Symbiotic System Models for Efficient IOT System Design and Test

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

***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, we propose a high-level model for efficient design-space exploration, which is called the *symbiotic system* (SS) model. Based on that, we propose a quorum-sensing model for SS to improve the reliability of IOT systems that contain subsystems. Experimental result shows that, with the SS approach, the lifetime of an example IOT storage system with 50K failure-in-time/Mb nodes can be extended by 74.41%. In addition, to double the lifetime of the system, only 5% additional repair resource is required. In short, by the SS model, it is possible for an IOT system to achieve low cost, low energy consumption, and high reliability.**

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

The global semiconductor business over the past 30 years shows an encouraging trend of growth, with only a few glitches that did not hinder the long-term trend, as shown in Fig. 1 [1]. In the figure, we can see that there are three periods of growth: 1) growth period of PC market, 2) growth period of 3C market, and 3) growth period of smart-handheld market. Each growth period spanned for about a decade. At the end of each growth phase, we were lucky to see a new market driver took over and extended the prosperity of the semiconductor industry.

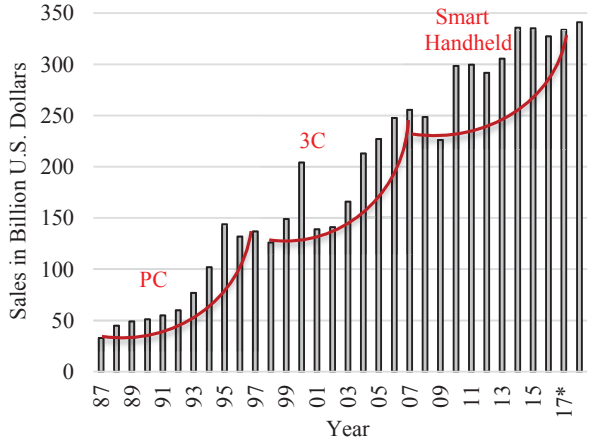

Fig. 1. Industry drivers for the continuous growth of global semiconductor sales in the past 30 years [1].

However, the semiconductor market in recent years is not as good as it used to be—almost flat as shown in the figure. The (predicted) market trend of smart phones and tablets from 2015 to 2020 as shown in Fig. 2 [2, 3] is almost flat, even negative for tablet shipments. The Internet of Things (IOT) had been identified, or expected, to be the main driver of market growth for many industries, including the semiconductor industry.

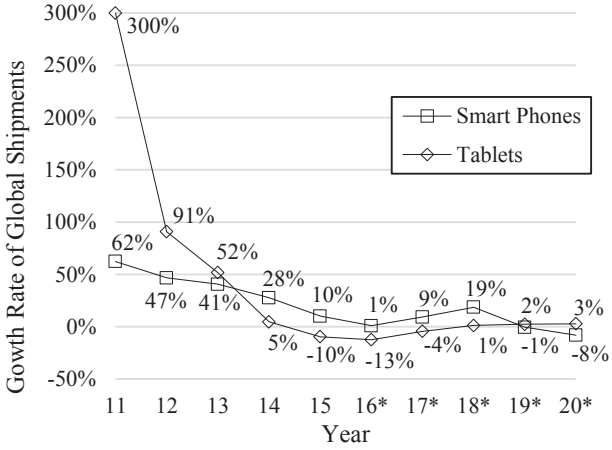

Fig. 2. Market Trend of Smart Phones and Tablets [2, 3].

In the past few years, the prediction of 50 billion IOT devices by 2020 [4, 5] has been well cited. However, so far there is not much evidence that the IOT will give a great boost to the semiconductor industry soon. Some newer forecasts have greatly lowered their expectations of the growth of IOT devices [6, 7]. As the semiconductor industry is quite mature, the growth of the semiconductor market generally follows that of the global GDP in recent years, and will be so in the near future, which shows no dramatic growth. If the global economy does not grow dramatically, so will the IOT market be. Another limiting factor is energy consumption. It has been a great concern that the entire world is over-using energy, creating environmental problems such as pollution, global warming, human-induced natural disasters, etc. Therefore, the dramatic growth of IOT devices will have to be under the constraint of no or only slight increase in energy supply, unless we find a sustainable clean and cheap energy, which simply is not in our sights. We need to invest more in basic research to be able to realize all the fancy IOT applications, e.g., real-time environmental monitoring/prediction (for disaster prevention) [8], scalable infrastructure management [9], manufacturing efficiency enhancement [10], productivity improvement for industrial workforce [11], smart coordinated healthcare [12, 13], connected vehicles [14], smart city/home [15], etc. If it is possible that there will be connected devices multiple orders of magnitude more than today, it is necessary to design, manufacture, and test these devices without dramatic spending or energy supply increase. The grand challenge for IOT, therefore, is 1) fast and low-cost system design

and manufacturing, 2) energy consumption containment, and 3) environmental sustainability.

Recent progress in artificial intelligence and neuromorphic computing shows that there is hope for developing much more energy efficient semiconductors for IOT devices and systems, including those used in datacenters [16]. With regard to IOT testing under the grand challenge, we seek to reduce or even remove manufacturing test of IOT devices and systems, while being able to enhance their reliability (useful lifetime). In this paper, we propose the basic theory and computing models for *Symbiotic System* (SS), which can be used for developing IOT devices and systems. By using the proposed theory and models, our experimental results show that the lifetime of a low-cost storage system can be extended by 74.41% as compared with the conventional way. In another case to double the lifetime of an IOT storage system, only 5% additional repair resources are required if we adopt the SS approach.

II. Symbiotic System Model

A *Symbiotic Relationship* (SR) is a relationship of mutual dependence between two (electronic) systems, where part of one system's input is from the other's output, and vice versa. A *Symbiotic System* (SS) is a twin system comprising the *primary (functional) system* and *secondary (test) system*, with SR. To model the symbiotic behavior of a novel class of electronic devices and systems, mainly for the Internet of Things (IOT), we propose a model for the symbiotic machine or system.

A. Symbiotic Machine

We use the finite-state machine (FSM) model denoted as $M = (Q, \Sigma, O, \delta)$, where Q is a set of states, Σ a set of inputs, O a set of outputs, and δ a set of transition functions defined by $Q \times \Sigma \rightarrow Q \times O$. Note that in an SS, the SR implies that the primary and secondary systems can communicate with each other directly, i.e., part of the outputs of the primary (secondary) system are inputs of the secondary (primary) system. We define a specific class of FSM to model the SS, which is called the *Symbiotic Machine* (SM) or *Symbiotic Device*, as $M_D = (Q_D, \Sigma_D, O_D, \delta_D)$, where $Q_D = Q_P \times Q_S$, $\Sigma_D = \Sigma_{P1} \times \Sigma_{S1}$, $O_D = O_{P1} \times O_{S1}$, and $\delta_D = \delta_P \times \delta_S$. The SM consists of the *primary machine* (M_P), which is defined as the functional system in the SS, and the *secondary machine* (M_S), which is the test system in the SS. As shown in Fig. 3, the primary machine is denoted as $M_P = (Q_P, \Sigma_P, O_P, \delta_P)$, where $\Sigma_P = \Sigma_{P1} \times \Sigma_{P2}$ (or $\Sigma_{P1} \cup \Sigma_{P2}$), $O_P = O_{P1} \times O_{P2}$ (or $O_{P1} \cup O_{P2}$), and δ_P: $Q_P \times \Sigma_P \rightarrow Q_P \times O_P$. Similarly, the secondary machine is denoted as $M_S = (Q_S, \Sigma_S, O_S, \delta_S)$, where $\Sigma_S = \Sigma_{S1} \times \Sigma_{S2}$ (or $\Sigma_{S1} \cup \Sigma_{S2}$), $O_S = O_{S1} \times O_{S2}$ (or $O_{S1} \cup O_{S2}$), and δ_S: $Q_S \times \Sigma_S \rightarrow Q_S \times O_S$. Figure 3 shows the graphical model of M_D in terms of M_P and M_S.

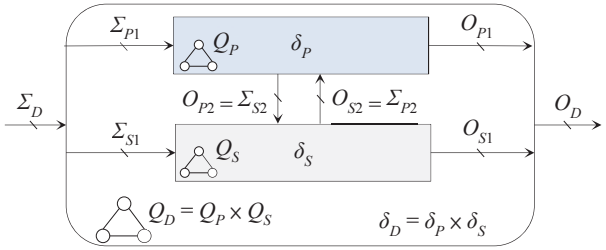

Fig. 3. The FSM model of a symbiotic device.

According to the definition, there is an SR between M_P and M_S. Specifically, there exist O_{S2} (the output from M_S) and O_{P2} (the output from M_P), meaning that the state of M_P can be monitored and changed by M_S, and vice versa. In an SS with *symbiosis-based test* (SBT), monitoring of Q_P can be done by a sensing mechanism. Based on current technology, e.g., it can be an on-line test and diagnosis scheme. In addition, change of Q_P is a healing or repair mechanism, e.g., it can be an on-line repair or fault tolerance scheme. As shown in Fig. 3, Σ_{P2} and Σ_{S2} (i.e., O_{S2} and the O_{P2}) are the internal inputs and outputs between M_P and M_S. We show in Figs. 4(a) and (b) the state diagrams for M_P and M_S, respectively, of an example SM. In the M_P, Q_P = {Active, Idle, Test/Config}, and in the M_S, Q_S = {Monitor/Report, Test, Config}. The signals TC (Test/Configuration), ACT (Active), IDLE, MR (Monitor/Report), CF (Configuration), and F (Failure) are all binary flags, which will be 1 if certain conditions are met. For example, in Fig. 4(a), TC = 1 if M_P needs test or configuration, ACT = 1 if M_P can be operated in normal mode, and IDLE = 1 if M_P has no waiting task to be executed or is forced to rest due to some detected faults or errors. In Fig. 4(b), MR = 1 if M_S should monitor the behavior of M_P and report it to the controller (local gateway or cloud) and/or other devices. Also, F = 1 if M_S detects that M_P is faulty, and CF = 1 if M_S needs to configure M_P.

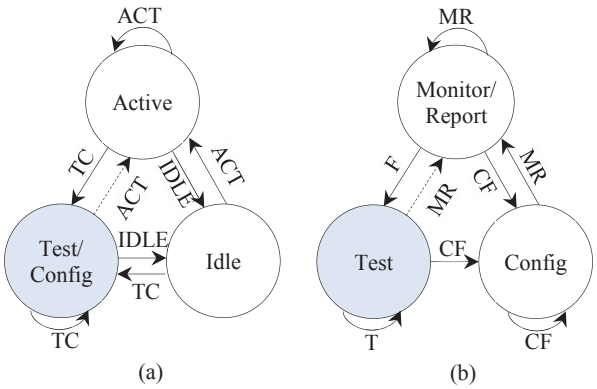

Fig. 4. State diagrams for (a) M_P and (b) M_S of a symbiotic device.

We can let, e.g., Σ_{P1} = {ACT, IDLE} and Σ_{S1} = {MR}, i.e., the external controller can change the state of M_P to Active or Idle, and ask M_S to monitor/report the state/behavior of M_P. We can also let Σ_{P2} (O_{S2}) = {TC} and Σ_{S2} (O_{P2}) = {F}, i.e., M_S can send M_P from another state to the Test/Config state. When M_P is sensed abnormal or for any reason it needs test (and diagnosis), it will set F to 1 and send M_S into the test mode. On the other hand, if M_P is confirmed faulty or for any reason it needs configuration, then CF = 1, and M_S will transit to the Config state to configure/repair the faulty M_P and try to bring it back. When M_S is in the Test or Config state, it will set TC to 1, which will pause the current job of M_P, if there is any, and send M_P to the Test/Config state. After configuration, if the job can be resumed, ACT = 1 and M_P will transit to the Active state. Otherwise, IDLE = 1 and M_P will transit to the Idle state, waiting for the job to be restarted from the beginning, when M_P can transit to the Active state.

Note that for different application requirements or specifications, different M_P and M_S as opposed to those shown in Fig. 4 can be defined. It is possible, e.g., to merge the Test and Config states of M_S into a single state. For more sophisticated repair

behavior, on the other hand, there may be more states in M_S. Examples will be shown later.

B. Reliability of Symbiotic System

In the proposed SS model, M_P can be tested (repaired) by M_S. Let the reliability function of the original system be $R(t) = e^{-\lambda t}$, where λ is the constant failure rate during useful life (i.e., when mean time between failures (MTBF) is $1/\lambda$) [17]. Then, the reliability of M_D is $R_D(t) = e^{-\lambda(1-\alpha)t}$, where α is the repair rate of M_D. Figure 5 plots the reliability of the original system as well as those of the SSs with 20%, 30%, and 40% repair rates, respectively. In Fig. 5, we assume that $\lambda = 4 \cdot 10^{-6}$. We can see

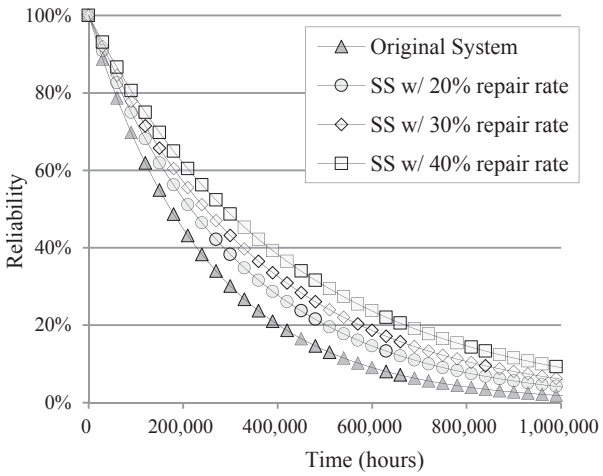

Fig. 5. Reliability of the original and symbiotic systems.

that the reliability of M_D after operating for 3 years (26,280 hours) is still higher than 90%, and the M_D reliability with respect to different repair rates are consistently higher than that of the original system. After 30,000 hours of operation, e.g., the reliability of the original system is 88.7%, which is 4.3% lower than that of the M_D with 40% repair rate.

III. Symbiotic System with Subsystems

A. Symbiotic Subsystem

A symbiotic system can be a class (network) of smaller symbiotic systems, i.e., *symbiotic subsystems*. An important objective for the model is to explore efficient architectures to guarantee a fault-free SS within a specified lifetime, for IOT applications with hierarchical symbiotic subsystems. Another important objective is to explore efficient architectures to minimize energy consumption throughout the specified service time of the SS. For such purposes, the SS formed by symbiotic subsystems can have hierarchical and/or peer repair capability at the system (network) level, in addition to self-repair or self-healing capability at the subsystem level.

An SS with symbiotic subsystems is *fault-free* if and only if the number of fault-free symbiotic subsystems is larger than or equal to the *working threshold* (T_w), i.e., T_w is the minimum number of fault-free subsystems in a fault-free system. If the SS is not fault-free, it *fails* (is dead) and cannot be recovered. For example, if a system contains 1,000 subsystems, and T_w = 800, then the SS is faulty if less than 800 subsystems are fault-free. We define the reliability of an SS at time t, with a working threshold of T_w subsystems, as the probability of the system that has at least T_w fault-free subsystems up to time t, i.e.,

$$R(T_w, t) = \frac{\sum_{j=T_w}^{N} \binom{N}{j} R(t)^j (1-R(t))^{N-j}}{\sum_{i=0}^{N} \binom{N}{i} R(t)^i (1-R(t))^{N-i}}$$

where N is the total number of symbiotic subsystems in the SS, and $R(t)$ is the reliability of a symbiotic subsystem (assuming they are all the same).

Figure 6 shows the reliability of the original system (with original subsystems) and systems of symbiotic subsystems with 20%, 30%, and 40% repair rates, respectively. Again, $\lambda = 4 \cdot 10^{-6}$. We can see that the reliability of systems with symbiotic subsystems is much higher than that of the original system. The SS with 20% repair rate, e.g., can achieve 95% reliability after 28,000 hours of operation. However, the reliability of the original system drops to 29% after the same operation time.

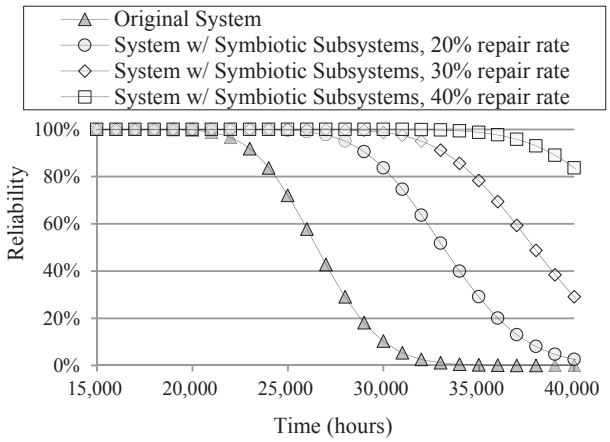

Fig. 6. Reliability of original and symbiotic systems of subsystems.

B. Quorum Sensing

To fulfill simple and effective monitoring of the entire system (or a neighborhood) by M_S, we adopt the *quorum-sensing* model from biology. It is defined here 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. The behavior of a *device under quorum* (DUQ, i.e., a device not yet reaching the specified quorum χ) is denoted as Θ, and that of a *device over quorum* (DOQ) is denoted as Ω. The objective is that, based on this model, symbiotic subsystems can test/repair each other without a centralized monitoring and configuration mechanism, which normally is more costly.

As an example, we extend the M_S in Fig. 4 to obtain a secondary machine as shown in Fig. 7, i.e., an M_S with quorum sensing, where Q_S = {Monitor/Report, Test, Self-Repair, Peer-Repair Replay, Peer-Repair Request}. The signals MR (Monitor/Report), CF (Configuration), F (Failure), T (Test), PR (Pear-Repair), and PCF (Configure requested from peer) are all binary flags, which will be 1 if certain conditions are met. The conditions for MR, CF, and F are the same as those in Fig. 4. Also, T = 1 if M_P needs test, PR = 1 if self-repair is not available, and PCF = 1 if M_S is requested to repair a faulty peer symbiotic subsystem. For example, Ω can be defined as the peer-repair behavior, and PR and PCF can be set to 1 only by Ω-driven flags, i.e., a fault-free DOQ can repair a faulty device as defined by Ω. Note that a different M_S can be defined, e.g., add a direct path from the Test state to the Peer-Repair Request state, i.e., M_S can transit from Test to Peer-Repair Request in certain conditions directly, without passing through Self-Repair.

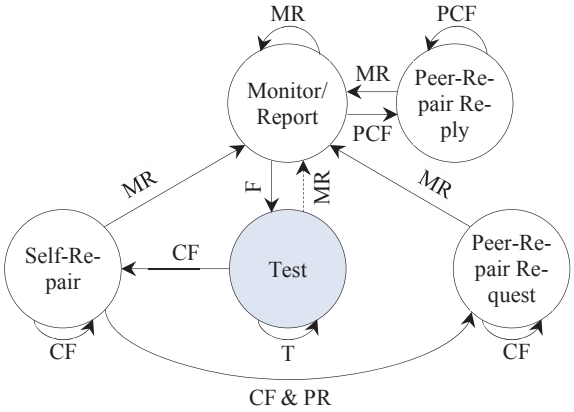

Fig. 7. State diagram for M_S of a symbiotic subsystem with quorum-sensing.

To specify the neighborhood where a symbiotic subsystem can sense other subsystems, we define the *sensing distance* (δ) as the maximum distance between any two symbiotic subsystems who can sense each other. To observe the efficiency of the quorum-sensing-based peer-repair, in this experiment, a faulty symbiotic subsystem can be repaired if and only if it is a DOQ. In our experiment, we inject 128 faulty and 128 fault-free symbiotic subsystems randomly in a 20×20 array. We vary δ and χ and simulate the repair rate of the symbiotic subsystems with 1,000 instances. We show in Fig. 8 the average repair rate under different conditions, where up to 83.5% faulty subsystems can be repaired when χ = 10 and δ = 5. If the subsystem can sense the whole space (δ = 20), almost 100% repair rate can be achieved even if χ is up to 90. The experimental result shows that, although the original manufacturing yield of the subsystems is only 50%, most of the faulty subsystems can be repaired by the quorum-sensing-based peer-repair mechanism.

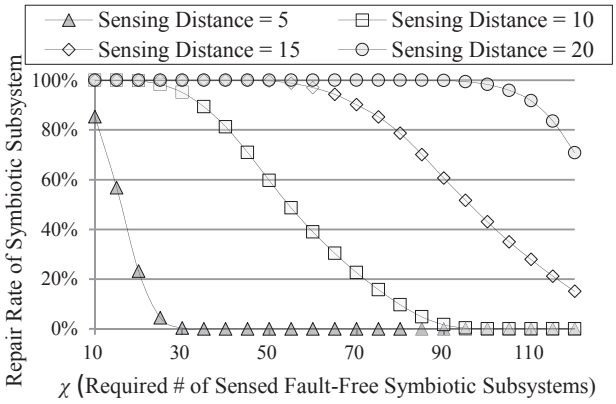

Fig. 8. Average repair rate of symbiotic subsystems with quorum sensing.

IV. SS Model for IOT Systems

A. IOT Device Network

We apply the SS model to a simple IOT system to evaluate its reliability and cost, which is a symbiotic system of 1,000 identical nodes. Each node is a symbiotic subsystem, which can communicate with its neighboring nodes (i.e., *peers*). We assume the enhanced M_S as depicted in Fig. 9. When a fault is detected, M_S will try to bring back M_P by allocating its self-repair resources, R_S, or asking its peers to allocate their peer-repair resources, R_P. In normal situation, M_S stays in the Monitor/Report state. When a fault is detected, F = 1, the state will change to Test, and TC = 1 to force M_P to Test/Config. In the Test state, if the fault is repairable and R_S is available, it will transit to Self-Repair, allocate repair resource (i.e., $R_S = R_S - 1$), and repair the faulty M_P. If $R_S = 0$, M_S will seek help from its peers. The M_S will go to Peer-Repair Request, and one of its peer's PCF will be set to 1. The latter's M_S will then enter the Peer-Repair Reply state and try to allocate R_P (i.e., $R_P - 1$) to repair the faulty node's M_S. If the peer has no more R_P, the faulty node's M_S will turn to next peer for help, and so on. Once all of its peers has run out of R_P, peer-repair fails.

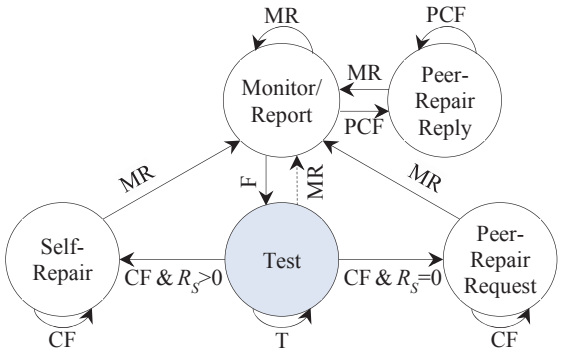

Fig. 9. State diagram for M_S of a symbiotic IOT device.

We use an in-house simulator. Let the reliability of the original M_D be $R(t) = e^{-\lambda t}$, where λ is a constant failure rate, and node failure is caused by a fault. Once a fault occurs, M_S will try to repair the faulty M_P by self-repair and/or peer-repair methods as mentioned above. If it is unrepairable, the node fails. The simulation process will stop when the number of fault-free nodes drops below T_w.

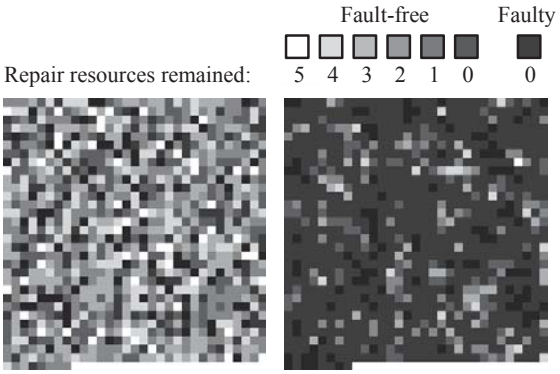

Fig. 10. Status of (a) self-repair only system (b) self-repair + peer-repair system, the moment when fault-free nodes < T_w.

1) Self-Repair vs. Peer-Repair

Figure 10 shows the repair resource status of the IOT devices when the system fails (i.e., fault-free nodes < T_w) under (a) system with self-repair and (b) system with peer-repair. Both systems have 1,000 nodes, each having 5 repair resources, and λ = 0.00001. Each cell in the array represents a node, whose color shows the number of its available repair resource, ranging from 0 (no resource left) to 5 (all resource available). On average, when the system fails, the peer-repair enhanced system has

30% better utilization of the repair resource and 50% longer system lifetime.

2) Number of Peers

The *number of peers* of a node is denoted as ρ, which is usually limited by the communication complexity. Selecting an appropriate ρ is to weigh between benefit (system reliability or lifetime) and cost (hardware, energy, timing penalty, etc.). Figure 11 shows the system lifetime under different R_P per node. When ρ increases, the system lifetime grows rapidly. As ρ reaches 5, the system lifetime saturates.

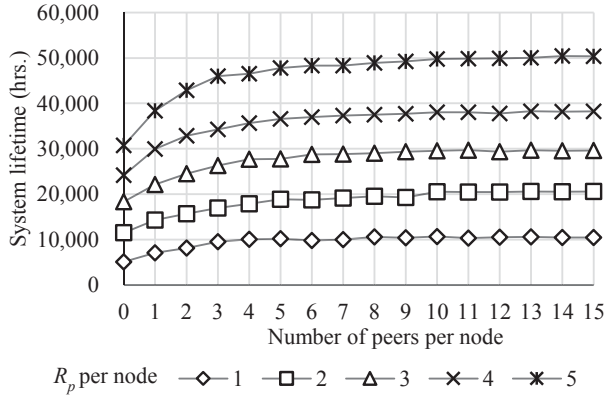

Fig. 11. System lifetime as ρ varies.

3) Operation Cost

Figure 12 shows the 10-year operation cost of the IOT system. We replace faulty nodes which cannot be repaired or nodes that are going to fail (i.e., $R_S = 0$, or $R_S = 0$ and peers' $R_P = 0$) every time when the system reaches T_w. The operation cost (C_O) is the device cost (C_D) plus the replacement cost (C_R):

$$C_O = N \cdot C_D + n \cdot C_R$$

where N is the total number of devices used, including initially deployed and replaced devices, and n is the times of batch replacement. Note that the cost is normalized to C_D (i.e., $C_D = 1$).

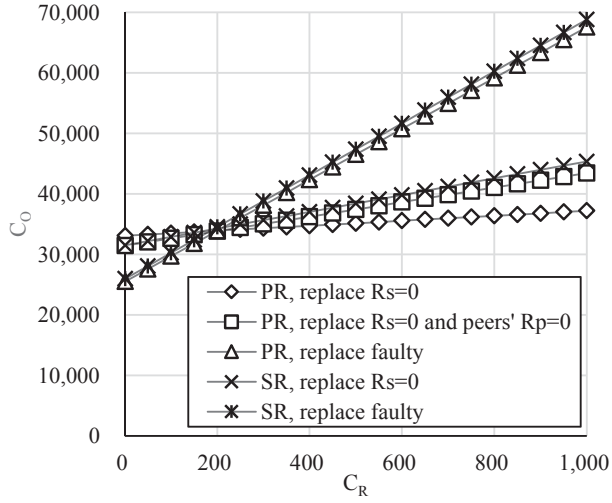

Fig. 12. 10-year operation cost.

The replacement is done manually, which can be very expensive for a large system. In Fig. 12, we show that when C_R exceeds 200 times of C_D, merely replacing faulty nodes is not enough. With peer-repair and replacing those with $R_S = 0$, the system is able to achieve the lowest cost when C_R exceeds 200 times of C_D.

B. IOT Storage Network

We now apply the SS model to a simple IOT storage system to evaluate its lifetime. Each of the 1,000 nodes is an SS with peer-repair when running out of R_S. Note that we focus on storage nodes only. The structure and behavior of our 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 (SU). Self-healing mechanism in a storage node includes the allocation of R_S as well as the correction of faults during the read/write process. Repair resource allocation is defined on the MU basis. The correction capability of a storage node can be determined by the maximum number of correctable SUs.

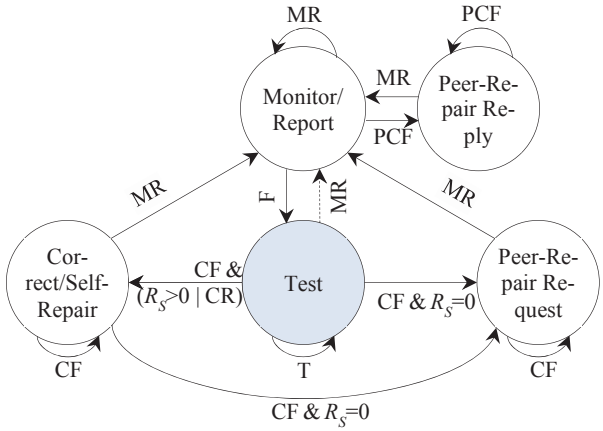

Fig. 13. State diagram for M_S of a symbiotic storage node.

The M_S is depicted in Fig. 13, which is similar to Fig. 9, except for Q_S = {Correct/Self-Repair} and signal CR (Correctable). In the model, faults can occur in a number of MUs, and the self-healing mechanism depends on the fault conditions in each MU. If a faulty MU is correctable, CR = 1, or if $R_S > 0$, M_S will transit to Correct/Self-Repair to perform correction or allocation of R_S. If R_S is sufficient for repairing the faults and MR = 1, M_S will transit to Monitor/Report. Otherwise, M_S will transit to Peer-Repair-Request to ask for peer repair.

We define the *Failure-in-Time* (FIT) as the number of faulty bits per billion device-hours of operation. The lower the FIT/Mb, the higher the reliability of the storage devices. In our experiments, we compare the lifetime of the low-FIT (expensive) systems without peer-repair with that of the high-FIT (cheap) systems with peer-repair capability. Consider a 1,000-node symbiotic system, where $T_w = 100$. Each storage node has 4GB capacity and ρ = 3. Also, each storage node has 0.01% repair resources, i.e., 0.01% additional MUs for self and/or peer repair. We vary the FIT/Mb of the nodes from 10K to 50K, and simulate the lifetime of the storage systems. In addition, we simulate lifetime enhancement by the peer-repair mechanism for systems with 50K FIT/Mb storage nodes, with additional repair resources (in addition to the original 0.01% repair resources) from 0.1% to 10%. As shown in Table 1, even when node FIT/Mb = 50K, by just 0.1% additional repair resource and peer-repair, the storage system lifetime will be longer than that with 16K FIT/Mb nodes.

Table 2 shows the extended lifetime of the storage systems with peer-repair when we vary the percentage of additional repair resource. From the table, the lifetime of the storage system

with 50K FIT/Mb nodes will be 74.41% longer by peer-repair. The extended lifetime is higher than that of the storage system with 16K FIT/Mb nodes. In addition, to extend the lifetime of the storage system with 50K FIT/Mb nodes by 98.46%, only 5% additional repair resource is required.

Table 1. Simulated Lifetime of Storage System with Different FIT/Mb Nodes, with/without Peer-Repair

Storage systems with peer-repair FIT/Mb: 50,000	
Additional Resource	Simulated Lifetime (yr)
0%	22.52
0.1%	24.48
1%	24.85
5%	25.62
10%	26.68

Storage systems without peer-repair	
FIT/Mb	Simulated Lifetime (yr)
50,000	12.91
45,000	13.17
40,000	13.59
35,000	14.73
30,000	15.23
25,000	16.37
20,000	18.79
18,000	20.32
16,000	22.46
14,000	25.55
12,000	29.99
10,000	36.21

Table 2. Extended Lifetime (%) by Additional Repair Resource

	Percentage of additional repair resource (%)				
FIT/Mb	0%	0.1%	1%	5%	10%
50,000	74.41	89.64	92.52	98.46	106.66
45,000	70.94	85.86	88.69	94.51	102.55
40,000	65.74	80.21	82.95	88.59	96.39
35,000	52.81	66.15	68.68	73.88	81.07
30,000	47.85	60.76	63.21	68.25	75.20
25,000	37.54	49.55	51.82	56.51	62.97
20,000	19.84	30.31	32.29	36.37	42.01
18,000	10.81	20.48	22.31	26.09	31.30
16,000	0.24	8.99	10.65	14.06	18.77
14,000	-11.87	-4.17	-2.71	0.29	4.43
12,000	-24.93	-18.37	-17.13	-14.57	-11.05
10,000	-37.81	-32.38	-31.35	-29.23	-26.31

V. Conclusions and Discussion

We propose the basic theory and models for symbiotic system (SS), which can help in developing cost-effective, reliable, and energy-efficient IOT devices and systems. Experimental results based on FSM models show that the reliability of SS is much higher than that of the original system. When failure rate is $4 \cdot 10^{-6}$, e.g., the SS with 20% repair rate can achieve 95% reliability after 28,000 hours of operation. However, the reliability of the original system drops to 29% after the same amount of operation time. In addition, by the proposed quorum-sensing-based peer-repair mechanism, the system can survive even when the initial subsystem yield is as low as 50%.

We apply the proposed theory and models to a couple of cases. The experimental results show that the lifetime of an example IOT storage system with 50K FIT/Mb nodes can be extended by 74.41%. In addition, only 5% additional repair resources are required to double the lifetime of the system. In short, by the SS model, it is possible for an IOT system to achieve low cost, low energy consumption, and high reliability.

References

[1] Statista, "Semiconductor sales revenue worldwide from 1987 to 2018," https://www.statista.com/statistics/266973/global-semiconductor-sales-since-1988/.

[2] Statista, "Global smartphone shipments forecast from 2010 to 2020," https://www.statista.com/statistics/263441/global-smartphone-shipments-forecast/.

[3] Statista, "Forecast unit shipments of tablets worldwide from 2010 to 2020," https://www.statista.com/statistics/269912/worldwide-tablet-shipments-forecast/.

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

[5] Cisco, "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.

[6] A. Nordrum, "The internet of fewer things," *IEEE Spectrum*, pp. 12–13, Oct. 2016.

[7] Cisco, "Cisco visual networking index predicts near-tripling of IP traffic by 2020," https://newsroom.cisco.com/press-release-content?articleId=1771211.

[8] S. Fang *et al*., "An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1596–1605, May 2014.

[9] P. Gope and T. Hwang, "Untraceable Sensor Movement in Distributed IoT Infrastructure," *IEEE Sensors Journal*, vol. 15, no. 9, pp. 5340–5348, Sept. 2015.

[10] F. Tao *et al*., "IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1547–1557, May 2014.

[11] L. D. Xu *et al*., "Internet of Things in Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[12] S. Amendola *et al*., "RFID Technology for IoT-Based Personal Healthcare in Smart Spaces," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 144–152, Apr. 2014.

[13] Y. Zhang *et al*., "Ubiquitous WSN for Healthcare: Recent Advances and Future Prospects," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 311–318, Aug. 2014.

[14] N. Lu *et al*., "Connected Vehicles: Solutions and Challenges," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[15] A. Zanella *et al*., "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[16] R. Evans and J. Gao, "DeepMind AI Reduces Google Data Centre Cooling Bill by 40%," https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/.

[17] L.-T. Wang *et al*., *Design for Testability*: *VLSI Test Principles and Architectures*. San Francisco, CA: Elsevier (Morgan Kaufmann), 2006.