Nano-scale On-chip Irregular Network Analysis

Yang Liu, Alvin R. Lebeck
Department of Computer Science
Duke University
Durham, NC, USA
{yliu,alvy}@cs.duke.edu

***Abstract*— Shrinking CMOS feature sizes and the integration of novel nanotechnologies onto silicon platforms are both likely to increase fabrication defects. As a result, on-chip networks become more and more irregular due to defects and it becomes more challenging to map computation and data onto the networks. One way to overcome this challenge is to configure the irregular network into a more conventional regular topology.**

In this paper we analyze nano-scale on-chip irregular networks to determine the regular topology most similar to a given irregular network. The results show that an irregular network is most similar to a tree. Further analysis is conducted based on configuring an irregular network into a tree structure to show whether there are opportunities to utilize links that are not included in the tree.

Nano-scale; irregular networks; analysis

I. INTRODUCTION

The trend toward increased functional capabilities on a single chip requires commensurate development of on-chip communication networks. The pressure to provide ever more cores on a single chip requires aggressive scaling of CMOS technology (including 3D [1]) or the incorporation of novel nanotechnologies (e.g., carbon nanotubes [2]). Unfortunately, decreasing feature sizes may decrease control over the entire fabrication process (e.g., with self-assembly [3]) and thus increase fabrication defects. As a result, it may be increasingly difficult to reliably create interconnects and networks may become highly irregular.

Irregular networks present challenges for programming, system designers (e.g., coherence protocols) and developing efficient routing algorithms, particularly for direct networks. It is difficult to map computation and data placement onto irregular topologies. One way to overcome the challenge is to configure the irregular network into a more conventional regular topology (e.g., a mesh, a tree or a ring).

We can configure an irregular network into different regular topologies. However, if a given irregular network is similar to a mesh and we configure it into a tree, a lot of links will be left unused. In order to decide which regular topology to choose, it is helpful to analyze the structural properties of an irregular network. For example, if the number of the links is roughly the same as the number of the nano-scale devices (nodes) in an irregular physical network, the network might be more like a tree than a mesh; while the network might be more like a mesh if the degree of each node is around four.

In this paper we analyze nano-scale on-chip irregular networks to determine the regular topology most similar to a given irregular network. The results show that an irregular network is most similar to a tree. Further analysis is conducted based on configuring an irregular network into a tree structure to show whether there are opportunities to utilize links that are not included in the tree.

II. METHODOLOGY

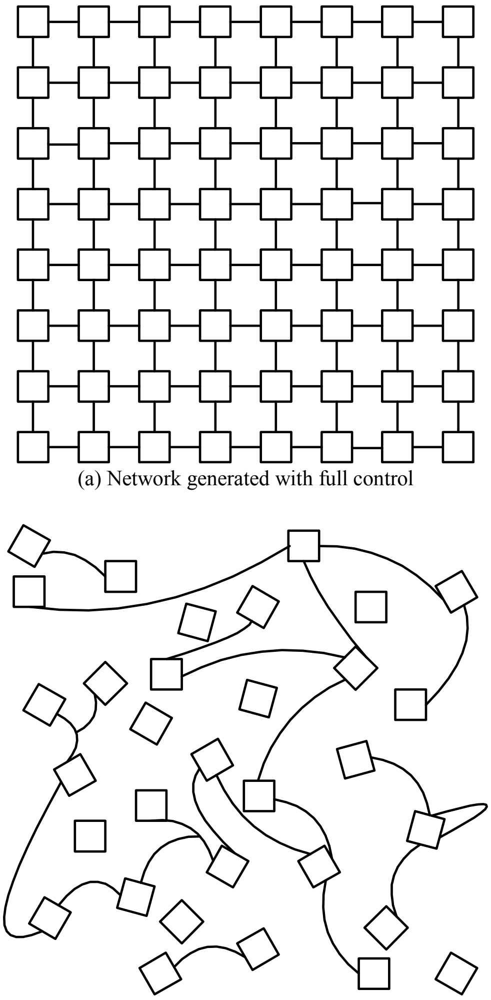
In order to analyze nano-scale on-chip irregular networks, we need to first generate them. Within the context of a bottom-up integrated nano-scale network there are three principle controls over the integration of components: placement, orientation, and interconnection [4]. Placement control decides whether a node is placed at a pre-fixed point or a random point on a planar surface during the integration. Orientation control decides whether a node should always face the same direction or can be turned by a certain angle. Interconnection control decides whether the placed nodes are connected through straight links or curved links. Random seeds are used to control the locations that nodes are placed, the angles that nodes are turned, and the curving extent of the links.

Future self-assembly technology may provide all three types of control and if there are no defective nodes or links, the integrated network can be a regular network. However, whether and when full control can be achieved remains an open question. To fully analyze the characteristics of nano-scale irregular networks we simulate all eight combinations of controls and also networks with full control but different fractions of defective nodes or transceivers.

We use square nodes with one transceiver at each of the four sides, which means that each node can be connected to at most four other nodes. A square grid is used as the substrate and nodes are placed on the grid and links are grown from transceivers to connect nodes with nearby neighbors following different control options to generate networks.

Fig. 1 shows two examples of the generated networks, in which a square represents a node. Fig. 1(a) is a part of a network generated with full control. We can see that nodes are grouped together, placed at each cross point of the grid and connected to its four nearest neighbors through straight links. The finished network is exactly a mesh structure. Fig. 1(b) is a part of a network generated without any control. We can see that nodes are scattered on the grid and links have turns and fuse with each other.

978-1-4244-4581-3/09/$25.00 ©2009 IEEE



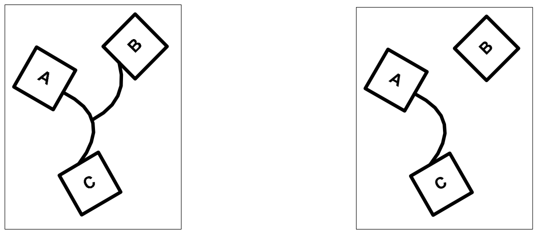
(a) Network generated with full control

(b) Network generated with no control

Figure 1. Examples of Generated Networks

As shown in Fig. 1(b), there are fused links in networks generated without any control. Before configuring the irregular network into a regular structure, we must cut fused links to ensure that only one pair of nodes are connected through a fused link. Fig. 2(a) shows a fused link connecting three nodes: A, B, and C. Fig. 2(b) shows the fused link connecting only A and C after the part connecting to C is cut off.

When more than two nodes are connected together through one fused link, there are multiple choices about which part of the link to cut. For example, Fig. 2(b) is only one option to do the cutting. We can also cut the fused link to make A and B connected, or B and C connected. In the cutting mechanism we use, the probability of a node remaining connected to a fused link after cutting is based on the number of the alive transceivers of the node and the number of nodes on that fused link [5].This cutting mechanism provides higher connectivity than random cutting. An *alive transceiver* is an active transceiver without defects.



(a) Before cutting

(b) After cutting (one option)

Figure 2. A Fused Link

III. NETWORK ANALYSIS

To fully analyze the characteristics of nano-scale irregular networks we simulate all eight combinations of control and also networks with full control but different fractions of defective nodes or transceivers. We use a 3-bit binary number to represent different control combinations where the leftmost bit indicates placement control, the middle bit indicates orientation control, and the rightmost bit indicates interconnect control. A bit with value 1 means that we have the corresponding control over the integration.

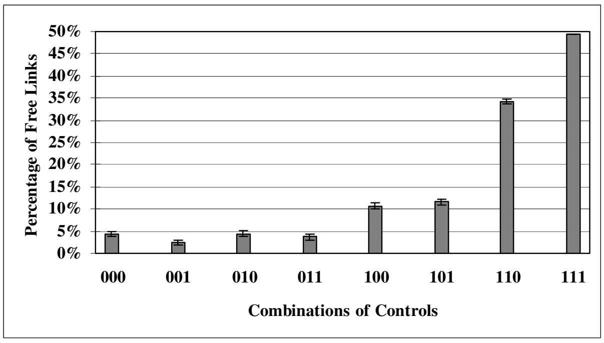
We use a variant of the reverse-path-forwarding algorithm [6] to generate a tree structure from a network. This variant of RPF can also isolate defective nodes and transceivers [7]. The analysis to determine the regular topology most similar to a given irregular network can then be conducted by calculating the fraction of free links. A *free link* is a link that connects two nodes in the generated tree structure but is not included in the tree structure.

A. Free Link Percentage

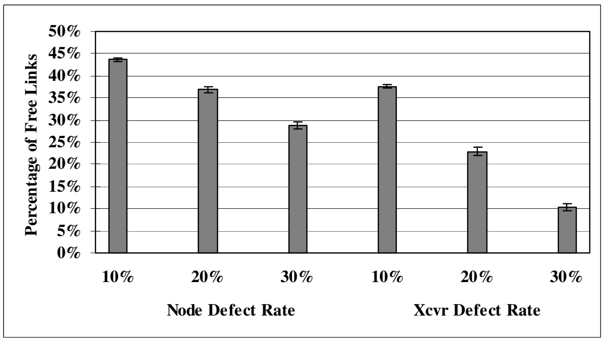
For each of the eight combinations of control except full control, we generate and analyze 20 networks using different random seeds, with 4,500 nodes in each network. When we have full control on the integration process, the generated network is always a regular mesh thus we don't need 20 networks. However, we do analyze the regular mesh network with different fractions of defective nodes or transceivers. The defects are randomly injected into the generated networks and for each defect rate we inject defects into 20 regular mesh networks and analyze them. The metric we use is free link percentage, which is shown by (1).

$$\text{Free Link Percentage} = \text{Number of Free Links}/(\text{Number of Free Links} + \text{Number of Links in the Tree}) \quad (1)$$

Fig. 3 (a) shows the percentage of free links in the networks without defects, and Fig. 3(b) shows the percentage of free links in the networks with different node defect rates and transceiver defect rates. In order to have a physical mesh-like structure, the free link percentage needs to be around 50%, which is shown as the rightmost bar in Fig. 3(a). However, we can see that in the networks without defects, the free link percentage is around or less than 10% except when there are at least both placement and orientation control. Even with almost 50% free links in the networks with full control, as the fraction of defective nodes or transceivers increases, the percentage of free links decreases. As a result, there are not enough links to form a mesh-like physical network.



(a) Networks without Defects



(b) Networks with Defects

Figure 3. Free Link Percentage in Networks with Different Control and Defect Rates

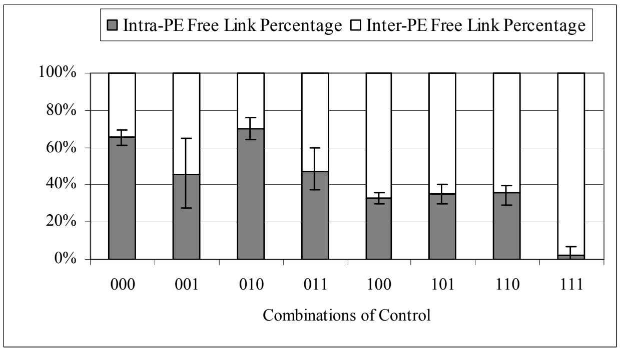
B. Free Link Distribution

The results from the previous section show that an irregular network is most similar to a tree, no matter if the irregularity results from the lack of integration control or inherent high defect rates. Since routing in a tree is generally not so efficient and flexible as in a mesh, we need to determine whether there are opportunities to utilize free links to improve routing performance (e.g., to shorten paths between certain pairs of nodes).

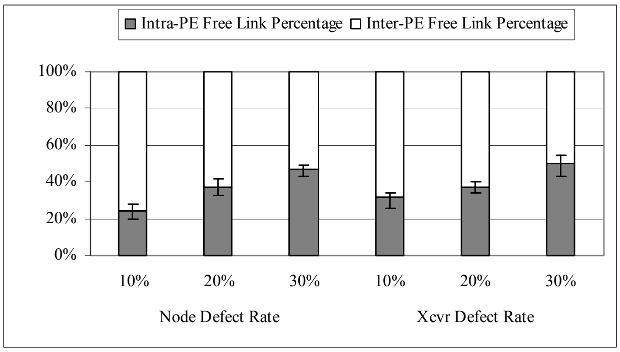
In nano-scale on-chip irregular networks, because the functionality of a single node is limited, several nodes have to be grouped together to form a functional unit [4][6]. We call such a functional unit a processing element (PE). One simple but efficient method to configure a network into PEs is traversing the tree and grouping nodes in depth-first order. Static routing paths can also be established during the traversal and routing among nodes can then follow the depth-first path or the reverse depth-first path. The choice of static routing paths results from limited resources at a single node. The depth-first path and the reverse depth-first path form an Euler path in a tree with bi-directional links. The Euler path based routing is simple to configure but not efficient since the path between one pair of nodes may go through one physical link multiple times. Free links could be used to form shorter paths between certain pairs of nodes.

In order to determine whether there are opportunities to utilize free links to improve routing performance, we first analyze how free links are distributed in the networks and then analyze how a free link shortens the Euler path between the two connected nodes. For each of the eight combinations of control except full control, we generate and analyze 20 networks using different random seeds. For each defect rate we also inject defects into 20 regular mesh networks and analyze them. Each network is configured into PEs with 18 nodes per PE (16 computational nodes, one head node and one tail node). The *head node* and the *tail node* are the first node and the last node in a PE in depth-first order separately.

Fig. 4 shows the fractions of intra-PE free links and inter-PE free links. An *intra-PE free link* is a free link connecting two nodes in the same PE, and an *inter-PE free link* is a free link connecting two nodes in two different PEs. We can see that the percentage of intra-PE free links increases as we have less control and higher defect rates. We can also see that there are very few intra-PE free links (near 0%) in a regular mesh network, which is shown as the rightmost bar in Fig. 4(a).



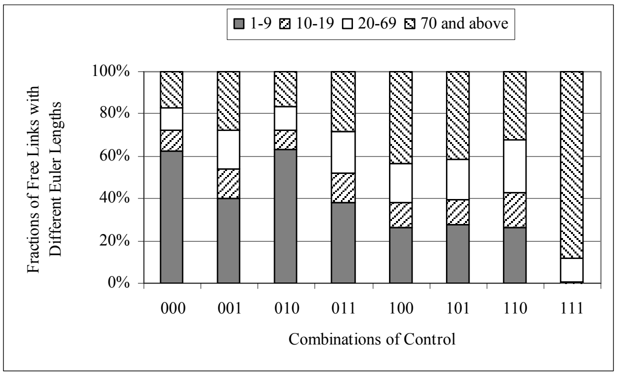
(a) Networks without Defects



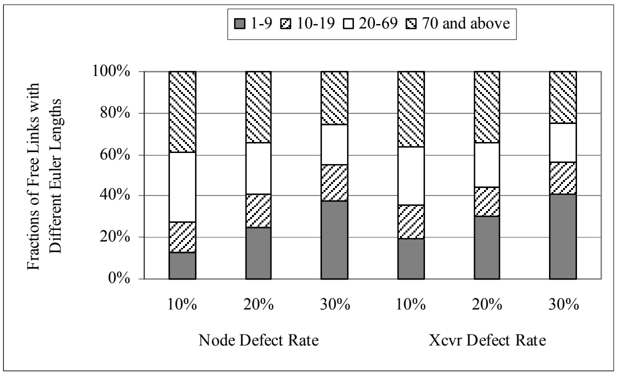
(b) Networks with Defects

Figure 4. Free Link Distribution in Networks with Different Control and Defect Rates

Fig. 5 shows how a free link shortens the Euler path between the two connected nodes. We define the *Euler length* of a free link as the number of links between two nodes connected by the free link. We can see that there are more free links with Euler length 1-9 when we have less control and higher defect rate. Our further analysis also shows that most intra-PE free links have Euler length between two and nine. We can draw the conclusion from Fig. 4 and Fig. 5 that it is possible to use free links to shorten paths between certain pairs of nodes. However, we need more analysis to determine whether it is worth utilizing free links.



(a) Networks without Defects



(b) Networks with Defects

Figure 5. Fractions of Different Path Length Reduction by One Free Link in Networks with Different Control and Defect Rates

C. *Tail-to-Head Path Length*

In order to determine whether it is worth utilizing free links, we compare the average lengths of the Euler path, the shortest path in the graph (which means that free links are used to form the shortest path) and the shortest path in the tree between the tail node and the head node in the same PE. We focus on the tail and head node pairs because the Euler path between the tail and head nodes is the longest within the same PE and can become the bottleneck for inter-PE routing.

For each of the eight combinations of control except full control and different number of nodes (3000, 4500, 6000, 7500, and 9000), we generate and analyze 20 networks using different random seeds. For each defect rate and different number of nodes we also inject defects into 20 regular mesh networks and analyze them. Each network is configured into PEs with 18/34/66 nodes per PE. A 34-node PE has 32 computational nodes, one head node and one tail node. Similarly, a 66-node PE has 64 computational nodes, one head node and one tail node.

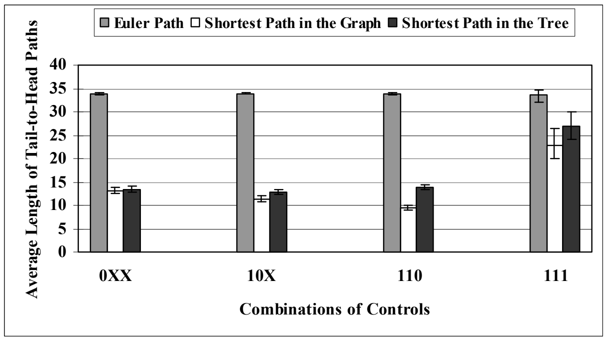


Figure 6. Comparison of Average Length of Tail-to-Head Paths in Networks with 18-node PEs and without Defects (X can be either 0 or 1)

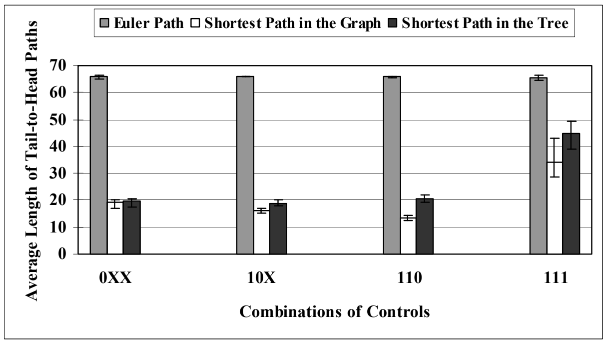


Figure 7. Comparison of Average Length of Tail-to-Head Paths in Networks with 34-node PEs and without Defects (X can be either 0 or 1)

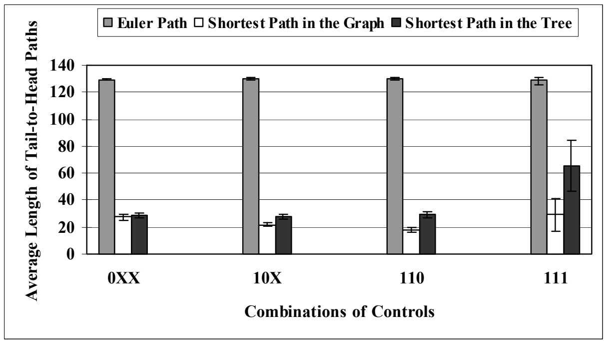


Figure 8. Comparison of Average Length of Tail-to-Head Paths in Networks with 66-node PEs and without Defects (X can be either 0 or 1)

Fig. 6, Fig. 7 and Fig. 8 compare the average lengths of the three different paths in the networks without defects. When we have no placement control, the results are similar. When we have placement control but no orientation control, the results are also similar. We can see that except in the networks with full control, there is at least a 50% reduction in the average tail-to-head path length if we follow the shortest path in the graph or the tree instead of the Euler path. Furthermore, our experiments show that this reduction does not depend on the number of nodes. In the networks with full control, the average length of the tail-to-head paths along the Euler path is similar to the networks with other types of control but the average lengths of the shortest path in the graph and in the tree between the tail and head nodes are both longer. The possible cause of the longer shortest tail-to-head paths in the graph is that there are almost no intra-PE free links in a regular mesh network while intra-PE free links has more contribution to the path length reduction than inter-PE free links. The likely reason why the shortest tail-to-head paths in the tree is also longer is that the networks with full control have regular mesh structures and the physical structure of a single PE is more likely to be linear, making the shortest path in the tree overlap more with the Euler path.

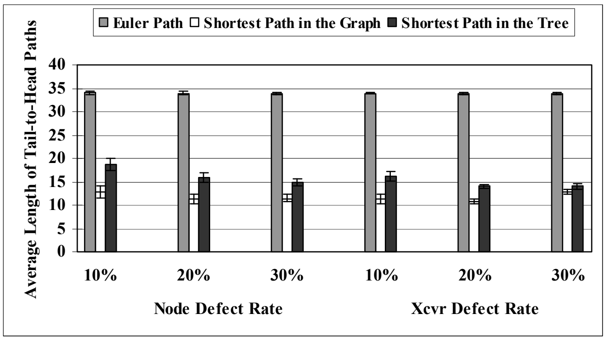


Figure 9. Comparison of Average Length of Tail-to-Head Paths in Networks with 18-node PEs and with Defects

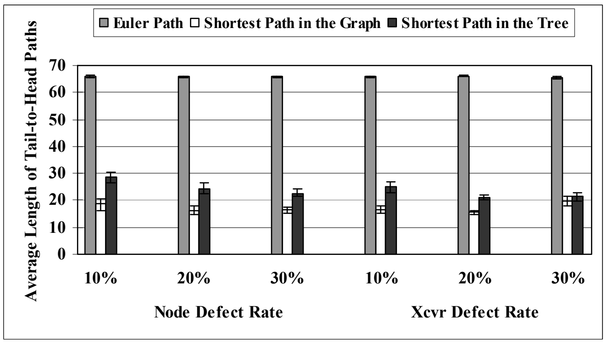


Figure 10. Comparison of Average Length of Tail-to-Head Paths in Networks with 34-node PEs and with Defects

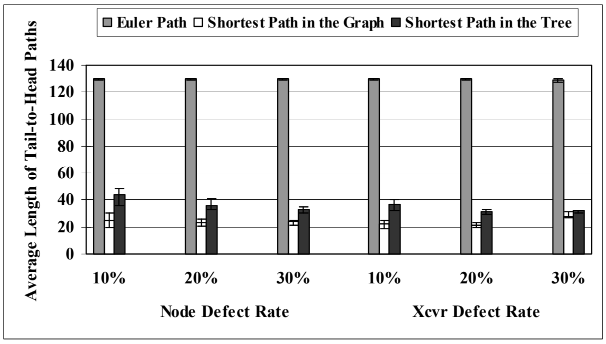


Figure 11. Comparison of Average Length of Tail-to-Head Paths in Networks with 66-node PEs and with Defects

Fig. 9, Fig. 10 and Fig. 11 compare the average lengths of the three different paths in the full-controlled networks with different node defect rates and different transceiver defect rates. When there are more defects in the networks with full control, the regular mesh structures become more irregular. Consequently, there are more intra-PE free links and the single PE structure becomes more non-linear. As a result, both the average lengths of the shortest path in the graph and in the tree become shorter and so do the differences between them.

The results in this section establish that in both irregular networks and regular networks with defects, the shortest paths in the graph and in the tree are both much shorter than the Euler path between the tail and head nodes. Moreover, in these two types of networks, the shortest paths in the graph and in the tree have similar average length, which indicates that free links actually do not have a large effect in reducing average tail-to-head path length. It would be more efficient not to use free links to form the shortest path since in both cases the average tail-to-head length paths are almost the same but it will cost extra time to configure routing paths using links that are not within the already configured tree structure.

IV. Conclusion

In this paper we analyze nano-scale on-chip irregular networks to determine the regular topology most similar to an irregular network. The results show that an irregular network is most similar to a tree, no matter if the irregularity results from the lack of integration control or inherent high defect rates. Further analysis is conducted based on configuring an irregular network into a tree structure to show whether there are opportunities to utilize links that are not included in the tree, and whether it is worth doing so. The results show that the utilization of free links does not have significant impact on shortening the paths between certain pairs of nodes.

V. Acknowledgments

This work is supported in part by the National Science Foundation (CCF-0702434).

References

[1] C. Dong, D. Chen, S. Haruehanroengra, and W. Wang, "Performance and Power Evaluation of a 3D CMOS/Nanomaterial Reconfigurable Architecture," IEEE/ACM International Conference on Computer-Aided Design, Nov. 2007.

[2] S. F. Bush, and S. Goel, "Graph Spectra of Carbon Nanotube Networks," in Proceedings of the First International Conference on Nano-Networks (NANONETS), 2006.

[3] S-H. Park, P. Yin, Y. Liu, J. H. Reif, T. H. LaBean, and H. Yan (2005) Programmable DNA Self-assemblies for Nanoscale Organization of Ligands and Proteins. Nano Letters 5, 729-733.

[4] J. P. Patwardhan, C. Dwyer, and A. R. Lebeck, "Self-Assembled Networks: Control vs. Complexity," in Proceedings of the First International Conference on Nano-Networks (NANONETS), 2006.

[5] R. C. Harting, "Computation on Self-Organized Networks," http://www.ece.duke.edu/files/ece/Harting_Rpt2007.pdf, Undergraduate Thesis, 2007.

[6] J. P. Patwardhan, C. Dwyer, A. R. Lebeck, and D. J. Sorin, "NANA: a Nano-scale Active Network Architecture," J. Emerg. Technol. Comput. Syst., vol. 2 (1), pp. 1-30, 2006.

[7] L. Demoracski, and F. Lombardi, "Connecting and Configuring Defective Nano-Scale Networks for DNA Self-Assembly," in Proceedings of the First International Conference on Nano-Networks (NANONETS), 2006.