Energy-Efficient VFI-Partitioned Multicore Design Using Wireless NoC Architectures
Ryan Kim*, Guangshuo Liu†, Paul Wettin*, Radu Marculescu†, Diana Marculescu†, Partha Pratim Pande

ABSTRACT
In recent years, multiple Voltage Frequency Island (VFI)-based designs have increasingly made their way into both commercial and research multicore platforms. On the other hand, the wireless Network-on-Chip (WiNoC) architecture has emerged as an energy-efficient and high bandwidth communication backbone for massively integrated multicore platforms. It becomes therefore possible to exploit the small-world effects induced by the wireless links of a WiNoC to achieve efficient inter-VFI data exchanges. In this work, we demonstrate that WiNoCs can provide better latency and energy profiles compared to traditional mesh-like architecture for VFI-partitioned multicore designs. The performance gains and energy efficiency are achieved due to the low-power wireless shortcuts in conjunction with the small-world architecture. Indeed, our experimental results show energy improvements as large as 40% for multithreaded application benchmarks.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design

General Terms
Algorithms, Performance, Design.

Keywords
NoC, Wireless, Multicore, VFI, low-power.

1. INTRODUCTION
Network-on-Chip (NoC) architectures partitioned into multiple voltage-frequency islands (VFI) are capable of minimizing energy dissipation subject to performance constraints. With this architecture, it becomes possible to do efficient power and thermal management of multicore platforms via dynamic voltage and frequency scaling (DVFS). DVFS-based approaches can be applied to individual cores independently, in a centralized or distributed manner. Either way, the state of the system needs to be known entirely or to a certain extent; this involves information transfer among various parts of the multicore platform via the backbone NoC.

Towards this end, most of the existing VFI-partitioned designs use the conventional multi-hop mesh-based NoC architecture, limitations of which are well known. For large-scale systems, the inter-VFI data exchange through traditional mesh NoCs may introduce unnecessary latency and energy overheads. However, it has been already shown that small-world network architectures with long-range wireless shortcuts can significantly improve the energy consumption and achievable data rates of massive multicore-based computing platforms [1].

In this context, the millimeter-wave small-world wireless NoC (mSWNoC), a type of WiNoC, is an enabling architecture to achieve high performance and energy efficiency [1]. Fortunately, the multi-VFI platform represents a natural fit for the mSWNoC architecture; indeed, the synchronous buffers in some of the NoC routers can simply be replaced with multi-clock FIFOs as needed. We conjecture that mSWNoC can and should complement the core-level energy efficiency achieved by a suitable assignment of voltage/frequency pairs by providing a low-power and low-latency communication infrastructure for inter-VFI data exchanges.

As the main contribution to the state-of-the-art, we show that each VFI in the mSWNoC architecture can implement a suite of power management capabilities and exploit the small-worldness of the existing wireless shortcuts in order to make the power management process more efficient. Consequently, in this work, we aim at presenting the design of an energy-efficient mSWNoC architecture as the communication backbone for VFI-partitioned multicore chips. Towards this end, we demonstrate that mSWNoC provides lower latency and less energy dissipation with respect to a conventional mesh-based counterpart in presence of several SPLASH-2 and PARSEC benchmarks for a VFI-based system. Instead of presenting a generic wireless NoC design, we focus on demonstrating how this emerging paradigm can be customized for improving latency and energy dissipation profiles of VFI data exchanges.

The remaining of this paper is organized as follows: In section 2, we outline the related work. Section 3 elaborates the VFI configuration and the associated mSWNoC architecture. Section 4 presents the experimental results. Finally, Section 5 concludes the paper by summarizing the salient contributions of this work.
2. RELATED WORK

Multiple VFI designs have become the de facto platforms for both embedded and high-performance multicore platforms [2], [3], [4], [5]. Most of these designs use mixed-clock/mixed-voltage FIFO interfaces for inter-VFI communication, similar to the one described in [6] and popularly referred to as Globally Asynchronous, Locally Synchronous (GALS) architectures [7].

Conventional NoCs use multi-hop, packet-switched communication. Design and optimization of multi- and many-core systems-on-chip (SoCs) that exploit small-world effects have already been demonstrated [1], [8]. A comprehensive survey regarding various WiNoC architectures and their design principles is presented in [9], which shows the possibility of creating novel architectures by inserting on-chip wireless links.

Although exploiting the small world effects has been initially used to improve the multicore performance [10], [11], it has been later demonstrated that small-worldness can also benefit the power management via control-theoretic approaches [12]. More recently, it has been shown that mSWNoC can help improving the temperature profile of the NoC switches and links compared to a traditional mesh in presence of DVFS [13]. But this work considers distributed DVFS where each voltages and frequencies on each NoC were fine tuned as per the traffic distribution. However, it is acknowledged that by considering voltage frequency island (VFI)-based designs, the area overhead of implementing per core DVFS can be reduced.

The VFI-partitioned systems require efficient communication architectures to realize their full potential. Hence, in this work we improve the state-of-the-art by proposing a new design methodology that exploits precisely the emerging wireless NoC paradigm to bring improved energy dissipation and latency profiles to VFI-partitioned multicore chips when compared to the traditional mesh architecture.

3. VFI ARCHITECTURE ENABLED BY mSWNoC

In this section, we first describe how exactly we create various VFI clusters. Then, we elaborate on the design of the mSWNoC architecture for the VFI-partitioned system. Finally, we describe how the overall architecture needs to be fine-tuned to make it more suitable for the set of target applications.

3.1 VFI Creation

VFI clustering in multicore systems is achieved by clustering cores into beneficial subsets. The clustering can be desirable from two perspectives: 1) Reducing the overall communication cost; and 2) Improving the energy efficiency by producing better opportunities for voltage/frequency scaling. From the perspective of overall network cost, the communication between two cores within the same VFI is significantly cheaper than that between two cores residing in two different VFIs. This is because in the former case, fewer hops are needed to exchange data and so no mixed clock and voltage interfacing is required. Thus, it is desirable that the subsets of cores that heavily communicate with each other get clustered into the same VFI. From the perspective of energy efficiency, it is also desirable to cluster together cores that have similar utilizations. This way, cores running similarly behaved workloads can share the same voltage/frequency pair; this can help reducing the number of such distinct pairs without violating the performance constraints.

We consider both versions of VFI-based clustering mentioned above, namely: 1) clustering based on communication and 2) clustering based on core utilization. Our main aim in this work is to study the impacts of these clustering methodologies on the network performance and energy. The baseline architecture consists of 64 homogeneous cores, which are clustered into four 4×4 equally sized VFIs.

We start by introducing clustering notations. A vector of length 64 containing per-core utilization is denoted by \( u \). \( F \) is a 64×64 matrix containing the traffic data between every core pairs. We also use a matrix \( X \) to indicate whether or not a core is assigned to a cluster. Since we assume a total of four clusters, \( X \) is of size 64×4. Elements in \( F \) and \( u \) are normalized with respect to their maximum values. Elements in \( X \) can only be either 0 or 1. Since \( F \) and \( u \) vary for different benchmarks, the creation of VFIs is benchmark specific.

3.1.1 Communication-Based VFI Creation

We define the objective of communication cost as:

\[
\min \sum_{p,q} u_{ij} f_{comm}(p,q),
\]

where \( f_{comm} \) is defined as:

\[
f_{comm}(j,q) = \begin{cases} 1 & \text{if } j \neq q \\ \frac{1}{2} & \text{if } j = q. \end{cases}
\]

This assumes that the intra-cluster communication cost evaluates as half of the inter-cluster communication cost as an example. Intuitively, on average, the number of hops required for inter- vs. intra-cluster communication is in a ratio of 2:1 with the 4-cluster (4 equal quadrants) implementation assumed. However, this assumption can be relaxed to include a more accurate cost function reflecting exact number of intra- and inter-cluster hops based on uniform or non-uniform clustering.

The optimization is subject to the following constraints. Firstly, a core can only be part of one VFI:

\[
\forall i: \sum_j X_{ij} = 1.
\]

Also, each VFI contains precisely 16 cores,

\[
\forall j: \sum_i X_{ij} = 16.
\]

Finally,

\[
X_{ij} \in \{0, 1\}.
\]

The objective is a quadratic function in \( X \); the problem is therefore a 0-1 quadratic program and needs be solved using the branch-and-bound approach since the variables are all discrete. We use a commercial solver named Gurobi [14] to solve it.

![Figure 1. Communication-based clustering for the LU benchmark. Inter-core communication patterns before and after clustering are shown on left and right, respectively.](image-url)
Fig. 1 visualizes the effect of communication-based VFI creation for one of the SPLASH-2 benchmarks, LU [15]. The heat maps represent inter-core communication patterns, with “hot spots” being the core pairs that produce large network traffic. The left plot is the visualization of the original F matrix before clustering. The right one is the pattern after clustering. It is noticeable that originally, the hotspots are spread, while after clustering they are more grouped together. The highlighting rectangle in Fig. 1 indicates the inter-core communication that falls into the second cluster (cores 17 – 32). We can see that most of the hot spots are encapsulated in the cluster. This also proves that the assumptions made in (2) lead to reasonably good solutions.

### 3.1.2 Utilization-Based VFI Creation
To find the optimal utilization-based clustering, we group the cores that are in each quartile of the utilization values within a single VFI. In other words, the solution is to rank cores by their utilization and take the four quarters of the ranked data as the four VFIs. The resulting VFIs have minimum utilization variation, which can be easily proved by contradiction.

### 3.1.3 Voltage/Frequency Pair Assignment

Given a VFI clustering configuration, a set of V/F pairs and a performance constraint, we employ the power and performance models proposed in [16] for assigning V/F pairs. The power model, as a function of frequency, consists of dynamic power and static power. Dynamic power is modeled as the product of peak dynamic power, a fitted polynomial and a utilization term, which is also a function of frequency. Static power is similarly captured by a peak value multiplied by a polynomial. Mathematically, it is given by:

\[
\text{Power}(f) = P_{\text{peak, dyn}}^\text{VFI} \cdot (\sum_{j=0}^{N-1} a_j f^j) \cdot u(f) + c_2 + P_{\text{peak, sta}}^\text{VFI} \cdot (\sum_{j=0}^{N-1} a_j f^j)
\]

where \( f \) denotes frequency, \( P_{\text{peak, dyn}} \) and \( P_{\text{peak, sta}} \) represents peak dynamic and static powers, respectively, \( u(f) \) models utilization and all \( a_j, c_2 \) are fitting coefficients. The performance metric is:

\[
\text{Performance}(f) = f \cdot u(f)
\]

We optimally solve for the V/F value of each VFI to minimize total power of the subject system to a performance constraint specified by a target \( \alpha \in (0,1) \) which indicates that the resulting system delivers at least \( \alpha \) of performance observed at its nominal frequency.

### 3.2 mSWNoC Architecture to Support VFI

An mSWNoC architecture, where the long-range shortcuts are implemented through mm-wave wireless links operating in the 10-100 GHz range, is shown to improve the energy dissipation profile and latency characteristics of multicore chips [1]. Fig. 2 represents such an mSWNoC with 16 cores, which are divided into 3 VFIs. Each core is also associated with a NoC switch (not shown for clarity). The mSWNoC has many short-range local links, as well as, a few long-range shortcuts schematically represented by the dashed interconnects. This mSWNoC can be used as the communication backbone for a VFI-partitioned multicore platform. We note that the long-range wireless links can play important roles in achieving various goals, e.g. optimization for performance, exchanging control signals between multiple VFIs for efficient power management, etc. Hence, in this work we design mSWNoC architecture to support efficient data exchange among various VFI domains.

#### 3.2.1 Overall Network Architecture

The mSWNoC topology is designed using a power-law model [17]. The cores were first assigned to VFI clusters depending on either their communication or computation characteristics that were described in subsection 3.1. In addition to grouping up the cores into their respective VFI clusters, the cores are also arranged in such a way to decrease the distance between the highly communicating cores for both intra- and inter-VFI communication.

As the mSWNoC has an overall irregular architecture, all the switches do not have an equal number of ports. We assume an average number of connections, \(<k>\), from each NoC switch to the other switches. The value of \(<k>\) is chosen to be four so that the mSWNoC does not introduce any additional switch overhead with respect to a conventional mesh. Also, an upper bound, \(k_{\text{max}}\), is imposed on the number of ports attached to a particular switch so that no switch becomes unrealistically large in the mSWNoC. This also reduces the skew in the distribution of links among the switches. Both \(<k>\) and \(k_{\text{max}}\) do not include the local NoC switch port to the core.

Due to the nature of the VFI clustering, additional constraints need to be applied to the connectivity of the mSWNoC. The distribution of links is divided into two steps: VFI intra-cluster connections needed to ensure each cluster’s connectivity and communication, and VFI inter-cluster connections, to ensure communication between the clusters. This is to ensure that both intra-cluster and inter-cluster communications have sufficient resources and none of them becomes a bottleneck in the overall data exchange.

For each switch, \(<k>\) is divided into two parts, \(<k_{\text{intra}}>\) and \(<k_{\text{inter}}>\), the average number of intra-cluster and inter-cluster connections to other switches respectively, such that (8) is satisfied.

\[
<k> = <k_{\text{intra}}> + <k_{\text{inter}}>
\]

For the VFI intra-cluster connections, each cluster is treated separately. A network is created for each cluster such that the connectivity follows the power-law model in [17]; the network is fully connected, has average intra-cluster connectivity, \(<k_{\text{intra}}>\), and has an upper bound, \(k_{\text{maxintra}}\), on the number of wiring links attached to a particular switch at the intra-cluster connection stage. \(k_{\text{maxintra}}\) is bounded as in (9):

\[
<k_{\text{intra}}> \leq k_{\text{maxintra}} \leq k_{\text{max}}
\]

These bounds are created since \(k_{\text{maxintra}}\) can be at most the maximum number of connections to a switch of the network, \(k_{\text{max}}\) and should be at least the average connectivity of the network \(<k>\). The value \(k_{\text{maxintra}}\) is imposed while creating the intra-cluster connections in order to give to the (relatively) highly
communicating cores within different clusters a chance to have a certain amount of connections.

The VFI inter-cluster connections are created such that the connectivity also follows the same power-law model as the intra-cluster connections, has an average inter-cluster connectivity, $<k_{\text{inter}}>$, and the number of links going from one cluster to another is given by:

$$\text{InterLinks}_{ij} = \frac{c_{ij}}{\sum_{k} c_{ij}}$$

where $\text{InterLinks}_{ij}$ is the proportion of the inter-cluster links allocated for cluster $i$ that connect to cluster $j$, and $c_{ij}$ is the inter-cluster traffic going from cluster $i$ to cluster $j$.

### 3.2.2 Wireless Link Placement

To help facilitate the long distance (predominantly inter-cluster) communication, we use mm-wave wireless links to connect switches that are separated by long distances. In [1], it is demonstrated that it is possible to create three non-overlapping channels with on-chip mm-wave wireless links. Using these three channels, we can 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 the wireless interfaces (WIs) tuned to one of the three different frequency channels. From [9] it is shown that WI placement is most energy-efficient when the distance between them is at least 7 mm for the 65 nm technology node. The optimum number of WIs is 12 for a 64-core system size [13].

Therefore, we assign three WIs, working in the three non-overlapping channels, to each of the four clusters. Simulated annealing is used to find the optimal WI placements such that there are three WIs in each cluster, wireless transmissions will traverse at least 7 mm, and maximizes the metric in (11):

$$\Sigma_{w} \Sigma_{m} \Sigma_{n} \text{path}(m, n) \cdot f_{mn}$$

where $w$ is a particular wireless channel, $m$ and $n$ are the switches with attached WIs that are tuned to wireless channel $w$, path$(m, n)$ is the path length from switch $m$ to switch $n$, and $f_{mn}$ is the frequency of interaction between switches $m$ and $n$. This ensures that the highly communicating, but distant nodes, can communicate via wireless channels.

### 3.3 Communication and Channelization

This section describes the WI components and overall communication mechanism, which includes flow control and routing strategies for the mSWNoC.

#### 3.3.1 Wireless Interface (WI)

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 [18]. 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 [18]. 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$^2$.

#### 3.3.2 WI Flow Control

In the mSWNoC, data is transferred via a flit-based, wormhole routing [19]. Between a source-destination pair, the flitless links, through the WIs, are only chosen if the wireless path reduces the total path length compared to the wireline path. This can potentially give rise to hotspot situations in the WIs. Many messages will try to access the wireless shortcuts simultaneously, thus overloading the WIs, which would result in higher latency and energy dissipation. Token flow control [20] is used to alleviate overloading at the WIs. An arbitration mechanism is designed to grant access to the wireless medium to a particular WI, including the gateway WI, at a given instant to avoid interference and contention between the WIs that have the same frequency.

To avoid the need for centralized control and synchronization, the arbitration policy adopted is a wireless token passing protocol [1]. 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 at 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, they do not have the token. As rerouting packets can potentially lead to deadlock, a rerouting strategy similar to Dynamic Quick Reconfiguration (DQR), as presented in [21], is used to ensure deadlock freedom. In this situation, the current WI becomes the new source for the packet, which then is forced to take a wireline only path to the final destination.

#### 3.3.3 Adopted Routing Strategy

The adaptive layered shortest-path routing (ALASH) algorithm is the routing adopted for the mSWNoC as it represents the most suitable routing algorithm for this architecture [22]. ALASH is built upon the layered shortest path (LASH) algorithm [23], but has more flexibility by allowing each message to adaptively switch paths, letting the message choose its own route at every intermediate switch. Here we briefly explain the basic characteristics of the routing mechanism.

The LASH algorithm takes advantage of the multiple virtual channels in each switch port of the NoC switches in order to route messages along the shortest physical paths. In order to achieve deadlock freedom, the network is divided into a set of virtual layers, which are created by dedicating the virtual channels from each switch port into these layers. The shortest physical path between each source-destination pair is then assigned to a layer such that the layer’s channel dependency graph remains free from cycles. A channel dependency is created between two links in the source-destination path when a link from switch $i$ to switch $j$ and a link from switch $j$ to switch $k$ satisfies the following condition, $\text{pathlength}(i) < \text{pathlength}(j) < \text{pathlength}(k)$, where $\text{pathlength}(X)$ is the length of the minimal path between switch $X$ and the original source switch. When a layer’s channel dependency graph has no cycles, it is free from deadlocks as elaborated in [23].

For ALASH, the decision to switch paths is based on the current network conditions. We use virtual channels’ availability and current communication density of the network as the two relevant parameters for this purpose. The communication density is defined as the number of flits traversing a given switch or link over a certain time interval. In order to increase the adaptability of the routing, multiple shortest paths between all destination pairs are found and then included into as many layers as possible. It is possible to induce deadlocks if a message is allowed to switch back and forth between two or more layers. Hence, a message is not allowed to revisit a layer to maintain deadlock freedom.
4. EXPERIMENTAL RESULTS

In this section, we evaluate the latency and energy characteristics of the mSWNoC compared to the conventional wireline mesh-based NoC architectures in the presence of communication-based and utilization-based VFI clustering as elaborated in section 3.1. We use GEM5 [24], a full system simulator, to obtain detailed processor and network-level information. In all experiments, we consider a system of 64 alpha cores running Linux within the GEM5 platform. The memory system is MOESI_CMP_directory setup with private 64KB L1 instruction and data caches and a shared 64MB (1MB distributed per core) L2 cache. Three SPLASH-2 benchmarks, i.e., FFT, RADIX, LU [15], and two PARSEC benchmarks, i.e., CANNEAL and BODYTRACK [25] are considered.

The width of all wired links is considered to be the same as the flit width, which is considered to be 32 bits in this paper. Each packet consists of 64 flits. Similar to the wired links, we have adopted wormhole routing in the wireless links too. The NoC simulator uses switches synthesized from an RTL level design using TSMC 65-nm CMOS process, using Synopsys™ Design Vision. The particular NoC switch architecture has three functional stages, namely, input arbitration, routing/switch traversal, and output arbitration.

Each switch port has 4 virtual channels. Hence, four 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. Increasing the buffer depth beyond this limit does not produce any further performance improvement for this particular packet size, but will give rise to additional area overhead [1]. Energy dissipation of the network switches, inclusive of the rerouting block, was obtained from the synthesized netlist by running Synopsys™ Prime Power.

Table 1. V/F assignments for utilization-based clustering

<table>
<thead>
<tr>
<th></th>
<th>cluster 1</th>
<th>cluster 2</th>
<th>cluster 3</th>
<th>cluster 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>BODYTRACK</td>
<td>0.8/0.8</td>
<td>0.8/0.8</td>
<td>0.8/0.8</td>
<td>1.0/1.0</td>
</tr>
<tr>
<td>CANNEAL</td>
<td>0.8/0.8</td>
<td>0.8/0.8</td>
<td>1.0/1.0</td>
<td>1.0/1.0</td>
</tr>
<tr>
<td>FFT</td>
<td>0.8/0.8</td>
<td>0.8/0.8</td>
<td>0.9/0.9</td>
<td>0.9/0.9</td>
</tr>
<tr>
<td>LU</td>
<td>0.8/0.8</td>
<td>0.8/0.8</td>
<td>0.9/0.9</td>
<td>0.9/0.9</td>
</tr>
<tr>
<td>RADIX</td>
<td>0.8/0.8</td>
<td>0.8/0.8</td>
<td>0.9/0.9</td>
<td>0.9/0.9</td>
</tr>
</tbody>
</table>

Table 2. V/F assignments for communication-based clustering

<table>
<thead>
<tr>
<th></th>
<th>cluster 1</th>
<th>cluster 2</th>
<th>cluster 3</th>
<th>cluster 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>BODYTRACK</td>
<td>0.9/0.9</td>
<td>1.0/1.0</td>
<td>0.8/0.8</td>
<td>0.8/0.8</td>
</tr>
<tr>
<td>CANNEAL</td>
<td>0.9/0.9</td>
<td>0.9/0.9</td>
<td>0.9/0.9</td>
<td>1.0/1.0</td>
</tr>
<tr>
<td>FFT</td>
<td>0.9/0.9</td>
<td>1.0/1.0</td>
<td>1.0/1.0</td>
<td>0.9/0.9</td>
</tr>
<tr>
<td>LU</td>
<td>1.0/1.0</td>
<td>0.9/0.9</td>
<td>1.0/1.0</td>
<td>0.9/0.9</td>
</tr>
<tr>
<td>RADIX</td>
<td>1.0/1.0</td>
<td>0.9/0.9</td>
<td>0.9/0.9</td>
<td>1.0/1.0</td>
</tr>
</tbody>
</table>

Note: The notation is as follows: (x, y) where x is the value of $<k_{maxintra}>$ and y is the value of $<k_{maxinter}>$
while the energy dissipated by wireline links was obtained through HSPICE simulations taking into consideration the length and layout of the wireline links.

### 4.1 VFI Parameters

Following the methodology described in 3.1, we determine the VFI pairs for all clusters, both in communication- and utilization-based clustering. We simulate SPLASH-2 [15] and PARSEC [25] benchmarks on a 64-core system with all cores running at the nominal frequency. For each benchmark, we collect core utilization and the normalized amount of traffic between each two distinct core pairs. Tables 1 and 2 show these parameters at a performance target of $\alpha = 95\%$, which means to achieve at least 95% of performance of the system with all clusters running at the nominal voltage/frequency, i.e. 1.0/1.0.

#### 4.2 Creation of mSWNoC Networks

First, we determine the mSWNoC architecture suitable for the VFI configurations mentioned above. To do so, our first aim is to determine the distribution of $\langle k \rangle$ between $\langle k_{\text{intra}} \rangle$ and $\langle k_{\text{inter}} \rangle$ following (8) as mentioned in section 3.2.1 for all the benchmarks. Due to the nature of fully connected networks, $\langle k_{\text{intra}} \rangle$ has a lower bound of

$$\langle k_{\text{intra}} \rangle \geq \left( \frac{\text{connection/edge}}{\text{min. edges/cluster}} \right) \left( \frac{\text{nodes/cluster}}{N_{\text{cluster}} - 1} \right)$$

where (connection/edge) is 2, (min. edges/cluster) is $N_{\text{cluster}} - 1$ which is the minimum number of edges required to fully connect a graph with $N_{\text{cluster}}$ nodes, and (nodes/cluster) is $N_{\text{cluster}}$. Substituting the proper terms into (12) we get:

$$\langle k_{\text{intra}} \rangle \geq 2 \times \frac{N_{\text{cluster}} - 1}{N_{\text{cluster}}}$$

For this work, our VFI clustering, $N_{\text{cluster}}$, is 16; therefore the lower bound on $\langle k_{\text{intra}} \rangle$ is 1.875. Hence, for ($\langle k_{\text{intra}} \rangle$, $\langle k_{\text{inter}} \rangle$) we choose (2, 2) and (3, 1) when creating our mSWNoC for the VFI application as the value of overall $\langle k \rangle$ is 4. As mentioned in section 3.2.1, we impose this constraint so that we do not introduce any additional switch area overhead with respect to a traditional mesh. In addition, Fig. 5 demonstrates the throughput and energy characteristics for an mSWNoC without any clustering for different values of $k_{\text{max}}$. It is evident that when $k_{\text{max}}$ is 7 that provides highest throughput with lowest energy dissipation [26]. Hence, for each ($\langle k_{\text{intra}} \rangle$, $\langle k_{\text{inter}} \rangle$) we sweep $k_{\text{maxintra}}$ from 4 to 7 in increments of 1 to evaluate all possible values of $k_{\text{maxintra}}$ as established by (9) in section 3.2.1. Therefore for each VFI clustering type, networks were created for each ($\langle k_{\text{intra}} \rangle$, $\langle k_{\text{inter}} \rangle$)

![Figure 6. Network latency for communication-based clustering.](image)

![Figure 8. Network latency for utilization-based clustering.](image)

![Figure 7. Normalized energy savings in mSWNoC compared to mesh for communication-based clustering.](image)

![Figure 9. Normalized energy savings compared to mesh for utilization-based clustering.](image)
and \( k_{\text{maxintra}} \). We choose the values for \((<k_{\text{ intra}}, k_{\text{ inter}}>)\) and \( k_{\text{maxintra}} \) that provide the lowest latency and energy dissipation.

### 4.3 Performance Evaluation

In this section, we present the latency and network-level energy dissipation per flop of the mSWNoC in the presence of two types of VFI clustering mentioned above. The V/F pairs for each VFI cluster are chosen to maintain a 95% performance compared to all the clusters running with nominal V/F pairs. For completeness, we also show the characteristics of the conventional wireline mesh architecture also incorporating these two separate types of VFI clustering. While determining the overall network latency and energy dissipation, we incorporate the overheads introduced by the synchronization circuits and voltage regulators.

#### 4.3.1 Communication-Based VFI Clustering

First, we determine the configuration of the switches. Hence, our aim is to find out the values of \(<k>\) in terms of \(k_{\text{ intra}}\) and \(k_{\text{inter}}\). Corresponding to each combination of \((<k_{\text{ intra}}, k_{\text{ inter}}>)\), we determine the value of \(k_{\text{maxintra}}\) too. We consider the energy savings and gain in latency as a function of \((<k_{\text{ intra}}, k_{\text{ inter}}>)\) and \(k_{\text{maxintra}}\).

Figs. 3 and 4 show the network latency and energy respectively for both (3, 1) and (2, 2) networks with \(k_{\text{maxintra}}\) swept from 4 to 7. It can be seen that for all benchmarks considered, the (3, 1) mSWNoC network provides lower latency and energy profiles when compared to the (2, 2) counterpart for all values of \(k_{\text{maxintra}}\). For example, in Fig. 4 it can be seen that (3,1) with \(k_{\text{maxintra}}\) of 4 provides the best energy and latency savings for CANNEAL. Using the communication-based VFI clustering, Table 3 shows the optimal \(<k_{\text{ intra}}, k_{\text{ inter}}\)\) and \(k_{\text{maxintra}}\) values for communication-based VFI clustering for all of the benchmarks considered. It can also been seen from Table 3 that (3, 1) outperforms (2, 2) in every situation for both types of clustering. For example for the CANNEAL benchmark and communication-based VFI clustering, the optimal configuration \(<k_{\text{ intra}}, k_{\text{ inter}}\) is 3, \(k_{\text{ inter}}\) is 1 and \(k_{\text{maxintra}}\) is 4.

We consider the optimum mSWNoC configurations for each benchmark for further analysis. Figs. 6 and 7 show the latency and energy savings respectively compared to a conventional mesh for the communication-based VFI clustering. While determining the overall network latency, we incorporate the additional delay introduced by the synchronization circuits. It can be seen from these figures that the mSWNoC is able to achieve lower network latency than the mesh. This is due to the small-world network interconnect infrastructure of the mSWNoC with direct long-range wireless links that enables a smaller average hop-count than that of mesh [22]. However, it should be noted that the difference in latency is small due to the fact that the traffic injection load for all these benchmarks is low and the network is operating much below saturation.

When we consider the energy dissipation profile then the advantages of mSWNoC become more prominent. It can be observed from Fig. 7 that for each benchmark the network energy dissipation is much lower for the mSWNoC compared to the mesh architecture when using communication-based VFI clustering. The two main contributors of the energy dissipation are from the switches and the interconnect infrastructure. For mSWNoC, the overall switch energy decreases significantly compared to mesh as a result of the better connectivity of the architecture. In this case, the hop-count decreases significantly, and hence, on the average, packets have to traverse through less number of switches and links. In addition, a significant amount of traffic traverses through energy-efficient wireless channels; consequently allowing the interconnect energy dissipation to decrease. Also since the mesh messages are in the network longer (higher latency) they dissipate more energy than mSWNoC. For communication-based VFI clustering, mSWNoC provides highest latency savings for the CANNEAL benchmark, while it is the least for LU. The highest energy savings also obtained for CANNEAL, while the least savings is again for LU with respect to mesh.

![Figure 10. Traffic interactions between switches for LU and CANNEAL.](image)

![Figure 11. Percentage of intra-cluster communication of total traffic for all benchmarks considered and for both VFI clustering techniques.](image)

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>Communication VFI Clustering</th>
<th>Utilization VFI Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>((&lt;k_{\text{ intra}}, k_{\text{ inter}}&gt;))</td>
<td>(k_{\text{maxintra}})</td>
</tr>
<tr>
<td>FFT</td>
<td>(3, 1)</td>
<td>7</td>
</tr>
<tr>
<td>RADIX</td>
<td>(3, 1)</td>
<td>4</td>
</tr>
<tr>
<td>LU</td>
<td>(3, 1)</td>
<td>4</td>
</tr>
<tr>
<td>CANNEAL</td>
<td>(3, 1)</td>
<td>4</td>
</tr>
<tr>
<td>BODYTRACK</td>
<td>(3, 1)</td>
<td>7</td>
</tr>
</tbody>
</table>
4.3.2 Utilization-Based VFI Clustering

We followed the same methodology to determine the optimum mSWNoC configuration for utilization-based clustering too. Table 3 shows the optimal \( <k_{\text{max}}>, <k_{\text{min}}>, \) and \( k_{\text{max}} \) values using utilization-based VFI clustering for all of the benchmarks considered. We determine the latency and energy dissipation profile of the mSWNoC using these optimum parameters. Figs. 8 and 9 show the latency and energy dissipation of the mSWNoC compared to mesh respectively. Like the communication-based VFI clustering, it can be seen that the mSWNoC provides less latency and energy dissipation compared to the mesh in case of utilization-based VFI clustering too. Among all the benchmarks considered, CANNEAL provides the highest energy and latency saving, whereas the achievable gain in LU is the least, with respect to mesh.

The main reason behind the difference in savings in terms of energy dissipation and latency for CANNEAL and LU can be explained by considering their traffic distribution patterns. Fig. 10 shows the traffic interaction rates between switches for CANNEAL and LU. Benchmarks with lower injection rates and lower traffic skew (difference between Q1 and Q3 is low), such as LU, have a lower potential for energy and latency savings due to the nature of the benchmark. Whereas benchmarks with higher injection rates and greater traffic skew (very high difference between Q1 and Q3), such as CANNEAL, have more potential for energy and latency savings. When the interaction rate is higher and traffic pattern is skewed, the network becomes more stressed. Hence, mSWNoC has much greater potential to improve latency and energy compared to mesh and it is manifested in case of CANNEAL as an example.

4.3.3 Intra- and Inter-Cluster Traffic Distribution

Our analysis has shown that there isn’t a significant difference between the latency and energy dissipation characteristics of the mSWNoC in presence of communication-based and utilization-based clustering. Therefore, we analyze the traffic distribution of the two different clustering techniques, specifically the inter- and intra-cluster traffic distributions. Fig. 11 shows the percentage of intra-cluster communication against the total traffic. It can be seen that for all the benchmarks considered, the intra- to inter-cluster communication does not change significantly between the communication-based VFI clustering and the utilization-based VFI clustering. For this reason, the mSWNoC latency and energy results are very similar between the two different clustering techniques.

For a more detailed comparison, we show the inter- and intra-cluster traffic for both VFI clustering approaches across all benchmarks in Fig. 12. Each bar in the plots represents the total amount of communication traffic between two clusters. Thus, the bars on diagonals illustrate intra-cluster communication patterns while the other bars are for inter-cluster communication. Although communication-based clustering tries to encapsulate network traffic within VFI, due to the relatively homogeneous inter-core communication pattern, the effect is not significant. Therefore, the difference between communication-based clustering and utilization-based clustering is not very big.

5. CONCLUSION

The use of VFI-partitioned designs for NoC-based architectures allows for fine-grain system-level power management. VFI-partitioned NoCs can provide significant improvement in energy dissipation without introducing undue performance degradation. However, the achievable benefits are limited by the overall NoC architecture, which is traditionally a mesh.

In this work we have demonstrated that by using wireless NoC as the communication backbone for a VFI-partitioned design, it is possible to improve both the latency and energy dissipation compared to a conventional wireline mesh. The performance benefits introduced by the wireless NoC are not bound by any particular methodology to create the VFI clusters.

The gains in energy dissipation and improved latency characteristics are also benchmark-dependent. Between the SPLASH-2 and PARSEC benchmarks considered in this work, for CANNEAL we achieve the highest energy savings of more than 39.46% while for LU it is the least (25.86%).

6. ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation (NSF) grants CCF-0845504, CNS-1059289, NSF...
grant CNS-1128624, and CCF-1162202, GRC 2012-HJ-2168 and 2011-HJ-2232, as well as Army Research Office grant W911NF-12-1-0373.

7. REFERENCES