



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação



PROPOSTA DE UMA INFRAESTRUTURA DE GERAÇÃO E AVALIAÇÃO PARA REDES INTRACHIP HERMES-G

RAFFAEL BOTTOLI SCHEMMER

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

ORIENTADOR: PROF. DR. NEY LAERT VILAR CALAZANS

PORTO ALEGRE, AGOSTO DE 2012

PROPOSTA DE UMA INFRAESTRUTURA DE GERAÇÃO E AVALIAÇÃO PARA REDES INTRACHIP HERMES-G

RESUMO

Os avanços relacionados à tecnologia de fabricação de circuitos integrados impulsionam a complexidade e o número de funcionalidades dos produtos eletrônicos. A literatura aponta que até 2015 tarefas do nível comportamental de projeto de sistemas eletrônicos ocuparão cerca de 50% do esforço de projeto, o que reforça a necessidade do desenvolvimento de ferramentas de automação e geração automática de circuitos. Além disso, o projeto de circuitos atuais faz uso do paradigma de projeto síncrono, que associado ao crescimento da complexidade dos mesmos impõe restrições importantes com relação ao consumo de energia e à dissipação de potência.

Este trabalho se apresenta como uma solução alternativa para alguns destes problemas, pela proposta de um ambiente de geração e avaliação de redes intrachip. Tais redes permitem, além de conectar módulos de processamento que operem em diferentes frequências, podem ajudar a garantir o atendimento de restrições temporais impostas pelos requisitos de tráfego destes módulos. Durante a geração da rede, o ambiente permite, em tempo de projeto, variar as características da mesma, tais como as frequências de operação dos roteadores, de forma individualizada.

Além da geração da rede, o ambiente ainda habilita avaliar restrições temporais de diferentes modelos de tráfegos, dando suporte à parametrização do tráfego na rede. Esta característica de permitir avaliar a rede através de um modelo de tráfego oferece alternativas para reduzir o esforço do projeto dos sistemas eletrônicos ainda nas fases de especificação de requisitos do sistema. Isto ocorre por que o ambiente facilita a visualização do comportamento de um modelo de rede denotando se o mesmo atende ou não a requisitos esperados para um cenário de tráfego.

Palavras chave: Geração de Redes, Geração de Tráfego, Redes Intrachip Não Síncronas, Avaliação de Tráfego.

PROPOSAL OF AN INFRASTRUCTURE FOR THE GENERATION AND EVALUATION OF HERMES-G NETWORKS ON CHIP

ABSTRACT

The advances in integrated circuits manufacturing technology increment the complexity and number of functionalities of electronic products. Current literature points to the fact that in 2015 behavioral level tasks of electronic systems design will take as much as 50% of the design effort, which reinforces the need for the deployment of automatic tools for circuit generation. Besides that, the design of current circuits capitalizes on the use of the synchronous paradigm, which coupled to the growing complexity of the later imposes relevant constraints on energy consumption and power dissipation.

This work presents an alternative solution for some of these problems, by proposing an environment for the generation and evaluation of intrachip networks. The addressed class of networks allows connecting processing modules operating at distinct clock frequencies. In addition, they may help guaranteeing the fulfillment of temporal restrictions imposed by the traffic requirements of the involved modules. During the network generation process, the environment allows choosing the network characteristics at design time, such as specifying the network routers operating frequencies individually.

Besides network generation, the environment also enables evaluating temporal constraints of different traffic models, providing support to network traffic parameterization. This feature offer alternatives to reduce the overall electronic system design effort early, during system requirements specification phases. This occurs because the environment eases the visualization of the network behavior, denoting if the model fulfills or not the expected requirements for a given traffic scenario.

Keywords: Network Generation, Traffic Generation, Non-Synchronous Networks on Chip, Traffic Evaluation.

LISTA DE FIGURAS

Figura 1: Topologias suportadas pela ferramenta OPNET: (a) 2D Mesh (b) Fat Tree (c) Big Fat Tree.	25
Figura 2: Modelos propostos por Kreutz et al. (a) Modelo ACP, define os custos de computação e dependências entre as tarefas. (b) Modelo CRG, define a arquitetura de comunicação onde a tarefa vai ser mapeada.	27
Figura 3: Arquitetura típica de um cluster na rede DSPIN, formada por três IPs, uma arquitetura de interconexão local e um adaptador de rede, que conecta o cluster a um roteador, este formado por duas portas, uma de envio e outra de recebimento de dados.	28
Figura 4: Interface gráfica de geração de redes do ambiente MAIA. (1) Permite o usuário selecionar os parâmetros da rede. (2) Representa uma rede 4x4 Mesh. (3) Dispara o ambiente que gera a rede selecionada.	29
Figura 5: Fila bi síncrona utilizada em redes HERMES-G e HERMES-GLP. Nesta arquitetura de fila, a escrita e a leitura podem ou não operar na mesma fase ou na mesma frequência. A codificação de Gray é utilizada nos ponteiros de leitura e de escrita da fila para garantir que ambos os ponteiros estejam sincronizados, mesmo sendo manipulados em diferentes domínios de fase e de frequência.	30
Figura 6: Topologias suportadas pelo ambiente de geração: (a) malha 2D, (b) anel, (c) Spidergon, (d) crossbar.	31
Figura 7: (a) Categoria de serviço CBR do inglês “Constant Bit Rate” em que todos os pacotes são injetados na rede na mesma taxa de injeção. (b) Categoria de serviço VBR, do inglês <i>variable bit rate</i> em que a taxa de injeção é variável ao longo do tempo. (c) Distribuição espacial do cenário de tráfego composto por um gerador de vídeo e um gerador de voz.	33
Figura 8: Curva exponencial decrescente de 1000 pacotes gerados em um intervalo de 100 Mbps a 200 Mbps com incremento de 10 Mbps e média de 151 Mbps.	36
Figura 9: Fluxo de projeto de geração de uma rede no ambiente ATLAS, composto por quatro passos (1) Criação de um projeto. (2) Definição das características desejadas para uma rede. (3) Geração da rede. (4) Edição de um projeto de rede gerado.	42
Figura 10: Curvas proibidas e permitidas para os algoritmos adaptativos: (a) Negative First proíbe curvas para Oeste se o pacote está subindo e curvas para o sul se o pacote está indo para a direita. (b) West First proíbe curvas para Oeste. (c) North Last proíbe qualquer curva após o sentido Norte ter sido tomado.	44
Figura 11: Diagrama de blocos da fila bi síncrona utilizada pela rede HERMES-G.	45
Figura 12: Conexões de entrada e saída das arquiteturas (a) Async_Fifo e (b) fila bi síncrona.	47
Figura 13: Máquina de estados de leitura da fila bi síncrona, composta por cinco estados, sendo um deles responsável pela inicialização, dois responsáveis pelo roteamento do pacote e dois pela transmissão dos <i>flits</i> do pacote.	47

Figura 14: Interface principal do ambiente de geração de tráfego HERMES-G e suas opções de seleção de características de geração de redes.....	50
Figura 15: Componente da interface gráfica utilizável para gerenciar valores de frequência definidos pelo usuário. Estas são utilizadas para definir as frequências de operação de roteadores e de módulos IP a estes conectados.	51
Figura 16: Interface de adição de uma nova frequência. Campos (1) e (2) possibilitam informar um nome para a frequência e um valor para a frequência.	52
Figura 17: Interface de remoção de frequências. Esta interface possui apenas um campo, onde se pode selecionar uma frequência a remover.....	52
Figura 18: Interface de edição de frequências. Permite selecionar uma frequência cadastrada e modificar seus campos referentes a nome, valor e unidade de frequência.	53
Figura 19: Interface de seleção de frequências cadastradas para o roteador (1), módulo de transmissão de pacotes (2) e módulo de recepção de pacotes (3).	53
Figura 20: Combinações existentes com relação ao número de portas utilizadas pelos roteadores em uma rede com topologia malha 2D.	55
Figura 21: (a) Descrição de um roteador formado por cinco portas, um <i>crossbar</i> e um <i>SwitchControl</i> (SWC). (b) Descrição de uma rede malha formada por nove roteadores.....	56
Figura 22: Distribuições espaciais suportadas pelo modelo de tráfego sintético. (a) Aleatória: Roteador endereço zero envia pacotes de maneira aleatória para os roteadores endereços entre 1 e 8. (b) Destino único: Roteador endereço zero envia todos os pacotes para um único endereço. (c) Complemento: por exemplo, o roteador 4 (em coordenadas XY 11, em quatro bits 0101) envia pacotes para o roteador 8 (em coordenadas XY 22, em quatro bits 1010 ou seja, o complemento de 0101).	58
Figura 23: Modelo de variação das taxas de injeção dos pacotes. O tamanho dos pacotes é sempre o mesmo, onde o tempo ocioso é variado conforme a taxa de injeção de cada pacote.....	58
Figura 24: Equação que calcula o tempo ocioso de cada pacote a partir do tamanho do pacote, da capacidade máxima de transmissão e da taxa de injeção do pacote.....	58
Figura 25: Equação que calcula a função de probabilidade seguindo uma distribuição normal de uma taxa de injeção informada.	59
Figura 26: Exemplo de uma distribuição normal de 1000 pacotes, distribuídos em um intervalo com taxa mínima de 1Mbps e máxima de 1000Mbps, média de 500Mbps, desvio padrão de 100Mbps e incremento de 10Mbps.....	60
Figura 27: Equação que calcula a função de probabilidade seguindo uma distribuição exponencial de uma taxa de injeção informada.	60
Figura 28: Exemplo de uma distribuição exponencial de 1000 pacotes distribuídos em um intervalo com taxa mínima de 1Mbps e máxima de 1000Mbps, média de 100Mbps e incremento de 10Mbps.	61

Figura 29: Interface principal do gerador de tráfego, que possibilita criar múltiplos cenários de tráfego.....	62
Figura 30: Interface que possibilita parametrizar o tráfego, tais como número de pacotes, tamanho dos pacotes, distribuição espacial e temporal.....	63
Figura 31: Formato de um pacote de tráfego. Os campos representam: (a) Tempo de transmissão do pacote; (b) Roteador destino do pacote; (c) Tamanho do pacote; (d) Roteador origem do pacote; (e) Tempo de injeção em decimal; (f) Número de sequência do pacote; (g) Carga útil (dados) do pacote.	64
Figura 32: Definição de um pacote, formado pelos campos de Header e Payload, onde um representa informação de controle de transmissão e o outro contém os dados do pacote.....	65
Figura 33: Descrição da interface da entidade SC_Input_Module, que implementa o componente transmissor de pacotes para a rede.....	66
Figura 34: Formato de um pacote gerado pelo transmissor de pacotes.....	67
Figura 35: Descrição da interface da entidade SC_Output_Module, que descreve o componente receptor de pacotes.	67
Figura 36: Pacote gerado pelo receptor de pacotes utilizado para avaliação do tráfego dos pacotes.	68
Figura 37: Interface linha de comando da ferramenta de testes responsável pela geração e verificação de cenários.	70
Figura 38: Uso de modelos de aplicações CDCM no ambiente Atlas: (a) criação e geração de um grafo de aplicações CDCM; (b) geração do Testbench pela ferramenta ATLAS; (c) geração dos arquivos de tráfego e o ambiente de transmissão e recepção de pacotes.....	71
Figura 39: Ferramenta de criação de aplicações CDCM através de grafos CDCG no ambiente CAFES.	72
Figura 40: Ferramenta de mapeamento do ambiente CAFES. Cada tarefa da aplicação é mapeada em um dos endereços da rede.	73
Figura 41: Arquivo que contém as características de um modelo de aplicação CDCM.	74
Figura 42: Interface principal do gerador de tráfego, que possibilita carregar arquivos que descrevem modelos de aplicações.	74
Figura 43: Formato de um pacote de tráfego. Os campos são: (a) Origem do pacote; (b) Roteador destino do pacote; (c) Tempo de processamento do pacote em ciclos; (d) Tamanho do pacote; (e) Número de sequência do pacote.	75
Figura 44: Descrição das portas de entrada e saída da entidade SC_Input_Module, que descreve o componente transmissor de pacotes da rede para o modelo de Testbench de modelos de aplicações CDCM.....	76

Figura 45: Equação que realiza o cálculo da latência de um pacote. A latência é calculada a partir da subtração do instante em que o pacote está apto a entrar na rede do instante em que o último <i>flit</i> do pacote deixou a rede.	79
Figura 46: Equação de cálculo da vazão de um pacote. A vazão é calculada a partir da divisão do tamanho do pacote pelo tempo gasto (latência) de transmissão do pacote. Logo após o valor é multiplicado por 1000 para ser expresso em Mbps.	80
Figura 47: Equação que calcula a latência média de um conjunto de pacotes a partir do somatório das latências medidas dos pacotes dividido pelo número total de pacotes.	82
Figura 48 Equação que calcula o desvio padrão da latência de um conjunto de pacotes.	82
Figura 49: Equação que calcula a vazão ideal de um pacote pela divisão do tamanho do pacote em bits pela latência ideal do pacote em tempo absoluto. Depois de calculado o valor é multiplicado por 1000 para ser convertido em Mbps.	83
Figura 50: Equação que calcula a vazão média de um conjunto de pacotes, a partir do somatório das vazões calculadas dos pacotes dividido pelo número total de pacotes.	83
Figura 51: Equação que calcula o desvio padrão da vazão de um conjunto de pacotes.	84
Figura 52: Interface da ferramenta distribuição de vazões do ambiente de avaliação de tráfego. No exemplo, é usado um tráfego uniforme de 400 Mbps de dois pacotes transmitidos do endereço 00 ao endereço 11.	84
Figura 53: Interface da ferramenta distribuição de latências do ambiente de avaliação de tráfego. O exemplo usa um tráfego uniforme de 400 Mbps de dois pacotes transmitidos do endereço 00 ao endereço 11.	85
Figura 54: Interface da ferramenta analisador de latências do ambiente de avaliação de tráfego, que permite visualizar latências específicas de pacotes de tráfegos.	86
Figura 55: Interface da ferramenta relatório global que permite visualizar latência e a vazão ideais e vazão e a latência medidas de todos os tráfegos.	86
Figura 56: Rede 3x3 contendo duas ilhas de frequência. O quadrado pontilhado simula quatro processadores operando a 500 MHz. Neste domínio todos os roteadores se comunicam utilizando filas síncronas (S). O restante dos roteadores (polígono preto) simula duas memórias e três controladores de entrada e saída, operando a 50 MHz cada. Repare que as filas bi síncronas (BS) são utilizadas para comunicação unicamente entre os diferentes domínios de frequência.	87
Figura 57: Latência média dos três cenários de tráfego propostos para o modelo de rede síncrona e o modelo de rede não síncrona.	90
Figura 58: Vazão média dos três cenários de tráfego propostos para o modelo de rede síncrona e o modelo de rede não síncrona.	90
Figura 59: Rede 2x2 contendo duas ilhas de frequência, uma formada pelos roteadores (00/10/01) onde a fila síncrona é utilizada para comunicação entre os roteadores. O roteador 11, por possuir frequência de operação diferente dos demais usa filas bi síncronas.	91

Figura 60: Vazão e latência média dos cenários de tráfego (00-11) e (11-00) para os modelos de rede que fazem uso unicamente de filas bi síncronas (2x2J – 2x2G) e para os modelos que fazem uso de filas síncronas e bi síncronas (2x2GS – 2x2JS).....93

Figura 61: Cenário de tráfego proposto, formado por uma aplicação composta por quatro tarefas.96

LISTA DE TABELAS

Tabela 1: Resumo do estado da arte nos temas de projeto, e geração de redes intrachip.....	38
Tabela 2: Resumo do estado da arte nos temas de caracterização e geração de tráfego para redes intrachip.	40
Tabela 3: Cálculo do <i>TimeStamp</i> para três pacotes de 13 <i>flits</i> de tamanho injetados a 100Mbps por um transmissor operando a 100 MHz em uma rede de comprimento de <i>flit</i> igual a 8 bits.	64
Tabela 4: Valores de vazão e latência média de todos os pacotes de cada tráfego, obtidos durante a simulação dos tráfegos para os modelos de rede síncrona e não síncrona.	89
Tabela 5: Resultado de área para cada um dos modelos de rede, em LUTs de quatro entradas.	91
Tabela 6: Valores de vazão e latência média de todos os pacotes dos tráfegos (00-11) e (11-00) obtidos durante a simulação dos tráfegos para os quatro modelos de rede propostos.	92
Tabela 7: Resultados de área obtidos para cada uma das redes propostas, em LUTs de quatro entradas.	93
Tabela 8: Valores de vazão e latência médias do tráfego (00-11), obtidos durante a simulação dos modelos de rede propostos.	94
Tabela 9: Resultado de área obtida para cada um dos sete modelos de rede propostos em LUTs de quatro entradas.	95
Tabela 10: Valores de vazão e latência média para cada um dos três tráfegos simulados nos sete modelos de redes propostas.....	95
Tabela 11: Valores de vazão e latência média dos cenários de tráfego obtidos durante a simulação.	97

LISTA DE SIGLAS

ACP	APPLICATION COMMUNICATION PATTERN
CAFES	COMMUNICATION ANALYSIS FOR EMBEDDED SYSTEMS
CDCG	COMMUNICATION DEPENDENCE AND COMPUTATION GRAPH
CDCM	COMMUNICATION DEPENDENCE AND COMPUTATION MODEL
CRG	COMMUNICATION RESOURCE GRAPH
EEPROM	ELECTRICALLY ERASABLE PROGRAMMABLE READ-ONLY MEMORY
FPGA	FIELD PROGRAMMABLE GATE ARRAY
GALS	GLOBALLY ASYNCHRONOUS LOCALLY SYNCHRONOUS
HPC	HIGH PERFORMANCE COMPUTING
IP	INTELLECTUAL PROPERTY
LUT	LOOK UP TABLE
MPEG	MOVING PICTURE EXPERTS GROUP
NOC	NETWORK ON CHIP
OCP	OPEN CORE PROTOCOL
PARIS	PARAMETERIZABLE INTERCONNECT NETWORK
RTL	REGISTER TRANSFER LEVEL
SOC	SYSTEM ON CHIP
SVN	SUBVERSION
TCL	TOOL COMMAND LANGUAGE
TL	TRANSACTION LEVEL
VHDL	VHSIC HARDWARE DESCRIPTION LANGUAGE
VHSIC	VERY HIGH SPEED INTEGRATED CIRCUIT

SUMÁRIO

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL	1
FACULDADE DE INFORMÁTICA	1
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO	1
PROPOSTA DE UMA INFRAESTRUTURA DE GERAÇÃO E AVALIAÇÃO PARA REDES INTRACHIP HERMES-G	3
PROPOSAL OF AN INFRASTRUCTURE FOR THE GENERATION AND EVALUATION OF HERMES-G NETWORKS ON CHIP	5
LISTA DE FIGURAS.....	7
LISTA DE TABELAS.....	13
LISTA DE SIGLAS.....	15
SUMÁRIO	17
1 INTRODUÇÃO	19
1.1 MOTIVAÇÃO DO TRABALHO	21
1.2 OBJETIVOS DO TRABALHO.....	22
1.3 ORGANIZAÇÃO DO DOCUMENTO	22
2 TRABALHOS RELACIONADOS	25
2.1.1 Fen et al.....	25
2.1.2 Kreutz et al.	26
2.1.3 Panades et al.....	27
2.1.4 Ost et al.....	28
2.1.5 Pontes et al.	29
2.1.6 Bononi et al.	31
2.1.7 Tedesco et al.	31
2.1.8 Liu et al.....	33
2.1.9 Hong et al.....	34
2.1.10 Bruch et al.	35
2.1.11 Schemmer et al.....	35
2.1.12 Gratz et al.....	37
2.1.13 Posicionamento do trabalho em relação ao estado da arte	37
3 GERAÇÃO DE REDES INTRACHIP NÃO SÍNCRONAS.....	41
3.1 ESTUDO INICIAL DO AMBIENTE ATLAS E DA REDE HERMES-G	41
3.1.1 Primeiro Contato com o Ambiente ATLAS.....	41
3.1.2 Primeiro Contato com a Rede HERMES-G.....	42
3.2 PROCESSO DE PARAMETRIZAÇÃO DA REDE HERMES-G	48
3.3 PROJETO DA INTERFACE GRÁFICA DO AMBIENTE DE GERAÇÃO.....	50
3.3.1 Geração e Definição de Relógios.....	51
3.4 PROCESSO DE GERAÇÃO DA REDE HERMES-G	53
3.4.1 Criação dos Diretórios e Arquivos do Projeto da Rede.....	53
3.4.2 Geração dos Roteadores e seus Componentes Internos	54
4 GERAÇÃO DE TRÁFEGO PARA REDES INTRACHIP NÃO SÍNCRONAS	57
4.1 CARACTERIZAÇÃO DE TRÁFEGO SINTÉTICO	57
4.1.1 Distribuição Espacial	57
4.1.2 Distribuição Temporal.....	58
4.2 AMBIENTE DE GERAÇÃO DE TRÁFEGO SINTÉTICO	61
4.2.1 Definição do Arquivo de Projeto do Tráfego	61

4.2.2	<i>Projeto da Interface do Gerador de Tráfego Sintético</i>	62
4.2.3	<i>Geração e Formato dos Pacotes do Tráfego</i>	63
4.2.4	<i>Geração do Testbench para Tráfego Sintético</i>	65
4.2.5	<i>Validação da Proposta</i>	70
4.3	AMBIENTE DE GERAÇÃO DE TRÁFEGO DE MODELO DE APLICAÇÕES	71
4.3.1	<i>O Ambiente CAFES: Criação e Mapeamento de um Modelo de Aplicação</i>	71
4.3.2	<i>Formato de Representação de um Modelo de Aplicação</i>	73
4.3.3	<i>Adaptação da Interface do Gerador de Tráfego</i>	74
4.3.4	<i>Geração e Formato dos Pacotes do Tráfego</i>	74
4.3.5	<i>Geração do Testbench para Modelos de Aplicação</i>	75
5	AVALIAÇÃO DE TRÁFEGO EM REDES INTRACHIP NÃO SÍNCRONAS	79
5.1	MÉTRICAS DE AVALIAÇÃO DE TRÁFEGO	79
5.1.1	<i>Latência</i>	79
5.1.2	<i>Vazão</i>	80
5.2	MEDIDAS ESTATÍSTICAS NAS MÉTRICAS DE AVALIAÇÃO	80
5.2.1	<i>Latência</i>	80
5.2.2	<i>Vazão</i>	82
5.3	AMBIENTE DE AVALIAÇÃO DE TRÁFEGO	84
5.3.1	<i>Distribuição de Vazões de um Tráfego</i>	84
5.3.2	<i>Distribuição de Latências de um Tráfego</i>	85
5.3.3	<i>Analizador de Latências</i>	85
5.3.4	<i>Relatório Global</i>	86
6	RESULTADOS OBTIDOS	87
6.1	ESTUDO DE CASO 1	87
6.2	ESTUDO DE CASO 2	90
6.3	ESTUDO DE CASO 3	93
6.4	ESTUDO DE CASO 4	94
6.5	ESTUDO DE CASO 5	96
7	CONCLUSÕES E TRABALHOS FUTUROS	99
7.1	CONTRIBUIÇÕES DO TRABALHO	99
7.2	TRABALHOS FUTUROS	100
8	REFERÊNCIAS BIBLIOGRÁFICAS	103

1 INTRODUÇÃO

Durante as últimas décadas, a evolução da tecnologia de fabricação de circuitos integrados sempre esteve diretamente relacionada às mudanças que ocorrem no processo de projeto de produtos eletrônicos. Parte desta mudança tem relação com o crescimento do número de transistores dos circuitos integrados, componentes fundamentais na concepção dos produtos eletrônicos. Em [MOO65], após um estudo do crescimento anual dos circuitos, Moore identificou que a cada 18 a 24 meses dobra-se a quantidade de transistores que se podia construir sobre a mesma área de silício, resultando em uma redução de custos e aumento do desempenho dos circuitos integrados segundo uma curva exponencial. Este fato continuou a observar-se em décadas posteriores, dando origem à chamada Lei de Moore.

O estado da arte em semicondutores faz uso da Lei de Moore como o principal incentivo ao crescimento da complexidade dos circuitos. O desafio para atender aos pressupostos da lei leva indústria e academia a investirem somas consideráveis em pesquisa e desenvolvimento para manter o crescimento dentro das expectativas da Lei. Uma organização, o *International Technology Roadmap for Semiconductors* ou ITRS, tem como objetivo analisar e prever o crescimento do mercado de semicondutores. Como resultado destes esforços, em [ARD10], Arden et al. demonstram existirem subsídios tecnológicos para o projeto de circuitos integrados capazes hoje de superar os pressupostos da Lei de Moore. Em [ITR11a], [ITR11b] e [ITR11c] são apresentados os desafios a serem vencidos para que as próximas escalas de fabricação capazes de superar o estado da arte atual estejam operacionais. Os documentos detalham que até 2016 será possível desenvolver circuitos com 14.2 nm de *min-feature size*, sendo fabricados sobre *wafers* de silício com dimensões de até 450 mm de diâmetro.

O reflexo do esforço e da disponibilidade tecnológica oferecida é visualizado em produtos que fazem uso das tecnologias de fabricação mais recentes. Até o Autor pode investigar, o menor processo de fabricação comercial utilizado nos dias atuais é descrito em [INT12] com uso da tecnologia de fabricação com 22 nm de *min-feature-size*, e fazendo uso de *wafers* de silício de 300 mm. Como outro exemplo, em [NVI12] apresenta-se um circuito integrado que contém 7.1 bilhões de transistores, capaz de atingir desempenho sustentável de 1 Teraflops, ou seja um trilhão de operações de ponto flutuante em precisão dupla por segundo. Previsões indicam que este circuito estará disponível no terceiro trimestre de 2012 com tecnologia de fabricação 28nm. Em [INT11] apresenta-se um circuito integrado com função de coprocessador, que apresenta desempenho sustentável a nível de Teraflops. Este circuito já é fabricado utilizando tecnologia de fabricação 28nm, e aplicado em supercomputadores de propósito específico.

Ainda durante as últimas décadas, a evolução nos projetos de circuitos integrados evoluiu do desenho manual de transistores ao uso de linguagens de descrição de hardware e ferramentas de síntese, que a partir da descrição abstrata efetuam a geração automática do projeto em nível físico do circuito. Um dos maiores desafios do projeto de sistemas eletrônicos está no atendimento de restrições e pré-requisitos do sistema. Uma das alternativas para superar estes

desafios está no desenvolvimento de ferramentas de automação e geração automática de circuitos. Estas devem ser tais que possam utilizar como entrada as restrições e pré-requisitos do circuito a ser gerado. Em [ITR11a] aponta-se que até 2015, o nível comportamental do projeto de sistemas eletrônicos ocupará cerca de 50% do esforço de projeto, o que reforça a necessidade de desenvolvimento de ferramentas de automação e geração automática de circuitos.

Nas últimas duas décadas, o desenvolvimento de sistemas eletrônicos compostos múltiplos elementos de processamento implicava sistemas compostos por diversos circuitos integrados distintos. Com a disponibilidade tecnológica atual, é possível desenvolver sistemas eletrônicos completos em um único circuito integrado. Para esse tipo de circuito, se dá o nome de sistema integrado em chip (do inglês *System on Chip* ou SoC) onde todos os componentes básicos do sistema eletrônico estão encapsulados em um mesmo circuito integrado. Tal tipo de sistema reúne uma série de vantagens, algumas delas diretamente relacionadas à redução dos custos de encapsulamento e de manufatura do produto. Esta tendência de projeto tende a reduzir os esforços em tempo de projeto. Por consequência, o sistema se torna mais competitivo. Outra técnica para aumentar a competitividade, e capaz de fazer uso da disponibilidade tecnológica para aumentar o número de funcionalidades dos sistemas eletrônicos, é o reuso de componentes que compõem um sistema eletrônico.

Com o reuso maciço de componentes, o projeto de um SoC passa a consistir da montagem do sistema a partir de uma maioria de módulos pré-validados e pré-caracterizados. Contudo, a interconexão em si de grande número de módulos não é tarefa trivial, sobretudo quando da interconexão depende a viabilidade do sistema como um todo. Uma arquitetura de interconexão deve ser capaz de permitir que componentes se comuniquem de forma eficiente. Diversos modelos de arquiteturas de interconexão foram propostos e difundidos. Em geral, uma arquitetura de interconexão deve garantir: (i) eficiência energética e confiabilidade (ii) escalabilidade e largura de banda (iii) reusabilidade. Um modelo de arquitetura de interconexão capaz de atender a estes requisitos, e que contém atualmente grande apelo comercial e gera muitos esforços em pesquisa são as redes intrachip, também referenciadas na literatura pelo termo *Network on Chip* ou NoC. Assume-se por convenção neste trabalho, que o termo rede intrachip será referenciado unicamente pelo termo rede. Neste modelo de rede, as decisões de arbitragem e roteamento da informação são tratadas de maneira distribuída, variando conforme as políticas de roteamento adotadas. Este modelo de arquitetura deriva suas propriedades da adaptação de conceitos provenientes de redes de computadores e/ou de sistemas distribuídos, existindo assim, semelhanças quer seja na disposição de protocolos, seja no encaminhamento de informação e até na sincronização da informação.

Por outro lado, o projeto dos circuitos integrados atuais faz uso majoritário do paradigma de projeto síncrono, em que um único sinal sincronizador é usado para coordenar todos os eventos que acontecem no circuito. Este paradigma foi adotado por favorecer a simplicidade e exigir pouca lógica para sincronizar eventos. Devido à crescente redução na escala dos componentes eletrônicos, o número de componentes a serem sincronizados, e por consequência o número de fios necessários para interconectar estes componentes ao sinal sincronizador, aumentam na mesma proporção. Em [AMD05], os autores apontam que 45% do total da potência consumida por

um processador de alto desempenho da época, era devida ao processo de distribuição do sinal de sincronização (relógio) do circuito.

A solução para as restrições impostas pelo modelo síncrono, que estão ligadas diretamente ao consumo de energia e à dissipação de potência, são temas de pesquisa e de interesse crescente do mercado. Dentre algumas das soluções apresentadas na literatura, destacam-se os circuitos que independem de sincronizadores, e circuitos que fazem uso de múltiplos sinais de sincronização. O modelo que independe de sincronizadores, referenciado por modelo assíncrono, faz uso de protocolos de comunicação e sincronização locais no lugar da lógica baseada em um sinal de sincronização global. As principais vantagens dessa abordagem estão na eliminação dos problemas causados pelo sinal de sincronização, apresentando maior robustez se comparada a circuitos que fazem uso do sinal de sincronização. Esta técnica é pouco adotada em SoCs hoje em dia, pela carência de ferramentas de projeto e de recursos humanos aptos para dar suporte à tecnologia.

Outra solução para libertar-se do modelo síncrono é aquela que faz uso de múltiplos sinais de sincronização. Nesta abordagem, conhecida em inglês pelo termo *Globally Asynchronous Locally Synchronous* ou GALS, o circuito é particionado, e as partes resultantes costumam ser referenciadas na literatura como ilhas, onde cada ilha possui um sinal de sincronização, podendo estes sinais ser diferentes uns dos outros. Cada sinal de sincronização possui uma árvore de distribuição de relógio própria. Para garantir a sincronização entre as ilhas, usam-se interfaces de sincronização, capazes de garantir a comunicação entre ilhas. De acordo com o ITRS [ITR11a], o uso da técnica GALS no projeto de circuitos com foco em baixo consumo de energia, é considerada como a segunda melhor técnica para redução no consumo de energia dinâmica de sistemas complexos, no mesmo nível de qualidade de técnicas de DVFS, do inglês *Dynamic Voltage and Frequency Scaling*, em que a voltagem e a frequência do circuito são ajustadas dinamicamente, conforme a demanda de operação do circuito.

1.1 MOTIVAÇÃO DO TRABALHO

A carência de recursos de especificação, verificação e síntese para sistemas não síncronos dá sentido a um esforço em pesquisa para fazer evoluir as técnicas de especificação, verificação e síntese de SoCs GALS em geral, e de arquiteturas de comunicação para tais SoCs em particular.

O interesse em redes de interconexão que utilizam a abordagem GALS como tema de pesquisa possibilita explorar alternativas no projeto de circuitos em que as restrições ocasionadas pela distribuição e consumo de potência do sinal de relógio possam ser evitadas. Além disso, a abordagem GALS favorece o reuso de componentes. Nesta, componentes pré-projetados e pré-validados sobre diferentes domínios de frequência podem garantir que requisitos de projeto como o tempo máximo para que um produto chegue ao mercado, sejam atendidos. Tal realidade reforça a motivação para existir um conjunto de técnicas capazes de permitir a geração e avaliação de tráfego em NoCs que operem sobre diferentes domínios de frequência em um nível mais alto de abstração que aquele descrito na bibliografia atual.

O ambiente ATLAS [TED05], proposto e mantido pelo grupo de pesquisa do Autor dá suporte hoje ao processo de projeto de redes síncronas apenas. Assim, existe uma infraestrutura de software extensa da qual se pode partir para desenvolver e validar técnicas de especificação, verificação e síntese de NoCs não-síncronas. Ainda, uma rede com suporte a SoCs GALS já foi proposta e validada pelo grupo de pesquisa do Autor, e denomina-se Hermes-G [PON08]. Ou seja, à infraestrutura de software disponível (ATLAS) se junta um suporte de hardware já previamente projetado e validado (Hermes-G).

1.2 OBJETIVOS DO TRABALHO

Como objetivo geral neste trabalho, pretendeu-se desenvolver uma infraestrutura capaz de viabilizar a geração e avaliação da rede GALS HERMES-G. Para que o objetivo geral fosse atingido, um conjunto de objetivos específicos foi desenvolvido. Estes consistem na inserção de suporte na ferramenta ATLAS para:

- ❖ A geração automatizada de instâncias quaisquer de redes GALS Hermes-G.
- ❖ A geração de tráfego sintético para redes GALS Hermes-G.
- ❖ A partir de grafos de aplicação do tipo CDCG [MAR05], possibilitar gerar tráfego para redes GALS HERMES-G.
- ❖ Permitir a avaliação de tráfegos sintéticos em redes GALS Hermes-G.
- ❖ Possibilitar a avaliação de tráfegos gerados a partir de grafos CDCG em redes Hermes-G.

1.3 ORGANIZAÇÃO DO DOCUMENTO

O restante do documento é detalhado em 6 capítulos, cada um contendo os seguintes assuntos:

- ❖ Capítulo 2: Apresenta uma revisão do estado da arte da literatura relacionada ao presente trabalho. Também se compara a presente proposta com os trabalhos revisados.
- ❖ Capítulo 3: Descreve em detalhes como é feito o processo de geração automática da rede HERMES-G, uma das contribuições do trabalho.
- ❖ Capítulo 4: Descreve o processo de geração de tráfego sintético e tráfego baseado em aplicações para redes do tipo HERMES-G, outra das contribuições do trabalho.
- ❖ Capítulo 5: Descreve o processo de avaliação de tráfego sintético e tráfego baseado em modelos de aplicações para redes do tipo HERMES-G, mais uma das contribuições.
- ❖ Capítulo 6: Apresenta resultados do trabalho. Nele descrevem-se os experimentos realizados, mostrando os ganhos obtidos em desempenho em redes do tipo HERMES-G. Ainda neste Capítulo, mostra-se o impacto de um tráfego sintético gerado pelo ambiente de geração de tráfego da ATLAS comparado a um tráfego baseado em modelo de aplicação real.

- ❖ Capítulo 7: Apresenta as conclusões e resume os resultados alcançados. Além disso, o Capítulo descreve um conjunto de trabalhos futuros a desenvolver a partir do que foi apresentado neste trabalho.

2 TRABALHOS RELACIONADOS

Este Capítulo aborda um conjunto de trabalhos relacionados aos temas de pesquisa explorados por este trabalho. Dentre os temas pesquisados, está o projeto de redes intrachip capazes de operar em diferentes frequências de operação, e que de alguma forma, sejam passíveis de sofrer parametrização, e que estejam vinculadas a um ambiente de geração automatizada. Outro tema de pesquisa busca esclarecer como se caracteriza tráfego em redes intrachip na literatura disponível. O último tema pesquisado relaciona as métricas de avaliação de tráfego, como são calculadas e como o desempenho de rede é mensurado na literatura. Por fim, faz-se uma compilação e uma comparação dos temas pesquisados com a proposta deste trabalho.

A seguir, será feita a revisão, partindo da seguinte estratégia para cada um dos trabalhos: primeiramente, resume-se o trabalho; a seguir analisa-se o trabalho em questão com ênfase nos temas de pesquisa detalhados no início deste Capítulo; finalmente, conclui-se apresentando os resultados do trabalho em questão.

2.1.1 FEN ET AL.

Fen et al. [FEN07] propõem criar diretrizes para que projetistas de redes intrachip escolham configurações de redes capazes de atender os requisitos de desempenho das aplicações que usam a rede intrachip como meio de comunicação.

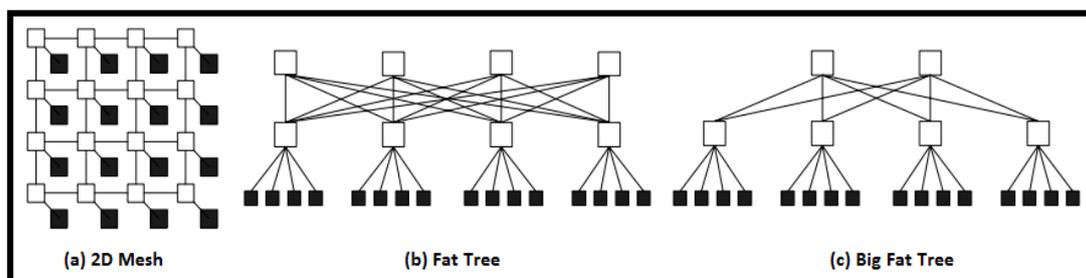


Figura 1: Topologias suportadas pela ferramenta OPNET: (a) 2D Mesh (b) Fat Tree (c) Big Fat Tree.

Os autores utilizam o ambiente OPNET para geração e simulação de redes intrachip. Durante a geração, o autor descreve a possibilidade de gerar redes com topologias malha 2D, árvore gorda e árvore gorda *butterfly*. A Figura 1 ilustra a disposição dos roteadores e dos elementos de processamento em cada uma das topologias possíveis de serem geradas. Para cada topologia, o gerador permite utilizar duas políticas de chaveamento de pacotes, a *wormhole* e a *virtual cut through*. Cada uma das portas dos roteadores faz uso de uma fila, com capacidade máxima de até dois pacotes, sendo que cada pacote possui tamanho igual a 256bits.

O processo de geração do tráfego proposto leva em consideração a distribuição espacial que define a relação entre a origem e o destino dos pacotes e a distribuição temporal que define o intervalo entre a geração dos pacotes do tráfego. Os autores fazem uso de duas distribuições espaciais, a primeira, define os destinos de maneira aleatória, e a segunda, define os destinos do

tráfego seguindo um critério de afinidade, em que os endereços vizinhos mais próximos do endereço gerador têm maior probabilidade de serem os destinos do tráfego. A distribuição temporal não é detalhada, porém, o autor durante a avaliação do trabalho utiliza cenários variados com taxas de injeção definidas, o que caracteriza tráfego com distribuição temporal uniforme.

O processo de avaliação de tráfego utiliza duas métricas de desempenho. A primeira destas é a vazão da rede, caracterizada pela média de bits recebidos pelos elementos de processamento acoplados a esta. A segunda métrica é a latência de pacote, caracterizada pelo intervalo de tempo entre a entrada e a saída de um pacote da rede, tempo este medido em ciclos de relógio da rede.

Por fim, após a realização de um conjunto de experimentos, variando as características de rede propostas pelo trabalho, os autores concluem que a arquitetura de rede com topologia árvore gorda utilizando chaveamento de pacotes *wormhole*, foi aquela capaz de atingir a maior vazão de rede e a menor latência de pacote. Os autores destacam ainda que durante a avaliação dos experimentos propostos, as características impostas ao tráfego foram decisivas na obtenção dos resultados.

2.1.2 KREUTZ ET AL.

Kreutz et al. [KRE05], propõem o desenvolvimento de uma técnica capaz de encontrar a arquitetura de rede intrachip ótima para uma aplicação, onde para isso deva ser levando em consideração o compromisso de latência e consumo mínimo de energia.

Os autores descrevem que a proposta do trabalho faz uso de redes com topologias diretas e indiretas. Nas topologias diretas, o trabalho dá suporte a malha 2D e *toro*, usando roteamento determinístico XY. Na topologia indireta, o trabalho dá suporte a *árvore gorda* usando roteamento adaptativo. Em ambas as topologias são utilizadas as técnicas de chaveamento de pacotes *wormhole* e controle de fluxo baseado em créditos. Os autores não descrevem existir suporte tanto a parametrização quanto de uma ferramenta de geração automatizada de redes.

O processo de geração do tráfego é feito através de modelos. O trabalho faz uso de dois modelos, o primeiro denominado *Application Communication Pattern* (ACP), que define os custos de comunicação e dependências entre as tarefas. O segundo chama-se *Communication Resource Graph* (CRG), que define a arquitetura de comunicação onde a tarefa deve ser mapeada. A Figura 2 ilustra um exemplo da proposta, contendo um modelo ACP que define uma aplicação e um modelo CRG que define a arquitetura de comunicação, juntamente com a aplicação mapeada. No trabalho, os autores propõem a utilização de um algoritmo de mapeamento e minimização de caminhos entre tarefas, denominado *Tabu Search*. Este algoritmo busca heurísticamente o melhor mapeamento possível, procurando encontrar um compromisso entre a menor latência e o menor consumo de energia.

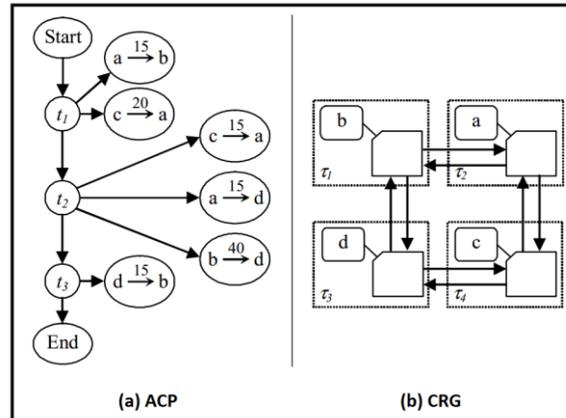


Figura 2: Modelos propostos por Kreutz et al. (a) Modelo ACP, define os custos de computação e dependências entre as tarefas. (b) Modelo CRG, define a arquitetura de comunicação onde a tarefa vai ser mapeada.

O processo de avaliação de tráfego utiliza como métrica de desempenho a latência média gasta por todos os pacotes, medida em ciclos de relógio, e o consumo de energia gasto por uma aplicação para ser executada, medida em micro joules.

Por fim, após a realização de um conjunto de experimentos, variando as características de redes propostas, e usando um conjunto definido de aplicações descritas pela proposta do trabalho, os autores concluem que a rede *árvore gorda* foi a que apresentou a menor latência dentre as redes propostas e as aplicações utilizadas, mas que a topologia malha 2D foi a que apresentou o melhor compromisso entre latência e consumo de energia.

2.1.3 PANADES ET AL.

Panades et al. [PAN06], apresentam uma rede intrachip chamada DSPIN, com suporte a serviços de entrega de pacotes na rede, capazes de sustentar em um fluxo de pacotes um conjunto de restrições como latência máxima e vazão mínima.

A rede proposta pelo autor é projetada como uma arquitetura de rede intrachip voltada para multiprocessadores com memória compartilhada. DSPIN é caracterizada por uma topologia malha 2D, chaveamento de pacotes *wormhole* e algoritmo de roteamento determinístico XY. As principais características da rede DSPIN são (i) O uso de canais virtuais nas portas de entrada dos roteadores, onde cada canal é responsável por transportar uma classe de serviço. A rede proposta dá suporte a dois tipos de pacote, um considerado como de melhor esforço, do inglês *best effort* e outro com garantia de serviço, do inglês *guaranteed service*. (ii) Suporte à técnica GALS, entre roteadores e IPs. Entre roteadores é possível existir defasagem entre as bordas dos relógios dos roteadores, mas todos os relógios devem trabalhar na mesma frequência. Entre os roteadores e IPs a relação é inversa, sendo que é possível os relógios operarem em diferentes frequências, mas devendo estar sincronizados na mesma fase. A rede DSPIN considera um módulo de processamento como um cluster, que podendo ser formados por diversos sub módulos, interconectados por sua rede local. Cada cluster é conectado a um único roteador, que possui dois canais físicos, um para enviar e outro para receber dados da rede. A Figura 3 ilustra uma arquitetura de cluster composta por três núcleos de propriedade intelectual (do inglês, *intellectual*

property core, ou IP ou *IP core*), uma rede local e um adaptador de rede. Durante o processo de avaliação do trabalho, o autor cita algumas variações nas características da rede DSPIN, mas não detalha em nenhum momento se a rede é passível de parametrização, ou da existência um ambiente para geração automática da rede.

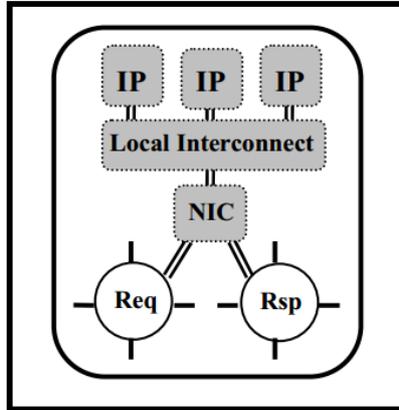


Figura 3: Arquitetura típica de um cluster na rede DSPIN, formada por três IPs, uma arquitetura de interconexão local e um adaptador de rede, que conecta o cluster a um roteador, este formado por duas portas, uma de envio e outro de recebimento de dados.

O processo de geração de tráfego é pouco detalhado, os autores citam durante o processo experimental fazer uso de distribuição espacial aleatória e de distribuição temporal uniforme, variando o comprimento dos pacotes em um intervalo de um a dezesseis *flits*, sendo o *flit*, de tamanho fixo em 34 bits. Não é mencionada a maneira como o comprimento dos pacotes é variado. O processo de avaliação de tráfego utiliza como métrica de desempenho a latência medida em ciclos de relógio. Como processos experimentais são propostos cenários de tráfego com variação nas taxas de injeção. Os autores concluem que durante a utilização do serviço *guaranteed service*, mesmo tráfegos injetados na taxa máxima da rede, não há variação na latência dos pacotes. Em contrapartida, durante a utilização do serviço *best effort*, tráfegos injetados em 25% da capacidade da rede, apresentam variações bruscas no aumento da latência.

2.1.4 OST ET AL.

Em Ost et al. [OST05], os autores apresentam o ambiente MAIA, que possibilita geração e avaliação de redes intrachip. O trabalho apresenta o ambiente e demonstra algumas das funcionalidades existentes.

O ambiente MAIA, possibilita a geração de redes intrachip, modeladas através de *templates* parametrizáveis da rede HERMES. Esta rede dá suporte a diferentes topologias de redes, comprimentos nas filas de entrada dos roteadores, algoritmos de roteamento determinísticos e adaptativos e diferentes técnicas de controle de fluxo. Durante o processo de geração da rede, o projetista pode optar pela geração de interfaces externas a rede, existindo duas possibilidades, a interface nativa da rede ou uma interface *open core protocol* (ou OCP). A Figura 4 ilustra a interface gráfica de usuário de geração de redes do ambiente MAIA, e descreve suas funcionalidades.

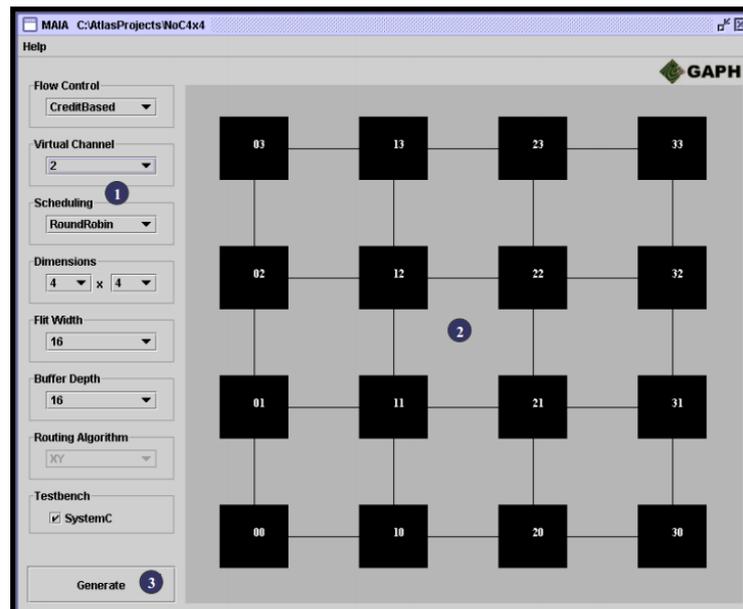


Figura 4: Interface gráfica de geração de redes do ambiente MAIA. (1) Permite o usuário selecionar os parâmetros da rede. (2) Representa uma rede 4x4 Mesh. (3) Dispara o ambiente que gera a rede selecionada.

O processo de geração de tráfego do ambiente MAIA, é feito a partir de um módulo do ambiente chamado *Traffic Generation*, responsável por implementar a interface gráfica que permite parametrizar o tráfego, e por gerar os arquivos que compõem o tráfego a ser transmitido na rede. O gerador permite variar os parâmetros que definem o tráfego, sendo eles, a taxa de injeção do tráfego, o número de pacotes, o tamanho dos pacotes e os destinos do tráfego, que podem ser fixados em um endereço da rede, ou definidos de maneira aleatória.

A avaliação do tráfego é feita a partir de um módulo do ambiente MAIA chamado *Traffic Analysis*, que usa arquivos que coletam informações dos pacotes durante a simulação da rede. O processo de avaliação é feito após o término da simulação. Dentre as métricas de desempenho utilizadas para avaliar o tráfego estão o número de pacotes recebidos, o tempo médio de entrega dos pacotes, o tempo total de entrega dos pacotes e o tempo total de simulação da rede, todos eles calculados em ciclos de relógio da rede.

Durante o processo experimental os autores propõem variar as características de algumas redes disponibilizadas pelo ambiente, obtendo algumas conclusões como que tráfego que contem pacotes menores obtém melhor desempenho em redes com algoritmos de roteamento adaptativos e tráfegos com pacotes maiores obtém maior desempenho em algoritmos de roteamento determinísticos.

2.1.5 PONTES ET AL.

Em Pontes et al. [PON08], propõem o desenvolvimento de dois roteadores GALS, baseados no roteador HERMES, ambos com objetivo de redução no consumo de energia da rede. As redes propostas por este trabalho são: (i) HERMES-G. (ii) HERMES-GLP. Ambos os roteadores são baseados no roteador HERMES, descrito originalmente por [MOR04]. Ambas as redes são compostas pelas seguintes características, roteamento determinístico XY, controle de fluxo

baseado em créditos, chaveamento de pacotes *wormhole* e largura dos canais dos roteadores igual a 16bits.

O principal componente que diferencia as redes propostas da rede HERMES são as filas utilizadas nos roteadores. Na rede HERMES, é feito uso de uma fila síncrona como componente de entrada nos roteadores, onde leituras e escritas entre os roteadores são feitas na mesma fase e na mesma frequência. As redes HERMES-G e HERMES-GLP utilizam uma fila bi síncrona, que permite que escritas e leituras na fila sejam feitas tanto na mesma, como em diferentes frequências, podendo ou não estar na mesma fase. A Figura 5 ilustra a arquitetura da fila, composta por uma memória que armazena os dados. Também existe na arquitetura uma lógica de codificação de ponteiros de leitura e de escrita, que garante que mesmo que ambos os ponteiros utilizados para ler e escrever na fila operem em diferentes domínios de relógio, sempre estejam sincronizados. Além da fila bi síncrona, o roteador HERMES-GLP possui um componente que ajusta a frequência do roteador, do inglês *clock gating* com base em valores de prioridade definidos e implementados através de um campo de prioridade nos pacotes do tráfego.

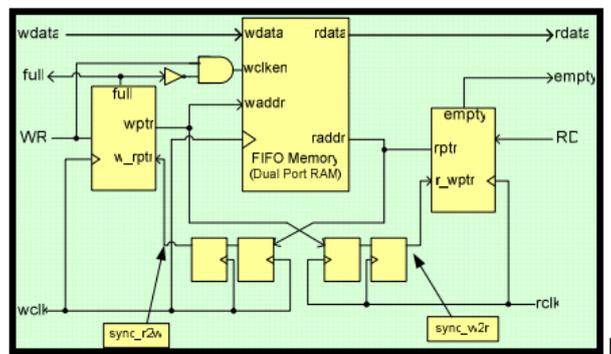


Figura 5: Fila bi síncrona utilizada em redes HERMES-G e HERMES-GLP. Nesta arquitetura de fila, a escrita e a leitura podem ou não operar na mesma fase ou na mesma frequência. A codificação de Gray é utilizada nos ponteiros de leitura e de escrita da fila para garantir que ambos os ponteiros estejam sincronizados, mesmo sendo manipulados em diferentes domínios de fase e de frequência.

O processo de geração de tráfego é pouco detalhado, uma vez que os autores o citam unicamente durante o processo experimental do trabalho. Em resumo, a distribuição espacial do tráfego não é explorada, os autores apenas utilizam tráfego com origens e destinos estáticos. A distribuição temporal é uniforme, usando a taxa máxima de injeção dos IPs. Como avaliação do tráfego, é feito uso da latência média de pacote como métrica de desempenho. Durante o processo experimental, é comparado o tempo adicional ocasionado pelo mecanismo que varia a frequência dos roteadores com base nas prioridades dos pacotes, e quais são as taxas de ativação dos roteadores em uma rede HERMES-GLP. Os resultados demonstram que não houve acréscimo significativo na latência dos pacotes pelo uso do mecanismo que altera a frequência do roteador com base na prioridade nas redes HERMES-GLP. Além disso, os resultados mostram que existem diferenças significativas nas taxas de ativação entre roteadores (sendo esta uma forma de estimar potência de forma grosseira), o que demonstra a principal vantagem da rede HERMES-GLP na redução de energia comparada à rede HERMES-G.

2.1.6 BONONI ET AL.

Bononi et al. [BON07] comparam diferentes arquiteturas de redes intrachip utilizando tráfego sintético e tráfego real, demonstrando o impacto do tráfego e da arquitetura utilizada nos resultados obtidos. Os autores utilizam o ambiente OMNeT++ para geração e simulação da rede. Este ambiente permite gerar diferentes topologias. O trabalho faz uso de quatro topologias, ilustradas pela Figura 6 e descritas a seguir: (i) malha 2D (ii) anel (iii) *Spidergon* (iv) *crossbar*. Em todas as topologias são utilizados algoritmos de roteamento determinísticos mínimos, livres de situações de impasse, do inglês *deadlock*. O simulador utilizado usa precisão em nível de *flit*, onde todos os roteadores se comunicam de maneira síncrona.

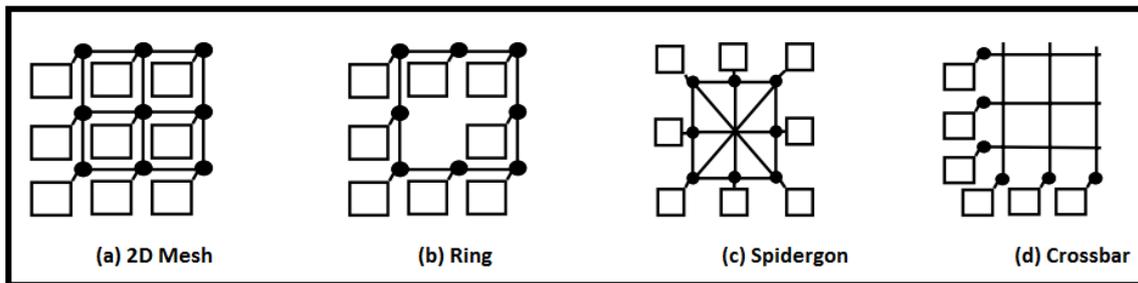


Figura 6: Topologias suportadas pelo ambiente de geração: (a) malha 2D, (b) anel, (c) Spidergon, (d) crossbar.

O processo de geração de tráfego é feito através da ferramenta SCOTCH que a partir de grafos de aplicações, permite modelar tráfego sintético e tráfego real. A ferramenta SCOTCH realiza o particionamento e o mapeamento da aplicação através de dois grafos, um que descreve a aplicação e outro que descreve a topologia e as demais características da rede, e leva em consideração a melhor combinação possível que explore o melhor desempenho da rede.

O processo de avaliação do tráfego utiliza como métrica de desempenho o tempo de processamento de uma aplicação real em ciclos de relógio, e a vazão média da rede. Durante o processo experimental, os autores avaliam todas as topologias de rede propostas sobre diferentes cenários de aplicações sintéticas, onde o modelo de particionamento e mapeamento das tarefas é variado. Além disso, os autores fazem uso de um cenário de aplicação real MPEG, do inglês *Moving Picture Experts Group* utilizando diferentes topologias de rede propostas. Como resultados os autores concluem que a topologia de rede *crossbar* foi a que atingiu melhor desempenho dentre as demais topologias comparadas. Além disso, os autores destacam a importância do particionamento e do mapeamento, sendo eles os critérios mais importantes para obter o melhor desempenho da rede.

2.1.7 TEDESCO ET AL.

Em Tedesco et al. [TED05a], os autores apresentam métodos para geração e avaliação de tráfego em redes intrachip, onde é proposta uma abordagem alternativa de avaliação de desempenho, que leva em consideração os canais que interligam os roteadores, possibilitando avaliar em quais pontos da rede os requisitos do tráfego não estão sendo atendidos.

Os autores utilizam a rede intrachip HERMES em seu trabalho, esta rede, possui topologia malha 2D, suporte a canais virtuais, roteamento determinístico XY e adaptativos *west-first*, chaveamento de pacotes *wormhole*, controle de fluxo baseado em créditos e *handshake* e suporte a parametrização do comprimento dos canais que interligam os roteadores.

O tráfego proposto pelo autor faz uso da distribuição espacial complemento para definir os iniciadores e destinatários do tráfego. O intervalo em que os pacotes são colocados na rede é dado de maneira uniforme. Além da taxa de injeção, o autor permite parametrizar o número de pacotes em *flits* de cada tráfego injetado por um IP.

O processo de avaliação do tráfego é descrito ao longo do trabalho como a principal contribuição. O modelo proposto permite que o tráfego seja avaliado tanto nos pontos de injeção e coleta de dados da rede, caracterizado como “Abordagem Externa” quanto nos canais que interconectam os roteadores, caracterizado como “Abordagem Interna”. Na abordagem externa, os resultados são obtidos nas interfaces externas da rede, já na abordagem interna, é possível obter detalhes entre os canais que interligam os roteadores, e entender em quais pontos, a rede apresenta maior e menor índice de atividade ou ociosidade. Como métricas de avaliação de desempenho, os autores fazem uso de vazão e de latência, tanto para a abordagem externa, quanto para a abordagem interna.

O processo experimental é feito através de cenários de redes HERMES, fazendo uso de algoritmos de roteamento determinísticos e adaptativos, e uso de canais virtuais, que multiplexam um canal físico, o que permite em alguns casos, um melhor aproveitamento da rede. Através da técnica de avaliação interna, o autor demonstra a possibilidade de detectar em quais pontos a rede estava causando a saturação no tempo de transmissão dos pacotes. Além disso, os autores demonstram as vantagens no uso de canais virtuais, capaz de em alguns cenários aumentarem as taxas de tráfego aceito.

Ainda em Tedesco et al. [TED06], os autores desenvolvem um novo trabalho propondo evolução no processo de geração de tráfego, com objetivo de comparar o desempenho de uma rede quando diferentes modelos de tráfegos são utilizados. As questões relacionadas à geração de redes permanecem as mesmas descritas em [TED05a].

O processo de geração de tráfego proposto pelos autores destaca a importância de dirigir a modelagem do tráfego conforme as características de uma aplicação. O trabalho de caracterização do tráfego proposto leva em consideração os requisitos de entrega das aplicações. Os autores propõem um modelo de tráfego capaz de variar os seguintes parâmetros de um tráfego: (i) Tamanho dos pacotes (ii) Intervalo de geração de um pacote (iii) Intervalo de ociosidade entre pacotes. Os autores fazem uso da distribuição de probabilidade normal e Pareto *on-off* para variar os intervalos de geração e ociosidade dos tráfegos.

O ambiente de avaliação de tráfego faz uso das mesmas métricas descritas em [TED05a], sendo que durante o processo experimental, são propostos dois cenários compostos por redes HERMES de tamanho 8x8 com 16bits de largura dos canais entre os roteadores e 8 *flits* de profundidade nas filas de entrada dos roteadores, que operam a 50 MHz. Em ambos os cenários, a distribuição espacial é ilustrada pela Figura 7(c), sendo um tráfego de voz, gerado pelos roteadores

0 e 7 para o roteador 63, e um tráfego de vídeo, gerado do roteador 24 para o roteador 52, e do roteador 39 para o roteador 60. A diferença entre ambos os cenários está na forma como os pacotes são injetados na rede. O primeiro cenário representado pelo tráfego de voz realiza a injeção de pacotes conforme ilustra a Figura 7(a), seguindo as características de tráfego CBR, do inglês “Constant Bit Rate”, onde a taxa de injeção é contínua ao longo do tempo. O segundo cenário representado pelo tráfego de vídeo realiza a injeção de pacotes conforme ilustra a Figura 7(b), seguindo as características de tráfego VBR, do inglês *variable bit rate* onde a taxa de injeção varia entre períodos *on-off*. A mesma quantidade de pacotes foi utilizada em ambos os tráfegos, injetados sob as mesmas taxas de injeção. Como resultados, os autores concluem que o segundo cenário que utiliza o modelo VBR apresentou melhores resultados em relação à variação da latência média de todos os pacotes.

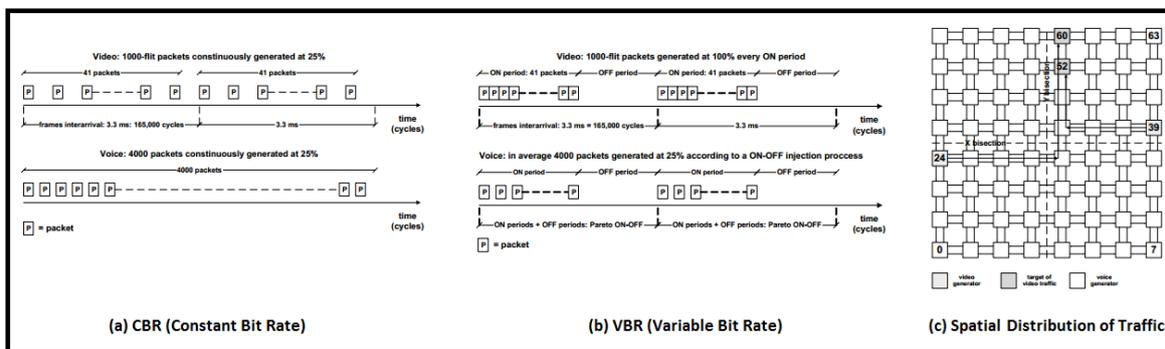


Figura 7: (a) Categoria de serviço CBR do inglês “Constant Bit Rate” em que todos os pacotes são injetados na rede na mesma taxa de injeção. (b) Categoria de serviço VBR, do inglês *variable bit rate* em que a taxa de injeção é variável ao longo do tempo. (c) Distribuição espacial do cenário de tráfego composto por um gerador de vídeo e um gerador de voz.

2.1.8 LIU ET AL.

Liu et al. [LIU07], propõem uma abordagem de caracterização de tráfego em NoCs a partir de aplicações reais. Os objetivos do trabalho demonstram a complexidade envolvida no processo de caracterização do tráfego e nos ganhos da proposta comparadas a distribuições de tráfego sintéticas, amplamente utilizadas por trabalhos relacionados.

Os autores utilizam o ambiente MCSL, que dá suporte a simulação de redes com precisão de ciclo, e dá suporte as topologias malha 2D, toro 2D e árvore gorda. O processo de geração de tráfego, descrito pelo autor como uma metodologia de caracterização de tráfego foi proposto a partir de um conjunto de oito aplicações reais, descritas originalmente para *Multiprocessor Systems on Chip* (MPSoCs). A proposta é dividida em múltiplas partes, que descrevem as características que um tráfego deve assumir. Dentre elas estão o modelo da aplicação e o modelo da arquitetura, ambos descritos através de grafos. O modelo da aplicação descreve o tráfego, no que tangem os custos de comunicação e de computação de uma aplicação. Já o modelo da arquitetura descreve as características desejadas da rede, incluindo os parâmetros dos recursos e a capacidade dos elementos de processamento.

O processo de avaliação de tráfego utiliza como métrica de desempenho a vazão total da rede e os atrasos de todos os pacotes medidos entre o envio e recebimento dos pacotes do tráfego. O processo experimental varia as topologias suportadas pelo trabalho sob diferentes tamanhos de redes, onde tráfego sintético uniforme é comparado com modelos de aplicações descritas através da abordagem proposta. Em geral, os autores concluem avaliando vazão da rede e latência dos pacotes, que o modelo proposto comparado a tráfego sintético uniforme, por ter a capacidade de concentrar tráfego tanto espacialmente quanto temporalmente trouxe maior precisão aos resultados.

2.1.9 HONG ET AL.

Hong et al. [HON08], propõem avaliar o desempenho de uma rede de topologia *toro* com suporte a roteamento *backtracking*, fazendo uso de tráfego sintético orientado a aplicações.

Os autores utilizam o ambiente OMNeT++ para geração e simulação de redes intrachip. O ambiente utilizado possui licença pública, é desenvolvido em C++ e dá suporte a interface gráfica. A rede utilizada possui como características, topologia torp 2D, 32bits de largura de dados entre os roteadores e algoritmo de roteamento *backtracking*. O roteamento segue a ideia do menor caminho entre a origem e o destino do tráfego, possuindo protocolo de chaveamento de pacotes de três fases, sendo elas: (i) Chaveamento do circuito (ii) Transmissão do pacote (iii) Liberação do circuito. O chaveamento é feito através de um pacote que contém o destino do tráfego, que pode ou não assumir um caminho alternativo ao menor caminho caso encontre bloqueios durante o chaveamento do circuito. A transmissão dos pacotes ocorre *flit a flit*, já a liberação do circuito é feita através de um pacote que libera o caminho.

O processo de geração de tráfego dá suporte a tráfego sintético dirigido a aplicações. Pacotes são gerados conforme três parâmetros, sendo eles a distribuição espacial, a distribuição do intervalo de geração entre os pacotes e o tamanho dos pacotes. A distribuição espacial, que descreve a relação entre a origem e o destino dos pacotes, é variada de três maneiras, sendo elas: (i) Uniforme (ii) *Locality* (iii) *Transpose*. Na maneira uniforme, a quantidade total de pacotes é distribuída de maneira proporcional entre todos os destinos disponíveis para receber tráfego na rede. Na distribuição *Locality*, os pacotes são distribuídos de maneira uniforme, porém, com uma relação de afinidade entre os vizinhos próximos a origem do tráfego. Na distribuição *Transpose*, o destino do tráfego é a sua matriz transposta do endereço origem do tráfego. A distribuição temporal é feita através de uma distribuição de probabilidade Poisson onde a taxa de injeção dos pacotes é fixa e o tempo de intervalo entre pacotes consecutivos é variado. A distribuição de tamanho de pacote é parametrizável, podendo ser feita de três formas, sendo elas de tamanho de pacote curto, médio e longo.

O processo de avaliação de tráfego utiliza como métrica de desempenho o atraso médio de pacotes medido em ciclos de relógio e a vazão média da rede. O processo experimental é feito em uma rede toro 2D com dimensões 4x4. No modelo de tráfego proposto, cada IP envia 10.000 pacotes, variados em três tamanhos de pacotes, sendo o curto de 128bytes, o médio de 512bytes e o longo de 2048bytes. Como resultados, os autores concluem que o fator de localidade do

tráfego trouxe ganhos no atraso e na vazão dos pacotes, o que indica um possível ganho de desempenho quando uma estratégia de mapeamento é utilizada. Além disso, o autor detectou que a distribuição espacial uniforme foi aquela que atingiu os menores resultados envolvendo saturação do tráfego, quando comparadas a distribuições espaciais *Locality* e *Transpose*.

2.1.10 BRUCH ET AL.

Bruch et al. [BRU09], apresentam um ambiente que permite avaliar redes intrachip projetadas em nível de transferência entre registradores (RTL) como no nível de transação (TL), sobre diferentes configurações, utilizando ferramentas com suporte a interface gráfica.

O ambiente proposto, referenciado por BrownPepper, implementado com uso de interface gráfica, e de domínio público, permite a geração parametrizável da rede SoCIN, acrônimo de *SoC Interconnection Network*, que possui topologia malha 2D, e composta pelo roteador ParIS, acrônimo de (*Parameterizable Interconnect Network*), formado por 5 portas, onde 4 delas são responsáveis pela comunicação entre os roteadores e uma delas responsável pela comunicação do roteador com um módulo IP. Cada porta do roteador possui dois canais, um responsável por receber e outro pelo envio dos dados. A rede SoCIN permite a transmissão de pacotes de tamanhos ilimitados.

O processo de geração de tráfego é implementado em um dos módulos do ambiente BrownPepper através de um ambiente parametrizável de configuração de tráfego utilizando interface gráfica. O modelo de tráfego proposto segue o mesmo definido em [TED05], com acréscimo de um novo campo aos parâmetros do tráfego, que o classifica conforme sua qualidade de serviço (do inglês *quality of service* ou QoS). Através deste valor é possível avaliar se um requisito do tráfego, como um tempo máximo de atraso, por exemplo, foi atendido.

O processo de avaliação de tráfego utiliza como métrica de desempenho a latência média da rede, medida em ciclos de relógio, e o percentual de pacotes que tiveram seu tempo de transmissão ideal atendidos. O ambiente de avaliação é disponibilizado através de uma interface gráfica. Durante o processo experimental o autor avalia dois cenários de tráfego, que demonstram as funcionalidades do ambiente de maneira geral, demonstrando que modificando certos parâmetros da rede, é possível atender restrições impostas pelos tráfegos gerados.

2.1.11 SCHEMMER ET AL.

Schemmer et al. [SCH10], apresentam uma proposta para estender o gerador de tráfego do ambiente ATLAS, utilizando a distribuição de probabilidade exponencial decrescente para variar a distribuição temporal de um tráfego de pacotes, que define o intervalo entre a transmissão dos pacotes de um tráfego.

Os autores fazem uso da rede HERMES e da ferramenta de geração de redes presente no ambiente ATLAS, que permite gerar variações da rede HERMES, variando suas dimensões, profundidade das filas nas portas dos roteadores, largura dos canais e número de canais virtuais. Além disso, o ambiente possibilita a seleção de um dentre sete algoritmos de roteamento, sendo 6

deles adaptativos e um determinístico, todos livres de situação de impasse (em inglês *deadlock free*).

O processo de geração de tráfego permite variar os parâmetros deste, no que diz respeito ao número de pacotes, tamanho dos pacotes, destinos do tráfego e intervalos entre a geração dos pacotes. Os destinatários dos pacotes do tráfego podem ser parametrizados de três maneiras: (i) Endereço destino único: Todos os pacotes de um tráfego são enviados para um único destino; (ii) Complemento: Todos os pacotes de um tráfego são enviados para endereço binário negado; (iii) Aleatório: Os destinos são escolhidos de maneira aleatória pelo gerador de tráfego. O intervalo de injeção entre os pacotes segue uma distribuição exponencial decrescente, calculado entre um intervalo de taxas de injeção, passível de parametrização. A Figura 8 descreve através de um exemplo uma das interfaces do ambiente de geração, que permite a visualização gráfica da distribuição dos pacotes para um intervalo informado. No exemplo ilustrado pela figura, é proposto um tráfego de 1000 pacotes variados em um intervalo de 100 Mbps a 200 Mbps variados em incrementos de 10 Mbps, com valor de média que define a intensidade do decaimento da distribuição definido em 151 Mbps.

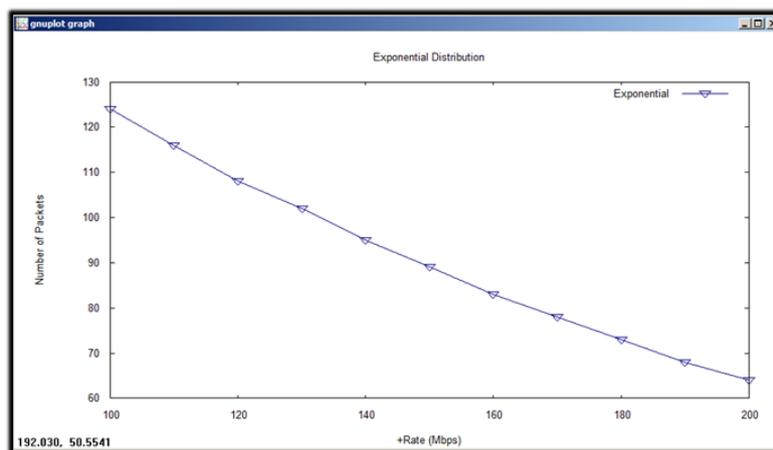


Figura 8: Curva exponencial decrescente de 1000 pacotes gerados em um intervalo de 100 Mbps a 200 Mbps com incremento de 10 Mbps e média de 151 Mbps.

O processo de avaliação do tráfego utiliza a mesma abordagem descrita por [TED05]. Como métricas de avaliação de desempenho, é feito uso de vazão e de latência. Durante o processo experimental os autores comparam a distribuição temporal de tráfego exponencial decrescente proposta com uma distribuição normal já existente no ambiente. Dois fluxos de 500 pacotes são propostos, um do roteador 00 para o roteador 22, e um do roteador 20 para o roteador 22, ambos concorrentes pelo mesmo caminho da rede. O primeiro tráfego (00-22) é gerado de maneira uniforme, o segundo tráfego (20-22) é gerado em um cenário através de uma distribuição normal, e em outro cenário através de uma distribuição exponencial. O objetivo do experimento é avaliar a capacidade de saturação da distribuição exponencial comparada a distribuição normal, sendo que ambas enviam a mesma quantidade de informação, em intervalos de tempo correlatos para os mesmos destinos.

Como resultados, os autores detectam existir 20% de aumento na latência média do tráfego uniforme (00-22) utilizando tráfego exponencial (20-22), quando comparado a um mesmo cenário,

que utiliza de tráfego normal (20-22). A vazão por outro lado diminuiu 9% no tráfego uniforme utilizando tráfego exponencial, quando comparado a tráfego normal. Por fim, o tráfego exponencial apresenta melhores resultados com relação a capacidade de saturar a vazão e a latência dos pacotes, comparado a tráfego normal.

2.1.12 GRATZ ET AL.

Gratz et al. [GRA06], apresentam o projeto, implementação e a avaliação da rede intrachip TRIPS OCN, proposta como uma rede intrachip para interconexão de processadores.

Os autores propõem o desenvolvimento da rede TRIPS OCN, que contem como características, tamanho 4x10, topologia malha 2D, chaveamento de pacotes *wormhole*, 4 canais virtuais em cada uma das portas dos roteadores, formados por 5 portas, sendo 4 delas para comunicação entre roteadores e uma delas para comunicação da rede componentes.

O processo de geração de tráfego propõe o uso de tráfego sintético e o uso de aplicações reais. O tráfego sintético é modelado utilizando distribuições *bit complement* e *random* que variam os destinos dos pacotes, já o tempo de injeção dos pacotes é dado por uma distribuição uniforme. Os autores utilizam o SPEC CPU2000 para extrair as aplicações reais, sendo utilizadas 20 aplicações no total.

O processo de avaliação do tráfego utiliza como métrica de desempenho a latência média dos pacotes, medida em ciclos de relógio e a vazão aceita dos tráfegos. Durante o processo experimental, os autores comparam os resultados obtidos entre tráfego sintético e as aplicações reais. Concluem então que o tráfego sintético, além de não reproduzir o mesmo universo da aplicação, no que diz respeito ao momento e a quantidade de informação injetada, não reproduz resultados de saturação próximos da aplicação. Estes, em relação aos cenários avaliados, são justamente os causadores de aumento na latência média dos pacotes, e redução na vazão total da rede.

2.1.13 POSICIONAMENTO DO TRABALHO EM RELAÇÃO AO ESTADO DA ARTE

Esta Seção resume alguns dos assuntos em comum propostos por este trabalho, coexistentes com os trabalhos pesquisados. A seguir, são feitas comparações, justificando através destas a proposta deste trabalho.

A Tabela 1 apresenta cinco aspectos coletados dos trabalhos pesquisados seguindo o tema de projeto e geração parametrizável de redes intrachip. Os itens coletados partem dos seguintes questionamentos: (i) Trabalho descreve uma ferramenta de geração automática de rede intrachip? (ii) Processo de geração de redes é passível de parametrização? (iii) Quais topologias são suportadas pela rede? (iv) Quais algoritmos de roteamento são suportados pela rede? (v) Rede utilizada é síncrona ou não síncrona? (vi) Qual nível de abstração de projeto a rede é descrita?

Tabela 1: Resumo do estado da arte nos temas de projeto, e geração de redes intrachip.

Autor/Ano	Objetivo	Ferramenta de Geração de Rede Intrachip	Suporte a Geração Parametrizável	Topologias Suportadas	Roteamento Suportado	Rede Síncrona ou Não Síncrona (GALS)	Nível de Abstração do Projeto da Rede Intrachip
Fen et al. [FEN07]	Apresentam diretrizes para gerar redes voltadas para requisitos de aplicações	Ferramenta OPNET		Malha 2D / Toro 2D / Árvore gorda			
Kreutz et al. [KRE05]	Técnica para encontrar arquitetura ótima de rede			Malha 2D / Toro 2D / Árvore gorda	Determinístico XY/Adaptativo		
Panades et al. [PAN06]	Apresenta rede DSPIN com suporte a serviços de rede			Malha 2D	Determinístico XY	Não Síncrona (GALS)	
Ost et al. [OST05]	Apresentam ambiente MAIA para geração e avaliação de redes	Ferramenta MAIA	Sim	Malha 2D	Determinístico XY/Adaptativos West-First, North-Last, Negative-First	Síncrona	RTL (VHDL)
Pontes et al. [PON08]	Apresentam dois roteadores GALS <i>Low Power</i>			Malha 2D	Determinístico XY	Não Síncrona (GALS)	RTL (VHDL)
Bononi et al. [BON07]	Comparam tráfego real com tráfego sintético	Ferramenta OMNeT++	Sim	Malha 2D / Anel/ Spidergon/ <i>crossbar</i>	Determinístico	Síncrona	
Tedesco et al. [TED05a][TED06]	Apresentam ambiente de geração e avaliação de tráfego			Malha 2D	Determinístico XY/Adaptativo West-First	Síncrona	RTL (VHDL)
Liu et al. [LIU07]	Apresentam abordagem de caracterização de tráfego			Malha 2D / Toro 2D / Árvore gorda			
Hong et al. [HON08]	Avaliam o desempenho de uma rede com roteamento <i>backtracking</i>	Ferramenta OMNeT++	Sim	Toro 2D	Adaptativo <i>backtracking</i>		
Brunch e al. [BRU09]	Apresentam um ambiente de avaliação de redes	Ferramenta BrownPepper	Sim	Malha 2D	Determinístico XY	Síncrona	RTL(SystemC) e TL(SystemC)
Schemmer et al. [SCH10]	Apresentam proposta de tráfego sintético	Ferramenta Atlas	Sim	Malha 2D / Toro 2D	Determinístico XY	Síncrona	RTL (VHDL)
Gratz et al. [GRA06]	Apresentam o projeto de uma rede intrachip para MPSoCs			Malha 2D		Síncrona	RTL (VHDL)
Este trabalho	Introduz um ambiente de geração parametrizável de redes GALS baseadas na rede HERMES-G	Ferramenta Atlas com suporte a geração de redes HERMES-G	Sim	Malha 2D	Determinístico XY/ Adaptativos West-First, West First Minimal, West-First Non Minimal, North-Last, North-Last Minimal, North- Last Non Minimal	Síncrona e não síncrona (GALS)	RTL (VHDL)

A partir da coleta dos dados, constata-se que nos itens ferramentas de geração de redes e suporte a geração parametrizável, entende-se a existência de cinco ferramentas, onde somente em uma delas os autores não descrevem existir suporte a parametrização durante a geração da rede. Dentre as topologias utilizadas, com exceção de um, todos os outros fazem uso da topologia malha 2D, além disso, quase todos dão suporte a topologia *toro*. Com relação ao tipo de roteamento suportado, a maior parte faz uso de roteamento determinístico. Com relação ao nível de abstração do projeto da rede, na grande maioria dos trabalhos é feito uso do *Register Transfer Level* (RTL), utilizando linguagens como Verilog ou VHDL para descrever a rede. O último item que pesquisa se as redes utilizadas são síncronas ou não síncronas, demonstra que a grande maioria dos trabalhos pesquisados faz uso de redes síncronas, com exceção de dois deles.

No contexto deste trabalho é apresentada uma ferramenta de geração parametrizável de redes com suporte a definição de múltiplos domínios de frequência para os roteadores, descritos em nível de projeto RTL utilizando linguagem VHDL. Conforme o levantamento feito, nenhum dos trabalhos pesquisados apresenta esta proposta. A rede utilizada por este trabalho faz uso da topologia Malha 2D seguindo a tendência utilizada pela maioria dos trabalhos pesquisados. Por fim, a maior parte dos trabalhos faz uso de algoritmos de roteamento determinísticos, este trabalho, além do uso de algoritmos de um roteamento determinístico, dá suporte também a seis algoritmos de roteamento adaptativos, suportados somente por alguns dos trabalhos pesquisados.

A Tabela 2 apresenta sete aspectos coletados dos trabalhos pesquisados seguindo os temas de caracterização de tráfego de rede e métricas de avaliação de tráfego utilizadas. Os itens coletados partem dos seguintes questionamentos: (i) Trabalho descreve ferramenta de geração de tráfego? (ii) Processo de geração de tráfego é passível de parametrização? (iii) Trabalho utiliza grafo de aplicação para descrever tráfego? (iv) Se grafo for utilizado, trabalho descreve alguma técnica de particionamento ou de mapeamento? (v) Qual é a distribuição espacial do tráfego utilizada pelo trabalho? (vi) Qual é a distribuição temporal do tráfego utilizada pelo trabalho? (vii) Quais são as métricas de avaliação de desempenho utilizadas? .

A partir da coleta dos dados, constata-se que no item ferramenta de geração de tráfego com suporte a parametrização, somente alguns trabalhos detalham existir uma ferramenta apta a possibilitar geração de tráfego parametrizável. Em alguns trabalhos, foi detectado o uso de grafos de aplicações para descrever tráfego, sendo que somente em um deles os autores descrevem a existência de uma técnica de particionamento e mapeamento utilizada para sintetizar o grafo que descreve o tráfego a arquitetura da rede. Com relação a distribuição temporal, grande parte dos trabalhos utiliza tráfego aleatório e complemento, existindo em alguns deles, uma proposta evolutiva ao modelo aleatório, em que propriedades estatísticas são utilizadas na definição dos destinos do tráfego. Já o item distribuição temporal, a grande maioria os autores utilizam distribuição uniforme, existindo alguns casos em que proposto o uso de modelos estatísticos como uma proposta evolutiva para aumentar a precisão do tráfego. Por fim, o item avaliação de desempenho, resume as métricas utilizadas para avaliar o tráfego na rede, sendo que em grande maioria, os trabalhos utilizam a vazão da rede e a latência média dos pacotes, avaliadas sobre diferentes óticas durante a injeção, transmissão e recebimento dos pacotes.

No contexto deste trabalho, é apresentada uma ferramenta de geração de tráfego parametrizável, que possibilita variar o tráfego, com relação a quantidade e ao tamanho dos pacotes, e varia a questão espacial que define os destinos dos pacotes e a questão temporal que define o momento de injeção dos pacotes, utilizando distribuições estatísticas. Além disso, este trabalho dá suporte a geração de tráfego através de grafos de aplicações, que além de permitirem caracterizar a questão espacial e temporal dos pacotes, permitem modelar as dependências existentes entre os tráfegos, que de certa forma caracterizam um maior grau de aproximação ao comportamento de uma aplicação. Conforme o levantamento feito, nenhum dos trabalhos pesquisados descreve um ambiente de geração de tráfego parametrizável, capaz de modelar as características do tráfego conforme é proposto. Por fim, este trabalho apresenta uma ferramenta de avaliação de tráfego, com suporte a interface gráfica, que faz uso das métricas de vazão e

latência conforme os demais trabalhos pesquisados. Esta ferramenta leva em consideração o cálculo da latência e da vazão ideal e calcula, levando em consideração quando uma rede intrachip não síncrona é utilizada, uma vez que este trabalho originalmente dá suporte a geração de redes não síncronas. As particularidades referentes a estes detalhes, e a proposta integral descrita até então serão detalhadas em profundidade nos próximos capítulos.

Tabela 2: Resumo do estado da arte nos temas de caracterização e geração de tráfego para redes intrachip.

Autor/Ano	Objetivo	Ferramenta de Geração de tráfego	Suporte a geração de tráfego Parametrizável	Utiliza grafos para descrever tráfego	Descreve técnica de particionamento/ Mapeamento	Distribuição espacial	Distribuição temporal	Avaliação de desempenho
Fen et al. [FEN07]	Apresentam diretrizes para gerar redes voltadas para requisitos de aplicações	_____	_____	_____	_____	Aleatório / <i>Locality</i>	Uniforme	Vazão média da rede/ Latência de pacote
Kreutz et al. [KRE05]	Técnica para encontrar arquitetura ótima de rede	Sim	Sim	Sim	Sim Tabu Search	_____	_____	Latência média dos pacotes
Panades et al. [PAN06]	Apresenta rede DSPIN com suporte a serviços de rede	_____	_____	_____	_____	Aleatório	Uniforme	Latência total do tráfego
Ost et al. [OST05]	Apresentam ambiente MAIA para geração e avaliação de redes	Ferramenta <i>Traffic Generation</i>	Sim	_____	_____	Aleatório	Uniforme	Tempo médio de entrega dos pacotes/ Tempo de entrega dos pacotes
Pontes et al. [PON08]	Apresentam dois roteadores GALS Low Power	_____	_____	_____	_____	Destino fixo	Uniforme	Latência média de pacote
Bononi et al. [BON07]	Comparam tráfego real com tráfego sintético	Ferramenta <i>Scotch</i>	Sim	Sim	_____	_____	_____	Vazão média da rede
Tedesco et al. [TED05][TED06]	Apresentam ambiente de geração e avaliação de tráfego	Ferramenta <i>Traffic Generation</i>	Sim	_____	_____	Destino fixo/ Complemento/ Aleatório	Uniforme/ Normal/ Pareto on-off	Latência média de pacote/Vazão média da rede
Liu et al. [LIU07]	Apresentam abordagem de caracterização de tráfego	_____	_____	Sim	_____	_____	_____	Vazão da rede/ Latência dos pacotes
Hong et al. [HON08]	Avaliam o desempenho de uma rede com roteamento <i>backtracking</i>	_____	_____	_____	_____	Uniforme/ <i>Locality</i> / <i>Transpose</i>	Poisson	Latência média dos pacotes/ Vazão da rede
Brunch e al. [BRU09]	Apresentam um ambiente de avaliação de redes	Sim	Sim	_____	_____	Destino fixo/ Complemento/ Aleatório	Uniforme/ Normal/ Pareto on-off	Latência média da rede
Schemmer et al. [SCH10]	Apresentam proposta de tráfego sintético	Ferramenta <i>Traffic Generation</i>	Sim	_____	_____	Destino Fixo/ Complemento/ Aleatório	Exponencial	Vazão da rede/ Latência dos pacotes
Gratz et al. [GRA06]	Apresentam o projeto de uma rede intrachip para MPSoCs	_____	_____	_____	_____	<i>bit complement</i> / Aleatório	Uniforme	Latência média dos pacotes/ Vazão aceita da rede
Este trabalho	Introduz um ambiente de geração parametrizável de redes GALS baseadas na rede HERMES-G	Ferramenta Atlas com geração e avaliação de tráfego	Sim	Sim	Sim "Tabu Search"	Destino Fixo/ Complemento/ Aleatório	Uniforme/ Normal/ Pareto on-off/ Exponencial	Latência média dos pacotes/ Vazão média dos pacotes

3 GERAÇÃO DE REDES INTRACHIP NÃO SÍNCRONAS

Este Capítulo descreve o processo de parametrização e projeto da ferramenta de geração de redes GALS, baseadas na arquitetura original da rede HERMES-G. Ele descreve quais foram as dificuldades e as decisões tomadas durante o desenvolvimento do gerador. Este gerador foi completamente integrado ao fluxo de projeto de geração do ambiente ATLAS. Na visão do autor, este Capítulo é considerado com uma das contribuições apresentadas por este trabalho.

3.1 ESTUDO INICIAL DO AMBIENTE ATLAS E DA REDE HERMES-G

Esta Seção descreve de através de uma abordagem cronológica quais foram as primeiras atividades desenvolvidas, para obter e entender a ferramenta ATLAS e a rede HERMES-G, para então dar início a parametrização da rede e desenvolvimento da ferramenta de geração. A seguir, serão detalhadas as principais características do processo de geração de redes no ambiente ATLAS, que influenciaram no projeto do gerador de redes não síncronas. Logo após é detalhado como a rede HERMES-G foi obtida, e algumas das principais características e funcionalidades adicionadas durante o desenvolvimento deste trabalho a rede.

3.1.1 PRIMEIRO CONTATO COM O AMBIENTE ATLAS

O ambiente ATLAS foi proposto pelo grupo de pesquisa GAPH, ou grupo de apoio ao projeto de hardware, inicialmente com objetivo de integrar as ferramentas de geração e avaliação de tráfego e de energia para redes do tipo HERMES. Hoje, este ambiente dá suporte à geração e avaliação de tráfego para outros seis tipos de redes, originalmente concebidas como uma evolução da rede HERMES, ou levando em consideração algumas das características utilizadas por esta rede. Atualmente o ambiente ATLAS é mantido sobre controle de versionamento via uso do software SVN. Todas as contribuições propostas por este trabalho estão mantidas em um dos inúmeros diretórios ramo (em inglês *branch*), derivados a partir do diretório tronco (em inglês *trunk*) do ambiente ATLAS. Para este trabalho foi criado um ramo derivado do tronco, chamado GALS. Todas as contribuições desenvolvidas por este trabalho estão disponíveis publicamente em <https://corfu.pucrs.br/svn/atlas/branches/gals>, incluindo o gerador de redes não síncronas apresentado por este capítulo.

Após obter uma versão estável do ambiente ATLAS, o autor deste trabalho propôs um estudo, com objetivo de entender como o processo de geração era feito. A saber, o código fonte que compõe o ambiente ATLAS não possui documentação, tanto no que tange o código fonte, quanto a suas funcionalidades, o que encarece o processo de aprendizagem para o desenvolvimento de novas funcionalidades. Sendo assim, a atividade de estudar o ambiente antes de projetar uma nova funcionalidade acaba se tornando algo necessário. O ambiente ATLAS dá suporte a geração de sete tipos de redes, sendo elas: (I) HERMES; (II) HERMES-TB; (III) HERMES-TU; (IV) HERMES-SR;

(V) HERMES-CRC; (VI) Mercury. Como resultados do estudo do ambiente de geração, o autor deste trabalho concluiu que:

- ❖ Cada uma das redes suportadas pelo processo de geração de redes do ambiente ATLAS possui seu próprio ambiente de geração, incluindo desde a interface gráfica do gerador até os arquivos que geram a rede intrachip.
- ❖ Todas as redes seguem um padrão, no que diz respeito aos nomes dos diretórios que armazenam o código fonte dos geradores e aos diretórios que armazenam os modelos das redes, utilizados durante o processo de geração de rede.
- ❖ Todas as redes propostas seguem um fluxo de projeto ilustrado pela Figura 9 durante a definição de um projeto e geração da rede, composto por quatro etapas: (1) Criação de um projeto de rede; (2) Definição das características da rede; (3) Geração da rede. (4) Dar suporte a edição de um projeto de rede gerada.
- ❖ Toda rede é vista no ambiente como um modelo universal, durante o processo de geração da rede, os modelos são lidos, manipulados e gerados conforme as características selecionadas pelo usuário.

Partindo destes conhecimentos, o novo ambiente de geração proposto, irá seguir o mesmo modelo de diretórios e nomenclaturas utilizadas para armazenar os arquivos que compõem o ambiente de geração e os modelos da rede. Além disso, o ambiente de geração de redes HERMES-G também irá dar suporte a edição de um projeto de rede.

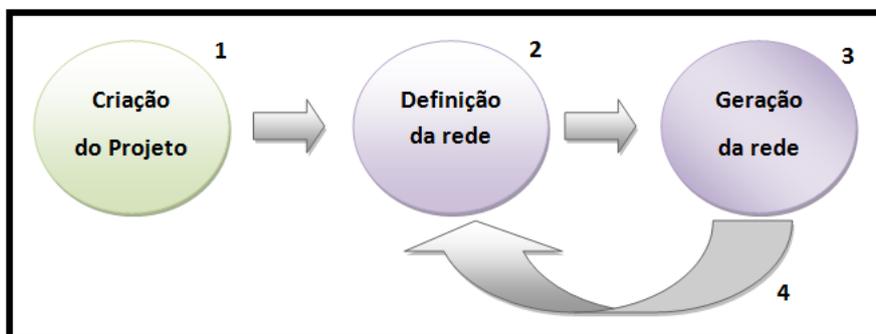


Figura 9: Fluxo de projeto de geração de uma rede no ambiente ATLAS, composto por quatro passos (1) Criação de um projeto. (2) Definição das características desejadas para uma rede. (3) Geração da rede. (4) Edição de um projeto de rede gerado.

3.1.2 PRIMEIRO CONTATO COM A REDE HERMES-G

De acordo com [PON08], a rede HERMES-G é uma extensão da rede HERMES, proposta originalmente por [MOR04]. Ambas as redes, são formadas pelas seguintes características, topologia de interconexão malha bidimensional, controle de fluxo utilizando créditos, algoritmo de roteamento determinístico XY e chaveamento de pacotes utilizando a técnica wormhole. A principal diferença entre uma rede HERMES de uma rede HERMES-G, está nas filas de entrada dos roteadores. Na rede HERMES, é feito uso de uma fila síncrona, onde a frequência de escrita e a frequência de leitura são feitas sempre, na mesma frequência e na mesma fase. Na rede HERMES-

G, é feito uso de uma fila bi síncrona, que possibilita que leituras e escritas, sejam realizadas, tanto na mesma fase e na mesma frequência, como em fases e frequências diferentes.

Em uma rede HERMES-G, o principal elemento que compõe a rede é o roteador. Uma rede é composta por diversos roteadores. Cada roteador é composto por três componentes fundamentais. O primeiro deles, chamado de controlador de chaveamento, do inglês *switch control*, é o responsável por realizar a arbitragem das portas de entrada do roteador, e pelo cálculo do endereço destino de um pacote. O segundo componente, também presente em todas as chaves, chamado de barra cruzada, do inglês *crossbar*, é o responsável por interconectar todas as portas de entrada com todas as portas de saída do roteador. O terceiro e último componente, presente em todas as chaves, chamado de fila de entrada, do inglês *buffer*, é responsável por dar suporte duas funcionalidades em redes HERMES-G. A primeira delas é garantir que independente a fase, ou a frequência em que o roteador estiver operando, em relação ao roteador ou elemento de processamento ao qual a porta do roteador estiver conectada, a fila presente nesta porta irá garantir que a comunicação ocorra da mesma forma. A segunda funcionalidade é realizar o armazenamento dos *flits* dos pacotes de maneira temporária, quando houver contenção na rede, causado pela ocupação do caminho por um fluxo de pacotes concorrente durante o chaveamento da conexão.

Este trabalho obteve acesso a um projeto de rede HERMES-G, com características definidas, tamanho 3x3, largura de canais igual a 16bits e profundidade das filas de entrada dos roteadores igual a 4*flits* de 16bits cada. Para este projeto foi feito um estudo dirigido, com objetivo de identificar quais eram as estruturas que formavam a rede. Durante este estudo os arquivos da rede foram documentados. Além disso, executou-se um conjunto de experimentos variando algumas das características da rede, para validar o funcionamento da rede. Para este projeto foi criado no mesmo servidor de SVN descrito na Seção anterior um diretório para armazenar estes arquivos. A rede HERMES-G parametrizada e inserida no ambiente ATLAS parte deste projeto de rede, disponível publicamente em <https://corfu.pucrs.br/svn/hermesg/branches/rschemmer/>.

Antes de descrever com maior grau de detalhamento os componentes que caracterizam uma rede HERMES-G, serão apresentadas algumas atualizações feitas na rede durante o desenvolvimento deste trabalho, umas no sentido de corrigir falhas existentes, outras adicionando novas características a rede. Ao todo, três novas funcionalidades foram adicionadas, e duas correções foram feitas, com objetivo de aumentar a robustez das filas utilizadas nos roteadores. Todas estas modificações feitas na rede são suportadas pelo ambiente de geração de redes HERMES-G.

3.1.2.1 ADIÇÃO DE NOVOS ALGORITMOS DE ROTEAMENTO

A implementação original da rede HERMES-G dá suporte a roteamento determinístico XY, em que qualquer pacote segue por um caminho predeterminado a partir da fonte e do destino do mesmo. Ao longo do desenvolvimento da rede Hermes agregaram-se, entre outros, seis novos algoritmos de roteamento adaptativos para a rede HERMES, como descrito em [MOR10] sendo estes: (I) *West First*; (II) *West First Minimal*; (III) *West First Non Minimal*; (IV) *North Last*; (V) *North*

Last Minimal; (VI) *North Last Non Minimal*, Devido às características de semelhança existentes entre as redes HERMES e HERMES-G, os algoritmos de roteamento HERMES são em geral totalmente compatíveis com a HERMES-G. Este trabalho usa roteamento adaptativo de redes HERMES na rede HERMES-G.

Para garantir o funcionamento destes algoritmos adaptativos, o autor desenvolveu uma ferramenta de testes capaz de gerar, simular e verificar um conjunto arbitrário de redes HERMES e HERMES-G, de maneira automática. Esta ferramenta faz uso dos geradores de rede e de tráfego do ambiente ATLAS e será detalhada posteriormente, tendo sido desenvolvida para ajudar na verificação do funcionamento do gerador de redes HERMES-G.

Um conjunto composto por diversos casos de teste foi proposto para validar os algoritmos adaptativos, nas redes HERMES e HERMES-G, utilizando a dita ferramenta de testes. Durante os experimentos, foram detectados alguns erros em alguns cenários com relação ao roteamento e à transmissão dos tráfegos. Algumas mudanças nos algoritmos adaptativos foram propostas e desenvolvidas por outro membro do grupo de pesquisa do autor. Os mesmos cenários de teste foram novamente executados sob as mudanças propostas, quando então a ferramenta de testes relatou êxito na transmissão dos cenários de tráfegos propostos em todos os cenários testados.

3.1.2.2 MINIMIZAÇÃO DO *CROSSBAR*

Outra funcionalidade adicionada à rede HERMES-G, desenvolvida originalmente para redes HERMES, é a minimização do número de portas de saída do *crossbar* para cada um dos seis algoritmos descritos na Seção anterior. Dependendo o algoritmo de roteamento adaptativo utilizado, algumas portas de entrada nunca irão enviar pacotes para algumas das portas de saída do roteador. A Figura 10 mostra para cada um dos algoritmos adaptativos as curvas permitidas e proibidas para pacotes. Estas foram utilizadas para desenvolver as versões minimizadas de *crossbars* para cada um dos algoritmos conforme descrito pela Seção 3.1.2.1.

Um conjunto de diversos casos de teste foi proposto para validar as minimizações de *crossbars*, nas redes HERMES e HERMES-G, fazendo uso da ferramenta de testes. Para todos os casos propostos, a ferramenta de testes relatou êxito na transmissão dos tráfegos utilizando diferentes algoritmos de roteamento com duas versões de *crossbars* minimizados.

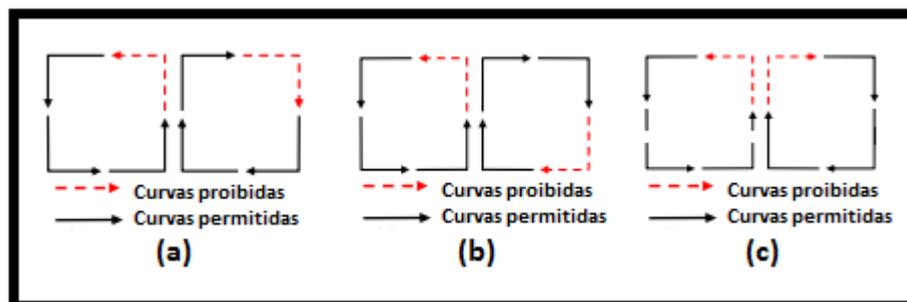


Figura 10: Curvas proibidas e permitidas para os algoritmos adaptativos: (a) Negative First proíbe curvas para Oeste se o pacote está subindo e curvas para o sul se o pacote está indo para a direita. (b) West First proíbe curvas para Oeste. (c) North Last proíbe qualquer curva após o sentido Norte ter sido tomado.

3.1.2.3 MODIFICAÇÃO DA CODIFICAÇÃO DE PONTEIROS DO *BUFFER* DOS ROTEADORES

Durante o desenvolvimento do trabalho de mestrado paralelo ao presente, detectou-se existirem deficiências funcionais no módulo de comparação de ponteiros da fila bi síncrona, componente este utilizado em certas portas de entrada dos roteadores na rede HERMES-G. A Figura 11 ilustra através de um diagrama de blocos os componentes que formam a fila. Dentre eles estão os módulos que implementam os comparadores de ponteiros de fila cheia e fila vazia, utilizados respectivamente como controle de fluxo para escrita na fila, através do sinal de fila cheia e para consumo da fila, através do sinal de fila vazia. Além deles, a fila é composta por um módulo de memória, que armazena os dados na fila, e por uma lógica de comparadores, que sincroniza a leitura e a escrita, ambas podendo ou não ser feitas em diferentes domínios de frequência e de fase. Através desta funcionalidade é possível configurar os roteadores para operar em diferentes domínios de relógio. Para resolver as condições de exceção encontradas na sincronização de ponteiros, a referência [HEC11] descreve a proposta de um módulo comparador utilizando três ponteiros, sendo um de leitura, um de escrita e um de escrita anterior.

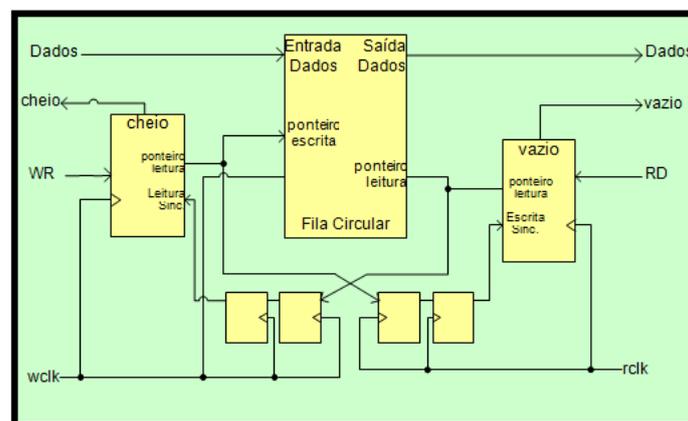


Figura 11: Diagrama de blocos da fila bi síncrona utilizada pela rede HERMES-G.

Além disso, no mesmo trabalho detecta-se também que a codificação Gray utilizada originalmente no comparador de ponteiros pela fila bi síncrona, faz uso de endereços binários, convertido através da operação ou exclusivo (XOR). Esta conversão possibilita a geração de *glitches*, efeitos indesejáveis que podem levar a fila a um estado inválido. Como solução para este problema, propõe-se o desenvolvimento de um novo comparador de ponteiros utilizando codificação Johnson. Esta faz uso de apenas um registrador de deslocamento e uma porta inversora para realizar o processo de incremento dos ponteiros de leitura e de escrita, eliminando a possibilidade de gerar *glitches* durante o incremento. Em [HEC11], o autor ainda desenvolve um estudo do crescimento da área das filas bi síncronas. Para tanto, compara a codificação de ponteiros Johnson com a codificação Gray originalmente existente na fila bi síncrona. Durante os experimentos varia-se a profundidade das filas em 4, 8, 16, 32, 64 e 128 endereços possíveis. Os resultados são obtidos através do processo de síntese para FPGAs, extraíndo o número de LUTs de quatro entradas. Os resultados obtidos apontam que para casos em que 4, 8 e 16 endereços da fila, o número de LUTs gastas por ambas as codificações são aproximadamente iguais. Para

profundidades de endereços de filas de 32,64 e 128 endereços, a codificação Gray, apresenta menor consumo em área que a codificação Johnson.

Por existir ganhos em cada uma das abordagens, o autor deste trabalho adotou a possibilidade de parametrizar a codificação de ponteiros durante a geração de redes HERMES-G, permitindo o uso de arquiteturas de rede com codificação Johnson e codificação Gray.

3.1.2.4 MODIFICAÇÃO DA ARQUITETURA DO COMPONENTE ASYNC_HERMES.VHD

O projeto da rede HERMES-G, por permitir que os roteadores operem em frequências diferentes, cria condições em que os elementos que transmitem e recebem o tráfego da rede necessitem de um componente capaz de sincronizar a transmissão. Com relação ao componente externo transmissor, conectado a uma porta local de um roteador, independente de este operar em uma frequência maior ou menor do que o roteador, a sincronização entre ambos os componentes é garantida pela fila bi síncrona presente na porta local do roteador. Já em um cenário em que componente externo receptor estiver conectado à porta local, tanto operando em uma frequência de leitura maior ou menor que a frequência do roteador, se faz necessário existir um componente sincronizador entre a saída da porta local do roteador, e o componente de recebimento do tráfego. Na visão de um projeto, este sincronizador é visto como um componente necessário para garantir a recepção do tráfego. Porém ele não faz parte dos módulos que compõem a rede.

A proposta original descrita em [PON08] propõe a utilização de um componente denominado *Async_Fifo*, desenvolvido a partir da arquitetura da fila bi síncrona, mas com diferenças no protocolo de leitura da fila. Esta arquitetura compartilha das mesmas deficiências relacionadas à sincronização dos ponteiros detectadas e corrigidas em [HEC11]. Sendo assim, visando corrigir as deficiências existentes, propôs-se o desenvolvimento de uma nova arquitetura *Async_Fifo*, baseada na arquitetura da fila bi síncrona, que dá suporte às correções propostas em [HEC11], desenvolvida unicamente utilizando a codificação de ponteiros Johnson. A Figura 12 ilustra as conexões de entrada e de saída da fila bi síncrona (na Figura 12(b)), e do componente *Async_Fifo* (na Figura 12(a)). Em ambos os casos, o protocolo de entrada das filas é o mesmo, pelo motivo de estarem sempre conectadas às portas dos roteadores. Na Figura 12 é possível visualizar que existem diferenças nas interfaces, e estas se refletem em diferenças no protocolo de saída das filas. Na fila bi síncrona, parte de suas saídas são conectadas ao módulo que realiza o roteamento, e parte delas ao *crossbar*. Já na *Async_Fifo*, a saída da fila possui apenas conexões de transmissão, pelo motivo de não haver necessidade de realizar o roteamento para transmitir os dados.

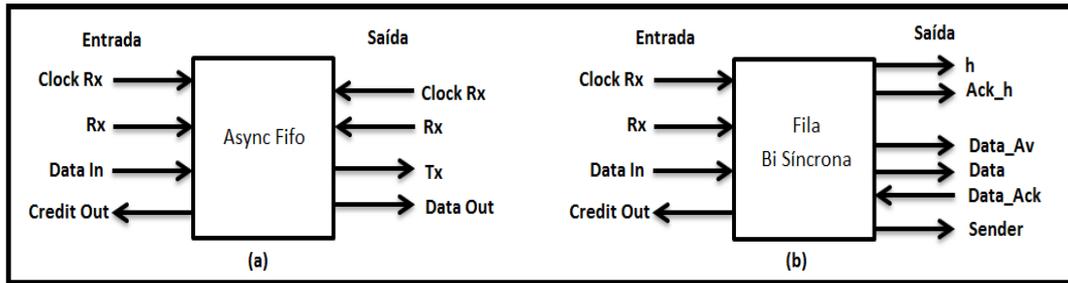


Figura 12: Conexões de entrada e saída das arquiteturas (a) Async_Fifo e (b) fila bi síncrona.

A Figura 13 descreve a máquina de estados de leitura da fila bi síncrona, composta ao todo por cinco estados, sendo o estado S_INIT responsável por inicializar a fila, os estados S_HEADER e S_SHEADER por realizar o roteamento, e S_TX e S_END pela transmissão do pacote. A nova arquitetura Async_Fifo conta com apenas dois destes cinco estados, sendo eles o S_INIT que realiza a transmissão do dado, quando existir algum e S_END, que aguarda e detecta até que exista um dado na fila para ser transmitido.

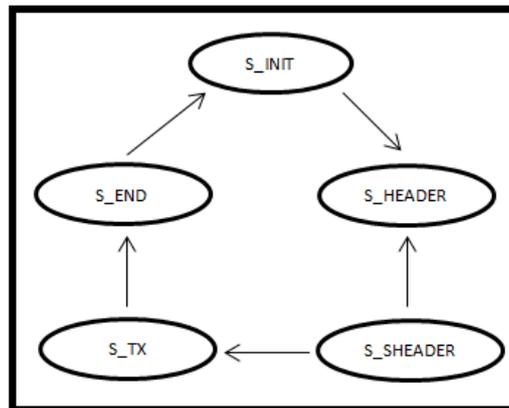


Figura 13: Máquina de estados de leitura da fila bi síncrona, composta por cinco estados, sendo um deles responsável pela inicialização, dois responsáveis pelo roteamento do pacote e dois pela transmissão dos *flits* do pacote.

3.1.2.5 EVOLUÇÃO DA ARQUITETURA DA FILA HERMES_BUFFER.VHD

Dentre os diversos tópicos propostos originalmente por este trabalho, um deles propõe utilizar filas síncronas e filas bi síncronas de maneira simultânea em uma rede, uma vez que ambas compartilham do mesmo protocolo de comunicação, e teoricamente, são compatíveis. A ideia por trás do uso de ambas as filas está relacionada ao aumento no tempo de transmissão e na área da fila bi síncrona comparada à fila síncrona. A seleção no uso das filas é feita durante a geração da rede a partir da frequência de operação de cada roteador. Em contextos onde as frequências são diferentes é utilizada a fila bi síncrona, enquanto que em contextos onde a frequência é igual, é utilizada a fila síncrona.

Durante o processo experimental, em que cenários foram gerados de maneira manual, o autor detectou situações de impasse, onde tráfegos de pacotes não eram completamente transmitidos. Em uma inspeção detalhada nas simulações utilizando formas de onda, detectou-se que a arquitetura interna da fila síncrona não era capaz de tratar algumas condições existentes em

diferentes cenários de tráfego. As alterações necessárias de serem feitas modificam algumas das propriedades da fila síncrona do roteador HERMES. Sendo assim, propõe-se fazer evoluir a arquitetura síncrona do roteador HERMES, criando uma variação desta. Para classificar a nova variação de arquitetura proposta, assume-se a nomenclatura HERMES-GS, uma vez que a arquitetura é desenvolvida especificamente para trabalhar em redes HERMES-G, possuindo a característica de permitir leituras e escritas unicamente de maneira sincronizada, donde o sufixo GS, acrônimo de *GALS Synchronous*.

A nova proposta soluciona uma deficiência encontrada no processo de leitura de filas síncronas, que assumem depois de feito roteamento do pacote, que a cada ciclo de relógio do roteador, um novo *flit* deve ser transmitido. Esta condição só funciona em dois casos: em um modelo ideal, onde não há atrasos na transmissão dos *flits* dos pacotes, e quando o chaveamento de pacotes garante que uma vez realizado o chaveamento do primeiro *flit*, os demais *flits* estarão disponíveis para consumo da fila no próximo ciclo da frequência de leitura do roteador. A rede HERMES-G assume, por outro lado, que diferentes frequências podem ser definidas estaticamente a cada roteador, o que na prática implica em diferentes atrasos de transmissão para os fluxos de pacotes. Para isso, a máquina de estados que dá suporte ao processo de leitura da fila foi reescrita, levando em consideração a condição que o sinal da fila que informa que existe uma transmissão de um *flit* só seja habilitada quando os ponteiros da fila indiquem que um novo dado está presente na fila.

3.2 PROCESSO DE PARAMETRIZAÇÃO DA REDE HERMES-G

O projeto de uma rede HERMES-G pode ser decomposto em três grandes partes: (i) os arquivos VHDL que compõem a rede; (ii) os arquivos SystemC que compõem os geradores e receptores do tráfego; (iii) um arquivo VHDL e um TCL, responsáveis por conectar a rede com os geradores e receptores e automatizar a compilação e simulação do projeto da rede. No ambiente ATLAS, o processo de geração de redes consiste em gerar de maneira simultânea as três partes. Aqui, por existirem outros fatores que diferenciam o que é rede do que é a estrutura de geração e avaliação de tráfego, preferiu-se optar por desenvolver o gerador de redes com habilidade apenas para gerar a rede em si. A seguir detalha-se como gerar redes HERMES-G via parametrização. Primeiro, descreve-se quais são os componentes que caracterizam a rede, e como eles podem ser parametrizados e adaptados para permitir sua geração. Nos capítulos seguintes, os demais componentes que formam o ambiente de geração de tráfego da rede serão detalhados.

Uma instância de rede HERMES-G é composta ao todo por seis arquivos. Por convenção, renomearam-se os arquivos que compõem a rede, prefixando todos os arquivos com o rótulo HermesG, seguindo a tendência utilizada por outros tipos de redes que o ambiente ATLAS pode gerar. A seguir, detalham-se os arquivos que descrevem a rede, e quais foram as decisões de projeto assumidas para permitir a parametrização dos mesmos.

- ❖ **HermesG_Buffer**: Arquivo utilizado como modelo para descrever quatro arquiteturas distintas de fila. A parametrização com relação ao tipo da fila e a profundidade têm suporte em outros arquivos. Durante a geração da rede, apenas se copia este arquivo.

- ❖ **HermesG_SwitchControl**: Arquivo modelo que descreve seis arquiteturas de SwitchControl. A parametrização com relação ao algoritmo de roteamento a utilizar tem suporte em outro arquivo. Durante a geração da rede, apenas se faz uma cópia deste.
- ❖ **HermesG_Crossbar**: Arquivo modelo, que descreve seis arquiteturas distintas de *crossbar*. A parametrização com relação ao tipo de *crossbar* utilizado novamente é mantida em outro arquivo. Durante a geração da rede, apenas se faz uma cópia deste.
- ❖ **Router**: Arquivo utilizado como modelo para geração dos roteadores da rede. Este arquivo possui uma série de marcadores, que indicam os locais onde devem ser declaradas as arquiteturas de Switch Control, *crossbar* e *buffer* da rede. Durante a geração da rede, uma cópia deste arquivo é feita, substituindo os marcadores existentes pelas arquiteturas dos componentes SwitchControl, *crossbar* e *buffer* selecionados através da interface do ambiente.
- ❖ **NOC**: Arquivo utilizado como modelo para geração da rede, interliga todos os roteadores. Este arquivo possui uma série de marcadores, que indicam os locais onde devem ser declarados e interconectados os roteadores, como também as portas de entrada e saída do componente. Durante a geração da rede, faz-se uma cópia deste arquivo, substituindo os marcadores pelos roteadores da rede e por suas conexões.
- ❖ **HermesG_Package**: Arquivo utilizado como biblioteca de constantes, que armazena, entre outros, os valores do comprimento dos canais e profundidade dos buffers. Durante a geração da rede, uma cópia deste arquivo é feita, substituindo os marcadores existentes pelos valores referentes ao comprimento dos canais e à profundidade dos buffers selecionados através da interface do ambiente.

De maneira resumida, o projeto da interface gráfica do ambiente de geração de redes HERMES-G dá suporte à seleção de oito características da rede, sendo elas:

- ❖ Dimensões da rede, em número de linhas e de colunas (assumindo topologia malha 2D);
- ❖ Profundidade das filas de entrada dos roteadores;
- ❖ Largura dos canais que interligam os roteadores (=largura do *flit*).
- ❖ Algoritmo de roteamento utilizado pelos roteadores.
- ❖ Frequência de operação de cada roteador.
- ❖ Frequência de operação de cada gerador e/ou receptor do tráfego.
- ❖ Codificação de ponteiros das filas bi síncronas.

A seguir detalha-se o projeto da interface gráfica do gerador de redes, apresentando uma solução que dá suporte à seleção de valores para todas as características da rede HERMES-G conforme resumidas aqui.

3.3 PROJETO DA INTERFACE GRÁFICA DO AMBIENTE DE GERAÇÃO

O projeto da interface gráfica do ambiente de geração de redes HERMES-G segue o mesmo padrão usado em outros tipos de redes no ambiente ATLAS, que estendem a interface gráfica do gerador de redes HERMES. A Figura 14 descreve a interface principal do ambiente de geração, onde campos que dão suporte à parametrização das características da rede estão numerados de 1 a 7. Os itens 1 a 4 e 6 são detalhados a seguir. Discute-se os demais (5 e 7) na Seção 3.3.1.

- ❖ Dimensões (1): permite parametrizar as dimensões da rede, suportando redes com tamanho mínimo de dois roteadores em X e dois em Y, e tamanho máximo de dezesseis em X e dezesseis em Y.
- ❖ Comprimento do *flit* (2): permite parametrizar o comprimento dos canais que interligam os roteadores, em 8, 16, 32 ou 64bits.
- ❖ Profundidade do buffer (3): permite parametrizar a profundidade das filas utilizadas pelos roteadores, em 4, 8, 16 e 32 *flits*.
- ❖ Algoritmo de roteamento (4): permite parametrizar o algoritmo de roteamento utilizado pelos roteadores. Os algoritmos de roteamento atualmente disponíveis são: (I) *XY*; (II) *West First Minimal*; (III) *West First Non Minimal*; (IV) *North Last Minimal*; (V) *North Last Non Minimal*; (VI) *Negative First Minimal* (VII) *Negative First Non Minimal*.
- ❖ Codificação de ponteiros (6): Opção do gerador que permite parametrizar a codificação de ponteiros utilizada pelas filas bi síncronas. Dentre as opções disponíveis estão a codificação Johnson e a codificação Gray.

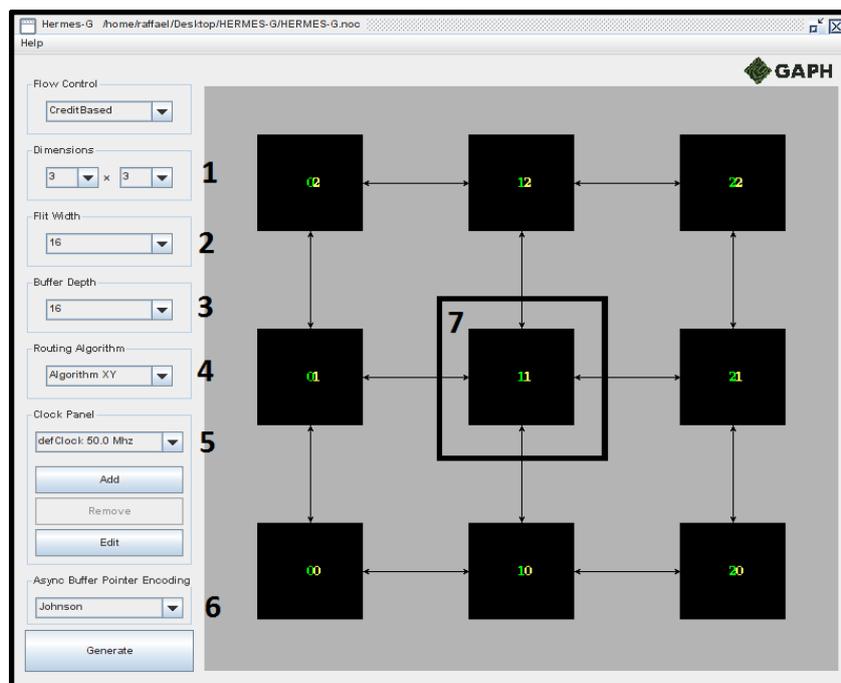


Figura 14: Interface principal do ambiente de geração de tráfego HERMES-G e suas opções de seleção de características de geração de redes.

3.3.1 GERAÇÃO E DEFINIÇÃO DE RELÓGIOS

O item 5 da Figura 14 dá suporte à criação, edição e remoção de frequências usadas nas redes e nos IPs a esta conectados. Estas frequências, depois de cadastradas podem ser usadas (via item 7 da Figura 14) para definir a frequência de operação dos roteadores e seu respectivo módulo de transmissão e recepção de pacotes. Este componente é responsável pela gerência de frequências e possui ao todo quatro funcionalidades, conforme ilustradas na Figura 15.

A funcionalidade 1 permite a visualização dos valores de frequências cadastradas. A funcionalidade 2 permite que novas frequências sejam cadastradas. A funcionalidade 3 permite remover frequências cadastradas, e a funcionalidade 4 permite a edição de frequências cadastradas. Toda frequência é definida por três campos: nome, valor e a unidade de frequência. Descreve-se a seguir as funcionalidades de adição, remoção e edição de frequências.

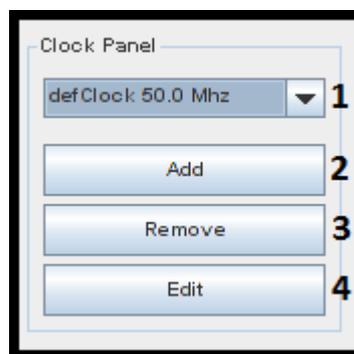


Figura 15: Componente da interface gráfica utilizável para gerenciar valores de frequência definidos pelo usuário. Estas são utilizadas para definir as frequências de operação de roteadores e de módulos IP a estes conectados.

3.3.1.1 ADICIONANDO NOVAS FREQUÊNCIAS

A adição de novas frequências permite especificar frequências usadas por roteadores e módulos IP. A Figura 16 ilustra a interface gráfica que dá suporte a esta funcionalidade. A adição de uma nova frequência consiste em definir um nome, um valor e uma unidade de frequência. A funcionalidade de adição de frequências leva em consideração um conjunto de restrições descritas a seguir:

- ❖ Nomes de frequências devem obrigatoriamente iniciar por uma letra do alfabeto, e seus caracteres remanescentes podem ser letras, números ou o caractere ponto. Além disso, palavras reservadas da linguagem VHDL não podem ser usadas como nomes. Estas restrições seguem as restrições impostas para nomes de variáveis e sinais da linguagem VHDL, uma vez que o nome das frequências é utilizado durante a geração do projeto da rede. Duas frequências de nome distinto podem ter exatamente o mesmo valor;
- ❖ Valores para as frequências válidos devem estar entre um intervalo de 0.1 MHz (100 KHz) e 5,000 MHz (5 GHz);
- ❖ O ambiente verifica se o nome informado para uma nova frequência já está cadastrado. Não é possível definir duas frequências distintas com o mesmo nome;

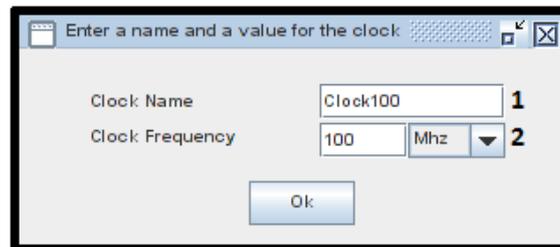


Figura 16: Interface de adição de uma nova frequência. Campos (1) e (2) possibilitam informar um nome para a frequência e um valor para a frequência.

3.3.1.2 REMOVENDO FREQUÊNCIAS CADASTRADAS

A Figura 17 apresenta a interface de remoção de frequências, que possui um único campo, onde é possível selecionar e remover uma frequência cadastrada. A funcionalidade remoção leva em consideração um conjunto de restrições na remoção de frequências. Para toda rede é gerada uma *frequência padrão da rede*, que durante a inicialização é definida pela tripla “def Clock/50.0/MHz”. Esta é automaticamente atribuída a todos os roteadores e seus respectivos transmissores e receptores (IPs) no início de um processo de geração de uma instância da rede HERMES-G. Por regra, este valor não pode ser removido, mas pode ser alterado à vontade.

Uma vez removida uma frequência, todos os roteadores e módulos de transmissão e recepção de pacotes que fazem uso desta frequência recebem automaticamente a frequência padrão da rede.

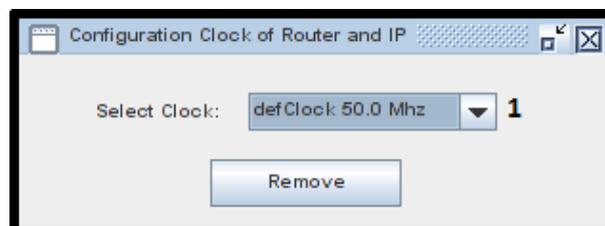


Figura 17: Interface de remoção de frequências. Esta interface possui apenas um campo, onde se pode selecionar uma frequência a remover.

3.3.1.3 EDITANDO FREQUÊNCIAS CADASTRADAS

A Figura 19 apresenta a interface da funcionalidade de edição, compostas pelos seguintes campos: 1-Selecionar uma das frequências cadastradas; 2-Informar um novo nome para a frequência; 3-Informar um novo valor e selecionar uma unidade de frequência. As regras com relação a nomes e valores de frequências válidas seguem as restrições já definidas na Seção 3.3.1.1. Quando uma frequência é editada, todos os roteadores e módulos de transmissão e recepção que fazem uso desta frequência recebem os novos valores informados para a frequência durante a edição.

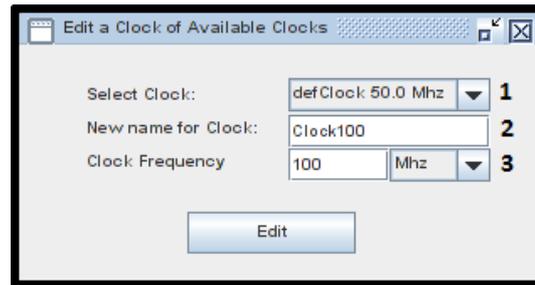


Figura 18: Interface de edição de frequências. Permite selecionar uma frequência cadastrada e modificar seus campos referentes a nome, valor e unidade de frequência.

3.3.1.4 SELECIONANDO UMA FREQUÊNCIA CADASTRADA

O item 7 da Figura 14 define funcionalidade que permite selecionar uma das frequências cadastradas para cada um dos roteadores e seus respectivos módulos de transmissão e recepção de pacotes uma frequência de operação. Ao se clicar com o *mouse* no roteador 11 da Figura 14, surge a Janela detalhada na Figura 19, que permite selecionar uma das frequências cadastradas para o roteador (1), para o módulo transmissor do IP (2) e para o módulo receptor do IP (3).

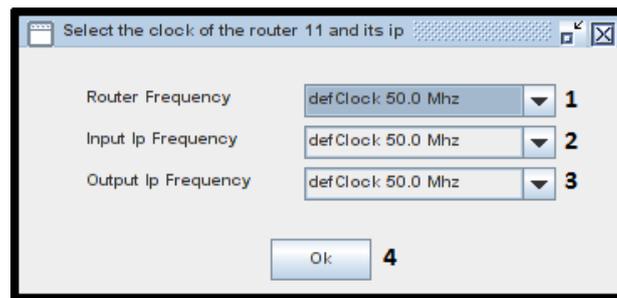


Figura 19: Interface de seleção de frequências cadastradas para o roteador (1), módulo de transmissão de pacotes (2) e módulo de recepção de pacotes (3).

3.4 PROCESSO DE GERAÇÃO DA REDE HERMES-G

Detalha-se a geração da rede HERMES-G em duas partes. Primeiro, descrevem-se as questões relacionadas aos diretórios e ao arquivo de projeto da rede, responsáveis respectivamente por armazenar o projeto desta e por permitir a edição da rede gerada. O segundo passo descreve como se dá a geração dos arquivos da rede, detalhando algumas particularidades na escolha das arquiteturas dos componentes.

3.4.1 CRIAÇÃO DOS DIRETÓRIOS E ARQUIVOS DO PROJETO DA REDE

O diretório onde a rede é criada é escolhido no momento da criação do projeto da rede. Neste diretório a pasta “NOC” irá armazenar os arquivos que definem a rede. Um arquivo com extensão “noc” irá armazenar as características da rede gerada, incluindo as frequências definidas na Seção 3.3.1, e as frequências cadastradas para os roteadores e seus respectivos módulos de transmissão e recepção de pacotes, bem como a codificação de ponteiros definida para a rede. A função deste arquivo é também permitir a edição de um projeto de rede previamente gerado.

Uma vez gerado o diretório da rede e o arquivo de projeto, os arquivos que compõem a rede são produzidos, conforme descrito a seguir.

3.4.2 GERAÇÃO DOS ROTEADORES E SEUS COMPONENTES INTERNOS

O processo de geração da rede é obtido através da cópia dos arquivos utilizados como modelos parametrizáveis dos componentes da rede, detalhados na Seção 3.2. Os componentes `HermesG_Buffer`, `HermesG_Crossbar` e `HermesG_SwitchControl`, que descrevem respectivamente as arquiteturas das filas, as arquiteturas de *crossbar* e as arquiteturas de roteamento utilizadas para a geração dos roteadores são unicamente copiados do diretório do ambiente ATLAS, que armazena os modelos para o diretório destino do projeto da rede. A seguir são detalhados como os roteadores são gerados.

3.4.2.1 GERAÇÃO DO COMPONENTE HERMESG_PACKAGE

O componente `HermesG_Package` é copiado do diretório do ambiente ATLAS que armazena o modelo para o diretório destino do projeto. Durante a cópia do arquivo é feita uma busca e substituição dos marcadores “`tam_flit`” e “`tam_buffer`”, pelos valores referentes ao comprimento dos canais e pela profundidade das filas informada pela interface gráfica durante a definição dos parâmetros da rede. O componente “`HermesG_Package`” é utilizado como um arquivo que armazena constantes responsáveis pelos valores dos componentes utilizados.

3.4.2.2 GERAÇÃO DO COMPONENTE ROUTER

O componente *Router* contém a descrição de um roteador. Este projeto assume que cada roteador em uma rede com N roteadores será implementado por um arquivo VHDL. Sendo assim, um projeto com N roteadores terá na pasta da rede N cópias do componente Router. Esta estratégia foi adotada por conveniência e facilidade de geração dos roteadores, e devido à flexibilidade necessária para definir os diferentes domínios de frequência do projeto. Durante a realização de N cópias do arquivo Router, faz-se uma busca e substituição de cinco marcadores, explicados a seguir.

O primeiro marcador `$router_name$` indica o local onde o nome do roteador deve ser escrito. A estratégia adotada assume que para cada roteador deva existir um nome único. Para tanto, adotou-se a seguinte nomenclatura como nomes dos roteadores. Todo nome contém como prefixo a palavra Router que indica que o arquivo implementa um roteador, seguido do endereço do roteador em hexadecimal, tendo como sufixo o tipo de fila utilizado em cada uma das portas do roteador. O sufixo parte da seguinte proposta. Todo roteador possui cinco portas, denominadas leste, oeste, norte, sul e local. Nas redes HERMES-G uma porta pode possuir uma fila síncrona (HERMESGS), uma fila bi síncrona (HERMESG), ou estar desconectada. Adotou-se o caractere (O) para porta desconectada, o caractere (A) para fila bi síncrona e o caractere (S) para fila síncrona. O sufixo deverá descrever o tipo de fila, ou se a porta estiver desconectada para cada uma das portas do roteador, seguindo a sequência na ordem leste, oeste, norte, sul, local, segundo convenção de ordem já existente no ambiente ATLAS. Por fim, um exemplo de nome

gerado para um roteador seria “Router00S0A0S”, onde seu prefixo indica que é o roteador das coordenadas X=0, Y=0, e que possui três portas conectadas, sendo duas delas com filas síncronas, sendo elas a porta leste e local, e possuindo a porta norte conectada a uma fila bi síncrona.

O segundo marcador $\$Algorithm_type\$\$$ indica o local onde o algoritmo de roteamento escolhido deve ser declarado. O terceiro marcador $\$Crossbar_type\$\$$ indica o local onde a arquitetura do *crossbar* deverá ser escrita. A arquitetura do *crossbar* escolhida pela ferramenta de geração de redes é sempre aquela compatível com a arquitetura de roteamento escolhida, uma vez que como explicado na Seção 3.1.2.2, hoje cada arquitetura de *crossbar* é otimizada, ou tem suas portas minimizadas especificamente para cada algoritmo de roteamento.

O quarto marcador $\$Pin_ground\$\$$ indica o local onde as portas não utilizadas pelo roteador devem ser desconectadas. Em uma topologia malha 2D, podem existir nove combinações de roteadores com relação ao número de portas dos roteadores. A Figura 20 ilustra as combinações, sendo possível existirem roteadores com 3, 4 e 5 portas. O gerador de redes detecta a partir do endereço do roteador durante a geração quais portas estarão desconectadas.

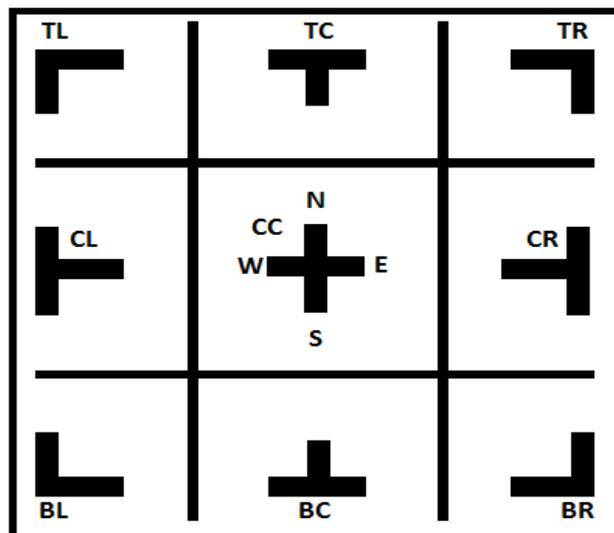


Figura 20: Combinações existentes com relação ao número de portas utilizadas pelos roteadores em uma rede com topologia malha 2D.

O quinto e último marcador $\$port_type\$\$$ indica o local onde as arquiteturas que implementam as filas dos roteadores devem ser declaradas e interconectadas aos demais componentes do roteador. O componente *HermesG_Buffer* dá suporte a quatro arquiteturas de fila, selecionadas pelo gerador a partir das seguintes condições:

- ❖ Quando somente uma frequência é definida para todos os roteadores e módulos de geração e recepção de pacotes, todos os roteadores irão fazer uso da arquitetura da fila do tipo HERMES. Esta rede, além de gastar menos tempo para rotear e transmitir um pacote ocupa menos área que a fila bi síncrona. Como detalhado na Seção 3.1.2, a rede HERMES-G só difere da rede HERMES pelo uso das filas bi síncronas. Dando suporte ao uso de filas do tipo HERMES quando uma única frequência é utilizada este trabalho permite

estender as ferramentas de geração de avaliação de tráfego propostas também para redes HERMES.

- ❖ Quando mais de uma frequência é utilizada, seja por roteadores ou por módulos de transmissão ou recepção, o ambiente de geração faz uso das filas síncronas (HERMESGS) e bi síncronas (HERMESG). Este utiliza a respectiva codificação de ponteiros, com base no que foi definido durante a escolha dos valores da rede. A escolha de qual fila será utilizada em cada porta é feita pela comparação da frequência do roteador com a frequência do roteador ou módulo transmissor/receptor conectado à porta em questão. Se as frequências forem iguais utiliza-se uma fila síncrona, se as frequências forem diferentes é utiliza-se fila bi síncrona. Esta pode garantir a comunicação, independente se o transmissor conectado à fila é mais lento ou mais rápido que a frequência de operação desta. Note-se que duas frequências idênticas com nomes distintos são tratadas como frequências diferentes, para permitir simular frequências distintas apenas em fase e/ou ciclo de serviço.

3.4.2.3 GERAÇÃO DO COMPONENTE NOC

Uma vez gerados os roteadores, o componente NOC, que interconecta todos os roteadores formando a topologia malha da rede é copiado do diretório do ambiente ATLAS que armazena o modelo parametrizável em questão para o diretório destino do projeto. Durante a cópia do arquivo é feita uma busca e substituição de três marcadores. O primeiro, `$clock_noc$` indica o local onde as conexões de entrada e saída do componente NOC devem ser declaradas. Estas conexões permitem que componentes externos transmitam e recebam dados da rede. O segundo e o terceiro marcadores, `map_router` e `$port_router$` são substituídos pelas instâncias dos roteadores detalhados na Seção 3.4.2.2 e pela interligação das portas de entrada com as portas de saída dos roteadores, formando a topologia malha da rede. A Figura 21(a) ilustra em um roteador formado por cinco portas e gerado através do processo descrito na Seção 3.4.2.2. A Figura 21(b) mostra uma rede composta por nove roteadores através do processo descrito aqui.

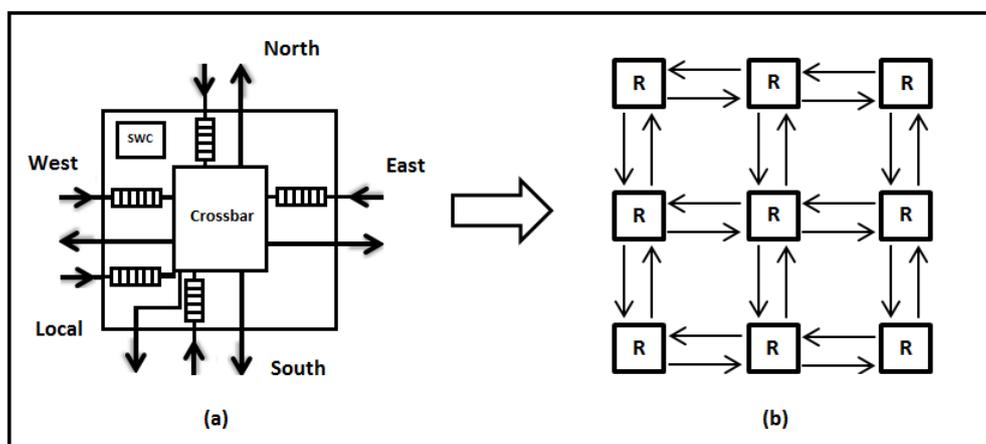


Figura 21: (a) Descrição de um roteador formado por cinco portas, um *crossbar* e um *SwitchControl* (SWC). (b) Descrição de uma rede malha formada por nove roteadores.

4 GERAÇÃO DE TRÁFEGO PARA REDES INTRACHIP NÃO SÍNCRONAS

Este Capítulo descreve a geração de tráfego sintético e utilizando modelos de aplicações CDCM para redes do tipo HERMES-G. Dentre os tópicos explorados estão a caracterização dos tipos de tráfegos, a descrição da implementação dos geradores de tráfego, e o projeto e geração parametrizável dos componentes responsáveis pela transmissão, recebimento e automação do processo de simulação da rede. Este Capítulo traz mais algumas das contribuições deste trabalho.

4.1 CARACTERIZAÇÃO DE TRÁFEGO SINTÉTICO

Esta Seção explora as características que compõem um tráfego sintético. No escopo deste trabalho, propõem-se duas modelagens de tráfego, sendo uma descrita a seguir. Usa-se o termo *modelo de tráfego* aqui para se referir a tráfegos sintéticos. Este difere de *modelos de aplicação*, pois não considera aspectos específicos de aplicações como, por exemplo, a dependência na transmissão de um conjunto de pacotes, ou a variação no comprimento das informações transmitidas.

O modelo de tráfego sintético permite que cada módulo transmissor conectado a um roteador, transmita uma quantidade ilimitada de pacotes, onde se varia o tamanho dos pacotes em número de *flits*. Cada transmissor pode enviar diferentes quantidades de pacotes com diferentes tamanhos. A relação entre destinos dos pacotes e intervalos entre a geração de cada um dos pacotes é dada por distribuições espacial e temporal, descritas a seguir.

4.1.1 DISTRIBUIÇÃO ESPACIAL

Distribuição espacial define a relação entre origens e destinos de um tráfego. Distribuições espaciais suportadas pelo modelo de geração de tráfego sintético utilizado são as propostas por [TED05], e descritas na Figura 22. Ao todo, dá-se suporte a três distribuições:

- ❖ Aleatória: Na distribuição aleatória, cada pacote de um tráfego terá um endereço destino escolhido de maneira aleatória pelo gerador.
- ❖ Destino único: Na distribuição destino único, todos os pacotes de um tráfego terão um único e mesmo endereço de destino.
- ❖ Complemento: Na distribuição complemento, todos os pacotes de um tráfego terão como destino o roteador cujo endereço é obtido pelo complemento do endereço em representação binária do roteador origem.

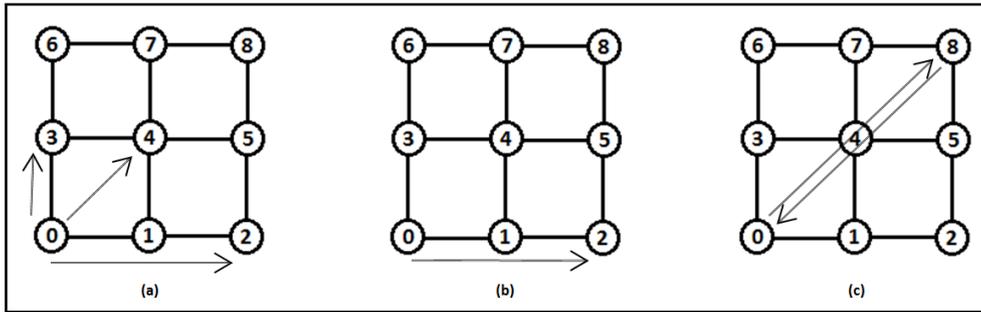


Figura 22: Distribuições espaciais suportadas pelo modelo de tráfego sintético. (a) Aleatória: Roteador endereço zero envia pacotes de maneira aleatória para os roteadores endereços entre 1 e 8. (b) Destino único: Roteador endereço zero envia todos os pacotes para um único endereço. (c) Complemento: por exemplo, o roteador 4 (em coordenadas XY 11, em quatro bits 0101) envia pacotes para o roteador 8 (em coordenadas XY 22, em quatro bits 1010 ou seja, o complemento de 0101).

4.1.2 DISTRIBUIÇÃO TEMPORAL

A distribuição temporal define a variação das taxas de injeção de um tráfego. Este trabalho faz uso do modelo de variação de taxas de injeção descrito pela Figura 23, onde todos os pacotes do tráfego têm tamanho fixo. Aqui, a variação do intervalo de geração entre os pacotes, conforme descrito na Figura pelo tempo ocioso, é quem define a taxa de injeção de cada pacote.

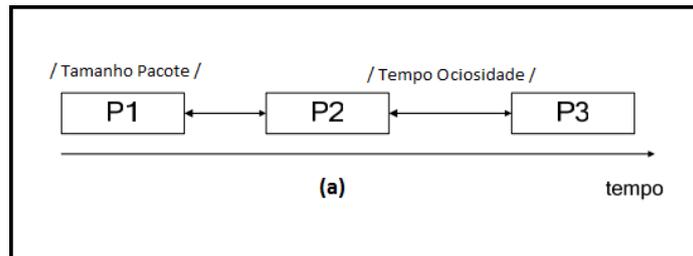


Figura 23: Modelo de variação das taxas de injeção dos pacotes. O tamanho dos pacotes é sempre o mesmo, onde o tempo ocioso é variado conforme a taxa de injeção de cada pacote.

A Equação que calcula o tempo ocioso de cada pacote é detalhada na Figura 24. O cálculo é feito a partir do tamanho do pacote a ser transmitido, multiplicado pela razão entre a taxa máxima do transmissor e a taxa de injeção do pacote. A taxa de injeção de um pacote pode ser fixa para todos os pacotes, ou pode ser variada conforme uma distribuição de probabilidade.

$$\text{Ociosidade} = \text{TamanhoPacote} * \text{NumCiclosFlit} * \left(\frac{\text{TaxaMaxIp}}{\text{TaxaInjPacote}} \right)$$

Figura 24: Equação que calcula o tempo ocioso de cada pacote a partir do tamanho do pacote, da capacidade máxima de transmissão e da taxa de injeção do pacote.

Onde:

Ociosidade: Período ocioso de um pacote

TamanhoPacote: Tamanho do pacote (em número de flits)

NumCiclosFlit: Número de ciclos gastos para transmissão de um flit

TaxaMaxIp: Capacidade máxima de transmissão do gerador de pacotes

TaxaInjPacote: Taxa de transmissão de um pacote

Este trabalho dá suporte a três distribuições de probabilidade: distribuição uniforme, distribuição normal e distribuição exponencial. Com exceção da distribuição uniforme onde todos os pacotes são transmitidos a uma mesma taxa de injeção, as demais distribuições são detalhadas a seguir.

4.1.2.1 DISTRIBUIÇÃO NORMAL

A distribuição normal é calculada para um número de pacotes a partir de cinco valores, sendo eles: (i) Taxa de injeção mínima; (ii) Taxa de injeção máxima; (iii) Taxa de injeção média; (iv) Desvio padrão; (v) Incremento. A Equação que calcula a função de probabilidade de cada uma das taxas de injeção de um intervalo é ilustrada na Figura 25. A Equação é formada pelos valores da média da curva (μ), pela taxa de injeção (rate), pelo número de Euler (e) e pelo desvio padrão (σ).

$$f(\text{Rate}) \equiv \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\text{Rate}-\mu)^2}{2\sigma^2}}$$

Figura 25: Equação que calcula a função de probabilidade seguindo uma distribuição normal de uma taxa de injeção informada.

Por conveniência, detalha-se como o cálculo da distribuição normal é realizado através de um estudo de caso. Assume-se um cenário com 10 pacotes com **taxa de injeção mínima de 100Mbps**, **taxa de injeção máxima de 200Mbps**, **taxa média de injeção de 150Mbps**, **desvio padrão de 10Mbps** e incremento de **10Mbps**. A partir do intervalo mínimo de **100Mbps** e máximo de **200Mbps** e pelo valor de incremento de **10Mbps** calculam-se 10 taxas de injeção no intervalo. Para cada uma das taxas de injeção do intervalo, calcula-se o fator de probabilidade através da Figura 25. O somatório de todos os fatores será sempre igual a um, ou seja, o total de pacotes informado para o tráfego. Cada fator indica um percentual do número de pacotes a serem transmitidos. Em uma distribuição normal, a Equação que calcula o fator de probabilidade tende a criar uma curva em formato de sino, onde todas as taxas de injeção entre o intervalo mínimo e máximo de taxas tendem a formar uma relação de simetria em torno da média.

Durante o cálculo da distribuição normal, poderão ocorrer casos, como o mostrado na Figura 26 que devido ao arredondamento no número de pacotes durante o cálculo do número de taxas de injeção, ocorrerão sobras de pacotes. Estes pacotes não calculados deverão, pelas regras definidas pela distribuição normal, ser acrescidas a taxa de injeção referente à média da curva ou próximo desta. A implementação original da distribuição normal descrita em [TED05] apresentou deficiências no cálculo da distribuição normal para algumas variações nos intervalos das taxas de injeção. Em [SCH10], o autor apresentou um conjunto de correções para o cálculo da distribuição normal, eliminando as falhas existentes. A Figura 26 mostra que o cálculo no exemplo utilizado

resultou em sobra de pacotes, acrescentados em uma taxa de injeção próxima ao valor da média de 500Mbps.

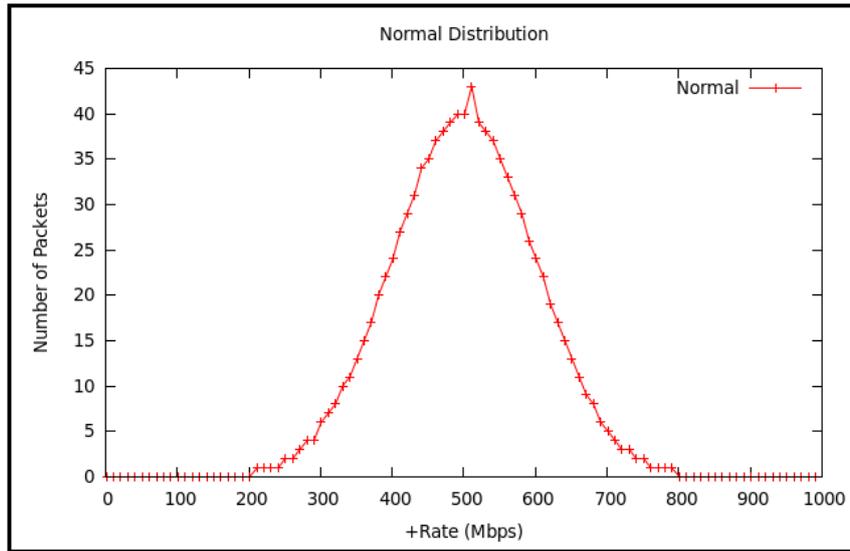


Figura 26: Exemplo de uma distribuição normal de 1000 pacotes, distribuídos em um intervalo com taxa mínima de 1Mbps e máxima de 1000Mbps, média de 500Mbps, desvio padrão de 100Mbps e incremento de 10Mbps.

4.1.2.2 DISTRIBUIÇÃO EXPONENCIAL DECRESCENTE

A distribuição exponencial decrescente é calculada para um número de pacotes a partir de quatro valores, sendo eles: (i) Taxa de injeção mínima; (ii) Taxa de injeção máxima; (iii) Taxa de injeção média; (iv) Incremento. A Equação que calcula a função de probabilidade de cada uma das taxas de injeção de um intervalo é ilustrada pela Figura 27. A Equação é formada pelos valores da média da curva (μ), pela taxa de injeção (rate) e pelo número de Euler (e).

$$f(\text{Rate}) = \frac{1}{(\mu)} e^{\frac{-\text{Rate}}{(\mu)}}$$

Figura 27: Equação que calcula a função de probabilidade seguindo uma distribuição exponencial de uma taxa de injeção informada.

Por conveniência, detalha-se como no caso anterior, o cálculo da distribuição exponencial através de um estudo de caso. Assume-se um cenário com 10 pacotes e taxa de injeção mínima de 100Mbps, taxa de injeção máxima de 200Mbps, taxa média de injeção de 150Mbps e incremento de 10Mbps. A partir do intervalo mínimo de 100Mbps e máximo de 200Mbps e pelo valor de incremento de 10Mbps calculam-se 10 taxas de injeção no intervalo. Para cada uma das taxas de injeção do intervalo é calculado o fator de probabilidade através da Figura 27. O somatório de todos os fatores será sempre igual a um, ou seja, o total de pacotes informado para o tráfego. Cada fator indica um percentual do número de pacotes a serem transmitidos. Em uma distribuição

exponencial decrescente, a Equação que calcula o fator de probabilidade cria uma curva decrescente. Conforme aumenta a taxa de injeção, menor a quantidade de pacotes gerados.

O mesmo tratamento com relação ao cálculo do número de taxas e o arredondamento de pacotes detalhado para a distribuição normal se aplica à distribuição exponencial. Seguindo o que diz a regra, o excesso de pacotes calculado é colocado na taxa de injeção mínima da curva, que representa o ponto onde o maior número de pacotes é gerado. A Figura 28 apresenta um exemplo de uma distribuição exponencial decrescente de 1000 pacotes distribuídos em uma taxa mínima de 1Mbps e uma taxa máxima de 1000Mbps com média de 100Mbps e incremento de 15Mbps.

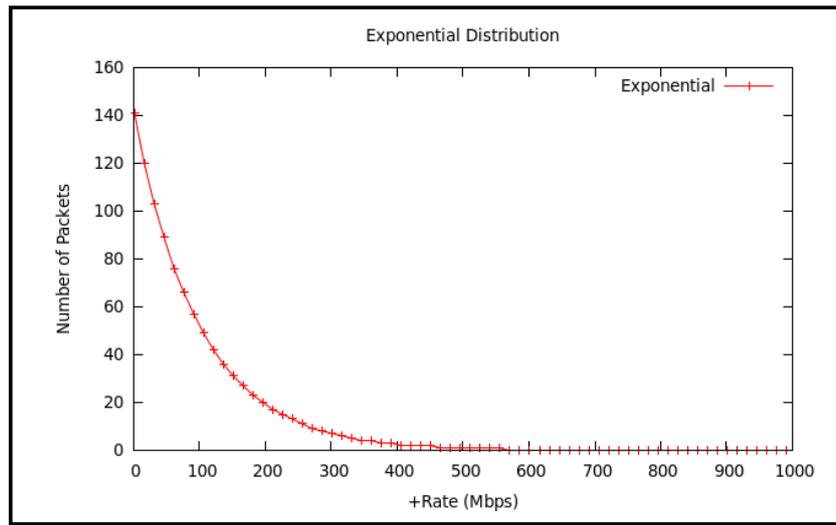


Figura 28: Exemplo de uma distribuição exponencial de 1000 pacotes distribuídos em um intervalo com taxa mínima de 1Mbps e máxima de 1000Mbps, média de 100Mbps e incremento de 10Mbps.

4.2 AMBIENTE DE GERAÇÃO DE TRÁFEGO SINTÉTICO

O ambiente de geração de tráfego sintético para redes HERMES-G diferentemente da abordagem utilizada na Seção 3.1, propõe evoluir o ambiente de geração de tráfego de [TED05]. Modificações foram introduzidas neste ambiente de geração, tanto nos aspectos relacionados à parametrização de um tráfego, como na geração dos arquivos do tráfego. Além disso, o ambiente de geração de tráfego para redes HERMES-G também é responsável pela geração dos arquivos do projeto referentes à transmissão, recepção e simulação da rede, produzidos em conjunto com os arquivos do tráfego.

4.2.1 DEFINIÇÃO DO ARQUIVO DE PROJETO DO TRÁFEGO

Antes de detalhar quais são as características da interface gráfica do gerador e o processo de geração de tráfego, detalham-se algumas questões fundamentais do processo de geração de tráfego. O gerador de tráfego permite produzir diversos cenários de tráfego para um mesmo projeto de rede existente. Cada projeto de tráfego possui um identificador ou nome. Este nome é utilizado para geração de um diretório e um arquivo de cenário durante a geração do tráfego, sendo no diretório armazenados os arquivos do tráfego e o arquivo cenário, com os parâmetros definidos do tráfego.

O arquivo de cenário também é usado durante a simulação e avaliação da rede. Neste trabalho, o gerador de tráfego com as modificações feitas para dar suporte a redes HERMES-G irá utilizar a mesma hierarquia de diretórios onde o tráfego é gerado. O mesmo arquivo de cenário possibilita a criação e a edição de mais de um cenário de tráfego para um mesmo projeto de rede.

4.2.2 PROJETO DA INTERFACE DO GERADOR DE TRÁFEGO SINTÉTICO

A interface gráfica do gerador de tráfego sintético é composta pela interface principal ilustrada na Figura 29 que apresenta ao todo três funcionalidades: (1) Menu de criação e edição de cenários salvos; (2) Interface de configuração de tráfegos para os módulos geradores; (3) Botão que dispara o processo de criação de um tráfego.

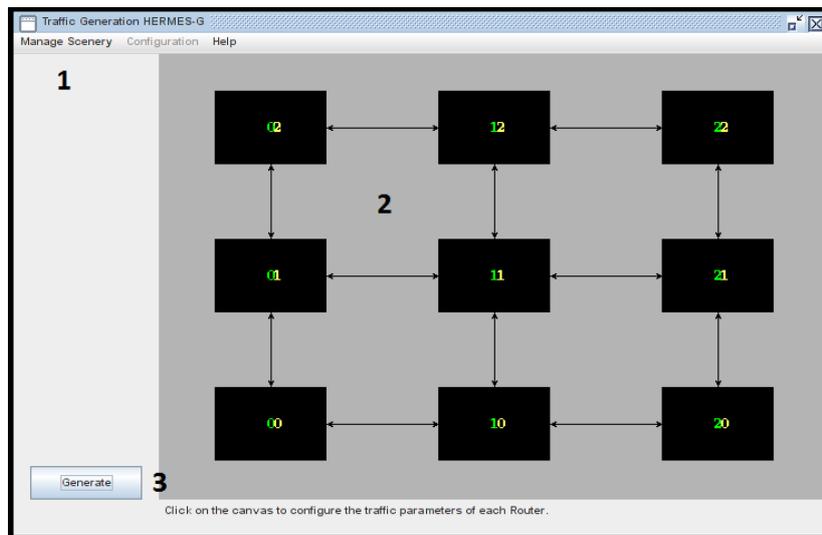


Figura 29: Interface principal do gerador de tráfego, que possibilita criar múltiplos cenários de tráfego.

O item (2) da Figura 29 permite, ao clicar sobre um dos quadrados que representam os transmissores de cada roteador com mesmo endereço XY, selecionar o tipo de tráfego para cada um dos geradores. A Figura 30 descreve a interface que possibilita a seleção dos parâmetros do tráfego para cada um dos geradores. No gerador de tráfego HERMES-G, a frequência de operação de cada um dos módulos transmissores é conhecida, uma vez que durante a geração da rede, este valor teve de ser informado. Sendo assim, na Figura 30 (1) o campo *Frequency* apresenta a frequência do módulo gerador de maneira sombreada, que não pode ser alterada. Na Figura 30 o campo *Target* define a distribuição espacial dos pacotes do tráfego, podendo como já descrito na Seção 4.1.1 ser aleatório, destino único ou complemento. Os campos (3) e (4) da Figura 30 permitem que o número de pacotes e o tamanho do pacote de um gerador sejam variados. O campo (5) da Figura 30 permite configurar as características temporais dos pacotes, através das distribuições normal, uniforme e exponencial descritas na Seção 4.1.2. O campo (6) da Figura 30 permite visualizar graficamente uma distribuição temporal.

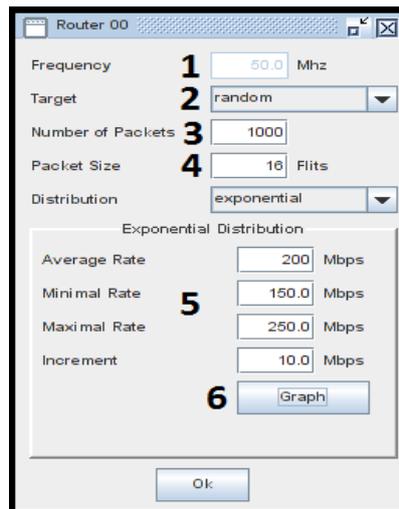


Figura 30: Interface que possibilita parametrizar o tráfego, tais como número de pacotes, tamanho dos pacotes, distribuição espacial e temporal.

Durante a adaptação da interface do gerador que possibilita parametrizar o tráfego, a não existência de um tratamento de exceções nos campos de entrada levou o autor a estudar e propor condições mínimas de entrada de valores baseado em alguns critérios, descritos a seguir:

- ❖ Para qualquer uma das três distribuições temporais, a taxa máxima de injeção deve ser sempre a frequência do gerador multiplicada pelo comprimento do *flit*. Por exemplo, em uma rede com comprimento de *flit* igual a 8bits e um gerador operando a 50 MHz, a **taxa de injeção máxima é de 400Mbps**.
- ❖ Em redes HERMES-G, o tamanho mínimo de pacote suportado para geração de tráfego é de 13 *flits*.
- ❖ O tamanho máximo do pacote em número de *flits* é limitado pela equação $2^{(\text{Comprimento do flit})}$. Por exemplo, em uma rede com comprimento de *flit* igual a 8bits, o tamanho máximo em *flits* de um pacote é de 256 *flits*.
- ❖ O número máximo de pacotes de um tráfego é limitado pela equação $2^{(\text{Comprimento do flit} * 2)}$. Por exemplo, em uma rede com comprimento de *flit* igual a 8bits, o número máximo de pacotes é de 65535 pacotes.

4.2.3 GERAÇÃO E FORMATO DOS PACOTES DO TRÁFEGO

A partir dos valores informados através da interface gráfica é produzido o tráfego. No modelo proposto em [TED05], o tráfego de cada transmissor é representado em arquivos texto, utilizados como entrada por estes módulos conectados nos roteadores. Em cada uma das linhas destes arquivos texto existem informações sobre um pacote. A Figura 31 descreve o formato do pacote produzido pelo gerador de tráfego em um arquivo texto que representa o tráfego de um módulo transmissor. A seguir detalha-se cada um dos campos da Figura 31 que compõem um pacote.

TimeStamp (1 flit)	Destino (1 flit)	Tam Payload (1 flit)	Origem (1 flit)	TimeStampDecimal (4 flits)	NúmeroSequência (2 flits)	Payload (.....)
(a)	(b)	(c)	(d)	(e)	(f)	(g)

Figura 31: Formato de um pacote de tráfego. Os campos representam: (a) Tempo de transmissão do pacote; (b) Roteador destino do pacote; (c) Tamanho do pacote; (d) Roteador origem do pacote; (e) Tempo de injeção em decimal; (f) Número de sequência do pacote; (g) Carga útil (dados) do pacote.

4.2.3.1 TIMESTAMP DE UM PACOTE

O *TimeStamp* define o momento ideal de injeção de um pacote em ciclos da frequência de operação do módulo transmissor. A partir de uma distribuição temporal, cada um dos pacotes assume uma taxa de injeção. De maneira incremental, o tempo de injeção de cada pacote é calculado a partir da Equação da Figura 24. A seguir detalha-se através de um exemplo o cálculo do tempo de injeção de três pacotes de 13 *flits* cada, com taxa de injeção uniforme de 100Mbps, calculado para uma rede com *flits* de 8bits e um módulo transmissor de 100 MHz. Aplicando a Equação da Figura 24 a cada um dos três pacotes obtêm-se os valores descritos na Tabela 3.

Tabela 3: Cálculo do *TimeStamp* para três pacotes de 13 *flits* de tamanho injetados a 100Mbps por um transmissor operando a 100 MHz em uma rede de comprimento de *flit* igual a 8 bits.

Pacote	<i>TimeStamp</i>
Pacote número 0	1 ciclo
Pacote número 1	105 ciclos
Pacote número 2	209 ciclos

Com base nos valores da Tabela 3 observa-se que o intervalo entre os pacotes é o mesmo de 104 ciclos, pelo motivo de possuírem a mesma taxa de injeção de 100Mbps. Além disso, o cálculo do *TimeStamp* de cada pacote é dado de maneira incremental, uma vez que cada pacote demanda um tempo para ser transmitido.

4.2.3.2 CAMPOS REMANESCENTES DO PACOTE

Além do *TimeStamp* ilustrado na Figura 31 outros seis campos compõem o pacote do tráfego. A seguir descreve-se cada um destes:

- ❖ Destino: O campo destino informa para qual roteador o pacote deve ser transmitido. Este é utilizado pelos roteadores da rede para calcular a rota a seguir.
- ❖ Tamanho do *Payload*: Por definição, um pacote na rede HERMES-G é formado por dois campos, sendo eles o *Header* que representa o cabeçalho do pacote e contém o destino e o tamanho do *Payload*, sendo o cabeçalho responsável por realizar a conexão entre origem e destino. Já o campo *Payload* contém os dados do pacote. A função do campo Tamanho do *Payload* é enviar ao transmissor o tamanho do pacote, utilizado durante a leitura para montagem do pacote final a ser transmitido na rede.



Figura 32: Definição de um pacote, formado pelos campos de Header e Payload, onde um representa informação de controle de transmissão e o outro contém os dados do pacote.

- ❖ **Origem:** O campo origem informa de qual roteador o pacote está sendo transmitido.
- ❖ **TimeStamp** em decimal: Este campo define o momento em que a transmissão do pacote deve iniciar, expresso em ciclos de relógio desde o início da simulação em notação decimal, utilizado pelo módulo transmissor durante a transmissão do pacote.
- ❖ **Número de sequência:** O número de sequência é utilizado para garantir que durante a avaliação do tráfego seja possível identificar pacotes a partir de um identificador único. Assim, durante a simulação é possível identificar se pacotes de um fluxo de tráfego foram perdido ou chegaram ao destino em ordem diversa da ordem de envio de pacotes (particularmente útil quando se emprega algoritmos de roteamento não-determinísticos).
- ❖ **Payload:** O *Payload* contém os dados de uma aplicação em uma em uma transmissão. Em tráfegos sintéticos gerados aqui, este campo é normalmente preenchido com valores aleatórios gerados por convenção a partir do número dez (em decimal).

4.2.4 GERAÇÃO DO TESTBENCH PARA TRÁFEGO SINTÉTICO

Além dos arquivos que definem o tráfego sintético, o ambiente de geração de tráfego HERMES-G realiza a geração dos arquivos que definem o ambiente de transmissão e recepção de pacotes, denominado doravante de *Testbench*. O diretório onde estes arquivos são criados é o mesmo do projeto da rede. Neste diretório o subdiretório SC_NOC armazena o Testbench, formado por quatro arquivos: (i) SC_Input_Module: Responsável pela transmissão dos pacotes; (ii) SC_Output_Module: Responsável pela recepção dos pacotes; (iii) HermesG_Fifo_Output: Responsável por sincronizar a recepção dos pacotes, quando o receptor opera em uma frequência diferente do roteador; (iv) TopNoC: Responsável por conectar os transmissores e os receptores a uma entidade de rede. As próximas seções descrevem o processo de geração destes arquivos.

4.2.4.1 O TRANSMISSOR DO TRÁFEGO: SC_INPUT_MODULE

O componente responsável pela leitura e transmissão do tráfego é implementado pela entidade SC_Input_Module. A versão original do componente, conforme descrito na Seção 3.1.2 foi modificada, tanto com relação às portas de entrada e saída do componente quanto a sua arquitetura, uma vez que um novo formato de pacote de tráfego contendo novos campos foi proposto. A Figura 33 ilustra quais são as portas de entrada e saída da entidade SC_Input_Module.

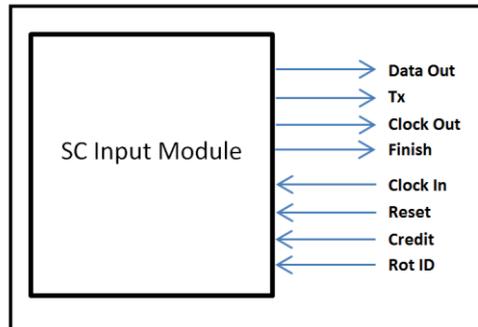


Figura 33: Descrição da interface da entidade SC_Input_Module, que implementa o componente transmissor de pacotes para a rede.

As portas de saída do componente são:

- ❖ Data Out: Transporta os dados do componente SC_Input_Module para o roteador. A largura desta porta é parametrizável, igual à do *flit* da rede.
- ❖ Tx: Indica à porta local do roteador que o transmissor deseja realizar uma comunicação com a rede.
- ❖ Clock Out: Repassa à porta local do roteador a frequência de operação do transmissor.
- ❖ Finish: Indica a finalização da injeção de tráfego na rede, explorado em mais detalhe na Seção 4.2.4.3.

As portas de entrada do componente são:

- ❖ Clock In: Repassa a frequência de operação do transmissor.
- ❖ Reset: É o sinal de inicialização do circuito.
- ❖ Credit: Informa se a porta local do roteador ao qual o transmissor está conectado disponibilidade para receber *flits* um pacote.
- ❖ RotID: Informa o endereço do roteador, utilizado para leitura dos arquivos do tráfego para o respectivo transmissor.

A arquitetura da entidade SC_Input_Module foi basicamente reescrita para comportar as condições de fim de transmissão e para possibilitar a leitura e a transmissão dos novos campos do pacote. Ao todo, o componente opera em três fases, descritas resumidamente a seguir:

- ❖ (1) – Realiza a leitura de um pacote do arquivo de tráfego. Se existir um pacote vai para o item (2), caso contrário informa que transmitiu todos os pacotes do tráfego.
- ❖ (2) – Aguarda enquanto o *TimeStamp* do pacote for menor que o contador de ciclos do relógio do transmissor. Quando o contador for maior ou igual ao *TimeStamp*, vai para o item (3).
- ❖ (3) – Anota o tempo em ciclos do relógio do transmissor, monta o pacote ilustrado na Figura 34 e fica em uma condição de envio até que todos os *flits* do pacote sejam transmitidos. Quando todos os pacotes forem transmitidos, volta novamente para o item (1).

O pacote transmitido possui sete campos conforme ilustra a Figura 34. Estes incluem o destino (a), utilizado pelos roteadores para encaminhar o pacote e o tamanho do pacote (b) utilizado pelas filas dos roteadores para garantir o chaveamento da conexão a partir do tamanho do pacote. Também existe o campo de origem (c) e de número de sequência (e), utilizados pela ferramenta de avaliação de tráfego descrita na Seção 5.3 para identificar os tráfegos. Os valores referentes ao *TimeStamp* (d) e ao Tempo de transmissão (f) são utilizados para os cálculos de vazão e latência dos pacotes, descritos nas seções 5.1.1 e 5.1.2, respectivamente. O item *Payload* (g) completa o pacote de tráfego.

Destino (1 flit) (a)	Tam Payload (1 flit) (b)	Origem (1 flit) (c)	TimeStampDecimal (4 flits) (d)	NúmeroSequência (2 flits) (e)	Tempo Trans (4 flits) (f)	Payload (...) (g)
----------------------------	--------------------------------	---------------------------	--------------------------------------	-------------------------------------	---------------------------------	-------------------------

Figura 34: Formato de um pacote gerado pelo transmissor de pacotes.

4.2.4.2 O RECEPTOR DE TRÁFEGO: SC_OUTPUT_MODULE

O componente responsável pelo recebimento dos pacotes da rede e pela escrita dos arquivos texto de saída do tráfego é descrito pela entidade *SC_Output_Module*. A versão original do componente, conforme descreve a Seção 3.1.2 foi modificada, tanto com relação das portas de entrada e saída do componente quanto em sua arquitetura, uma vez que um novo formato de pacote gerado pelo transmissor, contendo novos campos, foi proposto. A Figura 35 ilustra quais são as portas de entrada e saída da entidade “*SC_Output_Module*”.

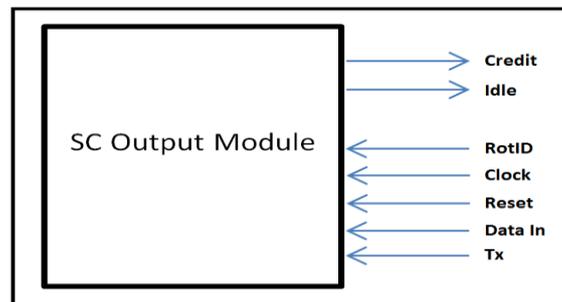


Figura 35: Descrição da interface da entidade *SC_Output_Module*, que descreve o componente receptor de pacotes.

As portas de entrada do componente são:

- ❖ RotID: Informa o número do roteador, utilizado para escrita dos arquivos de saída do tráfego, recebidos pelo receptor.
- ❖ Clock: Recebe a frequência de operação do receptor.
- ❖ Reset: É o sinal de inicialização do circuito.
- ❖ Data In: Recebe os dados da porta local dos roteadores para o componente *SC_Output_Module*. A largura desta porta é igual a do *flit* da rede.

- ❖ Tx: Indica ao componente SC_Output_Module que a porta local do roteador deseja realizar uma comunicação.

As portas de saída do componente são:

- ❖ Credit: Indica à porta local do roteador se o componente SC_Output_Module está apto a receber ou não um *flit* de um pacote.
- ❖ Idle: Indica que o componente receptor não está atualmente recebendo flits de nenhum pacote, e que não está aguardando flits de nenhum pacote incompleto. Descrito em mais detalhe na Seção 4.2.4.3.

A arquitetura da entidade SC_Output_Module foi basicamente reescrita para comportar os novos campos do pacote. Em resumo, ela aguarda a chegada dos pacotes, contendo uma máquina de estados capaz de detectar quais campos está recebendo, escrevendo estes valores no arquivo de saída de tráfego ao qual estiver conectado.

Os pacotes escritos no arquivo de saída são utilizados posteriormente para avaliação de tráfego. O formato do pacote é ilustrado na Figura 36, sendo composto por nove campos. Entre estes há o destino (a), a origem (c) e o número de sequência (e) utilizados pela ferramenta de avaliação descrita na Seção 5.3 para identificar os tráfegos. Os campos TimeStampDecimal (d), Tempo Transmitido (f), TempoChegadaPF (h) e TempoChegadaUF (i) são utilizados para os cálculos da vazão e da latência dos pacotes descritos nas Seções 5.1.1 e 5.1.2.

Destino	Tam Payload	Origem	TimeStampDecimal	NúmeroSequência	Tempo Transmitido	Payload	TempoChegadaPF	TempoChegadaUF
(1 flit)	(1 flit)	(1 flit)	(4 flits)	(2 flits)	(4 flits)	(...)	(Decimal)	(Decimal)
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)

Figura 36: Pacote gerado pelo receptor de pacotes utilizado para avaliação do tráfego dos pacotes.

4.2.4.3 O COMPONENTE TOPNOC

O componente TopNoC é responsável por declarar e conectar transmissores e receptores a uma instância de rede. A seguir descreve-se algumas das estruturas que o componente TopNoC é responsável por gerar e conectar:

- ❖ (1) Realiza a geração das estruturas que descrevem os geradores de frequência para cada uma das frequências definidas para os roteadores e para os módulos de transmissão e recepção de pacotes criados durante a geração da rede.
- ❖ (2) Declara N instâncias de arquivos transmissores e receptores, onde N é o número de roteadores da rede, e uma instância da rede gerada.
- ❖ (3) Atribui a cada transmissor um endereço e conecta suas portas de entrada e saída a cada uma das portas locais dos roteadores da rede com o mesmo endereço. Além disso, a partir da frequência definida para os módulos transmissores durante a geração da rede, atribui a cada transmissor seu respectivo gerador de frequência, conforme definido ao se definir a estrutura da rede.

- ❖ (4) A partir dos valores de frequência definidos para os roteadores, conecta os geradores de frequência às entradas de relógio dos roteadores e de suas portas.
- ❖ (5) Verifica em quais casos a frequência do roteador é diferente da frequência do componente receptor do tráfego. Caso haja pelo menos uma ocorrência dessa condição, o gerador do componente TopNoC realiza uma cópia do arquivo modelo *HermesG_Fifo_Output* para o diretório de projeto, no subdiretório NOC. Esta entidade é detalhada na Seção 3.1.2.4, que descreve o componente Async_Fifo. O nome *HermesG_Fifo_Output* foi escolhido para padronizar a nomenclatura com os demais arquivos da rede.
- ❖ (6) Atribui a cada receptor um endereço, sendo que se a frequência do receptor for igual a do roteador correspondente ao endereço, conecta diretamente o receptor ao roteador. Se a frequência do receptor for diferente da do roteador correspondente ao endereço, conecta o receptor a uma fila "*HermesG_Fifo_Output*" e então conecta este à fila ao roteador. Esta fila garante a comunicação nos casos em que as frequências do roteador e do receptor são diferentes.

Além destas condições, o componente TopNoC é dotado de uma estrutura que finaliza a simulação de maneira automática uma vez que não exista mais nenhum tráfego ativo na rede. Esta estrutura foi desenvolvida, pois redes que possibilitam operar com várias frequências criam possibilidades de existirem cenários que levam muito tempo para serem simulados, e ao mesmo tempo é difícil, como se faz no ambiente ATLAS original, estimar um valor de tempo de simulação. Esta estrutura faz uso de três valores. O primeiro destes são os sinais Finish de todos os transmissores, que indicam quando um transmissor terminou de transmitir seus tráfegos. O segundo são os sinais Idle de todos os receptores, que indicam se o receptor está recebendo tráfego ou se está ocioso. Finalmente usa-se a condição de fila vazia de todas as filas de todos os roteadores da rede, inclusive as filas utilizadas entre os roteadores e os receptores.

Uma vez que todos os transmissores transmitiram seus pacotes, que os receptores estão todos ociosos, e que todas as filas da rede estão vazias, a estrutura enviará um evento externo ao simulador. Este irá retornar a chamada ao arquivo que automatiza a simulação descrita na Seção a seguir. Este último irá então finalizar automaticamente a simulação.

4.2.4.4 ARQUIVO DE AUTOMAÇÃO DA SIMULAÇÃO

O arquivo Simulate.do é gerado para um projeto de rede com os comandos de compilação e simulação da rede, de maneira que este arquivo, uma vez disparado, automatiza o exercício da rede. Este arquivo contém uma condição que após disparar o comando de simulação da rede, aguarda que um evento externo do simulador que informe o fim da simulação, para então finalizar sua execução. A simulação de um tráfego sintético é feita a partir da leitura dos arquivos do tráfego e da transmissão dos pacotes na rede. Por fim, os receptores coletam os pacotes e escrevem os arquivos de saída, contendo os dados dos pacotes e os tempos gastos durante a transmissão.

4.2.5 VALIDAÇÃO DA PROPOSTA

Como já descrito em seções anteriores, desenvolveu-se uma ferramenta de testes em caráter experimental, com objetivo de verificar se as características funcionais da rede e do ambiente de Testbench desta estão operacionais, conforme os critérios do desenvolvimento. A ferramenta de testes permite, através da interface de linha de comando, mostrada na Figura 37, que diferentes abordagens sejam utilizadas para definir cenários de teste, estes sendo formados por arquivos de rede e de tráfego.

Para esta ferramenta, propôs-se uma distribuição de tráfego alternativa às demais existentes no gerador de tráfego do ambiente ATLAS, onde todos os elementos transmissores da rede geram tráfegos para todos os endereços da rede, com exceção dele mesmo. A distribuição temporal utilizada é uniforme, que assume a taxa máxima de transmissão do transmissor. O modelo de tráfego varia o número e o tamanho dos pacotes de diversas maneiras, seja a partir de parâmetros informados pelo usuário durante a geração dos cenários ou de maneira aleatória. Considera-se que esta abordagem permite que todas as combinações possíveis de caminhos sejam utilizadas pelos pacotes.

```

||||| Gerador e verificador de tráfego sintético para redes HERMES e HERMES-G |||||
-----
Java Gerador : TrafficOp | NumMaxX | NumMaxY | FlitWidth | TamPckt | NoCType | BufDepth | BufType | AlgoType | ClockType
-----
Example : java Gerador 2 2 16 16 0 4 J 1 1
Obs :
TrafficOp : (0) Gen Traffic (1) Ver Traffic (2) Gen NoCs (3) Gen NoCs + Traffic (4) Ver Traffic Ger (5) Gen Auto Cenery
FlitWidth : (8) bits (16) bits (32) bits (64) bits
NoCType : (0) HERMES-G (1) HERMES
BufDepth : (8) flits (16) flits (32) flits
BufType : (J) Johnson Coding (G) Gray Coding
ClockType : (0) DefClock 50 Mhz (1) Random 0Mhz - 5Ghz
AlgoType : (0) XY (1) WFM (2) WFNM (3) NLM (4) NLNM (5) NFM (6) NFNM

***** Warning : This Software dont verifies wrong values !!! *****
|||||

```

Figura 37: Interface linha de comando da ferramenta de testes responsável pela geração e verificação de cenários.

Utilizando uma opção do ambiente que permite gerar todas as combinações de redes em um intervalo de parâmetros, obteve-se ao todo 50.400 cenários possíveis de teste para redes do tipo HERMES-G. Estes cenários variam opções de dimensões da rede, profundidade de filas, largura de *flit*, algoritmo de roteamento e codificação de ponteiro das filas. Como já descrito, a ferramenta, além de possibilitar que a simulação seja feita de maneira automática, é finalizada quando os tráfegos de um cenário forem transmitidos. Também se adicionaram diretivas na ferramenta para detectar e usar capacidade do computador hospedeiro para *multithreading*, quando existirem. Durante as simulações detectaram-se ganhos significativos no tempo de simulação com o uso destas técnicas. Além da simulação, a ferramenta também verifica o tráfego gerado e compara se os pacotes do tráfego foram corretamente recebidos nos destinos. Para os 50.400 cenários de testes simulados, em todos os casos houve sucesso na transmissão e recepção dos pacotes.

4.3 AMBIENTE DE GERAÇÃO DE TRÁFEGO DE MODELO DE APLICAÇÕES

Além da proposta de um modelo de tráfego sintético, este trabalho também propõe fazer uso de modelos de aplicações para descrever tráfego para uma rede HERMES-G. Usa-se o modelo de aplicações CDCM, do inglês *Communication Dependence and Computation Model*, proposto por Marcon [MAR05], que representa uma aplicação pelas suas dependências de computação e comunicação. De acordo com Marcon [MAR05], o modelo permite capturar o tempo de computação, o tamanho da mensagem e suas dependências. O modelo CDCM é representado através de um grafo CDCG, do inglês *Communication Dependence and Computation Graph*, sendo utilizado para descrever as aplicações.

Marcon propôs o ambiente denominado *Communication Analysis for Embedded Systems* (CAFES), que permite a geração e o mapeamento de modelos CDCM sobre um MPSoC baseado em NoCs. O processo de criação e geração de um modelo de aplicação no escopo deste trabalho é conduzido em três fases, ilustradas na Figura 38. O ambiente CAFES da Figura 38(a) é usado nas etapas de criação e mapeamento de um modelo de aplicação sobre uma topologia de NoC. Esta etapa gera um arquivo intermediário, que servirá de entrada para a ferramenta de geração de tráfego do ambiente ATLAS, ilustrado na Figura 38(b). A partir das definições contidas neste arquivo gera-se o *Testbench*, contendo os arquivos de tráfego e o ambiente de transmissão e recepção adaptado para as características do modelo de aplicação CDCM informado.

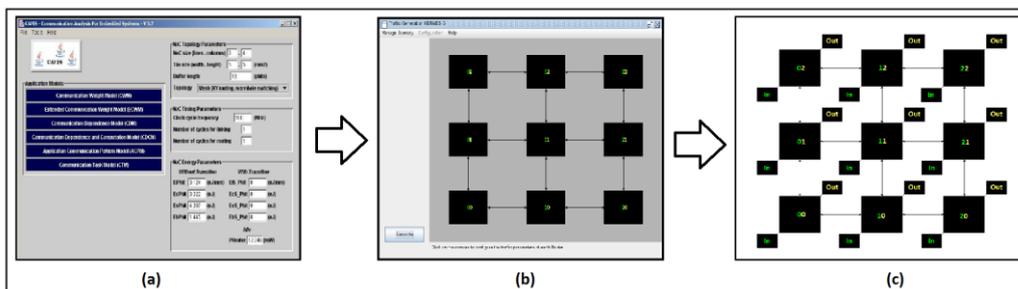


Figura 38: Uso de modelos de aplicações CDCM no ambiente Atlas: (a) criação e geração de um grafo de aplicações CDCM; (b) geração do Testbench pela ferramenta ATLAS; (c) geração dos arquivos de tráfego e o ambiente de transmissão e recepção de pacotes.

4.3.1 O AMBIENTE CAFES: CRIAÇÃO E MAPEAMENTO DE UM MODELO DE APLICAÇÃO

O ambiente CAFES permite descrever modelos de aplicação CDCM a partir de grafos CDCG. Em um grafo CDCG, ilustrado na Figura 39 vértices correspondem a tarefas. Toda aplicação possui um início, descrito no grafo pelo vértice especial START e um fim, descrito no grafo pelo vértice especial END. Cada tarefa é descrita pelos seguintes campos:

- ❖ Tempo de processamento: Define um tempo de processamento da tarefa expressa em nanossegundos.
- ❖ Identificador origem e destino: Define quem são a origem e o destino da comunicação realizada durante a execução da tarefa. Durante o mapeamento estes identificadores são traduzidos em endereços da rede.

- ❖ Mensagem de transmissão: Define um tamanho da mensagem a ser transmitida após o processamento da tarefa ocorrer, expresso em *flits*.
- ❖ Dependências de entrada: Definem os eventos que devem acontecer para que a tarefa seja executada.
- ❖ Dependências de saída: Definem que outras tarefas devem ser disparadas após a execução da tarefa.

O grafo CDCG permite definir as características de tempo de processamento e quantidade de comunicação. Ele também permite particionar a aplicação em tarefas e definir quais são os eventos que devem acontecer antes que cada tarefa seja executada, e quais eventos devem ser disparados após sua execução.

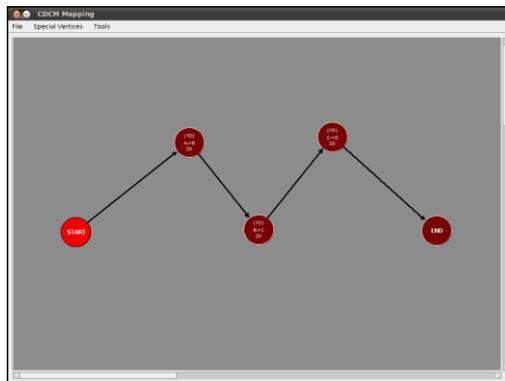


Figura 39: Ferramenta de criação de aplicações CDCM através de grafos CDCG no ambiente CAFES.

Uma vez descrita a aplicação, o ambiente CAFES permite realizar o mapeamento das tarefas para um modelo de rede de dimensões definidas. O mapeamento considera um conjunto de requisitos, visando minimizar o consumo de energia e a redução no tempo de execução da aplicação na rede. Ao todo, o ambiente permite três estratégias de mapeamento, sendo que no escopo deste trabalho, o mapeamento só é utilizado para conhecer em quais endereços tarefas que compõem uma aplicação devem ser mapeadas, não sendo detalhado. Na Seção 7.2, comenta-se uma adaptação do processo de mapeamento, uma vez que o mapeamento como é feito atualmente assume uma rede síncrona. A Figura 40 descreve a ferramenta de mapeamento do ambiente CAFES para o grafo ilustrado pela Figura 39, onde cada endereço comporta uma tarefa.

Uma vez realizado o mapeamento, o ambiente CAFES foi adaptado para gerar um arquivo contendo os valores referentes à aplicação e ao mapeamento, utilizadas pela ferramenta de geração de tráfego do ambiente ATLAS, conforme ilustrado na Figura 38(b) para geração dos arquivos de tráfego e do Testbench. A seguir, detalham-se as características da aplicação colocadas neste arquivo.

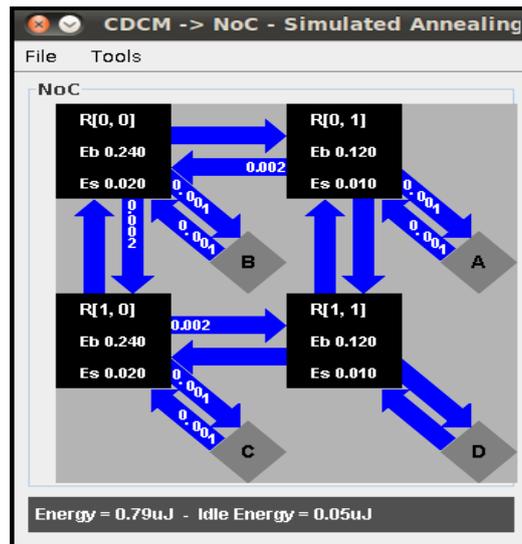


Figura 40: Ferramenta de mapeamento do ambiente CAFES. Cada tarefa da aplicação é mapeada em um dos endereços da rede.

4.3.2 FORMATO DE REPRESENTAÇÃO DE UM MODELO DE APLICAÇÃO

O arquivo do modelo de aplicação é gerado em um diretório informado pelo usuário durante o processo do mapeamento, e contém o nome do projeto e uma extensão model. Este contém quatro campos que representam os dados da aplicação, e são utilizados pela ferramenta de geração de tráfego do ambiente ATLAS para gerar os arquivos de tráfego e o Testbench para o modelo de aplicação CDCM. A Figura 41 mostra um exemplo de tal arquivo, onde os quatro campos que representam os dados da aplicação são interpretados da seguinte forma:

- ❖ **Noc Size:** Informa o tamanho da rede utilizada pela aplicação. A ferramenta de geração utiliza este valor para verificar se a rede gerada pelo usuário irá comportar o número de tarefas da aplicação.
- ❖ **Cost of Applications:** Informam a origem e o destino das tarefas, bem como o tempo de processamento em nanossegundos, a quantidade de *flits* a serem transmitidos e o tempo máximo de transmissão dos dados, sempre em nanossegundos.
- ❖ **Dependence of Applications:** Informam as dependências entre as tarefas, e quais delas estão relacionadas ao processo de inicialização e término da simulação.
- ❖ **Mapping of Applications:** Informam em quais roteadores as tarefas devem ser mapeadas.

A partir destes campos a ferramenta de geração de tráfego realiza a geração dos arquivos que compõem as mensagens a serem transmitidas e o Testbench, formado pelos arquivos transmissores e receptores do tráfego.

```

1 #
2 Noc Size
3 #
4 2 2
5 #
6 Cost of Applications
7 0 A - B 20 : 60 : 0
8 1 B - C 40 : 40 : 0
9 2 C - D 60 : 20 : 0
10 #
11 Dependence of Applications
12 START 0
13 END
14 0 1
15 1 2
16 2 END
17 #
18 Mapping of Applications
19 A 0 0
20 B 1 0
21 C 0 1
22 D 1 1

```

Figura 41: Arquivo que contém as características de um modelo de aplicação CDCM.

4.3.3 ADAPTAÇÃO DA INTERFACE DO GERADOR DE TRÁFEGO

A interface gráfica do gerador de tráfego foi adaptada para dar suporte à leitura do arquivo que descreve o modelo de uma aplicação, conforme ilustrado na Figura 42. Após o modelo ser carregado verifica-se se a extensão do arquivo é válida, e então se efetua a leitura dos campos do arquivo. Durante a leitura do arquivo, o ambiente verifica se as dimensões da rede utilizada pela aplicação são iguais ao projeto da rede gerada, bem como outros valores da aplicação relacionada às capacidades da rede. Entre estes, pode-se citar, por exemplo, o tamanho das mensagens ou o tempo de processamento, que variam conforme o comprimento do *flit* utilizado na rede.

Durante a leitura, o ambiente informa se o arquivo que descreve a aplicação está de acordo com as restrições impostas por um projeto de rede. Uma vez que estão de acordo, apresenta-se a mensagem de sucesso conforme ilustrado pela Figura 42, sendo que o botão Generate é então liberado. A ativação deste gera os arquivos do tráfego e o Testbench composto pelos transmissores, receptores e pelo arquivo que interconecta a rede aos transmissores e os receptores. Durante a geração destes arquivos o ambiente produz também um arquivo cenário de tráfego.

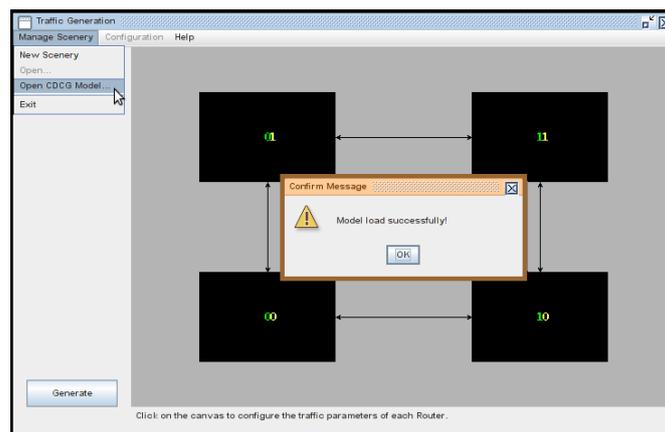


Figura 42: Interface principal do gerador de tráfego, que possibilita carregar arquivos que descrevem modelos de aplicações.

4.3.4 GERAÇÃO E FORMATO DOS PACOTES DO TRÁFEGO

A partir dos valores expressos nas descrições de tarefas faz-se a geração do tráfego. Utiliza-se aqui a mesma abordagem do tráfego sintético, descrito na Seção 4.2.3, onde o tráfego de cada

transmissor é produzido em arquivo texto usado como entrada por estes módulos, cada conectado a um roteador. Cada tarefa possui uma mensagem com um conjunto de informações a ser transmitida. Definiu-se que cada mensagem seria transmitida como um único pacote. Sendo assim, as mensagens de cada tarefa mapeada para um endereço da rede são vistas como um pacote descrito em um arquivo de texto que caracteriza o tráfego.

O formato dos pacotes presentes nos arquivos texto é descrito conforme ilustrado na Figura 43, que representa o modelo de um pacote do tráfego, com cinco campos, sendo estes:

- ❖ Origem: O campo origem informa de qual endereço o pacote está sendo transmitido. Ele é obtido a partir do identificador origem da tarefa, associado a um endereço da rede.
- ❖ Destino: O campo destino informa para qual roteador o pacote deve ser encaminhado. É obtido a partir do identificador destino da tarefa, associado a um endereço da rede.
- ❖ Tempo de processamento: O tempo de processamento é definido em ciclos de relógio da frequência de cada transmissor, calculado a partir do tempo de transmissão de cada tarefa.
- ❖ Tamanho do pacote: Define o tamanho do pacote em *flits*, calculado a partir do tamanho da mensagem definido para cada tarefa.
- ❖ Número de sequência: Utilizado para garantir que durante a avaliação do tráfego seja possível identificar os pacotes a partir de um identificador único em relação a todos os pacotes que trafegam na rede.

Origem (1 flit) (a)	Destino (1 flit) (b)	Tempo Processamento (1 flit) (c)	Tamanho Pacote (1 flit) (d)	Número Sequência (1 flit) (e)
---------------------------	----------------------------	--	-----------------------------------	-------------------------------------

Figura 43: Formato de um pacote de tráfego. Os campos são: (a) Origem do pacote; (b) Roteador destino do pacote; (c) Tempo de processamento do pacote em ciclos; (d) Tamanho do pacote; (e) Número de sequência do pacote.

4.3.5 GERAÇÃO DO TESTBENCH PARA MODELOS DE APLICAÇÃO

A partir dos valores informados pelo arquivo da aplicação, geram-se os respectivos transmissores, receptores e um componente que interliga a rede produzida aos transmissores e receptores. Por existirem mudanças com relação à maneira como a transmissão é feita, um novo componente transmissor foi desenvolvido. O componente CDCG_TopNoC que interliga a rede aos transmissores e receptores sofreu uma revisão em suas estruturas, uma vez que agora este arquivo coordena a transmissão das tarefas, considerando as dependências existentes.

A fim de garantir que a ferramenta de avaliação e as técnicas associadas tenham suporte para este modelo de tráfego, o formato do pacote, a arquitetura do receptor de tráfego e o formato dos pacotes de saída do tráfego permanecem os mesmos. Isto permite que o ambiente de avaliação descrito no Capítulo 5 dê suporte à avaliação de tráfego de modelos de aplicações CDCM em redes HERMES-G.

4.3.5.1 MÓDULO TRANSMISSOR DE TRÁFEGO: SC_INPUT_MODULE

O componente responsável pela leitura e transmissão do tráfego de modelos de aplicações CDCM tem suporte na entidade SC_CDCG_Input_Module. Com relação à variação nas portas de entrada e saída, comparado à entidade SC_Input_Module, utilizada para transmissão de tráfego sintético e descrita na Seção 4.2.4.1, o componente SC_CDCG_Input_Module possui uma porta de entrada *Send* e uma porta de saída *Receive*, conforme ilustra a Figura 44, utilizadas para comunicação entre o transmissor e a entidade CDCG_TopNoC. A porta de entrada *Send* indica ao componente transmissor que um pacote deve ser lido e transmitido. Já a porta *Receive* é utilizada pelo transmissor para confirmar quando o pacote foi transmitido à rede.

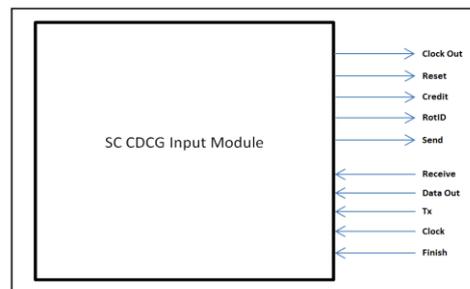


Figura 44: Descrição das portas de entrada e saída da entidade SC_Input_Module, que descreve o componente transmissor de pacotes da rede para o modelo de Testbench de modelos de aplicações CDCM.

A arquitetura da entidade SC_CDCG_Input_Module foi projetada para ter o seguinte comportamento:

- ❖ (1) – Aguarda até que o sinal de entrada *Send* esteja em '1', o que indica que um pacote deve ser transmitido, e o estado deve avançar para o item (2).
- ❖ (2) – Realiza a leitura de um pacote do arquivo de tráfego, captura o tempo de processamento em ciclos do relógio do transmissor em um contador e avança para o item (3). Se o arquivo de tráfego estiver vazio, o estado deve ser atualizado para o item (6).
- ❖ (3) – Permanece neste estado, decrementando em um ciclo o contador criado no item (2) a cada ciclo do relógio do transmissor. Quando o contador atingir 0, o que indica que o tempo de processamento da tarefa foi atingido, avança para o item (4).
- ❖ (4) – Monta o pacote e tenta transmiti-lo. Neste instante, anota o tempo que será adicionado (como Tempo Ideal do pacote), ou o momento em que a tarefa após ser executada quis ganhar acesso à rede. No momento em que o pacote ganhar acesso à rede o tempo deve ser anotado como (Tempo Transmitido), ou seja, o momento em que o pacote entrou na rede. Uma vez transmitido, avança-se para o item (5).
- ❖ (5) – Informar o componente CDCG_TopNoC que o pacote foi transmitido através do sinal de saída *Receive* e ir para o item (1).

- ❖ (6) – Informar o componente CDCG_TopNoC através do sinal de saída Finish que o transmissor de um determinado endereço já enviou todos os pacotes referentes a seu arquivo de tráfego.

4.3.5.2 MÓDULO CDCG_TopNoC.VHD

O componente CDCG_TopNoC é responsável por declarar e conectar os transmissores e receptores a uma instância de uma rede. A seguir descrevem-se algumas das estruturas que o componente CDCG_TopNoC é responsável por gerar e conectar:

- ❖ (1) Produz as estruturas que descrevem os geradores de frequência para cada frequência de roteador definida e para os módulos de transmissão e recepção de pacotes criados.
- ❖ (2) Declara N instâncias de arquivos transmissores e receptores, onde N é o número de roteadores da rede. Gera ainda uma instância da rede.
- ❖ (3) Atribui a cada transmissor um endereço e conecta suas portas de entrada e saída a cada uma das portas locais dos roteadores da rede com o mesmo endereço. Além disso, a partir da frequência definida para os módulos transmissores durante a geração da rede, atribui a cada transmissor seu respectivo gerador de frequência.
- ❖ (4) A partir dos valores de frequência dos roteadores, conecta os geradores de frequência dos roteadores às portas da rede que conectam as respectivas frequências aos roteadores.
- ❖ (5) Verifica em quais casos a frequência do roteador é diferente da frequência do componente receptor do tráfego. Caso haja pelo menos uma ocorrência dessa situação, o gerador do componente TopNoC realiza uma cópia do arquivo modelo HermesG_Fifo_Output para o diretório de projeto, subdiretório NOC. Esta entidade é detalhada na Seção 3.1.2.4, que descreve o componente Async_Fifo. O nome HermesG_Fifo_Output foi escolhido para manter a padronização de nomenclatura com os demais arquivos da rede.
- ❖ (6) Atribui a cada receptor um endereço, sendo que se a frequência do receptor for igual à do roteador correspondente ao endereço, conecta diretamente o receptor ao roteador. Se a frequência do receptor for diferente a do roteador correspondente ao endereço, conecta o receptor a uma fila HermesG_Fifo_Output e então conecta a fila ao roteador. Esta fila garante a comunicação nos casos em que as frequências do roteador e do receptor são diferentes.

Além destas condições, todo componente CDCG_TopNoC possui uma estrutura responsável por controlar a transmissão das tarefas, de maneira que as dependências existentes entre as tarefas sejam levadas em consideração. O componente CDCG_TopNoC, através dos sinais Send e Receive coordena a transmissão das tarefas, possuindo uma memória que armazena quais são as dependências de cada tráfego, e qual deve ser o fluxo de eventos a ser executado para que todas as tarefas executem, garantindo que suas dependências sejam consideradas.

Todo componente CDCG_TopNoC é dotado de uma estrutura que finaliza a simulação de maneira automática, uma vez que não exista mais tráfego na rede, nem haja qualquer possibilidade de tráfego existir no futuro. Uma vez que todos os transmissores tenham transmitido seus pacotes, e os receptores estejam em estado ocioso, e todas as filas da rede estiverem vazias, a estrutura produzirá um evento externo ao simulador. Este irá retornar a chamada ao arquivo que automatiza a simulação, e a seguir irá finalizar automaticamente a simulação.

4.3.5.3 ARQUIVO DE AUTOMAÇÃO DA SIMULAÇÃO

O arquivo Simulate.do é gerado para um projeto de rede contendo os comandos de compilação e simulação da rede, de maneira que este arquivo uma vez disparado automatiza a simulação da rede. Este arquivo contém uma condição que após disparar o comando de simulação da rede, aguarda que um evento externo do simulador informe o fim da simulação, para então finalizar sua execução. A simulação de um tráfego de modelos de aplicação CDCM é feita a partir da leitura dos arquivos do tráfego e da transmissão dos pacotes na rede. Por fim, os receptores coletam os pacotes e escrevem os arquivos de saída, que conterão os dados dos pacotes e os tempos gastos durante a transmissão.

5 AVALIAÇÃO DE TRÁFEGO EM REDES INTRACHIP NÃO SÍNCRONAS

Este Capítulo apresenta a avaliação de tráfego relacionada a este trabalho. Nele descrevem-se as métricas utilizadas para avaliar o tráfego na rede HERMES-G, e como estas são calculadas. Além disso, descreve-se se compara a proposta com o ambiente proposto por Tedesco em [TED05]. Este Capítulo contém mais uma das contribuições deste trabalho.

5.1 MÉTRICAS DE AVALIAÇÃO DE TRÁFEGO

Faz-se uso aqui de duas métricas para avaliar o tráfego. A primeira delas é a latência e a segunda a vazão. A seguir, detalha-se como estes valores são expressos, obtidos e calculados.

5.1.1 LATÊNCIA

A latência mede o tempo gasto por um pacote para ser transmitido. Aqui latência é expressa em nanossegundos (ns) ao invés de ciclos de relógio, por razões óbvias no contexto deste trabalho.

A latência de um pacote na rede é o intervalo entre o momento em que o primeiro *flit* do pacote está apto a entrar na rede até o momento em que o último *flit* do pacote sai da rede. Com base nas explicações das Seções 4.2.4.1 e 4.2.4.2, transmissor e receptor durante a transmissão de cada pacote anotam os momentos em que o pacote tentou entrar e saiu da rede. Estes valores podem ser obtidos nos arquivos de saída do tráfego gerados pelos receptores. A Figura 36 descreve o formato de informações sobre cada pacote nos arquivos de saída do tráfego. O item (f) na Figura 36 representa o instante, em ciclos de relógio da frequência do transmissor (a partir do início da simulação) quando o primeiro *flit* do pacote estava apto entrar na rede. O item (i) da mesma dá o instante, em ciclos de relógio da frequência do receptor quando o último *flit* do pacote deixou a rede. O cálculo da latência de um pacote ocorre durante a avaliação de tráfego. Primeiro, os valores que descrevem o momento em que cada pacote ficou pronto para entrar na rede, e o instante em que cada pacote saiu da rede são convertidos de ciclos de relógio para tempo absoluto após o início da simulação. As frequências do transmissor e do receptor podem ser diferentes. Por esta razão, o cálculo da latência ocorre durante a avaliação de tráfego, quando o ambiente conhece as frequências de ambos os componentes. Feito isso, a Equação da Figura 45 é usada. Nela, do instante de chegada do último *flit* do pacote no receptor subtrai-se o instante em que o pacote ficou apto a ser transmitido, ambos valores medidos em tempo absoluto. O valor resultante é o tempo gasto pelo pacote para ser transmitido, ou seja, a latência do mesmo.

$$\text{Latência pacote} = \text{Tempo chegada último flit pacote no receptor} - \text{Tempo transmissão pacote no transmissor}$$

Figura 45: Equação que realiza o cálculo da latência de um pacote. A latência é calculada a partir da subtração do instante em que o pacote está apto a entrar na rede do instante em que o último *flit* do pacote deixou a rede.

5.1.2 VAZÃO

A vazão mede a quantidade de informação transmitida entre dois pontos da rede em um intervalo de tempo. Aqui vazão é expressa em Megabits por segundo (Mbits/s ou Mbps).

Do ponto de vista da rede, a vazão é a quantidade de bits transmitidos divididos pelo tempo gasto para transmissão do pacote. A quantidade de bits transmitida refere-se ao tamanho do pacote, obtido através de campos do pacote no arquivo de saída do tráfego. Já o tempo de transmissão corresponde à latência do pacote, obtida pelo cálculo descrito na Seção 5.1.1.

O cálculo da vazão de um pacote é realizado durante o processo de avaliação de tráfego. A Equação da Figura 46 denota o cálculo da vazão, pela divisão do tamanho do pacote em bits pela latência do pacote em tempo absoluto. A vazão calculada é então multiplicada por 1000 para ser expressa em Mbps. Esta multiplicação claramente converte bits/ns em Mbits/s.

$$\text{Vazão do pacote (Mbps)} = \left(\frac{\text{Tamanho do pacote (bits)}}{\text{Latência do pacote (ns)}} * 1000 \right)$$

Figura 46: Equação de cálculo da vazão de um pacote. A vazão é calculada a partir da divisão do tamanho do pacote pelo tempo gasto (latência) de transmissão do pacote. Logo após o valor é multiplicado por 1000 para ser expresso em Mbps.

5.2 MEDIDAS ESTATÍSTICAS NAS MÉTRICAS DE AVALIAÇÃO

Além dos valores de vazão e latência medidos durante a transmissão de um pacote, o ambiente de avaliação de tráfego proposto por Tedesco [TED05] propõe a utilização de outras medidas durante a avaliação do tráfego. A seguir detalham-se as medidas utilizadas e como seu cálculo é feito.

5.2.1 LATÊNCIA

O ambiente de avaliação de tráfego faz uso de duas medidas de latência para um conjunto de pacotes: a latência média e o desvio padrão da latência. Além disso, o ambiente também calcula a latência ideal de transmissão de um pacote na rede, a partir de características da rede e do caminho utilizado durante o roteamento.

5.2.1.1 LATÊNCIA IDEAL DE UM PACOTE

A latência ideal de um pacote informa o tempo mínimo de transmissão de um pacote da origem ao destino, calculado a partir das características da rede. No escopo do ambiente de avaliação de tráfego, a latência ideal é utilizada para comparar a latência medida de um pacote com um tempo mínimo de transmissão, que pode em si ser uma medida otimista.

O cálculo da latência ideal considera uma série de características da rede, entre estas o número de roteadores existentes entre a origem e o destino, a frequência de operação de cada

um dos roteadores e o algoritmo de roteamento utilizado. Devido à necessidade de se conhecer durante a avaliação os roteadores do caminho, o cálculo da latência ideal só pode ser feito com precisão para roteamento determinístico, uma vez que é impossível conhecer exatamente o caminho usado em roteamento adaptativo. O tempo de transmissão de um pacote em cada roteador é composto pelo tempo gasto pelo primeiro *flit* do pacote para realizar o roteamento, que deve ser somado ao tempo de transmissão dos demais *flits* do pacote. Primeiramente detalha-se o cálculo do tempo de roteamento, o processo mais complexo de ser calculado. O primeiro passo executado no cálculo é verificar que roteadores se encontram entre a origem e o destino do pacote. Em seguida, verificam-se quais são as filas síncronas (HERMESGS) e quais são as filas bi síncronas (HERMESG) utilizadas entre os roteadores. Ao todo podem existir duas combinações de roteadores, sendo que cada uma delas consome um tempo diferente para realizar a escrita, a sincronização do dado e a requisição de roteamento.

Uma primeira combinação faz uso da fila síncrona (HERMESGS) conectada a outra fila síncrona (HERMESGS), o que consome cinco ciclos de relógio da frequência do roteador para escrever, sincronizar e realizar o roteamento do primeiro *flit* do pacote. Este valor em ciclos é convertido para tempo absoluto, utilizando o valor da frequência do roteador.

A segunda combinação faz uso da fila bi-síncrona (HERMESG), que pode estar conectada a uma fila síncrona (HERMESGS) ou a uma fila bi-síncrona (HERMESG). Este caso consome um tempo não determinístico entre a escrita do primeiro *flit* e a sincronização da fila. Por regra, a fila consome meio ciclo de relógio da frequência de escrita para escrever o primeiro *flit* do pacote, e gasta um tempo não determinístico, que no pior caso pode levar até um ciclo de relógio da frequência de leitura para sincronização da fila. Sendo assim, este trabalho assume o pior caso para o cálculo da latência ideal, considerando os efeitos não determinísticos causados pela sincronização da fila. Podem assim existir cenários em que o tempo medido para transmissão de um pacote seja menor que a latência ideal calculada. No total, o tempo gasto para escrever, sincronizar e rotear o primeiro pacote nesta combinação de roteadores é, no pior caso, de meio ciclo de relógio da frequência de escrita e sete ciclos de relógio da frequência de leitura. Quatro ciclos são gastos para o roteamento e três para a sincronização dos ponteiros da fila. Este valor em ciclos é convertido para tempo absoluto, utilizando os valores das frequências de leitura e de escrita do roteador.

Uma vez conhecidos os roteadores e seus respectivos tempos gastos para transmitir o primeiro pacote, é feito um somatório dos tempos gastos em cada um dos roteadores existentes onde é obtido o tempo ideal de transmissão do primeiro *flit*. Para os demais *flits* que compõem um pacote é feita uma busca pelo componente com a menor frequência entre a origem e o destino, sendo que se considera tanto os roteadores quanto o transmissor e o receptor do caminho. A menor frequência encontrada é convertida em tempo absoluto e multiplicada pelos demais *flits* que compõem um pacote, uma vez que depois de realizada a conexão entre a origem e o destino, cada *flit* irá gastar um ciclo da frequência mais lenta entre a origem e o destino para ser transmitida. O tempo obtido para os demais *flits* do pacote é somado ao tempo do primeiro *flit* do pacote, obtendo assim a latência ideal total do pacote.

5.2.1.2 LATÊNCIA MÉDIA DE UM FLUXO DE PACOTES

A latência média de um conjunto de pacotes é uma medida que representa o tempo médio gasto por um conjunto de pacotes para ser transmitido. O cálculo da média, ilustrado na Figura 47 é feito a partir da divisão do somatório de todos os valores de cada uma das latências dos pacotes medidas durante a transmissão de um determinado fluxo de pacotes origem-destino do tráfego pelo número total de pacotes.

$$\text{Latência média} = \frac{\sum \text{Latência de pacote}}{\text{Número de Pacotes}}$$

Figura 47: Equação que calcula a latência média de um conjunto de pacotes a partir do somatório das latências medidas dos pacotes dividido pelo número total de pacotes.

5.2.1.3 DESVIO PADRÃO DA LATÊNCIA DE UM FLUXO DE PACOTES

O desvio padrão da latência de um conjunto de pacotes é uma medida que representa a dispersão das latências dos pacotes em relação à média. Quanto menor for este valor, menor será a variação nos tempos de transmissão dos pacotes, indicando a uniformidade (ou não) da transmissão de um conjunto de pacotes. O cálculo do desvio padrão é ilustrado na Figura 48. Ele é feito a partir da raiz quadrada da divisão de um somatório. O somatório é feito sobre as latências dos pacotes, cada uma sendo previamente subtraído do valor da latência média dos pacotes, valor este elevado ao quadrado antes de ser acumulado. Após o somatório é dividido pelo número total de pacotes, antes de se aplicar a raiz quadrada.

$$\text{Desvio padrão da Latência} = \sqrt{\frac{\sum (\text{Latência pacote} - \text{média})^2}{\text{Número de Pacotes}}}$$

Figura 48 Equação que calcula o desvio padrão da latência de um conjunto de pacotes.

5.2.2 VAZÃO

O ambiente de avaliação de tráfego faz uso de duas medidas no cálculo da vazão de um conjunto de pacotes: a vazão média e o desvio padrão da vazão. Além disso, o ambiente também calcula a vazão ideal de transmissão de um pacote na rede, a partir das características de uma rede e do caminho utilizado durante o roteamento.

5.2.2.1 VAZÃO IDEAL DE UM PACOTE

A vazão ideal informa a quantidade de informação possível de ser transmitida entre um caminho da rede em um intervalo de tempo, calculado a partir das características da rede. No

escopo deste trabalho, a vazão ideal é utilizada para avaliar a vazão medida de um conjunto de pacotes contra a vazão ideal do caminho, que representa a quantidade máxima de informação possível de ser transmitida na unidade de tempo.

A vazão ideal de um pacote é calculada conforme a Figura 49, pela divisão do tamanho do pacote em bits pela latência ideal do pacote calculada conforme detalhado na Seção 5.1.1.1. O valor depois de calculado é multiplicado por 1000 para ser convertido em Mbps.

$$\text{Vazão ideal do pacote (Mbps)} = \left(\frac{\text{Tamanho do pacote (bits)}}{\text{Latência ideal do pacote (ns)}} * 1000 \right)$$

Figura 49: Equação que calcula a vazão ideal de um pacote pela divisão do tamanho do pacote em bits pela latência ideal do pacote em tempo absoluto. Depois de calculado o valor é multiplicado por 1000 para ser convertido em Mbps.

5.2.2.2 VAZÃO MÉDIA DE UM FLUXO DE PACOTES

A vazão média de um conjunto de pacotes é uma medida que representa a quantidade média de informação transmitida por um conjunto de pacotes. O cálculo da vazão média é feito usando a Equação da Figura 50, a partir da divisão do somatório de todos os valores das vazões calculadas de um conjunto de pacotes pelo número total de pacotes. O cálculo da vazão de cada pacote é feito conforme detalha a Seção 5.1.2, a partir da latência medida de cada pacote.

$$\text{Vazão média} = \frac{\sum \text{Vazão de pacote}}{\text{Número de Pacotes}}$$

Figura 50: Equação que calcula a vazão média de um conjunto de pacotes, a partir do somatório das vazões calculadas dos pacotes dividido pelo número total de pacotes.

5.2.2.3 DESVIO PADRÃO DA VAZÃO DE UM FLUXO DE PACOTES

O desvio padrão da vazão de um conjunto de pacotes é uma medida que representa a dispersão das vazões dos pacotes em relação a sua média. Quanto menor for este valor, menor será a variação da quantidade de informação transmitida pelos pacotes, indicando que a transmissão ocorreu de maneira uniforme para o conjunto de pacotes calculado. O cálculo do desvio padrão de um conjunto de pacotes é ilustrado na Figura 51. Ele é feito a partir da raiz quadrada da divisão do somatório das vazões dos pacotes subtraídas da média dos pacotes, valor este elevado ao quadrado antes da soma. A soma total é dividida pelo número total de pacotes antes de se calcular a raiz quadrada do valor assim obtido.

$$\text{Desvio Padrão da Latência} = \sqrt{\frac{\sum (\text{Vazão de pacote} - \text{média})^2}{\text{Número de Pacotes}}}$$

Figura 51: Equação que calcula o desvio padrão da vazão de um conjunto de pacotes.

5.3 AMBIENTE DE AVALIAÇÃO DE TRÁFEGO

A partir das métricas e das medidas descritas pelas Seções 5.1 e 5.2 o ambiente de avaliação de tráfego originalmente proposto Tedesco [TED05] foi adaptado para dar suporte aos cálculos e à avaliação de tráfego para redes HERMES-G. O ambiente de avaliação de tráfego faz uso do arquivo de cenário do tráfego, definido na Seção 4.2.1, como arquivo de entrada. Este permite que para o mesmo projeto de rede haja vários cenários já simulados, sendo possível avaliar cada um de maneira independente a partir do cenário de tráfego. Ao todo, quatro ferramentas que permitem avaliar o tráfego foram modificadas: (i) Distribuição de vazões; (ii) Distribuição de latências; (iii) Analisador de latências; (iv) Relatório global. Estas ferramentas foram adaptadas para dar suporte ao cálculo do tráfego para redes HERMES-G, e serão detalhadas a seguir.

5.3.1 DISTRIBUIÇÃO DE VAZÕES DE UM TRÁFEGO

A ferramenta de distribuição de vazões através de gráficos como o ilustrado na Figura 52 disponibiliza a visualização dos valores de vazão medidas de um determinado conjunto de pacotes entre um transmissor e um receptor.

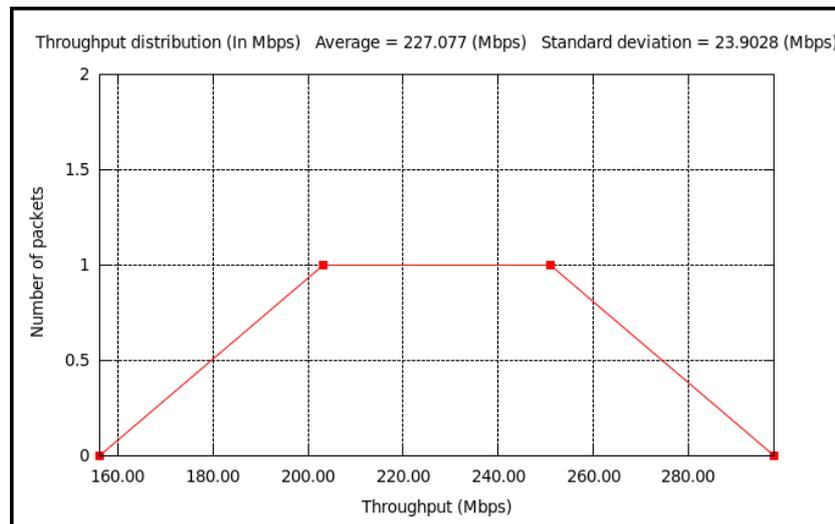


Figura 52: Interface da ferramenta distribuição de vazões do ambiente de avaliação de tráfego. No exemplo, é usado um tráfego uniforme de 400 Mbps de dois pacotes transmitidos do endereço 00 ao endereço 11.

Dentre os valores relacionados estão a vazão de cada um dos pacotes, calculada a partir do método descrito na Seção 5.1.2, a média das vazões dos pacotes, calculada a partir do método descrito na Seção 5.2.2.2 e o desvio padrão da vazão, calculado a partir do método descrito na

Seção 5.2.2.3. A Figura 52 descreve um tráfego uniforme injetado a 400 Mbps de dois pacotes enviados do endereço 00 ao endereço 11. A partir dos campos disponibilizados pela ferramenta de distribuição de vazões é possível avaliar que para o tráfego exemplo utilizado, um pacote teve vazão de aproximadamente 200 Mbps e o outro pacote vazão de 250 Mbps. Além disso, a média dos valores foi de 227.0 Mbps e o desvio padrão de 23.9 Mbps.

5.3.2 DISTRIBUIÇÃO DE LATÊNCIAS DE UM TRÁFEGO

A ferramenta distribuição de latências através de gráficos como o ilustrado na Figura 53 disponibiliza a visualização dos valores de latência de um determinado conjunto de pacotes enviados entre um transmissor e um receptor. Dentre os valores relacionados estão a latência de cada um dos pacotes, calculada a partir do método descrito na Seção 5.1.1, a média das latências, calculada a partir do método descrito na Seção 5.2.1.2 e o desvio padrão da vazão, calculado a partir do método descrito na Seção 5.2.1.3. A Figura 53 descreve um tráfego uniforme injetado a 400 Mbps de dois pacotes do endereço 00 ao endereço 11. A partir dos campos disponibilizados pela ferramenta de distribuição de latência é possível observar que para o tráfego exemplo utilizado, um pacote teve latência de aproximadamente 1020 ns e o outro pacote latência de 1260 ns. Além disso, a média dos valores foi de 1140 ns e o desvio padrão de 120 ns.

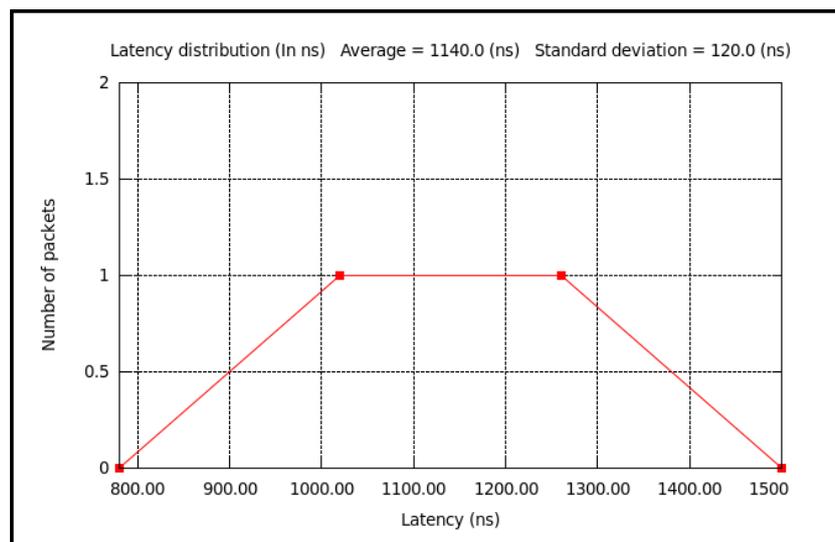


Figura 53: Interface da ferramenta distribuição de latências do ambiente de avaliação de tráfego. O exemplo usa um tráfego uniforme de 400 Mbps de dois pacotes transmitidos do endereço 00 ao endereço 11.

5.3.3 ANALISADOR DE LATÊNCIAS

A ferramenta analisadora de latências gera saída como a ilustrada na Figura 54. Esta permite a visualização de um conjunto de pacotes de todos os tráfegos que levaram mais ou menos tempo para serem transmitidos. Para todo valor de latência informa-se também o número de sequência e o endereço de origem e de destino do pacote. A latência dos pacotes é calculada a partir do método descrito na Seção 5.1.1. Os valores de número de sequência, origem e destino são capturados diretamente dos pacotes nos arquivos do tráfego.

	Latency	Sequency Number	Source	Target
1	2540.0	12	10	20
2	2420.0	11	10	20
3	2300.0	10	10	20
4	2180.0	9	10	20
5	2060.0	8	10	20
6	1940.0	7	10	20
7	1820.0	6	10	20
8	1700.0	5	10	20
9	1400.0	2	00	20
10	1140.0	4	10	20

Figura 54: Interface da ferramenta analisador de latências do ambiente de avaliação de tráfego, que permite visualizar latências específicas de pacotes de tráfegos.

5.3.4 RELATÓRIO GLOBAL

A ferramenta de relatório global, ilustrada na Figura 55, permite avaliar todos os tráfegos transmitidos entre cada par de endereços origem e destino na rede de duas formas. A primeira dela apresenta os valores de vazão e latência ideal de cada tráfego, calculados conforme os métodos descritos nas Seções 5.2.1.1 e 5.2.2.1. A segunda apresenta valores medidos de cada tráfego respectivo: a média da vazão e da latência, o desvio padrão da vazão e da latência e os valores de vazão e latência mínima e máxima. Além disso, o ambiente disponibiliza para cada tráfego dois botões que permitem visualizar gráficos de distribuição de vazões e latências detalhadas nas Seções 5.3.2 e 5.3.1.

HermesG NoC Flow Control: CreditBased (1 cycle per flit) Virtual Channels: 0 Scheduling: RoundRobin Dimensions: 3 x 3 (9 routers) Flit Width: 16 Buffer Depth: 16 Routing Algorithm: Algorithm XY															
Generated					Measured									Graphs	
Source	Target	Packets	Throughput In (Mbps)	Latency In (ns)	Packets	Throughput In (Mbps)				Latency In (ns)				Latency Distribution	Throughput Distribution
			Ideal(WC)	Ideal(WC)		Average	Standard Deviation	Minimum	Maximal	Average	Standard Deviation	Minimum	Maximal	VIEW	VIEW
00	20	2	341.33	750.0	2	216.91	34.06	182.85	250.98	1210.0	190.0	1020.0	1400.0	VIEW	VIEW
10	20	10	426.66	600.0	10	164.87	98.11	100.78	441.37	1868.0	574.60	580.0	2540.0	VIEW	VIEW
		12			12	190.89				1539.0					

Report generated on 16:6:37 Tuesday, July 24, 2012, AD by TrafficMeasurer

Figura 55: Interface da ferramenta relatório global que permite visualizar latência e a vazão ideais e vazão e a latência medidas de todos os tráfegos.

6 RESULTADOS OBTIDOS

Este Capítulo apresenta um conjunto de experimentos que visam validar os conceitos propostos aqui e avaliar de maneira resumida alguns cenários de rede e de tráfego apresentados nos capítulos anteriores. Com os estudos de caso desenvolvidos, pretende-se demonstrar como é possível aumentar o desempenho de um cenário de tráfego proposto, variando as características da rede como o tipo de roteamento utilizado e a frequência de operação. Além disso, realizam-se comparações de área para diversas variações da rede. HERMES-G.

6.1 ESTUDO DE CASO 1

Este estudo ilustrado pela Figura 56 analisa o impacto resultante no desempenho de tráfegos concorrentes em dois modelos de rede, uma operando totalmente de maneira síncrona e outra operando de maneira não síncrona com duas ilhas de frequência. O cenário sugerido faz uso de uma rede com dimensão 3x3, totalizando nove roteadores onde quatro deles conectam-se a processadores, dois conectam-se a elementos de memória, e três a elementos de entrada e saída. Este estudo de caso foi proposto seguindo tendências de projeto de SOC atuais, contendo quatro processadores operando em uma ilha de frequência mais alta, diferente dos componentes de memória e entrada e saída que operam em outras ilhas de frequência.

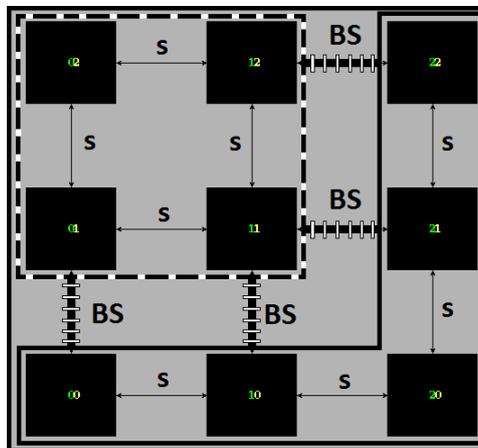


Figura 56: Rede 3x3 contendo duas ilhas de frequência. O quadrado pontilhado simula quatro processadores operando a 500 MHz. Neste domínio todos os roteadores se comunicam utilizando filas síncronas (S). O restante dos roteadores (polígono preto) simula duas memórias e três controladores de entrada e saída, operando a 50 MHz cada. Repare que as filas bi síncronas (BS) são utilizadas para comunicação unicamente entre os diferentes domínios de frequência.

Em ambos os modelos de rede utilizou-se largura de canal 16 bits, profundidade de filas de 16 *flits*, algoritmo de roteamento *Negative First Non Minimal* e codificação Johnson para ponteiros de filas. Neste cenário, optou-se por trabalhar unicamente com filas bi síncronas, uma vez que utilizando um cenário totalmente síncrono com filas síncronas do tipo HERMES, o tempo de sincronização e transmissão da fila influenciaria nos resultados. Os processadores são mapeados nos endereços (01/02/11/12), os elementos de memória nos endereços (00/20) e os elementos de

entrada e saída nos roteadores (10/21/22). O primeiro modelo de rede faz uso de um único domínio de frequência em todos os roteadores, transmissores e receptores. O segundo modelo faz uso de dois domínios de frequência, conforme ilustra a Figura 56, o primeiro deles no quadrado pontilhado onde os processadores estão mapeados, faz uso de uma frequência de 500 MHz para os roteadores, transmissores e receptores. Já no polígono preto, onde estão mapeados os elementos de memória e de entrada e saída é utilizada uma frequência de 50 MHz para os roteadores, transmissores e receptores.

O cenário de tráfego sugerido segue a definição do estudo de caso, onde três transmissões são utilizadas. A primeira transmissão propõe que um processador mapeado no endereço 01 envie a outro processador mapeado no endereço 02 50 pacotes de 16 *flits* cada, ou seja, 12800 bits transmitidos a uma taxa de injeção uniforme de 600 Mbps, o que representa 7.5% da capacidade de transmissão do processador. Esta transmissão poderia ser representada na prática como uma migração de uma tarefa, sendo que o processo de migração é executado de maneira concorrente no processador, uma vez que pouco mais de 7% da capacidade de transmissão do processador é utilizada.

A segunda transmissão propõe que um dos elementos de memória, mapeado no endereço 20 envie a um dos processadores, mapeado no endereço 02, 100 pacotes de 16 *flits* cada, ou seja, 256000 bits transmitidos a uma taxa de injeção exponencial decrescente em um intervalo de 200 Mbps a 800 Mbps com taxa média de 400 Mbps, o que representa respectivamente um intervalo de 25% a 100% e com média de 50% da taxa máxima de transmissão da memória. Esta transmissão poderia ser representada na prática como um elemento de memória de maneira concorrente a outros processadores transmitindo valores de seus respectivos endereços para o processador mapeado no endereço 12. Aqui, por existir concorrência no acesso à memória por outros processadores, a distribuição temporal do tráfego é dada como uma distribuição exponencial decrescente. Nesta, grande parte do total de pacotes é transmitida a uma taxa de injeção mais baixa, e poucos pacotes são transmitidos a uma taxa de injeção mais alta.

A terceira transmissão propõe que um dos elementos de memória mapeado no endereço 20 envie a um dos elementos de entrada e saída, mapeado no endereço 22 100 pacotes de 16 *flits* cada. Assim, transmitem-se 256000 bits a uma taxa de injeção normal variada em um intervalo de 200 Mbps a 600 Mbps com taxa média de 400 Mbps. Isto representa um intervalo de 25% a 75% e com média de 50% da taxa máxima de transmissão do elemento de memória. Esta transmissão poderia ser representada na prática como uma conexão com uma memória não volátil do tipo EEPROM ou *Flash*, responsável por armazenar de maneira permanente uma informação já computada pelos processadores do circuito que está presente em uma das memórias do circuito. Utiliza-se aqui distribuição normal, descrevendo um comportamento em que grande parte dos pacotes é transmitida com taxa próxima da média. Em um cenário real de uma EEPROM externa, existiriam atrasos durante a sincronização e o início da recepção, e então grande parte dos pacotes seria transmitida com taxa próxima do valor da média.

Os resultados obtidos após realizar a simulação dos modelos de rede síncrona e não síncrona são ilustrados na Tabela 4. Com relação à vazão e latência médias de transmissão do primeiro

cenário (migração de uma tarefa) entre processadores, representados na Tabela 4 pelo item (P - P) 01-02, detectou-se que o cenário não síncrono gastou 76% menos tempo para migrar a tarefa e conseguiu atingir uma vazão 73% maior que o cenário síncrono para migrar a tarefa. Este comportamento era previsível, uma vez que no cenário não síncrono os roteadores 01 e 02 responsáveis pela transmissão do tráfego estavam operando em 500 MHz, diferente do cenário síncrono onde os roteadores estavam operando em 50 MHz.

No segundo cenário, referente à comunicação de um elemento de memória com um processador, ilustrado através da Tabela 4 pelo item (M-P) 00-12, foi detectado com relação a latência que o cenário não síncrono gastou 33% menos tempo para transmitir os dados e com relação à vazão que o cenário não síncrono conseguiu atingir uma vazão 32% maior que o cenário síncrono para transmitir os valores da memória para o processador. Este comportamento era previsível, uma vez que o tráfego (P-P) 01-02 causa contenção, obrigando o roteamento adaptativo a utilizar os roteadores (00/10/11/12) para realizar a comunicação. No cenário síncrono, todos os roteadores possuem a mesma frequência. No cenário não síncrono os roteadores (11/12), que conectam os processadores possuem frequências 10 vezes maiores à frequência utilizada no cenário síncrono, o que justifica a existência de 33% redução na latência comparando ambos os cenários.

Tabela 4: Valores de vazão e latência média de todos os pacotes de cada tráfego, obtidos durante a simulação dos tráfegos para os modelos de rede síncrona e não síncrona.

Cenário de tráfego	Latência Média Síncrona (ns)	Vazão Média Síncrona (Mbps)	Latência Média Não Síncrona (ns)	Vazão Média Não Síncrona (Mbps)
(M-P) 00-12	988	261	659	388
(M-I/O) 20-22	780	328	862	306
(P-P) 01-02	1444	202	339	755
Média	1071	264	620	483

O terceiro cenário refere-se à comunicação de um elemento de memória com um elemento de entrada e saída. Isto é ilustrado na Tabela 4, item (M-I/O) 20-22. Detectou-se que com relação à latência o cenário síncrono gastou 9.5% menos tempo e com relação à vazão o cenário síncrono conseguiu atingir uma vazão 6% maior que o cenário não síncrono para transmitir os dados da memória para o controlador de entrada e saída. Era de se esperar a obtenção de resultados próximos para estes cenários de tráfego, uma vez que os roteadores que caracterizam o caminho entre a memória e o controlador de entrada e saída operam na mesma frequência.

A Figura 57 ilustra através de um gráfico os valores da latência média de cada um dos cenários de tráfego propostos. Por fim, foi calculada a média da latência a partir das médias de cada um dos tráfegos propostos para ambos os modelos de rede. A latência média da rede não síncrona foi 42% menor que a rede síncrona.

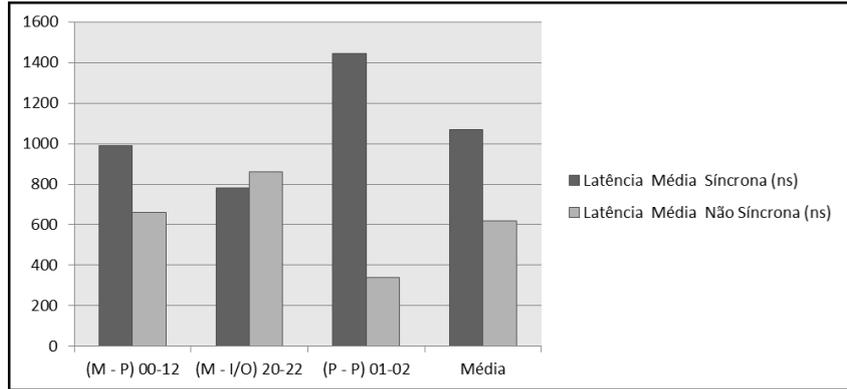


Figura 57: Latência média dos três cenários de tráfego propostos para o modelo de rede síncrona e o modelo de rede não síncrona.

A Figura 58 mostra os valores da vazão média de cada um dos cenários de tráfego propostos. Por fim, calculou-se a média da vazão a partir das médias de cada um dos tráfegos propostos para ambos os modelos de rede. A vazão média da rede não síncrona foi 45% maior que a rede síncrona.

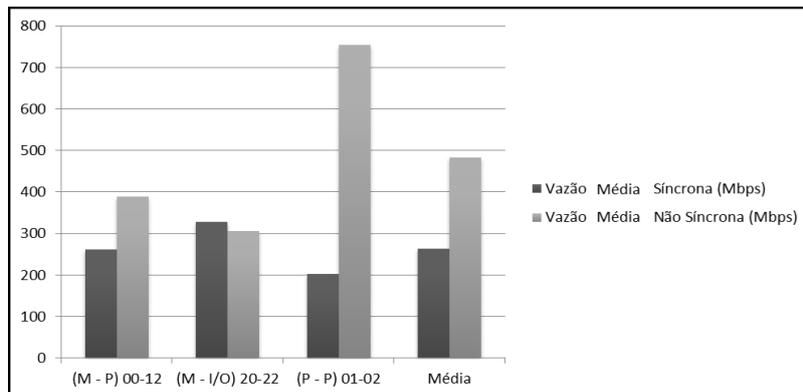


Figura 58: Vazão média dos três cenários de tráfego propostos para o modelo de rede síncrona e o modelo de rede não síncrona.

A partir dos resultados, conclui-se que o modelo não síncrono foi aquele que na média dos tráfegos conseguiu atingir 42% menos latência e 45% maior vazão. Chama a atenção o fato de redes que utilizam algoritmos de roteamento adaptativos, além de aceitar tráfegos concorrentes não fiquem limitadas a um único caminho possível entre um endereço origem e destino. Elas podem inclusive explorar em redes não síncronas a capacidade de transmissão de outros roteadores que operam em frequências mais elevadas, caso esta informação esteja disponível ao algoritmo de roteamento. O cenário de tráfego ilustrado na Tabela 4 pelo item (M-P) 00-12 mostra ganhos de 33% em latência e 32% em vazão, comparado a uma rede síncrona, por encontrar-se contenção na rede e fazer uso de caminhos alternativos, formado por roteadores operando em frequências mais elevadas.

6.2 ESTUDO DE CASO 2

Este estudo de caso propõe estudar o impacto no tempo de transmissão e no custo de área variando a codificação dos ponteiros das filas bi síncronas e no uso da fila síncrona descrita na

Seção 3.1.2.5, utilizada para geração de redes não síncronas. Ao todo, quatro modelos de rede são propostos, sendo que em todos eles utilizam-se as mesmas características: dimensão de duas linhas por duas colunas, comprimento de canais igual a 16bits, profundidade das filas igual 16 *flits* e algoritmo de roteamento XY. A frequência de operação de todos os roteadores, transmissores e receptores é de 50 MHz, com exceção do roteador 11 que possui frequência de operação de 100 MHz. O primeiro e o segundo modelos usam codificação Gray e codificação Johnson respectivamente, em todas as filas bi síncronas dos roteadores. O terceiro e o quarto modelos usam codificação Gray e fila bi síncrona e codificação Johnson com fila bi síncrona. A Figura 59 descreve em onde se usa filas bi síncronas e filas síncronas.

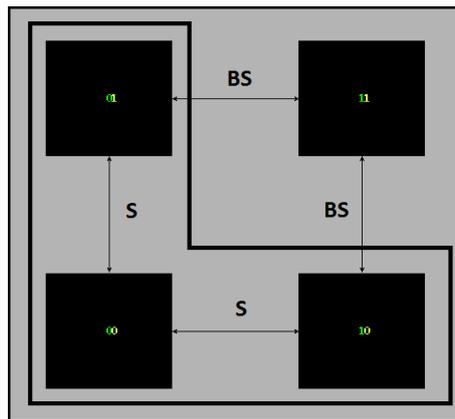


Figura 59: Rede 2x2 contendo duas ilhas de frequência, uma formada pelos roteadores (00/10/01) onde a fila síncrona é utilizada para comunicação entre os roteadores. O roteador 11, por possuir frequência de operação diferente dos demais usa filas bi síncronas.

O cenário de tráfego sugerido para os quatro modelos de rede explora o tempo gasto quando os dois tipos de filas são utilizados em uma transmissão. O primeiro cenário de tráfego propõe que o transmissor conectado ao roteador 00 envie ao roteador 11 a quantidade de 10 pacotes de 16 *flits* cada, totalizando um tráfego de 2560 bits distribuídos de maneira uniforme a 400 Mbps, ou 50% da capacidade do transmissor. O segundo cenário de tráfego propõe que o transmissor conectado no roteador 11 envie ao roteador 00 a quantidade de 10 pacotes de 16 *flits* cada, totalizando um tráfego de 2560 bits distribuídos de maneira uniforme a 800 Mbps, ou 100% da capacidade do transmissor. A Tabela 5 apresenta os resultados de área obtidos.

Tabela 5: Resultado de área para cada um dos modelos de rede, em LUTs de quatro entradas.

Modelo de rede	Área (em LUTs de quatro entradas)
2x2G	2061
2x2GS	1171
2x2J	2416
2x2JS	1275

Para cada modelo de rede realizou-se a síntese usando a ferramenta Xilinx ISE 10.1 voltada para um dispositivo FPGA Virtex5 XC5VLX330T. Este dispositivo foi escolhido por existirem disponíveis placas de prototipação com este FPGA no laboratório de pesquisa do grupo do autor. Os resultados de área são obtidos em número de LUTs de quatro entradas utilizadas pelo FPGA para representar o projeto de cada rede. A partir dos resultados de área coletados é possível concluir que a codificação Johnson conforme descrito na linha 2x2J da Tabela 5 apresenta

aumento de 14,6% de área em relação a codificação Gray, descrita na linha 2x2G da Tabela 5. Estes resultados se assemelham aos obtidos por [HEC11], conforme descrito na Seção 3.1.2.3, onde se compara a área de cada uma das codificações propostas. Já para os modelos de rede que utilizam a combinação de filas síncronas e bi síncronas, no cenário Johnson conforme descrito pelas linhas 2x2J e 2x2JS da Tabela, houve uma redução de 47% na área da versão que utiliza filas síncronas e bi síncronas comparada a versão que só faz uso de filas bi síncronas. No cenário Gray descrito nas linhas 2x2G e 2x2GS, houve uma redução de 43% na área da versão que utiliza filas síncronas e bi síncronas comparada a versão que só faz uso de filas bi síncronas. O autor esperava obter resultados de redução na área utilizando combinações de filas síncronas e bi síncronas, uma vez que [PON08] afirma que existe aumento na área de uma fila bi síncrona comparado a uma fila síncrona.

A Tabela 6 descreve os resultados obtidos após realizar a simulação dos tráfegos para os modelos de rede propostos. Como as filas com codificação Gray e codificação Johnson gastam o mesmo tempo para transmissão, os resultados obtidos para ambas as filas foram os mesmos, conforme mostra a Tabela 6. Nos casos onde é se emprega filas síncronas e filas bi síncronas ao mesmo tempo, detectou-se existir uma redução de 12% na latência e um aumento de 12% na vazão para o tráfego (00-11) e uma redução de 4% na latência e um aumento de 4% na vazão para o tráfego (11-00). O autor esperava obter resultados próximos aos medidos com relação à redução na latência e aumento na vazão, uma vez que esta diferença de tempo só é computada durante o chaveamento do pacote.

Tabela 6: Valores de vazão e latência média de todos os pacotes dos tráfegos (00-11) e (11-00) obtidos durante a simulação dos tráfegos para os quatro modelos de rede propostos.

Cenário de tráfego	2x2G		2x2GS		2x2J		2x2JS	
	Latência Média (ns)	Vazão Média (Mbps)						
00-11	667	384	587	436	667	384	587	436
11-00	1272	220	1232	229	1272	220	1232	229
Média	969	302	909	333	969	302	909	333

A Figura 60 ilustra através de um gráfico os valores da latência e vazão média de cada um dos cenários de tráfego propostos aqui. Por fim, calcula-se a média da vazão e da latência a partir das médias de cada um dos tráfegos propostos para ambos os modelos de rede. O modelo de rede que faz uso da fila síncrona e da fila bi síncrona, comparado ao modelo que faz uso unicamente da fila bi síncrona apresentou redução na média de todos os tráfegos ganhos de 6% na redução do tempo de transmissão e aumento na vazão em 10%.

A partir dos resultados, conclui-se, a partir dos modelos de rede propostos, que a codificação de ponteiros Gray apresenta 14,6% menor área comparado à codificação Johnson, sendo que ambas possuem o mesmo desempenho. Nas questões relacionadas ao uso de filas, os resultados apontam redução de 47% na área da rede utilizando filas bi síncronas com codificação Johnson e filas síncronas, e redução de 43% na área da rede utilizando filas bi síncronas com codificação Gray e filas síncronas. Além dos resultados de área, a combinação de filas síncronas e bi síncronas

possibilitou uma redução de 6% na latência dos pacotes e um aumento de 10% na vazão do tráfego.

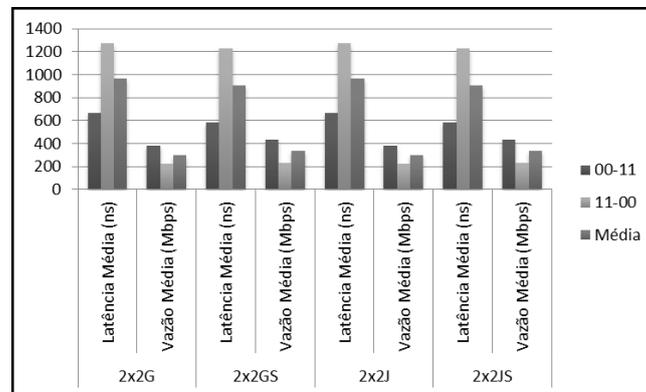


Figura 60: Vazão e latência média dos cenários de tráfego (00-11) e (11-00) para os modelos de rede que fazem uso unicamente de filas bi síncronas (2x2J – 2x2G) e para os modelos que fazem uso de filas síncronas e bi síncronas (2x2GS – 2x2JS).

6.3 ESTUDO DE CASO 3

Este estudo de caso propõe demonstrar é possível gerar redes menores com desempenho equivalente a redes maiores, a partir da variação da frequência dos roteadores. Neste estudo de caso propõem-se duas redes, a primeira com dimensões 2x2, comprimento de canais de 32 bits, profundidade das filas igual a 16 *flits*, algoritmo de roteamento XY, codificação de ponteiros Johnson para a fila bi síncrona e frequência de operação de 50 MHz para todos os roteadores, transmissores e receptores do tráfego. A segunda rede possui dimensões 2x2, comprimento de canais igual a 16 bits, profundidade das filas igual a 16 *flits*, algoritmo de roteamento XY e codificação de ponteiros Johnson para a fila bi síncrona. A frequência de operação para os roteadores (00/10/11) foi definida em 500 MHz, os demais roteadores, transmissores e receptores operam na frequência de 50 MHz. Apenas um cenário de tráfego sem concorrência foi definido, onde o transmissor do endereço 00 envia 10 pacotes de 16 *flits*, ou seja, 2560 bits para o endereço 11 a uma taxa de injeção uniforme de 800 Mbps, o que representa 100% da taxa de injeção do transmissor. A Tabela 7 apresenta quais foram os resultados de área obtidos para cada modelo de rede proposto através da síntese com a ferramenta Xilinx ISE 10.1 para um FPGA Virtex5 XC5VLX330T. Os resultados de área são dados em número de LUTs de quatro entradas.

Tabela 7: Resultados de área obtidos para cada uma das redes propostas, em LUTs de quatro entradas.

Modelo da rede	Área em número de LUTs de quatro entradas
Primeiro Modelo (32bits)	3582
Segundo Modelo (16bits)	2424

A partir dos resultados de área é possível concluir que há um aumento de 32% na área no primeiro modelo de rede, que faz uso de roteadores com largura de canal de 32bits, comparado ao segundo modelo, que faz uso de largura de canal de 16bits. Não somente o número de fios

aumenta do primeiro para o segundo tipo de roteador, como também a área ocupada pelas filas, devido à mudança no tamanho do *flit*.

A Tabela 8 descreve os resultados obtidos após realizar a simulação do tráfego para ambos os modelos de rede propostos. O segundo modelo de rede que dá suporte a duas frequências é 50% menor. Ainda assim, gera 52% de ganho em latência e 6% mais vazão, comparado ao primeiro modelo. O autor esperava obter resultados com relação à vazão e latência próximos, porém, o aumento a frequência no segundo modelo, mesmo possuindo largura de canais entre os roteadores 50% menor que o primeiro modelo resultou em resultados acima do esperado com relação à latência.

Tabela 8: Valores de vazão e latência médias do tráfego (00-11), obtidos durante a simulação dos modelos de rede propostos.

Cenário de tráfego	Primeiro Modelo (32bits)		Segundo Modelo (16bits)	
	Latência Média (ns)	Vazão Média (Mbps)	Latência Média (ns)	Vazão Média (Mbps)
00 – 11	720	711	339	755

A partir dos resultados, conclui-se que o cenário que propõe alterar a frequência foi aquele que além de consumir 32% menos área por fazer uso de uma rede menor conseguiu atingir 52% menor latência e 6% maior vazão. Mesmo considerando-se as questões relacionadas ao aumento da frequência e ao consumo de energia, e também que o estudo de caso propõe diferença na frequência da ordem de 10 vezes entre os modelos de rede, este estudo de caso deixa claro que a frequência apresenta em alguns casos vantagens. Isto vale para a geração de redes menores e capazes de atender com maior eficiência um tráfego sem comprometer a área como em outras propostas que sugerem variar o tamanho dos canais ou a profundidade das filas.

6.4 ESTUDO DE CASO 4

Este estudo de caso propõe avaliar o impacto na área e no desempenho ao utilizar diferentes algoritmos de roteamento do ambiente de geração de redes, conforme detalhado na Seção 3.1.2.2. Ao todo são utilizadas sete redes, todas compostas pelas seguintes características, tamanho 3x3, largura de canais que interligam os roteadores de 32bits, profundidade das filas igual a 8 *flits*, codificação de ponteiros Johnson utilizado pela fila bi síncrona e frequência de operação igual a 50 MHz para os roteadores, transmissores e receptores. Em cada uma das sete redes propostas é variado o algoritmo de roteamento, sendo eles: (i) XY; (ii) *Negative First Minimal*; (iii) *Negative First Non Minimal*; (iv) *North Last Minimal*; (v) *North Last Non Minimal*; (vi) *West First Minimal*; (vii) *West First Non Minimal*.

O cenário de tráfego sugerido para os sete modelos de rede visa utilizar a capacidade do algoritmo de roteamento de usar caminhos alternativos para o tráfego em caso de congestionamento. Sendo assim, quatro tráfegos são propostos, sendo que o primeiro deles transmite do endereço 00 para o endereço 22 a quantidade de 10 pacotes de 32 *flits* cada, totalizando um tráfego de 10240 bits distribuídos de maneira uniforme a uma taxa máxima de 800 Mbps. O segundo tráfego é transmitido do endereço 22 para o endereço 00 a quantidade de 10 pacotes de 32 *flits* cada, totalizando um tráfego de 10240 bits distribuídos de maneira uniforme a

taxa máxima de 800 Mbps. O terceiro tráfego transmite do endereço 02 para o endereço 20 a quantidade de 10 pacotes de 32 *flits* cada, totalizando um tráfego de 10240 bits distribuídos de maneira uniforme a uma taxa máxima de 800 Mbps. O quarto e último tráfego transmite do endereço 20 para o endereço 02 a quantidade de 10 pacotes de 32 *flits* cada, totalizando um tráfego de 10240 bits distribuídos de maneira uniforme a uma taxa máxima de 800 Mbps.

A Tabela 9 apresenta quais foram os resultados de área obtidos para cada um dos sete modelos de rede propostos através de síntese com a ferramenta Xilinx ISE 10.1 voltado para um FPGA Virtex5 XC5VLX330T. Os resultados de área são dados em LUTs de quatro entradas.

Tabela 9: Resultado de área obtida para cada um dos sete modelos de rede propostos em LUTs de quatro entradas.

Algoritmo de roteamento	XY	NFM	NFNM	NLM	NLNM	WFM	WFNM
Área em LUTs de quatro entradas	9207	3934	10487	10138	10416	10233	10512

A partir dos resultados de área obtidos da Tabela 9 para cada um dos sete modelos de rede propostos é possível concluir que o algoritmo de roteamento *Negative First Minimal* foi aquele que apresentou menor área, e o algoritmo *Negative First Non Minimal* aquele que apresentou maior área.

A Tabela 8 descreve os resultados obtidos após realizar a simulação do tráfego para ambos os modelos de rede propostos. O campo (Lat) representa a latência média em (ns) para cada um dos tráfegos, o campo (Vaz) representa a vazão média em (Mbps) para cada um dos tráfegos. A partir dos modelos de rede propostos, a rede que atingiu o menor valor de latência média e o maior valor de vazão média para todos os tráfegos propostos foi a rede que utilizou algoritmo de roteamento XY. Comparando com o cenário utilizando o algoritmo de roteamento *North Last Non Minimal* que gerou o maior valor de latência média e o menor valor de vazão média, o algoritmo XY gastou 2.65 vezes menos tempo para transmitir os tráfegos e obteve 50% maior vazão para transmitir os tráfegos. Em uma comparação direta de área entre os dois algoritmos, o algoritmo XY além de ser mais eficiente para o cenário proposto ocupa 11,6% menos área que o algoritmo *North Last Non Minimal*.

O cenário que consumiu menos área foi o que utiliza o algoritmo de roteamento *Negative First Minimal*, que comparado ao algoritmo XY apresentou 57,2% menos área, mas em contrapartida, o algoritmo XY apresenta em relação ao algoritmo *Negative First Minimal*, 31% menos tempo para transmitir os tráfegos e 30% maior vazão.

Tabela 10: Valores de vazão e latência média para cada um dos três tráfegos simulados nos sete modelos de redes propostas.

	XY		NFM		NFNM		NLM		NLNM		WFM		WFNM	
	Lat	Vaz												
00-22	2680	423	3414	347	4600	296	5996	238	5996	238	3414	347	3414	347
22-00	2780	404	3968	297	6430	231	3178	368	3178	368	3968	297	3968	297
02-20	2580	443	3856	292	7794	175	8300	176	8300	176	3856	292	3856	292
20-02	2680	423	4464	245	9588	141	5778	198	5778	198	4464	245	4464	245
Média	2680	423	3925	295	7103	211	5813	245	5813	245	3925	295	3925	295

A partir dos resultados, conclui-se que para o cenário de rede e de tráfego proposto, o algoritmo XY foi aquele que apresentou a melhor relação de desempenho comparado ao consumo de área para ser sintetizado. Já o algoritmo *Negative First Minimal* foi o que apresentou menor consumo de área, e resultados de desempenho acima da média comparado aos demais tráfegos. O algoritmo de roteamento que gastou mais área para ser sintetizado e apresentou o pior desempenho de todos foi o algoritmo *Negative First Non Minimal*.

6.5 ESTUDO DE CASO 5

Este estudo de caso propõe comparar dois cenários de tráfego, um a partir de um modelo de aplicação CDCM e outro através do modelo de tráfego sintético. Para ambos os modelos de tráfego utilizados foi definido um cenário de rede de tamanho 2x2, largura de canais igual a 16 bits, profundidade das filas igual a 8 *flits*, algoritmo de roteamento XY e codificação de ponteiro Johnson para a fila bi síncrona. Optou-se por gerar uma única frequência de operação de 50 MHz para todos os roteadores, transmissores e receptores.

O cenário de tráfego proposto é ilustrado na Figura 61, onde a aplicação proposta é particionada em quatro tarefas. Cada tarefa possui um tempo de processamento e uma quantidade de informação a ser transferida. O cenário de tráfego que faz uso do modelo de aplicação CDCM foi criado e mapeado utilizando o ambiente CAFES e o ambiente de geração do Testbench e do tráfego da ferramenta de geração de tráfego do ambiente ATLAS, conforme detalhado na Seção 4.3. Durante o processo de mapeamento, a tarefa A foi mapeada no endereço 00, a tarefa B no endereço 10, a tarefa C no endereço 01 e a tarefa D no endereço 11. A partir dos valores de endereços onde as tarefas foram mapeadas e do valor a ser transmitido de cada tarefa, foi então criado um tráfego sintético similar.

O tráfego sintético é composto por três transmissões, sendo a primeira do endereço 00 para o endereço 10, referente à tarefa (A→B) ilustrada na Figura 61

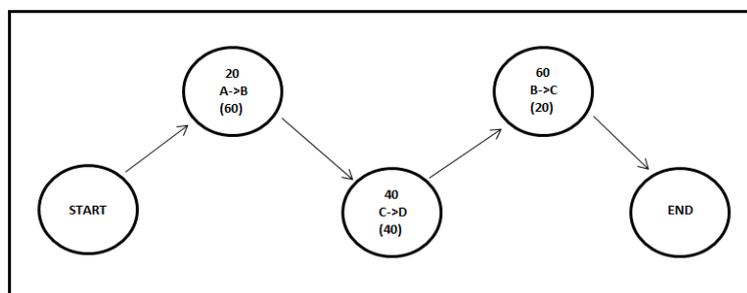


Figura 61: Cenário de tráfego proposto, formado por uma aplicação composta por quatro tarefas.

Transmite-se aqui um pacote de tamanho igual a 60 *flits*, o equivalente a 15360 bits a uma taxa uniforme de 800 Mbps, ou seja, 100% da capacidade do transmissor. A segunda transmissão é feita do endereço 10 para o endereço 01, referente à tarefa (C→D) conforme ilustrada pela Figura 61, que transmite um pacote de tamanho igual a 40 *flits*, o equivalente a 10240 bits a uma taxa uniforme de 800 Mbps, ou seja, 100% da capacidade do transmissor. A terceira e última transmissão é feita do endereço 01 para o endereço 11, referente à tarefa (B→C) conforme

ilustrado na Figura 61, que transmite um pacote de tamanho igual a 20 *flits*, o equivalente a 5120 bits a uma taxa uniforme de 800 Mbps, ou seja, 100% da capacidade do transmissor. A Tabela 11 descreve os resultados obtidos após realizar a simulação dos tráfegos propostos. Os resultados obtidos demonstram que o tráfego utilizando modelo de aplicação CDCM aumentou em 6% a latência e reduziu a vazão em 10%.

Tabela 11: Valores de vazão e latência média dos cenários de tráfego obtidos durante a simulação.

Cenário de tráfego	Tráfego Sintético		Modelo de aplicação CDCM	
	Latência Média (ns)	Vazão Média (Mbps)	Latência Média (ns)	Vazão Média (Mbps)
00 - 10	657	1460	685	1400
01 - 11	533	1200	581	1100
10 - 01	484	660	533	600
Média	558	1106	599	1033

O autor esperava obter resultados em que o tráfego sintético resultasse em saturação do tráfego, ou seja, aumento na latência e redução na vazão. Isto porque em um modelo sintético todo tráfego é transmitido a partir do início da simulação, diferentemente do modelo de aplicação que só transmite uma tarefa a partir do momento que sua dependência foi atendida (enviou o último *flit* do pacote para a rede). O autor acredita que pelo modelo de rede proposto, e pelo mapeamento da aplicação não causar concorrência na rede, o tráfego sintético, mesmo transmitindo todos os tráfegos ao mesmo tempo, acabou não conseguindo reproduzir o comportamento esperado.

Mesmo não conseguindo reproduzir o comportamento esperado no estudo de caso proposto, a tendência ao fazer uso de tráfegos sintéticos é que todas as transmissões aconteçam de maneira simultânea na rede. Isto obviamente tende a criar congestionamentos e aumentar o tempo de transmissão e reduzir a capacidade de transmissão da rede. Já o modelo de aplicação, garante que cada uma das transmissões só aconteça quando um conjunto de dependências definidas para cada tarefa de uma aplicação for concluída.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho propôs e desenvolveu um ambiente de geração e avaliação de redes intrachip não síncronas integrado ao ambiente ATLAS. O ambiente de geração de redes foi desenvolvido para permitir a geração parametrizável de instâncias da rede HERMES-G. O ambiente de geração e avaliação de tráfego foi desenvolvido a partir dos modelos de tráfego sintéticos propostos por Tedesco [TED05] e de um dos modelos de tráfego de aplicações proposto por Marcon [MAR05] para permitir realizar a avaliação das redes.

O estado da arte carece de referências a trabalhos com a proposta de redes intrachip não síncronas em nível VHDL que de alguma forma permitam a parametrização das características da rede. Este trabalho, além de disponibilizar um projeto de rede não síncrona passível de parametrização de suas características, disponibiliza um ambiente gráfico para geração de redes intrachip não síncronas de maneira intuitiva. Ainda com relação ao estado da arte, no tema relacionado a caracterização do tráfego, foram encontrados vários indícios que a maneira como o tráfego é gerado e mapeado tem relação direta com a desempenho da rede, e ainda, que o tráfego sintético utilizado por grande parte dos trabalhos pesquisados está distante de conseguir representar o comportamento de aplicações de forma precisa. Durante a investigação do estado da arte, diversas propostas foram encontradas que promovem a evolução de modelos de tráfego sintético. Neste trabalho, o autor além de propor novas características ao modelo de tráfego sintético, propõe também gerar tráfego a partir de modelos de aplicações. Além da proposta dos modelos de tráfego, este trabalho disponibiliza ainda um ambiente de geração de tráfego para redes não síncronas, que permite gerar tráfego de maneira parametrizável.

Por fim, os trabalhos do estado da arte no tema de avaliação de tráfego para redes detalham o uso de métricas de vazão e latência de pacotes. Na grande maioria dos trabalhos, a latência é medida em ciclos, e a vazão medida através de tráfego aceito. Neste trabalho, uma vez que uma rede opera com diferentes frequências, a latência, ou tempo de transmissão do pacote deve ser avaliada em tempo absoluto e não em ciclos. Uma vez que se possa conhecer o caminho do tráfego e as frequências dos roteadores, possível mensurar o tempo gasto por um pacote com precisão. Optou-se aqui por avaliar a vazão em quantidade de informação transmitida (Mbps), e não em tráfego aceito, uma vez que em uma rede não síncrona, a taxa de injeção de um pacote pode variar dependendo a frequência de operação da rede utilizada.

7.1 CONTRIBUIÇÕES DO TRABALHO

Este trabalho explorou trabalhos relacionados aos temas de geração e avaliação de redes não síncronas. Este trabalho contribui com um ambiente de geração de redes intrachip não síncronas em nível VHDL passíveis de parametrização através de uma interface gráfica intuitiva. Em nenhum trabalho revisado é apresentado uma proposta semelhante.

Com relação à geração de tráfego, este trabalho promove a evolução do gerador de tráfego proposto por Tedesco [TED05] nas questões relacionadas ao modelo de tráfego sintético, introduzindo a distribuição exponencial decrescente e propondo algumas correções nos cálculos e nas interfaces do gerador. Além disso, introduz-se um modelo de geração de tráfego que faz uso de modelos de aplicações. A utilização deste novo modelo traz ao grupo de pesquisa do autor uma série de novas possibilidades de uso.

Com relação à avaliação do tráfego, este trabalho propôs uma evolução da ferramenta de avaliação de tráfego de Tedesco [TED05], adaptando as técnicas e as métricas utilizadas para avaliação do tráfego gerado para redes não síncronas. Durante a revisão do estado da arte, o autor não encontrou nenhum trabalho propondo uma ferramenta de avaliação de tráfego para redes não síncronas. Em todos aqueles que de alguma forma demonstraram resultados para redes não síncronas, nenhum apresentou uma proposta similar à proposta por este trabalho para avaliar o tráfego.

Por fim, o autor ainda cita como contribuição os esforços desenvolvidos e pouco documentados neste trabalho relacionados à integração e verificação das ferramentas propostas ao fluxo de desenvolvimento de redes do ambiente ATLAS. Por este ambiente ter ampla aceitação da comunidade acadêmica no escopo de geração e avaliação de redes intrachip, ele poderá ser utilizado no futuro por outros trabalhos. Na visão do autor, a maior contribuição deste trabalho é reunir outros trabalhos, utilizando os ambientes descritos em [OST05] e [TED05], e a rede intrachip não síncrona proposta por Pontes et al. em [PON08] em algo maior, criando novos tópicos de pesquisa a serem explorados pelo grupo de pesquisa do autor.

7.2 TRABALHOS FUTUROS

A partir da proposta descrita por este trabalho, é possível definir um conjunto de trabalhos futuros a serem realizados:

- ❖ Desenvolvimento de um componente de avaliação interna, responsável por coletar o tráfego entre os canais que interligam os roteadores. Este componente deve levar em consideração que cada canal irá transmitir informação em uma determinada frequência, e que as características da rede por serem parametrizáveis irão influenciar na geração da sua estrutura.
- ❖ Adaptação do ambiente de avaliação de tráfego, para levar em consideração os valores coletados pelo componente de avaliação interna conforme proposto por Tedesco [TED05]. As métricas utilizadas para avaliação do tráfego de maneira interna em redes síncronas levam em consideração o ciclo de relógio para realizar os cálculos de vazão e latência dos canais, devendo ser adaptadas para tempo absoluto, levando em consideração a frequência de operação dos roteadores.
- ❖ Evolução das técnicas de mapeamento propostas por Marcon [MAR05] e utilizadas no ambiente CAFES, para considerar a frequência de operação dos roteadores e um tempo de transmissão máximo de cada uma das mensagens das tarefas da aplicação. A modificação

implica que o ambiente CAFES reconheça a frequência dos roteadores para realizar o mapeamento. Sugere-se que o arquivo de projeto da rede seja utilizado como entrada pelo ambiente CAFES durante o mapeamento, uma vez que contém os valores das frequências dos roteadores definidas no projeto da rede.

- ❖ Evoluir o modelo de aplicações CDCM, permitindo que durante a geração de um modelo seja possível definir um número de pacotes para uma mensagem em *flits*. Além disso, permitir que os pacotes de uma mensagem sejam transmitidos na rede conforme uma distribuição temporal a partir das distribuições suportadas pelo ambiente de geração de tráfego do ambiente ATLAS, uma vez que hoje uma mensagem é transmitida a taxa máxima de injeção do transmissor.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- [AMD05] Amde, M.; Felicijan, T.; Efthymiou, A.; Edwards, D.; Lavagno, L. "Asynchronous on-chip networks". IEE Proceedings - Computers and Digital Techniques, 152(2), March 2005, pp. 273-283.
- [ARD10] Arden, W.; Brillouët, M.; Cogez, P.; Graef, M.; Huizing, B.; Mahnkopf, R. "More-than-Moore" White Paper". Capturado em: <http://www.itrs.net/Links/2010ITRS/IRC-ITRS-MtM-v2%203.pdf> em Junho 2012, Novembro, 2010.
- [BON07] Bononi, L.; Concer, N.; Grammatikakis, M. Coppola, M.; Locatelli, R. "NoC Topologies Exploration based on Mapping and Simulation Models". In: 10th Euromicro Conference on Digital System Design Architectures (DSD'07), 2007, pp. 543-546.
- [BRU09] Bruch, J.; Pizzoni, M.; Zeferino, C. "BrownPepper: a SystemC-based Simulator for Performance Evaluation of Networks-on-Chip". In: 17th Annual International Conference on Very Large Scale Integration (VLSI-Soc'09), 2009, pp. 223-226.
- [FEN07] Fen, G.; Ning, W.; Qi, W. "Simulation and Performance Evaluation for Network on Chip Design Using OPNET". In Taipei International Convention Center (TENCON'07), 2007, pp.1-4.
- [GRA06] Gratz, P.; Kim, C.; McDonald, R. Keckler, S.; Burger, D. "Implementation and Evaluation of On-Chip Network Architectures". In: 24th International Conference on Computer Design (ICCD'06), 2006, pp. 477-484.
- [HEC11] Heck, G. "Um MPSoC GALS Baseado em Redes Intrachip com Geração Local de Relógio". Seminário de andamento de Mestrado, PPGCC-FACIN-PUCRS. Janeiro 2011. 20p.
- [HON08] Hong, P.; Pham, P.; Tran, X.; Kim, C. "Analysis and Evaluation of Traffic-Performance in a Backtracked Routing Network-on-Chip". In Second International Conference on Communications and Electronics (ICCE'08), 2008, pp. 13-17.
- [INT11] INTEL INC. "Teraflops Research Chip: Project". Capturado em: <http://www.intel.com/content/www/us/en/research/intel-labs-teraflops-research-chip.html>, Junho 2012.
- [INT12] INTEL FOUNDRIES INC. "Introducing the World's First 3-D Transistor Ready for High-Volume Manufacturing". Capturado em: <http://www.intel.com/content/www/us/en/silicon-innovations/intel-22nm-technology.html>, Junho 2012.
- [ITR11a] ITRS INC. "International Technology Roadmap for Semiconductors 2011 Edition Design". Capturado em: <http://www.itrs.net/Links/2011ITRS/2011Chapters/2011Design.pdf>, Junho 2012.
- [ITR11b] ITRS INC. "International Technology Roadmap for Semiconductors 2011 Edition Lithography". Capturado em: <http://www.itrs.net/Links/2011ITRS/2011Chapters/2011Lithography.pdf>, Junho 2012.
- [ITR11c] ITRS INC. "International Technology Roadmap for Semiconductors 2011 Edition Yield Enhancement". Capturado em: <http://www.itrs.net/Links/2011ITRS/2011Chapters/2011Yield.pdf>, Junho 2012. [GLO12]

GLOBAL FOUNDRIES INC. "GLOBALFOUNDRIES Fab 8 Adds Tools To Enable 3D Chip Stacking at 20nm and Beyond". Capturado em: <http://www.globalfoundries.com/newsroom/2012/20120426.aspx>, Junho 2012.

- [JAN05] Lu, Z.; Jantsch A. "Traffic Configuration for Evaluating Networks on Chips". In: Fifth International Workshop on System-on-Chip for Real-Time Applications (IWSOC '05), 2005, pp. 535-540.
- [KRE05] Kreutz, M.; Marcon, C.; Carro, L.; Calazans, N.; Susin, A. "Energy and Latency Evaluation of NoC Topologies". In: International Symposium on Circuits and Systems (ISCAS'05), 2005, pp. 5866-5869.
- [LIU07] Liu, W.; Xu, J.; Wu, X.; Ye, Y.; Wang, X.; Zhang, W.; Nikdast, M.; Wang, Z. "A NoC Traffic Suite Based on Real Applications". In Computer Society Annual Symposium on VLSI (ISVLSI'07), 2007, pp. 66-71.
- [MAR05] Marcon, C. "Modelos para o Mapeamento de Aplicações em Infra-estruturas de Comunicação Intrachip". Tese de Doutorado, PPGCC-FACIN-PUCRS. Dezembro 2005. 192p. Disponível em http://www.inf.pucrs.br/~calazans/publications/Marcon_PhDTh.pdf
- [MOO65] Moore, G. "Cramming More Components onto Integrated Circuits". Electronics, 38(8), April 1965, pp.114-117.
- [MOR04] Moraes, F.; Calazans, N.; Mello, A.; Möller, L; Ost, L. "HERMES: an infrastructure for low area overhead packet-switching networks on chip". Integration the VLSI Journal, 38(1), October 2004, pp. 69-93.
- [MOR10] Moreno, E. Mapeamento e Adaptação de Rotas de Comunicação em Redes em Chip. Tese de Doutorado, PPGCC-FACIN-PUCRS, Porto Alegre, Brazil. Janeiro 2010.
- [NVI12] NVIDIA INC. "NVIDIA's Next Generation CUDA Compute Architecture: Kepler GK110". Capturado em: <http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf>, Junho 2012.
- [OST05] Ost, L.; Mello, A.; Palma, J.; Moraes, F.; Calazans, N.; "MAIA: a framework for networks on chip generation and verification". In: 10th Annual Asia and South Pacific Design Automation Conference (ASP-DAC'05), 2005, pp49-52.
- [PAN06] Miro Panades, I.; Greiner, A.; Sheibanyrad, A. "A Low Cost Network-on-Chip with Guaranteed Service Well Suited to the GALS Approach" . In NanoNet, Sep 2006, pp.1-5.
- [PON08] Pontes, J.; Moreira, M.; Soares, R.; Calazans, N. "Hermes-GLP: A GALS Network on Chip Router with Power Control Techniques". In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI '08), April 2008, pp. 347-352.
- [SCH10] Schemmer, R.; Calazans, N. "Proposta e Implementação de um Novo Modelo de Geração de Tráfego Para a Plataforma Atlas". Introdução a Pesquisa II, PPGCC-FACIN-PUCRS. Dezembro 2010. 60p.
- [SOT06] Soteriou, V.; Wang, H.; Peh, L.-S. "A Statistical Traffic Model for On-Chip Interconnection Networks". In: 14th IEEE International Symposium on Modeling, Analysis, and Simulation (MASCOTS '06), September 2006, pp. 104-116.
- [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.

- [TED05a] Tedesco, L.; Mello, A.; Garibotti, D.; Calazans, N.; Moraes, F. "Traffic Generation and Performance Evaluation for Mesh-based NoCs". In: 19th Annual Symposium on Integrated Circuits and Systems design (SBCCI'05), 2005, pp. 184-189.
- [TED06] Tedesco, L.; Mello, A.; Giacomet, L.; Calazans, N.; Moraes, F. "Application driven traffic modeling for NoCs". In: 19th Annual Symposium on Integrated Circuits and Systems design (SBCCI'06), 2006, pp. 62-67.