ShuttleNoC: Boosting On-chip Communication Efficiency by Enabling Localized Power Adaptation

Hang Lu[†‡], Guihai Yan[†], Yinhe Han[†], Ying Wang[†] and Xiaowei Li[†‡]
[†]State Key Laboratory of Computer Architecture, Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China
[‡]University of Chinese Academy of Sciences, Beijing, China
{luhang, yan, yinhes, wangying2009, lxw}@ict.ac.cn

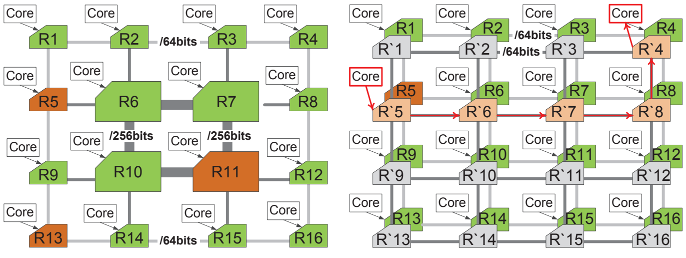
***Abstract*—Networks-on-Chip (NoC) gradually becomes a main contributor of chip-level power consumption. Due to the temporal and spatial heterogeneity of on-chip traffic, existing power management approaches cannot adapt the NoC power consumption to its traffic intensity, and hence lead to a suboptimal power efficiency. They either resort to over-provisioned NoC design that only suits for traffic spatial distribution, or coarse-grained power gating that only serves traffic temporal variation. In this paper, we propose a novel NoC architecture called *Shuttle Networks-on-Chip (ShuttleNoC)*. By permitting packets shuttling between multiple subnetworks, localized power adaptation can be achieved. Experimental results show that ShuttleNoC could achieve optimal power efficiency with up to 23.5% power savings and 22.3% performance boost in comparison with traditional heterogeneity-agnostic NoC designs.**

I. Introduction

Along with the technology scaling, power consumption has become a top design constraint for manycores. The power consumption of a manycore processor can be roughly breakdown to computation power for cores and communication power for Networks-on-Chip (NoCs). Compared with computation power which is relatively easy to regulate with many core-level techniques such as DVFS, the management for communication power is more sophisticated to hit the power efficiency frontier, largely because the sporadic variations of on-chip traffic is hard to be adapted by current power management techniques [1][2][3][4].

However, without power-efficient NoC infrastructures, we have little chance to achieve optimal chip-wide power efficiency. Recent studies show that the power consumption of NoC could reach as high as 80 watts, a large slice of total chip power, under 16nm technology node for a mesh connected multicore [5]. The same trend also emerges at commercial designs, i.e. Sun's "Niagara" processor, the interconnect takes nearly 17% of total power [6]; this percentage reaches up to 28% in Intel's "Intel80" processor [7]. In future manycore era, NoC power consumption is expected to increase more rapidly.

Localized power adaptability is an essential design challenge for NoC power management. NoC should be delivered proportional power quota to each node (router + link) in accordance with its local traffic intensity [2]. This requirement, though intuitively simple, is hard to accomplish, given the sporadic traffic variations in not only *temporal*, but also *spatial* dimension. The traffic intensity may vary at nanoseconds magnitude [1], and distribute heterogeneously across different locations [8]. Due to these limitations, traditional NoC designs fail to catch such traffic heterogeneity in both temporal and spatial dimensions, and usually renders the NoC at a suboptimal power efficiency state.



(a) Spatial-oriented (b) Temporal-oriented

Fig. 1. NoC architectures supporting only *spatial* or *temporal* heterogeneity.

Some prior studies [8] devote to design a heterogeneous NoC based on traffic "spatial" distribution. For example, in Figure 1(a), big routers are relatively designed for boosting network performance by providing higher bandwidth (256bits), in comparison with small power efficient routers (64bits). This design philosophy assumes that routers in central area of a mesh will handle heavier traffic than those at boundaries, so a larger power consumption of big routers is essentially expected to obtain proportional performance enhancement. Whereas, on-chip traffic distribution is never fixed and hotspot may migrate anywhere in NoC, especially when it handles multi-program workloads, so small routers like $R5$ and $R13$ may also encounter intense traffic while big routers are almost idle. Under these circumstances, power efficiency will suffer.

At the other end of the spectrum, some solutions aim to design a configurable NoC to capture the traffic "temporal" heterogeneity. Multi-NoC [2][9], as a representative, evenly breaks down the original single NoC into multiple subnetworks (subnets). By the employment of power gating, an entire subnet could be powered on/off according to the temporal traffic variations, as shown in Figure 1(b). However, such temporal-oriented approach ignores the spatial traffic distribution, and is not a comprehensive solution either. For example in Figure 1(b), there are two subnets with each 64bits wide. If a traffic flow intends to traverse from node 5 to node 4 under dimensional order routing, and sub-router $R5$ is already a hotspot, the only solution is to wake up all sub-routers along the path in subnet 2 $(R'5, R'6, R'7, R'8, R'4)$ to serve this traffic flow, even though sub-routers $R6, R7, R8, R4$ in subnet 1 are sufficient to handle this traffic flow. Hence, such power management approach degrades the power efficiency of these nodes.

*This work is supported in part by National Basic Research Program of China (973) under grant No. 2011CB302503, in part by NSFC under grant No. (61432017, 61202056, 61100016, 61076037, 60921002).

978-1-4799-7792-5/15/$31.00 ©2015 IEEE

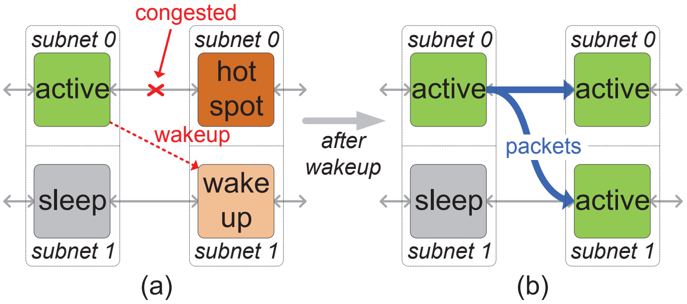


Fig. 2. Power adaptation based on local traffic intensity.

The *temporal/spatial heterogeneity* yields different network demands and power efficiency consequences. Ideally, we would expect each node's bandwidth to be in line with its *local* traffic intensity. Power management should be capable to *adapt* to both temporal and spatial heterogeneity of on-chip traffic. For example, Figure 2(a) shows a scenario that packets are blocked by a "hotspot" sub-router. If it could "wake up" the "higher level" subnet of neighboring node, and steer the congested packets into the newly powered sub-router, congestion condition would be effectively alleviated, just as Figure 2(b) shows. By contrast, if an active sub-router is more than required due to light traffic condition, we can offload the traffic back into the "lower level" subnet and control the offloaded sub-router to "sleep" state to save power.

This observation motivates a *localized* power management scheme that takes both temporal and spatial traffic heterogeneity into account. Therefore, this paper proposes a novel NoC architecture, called Shuttle Networks-on-Chip (*ShuttleNoC*), to achieve optimal power efficiency. By monitoring traffic intensity at each node, sub-router and its associate links could be powered on/off to implement localized power adaptation. Without losing connectivity, packets in a subnet are allowed to *shuttle* into active subnets, rather than waking up sleeping sub-router of the same subnet to proceed. In particular, this paper makes the following contributions:

- *We propose ShuttleNoC architecture to achieve localized power adaptation.* We leverage our insights from the weaknesses of existing heterogeneity-agnostic NoC designs. By providing *shuttle* ability, it avoids unnecessary activation of sub-routers, so temporal/spatial heterogeneity of on-chip traffic could be better supported.
- *We propose a localized power management mechanism and associate router microarchitecture to achieve optimal power efficiency.* By receiving power-gating or wake-up requests from other subnets, the bandwidth could be scaled locally, which provides a unique opportunity to achieve optimal power efficiency.

II. Shuttle Networks-on-Chip Architecture

In order to achieve the aforementioned packet shuttling to attain localized power adaptation, we need some modifications upon traditional NoC architectures. The proposed ShuttleNoC design is shown in Figure 3, without loss of generality, we start describing its microarchitecture using a 4x4 mesh connected NoC with two subnets. The packet shuttling is implemented through two hierarchies: 1) at chip level, apart from temporal-oriented approach, a particular hardware called "*Link Reconfiguration Module*" (referred to as LRM hereafter) is added between neighboring nodes, which makes previous separated subnets related to each other. The link of a subnet is reachable to other subnets after reconfigured in LRM; 2) for an individual sub-router, we need additional control paths connected to the LRM to transmit "shuttle requests", so LRM could reconfigure the links accordingly and steer the flit to the desired subnet.

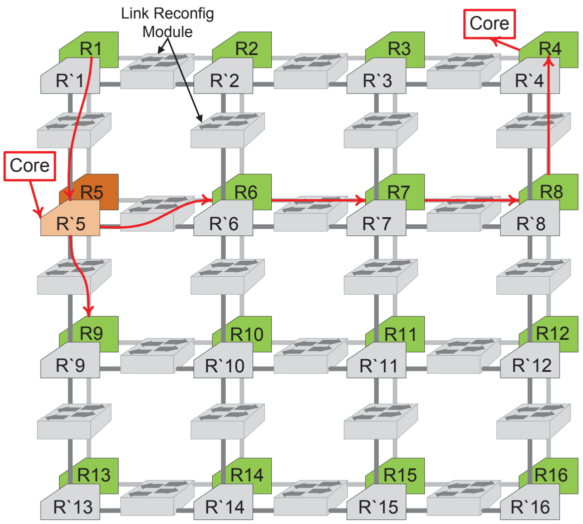


Fig. 3. Shuttle Networks-on-Chip architecture. Dedicated *link reconfiguration module* is responsible for packet shuttling between subnets. Packets are no longer required to stay within only one subnet after injected. This augmented link flexibility provides a unique opportunity to localized power adaptation.

ShuttleNoC resolves two power efficiency limitations associated with previous heterogeneity-agnostic approaches. First, it eliminates the unnecessary activation of sub-routers, hence avoiding the over-provisioned power consumption at less-congested nodes. This benefit, unique in ShuttleNoC, stems from the flexible link connectivity provided by LRM. Taking the same packet forwarding example in Section I, only $R'5$ is necessary to be powered on as Figure 3 shows. Packets could shuttle from $R'5$ to $R6$, which is a light-loaded sub-router ($R'6$ is still sleeping), and proceed to the destination in *subnet 1*. Thus, we have 4 less router activations. For other passing-by packets, i.e. $R1 \rightarrow R9$, $R'5$ could also be used for shuttling, leaving $R'1$ and $R'9$ at sleep state. Hence, overall power consumption could be reduced significantly for ShuttleNoC. Second, apart from spatial-oriented approach, the bandwidth of a node is never fixed but dynamically changed, so traffic spatial variations could be well adapted to further improve power efficiency.

Link Reconfigure Module. The detail implementation of LRM is shown in Figure 4(a). In order to implement packet shuttling, we need additional control and data paths between neighboring nodes. For example, at *east* output of $R1$ in the figure, sub-router could issue a shuttle request (`shuttle_req`) for the destination subnet, i.e. *west* input of $R'2$. The data path is reconfigured by controlling four multiplexers in LRM. According to different enabler combinations, we can get different subnet connections. In Figure 4(a), supposing a packet intends to shuttle from $R1$ to $R'2$, LRM configures the route as $R1 \rightarrow N_a \rightarrow N_d \rightarrow R'2$. Although we adopt LRMs, it only introduces a mild power and area overhead. Detailed evaluation of ShuttleNoC will be shown in Section IV.

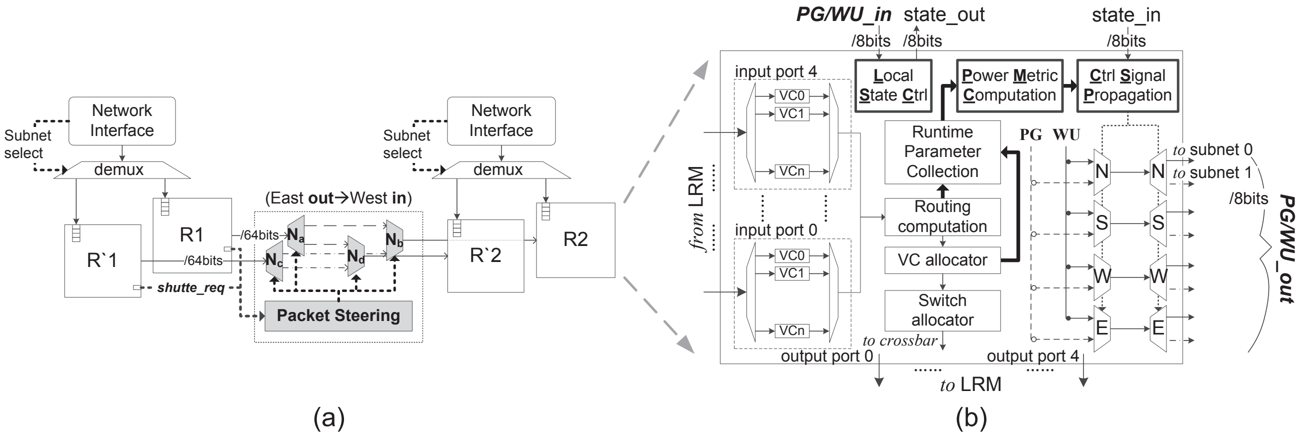


Fig. 4. Link reconfiguration module & dedicated router microarchitecture supporting localized power adaptation.

III. Localized Power Adaptation based on ShuttleNoC

In ShuttleNoC, packets are no longer required to stay within one subnet after injected, so opportunity arises that we can obtain optimal power efficiency by manipulating such packet shuttling operation. In this section, we firstly specify the router microarchitecture dedicated for ShuttleNoC, and then detailed power adaptation mechanism is discussed.

A. Router Microarchitecture in ShuttleNoC

Like previously proposed power management techniques [1][2][10], localized power adaptation also relies on runtime statistics as the reference to power on/off a sub-router. Packet shuttling is implemented between neighboring sub-routers, so their power state must be obtained in real time. In ShuttleNoC, three modules: *Local State Ctrl, Power Metric Computation* and *Ctrl Signal Propagation* are added to serve these purposes, as shown in Figure 4(b).

Local State Ctrl (LSC). LSC is used to control state transition of its host sub-router. Bandwidth adaptation signals, namely `PG/WU_in`, are received from neighboring sub-routers of 4 directions. “PG” and “WU” indicate a “power gating” or “wake-up” request, respectively. If we use 2 subnets in ShuttleNoC, it is hence an 8-bit signal (2 sub-routers×4 directions) and each bit is possible to be a PG or WU. By analyzing their numerical relationship, LSC decides to power on/off its host sub-router.

Power Metric Computation (PMC). In order to quantify traffic intensity, we employ PMC module to compute the microarchitectural parameters at runtime. Previous work [2] has proposed several plausible congestion detection metrics, such as local injection queue occupancy, average or maximum buffer occupancy, etc. These metrics can certainly be used in ShuttleNoC; however, to measure traffic intensity, the metric should be able to pinpoint the precise data path or direction that causes the packet contention. PMC then selects *average flits queueing delay for each output direction* as the intensity metric (Eq. 1).

$$QD_{outdir} = \frac{\sum_{i}^{N_{outdir}} delay_i}{N_{outdir}}, outdir \in (E, W, N, S) \quad (1)$$

N_{outdir} stands for the total number of flits heading direction $outdir$. $delay_i$ is the queuing delay that flit i must be retained in its input virtual channel, due to the failure of virtual channel allocation (VA) or switch allocation (SA). An increasing QD_{outdir} may indicate that output virtual channels in $outdir$ direction may be limited, and a power adaptation request is supposed to be issued. These parameters can easily be obtained as soon as a packet has finished certain pipeline stages, without introducing additional overhead.

Ctrl Signal Propagation (CSP). Note that once PMC intends to issue a power adaptation request (PG/WU) to a certain output direction, CSP module is designed to inform the target sub-router, in that direction, of its request. `state_in` includes the on/off status sent from neighbor LSCs. CSP, based on this information, propagates power adaptation request by controlling symmetrically organized MUXes, as Figure 4(b) shows. Each bit of `PG/WU_out` will connect to the corresponding sub-router’s LSC module.

B. Power Adaptation Mechanism

State Transition in LSC. Clearly, the efficacy of power management mechanism depends on the dynamic router status, so similar to [1][2], we also use three states to depict a router: `Active, Sleep, Wakeup`. The state is maintained in LSC module. `Active` indicates the router is currently working and packet shuttling is applicable, while `Sleep` and `Wakeup` means the router is power-gated or waking up, respectively. Packets are not allowed to shuttle into “sleep” and ”wakeup” routers.

In Section III-A, we have shown that LSC receives power adaptation requests (`PG/WU_in`) sent from neighbor CSPs (`PG/WU_out`), so LSC and CSP are coupled as two sides of handshaking operation. Figure 5 shows such interaction. Note that it only shows two neighboring sub-routers of the same subnet, while LSC also interacts with CSPs of other subnets.

For a particular sub-router, state transition from `Active` to `Sleep` must satisfy two conditions: 1) `Num(WU)` equals to 0, which denotes all bits in `PG/WU_in` are PGs; 2) no packet is remained in local input buffers. LSC can then safely power down this router to reduce power consumption. By contrast, if LSC detects arbitrary PG/WU combinations in `PG/WU_in`, state transition from `Sleep` to `Wakeup` depends on if `Num(WU)` attains a pre-defined threshold. If so, it means the local bandwidth is more requested to be broadened. LSC will then power up its host router. Once `Wakeup`, it may take $10 \sim 20$ cycles that `Active` state will be finally attained,

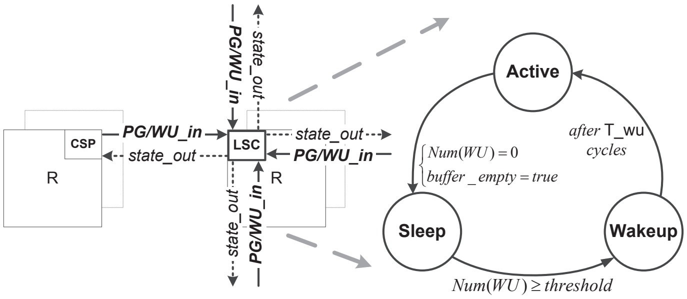


Fig. 5. LSC handshaking with CSP & router state transition.

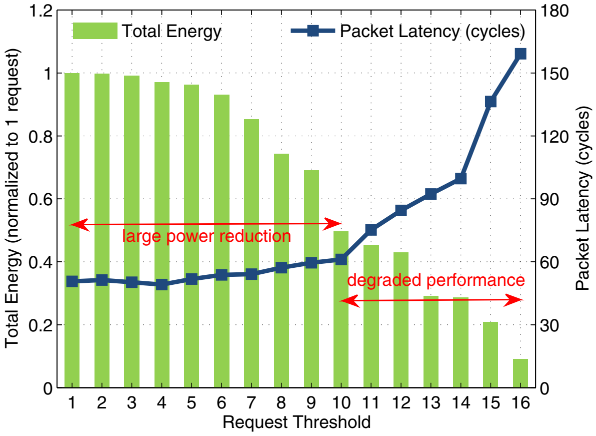


Fig. 6. Request threshold (`Num(WU)`) impact to ShuttleNoC power and performance.

according to [1][2][5]. We use `T_wu` to indicate this transition delay.

Subnet Selection in CSPs. As specified above, CSP processes the power adaptation requests generated by PMC. It must decide which sub-router is supposed to be activated/deactivated based on their on/off status. Detailed procedure is shown in 1. We stipulate the activation of subnets must be in order. For example, if subnet 1 is already active and subnet 2, 3 and 4 are off, subnet 2 is then chosen as the activation candidate (line 8 to 11). Shutting down, on the contrary, follows a reverse order by starting from the highest-level active subnet (line 2 to 5).

IV. Evaluation

In this section, we evaluate ShuttleNoC and the proposed localized power management approach. First, we introduce the platform and baselines we use. Second, we show various results in terms of performance and runtime power efficiency.

A. Experimental Setup

Platform. We modified Booksim2.0 [11] simulator to run application traces from full system simulation. The fundamental NoC topology is a 4x4 and 8x8 mesh. On-chip router is configured with a four-stage pipeline plus one cycle for link traversal or shuttling. We use 4 virtual channels for the input buffer with 5-flit depth each. For the power evaluation, we use DSENT [12] power model, which is fed with the statistics from NoC simulations of PARSEC [13] benchmark. We also employ Synopsis Design Compiler [14] to obtain the area overhead of various NoC designs under SMIC90 technology library.

Algorithm 1 Subnet Selection in CSP

```
Input: State of neighboring routers: $state\_in$; Subnetworks: $N$;
    PMC requests: $requests$;
Output: Subnet selected: $n$;
1: for each $req < dir, opa >\in requests$ do
2:   if $opa == PG$ then
3:     for ($i = N - 1; i >= 0; i - -$) //shut down from the highest-level do
4:       if $state\_in[dir][i] == WU$ then
5:         return $n$; //power off subnet $n$ at output $dir$
6:       end if
7:     end for
8:   else if $opa == WU$ then
9:     for ($i = 0; i < N; i + +$) //wake up from the lowest-level do
10:      if $state\_in[dir][i] == PG$ then
11:        return $n$; //wake up subnet $n$ at output $dir$
12:      end if
13:    end for
14:  end if
15: end for
```

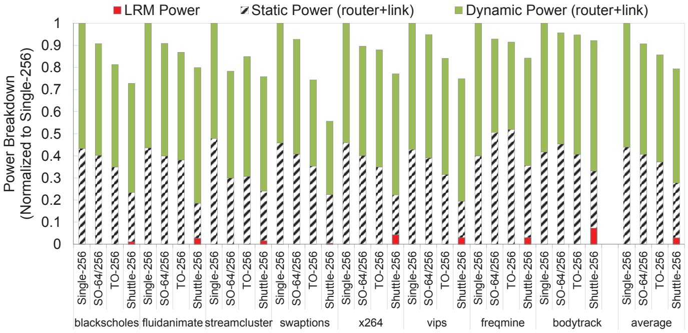


Fig. 7. Overall NoC power breakdown (normalized to Single-256).

Baselines. We use three baselines to prove the efficacy of ShuttleNoC in power adaptation: (1) the first one is a traditional single NoC with no power management involved. We configure the bandwidth of NoC platform as 256bits, so the first baseline is referred to as "single-256"; (2) the second baseline, referred to as "SO-64/256", is the "spatial-oriented" approach, and two bandwidth configurations are set as 64bits and 256bits; (3) the third one, referred to as "TO-256", is the "temporal-oriented" approach with 4 subnets and each one 64bits wide. The same configuration is also set for the proposed ShuttleNoC. Besides, `T_wu` is set as 20 cycles for a *Waking-up* sub-router to finally attain *Active* state [1][2]. Since we use 4 subnets, `state_in/out` and `PG/WU_out/in` are both 16bits (4 sub-routers × 4 directions) for ShuttleNoC. Note that even if the bandwidth configuration for the baselines and ShuttleNoC is not exactly the same, the maximum bandwidth (256bits) is equal.

B. Results and analysis

1) Request threshold & ShuttleNoC Responsiveness: We firstly evaluate the impact of "wakeup" request threshold, `Num(WU)` in Figure 5, to the ShuttleNoC responsiveness. Speaking of responsiveness, we evaluate the total energy consumption and average packet latency, with `Num(WU)` tuned from the minimum to the maximum. The energy results are normalized to the minimum-request scenario (1 on X-axis). As can be seen from Figure 6, the packet latency remains almost stable with threshold varied from 1 $\sim$ 10, while the total energy reduces nearly 50%. This phenomenon proves that larger threshold contributes to the power reduction due to the prolonged sub-router sleeping cycles without compromising performance. However, the latency starts to climb significantly to 160 cycles (207.7% degradation) from 10 $\sim$ 16. NoC per-

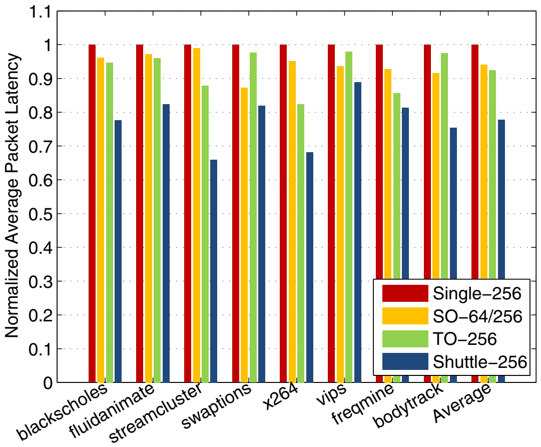


Fig. 8. Overall NoC performance (normalized to Single-256).

formance is suffered under this scenario, because sub-routers become too "lazy" to respond to the increasing WU requests, even if the total energy continues to decrease. Therefore, for the rest of the experiments, we set the request threshold as 10 to acquire a better energy/performance tradeoff, but ShuttleNoC can actually work at other threshold values, based on the total power budget available or the intended NoC power efficiency.

2) Overall Power & Performance: ShuttleNoC enables the localized power adaptation, so in this set of experiment, we show how much benefit it brings in terms of overall power reduction in a 8x8 mesh. Figure 7 plots the breakdown of NoC power to examine the impact of each component. Compared to "Single-256", "SO-64/256" and "TO-256", ShuttleNoC shows substantial static power reduction by 19.2%, 16.0% and 12.5%. The improvement comes from the localized power adaptation. ShuttleNoC only activates/deactivates sub-router based on the local traffic intensity, rather than fixing the bandwidth as done in "SO-64/256". Compared to "TO-256", it does not need to affect neighboring node to maintain the network connectivity, so static power is also reduced. On the other hand, ShuttleNoC employs LRM to achieve packet shuttling, so dynamic power may increase due to the frequent link reconfigure operation. As shown in the figure, it incurs 3.8% power overhead for LRM, and 1.7%, 3.3% dynamic power increase compared to "SO-64/256" and "TO-256", respectively. However, the abundant static power reduction still renders ShuttleNoC an overall power savings of 23.5%, 14.3% and 9.3%.

To further prove the effectiveness of ShuttleNoC, we then show the overall performance enhancement in comparison with three baselines. Average packet latency is used as the performance metric. The result in Figure 8 shows that ShuttleNoC improves network performance by 22.3%, 16.4% and 14.7% on average. The runtime support of traffic heterogeneity in ShuttleNoC effectively avoids potential hotspots. Generally speaking, ShuttleNoC offers improvements in both power and performance. Hence, power efficiency is undoubtedly boosted compared to the baselines.

3) Runtime Power Efficiency: To explore the implication of power/performance benefit brought by ShuttleNoC, we further evaluate the power efficiency variation periodically, by executing `canneal` benchmark under ShuttleNoC and the two baselines. We use Energy Delay Product (EDP) as the power efficiency representative. The result is shown in Figure 9, in which we observe that EDP values remain almost constant (5% variation on average) for ShuttleNoC. Whereas,TO-256 exhibits large fluctuations during execution, because of the coarse-grained, subnet-level power control. For SO-64/256, it suffers from the severe performance degradation when heavy traffic migrates to small routers.

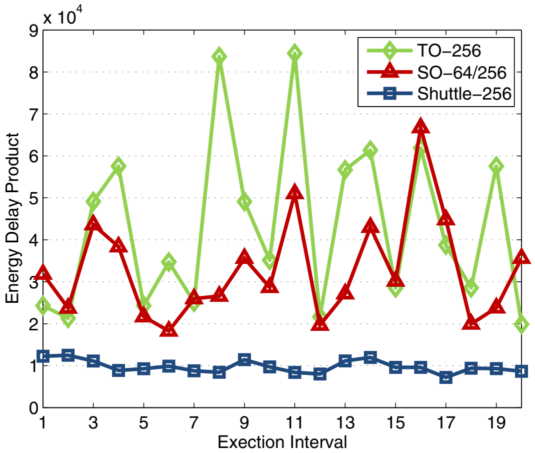


Fig. 9. Energy Delay Product (EDP) at runtime.

4) Heterogeneity Adaptation Analysis: The smooth power efficiency of ShuttleNoC stems from the effective adaptation of traffic heterogeneity. As evidence, this set of experiment traces the latency variation of every node in a 4x4 NoC, by executing the same `canneal` benchmark. As shown in Figure 10, SO-64/256 exhibits obvious latency variations between $40 \sim 190$ cycles. Due to the large bandwidth of big routers in the center, node $6, 7, 10, 11$ show a moderate latency variation compared to nodes at boundaries like 1 or 15, but still around 100 cycles. TO-256 shows a mild latency variation around 110 cycles on average. By sharp contrast, ShuttleNoC shows a more smooth latency variation around $30 \sim 40$ cycles. Such near-constant latency further proves that ShuttleNoC has the unique ability of localized bandwidth adaptation, and thus more possible to achieve optimal power efficiency.

C. Overhead Analysis

ShuttleNoC relies on link reconfiguration module and dedicated hardware in routers to fulfill the power adaptation purpose. We evaluate the implementation cost of ShuttleNoC by comparing its area overhead to other NoC designs, as shown in Figure 11. We analyze several area contributive components. ShuttleNoC incurs a mild increase of total NoC area, due to complicated link layout and dedicated modules. Fortunately, LRM only consists of several multiplexers, and does not introduce significant area overhead. Other control logics in total occupy 4.3% of overall NoC area. Note that although ShuttleNoC exhibits more area overhead, it does not consume a larger power, because the proposed power adaption mechanism brings more power savings, and even gives a better play in network performance.

V. Related Work

The concept of ShuttleNoC is similar to bandwidth scaling techniques proposed in [15] and [16], in which bi-directional links are employed to resolve low resource utilization. However, they do not target power efficiency in NoCs. [17] allows

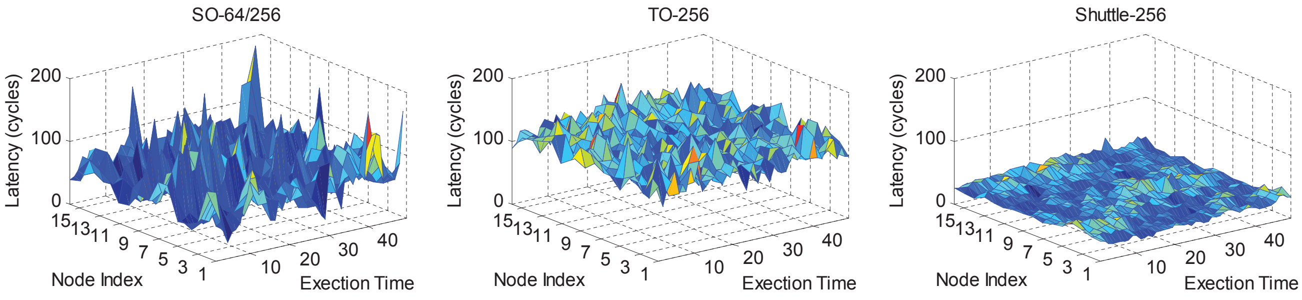


Fig. 10. Heterogeneity adaptation comparison.

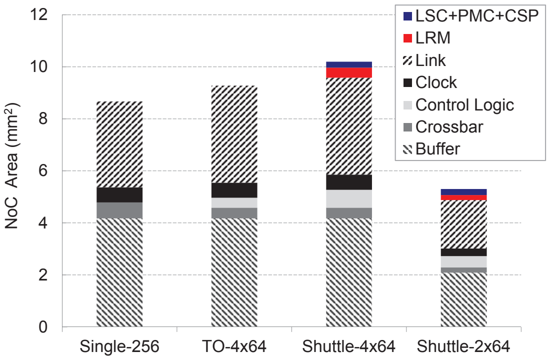


Fig. 11. Area overhead analysis of multiple NoC designs.

concurrent transmission of multiple flits on a link. [8] proposed that flit size can be decoupled from channel width and transmitted on big/little routers respectively. [18] combines multiple flits together if possible and send through a wider channel. These spatial-oriented approaches may encounter complex hardware design and suffer from runtime variations of traffic distribution.

Multiple Networks-on-Chip (MultiNoC) proposed in [2] can provide proportional energy consumption in accordance with the in-flight traffic, but it unnecessarily activates an entire subnetwork to adapt to local traffic intensity. [19] uses bandwidth/frequency asymmetric Multi-NoC for latency/bandwidth sensitive workloads; commercial processors like TILE64 [20] and TRIPS [21] also employ multiple NoCs to isolate different message classes. However, these designs do not consider localized power adaptation either. To the best of our knowledge, the proposed ShuttleNoC is the first work that targets localized power adaptation to achieve optimal power efficiency.

VI. Conclusion

This paper proposes ShuttleNoC, a novel NoC architecture to enforce optimal power efficiency. Unlike previous temporal and spatial-oriented approach, ShuttleNoC achieves localized power adaptation to serve runtime traffic heterogeneity. By precise state transition mechanism and dedicated link reconfiguration module, packets are allowed to shuttle between multiple subnetworks. Compared with temporal-oriented approach, ShuttleNoC avoids unnecessary activation of sub-routers to obtain lower power consumption. Besides, ShuttleNoC does not resort to fixed bandwidth configurations as in spatial-oriented approach, and hence yields higher network performance. We therefore believe that ShuttleNoC is a promising scheme to achieve optimal power efficiency in future many-core processors.

References

[1] L. Chen and T. M. Pinkston, "Nord: Node-router decoupling for effective power-gating of on-chip routers," in *MICRO2012*, pp. 270–281.

[2] R. Das, N. Satish, K. Satpathy, and G. Dreslinski, "Catnap: energy proportional multiple network-on-chip," in *ISCA2013*, pp. 320–331.

[3] H. Lu, G. Yan, Y. Han, B. Fu, and X. Li, "Riso: Relaxed network-on-chip isolation for cloud processors," in *DAC2013*, pp. 1–6.

[4] Y.-C. Huang, K.-K. Chou, C.-T. King, and S.-Y. Tseng, "Ntpt: On the end-to-end traffic prediction in the on-chip networks," in *DAC2010*, pp. 449–452.

[5] S. Borkar, "Networks for multi-core chips," in *Special Session at ISLPED 2007*.

[6] H. McGhan, "Niagara 2," in *Microprocessor Report, 2006*.

[7] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-ghz mesh interconnect for a teraflops processor," *Micro, IEEE*, vol. 27, pp. 51–61, Sept 2007.

[8] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for cmps," in *ISCA2011*, pp. 389–399.

[9] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. Brown, and A. Agarwal, "On-chip interconnection architecture of the tile processor," *Micro, IEEE*, vol. 27, pp. 15–31, Sept 2007.

[10] H. Matsutani, M. Koibuchi, K. Ikebuchi, D.and Usami, H. Nakamura, and H. Amano, "Ultra fine-grained run-time power gating of on-chip routers for cmps," in *NOCS2010*, pp. 61–68.

[11] Booksim2.0, https://nocs.stanford.edu/.

[12] C. Sun, C. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic, "Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *NOCS2012*, pp. 201–210.

[13] C. Bienia et al, "The parsec benchmark suite: characterization and architectural implications," in *PACT2008*, pp. 72–81.

[14] Synoposis, Design Compiler, Version D-2010.03-SP2, June 2010.

[15] R. Hesse, J. Nicholls, and N. Jerger, "Fine-grained bandwidth adaptivity in networks-on-chip using bidirectional channels," in *NOCS2012*, pp. 132–141.

[16] Y. C. Lan, S. H. Lo, Y. C. Lin, Y. H. Hu, and S. J. Chen, "Binoc: A bidirectional noc architecture with dynamic self-reconfigurable channel," in *NOCS2009*, pp. 266–275.

[17] L. Wang, P. Kumar, K. H. Yum, and E. J. Kim, "Apcr: an adaptive physical channel regulator for on-chip interconnects," in *PACT2012*, pp. 87–96.

[18] R. Das, S. Eachempati, A. Mishra, V. Narayanan, and C. Das, "Design and evaluation of a hierarchical on-chip interconnect for next-generation cmps," in *HPCA2009*, pp. 175–186.

[19] A. K. Mishra, O. Mutlu, and C. R. Das, "A heterogeneous multiple network-on-chip design: An application-aware approach," in *DAC2013*, pp. 1–10.

[20] S. Bell et al, "Tile64 - processor: A 64-core soc with mesh interconnect," in *ISSCC2008*, pp. 88–99.

[21] P. Gratz, C. Kim, K. Sankaralingam, H. Hanson, P. Shivakumar, S. Keckler, and D. Burger, "On-chip interconnection networks of the trips chip," *Micro, IEEE*, vol. 27, pp. 41–50, Sept 2007.