Highly Reliable and Low-Cost Symbiotic IOT Devices and Systems

Bing-Yang Lin, Hsin-Wei Hung, Shu-Mei Tseng, Chi Chen, and Cheng-Wen Wu
Department of Electrical Engineering, National Tsing Hua University
Hsinchu 30013, Taiwan

***Abstract*—IOT has seen countless potential applications that can improve our lives dramatically, but after years of efforts by numerous companies and organizations, the beautiful dreams are yet to be realized. The main obstacles are cost and energy consumption constraints of the devices and systems, which still cannot be contained. As a step forward in improving the reliability and reducing the cost and energy consumption of IOT devices and systems, at ITC-Asia17 we proposed a high-level model for efficient design-space exploration, which is called the *symbiotic system* (SS) model. Based on that, we also proposed a quorum-sensing model for SS to improve the reliability of IOT systems that contain subsystems. Preliminary experimental result shows that it is possible for an IOT system to achieve low cost, low energy consumption, and high reliability. In this extended work we discuss more evaluation results, and show that by using quorum-sensing-based peer-repair, up to 97% of the faulty devices can be repaired even when the raw yield is only 30%. It shows that by adopting the proposed approach, there is hope in eliminating production test of symbiotic IOT devices in the future.**

Index Terms—built-in self-test (BIST), built-in self-repair (BISR), on-line testing, resilient system, symbiotic computing, symbiotic system, yield improvement.

I. Introduction

With the advent of Internet, social media and networking service, smart hand-held devices, etc., the amount of global electronic data is growing at an exponential rate [1]. According to IDC Research, the size of the world's data is doubling every two years and will reach 44 zettabytes by 2020 [2]. In the near future, applications like autonomous vehicles and Internet of Things (IOT) also are expected to generate a huge amount of data. According to [1], there is a large gap between the supply and demand of storages, resulting in an increasing demand for both memories and storages. In [3], it is predicted that the NAND flash (DRAM) demand will grow at a CAGR of 40% (25%) from 2016 to 2019. There are two common types of storage technologies: *hard disk drive* (HDD) and *solid state drive* (SSD). Although HDDs are cheaper, they have lower performance and consume more energy compared to SSDs. According to [4], the energy consumed by datacenters in the US in 2014 is estimated to be 70 billion KWh, which is about 1.8% of the total electricity consumption of the US. Such huge amount of energy consumption leads to environmental problems such as air pollution, climate change, and human-induced natural disasters. Though SSDs are more energy-efficient than HDDs, their costs are higher.

The cost and energy issues are becoming more and more important in IOT systems, while IOT has long been expected as the key driving force for the continuous growth of the semiconductor industry, after the growth driven by smart handheld devices and consumer electronics seem to saturate. Several years ago, both Ericsson [5] and Cisco [6] predicted that there will be 50 billion connected devices by 2020. However, due to sluggish world economy and energy constraints [7], they have lowered the predictions in more recent forecasts [8]. To support a major growth of connected devices, the costs of IOT devices must be contained. In addition, sustainability and energy consumption are also important issues that need to be addressed in the development of future IOT systems. In this paper, we focus on the cost reduction and sustainability (as contributed by reliability/lifetime) issues of the IOT systems. The cost reduction issue can be addressed in two respects, i.e., by reducing manufacturing cost, and by extending product lifetime (or improving its reliability). As the test cost represents a large portion of semiconductor cost, it is important to reduce or even eliminate its manufacturing test cost, without sacrificing its reliability in the meantime. Manufacturing tests are performed at the end of the manufacturing process to ensure the chip quality, or defect level. *Built-in self-test* (BIST) and *built-in self-repair* (BISR) [9, 10] are popular in semiconductor logic and memory products. They perform test and repair off-line, i.e., during test mode. After test (and repair), the circuit can return to normal operation mode if they are fault-free. For enhanced reliability during normal operation, conventional methods rely on some kinds of redundancy in general. For example, the traditional *N-modular redundancy* (NMR) uses multiple (*N*) copies of functionally identical hardware and/or software, which take the same inputs and vote for the correct result [11]. *Error correction code* (ECC) is another way of error detection and correction during normal operation, by using extra (redundant) parity bits to detect and recover corrupted data bits [12]. Time-redundancy techniques also are popular, which guarantee correct result by performing the calculation or transmission multiple times, so transient errors can be identified and removed. However, as these techniques improve the reliability of electronic devices, the *space (hardware) redundancy* that takes additional chip area increases the product cost, and the *time redundancy* raises the energy consumption in addition to greatly increased computation/communication time. To reduce the costs, memory and storage vendors have tried to increase the device density by reducing the cell feature size or storing more bits into a single cell. These in turn can lead to reliability issues. To turn IOT into reality, we need to take all these concerns into

978-1-5386-3413-4/17/$31.00 ©2017 IEEE

consideration during design. Therefore, a software platform that can perform fast evaluation is important.

In this work, we adopt symbiotic system model [7] to evaluate the cost and reliability (lifetime) issues in future storage systems, and adopt the model in designing and modeling the IOT devices and networks, where each IOT device is equipped with *repair resources*. In addition, as extended from [7], we show more detailed simulation results to observe the efficiency of *quorum-sensing-based peer-repair* [7]. In our experiments, we show the effects of peer-repair mechanism on lifetime improvements in different conditions. The peer-repair mechanism brings more benefits for storage systems with a wide range of failure rate than storage systems with an identical failure rate. This means that it is especially suitable for improving the lifetime of, e.g., real-time environmental monitoring IOT systems, where nodes can be scattered in different terrain, temperature, moisture, etc. Moreover, in long-term operation where we need to replace faulty devices regularly to maintain system operability, the peer-repair enhanced system normally can achieve a lower cost as compared with the self-repair-only system, depending on the actual device cost and the batch replacement cost.

The organization of this paper is as follows. In Sec. II, we review the IOT storage and device networks [7] for our evaluation. In Secs. III and IV, we evaluate the cost and reliability of the IOT storage network and general device network, respectively. In Sec. V, we evaluate the efficiency of quorum-sensing-based peer-repair. Finally, in Sec. IV, we make some conclusions based on the experimental results.

II. IOT STORAGE/DEVICE NETWORK

We adopt the *Symbiotic System* (SS) model as proposed in [7], and apply it to the novel electronic systems that might appear in the future. A *finite-state machine* (FSM), called *Symbiotic Machine* (SM) or *Symbiotic Device* (M_D), is used to model the SS. By definition, the SM consists of a *primary machine* (M_P), which is the functional system in the SS, and a *secondary machine* (M_S), which is the test system in the SS. The FSM model of the M_P, as in a storage device and a general device, is the same as the M_P defined in [7], while the only difference between them lies in the M_S.

A. Secondary Machine for Symbiotic Storage Device

In an SS of the storage node, the *Symbiotic Relationship* (SR) [7] of changing the internal states of each other can be achieved by sensing and repairing mechanism. According to current technology, sensing of faulty behaviors in M_P can be done by on-line test and repair. Repairing the abnormal conditions and bringing the SM back to normal operations can be divided into two categories: self-healing and resource allocation. Self-healing in a storage node can be any mechanism that corrects the faulty outputs, e.g., ECC. Resource allocation is the preparation of redundancies in case of the faulty condition that self-healing mechanism is not able to handle, e.g., spare blocks in SSD. The state diagram of M_S in a storage node is depicted in Fig. 1 [7]. In the figure, we can see that M_S consists of five states: Monitor/Report, Test, Correct/Self-Repair, Peer-Repair Request, and Peer-Repair Reply. Note that all the state transition control signals, except R_S that will be defined later, are binary signals, including MR (Monitor/Report), F (Failure), CF (Configuration), CR (Correctable) and PCF (Configuration requested from peer).

The structure and behavior of each symbiotic subsystem in the symbiotic storage system is abstracted from current storage systems. A storage node is composed of a number of *memory-units* (MUs), each of which may contain multiple *sub-units* (SUs). Each storage node can communicate with a certain number (denoted as ρ) of neighboring nodes, i.e., *peers*, depending on its communication capabilities. In Fig. 1, if M_S detects that M_P is abnormal and needs test, i.e., F = 1, M_S will enter the test mode and force M_P into Test/Config state [7]. Once a fault is detected in M_P and M_P needs configuration, CF = 1, M_S will try to bring M_P back to normal mode.

Let the size of the total spare resources be R, size of *self-repair resources* be R_S, and size of *peer-repair resources* be R_P, respectively. We assume that the spare resources are counted by the number of MUs, and that $R = R_S + R_P$. For example, when a fault occurs, M_S will check whether the number of faulty SUs exceeds correction capability of the storage node or not. If not, the faulty MU is correctable and CR = 1. If the MU is correctable and $R_S > 0$, M_S will allocate spare resources to perform self-repair, or ask for peer-repair if $R_S = 0$. Note that PCF = 1 if a peer requests for peer-repair. In that case, M_S will transit to the Peer-Repair Reply state.

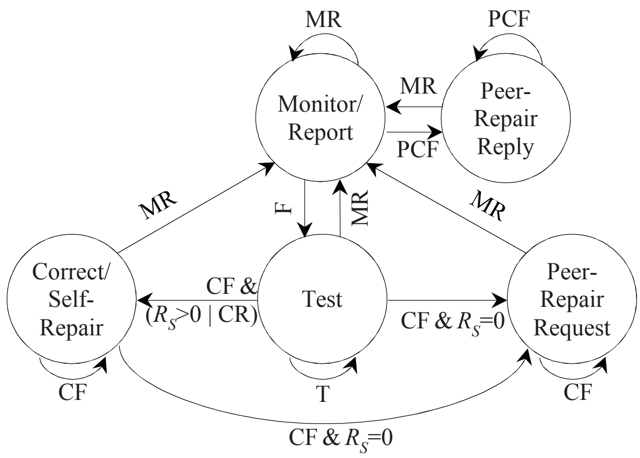


Fig. 1. State diagram for M_S of a symbiotic storage device [7].

B. Secondary Machine for Symbiotic IOT Device

The state diagram of the secondary machine of an IOT device based on the SS Model is shown in Fig. 2. The secondary machine, denoted as M_S, has a set of states Q_S = {Monitor, Test, Self-Repair, Peer-Repair Request, Peer-Repair Reply}. The signals MR (Monitor/Report), F (Fault), T (Test), R (repairable), CF (Configuration) and PCF (Configuration request from peer) are binary flags. Similar to the IOT storage node, the IOT device has a certain number of self-repair resources and peer-repair resources, denoted as R_S and R_P, respectively. Let the total number of spare resources be R, then $R = R_S + R_P$. Each of the spare resource in R can be used by M_S to repair its faulty M_P or one of its faulty neighboring M_P.

In normal condition, the IOT device regularly gathers information from the environment and goes into a low-power state to save energy, i.e., M_P regularly changes its state between Active and Idle. At the same time, M_S will stay in Monitor to monitor the signals and states of M_P and take action in case of any abnormal behavior (i.e., when a fault occurs). When there is a fault in M_P that is detected by M_S (F = 1), M_S will send M_P into Test/Repair and go to Test to test the faulty M_P. If the fault is repairable (R = 1) and self-repair resource is available ($R_S > 0$), M_S will allocate one self-repair resource to M_P and then, M_P will return to normal mode. If the fault is repairable but there is no available self-repair resource, M_S will change to Peer-Repair Request and try to bring back its faulty M_P by using *peer-repair*. First, M_S will ask for one of its peer for allocating by setting their PCF = 1. Then, the peer's M_S will change to Peer-Repair Reply. If the peer's $R_P > 0$, the repair is successful. If not, the M_S will turn to next peer for help until all its peers run out of peer-repair resources.

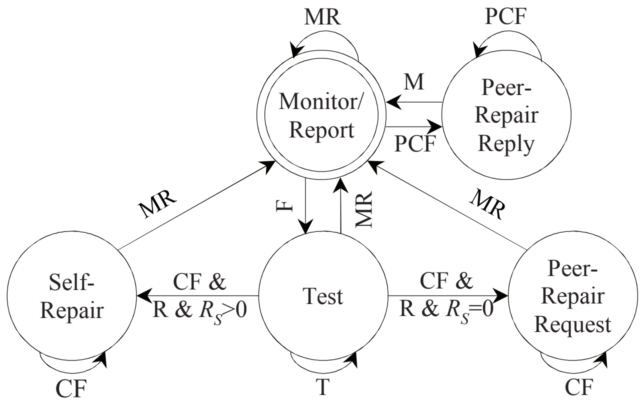


Fig. 2. State diagram for M_S of a symbiotic IOT device.

C. Network

An IOT device network is composed of IOT devices (i.e., nodes), where each IOT device is connected to a certain number of neighboring devices (i.e., *peers*). We define the average number of peers of the IOT devices in a network as the *number of peers*, denoted as ρ, which is usually constrained by the communication ability or energy. To maintain the correct functionality of the network, a minimum percentage of working nodes need to be guaranteed, which is the working threshold of the network, T_W. Figure 3 shows an example of a small IOT device network consisting of 20 nodes, where ρ = 4. The black node (node 7) represents a faulty node, which can turn to its peers (node 4, 8, 10, 11) for help if it has no self-repair resource left. If T_W is set to 90%, e.g., the system will fail when the third faulty node that cannot be repaired occurs.

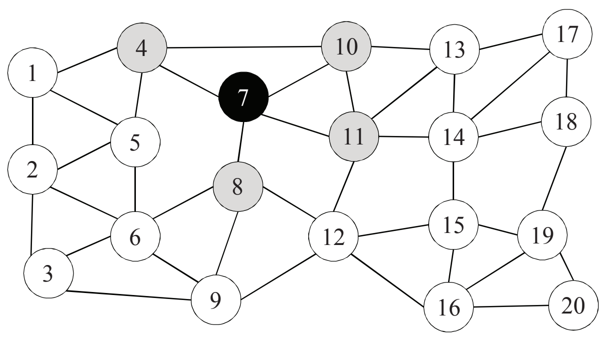


Fig. 3. An IOT device network example.

D. Cost Model

Some IOT device networks are expected to be deployed in a very wide range of area, and should last for a long time, e.g., the sensor network for gathering real-time traffic data, the one for keeping track of the soil water content, and the one for monitoring electricity usage in a power grid. Note that defects in IOT devices during manufacturing are inevitable, and during field use, device aging and environmental hazards will introduce other factors of failure, so the failure rate of the devices and system normally will grow over time, even when they are equipped with (on-line) repair resources. To maintain the functionality of a network, manual replacements normally are required. However, for many such applications, replacing failed devices in the field to maintain system functionality can be very expensive, if possible at all.

We model the operation cost, denoted as C_O, of an IOT device network as the summation of device cost (C_D) and replacement cost (C_R):

$$C_O = N_D \cdot C_D + N_R \cdot C_R, \tag{1}$$

where N_D is the total number of devices used throughout the entire period of operation, which includes the initially deployed devices and the devices used to replace the faulty ones, and N_R is the number of batch replacements.

III. Evaluation of IOT Storage Network

We use the *failures in time* (FIT) as a figure to represent the average failure rate, i.e., failures per billion hours of operation of each storage device. In the experiments, faults can occur in multiple MUs, with the total number of faults determined by Poisson distribution, whose mean is in FIT/Mb. Note that we define FIT as the number of faulty bits per billion hours of operation, which is also used to represent the reliability of a storage node. The higher the FIT/Mb, the lower the reliability of each storage device.

A. Relative Lifetime Increment with Identical Failure Rate

Consider a 1,000-node SS, where T_W = 900, i.e., 10% failed nodes can be tolerated. Assume each storage node has 4GB capacity, and ρ = 3. Also, redundant or spare resources are denoted in percentage, e.g., 0.01% means there are 0.01% additional MUs that are used for only self-repair or both self-repair and peer-repair. Faults are randomly distributed among multiple MUs, and FIT/Mb = 60K for each storage node. In Fig. 4, we compare the lifetimes of the storage systems with and without peer-repair mechanism under different amount of spare resources. We can see that the lifetime increases linearly with the size of spare resources. To show the impacts of peer-repair mechanism on lifetime, we define the *relative lifetime increment* (L_i) by peer-repair (as compared with pure self-repair) as follows:

$$L_i = \left(\frac{L_{pr} - L_{sr}}{L_{sr}}\right) \times 100\%,$$

where L_{pr} is the lifetime of the storage system with peer-repair mechanism and spare resources R, and L_{sr} is the lifetime of the storage system with only self-repair mechanism ($R_S = R$). We then calculate L_i for the same case, and the results are plotted in Fig. 5, where we magnify the region when R is very small. Note that $L_{pr} = L_{sr} = L$ (original lifetime without repair) when $R = 0\%$, which is not plotted in Fig. 5. The first point in the figure is when $R = 1.86\text{E-}7\%$, where L_i is about 68.22%. In the figure, we can see that when we increase R, the effectiveness of peer-repair over self-repair drops quickly. In this experimental case, when $R > 0.001\%$, L_i stays lower than 0.99%, which means that the relative lifetime increment by peer-repair is low. This is because we assume identical failure rates in all the storage nodes. When a given node is running out of self-repair resources, the available peer-repair resources in its peers may also be very few or even zero. Under this condition, the advantage of peer-repair on improving the lifetime is not much better than self-repair, unless when R is very small.

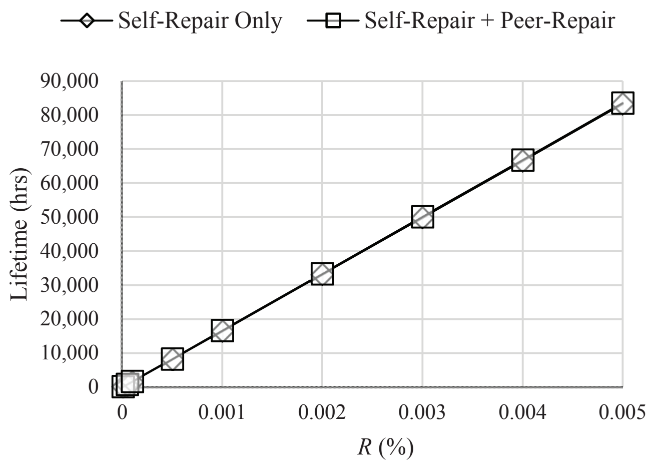


Fig. 4. Lifetime with/without peer-repair mechanism, with respect to different R, in a system which contains nodes with identical failure rates (where FIT/Mb = 60K).

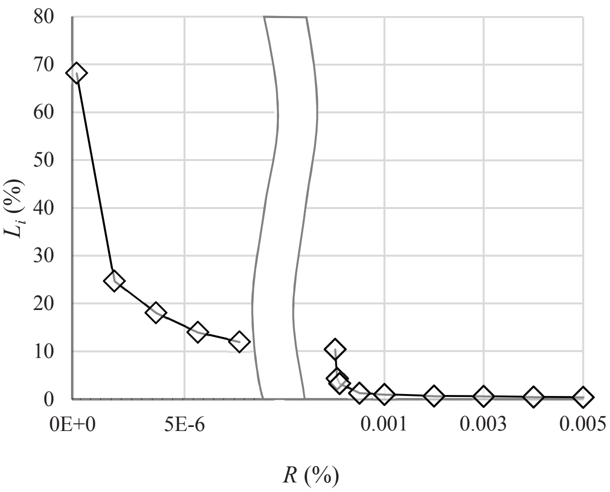


Fig. 5. Relative lifetime increment (L_i) by peer-repair mechanism, with respect to different R, in a system which contains nodes with identical failure rates.

B. Relative Lifetime Increment with Plural Failure Rates

According to the research on the faulty behavior of current large storage (and memory) systems [13-15], most of the faults actually occur in a relatively small portion of the storages (memories). Therefore, we also conduct an experiment and simulate the conditions that are closer to the observation results in reality. In this experiment, the average number of faults injected in a system is the same as the previous experiment, i.e., FIT/Mb = 60K, while the failure rates are higher in 20% of the storages (FIT/Mb = 100K), and lower in the other 80% (FIT/Mb = 50K). We replot Fig. 4, and the result is shown in Fig. 6. The lifetime difference between the two storage systems with and without peer-repair mechanism is clearly larger in Fig. 6 as compared to the results in Fig. 4, meaning that peer-repair is more effective when the failure rate is unevenly distributed in the system. We also replot Fig. 5 as shown in Fig. 7, without magnifying the region where R is very small. We can see that the L_i is quite significant: around 60%, and it does not decline when $R > 0.001\%$. Therefore, peer-repair is more effective than self-repair in this case, in the entire range of R as simulated. Based on the previous experimental results, where there is a large variation in L_i, we conduct the following experiment to show the

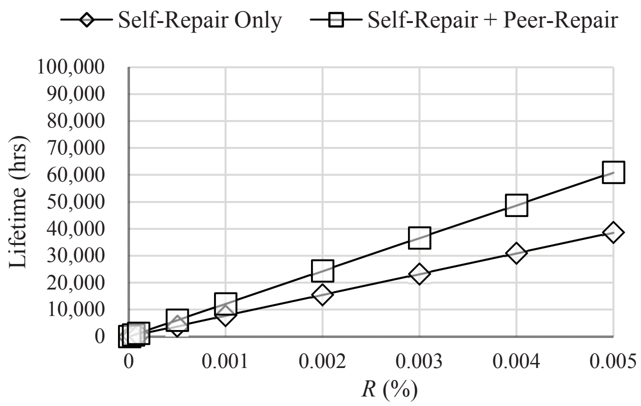


Fig. 6. Lifetime with/without peer-repair mechanism, with respect to different R, in a system with two different failure rates, where 20% of the nodes have a higher failure rate than the other 80%.

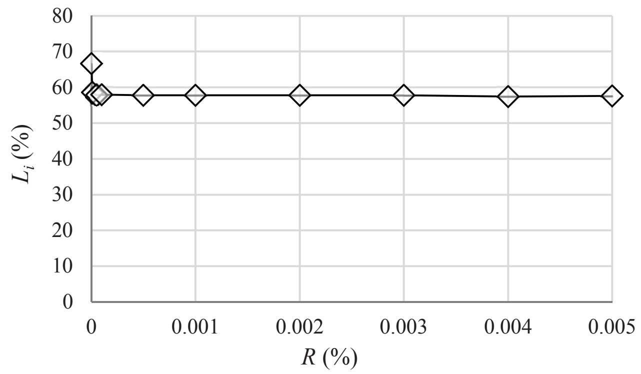


Fig. 7. Relative lifetime increment (L_i) by peer-repair mechanism under different R in a system with two different failure rates, where 20% of the nodes have a higher failure rate than the other 80%.

condition that favors peer-repair in terms of lifetime improvement. In the experiment, the FIT/Mb of each storage node in a system is randomly determined by normal distribution, with mean μ and standard deviation σ. The lifetime of the storage systems with R = 0.001%, μ = 60K FIT/Mb, and = 0, 5K, 10K, 15K, and 20K FIT/Mb are shown in Fig. 8. We can see that lifetime decreases as σ increases. Also, the lifetime difference between the peer-repair and self-repair systems increases with respect to σ. Therefore, L_i will be larger when σ increases. The results clearly illustrate the condition when peer-repair mechanism can be more helpful if the storage nodes have sufficient amount of R. It is more helpful as well when the variation of failure rates among nodes is large. The observation is important for, e.g., future IOT systems where nodes used for real-time environmental monitoring can spread out over a wide area, where the failure rates can vary a lot from place to place.

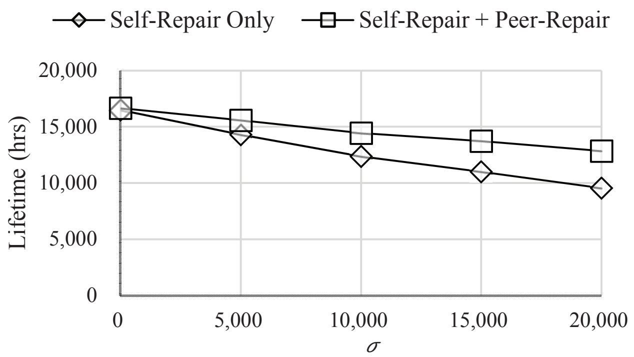


Fig. 8. Lifetime with/without peer-repair mechanism, with respect to different σ, in a system which contains nodes with failure rates in normal distribution.

C. *Cost Reduction With Different Replacement Policies*

After the evaluation of lifetime improvement by the peer-repair mechanism, we explore the cost reduction by peer-repair for 10 years of operation. We let C_D = 1 and sweep the C_R from 50 to 1,000 to evaluate the effects of replacement costs. Note that we only replace the failed nodes upon system failure in every replacement. The 10-year total costs of the storage systems, where (μ, σ) = (100K, 10K) and R = 0.001%, with/without peer-repair for different C_R are shown in Fig. 9.

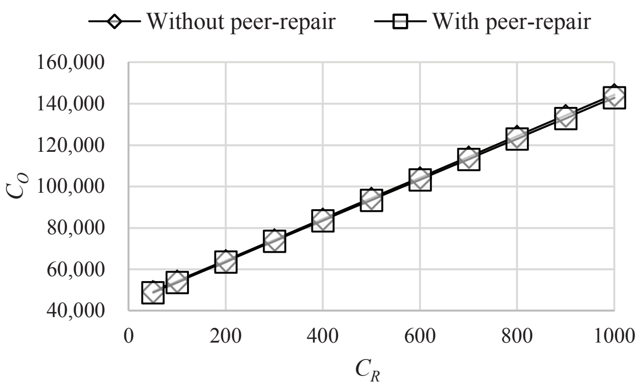


Fig. 9. 10-year total operation costs (C_O) of the storage systems with/without peer-repair mechanism for different C_R.

We can see that the C_O of the storage system with peer-repair mechanism is always larger than that without peer-repair mechanism, in the entire simulated range of C_R. This means that merely replacing all the failed nodes upon system failure is not enough, since the nodes with small R will fail very quickly after replacement. Therefore, we evaluate the C_O under different replacement policies: not only replacing the failed nodes but also replacing the good nodes whose R drops below a certain *replacement threshold* (T_R). To explain the experimental results more clearly, we also show the change of N_D and N_R for different T_R in Figs. 10 and 11, respectively. In the figures, when T_R = 0, both N_D and N_R of the storage system with peer-repair are larger than that without peer-repair (i.e., with self-repair only). Therefore, in Fig. 9, no matter how C_R changes, the storage systems with peer-repair will always have higher cost. The experimental results of the total operation costs (C_O) for different replacement policies and C_R is shown in Fig. 12. We can see that the storage systems with peer-repair mechanism have lower costs than those without peer-repair mechanism when $C_R \geq 200$. Although the difference in N_D when T_R is around 0.1 is small, the N_R drops very quickly when T_R = 0 ~ 0.1, as shown in Fig. 11. Therefore, as C_R becomes higher, the replacement costs will dominate the total costs. It shows that a system with peer-repair has a lower total cost than one without peer-repair and T_R = 0. Note that it also illustrates the effects of peer-repair mechanism: it utilizes the repair resources more evenly. If we only replace the failed nodes, the R of its peers must be 0, so they will also fail very quickly. Therefore, the time between replacements will decrease, hence increasing C_O. We

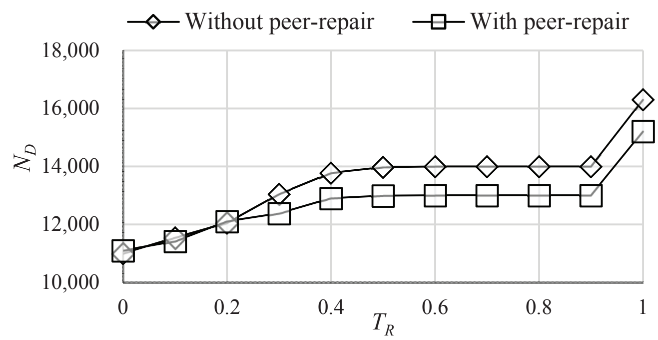


Fig. 10. Number of replaced devices (N_D) of the storage systems with/without peer-repair mechanism for different T_R in 10 years.

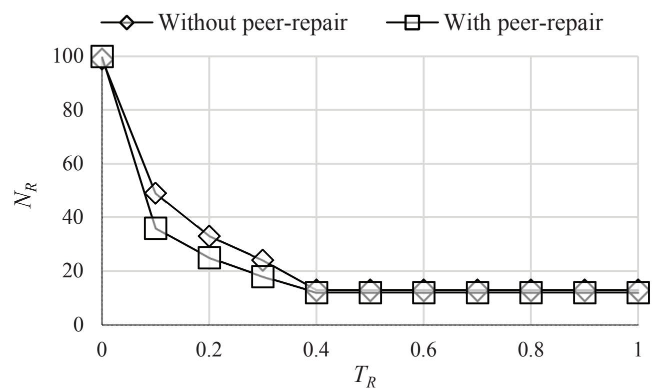


Fig. 11. Number of replacements (N_R) with/without peer-repair mechanism for different T_R in 10 years.

also show in Fig. 13 that, to achieve the lowest cost, $T_R = 0.4 \sim 0.8$.

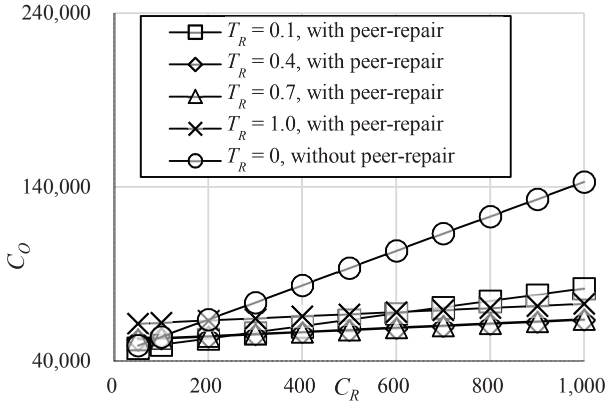


Fig. 12. Operation costs (C_O) of the storage systems with/without peer-repair mechanism for different T_R and C_R in 10 years.

In Fig. 13, we plot the C_O of the storage system with peer-repair mechanism with different C_R and T_R, to show the effect of T_R more clearly. We can see that when $C_R \leq 200$, the lowest cost lies around $T_R = 0.1$. As C_R becomes higher, the lowest cost moves from $T_R = 0.1$ to $T_R = 0.4 \sim 0.9$. Moreover, although N_R is the smallest when $T_R = 1$, C_O rises again when $T_R = 1$ as shown in Fig. 13. This is due to the rapid rise of N_D when $T_R = 1$ as indicated in Fig. 10.

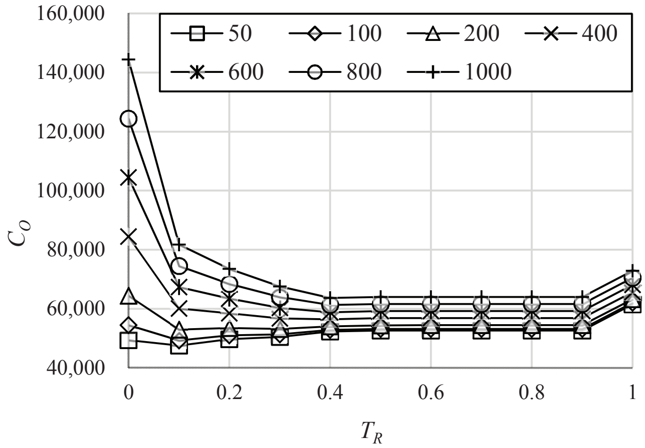


Fig. 13. Operation costs (C_O) of the storage systems with peer-repair for different T_R and C_R in 10 years.

D. Relative Lifetime Increment with Plural Failure Rates

To reduce the costs in future IOT storage systems, we explore the possibility of reducing the test costs during fabrication. The more the manufacturer spends on testing, the more high-FIT storages will be detected and discarded before shipping. Therefore, we simulate the lifetime of the test-enhanced storage system with different levels of testing efforts, and then compare the experimental results with the storage systems with peer-repair mechanism and different amount of additional spare resources, i.e., extra resources on top of the original 0.001%. Different levels of testing efforts are simulated, assuming there are no outlier cases with failure rates higher than $\mu + n \cdot \sigma$, where $n \leq 3$. The results of the experiments are shown in Table 1, where μ = 50K FIT/Mb and σ = 5K, 10K FIT/Mb.

We can see that by merely adding peer-repair mechanism with no additional spare resources, the lifetime of the original storage system can be longer than the lifetime of the test-enhanced storage systems with $n \geq 1$. In addition, with 5% additional spare resources when σ = 5K, the lifetime of the original storage system can be longer than the test-enhanced storage system with $n = 0.5$. Note that the testing costs can be higher than the costs of extra spare resources in advanced systems. To maintain a comparable lifetime with lower cost, therefore, it may be better by investing more in peer-repair while reducing testing.

Table 1. Simulated Lifetime of Test-Enhanced/Original Storage Systems Without/With Peer-Repair for (μ, σ) = (50K, 5K) FIT/Mb

Test-enhanced Storage Systems without peer-repair		Storage Systems with peer-repair	
n	Lifetime (hrs)	Additional Spare Resources	Lifetime (hrs)
0.5	21515.76	0%	20643.47
1	20237.66	1%	20988.31
1.5	19429.81	5%	21681.43
2	18785.51	10%	22659.06
2.5	18607.31	15%	23893.31
3	18554.61	20%	24817.52

IV. Evaluation of IOT Device Network

We build a simulator to evaluate the reliability and observe the behavior of the IOT device network under various conditions. Table 2 defines the input and output parameters that we use in the experiment. We simulate a large-scale IOT device network, consisting of 1,000 nodes. Each node is connected to 5 peers, as our previous work [7] has shown that $\rho = 5$ is good enough to achieve the ideal repair resources utilization. We assume that the IOT devices in the network have the same reliability, represented in mean time between failures (MTBF), i.e., $1/\lambda$, where λ is the failure rate. We assume that each repair resource can be used to repair any of the faults we inject with repair rate, μ. In a real system, neighborhood and proximity has to be considered due to communication complexities, so we assume R to be in the range [0, 10]. In the experiment, we will further show that R is even more restricted by other factors. The simulation will stop when the number of fault-free nodes drops below T_W. In addition, when we are analyzing long-term operation where replacements are involved whenever the system reaches T_W, the simulation will continue until we reach the given operation time, T. The values for all the input variables that we use in the simulations are shown in the table.

Table 2. Input and Output Parameters

Input	Description	Value
N_{DI}	No. initially deployed devices	1,000
R_S	No. self-repair resources	0 - 5
R	No. total repair resources	0 - 5
ρ	No. peers of a device	5
$1/\lambda$	Reliability of a device in MTBF	100,000 (hours)

μ	Repair rate	100% - 60 %
T_W	System working threshold	99% - 80%
T	Operation time	
C_D	Cost of a device	1 (normalized)
C_R	Cost of one batch replacement	0 - 1,000
Output	**Description**	
U	Repair resource utilization	
L	System lifetime	
N_D	No. total devices used	
N_R	No. batch replacements	
C_O	Operation cost	

A. Working Threshold

We examine the effect of peer-repair by observing the system lifetime under different T_W and R values. For the system using self-repair, R_S ranges from 0 to 5. For the system using peer-repair, where self-repair is automatically implied, R ranges from 0 to 5, which can be used either for self-repair or peer-repair. The experimental results are plotted in Fig. 14, where the system lifetimes for those with lower T_W are normalized to the one with $T_W = 99\%$.

As shown in Figs. 14(a) and (b), when self-repair and peer-repair are not used (R = 0), T_W plays an important role in the reliability of the system. In the self-repair system (Fig. 14(a)), e.g., when we merely increase the number of tolerable failure nodes from 10 ($T_W = 99\%$) to 50 ($T_W = 95\%$), the system lifetime increases by 4.4 times. This suggests that, for an IOT device network without spare repair resources, software fault-tolerance techniques can be essential and effective in improving system reliability. Once there are repair resources in the devices, the effect of T_W decreases. Especially for peer-repair systems, even if there is only one repair resource, the system lifetime for $T_W = 99\%$ is close to that for $T_W = 80\%$. The result shows that peer-repair at the device level is able to make up a low degree of fault

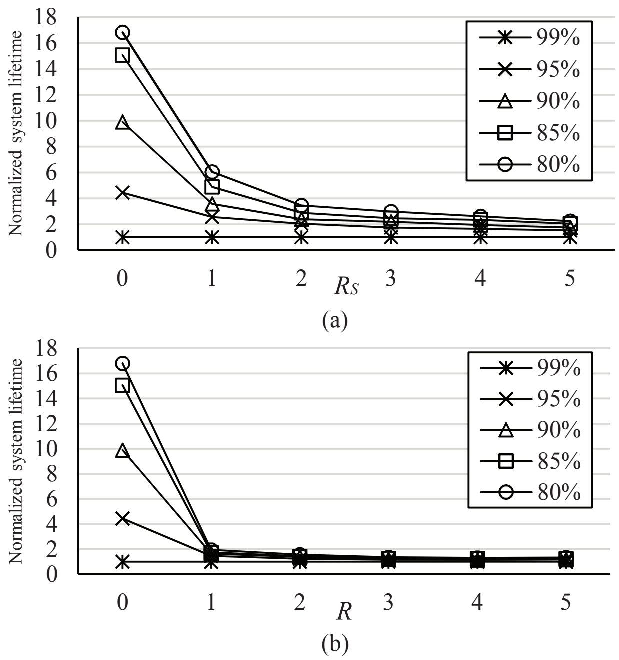


Fig. 14. Normalized system lifetime of the (a) self-repair (b) peer-repair system under different T_W and R.

tolerance (e.g., with much fewer redundant devices or nodes) at the system level.

B. Repair-Rate

In our previous work [7], we have shown that IOT device network with peer-repair has longer lifetime and better repair resource utilization. Figure 15 shows the trend of system lifetime (L) as R grows, where we can see that L is almost proportional to R, with or without peer-repair. On average, systems with peer-repair achieve 50% longer lifetime as compared with those with self-repair. However, considering manufacturing cost, design complexity and communication complexity, large R can be infeasible in real implementation. In addition, the effect of additional repair resources can be limited under some conditions, especially when the repair rate is not perfect (μ < 100%).

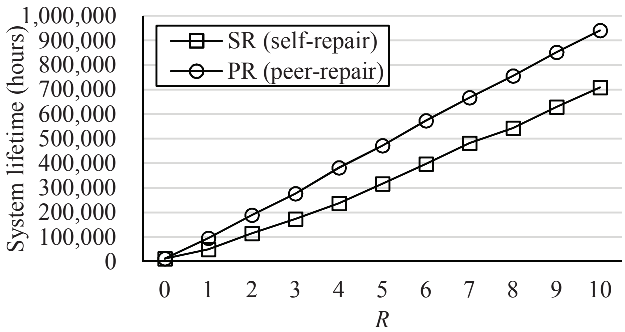
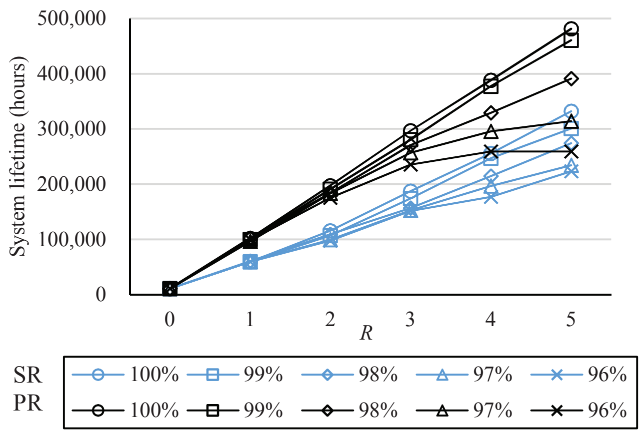
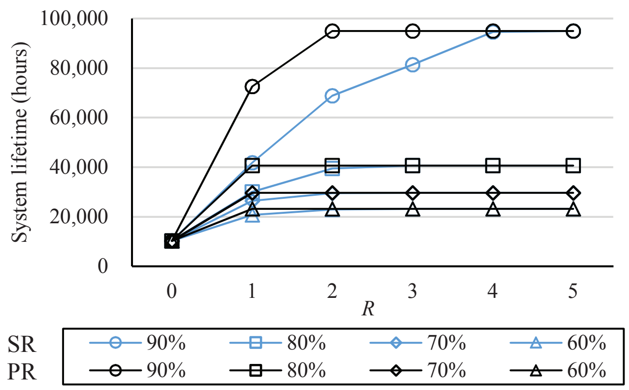


Fig. 15. System lifetime of a IOT device network with respect to R.

Figure 16 shows the lifetime of IOT device networks with different ranges of μ. In Fig. 16(a), for systems with near perfect μ (100% - 96%), the repair resources remain effective when R < 4. When $R \geq 4$ and μ < 97% the system lifetime quickly saturates, which is then dominated by μ. The situation gets even worse when μ is relatively low. As shown in Fig. 16(b), low μ becomes the bottleneck of the system lifetime. No matter how many repair resources are added, the system lifetime is not growing anymore, i.e., the additional repair resources are wasted. It shows that reliability analysis should be done at the system level during the design phase in order to make the best use of the resources. In general, both figures show that systems with peer-repair can achieve the same or higher reliability as compared to systems with self-repair.



(a)



(b)

Fig. 16. System lifetime of the self-repair and peer-repair system with, respectively, (a) higher μ and (b) lower μ as R grows.

C. Replacement Cost

In our previous work [7], we have also shown that merely replacing faulty devices is insufficient when the replacement cost is high, i.e., in a long-term operation, a high replacement cost (due to, e.g., human intervention) will dominate the total operation cost. Therefore, we adopt an aggressive replacement policy, which will replace both faulty devices and normal devices without available repair resources every time when the system reaches T_W. As a result, for the operation cost defined in (1), although N_D increases because more devices are replaced before failure, N_R is greatly reduced. Hence, when $C_R \gg C_D$, C_O can be significantly reduced, and it is our desire to know how large the C_R (as compared to C_D) should be when the system will benefit from peer-repair and the aggressive replacement policy. In our previous work [7], we perform a rough estimation of C_O of systems with different replacement policies for a 10-year operation period, i.e., T = 10 years, where the cost of systems with peer-repair outperform those with self-repair when C_R > 200.

Here, by detailed simulation, we further show in Fig. 17 the *breakeven* C_R of peer-repair system as compared to self-repair system with different R and T. The C_O of systems with peer-repair will be lower than those with self-repair only if C_R > breakeven C_R. For example, in the figure, a peer-repair system with minimum R (R = 1) has a breakeven C_R that is around 90 in long-term operation, i.e., $C_R \approx 90C_D$, which is fairly low. Figure 17 shows that the breakeven C_R grows with respect to T, since the peer-repair method can utilize repair resources more evenly in the system. More devices that run out of repair resources will be replaced during each replacement, so C_R becomes more important in long-term operation. Also, when R > 2, it is always cheaper to maintain a peer-repair system, i.e., even when C_R is negligible.

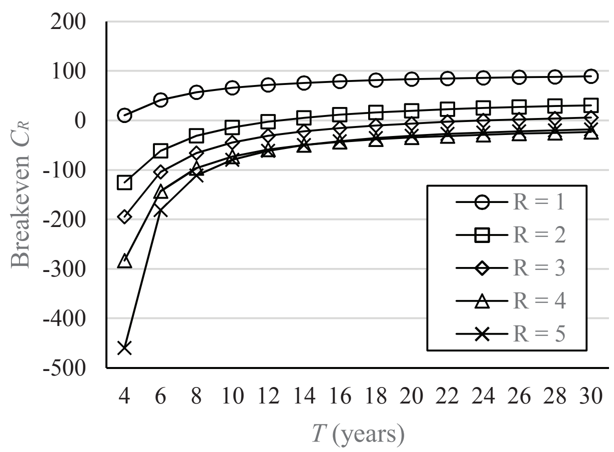


Fig. 17. Breakeven C_R of systems with peer-repair as compared with self-repair, with respect to T.

D. Operation Cost

In this subsection, we further analyze the operation cost. Figure 18 shows time points of replacement during a 20-year operation period, which can be divided into two phases, i.e., the *initial phase* and the *replacement phase*. The lengths of the two phases are denoted as T_I and T_R, respectively. In the beginning of the initial phase, the IOT devices and their repair resources are fault-free, so it normally takes longer time for the devices to become faulty, run out of repair resources, and reach T_W. When the system reaches T_W, according to the replacement policy, not only faulty devices but also normal devices without available repair resources are replaced, in order to restore the system functionality and reliability. After the first batch replacement, at time T_I, the replacement phase starts. In this phase, batch replacements occur in a period of T_R. Although all devices are fault-free after each replacement, some repair resources are already used. Therefore, the time for the system to reach T_W again (i.e., T_R) will be shorter than T_I. As shown in Fig. 18, assuming N_R batch replacements in the replacement phase, the total operation time is $T = T_I + N_R \cdot T_R$.

By moving the cost of devices used among replacement from the first term of (1) to its second term, the operation cost becomes

$$C_O = N_{DI} \cdot C_D + N_R \cdot \big((1 - T_W) \cdot N_{DI} \cdot C_D + C_R\big). \qquad (2)$$

Then, by substituting N_R with T_I and T_R, we can rewrite C_O as

$$C_O = N_{DI} \cdot C_D + \frac{T - T_I}{T_R} \cdot \big((1 - T_W) \cdot N_{DI} \cdot C_D + C_R\big), \qquad (3)$$

which is the combination of the initially deployed devices cost and the replacement cost (including the cost of new devices used to replace faulty ones). When C_R is much larger than C_D, which is highly possible, T_R will dominate C_O. Therefore, in long-term operation, maximizing the replacement period (T_R) will be the top concern.

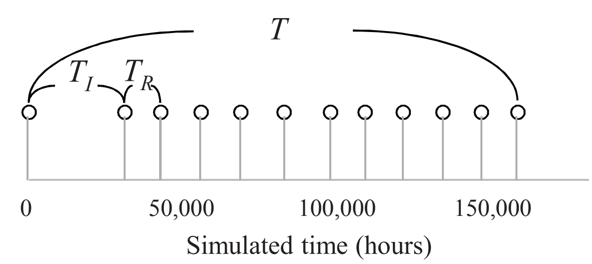


Fig. 18. Time points of replacement during a 20-year operation period.

Figure 19 shows the trend of T_R of the symbiotic systems with self-repair and peer-repair, respectively, as R grows. Note that the curves are normalized to the case where $T_R = 1$ when $R = 0$. Although peer-repair on average only shows 50% of extension on system lifetime compared to self-repair (see Fig. 15), its T_R is much longer than the one with self-repair only. For example, when $R = 1$, the T_R of peer-repair is twice as high as that of self-repair. The T_R extension of peer-repair as compared with self-repair is 3.5 times on average, as shown in the figure, which means the C_O can be further reduced to 28% that of self-repair. In addition, according to (3), compared to the system without repair ($R = 0$), the peer-repair system with $R = 1$ and $T_R = 10$ can reduce C_O by 90%.

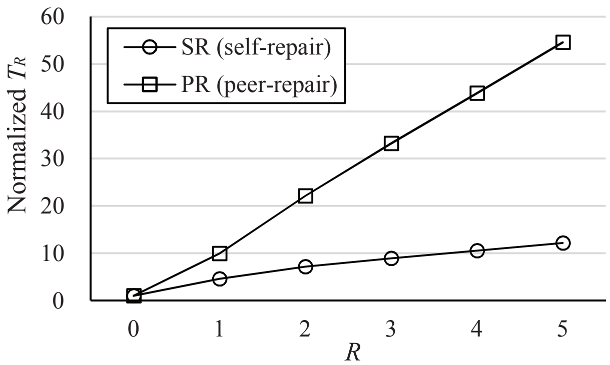


Fig. 19. Trend of T_R of a system with self-repair and peer-repair, respectively, with respect to R.

V. Quorum Sensing

To fulfill simple and effective monitoring of the entire system (or a neighborhood) by M_S, in [7], a *quorum-sensing* model is proposed for the symbiotic system. The model is defined as that an SS can change its behavior according to the *quorum* (minimum number or concentration) of neighboring devices or events. We define the quorum as χ, and the device over (under) quorum, i.e., a device (not yet) reaching a specified quorum χ, as Ω (Θ). The objective of this model is to obtain a self-stabilized symbiotic system, where each subsystems can test/repair each other without a centralized monitoring and configuration mechanism, so in normal cases the system cost (e.g., the human maintenance cost) can be reduced.

To observe the efficiency of quorum-sensing-based peer-repair, we define χ as the number of sensed fault-free subsystems. The Ω device (subsystem) is defined as one which, when faulty, can be repaired by other fault-free devices (subsystems) with a given peer-repair rate, where self-repair is automatically implied. The Θ device (subsystem) is defined as one which can only perform self-repair. In our experiment, we inject 256 symbiotic subsystems in a 20×20 array. Each subsystem can only sense other subsystems within distance 14, i.e., a device can sense at most 49% area of the 20x20 array.

In the first quorum-sensing experiment, we vary the raw yield of the 256 symbiotic subsystems with 100% peer-repair rate (μ_{peer}) and 0% self-repair rate (μ_{self}), and simulate the yield after injecting the subsystems into the array with 1,000 instances. Figure 20 shows the experimental result. As shown in the figure, as χ increases, the raw yield becomes more important because more fault-free subsystems are required to repair a faulty subsystem. However, when the raw yield is only 30% and $\chi \leq 30$, up to 97% faulty subsystems can still be repaired. This implies that the system yield can be greatly improved if quorum-sensing-based peer-repair is enabled. When the raw yield of devices cannot be guaranteed, e.g., in a future IOT system that consists of huge number of various devices, whose manufacturing test/repair becomes too costly to be performed, this kind of peer-repair is a promising approach to achieve high system yield.

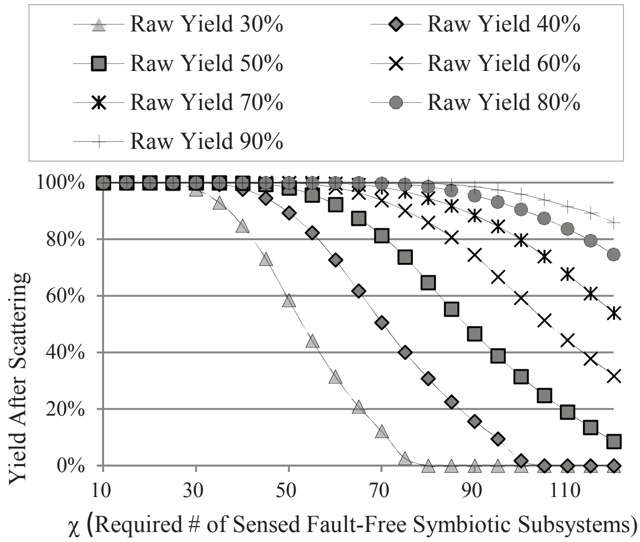


Fig. 20. Yield of symbiotic subsystems after being scattered into a 20x20 array under different raw yields and χ.

In real applications, perfect peer-repair is hard to achieve, so in the second quorum-sensing experiment, we vary the μ_{self} and μ_{peer} to observe the final repair rate of symbiotic subsystems.

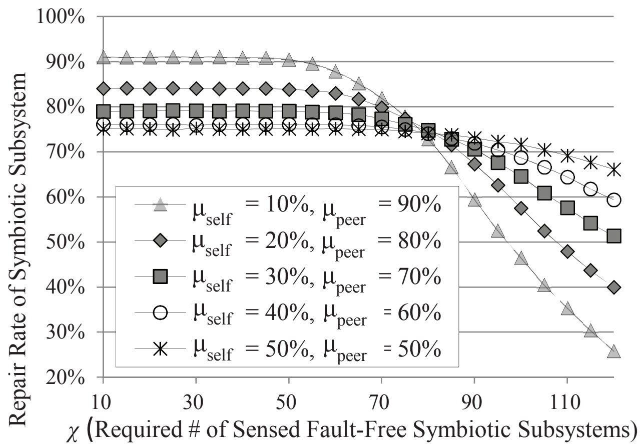


Fig. 21. Repair rate of symbiotic subsystems under different μ_{self}/μ_{peer} and χ.

For each pair of μ_{self} and μ_{peer}, 1,000 instances are simulated, whose results are shown in Fig. 21. In the figure, as χ increases, the μ_{self} becomes more important to obtain a higher repair rate.

However, when the μ_{self} is only 10%, by quorum-sensed-based peer repair, higher than 90% repair rate can be achieved when $\chi < 50$ and $\mu_{peer} = 90\%$.

VI. Conclusions

In this paper, we evaluate the cost and reliability (lifetime) issues of the repairable IOT storage and device networks, by the symbiotic system model. For the IOT storage network, we show that the relative lifetime of a storage system by peer-repair mechanism can be greatly increased. The peer-repair is especially suitable for future IOT systems where the storage nodes may spread in a wide area, with a large variation in device failure rates. For the IOT device network, on the other hand, we examine the reliability of the system with several factors: (1) system working threshold, (2) repair rate, (3) number of repair resources. Experimental results show that, for a system with redundant devices (such that $T_W = 80\%$), but without spare repair resources in the devices, software fault-tolerance technique can extend the system lifetime by 17 times.

In addition to the impact on lifetime, we evaluate the total costs, taking into account the device and replacement costs. For the IOT storage network, experimental results show that replacing the failed nodes as well as nodes whose R drops below 40% can achieve the lowest cost as C_R is reasonably higher than C_D, and a system with peer-repair can achieve the same lifetime as a much more test-enhanced system, with much lower cost. For the IOT device network, we show that for long-term operation, the replacement cost will dominate the operation cost, where the cost of a batch replacement can be several orders of the cost of a device in general. Therefore, the key to reducing the operation cost is to maximize the period of replacement (i.e., the period between human interventions in general) during operation period. Our experiment shows that by merely equipping each IOT device with a repair resource, peer-repair with the aggressive replacement policy can potentially reduce the operation cost by 90%. Moreover, it only requires $C_R \geq 90C_D$ to become profitable as compared to the self-repair-only devices.

To reduce the cost of a future IOT system that consists of a huge number of various devices (subsystems), a self-stabilized peer-repair mechanism is helpful. We show that by using quorum-sensing-based peer-repair, up to 97% faulty devices can be repaired even when the raw yield is only 30%. It implies that by adopting this mechanism, the manufacturing test can possibly be removed, as the impact of low raw yield can be greatly reduced. In summary, a symbiotic IOT system with peer-repair can be much more reliable and cost-effective.

References

[1] L. Rizzatti, "Digital Data Storage is Undergoing Mind-Boggling Growth," *EE Times* http://www.eetimes.com/author.asp?section_id=36&doc_id=1330462.

[2] IDC, "The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things," Apr. 2014.

[3] Micron Technology, Micron and Industry Analyst, 2016 http://files.shareholder.com/downloads/ABEA-45YXOQ/0x0x875021/4BEAA02E-BBC2-402C-A51D-B3B2C6B8C3D4/Winter_Analyst_Day_2016.pdf.

[4] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, W. Lintner, "United States Data Center Energy Usage Report," Jun. 2016 https://eta.lbl.gov/sites/all/files/publications/lbnl-1005775_v2.pdf.

[5] Ericsson, "CEO to shareholders: 50 billion connections 2020," https://www.ericsson.com/thecompany/press/releases/2010/04/1403231

[6] D. Evans, "The internet of things: how the next evolution of the internet is changing everything," https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[7] C.-W. Wu, B.-Y. Lin, H.-W. Hung, S.-M. Tseng and C. Chen, "Symbiotic System Models for Efficient IOT System Design and Test," in *Proc. IEEE International Test Conference in Asia* (ITC-Asia), Sept. 2017.

[8] A. Nordrum, "The Internet of Fewer Things," *IEEE Spectrum*, pp. 12–13, Oct. 2016.

[9] P. H. Bardell, W. H. McAnney, J. Savir, Built-In Test for VLSI, John Wiley & Sons, 1987.

[10] L.-T. Wang, C.-W. Wu, and X. Wen. VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann, 2006.

[11] P. K. Lala, Self-Checking and Fault-Tolerant Digital Design, CA, USA: Morgan Kaufmann, 2001.

[12] E. Fujiwara, Code Design for Dependable Systems: Theory and Practical Applications, John Wiley & Sons, 2006.

[13] B. Schroeder, E. Pinheiro, and W.-D. Weber, "DRAM Errors in the Wild: A Large-Scale Field Study," in *Proc. SIGMETRICS '09*, pp. 193-204, 2009.

[14] J. Meza, Q. Wu, S. Kumar and O. Mutlu, "A large-Scale Study of Flash Memory Failures in the Field," in *Proc. SIGMETRICS '15*, vol. 43, no. 1, pp. 177-190, Jun. 2015.

[15] J. Meza, Q. Wu, S. Kumar, O. Mutlu, "Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field," in *Proc. IEEE/IFIP Int. Conf. Dependable Systems and Networks* (DSN), 2015.