes.

Each non-faulty process \( k \) executes a convergent voting algorithm \( F(V_k) \). A voting algorithm is convergent if it guarantees both of the following conditions:

[C1] For each non-faulty process \( k \), the voted value is within the range of correct values, i.e. \( F(V_k) \in \rho(U_{irj}) \).

[C2] 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. \( |F(V_i) - F(V_j)| \leq C \delta(U_{irj}) \), where \( 0 \leq C < 1 \).

Parameter \( C \) is the Convergence Rate of a voting algorithm, the primary measure of its effectiveness. Assuming that \( \delta(U_{irj}) > 0 \), \( C \) is defined as the maximum possible value of:

\[ C = \frac{|F(V_i) - F(V_j)|}{\delta(U_{irj})} \]

The Robustness of a voting algorithm is the minimum number of processes \( N \) required to tolerate \( t \) faults.

### 2.2.2 Multiple Mode Fault Model

In real-world systems truly Byzantine behavior occurs only under highly improbable conditions. The behavior of most faults is much less malicious. Accordingly, Meyer and Pradhan [Mey87] have partitioned the space of all possible faults into
two distinct modes: *Benign* faults, defined as those which are self-incriminating, or immediately self-evident to all non-faulty processes, and *Malicious* faults, defined as all faults which do not qualify as benign. Thambidurai and Park [Tha88] have further partitioned malicious faults into two sub-modes: *Symmetric* faults, whose behavior is perceived identically by all non-faulty processes, and *Asymmetric* faults, whose behavior may be perceived differently by different non-faulty processes. It is the asymmetric fault mode which fits the definition of a Byzantine fault.

The total number of faulty processes \( t \) in a system is given by \( t = a + s + b \), where \( a \) is the number of asymmetric faults, \( s \) is the number of symmetric faults, and \( b \) is the number of benign faults. A major advantage of this three-mode fault model is that systems can often be designed to minimize the probability of the most harmful faults occurring. For example, the probability of two coincident symmetric faults is generally much higher than the probability of two coincident asymmetric faults. Thambidurai and Park used this model to derive tighter bounds on the robustness of selected Byzantine Agreement algorithms [Tha88]. Recently, Kieckhafer and Azadmanesh applied this same fault model to Approximate Agreement in completely connected networks [Kie91].

### 2.2.3 MSR Voting Algorithms

Convergence properties have been determined for an entire family of voting algorithms with the general form:

\[
F(V) = \text{mean} [\text{Sel}_\sigma (\text{Red}^\tau (V))].
\]

The reduction function \( \text{Red}^\tau \) removes the \( \tau \) largest and \( \tau \) smallest elements from multiset \( V \), The function \( \text{Sel}_\sigma \) selects a submultiset of \( \sigma \) elements from the reduced multiset \( \text{Red}^\tau (V) \). The final voted value is the arithmetic mean of the selected multiset. If \( \text{Sel}_\sigma \) produces a subsequence of \( \text{Red}^\tau (V) \), then \( F(V) \) is the *Mean* of a *Subsequence* of the *Reduced* multiset. Voting algorithms with this property are called *Mean-Subsequence-Reduced* (MSR) algorithms [Kie91].
Members of the MSR family differ from each other only in their definition of the selection function $Sel_\sigma$. Examples of MSR algorithms include the Fault-Tolerant Midpoint, the Fault-Tolerant Mean [Dol83], Dolev’s Optimal algorithm [Dol86], the Mixed-Mode Optimal algorithm, the Binary Mean, the Binary Suboptimal Algorithm [Kie91] and several algorithms for synchronizing phase-locked loops [Vas88, Vas89].

2.2.4 MSR Convergence With Complete Connectivity

Simple expressions have been found for the convergence rate and robustness of any MSR algorithm in a completely connected system [Kie91]. These results show that it is advantageous to discard recognized benign errors prior to voting provided that all processes do so. Thus, given a total of $N$ processes containing $b$ benign faults, $V = N - b$.

For completely connected systems, an MSR algorithm can be convergent only if:

\[ V \geq 2\tau + \max(a + 1, \sigma), \]  \hspace{1cm} (2.1)

\[ \tau \geq a + s, \]  \hspace{1cm} (2.2)

\[ \sigma \begin{cases} 
\geq 1 & : a = 0 \\
\geq 2 & : a > 0 
\end{cases}. \]  \hspace{1cm} (2.3)

Substituting the minimum allowable values for $\sigma$ and $\tau$ from (2.2) and (2.3) into (2.1) shows that a convergent MSR voting algorithm exists if:

\[ \tau \geq \tau_c \equiv a + s \]  \hspace{1cm} (2.4)

\[ V \geq V_c \equiv 3a + 2s + 1, \]  \hspace{1cm} (2.5)

\[ \text{hence } N \geq N_c \equiv 3a + 2s + b + 1. \]  \hspace{1cm} (2.6)

An important result of [Kie91] is the ease with which the convergence rate $C$ can be determined for any MSR voting algorithm $F(V) = \text{mean}[Sel_\sigma(\text{Red}^\tau(V))]$. First, define the Medial Multiset $M = \text{Red}^\tau(V) = \langle m_1, \ldots, m_M \rangle$ and the Selected Multiset $S = Sel_\sigma(M) = \langle s_1, \ldots, s_\sigma \rangle$. Then observe that for each $\ell \in \{1, \ldots, \sigma\}$ there
exists exactly one $k(\ell) \in \{1, \ldots, M\}$ which guarantees that $s_{\ell} = m_{k(\ell)}$. For a given subsequence function, define $g, h \in \{1, \ldots, \sigma\}$ such that $g \leq h$. Then define:

$$\Delta k(g, h) = [k(h) - k(g)]$$. If $g = h$, then $\Delta k(g, h) = 0$. However, if $g < h$, then $\Delta k(g, h)$ is the number of elements of $M$ in $\langle m_{k(g)+1}, \ldots, m_{k(h)} \rangle$.

$$\gamma = \min(h - g) : \Delta k(g, h) \geq a \ \forall \ g, h \in \{1, \ldots, \sigma\}$$. Thus, $\gamma$ is the minimum value of $(h - g)$ which assures that $k(h) - k(g) \geq a$. By this definition, $\gamma$ exists only if $M \geq a + 1$.

It has been shown that the convergence rate of an MSR algorithm is:

$$C = \frac{\gamma}{\sigma}. \quad (2.7)$$

This result has the advantage that parameters $\gamma$ and $\sigma$ are easily determined, given $\tau$, and $V$, and any subsequence selection function $Sel_\sigma$.

### 3 LOCAL CONVERGENCE

A partially connected system differs from a completely connected system in that a given process does not receive values from all non-faulty processes. Rather, it receives values only from a specific subset of processes. There are now two types of convergence to be considered: local convergence over a specified subgraph, and global convergence over the entire system graph.

Unlike a completely connected system, global convergence can not be inferred simply from the existence of a convergent voting algorithm. Local convergence is a necessary condition for global convergence. In addition, the system topology must permit convergence to “diffuse” across overlapping regions of local convergence. This condition raises two questions: (1) under what conditions is local convergence attainable with MSR algorithms, (2) under what conditions is local convergence sufficient for global convergence. This paper addresses primarily the first question, leaving the second question for future work.
The following are assumed regarding the system:

1. The system topology is a non-hierarchical, regular, homogeneous, undirected graph of $N$ processor nodes, each with degree $d$.

2. Messages received by a process may not be relayed to another process. Thus, the physical and logical connectivity are identical. Since a process receives its own messages as well as those of its immediate neighbors, $V = d + 1$ for all non-faulty processes.

3. $N >> V$. Then “wrap-around” effects can not assist the local convergence process in a given voting round.

The following sets and multisets define the relationships between the various values received by two nodes $i$ and $j$ in a partially connected system.

$P_i = \text{The set of processes whose values are receivable by process } i \text{ (} P_j \text{ is similarly defined for process } j \text{).}$

$P_{i\cap j} = P_i \cap P_j$, the set of processes whose values are receivable by both process, $i$ and process $j$.

$P_{i\cup j} = P_i \cup P_j$, the set of processes whose values are receivable by process $i$, process $j$, or both.

$U_{i\cap j} = \text{The multiset of all values generated by non-faulty processes in } P_{i\cap j}.$

$U_{i\cup j} = \text{The multiset of all values generated by non-faulty processes in } P_{i\cup j}.$

In a completely connected system, $U_{i\cap j} \equiv U_{i\cup j}$. However, in a partially connected system $U_{i\cap j} \subset U_{i\cup j}$. We must therefore define two types of local convergence for partially connected systems.

**Intersection Convergence:** Given a voting algorithm $F(V)$, two processes $i$ and $j$ are **Intersection-Convergent** if the following conditions are both true:

[I1] $F(V_i) \in \rho(U_{i\cap j})$, and $F(V_j) \in \rho(U_{i\cap j})$, 

11
[I2] \[|F(V_i) - F(V_j)| \leq C \delta(U_{i\cap j}), \text{ where } 0 \leq C < 1.\]

**Union Convergence:** Given a voting algorithm \(F(V)\), two processes \(i\) and \(j\) are
*Union-Convergent* if the following conditions are both true:

[U1] \(F(V_i) \in \rho(U_{i\cup j}), \text{ and } F(V_j) \in \rho(U_{i\cup j}),\)

[U2] \[|F(V_i) - F(V_j)| \leq C \delta(U_{i\cup j}), \text{ where } 0 \leq C < 1.\]

A major difference between completely connected and partially connected systems is the strategy for handling benign faults. In a completely connected system, benign faults can be ignored because all processes can delete the benign errors from \(V\) and vote with a smaller sized multiset. However, in a partially connected system, the benign fault is not self-evident to all processes. This strategy would cause different processes to vote using different sized multisets so that \(V\) would not be identical for all processes. Thus, only symmetric and asymmetric faults are considered in this analysis, leaving \(t = \tau = a + s.\)

### 3.1 Intersection Convergence

The conditions necessary to ensure that two processes \(i\) and \(j\) are Intersection-Convergent can be derived as a variant on the completely connected system previously described. We begin with the following definitions:

\[
a = \text{The number of asymmetrically faulty processes in } P_{i\cap j}.
\]

\[
s = \text{The number of symmetrically faulty processes in } P_{i\cap j}.
\]

\[
\chi = |P_i| - |P_{i\cap j}| = |P_j| - |P_{i\cap j}|, \text{ the number of processes whose values are receivable by either } i \text{ or } j, \text{ but not by both.}
\]

Given an MSR voting algorithm \(F(V) = \text{mean } [Sel_\sigma(\text{Red}^\tau(V))],\) medial multiset \(M = \text{Red}^\tau(V) = \langle m_1, \ldots, m_M \rangle\) and selected subsequence \(S = Sel_\sigma(M) = \langle s_1, \ldots, s_\sigma \rangle,\) define \(\Delta k(g, h)\) as previously defined in Subsection 2.2. Then define
\( \gamma' \) as a variation on \( \gamma \) obtained by replacing each occurrence of the value \( a \) with the expression \( [a + \chi] \), i.e.

\[
\gamma' = \min(h - g) : \Delta k(g, h) \geq [a + \chi] \quad \forall \ g, h \in \{1, \ldots, \sigma\}.
\]

Thus, \( \gamma' \) is the minimum value of \( (h - g) \) which assures that \( k(h) - k(g) \geq [a + \chi] \). By this definition, \( \gamma' \) exists only if \( M \geq [a + \chi] + 1 \).

**THEOREM 1** : Given non-negative integers \( a, s, \chi, \) and \( \tau \), where \( [a + \chi] + s \leq \tau \), and multisets \( V_i \) and \( V_j \), where \( V = |V_i| = |V_j| \geq 2\tau + \max([a + \chi], \sigma) \), and an MSR voting algorithm \( F(V) = \text{Sel}_\sigma(\text{Red}^\tau(V)) \), then the algorithm can be intersection convergent only if:

\[
V \geq 2\tau + \max([a + \chi] + 1, \sigma), \quad \text{(1)}
\]

\[
\tau \geq [a + \chi] + s, \quad \text{(2)}
\]

\[
\sigma \begin{cases} 
\geq 1 & : [a + \chi] = 0 \\
\geq 2 & : [a + \chi] > 0 
\end{cases} \quad \text{(3)}
\]

Furthermore, if the algorithm is intersection convergent, then:

\[
C' = \frac{\gamma'}{\sigma}. \quad \text{(4)}
\]

**PROOF** : The key to proving this theorem is the observation that each process \( x_i \in P_i \setminus P_{i \cap j} \) sends a value to process \( i \) but not to process \( j \). For each such \( x_i \) there is a corresponding \( x_j \in P_j \setminus P_{i \cap j} \) which sends a value to process \( j \) but not to process \( i \). For a given pair of processes \((x_i, x_j)\) their transmitted values are not guaranteed to be identical. Furthermore, since \( x_i, x_j \notin P_{i \cap j} \), their respective values are not guaranteed to be within \( \rho(U_{i \cap j}) \). Thus, the pair of processes \((x_i, x_j)\) exhibits the same behavior as a single asymmetrically faulty process regardless of the health of \( x_i \) and \( x_j \). Since there are \( \chi \) such process pairs, the effective number of asymmetric faults is \( [a + \chi] \), rather than \( a \). Thus, for any pair of processes \( i \) and \( j \), the results of [Kie91] for completely connected systems, as stated in (2.4) – (2.7) are valid for Intersection-Convergent partially connected systems if \( [a + \chi] \) is substituted for \( a \), producing (1) – (4) above. \( \square \)
Applying the minimum allowable values for $\sigma$ and $\tau$ from Theorem 1 shows that the minimum conditions necessary for the existence of an Intersection-Convergent MSR voting algorithm are:

$$V \geq V_I \equiv 3[a + \chi] + 2s + 1, \quad (3.1)$$

$$\tau \geq \tau_I \equiv [a + \chi] + s. \quad (3.2)$$

The minimum size of $P_{ij}$ can be derived from (3.1) by noting that $|P_{ij}| = V - \chi$. Thus, Intersection Convergence between processes $i$ and $j$ is possible only if:

$$|P_{ij}| \geq 3a + 2\chi + 2s + 1. \quad (3.3)$$

### 3.2 Union Convergence

Union Convergence is a less restrictive convergence criterion than Intersection Convergence. Accordingly, it will be shown that Union Convergence is possible under less restrictive conditions that Intersection Convergence. The main results are embodied in three theorems which (1) define a general criterion for bounding convergence rates, (2) identify a worst case scenario which maximizes $|F(V_i) - F(V_j)|$, (3) derive expressions for the convergence rate of any MSR algorithm, and the conditions under which Union Convergence can be guaranteed.

#### 3.2.1 Preliminary Lemmas

Two preliminary lemmas are presented describing essential properties of multisets. Lemma 1 defines conditions under which the range of the medial set is guaranteed to be within the range of the correct values. Lemma 2 defines a property of dominance which permits the mean values of two subsequences to be compared.

**Lemma 1**: Given non-empty multisets $U$ and $V$ and non-negative integer $\tau$:

**IF**: $|V| \geq 2\tau + 1,$

**AND**: $|V - U| \leq \tau,$

**THEN**: $\rho(\text{Red}^\tau (V)) \subseteq \rho(U).$
LEMMA 2: Given two non-empty multisets of size $W$, $W = \langle w_1, \ldots, w_W \rangle$ and $W' = \langle w'_1, \ldots, w'_W \rangle$, if $W$ dominates $W'$, then

$$\sum_{i=1}^{W} w_i \geq \sum_{i=1}^{W'} w'_i.$$ 

PROOF: See [Kie91, Lem 3]. \(\square\)

3.2.2 General Convergence Criterion

The impact of errors generated within $P_{i \cap j}$ is different from the impact of those generated in $P_{i \cup j} \setminus P_{i \cap j}$. The definitions of $a$ and $s$ remain, as in the previous section, the number of asymmetric and symmetric faults in $P_{i \cap j}$, respectively. In addition to $a$ and $s$ we define:

$$f = \text{The maximum number of faulty processes in either } P_{i \setminus P_{i \cap j}} \text{ or } P_{j \setminus P_{i \cap j}} \text{ regardless of the fault modes exhibited.}$$

The following theorem establishes a criterion for bounding the convergence rate of MSR voting algorithms with respect to $U_{i \cup j}$.

THEOREM 2: Given non-negative integers $a$, $s$, $f$, $\tau$, and $\sigma$, where $\tau \geq a + s + f$, two non-disjoint multisets $V_i$ and $V_j$ such that $V = |V_i| = |V_j| \geq 2\tau + \sigma$, and MSR algorithm $F(V) = \text{mean} \left[ \text{Sel}_\sigma \left( \text{Red}^\tau (V) \right) \right]$, define two equal-sized multisets $S_i^* \subseteq S_i$ and $S_j^* \subseteq S_j$. Also define $s_i^*$ and $s_j^*$ as the sums of all elements of $S_i^*$ and $S_j^*$, respectively, i.e.

$$s_i^* = \sum_{s_i \in S_i^*} s_i, \quad \text{and} \quad s_j^* = \sum_{s_j \in S_j^*} s_j.$$ 

15
IF: \( F(V_i) \geq F(V_j) \),
AND: \( \exists S_i^* \) and \( S_j^* \) such that \( S_j^* \) dominates \( S_i^* \),
THEN: \(|F(V_i) - F(V_j)| \leq \left[ (\sigma - \sigma^*)/\sigma \right] \delta(U_{i\cup j})\), where \( \sigma^* = |S_i^*| = |S_j^*| \).

PROOF: The constraint \( V \geq 2\tau + \sigma \) ensures that \( M \geq \sigma \), so that the \( Sel_\sigma \) function is valid in \( M_i \) and \( M_j \). \( S_i^* \) and \( S_j^* \) contain \( \sigma^* \) elements of \( S_i \) and \( S_j \), respectively, leaving \( (\sigma - \sigma^*) \) elements of \( S_i \) and \( S_j \) which are not contained in \( S_i^* \) and \( S_j^* \).

Only values from faulty nodes can be outside of \( \rho(U_{i\cup j}) \). According to Lemma 1, the constraint that \( \tau \geq a + s + f \) ensures that \( \rho(M_i) \subseteq \rho(U_{i\cup j}) \), and \( \rho(M_j) \subseteq \rho(U_{i\cup j}) \). Therefore, the values of all elements of \( S_i \) and \( S_j \) are within \( \rho(U_{i\cup j}) \). This fact yields the bounds:

\[
F(V_i) = \text{mean}[S_i] = \frac{1}{\sigma} \sum_{r \in \mathbb{R}} S_i(r) r \leq \frac{1}{\sigma} \left[(\sigma - \sigma^*) \max(U_{i\cup j}) + s_i^*\right]
\]

\[
F(V_j) = \text{mean}[S_j] = \frac{1}{\sigma} \sum_{r \in \mathbb{R}} S_j(r) r \geq \frac{1}{\sigma} \left[(\sigma - \sigma^*) \min(U_{i\cup j}) + s_j^*\right]
\]

By hypothesis, \( F(V_i) \geq F(V_j) \), then:

\[
|F(V_i) - F(V_j)| \leq \frac{1}{\sigma} \left[(\sigma - \sigma^*) \max(U_{i\cup j}) + s_i^*\right] - \frac{1}{\sigma} \left[(\sigma - \sigma^*) \min(U_{i\cup j}) + s_j^*\right]
\]

\[
= \frac{1}{\sigma} \left[(\sigma - \sigma^*) \max(U_{i\cup j}) - (\sigma - \sigma^*) \min(U_{i\cup j}) + (s_i^* - s_j^*)\right]
\]

\[
= \left[\frac{\sigma - \sigma^*}{\sigma}\right] \delta(U_{i\cup j}) + \frac{s_i^* - s_j^*}{\sigma}.
\]

If \( S_j^* \) dominates \( S_i^* \), then by Lemma 2, \( s_j^* \geq s_i^* \), so this expression reduces to:

\[
|F(V_i) - F(V_j)| \leq \left[\frac{\sigma - \sigma^*}{\sigma}\right] \delta(U_{i\cup j}).
\]
Theorem 2 is symmetric, i.e. if $F(V_i) \leq F(V_j)$ and $s_i^* \geq s_j^*$, then the same convergence rate can be derived. However, since the indices $i$ and $j$ are arbitrary, it can always be stated without loss of generality that $F(V_i) \geq F(V_j)$.

3.2.3 Worst-Case Scenario

A worst-case scenario maximizes $|F(V_i) - F(V_j)|$. To characterize one such scenario, the multiset of all values received by nodes $i$ and $j$ is partitioned into several disjoint multisets. First, we define multisets containing only symmetric errors generated within $P_{i\cap j}$.

$$U_c = U_{i\cap j} \cup \text{all symmetric errors generated within } P_{i\cap j} \text{ whose values are within } \rho(U_{i\cup j})$$. The symmetric errors in $U_c$ are indistinguishable from correct values, and thus do not inhibit either convergence condition [U1] or [U2].

$$L = \langle \ell_1, \ldots, \ell_\Lambda \rangle$$, the multiset of all symmetric errors generated within $P_{i\cap j}$ whose values are less than $\min(U_{i\cup j})$.

$$G = \langle g_1, \ldots, g_\Gamma \rangle$$, the multiset of all symmetric errors generated within $P_{i\cap j}$ whose values are greater than $\max(U_{i\cup j})$.

Next, we define multisets containing only errors received asymmetrically, i.e. by one process, but not by the other.

$$A = \text{The multiset of errors received by process } j \text{ but not by process } i. \text{ A thus contains all asymmetric errors received by } j \text{ which were generated within } P_{i\cap j} \cup \text{all errors generated within } P_j \setminus P_{i\cap j}. \text{ Thus, } |A| \leq a + f.$$ 

$$B = \text{The multiset of errors received by process } i \text{ but not by process } j. \text{ B thus contains all asymmetric errors received by } i \text{ which were generated within } P_{i\cap j} \cup \text{all errors generated within } P_i \setminus P_{i\cap j}. \text{ Thus, } |B| \leq a + f.$$ 

Finally, we define multisets containing only correct values received asymmetrically, i.e. by one process but not by the other.
\( \mathbf{X} = \langle x_1, \ldots, x_{\chi-z} \rangle \), where \( z \) is an integer such that \( 0 \leq z \leq f \). \( \mathbf{X} \) is the minimal multiset of correct values received by process \( j \), but not by process \( i \), i.e. all correct values generated within \( P_{i\cup j} \setminus P_i \).

\( \mathbf{Y} = \langle y_1, \ldots, y_{\chi-z} \rangle \), where \( z \) is an integer such that \( 0 \leq z \leq f \). \( \mathbf{Y} \) is the minimal multiset of correct values received by process \( i \), but not by process \( j \), i.e. all correct values generated within \( P_{i\cup j} \setminus P_j \).

**Example 1:** Figure 1 illustrates these multisets for an octagonally connected mesh network. In this figure, \( P_i \) consists of the top three rows of processors, while \( P_j \) consists of the bottom three rows. Thus, \( P_{i\cup j} \) consists of all nodes shown in the figure, while \( P_{i\cap j} \) consists of only those nodes within the dashed box. As defined above, all symmetric errors generated within the dashed box are contained in either \( U_c \), \( L \), or \( G \), depending on their values with respect to \( \rho(U_{i\cup j}) \). \( A \) contains all asymmetric errors generated within the dashed box (as received by node \( j \)) plus all errors generated in the bottom row of the diagram. Similarly, \( B \) contains all asymmetric errors generated within the dashed box (as received by node \( i \)) plus all errors generated in the top row of the diagram. Finally, \( X \) consists of all correct values generated in the bottom row, while \( Y \) consists of all correct values generated in the top row.

The following three Lemmas establish preliminary conditions which permit a worst-case scenario to be derived in Theorem 3.

**Lemma 3:** Given non-negative numbers \( z \), \( \tau \), and \( \sigma \), where \( z \leq \tau \), a multiset \( V \), where \( V \geq 2\tau + \sigma \), and an MSR algorithm \( F(V) = \text{Sel}_\sigma(\text{Red}^\tau(V)) \), assume \( V \) can be partitioned into two disjoint multisets \( U \) and \( Z \) such that: \( U \) is a multiset of fixed values, \( Z \) is a multiset of variable values, and \( |Z| = z \).

**IF:** each \( z_i \in Z \) assumes its maximum possible value,

**THEN:** \( F(V) \) is the maximum possible value of \( F(V) \), given \( U \).

Similarly,
Figure 1: Octagonal Mesh Example

IF: each \( z_i \in \mathbb{Z} \) assumes its minimum possible value,

THEN: \( F(V) \) is the minimum possible value of \( F(V) \), given \( U \).

PROOF: (See [Kie91, Lemma 4 and Lemma 5]). \( \square \)

**LEMMA 4:** The maximum possible value of \( F(V_i) \) given \( L, U_c, \) and \( G \) is obtained if:

\[
y_k = \max(U_{i,j}) \forall y_k \in Y, \text{ and } b_k \geq \max(U_{i,j}) \forall b_k \in B.
\]

PROOF: From the preceding definitions, it can be seen that \( V_i = L + U_c + Y + G + B \). Multisets \( L, U_c, \) and \( G \) are common to both \( V_i \) and \( V_j \) and are thus considered fixed. However, the values comprising \( Y \) and \( B \) are unique to \( V_i \) and may thus be
considered variable. For the correct values in $\mathbf{Y}$, the maximum value is bounded by $\max(U_{i,j})$, while the errors in $\mathbf{B}$ can exceed $\max(U_{i,j})$. According to Lemma 3, this scenario produces the maximum possible value of $F(V_i)$.

**LEMMA 5**: The minimum possible value of $F(V_j)$ is obtained if:

$$x_k = \min(U_{i,j}) \quad \forall \ x_k \in \mathbf{X}, \quad \text{and} \quad a_k \leq \min(U_{i,j}) \quad \forall \ a_k \in \mathbf{A}.$$  

**PROOF**: From the preceding definitions, it can be seen that $V_j = \mathbf{A} + \mathbf{L} + \mathbf{X} + U_c + \mathbf{G}$. The remainder of the proof is symmetric to the proof of Lemma 4.

Based on previously defined multisets, we now define:

$$\mathbf{W} = \mathbf{L} + \mathbf{X} + U_c + \mathbf{Y} + \mathbf{G},$$  

the multiset of all symmetrically transmitted values originating in $P_{i,j}$. Thus, $|\mathbf{W}| \geq V + \chi - (a + 2f)$.

**THEOREM 3**: Given non-negative integers $a, f, s, \sigma$ and $\tau$, where $\tau \geq a + f + s$, multisets $\mathbf{V}_i$ and $\mathbf{V}_j$ where $V = |\mathbf{V}_i| = |\mathbf{V}_j| \geq 2\tau + \sigma$, multisets $U_c, \mathbf{L}, \mathbf{G}, \mathbf{A}, \mathbf{B}, \mathbf{X}, \mathbf{Y}$ (as previously defined) and MSR algorithm $F(V) = \text{mean}[Sel_\sigma(\text{Red}_\tau(V))]$,

**IF**:  

$$y_k = \max(U_{i,j}) \quad \forall \ y_k \in \mathbf{Y}, \quad \text{and} \quad b_k \geq \max(U_{i,j}) \quad \forall \ b_k \in \mathbf{B},$$  

AND:  

$$x_k = \min(U_{i,j}) \quad \forall \ x_k \in \mathbf{X}, \quad \text{and} \quad a_k \leq \min(U_{i,j}) \quad \forall \ a_k \in \mathbf{A},$$

**THEN**: $|F(V_i) - F(V_j)|$ is the maximum obtainable, given $\mathbf{L}, U_c, \text{and} \mathbf{G}$.

AND:  

$$\text{Red}_\tau(\mathbf{V}_i) = \langle w_{(1-f+\tau+\chi)}, \ldots, w_{(V-f-\tau+\chi)} \rangle, \quad \text{and}$$  

AND:  

$$\text{Red}_\tau(\mathbf{V}_j) = \langle w_{(1-f+\tau-a)}, \ldots, w_{(V-f-\tau-a)} \rangle.$$  

**PROOF**: The constraint $V \geq 2\tau + \sigma$ ensures that the function $Sel_\sigma$ is valid within $\text{Red}_\tau(\mathbf{V}_i)$ and $\text{Red}_\tau(\mathbf{V}_j)$. The constraint $\tau \geq a + s + f$ further ensures (by Lemma 1) that $\rho(\text{Red}_\tau(\mathbf{V}_i)) \subseteq \rho(U_{i,j})$ and $\rho(\text{Red}_\tau(\mathbf{V}_j)) \subseteq \rho(U_{i,j})$. Thus, this scenario is non-divergent.
Without loss of generality assume \( F(V_i) \geq F(V_j) \). The proof of maximality follows directly from the previous two lemmas. By Lemma 4, the stated hypotheses yield the maximum possible value for \( F(V_i) \). Similarly, by Lemma 5, the stated hypotheses yield the minimum possible value for \( F(V_j) \). Therefore, the maximum possible value of \( |F(V_i) - F(V_j)| \) is obtained.

Under the stated hypotheses a complete ordering exists for \( W \) such that:

\[
W = \{L + X + U_c + Y + G\} = \langle L, X, U_c, Y, G \rangle
\]

\[
= \langle w_1, \ldots, w_{(V-a+f)+(\chi-f)} \rangle = \langle \ell_1, \ldots, \ell_{\Lambda}, x_1, \ldots, x_{(\chi-f)}, u_1, \ldots, u_{(V-a-\chi-\Lambda-\Gamma)}, y_1, \ldots, y_{(\chi-f)} \rangle
\]

Mapping the indices of \( W \) into the indices of the components of \( W \) yields:

\[
\langle w_1, \ldots, w_{\Lambda} \rangle = L = \langle \ell_1, \ldots, \ell_{\Lambda} \rangle
\]

\[
\langle w_{(\Lambda+1)}, \ldots, w_{(\Lambda+\chi-f)} \rangle = X = \langle x_1, \ldots, x_{(\chi-f)} \rangle
\]

\[
\langle w_{(\Lambda+\chi-f+1)}, \ldots, w_{(V-a+f-\Gamma)} \rangle = U_c = \langle u_1, \ldots, u_{(V-a-\chi-\Lambda-\Gamma)} \rangle
\]

\[
\langle w_{(V-a+f-\Gamma+1)}, \ldots, w_{(V-a+f+\chi-f)} \rangle = Y = \langle y_1, \ldots, y_{(\chi-f)} \rangle
\]

\[
\langle w_{(V-a+f-\Gamma+(\chi-f)+1)}, \ldots, w_{(V-a+f)+(\chi-f)} \rangle = G = \langle g_1, \ldots, g_{\Gamma} \rangle
\]

Applying the worst-case conditions of Lemma 4 to process \( i \) to maximize \( F(V_i) \) yields:

\[
V_i = L + U_c + Y + G + B
\]

\[
= \langle L, w_{(\Lambda+(\chi-f)+1)}, \ldots, w_{(V-(a+f)-\Gamma+(\chi-f))}, G + B \rangle.
\]

Convergence condition [U1] requires that all values not within \( \rho(U_{i,j}) \) be discarded by the voting algorithm. For \( V_i \), this means that all elements of \( L \) and \( G + B \) must be removed by \( Red^r \). By definition, \( |L| = \Lambda \leq s \), \( |G| = \Gamma \leq s \) and \( |B| \leq a + f \).
The constraint $\tau \geq a + f + s$ ensures that $\tau \geq |L|$ and $\tau \geq |G + B|$, so that $\text{Red}^\tau (V_i) \subseteq \langle U_c, Y \rangle$. Reducing multiset (2) by $\tau$ thus leaves

$$M_i = \text{Red}^\tau (V_i)$$

$$= \langle w(\Lambda+(\chi-f)+1+[\tau-\Lambda]), \ldots, w(V-(a+f)-\Gamma+(\chi-f)-(\tau-(a+f+\Gamma))) \rangle$$

$$= \langle w(1-f+|\pi+\chi|), \ldots, w(V-f-(a+\chi)) \rangle.$$

Similarly, applying the worst-case conditions from Lemma 5 to minimize $F(V_j)$ yields:

$$V_j = A + L + X + U_c + G$$

$$= \langle A + L, w(\Lambda+1), \ldots, w(V-(a+f)) - \Gamma, G \rangle$$

A similar argument can be applied to $V_j$ as to $V_i$ above regarding the sizes of $A$, $L$, and $G$. This argument yields a similar result: if $\tau \geq a + f + s$, then $\text{Red}^\tau (V_j) \subseteq \langle X, U_c \rangle$. Reducing multiset (7) by $\tau$ thus leaves

$$M_j = \text{Red}^\tau (V_j)$$

$$= \langle w(\Lambda+1+[\tau-(a+f+\Lambda)]), \ldots, w(V-(a+f)-\Gamma-[\tau-\Gamma]) \rangle$$

$$= \langle w(1-f+\tau-\chi), \ldots, w(V-f-\tau-\chi) \rangle.$$

□

3.2.4 Convergence Rate Determination

Theorem 3 revealed a worst-case scenario in which the convergence criterion of Theorem 2 could be applied so that $C = (\sigma - \sigma^*)/\sigma$. However, the maximum value of $\sigma^*$ is not immediately obvious for a given MSR algorithm. The following theorem simplifies the task of finding $C$, given $\gamma'$ (as defined in subsection 3.1), $M$, $a$, $\chi$, and any subsequence function $Sel_\sigma (M)$.

THEOREM 4: Given non-negative integers $a$, $s$, $f$, $\chi$, and $\tau$, where $\tau \geq a + f + s$, multisets $V_i$ and $V_j$, where $V = |V_i| = |V_j| \geq 2\tau + \max(a + \chi + 1, \sigma)$, and function $F(V) = \text{mean} [Sel_\sigma (\text{Red}^\tau (V))]$,
IF: \( Sel_\sigma \) is a subsequence function with parameters \( \sigma \) and \( \gamma' \),

THEN: \(|F(V_i) - F(V_j)| \leq \lfloor \gamma'/\sigma \rfloor \delta(U_{i,j}) \).

PROOF: According to Lemma 1, the constraint \( \tau \geq a + f + s \) ensures that \( \rho(M_i) \subseteq \rho(U_{i,j}) \) and \( \rho(M_j) \subseteq \rho(U_{i,j}) \), so that convergence condition [U1] is satisfied. The constraint \( V \geq 2\tau + \max(a + \chi + 1, \sigma) \) ensures that \( M_i \) and \( M_j \) are of sufficient size that (1) \( M \geq \sigma \), so that \( Sel_\sigma \) is valid, (2) \( M \geq a + \chi + 1 \), so that the medial multiset is large enough for \( \gamma' \) to exist.

The following notation is defined for process \( i \) (with similar notation for process \( j \)).

\[
m_{i,\ell} = \text{the } \ell\text{th element of } M_i, \text{ where } \ell \in \{1, \ldots, M\}.
\]

\[
s_{i,\ell} = \text{the } \ell\text{th element of } S_i, \text{ where } \ell \in \{1, \ldots, \sigma\}.
\]

Assume the worst-case scenario of Theorem 3, which yields the two Medial Multisets:

\[
M_i = \langle w_{(1-f+\tau+\chi)}, \ldots, w_{(V-f-\tau+\chi)} \rangle \quad \text{and} \quad M_j = \langle w_{(1-f+\tau-a)}, \ldots, w_{(V-f-\tau-a)} \rangle.
\]

Then:

\[
s_{i,g} = m_{i,k(g)} = w_{k(g)-f+\tau + \chi},
\]

\[
s_{j,h} = m_{j,k(h)} = w_{k(h)-f+\tau - a} = w_{k(g)-f+\tau + \lfloor \Delta k(g,h) - a \rfloor}.
\]

Comparing (1) and (2), shows that if \( \lfloor \Delta k(g,h) - a \rfloor \geq \chi \), then \( s_{j,h} \geq s_{i,g} \). If \( h \) is replaced by \( (g + \gamma') \), then this result generalizes to include all indices \( g \in \{1, \ldots, \sigma - \gamma'\} \). Thus,

\[
s_{j,(g+\gamma')} \geq s_{i,g} \quad \forall \ g \in \{1, \ldots, (\sigma - \gamma')\}.
\]

To apply the convergence criterion of Theorem 2, we define \( S_i^{\ast} \) and \( S_j^{\ast} \) as follows and consider the \( g \text{th} \) element of each.

\[
S_i^{\ast} = L^{\gamma'}(S_i) = \langle s_{i,1}, \ldots, s_{i,(\sigma-\gamma')} \rangle \Rightarrow s_{i,g}^{\ast} = s_{i,g};
\]

\[
S_j^{\ast} = S^{\gamma'}(S_j) = \langle s_{j,(1+\gamma')}, \ldots, s_{j,\sigma} \rangle \Rightarrow s_{j,g}^{\ast} = s_{j,(g+\gamma')}.
\]
By (3), \( s_{j,g+\gamma'} \geq s_{i,g} \Rightarrow s_{j,g} \geq s_{i,g} \forall g \in \{1, \ldots, \sigma - \gamma'\} \). Thus, \( S_j^* \) dominates \( S_i^* \).

With \( S_i^* \) and \( S_j^* \) defined as in (4) and (5), \( \sigma^* = (\sigma - \gamma') \) so that Theorem 2 yields:

\[
|F(V_i) - F(V_j)| \leq \left[ \frac{\sigma - \sigma^*}{\sigma} \right] \delta(U_{i\cup j})
\]

\[
= \left[ \frac{\sigma - (\sigma - \gamma')}{\sigma} \right] \delta(U_{i\cup j})
\]

\[
= \left[ \frac{\gamma'}{\sigma} \right] \delta(U_{i\cup j}).
\]

\( \square \)

Theorem 4 is valid only if \( V \geq 2\tau + \max(a + \chi + 1, \sigma) \), and \( \tau \geq a + s + f \). As was the case with Intersection Convergence, \( \sigma \) must be large enough that \( \gamma' \) exists. Thus:

\[
\sigma \begin{cases}
\geq 1 : & [a + \chi] = 0 \\
\geq 2 : & [a + \chi] > 0
\end{cases}
\]

Applying the minimum values for \( \sigma \) and \( \tau \) yields:

\[
\tau \geq \tau_U = a + s + f \quad (3.4)
\]

\[
V \geq V_U = (3a + 2s + 1) + \chi + 2f. \quad (3.5)
\]

In addition, the minimum required overlap between \( P_i \) and \( P_j \) is:

\[
|P_{i\cap j}| \geq V - \chi = 3a + 2s + 2f + 1. \quad (3.6)
\]

### 3.3 Application Examples

Table 1 summarizes the relevant parameters for convergence in completely connected systems and for Intersection Convergence or Union Convergence in partially connected systems. Applying these results to specific error scenarios in several interconnection topologies illustrates the relative robustness of these networks for both types of local convergence.
<table>
<thead>
<tr>
<th>Completely Connected</th>
<th>Partially Connected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intersection</td>
<td>Union</td>
</tr>
<tr>
<td>( \tau_c = (a + s) )</td>
<td>( \tau_I = (a + s) + \chi )</td>
</tr>
<tr>
<td>( V_c = (3a + 2s + 1) )</td>
<td>( V_I = (3a + 2s + 1) + 3\chi )</td>
</tr>
<tr>
<td>(</td>
<td>P_{i\cap j}</td>
</tr>
<tr>
<td>( \Delta k(g, g + \gamma) \geq a )</td>
<td>( \Delta k(g, g + \gamma') \geq a + \chi )</td>
</tr>
<tr>
<td>( C = \gamma/\sigma )</td>
<td>( C = \gamma'/\sigma )</td>
</tr>
</tbody>
</table>

Table 1: Summary of Convergence Parameters

### 3.3.1 Mesh Networks

Three common mesh networks are shown in Figure 2, with degrees \( d = 4, d = 6, \) and \( d = 8, \) respectively. Since each node also receives its own messages, these degrees yield \( V = 5, V = 7, \) and \( V = 9, \) respectively. For each network, two nodes are selected and labelled \( i \) and \( j \) such that \( |P_{i\cap j}| \) is maximized. In figure 2, the nodes enclosed within a dashed box comprise \( P_{i\cap j} \) for that mesh.

Inspection of figure 2 reveals that in all three meshes \( \chi = 3 \) for nodes \( i \) and \( j \). Thus, in the best case of a fault-free system, \( V_I = 3\chi + 1 = 3 \times 3 + 1 = 10. \) Since \( V_I > V \) for all three meshes, there exists no MSR voting algorithm which is Intersection-Convergent for any of these systems.

In a fault free system \( V_U = \chi + 1 = 3 + 1 = 4. \) Thus, all three systems are Union-Convergent in the fault-free case. Assuming a single-fault scenario, the worst case would be if that fault were asymmetric, in which case \( V_U = 3a + \chi + 1 = 3 + 3 + 1 = 7. \)
Thus, in any single-fault scenario, both the Hexagonal and Octagonal meshes are Union-Convergent. It can also be shown that the Octagonal mesh can tolerate a double fault as long as $a \leq 1$, while none of these networks can tolerate a double asymmetric fault or any triple fault.

### 3.3.2 Hypercubes

In a hypercube, $d = \log_2(N)$, so that $V = \log_2(N) + 1$. Each node is connected to all nodes whose binary address is at a Hamming distance of unity from its own address. Thus, for any two nodes $i$ and $j$, $|P_{i \cap j}| \leq 2$. By definition, $\chi = V - |P_{i \cap j}| \geq V - 2 = \log_2(N) - 1$. The fault-free condition for Intersection-Convergence is $V_I = 3\chi + 1 = 3[\log_2(N) - 1] + 1$. Therefore, an Intersection-Convergent MSR algorithm exists for a hypercube only if:

$$
V \geq V_I,
$$

$$
V \geq 3\chi + 1,
$$

$$
\log_2(N) + 1 \geq 3[\log_2(N) - 1] + 1,
$$

$$
\log_2(N) \leq \frac{3}{2}.
$$
Since \( \log_2 (N) \) must be an integer, the largest hypercube for which an Intersection-Convergent MSR algorithm exists is defined by \( \log_2 (N) = 1 \), or \( N = 2 \). This is the trivial system comprised of two nodes connected by a single link.

Performing a similar analysis for Union-Convergence yields:

\[
\begin{align*}
V & \geq V_U, \\
V & \geq \chi + 1, \\
\log_2 (N) + 1 & \geq (\log_2 (N) - 1) + 1, \\
\log_2 (N) + 1 & \geq \log_2 (N).
\end{align*}
\]

Thus, any fault-free hypercube will be Union-Convergent. However, any single fault within \( P_{i,j} \) will add at least 2 to the right-hand side of this expression, making nodes \( i \) and \( j \) non-convergent.

4 GLOBAL CONVERGENCE

For most system applications, the goal of convergent voting is to achieve Approximate Agreement on a global level, i.e. to reduce the range of values held by all non-faulty processes. While local convergence is a necessary pre-condition to global convergence, it is not by itself sufficient to guarantee global convergence. Global convergence also depends on the topology of the system, the distribution of initial values, and the distribution of faults throughout the system. A complete and general treatment of global convergence is the subject of active research and is beyond the scope of this paper. Accordingly, this section is confined to a simple case study which demonstrates the issues involved, and the methodology to be used in analyzing global convergence properties.
4.1 Basic Definitions

Single-step local convergence does not necessarily produce single-step global convergence. At the global level convergence may be “eventual” rather than “immediate”. Let \( v^k_{\min} \) and \( v^k_{\max} \) be, respectively, the global minimum and maximum values held by non-faulty processes after \( k \) rounds of voting. Similarly, \( \delta^k \) and \( \rho^k \) are the global diameter and range of correct values after \( k \) rounds of voting. The corresponding initial parameters of the system are thus \( v^0_{\min}, v^0_{\max}, \delta^0 \) and \( \rho^0 \). Assuming \( \delta^0 > 0 \), global convergence occurs if there exists a finite positive value of \( k \) such that:

\[
\text{[G1]} \quad v^k_i \in \rho(v^0_{\min}, v^0_{\max}) \quad \forall \text{ non-faulty } i \in \{0, \ldots, N-1\},
\]

\[
\text{[G2]} \quad \delta^k < \delta^0.
\]

Global convergence is not a single-step process as is local convergence. It is therefore not appropriate to employ a single-step convergence rate as a performance measure. A better measure of performance is the asymptotic convergence rate \( C_\infty \). Assuming \( \delta^0 \neq 0 \), \( C_\infty \) describes the long term average convergence rate of the process, i.e.:

\[
C_\infty = \lim_{k \to \infty} \frac{1}{k} \delta^k.
\]

Unless the system is initially converged (\( \delta^0 = 0 \)), the two global extremes are distinct, i.e. \( v^0_{\min} \neq v^0_{\max} \). At least one of these extremes must be held by no more than half of the non-faulty processes. We define the minority extreme \( v_{me} \) as the extremal value which is held by the fewest non-faulty processes. If global convergence is possible, then the number of instances of the minority extreme will eventually decrease to zero. At that point there exists a new value of the minority extreme, and the process begins again. For process \( j \) we define:

\[
x_j = \begin{cases} 
0 & : v_j \neq v_{me} \\
1 & : v_j = v_{me}
\end{cases}
\]
Let $X_i$ be the number of minority extreme values received by process $i$:

$$X_i = \sum_{j \in P_i} x_j.$$ 

Finally, let $X$ = the total number of minority extreme values in the system:

$$X = \sum_{j=0}^{N-1} x_j.$$ 

If a system is convergent, then the number of minority extremes $X$ will eventually become zero. We define a zero-virus as a contiguous pattern of zeroes and ones such that the number of zeroes in and adjacent to the virus increases with each voting round. Thus, if a particular system does not converge such that $X = 0$, then there can be no zero-viruses anywhere in the system.

The choice of a voting algorithm can have a major impact on global convergence. For an MSR algorithm, $F(V) = \text{mean}(S)$. Thus, the voted value will equal $v_{me}$ only if $s_j = v_{me} \ \forall \ s_j \in S$. We define a Spanning MSR algorithm as one in which both the minimum and maximum values of the medial multiset $M$ are included in $S$, i.e. $\{m_1, m_M\} \subseteq S$. For a spanning algorithm, the voted value will equal $v_{me}$ only if $m_j = v_{me} \ \forall \ m_j \in M$. For process $i$, this situation can occur only if $X_i \geq V - \tau$.

### 4.2 Application Example

If a system fails to converge, then there can be no zero-viruses anywhere in the system. Proving that a system is globally convergent involves proving that a total absence of zero-viruses can occur only if the minority extreme is held by a majority of the nodes. Hence there is a contradiction, so the system must be convergent.

To demonstrate the method, a one-fault-tolerant spanning MSR algorithm is applied to a fault-free hexagonal mesh network. Figure 3 shows a $2 \times 2$ block of a hexagonal mesh in which processors are labeled by row and column. Assume that the network
contains $I$ rows and $J$ columns, where $N = I \times J$. Links at the periphery wrap around so as to create a homogeneous regular network. In this system, $V = d + 1 = 7$, and $\tau = 1$, so that $F(V_{i,j}) = v_{me}$ only if $X_{i,j} \geq 6$. It is not difficult to show that any $2 \times 2$ block which contains two or more zeroes is a zero-virus. The number of zero-viruses in the mesh can be bounded using a $4 \times N$ cyclic matrix:

$$X = \begin{bmatrix}
  x_{0,0} & x_{0,1} & \ldots & x_{i,j} & \ldots & x_{I-1,J-2} & x_{I-1,J-1} \\
  x_{1,0} & x_{1,1} & \ldots & x_{i+1,j} & \ldots & x_{0,J-2} & x_{0,J-1} \\
  x_{1,0} & x_{1,1} & \ldots & x_{i+1,j+1} & \ldots & x_{0,J-1} & x_{1,0}
\end{bmatrix}.$$ 

![Figure 3: 2 x 2 Block of a Hexagonal Network](image)

Each row of the matrix represents the entire network with the processors listed in row-major order. Thus, summing all values of $x_{i,j}$ along each row yields $X$. Since there are four rows in the matrix, the total number of ones in the matrix is $4X$. 

30
Each row is skewed from the previous row in such a manner that each column of the matrix represents one $2 \times 2$ block. The sum of a column is the number of ones in the block. If the system contains no zero viruses, then there are at least three one’s in each block, so the sum of each column is at least 3. Given $N$ columns, the number of ones in the matrix of a non-convergent network is thus $\geq 3N$.

The number of ones in the matrix is fixed regardless of the order in which they are counted. Thus, $4X \geq 3N$, which implies:

$$X \geq \frac{3}{4}N.$$ 

By hypothesis, $X$ is the number of minority extremes in the network, so that $X \leq N/2$. Hence by contradiction, the network must contain at least one zero-virus, and is therefore convergent.

The viruses used in this simple example are not fault-tolerant, i.e. they are not guaranteed to act as a virus if one or more values in or adjacent to the virus does not behave as expected. Convergence under a specific fault scenario depends on the existence of fault-tolerant zero-viruses. In a faulty system, there is also the possibility of a zero-vaccine, a cluster of faulty processors which prevents expansion or propagation of zero-viruses. The interactions of zero-viruses with zero-vaccines may be subtle, and require considerably deeper study.

5 SUMMARY AND CONCLUSIONS

The problem of reaching low-cost Approximate Agreement in large partially connected networks has been examined. The objective was to achieve convergence without the overhead of relaying messages between nodes. Two types of local convergence have been defined, intersection convergence and union convergence. Local convergence has been examined in the presence of both symmetric and asymmetric faults. As shown in Table 1, simple expressions have been derived for the convergence rate and robustness of any Mean-Subsequence-Reduced (MSR) voting algorithm. These
results permit convergent voting to be employed in very large sparsely connected systems, for which complete message broadcast is impractical.

The zero-virus method was demonstrated for determining the conditions under which global convergence will occur as a result of local convergence. Currently, fault-tolerant zero-viruses and zero-vaccines are being identified for various network topologies and fault scenarios. This work is permitting global convergence properties to be examined for various system topologies.

At this point, two extreme approaches are known for achieving approximate agreement. The conventional approach employs complete message relay to emulate a completely connected network, while the approach developed herein employs no message relays. The complete relay approach offers faster convergence, while the no relay approach imposes lower message passing overhead. Current research is examining the continuum of limited relay policies lying between these two extremes. For example, nodes in a system may relay only those messages originated by an immediate neighbor. This approach yields better robustness and convergence rates than the no relay approach, while still imposing much lower overhead than that of the complete relay approach. This work is based on the results presented herein.

References


