Idea Paper

Unmasking Faulty Participants in a Byzantine Network sans Omission Faults

Damon Bruccoleri

Nova Southeastern University

Keeping all the subsystems in a modern aircraft in 'agreement' is a critical task (Wensley, et. al., 1978). For instance a critical aircraft subsystem could give conflicting data to other subsystems during a landing maneuver. Perhaps a faulty processor reports a failure to one subsystem and good or nothing at all to another. Or, perhaps a computer malfunction cripples the navigation and atmospheric systems of the International Space Station.

In June of 2007 the International Space station had a crisis with three Russian astronauts on board. At stake were the lives of these men, the Space station itself, and Russian-American relations. The triply redundant, Russian made computer systems that were responsible for flight control, as well as the atmospheric control system, were not functioning properly. The astronauts bypassed the computer temporarily with a jumper cable. They were able to get the Space station operative, but they needed to get the computers back on-line. It was a tense week while the problem was diagnosed. Claims by the Russians about how the American +28V bus 'polluted' their computers, as well as other false accusations ensued. There were many bad guesses as to the cause of the problem. Meanwhile, the engineers, program managers and quality control people down below could not understand why the astronauts temporary 'computer jumper' solution actually worked. There was finger pointing between the Russians and Americans. The astronauts were able to finally track the problem down to a mal-functioning de-humidifier, intended to protect the computer, dripping water on a connector and shorting control lines (Oberg, 2007).

It is safe to say that this is an application where reliability, and the diagnosis of faults, was of paramount importance. It is possible to achieve both of these goals.

Other examples of aerospace mishaps, both pilot error and equipment failure caused, abound. The Accident Investigation Board (AIB) of the United States Air Force Judge Advocate General's Corps discloses to the public the result of all Class A Aerospace Mishaps. A Class A mishap is a serious aerospace accident that results in loss of the aircraft, life or more than $2M in property damages. Mishaps for the Fiscal years 2000 to 2012 are available on the AIB website (http://*usaf.aib.law.af.mil*). From 2005 to 2012 they list over 200 aerospace mishaps alone! Posteriori diagnosis of aerospace failures is taken seriously and enormous effort is expended to diagnose the cause.

Reliable systems and the diagnosis of problems are valued in the commercial and military aerospace field. Other applications that may benefit from reliable systems include the medical (Jafari, Dabiri, Brisk, & Sarrafzadeh, 2005). and nuclear field (Vaurio, 1980)(Hayden, 1976). To meet the challenges of reliable and survivable systems the Byzantine Generals problem was introduced over 30 years ago.

The Byzantines Generals problem was introduced to present a basis for distributed process agreement. It is a general solution to distributed agreement in the presence of faulty or malicious nodes. Byzantine agreement is a form of N modular redundancy (NMR). An NMR system attempts to increase reliability by keeping N copies of a process (nodes) and voting on the results. Lamport, Shostack and Pease (1982) introduced the Byzantine Generals as a metaphor for these processes or nodes. In the literature review section of this paper the Byzantine Generals metaphor is presented. In their often cited paper they propose that processes act in concert to increase the reliability of the system as a whole. They place no limit on the modes of failure for their system nodes. They proposed that a faulty node in a distributed system may even exhibit intermittent failure that seems to be intelligently malicious. The apparently

malicious node(s) may send commands to other nodes in a seemingly deliberate attempt to make the system as a whole fail. Many node's may act in concert to cause parts of the system to diverge in their actions. In a system that uses Byzantine Agreement healthy nodes vote on a proposed action and malicious nodes are masked. All healthy nodes must agree. Contrary to intuition a simple vote is not sufficient to mask nodes capable of Byzantine failure. A Byzantine vote is actually the result of a series of exchanged messages. Consider for example an aircraft with a distributed computer system. Four nodes, two on each wing, collect temperature from RTD's and report Total Air Temperature (TAT) to the other nodes in the system. A most important requirement is that all nodes MUST be using the same value of temperature. Once all nodes agree on a TAT they can then make other system calculation and decisions based on that. At first it seems like a simple matter of the nodes transmitting their temperature and the rest of the nodes taking some sort of average, or voting, on the four sensor readings. If we consider that a defective node could possibly report differing values to nodes, or be intermittent, then we see that a simple vote or average would not work. A system using a Byzantine agreement protocol would be able to mask the actions of a single malicious sensor node in this example.

It is important to realize that in a distributed system that implements a Byzantine agreement protocol the reliability of the system as a whole is increased. This in fact is the intended goal. This is in contrast to some systems that implement a voting scheme, or simple cross checking, where the reliability actually decreases. In systems where the nodes simply cross check each other, if either node fails, then the whole system is marked as unreliable. In a system such as this the reliability decreases with each additional node added by the product of each individual nodes reliability. What does improve is the *integrity* of the system. Integrity is the absence of improper system state alterations (Avizienis, Laprie, Randell, 2001)

Byzantine Agreement is a distributed system protocol where healthy nodes (or processes) mask defective nodes (or processes.) Reliability is increased in these systems (Thambidurai and Park, 1988). In this paper we will use the term 'node' to also cover processors or processes agreement as well. The terms malicious and faulty are used interchangeably as well.

For the system to reach consensus and mask malicious participants a costly protocol is required. The primary cost is in the length and number of exchanged messages. Some proposed protocols require authentication of messages and this can be a costly process as well although it reduces the number and length of exchanged messages. On the other hand, in some systems, authentication may be as simple as a checksum or CRC. Much work to reduce the complexity and cost of the original Byzantine Algorithm (BA) has been proposed by researchers. Fisher and Lynch (1982) placed a lower bound on the BA as proposed by Lamport, Shostack and Pease. Since then much research and work has been devoted to reducing the complexity of the BA algorithm by limiting a nodes mode of failure. For instance by using a CRC for authentication rather than something like public key encryption we limit the mode of failure of a malicious node to being something less than 'intelligent'. But, for many real applications the far encompassing modes of failure as the originally proposed BA may not be necessary. A 2012 paper by Khosravi and Kavian presents a BA algorithm where a transmission omission fault is always detectable by non-faulty nodes. A transmission omission fault is one where a message is not delivered to a node. In their system, if this type of fault did occur it would be detectable by all non-faulty nodes. An example if where this would be where the data link provided the services to ensure data reliability. These researchers present a BA with greatly simplified complexity with this one additional system requirement. The algorithm presented here has a reduction in the number of rounds of information exchanged. The original BA required m+1 rounds of information

exchanged (Fisher and Lynch, 1982). The researchers are able to achieve a reduction down to a fixed 3 rounds of information exchanged. In some cases the algorithm is able to detect a faulty source node. It is not able to detect any other faulty participating nodes or the instantaneous resiliency left in a system operating with defective nodes.

In addition to reaching consensus and 'masking' faulty or malicious participant, much research appears in the literature to 'unmask' or expose faulty participants. An unmasking process is important to diagnosis a system syndrome. While a node exhibits a mode of failure a system exhibits a syndrome which is the sum of the node failure modes. The results of this unmasking process would be useful for particular applications. For instance the results could be useful to indicate a node to be repaired, replaced, reset, logged, or perhaps ignored. A paper by Ramero and Adams (1988) presents some early work on a general BA unmasking procedure.

Previous attempts at the unmaking process were 'test based' whereas this paper was the first to introduce an 'evidence based' BA type algorithm. The authors point out that a malicious node could pass a test based diagnosis, whereas an evidence based method is required to unequivocally mark a malicious node who might try to hide, as the faulty participant. An example of where a test based diagnosis may miss a fault and an evidence based test would catch the fault is an intermittent fault. An earlier paper by Shin and Ramanathan (1987) attempted a diagnosis of Byzantine system nodes, but the algorithm was not complete and it is an off-line method. It did not catch all node fault modes. In addition the algorithm by Ramero and Adams is more readily generally applied. Both of these researcher depended on authenticated messages. Ayeb and Farhat (2003) present a framework to unmask faulty participants using non-authenticated (or oral) messages. Their algorithm is asymptotic to $O(n^3)$ messages exchanged. In the literature section of this paper we will discuss more work on fault identification.

This idea paper proposes to continue the research work presented by Khosravi and Kavian (2012) by proposing to extend their BA protocol to unmask faulty participants and retain their save on transmissions and exchange rounds.

First this idea paper presents the problem statement and the goal of proposed research to meet the problem. The relevance and significance of this work is discussed. A review of the literature is presented next. This paper concludes with an approach (methodology) and the required resources.

Problem Statement

This idea paper proposes to extend the research of Khosravi and Kavian (2012) to identify faulty nodes using an evidence based protocol and algorithm. This paper makes some assumptions about the distributed system. It is assumed that the network is Fully Connected and synchronous. When we look close enough at a system we see that there is never any truly synchronous network. Although this may be true, it is assumed the details of the asynchronous parts of the system would be handled transparently or in another layer. To meet the requirement of a fully connected network of N nodes, the number of links required, L, is =N*(N-1)/2.

The types of faults Khosravi and Kavian are interested in are Byzantine faults. A Byzantine fault is one where the node can display even intelligently malicious behavior. This behavior may include lying about what it was told, telling different parts of the system different things, not telling parts of the system data, or acting in concert with other nodes. The researchers, as well as the original BA, assume that a node is able to determine the sender of a message. A message cannot be spoofed by another node although faulty nodes are assumed to be able to lie. This can be accomplished by having complete connectivity within the system. The system, S, is assumed to consist of N nodes and L links, of which up to t nodes can be faulty

and behave in any manner (Byzantine). When a non-faulty node sends a message, the receiver is able to determine who the sender is. A faulty processor thus cannot interfere with a transmission between two healthy nodes.

The system is assumed to be deterministic. In a deterministic system, two healthy nodes that are given identical inputs and history, have the same output. No randomness is allowed in subsequent system state. Issues of determinism and synchronism are assumed to be taken care of, and exist, by a different part of the system or by careful design.

The target, or goal, of the BA is to have *interactive consistency*. This goal is usually stated as two conditions, an agreement condition and a validity condition:

> *Agreement:* When healthy nodes choose a value, it must always be the same value, even if it is a default value.
>
> *Source Validity:* If the node initiating the command is a healthy node, then all healthy nodes in the system must select the initiators value.

In the original BA presented by Lamport, Shostack and Pease, a limitation was placed on the number of faulty nodes. This limitation is N> t*3. The number of nodes in the network must be greater than three times the number maximum number of faulty nodes. To tolerate at least 1 faulty node, a system needs a minimum of 4 nodes.

Researchers Khosravi and Kavian place one additional requirement on the system to achieve their reduction in transmission and protocol cost over the original BA. They assume that a healthy node can detect and sense transmissions on the communications link between all other nodes. A healthy node will know if a transmission between two nodes fails or there is an

omission fault. A transmission omission fault is one where the message is not delivered to the intended recipient. Their network model assumes fallible nodes and reliable links. The researchers point out that although many communications media are fallible, lower layer protocols can exist that provide the reliability of the links or notification of failed transmissions. This is their intended application. Chandy and Misra (1986) demonstrate that a system that is able to detect and correct omission faults must not be completely asynchronous.

The researchers provide a protocol for all nodes to reach agreement within these limitations of the system. Their paper also touches on the important topic of unmasking faulty source nodes, although it does not go into any great detail on this important subject.

Khosravi and Kavian's proposed protocol and algorithm has three rounds of information exchange. In the first round the source sends the command to all nodes. In the second round all nodes tell all other nodes what they heard from the source. Each node then does some local processing to determine if the source node may be faulty. If a node determines the source node may be faulty, it broadcasts a fixed message to all other nodes. This constitutes the third and final round of message exchanges. After this final round of message exchanges each node uses an internal voting mechanism to determine what the original command was regardless of what they were told in the first round of transmissions by the source. If enough nodes (>1/3n) transmitted the fixed message in the final round, all nodes unequivocally determine the source node to be faulty and do not use its' command. They use a default value instead. The researchers present a proof that this algorithm, when run on each local node, will result in the same value accepted by each node.

This algorithm provides interactive consistency for a distributed system in the absence of transmitter omission faults. (Interactive consistency loosely means all healthy nodes agree, a

more rigorous definition is presented in the literature section.) In addition it can provide, in certain instances, an evidence based diagnosis of a faulty source node. It provides a reduction in the length and number of exchanged messages to reach interactive consistency over the original BA. The goal of the research proposed by this paper is to continue the work done by researchers Khosravi and Kavian. Their research provides a BA algorithm, with a reduction in complexity of number of rounds of messages exchanged and number of bits exchanged. Their research only unmasks a faulty source node when the >3m nodes detect the source node as faulty.

Within the limitation as just discussed, **the problem is** *the reduced cost Byzantine algorithm as presented by Khosravi and Kavian, where transmission omission faults are detectable by all healthy nodes, does not fully identity faulty participants or indicate the health of the system. It identifies a faulty source node when >1/3n nodes agree the source is faulty. The problem is to more fully unmask faulty participants and integrate their protocol.*

The proposed protocol should have less than the optimal limits on the rounds of information exchanged as discovered by Ayeb and Farhat (2003) to unmask faulty participants

A question to be answered is what gains to the Byzantine agreement complexity can be had by more permanently marking a node as faulty.

Past attempts at the unmasking problem did not have the benefit of the research work by Khosravi and Kavian and their reduced complexity BA where transmission omission faults are always detectable.

Goal

To address this problem it is proposed to extend the protocol as proposed by Khosravi and Kavian. It is proposed to develop an evidence based protocol and algorithm that will unmask faulty participants and uncover the cardinality of the faulty participant set. This unmasking

protocol will be integrated into the masking protocol developed by Khosravi and Kavian. A secondary goal of the proposed research will be to uncover the set of faulty nodes.

The number of faulty nodes in an instance of a network can be used to indicate the resilience left in the network. We can envision, in a critical network, the user will be presented with 'meter' that shows the resilience left in the network (in a real system the resilience reading would probably be integrated in other aspects of a system fault or warning.) When the meter is green, then the distributed system can tolerate maliciousness. When the meter turns yellow, instantaneous malicious nodes equals tolerable malicious nodes. When red, then there is some criticality that the interactive consistency is not guaranteed. As per Thambidurai and Park (1988) there may be different type's of node failure modes and these might be indicated as non-integer values of resilience.

Figure 1 - Possible real time meter of system instantaneous resilience.

Success will be determined by presenting an algorithm that masks malicious nodes in the absence of omission faults and at the same time unmasking faulty participants in less than the optimal $O(n^3)$ messages exchanged per the limits as discovered by Ayeb and Farhat (where faults due to transmission omission ARE allowed.)

Research Questions and Issues

Several questions are raised by this proposal. The first is what is the range of system syndromes that can be diagnosed in the 3 rounds of transmitted messages as proposed by Khosravi and Kavian? What is the range of syndromes that can be diagnosed if we exchange more information in the same three rounds of transmissions? What if we allow more than 3

rounds of information exchange? If a node is found malicious by healthy nodes, and marked as non-participatory, will this accelerate Byzantine agreement in subsequent exchanges? How do we, or can we, integrate prosecution sets over multiple agreement exchanges to determine faulty nodes?

An important question to be answered as part of this research is to determine whether an indication of the *resilience* of the Network can be indicated. Khosravi and Kavian do not provide any sort of indicator or metric as part of their algorithm. Resilience can be defined as the number of failures the byzantine network can tolerate. For instance a Byzantine network with 7 nodes can theoretically tolerate up to 2 malicious nodes. Its resilience is 2 in the absence of any maliciousness. In a fully functioning network as proposed by Khosravi and Kavian, as more nodes are marked as malicious, where is the indication that the resilience has dropped to 0? Where or what is the indication in their protocol/algorithm that the resilience has dropped below 0, i.e. the network is no longer reliable. What happens when the resilience drops below 0?

Literature Review

An early paper leading to Byzantine fault tolerance was “SIFT: Design and Analysis of Fault Tolerant Computing for Aircraft Control” (Wensley et al, 1978). SIFT, or Software Implemented Fault Tolerance was a collaboration between SRI and Bendix corporation. The SIFT distributed system was designed for the critical function of aircraft control. The processors used were the ‘well known’ 8080 processors. Three significant claims are made in this early paper. Earlier work looked at fault tolerance on a much lower level, the level of gates and adders. They claim they are the first to look at redundancy at the processor and memory level. They also claim that they are the first to not place any limitations on the failure modes (thus the precursor to the later Byzantine failure modes). Earlier work, they claim, focused on stuck at

one and stuck at zero type errors. These two features of SIFT simplifies the Failure Error Mode Analysis or FEMA required of certain airborne electronic equipment. Their processors use a two out of three vote. Fault isolation is important to SIFT processors. Fault isolation means that processors can read information from any other processor, but can only write into their own memory. Each processor then gets information from other processors in the system and each processor 'votes' on the outcome. Processors can also diagnose or unmask faulty participants. If a processor is determined to be faulty then its tasks are assigned to another processor. If a bus is determined to be faulty then processors will request their data over a non-faulty bus. After reconfiguration the SIFT system can tolerate another failure if there are enough healthy participants remaining. Since the SIFT system was design for aircraft control, real time design requirement had to be met. Both throughput and latency were SIFT design considerations. Another very important design consideration is that processors clocks are not locked to each other. From time to time processors resynchronize themselves using the median clock resynchronization algorithm. An early result of this clock synchronization is they prove that no algorithm can 100% reliably synchronize the three clocks in the presence of 1 clock failure. You need at least four clocks. The equation they give is $N > 3M$ for M failures out of N nodes. The algorithm they use to synchronize the clock is the *interactivity consistency vector*. The researchers do not seem to generalize this result for the 'clock fixup' algorithm to the task at hand of getting nodes to agree. This clock resynchronization seems to be the precursor to the later Byzantine agreement paper. Two of the authors of this paper were Leslie Lamport and Robert Shostak.

In 1980 Leslie Lamport, Robert Shostak and Marshall Pease wrote a significant paper leading to their later formalization of the Byzantine Generals Problem metaphor. The paper was

"Reaching Agreement in the Presence of Faults". This paper introduces the basic elements of the problem of distributed nodes 'agreeing' on information and lays the groundwork for the Byzantine Generals. The paper introduces the term and concept of '*interactive consistency'* in more general terms than the 1978 "SIFT" paper. Two specific examples are given where processor may have to agree; the synchronization of clocks and on values read from sensors. As they point out in the 1978 "SIFT" paper clocks of distributed systems may periodically need to be synchronized. A second example they give, which they did not originally realize, is that multiple nodes in a distributed system may read sensor information that the system as a whole needs to agree on. At the beginning of their SIFT project they naïvely assumed that a simple voting algorithm would be sufficient to maintain 'interactive consistency.' They point out that the situation becomes much more difficult if we assume that nodes, including the sending node, can be maliciously faulty (or intermittent). A node may send differing values to other nodes thus foiling a voting algorithm. This paper presents several major results. They present the conditions that will become their interactivity consistency condition in their later paper introducing Byzantine Agreement. They show that interactivity consistency is possible for N>= 3M + 1 nodes where M is the number of faulty nodes and N is the total number of system nodes (faulty + healthy). The paper point out that the number of rounds of information required is M+1. If authentication of messages is allowed they show that the number of rounds of information is reduced. They point out that if faults are due to simpler errors rather than malicious intelligence than the problem of authentication is reduced from a cryptographic problem to something simpler. For this situation they show that N >= M >= 0 is sufficient.

They Byzantine Generals metaphor was presented by Leslie Lamport, Robert Shostak and Marshall Pease in their 1982 paper, "The Byzantines Generals Problem." In this paper they

introduce the Byzantine Generals. Before this paper this problem was known as reaching interactive consistency, after this paper it is known as the Byzantine Generals Problem. They present an interesting metaphor. The Byzantine Empire purportedly had a treachery problem in its high ranks. It could not be guaranteed that a General in the Byzantine army was not malicious or 'defective'. They problem, they state, is that there is a number of Byzantine Generals surrounding a enemy city. The Generals must reach agreement on a decision, to attack or retreat (for example). As we now know the term "Byzantine Failure" or "Byzantine Node" can mean a node, or set of nodes, that can fail in any failure mode. The mode can be one that seems to be maliciously intelligent among cooperating corrupt nodes to cause the failure of the complete system. Of course, the generals reaching agreement on a plan of attack is a metaphor for nodes of a distributed system reaching agreement. For simplicity of discussion the paper reformulates the Byzantine Generals problem as a problem of a General and his two lieutenants. Figure 2 illustrates why interactivity consistency requires at least four nodes. Here we see the red node, a malicious "source" general, giving differing commands to each of two lieutenant's, the blue nodes. Node 2 is told by Node 1 to "Go 1." Node 2 is told by Node 3 that the general said to "Go 0." Node 2 is cannot determine who is the traitor. It must listen to the source and "Go 1." Node 3, on the other hand, has the opposite problem. It is told by the source node to "Go 0". It must listen to the source node and "Go 0".

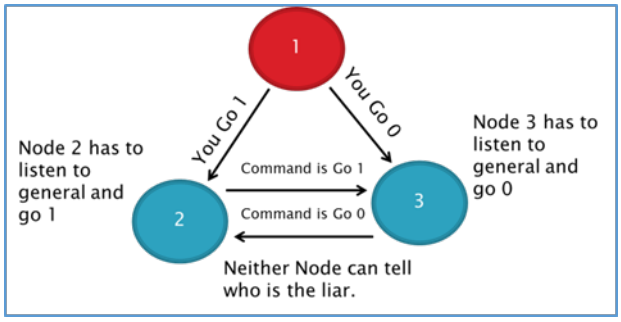

Figure 2 - Three Node Problem. A fourth Node is required to break a tie in the case of a faulty source node.

The healthy nodes of this distributed system each conclude differing actions. The system diverges and does not reach interactive consistency. What is missing is a fourth node to break the tie. There are two other significant contribution of this paper. This paper more formally states the *interactivity consistency* condition. Secondly, it analyses' the requirements for connectivity of the nodes. They show that if there are 3M+1 generals (the maximum number of traitors) then complete connectivity is required of healthy nodes.

In another paper, the lower bound for the number of rounds of transmission to ensure interactive consistency was proven (Fisher & Lynch, 1982). Their paper presents a formal and rigorous proof that no Byzantine Algorithm exists for k<=m where k is the number of rounds and M is the number of faulty nodes. The only complexity measure they are concerned with is the number of rounds. The authors note that a tremendous amount of information is exchanged, $O(n^{m+1})$ values. They question if another algorithm exists with reduced communications by, for instance, increasing the number of rounds of information with less information transmitted each round.

A significant paper was published by Thambidurai and Park (1988) that demonstrates a reduction in complexity (the number of rounds of information) can be had by partitioning the type of BA failures into different modes. Here they partition the failure into asymmetric malicious faults (a), symmetric faults (s), and non-malicious faults (b). The result they obtain is N>= 2a +2s +b + r where N is the total number of nodes and r is an 'algorithm dependent term'. An observation the researcher present is of notable interest. They observe that the Byzantine Algorithm makes Failure Mode Error Analysis easier by essentially wrapping any type of failure up into a "Byzantine Failure" at the cost of higher complexity by considering only this worst case situation. If on the other hand in a particular application only non-malicious faults can

occur then the results here show that a less costly algorithm exists. They also show that there is no benefit in increasing from 4 to 5 or 6 nodes in a Byzantine system. Reliability goes down with no improvement in system resilience. Table 1 lists probabilities of system failure for Byzantine networks of 4, 5, and 6 nodes. The individual node failure rate was chosen to be 1.0×10^{-4} per hour. The equation for probability of failure is $\approx (N(N-1)\ \lambda^2\ t^2\ )/2$, where N is the total number of nodes and λ is the failure rate of an individual node at time t. They show that by partitioning the failure modes they are able to obtain a less granular increase in reliability.

N	P(Failure)
4	6×10^{-8}
5	1.0×10^{-7}
6	1.5×10^{-7}

Table 1 - Reliability of Byzantine System for 3, 4 and 5 nodes. Adapted from "Interactive Consistency with Multiple Failure Modes" (Thambidurai & Park, 1988).

One of the first paper to propose unmasking or exposing faulty nodes specifically in a Byzantine system is written by Shin and Ramanathan (1987) . The researchers propose that the Byzantine algorithm can be considered the process of masking faulty nodes. When a faulty node is unmasked it is exposed as faulty. This would be useful in the diagnosis of a faulty system. In this paper they propose an offline method of fault analysis. The researcher's algorithm depends on a diagnosis process where nodes exchange a predetermined authenticated message. Processes observe the message exchange. Authenticated messages allow a process to relay a message to another node, and append its' own information, with the guarantee that the relaying node cannot alter the original message contents. Fault free links are assumed. They prove that any faulty processor can be diagnosed with full certainty if it exhibits if faultiness m+1 times where m is the maximum number of faulty processes in the system.

In 1988 Ramarao and Adams present an interesting algorithm for exposing faulty nodes. Here they suggest that the method proposed by Shin and Ramanathan is test based. The authors propose an evidence based algorithm. They note that for a general type of failure, such as an intelligent malicious node, the software could be compromised and pass an off-line test yet still cause failures in an operating environment. The intermittent fault may also fall into this category. They note that the only true way of diagnosing these type of failures is by evidence, or information collected by healthy nodes about the fault during normal operation. They propose that for a fault diagnosis algorithm to be *fair* it must not diagnose any healthy processes as faulty. They do not attempt to unambiguously diagnose all faulty processors. Two aspects of Byzantine systems that complicate their detection algorithm is that a malicious node may hide information about other malicious nodes, or, it may accuse healthy nodes. The algorithm they present is based on the information exchanged during the byzantine agreement protocol information exchange. The results of the diagnosis are then exchanged among nodes using a Byzantine Algorithm. During the Byzantine Agreement process information is exchange by each node as to what every other node claims they heard. A node has information as to an accusation by another node and who is at fault. The node takes this information and creates intersecting sets of the nodes at fault. If greater than t+1 nodes claim another node to be faulty then that node is marked as faulty.

A later paper by Ramarao and Adams (1989) proposes a new evidence based algorithm which is optimal. Again, they rely on authenticated messages. The researchers draw a parallel between evidence based methods (as presented here) and test based methods. The test based method they choose to compare to are PMC based algorithms. They note that in both methods there are two phases. First information is collected about the system. This information may

include *accusations* that nodes make about the state of other nodes. The second phase is a where the system processes those evidences. The only difference is in *how* the evidence is collected. The algorithm presented accepts a *testimony graph* as an input and produces a set of faulty processes and links as output. They note that the work by Shin and Ramanthan assumes non-faulty links and the algorithm presented here can diagnose faulty links as well as faulty processes/nodes. The researchers maintain their *correctness property* fro their previous research. The correctness property of their algorithm implies that no non-faulty process can be identified as faulty. They add in this paper a *completeness property*. Completeness of the algorithm means that all faulty processes and links are detected.

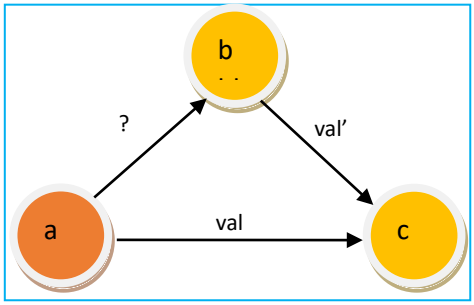

Figure 3 - Prosecution Principle

Ayeb and Farhat (2003) find an algorithm for identifying or unmasking faulty nodes. The paper they present is unique in that it relies on the original Byzantine algorithm. Their algorithm can accelerate the agreement process as well using the exchanged information although their diagnosis algorithm is independent of the agreement algorithm. Figure 3 illustrates the *prosecution principle*. In this example we see a source node, a, sending val to node c. We also see node c receiving a different message from node b, val'. Node c does not 'know' who is lying. We say that node c prosecutes both node's a and b. In the algorithm Ayeb and Farhat propose implicant sets are built. By inspecting overlapping sets either an actual diagnosis or partial diagnosis is built. An example of where a partial diagnosis could occur from

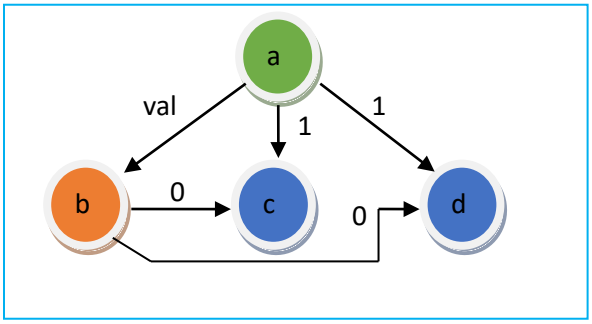

Figure 4 - Ambiguous situation.

ambiguity in the system of exchanged messages is shown in Figure 4. Here we see a situation where previous methods of diagnosis fail. In this example neither nodes a or b can tell where the error lies. Either node b sent an incorrect value to nodes c and d, or node a originally sent an incorrect value to node b. Ayeb and Farhat propose that by nodes exchanging implicant sets this ambiguity in some cases can be resolved.

Research Approach

Ayeb and Farhat present a method of diagnosing a system syndrome by way of passing implicant sets. A first step would be to examine see what syndromes can be detected in three rounds by passing implicant sets between nodes to make a diagnosis..

A goal will be to find a method of fully unmasking a faulty source node. The source node is potentially adding information and influencing other healthy nodes in the system and should be of prime importance. The most unequivocal way of diagnosing a node is evidence based. This research will build a tree of all the possibly ways the source node can affect other nodes during the Khosravi and Kavian protocol to understand the failure vectors. It will then look at what additional information is required of each node to effect a diagnosis of a faulty source node. Research will determine the minimum number of rounds of information that need to be exchanged to fully diagnose a faulty source node only. If it cannot be determined if the source node is unequivocally faulty then this research will look at way of marking a node suspect and judging the node faulty in subsequent message transactions. It is expected that this research will result in a further decrease in the amount of messages exchanged between nodes if the list of faulty nodes can be used to mark nodes determined faulty as not participatory during the byzantine agreement process. Finally, this research will look at determining if nodes other than the source node can be diagnosed as faulty during the byzantine agreement process. If the faulty

source node cannot be unambiguously determined, the goal will be to at least determine the number of faulty nodes to effect an instantaneous network resilience metric.

Importance of Research

In any system instance, the resilience of a Byzantine network is indicated by t – m, where t is the number of tolerable faults in a Byzantine Network (t = N/3 + 1 for non-authenticated messages), and m is the actual number of faulty or malicious nodes. When the resilience drops to 0 any further maliciousness could result in the non-agreement of nodes.

In many critical applications, such as the examples given at the beginning of this paper, it is not only important to diagnose nodes for replacement, etc., but it important to know when the system cannot be relied on anymore. If a military aircraft is flying into a dangerous mission, perhaps the mission needs to be aborted or nodes replaced. The purpose of a Byzantine System is to mask errors. Masking errors is a good thing as long as the overall system can be relied upon.

Resources

Limited resources are required for this research. Access to databases of scholarly work such as the IEEE and ACM is required. A computer for word processing and internet access is required. The time required will be 6 weeks.

REFERENCES

Adams, J. C., & Ramarao, K. V. S. (1989, June). Distributed diagnosis of Byzantine processors and links. In *Distributed Computing Systems, 1989., 9th International Conference on*, 562-569. IEEE.

Avizienis, A., Laprie, J. C., & Randell, B. (2001). Fundamental concepts of dependability. Technical Report Series-University Of Newcastle Upon Tyne Computing Science.

Ayeb, B., & Farhat, A. (2004). A flexible formal framework for masking/demasking faults. *Information Sciences*, *159*(1), 29-52.

Chandy, K. M., & Misra, J. (1986). How processes learn. Distributed Computing, 1(1), 40-52.

Fischer, M. J., & Lynch, N. A. (1982). A lower bound for the time to assure interactive consistency. *Information Processing Letters*, *14*(4), 183-186.

Hayden, K. C. (1976). Common-mode failure mechanisms in redundant systems important to reactor safety. Nucl. Saf.;(United States), 17(6).

Jafari, R., Dabiri, F., Brisk, P., & Sarrafzadeh, M. (2005, March). Adaptive and fault tolerant medical vest for life-critical medical monitoring. In *Proceedings of the 2005 ACM symposium on Applied computing* , 272-279. ACM.

Khosravi, A., & Kavian, Y. S. (2012, July). Reaching an agreement in a distributed environment in absence of omission faults. In *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*, 1-4, IEEE.

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, *4*(3), 382-401.

Oberg, J. (2007, October). Space station: Internal NASA reports explain origins of June computer crisis. *IEEE Spectrum*.

Pease, M., Shostak, R., & Lamport, L. (1980). Reaching agreement in the presence of faults. *Journal of the ACM*, *27*(2), 228-234.

Ramarao, K. V. S., & Adams, J. C. (1988, October). On the diagnosis of Byzantine faults. In *Reliable Distributed Systems, 1988. Proceedings., Seventh Symposium on*, 144-153. IEEE.

Shin, K., & Ramanathan, P. (1987, July). Diagnosis of processors with Byzantine faults in a distributed computing system. In *Proceedings of the 17th International Symposium on Fault-Tolerant Computing*, 55-60.

Thambidurai, P., & Park, Y. K. (1988, October). Interactive consistency with multiple failure modes. In *Reliable Distributed Systems, 1988. Proceedings., Seventh Symposium on*, 93-100. IEEE

Vaurio, J. K. (1980). Availability of redundant safety systems with common-mode and undetected failures. *Nuclear Engineering and Design*, *58*(3), 415-424.

Wensley, J. H., Lamport, L., Goldberg, J., Green, M. W., Levitt, K. N., Melliar-Smith, P. M & Weinstock, C. B. (1978). SIFT: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE*, *66*(10), 1240-1255.