

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL**  
**FACULDADE DE INFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**MAPEAMENTO E ADAPTAÇÃO  
DE ROTAS DE COMUNICAÇÃO  
EM REDES EM CHIP**

**EDSON IFARRAGUIRRE MORENO**

TESE SUBMETIDA COMO REQUISITO  
PARCIAL PARA A OBTENÇÃO DO  
GRAU DE DOUTOR EM CIÊNCIA DA  
COMPUTAÇÃO

**PROF. DR. NEY LAERT VILAR CALAZANS**  
ORIENTADOR

PORTO ALEGRE, BRASIL  
JANEIRO 2010

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



## **Dados Internacionais de Catalogação na Publicação (CIP)**

M843m    Moreno, Edson Ifarraguirre  
             Mapeamento e adaptação de rotas de comunicação em  
             redes em chip / Edson Ifarraguirre Moreno – Porto Alegre, 2010.  
             177 f.

             Tese (Doutorado) – Fac. de Informática, PUCRS.  
             Orientador: Prof. Dr. Ney Laert Vilar Calazans.

             1. Informática. 2. Arquitetura de Redes. 3. Roteamento –  
             Redes de Computadores. 4. Multiprocessadores. 5. Redes de  
             Comunicação de Computadores. I. Calazans, Ney Laert Vilar.  
             II. Título.

CDD 004.6

**Ficha Catalográfica elaborada pelo  
Setor de Tratamento da Informação da BC-PUCRS**





Pontifícia Universidade Católica do Rio Grande do Sul  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO

Tese intitulada "Mapeamento e Adaptação de Rotas de Comunicação em Redes em Chip", apresentada por Edson Ifarraguirre Moreno, como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação, Sistemas Embarcados e Sistemas Digitais, aprovada em 27/01/10 pela Comissão Examinadora:

  
Prof. Dr. Ney Laert Vilar Calazans -  
Orientador

PPGCC/PUCRS

  
Prof. Dr. Fernando Ghem Moraes -

PPGCC/PUCRS

  
Prof. Dr. César Augusto Missio Marcon -

FACIN/PUCRS

  
Prof. Dr. Ivan Saraiva Silva -

UFRN

Homologada em 17/09/2010, conforme Ata No. 18/10 pela Comissão Coordenadora.



Prof. Dr. Fernando Gehm Moraes  
Coordenador.

**PUCRS**

### Campus Central

Av. Ipiranga, 6681 - P. 32 - sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: [ppgcc@pucrs.br](mailto:ppgcc@pucrs.br)

[www.pucrs.br/facin/pos](http://www.pucrs.br/facin/pos)



## AGRADECIMENTOS

Este é um momento tão difícil quanto prazeroso. Listar as pessoas que contribuíram no desenvolvimento de minha tese de doutorado torna-se difícil por ter de priorizar a ordem em que aparecerão neste impresso. Prazeroso pelo fato de poder lembrar com carinho dos suportes carinhosos e dos choques de realidade que por muitas vezes também foram necessários.

Sílvia Casonato! Esta mulher é fantástica. Lembro ainda hoje o dia em que disse a ela que iria trocar o ramo que havia trilhado durante minha graduação para encarar a pós-graduação. Sem titubear ela disse que me daria a maior força. Não imaginávamos pelo que iríamos passar e talvez ela não imagine o quanto foi forte. Sete longos anos desde a minha decisão até o final do doutorado. E ela me agüentou, quando nem eu me agüentava mais! Não houve um só dia em que ela deixou de apoiar minha decisão. Nos momentos em que quase desisti, e não foi um só, lá estava ela me incentivando, acreditando em mim, me dando suporte. Doutora, eu te amo. Tu és a mulher da minha vida.

Aos meus pais, seu Moreno e dona Ilsa. Esses dois são o exemplo de força, fé, respeito e dedicação que busco seguir a cada dia. Muito obrigado pelo que são e pela oportunidade que temos de conviver. Às minhas irmãs, Elis e Estela, e ao meu sobrinho/afilhado/irmão/amigo, Arthur, que tanta aleatoriedade de sentimentos trazem a minha vida, mas que me ensinam todo o dia a tomar partido da família, independente do que ocorra.

Aos meus amigos “gapheanos”, colegas de laboratório, de eventuais cervejinhas e de, cada vez mais raros, churrascos. Julian Pontes, Leonel Tedesco, Rafael Soares, amigos sempre dispostos a discussões sobre assuntos técnicos, profissionais, pessoais e futebolísticos. Ao Luciano Ost, figura ímpar com quem tive o prazer de conviver durante todos estes anos, trazendo alegria nos momentos mais oportunos e nos menos também. Ao Ewerson Carvalho, meu irmão, um cara indescritível com quem tive a felicidade de conviver. Este cara é tão especial que não há palavras que eu possa usar para descrevê-lo e que me deixem confortável de tê-lo descrito como ele realmente é. Ewerson, tua felicidade, teu sucesso e nossa amizade são certezas que trago comigo. Ao Alexandre Amory, um cara com quem construí uma amizade muito grande e que admiro muito.

Aos amigos que conheci e convivi durante meu estágio na França. Leonardo Brenner e Marina Sovaldi, obrigado pela acolhida, pela amizade e pelas *randonnées*. Ao Lucas Schnorr, meu companheiro de caminhadas nas montanhas. Um especial agradecimento ao Ricardo Czekster, meu amigo desde o tempo de colégio, e a Thaís Webber. *To Katalin Popovici and Marius Bonacciu, thank you two for the friendship and discussion times. Kititonton and Little, you two are great!*

Aos meus grandes amigos e mentores César Marcon, Fernando Moraes, Ney Calazans e Fabiano Hessel. Este tempo de convívio me deram a oportunidade de conhecê-los bem. Busquei, e ainda busco, muitos conselhos com eles. Muito devo de minha formação profissional e pessoal a todos vocês. Marcon muito obrigado por auxiliar em momentos tão difíceis. Moraes muito obrigado por me dar a tranquilidade que eu precisava para concluir meus trabalhos. Ney muito obrigado pelas orientações e formação. Hessel, muito obrigado pelas conversas e pelos momentos de descontração.

Finalmente, agradeço o suporte financeiro para a realização do doutorado no país, advindo do CNPq na forma da bolsa de fomento do Programa Nacional de Microeletrônica (PNM), e à CAPES pela oportunidade de financiamento para a realização do doutorado sanduíche na França. Agradeço ainda à PUCRS pela infraestrutura disponibilizada e pelos profissionais envolvidos para a construção de uma universidade forte. Agradeço em especial ao pessoal da secretaria de pós-graduação da ciência da computação Thiago Lingener, Régis Escobal, Talita Utteich e à Sandra Rosa. Muito obrigado a todos.





# MAPEAMENTO E ADAPTAÇÃO DE ROTAS DE COMUNICAÇÃO EM REDES EM CHIP

## RESUMO

A constante evolução das necessidades de mercado exige que sejam disponibilizados sistemas computacionais com poder de processamento cada vez maior. O aumento da frequência de operação e o paralelismo de instruções em microprocessadores não são mais suficientes para garantir a melhora do desempenho destes sistemas. Uma forma de garantir tal aumento no poder de processamento é o desenvolvimento de sistemas multiprocessados em um único chip (MPSoC), o que permite dividir os custos de computação de aplicações pelos elementos de processamento que o formam. É tendência que o número de elementos de processamento que compõe um MPSoC cresça com o avanço em direção a tecnologias submicrônicas. Para interconectar tais elementos de processamento são necessárias infraestruturas de comunicação mais eficientes do ponto de vista de características elétricas, facilidade de adoção em projetos e desempenho. Redes em chip (do inglês, *Networks on Chip* ou NoCs) são vistas como uma tendência neste processo. Assim como o aumento do desempenho da computação prevê-se também o aumento do desempenho da comunicação entre os elementos de processamento. Obviamente, NoCs podem sofrer com fenômenos de congestionamento, que degradam a qualidade das comunicações devido, por exemplo, ao aumento da latência de entrega de mensagens. O uso de algoritmos adaptativos em NoCs permite reduzir o congestionamento, mas decisões de adaptação são normalmente baseadas no estado instantâneo da rede e apenas no uso de informação local. O problema deste tipo de abordagem é a imprevisibilidade da latência de entrega de pacotes, visto que a rota a ser utilizada por um pacote depende do estado da rede e da regra adotada pelo algoritmo de roteamento. Adicionalmente, o desvio de uma rota considerada congestionada pode levar a outras com concentração de tráfego ainda maior. O presente trabalho propõe duas infraestruturas de comunicação que permitem maior previsibilidade, sendo assim úteis para melhor atender requisitos de comunicação de aplicações. Ambas as infraestruturas propostas empregam roteamento na origem. A primeira, denominada NoC Hermes-SR explora o mapeamento de rotas de comunicação realizado em tempo de projeto. Resultados iniciais mostram um ganho desta infraestrutura de comunicação quando comparada à NoC Hermes com roteamento determinístico XY, uma NoC sem mecanismos para reduzir congestionamentos. Na segunda infraestrutura de comunicação, chamada MoNoC (de NoC Monitorada), exploram-se recursos que contribuem para permitir adaptação de rotas, tais como interfaces de rede, monitores e sondas de rede. Resultados capturados para tal infraestrutura apresentaram reduções significativas de latência de aplicação. Em ambos os casos, a adoção de algoritmos de roteamento adaptativos quando utilizados como base para a definição de rotas permite contornar caminhos congestionados na rede aumentando a previsibilidade de latência de entrega de pacotes.

Palavras chave: *redes em chip, roteamento na origem, balanceamento de carga, controle de fluxo*



# MAPPING AND ADAPTATION OF COMMUNICATION ROUTES IN ON CHIP NETWORKS

## ABSTRACT

The constant evolution of market needs requires the availability of computing systems with ever-growing performance. Increases in operating frequencies and instruction level parallelism in microprocessors are not alone sufficient anymore to guarantee performance scaling for such systems. A way to achieve scaling performance has been the adoption of Multiprocessor Systems on a Chip (MPSoCs), which allow distributing application computation costs along a set of processing elements in the MPSoC. The increase in the number of MPSoC processing elements as technology advances into the deep submicron domain is a clear trend. To interconnect such elements it is necessary to employ more efficient communication infrastructures in what concerns electrical characteristics, facility of adoption by designers and performance. Networks on Chip (NoCs) or on chip networks are a clear trend in this sense. In the same way computation performance is expected to increase, so is the performance of communication among processing elements in future MPSoCs. Obviously, NoCs may still suffer from congestion, which degrades the communication quality due, for example, to the increase of latency while delivering messages. The use of adaptive routing algorithms in NoCs allows altogether to solve or at least to alleviate congestion scenarios, but adaptation decisions usually employ only the instantaneous state of the communication network and are based on local information. The problem with this kind of approach is the unpredictability of latency in delivering network packets since the path followed by each packet depends on the state of the network and on the rules adopted by the routing algorithm as well. Additionally, deviations from a route considered congested can take packets to regions with even higher traffic. This thesis proposes two communication infrastructures that allow an improved degree of predictability and are thus more useful to fulfill application communication requirements. Both infrastructures employ source routing strategies. The first, called Hermes-SR NoC, explores the mapping of communication routes at design time. Initial results demonstrate this infrastructure displays gains when compared to the Hermes NoC, a network without congestion solving mechanisms, which uses an XY deterministic routing algorithm. The second infrastructure, called MoNoC (Monitored NoC), explores resources that enable dynamic route adaptation to take place. These resources include special network interfaces, monitors e network probes. Experimental results achieved with MoNoC showed significant reductions for application latency. In both cases, the adoption of adaptive routing algorithms as a base to compute routes enables to turn traffic away from congestion points in the network, which naturally increases latency and packet delivery predictability.

Keywords: *networks on chip, source routing, load balancing, flow control*



## LISTA DE FIGURAS

Figura 1.1 – Evolução da densidade de transistores e do poder de processamento [INT09].	23
Figura 1.2 – Exemplos da evolução no número de PEs em CMPs.	26
Figura 2.1 – Exemplos de topologias regulares e de NoCs que empregam estas.	35
Figura 2.2 – Exemplo da infraestrutura Octagon de dimensões superiores a topologia base.	36
Figura 2.3 – Infraestruturas de comunicação híbridas propostas em [DUM04].	37
Figura 2.4 – Infraestrutura de comunicação híbrida proposta por Richardson et al. [RIC06].	37
Figura 2.5 – Infraestrutura de comunicação QNoC resultante do processo de mapeamento de núcleos IP e da adaptação de uma NoC de topologia malha 2D.	38
Figura 2.6 – Associação de três fluxos compartilhando um canal. Em (a) mostra-se a alocação quando os fluxos têm mesma taxa de requisição, enquanto que em (b) mostra-se uma alocação max-min justa para taxas com requisitos diferentes.	51
Figura 3.1 – Estrutura do pacote na NoC Hermes-SR. Cabeçalho contendo a rota e finalizado pelo <i>flit</i> terminador. O campo Tamanho identifica a quantidade de <i>flits</i> no campo Carga útil ou corpo do pacote.	60
Figura 3.2 – Estrutura interna de roteadores da NoC Hermes-SR. Não há um módulo centralizado para arbitragem e roteamento como na rede Hermes. Esta função é realizada de forma distribuída nas diferentes portas de saída. Em (a) apresenta-se a estrutura da porta de entrada com seus sinais para recepção de dados e uma fila de armazenamento. Em (b) apresenta-se a estrutura da porta de saída, ligada a todas portas de entrada, à fila de armazenamento de requisições e aos sinais de transmissão.	62
Figura 3.3 – Exemplo de processo de transmissão de pacote na NoC Hermes-SR, ilustrando o tratamento dado ao campo rota.	63
Figura 3.4 – Diagrama de blocos de roteadores com (a) o esquema de arbitragem e roteamento centralizados adotado na NoC Hermes e (b) a arbitragem e roteamento descentralizados empregados na NoC Hermes-SR.	64
Figura 3.5 – Diagrama de curvas para o algoritmo de roteamento XY puro. Em (a) e (b) nota-se a proibição dos quatro diferentes tipos de curvas em direção ao eixo X, após iniciado deslocamento do pacote sobre o eixo Y.	67
Figura 3.6 – Diagrama de curvas para o algoritmo de roteamento <i>negative first</i> . Proíbe-se curvas para direções negativas se uma direção anteriormente tomada seguiu um sentido positivo. Em (a) proíbe-se curvas para OESTE (direção negativa) se o pacote está subindo e em (b) proíbe-se curvas para SUL (direção negativa) se o pacote está indo para a direita. Note-se que não há restrições para iniciar o caminho de um pacote seguindo para OESTE ou para o SUL, e mesmo relizar curvas entre estas direções.	68
Figura 3.7 – Situações de rota única no algoritmo <i>nfm</i> . Em (a), o destino possui o valor de coordenada x menor que o destino. Em (b), o destino possui o valor de coordenada y menor que o destino. Em (c), o destino encontra-se sobre a mesma linha (c.1) e (c.2) ou sobre a mesma coluna (c.3) e (c.4).	69
Figura 3.8 – Caminhos válidos entre origem e destino com distância 2 em X e em Y. Em (a) mostra-se o posicionamento totalmente positivo do destino em relação a origem. Em (b) mostra-se o posicionamento totalmente negativo do destino em relação a origem. Em ambos os casos, nota-se o respeito aos limites impostos pelo QE.	70
Figura 3.9 – Crescimento do número de rotas possíveis para a o roteador localizado no canto inferior esquerdo tendo como destino qualquer dos demais roteadores quando é assumido os algoritmo de roteamento mínimo explorados no presente trabalho.	71
Figura 3.10 – Alternativas de rotas quando inicialmente se usa algoritmo <i>nfm</i> . Em (a) apresenta-se a quantidade de rotas a partir da origem para chegar a algum roteador alcançável, ao afastar-se do destino. Em (b), mostra-se a quantidade de rotas possíveis quando, ao alcançar um roteador mais afastado, assume-se direção positiva.	72
Figura 3.11 – Diagrama de curvas para o algoritmo de roteamento <i>West first</i> . Em (a) e em (b), nota-se que curvas para OESTE são proibidas após caminhamentos em sentido diferente de OESTE. Note-se que sempre se pode ir inicialmente para OESTE.	73
Figura 3.12 – Disposição relativa dos destinos em relação à origem da comunicação em redes malha 2D.	74
Figura 3.13 – Características do roteamento com o algoritmo <i>West first</i> . Em (a), quando o destino encontra-se a esquerda da origem, o roteamento inicial para OESTE é obrigatório até que a coluna do destino seja alcançada, podendo continuar para OESTE caso o algoritmo seja não mínimo. Em (b), quando o destino encontra-se à direita da origem, o caminhamento nunca pode ultrapassar a coluna do destino.	74
Figura 3.14 – Situações de rota única para o algoritmo <i>wfm</i> . Em (a) e (b) os destinos encontram-se respectivamente em $QdX_{Pos}Y_{Neg}$ e $QdX_{Neg}Y_{Neg}$ em relação à origem. Em (c) os destinos encontram-se sobre o mesmo eixo X ou mesmo eixo Y em relação à origem.	75

Figura 3.15 – Opções de rotas entre um par origem/destinos com distância 2 em X e 2 em Y ao adotar o algoritmo <i>wfm</i> . Em (a) o destino encontra-se no $QdX_{Pos}Y_{Pos}$ em relação à origem. Em (b) o destino encontra-se em $QdX_{Pos}Y_{Neg}$ em relação à origem. Em ambos os casos, há respeito a limites colocados de QE. ....	75
Figura 3.16 – Áreas de roteamento para o algoritmo <i>wfm</i> . Em (a), o destino encontra-se no $QdX_{Neg}Y_{Neg}$ em relação à origem, o que reduz a área de roteamento livre. Em (b), o destino encontra-se no $QdX_{Pos}Y_{Neg}$ em relação à origem, aumentando a área de roteamento livre. ....	76
Figura 3.17 – Diagrama de curvas para o algoritmo de roteamento <i>North last</i> . Tanto em (a) quanto em (b) é ilustrado que curvas em todos os sentidos são válidas até que o sentido NORTE seja tomado. A partir daí nenhuma curva é mais permitida. ....	76
Figura 3.18 – Características do algoritmo de roteamento <i>North last</i> . Em (a), quando o destino encontra-se em $QdX_{Neg}Y_{Neg}$ , ou em $QdX_{Pos}Y_{Neg}$ , ou em $MmoYNeg$ , o roteamento para a direção NORTE pode não ocorrer, no caso do algoritmo mínimo. Em (b), a exceção da linha onde se encontra à origem, a área entre a origem e o destino deve ser coberta apenas por roteamentos no sentido NORTE. ....	77
Figura 3.19 – Situações de rota única para o algoritmo <i>nlnm</i> . Em (a) e (b) os destinos encontram-se em relação à origem respectivamente em $QdX_{Neg}Y_{Pos}$ e em $QdX_{Pos}Y_{Pos}$ . Em (c), (c.1) e (c.2) os destinos encontram-se em $MmoX$ , enquanto em (c.3) e (c.4) encontram-se em $MmoY$ em relação à origem. ....	77
Figura 3.20 – Rotas possíveis entre origens e destinos com distância 2 em X e 2 em Y e adotando o algoritmo de roteamento <i>nlnm</i> . Em (a) o destino encontra-se em $QdX_{Neg}Y_{Neg}$ em relação à origem e em (b) o destino encontra-se em $QdX_{Pos}Y_{Neg}$ em relação à origem. ....	78
Figura 3.21 – Áreas de roteamento livre, exclusivo para NORTE e proibida no algoritmo <i>nlnm</i> . Em (a), a localização do destino em relação à origem aumenta a área de roteamento livre e em (b), a localização do destino em relação à origem reduz a área de roteamento. ....	78
Figura 3.22 – Situação de ocorrência de ponto quente e consequente congestionamento. Ponto quente aqui ocorre devido a sobrecarga de fluxos de pacotes em canais, superando suas capacidades de vazão. ....	79
Figura 3.23 – Distribuições alternativas para os fluxos de comunicação entre os pares origem e destino visando eliminar a ocorrência de pontos quentes e consequentes congestionamentos. Em (a) a escolha de boas rotas, enquanto em (b) obtém-se uma boa distribuição, usando caminhos mais longos na rede. ....	80
Figura 3.24 .....	81
Figura 3.25 Exemplo de um grafo de aplicação hipotético, com taxas em Mbps. ....	81
Figura 4.1 – Métricas distintas de latência. ....	89
Figura 4.2 – Latência média obtida para as NoCs Hermes (a) e Hermes-SR (b) variando-se o tamanho da fila de armazenamento. ....	90
Figura 4.3 – Estimativas de distribuição de carga para o cenário de tráfego todos para todos. ....	94
Figura 4.4 – Estimativa de ocupação de canais para o cenário de tráfego todos para todos. ....	95
Figura 4.5 – Comparação de latências médias para os algoritmos não mínimos (a) e mínimos (b). ....	96
Figura 4.6 – Intervalo de latência e percentual de entrega no cenário de tráfego todos para todos. ....	97
Figura 4.7 – Cenários de tráfego com pares comunicantes nas bordas explorando sobreposição de rotas. ....	98
Figura 4.8 – Estimativa de distribuição de carga para o cenário de tráfego sobreposição de rotas. ....	98
Figura 4.9 – Estimativa de ocupação de canais para o cenário de tráfego sobreposição de rotas. ....	99
Figura 4.10 – Intervalo de latências obtidos para o cenário de tráfego sobreposição de rotas. ....	100
Figura 4.11 – Cenários de tráfego com apenas dois destinos possíveis ( <i>hotspot</i> ). ....	101
Figura 4.12 – Estimativa de distribuição de carga para o cenário de tráfego <i>hotspot</i> . ....	101
Figura 4.13 – Estimativa de ocupação de canais para o cenário de tráfego <i>hotspot</i> . ....	102
Figura 4.14 – Distribuição das latências na simulação do cenário um utilizando rotas calculadas com os diferentes algoritmos de roteamento determinístico e adaptativos. ....	104
Figura 4.15 – Comparando Latências para decisões em tempo de projeto (TProj) e de execução (TExec). ....	106
Figura 5.1 – Rotas resultantes do emprego dos algoritmos complementares. Em (a) emprega-se <i>East first</i> . Em (b) emprega-se <i>West first</i> . ....	113
Figura 5.2 – Infraestrutura de comunicação MoNoC resultante da integração dos mecanismos de suporte a adaptação de rotas. Interfaces de rede são posicionadas nas fronteiras para interconexão com os PEs, enquanto sondas de rede são posicionados internamente à NoC, nas portas de saída de cada roteador, onde observam a utilização dos canais. ....	114
Figura 5.3 – Roteador da MoNoC empregando canais virtuais para a diferenciação de serviços. ....	115
Figura 5.4 – Exemplo de ganho de prioridade para transmissão de pacotes a partir dos canais virtuais da MoNoC. Os pacotes (A, B, C, D) são compostos de 4 <i>flits</i> . Os pacotes A, B, C e D chegam aos distintos canais virtuais nos instantes $t_1$ , $t_2$ , $t_2$ e $t_7$ respectivamente. ....	116
Figura 5.5 – Comportamento projetado para o ganho de prioridade sob situações de bloqueio na transmissão de pacotes a partir dos canais virtuais da MoNoC. Os pacotes (A, B, C, D) são compostos de 4 <i>flits</i> . Os pacotes A, B, C e D chegam aos distintos canais virtuais nos instantes $t_1$ , $t_2$ , $t_2$ e $t_7$ respectivamente. ....	118
Figura 5.6 – Localização de uma sonda de rede em um roteador. Em destaque uma sonda de rede da	

porta West de um roteador central.....	119
Figura 5.7 Interface da sonda de rede. ....	119
Figura 5.8 - Sinais da interface de observação analisados pela sonda de rede da MoNoC. ....	120
Figura 5.9 - Estrutura de um pacote para operação com as sondas de rede. ....	122
Figura 5.10- Interfaces da NI para transmissão entre o Núcleo IP e a MoNoC. ....	123
Figura 5.11 – Exemplo de transmissão de uma mensagem de <i>flits</i> de carga útil através da MoNoC. Ilustra-se o processo adotado na NI para o envio de mensagens pela rede. Em (A) a mensagem encaminhada é fragmentada pela NI. Em (B) aparecem os pacotes resultantes da fragmentação, que trafegarão na rede. Em (C) mostra-se a entrega dos pacotes da mensagem.....	125
Figura 5.12 - Diagrama de blocos da NI.....	126
Figura 5.13 - Diagrama de sequência de mensagens para a abertura de uma sessão. ....	128
Figura 5.14 – Transmissão de dados pela MoNoC. Em destaque os núcleos <i>slvProbe</i> e <i>mstProbe</i> e seus respectivos pontos de observação de tráfego para coleta de estatística sobre taxa de injeção ( <i>mstProbe</i> ) e recepção ( <i>slvProbe</i> ) de dados.....	129
Figura 5.15 - Propostas de alternativas de rotas. Das rotas apresentadas, a rota (A) é a que proporciona menor taxa média e menor pico de ocupação. ....	131
Figura 5.16 – Diagrama de estados das NIs com foco no suporte a adaptação de rotas. ....	131
Figura 6.1 – Estrutura do roteador da NoC_Complementar, mostrando a diferenciação dos canais virtuais por algoritmo de roteamento empregado. ....	136
Figura 6.2 – Escolha do canal virtual para transmissão na NoC_Complementar para cada um dos algoritmos de roteamento. A letra “O” representa o emprego do canal virtual <i>cv_orig</i> , enquanto “C” representa o uso do canal virtual <i>cv_comp</i> para a transmissão dos pacotes. ....	137
Figura 6.3 - Ambiente empregado na validação das NoC_complementares. ....	138
Figura 6.4 - Cenários de tráfego Complemento e <i>Hotspot</i> aplicados durante etapa de validação. ....	140
Figura 6.5 – À esquerda, pares comunicantes analisados no cenário de tráfego <i>todos para todos</i> . À direita, os três conjuntos possíveis de canais de comunicação para a transmissão de pacotes do centro para os extremos quando se usam diferentes algoritmos de roteamento. Complemento ( <i>West/East first mínimo</i> ), WFM empregando somente <i>West first mínimo</i> e XY empregando XY puro. ....	141
Figura 6.6 – Latências de rede e de aplicação médias em número de ciclos de relógio, capturadas durante simulação com emprego do cenário de tráfego <i>todos para todos</i> e variando-se a taxa de injeção de dados. Mostram-se os resultados obtidos para a NoC Complementar ( <i>comp</i> ), para a NoC usando apenas o algoritmo <i>West first mínimo</i> ( <i>wfm</i> ) e a NoC aplicando o algoritmo XY ( <i>xy</i> ). ....	143
Figura 6.7 – No topo, pares comunicantes analisados no cenário de tráfego Complemento. Seguem-se os possíveis caminhos usados para a transmissão de pacotes entre cada par comunicante quando da utilização das diferente infraestruturas de comunicação. ....	144
Figura 6.8 - Latências médias, em ciclos de relógio, capturadas durante simulação com emprego do cenário de tráfego complemento e variando-se a taxa de injeção de dados na rede. Os resultados derivam de dados de simulação obtidos para a NoC_Complementar ( <i>comp</i> ), para a NoC_WFM ( <i>wfm</i> ) e para a NoC_XY ( <i>xy</i> ). ....	145
Figura 6.9 - No topo, pares comunicantes analisados no cenário de tráfego <i>hotspot</i> . Abaixo, os possíveis caminhos de comunicação usados para a transmissão de pacotes entre cada par comunicante quando da utilização das diferente infraestruturas de comunicação.....	146
Figura 6.10 – Disposição dos pares comunicantes para dois cenários de tráfego utilizados na validação da MoNoC. Em (a), um par comunicante opera uma comunicação que sofre interferência de tráfegos de ruído concorrentes. Em (b) quatro pares origem ( $S_i$ ) destino ( $T_i$ ) estaticamente definidos se comunicam sob interferência de tráfegos de ruído aleatórios de todos os demais pontos da rede. .	151
Figura 6.11 – Latências obtidas para o cenário de tráfego <i>Interferência</i> quando não é realizada adaptação nas rotas de comunicação. ....	152
Figura 6.12 – Latências obtidas para o cenário de tráfego <i>interferência</i> quando empregada a funcionalidade de adaptação de rotas na MoNoC.....	153
Figura 6.13 – Custos de tempo de abertura/fechamento de sessão e adaptação de rota. ....	154
Figura 6.14 – Disposição do par origem destino observado na MoNoC sob carga dinâmica de comunicação. ....	156
Figura 6.15 – Resultados de simulação para um cenário dinâmico de tráfego usando a MONOC. A mudança de pares comunicantes ocorre após 50 pacotes enviados para um mesmo destino, respeitando uma taxa de injeção de dados entre 20% e 30% da capacidade de transmissão de um canal. Em (a) tem-se os resultados de latência quando a adaptação não é explorada; em (b) aparecem os resultados com adaptação de rotas. ....	158
Figura 6.16 – Cenário dinâmico de tráfego. Mudança de pares comunicantes ocorrendo após 100 pacotes enviados, respeitando uma taxa de injeção de dados entre 30% e 40% da capacidade de transmissão de um canal. Em (a) os resultados de latência quando a adaptação não é aplicada. Em (b) nota-se a redução da latência de aplicação quando do emprego da adaptação. ....	159
Figura 7.1 – Comparação das NoCs Hermes-SR, MoNoC e Hermes. ....	165





## LISTA DE TABELAS

Tabela 2.1 – Comparativo das NoCs com topologia irregular investigadas. ....	40
Tabela 3.1 - Identificação das direções adotadas em rotas da NoC Hermes-SR. O prefixo 0x indica um valor hexadecimal. ....	61
Tabela 4.1 –Comparação de área ocupada por roteadores centrais das NoCs Hermes e Hermes-SR em FPGA. O dispositivo alvo usado foi um FPGA da família Virtex-II PRO, o xc2vp30-7. Resultados da síntese utilizando-se a ferramenta XST do ambiente ISE versão 9.1i. ....	91
Tabela 4.2 - Comparação de latência média para o cenário de tráfego todos para todos. ....	96
Tabela 4.3 - Comparação de resultados de latência média para o cenário da Figura 4.7(c).....	100
Tabela 4.4 - Comparação de resultados de latência média para o cenário da Figura 4.11(d).....	103
Tabela 4.5 – Comparação da latência média das NoCs (em ciclos de relógio) .....	107
Tabela 5.1 – Resultado de síntese de um roteador central da MoNoC (5 portas de entrada e saída) .....	132
Tabela 5.2 – Resultado de síntese de uma interface de rede da MoNoC .....	133
Tabela 6.1 – Latências capturadas para o cenário de tráfego <i>todos para todos</i> com 80 Mbps de injeção de tráfego quando a origem é o nó 12 da rede e os destinos são os nós 0, 4, 20 e 24.....	141
Tabela 6.2 - Latências capturadas para o cenário de tráfego Complemento quando a taxa de injeção por nó é de 160 Mbps. ....	144
Tabela 6.3 - Latências capturadas para o cenário de tráfego <i>hotspot</i> quando a taxa de injeção por nó é de 80 Mbps. As linhas de cada tabela parcial são diferenciadas pela distância em <i>hops</i> entre origens e destinos.....	146
Tabela 6.4 – Valores de latências obtidos para o cenário de tráfego <i>interferência</i> quando adaptada a rota de comunicação. ....	154
Tabela 6.5 – Carga de ocupação e de pico das rotas avaliadas durante a adaptação de rota.....	155



## LISTA DE SIGLAS

<b>ACG</b>	<i>Application Communication Graph</i>
<b>CI</b>	<i>Circuito Integrado</i>
<b>CMP</b>	<i>Chip Multiprocessor</i>
<b>DyAD</b>	<i>Dynamic Adaptive Deterministic</i>
<b>FCFS</b>	<i>First Come First Served</i>
<b>FPGA</b>	<i>Field Programmable Gate Array</i>
<b>GPP</b>	<i>General Purpose Processor</i>
<b>HP</b>	<i>Hot Potato</i>
<b>HSC</b>	<i>High Speed Computing</i>
<b>IP</b>	<i>Intellectual Property</i>
<b>ITRS</b>	<i>International Technology Roadmap of Semiconductors</i>
<b>LUT</b>	<i>Look Up Table</i>
<b>Mbps</b>	<i>Megabits per second</i>
<b>MPSoC</b>	<i>Multiprocessor System on a Chip</i>
<b>nfm</b>	<i>Negative first Mínimo</i>
<b>nfnm</b>	<i>Negative First Não Mínimo</i>
<b>NI</b>	<i>Network Interface</i>
<b>nlm</b>	<i>North Last Mínimo</i>
<b>nlNm</b>	<i>North Last Não Mínimo</i>
<b>NoC</b>	<i>Network-on-Chip</i>
<b>OCP</b>	<i>Open Core Protocol</i>
<b>PDSP</b>	<i>Programmable Digital Signal Processor</i>
<b>PE</b>	<i>Processing Element</i>
<b>QE</b>	<i>Quadrado Envolvente</i>
<b>QoS</b>	<i>Quality of Service</i>
<b>RCA</b>	<i>Region Congestion Awareness</i>
<b>RTL</b>	<i>Register Transfer Level</i>

<b>SoC</b>	<i>System on a Chip</i>
<b>SR</b>	<i>Source Routing</i>
<b>VC</b>	<i>Virtual Channel</i>
<b>VCT</b>	<i>Virtual Cut-Through</i>
<b>wfm</b>	<i>West First Mínimo</i>
<b>wfnm</b>	<i>West First Não Mínimo</i>
<b>WH</b>	<i>Wormhole</i>

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>23</b>
1.1 Motivação .....	27
1.2 Objetivos .....	29
1.3 Originalidade e contribuições da tese .....	30
1.4 Organização do documento .....	31
<b>2. TRABALHOS RELACIONADOS .....</b>	<b>33</b>
2.1 Infraestrutura de comunicação em chip .....	33
2.1.1 NoCs de topologia regular .....	34
2.1.2 NoCs de topologia irregular .....	36
2.2 Congestionamento .....	40
2.2.1 Pontos quentes .....	42
2.3 Controle de fluxo de comunicação .....	43
2.3.1 Mapeamento de tarefas .....	44
2.3.2 Adaptação local .....	46
2.3.3 Adaptação global .....	50
2.3.4 Emprego de monitores .....	53
<b>3. MAPEAMENTO DE ROTAS .....</b>	<b>59</b>
3.1 A Infraestrutura de comunicação Hermes-SR .....	59
3.1.1 Estrutura do pacote .....	60
3.1.2 Arquitetura do roteador Hermes-SR .....	61
3.2 Algoritmos de roteamento base para definição de rotas .....	66
3.2.1 Algoritmo de roteamento <i>negative first</i> .....	68
3.2.2 Algoritmo de roteamento <i>West first</i> .....	72
3.2.3 Algoritmo de roteamento <i>North last</i> .....	76
3.3 Avaliação de rotas .....	78
3.3.1 Organização do mapeamento de rotas .....	80
3.3.2 Exploração de cenários de rotas .....	84
<b>4. AVALIAÇÃO DE RESULTADOS .....</b>	<b>87</b>
4.1 Infraestrutura de comunicação .....	87
4.1.1 Experimentos conduzidos .....	88
4.1.2 Resultados obtidos .....	90
4.2 Algoritmos usados como base para definição de rotas .....	91
4.2.1 Experimentos conduzidos .....	91
4.2.2 Resultados obtidos .....	93
4.3 Cenário de rotas versus adaptação em tempo de execução .....	104
4.4 Consideração final .....	107
<b>5. ADAPTAÇÃO DE ROTAS .....</b>	<b>109</b>
5.1 Proposta de adaptação de rota .....	110
5.1.1 Adaptação a partir das fronteiras da rede .....	111
5.1.2 Garantia de alternativas de rotas .....	111
5.1.3 Considerações finais .....	113
5.2 A infraestrutura de comunicação MoNoC .....	114
5.2.1 Serviços diferenciados .....	115
5.2.2 Sonda de rede .....	118
5.3 Interface de rede .....	122
5.3.1 Serviços disponibilizados pela NI .....	122
5.3.2 Suporte à adaptação de rotas .....	126
5.4 Implementação em <i>hardware</i> .....	132
<b>6. VALIDAÇÃO E RESULTADOS .....</b>	<b>135</b>
6.1 Emprego de algoritmos complementares .....	135
6.1.1 Suporte a complementaridade de algoritmos .....	135

6.1.2	Validação do emprego de algoritmos complementares .....	138
6.1.3	Avaliação de resultados .....	140
6.2	Emprego da infraestrutura de comunicação MoNoC .....	147
6.2.1	Validação da infraestrutura de comunicação .....	147
6.2.2	Validação da Interface de rede.....	149
6.2.3	Validação completa da MoNoC .....	150
6.2.4	Resultados obtidos para cenários estáticos de tráfego.....	151
6.2.5	Resultados obtidos para cenários dinâmicos de tráfego .....	155
6.3	Considerações finais .....	159
<b>7.</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS .....</b>	<b>161</b>
7.1	Contribuições do trabalho.....	161
7.2	Conclusões .....	163
7.3	Trabalhos futuros .....	166
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>169</b>

# 1. INTRODUÇÃO

Na década de 60 do século XX, Gordon Moore observou que a capacidade dos circuitos integrados (CIs) duplicava a cada 18 meses [SCH97], o que vem se mantendo válido desde então. A esta observação denominou-se mais tarde “Lei de Moore”, a qual tem funcionado como um propulsor da evolução da indústria de semicondutores. Nota-se tal avanço pelo aumento exponencial do número de transistores por unidade de área de CI ocorrido nos últimos 40 anos [WES94], conforme ilustra a Figura 1.1.

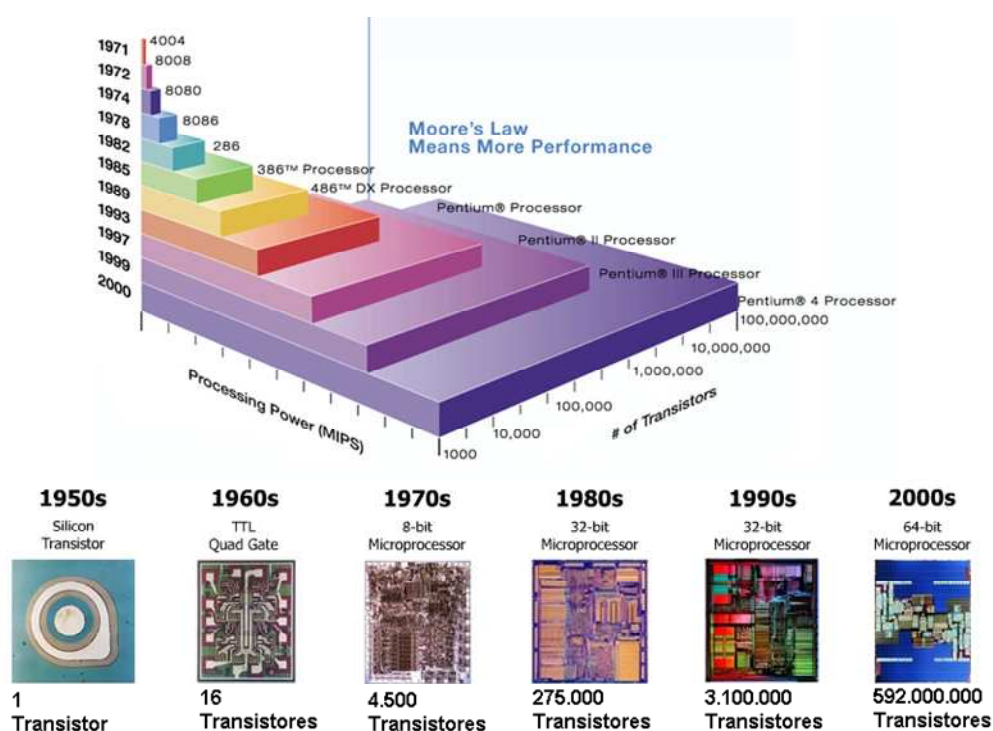


Figura 1.1 – Evolução da densidade de transistores e do poder de processamento [INT09].

O avanço contínuo da tecnologia submicrônica (em inglês, *deep submicron*) e o consequente aumento do número de transistores por unidade de área de CI, permitem a construção de sistemas completos em um único CI, criando-se o conceito de SoC (em inglês, *System on a Chip*) [RAJ00]. O projeto de SoCs é visto como uma tendência [MAR01], pois permitem resolver parcialmente problemas tais como (i) gargalos de desempenho na comunicação entre diferentes módulos de *hard-*



*ware* dentro de um chip, (ii) o consumo excessivo de área e de energia de sistemas [ZHU02]. No *International Technology Roadmap for Semiconductors* (ITRS) [ITR05], conjunto de documentos elaborado por representantes de grupos industriais da área de semicondutores, prevê-se que CIs fabricados na próxima década conterão dezenas de bilhões de transistores com dimensão em torno de 50nm e frequência de operação acima de 10 GHz.

Estruturalmente um SoC é constituído a parti do agrupamento de um conjunto de módulos de *hardware* comumente interconectados por infraestruturas de comunicação tais como conexões ponto a ponto e/ou barramentos. Conexões ponto a ponto são eficientes tanto em desempenho quanto no suporte ao controle do consumo de energia por disponibilizarem soluções dedicadas, tendo como desvantagem a limitação da escalabilidade. Barramentos são mais eficientes quanto à escalabilidade quando comparados a soluções ponto a ponto. Embora vantajoso sobre tal aspecto, o emprego de barramentos acarreta na utilização de fios globais para distribuir sinais, o que é visto como uma abordagem ineficaz para o projeto de SoCs [HO01] [DAL01], por ser suscetível a problemas elétricos (e.g. *crosstalk*) e estruturais (e.g. escalabilidade limitada), além de permitir apenas uma comunicação a cada instante. O uso de barramentos segmentados pode ser adotado como uma forma de contornar o último problema. Porém, a complexidade envolvida em sua adoção e o potencial bloqueio de todo o barramento segmentado são desvantagens claras desta abordagem.

Uma proposta que atenda os requisitos de SoCs futuros, quanto à melhora da eficiência na comunicação, vem do uso de redes em chip (em inglês, *Networks on Chip*, NoCs). NoCs adaptam conceitos de redes de computadores e sistemas distribuídos para o projeto de infraestruturas de comunicação em CIs complexos. Comparado a mecanismos de comunicação previamente citados (ponto a ponto, barramento simples e barramento segmentado), o uso de NoCs é vantajoso por permitir melhorar parâmetros tais como eficiência energética, reusabilidade, comunicação não bloqueante e escalabilidade [GUE00] [BEN02]. Um exemplo de vantagem do emprego de NoCs vem da possibilidade de estruturar a comunicação global de tal forma a explorar o controle de certos efeitos elétricos, tal como o *crosstalk*. Isto facilita o uso de circuitos de sinalização mais eficientes que permitam reduzir a dissipação de potência por até uma ordem de grandeza e aumentar em até três vezes a velocidade de propagação de sinais [DAL98]. Tais vantagens levam à aceitação de NoCs como mecanismo de interconexão em projeto de SoCs [BEN01] [KUM02] [WIN01], contribuindo ainda para o paralelismo tanto da computação quanto da comunicação.

Avanços na tecnologia de construção de microprocessadores permitiram a evolução do poder computacional destes dispositivos nas últimas décadas. Exemplos destes avanços são: o aumento da frequência de relógio e a capacidade de execução simultânea de múltiplas instruções. No entanto, assim como os dispositivos de *hardware*, também evoluíram os requisitos de aplicações no que se refere à qualidade dos resultados e o tempo de processamento para obtenção dos mesmos [CUL99]. Cada vez mais exige-se resultados precisos, confiáveis e computados sobre um volume crescente de dados. Como o fechamento de um ciclo, tais requisitos de aplicações obrigam o aumento do desempenho dos dispositivos de *hardware*, que devem levar a um novo patamar de requisitos de aplicação e assim sucessivamente. Uma forma de garantir o aumento de desempenho vem do emprego de paralelismo que deve ser viabilizado por infraestruturas em *hardware* e explorado por descrições de *software* [GRA03]. No presente trabalho, o foco está em sistemas de *hardware* multiprocessados.

Multiprocessadores integrados em um único CI (em inglês, *Chip Multiprocessor*, CMP) são classes de SoCs que dão suporte ao paralelismo através do emprego de múltiplos elementos de *hardware*, também chamados de núcleos de *hardware* de propriedade intelectual (em inglês, *Intellectual Property Cores*, *IP cores* ou núcleos IP em português), que podem ser de processamento, de armazenamento, de comunicação ou de entrada e saída. Os elementos de processamento (em inglês, *Processing Elements*, PEs), podem ser de aplicação específica, quando se destinam a uma aplicação ou domínio de aplicação (e.g. Processadores digitais de sinal), ou de propósito geral, quando não tem uma aplicação específica (e.g. Microprocessadores).

CMPs são dispositivos compostos por PEs não necessariamente homogêneos, mas de propósito geral, disponibilizando plataformas genéricas no que se refere ao atendimento dos requisitos das aplicações [JER05a]. Sistemas multiprocessados em um único CI (em inglês, *Multiprocessor Systems on Chip*, MPSoCs) são uma classe de CMPs, compostas de PEs heterogêneos, formados, sobretudo por PEs de aplicação específica. Em CMPs as funcionalidades exigidas em aplicações são encontradas a partir da implementação em *software*. A vantagem de MPSoCs sobre CMPs é a eficiência do primeiro quanto a desempenho e/ou consumo de energia para o atendimento dos requisitos de uma dada aplicação ou classe de aplicações. Já a vantagem de CMPs sobre MPSoCs vem da generalidade, possibilitando o atendimento mais fácil do *time to market*, além de permitir o aumento da vida útil de um dado produto ou plataforma. *Time to market* é a necessidade que as indústrias têm de lançar um produto no mercado no prazo mais curto possível para obter maior lu-

cratividade [RAU94]. Ambas abordagens provêem infraestruturas que potencializam a adoção de paralelismo com prática de projeto, contribuindo para a consequente evolução do poder computacional. Exemplos de CMPs e o avanço no número de PEs que compõe tais plataformas são dados na Figura 1.2.

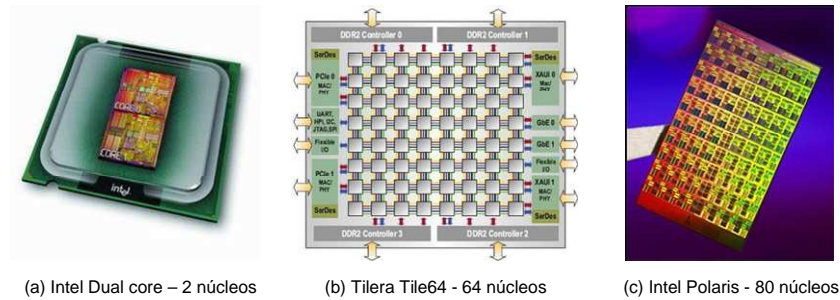


Figura 1.2 - Exemplos da evolução no número de PEs em CMPs.

MPSoCs são mais comumente empregados em aplicações específicas que exigem computação concorrente de tempo real, com requisitos de eficiência em área e energia [JER05a]. Telefones celulares, equipamentos de telecomunicação, televisão digital, e sistemas de vídeo game são exemplos de aplicações embarcadas que empregam MPSoCs. Soluções baseadas puramente em processadores de propósito geral (em inglês, *general purpose processors*, GPPs) tendem a não ser usadas em aplicações embarcadas devido ao seu alto custo, baixa eficiência energética e inabilidade de atender requisitos de tempo real. Uma solução mais eficiente vem de dispositivos que proporcionam o atendimento de requisitos de desempenho disponibilizando a flexibilidade de configuração/programação. Fabricantes como a Texas Instruments [TEX09] e a Freescale Semiconductors [FRE09] disponibilizam comercialmente processadores de sinais digitais programáveis (em inglês, *programmable digital signal processor*, PDSP). Adicionalmente, fabricantes de dispositivos reconfiguráveis tais como a Xilinx [XIL09] e a Altera [ALT09] também têm apresentado soluções que embarcam microprocessadores em FPGAs para auxiliar na criação de MPSoCs.

Como resultado do aumento do poder computacional associado a soluções embarcadas, pode-se notar o surgimento de dispositivos que cada vez mais acumulam funcionalidade, tais como aparelhos celulares atuais, que além de sua função básica, disponibilizam funcionalidades que vão de uma agenda até, mas não apenas, a visualização de filmes codificados (e. g. DivX) e reprodução de música (e. g. MP3). Cada uma destas funcionalidades possui seus próprios requisitos de computação e comunicação. Dado que diferentes aplicações podem coexistir neste tipo de plataforma, o

uso dos recursos disponíveis por uma aplicação não pode inviabilizar ou degradar consideravelmente a qualidade das outras aplicações. No que tange a comunicação, diferentes fluxos de comunicação entre pares origem/destino que compartilham canais podem levar a ocorrência de congestionamentos e à consequente degradação da qualidade da comunicação, aumentando o tempo de execução de aplicações e diminuindo sua qualidade.

Tendo em vista a tendência de que múltiplas aplicações sejam executadas em um mesmo sistema multiprocessado, cenários de congestionamento e a potencial queda de desempenho da comunicação tendem a ocorrer se não houver planejamento ou gerenciamento durante o desenvolvimento ou uso das plataformas multiprocessadas. Com base neste pressuposto, o presente trabalho tem por proposta investigar o controle de congestionamento através da reconfiguração de rotas de comunicação entre pares origem/destino em redes de comunicação em CI. O objetivo é prover meios que possibilitem garantir a qualidade da comunicação.

## 1.1 MOTIVAÇÃO

O aumento contínuo da capacidade de inclusão de transistores em uma mesma área de CIs, que hoje podem alcançar a ordem de centenas de milhões e que continuam seguindo a lei de Moore segundo o ITRS [ITR08], permite que cada vez mais PEs constituam um SoC. Isto é vantajoso, visto que possibilita a redução do tamanho dos dispositivos e do número de dispositivos que compõe um produto. Sua desvantagem vem do aumento da complexidade de descrição, validação e gerenciamento de seus projetos [RAJ00]. Tal complexidade potencializa o aumento do tempo de projeto, o que pode comprometer a vantagem do uso de SoCs pelo possível descumprimento do *time to market*.

Sendo uma evolução de SoCs e uma tendência em projetos futuros, MPSoCs tem por objetivo prover soluções ótimas para um dado problema, ou classe de problemas, quanto a desempenho e consumo de energia [JER05a]. CMPs podem ser uma proposta para cobrir a lacuna de produtividade de projeto (do inglês *design productivity gap*) de plataformas MPSoC até seu lançamento no mercado. Independente da abordagem, tanto CMPs como MPSoCs tiram proveito de conceitos bem fundamentados em áreas como sistemas operacionais e redes de computadores, no que tange o paralelismo. Apesar de plataformas multiprocessadas não serem novidade, sua adoção e adaptação em tecnologias submicrônicas é motivação de pesquisa.

A evolução de produtos tais como celulares (e. g. iPhone, Nokia N95) e *smartphones* tem disponibilizado cada vez mais recursos para o usuário final. A integração de câmeras digitais, filmadoras, tocadores de música e de filme codificados, navegadores de Internet, receptores de sinal de posicionamento global por satélite entre outras funcionalidades, deixaram de ser uma tendência e passaram a ser realidade neste tipo de dispositivo portátil. Individualmente, tais funcionalidades podem encontrar seus requisitos de execução se forem desenvolvidos módulos de *hardware* específicos. Outra forma de garantir tais requisitos vem da adoção do paralelismo na descrição de aplicações e sua execução em sistemas multiprocessados.

Após a especificação de uma aplicação, o paralelismo desta é alcançado com seu particionamento em um conjunto de tarefas. A associação destas tarefas a PEs de um sistema multiprocessado é chamada mapeamento, que pode ser realizado estática ou dinamicamente. No mapeamento estático, cada tarefa é associada a um PE onde será executada. No mapeamento dinâmico, cada tarefa é associada a um subconjunto de PEs que compõem a plataforma multiprocessada, sendo que tal subconjunto pode representar todos os PEs da plataforma. A decisão de qual PE será responsável por executar uma dada tarefa é feita em tempo de execução, de acordo com uma política adotada pelo mapeamento dinâmico. Apesar de ser importante para garantir o desempenho global da aplicação, o mapeamento pode levar a situações de congestionamento na infraestrutura de comunicação quando aplicado visando unicamente atender requisitos de computação. No caso de aplicações dependentes de comunicação, a decisão de onde mapear uma tarefa tem de levar em consideração a situação da rede durante a execução.

Uma proposta para melhorar o desempenho de aplicações no que tange a comunicação pode ser obtida através da adaptação de rota, o que permite o desvio de canais congestionados na rede. A adaptação de rotas tem como vantagem o desvio de canais considerados congestionados. As decisões de adaptação de rotas podem ser feitas externa ou internamente à rede. Decisões internas a rede normalmente consideram o estado local de contenção para a adaptação da rota. Tal abordagem tem como vantagem o tempo de reação e como desvantagem o pouco (ou nulo) conhecimento do estado de ocupação do restante da rede. Decisões externas à rede normalmente levam em consideração o estado de ocupação da rede. A vantagem desta abordagem é a potencial decisão de uso de rotas menos propensas a congestionamento e como desvantagem o tempo de necessário para computação de tais rotas. No presente trabalho considera-se decisões internas como aquelas feitas com base em algoritmos de roteamento implementados internamente a cada

roteador e computadas em tempo de execução. Decisões externas à rede são caracterizadas pelo mapeamento de rotas entre pares comunicantes, processo esse realizado em tempo de projeto, ou ainda adaptações de rota realizadas em tempo de execução por recursos especiais conectados aos roteadores das redes em chip.

As tendências de emprego de ambientes multiprocessados em projetos de SoC e os problemas de congestionamento previstos com a evolução destes motivam a exploração de infraestruturas que possibilitem o controle da queda de desempenho da comunicação.

## 1.2 OBJETIVOS

Esta tese aborda a redefinição de rotas entre pares de entidades comunicantes como forma de garantir a qualidade da comunicação. Esta Seção lista os objetivos estratégicos e específicos do trabalho a ser proposto.

Os objetivos estratégicos desta tese são:

- Dar apoio ao projeto de plataformas multiprocessadas cuja infraestrutura de comunicação é baseada em redes em chip;
- Dominar o problema de congestionamento em infraestruturas de comunicação em chip;
- Dominar a problemática de estratégias de controle de congestionamento em infraestruturas de comunicação.

Os objetivos específicos são:

- Desenvolver uma rede em chip capaz de:
  - Realizar arbitragem eficiente - uma forma de permitir maior precisão na previsão de latência dos pacotes passa por um controle mais eficiente de arbitragem;
  - Definir rotas adaptáveis - o uso de roteamento determinístico não garante o contorno de pontos de contenção. Assim, a exploração de algoritmos de roteamento adaptativos e a elaboração de métodos de definição de rotas estão incluídas no escopo deste trabalho;
- Propor mecanismos de monitoramento da comunicação de dois tipos:

- Propor mecanismos de avaliação interna à rede - para adaptação de rotas em uma rede, é indispensável o conhecimento de seu estado de utilização. Análises, tal como o percentual de ocupação de um canal, contribuem para as tomadas de decisão de adequação de rotas;
- Propor mecanismos de avaliação de fluxo fim a fim - a adaptação de rotas tem de ser realizada quando alguma situação indesejada é detectada. No presente trabalho, propõe-se o monitoramento das comunicações entre pares comunicantes. O objetivo é notificar a necessidade de adaptações nos casos em que algum serviço de rede não esteja sendo cumprido;
- Propor mecanismos de adaptação da comunicação
  - Propor interfaces de rede com poder de adaptação de rotas - a rede proposta adota roteamento na origem. Com a notificação de descumprimento de serviços de rede, este mecanismo permitirá adaptar rotas entre pares comunicantes;
  - Investigar propostas de redefinição de rotas - o suporte à redefinição de rotas necessitará de todos os mecanismos propostos. A exploração de canais virtuais e algoritmos de roteamento adaptativos são algumas das propostas investigadas.

### 1.3 ORIGINALIDADE E CONTRIBUIÇÕES DA TESE

A proposta de adaptação de caminhos em infraestruturas de comunicação com roteamento na origem em tempo de execução é original. Dos trabalhos revisados com tópicos similares, a infraestrutura alvo está em redes de computadores e não em NoCs. Concentrando-se em projetos de CIs, o foco dos trabalhos revisados está na definição de rotas em tempo de projeto, atendendo a objetivos de aplicações mapeadas estaticamente.

O presente trabalho tem como contribuições:

- Disponibilizar de infraestruturas de interconexão com roteamento na origem;
- Propor um fluxo de avaliação de rotas e mapeamento de rotas de forma estática em tempo de projeto para a comunicação entre tarefas;
- Propor uma plataforma com capacidade de decisão dinâmica de rotas (adaptação dinâmica de rotas) tomada com base no estado de ocupação de toda a rota, dando suporte ao atendimento de requisitos de desempenho.

## 1.4 ORGANIZAÇÃO DO DOCUMENTO

O restante deste documento é organizado da seguinte forma. O Capítulo 2 tem por objetivo apresentar trabalhos relacionados ao tema abordado. Revisam-se temas como propostas de infraestruturas de comunicação em chip, trabalhos sobre detecção e controle de congestionamento e mecanismos de controle de contenção. No Capítulo 3 apresenta-se a proposta de mapeamento de rotas, incluindo um detalhamento de infraestruturas de comunicação em chip, algoritmos de roteamento e mecanismos de avaliação de rotas. No Capítulo 4 discute-se um conjunto de experimentos relacionados à proposta de mapeamento de rotas. Busca-se evidenciar as contribuições alcançadas com cada um dos elementos apresentados no processo de mapeamento de rotas. O Capítulo 5 detalha a proposta de adaptação de rotas, incluindo minúcias sobre as infraestruturas de comunicação em chip e os mecanismos definidos para alcançar o objetivo de adaptação. O objetivo do Capítulo 6 é a validação das propostas de roteamento, da infraestrutura de comunicação e dos mecanismos de suporte a adaptação. O Capítulo 7 conclui a tese, destacando os trabalhos desenvolvidos no contexto do doutorado, as contribuições deste e apontando um conjunto de atividades de exploração futura.





## 2. TRABALHOS RELACIONADOS

A adoção de sistemas multiprocessados, aliada ao uso eficiente dos recursos destes, permite o aumento de desempenho quando comparado ao emprego de sistemas monoprocessados [JER05a]. Computacionalmente, as vantagens decorrem da divisão de uma aplicação em unidades menores, chamadas tarefas, e a exploração do paralelismo de execução destas. A distribuição das tarefas pelos PEs que compõem uma plataforma multiprocessada viabiliza a exploração do paralelismo. Sob o ponto de vista de comunicação, é natural que um conjunto de tarefas que define uma aplicação deva colaborar, implicando na necessidade de troca de informações entre tarefas. Desta forma, os ganhos que podem ser alcançados com a exploração do paralelismo em aplicações exigem a preocupação com aspectos de comunicação, tais como o modo pelo qual as tarefas colaboram entre si e a forma como são utilizados recursos de comunicação.

Diferentes aspectos influenciam na comunicação, indo desde características arquiteturais da infraestrutura de comunicação (e.g. topologia, algoritmo de roteamento) até as estratégias adotadas para a comunicação (e.g. uso de memória global, a definição do protocolo). Definidos tais aspectos, a ocorrência de congestionamentos é fator de degradação da comunicação e de consequente diminuição da eficiência do uso de sistemas multiprocessados. Desta forma, o congestionamento na comunicação é um efeito que deve ser controlado.

No presente trabalho, propõem-se duas infraestruturas de comunicação e mecanismos para controle da diminuição da eficiência da comunicação. Neste Capítulo são revisados trabalhos relacionados ao tema, divididos em três temas: *(i)* propostas de infraestruturas de comunicação em chip, *(ii)* o conceito de congestionamento e *(iii)* propostas de mecanismos de adaptação de tráfego em infraestruturas de comunicação em chip. Cabe ressaltar que, no que tange infraestruturas de comunicação, neste trabalho o foco está em redes em chip.

### 2.1 INFRAESTRUTURA DE COMUNICAÇÃO EM CHIP

Uma infraestrutura de comunicação em chip é o recurso básico necessário para garantir a

transmissão de dados entre entidades comunicantes em um sistema, sendo representados pelo *hardware* que o implementa e pelos serviços por ele disponibilizados, tal como envio e recepção de pacotes. Diferentes infraestruturas de comunicação em chip podem ser empregadas para a interconexão de módulos em sistemas embarcados, tal como conexões dedicadas, barramentos e redes em chip (i.e. NoCs). Destas, NoCs são vistas como tendência devido a suas vantagens de escalabilidade, diminuição/redução de fios globais, aumento da largura de banda de comunicação, suporte a baixa dissipação de potência entre outras [BEN02] [HO01]. No presente trabalho, NoCs são objeto central na pesquisa. Revisa-se agora algumas propostas de NoCs, divididas segundo a regularidade das topologias.

### 2.1.1 NOCS DE TOPOLOGIA REGULAR

Com a evolução das tecnologias submicrônicas e a constante busca pelo aumento de desempenho de dispositivos eletrônicos, a adoção de sistemas multiprocessados é tendência no projeto de CIs. MPSoCs deverão conter cada vez mais PEs, e o custo de lidar com esta multiplicação pode levar à adoção de PEs uniformes segundo aspectos tais como dimensões e fator de forma no intuito de facilitar o projeto automatizado e o reuso de módulos e subsistemas. Também, é possível que se adote a idéia de *clusters* de processamento ao invés de PEs de aplicação específica. Do ponto de vista da infraestrutura de comunicação, tende-se a adotar arquiteturas parametrizáveis, mas de topologia regular, o que facilita a replicação de PEs homogêneos em funcionalidade e dimensões. Revisa-se a seguir alguns trabalhos que propõem infraestruturas de comunicação em chip parametrizáveis para o atendimento genérico de requisitos de comunicação.

**Guerrier et al.** [GUE00] propuseram a rede em chip SPIN. SPIN adota uma topologia árvore gorda, onde cada núcleo IP que compõem a plataforma está localizado em um roteador folha da topologia, representados na Figura 2.1(a) como círculos escuros. Os núcleos IP colaboram entre si através da troca de mensagens, que são fragmentadas em pacotes antes de serem inseridas na rede. Sequências de palavras de 36 bits compõem um pacote. O cabeçalho do pacote fica na primeira palavra, onde um *byte* do cabeçalho identifica o endereço destino, e os demais *bits* são usados para informações de roteamento e controle. A carga útil dos pacotes pode ser de tamanho variável. Cada pacote é finalizado por um terminador que contém informações utilizáveis na detecção de erros de transmissão.

**Karim et al.** [KAR01] propuseram a rede Octagon. Conforme ilustrado na Figura 2.1(b), oito roteadores interconectados por doze canais de comunicação definem a topologia base desta NoC. Os roteadores são organizados em um anel cordal, com um canal para cada lado e um canal de cada roteador para o roteador localizado no ponto oposto do octógono virtual definido na topologia. Assim, cada roteador tem três conexões, uma com o roteador à sua esquerda, uma com roteador à sua direita e uma com o roteador oposto do octógono. Cada conexão entre dois roteadores interconectados diretamente é denominada *hop*. A distância entre dois roteadores quaisquer da rede é sempre no máximo dois *hops* dentro de um anel cordal. A rede Octagon é escalável. Se um dos roteadores é utilizado como ponte, mais octógonos podem ser agrupadas, conforme ilustra a Figura 2.2. Note-se que neste caso os roteadores que conectam dois octógonos são distintos dos demais, sendo denominados de *pontes*. A rede Octagon foi originalmente proposta para servir à escalabilidade na construção de processadores de rede (em inglês, *network processors*) maciçamente paralelos.

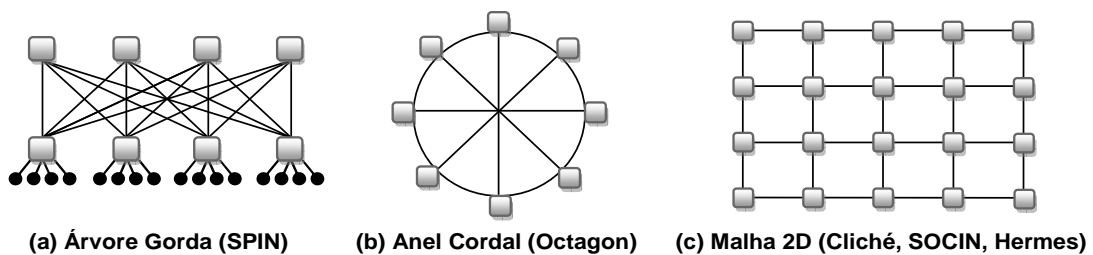


Figura 2.1 – Exemplos de topologias regulares e de NoCs que empregam estas.

**Kumar et al.** [KUM02] propuseram a rede em chip CLICHÉ, na qual foi adotada uma topologia em malha 2D, conforme ilustra a Figura 2.1(c). Cada roteador é conectado a um PE e o número de roteadores é igual ao número de PEs. Dependendo da posição do roteador na topologia, o número de roteadores vizinhos imediatos pode variar entre 2 e 4. Os Autores mostram que sob tráfego moderado, roteadores com tamanho de buffer igual a 8 é suficiente para reduzir a probabilidade de perda de pacotes na rede. Todavia, se a carga da rede é superior a metade da capacidade da mesma, então a perda de pacotes persiste mesmo quando se aumenta o tamanho das filas de armazenamento.

**Zeferino et al.** [ZEF03] propuseram a rede SoCIN, especificando-a de forma a permitir a adoção tanto da topologia malha 2D quanto toróide 2D. A SoCIN tem como elemento de roteamento um módulo denominado RASoC. Este roteador, descrito em VHDL, permite parametrizações em

quatro dimensões, quais sejam: (i) o número de portas de comunicação (limitado a 5), (ii) a largura do canal de comunicação, (iii) profundidade das filas de armazenamento e (iv) o número de bits utilizados no roteamento de pacotes (visando a escalabilidade da rede). O roteamento dos pacotes é definido pelo algoritmo determinístico XY.

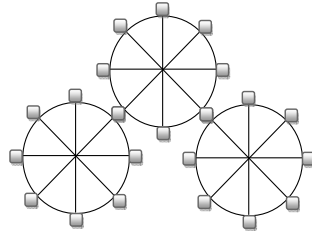


Figura 2.2 – Exemplo da infraestrutura Octagon de dimensões superiores a topologia base.

**Moraes et al.** [MOR04] propuseram a rede Hermes. Assim como a SoCIN, a rede Hermes permite a parametrização do número de portas dos roteadores, da largura dos canais e da profundidade das filas de armazenamento. Em sua primeira versão, a rede Hermes deu suporte somente à topologia malha 2D e ao algoritmo de roteamento XY determinístico. Com a evolução dos trabalhos em torno desta infraestrutura de comunicação, incluiu-se uma série de novas características, incluindo o suporte a topologia toro 2D, a adoção de algoritmos de roteamento adaptativos (e.g. *West first, North last, negative first*), a possibilidade de uso de canais virtuais, e a opção por controle de fluxo baseado em crédito (em inglês, *credit based*) ou *handshake*.

## CONSIDERAÇÕES FINAIS SOBRE NOCS REGULARES

As propostas de NoCs que assumem topologia regular, aqui revisadas, primam pela definição de infraestruturas genéricas. Isto é justificável, visto que são propostas pioneiras em projetos de NoCs. A partir do momento em que NoCs foram aceitas como necessárias para projetos de sistemas multiprocessados futuros, a busca por garantias de desempenho na comunicação começaram a ser melhor exploradas. Sob este aspecto, as pesquisas em torno da NoC Hermes têm buscado fortalecer o uso de topologias regulares e garantir a qualidade de serviços prestados pela rede.

### 2.1.2 NOCS DE TOPOLOGIA IRREGULAR

O desenvolvimento de (MP)SoCs e de aplicações específicas exige que sejam atendidos requisitos de projeto tal como redução de área e de dissipação de potência. No que se refere à infraestrutura de comunicação, tais requisitos podem ser mais facilmente atendidos em sistemas de



**Bolotin et al.** [BOL04] propõem a rede em chip QNoC. A construção desta NoC parte originalmente de uma topologia malha 2D regular, sendo adaptada de acordo com os núcleos IPs que compõem a plataforma alvo. Tal adaptação ocorre através da eliminação de roteadores e ligações da malha 2D, ação esta justificada pela heterogeneidade das dimensões de núcleos IP, conforme ilustrado na Figura 2.5. Podem ainda existir roteadores que não estão conectados a nenhum núcleo IP, mas fazem parte da rede em chip como nós de passagem. O roteamento de pacotes é definido por tabelas de roteamento, sendo as rotas entre pares origem/destino definidas de forma a atender os requisitos de comunicação, diminuindo a concorrência por recursos de comunicação e o tamanho das tabelas de roteamento [BOL07].

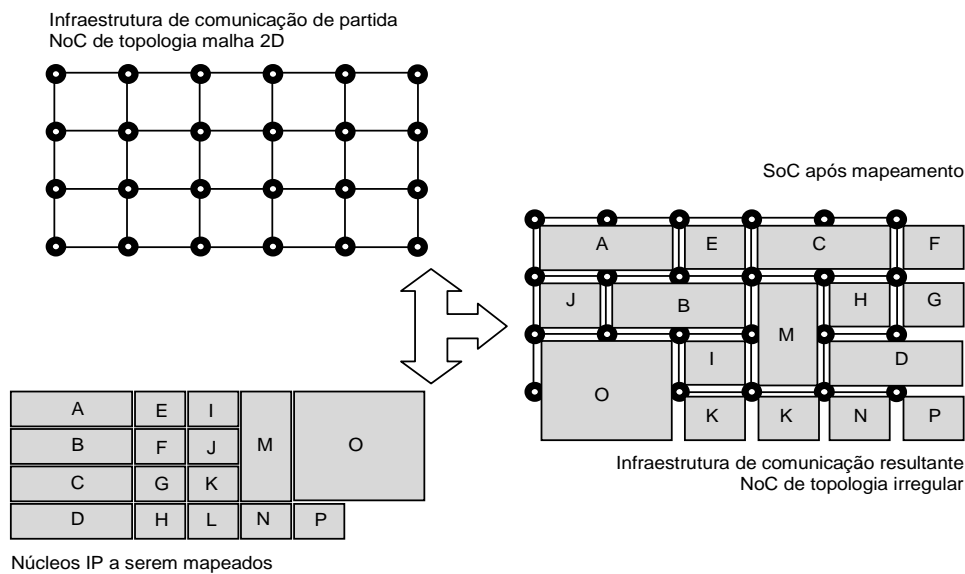


Figura 2.5 – Infraestrutura de comunicação QNoC resultante do processo de mapeamento de núcleos IP e da adaptação de uma NoC de topologia malha 2D.

**Ogras et al.** [OGR06a] propõem a inserção de fios longos como mecanismo para personalizar a NoC, contribuindo para a diminuição do congestionamento e da latência na comunicação entre pares origem/destino. Uma rede malha 2D é usada como infraestrutura de comunicação de base. A ela adicionam-se fios longos, respeitando um limite máximo de fios por roteador e observando a necessidade da aplicação alvo. Mantém-se o algoritmo de roteamento XY para atender a rede com topologia regular, adicionado-se um algoritmo livre de *deadlock* para uso nos enlaces inseridos.

**Srinivasan et al.** [SRI05] propõem uma técnica para a geração automática de topologia de NoCs e rotas entre pares comunicantes, objetivando aplicações específicas e levando em consideração o consumo de energia. Os autores computam a influência das taxas de injeção e do tamanho

dos fios de comunicação na energia dissipada nos canais. A técnica proposta se desenvolve em três fases, quais sejam: (i) criação da planta baixa inicial, (ii) mapeamento de núcleos sobre a NoC e (iii) definição das rotas de comunicação. Na primeira fase, busca-se a divisão dos elementos da aplicação de tal forma que se atenda a largura de banda exigida tanto para injeção quanto para recepção. Na segunda fase, os pares comunicantes são aproximados em uma infraestrutura de comunicação do tipo malha 2D, tendo por objetivo a diminuição do número de roteadores a serem atravessados desde a origem até o destino. Na última fase, tenta-se reduzir os caminhos utilizados entre os pares comunicantes, visando-se a redução do número de roteadores, porém respeitando a largura de banda a que cada canal dá suporte.

**Bertozi et al.** [BER04] propuseram a rede em chip Xpipes. Ela possui chaveamento por pacote, adotando a técnica *wormhole*, e faz uso de roteamento na origem denominado *street sign routing*. Neste tipo de roteamento, o cabeçalho do pacote contém a identificação dos roteadores em que se fará alguma mudança de direção e a direção que tomará. Adicionalmente, a NoC Xpipes permite parametrizações em tempo de projeto tais como: tamanho do *flit*, espaço de endereçamento dos núcleos, número máximo de roteadores entre dois núcleos, número máximo de bits para controle de fluxo fim-a-fim, profundidade da fila de armazenamento e número de canais virtuais por canal físico. Os autores também exploram a geração automática da XPipes, através da adoção de um fluxo de projeto que inclui a extração das características de comunicação da aplicação alvo.

**Goossens et al.** [GOO05] propuseram a rede em chip *Æthereal*. A *Æthereal* pode assumir diferentes topologias, regulares ou irregulares, de acordo com o interesse do projeto. A transmissão de pacotes na rede é garantida pelo uso de roteamento na origem. Nesta NoC são oferecidos serviços diferenciados com conexão e com nível de serviço associado. A aceitação da conexão pode incluir a reserva de recursos na rede, tal como filas de armazenamento e/ou percentual de largura de banda em canais. Finalizada a comunicação entre o par origem/destino, a conexão é eliminada e as reservas são desfeitas, permitindo assim a criação de novas conexões. O estabelecimento de conexão entre pares comunicantes propicia o atendimento de requisitos tal como a taxa de transmissão. Todavia, a reserva de caminhos consequente do estabelecimento da conexão pode acarretar congestionamento e possível dificuldade de cumprimento de requisitos de outros serviços da rede.



## CONSIDERAÇÕES FINAIS SOBRE NOCS IRREGULARES

O uso de topologias irregulares de NoCs para o projeto de SoCs também garante a comunicação paralela entre núcleos IP, adicionalmente potencializando otimizações de área e de dissipação de potência, quando comparado a topologias regulares. Em todos os casos revisados, a aplicação e seus requisitos comandam as definições arquiteturais da infraestrutura de comunicação. As diferentes abordagens adotadas para a geração das infraestruturas de comunicação investigadas são apresentadas na Tabela 2.1.

Tabela 2.1 – Comparativo das NoCs com topologia irregular investigadas.

	Topologia base	Abordagem	Objetivo
Dumitras et al. [DUM04]	Não há	Mescla infraestruturas e tecnologias	Garantir melhor infraestrutura de comunicação por aplicação
Richardson et al. [RIC06]	Malha 2D	Barramento + NoC	Reduzir o número de roteadores, explorando a vantagem do uso de barramentos
Bolotin et al. [BOL04]	Malha 2D	Eliminação de recursos dispensáveis	Viabilizar a geração de uma infraestrutura de comunicação com o poder de atendimento de diferentes níveis de serviço
Ogras et al. [OGR06a]	Malha 2D	Inserção de fios longos	Diminuição da latência da transmissão dos pacotes
Srinivasan et al. [SIR05]	Não há	Fluxo automático	Prover solução completa para geração de de um SoC
Bertozzi et al. [BER04]	Não há	Fluxo automático	Prover solução completa, centrando na infraestrutura de comunicação
Goossens et al. [GOO05]	Não há	Fluxo automático	Disponibilizar uma infraestrutura de comunicação personalizada com garantias de desempenho

## 2.2 CONGESTIONAMENTO

O termo congestionamento corresponde ao bloqueio, total ou parcial, de recursos de uma infraestrutura de comunicação, tal como enlaces ou meios de armazenamento temporário. A presente Seção tem por objetivo revisar algumas das definições relacionadas a congestionamento encontradas na literatura de infraestrutura de comunicação.

**Yang e Reddy** [YAN95] definem congestionamento em redes de chaveamento por pacotes como um estado no qual o desempenho da infraestrutura de comunicação degrada devido à saturação dos recursos da rede, tal como canais de comunicação e meios de armazenamento. Os autores citam como efeitos adversos resultantes do congestionamento o atraso na entrega de mensagens, o desperdício de recursos do sistema e um possível colapso na rede caso toda a comunicação seja impossibilitada (tal como no caso de ocorrência de *deadlocks*).

**Monteiro et al** [MON96] definem que um sistema de comunicação está congestionado quando o funcionamento dos serviços for afetado de tal forma que possa ser percebido pelo usuário. Ainda segundo estes autores, congestionamento é um fenômeno associado a todos os sistemas de comunicação de geometria variável, ou seja, sistemas de comunicação onde haja um número dinâmico de pares comunicantes (i.e. pares origem/destino), onde características de tráfego podem mudar, ou onde recursos de comunicação disponíveis não são constantes. De forma simplificada, a ocorrência de congestionamento em sistemas de comunicação pode ser identificada quando o fluxo de entrada de dados no sistema, em um intervalo fixo de tempo, é maior do que o fluxo de saída no mesmo intervalo, levando a um estado de sobrecarga da rede.

**Culler e Singh** [CUL99] expõem que congestionamento pode ter duas causas, quais sejam, (i) meio ou (ii) fim. O congestionamento causado pelo meio ocorre quando se ultrapassa a capacidade de atendimento de transmissão de dados por alguns recursos de rede, devido à competição de fluxos de dados pelos recursos de comunicação. Um dos reflexos deste tipo de congestionamento é o atraso da transmissão de dados pela rede, que pode ser reduzido pelo mapeamento de processos e o escalonamento da comunicação de forma apropriada à topologia de rede específica. O congestionamento causado pelo fim da comunicação pode ocorrer, por exemplo, quando vários PEs necessitam comunicar-se com um mesmo nodo ao mesmo tempo, ou ainda quando há discrepância entre as frequências de operação de elementos comunicantes. Em ambos os casos, o alvo da comunicação não consegue consumir a quantidade de dados a ele enviados, o que leva ao potencial bloqueio de recursos do dispositivo de comunicação.

## CONSIDERAÇÕES FINAIS SOBRE CONGESTIONAMENTO

Em redes de computadores, uma das formas de detectar a ocorrência de congestionamento é pela observação da ocorrência de perda de pacotes, causada pela expiração de um tempo de vida associado a cada pacote (em inglês *Time To Live*, TTL) no trânsito entre a origem e o destino. A perda de pacotes ocorre, pois o congestionamento prolonga a permanência de pacotes na rede, o que pode levar a superação de um tempo previsto de entrega do mesmo, ocasionando sua eliminação na rede. O não recebimento de pacotes pelo destino pode ocasionar a transmissão de pacotes de controle, do destino para a origem, contendo requisições de retransmissão ou sincronizações. Em algumas NoCs adota-se o conceito de TTL [DUM03], mas seu uso ainda é pouco usual. Dois fatores contribuem para isto, quais sejam, (i) requisitos de limitação de ocupação de área por

redes em chip e (ii) a busca pela localidade da comunicação entre pares origem/destino. Todavia, o avanço deste tipo de infraestrutura de comunicação, com redes de dimensões cada vez maiores, bem como o uso de mapeamento dinâmico de tarefas pode tornar mandatória a adoção de TTL em NoCs.

Neste trabalho, a rede é dita congestionada quando um fluxo de pacotes, trafegando por uma determinada infraestrutura de comunicação, tem o seu tempo ideal de entrega superado em decorrência de competição por recurso(s) compartilhado(s) com outro(s) fluxo(s) de pacote(s), impossibilitando o atendimento de requisitos tal como a taxa de transmissão contratada entre pares comunicantes. Define-se *tempo ideal de entrega* como o tempo mínimo necessário para a entrega de um pacote em um cenário onde não há concorrência de utilização por nenhum dos recursos de comunicação envolvidos na transmissão do pacote. Outra definição relevante é a de *taxa de transmissão contratada*, o valor que descreve a frequência média com que uma dada quantidade de bits tem de estar disponível em seu destino, independente da distribuição temporal que a entrega destes dados venha a ter, usando distribuição uniforme de tráfego ou não. Sabe-se que quanto maior a diferença entre o tempo ideal e o tempo real de transmissão dos pacotes, maior a probabilidade de ocorrência de congestionamento.

O controle da degradação do desempenho da comunicação é o foco deste trabalho. Isto é explorado de duas formas: (i) a disponibilização de infraestruturas de comunicação em chip e (ii) os meios de uso de tais recursos em tempo de projeto ou execução. As distintas infraestruturas de comunicação e meios de uso são apresentadas nos Capítulos 3 e 5, focando respectivamente decisões a serem tomadas em tempo de projeto e tempo de execução respectivamente.

### 2.2.1 PONTOS QUENTES

O termo ponto quente (em inglês, *hotspot*) é usado em diferentes etapas de projeto e teste de semicondutores. Por exemplo, durante a fabricação de um CI, um ponto quente ocorre quando o descumprimento de algumas regras de projeto leva à criação de regiões críticas de atividade de chaveamento, o que pode por em risco a robustez do CI à ocorrência de falhas de operação [LIM08]. No presente trabalho, o foco limita-se à etapa de uso de CIs. Com o aumento do número de núcleos IP por CI, o planejamento inadequado de uso de uma arquitetura alvo pode acarretar uma maior competição por recursos, o aumento da dissipação de potência e da temperatura além

de limites aceitáveis [LIN06] [SCH08]. A detecção de extremos de atividade, dissipação de potência e aumento de temperatura durante o uso de um circuito também são algumas das características consideradas [LIN05] para definir o conceito de ponto quente.

**Link et al.** [LIN05] propõem a migração de tarefas como mecanismo para eliminar a ocorrência de pontos quentes em sistemas multiprocessados baseados em NoCs. Para estes autores, ponto quente está relacionado com a temperatura no CI, e não necessariamente com a idéia de congestionamento. A migração das tarefas é realizada através da reconfiguração dinâmica parcial. Como resultado, a adoção da técnica de reconfiguração proposta aplicada sobre estudos de caso obteve reduções de até 8°C na temperatura de pico do CI em comparação com uma distribuição estática da computação.

**Banerjee et al.** [BAN01] propõem a adoção de projeto 3D de CIs como forma de aumentar o desempenho das infraestruturas de comunicação e facilitar a integração em SoCs. Segundo os autores, assim como proposto em [DUM04] para o projeto de NoCs de arquitetura e tecnologia mistas, é benéfica a integração de diferentes tecnologias em um mesmo CI. Apesar disto, **Addo-Quaye** [ADD05] assinala que a integração de NoCs e CIs 3D pode sofrer com o aparecimento de pontos quentes, devido à concentração de potência em determinados pontos no CI, acarretando a diminuição da confiabilidade nas interconexões, que conforme o autor representa uma suscetibilidade de falha de cerca de 5% a cada 10°C de elevação. Como solução, Addo-Quaye propõe a adoção de algoritmos genéticos para o mapeamento da comunicação e da computação.

## CONSIDERAÇÕES FINAIS SOBRE PONTOS QUENTES

No presente trabalho, *ponto quente* é a ocorrência de sobrecarga de atividades de transmissão nos canais de uma infraestrutura de comunicação, tornando-se um gargalo na comunicação entre núcleos IP distintos. Pontos quentes são considerados como um fator local de possível deflagração de congestionamento na rede.

## 2.3 CONTROLE DE FLUXO DE COMUNICAÇÃO

A presente Seção tem por objetivo apresentar um conjunto de trabalhos relacionados com o controle de fluxo de comunicação em sistemas que empregam redes. Inicia-se com um levantamento de trabalhos que usam uma proposta de solução para o problema de mapeamento de tare-

fas como meio de controlar o fluxo de informações no interior de uma infraestrutura de comunicação. Segue-se um estudo de trabalhos que propõem estratégias de adaptação local de uma rede de comunicação com o fim de controlar o fluxo de dados. Em oposição a esta abordagem, estuda-se a seguir trabalhos que sugerem abordagens de adaptação via mecanismos globais à rede. A Seção conclui com um levantamento de trabalhos sobre o tema relacionado relevante, o emprego de monitores em redes em chip.

### 2.3.1 MAPEAMENTO DE TAREFAS

O mapeamento de tarefas em plataformas multiprocessadas influencia a distribuição da carga de comunicação que ocorre em uma NoC. Esta Seção revisa alguns trabalhos relacionados ao mapeamento de tarefas que consideram uma infraestrutura de comunicação subjacente.

**Marcon et al.** [MAR05] propõem o mapeamento de módulos de aplicações sobre NoCs. O trabalho é composto de quatro elementos: (i) um modelo de comunicação, (ii) um algoritmo de busca de solução, (iii) um modelo da infraestrutura de comunicação alvo e (iv) uma função custo. Usam-se diferentes modelos abstratos de comunicação para descrever os aspectos de comunicação das tarefas que compõem uma dada aplicação. Para a busca da solução, propõe-se algoritmos exaustivos, genéricos (e.g. *Simulated Annealing*, *Tabu Search*) e algumas heurísticas. O modelo da infraestrutura de comunicação alvo é uma NoC, porém outras estruturas podem ser empregadas. A função custo empregada no trabalho objetivava reduzir o consumo de energia causado pelas comunicações na rede.

**Zhou et al.** [ZHO06] propõem o mapeamento de tarefas de uma dada aplicação como mecanismo de controle de congestionamento em sistemas baseados em NoCs. Utilizando grafos de caracterização de aplicação quanto à comunicação, grafos de descrição da infraestrutura de comunicação e algoritmos genéticos para a busca de um bom mapeamento, os autores melhoram o desempenho da comunicação. Aplicando o mapeamento proposto sobre cenários compostos de diferentes NoCs e geração randômica de tráfego, os autores reduzem os tempos de execução em média 20% quando comparado a resultados obtidos com mapeamentos aleatórios.

**Wenbiao et al.** [WEN07] propõem uma abordagem combinada de mapeamento de núcleos IP sobre uma NoC e a definição das rotas a serem adotadas pelos pares comunicantes, objetivando

a redução do consumo de potência e equalizando a carga de ocupação dos canais. Segundo os autores, a escolha de rotas pode afetar parâmetros de desempenho tal como largura de banda, latência e utilização de recursos. Os autores utilizam um algoritmo de otimização de agrupamento de partículas (em inglês, *particle swarm optimization*) para computação do mapeamento dos PEs e das rotas de comunicação. Os resultados apontam reduções de 20% no consumo de energia após o mapeamento e redução de carga nos canais de 13% posteriormente à escolha das rotas.

O mapeamento de aplicações busca atender um ou mais objetivos, conforme os trabalhos revisados. Quando realizado em tempo de projeto, como nos casos apresentados, o conhecimento das características de computação e comunicação da aplicação é de suma importância para garantir o melhor mapeamento possível. Porém, quando não há um conhecimento prévio sobre o comportamento das aplicações que serão executadas, o uso de mecanismos de mapeamento dinâmico torna-se essencial para garantir o melhor aproveitamento da plataforma, conforme salientado nos trabalhos revisados a seguir.

**Carvalho et al.** [CAR07] propõem o mapeamento dinâmico de tarefas de uma aplicação levando em consideração o estado de utilização de uma rede em chip. Neste trabalho, três elementos são utilizados para a exploração do mapeamento das tarefas: (i) um grafo de descrição dos requisitos de comunicação de cada tarefa da aplicação, (ii) uma unidade de gerenciamento de mapeamento das tarefas sobre a rede em chip e (iii) um conjunto de algoritmos de mapeamento. O objetivo é buscar o mapeamento de tarefas em pontos da rede onde a comunicação será menos afetada, tendo em vista uma rede malha com roteamento XY. Um conjunto de outras propostas de mapeamento dinâmico de tarefas além de comparações entre abordagens dinâmicas e estáticas de mapeamento foram exploradas por Carvalho em [CAR09].

## CONSIDERAÇÕES FINAIS SOBRE MAPEAMENTO DE APLICAÇÕES

No presente trabalho, a forma como será definido o mapeamento de tarefas não é essencial, podendo este ser realizado em tempo de projeto ou em tempo de execução. O objetivo é prover suporte para que a infraestrutura de comunicação possibilite o atendimento de requisitos de comunicação dinamicamente, de forma a garantir a execução de aplicações com atendimento à qualidade de serviço requisitada.

### 2.3.2 ADAPTAÇÃO LOCAL

Em redes de comunicação, uma das formas de garantir o atendimento a serviços ou melhorar o uso dos recursos de comunicação vem da possibilidade de adaptações dos recursos de rede durante o uso. A presente Seção revisa alguns trabalhos que versam sobre tais adaptações.

**Vishnu et al.** [VIS07] propõem o uso de múltiplos caminhos na comunicação entre pares comunicantes como meio de garantir requisitos de desempenho. Os autores exploram esta proposta usando uma infraestrutura de comunicação de alta velocidade chamada InfiniBand [INF09]. Assim como as redes Myrinet [MYR09], InfiniBand foi desenvolvida para ambientes de computação de alto desempenho (em inglês, *High Speed Computing*, HSC) que incluem máquinas agregadas (em inglês, *clusters*) e as grades computacionais (em inglês *grids*). Originalmente, permitia-se apenas um caminho de comunicação entre um par comunicante na InfiniBand. Esta característica foi estendida pelos autores que exploram as vantagens do uso de múltiplos caminhos. O objetivo é a eliminação do potencial de geração de pontos quentes. A idéia de uso de múltiplos caminhos foi assumida em NoCs por Murali et al. [MUR07] e Carara et al. [CAR08]. A preocupação de Murali et al. é a entrega ordenada de pacotes no destino em situações onde múltiplos caminhos existem. As várias rotas entre cada par de entidades comunicantes são definidas em tempo de projeto. O uso de múltiplos caminhos de comunicação é considerado como uma boa solução para adequação de requisitos de comunicação. Todavia, o uso de recursos de adaptação em redes deve levar em consideração o estado de ocupação dos recursos de comunicação.

**Kim et al.** [KIM05] propõem uma arquitetura de roteador de baixa latência e com suporte a adaptação no roteamento, criando critérios de seleção de encaminhamento de pacotes a partir de cada roteador (em inglês, *output selection*). A baixa latência é alcançada através do emprego de uma arquitetura de roteador de pacotes com 2 estágios de *pipeline*. No primeiro estágio é (i) tomada a decisão de roteamento, baseado na observação dos roteadores vizinhos, (ii) feita a pré-seleção do melhor canal que o pacote deve ser encaminhado e (iii) alocado o canal virtual. No segundo estágio, o cabeçalho é transmitido através do roteador. A adaptação é obtida a partir do uso de uma unidade de controle em *hardware*, que permite a adaptação de caminhos em uma topologia 2D. A arquitetura proposta pelos autores foi avaliada sob diferentes padrões de tráfego e observou-se que sob tráfego não uniforme alcançou-se redução da latência quando comparado à mesma arquitetura adotando algoritmos de roteamento determinísticos.

**Nilsson et al.** [NIL03] investigam o desvio de pacotes durante seu caminhamento em redes com algoritmos de roteamento *batata quente* (em inglês, *Hot-Potato*, HP), tal como a NoC Nostrum [MIL04]. Infraestruturas de comunicação que empregam roteamento HP eliminam o uso de filas de armazenamento, recurso este utilizado com o objetivo de amortizar a ocorrência de congestionamento em redes que empregam chaveamento por pacote do tipo *wormhole*. Os autores abordam três características para a decisão de desvio: (i) desconsiderar o congestionamento na rede, (ii) considerar o congestionamento ciclo a ciclo e (iii) considerar o congestionamento através de uma média dentro de uma janela de observação. Uma janela de observação é definida pelos autores como um número preestabelecido de ciclos de relógio (e.g. 1 janela de observação = 1000 ciclos de relógio). A informação de congestionamento é compartilhada entre roteadores vizinhos e utilizada na decisão da porta de saída de um pacote. Resultados apresentam uma melhor distribuição das cargas de comunicação na abordagem (iii) de 6 vezes melhor que à abordagem (ii) e 20 vezes superior à (i).

**Daneshtalab et al.** [DAN06a] propuseram o uso de algoritmos de roteamento adaptativos que consideram o estado das portas de entrada e saída de cada roteador, objetivando reduzir a ocorrência de pontos quentes. Dois mecanismos estão presentes em cada roteador para dar suporte a roteamento adaptativo. O primeiro mecanismo avalia a condição de fila cheia, dada pela taxa de recepção de pacotes e pela taxa de repasse destes (i.e. *input selection*). O estado da fila de armazenamento é informado ao roteador vizinho conectado à porta e, caso a indicação de fila cheia seja confirmada, este roteador vizinho evita enviar pacotes a este roteador, preferindo outras rotas. O segundo mecanismo é o repasse de informação de congestionamento para os roteadores vizinhos, que darão prioridade de atendimento a pacotes oriundos do roteador congestionado (i.e. *output selection*). Os autores desenvolveram um modelo C++ que permite estimar o atraso médio da comunicação. Adicionalmente, confrontaram este modelo com um modelo da NoC DyAD [HU04], de uma NoC genérica empregando roteamento XY e outra com roteamento Odd-Even.

**Al Faruque et al.** [ALF03] propõem o emprego da adNoC, uma infraestrutura de comunicação com poder de adaptação levando em consideração os requisitos de comunicação de pares de tarefas e a quantidade de largura de banda disponível nos canais. Os autores apresentam um fluxo de emprego da adNoC, onde as aplicações são divididas em tarefas, e as tarefas são mapeadas sobre PEs, os quais são mapeados sobre blocos disponíveis na rede. Em tempo de execução adapta-se a comunicação através da alocação de canais virtuais e a atribuição destes às portas de saída. As



portas de saída são selecionadas de acordo com sua disponibilidade (percentual da largura de banda ainda livre), mas considerando a ocupação prevista de largura de banda para a comunicação e a distância entre o roteador corrente e o destino da comunicação. Isto é obtido através da execução de um algoritmo chamado wXY.

**Shin e Daniel** [SHI96] propõem a adaptação do modo de chaveamento a ser empregado na rede em tempo de execução, sendo opções *wormhole* (WH) e *virtual cut-through* (VCT). Segundo avaliação dos autores, VCT proporciona maior vazão e menor latência quando a carga de tráfego na rede é alta (i. e. superior a 30% de injeção de tráfego) quando comparado a WH. Sob tráfego moderado (i.e. até 30% de injeção de tráfego, WH permite o emprego de filas de armazenamento de tamanho menor e desempenho equivalente a redes VCT. A decisão sobre qual tipo de chaveamento usar está vinculada aos pacotes e não se baseia no estado de ocupação da rede. Cada pacote carrega em seu cabeçalho um campo (H) que informa por quantos *hops* a rede deve oferecer-lhe chaveamento WH, sendo que inicialmente o deslocamento ocorre por VCT. O valor contido neste campo é decrementado *hop a hop*, e ao atingir-se zero assume-se chaveamento VCT para este pacote.

**Nguyen et al.** [NGU06] avaliaram o impacto do uso de algoritmos de roteamento determinísticos e adaptativos em NoCs. Sob tráfego de baixa intensidade, onde não há detecção de congestionamento ou sua existência atrapalha o atendimento de serviços, algoritmos determinísticos permitem alcançar melhor desempenho. Na presença de tráfego intenso, algoritmos adaptativos contribuem para a diminuição da latência.

**Hu e Marculescu** [HU04] propuseram uma rede em chip que combina roteamento adaptativo e roteamento determinístico chamada DyAD (do inglês, *Dynamic Adaptive Deterministic switching*, DyAD). Esta infraestrutura de comunicação combina algoritmos garantidamente livres de ocorrência de dependência cíclica (em inglês, *deadlock*) e/ou de deslocamentos por tempo indefinido (em inglês, *livelock*) na rede. Os autores assumem XY como o algoritmo de roteamento determinístico e *Odd-Even* (OE) como o algoritmo de roteamento adaptativo mínimo. OS autores citam que a abordagem dá suporte a outros algoritmos adaptativos, desde que a regra de mistura dos algoritmos adaptativos e determinísticos possa garantir que *deadlock* e *livelock* não ocorrem. Congestionamento é detectado sempre que a quantidade de posições ocupadas em uma fila de armazenamento exceder um valor limiar pré-estabelecido em tempo de projeto. Quando um rote-

ador detecta a ocorrência de congestionamento, seus vizinhos são notificados e todos assumem roteamento adaptativo até que a notificação de congestionamento se desfaça. Na ausência de notificação de congestionamento, o algoritmo de roteamento determinístico impera.

**Dehyadgari et al.** [DEH05] propuseram abordagem similar àquela de Hu e Marculescu [HU04], sugerindo o uso de um algoritmo de roteamento determinístico (XY) quando a rede não encontra-se congestionada. Na presença de congestionamento, assume-se um algoritmo de roteamento adaptativo (pseudo-XY). O algoritmo pseudo-XY é uma variação desenvolvida pelos autores com poder de adaptação e que garantidamente não causa *deadlock* quando usado juntamente com o algoritmo XY. Não há maiores detalhes sobre o funcionamento do algoritmo, exceto que as adaptações assumidas com o pseudo-XY estão fortemente baseadas no congestionamento da rede. O congestionamento é avaliado sobre o percentual de ocupação das filas de armazenamento, havendo indicação de 4 situações, quais sejam: (i) baixa ocupação, (ii) ocupação a 50%, (iii) ocupação a 75% ou (iv) limite de ocupação alcançado. Os autores desenvolveram um ambiente para simulação baseado em SystemC para avaliar o desempenho da infraestrutura de comunicação proposta.

**Sobhani et al.** [SOB06] propõem o uso de um algoritmo de roteamento adaptativo em suas formas mínima e não mínima na mesma infraestrutura de comunicação, dependendo do estado da rede. Por padrão, o transporte de pacotes ocorre utilizando o algoritmo de roteamento mínimo. Se ao chegar a um roteador e a porta definida como saída for considerada congestionada então a porta de saída é definida usando o algoritmo não mínimo. Em um roteador, o congestionamento é definido quando a quantidade de posições ocupadas em uma fila de armazenamento é superior a um dado limite e a quantidade de *flits* encaminhados para uma dada porta de saída é inferior à quantidade de *flits* que são recebidos na porta de entrada do mesmo roteador. Os autores desenvolveram um modelo simulável em C++, sendo que os valores obtidos são resultado de uma estimativa da latência média.

## CONSIDERAÇÕES FINAIS SOBRE ADAPTAÇÃO LOCAL

O uso de mais de uma alternativa de caminhos para alcançar um dado destino em infraestruturas de comunicação pode ser uma boa solução em situações onde se requer uma maior taxa de transmissão ou uma melhor distribuição da carga de comunicação. Diferentes abordagens, que

vão desde a exploração de múltiplos caminhos, avaliação do estado de ocupação dos recursos de comunicação roteador a roteador e a mistura de características contribuem para o melhor uso da rede. No presente trabalho, mecanismos de balanceamento de carga são explorados em tempo de projeto e em tempo de execução. Em tempo de projeto, a decisão de qual caminho utilizar na comunicação entre duas entidades para todos os pares comunicantes tem por objetivo controlar a ocorrência de pontos quentes na rede. Em tempo de execução, disponibilizam-se contratos de serviço que permitam a exploração de mais de um caminho de comunicação entre pares comunicantes. Em ambos os casos, o objetivo é a tomada de decisão baseada no conhecimento do estado da rede, ou pelo menos de regiões de interesse desta.

### 2.3.3 ADAPTAÇÃO GLOBAL

Diferentemente da visão dos mecanismos de adaptação em nível de enlace, o uso de mecanismos fim a fim propõe a observação global do sistema para a tomada de decisão. Diferentes formas de adaptação fim a fim podem ser assumidas, tais como a adequação das taxas de injeção de tráfego e a adaptação de caminhos em infraestruturas de comunicação. Na presente Seção, revisa-se alguns dos trabalhos envolvendo controle fim a fim aplicados a sistemas baseados em NoCs.

**Pastrnak et al.** [PAS06] propõem uma infraestrutura hierárquica de qualidade de serviço, voltada para aplicações que serão executadas em uma estrutura de multiprocessador baseada em NoC. Para tanto, as aplicações são descritas a partir de um conjunto de *Jobs*. Cada *Job* é representado por um conjunto de tarefas comunicantes e pelos requisitos de computação e comunicação que serão necessários para sua execução. A QoS é avaliada em dois níveis, um local e outro global. Sempre que um *job* for escalonado, um módulo local avalia se os recursos necessários para garantir a QoS deste *Job* estão disponíveis em um PE para o qual este foi mapeado. Se isto ocorrer, o sistema aloca o recurso. Do contrário, o módulo de QoS local pede intervenção de um módulo de QoS global. Este último modifica os mapeamentos e/ou redefine as prioridades de acesso a alguns recursos, a fim de garantir que todas as aplicações em execução mantenham seus requisitos definidos em tempo de compilação.

**Jafari e Yaghmaee** [JAF08] exploram a adoção de um mecanismo de controle de fluxo usando um algoritmo de justiça max-min com pesos (em inglês, *weighted max-min fairness*). Um algoritmo é dito *max-min fair* [DAL03] se e somente se, dado um vetor de taxas de comunicação  $X$ , o

aumento de uma taxa  $x_a$  não leva à diminuição de uma taxa  $x_b$ , tal que  $x_b \leq x_a$ . Um cenário de compartilhamento justo é apresentado na Figura 2.6(a) abaixo, enquanto um compartilhamento max-min fair é apresentado na Figura 2.6(b). O algoritmo de peso max-min justo tem o mesmo princípio, porém assumindo que  $x_b/w_b \leq x_a/w_a$ , onde  $w^*$  é uma unidade de peso específica para cada taxa  $x^*$ . O modelo de comunicação proposto é similar ao definido por [VAN07], onde as ocupações dos canais são capturadas por monitores distribuídos pela NoC e a tomada de decisão é feita por uma unidade de controle centralizada.

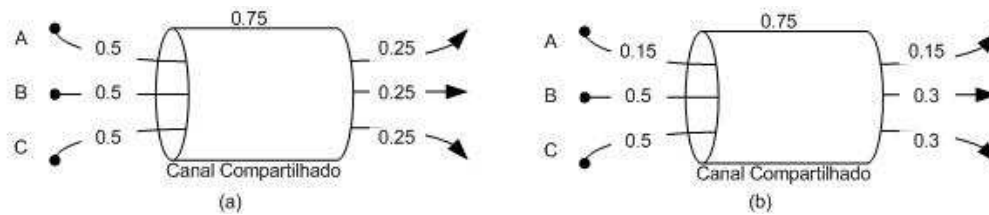


Figura 2.6 – Associação de três fluxos compartilhando um canal. Em (a) mostra-se a alocação quando os fluxos têm mesma taxa de requisição, enquanto que em (b) mostra-se uma alocação max-min justa para taxas com requisitos diferentes.

**Koundinya et al.** [KOU04] propõem algoritmos de redefinição de rotas em redes com roteamento na origem, visando o atendimento de múltiplos parâmetros de QoS. A busca pelo melhor caminho é realizada em duas etapas. A primeira é a fase de busca de caminhos e a segunda é a de confirmação da rota, onde o recurso alvo é reservado. Os resultados obtidos a partir do algoritmo são comparados ao algoritmo de Dijkstra, um algoritmo de base para a busca de melhores caminhos em redes.

**Radulescu et al.** [RAD05] propõem um projeto de interface de rede (em inglês, *network interface*, NI) que dá suporte a garantia de serviços, abstração de memória compartilhada e configuração da rede para projetos de SoCs baseados em NoCs. Para tanto, a NI é projetada de forma modular, tendo por base um modelo de transações no qual estão apoiados protocolos de barramentos, tais como AXI [ARM09], DTL [PHI09] e OCP [OCP09]. Cada um destes protocolos provê suporte a requisições de leitura e escrita entre mestre e escravo em uma comunicação. A NI é dividida em dois módulos, um NI Kernel e um NI Shell. O primeiro tem por objetivo garantir serviços e abstrações de acesso a memória. O segundo permite a criação de mecanismos de conexão em grupos, de um mestre para um conjunto de escravos. As configurações propostas no artigo referem-se à abertura e fechamento de conexões no sistema antes da execução de uma aplicação. A esta configuração está associada a alocação de recursos de comunicação, tais como fatias de tempo atribuídas a cada conexão em cada canal de comunicação (escalonamento da comunicação) e alocação da

quantidade de fila de armazenamento. O trabalho usa a infraestrutura *Æthereal* de comunicação, que provê QoS através da disponibilização de serviços garantido (vazão e latência garantidas através de multiplexação por divisão temporal nos canais) e melhor esforço.

**Nousias e Arslan** [NOU06] propõem uma infraestrutura de comunicação baseada em chaveamento *wormhole*, utilizando canais virtuais, que controla a carga dos canais de comunicação de forma descentralizados através do ajuste das taxas de injeção de tráfego possíveis a partir de cada origem. A tomada de decisão do ajuste da taxa de injeção de pacotes na rede é feita a partir da observação das filas de armazenamento presentes na infraestrutura de comunicação, sendo este um processo realizado de forma descentralizada. Busca-se localizar ocorrência de contenção, o que pode deflagrar o congestionamento na rede. Detectada a ocorrência de contenção, origens de comunicação, injetoras de tráfego, receberão pacotes de sinalização solicitando que suas taxas de injeção de pacotes sejam reduzidas.

**Ogras e Marculescu** [OGR06b] propõem um mecanismo de controle de fluxo do tipo laço fechado (em inglês, *closed-loop*), baseado em um algoritmo de predição de situações de congestionamento. A predição é feita através da avaliação do estado dos roteadores. Cada roteador tem o poder de avaliar sua capacidade de atendimento a pacotes que estão chegando, através da medição da ocupação média de sua fila de armazenamento e do histórico de taxa de chegada dos pacotes. Ocupações médias acima de um dado valor limiar (e. g. 80% do *buffer* cheio) e histórico de altas taxas de recepção de pacotes podem revelar a dificuldade de repasse dos pacotes e a possível ocorrência de congestionamento. Esta detecção de potencial congestionamento é repassada aos roteadores vizinhos, gerando uma onda de propagação desta informação. Ao receberem o alerta de congestionamento, as origens de comunicação, geradoras do tráfego, podem regular as taxas com que injetam pacotes.

**Kumar et al.** [KUM08] propuseram um esquema de controle de fluxo baseado em *tokens* para a definição dos melhores caminhos de comunicação. *Tokens* são indicações de disponibilidade na rede. Cada nodo na rede envia *tokens* para uma região limitada definida por um dado número máximo de *hops*. Os *tokens* contêm informações de disponibilidade de recursos tais como fila de armazenamento e canais virtuais em uma dada porta de entrada. Durante o roteamento, os pacotes usam individualmente as informações disseminadas pelos *tokens*, a fim de buscar a rota menos congestionada dentro de uma região. Esta abordagem descentraliza a tomada de decisão das ro-

tas, permitindo que uma região pertencente ao caminho seja avaliada.

**Gratz et al.** [GRA08] propõem uma infraestrutura de balanceamento de carga da comunicação em tempo de execução na infraestrutura de transporte, levando em consideração informações de contenção na rede por regiões (em inglês, *Regional Congestion Awareness*, RCA). Os autores utilizam como métrica de contenção a quantidade de requisições de transmissão feitas pelas portas de saída de cada roteador. Outras duas métricas também foram avaliadas por estes, a disponibilidade de canais virtuais e a ocupação de fila de armazenamento. Contudo, a primeira foi a que apresentou melhores resultados. Cada roteador coleta informações de ocupação e transmite aos roteadores vizinhos. Experimentaram-se três abordagens: (i) a RCA 1D, (ii) a RCA *fanin* e (iii) a RCA quadrante. Na primeira, os valores são disseminados por apenas um dos eixos da rede. Na segunda, a transmissão é equivalente àquela 1D, diferenciando-se por agregar informações dos roteadores vizinhos ao roteador corrente. Na terceira abordagem a informação é transmitida por quadrantes. O tráfego destes pacotes se dá por uma rede distinta da de dados e contribui para a tomada de decisão da porta de saída. Resultados mostram ganho em todas as abordagens que utilizam informação global de congestionamento da rede quando comparado a abordagens determinísticas, ou que utilizam o estado local de cada roteador. Das três abordagens de disseminação da informação de congestionamento, a RCA quadrante foi a que permitiu melhor desempenho da infraestrutura de comunicação.

## CONSIDERAÇÕES FINAIS SOBRE MECANISMOS DE ADAPTAÇÃO GLOBAL

No presente trabalho, será utilizado roteamento na origem, o que requer estabelecer uma função custo para a computação de rotas que garanta a menor latência para comunicação entre pares origem/destino. A adoção de múltiplas rotas entre pares comunicantes é uma possível solução dentro do escopo deste trabalho.

### 2.3.4 EMPREGO DE MONITORES

O emprego de monitores em redes em chip tem múltiplos objetivos, incluindo o apoio ao teste do circuito, a depuração do projeto, a coleta de informações para auxiliar na avaliação de uso de recursos e também na coleta de informações com o objetivo de adaptação do uso da infraestrutura de comunicação alvo. A seguir revisam-se alguns trabalhos relacionados ao monitoramento de redes em chip.

**Ciordas et al.** [CIO06] propõem uma infraestrutura de comunicação integrada com recursos de monitoramento de transações entre núcleos IPs para observação em tempo de execução. Considera-se transação os pedidos de leitura e escrita, bem como as respostas obtidas e as confirmações de atendimento da operação, realizados nas interfaces de rede da infraestrutura de comunicação. O objetivo deste sistema é contribuir para a depuração da comunicação que ocorre no sistema, a partir da reconstrução das transações para observação do que ocorre na infraestrutura de comunicação. Implementou-se um sistema composto de três partes: um “farejador” (em inglês *sniffer*), um gerador de eventos e uma interface de monitoramento de rede. O primeiro é responsável pela observação de transmissões *flit a flit* na rede, enquanto o segundo contabiliza as transações. A última é responsável pelo empacotamento e envio dos valores obtidos em direção a um ponto de acesso ao serviço de monitoramento.

**Mouhoub e Hammami** [MOU06] propõem uma plataforma de monitoramento para dar suporte a que projetistas de *software* avaliem o quão eficiente é o uso dos recursos comunicação. Isto é obtido através da coleta de informações de ocupação de filas de armazenamento e de latência de transmissão fim a fim de pacotes. Os valores são coletados de forma não intrusiva e transmitidos por um meio não compartilhado com a aplicação sob avaliação. O objetivo é evitar a influência que o tráfego de dados de monitoramento possa causar na comunicação da aplicação. O trabalho foi validado em uma NoC com dimensões 2x2, tendo *Microblaze* como PE.

**Shobaki** [SHO02] propõe uma arquitetura de monitoramento não intrusivo para sistemas operacionais mono e multiprocessados de tempo real. A arquitetura baseia-se no emprego de núcleos IP reconfiguráveis que detectam a ocorrência de eventos e coletam informações. Exemplo de tais eventos são a execução de processos, sincronização e atividades de entrada e saída. O trabalho desenvolvido pelos autores é dito não intrusivo, pois se baseia na observação do barramento que interconecta os elementos e processadores. À ocorrência de eventos são adicionadas informações temporais (i. e. o momento em que ocorreram), as quais são transmitidas para um sistema externo à arquitetura de monitoramento, que contém uma base de dados de eventos e ferramentas de monitoramento de aplicações.

**Tang e Xu** [TAN08] propõem uma plataforma de depuração para sistemas multiprocessados com comunicação baseada em NoCs. Tal plataforma permite a depuração concorrente de diferentes PEs através da observação de transações que ocorrem pela rede. A plataforma de depuração é

composta de: (i) um *software* de suporte a depuração, (ii) um controlador de depuração e (iii) uma arquitetura de depuração em chip. O primeiro permite o envio de comandos de depuração e a visualização de resultados através de uma interface gráfica ou linha de comando. O segundo funciona como um tradutor, capturando os comandos gerados pelo *software*, inserindo estes na plataforma de depuração através de uma interface JTAG. O terceiro é o mecanismo de depuração em chip, sendo posicionado entre os elementos sob depuração e sua interface de rede. Os autores assumem o uso de interfaces padronizadas (e. g. OCP-IP e AXI), extraíndo dali as transações que ocorrem no sistema multiprocessado. Um agente de depuração concentra a recepção dos comandos e encaminha para os mecanismos responsáveis por sua interpretação. Este mesmo agente é responsável por repassar os resultados de depuração para fora do chip.

**Kim et al.** [KIM07] propõem um ambiente de monitoramento para auxiliar em tomadas de decisão em tempo de projeto e contribuir para melhorar o desempenho da infraestrutura de comunicação em tempo de execução. O ambiente é composto por 3 elementos: (i) uma interface com o mundo externo, (ii) um controlador central e (iii) uma unidade de monitoramento. O primeiro provê um meio para capturar resultados de monitoramento a partir de um computador hospedeiro. O segundo habilita ou desabilita a monitoramento, de acordo com decisão do projetista. O último captura informações tais como latência fim-a-fim e utilização dos canais locais, recursos dispostos na interface de rede. Uma plataforma específica é implementada fazendo uso da infraestrutura de comunicação proposta, que foi prototipada em um FPGA. Varia-se o tamanho das filas de armazenamento, até encontrar o menor tamanho que garanta o mesmo desempenho para a aplicação alvo. Para a decisão sobre qual o melhor caminho, pacotes são enviados por caminhos diferentes, sendo assumido como melhor o caminho do pacote que apresentar a menor latência no deslocamento.

**Nollet et al.** [NOL04] propõem uma plataforma conectando PEs através de uma NoC, sendo esta infraestrutura de comunicação monitorada por elementos distribuídos na rede e gerenciada por um sistema operacional. O objetivo é garantir o melhor desempenho da infraestrutura de comunicação através do controle de injeção de tráfego e atualização de rotas de comunicação. A observação da utilização da NoC é realizada nas interfaces de rede, onde para cada PE se monitora o número de mensagens enviadas, recebidas ou bloqueadas. Um PE centralizado (Mestre) recebe periodicamente as informações coletadas nas interfaces de rede (Escravos). O Mestre possui um sistema operacional com o poder de gerenciamento da infraestrutura de comunicação, enquanto



os escravos possuem um sistema básico de atendimento de funções. Com base na métrica de bloqueio de mensagens, o Mestre envia comandos de alteração da permissão de injeção de tráfego na rede por PE ou ainda atualiza as tabelas de roteamento. As adaptações são realizadas através do uso de mensagens de controle que trafegam por uma rede distinta da rede de dados, para minorar a interferência causada por estes na comunicação das aplicações.

**Daneshtalab et al.** [DAN06b] propõem um mecanismo de adaptação de rotas entre pares comunicantes em redes em chip. Os autores baseiam-se em um algoritmo do tipo *AntNet* e tem como objetivo reduzir a ocorrência de pontos quentes e os congestionamentos consequentes. Adaptado para NoCs, o algoritmo *AntNet* foi inicialmente proposto por **Di Caro et al.** para redes de computadores [DIC98]. Este algoritmo baseia-se no comportamento de colônias de formigas, onde o deslocamento das mesmas em busca de alimento se dá por caminhos com maior a concentração de feromônios<sup>1</sup> o que está diretamente relacionado com a quantidade destes insetos que adota um dado caminho. No trabalho desenvolvido, as formigas são pacotes de controle e o alimento é definido como o destino das comunicações. A topologia de rede é uma malha 2D, com roteamento definido por tabelas, a qual contém dados a serem utilizados como valor probabilístico no deslocamento de pacotes de dados. Periodicamente, pacotes de controle são enviados para um destino, respeitando um algoritmo de roteamento. Durante o deslocamento, o pacote de controle coleta informações tais como a identificação do roteador pelo qual está passando e o estado de congestionamento do mesmo. Os autores assumem o estado de ocupação das filas de armazenamento como mecanismo de medida de congestionamento. Se o pacote de controle chegar ao destino, visto que a formação de um ciclo dentro da rede faz com que este seja eliminado, este é retransmitido ao roteador origem da comunicação, assumindo o caminho que utilizou para chegar ao destino. O caminho que foi adotado para chegar ao destino baseia-se nas probabilidades de uso de uma dada porta e no grau de congestionamento do roteador. Durante o retorno do pacote de controle, as tabelas de roteamento são atualizadas para que os pacotes de dados concentrem a transmissão dos dados pela rota menos congestionada. Os autores comparam seu trabalho, por simulação, com os algoritmos XY puro, *Odd-Even* e a infraestrutura de comunicação DyAD. Adota-se o padrão de tráfego transposto, tendo esta abordagem apresentado o maior afastamento do ponto de saturação.

---

<sup>1</sup> Substância química secretada por algumas espécies de animais e captada/interpretada por animais de mesma espécie, sendo capaz de provocar reações fisiológicas/comportamentais em um grupo ou em um indivíduo.

**Rantala et al.** [RAN08] propõem dois métodos de monitoramento de cargas na rede. O primeiro método concentra-se na ocupação das portas de saída do roteador local e dos roteadores vizinhos, no que se costuma chamar *monitoramento do estado da rede*, enquanto o segundo captura a ocupação das filas de armazenamento da chave local e de seus vizinhos, no que se chama *monitoramento de FIFOs*. O monitoramento é dito distribuído pelo fato de haver o conhecimento do estado do roteador local e dos vizinhos. O compartilhamento das informações entre roteadores vizinhos se faz através do emprego de canais dedicados, o que evita que a interferência no tráfego de dados. Ambas as abordagens são usadas como mecanismo de suporte à adaptação de rotas. Resultados apontam para um melhor desempenho da adaptação quando se utiliza monitoramento de FIFOs. Mesmo assim, em ambos os métodos houve ganho de desempenho quando comparado a uma solução que não adota monitoramento.

## **CONSIDERAÇÕES FINAIS SOBRE O EMPREGO DE MONITORES**

Como pode ser visto, o emprego de monitores em redes em chip tem objetivos distintos tais como no auxílio à depuração, tomada de decisão sobre escolhas de projeto, controle do uso de recursos da plataforma alvo e adaptação do uso da mesma. No presente trabalho, monitores são empregados para contribuir no uso da infraestrutura de comunicação principalmente como suporte à tomada de decisões de adaptação de rotas.



### 3. MAPEAMENTO DE ROTAS

No presente Capítulo apresenta-se uma abordagem para a escolha de rotas entre os pares comunicantes visando mitigar a sobrecarga em canais da rede em chip através da distribuição dos fluxos de comunicação, decisão esta tomada em tempo de projeto. Em redes em chip, denomina-se *rota* a sequência de direções que identifica os passos a serem seguidos para garantir a transmissão de um pacote entre entidades comunicantes. Uma *direção* é a identificação de uma escolha a ser tomada, podendo ser o endereço de um próximo roteador ou, como assumido no presente trabalho, a sinalização de qual porta de saída deve ser usada quando dentro de um roteador. Diferentes rotas podem existir para um mesmo par comunicante e, unitariamente, a definição de uma rota não é uma atividade complexa. Um *cenário de rotas* é a uma combinação de rotas de todos pares comunicantes sobre uma rede, sendo que somente uma rota é assumida por par comunicante. Ao processo de busca de cenários de rotas dá-se o nome de *mapeamento de rotas*. Para a escolha de um cenário de rotas ótimo pode-se aplicar critérios tais como vazão mínima, latência máxima de pacotes ou mesmo consumo de energia médio como funções custo.

No presente trabalho, os elementos utilizados para a exploração de cenário de rotas entre os pares comunicantes sobre uma infraestrutura de comunicação são: (i) uma infraestrutura de comunicação adotada para a validação do mapeamento, (ii) algoritmos de roteamento base para a busca de rotas e (iii) um método de avaliação de boas rotas. Cada um dos elementos citados acima é detalhado no decorrer deste Capítulo.

#### 3.1 A INFRAESTRUTURA DE COMUNICAÇÃO HERMES-SR

Nesta Seção será detalhada a organização da NoC Hermes-SR. Esta NoC baseia-se na NoC Hermes [MOR04], adaptando a lógica de controle e a estrutura do pacote para dar suporte a um esquema de roteamento definido na origem. As seguintes características são comuns a ambas as NoCs:

- Filas de armazenamento de entrada com tamanho parametrizável;

- Chaveamento por pacotes com adoção do modo de chaveamento *wormhole*;
- Topologia de rede malha 2D;
- Endereço dos roteadores seguindo o padrão XY;
- Controle de fluxo baseado em créditos.

### 3.1.1 ESTRUTURA DO PACOTE

A estrutura adotada pelos pacotes é similar à utilizada na NoC Hermes, sendo dividida em duas partes: (i) cabeçalho e (ii) corpo. Como a Hermes-SR emprega modo de chaveamento *wormhole*, um *flit* é a menor unidade na rede tendo sido fixado em 16 bits na NoC Hermes-SR. Apesar disto, diferentes tamanhos podem ser adotados para um *flit*, variando de 4, 8, 16, 32 e 64 bits.

O cabeçalho é subdividido em dois campos. O primeiro contém a Rota, um campo de tamanho variável que possui informação suficiente para o pacote alcançar o destino, como será descrito abaixo. O segundo campo, com tamanho de um *flit* dá o Tamanho do corpo do pacote. Um exemplo de pacote que trafega na Hermes-SR é ilustrado na Figura 3.1.

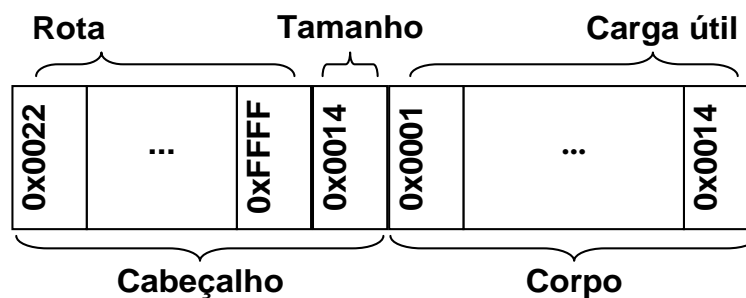


Figura 3.1 - Estrutura do pacote na NoC Hermes-SR. Cabeçalho contendo a rota e finalizado pelo *flit* terminador. O campo Tamanho identifica a quantidade de *flits* no campo Carga útil ou corpo do pacote.

A rota constitui-se de um conjunto de direções a tomar ao longo do caminho entre o roteador origem da comunicação e o roteador destino. A convenção numérica de direções adotada na Hermes-SR é dada na Tabela 3.1. Cada direção é representada por quatro bits. Das dezesseis configurações binárias possíveis apenas quatro são usadas na Hermes-SR, cujo roteador possui no máximo quatro portas ligadas a outros roteadores. A convenção permite acomodar roteadores com um número maior de portas. Uma das dezesseis configurações (0xF) é reservada para representar um valor inválido de direção.

Tabela 3.1 - Identificação das direções adotadas em rotas da NoC Hermes-SR. O prefixo 0x indica um valor hexadecimal.

Identificação	0x0	0x1	0x2	0x3	...	0xF
Porta equivalente	LESTE	OESTE	NORTE	SUL	LIVRE	INVÁLIDA

Cada *flit* do campo de rota armazena até quatro direções consecutivas que o pacote deve seguir. Por exemplo, na Figura 3.1 o primeiro *flit* indica que ao ser injetado na rede o pacote avançará dois *hops* para o Leste e em seguida dois *hops* para o Norte. Uma direção inválida em meio ao tratamento de um *flit* indica a chegada no roteador destino. Como o número de direções pode obviamente ser múltiplo de quatro e para facilitar a execução do roteamento toda rota termina obrigatoriamente por um *flit* com valor 0xFFFF, ou seja, com todas as direções inválidas. Este é o chamado *flit terminador*.

Como dito anteriormente, o campo tamanho identifica a quantidade de *flits* que está contida no corpo do pacote. Na NoC Hermes-SR não se aceitam pacotes com tamanho igual a zero, ou seja, todo pacote deve conter algum dado, mesmo que este não vá ser usado. Os *flits* que seguem o campo Tamanho são a Carga útil do pacote, com conteúdo ignorado pelos roteadores da Hermes-SR, que os transmite sem nenhuma forma de análise.

### 3.1.2 ARQUITETURA DO ROTEADOR HERMES-SR

As infraestruturas de NoC Hermes e Hermes-SR diferem fundamentalmente em dois aspectos: (i) o roteamento distribuído da rede Hermes é substituído na NoC Hermes-SR pelo roteamento na origem e (ii) a arbitragem e roteamento centralizada no roteador Hermes é substituída por uma arbitragem distribuída na Hermes-SR. A primeira diferença tem por objetivo dar suporte à exploração de possíveis rotas entre pares origem/destino de forma controlada, proposta deste trabalho. A segunda diferença objetiva otimizar desempenho, através de um melhor controle de requisições de transmissão de pacotes. A estratégia é implementada pela proposição de um mecanismo distribuído e de uma lógica que garante atendimento em ordem das requisições.

Cada roteador da NoC Hermes-SR possui módulos de entrada e módulos de saída, interconectados por um *crossbar*. Cada par de módulos de entrada e saída vinculados a uma direção constitui a *porta* do roteador para aquela direção. Cada porta em uma direção possui uma porta *de saída* e uma *porta de entrada*. Como a NoC Hermes-SR é uma rede com topologia malha 2D, existem três tipos de roteadores, de 3, 4 e 5 portas, dependendo de localização destes na topologia, con-

forme Figura 3.2. Cada roteador da rede possui uma porta para conexão ao núcleo IP vinculado a este roteador. Esta porta é denominada LOCAL. As demais portas são utilizadas para conectar-se aos roteadores vizinhos. Dependendo da disposição da porta no roteador, uma das seguintes nomenclaturas é adotada: porta NORTE, porta SUL, porta LESTE e porta OESTE, conforme ilustrado na Figura 3.2(a).

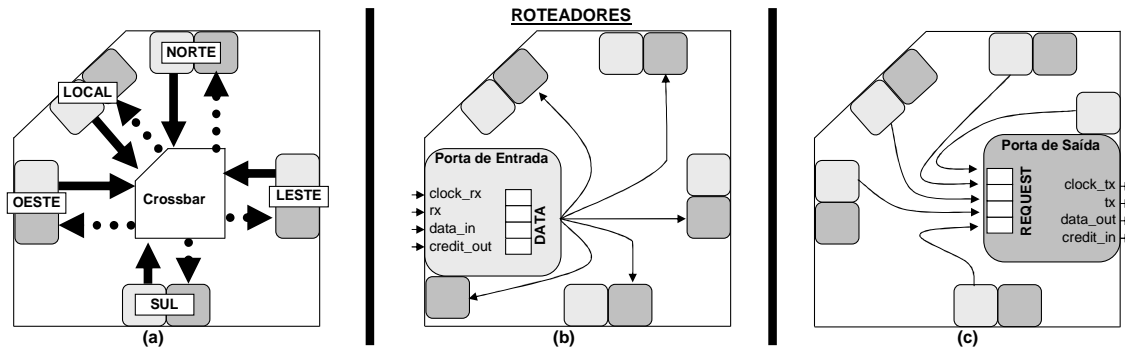


Figura 3.2 - Estrutura interna de roteadores da NoC Hermes-SR. Não há um módulo centralizado para arbitragem e roteamento como na rede Hermes. Esta função é realizada de forma distribuída nas diferentes portas de saída. Em (a) apresenta-se a estrutura da porta de entrada com seus sinais para recepção de dados e uma fila de armazenamento. Em (b) apresenta-se a estrutura da porta de saída, ligada a todas portas de entrada, à fila de armazenamento de requisições e aos sinais de transmissão.

Destaca-se na Figura 3.2(b) a porta de entrada do roteador, que possui duas funções principais. A primeira é armazenar *flits* do pacote que está chegando e a segunda é controlar o envio para uma das portas de saída. Para o cumprimento da primeira funcionalidade, é disponibilizada uma fila de armazenamento, a qual tem seu tamanho parametrizável em tempo de projeto. Para a segunda função, uma máquina de estados provê controle sobre os pacotes. Detectada a recepção do primeiro *flit* de um pacote, avalia-se a direção a ser seguida. Para isso, os quatro bits mais significativos do *flit* indicam a porta de saída. A relação entre o valor representado nestes quatro bits e as portas de destino aparece na Tabela 3.1. Quando um dos valores 0x0, 0x1, 0x2 ou 0x3 é identificado, os pacotes são roteados respectivamente para uma das portas LESTE, OESTE, NORTE ou SUL. Se o valor identificado com quatro bits for 0xF durante o estágio de definição da porta de saída, isto caracteriza uma porta inválida conforme a Tabela 3.1, então o pacote é encaminhado para a porta LOCAL.

O número de direções contidos no campo rota do cabeçalho está diretamente relacionado ao número de *hops* por onde o pacote passa até alcançar o destino. Cada pacote tem o tamanho inicial do campo rota reduzido à medida que avança pela rede. A Figura 3.3 ilustra o processo de

redução da rota. Nesta Figura utiliza-se uma NoC Hermes-SR de dimensão 2x2, e um par onde a origem da comunicação, injetora de pacotes, encontra-se no canto inferior esquerdo (i.e.  $x=0$  e  $y=0$ ) e o destino destes encontra-se no canto superior direito da NoC (i.e.  $x=1$  e  $y=1$ ). A sequência de direções utilizadas para a transmissão de pacotes entre este par é LESTE e NORTE, sendo então necessários 8 bits para sua representação (i.e. 4 bits para cada direção). Para diminuir a quantidade de dados transmitidos na NoC Hermes-SR, a porta LOCAL não é explicitamente informada no cabeçalho. Conforme ilustrado na Figura 3.3(a), a rota para este pacote possui 2 *flits*. Ao chegar ao roteador com  $x=0$  e  $y=0$ , o pacote da Figura 3.3(a) é avaliado. Baseado nos 4 bits mais significativos, uma porta de saída (i.e. LESTE ou 0x1) é alocada.

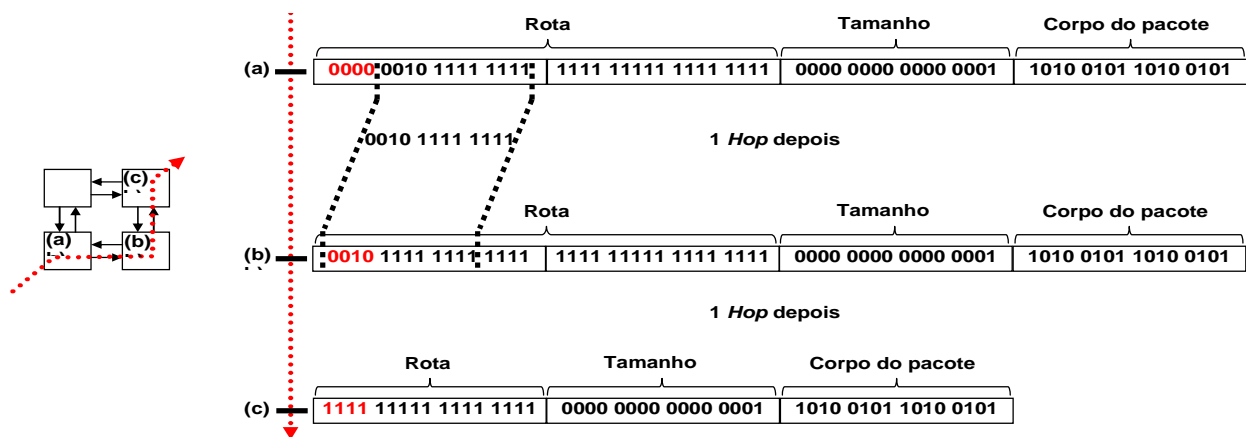


Figura 3.3 – Exemplo de processo de transmissão de pacote na NoC Hermes-SR, ilustrando o tratamento dado ao campo rota.

Antes de encaminhar o *flit*, os quatro bits mais significativos deste são descartados e o restante é deslocado para a esquerda com bits em '1' entrando pela direita. O resultado desta operação é ilustrado na Figura 3.3(b), e representa o estado do cabeçalho do pacote ao ser recebido no segundo roteador. Os demais *flits* do campo rota são mantidos inalterados. Neste roteador, os quatro bits mais significativos são novamente avaliados e definem uma porta de saída (i.e. NORTE ou 0x2). Como o *flit* não carrega mais informação de roteamento, visto que os 12 bits menos significativos estão todos com valor lógico 1, o *flit* inteiro é descartado, e o próximo *flit* do cabeçalho é encaminhado ao próximo roteador, neste caso o *flit* terminador. A Figura 3.3(c) ilustra o estado do pacote ao ser recebido pelo último roteador. Ao identificar este *flit* do cabeçalho como o *flit* terminador, o pacote é encaminhado à porta de saída LOCAL.

Na NoC Hermes, o roteamento de um pacote é realizado a partir de uma requisição a uma unidade de roteamento central ao roteador e compartilhada por todas as portas, conforme dia-



grama de blocos ilustrado na Figura 3.4(a). Na NoC Hermes-SR, a decisão de roteamento é distribuída nas portas de entrada, eliminando uma etapa de concorrência entre as portas, como pode ser observado na Figura 3.4(b). Apesar de reduzir o tempo necessário para a definição do roteamento dos pacotes, na Hermes-SR a abordagem adotada de roteamento na origem e rota armazenada no pacote aumenta o número de *flits* no cabeçalho, devido à necessidade de informar a sequência de direções para alcançar o destino. Com isso o número de ciclos de relógio necessários para a transmissão de todo o campo rota do cabeçalho de um pacote no roteador depende de quatro fatores. O primeiro é o armazenamento do primeiro *flit*, feito em 1 ciclo de relógio quando há lugar na fila de armazenamento. O segundo é a avaliação da direção a seguir, realizada em 1 ciclo de relógio. O terceiro é o início da transmissão do primeiro *flit*, o qual depende de quantidade de outros pacotes, provenientes de outras portas de entrada, que tenham previamente requisitado a mesma porta de saída. O quarto é a transmissão do *flit* terminador, a qual é dependente, entre outros fatores, do número de *flits* para representar a rota.

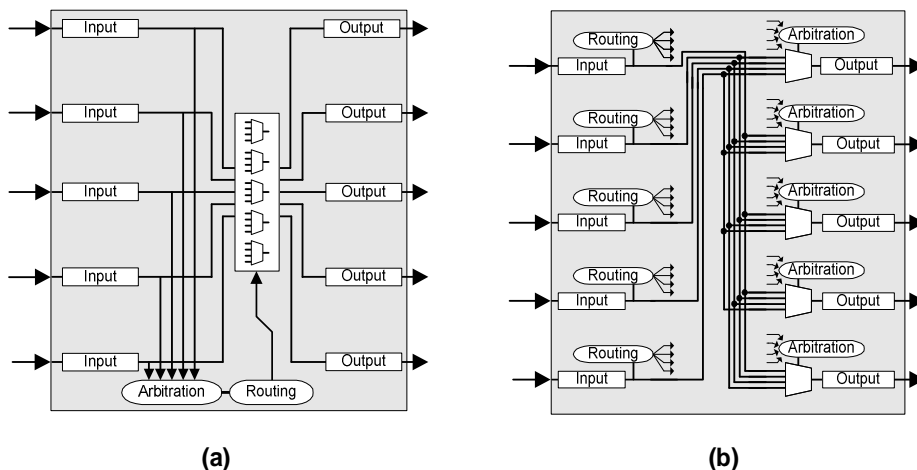


Figura 3.4 - Diagrama de blocos de roteadores com (a) o esquema de arbitragem e roteamento centralizados adotado na NoC Hermes e (b) a arbitragem e roteamento descentralizados empregados na NoC Hermes-SR.

A porta de saída do roteador da Hermes-SR aparece na Figura 3.2(c). Ela possui duas funções principais, quais sejam: (i) o gerenciamento de requisições de transmissão e (ii) o envio dos pacotes. Para a primeira função programou-se uma estratégia de arbitragem diferenciada na NoC Hermes-SR em relação à NoC Hermes. O algoritmo de arbitragem na NoC Hermes é *round-robin*, caracterizado por uma avaliação ordenada e circular das portas de entrada que requisitam roteamento. Suponha-se um roteador que empregue este algoritmo de arbitragem, em uma situação onde quatro portas de entrada solicitam a mesma porta de saída. Nesta situação, a primeira porta

a ser avaliada é a que ganhará o roteamento para aquela porta de saída. As demais portas mantêm seu pedido de roteamento ativo, pedido este que passará novamente pelo processo de arbitragem, terá o roteamento computado e que não poderá iniciar a transmissão enquanto a porta de saída estiver em uso, voltando novamente à etapa de requisição de roteamento. Assim, uma mesma porta de entrada pode ter sua requisição escalonada e o roteamento computado várias vezes até conseguir iniciar a transmissão de um pacote. Como não há uma garantia na ordem de atendimento e as requisições passam repetidas vezes pela arbitragem para a computação da rota, mas sem garantia de que a transmissão ocorrerá, a próxima porta de entrada a ser atendida depende do tempo de transmissão do pacote atualmente roteado, bem como do tempo gasto atribuindo arbitragem e roteamento para as demais portas. Isto compromete a previsibilidade da latência de pacotes em situações de competição. Esta condição potencializa a ocorrência de postergação indefinida na transmissão de pacotes. Na NoC Hermes-SR, sempre que uma dada porta de entrada requer transmissão para uma porta de saída, esta requisição é armazenada em uma fila de atendimento. O atendimento das portas de entrada respeita a ordem em que os pedidos são realizados, garantindo que situações de postergação indefinida nunca ocorrerão. Este algoritmo de arbitragem usa como critério de julgamento o momento da requisição, ou seja, o primeiro a solicitar é o primeiro a ser servido (em inglês, *first come, first served*, FCFS) [DAL03].

Para a segunda funcionalidade da porta da saída, a de envio de pacotes, observa-se a fila de atendimento de requisições. Havendo alguma requisição, ela é retirada da fila e a transmissão do pacote pode ter início, obviamente dependendo da disponibilidade do roteador vizinho ou do núcleo IP conectado à porta de saída. Assim como a funcionalidade de transmissão da porta de entrada, a função de transmissão da porta de saída tem de controlar o pacote para detectar o final da transmissão. Finalizada a transmissão de um pacote e detectada a presença de uma requisição na fila de atendimento, pode-se iniciar uma nova transmissão.

O desempenho da NoC Hermes-SR foi comparado ao da NoC Hermes quando da presença de alguns cenários críticos de transmissão de pacotes, tal como a presença maciça de origens de comunicação, onde ocorre a injeção de pacotes, com taxas variando entre 10% e 50% da capacidade máxima dos canais de dados, ou seja, 80 *Megabits* por segundo (Mbps) e 400 Mbps. A NoC Hermes-SR parametrizada com filas de armazenamento de profundidade quatro obteve latências máximas equivalentes e em alguns casos melhores que os mesmos cenários de tráfego aplicados sobre a NoC Hermes com filas de armazenamento de tamanho 32. Maiores detalhes sobre os re-

sultados obtidos quando da comparação da Hermes-SR com a Hermes são apresentados no Capítulo 4.

### 3.2 ALGORITMOS DE ROTEAMENTO BASE PARA DEFINIÇÃO DE ROTAS

Na NoC Hermes-SR, cada pacote contém em seu cabeçalho uma rota a ser adotada, a qual é interpretada durante o deslocamento do pacote pela rede, conforme descrito na Seção 3.1. Independentemente da abordagem de roteamento, é consenso que redes de interconexão devem prover serviços básicos, garantindo que não ocorrerão situações de *livelock*, *starvation* e *deadlock* [DUA97] [PAT96].

Uma situação de *livelock* é caracterizada quando um pacote evolui ao longo rede de interconexão sem nunca atingir seu destino. Quando se assume roteamento na origem, o campo rota do cabeçalho tem tamanho finito contendo a rota a ser percorrida. Desta forma, garantidamente uma rede com roteamento na origem é livre de *livelock*, se todas as rotas são computadas corretamente. Uma situação de *starvation* ocorre quando um pacote requer um determinado recurso, porém nunca ganha a prioridade de acesso ao recurso, pois o mesmo está sempre ocupado. No que tange a infraestrutura de comunicação, recurso aqui pode representar um enlace de comunicação entre dois roteadores ou entre um núcleo IP e um roteador. Na Hermes-SR emprega-se o algoritmo de arbitragem FCFS, o que garante o atendimento ordenado de pacotes aguardando um recurso e elimina a possível ocorrência de *starvation* para pacotes na rede de comunicação. Uma situação de *deadlock* é caracterizada pela dependência cíclica entre elementos que requisitam acesso a um determinado conjunto de recursos e que, independentemente da sequência de eventos que ocorra, não haverá progresso de um ou mais pacotes em direção aos respectivos destinos, ficando tais elementos perpetuamente bloqueados. Em redes de topologia regular que não possuem roteamento na origem, este problema pode ser evitado através da adoção de algoritmos de roteamento garantidamente livres de *deadlock*, adaptativos ou determinísticos. Em redes de topologia regular com roteamento na origem pode-se evitar uma situação de *deadlock* de duas formas. A primeira é através a definição de rotas calculadas a partir de algoritmos livres de *deadlock*. A segunda é através da definição de rotas quaisquer e posterior validação de que a coexistência destas rotas não leva a situações de *deadlock* na rede. Para isso, técnicas de verificação formal podem ser adotadas, onerando no tempo de projeto.

A NoC Hermes-SR pode ser suscetível a *deadlocks* na transmissão de pacotes, dependendo de como as rotas são definidas para cada pacote. *Deadlocks* podem ocorrer então não devido a falhas arquiteturais, mas como consequência da forma como é utilizada a rede de comunicação pelas entidades que calculam as diferentes rotas dos pacotes. Para evitar tal problema, exploram-se apenas rotas que sigam caminhos definidos por algoritmos livres de *deadlock*. Quatro algoritmos de roteamento distintos foram adotados para a definição das rotas, três destes variando entre roteamento mínimo e não mínimo, compondo um universo de sete soluções possíveis. Os últimos citados são algoritmos derivados do modelo de curvas (em inglês, *turn model*) de Glass e Ni [GLA94]. O conjunto de sete algoritmos investigados inclui os algoritmos: XY puro, *negative first* não mínimo, *negative first* mínimo, *West first* não mínimo, *West first* mínimo, *North last* não mínimo e *North last* mínimo. A definição de rotas em uma dada instância da rede deve respeitar apenas um dos algoritmos de roteamento, pois o uso de mais de um destes algoritmos para definir diferentes rotas pode criar situações de *deadlock*. Os diferentes algoritmos de roteamento foram adotados para dar suporte à proposta deste trabalho disponibilizando um conjunto grande de possibilidades de caminhos entre pares comunicantes. Experimentos foram conduzidos para avaliar os algoritmos que fornecem melhores resultados quanto ao controle de congestionamento. Detalham-se os resultados de tais experimentos no Capítulo 4.

O algoritmo de roteamento XY puro (*xy*) é determinístico. Nele, os *flits* que compõem o pacote são inicialmente roteados sobre o eixo X até alcançar a coluna onde se encontra o destino e a seguir percorrem o eixo Y até o roteador destino. Este algoritmo é garantidamente livre de *deadlock* uma vez que proíbe um conjunto de curvas que eventualmente poderiam levar a uma dependência circular, conforme mostra a Figura 3.5.

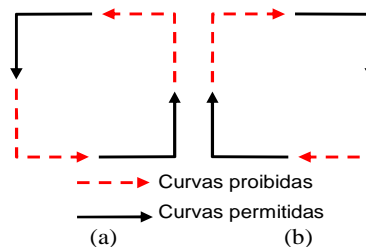


Figura 3.5 – Diagrama de curvas para o algoritmo de roteamento XY puro. Em (a) e (b) nota-se a proibição dos quatro diferentes tipos de curvas em direção ao eixo X, após iniciado deslocamento do pacote sobre o eixo Y.

A desvantagem do uso deste algoritmo está na quantidade de opções de rotas entre qualquer par origem/destino, limitado a exatamente uma única rota, independente da posição relativa

da origem e do destino. O número total de rotas que podem existir na rede ( $NRotas_{xy}$ ) é sempre proporcional ao número de pares comunicantes, este último dependente entre outros fatores do tamanho da rede. Em uma NoC que usa algoritmo de roteamento XY, o número total de rotas na rede é igual ao número total de potenciais origens de comunicação ( $NOrig$ ) multiplicado pelo número de potenciais destinos da comunicação ( $NDest$ ). Se cada roteador da NoC possuir um e somente um núcleo IP e este nunca envia pacotes à rede tendo como destino ele mesmo, então o número de potenciais destinos ( $NDest$ ) é igual a ( $NOrig - 1$ ). Assim, em uma NoC de dimensões 3x3, 9 roteadores podem ser origens de comunicação, mas para cada roteador apenas 8 destinos são possíveis, contabilizando um total de 72 rotas. Esta equação é representada abaixo.

$$\text{Equação 3.1} \quad NRotas_{xy} = NOrig * (NOrig - 1)$$

### 3.2.1 ALGORITMO DE ROTEAMENTO *NEGATIVE FIRST*

O algoritmo de roteamento *negative first* é adaptativo. Nele, os *flits* que compõem um pacote podem ser inicialmente roteados em direções negativas. Quando uma direção positiva é assumida, não é mais permitido realizar curvas para direções negativas. O roteamento para direções negativas, em uma rede malha 2D, caracteriza-se pelo envio de pacotes para as portas OESTE ou SUL, enquanto LESTE e NORTE são consideradas direções positivas. A Figura 3.6 detalha as curvas não permitidas pelo algoritmo e que garantem a ausência de *deadlock* [GLA94].

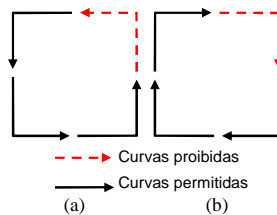


Figura 3.6 – Diagrama de curvas para o algoritmo de roteamento *negative first*. Proíbe-se curvas para direções negativas se uma direção anteriormente tomada seguiu um sentido positivo. Em (a) proíbe-se curvas para OESTE (direção negativa) se o pacote está subindo e em (b) proíbe-se curvas para SUL (direção negativa) se o pacote está indo para a direita. Note-se que não há restrições para iniciar o caminho de um pacote seguindo para OESTE ou para o SUL, e mesmo relizar curvas entre estas direções.

O algoritmo *negative first* foi explorado nas versões mínima e não mínima. No algoritmo mínimo define-se um “quadrado envolvente” (QE) (em inglês, *bounding box*) que limita as opções de roteamento. O QE é reduzido cada vez que o pacote avança para seu destino, nunca se afastando de seu alvo. Na versão não mínima, o QE com o destino pode ser inicialmente desconsiderado enquanto se utiliza apenas direções negativas. Porém, o QE passa a ser respeitado uma vez iniciado

o caminhamento no sentido positivo. A desvantagem do algoritmo mínimo é a redução do número de opções de rotas entre uma origem de comunicação e seu destino. O número de rotas entre uma origem e um destino, no algoritmo *negative first*, depende da minimalidade do algoritmo, da posição da origem em relação ao destino da comunicação e de suas posições, relativas aos limites da rede. Pressupondo uma topologia malha 2D de dimensões NxM, podem ocorrer as seguintes situações:

- Se o destino da comunicação encontra-se em uma posição negativa em relação à origem da comunicação sobre ambos os eixos coordenados, ou seja, se  $X_{\text{destino}} < X_{\text{origem}}$  e  $Y_{\text{destino}} < Y_{\text{origem}}$ , então necessariamente haverá roteamento inicial para direções negativas, independente da minimalidade;
- Se a posição do destino da comunicação é positiva em relação à origem sobre ambos eixos coordenados, ou seja se  $X_{\text{destino}} > X_{\text{origem}}$  e  $Y_{\text{destino}} > Y_{\text{origem}}$ , então somente poderá ocorrer roteamento em direções negativas se adotado o algoritmo não mínimo. Para o algoritmo mínimo, restringe-se o roteamento em direções positivas.

Para o algoritmo de roteamento *negative first* mínimo (*nfm*), a exploração de rotas deve respeitar o QE, conforme descrito anteriormente. O roteamento em direções negativas somente ocorre se alguma das coordenadas (X,Y) que define o posicionamento do destino na rede for negativa em relação à origem da comunicação. Se apenas um dos eixos coordenados do destino for negativo em relação à origem ou um dos eixos for igual ao da origem, então existe somente uma rota possível entre este par comunicante, conforme ilustrado na Figura 3.7.

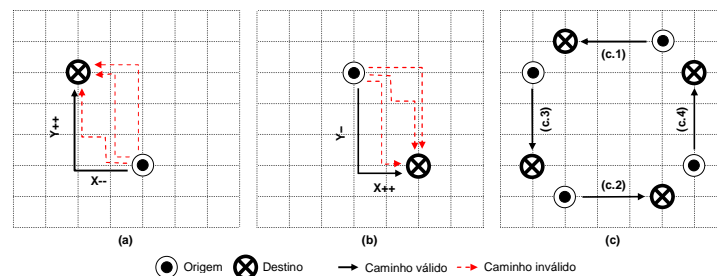


Figura 3.7 – Situações de rota única no algoritmo *nfm*. Em (a), o destino possui o valor de coordenada x menor que o destino. Em (b), o destino possui o valor de coordenada y menor que o destino. Em (c), o destino encontra-se sobre a mesma linha (c.1) e (c.2) ou sobre a mesma coluna (c.3) e (c.4).

Caso a posição do destino seja totalmente negativa ou totalmente positiva em relação à origem da comunicação, ou seja, ambos os valores da coordenada (X,Y) do destino são menores (po-

sição relativa negativa), ou maiores (posição relativa positiva) respectivamente, em relação à origem, as rotas serão unicamente compostas por caminhamento em sentidos negativos ou positivos. Este roteamento caracteriza o deslocamento sobre o QE. A Figura 3.8 exemplifica um posicionamento totalmente positivo (Figura 3.8 (a)) e um posicionamento totalmente negativo (Figura 3.8 (b)) entre uma dada origem de comunicação e seu destino.

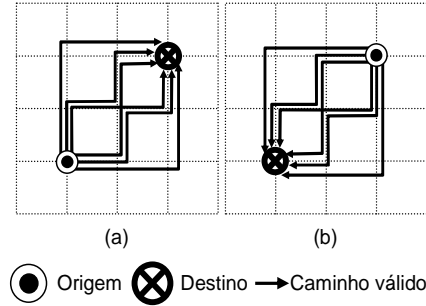


Figura 3.8 - Caminhos válidos entre origem e destino com distância 2 em X e em Y. Em (a) mostra-se o posicionamento totalmente positivo do destino em relação a origem. Em (b) mostra-se o posicionamento totalmente negativo do destino em relação a origem. Em ambos os casos, nota-se o respeito aos limites impostos pelo QE.

O roteamento realizado com algoritmos mínimos em uma rede com topologia malha 2D implica a avaliação de no máximo 2 arcos, no caso do algoritmo *negative first* sendo em direções positivas ou negativas. Para o roteamento mínimo sobre um QE, o número de rotas possíveis entre um par origem/destino é dado pela soma sucessiva do número de direções válidas a partir de vértices adjacentes. Quando mais de uma rota pode existir entre um par origem/destino sobre o QE, o número de rotas pode ser calculado tomando por base sua distância. Definindo-se a posição da origem como o par de coordenadas  $(X_{\text{Origem}}, Y_{\text{Origem}})$  e as coordenadas do destino como  $(X_{\text{Destino}}, Y_{\text{Destino}})$ , a distância  $\Delta x = |X_{\text{Origem}} - X_{\text{Destino}}|$  e a distância  $\Delta y = |Y_{\text{Origem}} - Y_{\text{Destino}}|$ , se ambos  $\Delta x$  e  $\Delta y$  forem diferentes de zero então a quantidade de opções de rotas é expressa pela Equação 3.2. Caso contrário, existe apenas 1 rota [GLA94].

$$\text{Equação 3.2} \quad NRotas_{Min} = \frac{(\Delta x + \Delta y)!}{\Delta x! \Delta y!}$$

Na Figura 3.9, ilustra-se o crescimento do número de rotas possíveis quando origens e destino distam de até 9 hops em X e 9 hops em Y. Supondo-se uma origem da comunicação no roteador do canto inferior esquerdo (i.e.  $X_{\text{Origem}}=0$  e  $Y_{\text{Origem}}=0$ ), detalha-se o número de rotas possíveis para chegar a cada outro roteador. É possível observar então que o número de rotas possíveis en-

tre cada par origem/destino é proporcional às distâncias sobre os eixos X e Y.

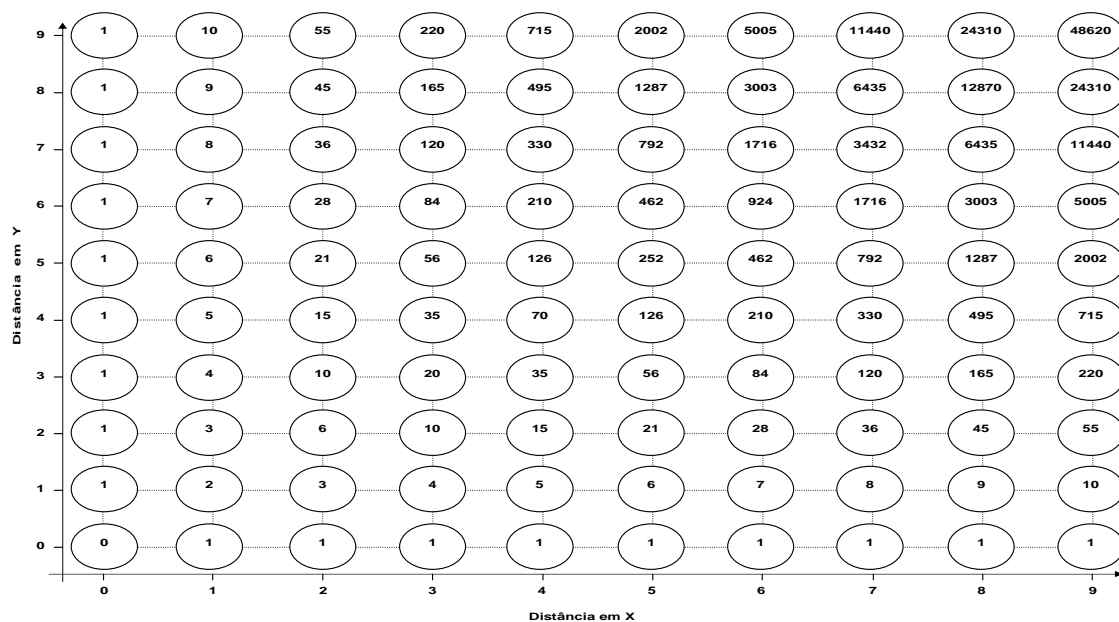


Figura 3.9 – Crescimento do número de rotas possíveis par a o roteador localizado no canto inferior esquerdo tendo como destino qualquer dos demais roteadores quando é assumido os algoritmo de roteamento mínimo explorados no presente trabalho.

Para o algoritmo *negative first* não mínimo (*nfnm*), o roteamento inicial em direções negativas é permitido mesmo quando o destino encontra-se em posição positiva em relação à origem da comunicação, mas é mandatório quando o destino está em uma posição negativa em relação à origem. A Figura 3.10 apresenta a situação onde se permite roteamento inicial em direções negativas, apesar de não ser obrigatório. Nela, o destino encontra-se a uma distância 2 em X e 2 em Y em relação à origem. Se o algoritmo *nfm* fosse assumido, somente seis rotas seriam possíveis. Porém, com a possibilidade de iniciar o roteamento em direções negativas, o número de rotas possíveis entre uma origem e seu destino aumenta. Quando o roteamento inicial ocorre em direções negativas, qualquer um dos pontos destacados da Figura 3.10(a) pode reverter o roteamento para direções positivas para alcançar o destino.

Nota-se que o número de pontos possíveis quando o roteamento inicial em direções negativas tem relação com a posição da origem e não do destino. No caso apresentado na Figura 3.10, além da origem da comunicação, 19 outros pontos podem iniciar o roteamento em direções positivas. Na Figura 3.10(a), são apresentados os roteadores que o roteamento não mínimo pode usar afastando-se do destino e a quantidade de opções de rotas para alcançar cada um destes roteadores. Já na Figura 3.10(b), é apresentado o número total de rotas possíveis a partir de um dos rotea-



dores Salientados na Figura 3.10 (a) até o destino.

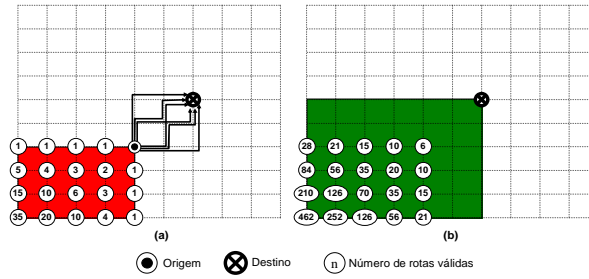


Figura 3.10 - Alternativas de rotas quando inicialmente se usa algoritmo *nfnm*. Em (a) apresenta-se a quantidade de rotas a partir da origem para chegar a algum roteador alcançável, ao afastar-se do destino. Em (b), mostra-se a quantidade de rotas possíveis quando, ao alcançar um roteador mais afastado, assume-se direção positiva.

No exemplo apresentado na Figura 3.10, se inicialmente toma-se uma direção positiva, a quantidade de rotas possíveis seria seis. Com o afastamento do destino, permitida pelo algoritmo não mínimo, a quantidade de rotas possíveis para alcançar o destino utilizando-se o algoritmo *nfnm* é definido pelo somatório da quantidade de rotas possíveis para cada roteador intermediário que se afasta do destino (RotInt) multiplicado pela quantidade de rotas possíveis entre o roteador RotInt até o destino, conforme a Equação 3.3. Nesta Equação, a função  $NRotas_{Min}$  representa a quantidade de rotas entre um par origem/destino usando o algoritmo *nfm*,  $NRotInt$  representa o número de roteadores intermediários RotInt podem ser utilizadas durante o afastamento do destino, Orig e Dest caracterizam o par origem/destino da comunicação. No caso da Figura 3.10, este número é igual a 28544.

Equação 3.3

$$NRotas_{NF\_Nao\_Minimo} = NRotas_{Min}(Orig, Dest) + \sum_{i=1}^{NRotInt} NRotas_{Min}(Orig, RotInt_i) * NRotas_{Min}(RotInt_i, Dest)$$

### 3.2.2 ALGORITMO DE ROTEAMENTO *WEST FIRST*

O algoritmo *West first* é adaptativo. Nele, os pacotes podem ser roteados inicialmente para OESTE, mas depois que o sentido NORTE, SUL ou LESTE for assumido, não é mais permitida a direção OESTE. A Figura 3.11 detalha as curvas proibidas através de setas pontilhadas, ilustrando que a ocorrência de *deadlock* é evitada a partir da eliminação de possíveis ciclos [DAL87].

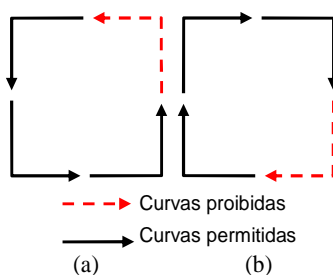


Figura 3.11 – Diagrama de curvas para o algoritmo de roteamento *West first*. Em (a) e em (b), nota-se que curvas para OESTE são proibidas após caminhamentos em sentido diferente de OESTE. Note-se que sempre se pode ir inicialmente para OESTE.

Em redes que empregam topologia malha 2D, a disposição dos destinos em relação à origem da comunicação assume uma das situações ilustradas na Figura 3.12. Nesta Figura, destinos identificados por X são os que se encontram no mesmo eixo X da origem da comunicação e são referenciados doravante como  $MmoX$ . Diferencia-se destinos  $MmoX$  como negativos ( $MmoX_{Neg}$ ), quando sua coordenada X é menor que da origem, ou como positivo ( $MmoX_{Pos}$ ), quando sua coordenada X é maior que a origem. Ainda na Figura 3.12, destinos identificados por Y são os que se encontram no mesmo eixo Y da origem e são referenciado como  $MmoY$ . Diferencia-se destinos  $MmoY$  como negativos ( $MmoY_{Neg}$ ), quando sua coordenada Y é menor que a origem, ou como positivos ( $MmoY_{Pos}$ ), quando sua coordenada Y é maior que a origem. Destinos identificados naquela Figura com o valor 1 encontram-se no quadrante inferior esquerdo, representando um posicionamento totalmente negativo em relação à origem e serão doravante classificados como no quadrante negativo negativo ( $QdX_{Neg}Y_{Neg}$ ). Destinos identificados com o valor 2 encontram-se no quadrante superior esquerdo e serão doravante classificados como no quadrante negativo positivo ( $QdX_{Neg}Y_{Pos}$ ). Destinos identificados pelo valor 3 encontram-se no quadrante superior direito e serão doravante classificados como no quadrante positivo positivo ( $QdX_{Pos}Y_{Pos}$ ). Destinos identificados pelo valor 4 encontram-se no quadrante inferior direito e serão doravante referenciados como no quadrante positivo negativo ( $QdX_{Pos}Y_{Neg}$ ).

Com base nas definições de posicionamento dos destinos em relação à origem em redes malha 2D, adotam-se as seguintes características para roteamento *West first*. Independente da minimalidade do algoritmo, sempre que o posicionamento do destino for menor do que a origem em relação ao eixo X, os primeiros passos necessariamente terão de ser para OESTE. Isto é válido para destinos localizado em  $MmoX_{Neg}$ ,  $QdX_{Neg}Y_{Neg}$  e  $QdX_{Neg}Y_{Pos}$ , conforme ilustrado na Figura 3.13(a). Outro fator a este algoritmo, independente de sua característica de minimalidade, é o caminhamento no sentido LESTE o qual é limitado ao eixo vertical onde se encontra o destino, sendo

proibido ultrapassá-lo, conforme ilustrado na Figura 3.13(b).

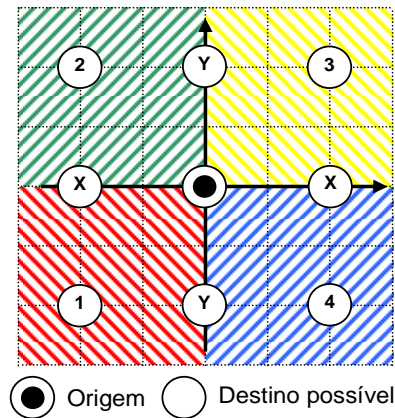


Figura 3.12 - Disposição relativa dos destinos em relação à origem da comunicação em redes malha 2D.

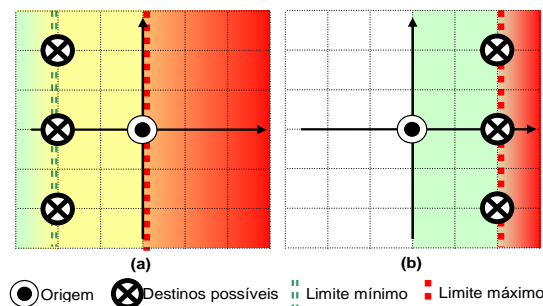


Figura 3.13 – Características do roteamento com o algoritmo *West first*. Em (a), quando o destino encontra-se à esquerda da origem, o roteamento inicial para OESTE é obrigatório até que a coluna do destino seja alcançada, podendo continuar para OESTE caso o algoritmo seja não mínimo. Em (b), quando o destino encontra-se à direita da origem, o caminhar nunca pode ultrapassar a coluna do destino.

Para o algoritmo *West first* mínimo (*wfm*), o roteamento na direção OESTE é obrigatório sempre que o posicionamento do destino for negativo em relação ao eixo X da origem da comunicação, situação esta ilustrada na Figura 3.13(a). Neste caso e devido à minimalidade do algoritmo, quando realizado o roteamento na direção OESTE, existe somente uma rota entre o par origem/destino. A Figura 3.14 (a), (b) e (c.1) representam tal condição. Outra situação em que somente uma rota é válida entre um par origem/destino ocorre quando o destino localiza-se em  $MmoX$  ou  $MmoY$ , conforme ilustrado na Figura 3.14 (c).

A quantidade de opções de rota entre um par origem/destino é superior a um quando o destino encontra-se em  $QdX_{Pos}Y_{Neg}$  ou  $QdX_{Pos}Y_{Pos}$  em relação à origem, permitindo roteamento livre nos sentidos NORTE, SUL e LESTE. Nestes casos, a quantidade de opções de rotas é definida pelo QE, assim como discutido para o algoritmo *nfm*. A Figura 3.15 ilustra as rotas permitidas para des-

tinhas que se encontram em  $QdX_{Pos}Y_{Neg}$  ou  $QdX_{Pos}Y_{Pos}$  em relação à origem.

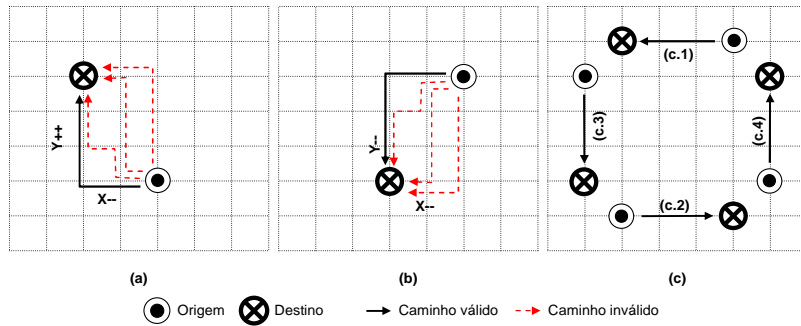


Figura 3.14 - Situações de rota única para o algoritmo *wfm*. Em (a) e (b) os destinos encontram-se respectivamente em  $QdX_{Pos}Y_{Neg}$  e  $QdX_{Neg}Y_{Neg}$  em relação à origem. Em (c) os destinos encontram-se sobre o mesmo eixo X ou mesmo eixo Y em relação à origem.

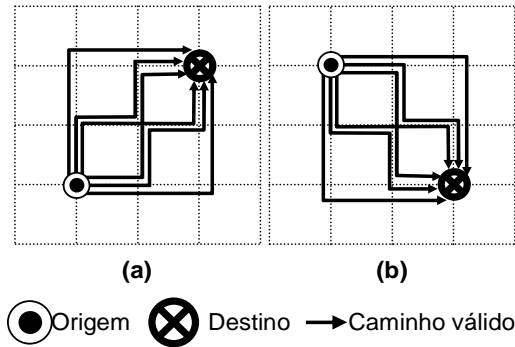


Figura 3.15 – Opções de rotas entre um par origem/destinos com distância 2 em X e 2 em Y ao adotar o algoritmo *wfm*. Em (a) o destino encontra-se no  $QdX_{Pos}Y_{Pos}$  em relação à origem. Em (b) o destino encontra-se em  $QdX_{Pos}Y_{Neg}$  em relação à origem. Em ambos os casos, há respeito a limites colocados de QE.

A quantidade de opções de rotas entre um par origem/destino é proporcional à distância em x e em y, conforme já definido na Equação 3.2.

Para o algoritmo *West first* não mínimo (*wfnm*), o roteamento inicial na direção OESTE também será obrigatório sempre que o posicionamento do destino for negativo em relação ao eixo X da origem da comunicação. Nos demais casos, pode-se inicialmente adotar a direção OESTE. Diferentemente do algoritmo *wfm*, no caso de obrigatoriedade de roteamento para OESTE, quando o pacote chega ao mesmo eixo Y do destino, a adoção da direção OESTE só é impedida em dois casos, quais sejam: (i) uma direção que não seja OESTE já foi utilizada no roteamento do pacote ou (ii) a borda da rede foi alcançada. A Figura 3.16 ilustra as áreas de possível roteamento quando *wfnm* é adotado.

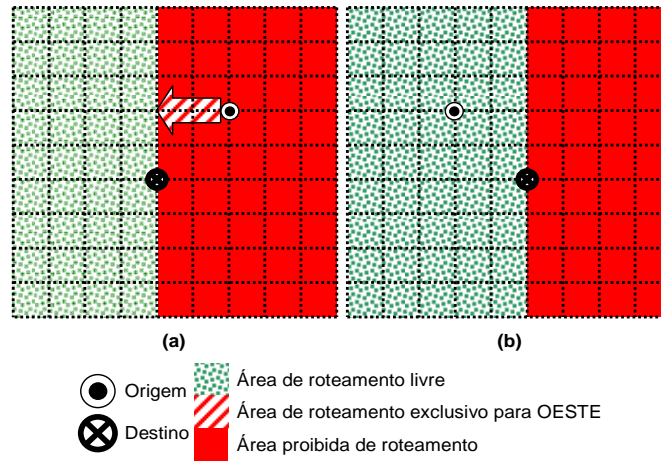


Figura 3.16 - Áreas de roteamento para o algoritmo *wfnm*. Em (a), o destino encontra-se no  $QdX_{Neg}Y_{Neg}$  em relação à origem, o que reduz a área de roteamento livre. Em (b), o destino encontra-se no  $QdX_{Pos}Y_{Neg}$  em relação à origem, aumentando a área de roteamento livre.

### 3.2.3 ALGORITMO DE ROTEAMENTO *NORTH LAST*

O algoritmo *North last* é adaptativo. O roteamento adotando este algoritmo define que pacotes podem ser roteados inicialmente nas direções LESTE, OESTE ou SUL. No momento em que a direção NORTE é assumida, nenhuma outra direção será permitida até que o destino seja alcançado, quando a porta LOCAL é utilizada. A Figura 3.17 detalha as curvas proibidas para este algoritmo através de setas pontilhadas ilustrando que a ocorrência de *deadlock* é evitada a partir da eliminação de possíveis ciclos [GLA94].

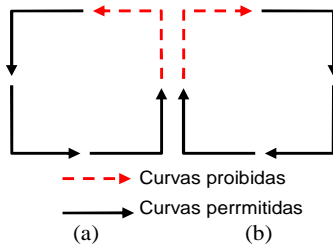


Figura 3.17 – Diagrama de curvas para o algoritmo de roteamento *North last*. Tanto em (a) quanto em (b) é ilustrado que curvas em todos os sentidos são válidas até que o sentido NORTE seja tomado. A partir daí nenhuma curva é mais permitida.

Com base na posição dos destinos em relação à origem da comunicação em redes com topologia malha 2D, as seguintes características são associadas ao algoritmo de roteamento *North last*. Independente da minimalidade do algoritmo, sempre que o destino estiver no  $QdX_{Neg}Y_{Neg}$ , no  $QdX_{Pos}Y_{Neg}$  ou na  $MmoY_{Neg}$  em relação à origem, não poderão ser usados os quadrantes  $QdX_{Neg}Y_{Pos}$  e  $QdX_{Pos}Y_{Pos}$  para o roteamento de pacotes, conforme ilustra a Figura 3.18(a). Quando o destino estiver em relação à origem em  $QdX_{Neg}Y_{Pos}$ , em  $QdX_{Pos}Y_{Pos}$  ou em  $MmoY_{Pos}$ , roteamentos na direção

NORTE somente pode ocorrer no eixo Y do destino e nunca se pode ultrapassar o eixo X onde este se encontra.

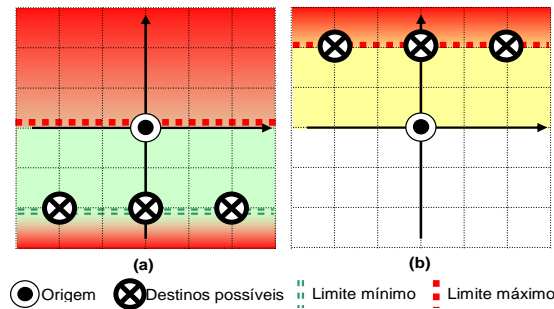


Figura 3.18 – Características do algoritmo de roteamento *North last*. Em (a), quando o destino encontra-se em  $QdX_{Neg}Y_{Neg}$ , ou em  $QdX_{Pos}Y_{Neg}$ , ou em  $MmoYNeg$ , o roteamento para a direção NORTE pode não ocorrer, no caso do algoritmo mínimo. Em (b), a exceção da linha onde se encontra à origem, a área entre a origem e o destino deve ser coberta apenas por roteamentos no sentido NORTE.

O algoritmo *North last* mínimo (*n/m*) permite mais de uma opção de rota quando o destino está abaixo da origem, ou seja, quando o destino está no  $QdX_{Pos}Y_{Neg}$ , no  $QdX_{Neg}Y_{Neg}$  ou no  $MmoYNeg$  em relação à origem. Em casos onde o destino encontra-se acima da origem ou no mesmo eixo X, existe apenas uma rota válida conforme ilustrado na Figura 3.19. Nos casos (a) e (b) da Figura 3.19, o roteamento se dá sobre o eixo X até alcançar o eixo Y onde se encontra o destino. A partir daí, permite-se apenas o roteamento para a direção NORTE. No caso (c) da Figura 3.19, em relação à origem, o destino encontra-se no  $MmoX$  em (c.1) e (c.2) e no  $MmoY$  em (c.3) e (c.4).

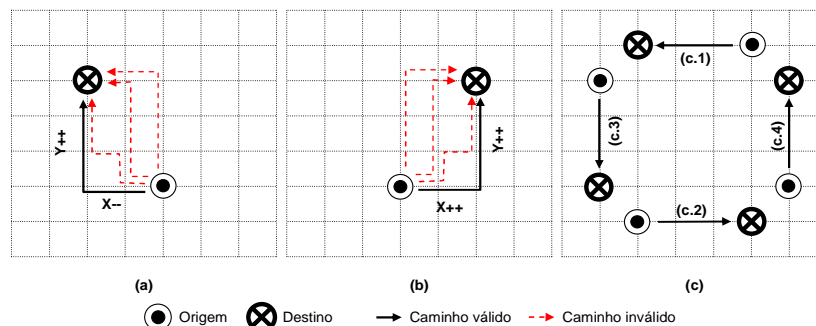


Figura 3.19 - Situações de rota única para o algoritmo *n/m*. Em (a) e (b) os destinos encontram-se em relação à origem respectivamente em  $QdX_{Neg}Y_{Pos}$  e em  $QdX_{Pos}Y_{Pos}$ . Em (c), (c.1) e (c.2) os destinos encontram-se em  $MmoX$ , enquanto em (c.3) e (c.4) encontram-se em  $MmoY$  em relação à origem.

Nas demais situações, ou seja, quando o destino encontra-se em  $QdX_{Neg}Y_{Neg}$  ou em  $QdX_{Pos}Y_{Neg}$  em relação à origem, o roteamento é livre nas direções LESTE, OESTE e SUL, respeitando o que foi apresentado na Figura 3.18(a). A Figura 3.20 ilustra o roteamento permitido nas situações apresentadas. A quantidade de opções de rotas entre pares origem/destino é proporcional a sua

distância em X e Y, conforme apresentado na Equação 3.2, quando a disposição deste par condiz com aquelas da Figura 3.20.

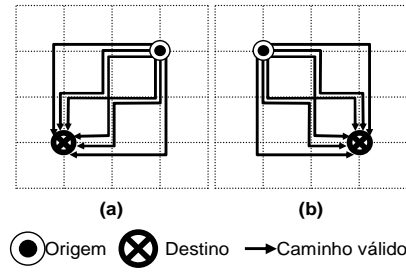


Figura 3.20 – Rotas possíveis entre origens e destinos com distância 2 em X e 2 em Y e adotando o algoritmo de roteamento *n/m*. Em (a) o destino encontra-se em  $QdX_{Neg}Y_{Neg}$  em relação à origem e em (b) o destino encontra-se em  $QdX_{Pos}Y_{Neg}$  em relação à origem.

Concentrando-se agora no algoritmo *North last* não mínimo (*nlnm*), o roteamento final na direção NORTE será obrigatório nas mesmas condições do algoritmo mínimo, ou seja, uma vez assumida esta direção nenhuma outra pode ser utilizada. A exceção é que no algoritmo não mínimo os pacotes podem ultrapassar os limites definidos pelo QE enquanto uma curva para NORTE não for realizada. A Figura 3.21 ilustra as áreas de roteamento livre para pacotes que seguem rotas definidas segundo as regras do algoritmo *nlnm*.

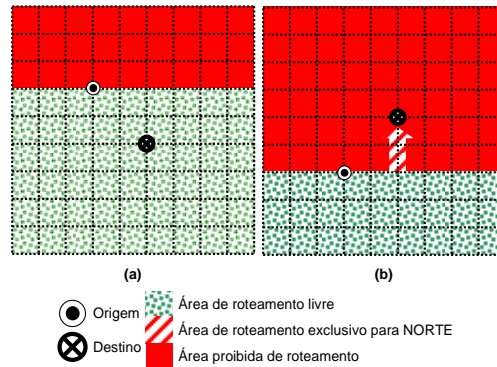


Figura 3.21 - Áreas de roteamento livre, exclusivo para NORTE e proibida no algoritmo *nlnm*. Em (a), a localização do destino em relação à origem aumenta a área de roteamento livre e em (b), a localização do destino em relação à origem reduz a área de roteamento.

### 3.3 AVALIAÇÃO DE ROTAS

Para transmissões através da NoC Hermes-SR, o campo rota do cabeçalho de cada pacote contém toda a sequência de direções a serem tomadas desde a origem da comunicação até o destino. A definição das rotas a serem utilizadas entre cada par comunicante deve garantir que não ocorrerá *deadlock* na infraestrutura de comunicação. Para isso, a escolha de rotas é feita através

da observação de regras impostas por algoritmos que garantem a inexistência de *deadlock*, restritos àqueles avaliados na Seção 3.2. Como visto anteriormente, o uso do algoritmo determinístico XY limita a existência de apenas uma opção de rota para qualquer par comunicante. Com o objetivo de aumentar a quantidade de opções de rota utilizou-se os algoritmos adaptativos como base. O aumento da quantidade de rotas permite mais facilmente ter controle sobre a ocorrência de congestionamento, pois permite a escolha de rotas que evitem pontos de maior concorrência.

Define-se *congestionamento* como a situação onde a latência de entrega de pacotes inviabiliza o atendimento a um serviço contratado, tal como uma taxa de transmissão. Congestionamentos podem ser causados por diferentes motivos, tal como o mau dimensionamento de filas de armazenamento de *flits* de pacotes e/ou a sobrecarga de tráfego em determinados pontos da rede. Esta última pode ser causada pela ocorrência de pontos quentes, no caso de sobrecarga em canais, ou módulos quentes (em inglês, *hot modules*), no caso de sobrecarga de requisições a um recurso computacional conectado à rede. A Figura 3.22 ilustra uma situação potencial de congestionamento causada por um ponto quente. Na Figura, cada seta representa um fluxo entre um par origem/destino. Hipoteticamente, cada fluxo da Figura tem uma taxa contratada de transmissão de 100 Mbps e, supondo que cada canal tem capacidade máxima de transmissão de 200 Mbps, o compartilhamento do canal pelos três fluxos supera em 100 Mbps sua capacidade, gerando um ponto quente.

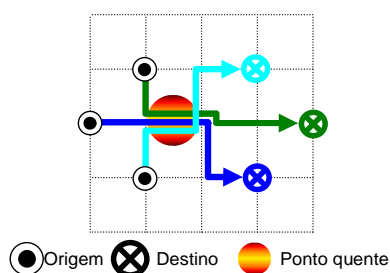


Figura 3.22 - Situação de ocorrência de ponto quente e consequente congestionamento. Ponto quente aqui ocorre devido a sobrecarga de fluxos de pacotes em canais, superando suas capacidades de vazão.

Como diferentes fluxos convivem em uma infraestrutura de comunicação e cada fluxo está tipicamente relacionado a um par origem/destino, a escolha de uma rota para cada fluxo tem de garantir que não haverá sobrecarga em canais. Adicionalmente, a escolha de uma rota para cada fluxo deve ser feita de modo que haja uma distribuição equalizada das cargas na infraestrutura de comunicação. No presente trabalho, assume-se que o congestionamento ocorre pela má escolha



de rotas, mesmo que a infraestrutura de comunicação disponibilize recursos que garantam a transmissão dos pacotes com qualidade. Uma rota entre um par comunicante é considerada boa se é a de menor distância e encontra menor concorrência, ou seja, se estiver o mais livre possível para transmitir pacotes de um dado fluxo. Na Seção 3.3.2 detalha-se o processo de escolha das rotas, bem com as prioridades dos critérios levados em consideração. Uma escolha de boas rotas para os fluxos apresentados na Figura 3.22, onde se elimina a concorrência nos canais e se garante uma menor distância entre os pares comunicantes é ilustrada na Figura 3.23(a).

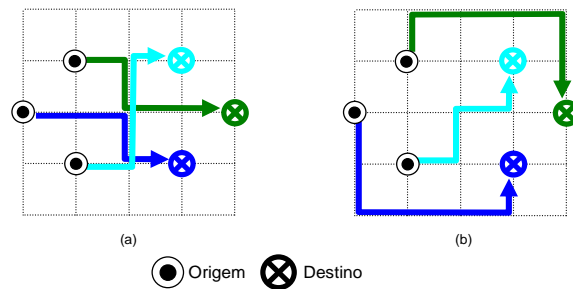


Figura 3.23 – Distribuições alternativas para os fluxos de comunicação entre os pares origem e destino visando eliminar a ocorrência de pontos quentes e consequentes congestionamentos. Em (a) a escolha de boas rotas, enquanto em (b) obtém-se uma boa distribuição, usando caminhos mais longos na rede.

Na presença de múltiplos fluxos, nem sempre uma boa rota pode ser obtida para cada um dos pares comunicantes. Para os mesmos fluxos apresentados na Figura 3.22, uma boa distribuição de rotas, mas que elimina a concorrência nos canais e garante uma distância razoável entre os pares comunicantes, é ilustrada na Figura 3.23(b). A uma distribuição de rotas dá-se o nome de cenário de rotas, formalmente definido a seguir.

### 3.3.1 ORGANIZAÇÃO DO MAPEAMENTO DE ROTAS

Redes que adotam topologia malha 2D podem ser representadas através de um grafo dirigido  $N = \langle R, C \rangle$ , onde o conjunto de vértices  $R = \{r_0, \dots, r_{i-1}\}$  representa os roteadores da rede e o  $A$ -qui,  $i$  é o número de roteadores na rede e  $j$  é o número de canais da rede. Assim, em uma topologia malha 2D, cada roteador possui 4, 6 ou 8 canais, utilizados para encaminhar ou receber dados de roteadores adjacentes. Cada elemento do conjunto de canais  $C$  de uma rede  $N$  pode ser definido como um par ordenado:

$$\forall c_i \in C, c_i = (r_a, r_b), \text{ onde } r_a \in R, r_b \in R.$$

Obs: Note-se que esta definição automaticamente exclui como canais da rede os componentes de enlaces locais entre um roteador e um IP. Logo enlaces locais e por conseguintes

os IPs não fazem parte de uma rede assim definida.

Ou seja, um canal de  $C$  conecta dois roteadores da rede unidirecionalmente. Uma conexão bidirecional é um subconjunto de dois elementos  $E$  de  $C$ , com  $E=\{a, b\}$ . Se  $a=(r_x, r_y)$ , então  $b=(r_y, r_x)$ . Cada subconjunto de  $C$  com a forma de  $E$  corresponde a um enlace em redes malha 2D, onde um elemento de  $E$  representa, para cada roteador envolvido, um canal para recepção de pacotes e um canal para a transmissão de pacotes.

Uma aplicação pode ser caracterizada por um grafo dirigido  $A=\langle T, S \rangle$ , onde  $T=\{t_0, \dots, t_{k-1}\}$  representa o conjunto de tarefas da aplicação e  $S=\{s_{od}=\langle (t_o, t_d), w \rangle \mid t_o, t_d \in T \text{ e } w \in \mathbb{R}^+\}$  representa o conjunto de arcos que designa a comunicação entre pares de tarefas comunicantes. Aqui,  $k$  é o número de tarefas no qual a aplicação foi dividida, e  $S$  representa nos seus elementos cada fluxo unidirecional de informação que flui na rede. Um elemento  $s_{od}$  de  $S$  possui um par de tarefas origem e destino do fluxo e a taxa de média de transmissão para o fluxo. A Figura 3.25 ilustra um grafo de caracterização da aplicação. Neste exemplo,  $k=6$  e existem 9 elementos em  $S$ .

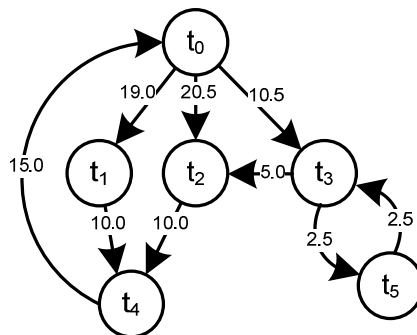


Figura 3.24

Figura 3.25 Exemplo de um grafo de aplicação hipotético, com taxas em Mbps.

Ao processo de definição de qual núcleo IP executará cada tarefa, seja uma tarefa de *hardware* ou de *software*, é dado o nome de *ligação de tarefas* (em inglês, *binding*). O processo de *mapeamento de tarefas* define qual a posição de cada tarefa e seu núcleo IP na infraestrutura de comunicação. Apesar de terem sido explorados e terem contribuído para a proposição deste trabalho, ambos os processos são aqui abstraídos. Com o objetivo de simplificar esta etapa, assume-se que a cada roteador, pertencente a  $R$  está associado um e somente um núcleo IP. Adicionalmente, a núcleo IP está ligado uma e somente uma tarefa, sendo que os núcleos IP são mono-tarefa. Dora-vante, sempre que se fizer referência a uma entidade ou par comunicante, estar-se-á referencian-

do o roteador  $r_o$  origem da comunicação e o roteador  $r_d$ , destino da comunicação.

Os algoritmos de roteamento permitem que uma ou mais direções sejam assumidas em cada roteador. No presente trabalho, o conjunto de direções possíveis  $D=\{d_{Leste}, d_{Oeste}, d_{Norte}, d_{Sul}\}$  limita-se àquelas definidas para o roteamento em redes de topologia malha 2D. Claramente, os conceitos podem ser estendidos para outras topologias, regulares ou não. Uma rota  $Rota_{od}=(d_1, \dots, d_n)$  de  $n$  hops entre dois roteadores  $r_o$  (i. e. o roteador de origem da comunicação) e  $r_d$  (i. e. o roteador destino da comunicação) é definida por uma  $n$ -tupla ordenada, uma sequência linear de  $n$  direções, sendo  $d_1$  a direção a ser tomada no roteador origem da comunicação e  $d_n$  a direção a ser tomada no roteador que precede o roteador destino. Dada uma rede  $N=<R, C>$ , uma rota  $R_{od}=(d_1, \dots, d_n)$  é *válida* se e somente se  $r_o \in R$ ,  $r_d \in R$  e um pacote saindo de  $r_o$  atinge  $r_d$ , após seguir exatamente as direções  $d_1, \dots, d_n$  nesta sequência. À sequência de direções estão relacionados os canais de rede que serão utilizados pela  $Rota_{od}$ . Uma rota  $Rota_{od}$ , tem as seguintes propriedades:

- O primeiro elemento  $d_0$  da tupla  $Rota_{od}$  define a direção a ser tomada em  $r_o$ , roteador origem da comunicação, e para qualquer rota válida,  $d_1 \in D$ ;
- O último elemento  $d_n$  da tupla  $Rota_{od}$  define a direção a ser tomada em um roteador que é vizinho do roteador destino da comunicação;
- $\forall d_1$  da tupla  $Rota_{od}$  assumido em  $r_o$ , está associada à utilização de um canal  $c_i=(r_o, r_x)$ ;
- $\forall d_n$  da tupla  $Rota_{od}$  assumido em  $r_y$ , leva à utilização  $c_y=(r_x, r_d)$ , onde  $r_d$  é o roteador destino do pacote.

Com base nas definições dadas na elaboração de rotas, tem-se que a sequência de canais a ser utilizada por uma rota  $Rota_{od}$  respeita algumas propriedades. Seja o primeiro canal alocado por uma rota  $Rota_{od}$   $c_0$  e o último canal alocado por uma rota  $Rota_{od}$   $c_n$ :

- $\forall c_p \neq c_n$ , tem-se que  $c_p=(r_a, r_b)$ , e  $c_{p+1}=(r_b, r_c)$ ;
- $\forall c_p \neq c_0$ , tem-se que  $c_{p-1}=(r_e, r_f)$ , e  $c_p=(r_f, r_g)$ .

Ou seja, dois canais consecutivos quaisquer utilizados em uma rota têm necessariamente um roteador em comum, que é destino do canal  $c_p$  e origem do canal  $c_{p+1}$  (ou, equivalentemente, o destino do canal  $c_{p-1}$  e a origem do canal  $c_p$ ).

Como dito anteriormente, os algoritmos de roteamento permitem que uma ou mais direções sejam tomadas em cada roteador. Com isto, diferentes opções de rotas podem existir para um mesmo par comunicante. Ao assumir-se uma das opções de rota para um dado par origem/destino, acumula-se as taxas médias de transmissão deste par (i.e. os valores de  $w$  do grafo de aplicação) aos canais a serem utilizados por esta rota. Se para cada par de núcleos IP que se comunica for associada uma e somente uma rota, a combinação destas define um cenário de rotas. Formalmente, dada uma rede  $N$ , um cenário de rotas  $Scen$  é um conjunto de rotas definido sobre  $N$  tal que:

$\forall r_o, r_d \in R$  existe exatamente uma rota  $Rota_{od}$  em  $Scen$  ou não existe nenhuma  $Rota_{od}$  em  $Scen$ .

Define-se naturalmente que, se um núcleo IP conectado a  $r_o$  não se comunica com o núcleo IP conectado a  $r_d$  então não existe uma  $Rota_{od}$  em  $Scen$ .

A combinação de todas as opções de rota de todos pares comunicantes em uma aplicação ligada e mapeada em um sistema que emprega uma NoC como arquitetura de comunicação define o conjunto universo de cenários de rotas. O conjunto de todos cenários de rotas tem sua cardinalidade definida pela Equação 3.4. Nesta equação,  $N_{Scen}$  é o número máximo de cenários de rota,  $N_{Pares}$  é o número de pares comunicantes e  $N_{Rotas}$  é o número de rotas existente para cada par. Por exemplo, assumindo-se uma rede de topologia malha 2D e de dimensões 2x2, onde todos os roteadores originam comunicação ( $N_{Origem}$ ) e cada um destes tem como destino todos os outros roteadores ( $N_{Destino}$ ), então  $N_{Pares}$  é igual a 12 (i.e para  $N_{Origem}=4$  e  $N_{Destino}=3$ , tem-se  $N_{Pares}=N_{Origem} * N_{Destino}=4*3=12$ ). Supondo-se que 7 pares têm apenas 1 opção de rota entre eles e 5 pares têm 2 opções de rota, o número máximo de cenários de rotas neste caso é definido pelo produto de  $N_{Rotas}$  de cada par comunicante, conforme Equação 3.4.

$$\text{Equação 3.4} \quad N_{Scen} = \prod_{i=1}^{N_{Pares}} N_{Rotas}(i)$$

### 3.3.2 EXPLORAÇÃO DE CENÁRIOS DE ROTAS

Somente são considerados cenários de rotas válidos aqueles onde garantidamente não é superada a capacidade de transmissão de qualquer canal de comunicação da rede. A otimização que se busca é a redução da ocupação de canais de comunicação, com o objetivo de aumentar o desempenho de uso da rede em chip através de uma melhor distribuição das cargas de comunicação e abrandar potenciais pontos quentes. Para atingir tal otimização, deve-se escolher a rota mais adequada para cada um dos pares comunicantes. Uma boa rota é aqui definida como aquela com a qual se obtém a menor taxa média de transmissão nos canais a serem utilizados, com menores picos de uso entre o par origem/destino e com a menor distância em *hops*. Esta definição foi assumida a partir dos seguintes pressupostos:

- Quanto maior a taxa de transmissão em um dado canal, menor a probabilidade de encontrar este canal livre. Pressupondo um canal com capacidade de transmissão de 200 Mbps e uma ocupação de 100 Mbps, dada pela soma de fluxos que passam por ali, o canal estará 50% do tempo ocupado transmitindo dados e 50% do tempo livre. Quanto mais próximo de 100% a ocupação do canal, maior será a probabilidade dos pacotes que trafegam por ali encontrarem-no ocupado. Esta tentativa de uso do canal ocupado acarreta aumento da latência de pacotes e contribui para o aumento do congestionamento na rede;
- Quanto mais baixa a taxa média de transmissão e menor o pico de transmissão de canais na rota, menor a probabilidade de se encontrar concorrência durante a transmissão de pacotes;
- Finalmente, quanto mais baixa a taxa média de transmissão de canais e menor o pico de transmissão e menor a distância entre a origem e o destino, menor será a latência de pacotes que trafegarão pela rota escolhida.

A exploração dos cenários de rotas pode ser realizada através de busca exaustiva. Esta abordagem efetivamente descobre o melhor cenário de rotas existente, visto que todos os cenários de rotas possíveis são avaliados. Contudo, seu emprego torna-se inviável quando a quantidade de cenários de rotas cresce demais. Assim, é importante utilizar algoritmos que garantam um resultado eficiente com tempo reduzido. Os algoritmos *Breadth first* e o algoritmo de Dijkstra são adotados em [KAV06] para a seleção de cenários de rotas. O algoritmo *Breadth first* realiza tal busca sem levar em consideração o estado de ocupação de rede. Já o algoritmo de Dijkstra leva em considera-

ção este estado. No presente trabalho, os algoritmos de avaliação de cenários de distribuição de rotas adotados respeitam o conceito adotado no algoritmo de Dijkstra, Três algoritmos são utilizados, quais sejam: (i) uma solução aleatória, (ii) um algoritmo heurístico e (iii) um algoritmo do tipo *simulated annealing*. Para todos os algoritmos de busca de melhor cenário de rotas foram adotadas duas métricas de comparação, quais sejam: (i) a média das taxas médias de transmissão de todos os canais a serem utilizados na rede e (ii) o pico de taxa média de transmissão. O cenário de rotas a ser escolhido é o que apresenta menor média e menor pico.

Na *solução randômica*, gera-se um número limitado de cenários de rotas. Na *solução heurística*, representada no Algoritmo 3.1, um cenário de rotas inicial é gerado através da seleção aleatória de uma rota para cada par comunicante. Em seguida, os pares têm suas rotas avaliadas. A escolha da melhor rota a ser utilizada para cada par comunicante passa pela avaliação das opções de rota, pela análise da taxa média de transmissão que será associada aos canais, pelo pico de taxa média de transmissão nos canais e pela distância em *hops* definida pela rota que será empregada pelo par origem/destino. Ao finalizar a avaliação de todos os pares origem/destino, uma nova avaliação pode ser realizada para analisar se outras rotas não são mais interessantes do que as atualmente definidas no cenário de rotas. Dois critérios são adotados para encerrar este algoritmo. O primeiro é a superação de um limite estabelecido *a priori*. O segundo ocorre quando nenhuma rota mais otimizada pode ser encontrada após avaliação de todos pares comunicantes.

Algoritmo 3.1 - Algoritmo heurístico de avaliação de cenários de rotas.

---

```

1 // Cria um cenário de rotas inicial
2 CENARIO_DE_ROTAS=GeraCenarioDeRotas(Aleatório);
3 LIMITE=ValorPreestabelecido;
4
5 // Controla o laço de otimização
6 ENQUANTO (LIMITE>ZERO) FAÇA
7     HouveTroca=FALSO;
8
9 // Avalia cada par comunicante individualmente
10 PARA (ParComunicante na ListaDeParesComunicantes) FAÇA
11     ROTA_A=RotaEmUso(CENARIO_DE_ROTAS, ParComunicante);
12     PARA (ROTA_B na ListaDeOpcoesDeRota(ParComunicante)) FAÇA
13
14 // Se a nova rota é melhor, gera um novo cenário de rotas
15 SE (ROTA_B é melhor que ROTA_A) ENTÃO
16     CENARIO_DE_ROTAS=TrocaDeRotas(ROTA_A por ROTA_B);
17     ROTA_A=ROTA_B;
18     HouveTroca=VERDADEIRO;
19 FIM SE;
20
21 FIM PARA;
22 FIM PARA;
23
24 // Critério de parada da otimização
25 SE HouveTroca==FALSO ENTAO retorna(CENARIO_DE_ROTAS);
26 SENAO LIMITE=LIMITE-1;
27 FIM SE;
28
29 FIM ENQUANTO;
30 retorna(CENARIO_DE_ROTAS);

```

---

Na *solução simulated annealing*, geram-se diferentes cenários de rotas iniciais. Para cada

cenário de rotas gerado, busca-se a otimização deste através da alternância de rotas para cada par origem/destino. Como proposto para os algoritmos *simulated annealing*, soluções inicialmente não favoráveis podem ser aceitas, desde que estejam dentro de uma faixa pré-estabelecida, tal como até 30% de piora sobre o melhor resultado obtido até então. Esta faixa é gradativamente reduzida com o objetivo de evitar resultados que desviem do objetivo final, o de otimização do cenário. Não podendo mais ser otimizado, este cenário é armazenado se os critérios apontados anteriormente o definem como o melhor cenário de rotas. Finalizada esta etapa de otimização, um novo cenário de rotas é gerado e o processo de otimização reinicia. O melhor cenário de rotas é entregue quando for excedida uma quantidade pré-estabelecida de buscas.

## 4. AVALIAÇÃO DE RESULTADOS

Neste Capítulo apresentam-se os experimentos conduzidos e os resultados obtidos para o mapeamento de rotas. Três conjuntos de experimentos foram conduzidos, sendo que no primeiro faz-se uma comparação da latência de entrega dos pacotes e o consumo de área entre as NoC Hermes e Hermes-SR. No segundo faz-se uma comparação de resultados obtidos quando os diferentes algoritmos adaptativos de roteamento são adotados como base para a definição de um cenário de rotas. Os resultados apresentados neste segundo conjunto de experimentos incluem estimativas de distribuição de cargas na rede, estimativas de ocupação de canais, as latências médias associadas às distâncias entre pares comunicantes e o intervalo de latências. Os experimentos aqui conduzidos têm por objetivo validar a NoC Hermes-SR e avaliar o uso de algoritmos adaptativos de roteamento como base eficaz para definição de rotas alternativas de comunicação em redes com roteamento na origem. No último conjunto de experimentos conduzidos fez-se uma comparação entre os resultados de latência da solução onde foi aplicado o mapeamento de rotas (decisão em tempo de projeto) com soluções onde empregou-se o algoritmo adaptativo (decisões em tempo de execução).

### 4.1 INFRAESTRUTURA DE COMUNICAÇÃO

O primeiro conjunto de experimentos conduzido teve dois objetivos: *(i)* validar a funcionalidade da NoC Hermes-SR e *(ii)* comparar o desempenho desta com o da NoC Hermes. Empregou-se a ferramenta Atlas [GAP09], uma evolução da ferramenta Maia [OST05], como suporte à geração das estruturas usadas nos experimentos. Enquanto o ambiente Maia [OST05] tem suporte limitado à NoC Hermes, o ambiente Atlas foi concebido para habilitar a integração de recursos para automatizar os processos de geração, geração de estímulos, simulação e análise de desempenho de múltiplas NoCs. Apesar de não trivial, o processo de integrar uma nova arquitetura de NoC ao ambiente Atlas é simplificado pela estrutura de código aberto e o projeto visando a expansibilidade do arcabouço de programação do ambiente Atlas. Dentro do escopo deste trabalho inseriu-se o suporte à NoC Hermes-SR como parte das arquiteturas de NoC que podem ser operadas a partir do ambiente Atlas. Isto possibilita: *(i)* gerar automaticamente descrições em VHDL RTL de instâncias da NoC Hermes-SR; *(ii)* gerar cenários de tráfego de simulação para esta rede; *(iii)* simular a rede



com os cenários de tráfego produzidos; (iv) e avaliar os resultados a partir de diversos recursos textuais e gráficos disponíveis no ambiente Atlas.

#### 4.1.1 EXPERIMENTOS CONDUZIDOS

Neste experimento, o parâmetro tamanho da fila de entrada dos roteadores foi variado a partir da ferramenta de geração automática do ambiente Atlas para as NoCs Hermes e Hermes-SR, resultando em diversas configurações distintas para cada rede. Os parâmetros a seguir foram fixados para a comparação das NoCs:

- Dimensão da rede: 5x5;
- Tamanho de *flit*: 16 bits;
- Controle de fluxo: baseado em créditos;
- Uso de canais virtuais: não;
- Roteamento: XY puro (determinístico).

Cenários de rotas são gerados e armazenados em um arquivo externo à descrição da NoC Hermes-SR denominado arquivo de rotas (via ambiente Atlas). Este arquivo é usado durante a simulação, sendo lido por módulos descritos em SystemC (produzidos pelo ambiente Atlas). Neste experimento, as rotas foram produzidas tomando por base o algoritmo de roteamento XY puro, para viabilizar uma comparação coerente das infraestruturas de comunicação Hermes e Hermes-SR. Assim, a rota de cada par comunicante é exatamente a mesma nas duas NoCs. O desempenho das infraestruturas de comunicação foi medido para os diferentes tamanhos de fila de entrada. Este parâmetro (i.e. tamanho da fila de entrada) é comum as duas NoCs e influencia diretamente os fenômenos de congestionamento. Quatro descrições diferentes de cada NoC resultaram da configuração do tamanho das filas de entrada, variado entre 4, 8, 16 e 32 posições.

Para a avaliação das NoCs definiram-se cinco cenários de tráfego. Em cada cenário adotou-se taxas distintas de inserção de pacotes (i. e. distribuição temporal) e distribuição espacial uniforme de tráfego. As taxas de inserção usadas foram 10%, 20%, 30%, 40% e 50% da capacidade máxima de cada canal da NoC. Como se emprega uma frequência de operação padronizada em 50MHz, controle de fluxo baseado em crédito (onde 1 *flit* pode ser transmitido a cada ciclo de relógio) e tamanho de *flit* igual a 16 bits, tem-se que a capacidade máxima de cada canal é igual a 800 Mbps

(de  $50\text{MHz} \times 16 \text{ bits} = 800\text{Mbps}$ ). Os percentuais definidos anteriormente para a geração de cenários de tráfego podem ser traduzidos como taxas de injeção de 80Mbps, 160Mbps, 240Mbps, 320Mbps e 400Mbps. Os pares origem/destino foram definidos de forma aleatória para os cenários de tráfego deste experimento, garantindo-se que para todo par o destino é diferente da origem. Definiu-se ainda que pacotes tenham tamanho *em torno* de 20 *flits*. A imprecisão deve-se às diferenças nos tamanhos dos campos que identificam o destino para as NoC Hermes e Hermes-SR. Na primeira, usa-se exatamente 1 *flit* para identificar o endereço do roteador destino, enquanto que na Hermes-SR são necessários ao menos 2 *flits*, conforme já descrito na Seção 3.1.1.

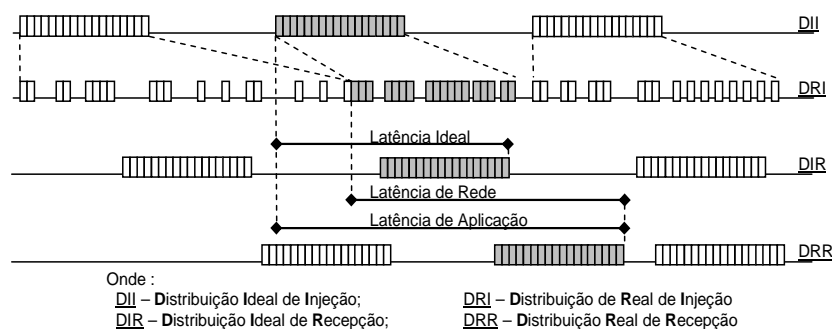


Figura 4.1 - Métricas distintas de latência.

Para a avaliação de desempenho baseado em simulações RTL (empregando o simulador Modelsim executado de dentro do ambiente Atlas), adotou-se como medida a latência dos pacotes. Todavia, a métrica de latência pode ser observada de diferentes formas, conforme ilustra a Figura 4.1. Os valores de latência aqui apresentados não se limitam ao atraso de transmissão na rede. A Figura 4.1 diferencia as latências de transmissão de acordo com a distribuição da injeção e recepção. A *distribuição de injeção ideal* (DII) é especificada no arquivo texto contendo o cenário de tráfego e descreve os momentos ideais planejados para injeção de cada pacote [TED05]. A *distribuição real de injeção* (DRI) considera o momento exato de inserção de um pacote na rede, que pode ser diferente daquele definido na DII devido à ocorrência de contenção em tempo de simulação. A *distribuição ideal de recepção* (DIR) representa os momentos estimados de entrega dos pacotes. A *distribuição real de recepção* (DRR) revela os momentos reais de entrega de pacotes em seus destinos. A Figura 4.1 mostra distribuições ilustrativas de tais cenários de injeção e recepção. A *latência ideal* é o número mínimo de ciclos de relógio que um pacote precisa para alcançar seu destino. Este valor é obtido a partir da diferença entre o momento ideal de injeção do pacote e o momento esperado de entrega do mesmo. A *latência de rede* é o atraso de transmissão de um pa-

cote durante seu trânsito da origem até o destino, que pode ser influenciado pela competição por recursos da NoC (e. g. canais, filas de armazenamento, arbitragem, roteamento). A *latência de aplicação* ilustra o impacto mais importante no desempenho ideal de uma comunicação. Ela é computada como a diferença entre o momento ideal de injeção de pacotes e seu momento efetivo de entrega no destino. A latência de aplicação é assumida como métrica de comparação neste trabalho.

#### 4.1.2 RESULTADOS OBTIDOS

A partir da configuração das NoCs e da geração dos cenários de tráfego, a ferramenta *Modelsim* da Mentor foi utilizada para simulação das NoCs. A Figura 4.2 apresenta os resultados de distribuição de latências obtidos com diferentes cenários tráfego e diferentes NoCs.

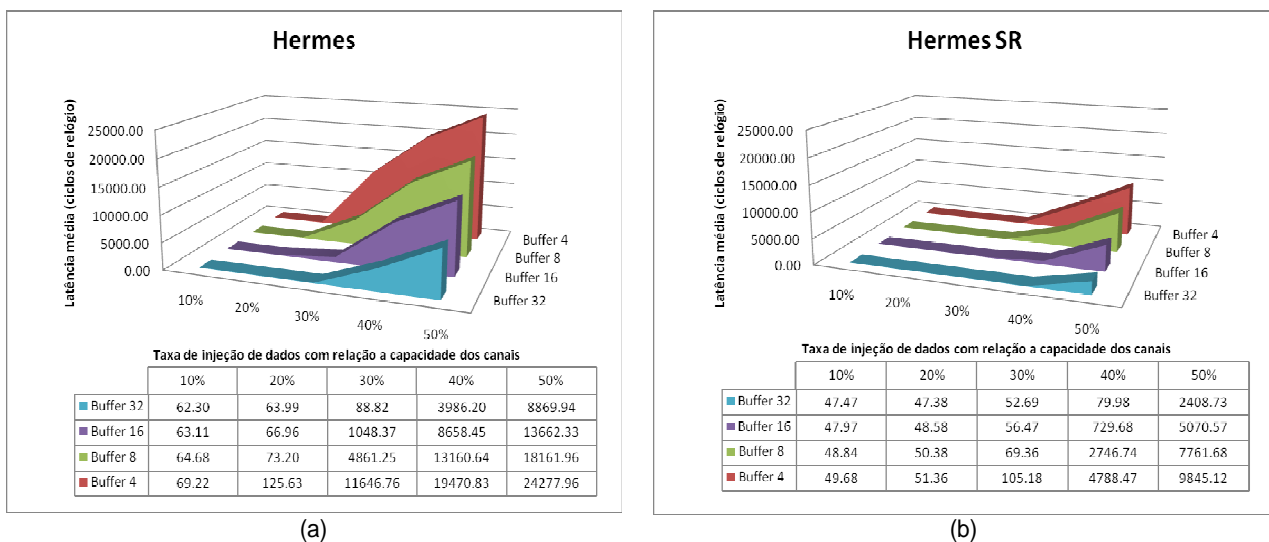


Figura 4.2 – Latência média obtida para as NoCs Hermes (a) e Hermes-SR (b) variando-se o tamanho da fila de armazenamento.

Na Figura 4.2 pode-se observar que as latências máximas alcançada com a NoC Hermes-SR são sempre inferiores às alcançadas na Hermes original. Isto se deve a três fatores: (i) a diminuição do tempo para encaminhar um pacote da porta de entrada para a porta de saída em um roteador, (ii) a adoção de um algoritmo de arbitragem mais eficiente e principalmente, (iii) a descentralização do mecanismo de arbitragem. Ao chegar um pacote em um roteador da NoC Hermes, são necessários entre 6 e 7 ciclos para a transmissão do primeiro *flit*. Na NoC Hermes SR a mesma operação requer entre 2 e 3 ciclos. Quanto ao algoritmo de arbitragem, a NoC Hermes utiliza *Round Robin*, que pode causar postergação indefinida para a transmissão de um pacote. Já na Hermes-SR o algoritmo FCFS torna o pior caso de espera na transmissão previsível, visto que todas as portas têm a mesma probabilidade de arbitragem, característica garantida pelo enfileiramento

de requisições que garante o atendimento destas de forma ordenada.

Pode-se observar na Figura 4.2 que a NoC Hermes-SR com filas de entrada de tamanho 4 proporciona latência média de transmissão similar àquela obtida com a NoC Hermes com 32 posições para a fila de entrada. A implementação inicial da NoC Hermes-SR mostrou-se eficiente, pois empregando tamanhos menores de filas de armazenamento proporcionou latências máximas e quiparáveis àquelas obtidas com a NoC Hermes. Enfatizando o critério desempenho, isto compensa o maior custo em área de *hardware* quando comparada à NoC Hermes com mesmo tamanho de fila de armazenamento. A Tabela 4.1 resume os valores de custo em área quando da síntese do roteador central das NoCs para um dispositivo do tipo FPGA da Xilinx.

Tabela 4.1 –Comparação de área ocupada por roteadores centrais das NoCs Hermes e Hermes-SR em FPGA. O dispositivo alvo usado foi um FPGA da família Virtex-II PRO, o xc2vp30-7. Resultados da síntese utilizando-se a ferramenta XST do ambiente ISE versão 9.1i.

	NoC Hermes		NoC Hermes-SR	
	LUTs	Portas lógicas equivalentes	LUTs	Portas lógicas equivalentes
Tamanho de fila 4	683	26,858	897	33,464
Tamanho de fila 8	644	26,469	986	33,876
Tamanho de fila 16	683	26,882	986	34,407
Tamanho de fila 32	782	27,711	1,046	34,987

## 4.2 ALGORITMOS USADOS COMO BASE PARA DEFINIÇÃO DE ROTAS

No segundo conjunto de experimentos é feita a comparação dos diferentes algoritmos de roteamento usados como base para o mapeamento de rotas e a consequente definição de cenários de rotas. O conjunto de algoritmos de roteamento experimentados corresponde àqueles detalhados no Capítulo 3: XY puro (*xy*), *negative first* não mínimo (*nfnm*), *negative first* mínimo (*nfm*), *North last* não mínimo (*nlnm*), *North last* mínimo (*nlm*), *West first* não mínimo (*wfnm*) e *West first* mínimo (*wfm*). Para todos experimentos conduzidos ao longo desta Seção utilizou-se unicamente a NoC Hermes-SR. O objetivo é realizar uma comparação dos valores de latência para os diferentes capturados para os diferentes algoritmos de roteamento citados.

### 4.2.1 EXPERIMENTOS CONDUZIDOS

Este experimento compõe-se de cinco passos: (i) geração da NoC Hermes-SR, (ii) definição dos cenários de comunicação (i. e. que pares se comunicantes e com que características), (iii) ma-

peamento estático das rotas de comunicação entre os pares origem/destino para a NoC Hermes-SR, (iv) simulação e (v) avaliação de resultados.

Para a geração da NoC Hermes-SR, manteve-se a configuração do experimento anterior, ou seja: NoC com dimensão 5x5 (5 linhas e 5 colunas), tamanho de *flit* igual a 16 bits, controle de fluxo baseado em créditos, sem emprego de canais virtuais. Quanto aos cenários de comunicação, três categorias foram geradas: (i) um cenário de transmissão *um para todos*, a partir de cada ponto da rede, (ii) um cenário com possível sobrecarga em alguns canais de rede ao adotar o algoritmo XY puro e (iii) um cenário concentrando os destinos de transmissão em dois pontos da rede. Nas duas primeiras categorias, adotaram-se taxas de injeção de tráfego na rede de 10% da capacidade máxima de um canal, conforme explicado na Seção 4.1.1. Na terceira categoria de cenários adotaram-se taxas de injeção de pacotes de 40% da capacidade de um canal. Os cenários de tráfego sintéticos descritos são de difícil associação com cenários reais de tráfego, todavia são bastante úteis para a exploração de casos extremos de utilização das redes em chip.

Durante a criação dos cenários de tráfego com o ambiente Atlas, gera-se automaticamente um arquivo contendo a informação sobre os pares comunicantes e a taxa média de transmissão exigida entre eles. Este arquivo, que descreve um grafo de comunicação da aplicação (em inglês, *Application Communication Graph*, ACG), é usado como entrada de uma ferramenta de mapeamento de rotas entre os pares que se comunicarão na NoC Hermes-SR. Para cada cenário de tráfego gera-se um cenário de rotas com cada um dos algoritmos de roteamento, totalizando sete cenários de rotas por cenário de tráfego. Os cenários de rota resultantes não são necessariamente distintos, mas a sua computação obrigatoriamente é diferente. Assim, um cenário de rotas definido com o algoritmo *West first* pode ser igual a um computado com o *North last* ou com o *negative first*.

A simulação foi realizada com a ferramenta *Modelsim*, tendo sido automaticamente executada a partir do ambiente Atlas. Para dar início à simulação deve ser definida a infraestrutura de comunicação (i. e. a NoC e suas configurações), criado o cenário de tráfego e gerando o arquivo que descreve o cenário de rotas entre os pares comunicantes. Todo o controle da simulação é realizado com módulos de validação funcional (i. e. *testbenches*) descritos em SystemC. Tais módulos usam como entrada os arquivos que descrevem os cenários de tráfego e os arquivos de cenário de rotas. Como saída, geram-se arquivos de saída contendo os resultados da simulação. Estes contêm

como informações principais o momento ideal de inserção de cada pacote na rede e o momento em que este atingiu ao destino.

O potencial de distribuição das cargas de comunicação pela rede a partir do uso dos diferentes algoritmos de roteamento é uma estimativa comum a todos os experimentos aqui realizados. Para esta avaliação foram observados os dados de latência de aplicação obtidos quando da transmissão dos pacotes na rede e os picos de taxa de utilização dos canais empregados na transmissão de pacotes entre pares origem/destino.

#### 4.2.2 RESULTADOS OBTIDOS

O primeiro cenário de tráfego é compatível com aquele adotado no primeiro experimento, ou seja, cada ponto da rede envia pacotes para um conjunto aleatório de outros destinos. Apresenta-se aqui um cenário de tráfego onde a taxa de injeção de pacotes de cada origem de comunicação, injetora de pacotes, foi configurada para 10% da capacidade do canal. Após a geração do cenário de comunicação utilizou-se uma ferramenta desenvolvida no escopo deste trabalho que é capaz de definir cenários de rotas a serem empregados durante a transmissão de pacotes entre pares comunicantes. A Figura 4.3 apresenta a estimativa de distribuição das cargas resultante do cenário de tráfego sobre os canais de comunicação da NoC Hermes-SR para os diferentes cenários de rotas computados com os algoritmos de roteamento.

Nota-se na Figura 4.3 a estimativa de uma má distribuição das cargas da comunicação para este cenário de tráfego quando utilizando-se o algoritmo de roteamento *negative first* não mínimo (*nfnm*) e *negative first* mínimo (*nfm*), observada naquela figura pela alta taxa média de utilização estimada no canto inferior esquerdo da NoC para tais algoritmos. Os demais algoritmos mostraram-se eficazes, conforme pode-se observar na Figura 4.3 a distribuição uniformidade de cargas de comunicação alcançada para *xy*, *wfnm*, *wfm*, *nlm*, *nlnm*. Na mesma figura em *nfnm* e *nfm* é possível notar picos mais altos em relação aos demais algoritmos estimados na utilização de canais na base e na lateral esquerda da infraestrutura de comunicação ali representada.

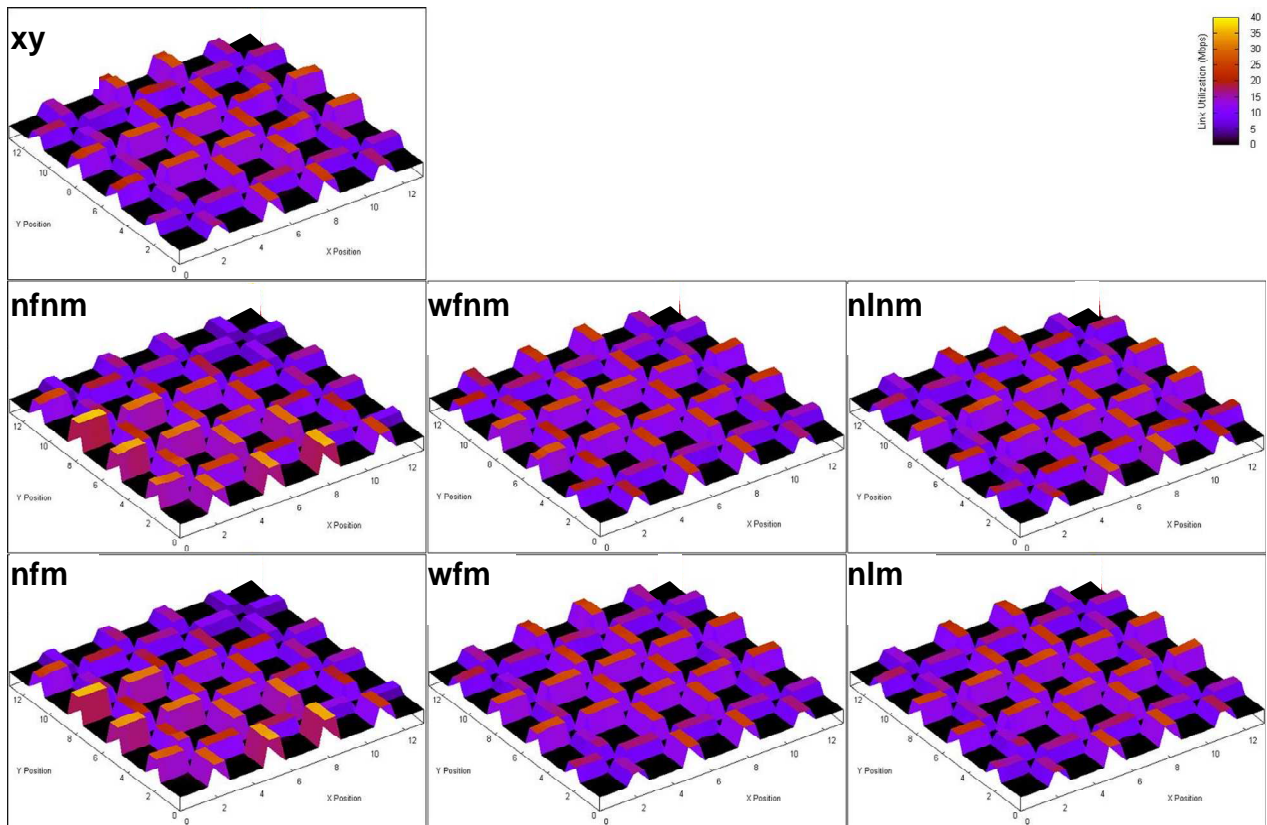


Figura 4.3 - Estimativas de distribuição de carga para o cenário de tráfego todos para todos.

Para a dimensão de NoC malha 2D adotada para esta avaliação (i.e. 5x5) estão disponíveis 80 canais para serem utilizados na transmissão de pacotes. Neste primeiro cenário, todos os canais foram utilizados independente do algoritmo de roteamento adotado para definição do cenário de rotas. Na Figura 4.4 são apresentadas as estimativas de ocupação dos canais em Mbps. No gráfico, a ordenação é dada pela taxa de ocupação estimada por canal, independente da localização do mesmo na rede. Comparado aos demais algoritmos é possível notar que os algoritmos *nfnm* e *nfm* possuem aproximadamente 25 roteadores com estimativa de ocupação superior e aproximadamente 15 canais com estimativa de ocupação inferior aos demais algoritmos, o que caracteriza a má distribuição das cargas na rede visto que há canais muito e outros pouco utilizados. A exceção dos algoritmos *nfnm* e *nfm*, os demais algoritmos mostraram permitir distribuição de cargas bastante similar no cenário proposto.

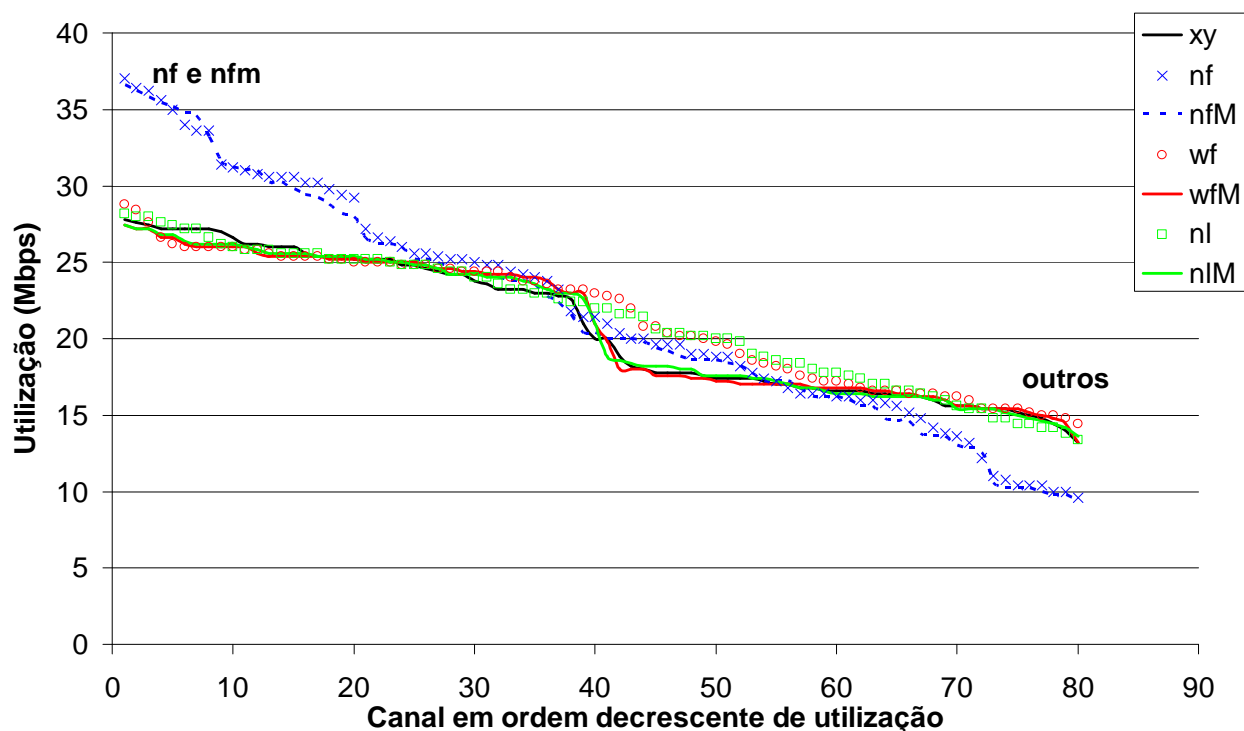


Figura 4.4 - Estimativa de ocupação de canais para o cenário de tráfego todos para todos.

A partir da avaliação das estimativas de distribuição da carga e de ocupação dos canais, foram realizadas simulações, sendo os resultados são apresentados na Tabela 4.2. Nela apresenta-se a latência média obtida quando da simulação utilizando os diferentes cenários de rotas obtidos com os diferentes algoritmos de roteamento. Naquela tabela, diferenciam-se os valores pela distância em *hops*, entre os pares comunicantes, conforme listado na primeira coluna. Na segunda coluna apresenta-se o número de ocorrências de pares comunicantes correspondente às distâncias apresentadas na primeira coluna. As demais colunas da tabela apresentam os resultados de latência média quando os diferentes algoritmos de roteamento são utilizados na criação dos cenários de rotas. Observa-se que os algoritmos *nfnm* e *nfm* apresentaram os resultados menos satisfatórios. Até 2 *hops*, *xy* mostra-se algoritmo mais eficaz quando comparados aos demais. A partir de 3 *hops*, os algoritmos *nlm*, *wf*, *wfM* e *wf* apresentam-se mais eficientes que o *xy* na distribuição de cargas para tal cenário de tráfego. Apesar disto, a última linha reporta que o algoritmo *xy* proporcionou a menor latência média. Isto se deve ao fato de que 56% dos pares têm distância de até 3 *hop*, sendo lá o principal ganho do algoritmo *xy*.



Tabela 4.2 - Comparação de latência média para o cenário de tráfego todos para todos.

Hops	Ocorrência	Latência média						
		xy	nfnm	nfm	nlnm	nlm	wfnm	wfm
1	80	<b>25,48</b>	26,35	61,34	53,29	53,23	53,59	53,44
2	123	<b>36,92</b>	42,77	58,76	51,08	50,97	52,44	51,03
3	131	<b>48,01</b>	61,42	54,48	<b>45,46</b>	<b>44,89</b>	<b>47,56</b>	<b>44,71</b>
4	119	56,97	71,54	55,88	<b>47,58</b>	<b>48,03</b>	<b>49,19</b>	<b>46,47</b>
5	79	65,97	84,65	71,05	<b>59,11</b>	<b>59,04</b>	<b>58,52</b>	<b>55,42</b>
6	40	71,73	90,33	66,38	<b>49,05</b>	<b>48,75</b>	<b>52,43</b>	<b>48,35</b>
7	14	82,93	98,57	79,79	<b>63,36</b>	<b>62,29</b>	<b>66,43</b>	<b>61,21</b>
8	4	84,00	90,75	39,25	<b>57,25</b>	<b>57,25</b>	<b>64,50</b>	<b>51,75</b>
Média		49,54	60,97	60,11	50,70	50,57	52,08	49,69

Na Figura 4.5, os eixos representam as distâncias em *hops* possíveis entre os pares origem destino e os valores representam a latência média quando do emprego de cada um dos algoritmos de roteamento. Conforme pode ser visto, o algoritmo *xy* garante menor latência quando a distância não ultrapassa 3 *hops*. Com o aumento da distância, pode-se notar a perda gradual da vantagem do uso deste algoritmo. A exceção dos algoritmos *nfnm* e *nfm* os demais algoritmos adaptativos utilizados como base para a computação de rotas mostraram bastante estabilidade nas latências, garantindo equivalentes das mesmas, independente da distância em *hops*.

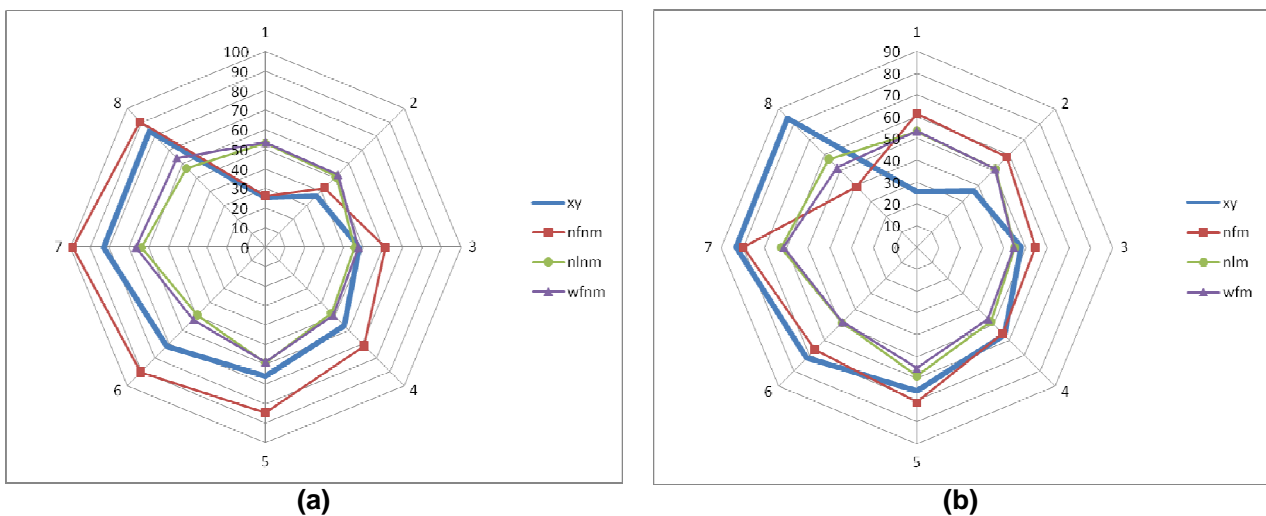


Figura 4.5 - Comparação de latências médias para os algoritmos não mínimos (a) e mínimos (b).

A última avaliação realizada para este cenário de tráfego separa em intervalos às latências obtidas para a transmissão de todos os pacotes pela rede. A Figura 4.6 ilustra o intervalo de latências obtidas com os diferentes cenários de rotas, destacado à direita da figura. Os valores percentuais representam a quantidade transmitida de pacotes ao final de cada intervalo. Assim, pode-se observar que os cenários de rotas definidos algoritmos *nfnm* e *nfm*, garantem que aproximadamente 50% dos pacotes transmitidos tiveram latência máxima dentro do primeiro e segundo inter-

valo. Já para os cenários de rotas definidos com os demais algoritmos, este percentual fica em torno de 70%. Adicionalmente, pode-se observar que para este cenário de tráfego, a exceção dos algoritmos *nfnm* e *nfm* todos os demais algoritmos propiciam aproximadamente 100% dos pacotes transmitidos até o sexto intervalo de latência.

Os resultados obtidos para o cenário de tráfego todos para todos permitem concluir que os esforços o algoritmo *xy* garante a menor taxa média com esforço mínimo de computação do cenário de rotas. Porém, a exceção dos algoritmos *nfnm* e *nfm*, o uso dos demais algoritmos garante uma distribuição de cargas na rede tão boa quanto aquela obtida com *xy* além de garantir latências médias similares entre os pares comunicantes, independente de sua distância em *hops*.

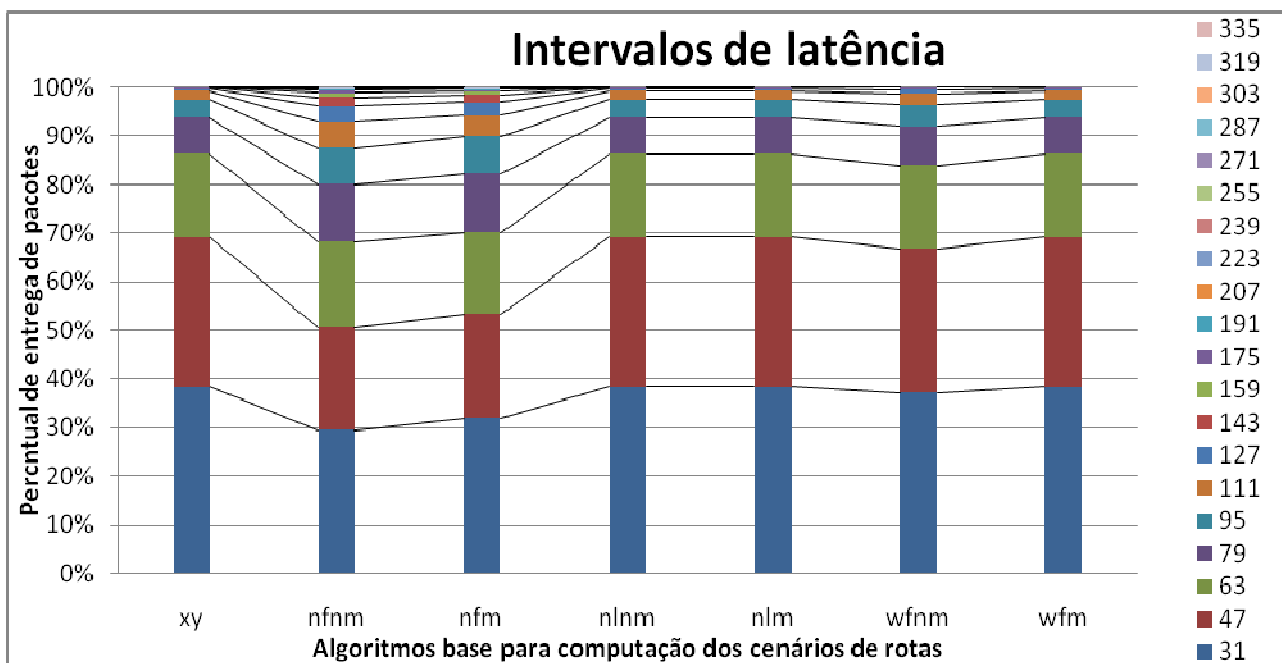


Figura 4.6 – Intervalo de latência e percentual de entrega no cenário de tráfego todos para todos.

A próxima categoria de cenários de tráfego explora a eficiência dos algoritmos adaptativos na elaboração do cenário de rotas, quando comparados ao algoritmo *xy*. Supõem-se cenários de tráfego onde pares comunicantes estão localizados nas bordas da infraestrutura de comunicação. A Figura 4.7 apresenta 4 cenários de tráfego que foram gerados e simulados. As rotas representadas na Figura 4.7 ilustram a sobreposição destas quando adotando-se o algoritmo *xy* para elaboração do cenário de rotas. A taxa de transmissão assumida entre os pares comunicantes é de 100 Mbps, ou seja, 12,5% da capacidade máxima de um canal de transmissão da NoC.

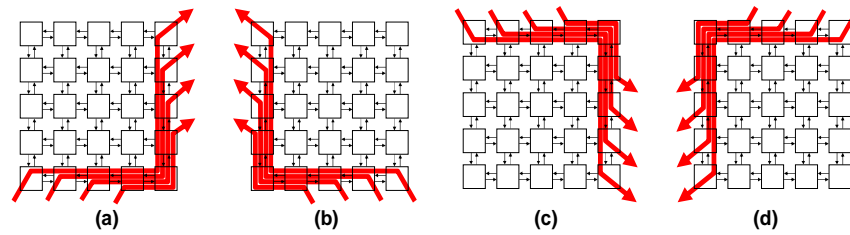


Figura 4.7 – Cenários de tráfego com pares comunicantes nas bordas explorando sobreposição de rotas.

O cenário de tráfego avaliado é aquele ilustrado na Figura 4.7(c), ilustrando uma situação em que todos os algoritmos adaptativos, quando adotados como base para o cálculo de rotas, permitem alcançar resultados idênticos ou melhores do que aqueles com o algoritmo *xy*. O processo seguiu com apresentado no experimento anterior, ou seja, um arquivo contendo a descrição do grafo de comunicação foi utilizado como entrada para a elaboração dos cenários de rotas e, a partir daí, capturou-se um conjunto de estimativas iniciais. A Figura 4.8 apresenta a estimativa de distribuição de cargas para o cenário de tráfego sob avaliação para os cenários de rotas definidos com os diferentes algoritmos de roteamento. Pode-se observar que, como esperado, o algoritmo *xy* concentrou a comunicação nas bordas enquanto os demais algoritmos distribuíram a carga equivalentemente pela rede.

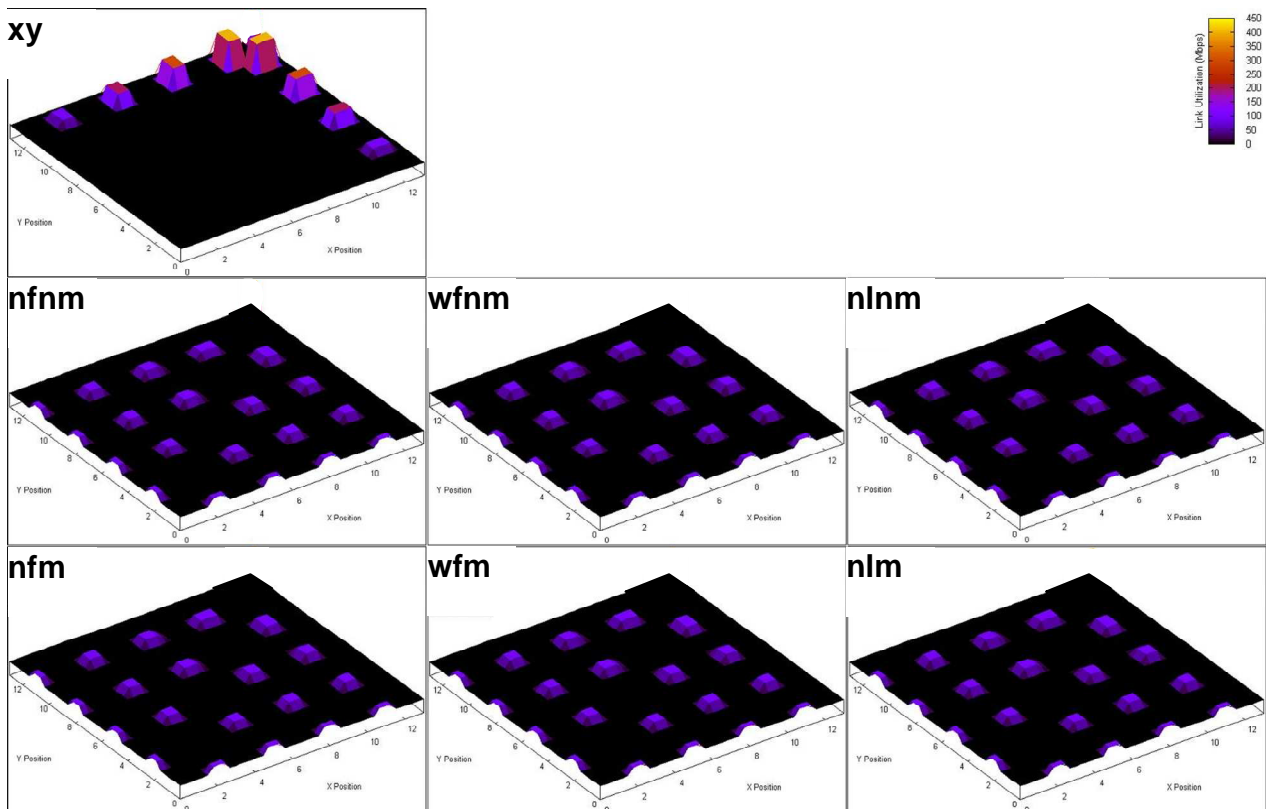


Figura 4.8 - Estimativa de distribuição de carga para o cenário de tráfego sobreposição de rotas.

A Figura 4.9 detalha as estimativas de uso dos canais da rede, ordenando os resultados de forma decrescente de acordo com a estimativa de ocupação, sem considerar a localização do canal na infraestrutura de comunicação. É possível observar que o algoritmo *xy* leva a uma má distribuição de cargas, visto que a estimativa de ocupação dos canais chega a ser 4 vezes maior quando comparada as estimativas de ocupação para os demais algoritmos. Assim, são previstos canais com cargas até 50% de sua capacidade máxima quando adotado o algoritmo *xy*, enquanto que para os demais algoritmos estimou-se ocupações máximas de 12,5%. Esta variação entre as cargas deve-se a distribuição das cargas de comunicação pelos canais de rede. O algoritmo *xy* distribuiu esta carga por 8 canais, enquanto os demais o fizeram sobre 20 canais. Estes últimos valores mostram a flexibilidade dos algoritmos adaptativos na busca de caminhos menos congestionados.

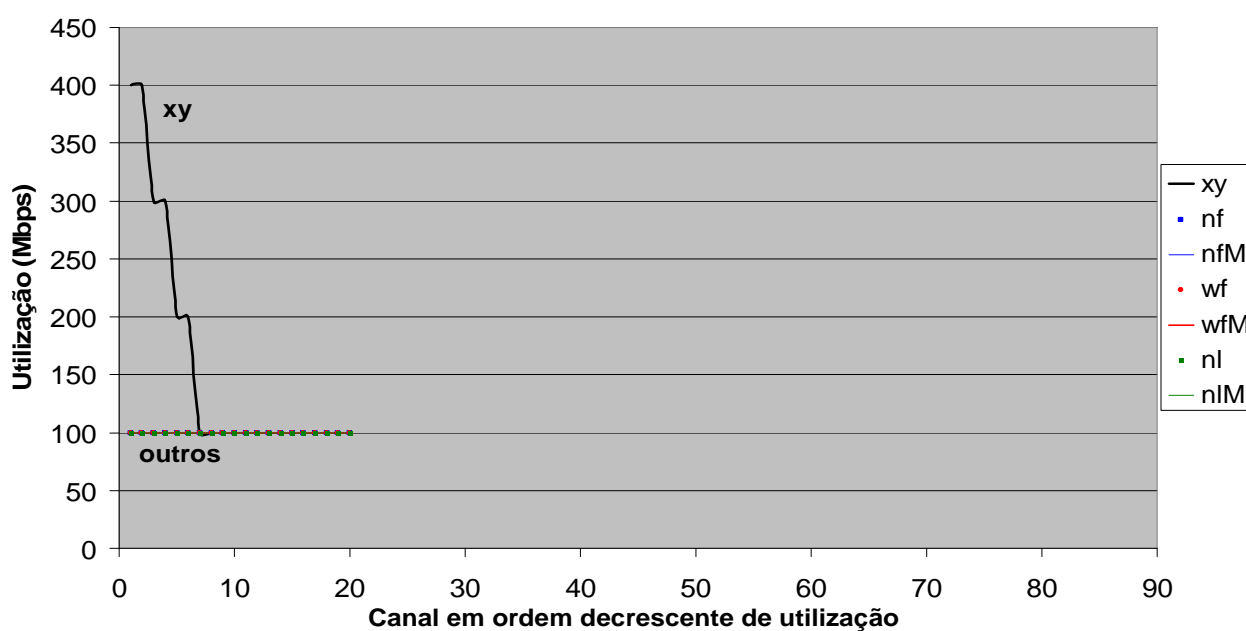


Figura 4.9 - Estimativa de ocupação de canais para o cenário de tráfego sobreposição de rotas.

Na Tabela 4.3 apresentam-se os resultados de latência média obtidos a partir da simulação do cenário de tráfego usando as diferentes definições de cenário de rotas. Similar ao ocorrido no experimento anterior, o algoritmo *xy* garantiu latências médias diretamente proporcionais as distâncias entre os pares comunicantes, tendo aqui crescimento aproximado de 25 ciclos de relógio a cada 2 *hops*. Os demais algoritmos de roteamento garantiram a criação de cenários de rotas com distribuições de carga que permitiram atingir resultados de latência iguais em todos os casos e melhores do que aqueles obtidos com o algoritmo *xy*. Como pode ser visto na tabela, enquanto a diferença entre a maior e a menor latência média para o algoritmo *xy* é de aproximadamente 75 ciclos de relógio, para os demais algoritmos esta diferença cai para 12.

Tabela 4.3 - Comparação de resultados de latência média para o cenário da Figura 4.7(c).

Hops	Latência média						
	xy	nfnm	nfm	nlm	nlnm	wfnm	wfm
2	26,03	25,04	25,04	25,04	25,04	25,04	25,04
4	50,05	29,06	29,06	29,06	29,06	29,06	29,06
6	75,06	33,08	33,08	33,08	33,08	33,08	33,08
8	101,06	37,10	37,10	37,10	37,10	37,10	37,10
Média	63,05	31,07	31,07	31,07	31,07	31,07	31,07

O aumento da latência média, detalhado na Tabela 4.3, permite melhor compreender influência que causado pela sobreposição de fluxo de transmissão de pacotes. Nas rotas definidas pelo algoritmo xy, nota-se que para cada fluxo, sua taxa de transmissão é resultado da distância entre os pares e da concorrência com outros fluxos. A Figura 4.10 ilustra os intervalos de latência obtidos para todos os pacotes transmitidos durante este experimento. Enquanto os pacotes transmitidos com cenário de rotas definidos com o algoritmo xy alcança latências de até aproximadamente 100 ciclos de relógio, para os demais algoritmos de roteamento usados, estes pacotes não ultrapassaram 50 ciclos em nenhum dos casos.

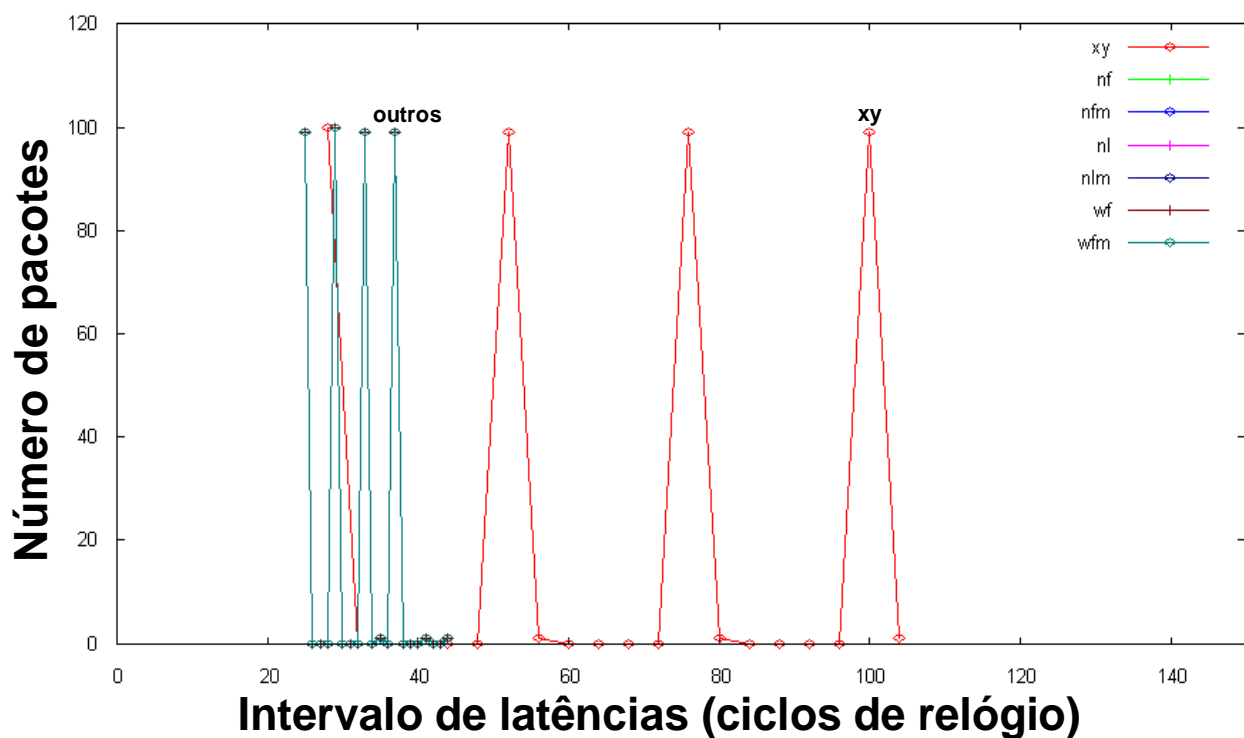


Figura 4.10 – Intervalo de latências obtidos para o cenário de tráfego sobreposição de rotas.

Na terceira e última categoria de cenários de tráfego explora-se a situação onde os pares comunicantes têm como destino possível um de dois pontos da rede, descrevendo um tráfego *hotspot*. A posição da origem de injeção de dados é definida de forma a explorar as vantagens e desvantagens do uso de algoritmos adaptativos na busca de rotas ótimas. Os cenários de tráfego

aqui propostos são ilustrados na Figura 4.11. Todos estes cenários de tráfego foram simulados e avaliados, sendo aqui apresentado aquele definido pela Figura 4.11(d). A taxa de injeção de pacotes para cada uma das origens nestes cenários é 100 Mbps.

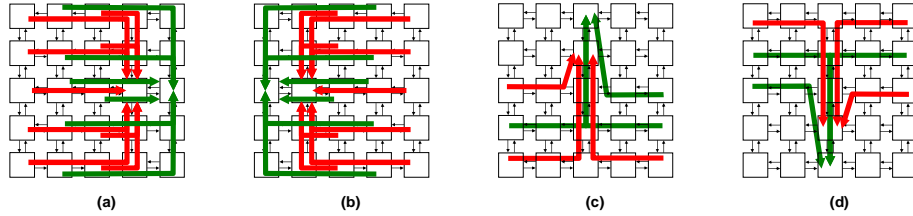


Figura 4.11 – Cenários de tráfego com apenas dois destinos possíveis (*hotspot*).

Extraíu-se um grafo de comunicação que descreve o cenário de tráfego aqui explorado (i.e. Figura 4.11(d)) para a busca dos cenários de rotas. A partir dos cenários de rotas, dos grafos de descrição do cenário de tráfego e da descrição da infraestrutura de comunicação, pôde-se estimar a carga esperada nos canais que compõem a NoC. A Figura 4.12 ilustra as estimativas de distribuição das cargas de comunicação. Nota-se que o algoritmo *xy* não favorece a eliminação de sobrecarga em certos canais no meio da rede, o que é corrigido pelos demais algoritmos.

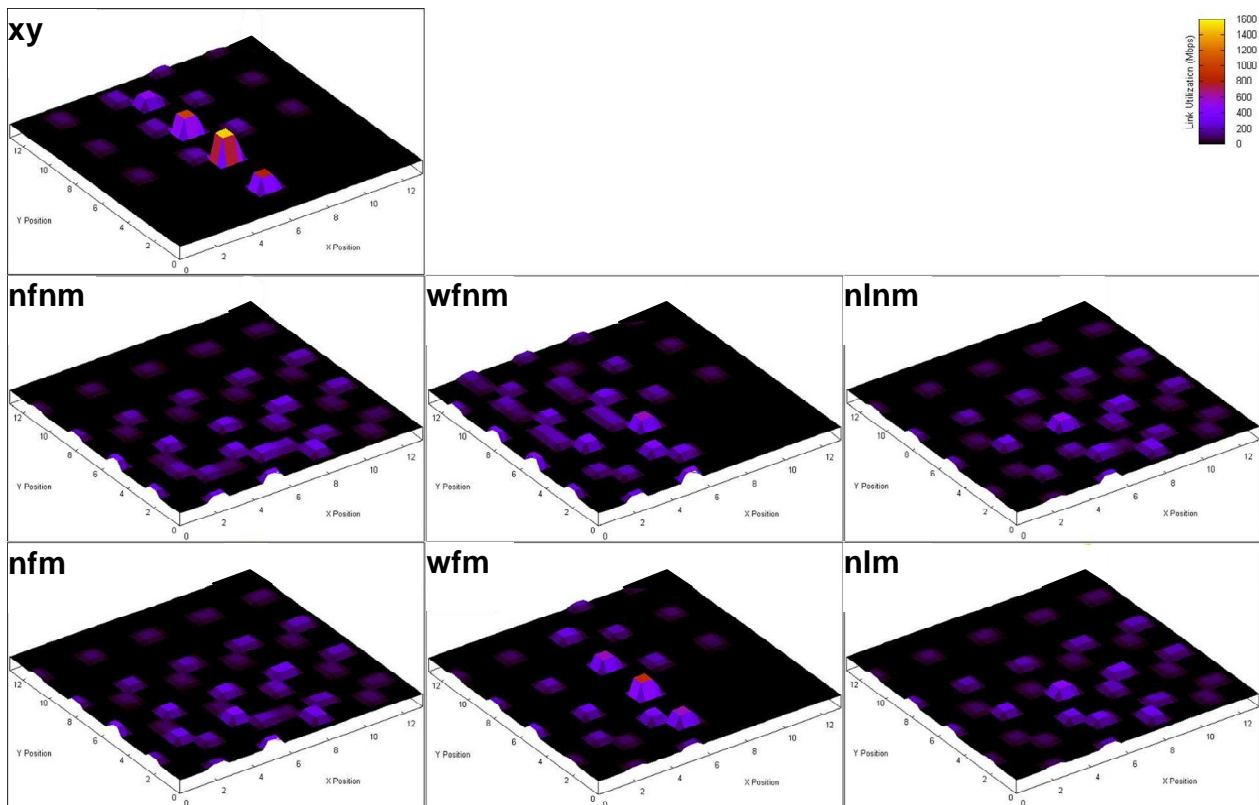


Figura 4.12 - Estimativa de distribuição de carga para o cenário de tráfego *hotspot*.

O gráfico da Figura 4.13 detalha as estimativas de ocupação nos canais da NoC. Novamente,

cabe ressaltar que o gráfico a seguir apresenta a estimativa de ocupação dos canais da rede, ordenando a apresentação destes de forma decrescente dos valores estimados de taxas de ocupação. Expõe-se na Figura 4.13 que o algoritmo *xy* não distribuiu bem os fluxos ocasionando a superação da capacidade máxima de canais, sendo que detectou-se este mesmo problema quando da adoção do algoritmo *wfnm* e *wfm* para a geração dos cenários de rota para este cenário de tráfego. Lembrando que a capacidade máxima de um canal, para o presente experimento é de 800 Mbps, para o algoritmo *xy* são esperadas taxas de transmissão de 1500 Mbps, enquanto que para o algoritmo *wfnm* e *wfm* são esperadas taxas de transmissão de 900 Mbps. Os demais algoritmos garantiram que os canais a serem utilizados na transmissão tivessem a taxa máxima de transmissão alcançando 500 Mbps. O motivo que levou a tais diferenças está ligado à natureza dos algoritmos de roteamento utilizados e a distribuição dos pares origem/destino para este cenário de tráfego. O número de canais a serem utilizado de acordo com os cenários de rotas definidas ficou abaixo da metade do número de canais de rede disponíveis.

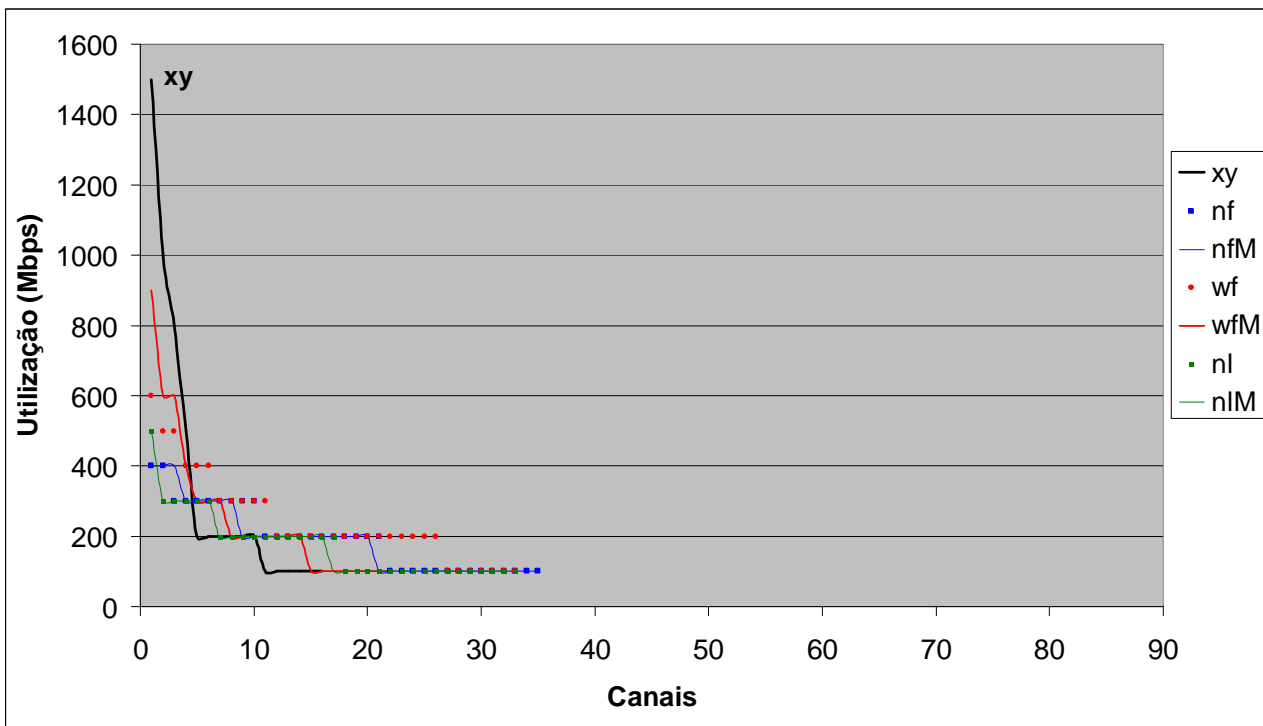


Figura 4.13 – Estimativa de ocupação de canais para o cenário de tráfego *hotspot*.

Com base nas estimativas de distribuição das cargas de comunicação e da ocupação dos canais de rede nos diferentes cenários de rotas, realiza-se a simulação e captura de seus resultados, que a Tabela 4.4 apresenta. A primeira coluna da tabela classifica os pares comunicantes quanto a sua distância em *hops* medida pelo QE. As demais colunas apresentam a latência média capturada

para os cenários de rotas definidos com os diferentes algoritmos de roteamento. Equivalentemente aos demais experimentos conduzidos, pode-se notar novamente o ganho quando da utilização do algoritmo *xy* na definição do cenário de rotas para pares pouco distantes (2 *hops*), perdendo apenas para os cenários de rota definidos pelos algoritmos *wfm* e *nfm*. A partir daí, para distâncias de 3, 4 e 5 *hops*, os demais algoritmos apresentam latência média significativamente melhor do que as do algoritmo *xy*, ficando entre 2,5 vezes (*nfnm* a 3 *hops*) e 74,7 vezes melhor (*nfnm* a 4 *hops*). Na média, os ganhos dos algoritmos adaptativos em relação ao *xy* ficaram entre 3,6 vezes (*wfm*) e 16.3 vezes (*nfm*).

Tabela 4.4 - Comparação de resultados de latência média para o cenário da Figura 4.11(d).

Hops	Latências médias						
	xy	nfnm	nfm	nlm	nlnm	wfnm	wfm
2	57,00	99,50	56,50	72,00	83,50	61,00	49,00
3	4449,25	1720,75	230,00	896,00	1399,75	108,50	1388,50
4	11281,40	151,00	1258,40	512,80	538,60	1373,00	3015,40
5	14219,25	222,50	293,50	1252,75	632,25	2251,00	3749,00
Média	7501,73	548,44	459,60	683,39	663,53	948,38	2050,48

O gráfico apresentado na Figura 4.14 apresenta as latências de todos os pacotes enviados durante a simulação executada para cada um dos cenários de rotas. A latência obtida para cada pacote em cada cenário de rota é dividida em 20 intervalos. Pode-se notar pela figura que as latências mais altas de pacotes são aquelas onde o algoritmo *xy* foi utilizado na geração do cenário de rotas, ultrapassando 20.000 ciclos de relógio. Com os demais algoritmos, a segunda latência mais alta é obtida com cenário de rotas obtido com o algoritmo *wfm* (aproximadamente 13.000 ciclos de relógio), seguida pela latência obtida com o cenário de rotas obtido com o algoritmo *wfnm* (próximo a 7.000 ciclos de relógio). Os demais algoritmos têm suas latências máximas próximas de 5.000 ciclos de relógio. Nota-se que em todos os casos onde a distribuição de cargas de comunicação foi feita com algoritmos adaptativos, a diminuição das latências máximas foi de no mínimo em torno de 35%, chegando a aproximadamente 75%, o que reforçando sua vantagem sobre o algoritmo *xy*.



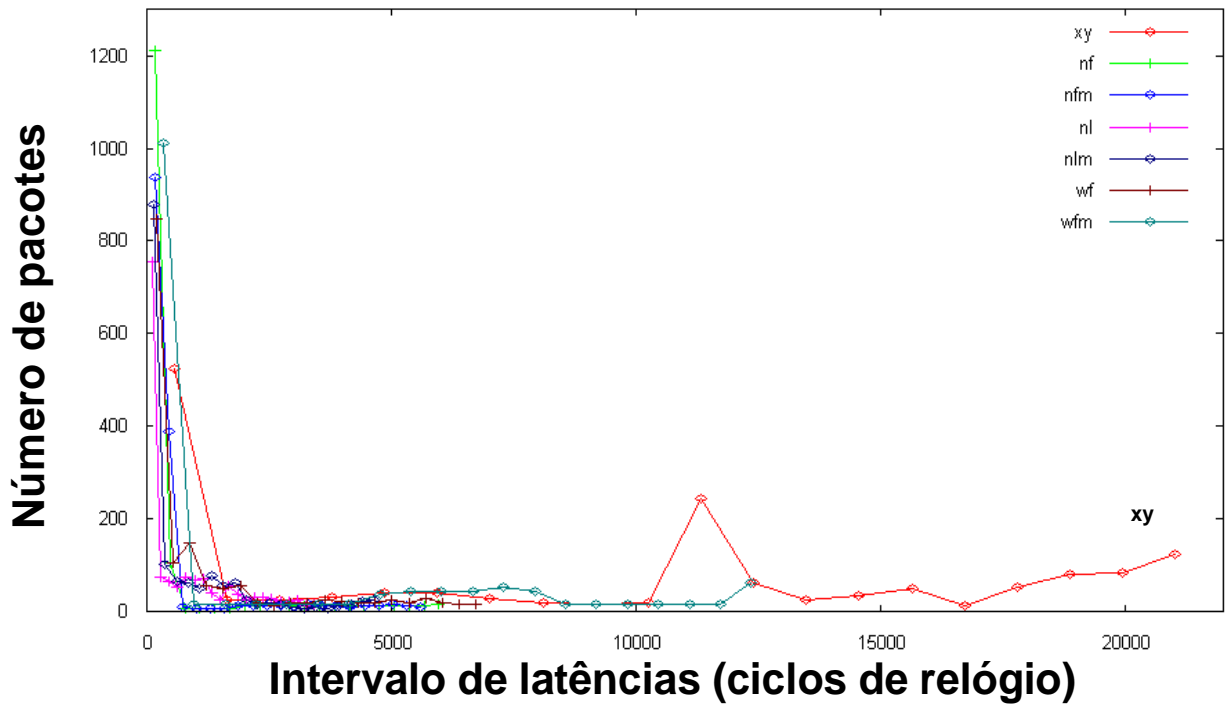


Figura 4.14 - Distribuição das latências na simulação do cenário um utilizando rotas calculadas com os diferentes algoritmos de roteamento determinístico e adaptativos.

### 4.3 CENÁRIO DE ROTAS VERSUS ADAPTAÇÃO EM TEMPO DE EXECUÇÃO

Como última avaliação desenvolvida para esta proposta decidiu-se pela comparação das propostas de decisão de roteamento em tempo de execução e em tempo de projeto. Conduziu-se dois experimentos, ambos comparando resultados de latência média, que serão melhor descritos a seguir.

Neste contexto foi desenvolvida uma nova NoC, aqui chamada AVAL, onde implementou-se quase todas especificações da NoC Hermes, exceto para o esquema de roteamento, modificando para aceitar roteamento na origem, conforme a especificação assumida para a NoC Hermes-SR (Seção 3.1). A NoC AVAL atende as configurações definidas no início deste Capítulo na Seção 4.1.1 para a simulação. Assim, a única diferença entre a NoC AVAL e a NoC Hermes é o esquema de roteamento. Já entre a NoC AVAL e a NoC Hermes-SR a única diferença é o esquema de arbitragem.

O primeiro experimento assumiu um cenário de tráfego todos para todos onde as origens de comunicação injetam tráfego a 30% da capacidade dos canais da NoC (i.e. 240 Mbps). É comparado o desempenho do emprego dos algoritmos de roteamento apresentados no contexto deste

trabalho quando do seu uso em tempo de execução (TExec) e tempo de projeto (TProj), este último representado pelos cenários de rotas. TExec é representado pela NoC Hermes, enquanto TProj é representado pela NoC AVAL. Tamanhos diferentes de fila de armazenamento foram utilizados durante a simulação deste experimento para adicionalmente permitir a avaliação dos potenciais ganhos na redução de latência para ambas as abordagens. Detalhado na Figura 4.15, todas decisões em TProj com os algoritmos *nfnm* e *nfm* levaram a latências maiores quando comparadas as decisões em TExec com os mesmos algoritmos. A exceção foi para em um caso que resultou em ganho para a decisão em TProj (1,12% - aproximadamente 350 ciclos de relógio mais rápido em média) sobre TExec quando assumido o algoritmo *nfm* e fila de armazenamento de tamanho igual a 4. Para o restante dos algoritmos de roteamento, as decisões em TProj garantiram redução na latência média em todos os casos quando comparada as decisões TExec. Os bons resultados obtido com os algoritmos *West first* e *North last* para decisões em TProj tem três explicações. A primeira é o grau de liberdade provido por estes quando comparado ao *negative first* durante o processo de mapeamento de rotas, conforme explicações da Seção 3.2. A segunda explicação é o conhecimento global da carga dos canais de rede quando planejadas as rotas a serem utilizadas pelos pares comunicantes. A terceira é a má decisão que pode ser tomadas em TExec, visto que o julgamento da direção a ser tomada baseia-se em informação de ocupação disponível localmente, o que pode desviar pacotes para regiões de maior concentração de tráfego.

Conforme pode ainda ser visto na Figura 4.15, as decisões em TExec atingiram latências menores quando comparando-se os algoritmos *wfnm* ao *nlnm* e *wfm* ao *nlm*. Todavia, quando os cenários de rotas são definidos, ou seja, em TProj, as vantagens comparadas anteriormente se invertem, sendo atingidas as menores latências comparando-se *nlnm* ao *wfnm* e *nlm* ao *nlnm*.

O segundo experimento combina todas as abordagens exploradas nos este capítulo de resultados, avaliando as decisões arquiteturais (i.e. arbitragem centralizada/distribuída e roteamento em tempo de execução ou em tempo de projeto). Adicionalmente, ele permite medir a eficiência do planejamento de rotas. Para tanto, o cenário de tráfego *hotspot* é novamente utilizado, sendo que as origens dos pares comunicantes injetam pacotes a 10% da capacidade dos canais de rede. A Tabela 4.5 apresenta as latências médias computadas a partir da simulação. Às infraestruturas de comunicação foram empregadas filas de armazenamento de tamanho igual a 4.

Para os valores apresentados na primeira linha da Tabela 4.5, apenas as decisões arquiteturais podem ser avaliadas, visto que as mesmas rotas são usadas durante a transmissão dos pacotes pelas NoCs (i.e.  $xy$ ). Comparando-se a primeira e a segunda colunas da primeira linha, pode ser visto que não houve ganho quando da escolha esquema de roteamento definido em tempo de execução e tempo de projeto. Já a escolha da arbitragem distribuída permitiu ganhos de até 40% na redução de latência quando comparada a abordagem centralizada, comparando-se a terceira coluna com as duas primeiras na primeira linha da tabela. Quando comparando-se a primeira e a segunda colunas para as demais linhas da tabela, pode-se observar a eficiência do planejamento de rotas em tempo de execução e das decisões tomadas em tempo de execução. O experimento mostrou reduções de latência de 24,93% ( $nlnm$ ) até 73,87% ( $nlnm$ ) quando definidos os cenários de rotas. Adicionalmente, pode-se notar o benefício da combinação do planejamento de rotas e da arbitragem distribuída, suportada pela Hermes-SR. Comparando-se os resultados obtidos para este experimento para a NoC Hermes e para a NoC Hermes-SR, obteve-se reduções de até 10,99 vezes a latência média de transmissão de pacotes para esta última infraestrutura de comunicação. Como esperado, os resultados apresentados naquela tabela mostraram que o algoritmo  $xy$  levou ao desempenho menos otimizado quando usado como base de computação do cenário de rotas.

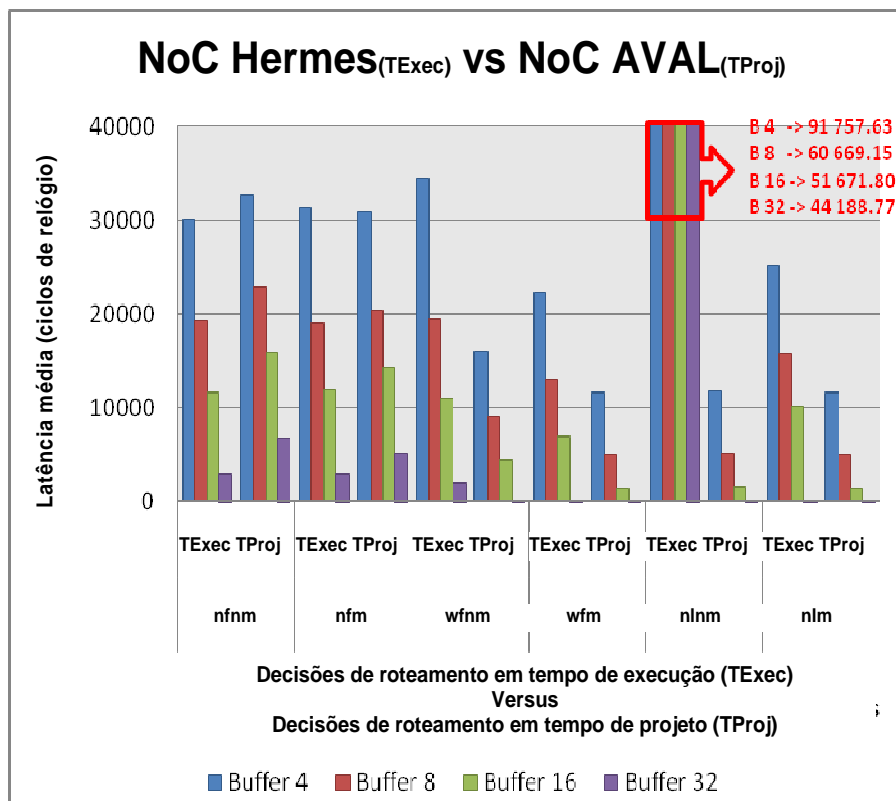


Figura 4.15 – Comparando Latências para decisões em tempo de projeto (TProj) e de execução (TExec).

Tabela 4.5 – Comparação da latência média das NoCs (em ciclos de relógio)

	<b>Hermes</b>	<b>AVAL</b>	<b>Hermes-SR</b>
<b>xy</b>	15 047.06	15 047.06	8 746.96
<b>nfnm</b>	3 364.56	1 894.57	582.29
<b>nfm</b>	4 443.94	1 697.26	567.06
<b>wfnm</b>	5 148.44	3 676.78	1 095.43
<b>wfm</b>	11 081.46	5 344.77	2 382.24
<b>nlm</b>	2 730.85	2 050.10	754.18
<b>nlm</b>	8 053.11	2 103.99	733.06

#### 4.4 CONSIDERAÇÃO FINAL

No presente Capítulo foram apresentados resultados de estimativa e simulação de transmissão de pacotes. No primeiro experimento, uma comparação de latências e ocupação de área foi realizada entre a NoC Hermes-SR e a NoC Hermes, sendo que a primeira proveu diminuição das latências sobre a segunda, porém consumiu mais área em FPGA. No segundo experimento, foram encontrados resultados encorajadores sobre a adoção de caminhos alternativos gerados a partir de algoritmos adaptativos, oferecendo em alguns cenários ganhos de desempenho da ordem de 45 a 75 vezes maior que nas rotas definidas pelo algoritmo xy. No terceiro experimento, conduziu-se uma comparação de desempenho entre abordagens de decisão de roteamento em tempo de execução e tempo de projeto, finalizando por uma comparação da combinação de esquemas distintos de roteamento e arbitragem. Para o primeiro, observou-se vantagens na maioria dos casos enquanto que para o segundo, a combinação da arbitragem com o planejamento de rotas mostrou-se muito eficaz no aumento de desempenho da infraestrutura de comunicação.



## 5. ADAPTAÇÃO DE ROTAS

O mapeamento de rotas pode garantir otimização de desempenho quando do uso da infraestrutura de comunicação. Isto é especialmente verdade quando aplicado no projeto de sistemas onde o comportamento da aplicação é conhecido a priori, permitindo o planejamento de distribuição de cargas de comunicação pelos canais que compõem a infraestrutura de comunicação, como proposto no Capítulo 3. Todavia, em sistemas onde a carga dinâmica de aplicações pode acarretar em desconhecimento do estado instantâneo de utilização da rede, o uso de mecanismos de adaptação de rotas é artifício importante para garantir, por exemplo, desempenho. No presente trabalho, desempenho é figura observada a partir da redução de latência de transmissão de pacotes em uma dada infraestrutura comunicação. Com estes mecanismos é possível evitar a travessia de canais sobrecarregados pelo tráfego de pacotes, potenciais pontos deflagradores de congestionamento, através da adoção de algoritmos de roteamento adaptativos, por exemplo. Apesar disto, duas desvantagens surgem do uso de algoritmos de roteamento adaptativo, quais sejam: *(i)* o conhecimento normalmente limitado que se tem do estado de uso da rede e *(ii)* a limitação da capacidade de adaptação sob circunstâncias específicas. No primeiro caso, uma boa escolha local (i.e. a seleção de uma porta de saída em um roteador que permite o desvio de um canal considerado sobrecarregado) pode levar a uma má decisão global, na qual pacotes são direcionados para regiões da rede onde canais estejam ainda mais carregados. No segundo caso, conforme visto na Seção 3.2 para os algoritmos de roteamento adaptativos avaliados, o posicionamento relativo do par origem/destino de comunicação pode levar à existência de apenas uma opção de rota, inviabilizando o desvio de potenciais canais sobrecarregados.

No presente Capítulo apresenta-se uma abordagem de adaptação de rotas em situações onde o congestionamento é detectado a partir das fronteiras da rede em chip. A adaptação é proposta de forma descentralizada, propondo-se assim uma tomada de decisão tendo por base o conhecimento mais abrangente do estado da rede. Para dar suporte à proposta de adaptação de rotas, dois elementos foram explorados neste trabalho: *(i)* uma infraestrutura de rede em chip empregando sondas para observação de tráfego, doravante chamada MoNoC (**M**onitored **N**etwork-**on-Chip**) e *(ii)* interfaces de redes disponibilizando diferentes serviços de comunicação. A MoNoC

segue as especificações da Hermes-SR e a explora-se uma proposta de adaptação de rotas nas interfaces com a rede, aqui também referenciado como fronteira da rede. Em ambos os mecanismos emprega-se módulos adicionais, que são utilizados ora na avaliação do uso de recursos de comunicação, ora para auxiliar na detecção de queda de desempenho na comunicação. Tais módulos são aqui chamados sonda e monitores, sendo diferenciados pela sua funcionalidade e localização. Monitores são núcleos IP conectados nas portas locais dos roteadores que podem iniciar reações de otimização de uso da rede com o objetivo de prover um determinado serviço contratado. Sondas são núcleos IP embarcados nos roteadores, responsáveis pela observação de eventos associados à transmissão de um dado e o repasse de informações coletadas, quando solicitadas por um monitor. As Seções que seguem, detalham a proposta e os mecanismos desenvolvidos de adaptação de rotas.

## **5.1 PROPOSTA DE ADAPTAÇÃO DE ROTA**

O emprego de algoritmos de roteamento adaptativo é recomendável principalmente para solucionar problemas relacionados a tolerância à falhas. Eles possibilitam ainda o aumento de desempenho de comunicação através da exploração de rotas alternativas entre pares comunicantes no uso da rede em chip. Com dito anteriormente, estas adaptações que levam em consideração o estado local de uso da rede (i.e. ocupação de buffers, uso dos canais de entrada) pode garantir o desvio de um canal considerado sobrecarregado. Todavia, esta ação pode encaminhar pacotes para regiões da rede com maior concentração de tráfego, levando a maior competição por recursos de comunicação e a consequente queda de desempenho da comunicação. O trabalho desenvolvido sobre mapeamento de rotas permitiu observar dois problemas quando empregados os algoritmos adaptativos avaliados. A primeira foi sobre o local onde a adaptação é decidida. Adaptações baseadas no estado local de ocupação dos canais (i.e. roteador a roteador) e em tempo de execução tiveram resultados ruins quando comparado a rotas fixadas por par comunicante através do processo de mapeamento de rotas. A segunda observação diz respeito a quantidade de opções de rotas. Em certos casos, o número de rotas possíveis entre um par comunicante é limitado a um, sendo este número definido pelo posicionamento relativo entre o par na rede e pela natureza do algoritmo de roteamento. No presente trabalho, a proposta de adaptação apresenta solução que resolva ambos os problemas observados e que são detalhados a seguir.

### 5.1.1 ADAPTAÇÃO A PARTIR DAS FRONTEIRAS DA REDE

A decisão de adaptação de uma rota pode ser tomada de forma distribuída ou centralizada. Mecanismos de adaptação distribuídos são comumente utilizados em rede em chip, onde a indicação da direção a ser adotada por um pacote é feita em cada roteador da rede por onde este passa. Com tais mecanismos, o desempenho da comunicação é definido pelo estado de ocupação dos canais de rede e pelas direções permitidas pelo algoritmo de roteamento. Como cada roteador tem uma visão limitada da rede, suas decisões locais podem não ser as mais eficientes. Soluções como aquela apresentada em [DAN06b], onde pacotes de controle são constantemente enviados para atualização de tabelas de roteamento, mostram uma preocupação com a situação global de uso da rede. O emprego de adaptações de forma centralizada pode garantir a melhor distribuição de cargas na rede. Todavia, com o aumento do número de núcleos IP em um único chip e o consequente aumento das dimensões das infraestruturas de comunicação, o tempo necessário para decidir tal distribuição pode ser inviável. Isto se justifica, pois para tal distribuição de cargas deve-se determinar a(s) rota(s) mais adequada(s) para cada par comunicante na rede, além da constante atualização do estado de utilização desta infraestrutura de comunicação. Com isto, uma abordagem centralizada pode não garantir a escalabilidade de sistemas integrados em chip.

A proposta aqui apresentada busca garantir escalabilidade, fator positivo dos mecanismos distribuídos, e conhecimento abrangente do estado da ocupação rede, fator positivo dos mecanismos centralizados. O julgamento da necessidade de adaptações de rota é feito nas fronteiras da rede, onde estão localizados os pares comunicantes, sempre que tiver sido requisitado o emprego de adaptação e que detectada a queda de um dado desempenho esperado. Esta abordagem garante a distribuição do potencial de adaptação e escalabilidade de infraestruturas de comunicação. Sempre que detectada a necessidade de uma adaptação, propõem-se a avaliação do estado de ocupação de potenciais rotas de comunicação entre cada par. O julgamento é feito tomando por base não mais a informação local, de um canal conectado a um roteador, mas sim do conjunto de canais que formam as rotas de conexão. Para tanto, mais de uma rota tem de ser definida para cada par comunicante e garantindo a inexistência de problemas básicos da comunicação, tal como *deadlock*, *livelock* e *starvation*.

### 5.1.2 GARANTIA DE ALTERNATIVAS DE ROTAS

Como dito anteriormente, observou-se um problema comum nos algoritmos de roteamen-



to adaptativos avaliados no capítulo 3. Para cada algoritmo de roteamento adaptativo, existem situações em que o posicionamento relativo entre um par origem/destino em uma rede malha 2D pode permitir somente uma rota para a comunicação. Adicionalmente, resultados mostram que dependendo das características de comunicação de uma aplicação mapeada sobre um sistema empregando uma rede em chip, os diferentes algoritmos de roteamento provêm diferentes resultados de desempenho. Desta forma, a exploração do algoritmo de roteamento a ser utilizado no mapeamento de rotas contribui para o aumento do desempenho da comunicação na rede em chip.

A proposta aqui apresentada busca garantir um desempenho de comunicação ótimo em cenários onde dinamicamente possam ser carregadas aplicações em uma plataforma cuja infraestrutura de comunicação emprega uma rede em chip. Neste tipo de cenário de carga dinâmica, o posicionamento dos pares comunicantes pode variar, o que dificulta definir um único algoritmo de roteamento como o mais apropriado. Uma solução vem da integração de mais de um algoritmo de roteamento em uma mesma infraestrutura de comunicação, definido como conjunto de algoritmos de roteamento. A escolha de algoritmos de roteamento que compõem um conjunto deve garantir que o funcionamento correto da rede será mantido, evitando a ocorrência de problemas tal como *deadlock*. Para criar um conjunto de algoritmos, ditos coexistentes em um mesmo canal, deve-se respeitar a seguinte regra construtiva:

Escolhe-se um algoritmo de roteamento inicial livre de *deadlock*;

Escolhe-se um novo algoritmo de roteamento a ser integrado de forma que este não insira curvas as quais são evitadas no conjunto de algoritmos de roteamento.

O problema da construção de um conjunto de algoritmos coexistentes em um mesmo canal é que ao invés de aumentar a quantidade de rotas possíveis, este tende a reduzi-las, uma vez que o número de curvas possíveis, no melhor caso, mantém o mesmo número de rotas a cada inserção de um novo algoritmo. A forma de uso de mais de um algoritmo de roteamento que venha a aumentar as alternativas de roteamento exige que sejam eliminadas as curvas proibidas pela regras que garantem a ausência de *deadlock* e, conseqüentemente, o emprego de múltiplos canais, virtuais [DUA97] ou físicos [CAR08], sendo cada um associado dominado por apenas um dos algoritmos.

Diferentes algoritmos de roteamento podem ser escolhidos para cada canal. Uma solução que resulta em melhor aproveitamento dos recursos vem do emprego da complementaridade de algoritmos de roteamento. Define-se aqui complementaridade como a situação onde existência de apenas uma rota entre um par origem destino é suprida pelo algoritmo de roteamento presente no outro canal para todos os casos. Três novos algoritmos de roteamento são implementados no presente trabalho, chamados *East first*, *South last* e *positive first*, e que respectivamente suprem as limitações dos algoritmos *West first*, *North last* e *negative first*, estes três últimos apresentados no Capítulo 3.

Apresenta-se aqui, a título de exemplo, o emprego do algoritmo *West first* e seu algoritmo complementar, *East first*. No entanto, a proposta não se limita a tal algoritmo, tendo sido sua escolha feita para limitar o escopo e facilitar as explicações sobre complementaridade. Toma-se por base a análise do algoritmo a ser empregado, realizada na Seção 3.2.2, e a Figura 3.12, onde define-se a relação de posição entre pares origem/destino. Assume-se o emprego de *West first* quando o destino encontra-se nos quadrantes  $QdX_{Pos}Y_{Pos}$  ou  $QdX_{Pos}Y_{Neg}$  em relação a origem. Assume-se o emprego de *East first* nos caso em que essa relação se de nos quadrantes  $QdX_{Neg}Y_{Pos}$  ou  $QdX_{Neg}Y_{Neg}$ . A Figura 5.1 ilustra a relação.

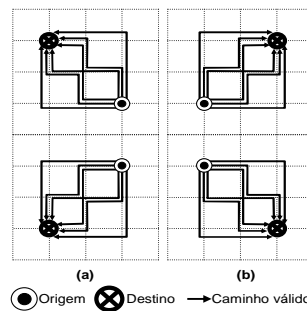


Figura 5.1 – Rotas resultantes do emprego dos algoritmos complementares. Em (a) emprega-se *East first*. Em (b) emprega-se *West first*.

### 5.1.3 CONSIDERAÇÕES FINAIS

Tomando-se por base a NoC Hermes-SR, apresentada no Capítulo 3, a implementação da proposta de adaptação de rotas exige que um conjunto de mecanismos sejam adicionados àquela infraestrutura de comunicação. À proposta de adaptação previu-se o desenvolvimento de interfaces e sondas de rede enquanto que para o emprego de complementaridade previu-se o uso de canais virtuais.

Interfaces de rede são os agentes responsáveis pela adaptação das rotas, enquanto as sondas de rede farão a observação da utilização dos canais da rede. A integração destes núcleos IP em uma rede em chip permite a criação da infraestrutura de comunicação MoNoC, ilustrada na Figura 5.2 e detalhada nas Seções a seguir.

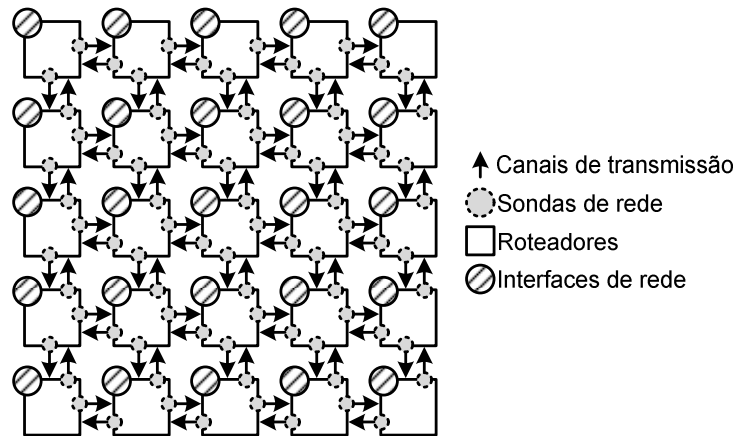


Figura 5.2 – Infraestrutura de comunicação MoNoC resultante da integração dos mecanismos de suporte a adaptação de rotas. Interfaces de rede são posicionadas nas fronteiras para interconexão com os PEs, enquanto sondas de rede são posicionados internamente à NoC, nas portas de saída de cada roteador, onde observam a utilização dos canais.

## 5.2 A INFRAESTRUTURA DE COMUNICAÇÃO MONOC

Nesta Seção será detalhada a organização arquitetural da MoNoC. Esta NoC adota especificações muito similares às aquelas empregadas na Hermes-SR, tendo como principais diferenças o emprego de serviços diferenciados para a transmissão de pacotes, além do emprego de sondas de rede. As seguintes características foram mantidas:

- Filas de armazenamento na entrada e de tamanho parametrizável;
- Chaveamento por pacote, com adoção de técnica *wormhole* de transmissão de pacotes;
- Topologia de rede malha 2D;
- Endereçamento XY de roteadores;
- Roteamento na origem;
- Arbitragem descentralizada para seleção de portas de entrada para transmissão;
- Controle de fluxo baseado em crédito.

Esta infraestrutura de comunicação não emprega mecanismos de garantia de serviço, tal como a reserva de recursos da rede, sendo uma rede que explora unicamente mecanismos de melhor esforço.

### 5.2.1 SERVIÇOS DIFERENCIADOS

O emprego de serviços diferenciados não é novidade em redes em chip [HAR04] [BOL04]. Na MoNoC, eles são diferenciados no nível físico, onde prioridades distintas são atribuídas a canais virtuais existentes em cada roteador para utilização do canal físico de transmissão de dados, canais virtuais estes ilustrados na Figura 5.3.

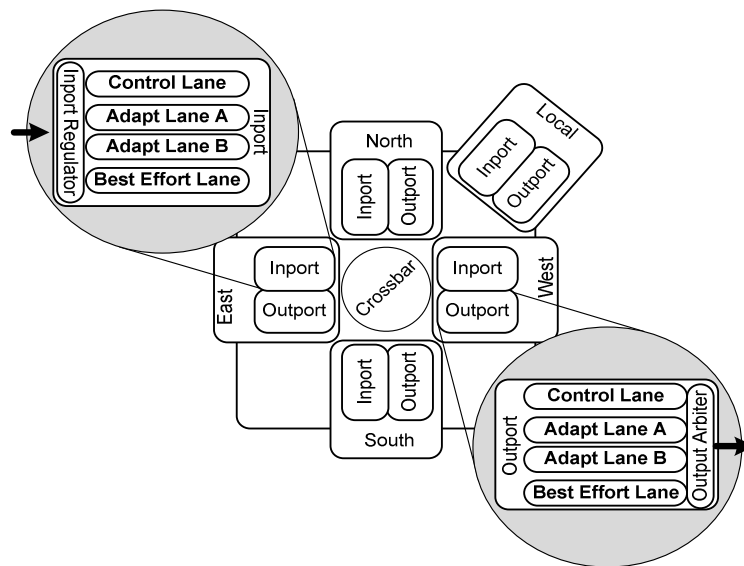


Figura 5.3 - Roteador da MoNoC empregando canais virtuais para a diferenciação de serviços.

O canal virtual de mais baixa prioridade é chamado de *Best Effort Lane* (BE), implementando o atendimento a serviços sem requisitos de latência ou com requisitos flexíveis. Dois canais, chamados *Adapt Lane A* (AdaptA) e *Adapt Lane B* (AdaptB), são utilizados para serviços com requisito alto de desempenho, sendo empregados para a adaptação de rotas no contexto deste trabalho. AdaptA e AdptB têm prioridades equivalentes, sendo superior a prioridade do canal virtual BE. O último canal virtual, chamado *Control Lane* (CTRL), representa o serviço de mais alta prioridade, sendo utilizado para notificações e troca de mensagens de controle. Na MoNoC, a prioridade destes serviços é definida nas portas de saída pelos árbitros dos canais virtuais que analisam a requisição de transmissão. Nas portas de entrada há um módulo direcionador, que encaminha os pacotes para seus devidos canais virtuais. Não há possibilidade de mudança de canal virtual após a entrada de um pacote na rede, o que mantém a prioridade de cada pacote durante seu trânsito pela rede.

A Figura 5.4 ilustra uma competição possível entre os canais virtuais e as regras de prioridade empregadas.

Na Figura 5.4 apresenta-se 4 pacotes (i.e. A, B, C e D), cada um composto por 4 *flits* assim divididos: um *flit* representando a rota (*h*), um *flit* terminador (*t*), um *flit* de tamanho do pacote (*s*) e um *flit* de corpo do pacote (*p*). O conteúdo destes *flits* é aqui ignorado para permitir focar na regra de prioridade dos canais virtuais. A ordem de aparecimento de pacotes para transmissão é identificada pela ordem alfabética associada a cada pacote. Assim, o pacote A tem sua transmissão requisitada em  $t_1$ , sendo BE o único canal virtual a usar o canal físico nos tempos  $t_1$  e  $t_2$ . Em  $t_3$ , os canais virtuais AdaptA e AdaptB requisitam transmissão. Como ambos possuem a mesma prioridade, que é superior àquela de BE, dos tempos  $t_3$  a  $t_6$  a transmissão dos pacotes é intercalada entre AdaptA e AdaptB. No momento  $t_7$  CTRL solicita transmissão e por ter a maior prioridade este ganha o canal físico até o tempo  $t_{10}$ . Quando CTRL conclui a transmissão, os canais AdaptA e AdaptB voltam a intercalar a transmissão até que concluem em  $t_{14}$ . Em  $t_{15}$ , a transmissão de BE é retomada e concluída em  $t_{16}$ .

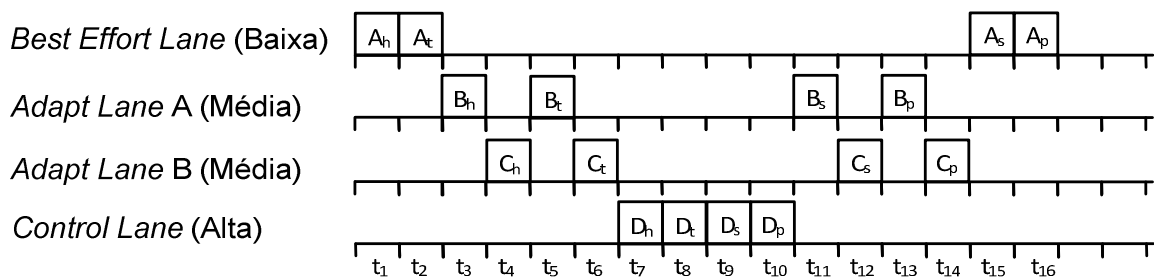


Figura 5.4 – Exemplo de ganho de prioridade para transmissão de pacotes a partir dos canais virtuais da MoNoC. Os pacotes (A, B, C, D) são compostos de 4 *flits*. Os pacotes A, B, C e D chegam aos distintos canais virtuais nos instantes  $t_1$ ,  $t_2$ ,  $t_2$  e  $t_7$  respectivamente.

Sendo  $\text{Max}_{bw}$  a capacidade máxima de transmissão por canal físico, a regra de prioridades permite definir a disponibilidade dos canais virtuais através das equações 5.1, 5.2 e 5.3. Nas Equações 5.1 e 5.3,  $\text{CTRL}_{bw}$  e  $\text{BE}_{bw}$  representam as larguras de banda disponíveis para os canais virtuais CTRL e BE, respectivamente. Na equação 5.2,  $\text{AdaptLanes}_{bw}$  representa a largura de banda disponível para compartilhamento entre os canais virtuais AdaptA e AdaptB.

$$\text{CTRL}_{bw} = \text{Max}_{bw}$$

Equação 5.1

Equação 5.2 
$$\text{AdaptLanes}_{bw} = (\text{Max}_{bw} - \text{CTRL}_{bwInUse}), \text{ onde:}$$

$\text{CTRL}_{bwInUse}$  é a largura de banda em uso pelo canal virtual CTRL.

Equação 5.3 
$$\text{BE}_{bw} = \text{Max}_{bw} - (\text{CTRL}_{bwInUse} + \text{AdaptLanes}_{bwInUse}), \text{ onde:}$$

$\text{CTRL}_{bwInUse}$  é a largura de banda em uso pelo canal virtual CTRL e  $\text{AdaptLanes}_{bwInUse}$  é a soma das larguras de banda em uso pelos canais virtuais AdaptA e AdaptB.

Como dito anteriormente, a prioridade é implementada por um árbitro de canais virtuais presente nas portas de saída de cada roteador. Note que na Figura 5.4 transmissões realizadas por canais virtuais de menor prioridade são bloqueadas até que aqueles de maior prioridade transmitam os *flits* de seus pacotes. Este bloqueio está sujeito a disponibilidade da transmissão dos canais virtuais de maior prioridade, ou seja, é observada a existência de crédito (i.e. controle de fluxo).

A Figura 5.5 explora potenciais situações de bloqueio e as decisões tomadas pelo árbitro de canais virtuais. Nela, a regra de prioridades é respeitada até  $t_8$  e a partir de  $t_9$  pode-se observar o bloqueio por falta de crédito dos canais virtuais CTRL, AdaptA e AdaptB, enquanto BE mantém-se com crédito. Neste caso BE ganha prioridade para a transmissão. Em  $t_{10}$ , AdaptA e AdaptB voltam a ter crédito e, por consequência, prioridade, sendo a transmissão de BE bloqueada. Como a última transmissão dos canais virtuais de mesma prioridade foi feita por AdaptB, em  $t_{10}$ , AdaptA ganha preferência. Dado que a prioridade é intercalada para estes canais virtuais, em  $t_{11}$  AdaptB volta a ganhar acesso ao canal físico. No momento em  $t_{12}$  o canal AdaptA deveria voltar a ganhar prioridade, porém este volta a ficar bloqueado por falta de crédito, e neste caso AdaptB mantém a preferência de uso do canal físico, concluindo a transmissão do pacote C. A partir de  $t_{13}$ , os bloqueios se desfazem e a regra de prioridades volta a ser seguida como ocorrera entre os tempos  $t_1$  e  $t_8$ .

Com relação à adaptação de rotas, o emprego de serviços diferenciados foi adotado para que notificações e mensagens de controle sofram o mínimo possível com concorrência por recur-

sos na rede. Para este tipo de mensagem, deve-se empregar o canal virtual CTRL. Os dois canais virtuais, AdaptA e AdaptB, têm por objetivo dar suporte à mencionada complementaridade dos algoritmos de roteamento. Um dos algoritmos é assumido em cada canal, evitando assim a quebra das regras que garantem a inexistência de *deadlock*.

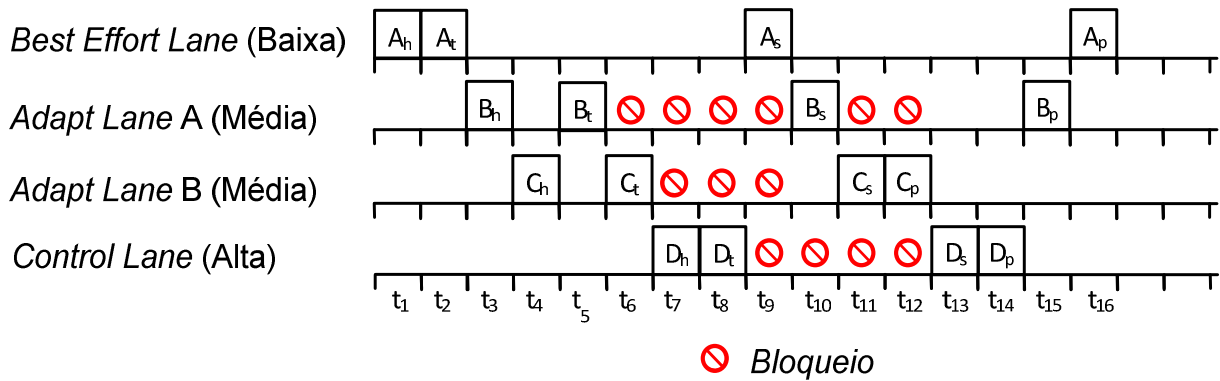


Figura 5.5 – Comportamento projetado para o ganho de prioridade sob situações de bloqueio na transmissão de pacotes a partir dos canais virtuais da MoNoC. Os pacotes (A, B, C, D) são compostos de 4 *flits*. Os pacotes A, B, C e D chegam aos distintos canais virtuais nos instantes  $t_1$ ,  $t_2$ ,  $t_2$  e  $t_7$  respectivamente.

Independente do canal virtual a ser utilizado na transmissão de pacotes, a estrutura dos pacotes respeita àquela definida na Seção 3.1.1. O cabeçalho do pacote contém a rota a ser utilizada, finalizada por um *flit* terminador, e o tamanho do corpo do pacote. O restante dos *flits* que compõem o pacote define o corpo deste, espaço este reservado à carga útil. Todavia, a MoNoC foi desenvolvida para dar suporte à adaptação de rotas. Tal adaptação leva em consideração o estado de ocupação de um conjunto de canais na rede, que é capturado em tempo de execução. A captura do estado de ocupação de canais é realizada a partir de pacotes especiais que trafegam pelo canal virtual CTRL e operam com sondas espalhadas pela rede. Assim, cuidado especial deve ser tomado para pacotes que trafegam por este canal virtual, dadas algumas peculiaridades de operação com sondas de rede, assumidos a tal canal. Isto, porém, não inviabiliza o uso deste canal para a transmissão de outros tipos de pacotes.

## 5.2.2 SONDA DE REDE

As sondas de rede têm papel fundamental na proposta de adaptação de rotas. Eles são núcleos IP passivos, visto que não provocam reação ou notificação. Sua função é a de observação do estado de utilização de canais da rede e a resposta a requisições de leitura e escrita feitas direta-

mente a eles a partir de pacotes específicos que trafegam pelo canal virtual CTRL. Os resultados capturados a partir das sondas de rede dão suporte à adaptação de rotas, detalhado nas seções a seguir. As sondas de rede localizam-se nas portas de saída de cada roteador, conforme ilustra a Figura 5.6. O número de sondas de rede por roteador é equivalente ao número de portas de saída do mesmo, sendo então cada sonda de rede dedicada à observação de apenas um canal. Cabe ressaltar que a presença da sonda de rede não exige a inclusão de sinais adicionais à interface entre os roteadores, o que garante equivalência àquela interface empregada nas especificações da NoC Hermes com canais virtuais.

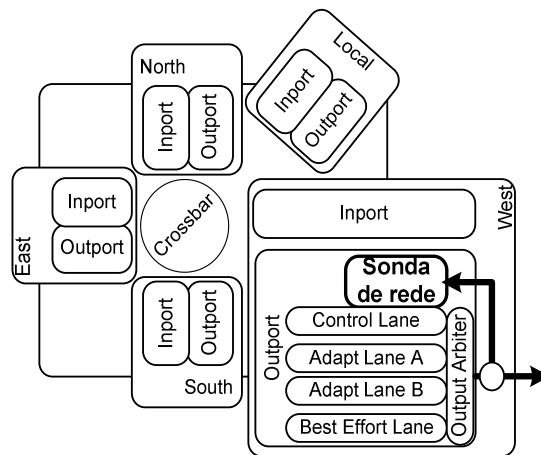


Figura 5.6 – Localização de uma sonda de rede em um roteador. Em destaque uma sonda de rede da porta *West* de um roteador central.

Definiram-se dois conjuntos de sinais, visto que uma sonda de rede é um núcleo IP de observação de transmissão e de resposta a requisições de leitura e escrita. A interface deste núcleo IP é delineada na Figura 5.7.

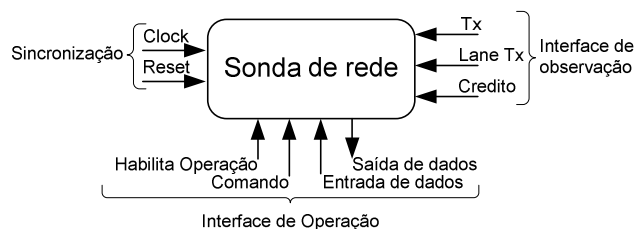


Figura 5.7 Interface da sonda de rede.

A sonda de rede possui uma interface de observação e uma de operação, sendo esta última a interface por onde são atendidas requisições de leitura e escrita. Já a interface de observação é



utilizada para capturar os eventos associados à transmissão de dados nos canais de rede. A interpretação dos eventos de transmissão é descrita a seguir, baseando-se na Figura 5.8.

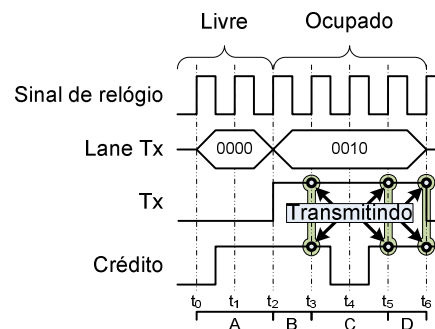


Figura 5.8 - Sinais da interface de observação analisados pela sonda de rede da MoNoC.

Dois estados podem ser identificados em um canal, quais sejam: (i) livre ou (ii) ocupado. No primeiro estado, não há requisição de transmissão, logo o canal não está sendo utilizado, conforme representado no intervalo A (i.e.  $[t_0; t_2]$ ) da Figura 5.8. No segundo estado, uma transmissão é requisitada, conforme representado na Figura 5.8 nos intervalos B, C e D (i.e.  $[t_2; t_6]$ ). Durante o estado de ocupado, distingue-se dois subestados, nos quais (i) o canal estava ocupado, mas não transmitindo, ou (ii) o canal estava ocupado transmitindo. Neste último subestado, a transmissão de um dado pelo canal de transmissão realmente ocorre, correspondendo ao final dos intervalos B, C e D (i.e.  $t_3$ ,  $t_5$  e  $t_6$ ).

A observação da ocorrência dos estados de ocupado e do subestado ocupado transmitindo são contabilizadas pelas sondas de rede. O evento que caracteriza a ocupação do canal relaciona uma borda de subida do sinal de relógio e o sinal **tx** com valor lógico '1'. O evento que caracteriza uma transmissão relaciona a borda de subida do sinal de relógio, o sinal **tx** e o sinal **crédito** com valor lógico '1'. Avaliando-se a Figura 5.8, e considerando os eventos especificados, contabiliza-se 4 ocorrências do estado ocupado e 3 ocorrências da ocupado transmitindo algum dado. Os valores contabilizados são acumulados dentro de uma *janela de observação*, definida por um número limitado de ciclos de relógio. A janela de observação é configurada em tempo de projeto, mas também pode ser modificada em tempo de execução a partir da interface de operação das sondas de rede. Os valores contabilizados são armazenados de três diferentes formas por janela de observação, quais sejam: (i) o valor corrente, (ii) o último valor contabilizado e (iii) a média de ocorrência dentro das janelas de observação. O valor corrente é definido pelo número de ocorrências dos estados anteriormente citados dentro da janela de observação em curso. O armazenamento do último va-

lor e a computação da média ocorre sempre que finda uma janela de observação.

A interface de operação permite que sejam alteradas configurações iniciais da sonda de rede (i.e. operação de escrita), tal como o tamanho da janela de observação definido em número de ciclos de relógio. Esta interface também permite a captura dos valores contabilizados de eventos relacionados a ocupação e transmissão por um dado canal (i.e. operação de leitura). O sinal **comando**, pertencente à interface de operação, é composto por 4 bits. O bit mais significativo indica se é requisitada uma operação de leitura ('1') ou escrita ('0'). A implementação atual permite a captura de até 6 tipos distintos de valores contabilizados pelas sondas de rede, resultante da relação entre as contabilizações dos estados de ocupado/transmitindo com o tipo de observação destas informações, ou seja, se o é valor corrente ou se é o último valor avaliado ou se é a média. Quanto à escrita, duas operações são possíveis. A primeira diz respeito à atualização da janela de observação, conforme já citado. A segunda permite alterar os canais virtuais que devem ser considerados quando da contabilização da ocorrência de eventos de ocupação e transmissão. Por padrão, dado que adaptação de rotas ocorre apenas nos canais AdaptA e AdaptB conforme será visto a seguir, em tempo de projeto define-se que apenas os canais virtuais de prioridade média devem ser observados. As requisições de operação com a sonda de rede somente têm efeito se o sinal **habilita operação** estiver com valor lógico '1'. Na MoNoC, as requisições de operação com as sondas de rede são resultado da interpretação de pacotes especiais que trafegam pela rede, mais precisamente pelo canal virtual CTRL.

Todo pacote que trafega pelo canal virtual CTRL e que tem por objetivo realizar operações com as sondas de rede contém no primeiro *flit* do corpo do pacote o bit menos significativo carregando o valor lógico '1'. A estrutura destes pacotes é ilustrada na Figura 5.9. O segundo bit menos significativo é reservado para a sinalização entre os mecanismos de adaptação de rota, sendo seu uso explorado nas próximas Seções. O comando, definido em 4 bits, fica armazenado entre os bits 5 e 2. No caso de uma operação de escrita, o dado encontra-se no próximo *flit* do corpo do pacote. No caso de uma leitura, o segundo *flit* contém um índice/endereço a ser utilizado pelo canal virtual CTRL para o controle no armazenamento do valor proveniente da operação requerida.

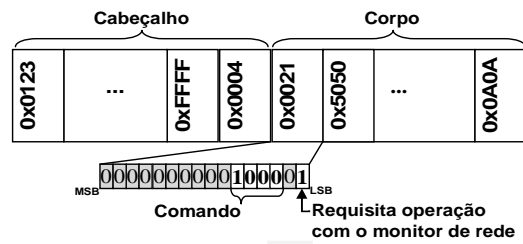


Figura 5.9 - Estrutura de um pacote para operação com as sondas de rede.

A fim de que não sejam alterados parâmetros de sondas de rede, deve-se garantir que o bit menos significativo do primeiro *flit* do corpo de um pacote que trafega pelo canal virtual CTRL, contenha valor lógico '0'. O objetivo do emprego deste tipo de operação (i.e. leitura e escrita) com as sondas de rede é de garantir uma forma genérica e básica de interação com os mesmos, provendo mecanismos para sua programação, por exemplo.

### 5.3 INTERFACE DE REDE

A infraestrutura de comunicação MoNoC provê (i) o suporte necessário para a transmissão de pacotes, adicionalmente disponibilizando (ii) recursos de diferenciação de tráfego de pacotes e (iii) mecanismos de avaliação dos estados dos canais da rede. Apesar disto, não é explorado qualquer mecanismo de adaptação pela MoNoC.

Normalmente, interfaces de rede (em inglês, *Network Interface*, NI) têm como papel específico a padronização do mecanismo de transmissão e o empacotamento/dempacotamento de dados que trafegam por um meio de comunicação. Sua funcionalidade pode ser descrita tanto em *software* quanto em *hardware*. Bhojwani e Mahapatra [BHO03] compararam ambas as implementações e mostraram que a versão em *software* exigiu 47 ciclos de relógio enquanto a implementação em *hardware* levou apenas 2 ciclos. No presente trabalho, acrescenta-se às NIs o poder de adaptação de rotas entre pares comunicantes, tomando-se por base o estado de utilização da infraestrutura de comunicação.

Na presente Seção é apresentada uma interface de rede que tem por objetivo garantir serviços de comunicação aos núcleos IPs conectados a ele. Tais serviços são detalhados nas Sessões a seguir, sendo dado maior destaque ao serviço que oferece suporte a adaptação de rotas.

#### 5.3.1 SERVIÇOS DISPONIBILIZADOS PELA NI

A NI provê ao núcleo IP associado a ela dois tipos de serviço, quais sejam: (i) transporte de

dados com esforço de atendimento de taxa de transmissão (ii) transporte sem tal esforço. O primeiro serviço somente é fornecido pelos canais virtuais AdaptA e AdaptB da MoNoC. Para o segundo serviço, qualquer um dos canais virtuais pode ser utilizado. O transporte de dados com esforço de atendimento de taxa de transmissão foi planejado para ser utilizado como recurso de exploração do suporte à proposta de adaptação de rotas. A Figura 5.10 detalha uma NI e os sinais que compõem suas portas de entrada e saída, por onde são solicitados serviços e encaminhados dados à MoNoC.

A interface entre a NI e o núcleo IP é muito similar àquela entre a NI e a MoNoC. A NI implementa o mesmo controle de fluxo adotado na MoNoC, ou seja, baseado em crédito. O sinal **sv\_tx** permite identificar que nível de prioridade da MoNoC será utilizado para o transporte dos dados, sendo interpretado pela NI e adequado para o sinal **vc\_tx**. O sinal **sv\_rx**, identifica o serviço que foi utilizado para o transporte do pacote que está chegando ao núcleo IP, sendo resultado da interpretação do sinal **vc\_rx**.

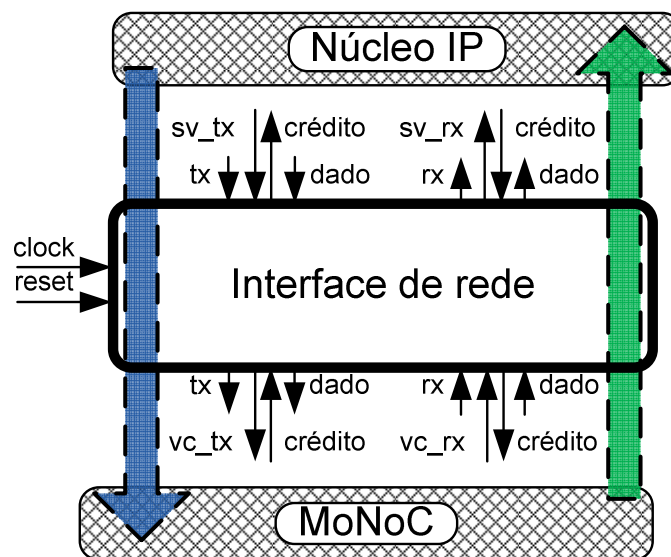


Figura 5.10- Interfaces da NI para transmissão entre o Núcleo IP e a MoNoC.

A contratação do serviço de transporte de dados com esforço de atendimento de taxa de transmissão exige que ambas as NIs de origem e destino da comunicação estejam cientes da solicitação deste serviço. Para tanto, solicitações que levem a tal contratação de serviço devem partir dos núcleos IP para as NIs vinculadas a eles, as quais são brevemente apresentadas aqui e detalhadas nas Seções a seguir. Inicialmente, o núcleo IP onde a comunicação é originada deve solicitar a abertura de uma sessão com o núcleo IP destino. Estabelecida a sessão, o núcleo IP estará apto a

utilizar o serviço de transporte de dados com esforço de atendimento de taxas, o que ocorre sempre que os parâmetros são informados: *(a)* o destino, *(b)* o tamanho da mensagem a ser transmitida e *(c)* a mensagem. Ao concluir o ciclo de transferências de dados, a sessão deve ser fechada. Solicitações de transporte de dados sem a prévia abertura de sessão caracterizam a utilização de serviço de entrega de dados sem observação do cumprimento de taxa de transmissão. Para a abertura de uma sessão é necessária a definição de alguns parâmetros, quais sejam: *(a)* o destino da comunicação, *(b)* o tamanho (em ciclo de relógios) da janela de observação, *(c)* a taxa média de transmissão a ser cumprida, *(d)* o canal virtual a ser utilizado (i.e. AdaptA ou AdaptB) e *(e)* as possíveis rotas a serem utilizadas. Para o fechamento de uma sessão, é necessário apenas informar o endereço do núcleo IP destino da comunicação. As consequências da abertura e fechamento de sessão refletem na adaptação de rotas. Por ora, cabe ressaltar que a abertura e o fechamento de sessão criam um mecanismo de controle de fluxo fim a fim entre os núcleos IP comunicantes.

A MoNoC foi projetada de forma a dar suporte à implementação de diferentes topologias de rede. Apesar disto, no presente trabalho, optou-se pelo uso da topologia malha 2D. Por esta característica, sobre a NI desenvolvida neste trabalho foi incluída uma funcionalidade que permite que o primeiro parâmetro a ser informado quando da requisição de um serviço de transmissão de dados (i.e. o destino) possa ser feito de dois modos: *(i)* com o endereço XY do alvo da comunicação ou *(ii)* como a sequência de direções que define a rota a ser utilizada. Quando informado o endereço XY do destino, a NI computa através do algoritmo de roteamento XY a sequência de direções necessária para alcançar o alvo da comunicação. Esta funcionalidade foi incorporada para permitir compatibilidade com o protocolo utilizado pelas demais NoCs desenvolvidas no grupo de pesquisa do autor com especificação baseada na NoC Hermes e que utilizam canais virtuais, facilitando a interoperabilidade de estruturas distintas.

Em tempo de projeto, três parâmetros devem ser definidos para a implementação de cada NI, quais sejam: *(i)* seu posicionamento XY na MoNoC, *(ii)* as dimensões de rede como um todo e *(iii)* o tamanho máximo da carga útil a ser transportada por pacote. O primeiro parâmetro é incorporado aos pacotes transferidos através da rede a partir o núcleo IP para que o destino reconheça a origem da transmissão, tornando o pacote entregue similar em estrutura àquele das implementações da NoC Hermes. O segundo parâmetro, aliado ao primeiro, provê recurso necessário para a computação de rotas, onde esta funcionalidade é requerida. Finalmente, o terceiro parâmetro é utilizado para a decisão da fragmentação da mensagem em pacotes. Esta última abordagem tem

por objetivo prover meios de compartilhamento dos recursos de forma justa. No caso de pacotes de tamanho muito grande, a técnica de chaveamento de pacote *wormhole* pode vir a ter comportamento similar ao chaveamento por circuito, visto que os recursos da rede são alocados pelo primeiro *flit* do pacote e liberados somente depois que o último *flit* do pacote passa por estes. Como cada roteador possui uma ordem de atendimento das portas de entrada que requisitam transmissão de pacotes, a fragmentação da mensagem pode contribuir para o uso justo dos recursos de comunicação. O processo de fragmentação da mensagem em pacotes, envio e recepção dos mesmos é ilustrado na Figura 5.11.

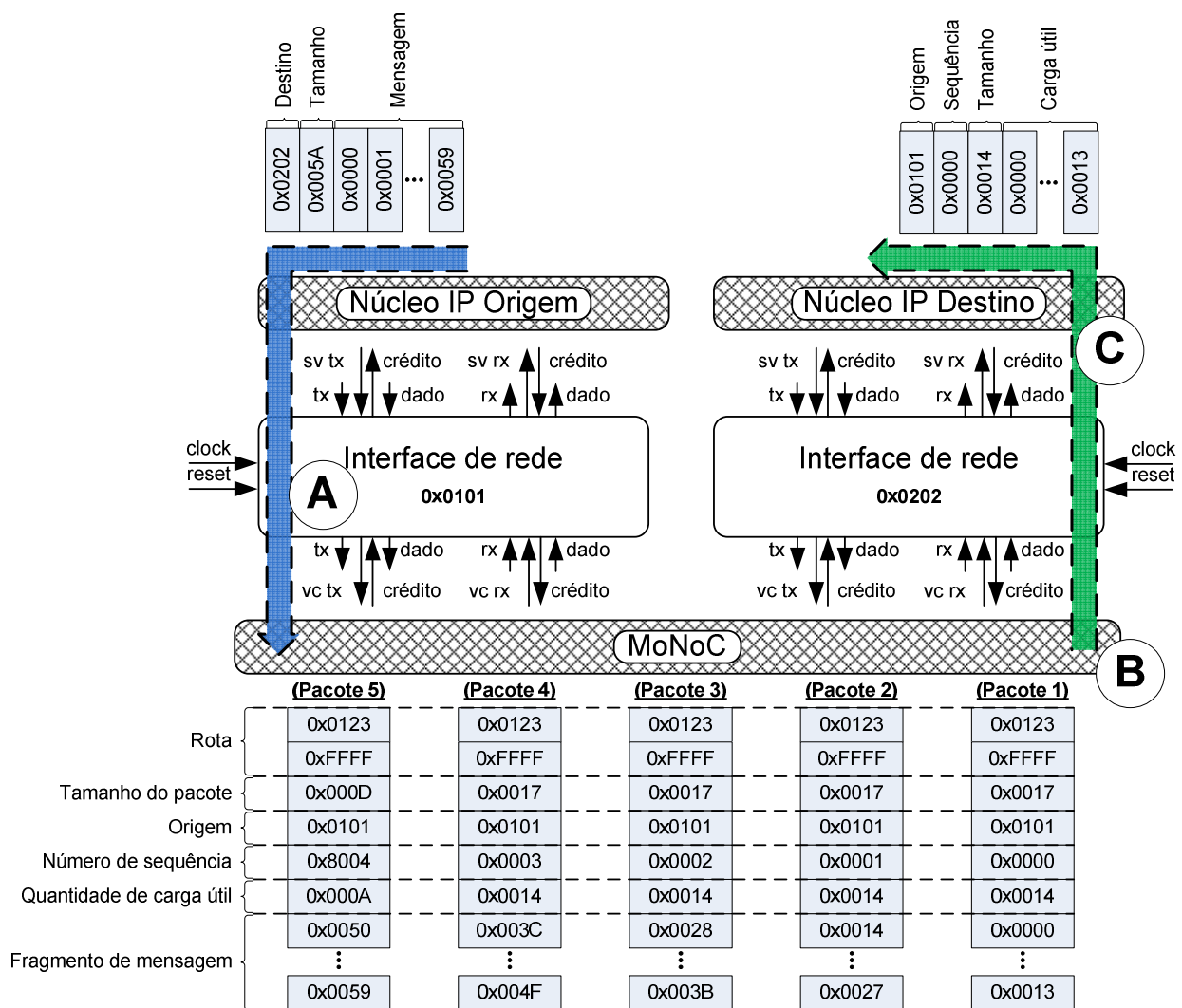


Figura 5.11 – Exemplo de transmissão de uma mensagem de *flits* de carga útil através da MoNoC. Ilustra-se o processo adotado na NI para o envio de mensagens pela rede. Em (A) a mensagem encaminhada é fragmentada pela NI. Em (B) aparecem os pacotes resultantes da fragmentação, que trafegarão na rede. Em (C) mostra-se a entrega dos pacotes da mensagem.

Para o processo ilustrado na Figura 5.11, definiu-se que os pacotes poderiam carregar no máximo 20 *flits* de carga útil. Adicionalmente, à NI do núcleo IP origem da comunicação foi associado o endereço 0x0101 e à NI do núcleo IP destino foi associado o endereço 0x0202. A mensagem a ser enviada contém o endereço do núcleo IP destino, o tamanho e a carga útil da mensagem, conforme ilustrado na Figura 5.11(A). A mensagem é fragmentada ao ser encaminhada à NI. No caso apresentado, o tamanho da mensagem é igual a 90 *flits*, o que resulta na fragmentação desta em 5 pacotes, onde 4 carregam 20 *flits* de carga útil e 1 carrega 10 *flits*. Aos pacotes gerados a partir fragmentação, são adicionadas as informações de endereço de origem de envio do pacote e um número de sequência para controle na recepção. O bit mais significativo do campo que representa o número de sequência é reservado para a indicação de último pacote resultante da fragmentação da mensagem, conforme pode ser notado no campo número de sequência do pacote 5 na Figura 5.11(B). Antes da entrega de pacotes no destino, são eliminados os dois primeiros campos do pacote, referentes a rota utilizada e o tamanho do pacote transferido. O restante do pacote é repassado, conforme ilustra a Figura 5.11(C), ou seja, a origem da comunicação, o número de sequência do pacote, a quantidade de carga útil e a carga útil.

### 5.3.2 SUPORTE À ADAPTAÇÃO DE ROTAS

Cerne deste Capítulo, a adaptação de rotas é possível quando se contrata o serviço de transporte de dados com esforço de atendimento de taxa de transmissão. A sequência de parâmetros informados pelo núcleo IP origem da comunicação, durante a abertura de sessão, é armazenada tanto na NIs de origem quanto na de destino da comunicação. Tais parâmetros servem de configuração de recursos que darão suporte a adaptação de rotas. O diagrama de blocos da Figura 5.12 ilustra os módulos que compõem a NI.

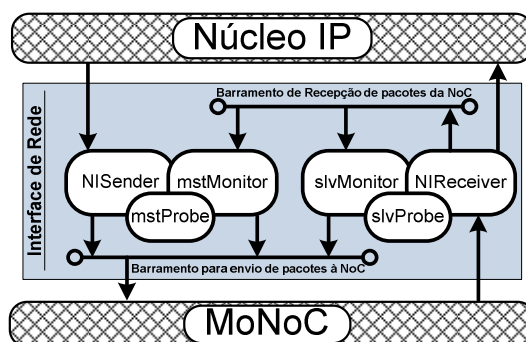


Figura 5.12 - Diagrama de blocos da NI.

As requisições de comunicação provenientes dos núcleos IP são sempre capturadas pelo

módulo NISender. Este é responsável pela interpretação do serviço que está sendo solicitado, bem como pela fragmentação das mensagens e encaminhamento de pacotes à rede. Adicionalmente, a funcionalidade de computação de rota, citada na Seção anterior, é implementada neste módulo. O módulo NIReceiver captura os pacotes que chegam da MoNoC e os encaminha para o destino. Os demais módulos ali presentes são usados no processo de adaptação de rotas, descrito mais adiante nesta Seção.

A sequência de ações que resultam na abertura de uma sessão é apresentada na Figura 5.13. A requisição de abertura de sessão é interpretada pelo NISender, o qual permite ao núcleo IP ter acesso ao módulo mstMonitor. O mstMonitor recebe e armazena os parâmetros informados, encaminha o pedido de abertura de sessão à NI destino e inicia o monitoramento da origem de transmissão de dados. Ao chegar ao destino, o módulo NIReceiver é responsável pela captura e interpretação de um pacote de controle. Os parâmetros contidos no pacote de controle são repassados ao módulo slvMonitor. Este último módulo armazena as configurações necessárias para iniciar o monitoramento do canal de recepção de dados a partir da infraestrutura de comunicação em suporte ao serviço de transmissão de dados com esforço de atendimento de taxa de transmissão.

A adaptação de rota é uma decisão bilateral, sendo tomada em conjunto pelas NIs origem e destino. Neste processo, cada NI tem papel distinto e fundamental. A NI destino da comunicação tem como responsabilidades *(i)* a notificação do descumprimento da taxa de transmissão contratada e *(ii)* a escolha da alternativa de rota mais adequada, no caso da necessidade de uma adaptação. Já a NI origem é responsável por *(i)* julgar as notificações de descumprimento que venha a receber e *(ii)* enviar alternativas de rota para a adaptação, se cabível.

No presente trabalho, define-se dois fatores que podem servir para a observação do descumprimento de uma taxa de transmissão contratada pelo destino da comunicação, quais sejam: *(i)* a queda na taxa de injeção de dados abaixo do que fora contratado a partir da origem de comunicação ou *(ii)* a identificação de congestionamento na rede. Se o primeiro caso for observado, a NI origem da comunicação encaminha uma notificação à NI destino na qual assume a culpa pelo descumprimento da taxa de transmissão esperada. Neste caso, é descaracterizada a ocorrência de congestionamento na rede o que elimina a necessidade de uma adaptação de rota. Já quando o segundo caso é percebido, a NI origem encaminha um conjunto de alternativas de rota para a NI destino. Após receber as alternativas de rota, a NI destino escolhe aquela mais livre para a utiliza-



ção, baseando-se para isso na avaliação da taxa média de ocupação dos canais.

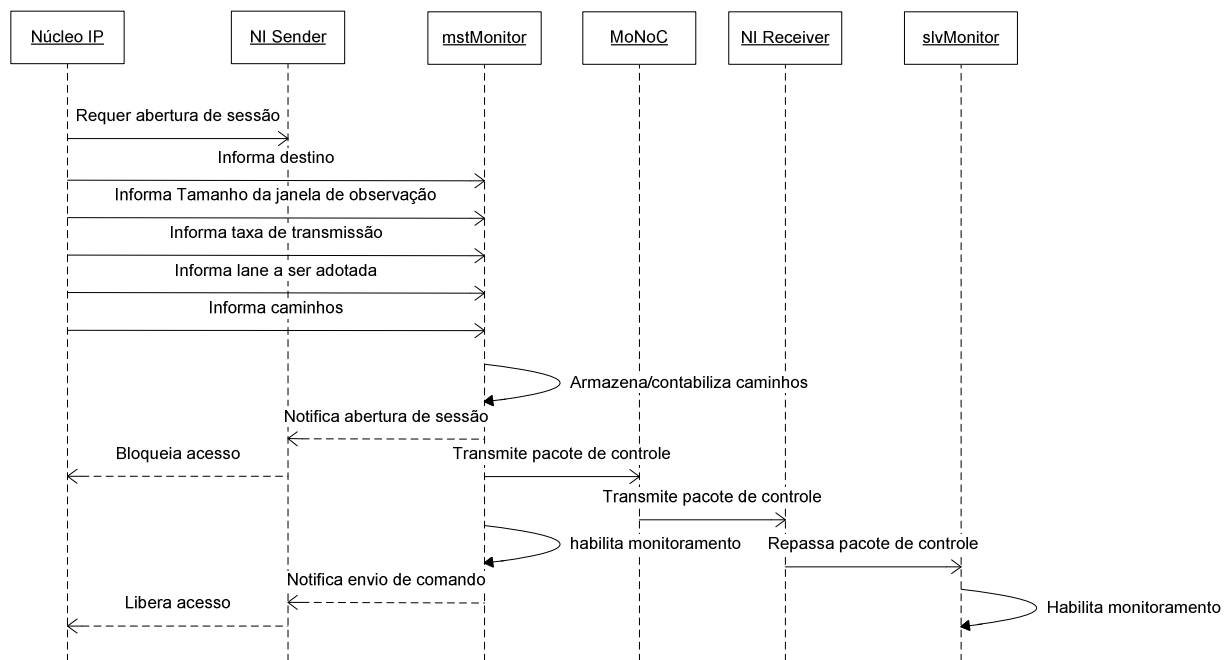


Figura 5.13 - Diagrama de sequência de mensagens para a abertura de uma sessão.

A partir da abertura de uma sessão, a observação das taxas de injeção e recepção de dados na fronteiras da rede são respectivamente de responsabilidade do mstProbe, na NI de origem, e do slvProbe, na NI destino. O tamanho da janela de observação, informada na abertura da sessão, é parâmetro utilizado pelo slvProbe para dois fins. O primeiro é garantir medidas de observação de taxas de transmissão na NI destino na mesma proporção daquelas que estão sendo obtidas na NI origem da comunicação. O segundo fim é gerar um evento, do slvProbe para o slvMonitor através de um sinal dedicado entre estes módulos, sempre que alcançada a quantidade de ciclos de relógio dada pelo tamanho da janela de observação. A geração deste evento faz com que o slvMonitor, que está diretamente conectado ao slvProbe, avalie as diferenças entre a taxa de transmissão contratada e a realizada, e decida pela necessidade de notificação de descumprimento da taxa de transmissão contratada.

Assim como as sondas de rede, os módulos mstProbe e slvProbe observam os estados de ocupação e transmissão, limitando-se porém às fronteiras dos canais locais do roteador aos quais a NI está conectada, conforme ilustra a Figura 5.14. O mstProbe avalia o canal de injeção enquanto o slvProbe, o de recepção. Na NI destino, o valor observado pelo slvProbe é o de transmissão efetiva

de dados, ou seja, quantas vezes um dado disponível na rede foi encaminhado para o núcleo IP. A contabilização destes valores segue os critérios de base definidos para as sondas de rede, ou seja, sempre que observada uma borda de subida do sinal de relógio, com os valores lógicos dos sinais **tx** e **crédito** iguais a '1' lógico, além da avaliação da origem da comunicação. Este último critério garante que somente serão contabilizadas transmissões de dados oriundas de origens de comunicação que solicitaram a abertura de sessão com o destino.

Uma notificação de descumprimento somente é gerada pela NI destino se a taxa média e a última taxa de transmissão de dados observada nos canais de recepção forem inferiores àquela contratada. Se a taxa média de transmissão é maior do que a contratada, mas a última taxa obtida é inferior, isto é interpretado pelo slvMonitor como uma possível diminuição momentânea da taxa de transmissão, e é ignorada. Se a taxa média de transmissão é menor que a contratada, mas a última taxa observada é maior, isto é interpretado pela slvMonitor como uma possível reação a uma queda prévia da taxa média de transmissão, e também é ignorado. Estes dois últimos casos não geram notificação à NI origem. Nos casos onde tem de ser gerada uma notificação de descumprimento de taxa contratada, a NI destino desabilita o monitoramento dos canais de recepção de dados até que seja dada uma resposta por parte da NI origem da comunicação. Esta ação tem por objetivo coibir a geração de novas notificações de descumprimento da taxa de transmissão até que a negociação corrente de uma nova rota seja finalizada.

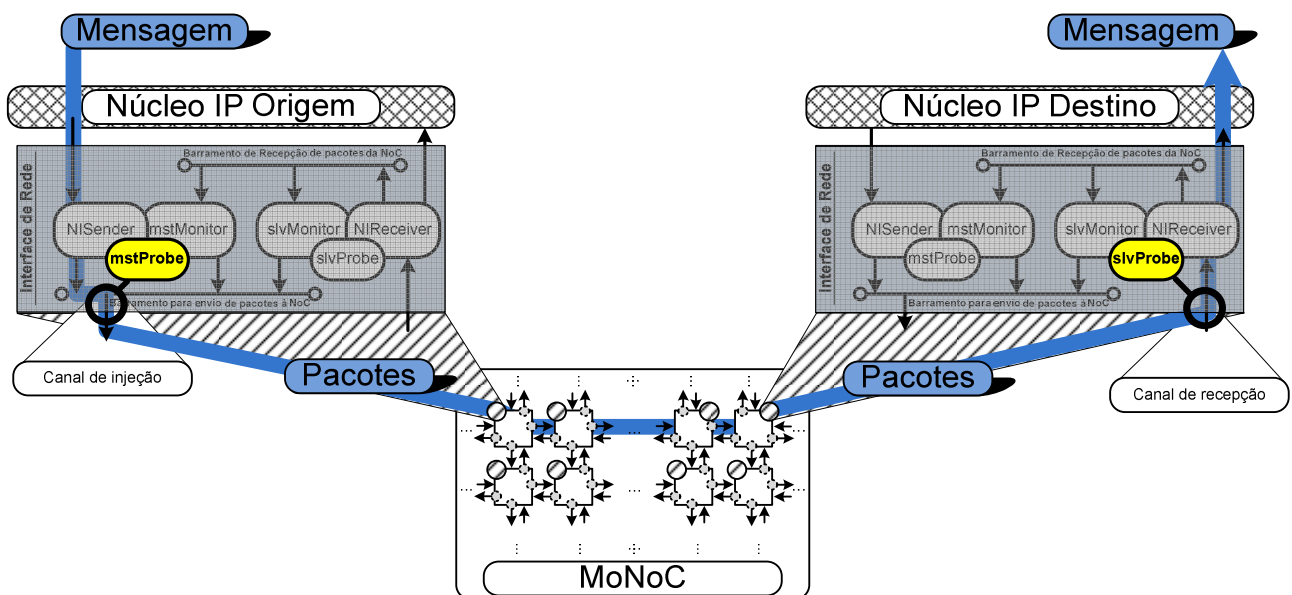


Figura 5.14 – Transmissão de dados pela MoNoC. Em destaque os núcleos slvProbe e mstProbe e seus respectivos pontos de observação de tráfego para coleta de estatística sobre taxa de injeção (mstProbe) e

recepção (slvProbe) de dados.

Quando a NI origem recebe uma notificação de descumprimento de taxa de transmissão contratada, o mstMonitor requisita ao módulo mstProbe a taxa média e a última taxa de ocupação observadas na injeção de dados para o destino reclamante. Note que a métrica de ocupação não é necessariamente igual à de transmissão de dados. Nos casos onde toda tentativa de transmissão é bem sucedida, os valores medidos de ocupação são iguais ao de transmissão. Sendo assim, se a taxa média ou a última taxa de ocupação são inferiores à taxa de transmissão contratada, o módulo mstMonitor interpreta que houve uma baixa injeção de dados a partir do núcleo IP origem da comunicação, o que é notificado à NI destino. Porém, se a taxa média e a última taxa de ocupação forem iguais ou superiores àquela contratada, então o mstMonitor considera que houve congestionamento na rota que está sendo utilizada para comunicação. Neste caso, o mstMonitor desabilita o monitoramento do seu canal de injeção e encaminha pacotes à NI destino sugerindo novas rotas (um pacote por rota). A desativação da observação das taxas de transmissão ocorre, pois não é interrompido o serviço de transmissão dos dados entre os núcleos IP, independente da adaptação de rota que está sendo negociada. O objetivo desta abordagem é concentrar as estatísticas para cada rota. No caso de uma troca de rota, se a coleta de estatísticas tivesse sido mantida, então uma próxima avaliação de congestionamento poderia não ser coerente. O monitoramento dos canais de injeção somente é retomado quando a NI origem da comunicação recebe o resultado sobre uma decisão de rota a ser utilizada.

Os pacotes contendo as propostas de rota, encaminhados pela NI origem, coletam os dados de ocupação dos canais de rede por onde passam. Cada um destes pacotes trafega na rota que está propondo. Ao chegarem à NI destino, eles são interpretados como propostas de rotas e encaminhados para o módulo slvMonitor, que tem por função colher todas as propostas e eleger a mais adequada. Elege-se a rota na qual for constatada a menor taxa média e o menor pico de ocupação (i.e. o menor ponto quente) dos canais que a compõem. Ilustrado na Figura 5.15, propõe-se um conjunto de alternativas de rotas para um par comunicante hipotético, sendo a rota (A) a escolhida. Após receber todas as propostas de rota e finalizar o julgamento daquela mais adequada, o módulo slvMonitor notifica à NI origem sobre o resultado de sua avaliação. Encaminhada a notificação, a NI destino volta a habilitar o monitoramento do canal de recepção de dados, tornando-se apta a voltar a notificar o descumprimento da taxa contratada.

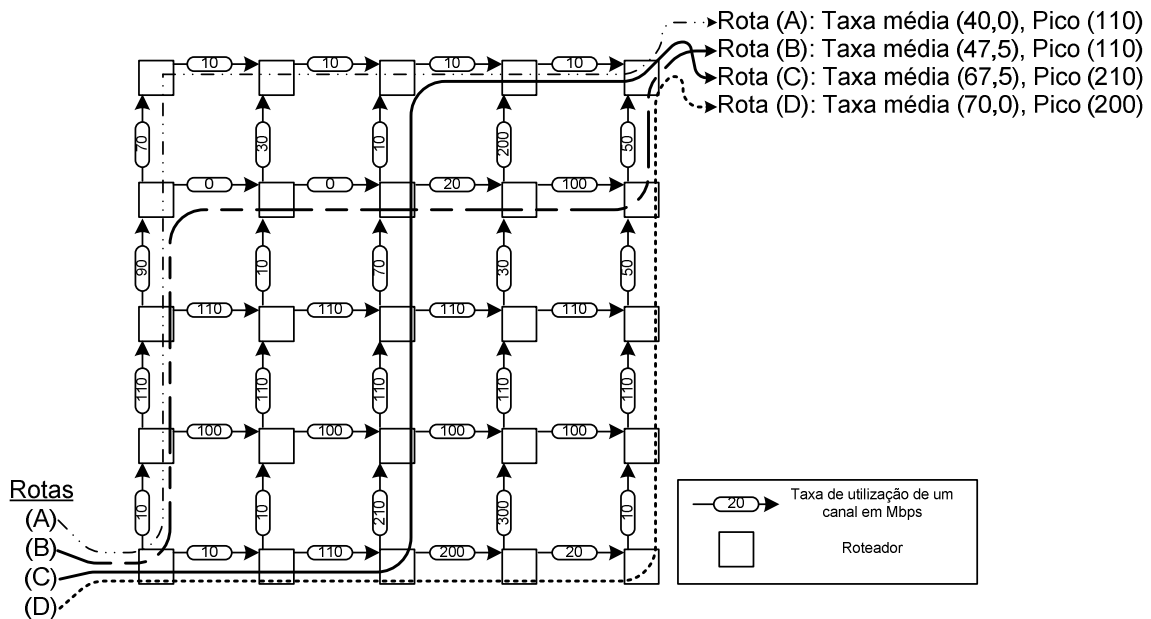


Figura 5.15 - Propostas de alternativas de rotas. Das rotas apresentadas, a rota (A) é a que proporciona menor taxa média e menor pico de ocupação.

Ao receber da NI destino a notificação do resultado, a NI origem assume a rota informada como aquela a ser adotada na comunicação entre os dois núcleos IP para o serviço de transporte de dados com esforço de atendimento de taxa de transmissão. Ao mesmo tempo, o monitoramento volta a ser habilitado na NI Origem. A Figura 5.16 resume as decisões aqui expostas desde a avaliação da necessidade de notificação por descumprimento taxas contratadas.

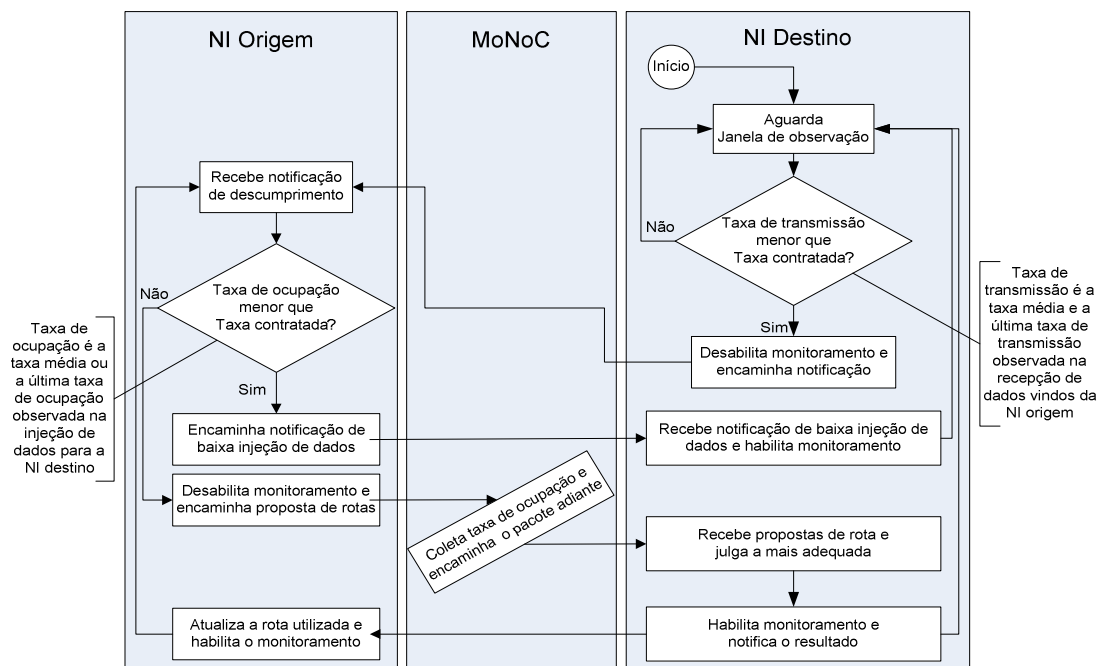


Figura 5.16 – Diagrama de estados das NIs com foco no suporte a adaptação de rotas.

## 5.4 IMPLEMENTAÇÃO EM *HARDWARE*

Todos os componentes que forma a MoNoC foram descritos em VHDL no nível de transferência entre registradores (em inglês, *Register Transfer Level*, RTL) utilizando o subconjunto sintetizável. As descrições tiveram como dispositivo alvo arquiteturas do tipo FPGA (xc2vp30-5fg676) da Xilinx. No processo de síntese utilizou-se a ferramenta XST que compõe o pacote de ferramentas ISE 10.1 para o projeto FPGA. Os resultados de área capturados após síntese para um roteador central da MoNoC são apresentados na Tabela 5.1.

Tabela 5.1 – Resultado de síntese de um roteador central da MoNoC (5 portas de entrada e saída)

Núcleo IP	LUTs
1 - Roteador central	10.212
1.1 - Porta de entrada	848
1.1.1 - Canal virtual de entrada	236
1.2 - Porta de saída	1.070
1.2.1 - Canal virtual de saída	113
1.2.2 - Canal virtual de saída + controle	278
1.2.3 - Sonda de rede	302

Na Tabela 5.1 discrimina-se o custo em LUTs (*look up tables*) de cada um dos componentes que formam o roteador central, conforme listagem hierarquizada. Um roteador central é composto por 5 portas de entrada, 20 canais virtuais de entrada, 5 portas de saída, 15 canais virtuais de saída, 5 canais virtuais de controle e 5 sondas de rede, conforme ilustrado na Figura 5.6. Cabe ressaltar que o roteador sintetizado utiliza filas de armazenamento na entrada de 4 (quatro) posições. Destaca-se principalmente o núcleo IP referente ao canal virtual de saída utilizado para controle (Tabela 5.1 – 1.2.2). Este é utilizado nas portas de saída e apenas para os canais virtuais de mais alta prioridade da MoNoC. Seu custo é maior em relação aos núcleos IP canal virtual de saída por adicionar mecanismos de interpretação de pacotes de controle e de comunicação com a sonda de rede. O mesmo procedimento de síntese foi aplicado ao núcleo IP correspondente a interface de rede. Os resultados de síntese são apresentados na Tabela 5.2.

Os núcleos IP que compõem a interface de rede são parcialmente ilustrados no diagrama de blocos da Figura 5.12. No presente trabalho, uma interface de rede é composta por uma instância de cada um dos núcleos IP listados na Tabela 5.2. O núcleo IP referente a memória foi implementado com 100 (cem) posições de 1 (um) *flit* cada. Destaca-se aqui o custo reduzido das sondas mestre e escravo, quando comparadas a sonda de rede. Isto se deve principalmente pelo fato desta

última sonda implementar uma máquina de estados para interpretação de comandos enviados pela rede, enquanto simplificou-se as sondas mestre e escravo presentes na interface de rede.

Tabela 5.2 – Resultado de síntese de uma interface de rede da MoNoC

<b>Núcleo IP</b>	<b>LUTs</b>
1 – Interface de rede	5.742
1.1 – NI Sender	1.254
1.2 – NI Receiver	205
1.3 – Monitor Mestre	1.286
1.4 – Sonda Mestre	193
1.5 – Monitor Escravo	1.150
1.6 – Sonda Escrava	193
1.7 – Memória	1.858
1.8 – Barramento Monitores	27
1.9 – Barramento Memória	57
1.10 – Barramento NoC	36

Cabe ressaltar que os resultados obtidos tanto para o roteador quanto para a interface de rede são frutos de uma implementação voltada para prova de conceito. Durante a descrição dos núcleos IP não houve qualquer tipo de cuidado especial de otimização de área, tendo sido mantido o foco na garantia de funcionalidade e de desempenho destes.



## 6. VALIDAÇÃO E RESULTADOS

Este Capítulo apresenta experimentos conduzidos para validar o uso de algoritmos de roteamento complementares em redes em chip (Seção 6.1) e a proposta de adaptação de rotas configurada pela MoNoC (Seção 6.2). Os resultados não se limitam a avaliar a infraestrutura de comunicação MoNoC, incluem outras infraestruturas de comunicação para a validação das propostas e a comparação de resultados.

### 6.1 EMPREGO DE ALGORITMOS COMPLEMENTARES

O primeiro experimento conduzido teve por objetivo principal validar funcionalmente a coexistência de algoritmos de roteamento distintos em uma mesma infraestrutura de comunicação. Para dar suporte a tal validação, desenvolveu-se uma NoC para o emprego de algoritmos complementares conforme descrito no Capítulo 5. Alguns cenários de tráfego sintéticos foram utilizados no processo de validação e reutilizados no processo de avaliação. Compararam-se os resultados com aqueles obtidos quando se aplicam os mesmos cenários de tráfego a infraestruturas de comunicação que empregam apenas um tipo de algoritmo de roteamento. As Seções a seguir detalham os recursos desenvolvidos nesta etapa.

#### 6.1.1 SUPORTE A COMPLEMENTARIDADE DE ALGORITMOS

Para dar suporte à proposta de emprego de algoritmos de roteamento complementares, desenvolveu-se uma rede em chip com dois canais virtuais, chamada NoC\_Complementar. Em um canal virtual é permitido o tráfego de pacotes roteados com um algoritmo de roteamento, tal como o algoritmo *West first mínimo*, enquanto no outro trafegam pacotes roteados com a versão complementar deste algoritmo de roteamento, tal como o *East first mínimo*. A Figura 6.1 ilustra a diferenciação entre os canais virtuais onde trafegam pacotes roteados com uma versão original do algoritmo de roteamento (cv\_orig) e onde trafegam pacotes com a versão complementar (cv\_comp).

Todos os algoritmos adaptativos detalhados no Capítulo 3 foram empregados na validação



da complementaridade, ou seja, (i) *West first*, (ii) *North last* e (iii) *negative first*. Implementou-se uma versão complementar para cada um destes algoritmos, ou sejam, o *East first* para o *West first*, o *South last* para o *North last* e o *positive first* para o *negative first*. Os algoritmos de roteamento foram implementados em sua versão mínima, na qual apenas são aceitos os caminhos mais curtos (dado em número de *hops*) entre um dado par comunicante. Na NoC\_Complementar as decisões de roteamento são feitas de forma distribuída (i. e. roteador a roteador) e em tempo de execução.

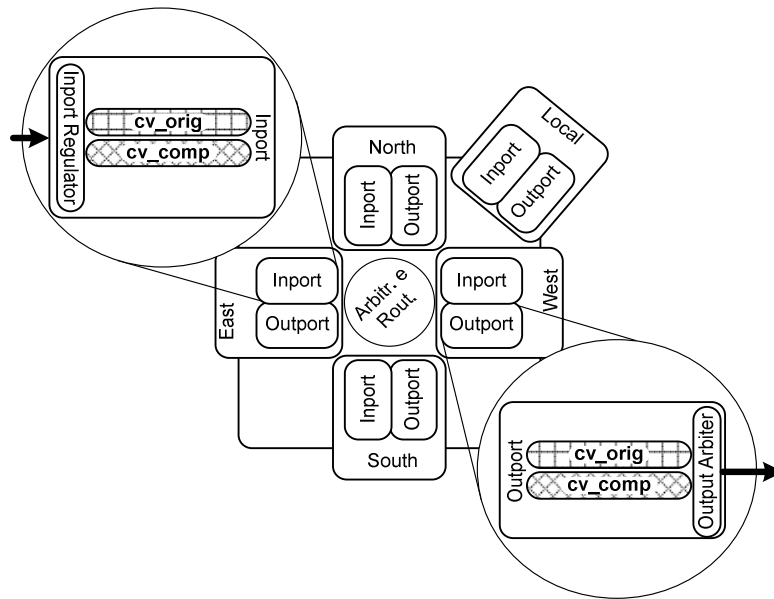


Figura 6.1 – Estrutura do roteador da NoC\_Complementar, mostrando a diferenciação dos canais virtuais por algoritmo de roteamento empregado.

Quando um pacote é injetado na NoC\_Complementar, decide-se, a partir da posição relativa entre dos elementos do par comunicante e dos algoritmos complementares em uso, qual canal virtual empregar na transmissão (i. e. *West first/East first* ou *North last/South last* ou *negative first/positive first*). Para redes de topologia malha 2D, o posicionamento relativo entre pares comunicantes referido a seguir respeita as definições de nomenclatura apresentadas na Seção 3.2.2. A Figura 6.2 ilustra as decisões de canal virtual usadas na transmissão, dadas pela relação entre os possíveis canais virtuais e algoritmos de roteamento na NoC\_Complementar.

Para o conjunto de algoritmos de roteamento *West/East first*, se o destino está à direita da origem de comunicação (i.e.  $QdX_{Pos}Y_{Pos}$  ou  $QdX_{Pos}Y_{Neg}$ ) ou se está alinhado horizontalmente com a origem (i.e.  $MmoX$ ) então utiliza-se o canal *cv\_orig*. Se um destino está à esquerda da origem da comunicação (i.e.  $QdX_{Neg}Y_{Pos}$  ou  $QdX_{Neg}Y_{Neg}$ ) ou se está alinhado verticalmente com a origem (i.e.  $MmoY$ ) então o canal a ser utilizado na transmissão é o *cv\_comp*, que emprega o algoritmo *East*

*first*. A Figura 6.2 (*West/East first*) ilustra as alternativas de decisão de roteamento, baseado na posição relativa do par comunicante. Denomina-se NoC\_Comp\_WEF a NoC\_Complementar que emprega os algoritmos *West/East first*.

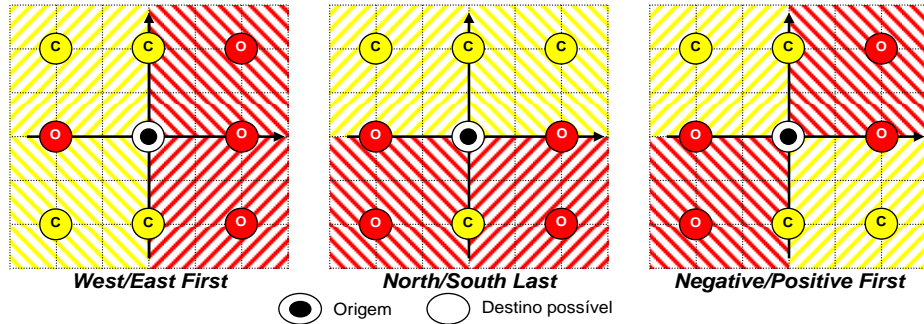


Figura 6.2 – Escolha do canal virtual para transmissão na NoC\_Complementar para cada um dos algoritmos de roteamento. A letra “O” representa o emprego do canal virtual  $cv\_orig$ , enquanto “C” representa o uso do canal virtual  $cv\_comp$  para a transmissão dos pacotes.

Para o conjunto de algoritmos de roteamento *North/South last*, se o destino está abaixo da origem da comunicação (i.e.  $QdX_{Pos}Y_{Neg}$  ou  $QdX_{Neg}Y_{Neg}$ ) ou se está alinhado horizontalmente com a origem (i.e.  $MmoX$ ) então utiliza-se o canal  $cv\_orig$ . Se o destino está acima da origem da comunicação (i.e.  $QdX_{Pos}Y_{Pos}$  ou  $QdX_{Neg}Y_{Pos}$ ) ou se está alinhado verticalmente com a origem (i.e.  $MmoY$ ) então o canal a ser utilizado na transmissão é o  $cv\_comp$ , que emprega o algoritmo *South last*. A Figura 6.2(*North/South last*) ilustra as alternativas de decisão de roteamento baseadas na posição relativa de um par comunicante. Denomina-se NoC\_Comp\_NSL a NoC\_Complementar que emprega os algoritmos *North/South last*.

Por fim, para o algoritmo de roteamento *negative/positive first*, se o destino está abaixo e à esquerda da origem de comunicação (i.e.  $QdX_{Neg}Y_{Neg}$ ), acima e à direita (i.e.  $QdX_{Pos}Y_{Pos}$ ) ou se está alinhado horizontalmente com a origem (i.e.  $MmoX$ ) então se utiliza o canal  $cv\_orig$ . Se o destino está acima e à esquerda da origem da comunicação (i.e.  $QdX_{Neg}Y_{Pos}$ ), abaixo e à direita (i.e.  $QdX_{Pos}Y_{Neg}$ ) ou se está alinhado verticalmente com a origem (i.e.  $MmoY$ ) então o canal a ser utilizado na transmissão é o  $cv\_comp$ , que emprega o algoritmo *positive first*. A Figura 6.2(*negative/positive first*) ilustra as alternativas de decisão de roteamento baseadas na posição relativa de um par comunicante. Denomina-se NoC\_Comp\_NPF a NoC\_Complementar que emprega os algoritmos *negative/positive first*.

Na NoC\_Complementar, a inserção dos pacotes na rede pode ocorrer por qualquer um dos

canais virtuais, independente do posicionamento relativo do par comunicante. A decisão do canal virtual a ser assumido durante o trânsito de um dado pacote pela rede em chip é de responsabilidade do roteador origem da comunicação. Assim, sempre que um dado pacote entra pela porta Local, o roteador decide qual o canal virtual adequado, de acordo com os algoritmos adotados. O canal virtual assumido no roteador origem da comunicação não muda até o pacote chegar ao roteador destino. Ao chegar ao destino, tal obrigatoriedade é extinta e o pacote pode sair por qualquer um dos canais virtuais. Tal característica foi assumida para facilitar o processo de adoção da NoC\_Complementar, delegando tal etapa de decisão à própria infraestrutura de comunicação.

### 6.1.2 VALIDAÇÃO DO EMPREGO DE ALGORITMOS COMPLEMENTARES

Para esta etapa, desenvolveu-se um ambiente de validação em SystemC, cujos principais módulos implementam produtores (e.g. geradores de estímulo) e consumidores (i.e. armazenadores de resultados), conforme ilustra a Figura 6.3. Cada módulo produtor captura o conteúdo de um arquivo de entrada, onde está especificado o comportamento esperado de injeção de um conjunto de pacotes. Nestes arquivos são definidos os momentos de inserção dos pacotes na rede (em ciclos de relógio) e o conteúdo de cada pacote. Ao iniciar a inserção de um pacote na rede, os produtores adicionam uma etiqueta de tempo que contém o momento da simulação em que o pacote foi realmente injetado na rede, representado pelo número de ciclos contabilizados desde o início da simulação. Os consumidores são responsáveis pela captura dos pacotes transmitidos pela rede e destinados ao núcleo IP que ele representam. Após a captura de cada pacote, o módulo consumidor armazena seu conteúdo e o momento de recepção em um arquivo de saída único para cada consumidor.

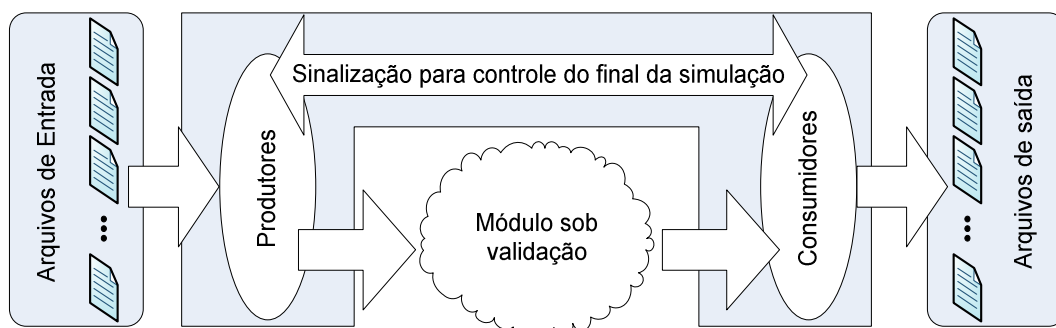


Figura 6.3 - Ambiente empregado na validação das NoC\_complementares.

Nesta etapa de validação, a garantia de funcionalidade das infraestruturas de comunicação

se dá através da entrega de todos os pacotes definidos nos arquivos de entrada em seus destinos corretos. A entrega de todos os pacotes provoca o fim da simulação. Um mecanismo de sinalização de fim de injeção de todos de pacotes, destacado na Figura 6.3, é comandado pelos produtores e observado pelos consumidores. Quando todos os produtores concluem o envio de todos os pacotes especificados em seus arquivos de entrada, uma notificação é encaminhada aos consumidores. A partir da notificação de que todos os pacotes definidos nos arquivos de entrada já foram injetados e dada a garantia de inexistência de pacotes em trânsito na rede em chip, os consumidores determinam o final da simulação.

A ferramenta Atlas foi utilizada para a geração dos arquivos de entrada e para as chamadas do simulador. Os parâmetros de geração destes cenários de tráfego foram similares àqueles empregados na criação dos cenários de tráfego utilizados nas simulações apresentadas no Capítulo 4. Para todos variou-se a taxa de injeção, e manteve-se fixa em 1000 a quantidade de pacotes inseridos a partir de cada origem de comunicação. As taxas de injeção foram variadas entre 10%, 20%, 30%, 40% e 50% da capacidade de injeção dos canais da NoC, definidas a partir do tempo de ocupação dos canais de transmissão assumindo-se a inexistência de contenção. Tomando-se por base a Hermes-SR e assumindo-se a mesma frequência de operação, ou seja, 50 MHz, os percentuais referidos acima representam uma taxa de injeção de 80 Mbps, 160 Mbps, 240 Mbps, 320 Mbps e 400 Mbps, respectivamente.

Foram gerados três cenários de tráfego: complemento, *hotspot* e *todos para todos*. A Figura 6.4 ilustra a distribuição espacial dos cenários de tráfego complemento e o *hotspot*. Quanto ao cenário de tráfego *todos para todos* a distribuição espacial assumida é aleatória. O emprego do cenário de tráfego *todos para todos* cria uma situação de difícil associação com aplicações reais, visto que os destinos de comunicação mudam com frequência. Nos casos gerados para injeção representando 10% da capacidade de transmissão de um canal, o percentual de pacotes enviados consecutivamente para um mesmo destino a partir de cada origem variou de 3,5% a 5,2%. Apesar do alto grau de variação dos destinos dos pacotes a partir de cada origem, tal cenário de tráfego permite mais facilmente deparar-se com eventuais fragilidades da infraestrutura de comunicação, tal como a inabilidade para evitar *deadlock*. Para todos os cenários de tráfego assumiu-se distribuição temporal uniforme para a definição dos momentos de envio de pacotes a partir de cada origem.

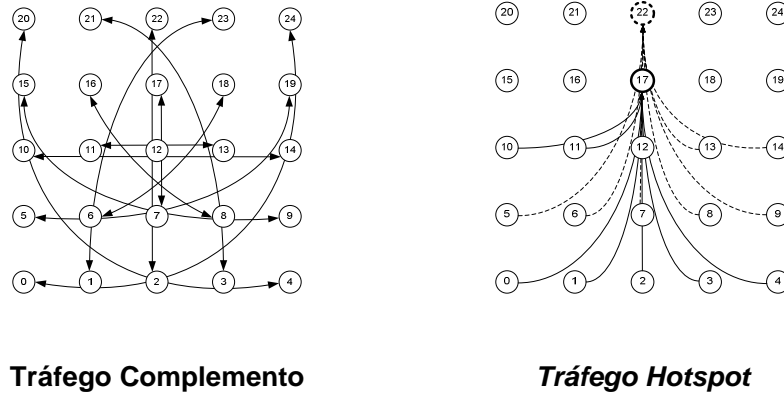


Figura 6.4 - Cenários de tráfego Complemento e *Hotspot* aplicados durante etapa de validação.

Ao todo foram executadas 180 simulações durante esta etapa, quantidade esta resultante das combinações de infraestruturas de comunicação (3 estratégias de roteamento complementares e 4 tamanhos distintos de *buffer* resultando em 12 NoCs) e de cenários de tráfego (3 padrões de distribuição espacial, 1 padrão de distribuição temporal e 5 taxas distintas de injeção de tráfego, resultando em 15 cenários de tráfego). Como resultado da etapa de validação, a funcionalidade das NoCs aplicando roteamento complementar foi comprovada com a entrega correta de todos os pacotes inseridos nas infraestruturas de comunicação. A entrega de todos os pacotes foi garantida pelo final automático das simulações e adicionalmente pela contabilização dos pacotes nos arquivos de saída. No cenário de tráfego *todos para todos* um total de 25000 pacotes foram entregues. No cenário complemento foram 24000 e enquanto no cenário *Hotspot* foram 15000. Comprovou-se a entrega correta dos pacotes nos destinos via análise de arquivos de saída e do destino de cada pacote ali armazenado.

### 6.1.3 AVALIAÇÃO DE RESULTADOS

Comparou-se os resultados de simulação obtidos para os cenários de tráfego utilizados nas diferentes NoC\_Complementares com quatro outras NoCs. Em três destas NoCs utilizaram-se os algoritmos de roteamento adaptativos base da complementaridade, ou seja *West first mínimo* (NoC\_WFM), *North last mínimo* (NoC\_NLM) e *negative first mínimo* (NoC\_NFM). Na quarta NoC, foi utilizado o algoritmo de roteamento *XY puro* (NoC\_XY). Os algoritmos *West first mínimo*, *North last mínimo*, *negative first mínimo* e *XY puro* seguem as parametrizações detalhadas na Seção 3.2.

À exceção da variação das características de roteamento das NoCs, as demais definições arquiteturais seguem a especificação da NoC Hermes [MOR04]: controle de fluxo baseado em créditos, filas de entrada para armazenamento temporário de *flits*, modo de chaveamento *wormhole*,

algoritmo de arbitragem *round-robin*, e roteador dotado de unidade compartilhada por todas as portas do roteador para arbitragem e roteamento. Todas as NoCs têm topologia malha 2D 5x5.

Os cenários de tráfego apresentados na Seção anterior (i. e. complemento, *hotspot* e *todos para todos*) foram utilizados para simulação com as infraestruturas de comunicação desenvolvidas neste ponto. Ao todo, 240 novas simulações foram executadas nesta etapa. Apresenta-se aqui uma primeira análise que reflete o cenário de tráfego *todos para todos*, onde se variaram as taxas de injeção a partir de cada ponto da rede de 10% a 50% (em passos de 10%) da capacidade de transmissão dos canais. Os resultados apresentados refletem a latência de rede e de aplicação (diferenciadas no Capítulo 4, e ilustradas na Figura 4.1) capturadas para os pacotes inseridos a partir do ponto central da rede, tendo como alvo os extremos na infraestrutura de comunicação, conforme ilustra a Figura 6.5.

Limitou-se a comparação de resultados àqueles obtidos para as infraestruturas de comunicação empregando três algoritmos de roteamento: algoritmo complementar (*West/East first mínimo*), *West first mínimo* e XY, sempre com filas de entrada com capacidade de até quatro *flits*. Cabe lembrar que a única diferença entre estas infraestruturas é o algoritmo de roteamento, todas empregam dois canais virtuais. A Tabela 6.1 apresenta os resultados capturados para a simulação utilizando o cenário de tráfego *todos para todos* com 80 Mbps de taxa de injeção a partir de cada origem.

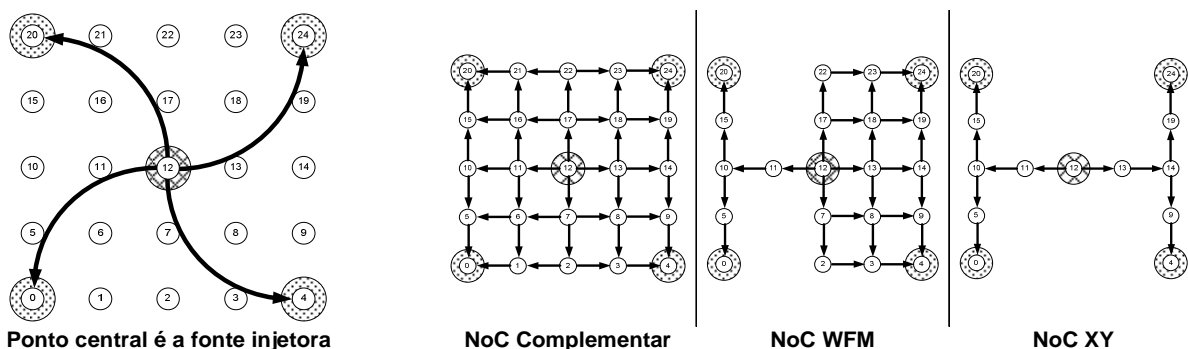


Figura 6.5 – À esquerda, pares comunicantes analisados no cenário de tráfego *todos para todos*. À direita, os três conjuntos possíveis de canais de comunicação para a transmissão de pacotes do centro para os extremos quando se usam diferentes algoritmos de roteamento. Complemento (*West/East first mínimo*), WFM empregando somente *West first mínimo* e XY empregando XY puro.

Tabela 6.1 – Latências capturadas para o cenário de tráfego *todos para todos* com 80 Mbps de injeção de

tráfego quando a origem é o nó 12 da rede e os destinos são os nós 0, 4, 20 e 24.

		Latência de rede (NL)				Latência de aplicação (AL)			
NoC Complementar		NL mínima	NL máxima	NL média	NL desvio padrão	AL mínima	AL máxima	AL média	AL desvio padrão
	12->0	40	115	49.44	15.05	40	115	49.44	15.05
	12->4	40	91	53.82	12.30	40	91	53.82	12.30
	12->20	40	197	51.87	30.54	40	197	51.87	30.54
	12->24	40	75	50.80	9.41	40	75	50.80	9.41

		Latência de rede (NL)				Latência de aplicação (AL)			
NoC WFM		NL mínima	NL máxima	NL média	NL desvio padrão	AL mínima	AL máxima	AL média	AL desvio padrão
	12->0	<b>40</b>	<b>83</b>	<b>49.02</b>	<b>12.12</b>	<b>40</b>	<b>83</b>	<b>49.02</b>	<b>12.12</b>
	12->4	43	112	56.14	15.65	43	112	56.14	15.65
	12->20	<b>40</b>	<b>122</b>	<b>54.33</b>	<b>20.43</b>	<b>40</b>	<b>122</b>	<b>54.33</b>	<b>20.43</b>
	12->24	40	79	52.34	10.48	40	79	52.34	10.48

		Latência de rede (NL)				Latência de aplicação (AL)			
NoC XY		NL mínima	NL máxima	NL média	NL desvio padrão	AL mínima	AL máxima	AL média	AL desvio padrão
	12->0	<b>40</b>	<b>95</b>	<b>49.21</b>	<b>13.34</b>	<b>40</b>	<b>95</b>	<b>49.21</b>	<b>13.34</b>
	12->4	<b>43</b>	<b>98</b>	<b>55.73</b>	<b>14.19</b>	<b>43</b>	<b>98</b>	<b>55.73</b>	<b>14.19</b>
	12->20	<b>40</b>	<b>122</b>	<b>54.13</b>	<b>20.50</b>	<b>40</b>	<b>122</b>	<b>54.13</b>	<b>20.50</b>
	12->24	<b>40</b>	<b>107</b>	<b>56.74</b>	<b>15.82</b>	<b>40</b>	<b>107</b>	<b>56.74</b>	<b>15.82</b>

A partir da Tabela 6.1 pode-se observar que as latências de rede médias obtidas para os pares comunicantes não variam significativamente entre as infraestruturas de comunicação. Nota-se um leve ganho da arquitetura que emprega roteamento complementar sobre as demais em 75% dos casos no que se refere à latência de rede média. Dado que a latência de rede é similar à latência de aplicação em todos os casos, e dadas as características das infraestruturas de comunicação, pode-se afirmar que não houve congestionamento durante a transmissão dos pacotes.

A Figura 6.6 sumariza os valores de latência capturados para os mesmos pares origem destino apresentados na Tabela 6.1, variando-se a taxa de injeção a partir de cada um dos pontos da infraestrutura de comunicação.

Conforme se pode observar neste cenário de tráfego, o emprego de algoritmos complementares permite uma diminuição da latência de rede média em todos os casos apresentados na Figura 6.6(a), quando comparado com os demais mecanismos de roteamento. As reduções variam de 1,48 a 15,99 ciclos sobre o valor médio. Já no que se refere à latência de aplicação, a Figura 6.6(b) deixa evidente que o ponto de saturação é encontrado quando a taxa de injeção de pacotes ultrapassa 20% da capacidade de transmissão dos pacotes (160 Mbps). Nota-se tal característica pela diferença das latências de aplicação até se alcançar cerca de 20% da capacidade de transmissão e os valores de latência além deste ponto. À exceção do ponto onde a taxa de injeção foi de

320 Mbps, onde a NoC que usa o algoritmo de roteamento XY apresentou maior ganho, nos demais casos a infraestrutura que usa algoritmos complementares permitiu reduções da latência de aplicação média de até, aproximadamente, 43% (relação observável na Figura 6.6(b) entre comp e XY quando a taxa de injeção é de 240 Mbps).

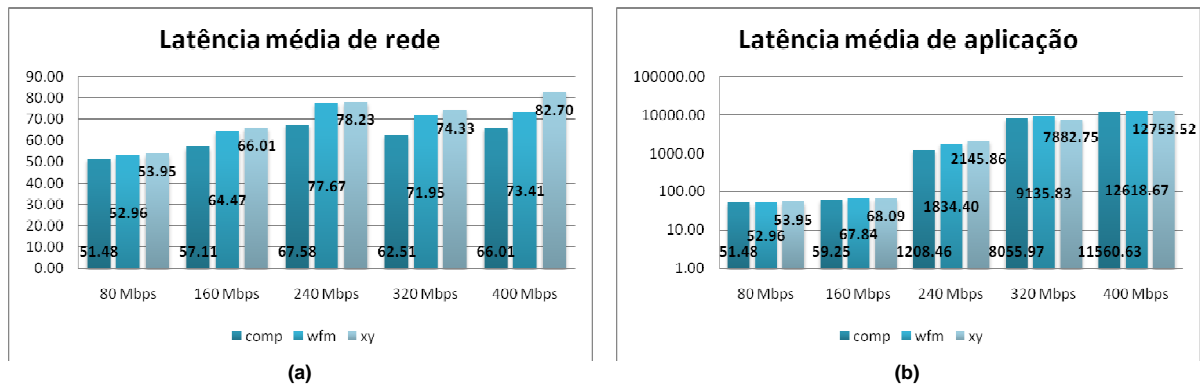


Figura 6.6 – Latências de rede e de aplicação médias em número de ciclos de relógio, capturadas durante simulação com emprego do cenário de tráfego *todos para todos* e variando-se a taxa de injeção de dados. Mostram-se os resultados obtidos para a NoC Complementar (comp), para a NoC usando apenas o algoritmo *West first minimo* (wfm) e a NoC aplicando o algoritmo XY (xy).

A segunda análise que se apresenta reflete o cenário de tráfego Complemento, onde se variam as taxas de injeção a partir de cada ponto da rede de 10% a 50% (com passos de 10%) da capacidade de transmissão dos canais. Os resultados apresentados na Tabela 6.2 refletem os valores das latências de rede e de aplicação capturados para os pacotes cujos pares encontram-se nos pontos extremos da rede, conforme ilustra a Figura 6.7.

A Tabela 6.2 apresenta valores de latência capturados quando a taxa de injeção de pacotes é de 160 Mbps para cada origem. As limitações de canais disponíveis para roteamento para as infraestruturas NoC\_WFM e NoC\_XY em relação a NoC\_Complementar, conforme ilustra a Figura 6.7, é evidente na Tabela 6.2. Em todos os casos, a latência de rede média da infraestrutura NoC\_Complementar foi melhor que as das demais infraestruturas de comunicação. Quando se observa a latência de aplicação alcançada por par comunicante, a vantagem da NoC\_Complementar destaca-se ainda mais. A redução máxima de latência de aplicação média alcançada na NoC\_Complementar em relação à NoC\_WFM chegou a 108 (cento e oito) vezes, quando a origem é o nó 4 (quatro) e o destino é o nó 20 (vinte). Comparando-se as latências de aplicação médias da infraestrutura para todos os pares comunicantes, a NoC\_Complementar reduziu em aproximadamente 35 (trinta e cinco) vezes a latência alcançada pela NoC\_XY.



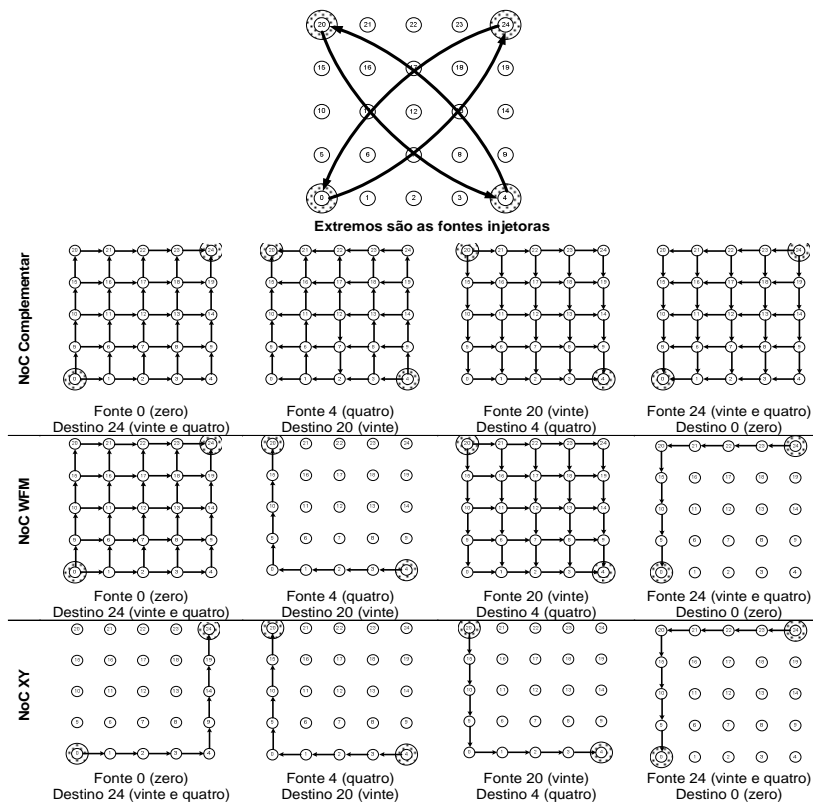


Figura 6.7 – No topo, pares comunicantes analisados no cenário de tráfego Complemento. Seguem-se os possíveis caminhos usados para a transmissão de pacotes entre cada par comunicante quando da utilização das diferente infraestruturas de comunicação.

Tabela 6.2 - Latências capturadas para o cenário de tráfego Complemento quando a taxa de injeção por nó é de 160 Mbps.

		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
NoC Complementar	0->24	60	361	104.61	46.30	60	895	183.76	125.96
	4->20	60	276	101.50	34.46	60	409	131.37	63.02
	20->4	60	400	110.35	51.65	60	1712	441.01	355.64
	24->0	60	299	107.69	39.04	60	758	228.57	153.10
NoC WFM		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
	0->24	61	453	111.62	42.06	61	513	183.55	98.27
	4->20	101	388	135.01	33.19	144	23333	14109.12	7191.84
NoC XY		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
	0->24	67	251	125.06	19.03	144	13548	7042.03	4032.10
NoC XY		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
	4->20	70	298	127.90	28.43	135	19541	10412.94	5841.63
NoC XY		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
	20->4	67	238	125.68	19.68	129	13989	6930.46	4153.17
NoC XY		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
	24->0	66	344	128.77	32.49	140	21034	10966.55	6337.17

A Figura 6.8 sumariza os resultados de latência obtidos para o cenário de tráfego complemento quando se aplica diferentes taxas de injeção com as infraestruturas de comunicação avalia-

das.

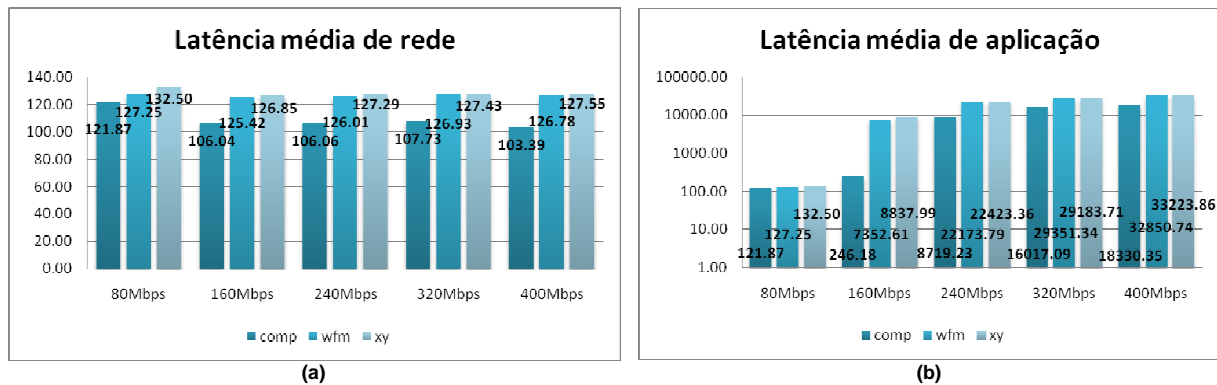


Figura 6.8 - Latências médias, em ciclos de relógio, capturadas durante simulação com emprego do cenário de tráfego complemento e variando-se a taxa de injeção de dados na rede. Os resultados derivam de dados de simulação obtidos para a NoC\_Complementar (comp), para a NoC\_WFM (wfm) e para a NoC\_XY (xy).

Como se pode observar, para o cenário de tráfego complemento, o emprego da infraestrutura de roteamento complementar permite uma diminuição da latência média de rede em todos os casos apresentados na Figura 6.8(a). Ainda no que se refere à latência de rede média, pode-se notar que mesmo aumentando-se a taxa de injeção, os valores de latência média diminuiram. Apesar disto, a avaliação do desvio padrão médio destas latências mostra que quanto maior a taxa de injeção, mais dispersa é a amostra, tendo variado de 20 a 40 ciclos de relógio em média. Quanto à latência média de aplicação, a Figura 6.8(b) permite observar ganho da NoC\_Complementar em todos os casos.

A terceira e última análise reflete o cenário *hotspot*. Assim como realizado para os outros cenários de tráfego, variou-se aqui a taxa de injeção de 10% a 50% (em passos de 10%) da capacidade de transmissão dos canais. A Figura 6.9 ilustra os canais disponíveis para o roteamento dos pacotes em cada uma das infraestruturas de comunicação para o cenário de tráfego sob análise.

A Tabela 6.3 apresenta os resultados de latência obtidos para o cenário de tráfego *hotspot* quando se aplica a taxa de injeção 80 Mbps na simulação com as infraestruturas de comunicação avaliadas. As linhas da Tabela diferenciam-se pela distância entre as origens de transmissão e o destino da comunicação, que varia entre dois e cinco *hops*.

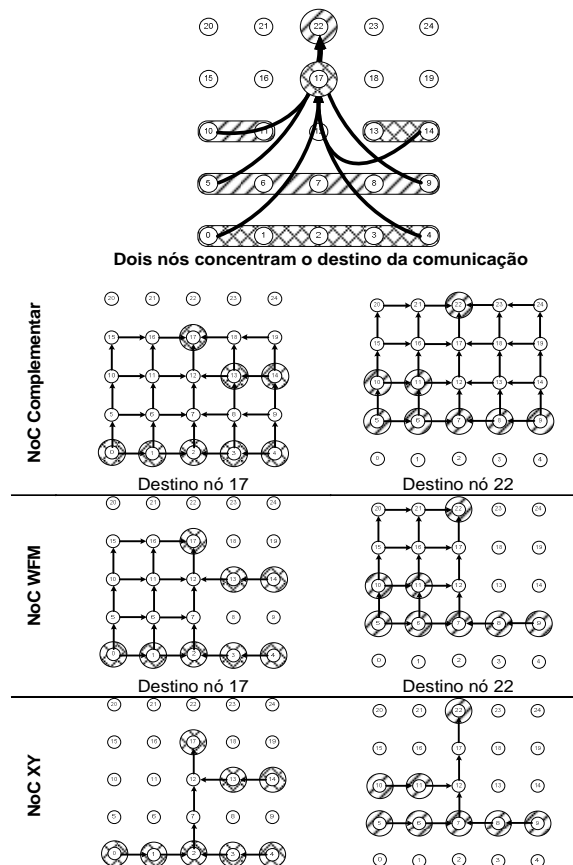


Figura 6.9 - No topo, pares comunicantes analisados no cenário de tráfego *hotspot*. Abaixo, os possíveis caminhos de comunicação usados para a transmissão de pacotes entre cada par comunicante quando da utilização das diferentes infraestruturas de comunicação.

Tabela 6.3 - Latências capturadas para o cenário de tráfego *hotspot* quando a taxa de injeção por nó é de 80 Mbps. As linhas de cada tabela parcial são diferenciadas pela distância em *hops* entre origens e destinos.

		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
NoC Complementar	2 hops	30	405	80.64	40.62	30	405	83.27	44.91
	3 hops	35	24928	146.51	460.86	35	160336	43540.02	52381.85
	4 hops	40	24877	170.68	408.63	40	161241	29007.30	48286.14
	5 hops	52	2334	167.66	136.31	72	39855	11881.37	14240.87
		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
NoC WFM	2 hops	31	235	93.61	43.24	31	258	95.86	45.45
	3 hops	35	36492	148.51	756.36	35	162182	43256.61	54486.34
	4 hops	51	14318	210.93	314.41	81	148716	51890.64	43930.18
	5 hops	51	1455	178.61	127.08	54	95884	24440.84	32428.10
		Latência de rede (NL)				Latência de aplicação (AL)			
		NL	NL	NL	NL	AL	AL	AL	AL
		mínima	máxima	média	desvio padrão	mínima	máxima	média	desvio padrão
NoC XY	2 hops	32	135	76.45	26.61	32	135	76.45	26.61
	3 hops	40	35518	184.53	747.54	40	154389	46740.40	56801.81
	4 hops	73	3278	231.39	304.54	77	198394	69292.98	62686.67
	5 hops	78	3039	263.46	325.21	186	198452	86702.89	58431.84

Para o cenário de tráfego *hotspot*, as vantagens do emprego da NoC\_Complementar ficam evidentes quando os núcleos IP comunicantes distam pelo menos 4 (quatro) *hops*. Isto se deve ao fato de que os nós injetores de dados com tal distância estão em posições com maior quantidade de alternativas de roteamento, sempre que empregada a infraestrutura de comunicação NoC\_Complementar.

## 6.2 EMPREGO DA INFRAESTRUTURA DE COMUNICAÇÃO MONOC

Apresentada no Capítulo 5, a MoNoC foi criada para disponibilizar uma infraestrutura de comunicação passível de síntese e servindo à investigação de diversos aspectos de comunicação em chip. Concluída a descrição da MoNoC empreendeu-se um conjunto de passos para validá-la, o que permitiu garantir a funcionalidade de cada módulo que a compõe. A validação foi dividida em etapas, iniciando-se pelos módulos da rede de interconexão, a seguir passando pelos módulos de interface de rede e finalizando pela integração de todos estes para a validação da infraestrutura completa.

Na presente Seção apresenta-se a organização adotada para a validação da arquitetura de comunicação, finalizando pela discussão dos resultados obtidos quando do emprego da MoNoC em cenários de tráfego estáticos e dinâmicos. Um *cenário de tráfego estático* é aquele em que os pares comunicantes são conhecidos em tempo de projeto e que não mudam ao longo de uma dada execução. Um *cenário de tráfego dinâmico* é aquele em que os pares comunicantes variam ao longo da execução, sendo que tal variação pode ser programada (e. g. via escalonamento) ou aleatória. O presente trabalho explora esta última alternativa.

### 6.2.1 VALIDAÇÃO DA INFRAESTRUTURA DE COMUNICAÇÃO

A primeira validação foi reservada para a rede de interconexão. Conforme detalhado no Capítulo 5, para o desenvolvimento da infraestrutura de comunicação a ser empregada na MoNoC adaptou-se as especificações da Hermes-SR pela adição de algumas características, incluindo a inserção de mecanismos para controle de canais virtuais e de comunicação com sondas de rede. A validação da infraestrutura de comunicação foi realizada a partir de treze (13) passos, realizando processo similar àquele aplicado para a infraestrutura de comunicação NoC\_Complementar, para a qual foram descritos ambientes em SystemC que integraram os núcleos IP descritos em VHDL da rede em chip e simulados com a ferramenta Modelsim da Mentor.

No primeiro passo validou-se a transmissão de pacotes pelo canal virtual de saída que permite operações com a sonda de rede. Diferente dos demais canais virtuais de saída, este exigiu a criação de uma máquina de estados específica para interpretar pacotes que trafegam, visto que ambos, pacotes de manipulação com a sonda de rede e pacotes de dados podem estar sendo transmitidos. Um ambiente específico foi desenvolvido para variar o tipo de pacote que trafega pela rede, o tamanho destes, os comandos e os parâmetros de cada comando.

No segundo passo de validação, o núcleo IP validado foi a sonda de rede e sua integração com o canal virtual de saída responsável pelo repasse de comandos vindos da rede. Os mesmos comandos aplicados no passo anterior foram reaplicados, ou seja, os de captura de valores observados pela sonda de rede e alteração de parâmetros de observação, tal como canais virtuais a serem analisados e o tamanho da janela de observação. Usou-se um cenário pré-definido de estímulos, com o objetivo de garantir o funcionamento de ambos os núcleos IP.

No terceiro passo, se valida o mecanismo de arbitragem da porta de saída de cada roteador. Desenvolveu-se um ambiente específico para esta etapa de validação, onde se insere pacotes para transmissão via porta de saída. Decisões de tamanho de pacotes e canal virtual de saída são feitas aleatoriamente em tempo de simulação. Para automatizar a etapa de validação definem-se dois campos no corpo do pacote: *(i)* a porta hipotética de entrada, *(ii)* o canal virtual a ser utilizado na saída. Após a captura de um pacote completo na saída, módulos consumidores analisam estes campos e verificam se o pacote transmitido pela porta de saída condiz com o esperado.

Do quarto ao décimo segundo passo validam-se os nove diferentes tipos de roteadores da infraestrutura de comunicação, que emprega topologia malha 2D. Estes roteadores são diferenciados pelo número e/ou pela posição das portas de entrada e saída. Desenvolveu-se um ambiente de simulação SystemC com implementação similar a do terceiro passo. Desenvolveu-se um módulo produtor, responsável pela injeção de pacotes nos roteadores, e um módulo consumidor, responsável pela coleta e análise dos pacotes. Define-se uma ordem de inserção de pacotes, percorrendo todas as portas de entrada e todos os canais virtuais de cada roteador. O corpo dos pacotes inclui três campos que dão: a porta origem da comunicação, o canal virtual empregado na transmissão e o roteador destino. Estes campos permitem analisar se a entrega ocorre conforme planejado.

O décimo terceiro passo corresponde à validação da rede em chip como um todo, mas sem incluir ainda interfaces de rede. Assume-se uma rede de topologia malha 2D de dimensões 5x5 (25 roteadores). Usa-se um ambiente similar àquele empregado na validação dos roteadores, ou seja, um produtor inserindo pacotes em série para todos os possíveis destinos da rede em cada um dos canais virtuais. Ao todo, inserem-se dois mil e quatrocentos pacotes. Adicionalmente, emprega-se o mesmo esquema de transporte de informação de controle no pacote (origem, canal virtual empregado na transmissão e roteador destino). Módulos consumidores em cada um dos roteadores da rede decodificam e processam esta informação. Esta etapa garante a funcionalidade completa da rede em chip.

### 6.2.2 VALIDAÇÃO DA INTERFACE DE REDE

Para a validação da interface de rede, foram realizados seis passos. No primeiro, validou-se o núcleo IP NI\_Sender, responsável pela recepção de comandos e dados vindos do PE conectado a ele e a transformação destes em requisições internas à interface de rede (e. g. abertura/fechamento de sessão) e transmissão de pacotes. A validação deste módulo deu-se em quatro subetapas, onde se explora gradualmente os serviços de transmissão (com e sem fragmentação de pacotes), geração automática de caminhos baseado no algoritmo de roteamento XY, abertura e fechamento de sessão. Um ambiente SystemC gera estímulos e captura a funcionalidade do núcleo IP.

No segundo passo valida-se os barramentos internos da interface de rede, que permitem compartilhar (i) a porta de entrada da rede em chip, (ii) a memória interna da interface de rede e (iii) a comunicação com os monitores mestre e escravo. No terceiro passo valida-se o monitor escravo da interface de rede. Nesta etapa se validam os mecanismos de abertura e fechamento de sessão, e a mecânica de observação de serviços contratados e as ações associadas quando da necessidade de uma alteração de rotas. Se valida ainda a política de definição de uma nova rota, segundo os critérios definidos no Capítulo 5, avaliando-se a carga de utilização dos canais que compõem um caminho.

No quarto passo combina-se pedidos de reação oriundos do monitor escravo (que representa o destino de uma comunicação) com as possíveis reações do monitor mestre (que representa a origem de uma comunicação). Cria-se situações de congestionamento e de baixa injeção de da-

dos na rede. Aqui, o objetivo é validar a interoperabilidade entre os núcleos IP responsáveis pelo disparo de ações de busca de novos caminhos de comunicação. O passo final da etapa de validação da interface de rede ocorre com a integração de todos os núcleos IP que o compõem e a criação de um ambiente misto, onde se incluem módulos para geração de estímulos e abstração da rede de interconexão.

### 6.2.3 VALIDAÇÃO COMPLETA DA MONOC

Para a validação completa da MoNoC descreveram-se quatro cenários de tráfego distintos, sendo os três primeiros estáticos e o último dinâmico. Para o primeiro cenário de tráfego empregou-se distribuição espacial *todos para todos* como característica de comunicação e utilizou-se apenas a contratação do serviço de transmissão, disponível nas interfaces da MoNoC. O objetivo é demonstrar a interoperabilidade entre a interface de rede e a infraestrutura de comunicação. O segundo cenário de tráfego explorou a distribuição espacial ilustrada na Figura 6.10(a), onde o par comunicante encontra-se em extremidades diagonalmente opostas da rede e a rota inicial de comunicação entre eles é parcialmente utilizada por outros pares comunicantes, o que produz uma interferência potencial na comunicação. O objetivo deste cenário de tráfego é explorar, de forma simples, a adaptabilidade da MoNoC. No terceiro cenário de tráfego definiu-se quatro pares comunicantes fixos, dispostos conforme ilustra a Figura 6.10(b). Para explorar o poder de adaptação, os demais pontos da rede que não estão definidos como fixos, injetam pacotes aleatoriamente para outros destinos, causando uma interferência descontrolada na comunicação. O objetivo deste cenário de tráfego é validar a contratação de serviços de taxa mínima de transmissão para múltiplos pares origem e destino operando simultaneamente. O quarto cenário de tráfego teve por característica a definição dos pares comunicantes ocorrendo de forma dinâmica e periódica em tempo de simulação. Para cada par foi definida uma taxa fixa de transmissão e a quantidade de pacotes a enviar. Depois de enviada a quantidade prevista de pacotes entre um par comunicante, sorteia-se aleatoriamente um novo destino, e define-se a taxa de comunicação e a quantidade de pacotes a enviar.

Empregou-se um ambiente SystemC similar àquele utilizado durante a fase de validação da NoC\_Complementar (ver Figura 6.3). A principal mudança com relação àquele ambiente está na forma de geração de estímulos, onde não se faz mais necessário um conjunto de arquivos de entrada. Aqui o comportamento de cada origem de injeção de pacotes pode ser descrito em SystemC

dentro do próprio ambiente de validação. A vantagem desta característica está na eliminação do passo de criação e armazenamento de arquivos que definem os cenários de tráfego. Apesar disto, foi adicionada à ferramenta Atlas a possibilidade de geração de cenários de tráfego para os diferentes níveis de prioridade na transmissão de pacotes disponíveis na MoNoC e a possibilidade de leitura destes arquivos a partir do ambiente empregado aqui.

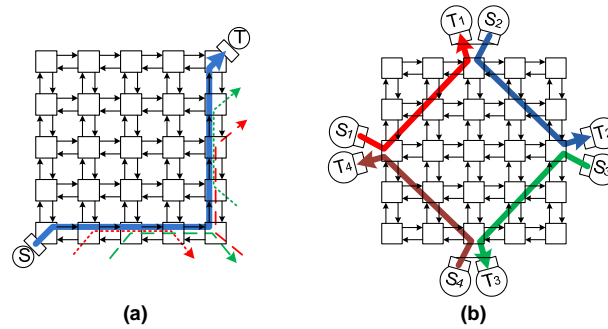


Figura 6.10 – Disposição dos pares comunicantes para dois cenários de tráfego utilizados na validação da MoNoC. Em (a), um par comunicante opera uma comunicação que sofre interferência de tráfegos de ruído concorrentes. Em (b) quatro pares origem ( $S_i$ ) destino ( $T_i$ ) estaticamente definidos se comunicam sob interferência de tráfegos de ruído aleatórios de todos os demais pontos da rede.

Realizou-se aproximadamente uma centena de simulações, onde se variou características de (i) cenários de tráfego e (ii) arquiteturas de comunicação. No primeiro caso, variou-se a quantidade de pacotes e a frequência de inserção destes na rede. Para a segunda característica, variou-se o tamanho das filas de armazenamento. Para todos os casos, a arquitetura de comunicação teve sua funcionalidade validada. Tal validação levou em conta a quantidade de pacotes inseridos e consumidos da rede, bem como a garantia de entrega dos pacotes no destino correto. Para tanto, o ambiente de validação controla, através de sinalização, o fim de transmissão de pacotes pela rede e adicionalmente valida o destino do pacote. Para tanto se inclui em cada pacote dois campos, contendo a identificação do destino e o canal virtual no qual o pacote foi originalmente inserido. Estes campos são automaticamente avaliados e validados no destino. Em caso de não conformidade, a simulação encerra-se com aviso de erro.

#### 6.2.4 RESULTADOS OBTIDOS PARA CENÁRIOS ESTÁTICOS DE TRÁFEGO

Esta Seção apresenta os resultados obtidos para o cenário destacado na Figura 6.10(a). Conforme observado no Capítulo 4, o tamanho da fila de armazenamento contribui para o controle da ocorrência de congestionamentos na rede. O objetivo é avaliar o poder de decisão em situações de carga acentuada na infraestrutura de comunicação. Emprega-se aqui uma versão da MoNoC com



filas de armazenamento de tamanho igual a 4 *flits*. Algumas das configurações da MoNoC relevantes nesta etapa são: dimensões 5x5, algoritmos complementares *West first mínimo* e *East first mínimo*. As demais características são as apresentadas na Seção 5.2, onde se discute, por exemplo, o mecanismo de arbitragem e o uso de canais virtuais.

Denomina-se o cenário de tráfego ilustrado na Figura 6.10(a) de *interferência*. Ele foi descrito em quatro formatos diferentes, conforme a variação de sua taxa de transmissão de pacotes entre cada par, incluindo versões: (i) de 10% a 20%, (ii) de 20% a 30%, (iii) de 30% a 40% e (iv) de 40% a 50% da capacidade de transmissão de um canal. Tais taxas de comunicação são aquelas aplicadas entre os pares comunicantes que estão no caminho do par sob análise, conforme destaca a Figura 6.10(a). A influência na latência de comunicação do par sob análise é apresentada na Figura 6.11.

Os resultados apresentados na Figura 6.11 ilustram uma situação onde nenhuma adaptação é executada, tendo sido utilizado o algoritmo XY para o roteamento dos pacotes entre os pares origem e destino. Conforme se pode observar, mesmo quando a taxa de comunicação entre os diferentes pares é baixa, a interferência causada na latência de comunicação é notória. Pode-se observar na Figura 6.11(a) que a latência de rede mantém-se baixa, porém o congestionamento acarreta atraso superior a quatro mil ciclos de relógio.

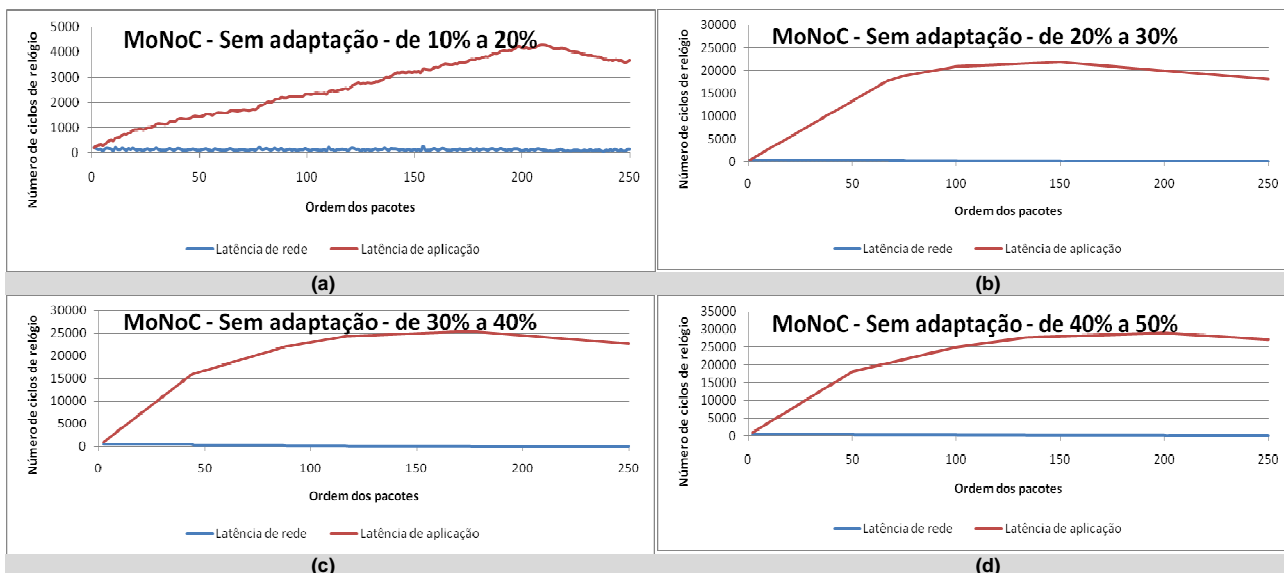


Figura 6.11 – Latências obtidas para o cenário de tráfego *Interferência* quando não é realizada adaptação nas rotas de comunicação.

Para as demais taxas apresentadas na Figura 6.11(b,c,d), a latência de aplicação é superior a

vinete mil ciclos de relógio. Tomando-se o mesmo cenário de tráfego apresentado na Figura 6.11, explorou-se a funcionalidade de adaptação da MoNoC. A Figura 6.12 detalha as latências alcançadas para o cenário de 10% a 20% da taxa de comunicação, similar àquela apresentada na Figura anterior. O presente documento omite os resultados para as demais taxas de comunicação, visto que os momentos de adaptação e os valores de latência ficaram muito próximos do mostrado na Figura 6.12.

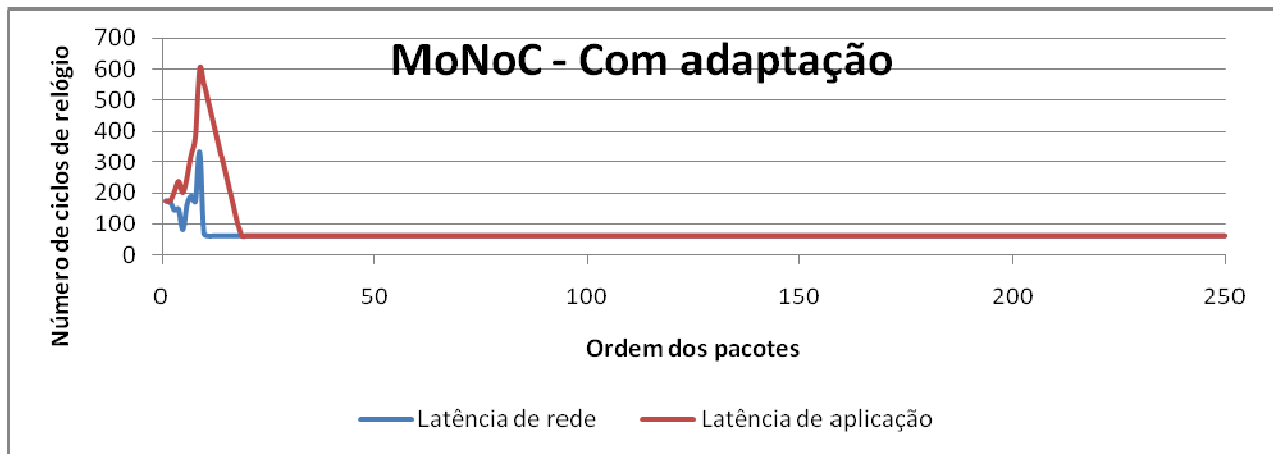


Figura 6.12 – Latências obtidas para o cenário de tráfego *interferência* quando empregada a funcionalidade de adaptação de rotas na MoNoC.

Observando-se a Figura 6.12, pode-se notar um pico de latência, tanto de rede quanto de aplicação. Tal pico é causado pelo processo de adaptação, que envia pacotes de maior prioridade de transmissão pela rede, causando um congestionamento momentâneo na transferência de pacotes de mais baixa prioridade. A Tabela 6.4 apresenta os valores de latência de rede e de aplicação dos doze primeiros pacotes transmitidos entre a origem e o destino sob análise. Pode-se observar que o pico de latência ocorreu durante a transmissão do nono pacote, momento em que uma nova rota estava sendo escolhida. A partir do décimo pacote, coluna destacada na Tabela 6.4, reduz-se a latência de rede em mais de 60% da média anterior, alcançando latências baixas, garantidas pelo uso de rotas livres de concorrência. Pode-se também observar a partir do décimo pacote, uma queda gradual da latência de aplicação, que se torna equivalente à latência de rede a partir do décimo nono pacote.

Tabela 6.4 – Valores de latências obtidos para o cenário de tráfego *interferência* quando adaptada a rota de comunicação.

Identificação do Pacote	1	2	...	7	8	9	10	11	12	...	17	18	19	20	...
Latência de rede	175	173	...	191	174	334	68	60	60	...	60	60	60	60	...
Latência de aplicação	176	175	...	325	383	601	553	497	441	...	161	105	60	60	...

Para que a adaptação de rotas ocorra, faz-se necessária a abertura de uma sessão entre a origem de injeção de dados e o destino, característica detalhada no Capítulo 5. Os custos de tempo da abertura de uma sessão, do tempo de reação e do fechamento de uma sessão, capturados durante a simulação do cenário de tráfego *interferência*, são apresentados na Figura 6.13. Estes tempos são precisos, pois foram obtidos a partir de análise de formas de onda da simulação RTL. O tempo apresentado para abertura de sessão (78 ciclos de relógio) compreende desde o momento em que o elemento de processamento conectado à NI origem requisita a abertura de sessão (no instante 380 nanosegundos) até o momento em que tal requisição é atendida pela NI destino (no instante 1.940 nanosegundos).

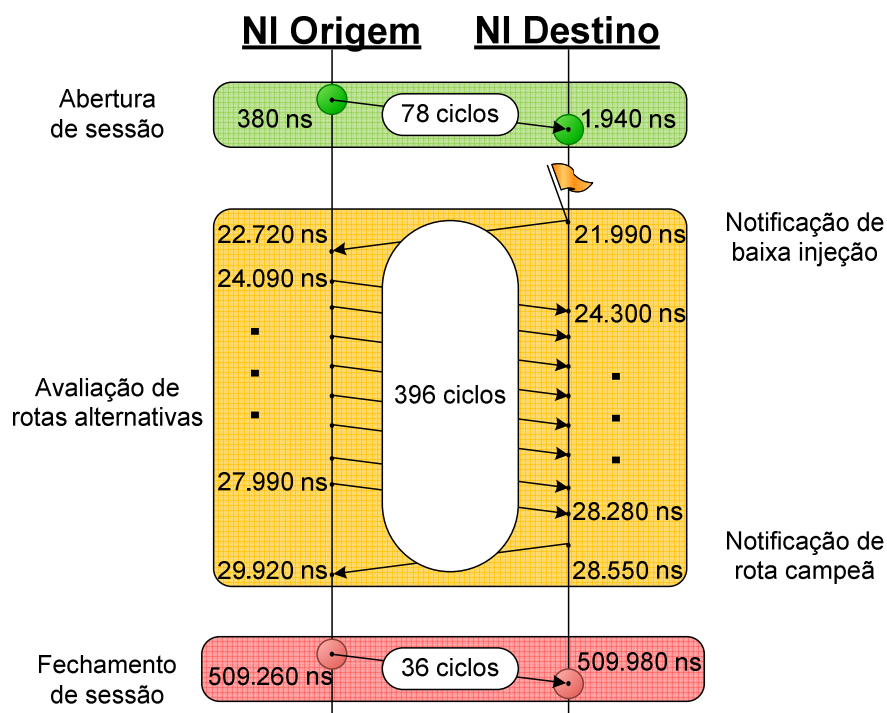


Figura 6.13 – Custos de tempo de abertura/fechamento de sessão e adaptação de rota.

A fase de adaptação da rota mostrada na Figura tem início quando a NI destino detecta baixa injeção de dados (no instante 21.990 nanosegundos) e encaminha uma notificação à NI origem, informando descumprimento da taxa contratada, a qual é interpretada (no instante 22.720 nanosegundos), identificando-se uma possível ocorrência de congestionamento. Sabendo-se que a taxa

de 20% da capacidade de transmissão do canal, ou seja, 160 Mbps deveria ser atendida, define-se 8 (oito) rotas possíveis entre a origem e o destino da comunicação, que são investigadas na busca da menos congestionada. Os valores de ocupação de cada rota são apresentados na Tabela 6.5. A NI destino escolhe a nova rota (no instante 28.280 nanosegundos) e a origem de transmissão assume esta nova rota (no instante 29.920 nanosegundos). A Tabela 6.5 é apresentada a rota escolhida (rota 0x15) e destaca suas características. Ao todo, da detecção de queda na taxa de transmissão até a definição de uma nova rota, consumiu-se cerca de 400 ciclos de relógio.

Tabela 6.5 – Carga de ocupação e de pico das rotas avaliadas durante a adaptação de rota.

<b>Identificação da rota</b>	<b>Carga total (média)</b>	<b>Pico de ocupação</b>
<b>0x06</b>	69	96
<b>0x09</b>	30	88
<b>0x0C</b>	21	88
<b>0x0F</b>	12	82
<b>0x12</b>	6	29
<b>0x15</b>	2	11
<b>0x18</b>	8	55
<b>0x1B</b>	12	65

\* Valores de carga e pico dados em número de ciclos de relógio

Os resultados de otimização de desempenho através da adequação da rota de comunicação entre um dado par comunicante mostrou bons resultados, conforme se pode notar pelos gráficos de latência, o custo em tempo para a adaptação da rota e sua escolha. Todavia, cenários estáticos de tráfego não são uma boa justificativa para o emprego de tal abordagem, pois propostas de solução em tempo de projeto como as do Capítulo 3 podem gerar resultados melhores. Na Seção a seguir explora-se o emprego de adaptação de rotas quando a ocupação da rede varia em tempo de execução.

## 6.2.5 RESULTADOS OBTIDOS PARA CENÁRIOS DINÂMICOS DE TRÁFEGO

Esta Seção apresenta os resultados obtidos para o cenário de tráfego ilustrado na Figura 6.14. No cenário de tráfego sugerido utiliza-se o par em destaque como base para a avaliação dos valores de latência, enquanto os demais pontos da rede injetam pacotes para destinos aleatórios definidos em tempo de execução, seguindo critérios que são mais bem descritos adiante. Escolheu-se a distribuição espacial do par comunicante em destaque, pois com frequência o centro de redes de topologia malha é local de maior concentração de fluxo de pacotes quando comparado às bordas. Isto facilita que interferências na comunicação entre o par origem/destino em destaque sejam

causadas pelos demais pares comunicantes definidos em tempo de execução. Isto contribui para criar situações onde se quer avaliar o comportamento da MoNoC sob carga dinâmica de comunicação. No restante desta Seção, a interferência causada pela comunicação dos demais pares que não aquele destacado na Figura 6.14 será referida como *ruído*.

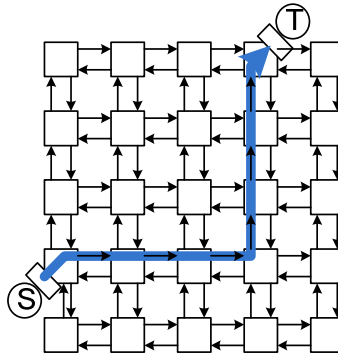


Figura 6.14 – Disposição do par origem destino observado na MoNoC sob carga dinâmica de comunicação.

As configurações assumidas para a infraestrutura de comunicação MoNoC mantiveram-se inalteradas com relação àquelas empregadas na Seção 6.2.4, ou seja, tamanho da fila de armazenamento igual a 4 *flits*, dimensões da rede 5x5 e algoritmos de roteamento complementares *West first mínimo* e *East first mínimo*.

Realizou-se 40 (quarenta) simulações distintas onde foram variadas: (i) a taxa de transmissão entre pares comunicantes, a exceção do par destacado na Figura 6.14, onde se manteve taxa constante de 20%, (ii) a quantidade de pacotes que uma determinada origem deve enviar para um destino antes de definir um novo destino e (iii) a adaptação ou não de rotas. As duas primeiras características tinham por objetivo definir critérios de carga dinâmica de comunicação. Para cada par gerando ruído a taxa de transmissão é variada aleatoriamente pela origem, selecionando um de um conjunto de intervalos fixos: 10% a 20%, 20% a 30%, 30% a 40% e 40% a 50%. Para cada escolha de taxa e destino, fazem-se seis rodadas de simulação com as seguintes quantidades de pacotes (antes de cada reconfiguração de taxa e destino): 1, 15, 30, 50, 60 e 100. A variação da quantidade de pacotes se justifica, pois espera-se que quanto menor a quantidade de pacotes a serem enviados antes de definir um novo destino de comunicação, menor a qualidade da adaptação da rota, visto que uma decisão de um bom caminho pode não persistir.

Para os pares comunicantes que geram ruído a rota de comunicação é definida usando o al-

goritmo XY, independente da exploração da adaptação de rotas. Para garantir a coerência na comparação dos resultados de simulação quando do emprego ou não da adaptação de rotas, adequou-se o ambiente de simulação para que este salve a sequência de destinos e as taxas assumidas em cada simulação, garantindo assim cenários de tráfego equivalentes para simulações distintas.

Para pares origem destino distantes, o crescimento super-exponencial das rotas alternativas com os algoritmos adaptativos aqui empregados inviabiliza a exploração completa de rotas alternativas. Assim, objetivando reduzir o número de pacotes de controle circulando na rede para exploração de rotas limitou-se a quantidade de alternativas. O objetivo é influenciar o mínimo possível o tráfego dos demais pacotes de mais baixa prioridade, permitindo ao mesmo tempo identificar um canal com maior pico de utilização. Nos experimentos descritos aqui, usou-se uma heurística que consiste em definir rotas com um algoritmo adaptativo onde o número máximo de curvas é restrito a duas (2). Com isto pode-se reduzir a quantidade de pacotes de controle que varia de acordo com a distância entre o par origem destino e é expressa pela Equação 6.1. Por exemplo, assumindo-se um algoritmo adaptativo *West first mínimo* e a distribuição espacial do par destacado na Figura 6.14, a quantidade de rotas possíveis é igual a vinte, número este calculado a partir da Equação 3.2. No experimento o número de rotas alternativas foi de apenas seis.

$$\text{Equação 6.1} \quad N_{\text{RotasExpbradas}} = ((\Delta x - 1) + (\Delta y - 1))$$

A Figura 6.15 apresenta os resultados alcançados para um cenário de tráfego onde o ruído aplicado tem características de taxa de injeção de tráfego entre 20% e 30% e a definição de um novo destino se dá após o envio de 50 pacotes. Na Figura 6.15(a), apresenta-se os resultados de latência obtidos para o caso onde não foi explorada a funcionalidade de adaptação. Conforme se pode observa, as latências de rede e de aplicação mantiveram-se baixas até o centésimo pacote, quando a ocorrência de congestionamento elevou a latência de aplicação a aproximadamente 12.000 ciclos. Já quando aplicado o poder de adaptação, pode-se observar claramente três picos de latência de aplicação sendo reduzidos, chegando a aproximadamente 1.200 ciclos de relógio.

A Figura 6.16 apresenta resultados alcançados para um cenário de tráfego onde o ruído aplicado possui características de taxa de injeção de tráfego entre 30% e 40% e a definição de um novo destino se dá após o envio de 100 pacotes. Na Figura 6.16 (a), apresenta-se os resultados de

latência sem adaptação. Conforme se pode observar, as latências de rede e de aplicação são afetadas significativamente pelo ruído desde o início da simulação, atingindo picos de aproximadamente 10.000 ciclos para a latência de aplicação. Ao aplicar-se adaptação (Figura 6.16(b)), pode-se observar claramente um pico inicial de latência causado pela busca de uma melhor rota, mas que não superou 900 ciclos de relógio. Após a definição da nova rota, nenhuma nova adaptação foi necessária, garantindo assim latências de rede e de aplicação bastante próximas.

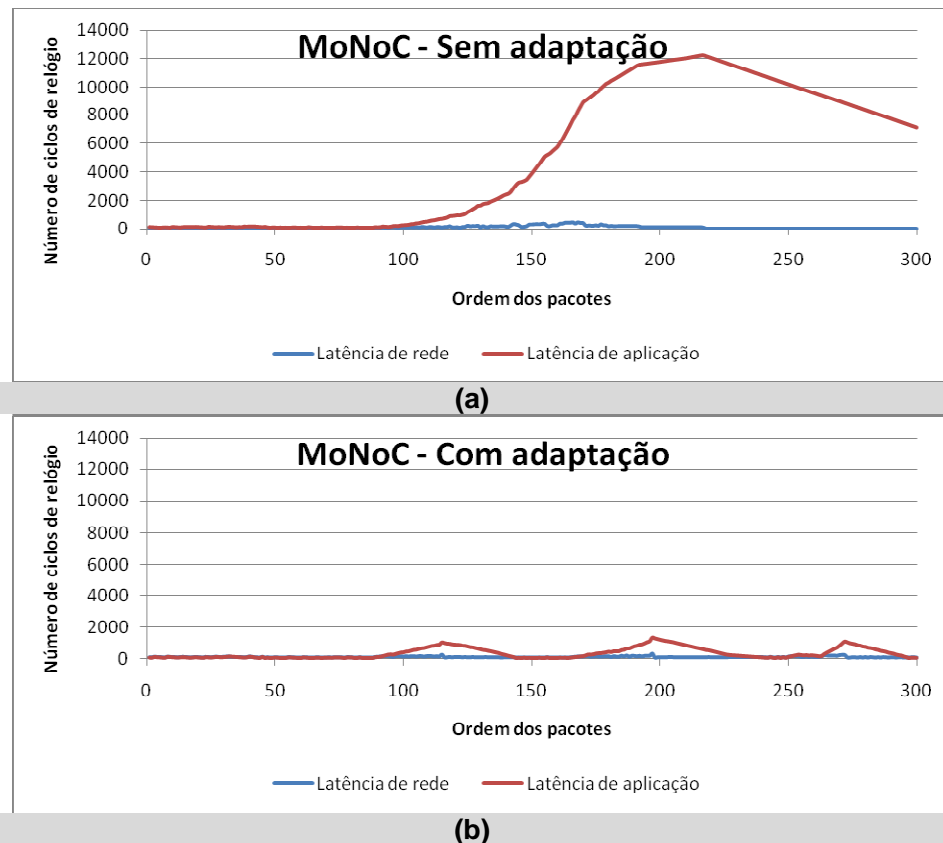


Figura 6.15 – Resultados de simulação para um cenário dinâmico de tráfego usando a MONOC. A mudança de pares comunicantes ocorre após 50 pacotes enviados para um mesmo destino, respeitando uma taxa de injeção de dados entre 20% e 30% da capacidade de transmissão de um canal. Em (a) tem-se os resultados de latência quando a adaptação não é explorada; em (b) aparecem os resultados com adaptação de rotas.

Dos casos avaliados, observou-se que o emprego do mecanismo de adaptação proposto permitiu melhores resultados principalmente quando as taxas de comunicação exploradas foram superiores à faixa entre 10% e 20% da capacidade de transmissão de um canal. Isto se justifica pelas características da infraestrutura de comunicação, que minimiza situações de sobrecarga a partir de seu mecanismo de arbitragem interna de roteamento e de priorização de canais virtuais para tais taxas de comunicação. Adicionalmente, a adaptação surtiu maior efeito quando a variação dos pares comunicantes era igual ou superior a 30 pacotes enviados antes de definir um novo destino

para cada par. Isto se deve principalmente pela instabilidade de cargas de comunicação nos canais de rede causada pela mudança frequente de pares de comunicação, observado quando o número de pacotes é inferior a 30.

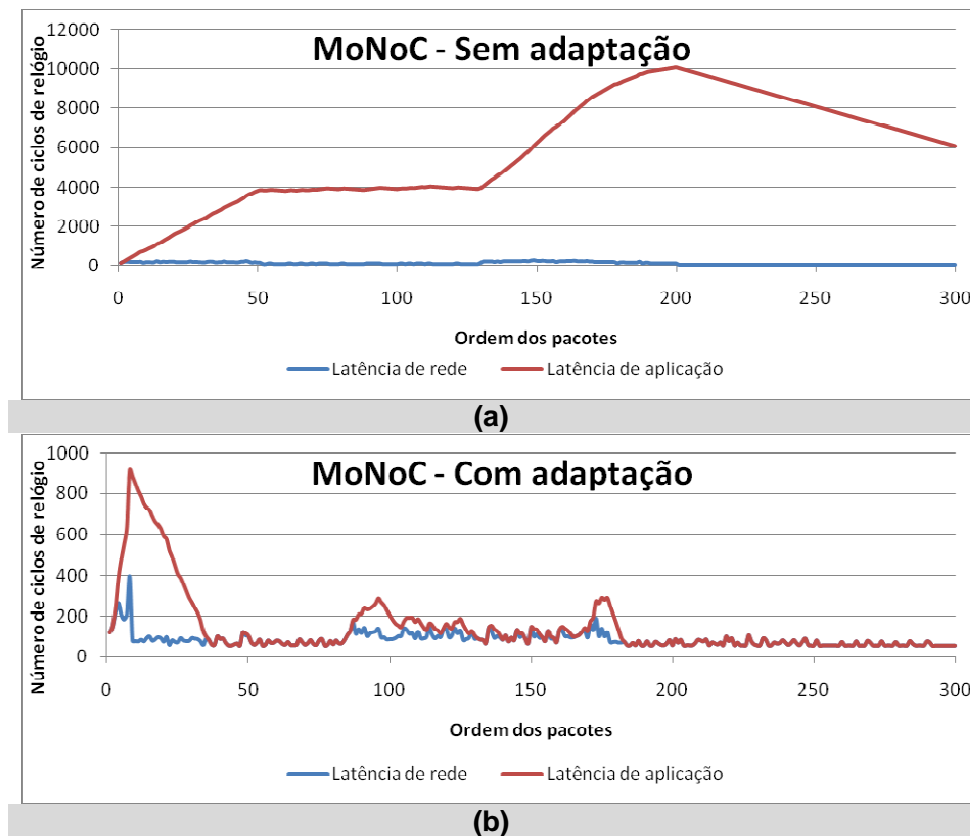


Figura 6.16 – Cenário dinâmico de tráfego. Mudança de pares comunicantes ocorrendo após 100 pacotes enviados, respeitando uma taxa de injeção de dados entre 30% e 40% da capacidade de transmissão de um canal. Em (a) os resultados de latência quando a adaptação não é aplicada. Em (b) nota-se a redução da latência de aplicação quando do emprego da adaptação.

### 6.3 CONSIDERAÇÕES FINAIS

No presente Capítulo buscou-se validar o emprego de arquiteturas que utilizam simultaneamente algoritmos de roteamento complementares em uma mesma arquitetura de comunicação, bem como demonstrar a funcionalidade da rede em chip MoNoC. Adicionalmente, foram conduzidas simulações objetivando a captura de resultados de latência para todas as infraestruturas de comunicação validadas. Para a realização das validações, utilizou-se a ferramenta Atlas para a geração de cenários de tráfego estáticos e criaram-se ambientes de validação para permitir a caracterização de cenários de tráfego dinâmicos.

A validação inicial do emprego de algoritmos de roteamento complementares em uma



mesma arquitetura de comunicação deu-se a partir de uma NoC desenvolvida especialmente para tal tarefa (a NoC Complementar) onde o algoritmo de roteamento é distribuído. Resultados de latência de rede e de aplicação foram capturados para diferentes cenários de tráfego. A latência de rede mostrou-se menor em praticamente todos os experimentos, garantindo de 10% a 20% de redução em relação às NoCs com algoritmos determinístico e adaptativos. A latência de aplicação foi o valor de maior impacto nas simulações, demonstrando-se diminuições significativas na maioria dos casos.

Por fim, os experimentos conduzidos para a MoNoC exploram o potencial de adaptação da arquitetura frente a situações estáticas e dinâmicas de carga de comunicação. Nas simulações onde com cenários estáticos de tráfego, a MoNoC garantiu rota ótima assim que detectada a queda da taxa contratada, tendo o tempo total de reação para o cenário de tráfego experimentado sido igual a 396 ciclos de relógio. Nas simulações onde se explora cenários dinâmicos de tráfego, a MoNoC garantiu reduções de latência de aplicação entre 10 e 50 vezes para a transmissão de pacotes nos experimentos apresentados.

## 7. CONCLUSÃO E TRABALHOS FUTUROS

O presente Capítulo apresenta uma relação das contribuições do trabalho, na Seção 7.1. Em seguida, apresenta-se um conjunto de conclusões (Seção 7.2), e algumas sugestões de trabalhos futuros (Seção 7.3).

### 7.1 CONTRIBUIÇÕES DO TRABALHO

Dentre as contribuições do trabalho pode-se citar as seguintes como principais:

**Infraestrutura de comunicação visando mapeamento** – A primeira contribuição do trabalho consiste na elaboração de uma infraestrutura de comunicação, chamada Hermes-SR, voltada para o mapeamento de rotas, definidas em tempo de projeto. Tal arquitetura teve por objetivo a otimização de mecanismos empregados na NoC Hermes, tal como o algoritmo de arbitragem, e adicionalmente permitir a exploração de roteamento na origem para uma infraestrutura até então baseada unicamente na implementação de algoritmos de roteamento distribuídos. Seu emprego exigiu a adaptação da estrutura dos pacotes de rede, que na Hermes-SR contêm as direções a serem tomadas desde a origem até o destino.

**Heurística de mapeamento de rotas** – A segunda contribuição do trabalho consiste na implementação de um ambiente que tem por objetivo a coleta de informações de características da comunicação de uma aplicação já mapeada na rede e a exploração das rotas mais adequadas para a eliminação de canais de maior carga da NoC. Para a busca da combinação de todas as rotas que levam ao uso otimizado da infraestrutura de comunicação foram explorados os algoritmos *simulated annealing* e heurístico. O primeiro garante uma solução otimizada, mas não necessariamente a melhor, a qual somente pode ser encontrada a partir de uma busca exaustiva. Esta última torna-se proibitiva dada a quantidade de combinações possíveis mesmo para NoCs de pequenas dimensões. A segunda permitiu encontrar soluções tão boas quanto as encontradas com o *simulated annealing*, porém em menos tempo.

**Infraestrutura de comunicação visando adaptação** – A principal contribuição do trabalho é a da implementação da infraestrutura de comunicação visando adaptações de rota, chamada MoNoC. O objetivo é propor recursos que permitam adaptação de rotas em infraestruturas dotadas de roteamento na origem. Isto ocorre a partir de uma visão global das cargas de ocupação dos canais de comunicação entre origens e destinos de comunicação sem a necessidade de um recurso de gerenciamento centralizado. Isto é possível através da implementação de interfaces de rede que disponibilizam um conjunto de serviços contratáveis externamente à rede.

**Emprego de sondas de rede** – O emprego de sondas de redes ligadas aos canais da MoNoC foi outra contribuição do trabalho. Esses módulos, inseridos nas portas de saída dos roteadores da MoNoC, permitem a medição da ocupação de cada canal da NoC, em tempo de execução. Os resultados coletados por tais sondas permitem avaliações relacionadas à ocupação dos canais, e ainda a medição do nível de congestionamentos de uma dada rota.

**Infraestruturas de comunicação adicionais** – No decorrer dos trabalhos necessitou-se desenvolver um conjunto de infraestruturas para a validação e comparação de resultados. Dentre as infraestruturas adicionais desenvolvidas para o presente trabalho destaca-se (a) a NoC Hermes com roteamento na origem e empregando algoritmo de arbitragem *round robin* e (b) as NoCs complementares, desenvolvidas a partir da Hermes original, porém com dois canais virtuais, onde combinou-se os algoritmos *West first mínimo* com *East first mínimo*, *North last mínimo* com *South last mínimo* e *negative first mínimo* com *positive first mínimo*.

**Publicações** – O desenvolvimento dos trabalhos realizados durante o período do doutorado contribuiu para as publicações listadas abaixo. Os assuntos trabalhados durante o doutorado contribuíram para a definição do tema da presente tese.

MARCON, César Augusto Missio, MORENO, Edson Ifarraguirre, CALAZANS, N. L. V., MORAES, Fernando Gehm. **Evaluation of Algorithms for Low Energy Mapping onto NoCs**. In: 20th Symposium on Circuits and Systems and Systems - ISCAS 2007, New Orleans. 2007. pp. 389-392.

MORENO, Edson Ifarraguirre, POPOVICI, Katalin, CALAZANS, Ney Laert Vilar, JERRAYA, Ahmed Amine. **Integrating Abstract NoC Models within MPSoC Design**. In: 18th Annual IEEE/IFIP

International Symposium on Rapid Systems Prototyping - RSP 2008, Monterrey, June, 2008. pp. 65-71.

MARCON, César Augusto Missio, MORENO, Edson Ifarraguirre, CALAZANS, Ney Laert Vilar, MORAES, Fernando Gehm. **Comparison of NoC Mapping Algorithms Targeting Low Energy Consumption**. IET Computers & Digital Techniques, vol. 2, no. 6. November, 2008. pp. 471-482.

## 7.2 CONCLUSÕES

A evolução das tecnologias submicrônicas de implementação de CIs e de técnicas de projeto de CIs têm contribuído para o surgimento de plataformas multiprocessadas em um único chip. Visto como uma tendência até pouco tempo, o lançamento de produtos comerciais com grande número de processadores (e. g. Tile64 [TIL09]) tem demonstrado a viabilidade prática de tais sistemas. Todavia, o aumento do número de elementos computacionais em uma mesma plataforma exige a presença de infraestruturas de comunicação mais eficientes. NoCs são vistas como tendência, pois entre outras vantagens garantem melhor escalabilidade quando comparadas a infraestruturas de comunicação como barramentos e conexões dedicadas. Mesmo agregando eficiência à plataforma, o número crescente de tais elementos computacionais pode levar a cenários de tráfego que inviabilizem o atendimento de restrições de qualidade de serviço na comunicação, degradando a eficiência total do sistema. No presente documento exploram-se mecanismos que contribuam para a melhor utilização dos recursos providos por redes em chip.

Para o desenvolvimento deste trabalho, realizou-se a revisão de trabalhos tais como a de proposta de NoCs, de topologias regulares e irregulares e mecanismos de controle de congestionamento, variando entre controle de pontos quentes, negociações fim a fim, adaptações na rede, mapeamento de aplicações e definições de rotas. Com base nos trabalhos revisados propuseram-se meios de controle de fluxo de comunicação no qual o conhecimento do estado de ocupação dos recursos de comunicação não se restringe a um roteador. Durante o desenvolvimento do trabalho contemplaram-se duas propostas, a primeira para decisões tomadas em tempo de projeto, através do mapeamento de rotas, e a segunda para decisões tomadas em tempo de execução, através da adaptação de rotas.

Para dar suporte ao mapeamento de rotas, idealizou-se inicialmente uma infraestrutura de

comunicação com roteamento flexível, mas controlável, chamada NoC Hermes-SR, uma rede em chip baseada nas especificações da NoC Hermes. A NoC Hermes-SR foi implementada com duas características diferenciais quando comparada à NoC Hermes: roteamento na origem e um esquema de arbitragem mais previsível, este último implementando o algoritmo *First Come First Served*. Para a definição das rotas entre pares origem/destino que se comunicam com o uso da NoC Hermes-SR, explora-se diferentes algoritmos de roteamento, utilizando-se uma função custo que leva em consideração a carga dos canais. Realiza-se a exploração das rotas a partir de uma ferramenta desenvolvida no âmbito do trabalho, que emprega algoritmos de busca tal como *simulated annealing* e um heurístico. Resultados apontam para uma diminuição da latência média dos pacotes quando adotada a NoC Hermes-SR ao invés da NoC Hermes, devendo-se este resultado principalmente ao algoritmo de arbitragem adotado. Adicionalmente, a adoção de algoritmos adaptativos para a definição de rotas entre os pares origem/destino mostrou a eficiência em controle de situações de congestionamento e a diminuição de pontos quentes na rede. Comparada a uma rede em chip implementada com algoritmos de roteamento adaptativos, os resultados de latência média obtidos com o mapeamento de rotas mostrou-se mais eficaz na maioria dos casos.

Os resultados obtidos com o mapeamento de rotas motivaram a implementação de uma solução que permite a adaptação de rotas em tempo de execução em arquiteturas de comunicação que empregam roteamento na origem. Nesta solução, procura-se dinamicamente propor caminhos alternativos de comunicação entre pares origem-destino em situações de queda das taxas contratadas de serviço, parâmetro utilizado como variável deflagradora da necessidade de adaptação. Para dar suporte à adaptação de rotas, desenvolveu-se uma infraestrutura de comunicação, cujas características básicas refletiam aquelas assumidas na NoC Hermes-SR. Adicionalmente, um conjunto de características foi aplicado sobre a infraestrutura de comunicação, tal como a implementação de canais virtuais definindo prioridades diferenciadas e um mecanismo de otimização nas portas de saída dos roteadores para a transmissão dos pacotes. A adaptação de rotas apoiou-se em dois outros recursos, quais sejam interfaces de rede e sondas de rede. As interfaces de rede garantem a contratação de serviços e monitoram o atendimento destes enquanto as sondas de rede coletam informações sobre a ocupação nos canais de rede. A infraestrutura de comunicação, juntamente com as interfaces de rede e as sondas de rede são empregadas na construção da rede em chip MoNoC. Uma última característica explorada para a adaptação de rotas diz respeito à exploração de rotas alternativas.

Os algoritmos de roteamento adaptativos explorados nesta tese mostraram-se limitados em algumas situações, devido à natureza dos mesmos e ao posicionamento dos pares comunicantes. Sendo assim, para a MoNoC assumiu-se mais de um tipo de algoritmo de roteamento, o que permitiu que mais de uma rota alternativa pudesse ser definida entre uma dada origem de transmissão e seu destino de comunicação. Experimentos conduzidos ilustraram a viabilidade da coexistência dos algoritmos de roteamento quando empregados em canais virtuais distintos, além de permitir a diminuição das latências de comunicação dada a maior quantidade de rotas e o consequente aumento do número de possibilidades de rotas para desviar de pontos congestionados. Outros dois conjuntos de experimentos permitiram validar o emprego da infraestrutura de comunicação MoNoC sob cenários de tráfego cujo comportamento não se alterava ao longo da simulação, chamado de cenário estático de tráfego, e sob cenários de tráfego cujos pares comunicantes eram definidos em tempo de execução. Em ambos os casos, quando aplicados cenários estáticos e dinâmicos de tráfego, pode-se observar o ganho promovido pela infraestrutura de comunicação, reduzindo significativamente a latência. A Figura 7.1 compara as NoCs Hermes-SR, MoNoC e Hermes quanto ao local de decisão de roteamento, tipo de roteamento e conhecimento do estado de ocupação dos recursos de comunicação das redes em chip.

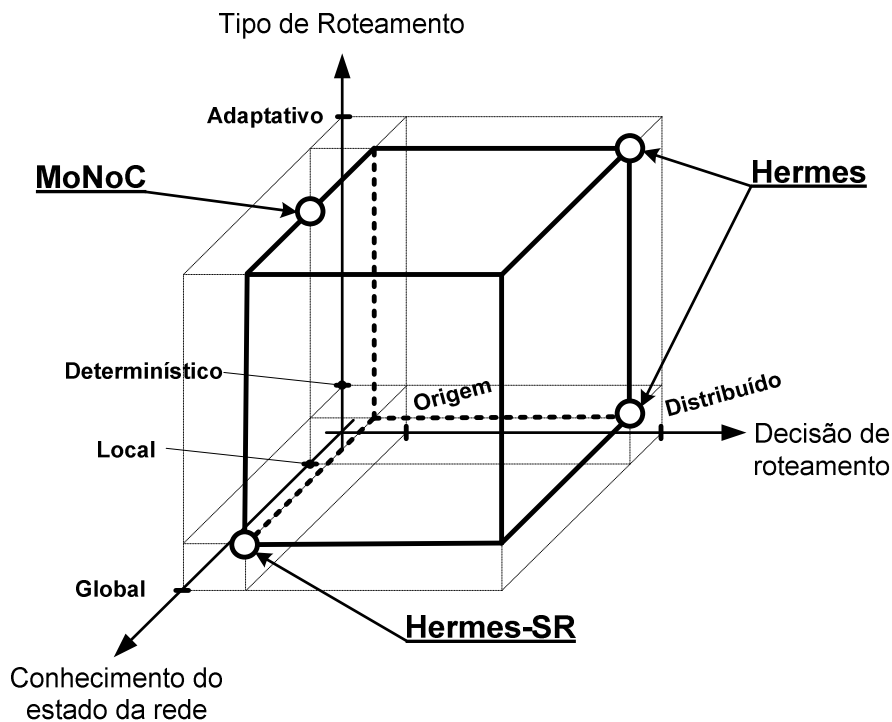


Figura 7.1 – Comparação das NoCs Hermes-SR, MoNoC e Hermes.

Conforme pode-se observar na Figura 7.1, a NoC Hermes é duplamente representada na figura visto que hoje sua implementação abrange algoritmos determinísticos ou adaptativos. A a-

daptação é realizada a partir do processamento de informação estritamente local entre roteadores. A NoC Hermes-SR, comparada à NoC Hermes, emprega determinismo na decisão do roteamento, que é feita em tempo de projeto, empregando roteamento na origem. O conhecimento do estado de ocupação da rede é dito global, visto que neste trabalho explora-se a estimativa de ocupação dos canais da rede para a definição das rotas de comunicação. Por fim, a MoNoC emprega decisão de roteamento similar à Hermes-SR com a diferença de que as mesmas são tomadas em tempo de execução, o que garante a adaptação apontada na Figura 7.1. Tais decisões são tomadas a partir do conhecimento parcial do estado de ocupação da rede, onde se explora apenas os recursos de comunicação intermediários aos roteadores entre a origem e o destino da comunicação, eliminando assim a necessidade do conhecimento do estado de ocupação de toda a rede.

### 7.3 TRABALHOS FUTUROS

Uma primeira sugestão de trabalhos futuros consiste na otimização da infraestrutura de comunicação MoNoC. As otimizações que se sugere referem-se à ocupação de área. A estimativa de consumo de área mostrada no final do Capítulo 5 mostra uma maior ocupação da MoNoC quando comparada com a Hermes-SR. Isto se deve em parte pelo fato da Hermes e a Hermes-SR serem NoCs minimalistas, mas é claro que os recursos adicionais incluídos na infraestrutura MoNoC causam aumento de lógica para efetuar as operações, resultando em aumento de área. Acredita-se que parte desta sobrecarga ocorre devido ao uso de técnicas não necessariamente otimizadas de descrição de *hardware*.

Uma segunda sugestão é a exploração de mapeamento de rotas ou mesmo adaptação de rotas visando a redução do consumo de energia. O ganho de desempenho atribuído a ambas infraestruturas de comunicação propostas, Hermes-SR e MoNoC, por certo reflete positivamente na redução do consumo de energia, visto que um maior esforço é empregado de forma útil, ou seja na efetiva transmissão de dados pela rede com a redução da ocorrência de congestionamentos. Adicionalmente, o emprego de estratégias, tais como uso de *clock gating*, são bastante importantes de serem exploradas neste contexto.

O trabalho realizado no contexto da tese focou principalmente a infraestrutura de comunicação e a avaliação dos ganhos obtidos a partir de cenários sintéticos de tráfego. A infraestrutura de comunicação MoNoC foi desenvolvida com a intenção de trazer para a fronteira da rede em

chip o poder de intervenção no desempenho da rede através de contratos de serviço que podem ser requisitados a partir dos elementos de processamento. Este passo inicial pode vir a ser explorado para a criação de um conjunto de serviços que, incluídos em um sistema operacional, permita a intervenção na adaptação, seja de rotas ou outro critério, por parte de elementos de processamento das características de tráfego, o que define uma terceira sugestão de trabalho futuro.

Uma última sugestão de trabalho futuro é a automatização do processo de elaboração de sistemas multiprocessados baseados em redes em chip. Tal automatização pode explorar otimizações de implementação de redes em chip, eliminando roteadores desnecessários, propondo rotas de comunicação para pares comunicantes de forma a reduzir a sobrecarga de canais e por fim incluir mecanismos de adaptação de rotas em pontos da rede onde se prevê a necessidade de contratação de serviços de transmissão.





## REFERÊNCIAS BIBLIOGRÁFICAS

- [ADD05] Addo-Quaye, C. ***“Thermal-Aware Mapping and Placement for 3-D Designs”***. In: IEEE International SoC Conference (SOCC’05), 2005, pp. 25-28.
- [ALF07] Al Faruque, M.; Ebi, T.; Henkel, J. ***“Run-time Adaptive on-chip Communication Scheme”***. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD’07), 2007, pp. 26-31.
- [ALT09] Altera Corporation. Capturado em: <http://www.altera.com>, Janeiro 2009.
- [ARM09] ARM Ltd. Capturado em: <http://www.arm.com>, Janeiro 2009.
- [BAN01] Banerjee, K.; Souri, S.; Kapur, P.; Saraswat, K. ***“3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration”***. Proceedings of IEEE (Special Issue on Interconnections), vol. 89-5, May 2001, pp. 602-633.
- [BEN01] Benini, L.; Micheli, G. ***“Powering Networks on Chip”***. In: International Symposium on System Synthesis (ISSS’01), 2001, pp. 33-38.
- [BEN02] Benini, L.; Micheli, G. ***“Network on Chips: A New SoC Paradigm”***. IEEE Computer, vol. 35-1, January 2002, pp. 70-78.
- [BER04] Bertozzi, D.; Benini, L. ***“Xpipes: A Network-on-chip Architecture for Gigascale Systems-on-Chip”***. IEEE Circuits and Systems Magazine, vol. 4-2, 2004, pp. 18-31.
- [BHO03] Bhojwani, P.; Mahapatra, R. ***“Interfacing cores with on-chip packet-switched networks”***. In: Proceedings of the International Conference on VLSI Design (VLSID’03), 2003, pp. 382-387.
- [BOL04] Bolotin E.; Cidon, I.; Ginosar, R.; Kolodny, A. ***“QNoC: QoS architecture and design process for network on chip”***. Journal of Systems Architecture: The EUROMICRO Journal, vol. 50-2/3, February 2004, pp. 105-128.
- [BOL07] Bolotin, E.; Cidon, I.; Ginosar, R.; Kolodny, A. ***“Routing Table Minimization for Irregular Mesh NoCs”***. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE’07), 2007, pp. 942-947.

- [CAR07] Carvalho, E.; Calazans, N.; Moraes, F. **"Congestion-Aware Task Mapping in NoC based MPSoCs with Dynamic Workload"**. In: Proceedings of the IEEE Computer Society Annual Symposium on Very Large Scale Integration (ISVLSI'07), 2007, pp. 459-460.
- [CAR08] Carara, E.; Pigatto, D.; Calazans, N.; Moraes, F. **"MOTIM – An industrial Application Using NoCs"**. In: Proceedings of the 21st Symposium on Integrated Circuits and Systems (SBCCI'08), 2008. pp 182-187.
- [CAR09] Carvalho, E. **"Mapeamento Dinâmico de Tarefas em MPSoCs Heterogêneos baseados em NoCs"**. Tese de Doutorado,PPGCC-FACIN-PUCRS, 2009, 168 p.
- [CIO06] Ciordas, C.; Goossens, K.; Basten, T.; Radulescu, A.; Boon, A. **"Transaction Monitoring in Networks on Chip: The On-Chip Run-Time Perspective"**. In: Proceedings of the Symposium on Industrial Embedded Systems (IES'06), 2006, pp. 1-10.
- [CUL99] Culler, D.; Singh, P. **"Parallel Computer Architecture: A Hardware/Software approach"**. Morgan Kaufmann, 1999, 1025 p.
- [DAL01] Dally, W.; Towles, B. **"Route Packets, Not Wires: On Chip Interconnection Networks"**. In: Proceedings of the Annual Conference on Design Automation (DAC'01), 2001, pp. 684-689.
- [DAL03] Dally, W.; Towles, B. **"Principles and Practices of Interconnection Networks"**. São Francisco (EUA):Morgan Kaufmann, 2003, 2nd ed., 577 p.
- [DAL87] Dally, W.; Seitz, C. **"Deadlock-free message routing in multiprocessor interconnection networks"**. IEEE Transactions on Computers, vol. C-36-5, May 1987, pp. 547-553.
- [DAL98] Dally, W.; Poulton, J. **"Digital Systems Engineering"**. Cambridge University Press, 1998, 663 p.
- [DAN06a] Daneshtalab, M.; Afzali-Kusha, A.; Mohammadi, S. **"Minimizing Hot Spots in NoCs through a Dynamic Routing Algorithm based on Input and Output Selections"**. In: International Symposium on System-on-Chip (SOC'06), 2006, pp. 49-52.
- [DAN06b] Daneshtalab, M.; Sobhani, A.; Afzali-Kusha, A.; Fatemi, O.; Navabi, Z. **"NoC Hot Spot minimization Using AntNet Dynamic Routing Algorithm"**. In: IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP'06), 2006, pp. 33-38.
- [DEH05] Dehyadgari, M.; Nickray, M.; Afzali-Kusha, A.; Navabi, Z. **"Evaluation of Pseudo Adaptive XY Routing Using an Object Oriented Model for NOC"**. In: Proceedings of the International Conference on Microelectronics (ICM'05), 2005, pp. 204-208.

- [DIC98] Di Caro, G.; Dorigo, M. **"AntNet: Distributed stigmergetic control for communications networks"**. Journal of Artificial Intelligence Research, vol. 9, December 1998, pp. 317-365.
- [DUA97] Duato, J.; Yalamanchili, S.; Ni, L. **"Interconnection Networks: An Engineering Approach"**. Los Alamitos, California:IEEE Computer Society Press, 1997, 1 ed, 515 p.
- [DUM03] Dumitras, T.; Kerner, S.; Marculescu, R. **"Towards On-Chip Fault-Tolerant Communication"**. In: Proceedings of the Asia South Pacific Design Automation Conference (ASP-DAC'03), 2003, pp. 225-232.
- [DUM04] Dumitras, T.; Kerner, S.; Marculescu, R. **"Enabling On-Chip Diversity through Architectural Communication Design"**. In: Proceedings of the Asia South Pacific Design Automation Conference (ASP-DAC'04), 2004, pp.799-805.
- [FRE09] Freescale Semiconductors, Inc. Capturado em: <http://www.freescale.com>, Janeiro 2009.
- [GAP09] Grupo GAPH. **"Atlas - An Environment for NoC Generation and Evaluation"**. Capturado em: [http://www.inf.pucrs.br/~gaph/AtlasHtml/AtlasIndex\\_us.html](http://www.inf.pucrs.br/~gaph/AtlasHtml/AtlasIndex_us.html), Janeiro 2009.
- [GLA94] Glass, C.; Ni, L. **"The Turn Model for Adaptive Routing"**. Journal of the ACM, vol. 41-5, September 1994, pp. 874-902.
- [GOO05] Goossens, K.; Dielissen, J.; Radulescu, A. **"Aetheral Network on Chip: Concepts, Architectures, and Implementations"**. IEEE Design & Test of Computers, vol. 22-5, September-October 2005, pp. 414-421.
- [GRA03] Grama, A.; Gupta, A.; Karypis, G.; Kumar, V. **"Introduction to Parallel Computing"**. The Benjamin/Cummings Publishing Company, 2 ed., 2003, 636 p.
- [GRA08] Gratz, P.; Grot, B.; Keckler, S. **"Regional Congestion Awareness for Load Balance in Networks-on-Chip"**. In: IEEE International Symposium on High Performance Computer Architecture (HPCA'08), 2008, pp.: 203-214.
- [GUE00] Guerrier, P.; Greiner, A. **"A Generic Architecture for on Chip Packet-Switched Interconnections"**. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE'00), 2000, pp.250-256.
- [HAR04] Harmanci, M.; Escudero, N.; Leblebici, Y.; Ienne, P. **"Providing QoS to connection-less packet-switched NoC by implementing DiffServ functionalities"**. In: Proceedings of the International Symposium on System-on-Chip (SOC'04), 2004, pp.: 37-40.

- [HO01] Ho, R.; Mai, K.; Horowitz, M. ***"The future of wires"***. Proceedings of the IEEE, vol. 89-4, April 2001, pp. 490-504.
- [HU04] Hu, J.; Marculescu, R. ***"DyAD: Smart Routing for Network-on-Chip"***. In: Proceedings of the Annual Conference on Design Automation (DAC'04), 2004, pp. 260-263.
- [INF09] Infiniband Trade Association. Capturado em: <http://www.infinibandta.org>, Janeiro 2009.
- [INT09] Intel Corporation. Capturado em: <http://www.intel.com>, Janeiro 2009.
- [ITR05] ITRS. ***"2005 International Technology Roadmap for Semiconductors"***. Capturado em: <http://www.itrs.net/Links/2005ITRS/Home2005.htm>, Janeiro 2009.
- [ITR08] ITRS. ***"International Technology Roadmap for Semiconductors, 2008 Update - Overview"***. Disponível em [http://www.itrs.net/Links/2008ITRS/Update/2008\\_Update.pdf](http://www.itrs.net/Links/2008ITRS/Update/2008_Update.pdf), capturado em Outubro de 2009, 2008.
- [JAF08] Jafari, F.; Yaghmaee, M. ***"A novel flow control scheme for best effort traffics in Network-on-Chip based on Weighted Max-Min-fairness"***. In: Proceedings of International Symposium on Telecommunications (IST'08), 2008, pp. 458-463.
- [JER05a] Jerraya, A.; Wolf, W. ***"Multiprocessor Systems-on-Chips"***. San Francisco, California: Morgan Kaufmann publisher, 2005, 581 p.
- [JER05b] Jerraya, A.; Tenhunen, H.; Wolf, W. ***"Guest Editors' Introduction: Multiprocessor Systems-on-Chips"***. IEEE Computer, vol. 38-7, July 2005. pp. 36-40.
- [KAR01] Karim, F.; Nguyen, A.; Dey, S.; Rao, R. ***"On-chip Communication Architecture for OC-768 Network Processors"***. In: Proceedings of the Annual Conference on Design Automation (DAC'01), 2001, pp. 678-683.
- [KAV06] Kavaldjiev, N.; Smit, G.; Wolkotte, P.; Jansen, P. ***"Providing QoS Guarantees in a NoC by Virtual Channel Reservation"***. In: Proceedings of the International Workshop on Applied and Reconfigurable Computing (ARC'06), 2006, pp. 299-310.
- [KIM05] Kim, J.; Park, D.; Theocharides, T.; Vijaykrishnan, N.; Das, C. ***"A Low Latency Router Supporting Adaptivity for On-Chip Interconnects"***. In: Proceedings of the Annual Conference on Design Automation (DAC'05), 2005, pp. 559-564.
- [KIM07] Kim, K.; Kim, D.; Lee, K.; Yoo, V. ***"Cost-efficient Network-on-Chip Design Using Traffic Monitoring System"***. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE'07), 2007.

- [KOU04] Koundinya, A.; Negi, A.; Sastry, V. ***“Unicast routing algorithm with multiple quality-of-service parameters”***. In: International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04), 2004, pp. 520-525.
- [KUM02] Kumar, S.; Jantsch, A. Millbert, M.; Berg, J.; Soininen, J.; Forsell, M.; Tiensyrj, K.; Hemani, A. ***“A Network on Chip Architecture and Design Methodology”***. In: Proceedings of the IEEE Computer Society Annual Symposium on Very Large Scale Integration (ISVLSI'02), 2002, pp. 105-112.
- [KUM08] Kumar, A.; Peh, L.; Jha, N. ***“Token Flow Control”***. In: IEEE/ACM International Symposium on Microarchitecture (MICRO'08), 2008, pp. 342-353.
- [LIM08] Lim, S. ***“Practical Problems in VLSI Physical Design Automation”***. New York:Springer-Verlag, 2008, 264p.
- [LIN05] Link, G.; Vijaykrishnan, N. ***“Hotspot Prevention Through Runtime Reconfiguration in Network-on-Chip”***. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE'05), 2005, pp. 648-649.
- [LIN06] Link, G.; Vijaykrishnan, N. ***“Thermal trends in emerging technologies”***. In: International Symposium on Quality Electronic Design (ISQED'06), 2006, pp.: 632-639.
- [MAR01] Martin, G.; Chang, H. ***“Tutorial 2 - System on Chip Design”***. In: International Symposium on Integrated Circuits, Devices & Systems (ISIC'01), 2001, 6 p.
- [MAR05] Marcon, C.; Calazans, N.; Moraes, F.; Susin, A.; Reis, I.; Hessel, F. ***“Exploring NoC Mapping Strategies: An Energy and Timing Aware Technique”***. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE'05), 2005, pp. 502-507.
- [MAR07] Marcon, C.; Moreno, E.; Calazans, N.; Moraes, F. ***“Evaluation of Algorithms for Low Energy Mapping onto NoCs”***. In: Proceedings of IEEE International symposium on Circuits and Systems (ISCAS'07), 2007, pp. 389-392.
- [MAR08] Marcon, C.; Moreno, E.; Calazans, N.; Moraes, F. ***“Comparison of NoC Mapping Algorithms Targeting Low Energy Consumption”***. IET Computers & Digital Techniques, vol. 2-6, November 2008, pp. 471-482.
- [MIL04] Millberg, M.; Nilsson, E.; Thid, R.; Kumar, S.; Jantsch, A. ***“The Nostrum Backbone - A Communication Protocol Stack for Networks on Chip”***. In: Proceedings of the International Conference on Very Large Scale Integration Design (VLSI'04), 2004, pp. 693-696.

- [MON96] Monteiro, E.; Quadros, G.; Boavida, F. **"A Scheme for the Quantification of Congestion in Communication Services and Systems"**. In: International Workshop on Services in Distributed and Networked Environments (SDNE'96), 1996, pp. 52-59.
- [MOR04] Moraes, F.; Calazans, N.; Mello, A.; Moller, L.; Ost, L. **"HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip"**. The VLSI Journal, Amsterdam, vol. 38-1, October 2004, pp. 69-93.
- [MOR08] Moreno, E.; Popovici, K.; Calazans, N.; Jerraya, A. **"Integrating Abstract NoC Models within MPSoC Design"**. In: IEEE/IFIP International Symposium on Rapid System Prototyping (RSP'08), 2008, p. 65-71.
- [MOU06] Mouhoub, R.; Hammami, O. **"NOC Monitoring Feedback for Parallel Programmers"**. In: Proceedings of the IEEE North-East Workshop on Circuits and Systems (NEWCAS'06), 2006, pp. 141-144.
- [MUR04] Murali, S.; De Micheli, G. **"Bandwidth-Constrained Mapping of Cores onto NoC Architectures"**. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE'04), 2004, pp. 896-901.
- [MUR07] Murali, S.; Atienza, D.; Benini, L.; De Micheli, G. **"A Method for Routing Packets Across Multiple Paths in NoCs with In-Order Delivery and Fault-Tolerance Guarantees"**. VLSI Design Journal, vol. 2007, 2007, 11 p.
- [MYR09] Myricom Corporation. Capturado em: <http://www.myri.com/myrinet>, Janeiro 2009.
- [NGU06] Nguyen, H.; Ngo, V.; Choi, H. **"Assessing Routing Behavior on On-Chip-Network"**. In: Proceedings of the International Conference on Computer Engineering and Systems (ICCES'06), 2006, pp. 62-65.
- [NIL03] Nilsson, E.; Millberg, M.; Öberg, J.; Jantsch, A. **"Load Distribution with the Proximity Congestion Awareness in a Network on Chip"**. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE'03), 2003, pp. 1126-1127.
- [NOL04] Nollet, V.; Marescaux, T.; Verkest, D. **"Operating System Controlled Network on Chip"**. In: Proceedings of the Annual Conference on Design Automation (DAC'04), 2004, pp. 256-259.
- [NOU06] Nousias, I.; Arslan, T. **"Wormhole Routing with Virtual Channels using Adaptive Rate Control for Network-on-Chip (NoC)"**. In: NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06), 2006, pp. 420-423.
- [OCP09] OCP International Partnership, Capturado em: <http://www.ocpip.org>, Janeiro 2009.

- [OGR06a] Ogras, U.; Marculescu, R. ***"It's a small world after all: NoC performance optimization via long-range link insertion"***. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14-7, July 2006, pp. 693-706.
- [OGR06b] Ogras, U.; Marculescu, R. ***"Prediction-Based Flow Control for Network-on-Chip Traffic"***. In: Proceedings of the Annual Conference on Design Automation (DAC'06), 2006, pp. 839-844.
- [OST05] Ost, L.; Mello, A.; Palma, J.; Moraes, F.; Calazans, N. ***"MAIA - A framework for networks on chip generation and verification"***. In: Proceedings of the Asia South Pacific Design Automation Conference (ASP-DAC'05), 2005, pp. 49-52.
- [PAL04] Palermo, G.; Silvano, C. ***"Pirate: A Framework for Power/Performance Exploration of Network on Chip Architectures"***. Lecture Notes in Computer Science, vol. 3254, August 2004, pp. 521-531.
- [PAS06] Pastrnak, M.; de With, P.; Meerbergen, J. ***"Realization of QoS management using negotiation algorithms for multiprocessor NoC"***. In: Proceedings of IEEE International symposium on Circuits and Systems (ISCAS'06), 2006, pp. 1912-1915.
- [PAT96] Patterson, D.; Hennessy, J. ***"Computer Architecture: A Quantitative Approach"***. San Francisco, California: Morgan Kaufmann, 1996, 3th ed., 760 p.
- [PHI09] Philips Semiconductors. Capturado em: <http://www.philips.com>, Janeiro 2009.
- [PON08] Pontes, J. ***"Projeto e Prototipação de Interfaces e Redes Intrachip Não-Síncronas em FPGAs"***. Dissertação de Mestrado, PPGCC-FACIN-PUCRS. Fevereiro 2008. 122p.
- [RAD05] Radulescu, A.; Dielissen, J.; Pestana, S.; Gangwal, O.; Rijpkema, E.; Wielage, P.; Goossens, K. ***"An Efficient On-Chip NI Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Configuration"***. IEEE Transactions on CAD of Integrated Circuits and Systems, vol. 24-1, January 2005, pp. 4-17.
- [RAJ00] Rajsuman, R. ***"System-on-a-Chip - Design and Test"***. Santa Clara: Artech House, 2000, 1 ed., 277 p.
- [RAN08] Rantala, V.; Lehtonen, T.; Liljeberg, P.; Plosila, J. ***"Distributed Traffic Monitoring Methods for Adaptive Network-on-Chip"***. In: Proceedings of the IEEE Norchip Conference (NORCHIP'08), 2008, pp. 233-236.
- [RAU94] Rauscher, T. ***"Time to Market Problems - The Organization is the Real Cause"***. In: Proceedings of the IEEE International Engineering Management Conference (IEMC'94), 1994, pp. 338-345.



- [RIC06] Richardson, T.; Nicopoulos, C.; Park, D.; Narayanan, V.; Xie, Y.; Das, C.; Degalahal, V. "**A Hybrid SoC Interconnect with Dynamic TDMA-Based Transaction-Less Buses and On-Chip Networks**". In: International Conference on VLSI Design (VLSID'06), 2006, pp.: 657-664.
- [SCH08] Schafer, B.; Kim, T. "**Hotspots Elimination and Temperature Flattening in VLSI Circuits**". IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16-11, November 2008, pp. 1475-1487.
- [SCH97] Schaller, R. "**Moore's Law: Past, Present and Future**". IEEE Spectrum, vol. 34-6, June 1997, pp. 52-59.
- [SHI96] Shim, K.; Daniel, S. "**Analysis and Implementation of Hybrid Switching**". IEEE Transactions on Computers, vol. 45-6, June 1996, pp. 684-692.
- [SHO02] Shobaki, M. "**On-Chip Monitoring of Single- and Multiprocessor Hardware Real-Time Operating Systems**". In: Proceedings of the International Conference on Real-Time Computing Systems and applications (RTCSA'02), 2002.
- [SOB06] Sobhani, A.; Daneshtalab, M.; Neishaburi, M.; Mottaghi, M.; Afzali-Kusha, A.; Fatemi, O.; Navabi, Z. "**Dynamic Routing Algorithm for Avoiding Hot Spots in On-chip Networks**". In: International Conference on Design and Test of Integrated Systems in Nanoscale Technology (DTIS'06), 2006, pp. 179-183
- [SRI05] Srinivasan, K.; Chatha, K.; Konjevod, G. "**An Automated Technique for Topology and Route Generation of Application Specific On-Chip Interconnection Networks**". In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'05), 2005, pp.: 231-237.
- [TAN08] Tang, S.; Xu, Q. "**A Debug Probe for Concurrently Debugging Multiple Embedded Cores and Inter-Core Transaction in NoC-Based Systems**". In: Proceedings of the Asia South Pacific Design Automation Conference (ASP-DAC'08), 2008, pp. 416-421.
- [TED05] Tedesco, L. "**Uma proposta para Geração de Tráfego e Avaliação de Desempenho para NoCs**". Dissertação de Mestrado, PPGCC-FACIN-PUCRS. Dezembro 2005. 126p. Disponível em [http://www.inf.pucrs.br/~moraes/my\\_pubs/papers/dissertacao\\_leonel.pdf](http://www.inf.pucrs.br/~moraes/my_pubs/papers/dissertacao_leonel.pdf)
- [TEX09] Texas Instruments, Inc. Capturado em: <http://www.ti.com>, Janeiro 2009.
- [TIL09] Tiler Corporation. "**TILE64™ Processor**". Capturado Em: [http://www.tiler.com/pdf/ProBrief\\_Tile64\\_Web.pdf](http://www.tiler.com/pdf/ProBrief_Tile64_Web.pdf), Janeiro 2009.

- [VAN07] van den Brand, J.; Ciorda, C.; Basten, T.; Goossens, K. **“Congestion-Controlled Best-Effort Communication for Networks-on-Chip”**. In: Proceedings of the Conference on Design, Automation and Test in Europe (DATE'07), 2007, pp. 948-953.
- [VIS07] Vishnu, A.; Koop, M.; Moody, A.; Mamidala, A.; Narravula, S.; Panda, D. **“Hot-Spot Avoidance With Multi-Pathing Over InfiniBand: An MPI Perspective”**. In: Proceedings of the IEEE International Symposium on Cluster and the Grid (CCGrid'07), 2007, pp. 479-486.
- [WEN07] Wenbiao, Z.; Zhang, Y.; Mao, Z. **“Link-load Balance Aware Mapping and Routing for NoC”**. WSEAS Transactions on Circuits and Systems, vol. 6-11, November 2007, pp. 583-591.
- [WES94] Weste, N.; Eshragian, K. **“Principles of CMOS VLSI Design: a Systems Perspective”**. Reading: Addison-Wesley, 1994, 2 ed., 713 p.
- [WIN01] Wingard, D. **“Micro-Network Based Integration for SoCs”**. In: Proceedings of the Annual Conference on Design Automation (DAC'01), 2001, pp. 673-677.
- [WOS07] Woszezenki, C. **“Alocação de Tarefas e Comunicação entre Tarefas em MPSoCs”**. Dissertação de Mestrado, PPGCC-FACIN-PUCRS. Fevereiro 2007. 121p. Disponível em [http://www.inf.pucrs.br/~moraes/my\\_pubs/papers/dissertacao\\_cris.pdf](http://www.inf.pucrs.br/~moraes/my_pubs/papers/dissertacao_cris.pdf)
- [XIL09] Xilinx, Inc. Capturado em: <http://www.xilinx.com>, Janeiro 2009.
- [YAN95] Yan, C.; Reddy, A. **“A Taxonomy for Congestion Control Algorithms in Packet Switching Networks”**. IEEE Network, vol. 9-5, July/August 1995, pp. 34-45.
- [ZEF03] Zeferino, C.; Susin, A. **“SoCIN: A Parametric and Scalable Network-on-Chip”**. In: Symposium on Integrated Circuits and System Design (SBCCI'03), 2003, pp. 169-174.
- [ZHO06] Zhou, W.; Zhang, Y.; Mao, Z. **“An application specific NoC mapping for optimized delay”**. In: International Conference on Design and Test of Integrated Systems in Nanoscale Technology (DTIS'06), 2006, pp. 184-188.
- [ZHU02] Zhu, Q.; Matsuda, A.; Shoji, M.; Kuwamura, S.; Nakata, T. **“An Object-Oriented Design Process for System-on-Chip Using UML”**. In: Proceedings of the International Symposium on System Synthesis (ISSS'02), 2002, pp. 249-254.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)