

Análise Arquitetural Comparativa do Desempenho de Redes-em-Chip baseada em Simulação

Eduardo Alves da Silva, Cesar Albenes Zeferino

Laboratório de Sistemas Embarcados e Distribuídos (LEDS)
Centro de Ciências Tecnológicas da Terra e do Mar (CTTMar)
Universidade do Vale do Itajaí (Univali) – Itajaí, SC – Brasil

{eas, zeferino}@univali.br

***Resumo.** As Redes-em-Chip são a infraestrutura de comunicação adotada nos sistemas many-core atuais e diversos estudos têm comparado o desempenho de arquiteturas de rede. Porém, esses trabalhos carecem de análises quantitativas ou de abrangência arquitetural. Para superar essas limitações, este artigo apresenta uma biblioteca de componentes de simulação que possibilita a modelagem de diferentes alternativas arquiteturais e a coleta de dados para uma comparação quantitativa. Um amplo conjunto de experimentos foi realizado para demonstrar a efetividade da biblioteca implementada. Os resultados obtidos permitem identificar as melhores alternativas de rede para diferentes cenários de tráfego.*

1. Introdução

As Redes-em-Chip (NoCs – Networks-on-Chip) se tornaram a arquitetura de comunicação preferencialmente adotada nos sistemas computacionais multiprocessados integrados em uma única pastilha de silício, os SoCs (Systems-on-Chip). Isso ocorreu devido às vantagens apresentadas por essas redes em relação às arquiteturas de comunicação tradicionalmente utilizadas nesses sistemas, como o barramento. Como principais vantagens destacam-se a escalabilidade de desempenho, a reusabilidade e o paralelismo de comunicação [Andriahantenaina et al. 2000].

Desde o surgimento das primeiras NoCs, várias alternativas arquiteturais foram propostas para atender a demanda de desempenho dos novos projetos. Usualmente, essas propostas são avaliadas por meio de uma descrição arquitetural ou por comparativos baseados em experimentos em relação a uma arquitetura de referência. Nessa segunda linha, os experimentos geralmente são realizados com o uso de simuladores [Ogras and Marculescu 2013].

De um lado, o problema é que descrições arquiteturais não dão subsídios para a tomada de decisão, pois carecem de dados quantitativos de desempenho das arquiteturas analisadas. De outro, cada grupo de pesquisadores desenvolve seu próprio simulador e realiza uma exploração arquitetural limitada, comparando a sua proposta com uma ou poucas arquiteturas de referência.

Inserido neste contexto, este trabalho busca auxiliar projetistas de sistemas integrados na tomada de decisão por meio de uma análise quantitativa do desempenho de arquiteturas de NoCs. Para isso, foi desenvolvida uma biblioteca de componentes simuláveis que foi integrada a uma ferramenta de simulação de arquiteturas de

interconexão intrachip. A biblioteca e a ferramenta foram implementadas de forma a flexibilizar a adição de novos componentes, o que possibilita que propostas futuras possam ser facilmente avaliadas frente às arquiteturas existentes e disponíveis no ambiente. Para demonstrar a diversidade dessa biblioteca e a sua flexibilidade, foram realizados experimentos que permitiram obter informações sobre as arquiteturas com melhor desempenho para diferentes cenários de tráfego.

As principais contribuições do trabalho residem na infraestrutura para avaliação unificada do desempenho de NoCs e uma análise comparativa de um conjunto de arquiteturas de comunicação intrachip com base em dados quantitativos.

O restante deste artigo está organizado em quatro seções. A Seção 2 discute os trabalhos relacionados que serviram para evidenciar o problema alvo deste trabalho. A Seção 3 apresenta aspectos do desenvolvimento da biblioteca proposta. Já a Seção 4 descreve os experimentos realizados e discute os resultados obtidos. Por fim, a Seção 5 apresenta as conclusões do trabalho.

2. Trabalhos Relacionados

Para evidenciar a lacuna na avaliação de desempenho de arquiteturas de Redes-em-Chip em que trabalhos descritivos não subsidiam escolhas e trabalhos experimentais não possuem abrangência arquitetural, uma Revisão Sistemática da Literatura foi aplicada. Essa revisão teve o objetivo de identificar os trabalhos mais recentes que realizam análises comparativas de NoCs. A pesquisa bibliográfica foi feita nas principais bibliotecas digitais da área de Ciência da Computação (IEEE Xplore, ACM Digital Library e ScienceDirect) e também com uso do Scopus e do Google Scholar. A expressão de busca foi adaptada para cada uma das bases de forma que correspondesse ao mesmo padrão de pesquisa. A forma genérica da expressão é a seguinte:

```
(noc OR "network on chip") AND  
(comparison OR compare OR comparing OR comparative)
```

Somente trabalhos escritos na língua inglesa e que contivessem os termos da expressão acima nos seus títulos foram buscados. Como critério de seleção, foi definido que apenas trabalhos publicados a partir do ano 2011 seriam analisados. Foram excluídos trabalhos cujos títulos e/ou resumos não abordassem a avaliação de desempenho de NoCs baseada em simulação ou que tratassem de tecnologias alternativas (*e.g.* redes ópticas e sem fio). Com a aplicação desses critérios, foram selecionados 17 trabalhos, os quais foram analisados e caracterizados, como ilustram as tabelas a seguir.

A Tabela 1 apresenta uma visão geral que ilustra o método de comparação utilizado, o atributo arquitetural avaliado e as alternativas arquiteturais exploradas em cada trabalho. O principal método de avaliação utilizado é a simulação. Alguns trabalhos realizam comparações qualitativas (método descritivo) que não dão subsídios suficientes às suas análises, enquanto outros realizam o mapeamento em silício (síntese). Os níveis de implementação dos simuladores variam na acurácia e no tempo de execução. Porém, para avaliar novas propostas, são utilizados modelos de alta acurácia (em nível de ciclos). Em alguns casos, são utilizados simuladores já existentes, como nos trabalhos de Yin et. al. (2012), Hao et al. (2011) e Pandey e Gupta (2012), em que foram adotados os simuladores ORION 2.0, Nirgam e NS-2, respectivamente. A

maior parte dos trabalhos analisados tem por objetivo comparar entre duas a três topologias ou algoritmos de roteamento (alguns comparam técnicas de chaveamento), porém, nenhum aborda mais de um atributo arquitetural. Em geral, os trabalhos comparam uma proposta de solução em relação à uma abordagem convencional.

Tabela 1. Visão geral dos trabalhos analisados na revisão sistemática

| Trabalhos | Método | Atributo | Alternativas |
|---|---------------------|-------------------------|---|
| [Sadawarte, Gaikwad e Patrikar 2011] | Descritivo | Chaveamento | Circuito, mensagem, pacote, Wormhole e Virtual Cut-through |
| [Hao et al. 2011] | Simulação | Roteamento | XY e Odd-Even |
| [Ghidini et al. 2012] | Simulação | Topologia | Malha 2D e Malha 3D |
| [Yin et al. 2012] | Simulação | Topologia | Malha 2D e Honeycomb |
| [Kalimuthu e Karthikeyan 2012] | Descritivo | Trabalhos da literatura | <i>n.i.</i> |
| [Ju e Yang 2012] | Simulação | Topologia | Malha 2D, Toróide 2D e Malha Hierárquica |
| [Romanov e Lysenko 2012] | Simulação | Topologia | Malha 2D, Toróide, Butterfly, Árvore gorda e 6 topologias irregulares |
| [Manna, Chattopadhyaya e Sengupta 2012] | Síntese e simulação | Topologia | Mesh of Tree original e modificada |
| [Parandkar, Dalal e Katiyal 2012] | Simulação | Roteamento | XY, Odd Even e DyAD (Dynamic Deterministic Adaptive) |
| [Pandey e Gupta 2012] | Simulação | Topologia | Malha 2D e topologias irregulares |
| [Du et al. 2013] | Simulação | Roteamento | XY, Turn-model e Retrograde-turn-model |
| [Jetly 2013] | Síntese e simulação | Chaveamento | Wormhole e Store-and-Forward |
| [Slame e Abdelkader 2014] | Simulação | Topologia | Malha 2D e Árvore gorda |
| [Wang et al. 2014] | Síntese | Chaveamento | Wormhole e Virtual Cut-Through |
| [Radfar, Zabihi e Sarvari 2014] | Simulação | Topologia | Malha 2 e 3D, Toróide 2 e 3D e Hiper cubo 2 e 3D |
| [Jaina, Kumar e Sharmac 2015] | Síntese | Topologia | Malha 2D, Toróide 2D e Anel |
| [Harbin e Indrusiak 2016] | Simulação | Níveis de abstração | Em nível de ciclos e transação |

n.i.: não informado

Apenas seis dos 17 trabalhos analisados foram identificados como fortemente relacionados ao tema alvo deste estudo. Os demais foram desconsiderados por não utilizarem simulação e pelo pouco detalhamento sobre a arquitetura das NoCs e os cenários de tráfego adotados, o que impossibilitaria a sua reprodução.

A Tabela 2 apresenta a caracterização dos trabalhos quanto às topologias e demais atributos arquiteturais analisados. A Tabela 3, por sua vez, identifica os métodos de injeção de tráfego e as métricas para avaliação de desempenho.

É possível verificar que a maior parte dos trabalhos é focada na avaliação de topologias e que a Malha 2D é a mais utilizada. Isso ocorre porque a topologia é um dos atributos que mais impacta nos limites de desempenho da infraestrutura de comunicação e a Malha 2D serve de referência para novas propostas. Poucos trabalhos avaliaram outros atributos, tais como a memorização e a arbitragem, sendo que nenhum dos que foram analisados abordou técnicas de controle de fluxo.

Um aspecto a ser destacado é que estabelecer cenários de avaliação justos para os comparativos não é uma tarefa trivial. Em alguns trabalhos, é nítido que a experimentação realizada é tendenciosa para “beneficiar” uma arquitetura em relação à outra, gerando vieses de avaliação. Os cenários de comunicação comumente são

implementados por meio de geradores de tráfego sintético para focar na avaliação do desempenho da rede. Porém, em um dos trabalhos, foi utilizado um modelo de sistema real justificando que tráfego sintético não representa o comportamento de aplicações.

Tabela 2. Atributos arquiteturais abordados

| Trabalho | Topologia | Chaveamento | Controle de fluxo | Arbitragem | Memorização |
|--|-------------------|--------------|---------------------|----------------|--|
| Ghidini et al. (2012) | Malha 3D | Wormhole | Baseado em créditos | RR | Nas entradas, dimensionáveis e 1 VC |
| | Malha 2D | | | | |
| Yin et al. (2012) | Malha 2D | <i>n.i.</i> | <i>n.i.</i> | <i>n.i.</i> | Nas entradas, dimensionáveis em profundidade e VCs |
| | Malha Honeycomb | | | | |
| Ju e Yang (2012) | Malha 2D | <i>n.i.</i> | Handshake | RR | Nas entradas com 64 posições de 8 bits (64 bytes) |
| | Toróide 2D | | | | |
| | Malha Hierárquica | | | | |
| Manna, Chattopadhaya e Sengupta (2012) | Mesh-of-Tree | Wormhole | <i>n.i.</i> | RR | Nas entradas com 6 flits de profundidade |
| Jetly (2013) | Malha 2D | Wormhole SAF | <i>n.i.</i> | <i>n.i.</i> | <i>n.i.</i> |
| Sllame e Abdelkader (2014) | Malha 2D | Wormhole | <i>n.i.</i> | RR | Nas entradas dimensionáveis em profundidade e VCs |
| | Árvore Gorda | | | | |
| Harbin e Indrusiak (2016) | Malha 2D | Wormhole | Baseado em créditos | Preemptivo | Com VC |
| | | | | Não-preemptivo | Sem VC |

Onde: VC: Virtual Channel; SAF: Store-and-Forward; RR: Round-robin; *n.i.*: não informado

Tabela 3. Modelos de tráfego e métricas de avaliação

| Trabalho | Modelo de tráfego | Métricas de desempenho |
|--|---|--|
| Ghidini et al. (2012) | Sintético; Uniforme e complemento; Injeção constante a 800Mbps | Latência, vazão e utilização de recursos |
| Yin et al. (2012) | Gerador sintético com taxas de injeção de 1%, 0.5% e 0.1% | Latência |
| Ju e Yang (2012) | Sintético; Uniforme; Injeção de pacotes sem pontos de contenção | Latência, vazão e utilização recursos |
| Manna, Chattopadhaya e Sengupta (2012) | Sintético; Uniforme; Tráfego em rajadas | Latência e vazão |
| Jetly (2013) | Sistema real; requisições mestre-escravo | Vazão e frequência de operação |
| Sllame e Abdelkader (2014) | Sintético; Uniforme; Pacotes de 200, 300 e 400 bytes | Latência e Vazão |
| Harbin e Indrusiak (2016) Ghidini et al. (2012) | Aplicação veicular, decodificador H.264 e sintético | Tempo de execução das aplicações |

Por fim, observou-se que há muita variação na precisão dos resultados apresentados. Alguns trabalhos apresentam resultados precisos e realizam uma discussão suficientemente detalhada. Outros, entretanto, apresentam resultados superficiais sem informar o quanto uma arquitetura é melhor em relação a outra, ou, ainda, carecem de uma discussão fundamentada para justificar o motivo de uma configuração ser superior à outra.

3. Desenvolvimento

O presente trabalho foi desenvolvido com a ferramenta de avaliação de desempenho de NoCs denominada RedScarf [Silva 2014]. Essa ferramenta é composta de uma interface gráfica (*front-end*), desenvolvida com o *framework* Qt [The Qt Company 2017], e um simulador (*back-end*) implementado utilizando SystemC [Accellera 2017] com acurácia em nível de ciclos. O simulador é baseado em uma biblioteca de componentes para a construção de roteadores e NoCs. O *front-end* é responsável pela interface com o usuário e oferece diversos recursos para configuração dos modelos, execução dos experimentos e análise dos resultados. Já o *back-end* realiza as simulações e gera os relatórios sobre os pacotes transferidos pela rede. Os dados desses relatórios são apresentados no *front-end* por meio de gráficos e tabelas, o que facilita a análise das métricas. A geração de tráfego é baseada nos métodos propostos por Tedesco (2005) e os modelos de latência e vazão baseiam-se nas definições de Dally e Towles (2004).

Neste trabalho, os componentes da biblioteca foram reimplementados e estendidos na forma de *plug-ins* para facilitar a adição de novos componentes. Os atributos arquiteturais suportados pela versão atual da biblioteca estão listados na Tabela 4, a qual evidencia a diversidade de alternativas em relação aos trabalhos supracitados. Destaca-se que o algoritmo de roteamento é específico a cada topologia.

Tabela 4. Atributos arquiteturais da biblioteca de componentes

| Topologia | Roteamento | Chaveamento | Controle de fluxo | Árbitro | Memorização |
|-------------|----------------------------|---------------------|---------------------------------|---|---|
| Anel | Volta Mínima e Restrição 0 | Circuito e wormhole | Baseado em créditos e handshake | Distribuído com as políticas de arbitragem: estática, rotativa, randômica e Round-Robin | Entradas e/ou saídas sem ou com até 8 canais virtuais |
| Anel Cordal | Crossfirst | | | | |
| Barramento | Endereçamento direto | | | | |
| Crossbar | | | | | |
| Malha 2D | XY, WF, NF, NL e Odd-Even | | | | |
| Torus 2D | DOR | | | | |
| Malha 3D | XYZXY ¹ | | | | |

Onde: WF: West-first; NF: Negative-first; NL: North-last; DOR: Dimension Order, XY para o Torus 2D

¹ Roteamento que já considera o uso de TSV (Through Silicon Via)

Para as simulações, um cenário de tráfego pode ser definido para avaliação entre diferentes topologias e cada topologia possui pelo menos um algoritmo de roteamento de caminho mínimo. Essa escolha foi feita na tentativa de estabelecer um critério comum aos roteamentos de diferentes topologias, já que cada topologia possui suas particularidades, bem como para evitar disparidades na avaliação e criar vieses. Os mecanismos de chaveamento, controle de fluxo, arbitragem e memorização podem ser combinados para qualquer configuração. Uma exceção é o barramento, no qual não é possível estabelecer canais virtuais.

As diferentes topologias foram agrupadas em três classes: (i) Não-ortogonais (ou de uma dimensão), Barramento, Anel, Anel Cordal e Crossbar; (ii) Ortogonais 2D, Malha 2D e Torus 2D; e (iii) Ortogonais 3D, a Malha 3D. O formato do cabeçalho do pacote na rede foi ajustado para cada uma das classes de topologia. Os campos de

endereço do cabeçalho permitem endereçar até 256 nodos. Nas redes não-ortogonais, os endereços são compostos de uma palavra de 8 bits. Nas redes ortogonais 2D, os endereços são divididos em coordenadas X e Y, com 4 bits para cada dimensão, o que permite um arranjo máximo de 16x16 nodos. Nas redes 3D, os endereços X, Y e Z são compostos de 3, 3 e 2 bits, respectivamente. O maior arranjo de uma rede 3D é 8x8x4. Nas redes 3D, foram reservados campos para endereçamento individual de TSVs (Xtsv e Ytsv), pois assume-se a possibilidade de que nem todos os nodos possuam links verticais. Porém, por simplificação, considera que um enlace vertical interliga todas as camadas da topologia.

O protocolo de comunicação utiliza dois bits de enquadramento em todos os *flits* (*flow control units* – unidades de controle de fluxo) para indicar se o *flit* é cabeçalho, carga útil ou terminador e pode ter no mínimo 32 bits e no máximo 510 bits. Além da identificação dos endereços de origem e de destino do pacote, o *flit* de cabeçalho contém campos para diferenciação de fluxo nos canais virtuais, identificação de *threads* e estabelecimento/cancelamento de circuito.

4. Experimentos

Os experimentos visaram avaliar cada atributo arquitetural das infraestruturas de comunicação. Os requisitos temporais de tráfego foram definidos e todos os experimentos feitos utilizaram a mesma configuração. A largura de banda requerida pelos fluxos foi fixada em 320 Mbps e a frequência de operação foi variada. Ou seja, é a capacidade da rede que varia, de 320 Mbps/canal (10 MHz) a 3200 Mbps/canal (a 100 MHz), como mostra a Tabela 5. A largura do canal de comunicação é de 32 bits e todos os pacotes possuem 128 bits, quatro *flits* com o cabeçalho.

Tabela 5. Relação entre frequência de operação e o tráfego oferecido

| Largura do canal (bits) | Frequência de operação (MHz) | Largura de banda do canal (Mbps) | Largura de Banda Requerida (Mbps) | Tráfego Oferecido (%) |
|-------------------------|------------------------------|----------------------------------|-----------------------------------|-----------------------|
| 32 | 100 | 3200 | 320 | 10,0 |
| | 90 | 2880 | | 11,1 |
| | 80 | 2560 | | 12,5 |
| | 70 | 2240 | | 14,3 |
| | 60 | 1920 | | 16,7 |
| | 50 | 1600 | | 20,0 |
| | 40 | 1280 | | 25,0 |
| | 30 | 960 | | 33,3 |
| | 20 | 640 | | 50,0 |
| | 10 | 320 | | 100,0 |

As distribuições espaciais utilizadas para a definição dos destinos foram: Bit-reversal, Perfect Shuffle, Butterfly, Complemento, Transposto e Uniforme. Geradores de tráfego sintético são usados para se ter total controle sobre o processo de injeção. Como critério de parada, 100.000 pacotes devem ser entregues. Desses, os primeiros 40.000 (40%) foram descartados para reduzir/evitar os vieses sistemáticos de inicialização dos simuladores relacionados ao período de aquecimento da rede. No total, cerca de 1.500 simulações foram executadas (parte delas ilustradas a seguir).

Para as topologias, foram realizados experimentos com 16 e 64 nodos nos arranjos 4x4 e 8x8, para as redes 2D, respectivamente, e 4x2x2 e 4x4x4 nas redes 3D. Para estabelecer um cenário de comparação justo, todas as topologias foram avaliadas com algoritmos de roteamento determinísticos e de caminho mínimo. Os demais atributos são: chaveamento Wormhole; controle de fluxo baseado em créditos; árbitro Round-robin; e *buffers* de entrada com quatro *flits* de capacidade (sem canais virtuais).

Dos experimentos, verificou-se que nem mesmo o Crossbar possui melhor desempenho em todos os cenários. Por exemplo, no tráfego Uniforme nas redes de 16 nodos, o Torus 2D e a Malha 3D foram superiores (Figura 1). Isso porque o tráfego aceito (Figura 1.a) é maior e a latência média (Figura 1.b) é menor. No *jitter* (Figura 1.c), destaca-se a maior concentração de pacotes entregues com latências pequenas e menor dispersão na distribuição, o que também evidencia a superioridade dessas NoCs.

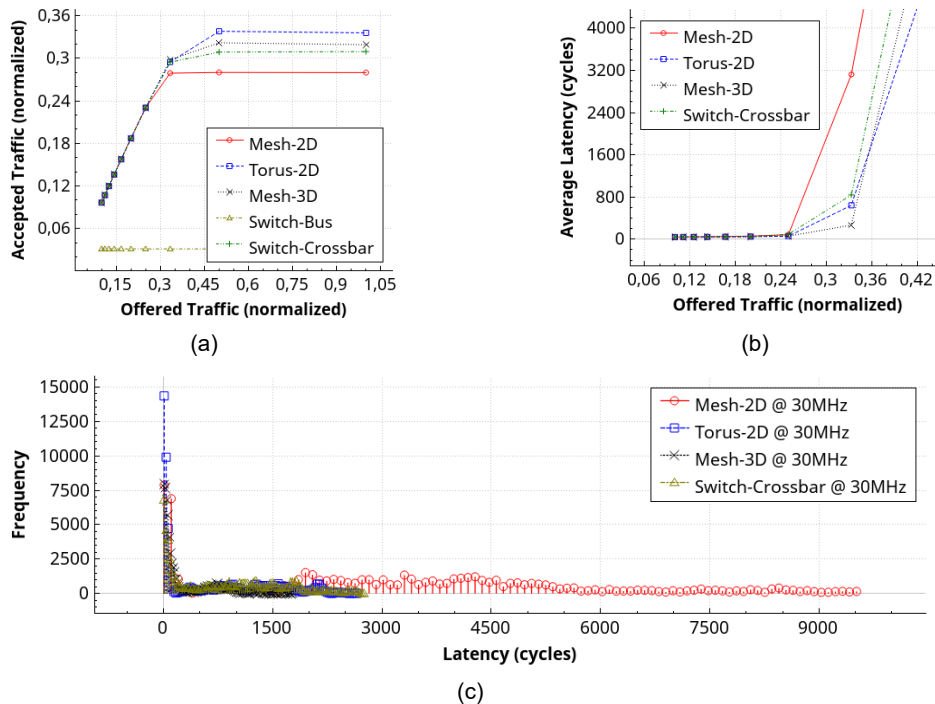


Figura 1. Tráfego Uniforme – topologias de 16 nodos: (a) Tráfego aceito x Tráfego Oferecido; (b) Latência média x Tráfego Oferecido; (c) *Jitter* próximo à saturação

Também foi possível perceber, que a escalabilidade da Malha 3D foi superior às demais configurações de NoC ao aumentar o tamanho do sistema para 64 nodos, sendo inferior apenas ao Crossbar no contexto de todos os cenários de tráfego. Na Figura 2, é possível visualizar essa situação no tráfego Perfect Shuffle. Nas redes com 16 nodos, tanto a latência (Figura 2.a) como a vazão (Figura 2.c) são semelhantes para as topologias Crossbar, Malha 3D e Torus 2D. Já nas redes com 64 nodos, nota-se que a Malha 3D é a NoC com menor latência e que suporta uma maior carga de tráfego até a rede saturar (Figura 2.b). O mesmo vale para o tráfego aceito (Figura 2.d).

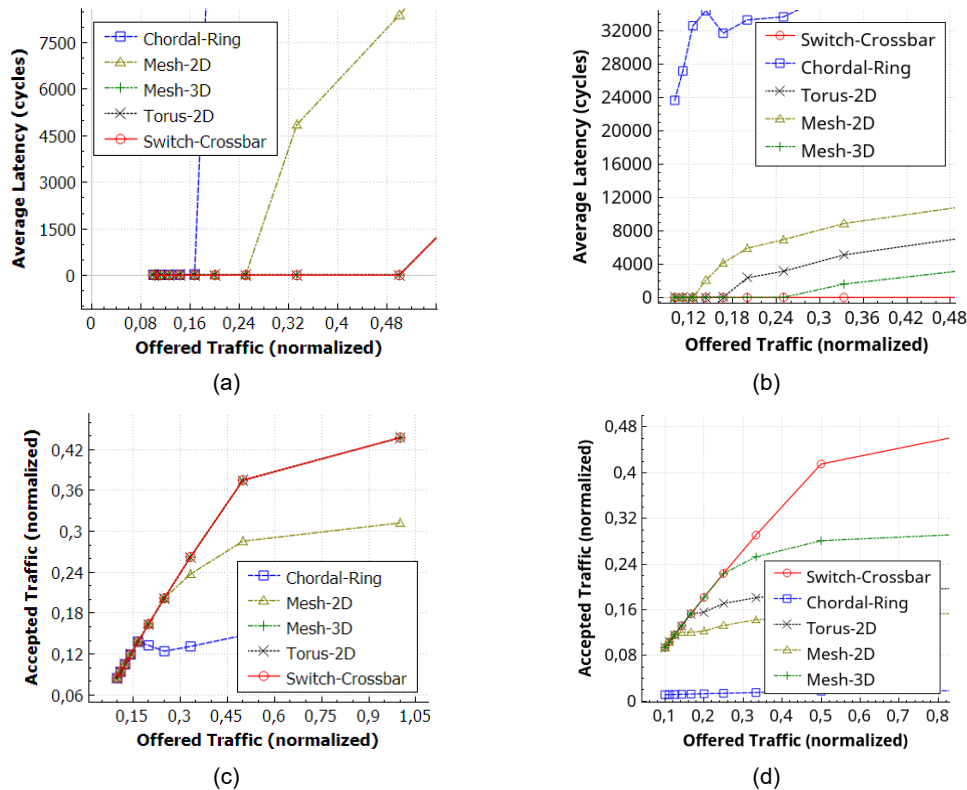


Figura 2. Tráfego Perfect Shuffle – escalabilidade: (a) Latência Média com 16 nodos; (b) Latência Média com 64 nodos; (c) Vazão com 16 nodos; e (d) Vazão com 64 nodos

Os roteamentos foram avaliados na Malha 2D, pois é a topologia com mais algoritmos implementados. Os roteamentos experimentados foram: o XY, o West-first, o Negative-first, o North-last e o Odd-even. O arranjo adotado foi o 4x4, sendo que os demais atributos da rede foram mantidos os mesmos dos experimentos das topologias.

Destaca-se que o algoritmo determinístico XY é o que melhor distribui os fluxos em cenários de alta distribuição de carga pela rede e suas bissecções, tais como o Uniforme e o Complemento. Enquanto os algoritmos adaptativos, melhor distribuem a carga em cenários de tráfego com maior localidade temporal (Bit-reversal, Perfect Shuffle, Butterfly e Transposto). Em cenários globalmente distribuídos, as rotas alternativas podem fazer com que um pacote encontre um caminho mais congestionado nos saltos posteriores. Já em cenários localmente distribuídos, as rotas alternativas tendem a estar disponíveis, o que faz com que mais pacotes enfrentem latências menores.

Ao analisar os algoritmos XY e Odd-even nos cenários de tráfego experimentados, torna-se perceptível o que foi mencionado, tanto na latência média (Figura 3) quanto no tráfego aceito (Figura 4).

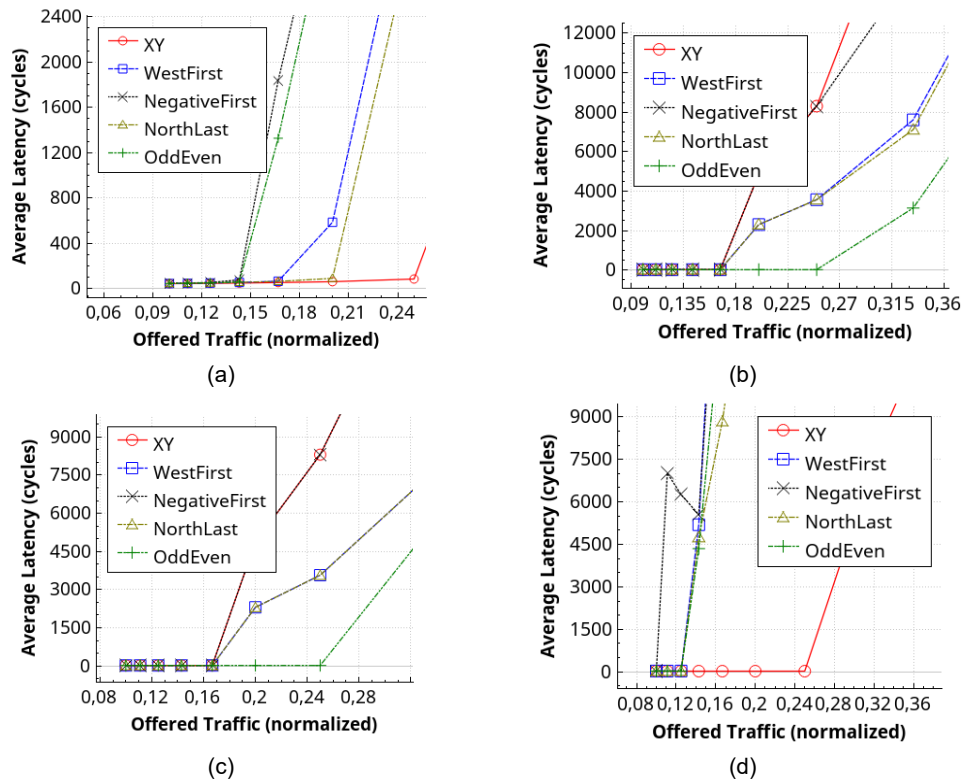


Figura 3. Latência média dos roteamentos na Malha 2D 4x4 sob diferentes cenários de tráfego: (a) Uniforme; (b) Bit-reversal; (c) Transposto; e (d) Complemento

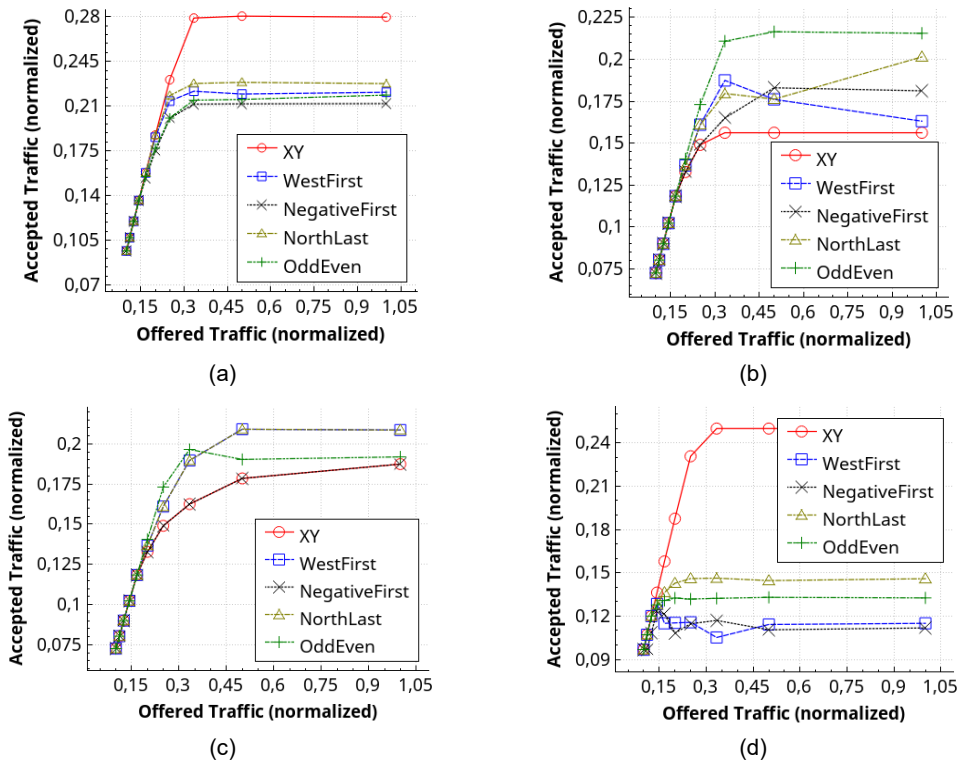


Figura 4. Tráfego aceito dos roteamentos na Malha 2D 4x4 sob diferentes cenários de tráfego: (a) Uniforme; (b) Bit-reversal; (c) Transposto; e (d) Complemento

Dos árbitros, as políticas de priorização Estática, Rotativa, Randômica e Round-robin foram avaliadas. A partir dos resultados obtidos, é possível confirmar que uma política sem equidade, caso do estático, degrada o desempenho das comunicações (Figura 5), conforme já discutido na literatura. Além disso, a entrega dos pacotes pode ser comprometida dependendo dos requisitos de comunicação. Os árbitros dinâmicos apresentaram similaridades nos seus desempenhos e o árbitro estático apresentou instabilidade na rede ao degradar o desempenho após a saturação.

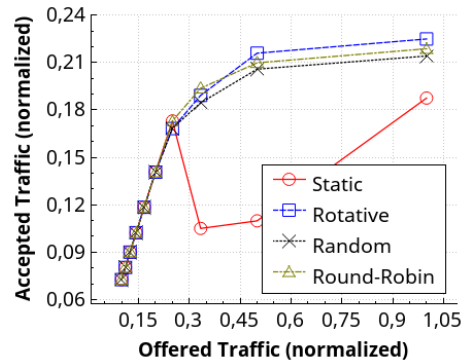


Figura 5. Tráfego aceito para diferentes tipos de árbitro

Por fim, um experimento com diferentes profundidades de *buffer* é discutido. Memórias maiores nos roteadores acarretam em melhor desempenho. Porém, ressalva-se que o ganho é significativo até uma determinada profundidade, a partir da qual o ganho obtido não justifica o custo adicional. A Tabela 6 exibe uma forma alternativa de análise dos dados de desempenho que mostra a relação de ganho de uma configuração sobre outra. As linhas em destaque identificam o melhor resultado obtido em cada métrica. Para a latência média e o *jitter*, apresenta-se quantas vezes a configuração da linha é melhor em relação à da coluna. Para o tráfego aceito, é mostrada a capacidade adicional de tráfego da configuração da linha em relação à da coluna.

Tabela 6. Desempenho para diferentes profundidades de *buffer* sob tráfego Uniforme

| Profundidade dos Buffers | IN4 | IN8 | IN16 | IN32 | IN16 OUT16 | Métrica |
|--------------------------|-------|--------|--------|------|---------------|-----------------|
| IN4 | | -x- | -x- | -x- | -x- | Latência Média* |
| IN8 | 5,83 | | -x- | -x- | -x- | |
| IN16 | 7,29 | 1,25 | | -x- | -x- | |
| IN32 | 7,75 | 1,32 | 1,06 | | 1,03 | |
| IN16 OUT16 | 7,45 | 1,27 | 1,02 | -x- | | |
| IN4 | | -x- | -x- | -x- | -x- | Tráfego Aceito |
| IN8 | 4,74 | | -x- | -x- | -x- | |
| IN16 | 6,53 | 1,79 | | -x- | -x- | |
| IN32 | 6,72 | 1,98 | 0,19 | | 0,05 | |
| IN16 OUT16 | 6,67 | 1,93 | 0,14 | -x- | | |
| IN4 | | -x- | -x- | -x- | -x- | Jitter* |
| IN8 | 12,67 | | -x- | -x- | -x- | |
| IN16 | 16,51 | 1,3027 | | -x- | -x- | |
| IN32 | 17,46 | 1,3777 | 1,0576 | | 1,0036 | |
| IN16 OUT16 | 17,39 | 1,3728 | 1,0538 | -x- | | |

OBS: IN: Buffer de entrada; OUT: Buffer de saída; INX/OUTX: Buffer com X posições; * Operando a 30 MHz

5. Conclusões

Este artigo apresentou um estudo realizado com o objetivo de comparar diferentes arquiteturas de comunicação intrachip com base em dados quantitativos extraídos por simulação. A partir dos experimentos realizados, é possível afirmar que não há uma única arquitetura de comunicação que possua melhor desempenho em latência e vazão para todos os cenários de comunicação, nem mesmo o Crossbar. Por outro lado, foi possível identificar quais configurações apresentam melhor desempenho em cada cenário de tráfego experimentado. Além disso, alternativas de implementação dos atributos arquiteturais foram avaliadas e seus desempenhos foram relacionados para indicar o quão uma implementação é superior/inferior a outra.

Como trabalhos futuros, pretende-se implementar ferramentas que automatizem a exploração do espaço de projeto e identifiquem a melhor configuração de rede para um determinado modelo de tráfego, considerando suas métricas de desempenho e custo. Também pretende-se disponibilizar as ferramentas e bibliotecas já desenvolvidas para uso em ensino e pesquisa em outras instituições.

Agradecimentos

Os autores gostariam de agradecer o apoio recebido da Capes – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

Referências

- Accelera (2017). “SystemC”, <http://www.accellera.org/downloads/standards/systemc>.
- Andriahantenaina, A., Charlery, H., Greiner, A., Mortiez, L., Zeferino, C. A. (2003) “SPIN: a scalable, packet switched on-chip micro-network”, In *Design Automation and Test on Europe*, IEEE CS, Los Alamitos, p. 70-73.
- Dally, W., Towles, B. (2004). Principles and Practices of Interconnection Networks. San Francisco: Morgan Kaufmann.
- Du, G., He, J., Song, Y., Zhang, D., Wu, H. (2013). “Comparison of NoC routing algorithms based on packet-circuit switching”. In *3rd Int. Conf. on Information Science and Technology*, IEEE, Yangzhou, p. 707-710.
- Ghidini, Y., Weber, T., Moreno, E., Quadros, I., Fagundes, R., Marcon, C. (2012). “Topological impact on latency and throughput: 2D versus 3D NoC comparison”. In *25th Symp. on Integrated Circuits and Systems Design*, IEEE, Brasília, p 1-6.
- Hao, P., Qii, H., Jiaqin, D., Pan, P. (2011). “Comparison of 2D MESH routing algorithm in NOC”. In *9th Int. Conf. on ASIC*, IEEE, Xiamen, p. 791-795.
- Harbin, J., Indrusiak, L. S. (2016). “Comparative performance evaluation of latency and link dynamic power consumption modelling algorithms in wormhole switching networks on chip”. *Journal of Systems Architecture*, New York, v. 63, p. 33-47.
- Jaina, A., Kumarb, A., Sharmac, S. (2015). “Comparative Design and Analysis of Mesh, Torus and Ring NoC”. In *Int. Conf. on Computer, Communication and Convergence*, Procedia Computer Science, v. 48, p. 330-337.
- Jetly, Krunal (2013). Experimental Comparison of Store-and-Forward and Wormhole NoC Routers for FPGA's. MSc Thesis, Dept. Electrical and Computer Engineering University of Windsor, Windsor.

- Ju, X., Yang, L. (2012). "Performance analysis and comparison of 2×4 Network on Chip topology". *Journal of Microprocessors and Microsystems*, New York, v. 36, n. 6, p. 505-509.
- Kalimuthu, A., Karthikeyan, M. (2012). "Comparative performance evaluation of power and area Network on Chip (NoC) architectures". In *Int. Conf. on Computing Intelligence and Computing Research*, IEEE, Coimbatore, p. 1-4.
- Manna, K., Chattopadhyaya, S., Sengupta, I (2012). "An efficient routing technique for mesh-of-tree-based NoC and its performance comparison". *Int. Journal of High Performance Systems Architecture*, Geneva, v. 4, n. 1, p. 25-37.
- Ogras, U. Y., Marculescu, R. (2013). *Modeling, Analysis and Optimization of Network-on-Chip Communication Architectures*, Springer, v. 184.
- Pandey, D., Gupta, K. (2012). "Comparison between Mesh and Custom Topologies of Network-on-Chip Architectures". *Int. Journal of Scientific Engineering and Technology*, Gulmohar, v. 1, n. 5, p. 243-247.
- Parandkar, P., Dalal, J. K., Katiyal, S. (2012). "Performance Comparison of XY, OE and DY Ad Routing Algorithm by Load Variation Analysis of 2-Dimensional Mesh Topology Based Network-on-Chip". *BVICAM Int. Journal of Information Technology*, New Delhi, v. 4, n. 1, p. 391-396.
- Radfar, F., Zabihi, M., Sarvari, R. (2014). "Comparison between optimal interconnection network in different 2D and 3D NoC structures". In *27th Int. System-On-Chip Conf.*, IEEE, Las Vegas, p. 171-176.
- Romanov, O., Lysenko, O. (2012). "The comparative analysis of the efficiency of regular and pseudo-optimal topologies of Networks-on-Chip based on Netmaker". In *Mediterranean Conf. on Embedded Computing*, IEEE, Bar, p. 13-16.
- Sadawarte, Y. A., Gaikwad, M. A., Patrikar, R. M. (2011). "Comparative study of switching techniques for network-on-chip architecture". In: *Int. Conf. on Communication, Computing and Security*, ACM, New York, p. 243-246.
- Silva, E. A. (2014). RedScarf: ambiente para avaliação de desempenho de Rede-em-Chip. BSCS Work, Universidade do Vale do Itajaí, Itajaí.
- Silva, E. A. (2017). Análise Comparativa do Desempenho de Arquiteturas de Redes-em-Chip baseada em Simulação. M.Sc. Thesis, Universidade do Vale do Itajaí, Itajaí
- Sllame, A. M., Abdelkader, A. H. (2014). "A comparative study between fat tree and mesh network-on-chip interconnection architectures". In *14th Middle Eastern Simulation and Modeling Multiconference*, Eurosis, Muscat, p. 31-37.
- Tedesco, L., Mello, A., Garibotti, D., Calazans, N., Moraes, F. (2005). "Traffic generation and performance evaluation for mesh-based NoCs". In *18th Symp. On Integrated Circuits and Systems Design*. Florianópolis, p. 184-189.
- The Qt Company (2017), "Qt", <https://qt.io>, August.
- Wang, P., Ma, S., Lu, H., Wang, Z. (2014). "A comprehensive comparison between virtual cut-through and wormhole routers for cache coherent network on-chips". *Journal IEICE Electronic Express*, Tokyo, v. 11, n. 14, p. 1-12.
- Yin, A., Chen, N., Liljeberg, P., Tenhunen, H. (2012). "Comparison of mesh and honeycomb network-on-chip architectures". In *Conf. on Industrial Electronics and Applications*, IEEE, Singapura, p. 1716-1720.