Continual On-Line Diagnosis of Hybrid Faults*

C. J. Walter, N. Suri and M. M. Hugue
AlliedSignal MTC
Columbia, MD 21045

Abstract

An accurate system-state determination is essential in ensuring system dependability. An imprecise state assessment can lead to catastrophic failure through optimistic diagnosis, or underutilization of resources due to pessimistic diagnosis. Dependability is usually achieved through a fault detection, isolation and reconguration (FDIR) paradigm, of which the diagnosis procedure is a primary component. Fault resolution in on-line diagnosis is key to providing an accurate system-state assessment. Most diagnostic strategies are based on limited fault models that adopt either an optimistic (all faults s-a-X) or pessimistic (all faults Byzantine) bias. Our Hybrid Fault-Effects Model (HFM) handles a continuum of fault types that are distinguished by their impact on system operations. While this approach has been shown to enhance system functionality and dependability, *on-line* FDIR is required to make the HFM practical. In this paper, we develop a methodology for utilization of the system-state information to provide continual on-line diagnosis and reconguration as an integral part of the system operations. We present diagnosis algorithms applicable under the generalized HFM and introduce the notion of fault *decay time*. Our diagnosis approach is based primarily on monitoring the system's message trafc. Unlike existing approaches, no explicit test procedures are required.

Keywords: on-line diagnosis, fault-effects, Byzantine faults

1 Introduction

Strategies which incorporate fault identication with high resolution are becoming an important feature of fault tolerant systems. These systems are designed to provide maximal system dependability at the lowest possible cost to the user. The ability to accurately diagnose faults is an important aspect of achieving this goal. In general, diagnosis has two primary objectives: (a) to identify a faulty unit so as to restrict the fault-effect on the system operations and (b) to

*Work supported in part by ONR # N00014-91-C-0014 support the fault isolation and reconguration process with information concerning the state of the system resources. The better the diagnostic resolution to identify a fault at the lowest level, the more efcient is the FDIR procedure, with minimal and very specic resources targeted for repair. This translates into an efcient management of the system resources, and the enhanced resolution can result in lower repair costs (cost of spares, time latency, fault-management costs). Also, if FDIR can be done during system operation, there will be more resources available for sustained system operations.

A variety of approaches have been proposed for system diagnosis - see survey in [B^{+}93], which have limited applicability for the following reasons. First, these methods are invariably off-line techniques due to the extensive overhead induced in collecting and interpreting the diagnostic information. Second, each approach is based on a particular system and fault model which inuences the amount of diagnostic resolution obtainable and determines the achievable degree of system dependability. Third, diagnosis is treated as an isolated objective in the system and not as a continual and integral phase of the system operations.

The dependability objective requires an accurate assessment of the state of the operational system, with imprecise assessment potentially resulting in either underutilization of resources or system failure due to incorrect assessment of the system-state. Off-line diagnosis procedures support the rst goal of identifying the faulty units in need of repair. The advantage of performing tests off-line is that interference with the system processing tasks does not occur. One major disadvantage in running the diagnosis off-line is that information is not available to the reconguration process which would make it more efcient. A second disadvantage is that transient or intermittent faults are not easily duplicated in an off-line environment. Actual stress placed on the system during active operation may be required to produce the fault, since it may be difcult to anticipate a test case or may be too expensive to simulate off-line. Also, these procedures impose a latency between the fault-occurrence and diagnosis phases, thus restricting the applicability of the diagnosis operations to actual system operations.

Although on-line diagnosis has some clear benets, it also has its challenges. The rst challenge is to implement a procedure with high resolution that does not severely degrade the performance of the system. Approaches, such as the PMC [PMC67] model (and variations - centralized and distributed), which rely on units explicitly testing each other to obtain syndrome information, are too costly in this regard and are invariably off-line procedures. Other methods, such as comparison testing [SD92] also have their shortcomings with the reliance primarily on comparison of test results for fault detection.

1.1 Diagnostic Approach

In [S^{+}92, Wal90], we showed that using a more rigorous fault model allows dependability to be more precisely and realistically evaluated for a given *fault set* instead of a single fault type. The stuck-at-X model usually results in overly optimistic evaluations as faults which exhibit less predictable manifestations are frequently present. The Byzantine model assumes worst-case behavior for each fault occurrence; this is a conservative model and results in a pessimistic assessment of dependability. The need is to nd a more realistic and exible fault model and to demonstrate its applicability in supporting fault-diagnosis.

The majority of diagnostic approaches assume permanent faults to prevent syndrome ambiguities due to arbitrarily test results from a node with a transient or intermittent fault. Thus, they are unable to handle Byzantine faulty nodes, which can send conicting information to testing nodes. In fact, it is shown in [SR87] that identication of a node that is Byzantine faulty is impossible using single pass syndrome collection. [SR87] also provide an off-line algorithm to tolerate Byzantine faults. In [Wal90], we have presented an on-line diagnosis and fault identication model that admits Byzantine faults.

We deviate from the conventional xed severity (both time-domain and data-domain) fault models to develop a composite and cohesive fault classication basis. Our model handles a continuum of fault and error types, classied according to their impact on system operations. Although this approach is shown to provide better assessments of operational indices, its practical usage requires an on-line FDIR process. In this paper we present diagnosis protocols, developing on our previous work. In [Wal90], we demonstrated that different error types could be detected, focusing on local detection primitives. We presented the basis for an on-line algorithm which handles mixed fault types. One important feature shown was that this approach could be implemented with low overhead. It was stated that in MAFT[W$^+$85] the overhead was $\leq$ 6% even though the algorithm executes continually. This method allows diagnosis to be directed to identifying the cause of detected errors and their impact on system resources. Also, understanding the cause and effect relationship, allows the reconguration process to be properly guided.

Our previous classication of detectable faults as either symmetric or asymmetric was from a low-level perspective on information ow in the system. We will provide a higher level classication from the standpoint of how detected faults affect the global state of the system based on the HFM. This reconciling of local views to a higher-level global view is accomplished by rst executing a distributed agreement algorithm to ensure consistent error syndrome information and penalty counts. This agreed upon information is used in the diagnosis phase by mapping low-level syndrome bits to high-level fault manifestation groups, such as benign, catastrophic, value, crash and Byzantine, which match the dependability model. Thus, we show that a more precise model for dependability can be constructed with signicant benets, supported by the requisite on-line algorithms. Our approach is not to treat diagnosis as a stand-alone function but to keep it as an integral and continually on-going phase of system operations. Thus, the algorithm is not scheduled, but uses the system communication trafc as test stimuli. If necessary, specic tests can be run to handle episodic faults.

In section 1.2 we describe the HFM and system model. We present the progressively rened diagnosis algorithms in section 1.3. The formalism for the diagnosis algorithms are based on the Hybrid Oral Messages (HOM) algorithm, whose properties are also discussed. In section 1.4 we introduce the issue of the temporal behavior of faults through a notion of fault decay time.

2 System and Fault Model

We consider a distributed system framework, comprising *nodes* that communicate using a synchronous deterministic message passing protocol. Individual nodes make decisions and compute values based on information received in messages from other nodes. We consider the system communication model as:

A1: A direct path exists from each node to all other nodes. A node issues a single message which is broadcast to all connected nodes.

A2: A non-faulty node can identify the sender of an incoming message, and can detect the absence of an expected message.

A3: Node and link faults are considered indistinguishable (This assumption is later alleviated in Alg. HD).

Our goal is to identify faulty nodes and to prevent system failure in the presence of a specied number of faults. The system fails when consistent decisions or computations across the system are no longer possible. Unlike previous work, we distinguish faults by their behavior, and place no limitations on the duration of a fault. Under this framework, the sole indicator of a faulty node is an error in its transmitted message, as viewed by all receivers of the message. The receiver acquires a *local view* of the sending node's health by applying fault detection mechanisms, such as range and framing checks, to the incoming message. The exchange of local views among receiving nodes is then used to acquire a global perspective of the effects of a faulty unit.

2.1 Fault Model

Our interest is in considering faults with unrestricted behavior. In this paper, we adopt the Hybrid Fault-Effects Model (HFM), in which faults are classied according to the fault manifestations they present across the system. We provide a two-fold motivation for using HFM as linked to the diagnosis objective.

First, we are basing diagnosis on consensus algorithms. These algorithms are very robust but also expensive (time and space complexity) to implement. Designed to provide coverage to the worst case Byzantine faults, these assume all fault occurrences to be Byzantine faults and require a node-redundancy of $N > 3m$ to cover m faults. The HFM assumes perfect coverage to a limited number of arbitrary faults, but recognizes that weaker fault types are typically more common than the classical Byzantine faults. The algorithms under the HFM do not compromise the system's capability of tolerating Byzantine faults, however they provide additional and concurrent coverage to *fault sets* of weaker manifestations. This also facilitates a higher resolution of fault granularity compared to considering all faults of a single fault severity. The distinction here is that hybrid faults can be of any type, as long as they can be tolerated by the system implementation *without causing system failure*. If the system is destroyed, or a sufciently large portion of the system is damaged, then the issue of diagnosis becomes moot.

Second, in HFM it is important to note that the fault classes are *disjoint*, preventing any ambiguities in discerning the fault behavior and effect and subsequent diagnostic ambiguities. Furthermore, as HFM considers fault classication based on the *effect* the fault causes to the system operation, it provides an uniform framework to handle both time-domain and data-domain faults.

Under the HFM, classication is based on **(1)** *Local-Classication* of fault-effects to the extent permitted by the fault-detection mechanisms built in at the node level, and **(2)** *Global-Classication* based on nodes exchanging their local-classication with other system nodes to develop a global opinion on the fault-effect.

HFM classies fault-effects into two types: *benign* and *value* (or *malicious*) faults. *Benign or non-malicious* are detectable by all non-faulty nodes within the scope of the fault detection mechanisms implemented *locally* in each node. Examples of *benign* faults consist of locally discernible effects such as bit garbling, framing errors, range violations, message omission or early/late message delivery. Interestingly, the *benign* fault set encompasses a number of fault classes dened previously as timing, omission, crash and stopping faults. These faults can still be classied individually under the HFM. Any detectable fault in a message will result in the sender being classied as *locally benign faulty*-**(b)** by the receiver.

Value or malicious faults are the locally undetectable class of faults. Messages which pass all data validity and range deviance checks, but provide a valid but *incorrect* value constitute *value* faults. Essentially, none of the constraints on good values or messages are violated. For example, if the valid range for an expected datum value is [0 - 100] and the expected *correct* value is 50, but instead the value 75 arrives, the node, locally, cannot discern any discrepancy.

A further classication for both *benign* and *value* faults is of symmetry. Suppose at least one non-faulty node cannot detect a node fault, and the same fault is detected by every other good node. The sending node is thus asymmetric value faulty, but is identied as *locally benign* by the nodes that can detect it, showing that the local view is not accurate. Thus, exchange and accumulation of global information across several time periods is often necessary to diagnose such faults. Value faults are also partitioned into *symmetric*-**(s)** and *asymmetric*-**(a)** value faults, according to their presentation throughout the fault scope. It should be noted that diagnosis of nodes as symmetric or asymmetric faulty requires a global perspective, or several rounds of message exchange to emulate one. Byzantine agreement [L+82] algorithms are often implemented to ensure that distributed nodes or processes arrive at the same decisions and computational results in the presence of arbitrary faults. However, with simple modication of existing fault detection and masking techniques, correct computations can be also be guaranteed under the HFM. Unlike the $N > 3m$ redundancy requirement of the classical Byzantine models, N nodes will tolerate $> 3a + 2s + b$ composite *(a, b and s)* faults under the HFM. We extend the classical Byzantine fault model to address mixed fault types, while ensuring correct system operation in the presence of faults. Instead of assuming that all faults are *arbitrary*, we adopt the hybrid fault taxonomy and handle faults according to the errors they produce.

3 HOM - Agreement under HFM

Our diagnosis protocols are based on the Exact Agreement or Consensus paradigm. Prior to developing the diagnostic procedure, we rst need to demonstrate that exact agreement is indeed possible under the HFM. Further, it is also required to demonstrate the stability and correctness of the diagnosis procedure in the presence of faults. This is unlike the off-line techniques which assume the sanity of diagnosis in a fault-free scenario. Both of these facets are handled by the HOM Agreement algorithm.

3.1 HOM(a) Algorithm

The *Oral Messages Algorithm* (OM) [L$^+$82] demonstrated that Agreement can be guaranteed if $n > 3t$ and $r \geq t$ where t is the number of Byzantine faults, n is the number of nodes, and r is the number of rebroadcast rounds. However, it makes the pessimistic assumption that all faults are asymmetric value faulty. This implies that a 4, 5, or 6 node system can tolerate only a single arbitrary fault, and that 7 nodes will be required to tolerate two faults. Note that the model covers only 2 faults, not necessarily 2 *arbitrary* faults. In contrast, the HFM covers both Byzantine and non-Byzantine faults under a single framework, without increasing the complexity of the underlying algorithms.

The *Hybrid Oral Messages* (HOM(a)) -Table 1- is a r-rebroadcast round protocol based on the OM[L$^+$82] algorithm, requiring $n > 2a + 2s + b + r$ nodes to mask $(a + s + b)$ faults, where b nodes are benign faulty, s nodes are symmetric value faulty, a nodes are asymmetric value faulty, and $a \leq r$. As $n > 2a + 2s + b + r$ indicates, several composite *(a,b,s)* fault scenarios are covered by HOM(a), not just a xed number of t arbitrary faults.

The algorithm $Z(r)$ of [TP88] tried to address the issue of consensus and composite fault types. However, as it was found to be incorrect [LR93], we present a hybrid algorithm that achieves Agreement under the assumption of hybrid faults, satisfying the Hybrid Agreement conditions HOM1 and HOM2:

HOM1 (Validity): If the Transmitter is *non-faulty*, then all non-faulty Receivers select the sender's original value. If the Transmitter is *benign faulty*, then all non-faulty Receivers will adopt a default value, $\mathcal{E}$. If the Transmitter is *symmetric faulty*, then all non-faulty Receivers will adopt z, the sender's original value.

HOM2 (Agreement): All non-faulty Receivers agree on the value of the Transmitter.

The HOM(a) uses a family of error values, $\{\mathcal{E}, R(\mathcal{E}), \ldots, R^r(\mathcal{E})\}$, where r is the number of rebroadcast rounds, to incorporate the HFM. If the value $R^k(\mathcal{E})$ is received, where $k \geq r - l$ in S2 of HOM($a - l$), then that too is recognized as an error, and $\mathcal{E}$ should be adopted. The function HOM-*maj*, used by HOM(a), is as follows. Given a set V of k values, $v_i, \ldots, v_k$, HOM-*maj*(V) is given by

Table 1: HOM(a) Algorithm

S1: The Transmitter sends its personal value, v, to all receivers.

S2: $\forall i$, let v_i denote the value that Receiver i gets from the Transmitter.

If $a = 0$, and a benign-faulty**(b)** value is received, Receiver i adopts $\mathcal{E}$. Otherwise, Receiver i adopts v_i. The algorithm then terminates.

If $a > 0$, each Receiver adopts $R(\mathcal{E})$, if a benign-faulty**(b)** value is received, and $R(v_i)$ otherwise. Each receiver then acts as the Transmitter in Algorithm HOM($a-1$) sending its personal value to the other $N-2$ nodes.

S3: $\forall i, j$, with $i \neq j$, let v_j denote the value Receiver i gets from sender j in Step 2 of HOM($a-1$). If no message is received or v_j is obviously incorrect, Receiver i adopts $\mathcal{E}$ for v_j; otherwise, v_j is used.

Since all Receivers act as senders in HOM($a-1$), each Receiver will have a vector containing (N-1) values at the end of HOM($a-1$). Receiver i adopts v=$HOM-maj(v_1, v_2, \ldots v_{N-1})$ as the Transmitter's value.

$$\text{HOM} - maj((V)) = \begin{cases} \mathcal{E}, & \text{if all of the } v_i \text{ satisfy } v_i = \mathcal{E}. \\ R^{-1}(v_{\mathcal{E}}), & \text{if } v_{\mathcal{E}} = maj(exclude(V, \mathcal{E})) \text{ exists.} \\ v_0, & \text{otherwise.} \end{cases}$$

The provision which adopts $\mathcal{E}$ if all the v_i are $\mathcal{E}$ cannot occur on a good node, but is included to provide a fail safe default value should that case occur on a partially faulty node.

3.2 Algorithm HOM(a) - Properties

We defer detailing the proof of the algorithm [LR93] and limit our discussion to stating the relevant Lemmas and properties of interest which help formalize the diagnosis objective.

Lemma 1 *Algorithm HOM(a) achieves Validity (***HOM1***) for any integers $a, b, s, N, r \geq 0$, such that $N > 2a + 2s + b + r$, and $a \leq r$.*

Lemma 2 *If $N > 2a + 2s + b + r$, for any $N, a, s, b \geq 0$, $r > 0$ and $a \leq r$, then HOM(a) satises Agreement (***HOM2***).*

Taken together, Lemmas 1 and 2 prove the following theorem.

Theorem 1 *Algorithm HOM(a) achieves Byzantine Agreement when $N > 2a + 2s + b + r$, for any $r \geq 0$, any $a \leq r$, any $s \geq 0$, and any $b \geq 0$, where a is the number of asymmetric value faults, s is the number of symmetric value faults, and b is the number of benign faults in the system during execution of HOM.*

The HOM algorithm assured agreement on a specied information value. The extension to obtaining exact agreement on the values of all constituent nodes leads to the Hybrid Interactive Consistency (HIC) problem. The HOM forms the proof basis for the HIC algorithms too. We briey mention the requirement for HIC.

Algorithm 1 *(HIC(a)) Let S be the set of nodes holding values upon which HIC is desired, with $|S| = N$. Each node sends its private value to all other nodes in S, acting as the transmitter in HOM(a), with the value of r identical for all nodes.*

At the conclusion, each good node in S will hold a nal vector which satises the following conditions.

HIC1 (Validity): Each element of the nal vector that corresponds to a non-faulty node is the private value of that node. Each element of the nal vector that corresponds to a benign faulty node is $\mathcal{E}$.

HIC2 (Agreement): All non-faulty nodes compute the same vector of values.

Theorem 2 *Algorithm HIC(a) achieves Agreement when $N > 2a + 2s + b + r$, for any $r \geq 0$, any $a \leq r$, any $s \geq 0$, and any $b \geq 0$, where a is the number of asymmetric value faults, s is the number of symmetric value faults, and b is the number of benign faults in the system during execution of HOM.*

Proof: By Theorem 1, at the conclusion of HOM(a) with node i as the transmitter, HOM1 and HOM2 will be satised. Thus, each good node will have a consistent view of node i's personal value. All nodes then execute HOM(a). By the proof of agreement and validity for HOM, if $N > 2a + 2s + b + r$, and all nodes act as the Transmitter in HOM(a), with $a \leq r$, and all nodes using the same value of r, then conditions **HIC1** and **HIC2** will be satised. Since i is arbitrary, this guarantees that all good nodes will hold the same vector of values, either a good node's original value or an agreed upon default value for a faulty node's original value. □

4 On-Line Diagnosis

In distributed diagnosis, each non-faulty node in the system is expected to cooperate with other non-faulty system nodes in detecting, identifying and removing faulty nodes from the system. In a typical diagnostic procedure, system components are tested and the results, or *syndromes*, are collected and analyzed by all active components. At a minimum, the status of a node (faulty or good) should be agreed upon by all non-faulty nodes. Once a faulty node has been identied, the decision to add or to exclude a node should also be consistent in all non-faulty components. Essentially, we desire a complementary usage of consensus and diagnosis facets to provide a 2-phase approach: *Phase(a):* local diagnosis, followed by *Phase(b):* global diagnosis based on consensus principles.

Table 2: Algorithm DD for Node i : Basic Consensus Syndrome

DD0. Initialize s_i^k to be the zero health vector.

DD1. *local detection*: Node i monitors message trafc from all other nodes throughout $\mathcal{D}(k)$. If a benign-faulty message is received from from node j during diagnosis interval k then set $\sigma_{ij}^k = (\sigma_{ij}^k$ OR 1), thus forming the local k^{th} round health vector s_i^k.

DD2. *Global assimilation*: Node i collects health vectors s_i^{k-1} computed during diagnosis interval $\mathcal{D}(k-1)$ into the $n \times n$ global syndrome matrix $S_i^{(k-1)}$. Row l of $S_i^{(k-1)}$ is the syndrome vector $s_l^{(k-1)}$ received from node l.

DD3. *Syndrome Matrix/Diagnosis:* Combine the values in column j of $S_i^{(k-1)}$, corresponding to other nodes' views of the health of node j, by using a hybrid voting function to generate a consistent health value, faulty or non-faulty, for node j. Node exclusion/inclusion as per consensus based heath value.

DD4. At the end of diagnosis interval k, node i sends its health vector, $s_i^k =< \sigma_{i1}^k, \sigma_{i2}^k, \ldots \sigma_{in}^k >$, to all other nodes.

Our primary goal in on-line diagnosis is to identify faulty system nodes while maintaining correct system operation. A node's analysis of messages received from another node, as well as messages expected but not received, constitutes implicit testing of the sending node. We dene diagnosis intervals, in which the following primitives are executed: *local detection and diagnosis*, *global information collection* and *global diagnosis* { on a concurrent, on-line and continual basis. The information collected locally by each node during diagnosis interval k, $\mathcal{D}(k)$, is broadcast to all other nodes, which then collect and analyze the information during $D(k+1)$, to formulate a global perspective on the fault behavior. The length of the diagnosis interval is bounded by the assumed frequency of asymmetric value faults in the system.

We present the diagnosis algorithms on a progressive basis. Algorithm **DD** (*Distributed Diagnosis*) represents the basic two-phase, on-line diagnostic approach linking the diagnosis and consensus procedures. In Algorithm **HD** (*Hybrid Diagnosis*), we extend on **DD** to provide the capability of discriminating between node and link faults, where possible; to assess the severity of the node fault from temporal perspectives, and to incorporate the facets of node recovery and re-integration.

4.1 Distributed Diagnosis: Node Health Monitoring

The goal of algorithm **DD**, shown in Table 2, is for all non-faulty nodes to acquire a consistent view of the health of all other nodes in the system. This is done by an exchange of local health assessments and subsequent computation of a consistent global health vector, using the HOM algorithm. It is pertinent to mention that the diagnosis algorithm running during diagnosis interval $\mathcal{D}(k)$ utilizes information collected across the system over the previous round $\mathcal{D}(k-1)$. All operations can be considered to be on-going in a pipelined manner.

Inter-node messages are considered as the sole indicators of the health of a node. Based on this, step **DD1**, the local detection phase, examines all received messages for errors. Since the detection of an error in a message by its receiver implies that the sender is *locally benign faulty*, local detection utilizes the parity checks, checksums, message framing checks, range checks, sanity checks, and comparison techniques. The failure to receive an expected message from a node or an early/delayed message is also logged as an error for that node.

During $\mathcal{D}(k)$, each node i locally formulates a health vector, $s_i^k =< \sigma_{i1}^k, \sigma_{i2}^k, \ldots, \sigma_{in}^k >$, containing an entry, σ_{ij}^k, corresponding to the perceived status of each system node, j. If any error is detected from a given node, j, its entry σ_{ij}^k is set to 1; otherwise it remains at the fault-free value of 0. This local diagnosis step is equivalent to the identity mapping, as no further local diagnosis occurs following detection.

To achieve consistency of this local diagnosis, and to build the global perspective to handle the \locally undetectable" class of *value* faults, information dispersal across the system is necessitated. At the end of each detection interval, in step **DD4**, the local health vector s_i^k of node i is sent to all other nodes. Thus, each node compiles (and analyzes) a global syndrome matrix during $\mathcal{D}(k+1)$ that contains the local health assessments of every node by other nodes over $\mathcal{D}(k)$.

During $\mathcal{D}(k)$, global diagnosis of each node's health during $\mathcal{D}(k-1)$ is performed in step **DD3**. The local health vectors computed during $\mathcal{D}(k-1)$ form the rows of the global health matrix $S_i^{(k-1)}$. If no health vector or an obviously erroneous vector is received from node l, then an error indicator value, $\mathcal{E}$, is adopted for each $\sigma_{lj}^{(k-1)}$ in $s_l^{(k-1)}$, and node l is assessed for an error by updating σ_{il}^k. The global health vector held by each non-faulty node i is denoted by $h_i^{(k-1)}$, with entries $\eta_{ij}^{(k-1)}$ giving the global status of node j during $\mathcal{D}(k-1)$. The global health vector for $\mathcal{D}(k-1)$ is computed during $\mathcal{D}(k)$ by applying a hybrid majority voting function, described below, to each column of $S_i^{(k-1)}$.

First, all elements of column j equal to $\mathcal{E}$ are excluded, along with node j's opinion of itself ($\sigma_{jj}^{(k-1)}$). The nal value, $\eta_{ij}^{(k-1)}$, is the majority of the remaining values. If no majority exists, the value 0 should be adopted to ensure that a good node is not identied as faulty. At the conclusion of $\mathcal{D}(k)$, each good node will contain a global health vector $h_i^{(k-1)} =< \eta_{i1}^{(k-1)}, \eta_{i2}^{(k-1)}, \ldots, \eta_{in}^{(k-1)} >$, where $\eta_{ij}^{(k-1)} = 1$ means that node i has diagnosed node j as being faulty during $\mathcal{D}(k-1)$. It needs to be stated that this diagnosis is consistently achieved by all non-faulty nodes, as this is the Agreement condition HIC1.

4.2 Hybrid Diagnosis: Fault Identication and Severity

In **DD**, an error in a single message from a node during a single diagnosis interval is sufcient to cause that node to be removed from the system. The local detection mechanism described previously for DD1 treats a node that sends a single erroneous message as if all messages from the node were indeed faulty. Essentially, a transient and a permanent fault will have an identical fault effect here. This is not an efcient strategy, and could lead to rapid depletion of system resources. Further, **DD** provides only the fault detection and isolation facets of FDIR. Recovery of a faulty node or node re-integration require rened temporal considerations, which the **DD** does not fully support. These issues, and increasing the diagnostic resolution, are dealt

Table 3: Fault Classication under HD

Type		Recorded in:
$\mathcal{MM}$	Missing Message	ϵ^k_{ij1}
$\mathcal{IFM}$	Improperly Formatted Message	ϵ^k_{ij2}
$\mathcal{ILM}$	Improper Logical Message	ϵ^k_{ij3}
$\mathcal{CVM}$	Failing Comparison to Voted Value	ϵ^k_{ij4}

with in **HD** (Table 4), building on the basic framework of **DD**. We also add temporal fault detection to the local diagnosis primitive, and replace the simple good{bad local diagnosis with a preliminary assessment of the node and/or link fault symmetry.

4.2.1 Local Primitives in HD

The scalar status value, σ^k_{ij}, used in step DD1 of the previous algorithm, corresponding to node i's local assessment of node j's health during $\mathcal{D}(k)$, is replaced in step HD1 by a local diagnosis vector, e^k_{ij}. This local diagnosis vector is used to indicate the type of detection mechanism which found errors in messages from node j, providing a preliminary diagnosis of the fault type. While the entries in $e^k_{ij} =< \epsilon^k_{ij1}, \epsilon^k_{ij2}, \ldots, \epsilon^k_{ijm} >$ can have a one-to-one correspondence with the m fault detection mechanisms implemented in the system, as in MAFT [W$^+$85, Wal90], we consider four potential diagnoses shown in Table 3.

We assign a penalty weight to each error type, commensurate with its assumed severity in the system implementation, and accumulate the weights for each node over $\mathcal{D}(k)$. By denition, these detected errors result from *benign* faulty nodes. However, discerning the potential symmetry of the errors is useful in discriminating between a crash faulty node and a less severe faulty communications link. The relationships among ω_{MM}, ω_{ILM}, ω_{IFM}, ω_{CVM} forms the basis of inferences on fault-type. Of course, more or fewer weights could be used. Also, an additional correlated weight can be used if a faulty node exhibits several of these behaviors during a single diagnosis interval. The nal extended health vectors and accumulated penalty weights from $\mathcal{D}(k)$ are sent to all nodes at the end of $\mathcal{D}(k)$.

4.2.2 HD: Global Diagnosis/Properties

During $\mathcal{D}(k)$, the extended health vectors and penalty weights from all nodes during $\mathcal{D}(k-1)$ are analyzed in a fashion similar to that in DD. Steps HD3 and HD4 ensure a consistent global perspective on the cumulative penalty values associated with each and every system node following global information collection in HD2. The overall relative value of a fault type e.g., $\epsilon_{ij1} > \epsilon_{ij2}$'s, etc are useful in attempting to identify the type of fault. The penalty weight for each node under diagnosis is its initial value in HOM(1). The asymmetric fault coverage is limited to one fault by the single re-broadcast round.

Since behavior of a faulty node is unrestricted, and a faulty node can send different corrupt health vectors (or none) to other nodes, good nodes may receive different health matrices. So, we must prove that the nal health vectors h_i computed by all good nodes i during each diagnosis interval are consistent. Furthermore, we must assess the correctness and completeness of diagnosis. Global diagnosis is *correct* for $\mathcal{D}(k-1)$ if each node identied as faulty by a good node in step **HD3** of HD (during $\mathcal{D}(k)$) is indeed faulty. Similarly, global diagnosis is *complete* for $\mathcal{D}(k-1)$ if all nodes that were faulty during $\mathcal{D}(k-1)$ are identied as such in step HD3 in that diagnosis interval.

While the statement of algorithm *HD* does not explicitly invoke any fault tolerance algorithm, it is implicit in the denition of HD and in the fault masking used in the the **HIC(a)** algorithm. The local health of node j as viewed by node i during $\mathcal{D}(k-1)$, assessed as either 0 or 1, is equivalent to node j holding a personal value of either 0 or 1 and transmitting it to node i during $\mathcal{D}(k-1)$. This corresponds to the initial round of the **HOM(a)**, with $r=1$. The sending by node i of its local assessment of node j's health to other nodes at the end of $\mathcal{D}(k-1)$ represents the rebroadcast round of HOM(1), with the hybrid majority column vote of $S_i^{(k)}$ during interval $\mathcal{D}(k)$ equivalent to the nal value for j as computed by node i in HOM(1).

Since *HD* is executed on all nodes, and each node i monitors all other nodes, completion of *HD* is equivalent to all good nodes achieving interactive consistency during $\mathcal{D}(k)$ on the health of all other nodes during $\mathcal{D}(k-1)$. Theorems 1 and 2 provide conditions under which the global health vectors h_i^{k-1} are guaranteed to be consistent on all good nodes. These theorems also permit us to prove the following correctness and completeness results.

Theorem 3 (Correctness) *If $n > 2t-b+1$, where $t = a+b+s$ faulty nodes are present during both $\mathcal{D}(k-1)$ and $\mathcal{D}(k)$, then diagnosis under* Algorithm HD *during $\mathcal{D}(k)$ is guaranteed to be correct for $\mathcal{D}(k-1)$.*

Proof: By Theorem 1, with the default majority value set to 0, all good nodes will have $h_{il}^{(k-1)} = 0$ for the values of all good nodes l. Thus, any node j for which $h_{ij}^{(k-1)} = 1$ must be faulty. □

Theorem 4 (Completeness) *If $n > 2t-b+1$ as dened above, and node j was faulty during $\mathcal{D}(k-1)$, then under* Algorithm HD, *node j will be diagnosed during $\mathcal{D}(k)$ as having been faulty during $\mathcal{D}(k-1)$.*

Proof: By denition, a node that is benign faulty during $\mathcal{D}(k-1)$ will be detected locally on each good node. Thus, the sender's initial value for these nodes is 1, with all good nodes adopting 1 (faulty) as the local diagnosis during $\mathcal{D}(k-1)$, and agreeing on 1 at the conclusion of $\mathcal{D}(k)$, by Thm 1. □

This covers the cases for data and node faults. Following the formalism of the HD algorithm, we now motivate the fault resolution and the temporal aspect of aggregating the penalty weights in steps HD1 and HD3.

4.3 HD: Temporal Perspectives

Algorithm HD improves the judgment of fault severity at any interval in time so that units with less severe fault indications are left operational. Additional processing is required in HD as we are interested in handling fault-effects over a longer period of time than the diagnostic interval. We can build on the results obtained by instantaneous diagnosis(DD) by placing them into a temporal framework(HD).

Errors can be viewed as the manifestation of faults which exist in the system. *Duration* is dened as the total time a fault and its effects are present in the system during actual operation. We can introduce the concept of *decay-time* to be the length of time an error would be present if the fault was instantaneously applied and removed. Thus, the error is the effect of an instantaneous fault injected at time t_0 which lasts for a time t_f. The concept of *decay-time* allows error information to be carried across multiple intervals so that instantaneous diagnosis information can be related over time.

It must be noted that the *decay-time* does not always correlate directly with the severity level of the error. If a function in the system hard core is impacted by the transient fault, it may be necessary to immediately deal with the effects rather than waiting for them to die out. This method can account for the possibility that errors may propagate due to lack of containment and cause other errors which have their own *decay-time* and severity.

Errors which have shorter *decay-times* will have less time to further impact system operation. For example, a lost bit on a communication link due to a transient fault should be considered as an error with a short *decay-time*. If a noise pulse affects the link, some time will need to pass before the energy is dissipated from the medium. During this time, the messages being sent may be corrupted, depending on the level of noise. Another example would be a memory module with scrubbing. When an an error occurs, there will be a time period where the error could propagate and induce further errors. Once the scrubbing mechanism detects the error and removes it, the immediate danger of error propagation will have lapsed (even though the faulty source may still be present and/or intermittent).

Decay rates can therefore be determined if there exist regular and predictable times where errors can be detected or removed. If the times of detection are not regular, it is prudent to assume a worst case scenario which can be arrived at in a number of ways. The rst approach would be to assess a penalty so severe it causes exclusion immediately so that further reliance on fault detection is not needed. A second method would be to attempt to identify the worst case detection time by a higher level mechanism. This method may allow for some error propagation until it begins to affect a critical higher level function. A third alternative is to schedule more extensive FDIR tasks to attempt to collect more information while imposing greater overhead on the system.

Based on the concept of *decay-time*, we can assume that if the fault is applied and then goes away immediately, the fault-effects should only last for a certain period of time. Faults with their associated *decay-times* and handled as follows. First, a penalty weight (W_e) should be assessed against the faulty unit in the interval $\mathcal{D}$(k). Second, during the following diagnostic intervals, for the node displaying sustained fault-free behavior, the penalty weight against it is reduced by a predetermined amount, referred to as the *decrement-count* (DC). The ratio, W_e/DC provides the *decay-time* for the given error.

This *decrement-count* is introduced so that temporary malfunctions do not result in permanent exclusion. Note, that if the fault persists, penalties will continue to accrue and the decrement amount will be offset by new increases in penalty weights. The duration aspect of faults is also handled in this model. For the fault being transient, the source of the fault is removed and the effects should disappear after the appropriate decay time has passed. For a permanent fault, the source of the fault will remain present and new penalties will continually accrue over each new diagnostic interval until an exclusion threshold is reached. In order to handle transient and intermittent faults which are severe enough to cause exclusion but allow re-admission, the exclusion and re-admission thresholds must be appropriately separated.

This approach alternately supports modeling of permanent faults by setting the decrement count to zero. If one wants to support graceful re-admission of system units with on-line repair, even permanent faults can have a relatively short *decay-time*. The decay time can also be based on the time of re-admission, since once the unit is repaired its count must be decremented for it not to appear faulty anymore.

5 Discussion

Most existing diagnosis strategies treat diagnosis as a stand-alone process in the system operations, and are primarily off-line techniques. In this paper, we have addressed the problem of performing on-line diagnosis as an integral phase of the system FDIR process. Unlike existing approaches, the strategy is based on monitoring the system message trafc rather than using explicit test procedures.

Extending beyond the xed fault severity models (time-domain and data-domain, s-a-X, Byzantine faults), the HFM framework is developed, which permits handling a continuum of fault types as groups of faults of varying fault manifestations under a single algorithm. The HFM's applicability to diagnosis is formally shown through the development of a distributed agreement algorithm (HOM) and diagnosis algorithms DD and HD. The integration of HFM into the diagnosis domain facilitates increased diagnostic resolution which can be used for improved resource management strategies.

The paper has also introduced the concept of fault *decay-time* and its impact in handling transient, intermittent and permanent faults in conjunction with Byzantine faults. The usage of penalty counts is shown as a basis for graceful node exclusion and re-admission protocols. In future work a detailed penalty count model will be developed. Overall we have shown that a more precise dependability model can be constructed, supported by on-line diagnosis algorithms under a generalized hybrid fault model.

References

[B$^+$93] M. Barborak et al. The consensus problem in fault-tolerant computing. *ACM Computing surveys*, 25(2):171-220, June 1993.

[L$^+$82] L. Lamport et al. The byzantine generals problem. *ACM Trans. on Prog. Languages and Systems*, 4:382-401, July 1982.

[LR93] P. Lincoln and J. Rushby. A formally veried algorithm for interactive consistency under a hybrid fault model. *FTCS-23*, pages 402-411, 1993.

[PMC67] F. Preparata, G. Metze, and R. T. Chien. On the connection assignment problem of diagnosable systems. *IEEE Trans. on Electronic Computing*, ec-16:848-854, Dec 1967.

[S$^+$92] N. Suri et al. Reliability modeling of large fault-tolerant systems. *FTCS-22*, pages 212-220, 1992.

[SD92] A. Sengupta and A. Dahbura. On self-diagnosable multiprocessor systems: diagnosis by the comparison approach. *IEEE TOC*, 41(11):1386-1396, Nov 1992.

[SR87] K. G. Shin and P. Ramanathan. Diagnosis of processors with byzantine faults in a distributed computing system. *FTCS-17*, pages 55-60, June 1987.

[TP88] P. Thambidurai and Y. K. Park. Interactive consistency with multiple failure modes. *Proc. of RDS*, pages 93-100, 1988.

[W$^+$85] C. J. Walter et al. MAFT: A multicomputer architecture for fault-tolerance in real-time control systems. *RTSS*, Dec 1985.

[Wal90] C. J. Walter. Identifying the cause of detected errors. *FTCS-20*, June 1990.

Table 4: Algorithm HD for node i : Rened Syndrome

HD0. The expanded health vector, $s_i^k =< e_{i1}^k, e_{i2}^k, \ldots, e_{in}^k >$, is initialized to zero at the beginning of $\mathcal{D}(k)$; restore penalty weight vector $p_i^k =< \rho_{i1}^k, \rho_{i2}^k, \ldots, \rho_{in}^k >$ from D(k-1).

HD1 *local diagnosis:* Node i monitors message trafc from all other nodes. The following steps for each received message from each node j, and for each expected message not received from each node j.

HD1.1 If no message ($\mathcal{MM}$) is received from node j, then set $\epsilon_{ij1}^k = \max(\epsilon_{ij}^k, 1)$ and $\rho_{ij}^k = \rho_{ij}^k + \omega_{MM}$.

HD1.2 If an incorrectly formatted message ($\mathcal{IFM}$) is received from j, then set $\epsilon_{ij2}^k = \max(\epsilon_{ij2}^k, 1)$ and $\rho_{ij}^k = \rho_{ij}^k + \omega_{ILM}$.

HD1.3 If an incorrect logical message ($\mathcal{ILM}$) is received from node j, then set $\epsilon_{ij3}^k = \max(\epsilon_{ij3}^k, 1)$ and $\rho_{ij}^k = \rho_{ij}^k + \omega_{IFM}$.

HD1.4 If an incorrect value ($\mathcal{CVM}$) is detected in a message from node j by comparison to a voted value, j, then set $\epsilon_{ij4}^k = \max(\epsilon_{ij}^k, 1)$, and $\rho_{ij}^k = \rho_{ij}^k + \omega_{CVM}$.

HD2. *Information assimilation*: Node i collects the health vectors computed during $\mathcal{D}(k-1)$ into the $n \times n$ syndrome matrix $S_i^{(k-1)}$, where row l of $S_i^{(k-1)}$ is the syndrome vector $s_l^{(k-1)}$ received from node l. Similarly, the penalty counts computed during $\mathcal{D}(k-1)$ are collected into the matrix $P_i^{(k-1)}$, where column j of $P_i^{(k-1)}$ contains the weights received from all nodes regarding node j.

HD3. *Diagnosis: Consensus*: Combine the values in column j of $S_i^{(k-1)}$ to generate a health value, faulty or good, for node j by rst OR-*ing* the entries in $e_{ij}^{(k-1)}$, and then performing a hybrid majority vote down the column. Combine the values in column j of $P_i^{(k-1)}$ to generate a consistent incremental penalty count for node j during $\mathcal{D}(k-1)$ using a hybrid voting function. *Threshold Matching: Inclusion/Exclusion*

HD4 At the end of diagnosis interval k, node i sends its health vector, $s_i^k =< \sigma_{i1}^k, \sigma_{i2}^k, \ldots \sigma_{in}^k >$, and its penalty count vector, $p_i^k =< \rho_{i1}^k, \rho_{i2}^k, \ldots, \rho_{in}^k >$, to all other nodes.