RESILIENT VOTING MECHANISMS FOR MISSION SURVIVABILITY IN CYBERSPACE: COMBINING REPLICATION AND DIVERSITY

Charles A. Kamhoua[1], Patrick Hurley[1], Kevin A. Kwiat[1] and Joon S. Park[2]

[1]Air Force Research Laboratory, Information Directorate, Rome, New York, USA
{charles.kamhoua.ctr, patrick.hurley, kevin.kwiat}@rl.af.mil
[2]Syracuse University, Syracuse, New York, USA
jspark@syr.edu

ABSTRACT

While information systems became ever more complex and the interdependence of these systems increased, mission-critical services should be survivable even in the presence of cyber attacks or internal failures. Node replication can be used to protect a mission-critical system against faults that may occur naturally or be caused by malicious attackers. The overall reliability increases by the number of replicas. However, when the replicas are a perfect copy of each other, a successful attack or failure in any node can be instantaneously repeated in all the other nodes. Eventually, the service of those nodes will discontinue, which may affect the system's mission. Therefore, it becomes evident that there must be more survivable approach with diversity among the replicas in mission-critical systems. In particular, this research investigates the best binary voting mechanism among replicas. Furthermore, with experimental results, we compare the simple majority mechanism with hierarchical decision process and discuss their trade-offs.

KEYWORDS

Diversity, Fault-tolerant Networks, Intrusion Resilience, Reliability, Survivability

1. INTRODUCTION

Researchers have recently investigated the Internet topology and concluded that it has a hierarchical structure [1]. Further, new developments in cloud computing make it possible to run applications using numerous computer nodes or virtual machines distributed around the world. This advancement in cloud computing facilitates the design of fault-tolerant networks. In fact, one approach to fault-tolerant networks is node replication. Using replicated nodes, the system-of-nodes can tolerate the failure of a few replicas while guarantying critical functionality.

One specific technique of fault-tolerant networks is binary voting. Binary voting is of great interest when the system's defender wants to make a binary decision from the monitoring of a binary event. The binary event of interest may be distributed in the Internet, the cloud, or in a large organization that has branches around the world. Moreover, most civilian and military organizations have a hierarchical structure.

For instance, let us consider that each soldier in a battlefield is equipped with a sensor that monitors a binary event. Soldiers are partitioned in subsets under the control of a captain. Each soldier's sensor directly reports its observation in the form of a binary vote to a minor decision center commanded by a captain. Each captain reports as a single binary vote its soldier's majority opinion to a colonel. Further, each colonel sends to the general a single binary vote

DOI : 10.5121/ijnsa.2012.4401

consistent with its captains' majority vote. Only the general has the power to decide and make its binary decision based uniquely on its colonel's majority vote.

One contribution of this paper is to analyze what constitutes the optimum vote aggregation mechanism between simple majority and hierarchical decisions. We find that in most circumstances, the simple majority rule is more robust than hierarchical decision. However, the hierarchical vote aggregation method is faster and more scalable. Further, a special consideration is given to intelligent malicious nodes that attempt to strategically defeat the aggregate results. The chance that the aggregate decision survives in the presence of compromised nodes is analyzed in both the hierarchical decision and the simple majority. In addition, we use the law of diminishing marginal utility to show how to calculate the optimum number of nodes that participate in the decision process.

In addition to the optimum vote aggregation method and the calculation of the optimum number of replicas, another contribution of this research is to look into the importance of diversity to survivability. Definitions and requirements of survivability have been introduced by previous researchers [2-4]. We define survivability as the capability of an entity to continue its mission even in the presence of cyber attacks, internal failures, or accidents. An entity ranges from a single component (object), with its mission in a distributed computing environment, to an information system that consists of many components to support the overall mission. An entity may support multiple missions. In order to support the pressing requirements for survivability in mission-critical systems, we identified the static and dynamic models and discussed their trade-offs with generic implementation architectures in our previous works [5-10].

The static survivability model is based on redundant components (e.g., multiple copies of a critical component), prepared before the operation, to support critical services continuously in a distributed computing environment. Redundant components can be located in the same machine or in different machines in the same domain or even different domains. The same service can be provided by identical components (e.g., copies of the original component) or by diverse components that are implemented in various ways. Isolated redundancy (in different machines or domains) usually provides higher survivability because the replaced component can be running in an unaffected area. For instance, if the redundant components are distributed in different places of the network, the services provided by those components can be recovered in the event of primary network service failures. However, if there is a successful attack to a component, replacing that component with an identical copy is not a fundamental solution, because identical components are vulnerable to the same exploit used in the previously successful attack.

In the dynamic survivability model, unlike the static model, there are no redundant components. The components that have failed or are under the control of malicious codes are replaced by dynamically generated components on-the-fly and deployed in runtime when they are required. Furthermore, this model allows the replacement of the malicious components with immunized components if possible, which enables it to provide more robust services than the static model. If we do not know the exact reason for the failures or types of malicious codes, or if it is hard to recover components against known failures or from the influence of malicious codes, we can simply replace the affected component with a new one - thereby creating in a renewed service. We call this a *generic immunization strategy*, which can be effective against cyber attacks. If a component (a machine or a whole domain) is under attack, the generic immunization strategy suggests generating a new copy of the component and deploying it in a new environment that is safe from the attack. Although the generic immunization strategy supports service availability continuously, the new component might still be susceptible to the same failures or attacks.

Technically, it is simpler to implement the static survivability model than the dynamic survivability model because the former basically requires redundant components prepared to be used if necessary, while the latter requires other support mechanisms to deal with the

component in runtime. In the static model, the service downtime is relatively short because the system just needs to change the service path with one of the previously-prepared redundant components. However, if the initially selected component is running in its normal state, we do not need to use other redundant components. We say that in this situation that the resource efficiency is low. The adaptation capability in this model is based on the reconfiguration among predefined alternatives. On the contrary, the dynamic model can adapt dynamically to the kind of failures or attacks that occur in runtime. Furthermore, if component immunization is possible, it can provide resistance to the same kinds of future failures and attacks. Therefore, the overall robustness in this model is higher than in the static model. However, the dynamic model has an inherent disadvantage in terms of service downtime. The recovery process can range from seconds to a few minutes. This downtime drawback will cause major problems in mission-critical systems because there will be no service provided by the component available during the recovery period.

Therefore, in order to compensate for the weaknesses in the two models and to enhance the overall survivability in a mission-critical system, we incorporate the idea of a hybrid model, which can be implemented by using diverse critical components – components that are functionally equivalent but whose make-ups are diverse. We refer to these functionally equivalent yet diversely implemented components as *diverse replicas*. In this way, we can reduce the complexity and the service downtime that are caused by the dynamic model, while we can improve the overall robustness of the static model. In this paper, we apply the idea of diverse replicas to the voting components in a survivable network. In particular, we consider the *simple majority* and *hierarchical troika* voting mechanisms with/without diversity and discuss our experimental results.

This paper is organized as follows. Section 2 is dedicated to the related works. Section 3 shows how to calculate the optimum number of nodes. After the optimum number of nodes is calculated, we will analyze in Section 4 the optimum nodes' arrangement. Section 5 exhibits our numerical results highlighting reliability, intrusion resilience, and diversity. Finally, Section 6 concludes the paper and proposes future research directions.

2. RELATED WORKS

In recent years, several researches have focused on binary voting. Kwiat *et al.* [11] analyzed the best way to aggregate the nodes' observations given the nodes' reliability. The nodes are assumed to be homogeneous. The reliability of a single node p is its probability to make the correct decision. They showed that Majority Rule (MR) performs better if the nodes' observations are highly reliable (p close to 1). But for low value of p, $\left(p < \frac{1}{2}\right)$ choosing a Random Dictator (RD) is better than MR. Random Troika (RT) combines the advantage of those two strategies when the node reliability is unknown $(0 \leq p \leq 1)$. Generally, it can be shown that if a small proportion of nodes are compromised and nodes are highly reliable, assuming that an odd number of nodes is used, we will have MR=Random N>….>Random 5>RT>RD. However, if the majority of nodes are compromised, the previous inequality is reversed. That is because, with a majority of compromised nodes, increasing the size of the subset of deciding nodes also increases the likelihood of compromised nodes taking part in the decision.

Following the previous research, Wang *et al.* [12] analyzed the nodes decision in a cluster. There are n clusters of m nodes, with a total of $n*m$ nodes. The attacker chooses the number of clusters to attack while the defender chooses how many nodes participate in the decision in each cluster. They formulated a zero-sum game in which the defender maximizes the expected number of clusters deciding correctly while the attacker minimizes that number. They proposed a general framework to find the Nash equilibrium of such a game. However, the cluster

structure is assumed to be fixed. This research will show that the defender has a better optimization strategy just by changing the cluster structure.

Malki and Reiter [13] analyze Byzantine quorum systems. They propose a masking quorum system in which data are consistently replicated to survive an arbitrary failure of data repositories. Their work also proposes a disseminating quorum system. Faulty server can fail to redistribute the data but cannot alter them.

Bhattacharjee *et al* [14] use a distributed binary voting model in cognitive radio. To compensate their noisy observation of channel utilization by primary spectrum users, each secondary user requests their neighbor's opinion (vote). Those interactions are repeated and the Beta distribution is used to formulate a trust metric. Nodes with low trust are eliminated to have a more accurate channel evaluation. Replica voting for data collection in an active environment is investigated in [15-16].

Park *et al*. [17-18] proposed a trusted software-component sharing architecture in order to support the survivability at runtime against internal failures and cyber attacks in mission critical systems. They defined the definition of survivability using state diagrams, developed static and dynamic survivability models, and introduced the framework of multiple-aspect software testing and software-component immunization. Amir et al. [19] presented a Byzantine fault-tolerant replication protocol that is resilient to performance attacks in malicious environments. Dai et al. [20] and Meng et al. [21] introduced self-healing approaches for reliable systems.

Alongside the research above, there is a large mathematical literature about binary voting starting with Condorcet [22]. Simply stated, the Condorcet Jury Theorem (CJT) shows that if a group of homogeneous and independent voters, with voter competence better than random, uses the simple majority rule to choose among two alternatives having equal a priori probability, then the group's decision accuracy monotonically increases and converges to one as the number of voters increases. Owen *et al.* [23] generalized the CJT while considering any distribution of voter competence. The original CJT was restricted to a uniform distribution of voter competence or reliability p. A mathematical survey of binary voting is provided in [24]. A preliminary version of this paper appears in [10].

3. CALCULATION OF THE OPTIMUM NUMBER OF REPLICATED NODES

The optimum number of replicated nodes has attracted less attention in the literature. The implicit assumption is that the number of nodes that participate in the decision is given. However, we believe that number of nodes strongly contributes to optimizing the decision center's reactions. We are proposing an optimization approach based on the law of diminishing marginal utility.

Without loss of generality, we assume in this section that the nodes are homogeneous and that each node's reliability is p. We also consider that $0.5 < p \leq 1$. Therefore, in the framework of Condorcet [22], using a simple majority and without malicious nodes, the reliability of the decision monotonically increases and converges to one as the number of voter grows to infinity. This is valid for either the simple majority rule or the hierarchical decision process.

In the democratic political system that Condorcet advocated, the government organizes the election and does not pay its citizens to vote. Thus, a larger electorate increases the result accuracy at no fee to the government. Accordingly, a larger electorate is always better in terms of vote accuracy. However, in fault-tolerant networks, there is a system designer's cost associated with any additional voter (e.g., node or sensor). Precisely, there is a tradeoff between costs and accuracy in fault-tolerant networks. We will show that above the optimum number of replicated nodes, any increase in replicas will result in diminishing marginal utilities. This is

because the cost of an additional replica exceeds the marginal payoff, and that the marginal payoff depends on the reliability increase due to an additional replica.

Let *C* be the cost of a node and *V* be the value of the target being protected by a mission. A binary voting mechanism is implemented to aggregate the decision of the *N* nodes. An odd number of nodes are used to avoid tie vote. Let us take $m = \frac{N+1}{2}$. The probability $P_N(p)$ that *N* nodes reach the correct decision in a majority rule can be calculated as:

$$P_N(p) = \sum_{k=m}^{N} \binom{N}{k} P^k (1-p)^{N-k}, \text{with } m = \frac{N+1}{2}. \quad (1)$$

When we increase two nodes, the new decision accuracy becomes:

$$P_{N+2}(p) = \sum_{k=m+1}^{N+2} \binom{N+2}{k} p^k (1-p)^{(N+2)-k}. \quad (2)$$

We will proceed in two steps. In the first step, we repeat the CJT to show that P_N monotonically increases. The second step will show that the rate of that increment decreases. Those two steps are enough to validate a diminishing marginal utility.

Theorem 1: The sequence P_N increases with *N* when $0.5 < p \leq 1$. (CJT, [22])

Proof: The following recursion formula holds:

$$P_{N+2} = P_N + p^2 \binom{N}{m} p^{m-1}(1-p)^m - (1-p)^2 \binom{N}{m} p^m (1-p)^{m-1}. \quad (3)$$

In fact, two additional voters can influence a binary election using the simple majority rules if and only if one alternative has one vote more than the other. The second term of the right hand side (RHS) of (3) is the probability that the incorrect alternative has one more vote than the correct one and the two new voters vote correctly. The third term of the right hand side (RHS) of (3) is just the reverse or the probability that the correct alternative has one more vote than the incorrect one and the two new voters vote incorrectly. After a few algebraic manipulations, we have:

$$P_{N+2} - P_N = (2p-1)\binom{N}{m}[p(1-p)]^m > 0 \text{ if } p > 0.5. \quad (4)$$

■

Theorem 2: The sequence $W_N = P_{N+2} - P_N$ decreases with *N* when $0.5 < p \leq 1$.

Proof:

$$\frac{W_{N+2}}{W_N} = \frac{P_{N+4} - P_{N+2}}{P_{N+2} - P_N} = \frac{(2p-1)\binom{N+2}{m+1}[p(1-p)]^{m+1}}{(2p-1)\binom{N}{m}[p(1-p)]^m} = \frac{2p(1-p)(N+2)}{m+1}$$

$$= \frac{4p(1-p)(N+2)}{N+3} < 4p(1-p) < 1. \quad (5)$$

Theorem 2 shows that the marginal reliability value of two additional nodes diminishes. Then increasing the number of nodes yields concave utility. Thus, applying the law of diminishing marginal utility, the optimum number of nodes to use in the decision process should be the larger number *N* such that:

$$(P_{N+2} - P_N)V \geq C. \tag{6}$$

Figure 1 provides a numerical example. We use $p = 0.9$.We can see that $P_1 = p = 0.9$, $P_3 = 0.91944$, $P_5 = 0.925272$. Thus, the increase in precision from the addition of the first two nodes (2%) is higher than that of the last two (0.5%).

We have provided an approach to calculate the optimum number of nodes when using the simple majority rule with uncompromised nodes. However, this approach can be generalized to the case of RD, RT, or when nodes are arranged in either a cluster or hierarchically while in the presence of compromised nodes.

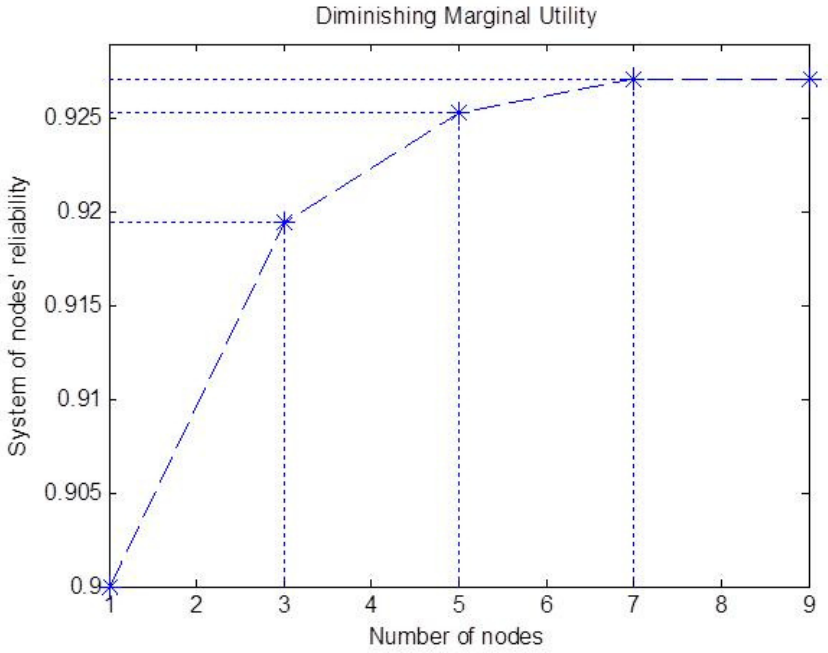

Figure 1. Decision Reliability as a Function of *N*

4. OPTIMUM NODE ARRANGEMENT

In this section, we discuss the optimum structure of the replicated nodes to maximize their fault tolerance. We will compare the simple majority vote with clustering and hierarchical vote. The nodes are considered to be diverse to prevent correlated failure.

4.1. Simple Majority

A simple majority is the most common vote aggregation scheme. An odd number of nodes are used to avoid tie vote. The aggregate result is the outcome having more than 50% of the vote. Figure 2 illustrates a simple majority voting protocol. For the purpose of the discussion in this section, we define the *tenacity* of a structure of nodes as the minimum proportion of nodes that an attacker must compromise to have a total control over the aggregate decision. Therefore,

using a simple majority with *N* nodes (*N* odd), the *tenacity* will be $m = \frac{N+1}{2}$, which approaches 50%. Figure 2 represents a simple majority voting scheme.

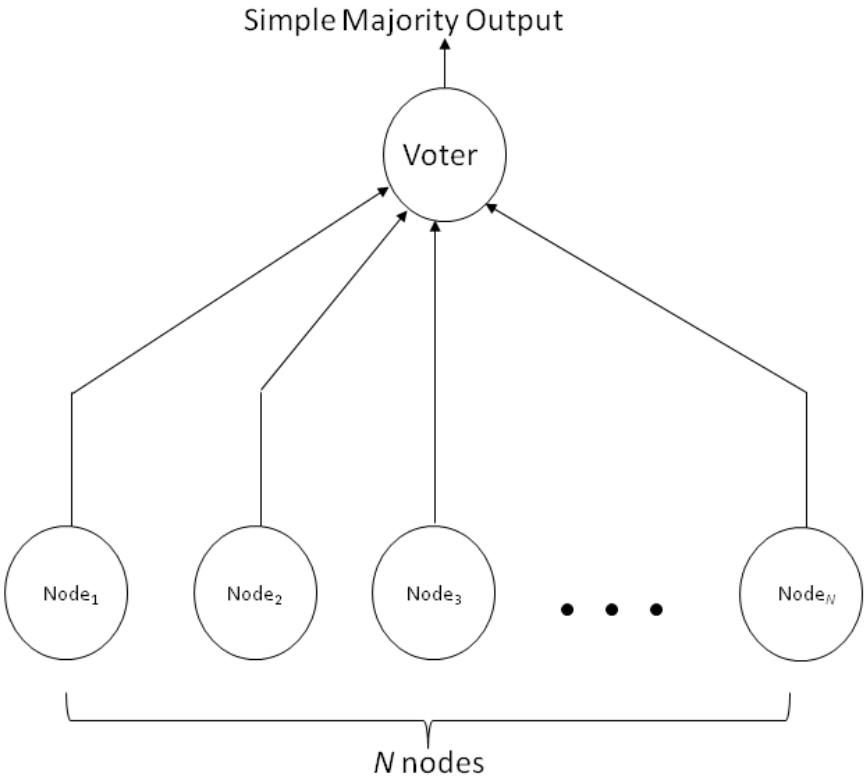

Figure 2: Simple Majority voting

4.2. Clustering

Presently, let us consider a rational agent (the defender) at the decision center that believes that more than 50% of its nodes have been compromised by an attacker. The attacker will then have full control over the decision outcome if using a simple majority rule to aggregate the votes. To respond to the situation, the defender may simply arrange its nodes in a cluster as presented in Table 1.

In Table 1, the *C* represents the compromised nodes and *R* the regular nodes. In the first three columns, *R* is the majority. Thus, we may have a correct decision in the majority of columns or clusters. Let us also consider that the defender aggregate's result is that of the majority of nodes in the majority of clusters. In this case, if we consider highly reliable nodes, 25 nodes can survive the failure of up to 16 nodes as illustrated in Table 1. This is a clear improvement compare to simple majority rule that can only survive the failure of 12 out of 25 nodes. However, the defender can take advantage of the cluster structure if and only if that structure is unknown to the attacker. For instance, an attacker that knows the cluster structure just needs to compromise 9 nodes out of 25 as presented in Table 2. In this case, using a simple majority rule is a superior solution. In fact, the attacker's optimum strategy is to compromise a bare majority of nodes in a bare majority of clusters.

Using $N = (2k + 1)^2$ nodes for instance, we can see that the attacker just needs to compromise $(k + 1)$ nodes in $(k + 1)$ clusters for a total number of $(k + 1)^2$ nodes out of the $(2k + 1)^2$ nodes (see Table 2). The ratio $\frac{(k+1)^2}{(2k+1)^2} = \frac{k^2+2k+1}{4k^2+4k+1}$ converges to 0.25 as k grows.

Table 1. 25 Nodes Illustration

C	C	C	C	C
C	C	C	C	C
R	R	R	C	C
R	R	R	C	C
R	R	R	C	C

C: Compromised Nodes
R: Regular Nodes

Table 2. 25 Nodes Illustration

R	R	R	R	R
R	R	R	R	R
R	R	C	C	C
R	R	C	C	C
R	R	C	C	C

C: Compromised Nodes
R: Regular Nodes

In contrast, when the defender knows the cluster structure while that structure is unknown to the attacker, the optimum attacker's strategy is to randomly attack the nodes. As a consequence, taking the ratio, the cluster structure can survive the failure of $\left[1 - \frac{(k+1)^2}{(2k+1)^2}\right]$ nodes (see Table 1), or 75%. By definition, the maximum tenacity of a cluster structure is 75%.

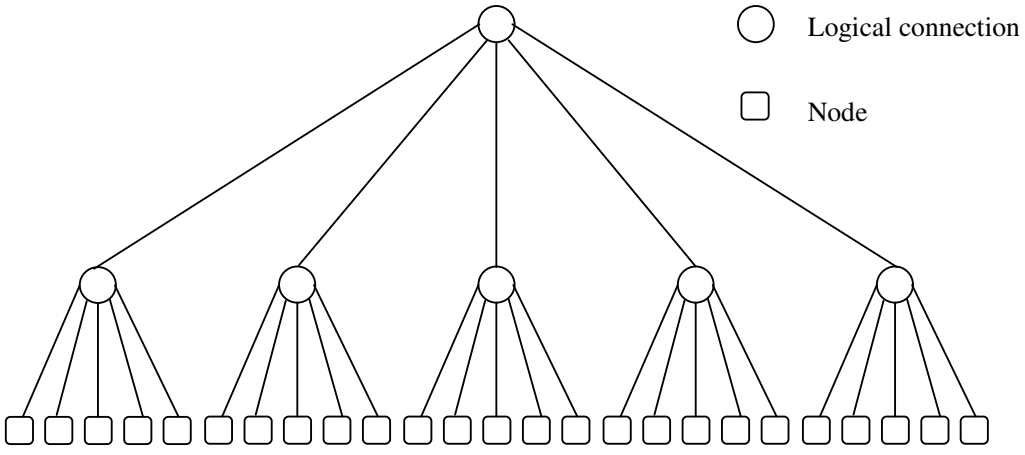

Figure 3: Hierarchical representation of the nodes in Table 1 and 2

In short, the clustering strategy in two dimensions (Table 1) cannot protect the aggregate decision when the number of compromised nodes is higher than 75%. To survive the compromising of more than 75% of the nodes, clustering should be applied in three dimensions

or higher. We see in the next subsection that arranging the nodes hierarchically (which is a higher dimension clustering) can possibly survive the compromising of more than 75% of nodes. For instance, Table 1 or 2 can be represented hierarchically as in Figure 3. From left to right in Figure 3, the first five nodes represent the nodes of the first column or cluster. The next five nodes represent the second column and so on. Thus we can see that the hierarchical vote of Figure 4 has three layers and is comparable to a three dimension clustering.

4.3. Hierarchical Troika

Figure 4 shows 27 nodes hierarchically arranged in subsets of three nodes (hierarchical troika). The 27 nodes make their decision in three layers. The resulting decision of the higher layer is that of at least 2 out of 3 nodes in the lower layer. Since 27 = 9*3 = 3*3*3, the first layer has 27 nodes, the second layer has 9 sub-results, the third layer has 3 sub-results, and the final aggregate result is obtained from the last three results.

The higher layers are not nodes but materialize the logical aggregate decision from the lower layers. Recall that in our scenario, only the soldiers on the ground are equipped with sensors. Thus, logical connections instantiate the hierarchical structure: a captain sending the partial aggregate vote to the colonel that in turn will partially aggregate the vote at his layer to send it to the general. In corporations, the logical connection could be the supervisor sending the vote to the head manager that will report to the director.

A careful analysis of this decision process shows that the 27 nodes can tolerate the failure of up to 19 nodes (see Figure 4). Figure 4 shows 19 nodes with a circle shape (compromised) and 8 nodes with a square shape (regular). The straight line shows the transmission of a correct vote to the higher layer while an interrupted line shows the transmission of an incorrect vote. We can see that the final decision is correct because two out of three votes are correct in the upper layer.

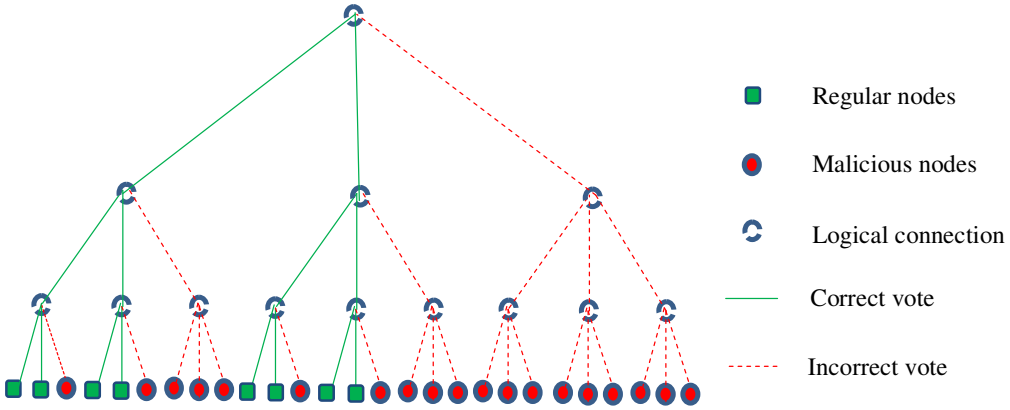

Figure 4. Hierarchical Troika nodes arrangement

Extending the process with 81 nodes, we see that the system tolerates the failure of up to 65 nodes or 80% of nodes. The truthful node can win an election with only 20% of the vote. In general, 3^n nodes require n layers decision process and can tolerate the failure of $(3^n - 2^n)$ nodes. Thus, the tolerance ratio is $\frac{(3^n-2^n)}{3^n} = 1 - \left(\frac{2}{3}\right)^n$ which converges to 100% as n grows. Therefore, we can see that a hierarchical node arrangement maximizes the *tenacity* of the network when the attacker does not know the nodes' assignment into the hierarchical structure. The analysis in this subsection remains valid in other hierarchical structures such as hierarchical 5, 7, …or a combination.

4.4. Structure Comparison

In summary, we have shown that clustering the nodes or arranging them hierarchically is a valuable strategy to the defender if and only if it is hard for the attacker to infer the cluster structure. One way to insure this is for the defender to randomly and periodically reassign the nodes into different clusters or troika arrangements. Moreover, we can see that there is a tradeoff between resisting the compromising of a large number of nodes and the risk of being exposed to the compromising of a few nodes. Therefore, the defender's belief about the distribution of the number of compromised nodes is the most important factor that determines the best structure to use.

Generally speaking, at a given time, if the defender believes that the attacker has compromised only a minority of nodes, the defender may choose among simple majority rule, clustering, or hierarchical troika. However, if the defender believes that the attacker has compromised between 50% and 75% of nodes, the defender must avoid simple majority and use clustering or hierarchical troika. When more than 75% of nodes are compromised, the only solution left is hierarchical troika.

We can perform a similar comparison when the defender does not know the exact number of compromised nodes but his belief about that number has a specific probability density function (PDF). Table 3 provides a summary. Definitely, more specific results will depend on the exact PDF (uniform, normal, exponential, etc.), the shape of the PDF (symmetric or skewed), a node's reliability, and the number of nodes.

Table 3. Comparison of the Structures

		Simple Majority	**Cluster**	**Troika**
Tenacity		Low	Medium	High
Scalability		Low	Medium	High
Speed of vote aggregation		Low	Medium	High
Proportion of compromised nodes	0-50%	Good	Good	Good
	50%-75%	Poor	Good	Good
	75%-100%	Poor	Poor	Good
Probability density function of defender belief's on the number of compromised nodes	Uniform, Symmetric	Good	Good	Good
	Positive skew	Good	Good	Good
	Negative skew	Poor	Medium	Good

Note that we can also have a rectangular cluster (e.g. 5 clusters of 9 nodes or 45 nodes in total). In a rectangular cluster, the number of rows should be as close as possible to the number of

columns to maximize the structure *tenacity*. Moreover, additional precautions can be taken to arrange the nodes hierarchically even though the number of nodes is not a specific power of an integer. Section 5 will reveal more structure comparisons.

5. EXPERIMENTAL RESULTS

This section shows our simulation results to support the different techniques analyzed in this paper. Recall that clustering is a special case of a hierarchical vote. Therefore, the clustering technique is not evaluated in our simulation. The simulation results are generated from MATLAB. We organized the results in three sets of experiments. Each set examines a specific factor. The first factor we examined is the system-of-nodes reliability when there is no malicious node in two structures: hierarchical troika and simple majority. The second factor we consider is the impact of a malicious node in a system-of-nodes. Again, we consider hierarchical troika and simple majority rule. We look into the power of a single malicious node to change the aggregate result. The third consideration is replicas' diversity. In fact, a system of nodes is more vulnerable to malicious attacks and natural faults if the replicas are a perfect copy of each other. That is because any successful attack in any replica can be used to compromise all the nodes. Moreover, with similar replicas, a natural fault can simultaneously damage all the replicas at once. In short, diversity increases the system resilience.

5.1. System-of-nodes Reliability Comparison

We can observe that hierarchical troika and simple majority are identical in the case of 3 nodes. Figures 5 and 6 show how the group of nodes decision reliability varies with individual node reliably using 9 and 27 nodes respectively.

The result is that hierarchical troika outperform simple majority when $0 \leq p < 0.5$ and the reverse is true when $0.5 < p \leq 1$. We forecast that this result holds for any number of nodes above 27. In fact, there is information lost at each layer of the hierarchical vote when a partial vote aggregation is performed (at the logical connection in Figure 4). That information lost decreases the aggregate vote reliability of hierarchical troika compared to a simple majority vote. Thus, if each node's reliability is $0.5 < p \leq 1$, and the defender's main concern is to increase the collective decision reliability while not considering the malicious nodes' action, a simple majority should be used.

However, the main concern of this research is the malicious nodes' action. We deal with malicious nodes' action in the next subsection. We can also see that hierarchical troika is also consistent with CJT. If we have $p = 0.5$, the system of node reliability stays at 0.5. When we have $p > 0.5$ the system of node reliability is above 0.5. On the contrary, when we have $p < 0.5$ the system of node reliability is below 0.5. Looking at the difference between Figure 5 and 6, we see a fast convergence to one when $0.5 < p \leq 1$ (zero respectively if $0 \leq p < 0.5$) as the number of nodes increases. We can also see that the convergence is faster as we move away from $p = 0.5$. Also, as the number of nodes increases or the node's reliability increases, the difference between simple majority and hierarchical troika becomes negligible.

To generalize our analysis above 27 nodes, we have already shown that the aggregate decision reliability using simple majority increases according to the sequence (3) and (4). A similar sequence can be derived using hierarchical troika. First, we need to observe that using three nodes, the aggregate decision reliability in a troika is:

$$P^T{}_3 = 3p^2 - 2p^3. \quad (7)$$

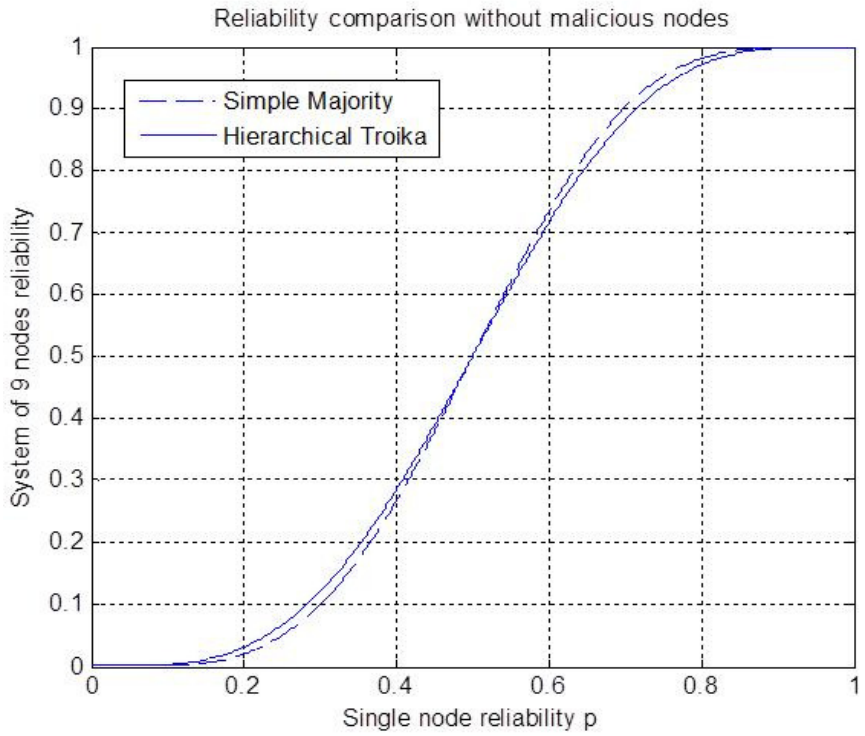

Figure 5. Troika vs. Majority group of 9 nodes decision reliability

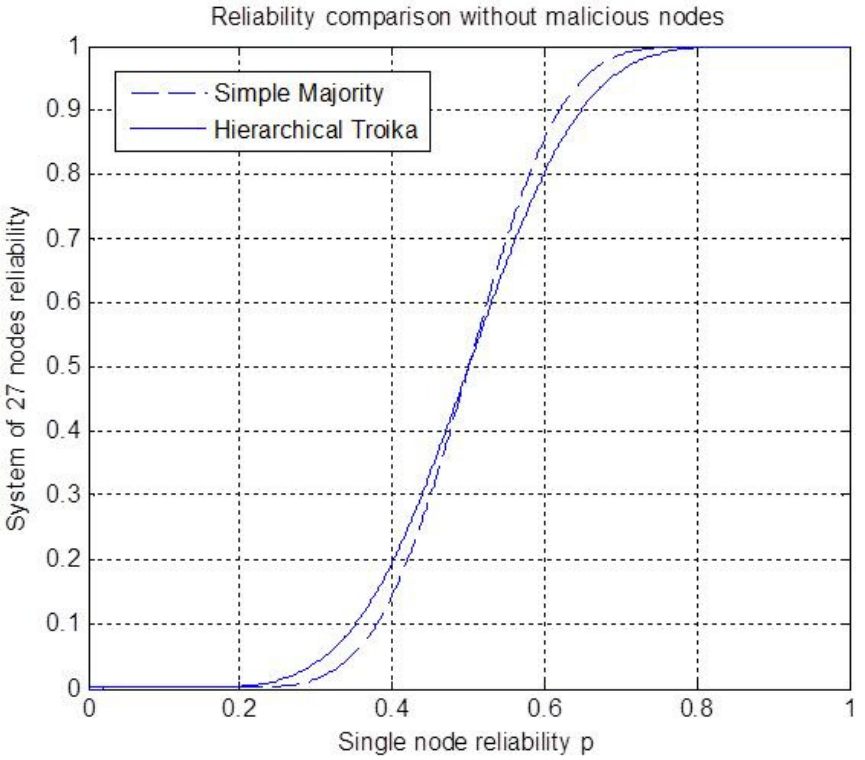

Figure 6. Troika vs. Majority group of 27 nodes decision reliability

Given the recursive structure of hierarchical troika, we have:

$$P^T{}_{3N} = 3(P^T{}_N)^2 - 2(P^T{}_N)^3. \quad (8)$$

Equations (4) and (8) allow a direct comparison of simple majority for any number of nodes that is a power of 3.

5.2. Measuring a Malicious Node's Influence

One metric we can use to measure a malicious node's influence is the probability that a single vote changes the aggregate decision. Again, we compare troika and simple majority using 9 and 27 nodes because they are powers of 3.

Using simple majority with an odd number of nodes, a single malicious node can change the aggregate decision if and only if the vote from other nodes breaks even. The probability of a tie vote in a simple majority is:

$$C_N = \binom{N-1}{m-1} p^{m-1}(1-p)^{m-1}, \text{with } m = \frac{N+1}{2}. \quad (9)$$

Using hierarchical troika, the process is different. First, we can see that if there are only three nodes, a malicious node changes the aggregate decision if the two other nodes have different votes, that happens with probability

$$C^T{}_3 = 2p(1-p). \quad (10)$$

Second, with 9 nodes, there are two decision layers (see Figure 4). A single node can influence the 9 nodes decision if at the first layer the two other nodes have different votes (which happen with probability $C^T{}_3 = 2p(1-p)$) and, at the second layer, the two logical connections have different results (which happen with probability $2P^T{}_3(1-P^T{}_3)$). Thus, we have:

$$C^T{}_9 = [2p(1-p)][2P^T{}_3(1-P^T{}_3)]. \quad (11)$$

More generally, we have

$$C^T{}_N = \prod_{k=1}^{\log_3 N} 2P^T{}_k(1-P^T{}_k)\,, k \text{ a power of 3.} \quad (12)$$

Figure 7 shows that a malicious node has a stronger influence on simple majority than on hierarchical troika if $\frac{1}{3} < p < \frac{2}{3}$. The reverse is true elsewhere. Further, if we take the integral for all values of p $(0 \le p \le 1)$, hierarchical troika and simple majority have equal results. Figure 8 shows a similar result to Figure 7 but with the interval in which hierarchical troika is more effective than simple majority is reduced to $0.4 < p < 0.6$. We foresee that this interval will continue to be reduced as the number of nodes increases.

Recall that in Section 4 we considered highly reliable nodes and showed that hierarchical troika will outperform simple majority if a high proportion of nodes (approximately more than 50%) are compromised. Thus, we anticipate that when nodes are highly reliable $\left(\frac{2}{3} < p \le 1 \text{ for 9 nodes}\right)$, if there is a small proportion of compromised nodes, simple majority should be used. However, as the number of compromised nodes increases, there must be a critical proportion above which hierarchical troika is superior to simple majority.

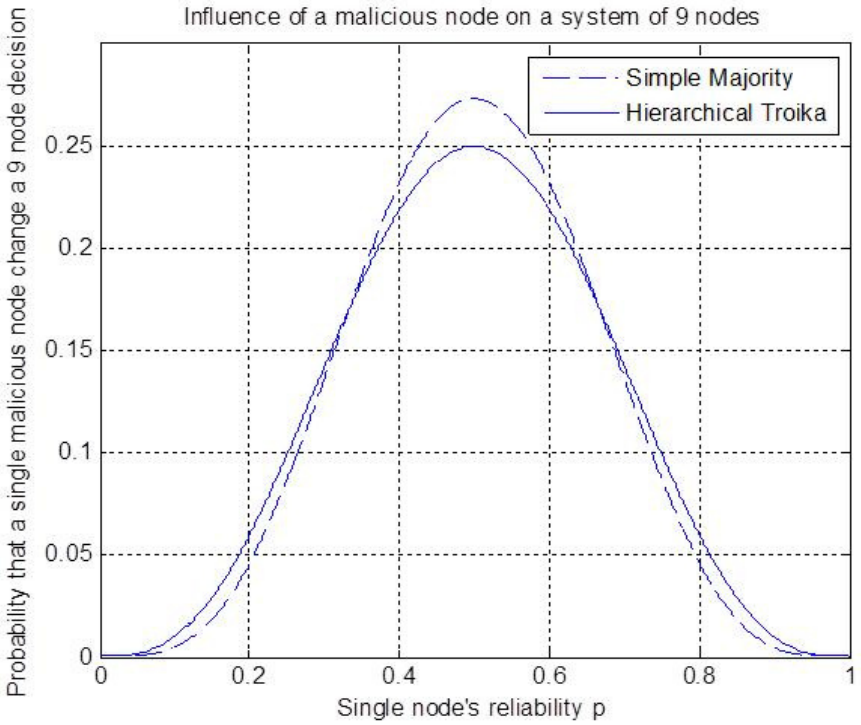

Figure 7. Troika vs. Majority in the mitigation of malicious nodes' vote

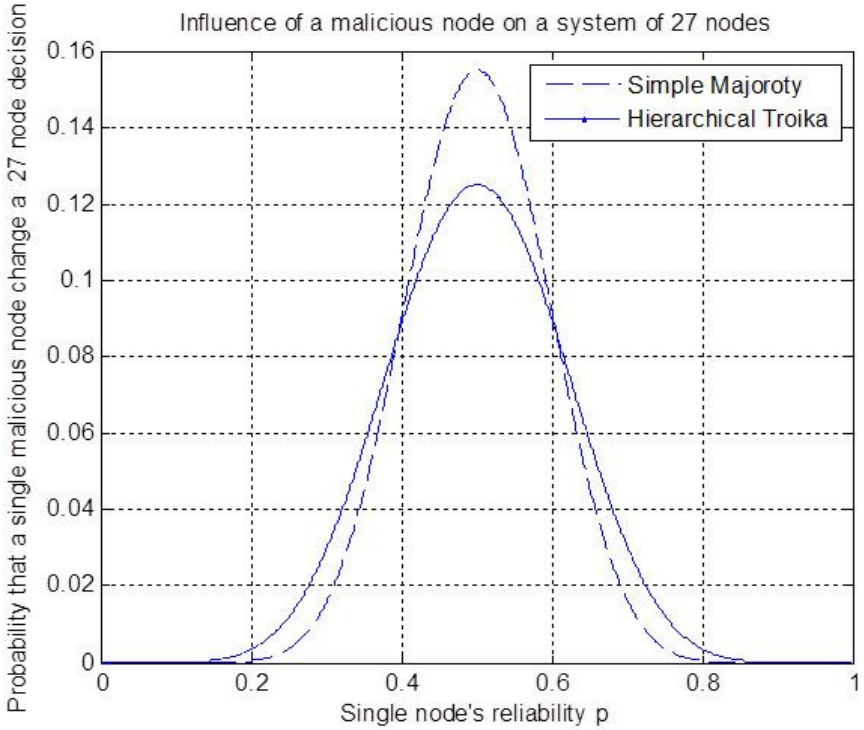

Figure 8. Troika vs. Majority in the mitigation of malicious nodes' vote

In this comparison, we have focused our attention to simple majority and hierarchical troika. In fact, those are the two extreme structures. For instance, the cluster of Table 1 and 2 can be represented as hierarchical *5*. We may also have hierarchical *7, 9*…, and so on. Furthermore, simple majority can be represented as hierarchical *N* with the entire vote aggregated in one step.

The properties of hierarchical *5, 7*, and others can be inferred from the comparison of simple majority and hierarchical troika. For instance, in Subsection 5.1, we can easily infer that if $0.5 < p \leq 1$, the aggregate decision reliability must be such that: simple majority>…>hierarchical 5> hierarchical troika.

5.3. Replica Diversity

Up to now we have assumed that the replicas are *functional replicas* – that is, they perform the same function but none are identical. This prevents a monoculture that would be susceptible to a global vulnerability. However, as the number of replicas' types requiring diversification increases the associated costs in design, development, and testing may become prohibitive. This is because it is easier to duplicate a node rather than creating an entirely new one. Therefore, we now consider diversity in a more restrictive sense where the number of functional replicas is, due to cost constraints, limited to a small number of types.

We have conducted an analysis with three types of replicas within a voting mechanism. This is to guarantee that the other two types of replicas can reach the correct result in case one fails. To increase the aggregate reliability, we have used three nodes of each type. Thus, we have a total of 9 nodes. The 9 nodes are just a prototype for a possible large scale replication. Each node has a reliability p ($0 \leq p \leq 1$) regardless of its type. The replicas of each type are a perfect copy of each other. Therefore, we assume that a malicious attack that compromises one node will also compromise all the nodes of the same type instantaneously. However, we assume that nodes of different types cannot be instantaneously compromised with the same attack. That is because the nodes of different types incorporate sufficient diversity, for example, running different software or having undergone a different fabrication process. If they incorporate diversity, then we assume that they have different vulnerabilities and hence fail independently due to attack.

In the first experiment, we consider that no replica has failed. We compare two node arrangements: simple majority and hierarchical troika. Figure 5 and 6 and Subsection 5.1 indicate that simple majority is superior to hierarchical troika when the node reliability is greater than 0.5.

In the second experiment, we consider the failure of three nodes of the same type. We again compare the performance of a simple majority and hierarchical troika. This time, we focus on two forms of node arrangement into the troika: homogeneous (see Figure 9) and heterogeneous (see Figure 10). We assume in this analysis that a compromised node never votes truthfully; therefore, the reliability of a compromised node is $p = 0$.

In the case of a simple majority, at least 5 out of 9 nodes have to vote correctly in order for the aggregate result to be correct. Given that three nodes are compromised and cannot vote correctly, there are only two ways that the aggregate result can be correct. The first possibility is that all the six uncompromised nodes vote correctly, which happens with probability p^6. The second alternative is that five of the six uncompromised nodes vote correctly while one votes incorrectly, which happens with probability $\binom{6}{5}p^5(1-p) = 6p^5 - 6p^6$. In short, when simple majority is used to aggregate the vote and three nodes of the same type are compromised, the probability P^{SM} that the majority will be correct is:

$$P^{SM} = 6p^5 - 5p^6. \qquad (13)$$

There are several ways to arrange 9 nodes of three types (three of each type) into the troika as in Figure 9 and 10. There is a total of $\binom{9}{3} * \binom{6}{3} * \binom{3}{3}$ or 1680 possibilities. Among those 1680 possibilities, one is a homogeneous troika (Figure 9) and the 1679 others are Heterogeneous. We use Figure 10 to get some insight into the characteristics of the different forms that are possible with Heterogeneous Troika. In fact, other forms of heterogeneous troika will combine the features of Figures 9 and 10.

Let us consider that the 9 nodes are arranged as in Figure 9 (homogeneous troika) and one type has failed, say type 3 (the hexagon). Thus, the troika containing the three hexagons must yield the wrong result. Therefore, the two other troikas (represented by the square and pentagon) must simultaneously send the correct result in order for the aggregate result to be correct. In fact, one troika yields the correct result if at least two out of three nodes vote correctly. This occurs with probability $p^3 + \binom{3}{2}p^2(1-p) = 3p^2 - 2p^3$. Then, two troikas will yield the correct result with probability $(3p^2 - 2p^3)^2 = 4p^6 - 12p^5 + 9p^4$. In summary, when a homogeneous troika is used to aggregate the vote as shown in Figure 9 and three nodes of the same type are compromised, the probability P^{HoT} that the aggregate result will be correct is:

$$P^{HoT} = 4p^6 - 12p^5 + 9p^4 \qquad (14)$$

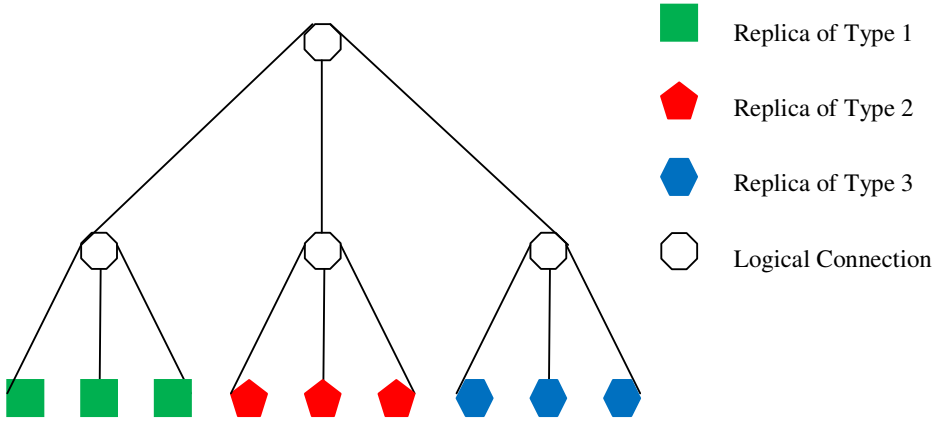

Figure 9: Homogeneous Troika

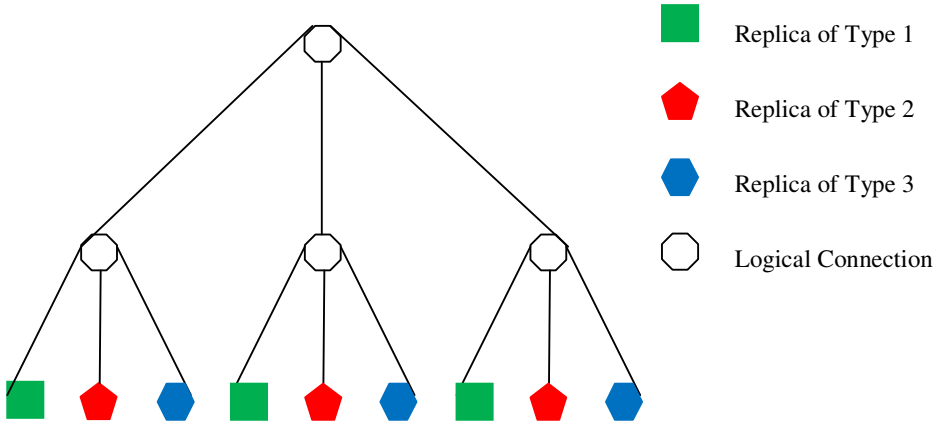

Figure 10: Heterogeneous Troika

Let us consider that the 9 nodes are arranged as in Figure 10 (Heterogeneous Troika) and assume that one type fails, say type 3 (the hexagon). In each troika, the result will be correct if and only if the troikas represented by the square and the pentagon vote correctly, which will happen with probability p^2. Since there are three troikas, the aggregate result is correct if at least two of the three troikas yield the correct result. Therefore, the probability P^{HeT} that the aggregate result will be correct is:

$$P^{HeT} = (p^2)^3 + \binom{3}{2}(p^2)^2(1-p^2) = 3p^4 - 2p^6 \qquad (15)$$

Figure 11 shows the resulting aggregate reliability to withstand both naturally-occurring faults and those that are attacker induced given that the replicas of one type have failed. We can see that a homogeneous troika is the best arrangement when there is a failure in the nodes; however, it merits repeating that a simple majority is favored when the malicious influence is absent (see Figure 5 and 6). This compels the consideration of deploying dynamic voting mechanisms: at the onset of the mission: a simple majority is instantiated for withstanding naturally occurring faults and later, as the mission advances, switching to homogeneous troika as warranted by the system's exposure to attack.

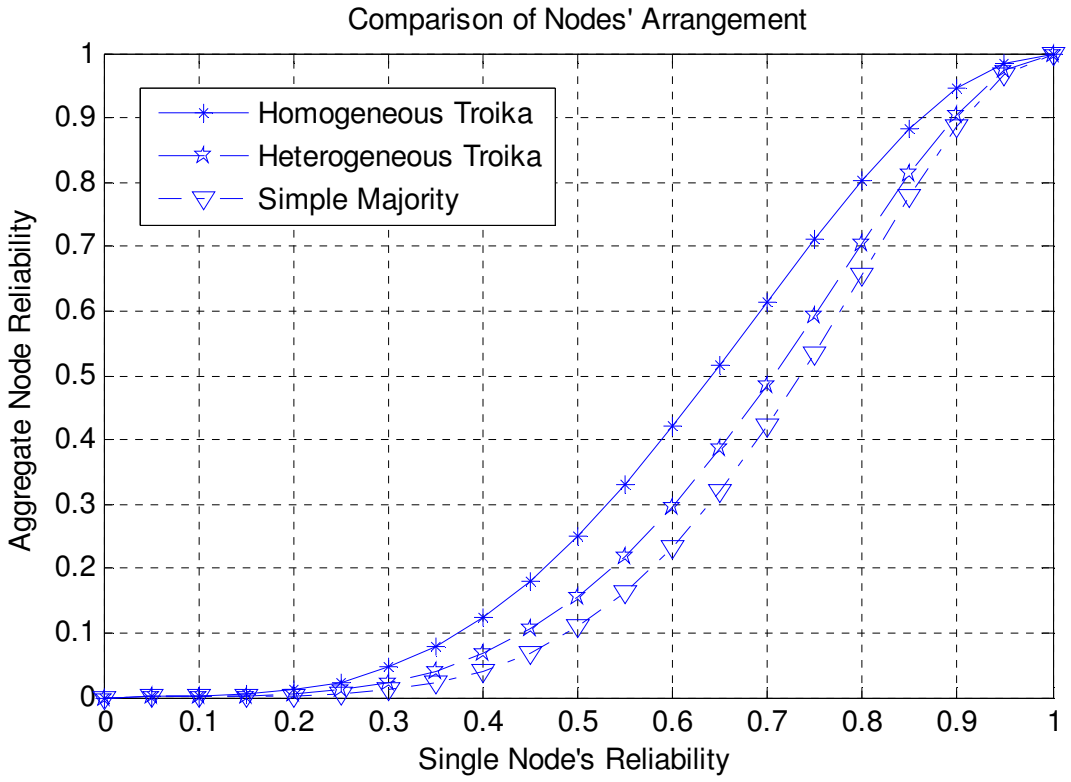

Figure 11: Comparison of Simple Majority and Hierarchical Troika Having a Failure of One Type.

6. CONCLUSIONS AND FUTURE WORKS

This research has compared the simple majority rule with a hierarchical decision process in a strategic environment. When there is no compromised node, we have shown that a simple majority decision rule is generally preferable to a hierarchical decision. Also, a hierarchical decision process should only be chosen when fast vote aggregation in a large scale network is the primary metric to consider. However, there are advantages of a hierarchical decision processes when nodes are potentially compromised and the application of diversity among the replicas is taken into account. In this case, the replicas of the same type should be grouped together as in Homogeneous Troika. We will investigate in more detail the scalability and speed of troika in our future research. Also, when investigating replicas' diversity, we have considered that the replicas of different types are vulnerable to different threats and that they fail independently. Our future research will look into the consequences of potential failure correlation among different replicas' types.

ACKNOWLEDGEMENTS

This research was performed while Charles Kamhoua and Joon Park held a National Research Council (NRC) Research Associateship Award at the Air Force Research Laboratory (AFRL). This research was supported by the Air Force Office of Scientific Research (AFOSR). Approved for Public Release; Distribution Unlimited: 88ABW-2012-3467 Dated18 June 2012.

REFERENCES

[1] Ge, Z., Figueiredo, D., Jaiswal, S., Gao, L.: Hierarchical structure of the logical Internet graph. in the Proceeding of SPIE 4526, 208, (2001).

[2] J. Knight and K. Sullivan, "Towards a definition of survivability," in the 3rd IEEE Information Survivability Workshop (ISW), Boston, MA, October 2000.

[3] H. Lipson and D. Fisher, "Survivability – a new technical and business perspective on security," in New Security Paradigms Workshop (NSPW99), Ontario, Canada, September 21-24, 1999.

[4] V. Westmark, "A definition for information system survivability," in System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, jan. 2004, p. 10 pp.

[5] J. S. Park and P. Chandramohan, "Component recovery approaches for survivable distributed systems," in the 37th Hawaii International Conference on Systems Sciences (HICSS-37), Big Island, HI, January 5-8, 2004.

[6] J. S. Park, P. Chandramohan, and J. Giordano, "Survivability models and implementations in large distributed environments," in the 16th IASTED (International Association of Science and Technology for Development) Conference on Parallel and Distributed Computing and Systems (PDCS), MIT, Cambridge, MA, November 8-10, 2004.

[7] J. S. Park, P. Chandramohan, G. Devarajan, and J. Giordano, "Trusted component sharing by runtime test and immunization for survivable distributed systems," in the 20th IFIP International Conference on Information Security (IFIP/SEC 2005), Chiba, Japan, May 30 – June 1, 2005.

[8] J. S. Park, P. Chandramohan, and J. Giordano, "Component-abnormality detection and immunization for survivable systems in large distributed environments," in Proceedings of the 8th IASTED (International Association of Science and Technology for Development) Conference on Software Engineering and Application (SEA). MIT, Cambridge, MA, USA: ACTA Press, November 2004, pp. 102–108.

[9] C. Kamhoua, K. Kwiat, J. Park "Surviving in Cyberspace: A Game Theoretic Approach" in the Journal of Communications, Special Issue on Future Directions in Computing and Networking, Academy Publisher, Vol. 7, NO 6, June 2012.

[10] C. Kamhoua, K. Kwiat, J. Park "A Binary Vote Based Comparison of Simple Majority and Hierarchical Decision for Survivable Networks" in the proceedings of the Third International Conference in Communication Security and Information Assurance (CSIA 2012) Delhi, India, May 2012. Published by Springer.

[11] Kwiat, K., Taylor, A., Zwicker, W., Hill, D., Wetzonis, S., Ren, S.: Analysis of binary voting algorithms for use in fault-tolerant and secure computing. International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt. December 2010.

[12] Wang, L., Li, Z., Ren, S., Kwiat, K.: Optimal Voting Strategy Against Rational Attackers. The Sixth International Conference on Risks and Security of Internet and Systems CRiSIS 2011, Timisoara, Romania, September 2011.

[13] Malki, D., Reiter, M.: Byzantine quorum systems. Distributed computer system, pp203-213 (1998).

[14] Bhattacharjee, S., Debroy, S., Chatterjee, M., Kwiat, K.: "Trust based Fusion over Noisy Channels throught Anomaly Detection in Cognitive Radio Networks" 4th International Conference on Security of Information and Networks (ACM SIN 2011), Sydney, Australia, November 2011.

[15] Ravindran, K., Rabby, M., Kwiat, K., Elmetwaly, S.: Replica Voting based Data Collection in Hostile Environments: A Case for QoS Assurance With Hierarchical Adaptation Control. Journal of Network and Systems Management, (2011).

[16] Ravindran, K., Rabby, M., Kwiat, K.. Data Collection in Hostile Environments: Adaptive Protocols and Case Studies. The Second International Conference on Adaptive and Self-Adaptive Systems and Applications.(2011).

[17] An, G., Park, J.: Cooperative component testing architecture in collaborating network environment. In Proceedings of the 4th International Conference on Autonomic and Trusted Computing (ATC), Lecture Notes in Computer Science (LNCS), pages 179-190,Hong Kong, China, July 11-13, 2007. Springer.

[18] Park, J., Chandramohan, P., Devarajan, G., Giordano, J.: Trusted component sharing by runtime test and immunization for survivable distributed systems. In Ryoichi Sasaki, Sihan Qing, Eiji Okamoto, and Hiroshi Yoshiura, editors, Security and Privacy in the Age of Ubiquitous Computing, pages127–142. Springer, 2005. Proceedings of the 20th IFIP TC11 International Conference on Information Security (IFIP/SEC), Chiba, Japan, May 30-June 1, (2005).

[19] Y. Amir, B. Coan, J. Kirsch, and J. Lane. 2011. Prime: Byzantine Replication under Attack. *IEEE Transactions on Dependable and Secure Computing (TDSC)* 8, 4 (July 2011), 564-577.

[20] Y. Dai, Y. Xiang, Y. Li, L. Xing, and G. Zhang, "Consequence oriented self-healing and autonomous diagnosis for highly reliable systems and software," Reliability, IEEE Transactions on, vol. PP, no. 99, p. 1, 2011.

[21] Q. Meng, R.-p. Zhou, and X.-h. Yang, "Design and implementation of an intrusion-tolerant self-healing application server," in Proceedings of the 2010 International Conference on Communications and Intelligence Information Security, ser. ICCIIS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 92–95.

[22] Condorcet: Essai sur l'application de l'analyse a la probabilite des decisions rendues a la pluralite des voix. Paris: Imprimerie Royale. (1785).

[23] Owen, G., Grofman, B., Feld, S.: Proving a Distribution-Free Generalization of the Condorcet Jury Theorem. Mathematical Social Sciences 17, 1-16, 1989.

[24] Grofman, B., Owen, G., Feld, S.: Thirteen Theorems in Search of the Truth. Theory and Decision 15, 261-278, (1983).

Authors

Charles A. Kamhoua received his B.S. in Electronic from the University of Douala/ENSET, Cameroon in 1999. He received his M.S. in Telecommunication and Networking and his PhD in Electrical Engineering from Florida International University in 2008 and 2011 respectively. He is currently a postdoctoral fellow at the Air Force Research Laboratory. His interdisciplinary research area includes game theory, cybersecurity, survivability, fault tolerant networks, and ad hoc networks. Dr. Kamhoua is a member of IEEE and the National Society of Black Engineer (NSBE). He is the recipient of the National Academies Postdoctoral Fellowship award at the Air Force Research Laboratory, Rome, New York in March 2011, extended in February 2012.

Patrick Hurley is a Senior Computer Engineer at Air Force Research Laboratory in Rome, New York. He received a Bachelor of Science degree in Computer Science from SUNY Oswego and a Master of Science degree in Computer Information Science from SUNY College of Technology. Mr. Hurley has worked closely with DARPA on various aspects of cyber defence over the past 15 years, especially focusing on survivability architectures, adaptive security, and advanced distributed systems technologies that lead to more agile and better managed systems. Mr. Hurley is currently the lead on two AFRL capability concepts focused on fighting through cyber attacks while maintaining mission essential function. He is a member of the IEEE and has over 20 technical publications.

Kevin A. Kwiat is a Principal Computer Engineer in the Cyber Science Branch of the U.S. Air Force Research Laboratory (AFRL) in Rome, New York where he has worked for over 28 years. He received a Ph.D. in Computer Engineering from Syracuse University. Dr. Kwiat holds 4 patents and has published more than hundred journal and conference papers. In addition to his duties with the Air Force, he is an adjunct professor of Computer Science at the State University of New York at Utica/Rome, an adjunct instructor of Computer Engineering at Syracuse University, and a Research Associate Professor with the University at Buffalo.

Joon S. Park is an associate professor at the School of Information Studies (*iSchool*), Syracuse University, Syracuse, New York, USA. Currently, he is the director of the Certificate of Advanced Study (CAS) in Information Security Management (ISM) at the iSchool. Dr. Park has been involved with research/education in information and systems security over the past decades. Before he joined the iSchool, he worked for the Center for High Assurance Computer Systems (CHACS) at the U.S. Naval Research Laboratory (NRL), Washington, D.C. He received a PhD in Information Technology, specialized in Information Security, from George Mason University, Fairfax, Virginia, in 1999.