# Improving EDP in Wireless NoC-Enabled Multicore Chips via DVFS Pruning

Wonje Choi, Shervin Hajiamin, Ryan Gary Kim, Armin Rahimi, Nillofar hezarjaribi, Partha Pratim Pande and Behrooz Shirazi
Email: {wchoi1, shajiami, rkim, arahimi1, nhezarja, pande, shirazi}@eecs.wsu.edu
The School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, USA

***Abstract*— The millimeter-wave small-world wireless NoC (mSWNoC) is shown to be capable of improving the overall latency and energy dissipation characteristics compared to the conventional wireline mesh-based counterpart. The mSWNoC helps in improving the energy dissipation even further in presence of dynamic voltage and frequency scaling (DVFS). On-chip voltage regulators are required to tune the voltage depending on the workload. Though it is possible to have multiple voltage levels by designing suitable on-chip regulators, certain voltage levels are underutilized for specific applications. Hence, unnecessary voltage levels should be pruned, reducing the design complexity of the on-chip voltage regulators. In certain circumstances, the pruned DVFS method improves the energy-delay product (EDP) compared to the fine-grained DVFS while still remaining within an acceptable performance boundary.**

***Keywords—Network-on-chip; wireless; mm-wave; dynamic voltage and frequency scaling; pruning***

## I. INTRODUCTION

The millimeter-wave small world wireless NoC (mSWNoC) is an emerging paradigm to design low-power and high-bandwidth massive multicore chips. By placing the wireless shortcuts between distant and highly communicating cores, wireless shortcuts in mSWNoC architectures carry a significant amount of the overall traffic. The amount of traffic detoured in this way is substantial, saving significant energy through the low-power wireless links [1]. The overall energy dissipation of the mSWNoC can be improved even further if the characteristics of the cores and the network (wireline links and associated switches) are optimized according to the specific workload. By implementing dynamic voltage and frequency scaling (DVFS) in an mSWNoC, its energy dissipation can be reduced without significantly compromising the overall achievable performance [1]. The range of required voltage/frequency (V/F) levels for DVFS could be optimized depending on the specific benchmark workload. In this work, we demonstrate that if the V/F levels are suitably pruned according to the workload, then the energy-delay tradeoff of the mSWNoC is not compromised and a coarse-grain regulator design will suffice. The proposed methodology is general, in the sense that it is not bound to the specific DVFS mechanism that we consider here. Instead, we focus on demonstrating how the number of V/F levels in a DVFS-based mechanism should be chosen to maintain or improve the energy-delay tradeoff.

## II. RELATED WORK

The limitations and design challenges associated with existing NoC architectures are elaborated in [2]. NoCs have been shown to perform better by inserting long-range wired links following the principles of small world graphs [3]. Wireless Network-on-Chip (WiNoC) has been shown as an enabling technology to design high-bandwidth and low-power multicore architectures [1], [4]. A comprehensive survey regarding various WiNoC architectures and their design principles are presented in [4].

DVFS is a popular methodology to optimize the power usage/heat dissipation of electronic systems without significantly compromising overall system performance. Most of the existing works principally address power and thermal management strategies for the processing cores only. The network consumes a significant part of the chip's power budget; as shown in [5] it can be almost 50% depending on the application under consideration. The works of [1] and [5] show that by incorporating DVFS in an mSWNoC, the thermal profile of a multicore chip can be improved. However, in these works the V/F levels are fine-grained which, depending on the traffic patterns generated by the benchmark application, may be unnecessary. By incorporating machine learning concepts, it is possible to determine the optimal operating point for many-core systems with extended range V/F scaling [6]. In this work, we propose evaluating the performance of a DVFS mechanism with a reduced number of allowable V/F states, to improve the energy dissipation profile of mSWNoC architectures.

## III. MM-WAVE WIRELESS NOC ARCHITECTURE

### A. mSWNoC Topology

The topology of the mSWNoC is a small-world network where the links between switches are established following a power-law model [1], [7]. In this small-world network there are still several long wireline interconnects. As these are costly in terms of power and delay, we use mm-wave wireless links to connect switches that are separated by a long distance. In [8], it is demonstrated that it is possible to create three non-overlapping channels with central frequencies of 31 GHz, 57.5 GHz, and 120 GHz respectively. Using these three channels we overlay the wireline small-world connectivity with the wireless links such that a few switches get an additional wireless port. Each of these wireless ports will have wireless interfaces (WIs) tuned to one of the three frequency channels.

The average number of connections from each switch to other switches is chosen to be 4 so that the mSWNoC does not introduce any additional switch overhead with respect to a conventional mesh. Also an upper bound is imposed on the number of wireline links attached to a particular switch so that

This work was supported in part by the US National Science Foundation (NSF) grants CCF-0845504, CNS-1059289, and CCF-1162202, CNS-1359461, and Army Research Office grant W911NF-12-1-0373.

no switch becomes unrealistically large in the mSWNoC. For a 64-core system the optimum maximum was found to be 7 [1].

### B. Routing and Communication

For each source-destination pair in the mSWNoC architecture, the wireless links, through the WIs, are only chosen if the wireless path is shorter compared to the wireline path. Token flow control [9] is used to alleviate overloading at the WIs. To avoid centralized control and synchronization, we adopt a simple wireless token passing protocol [8] as the channel access mechanism. In this scheme, a single flit circulates as a token in each frequency channel. The particular WIs possessing a wireless token can broadcast flits into the wireless medium in their respective frequencies. The wireless token is forwarded to the next WI operating in the same frequency channel after all flits belonging to a message at a particular WI are transmitted. Packets are rerouted, through an alternate wireline path, if the WI buffers are full or, it does not have the token. Since mSWNoC principally has an irregular topology [1], we consider a topology agnostic routing mechanism, Adaptive Layered Shortest-Path (ALASH) routing to ensure deadlock-free routing.

The two principal WI components are the antenna and the transceiver. The on-chip antenna for the mSWNoC has to provide the best power gain for the smallest area overhead. A metal zigzag antenna has been demonstrated to possess these characteristics, and hence is used for this work [8]. To ensure high throughput and energy efficiency, the WI transceiver circuitry has to provide a very wide bandwidth as well as low power consumption. The detailed description of the transceiver circuit is out of the scope of this paper. However, the transceiver was designed following [8]. With a data rate of 16 Gbps, the wireless link dissipates 1.95 pJ/bit. The total area overhead per wireless transceiver is 0.25 $mm^2$.

## IV. Dynamic Voltage and Frequency Scaling

The execution flow of an application running on multi-core NoCs generally contains periods of heavy computation followed by periods of inter-core data exchange. During heavy computation periods, network usage may be at a minimum, allowing the V/F of the links and switches to be tuned down in order to save energy without incurring a significant penalty to network latency [5]. On mSWNoC architectures, a significant amount of the traffic is carried by energy-efficient wireless links, reducing the wireline link utilization. Therefore, the mSWNoC increases the opportunity to perform DVFS on the network switches and links [10]. During inter-core data exchange, cores potentially idle (in the presence of memory-bounded operations) until the required data is fetched. In this period, the V/F of the cores can be tuned appropriately to reduce power dissipation. We incorporate this complementary relationship between the core utilization and the network traffic to optimize both core- and the network-level DVFS.

### A. DVFS Overhead

Voltage regulators are required in order to dynamically adjust the voltage and frequency. By using on-chip voltage regulators with fast transitions, latency penalties and energy overheads from voltage transitions can be kept low. We estimate the energy overhead introduced by the regulators due to voltage transitions as:

$$E_{regulator} = (1-\eta)\cdot C_{filter}\cdot |V_2^2 - V_1^2| \quad (1)$$

where, $E_{regulator}$ is the energy dissipated by the voltage regulator due to a voltage transition, $\eta$ is the power efficiency of the regulator, $C_{filter}$ is the regulator filter capacitance, and $V_1$ and $V_2$ are the two voltage levels before and after the transition.

In order to implement DVFS, we sample the benchmark at a fixed rate. At the beginning of each sample V/F decisions are made for all core and network elements, as we will describe in section IV.B. The optimal sample rate which gives the minimum Energy Delay Product (EDP) is experimentally found for each benchmark considered in this work. A high sample rate may reduce mispredictions by reacting to quick changes in the application characteristics, but may increase the V/F switching frequency. This may increase the regulator conversion energy described in (1) reducing the benefits of a lower misprediction penalty. On the other hand, with lower sample rates, the regulator overhead will decrease, but the penalty due to misprediction may increase.

### B. DVFS Algorithm

In this work, we develop a Markov Decision Process (MDP) -based approach to implement DVFS. We use the MDP to formulate our DVFS problem wherein we decide the V/F in each window in order to optimize the cost function. Each V/F decision taken by the MDP is given a reward or penalty value as feedback to improve the accuracy of the prediction. In the context of our work we define the MDP procedure as follows.

The MDP approach evaluates a system with $N$ V/F states, for each sample, $t$. Then, the V/F state space is $S = \{VF_1..VF_N\}$. Given a current V/F state $s \in S$, a possible next V/F state $s' \in S$, and the system-level observations (core utilization and traffic) $o$, we can compute the cost of transitioning to $s'$:

$$\begin{aligned} C(s',s,o) = (1-\gamma)[\, f(E_{comp}(s',s,o), Exec_{comp}(s',s,o)] \\ + \gamma[\, f(E_{comm}(s',s,o), Exec_{comm}(s',s,o)] \end{aligned} \quad (2)$$

where $E_{comm}$ and $E_{comp}(s',s,o)$ are the predicted energy for sample $(t+1)$, after moving to $s'$ from $s$ given $o$, for the core communication and computation respectively. $Exec_{comm}$ and $Exec_{comp}(s',s,o)$ are the predicted execution time for sample $(t+1)$, after moving to $s'$ from $s$ given $o$, for the core communication and computation respectively. $f(\bullet)$ is used to convert and normalize the energy and execution time for both computation intensity and communication into a metric such as EDP. The parameter $\gamma$ is the ratio of traffic to computation intensity for each benchmark under consideration and is used to weigh the computation to the communication portions of the cost function (2). Intuitively, the cost function is the weighted sum of the computation intensity and the communication characteristics.

To solve the MDP represented above we use the Viterbi algorithm, a dynamic programming technique, which performs the full system DVFS during the application's runtime. The Viterbi algorithm starts by capturing the characteristics of the application, making "observations", of the current time sample. At the end of each sample, $t$, the Viterbi algorithm chooses suitable V/Fs using the new system-level

observations, $o_t$, to calculate the next V/F state, $s(t+1)$, by calculating $s'_{best}$:

$$s(t+1) = s'_{best} = \arg\min_{s' \in S} C(s', s, o_t) \quad (3)$$

Ultimately, we choose *s'* that minimizes the cost function.

To provide feedback we evaluate the V/F selection compared to the expected V/F state that matches the core's utilization and network traffic in order to enhance the outcome of the algorithm. A reward is given to a V/F state if the prediction matches the expected V/F state, otherwise the prediction is given a penalty (negative reward). The accumulated reward of the V/F assignments for each core and network element will affect the cost function (2) inversely. Intuitively, with high reward, the cost will be reduced, raising the likelihood that this V/F state will be chosen again in future samples. Conversely, the lower the reward, the higher the cost and the lower the likelihood of transitioning to this state.

Fig. 1 shows an example of the Viterbi algorithm with three states for three windows. Given the second state, $s_1$ at time interval *t*, the algorithm considers the observations, $o_t$, and evaluates the cost for all branches, $C(s'_i, s_1, o_t), 0 \le i \le 2$ and takes the branch with the minimum cost, $s_1$ to $s_2$.

### C. DVFS Pruning

The required V/F levels for the network and core elements depend on the traffic patterns and core utilizations, respectively. It has been seen that for some commonly used benchmarks like SPLASH-2 and PARSEC, a majority of the links, switches and cores frequently use only a few V/F states [10]. In this case, a fine-grained DVFS mechanism with many V/F states may be unnecessary, and a DVFS mechanism implemented with a reduced number of V/F states may be sufficient.

By eliminating unnecessary V/F states, we can reduce the regulator area overhead and increase regulator efficiency. The number of V/F levels to prune was determined by first profiling the V/F level usage throughout the execution of each benchmark. The next V/F level to prune was chosen using:

$$P_{V/F} = \arg\min_x f(x) \quad (4)$$

where, $P_{V/F}$ is the V/F level to prune and $f(x)$ is the usage frequency of the particular V/F level $x$. To determine the appropriate number of pruned V/F levels, we prune the V/F level one at a time according to (4) and calculate the EDP by running full-system simulations using the DVFS algorithm in section IV.B. Then the optimal number of pruned V/F levels is given by:

$$Opt_{V/F} = \arg\min_i EDP(i) \quad (5)$$

where, $EDP(i)$ is the EDP given $i$ pruned V/F states and $Opt_{V/F}$ is the optimal number of pruned states. During the analysis in section V we investigate the effect of the number of pruned states on the execution time and EDP.

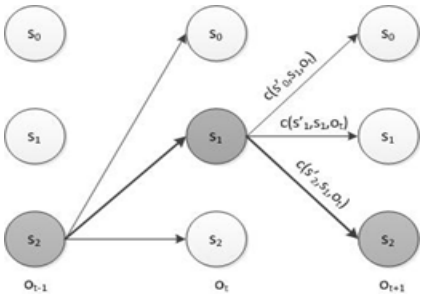


Fig.1. Example of the Viterbi algorithm for two iterations.

## V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the mSWNoC and mesh-based NoC architectures in presence of the DVFS strategy developed in section IV. We use GEM5 [11], a full system simulator, to obtain detailed processor and network-level information. We consider a system of 64 alpha cores running Linux within the GEM5 platform for all experiments. Four SPLASH-2 benchmarks, FFT, RADIX, LU, WATER [12], and one PARSEC benchmark CANNEAL [13] are considered.

Each switch port has 4 virtual channels. Hence, 4 layers are created for ALASH. All ports except those associated with the WIs have a buffer depth of two flits. The ports associated with the WIs have an increased buffer depth of eight flits to avoid excessive latency penalties while waiting for the token [8]. Energy dissipation of the network switches was obtained from the synthesized netlist by running Synopsys™ Prime Power, while the energy dissipated by wireline links was obtained through HSPICE simulations taking into consideration the length and layout of the wireline links.

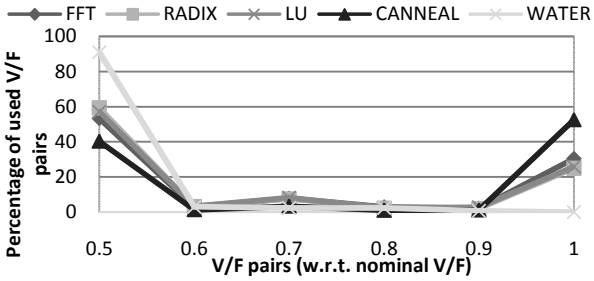


Fig. 3. Network-level V/F usage distribution.

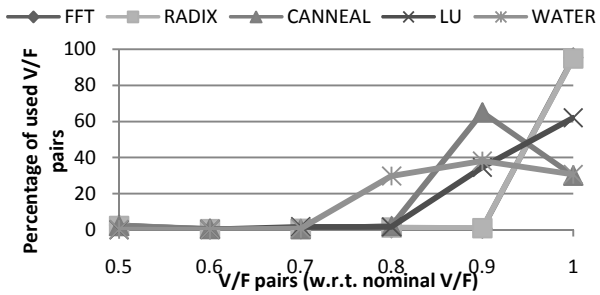


Fig. 2. Core-level V/F usage distribution.

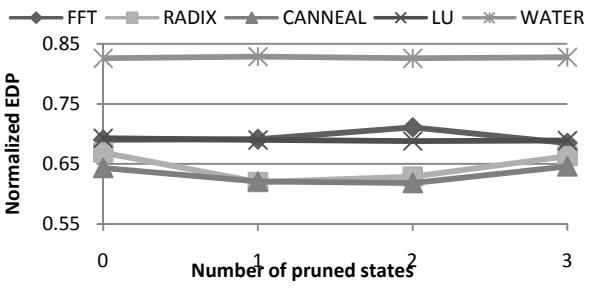


Fig. 4. Effects of the number of pruned V/F states on EDP.

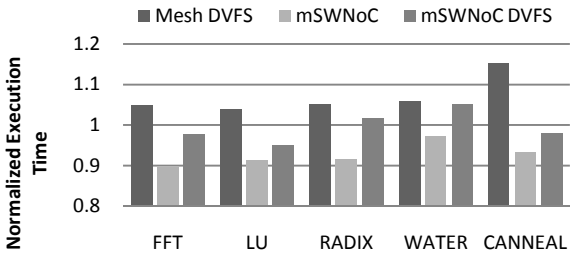


Fig. 5. Normalized execution time with respect to non-DVFS Mesh with the optimum number of V/F states pruned.

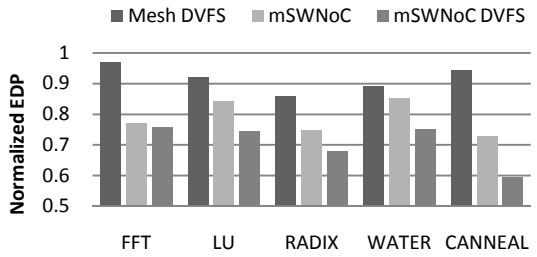


Fig. 6. Normalized EDP with respect to non-DVFS Mesh with the optimum number of V/F states pruned.

We first present the EDP of the mSWNoC by incorporating the pruned DVFS technique described in section IV.C. Fig. 2 shows the core-level V/F utilizations. It is evident that the lower states are rarely used for most of the benchmarks. As an example, for RADIX, more than 90% of the time, the cores are operating above 90% utilization. In this case, we can prune many of the lower states without paying large penalties to either energy or performance. For mSWNoC, a significant amount of the network V/Fs falls under the lowest V/F states as shown in Fig. 3. Therefore, it is possible to prune many middle states without compromising energy savings and performance. To conduct the rest of the experiments we vary the amount of pruned states from 0 to 3. Fig. 4 shows that pruning several states does not significantly reduce the performance for all benchmarks considered. In fact, in certain cases, it can even help to reduce the EDP further.

Next, we consider the overall execution time with respect to the non-DVFS mesh. Fig. 5 shows the execution time relative to non-DVFS mesh for DVFS mesh, non-DVFS mSWNoC and DVFS mSWNoC. Due to the misprediction and regulator latency, conventional wireline mesh suffers from the degradation of execution time. However, on the mSWNoC architecture, the penalties are compensated by the better network, almost zero penalty is seen with respect to the non-DVFS mesh. In fact, it even outperforms the non-DVFS mesh for many of the benchmarks, such as FFT, LU, and CANNEAL. On the mSWNoC architecture, wireline link utilizations are heavily reduced due to a reduction in hop count as well as a significant amount of traffic traversing the energy-efficient wireless channels. This reduction in wireline link utilization facilitates the network DVFS, while the mesh does not have the same architectural benefits. Thus we have more opportunity to perform DVFS on the network elements of mSWNoC. The reduction in hop-count of mSWNoC architecture also translates into execution time improvement. It reduces non-local L2 access time and hence the overall execution time. The extent of the execution time reduction depends on the communication intensity of the benchmark under consideration. A benchmark with more frequent data exchanges between the computing cores will have a better execution time improvement by using mSWNoC.

It can be observed from Fig. 6 that in each benchmark, EDP is much lower for the mSWNoC (20% on average) compared to the mesh architecture. By adding the DVFS strategy on top of the mSWNoC, we can reduce the EDP by an additional 9.5% on average without incurring performance penalty with respect to the non-DVFS mesh. In the best case (CANNEAL), mSWNoC shows 27% less EDP than the baseline mesh, and when enhanced with DVFS, EDP reduction increases to 40% while execution time outperforms the baseline mesh.

## VI. Conclusion

Millimeter-wave small-world wireless NoC (mSWNoC) is an enabling technology to design energy-efficient, high-bandwidth multicore architectures. The overall energy dissipation of the mSWNoC can be improved even further by incorporating a suitable DVFS mechanism. In this paper we demonstrate that by selectively pruning underutilized V/F states lower EDPs can be achieved. The optimal number of pruned V/F states depends on the core and link utilizations of the benchmark under consideration. For most of the benchmarks, it is possible to reduce the number of V/F levels to 3 for a system with 6 initial V/F configurations without increasing EDP.

## References

[1] P. Wettin, et al., "Energy-efficient multicore chip design through cross-layer approach," Proc. of DATE 2013.

[2] R. Marculescu, et al., "Out-standing Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives," IEEE Trans. Comput. Aided Design of Integr. Circuits Syst., vol. 28, no. 1, 2009, pp. 3-21.

[3] U. Y. Ogras and R. Marculescu, "It's a small world after all: NoC performance optimization via long-range link insertion," IEEE Trans. Very Large Scale Integr. Syst., vol. 14, no. 7, 2006, pp. 693-706.

[4] S. Deb, et al., "Wireless NoC as interconnection backbone for multicore chips: promises and challenges," IEEE JETCAS vol. 2, no. 2, 2012, pp. 228-239.

[5] J. Murray, et al., "Sustainable dual-level DVFS-enabled NoC with on-chip wireless links," Proc. of ISQED 2013.

[6] J. Da-Cheng, et al., "Learning the optimal operating point for many-core systems with extended range voltage/frequency scaling," Proc. of CODES+ISSS 2013, pp. 1-10.

[7] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," Nature 393, 1998, pp. 440-442.

[8] S. Deb, et al., "Design of an energy efficient CMOS compatible NoC architecture with millimeter-wave wireless interconnects," IEEE Trans. Comput. Vol. 62, No. 12, pp. 2382-2396, 2013.

[9] A. Kumar, et al., "Token flow control," Proc. of MICRO 2008, pp. 342-353.

[10] Jacob Murray, Nghia Tang, Partha Pratim Pande, Deukhyoun Heo and Behrooz Shirazi, "DVFS Prunning for Wireless NoC Arthitecture", IEEE Design and Test, Vol. 32, Issue 2, March 2015, pp. 29-38.

[11] N. Binkert, et al., "The GEM5 Simulator," ACM SIGARCH Computer Architecture News, 39(2), 2011, pp. 1-7.

[12] S. C. Woo, et al., "The SPLASH-2 programs: characterization and methodological considerations," Proc. of ISCA, 1995, pp. 24-36.

[13] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. Dissertation, Princeton Univ., Princeton NJ, Jan. 2011.