

**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**ALAN CADORE PINHEIRO**

**NOIA: UMA FERRAMENTA PARA ANÁLISE DE NETWORKS-ON-CHIP**  
**IRREGULARES**

**FORTALEZA**  
**2016**



ALAN CADORE PINHEIRO

NOIA: UMA FERRAMENTA PARA ANÁLISE DE NETWORKS-ON-CHIP  
IRREGULARES

Monografia apresentada ao Curso de Engenharia de Computação do Departamento de Engenharia de Teleinformática da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Jarbas Aryel Nunes da  
Silveira

FORTALEZA  
2016

---

Alan Cadore Pinheiro      NoIA: uma ferramenta para análise de networks-on-chip irregulares/ Alan Cadore Pinheiro. – Fortaleza, 2016-      41 p. : il.(alguma color.); 30 cm.

Orientador: Prof. Dr. Jarbas Aryel Nunes da Silveira

Monografia apresentada ao Curso de Engenharia de Computação do Departamento de Engenharia de Teleinformática da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação. – Universidade Federal do Ceará

Centro de Tecnologia

Departamento de Engenharia de Teleinformática

Curso de Engenharia de Computação, 2016.

1. Palavra-chave1. 2. Palavra-chave2. I. Jarbas Aryel Nunes da Silveira. II. Universidade Federal do Ceará. III. Centro de Tecnologia. IV. NoIA: Uma Ferramenta para Análise de Networks-on-Chip Irregulares

CDU

---

ALAN CADORE PINHEIRO

NOIA: UMA FERRAMENTA PARA ANÁLISE DE NETWORKS-ON-CHIP  
IRREGULARES

Monografia apresentada ao Curso de Engenharia de Computação do Departamento de Engenharia de Teleinformática da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Trabalho aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_.

BANCA EXAMINADORA

---

Prof. Dr. Jarbas Aryel Nunes da Silveira  
(Orientador)  
Universidade Federal do Ceará (UFC)

---

<Nome do Primeiro examinador da banca>  
<IES do primeiro membro>

---

<Nome do Segundo examinador da banca>  
<IES do segundo membro>



## **RESUMO**

**Palavras-chave:** Palavra1. Palavra Chave2. Palavra3. Palavra Chave Quatro.





## **ABSTRACT**

**Keywords:** Keyword1. Keyword2. Word3. Keyword four.



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>11</b>
<b>1.1</b>	<b>Objetivos . . . . .</b>	<b>12</b>
<b>1.1.1</b>	<b>Objetivo Geral . . . . .</b>	<b>12</b>
<b>1.1.2</b>	<b>Objetivos Específicos . . . . .</b>	<b>12</b>
<b>1.2</b>	<b>Organização deste Trabalho . . . . .</b>	<b>13</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>15</b>
<b>2.1</b>	<b>Sistemas Intrachip . . . . .</b>	<b>15</b>
<b>2.2</b>	<b>Redes Intrachip . . . . .</b>	<b>16</b>
<b>2.2.1</b>	<b>Conceitos . . . . .</b>	<b>16</b>
<b>2.2.2</b>	<b>Topologia . . . . .</b>	<b>17</b>
<b>2.2.2.1</b>	<b>Malha Regular . . . . .</b>	<b>17</b>
<b>2.2.2.2</b>	<b>Torus . . . . .</b>	<b>18</b>
<b>2.2.2.3</b>	<b>Árvore Binária . . . . .</b>	<b>18</b>
<b>2.2.3</b>	<b>Redes Irregulares . . . . .</b>	<b>18</b>
<b>2.2.4</b>	<b>Memorização . . . . .</b>	<b>19</b>
<b>2.2.5</b>	<b>Controle de fluxo . . . . .</b>	<b>19</b>
<b>2.2.5.1</b>	<b>Handshake . . . . .</b>	<b>20</b>
<b>2.2.5.2</b>	<b>Baseado em Créditos . . . . .</b>	<b>20</b>
<b>2.2.6</b>	<b>On-Off . . . . .</b>	<b>20</b>
<b>2.2.7</b>	<b>Chaveamento . . . . .</b>	<b>21</b>
<b>2.2.7.1</b>	<b>Chaveamento por Circuito . . . . .</b>	<b>21</b>
<b>2.2.7.2</b>	<b>Chaveamento por Pacote . . . . .</b>	<b>21</b>
<b>2.2.7.3</b>	<b>Store and Forward . . . . .</b>	<b>22</b>
<b>2.2.7.4</b>	<b>Virtual Cut Through . . . . .</b>	<b>22</b>
<b>2.2.7.5</b>	<b>Wormhole . . . . .</b>	<b>22</b>
<b>2.2.8</b>	<b>Arbitragem . . . . .</b>	<b>22</b>
<b>2.2.8.1</b>	<b>Round Robin . . . . .</b>	<b>22</b>
<b>2.2.8.2</b>	<b>First Come First Serve . . . . .</b>	<b>23</b>
<b>2.2.9</b>	<b>Roteamento . . . . .</b>	<b>23</b>
<b>2.2.9.1</b>	<b>Algoritmos Determinísticos . . . . .</b>	<b>23</b>
<b>2.2.9.2</b>	<b>Algoritmos Adaptativos . . . . .</b>	<b>23</b>
<b>2.2.9.3</b>	<b>Algoritmos para Redes Irregulares . . . . .</b>	<b>24</b>
<b>2.2.9.3.1</b>	<b>UP*/DOWN* . . . . .</b>	<b>24</b>
<b>2.2.9.3.2</b>	<b>Segment-based Routing . . . . .</b>	<b>24</b>
<b>2.2.9.4</b>	<b>Implementação dos Algoritmos . . . . .</b>	<b>26</b>

2.2.9.4.1	Region-based Routing . . . . .	26
3	MATERIAIS E MÉTODOS . . . . .	29
3.1	Geração de Topologias . . . . .	29
3.2	Algoritmos Implementados . . . . .	31
3.2.1	SBR . . . . .	31
3.2.2	Computação de Caminhos . . . . .	32
3.2.3	RBR . . . . .	34
3.2.4	Métricas . . . . .	36
3.2.5	Interface Gráfica . . . . .	36
4	DISCUSSÃO DOS RESULTADOS . . . . .	39
5	CONCLUSÃO . . . . .	41

## 1 INTRODUÇÃO

Nas últimas décadas, a tecnologia acompanhou a Lei de Moore, a qual prevê que a quantidade de transistores em um circuito integrado irá duplicar a cada ano. Elementos com funcionalidades diferenciadas e com maior poder de processamento funcionaram como uma alavanca para o desenvolvimento de sistemas mais eficientes. Nessa linha de avanço, temos os sistemas embarcados que são desenvolvidos para um propósito específico. Dimensionando recursos e restringindo suas funcionalidades.

Os sistemas embarcados atuais vêm utilizando sistemas intrachip, em inglês System-on-Chip (SoC), de forma diversificada e em larga escala. Um SoC é composto por diferentes componentes agrupados de forma a termos sistemas complexos e completos em um único chip. Tais componentes necessitam ser interconectados por uma estrutura de comunicação robusta e eficiente. A modelagem e desenvolvimento desta estrutura implica não somente na performance de todo o chip, mas também no consumo de potência, nos requisitos de tolerância a falhas, na dimensão física, no custo e tempo de projeto.

Com o avançar da tecnologia, a área ocupada por um transistor está cada vez menor, permitindo que a quantidade de elementos dentro de um único chip cresça consideravelmente. Como consequência disso, o nível de integração dos circuitos se intensifica de forma que a comunicação entre esses elementos se torna ineficiente. Pois as soluções tradicionais, como conexões por barramento e ponto-a-ponto, acabam se comportando como um gargalo, em termos de comunicação.

Diante dessa limitação, surge o conceito de redes intrachip, em inglês Network-on-Chip (NoC), que seguem os conceitos de redes de chaveamento de computadores. Uma NoC é constituída de roteadores interligados por meio de links e cada roteador está interligado a um elemento de processamento da SoC. Esses roteadores se conectam segundo uma topologia, por exemplo, malha regular que se assemelha a uma grelha ou torus, onde é uma malha regular em que os roteadores das extremidades se conectam aos seus opostos. Também são implementados algoritmos de roteamento e técnicas de chaveamento. Assim, a NoC atua gerenciando toda a comunicação entre os elementos da SoC.

As NoCs irregulares são assim chamadas por apresentarem falhas na sua topologia, sendo classificadas como uma derivação das NoCs regulares (totalmente conectadas). Essas falhas acontecem em links ou roteadores. Essa irregularidade na topologia limita o conjunto de algoritmos de roteamento aplicáveis. Gerando a necessidade da implementação de algoritmos adaptativos. Estes algoritmos precisam ser robustos e escaláveis para se adequar à topologia aplicada.

Como forma de avaliar a performance desses algoritmos na rede são necessárias métricas analíticas e de simulação. Por meio das métricas analíticas é medida a eficiência do roteamento utilizando, não sendo necessário realizar simulações, apenas é executado o

algoritmo de roteamento e definido os caminhos entre os roteadores.

Por outro lado, as métricas extraídas de simulações revelam o desempenho da rede. Geralmente demandam um tempo maior para serem geradas, pois, além da execução do algoritmo de roteamento, também é necessário simular a NoC, injetar tráfego e finalmente extrair os dados.

Este trabalho foca na ferramenta que gera métricas analíticas, utilizando o algoritmo SBR (MEIJA et al, 2006), em inglês Segment-based Routing, em conjunto com o mecanismo de roteamento RBR (MEIJA et al, 2009), em inglês Region-based Routing, que implementa o algoritmo SBR.

Também apresenta um recurso visual para análise do comportamento resultante do processamento do algoritmo na topologia dada, onde, além de visualizar a forma da topologia, é possível observar a formação de segmentos, o posicionamento de restrições, a formação de regiões, os pesos dos links e as métricas analíticas. Por fim, é gerada uma tabela de roteamento que é usada para fins de simulação.

## 1.1 Objetivos

Os objetivos gerais e específicos deste trabalho são apresentados a seguir.

### 1.1.1 *Objetivo Geral*

O objetivo principal deste trabalho é a construção da ferramenta NoIA e análise das métricas analíticas geradas por esta ferramenta gráfica para diferentes topologias de entrada. Visando uma previsão do desempenho das NoCs referentes a essas topologias, que irão utilizar as tabelas de roteamento geradas pela ferramenta.

### 1.1.2 *Objetivos Específicos*

- Implementar o algoritmo de roteamento SBR;
- Implementar o mecanismo de roteamento RBR;
- Visualizar as topologias de NoCs e suas falhas;
- Visualizar o resultado da segmentação;
- Visualizar as regiões formadas;
- Gerar métricas analíticas;
- Analisar resultados.

## **1.2 Organização deste Trabalho**

Este trabalho está organizado em 5 capítulos. O capítulo 2 trata da fundamentação teórica sobre redes intrachip. O capítulo 3 trata da metodologia usada para o desenvolvimento da ferramenta e materiais para avaliação dos seus resultados. No capítulo 4 são apresentados e discutidos os resultados deste trabalho. Por fim, no capítulo 5 é apresentada a conclusão deste trabalho.





## 2 FUNDAMENTAÇÃO TEÓRICA

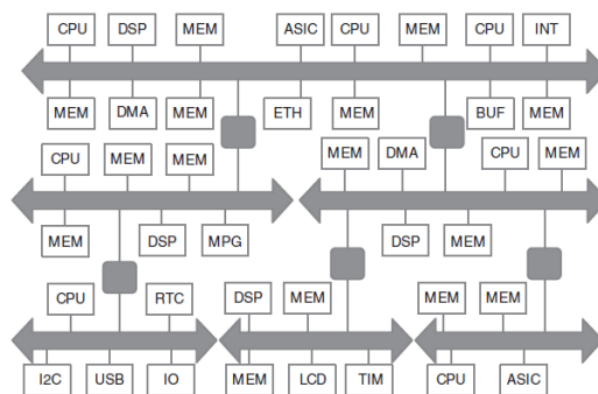
### 2.1 Sistemas Intrachip

Com o avanço tecnológico dos circuitos integrados foi elevado o nível de integração permitindo, assim, o desenvolvimento de um sistema que embarca mais de uma unidade de hardware, cada uma projetada de forma independente. Esses sistemas são chamados de sistemas intrachip (SoC) e têm alta integração entre os seus elementos heterogêneos (PASRICHA; DUTT, 2008).

Essas unidades de hardware são chamadas de propriedade intelectual, do inglês Intellectual Property (IP), e precisam de uma estrutura de comunicação que permita a comunicação entre esses IPs. Essa estrutura deve garantir o correto fluxo de dados da origem ao destino de cada envio, bem como oferecer garantias sobre o atraso e a vazão (PASRICHA; DUTT, 2008).

Uma das estruturas de comunicação mais usadas é a barramento. Nela os elementos são conectados por meio de uma conexão compartilhada e somente um elemento pode fazer uma transmissão por vez. A Figura 1 ilustra um exemplo de barramento.

Figura 1 – Barramento



Fonte: (PASRICHA; DUTT, 2008).

Essa concorrência de recurso limita o paralelismo e reduz o desempenho (PASRICHA; DUTT, 2008). No caso de aplicações que possuam um número considerável de processadores, o barramento se torna proibitivo. Inclusive, sendo o barramento tradicional, se torna inadequada às características de comunicação de um SoC (NICOPOULOS; NARAYANAN; DAS, 2010). Em aplicações que utilizam altas frequências, essa estrutura de comunicação se torna mais complexa de se operar devido às conexões dos módulos que causam descontinuidades de impedância (DALLY; TOWELS, 2003).

Visando contornar essas barreiras emergentes, um novo conceito foi proposto: uma arquitetura baseada em redes de computadores para gerenciar a comunicação entre os

elementos de processamento. Esse tipo rede de interconexão chaveada é conhecida como Network-on-Chip (NoC). Uma NoC contribui com um alto grau de paralelismo, alta escalabilidade e reusabilidade da arquitetura de comunicação (CARDOZO, 2005).

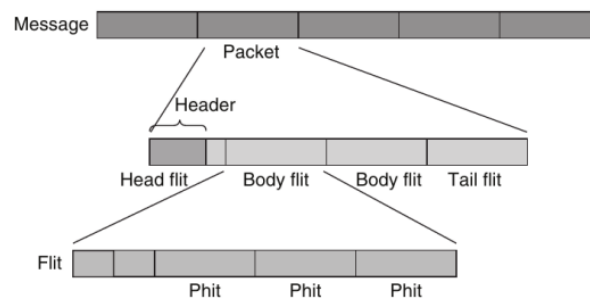
## 2.2 Redes Intrachip

### 2.2.1 Conceitos

A NoC é constituída de roteadores e links. Geralmente o par IP-Roteador é referenciado como nó ou nodo (BENINI; MICHELLI, 2006). A NoC atua como uma rede de roteadores interligados que se conectam aos elementos de uma SoC. Utiliza-se conceitos de redes de computadores e sistemas distribuídos. O projetista precisa atentar em reduzir a área de silício ocupada e o consumo de energia. Esses pontos não são preocupantes em redes de computadores tradicionais (TEDESCO, 2005).

Na transmissão de informações na NoC, são considerados quatro unidades de comunicação: a mensagem, que é o conjunto de dados enviado entre IPs; o pacote, o qual é uma parte da mensagem sendo enviada e que é trabalhada pelo algoritmo de roteamento, também é adequada às restrições da rede; o flit, do inglês flow control unit, que é a unidade de divisão dos pacotes e a unidade em que o controle de fluxo age; o phit, do inglês physical unit, é a unidade física em que atravessa o link num ciclo de relógio. A Figura 2 mostra essas unidades relacionadas entre si.

Figura 2 – Relação entre unidades de comunicação



Fonte: (PASRICHA; DUTT, 2008).

As principais vantagens de uma NoC são: escalabilidade, paralelismo, reusabilidade, decisões de roteamento distribuídos e baixo consumo de potência. Entretanto, existem algumas desvantagens como: área de silício ocupada, latência dos pacotes e a complexidade na fase de projeto (TEDESCO, 2005).

Uma NoC é definida em dois aspectos: a topologia e os mecanismos de comunicação. Esses mecanismos são divididos em memorização, controle de fluxo, chaveamento, arbitragem e roteamento.

### 2.2.2 Topologia

A topologia diz respeito à interligação dos nodos através dos links. A escolha da topologia é muito importante, pois, dependendo da topologia escolhida, temos uma grande ou baixa diversidade de caminhos entre nodos de origem e nodos de destino e temos uma grande ou pequena distância de roteamento. Com base na topologia são escolhidos os algoritmo de roteamento e controle de fluxo.

Existem as topologias diretas e indiretas. No caso das diretas, todos os roteadores estão ligados a um elemento de processamento. Logo, qualquer roteador pode ser uma origem ou um destino numa transmissão da dados. Esses roteadores se conectam a um número limitado de roteadores vizinhos.

Em se falando de topologias indiretas, os roteadores não estão obrigatoriamente ligados a um IP. Portanto, existem os roteadores intermediários, que não se conectam diretamente a um IP, e os terminais que têm um de seus links conectados a um IP.

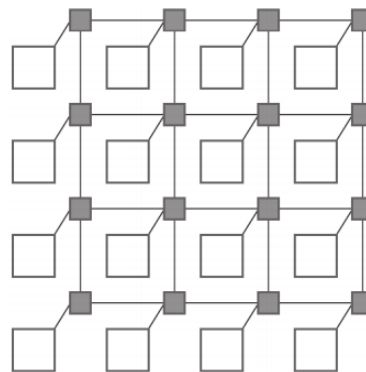
A seguir são exemplificados duas topologias diretas (malha regular e torus clássica) e uma indireta (árvore binária).

#### 2.2.2.1 Malha Regular

É a topologia mais comumente utilizada. Os roteadores que não estão nas extremidades possuem cinco conexões, uma para o elemento de processamento e quatro para conectar-se aos roteadores adjacentes. Os nodos das laterais possuem quatro conexões (sendo uma para o IP) e os dos cantos possuem três conexões, com uma conexão para o elemento de processamento.

Essa estrutura permite a incorporação de vários elementos de processamento em uma estrutura regular (REDDY et al., 2014). A Figura 3 exemplifica uma topologia de malha regular de dimensão 4.

Figura 3 – Topologia malha regular

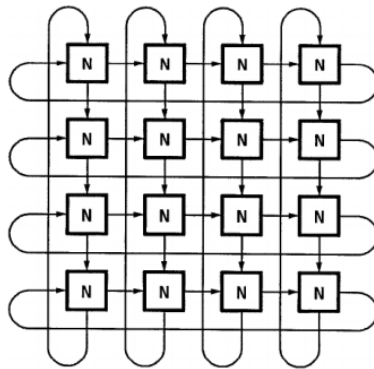


Fonte: (PASRICHA; DUTT, 2008).

### 2.2.2.2 Torus

Essa estrutura se assemelha à malha regular, diferenciando no fato de que os roteadores das extremidades se conectam com os roteadores da extremidade oposta (REDDY et al., 2014). Com isso, todos os roteadores da topologia têm cinco links, sendo um para o elemento de processamento e os outros quatro para roteadores vizinho (e de extremidade oposta). A Figura 4 ilustra uma topologia torus clássica de dimensão 4.

Figura 4 – Topologia torus regular

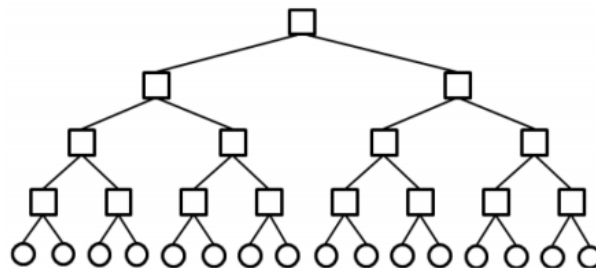


Fonte: (DALLY; SEITZ, 1986).

### 2.2.2.3 Árvore Binária

A árvore binária segue a forma de uma árvore em que os elementos de processamento estão nas folhas, portanto, no nível mais baixo. Cada elemento da rede é representado pelo par (nível, posição). Onde nível indica a posição vertical na árvore e posição representa a localização da esquerda para a direita. A figura 5 exemplifica uma árvore binária.

Figura 5 – Topologia árvore binária



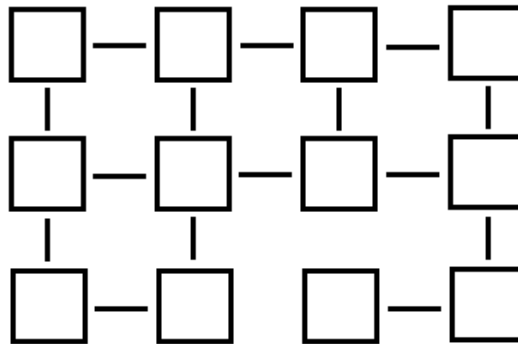
Fonte: (REDDY et al., 2014).

### 2.2.3 Redes Irregulares

Redes irregulares podem derivadas de topologias regulares. Essa irregularidade se dá devido a falhas decorrentes de links e/ou roteadores. Tais falhas são provenientes de

defeitos de fabricação ou falhas ocasionadas em tempo real. Neste trabalho trataremos de redes irregulares provenientes de falha em links. A figura 6 exemplifica uma topologia irregular derivada de uma malha regular de dimensão 4x3.

Figura 6 – Rede irregular



Fonte: Elaborada pelo autor.

#### 2.2.4 Memorização

A memorização é o mecanismo que trata do armazenamento do pacote no caminho de envio. Portanto, influencia na eficiência do compartilhamento da largura de banda dos links.

Esse armazenamento é feito por meio de buffers. Os buffers armazenam pacotes ou flits que não podem avançar de imediato e estão localizados na porta de entrada ou de saída do roteador (JERGER; PEH, 2009).

São definidas três possíveis arquiteturas: enfileiramento de entrada, enfileiramento de saída e enfileiramento combinado, o qual é de entrada e saída.

#### 2.2.5 Controle de fluxo

O controle de fluxo é o mecanismo que controla a alocação de recursos na rede. Ou seja, indica quando os recursos estão livres para serem usados, bem como determina a forma como são compartilhados entre as mensagens em tráfego.

Um controle de fluxo que tende ao ideal minimiza a latência das mensagens em baixa carga e engrandece a vazão por meio de um eficiente compartilhamento de recurso. Logo, é determinante no consumo de potência e energia na rede (JERGER; PEH, 2009).

No ato de injeção de uma mensagem na rede, esta é segmentada em pacotes que são constituídos por flits. Esses flits são organizados em flits de cabeçalho, intermediários e terminais. Os flits de cabeçalho indicam o destino da mensagem, a sua origem e o tamanho do pacote. Flits intermediários são os dados propriamente ditos do pacote. Por fim, os flits terminais são indicadores do fim do pacote.

Os flits são comumente definidos como a menor divisão de um pacote, sendo constituídos por somente um phit (JERGER; PEH, 2009).

### 2.2.5.1 Handshake

Nessa técnica ocorre uma verificação e confirmação para alocação de flits em buffer. Quando o roteador de origem precisa enviar um flit para o roteador de destino, é enviado um sinal de requisição através da linha de requisição. Caso o roteador de destino tenha recurso disponível, é retornado um sinal de confirmação, através da linha de confirmação, indicando a permissão de envio e, então, ocorre o envio do flit. Esse procedimento se repete para cada envio de flit.

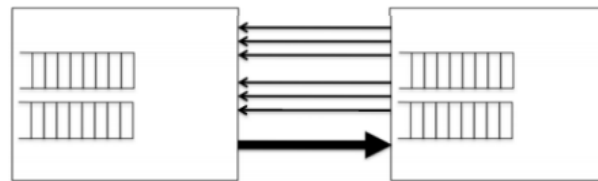
### 2.2.5.2 Baseado em Créditos

Essa técnica trabalha com uma linha de indicação de dado para transmissão e um conjunto de linhas que indicam o valor do crédito. O crédito é uma grandeza que indica a o espaço livre existente no buffer do roteador receptor (JERGER; PEH, 2009).

Enquanto houver sinalização de dado a ser enviado e o valor de crédito for maior que zero, flits são enviados ao roteador receptor e o valor de crédito é decrementado. Da mesma forma que, se o roteador receptor enviar mensagem a um próximo roteador, o valor de crédito é incrementado.

A figura 7 exemplifica um controle de fluxo baseado em créditos.

Figura 7 – Controle baseado em créditos



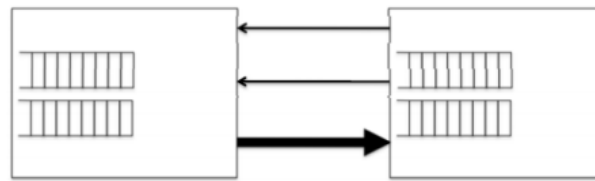
Fonte: (JERGER; PEH, 2009).

### 2.2.6 On-Off

Nessa técnica há uma semelhança com a técnica baseada em créditos, diferenciando no modo como se indica uma permissão de envio de dado. Essa indicação se dá por meio de um sinal que indica um estado OFF quando não se é permitido o envio de dados e um estado ON o qual indica a permissão do envio de dados.

Esses estados estão relacionados aos limites superior e inferior, respectivamente, da capacidade do buffer. Esses limites são determinados com base no atraso da recepção do sinal OFF, o que permite que um flit enviado durante o atraso tenha espaço no buffer do receptor. A figura 8 ilustra o controle de fluxo on-off.

Figura 8 – Controle on-off



Fonte: (JERGER; PEH, 2009).

### 2.2.7 Chaveamento

O chaveamento é o mecanismo que define como uma mensagem será transmitida pela rede passando pelos roteadores.

#### 2.2.7.1 Chaveamento por Circuito

Inicialmente o cabeçalho do pacote percorre a rede reservando o caminho entre o roteador de origem e o roteador de destino. Após o caminho ser reservado por completo, o restante do pacote é transmitido por meio desse caminho. Os recursos não mais usados pelo pacote são liberados na medida que o final do pacote avança na rede. Nesse processo de construção do caminho pode ocorrer deadlock num caso de concorrência cíclica de recursos. O algoritmo de roteamento é quem deve evitar isso.

Com o uso desse caminho reservado, não há necessidade do uso de buffers nos nodos intermediários e, portanto, só ocorre latência de transmissão nesses nodos intermediários. Diminuindo, assim, atrasos e a área ocupada pela rede.

Apesar de existir um atraso inicial de configuração, no caso de mensagens grandes, esse custo é considerado reduzido (JERGER; PEH, 2009). Um detalhe a ser considerado é que, quando um caminho está reservado, seus recursos são bloqueados para novos caminhos que iniciam uma concorrência desses recursos. Se torna recomendada em redes de fluxo constante ou com transações infrequentes e volumosas (NURMI et al., 2004).

#### 2.2.7.2 Chaveamento por Pacote

Os pacotes avançam usando o mínimo necessário de recurso e liberam o que não é mais usado. Não há caminho pré definido ou fixo, logo, a rede tem mais recursos livres para o tráfego de pacotes, em contraste com o chaveamento por circuito (CARDOZO, 2005).

Os roteadores têm que ter suporte a memorização. Apesar de ocorrer um aumento no desempenho, já não existem mais previsões de latência, comparado com o chaveamento por circuito.

### 2.2.7.3 *Store and Forward*

O roteador tem um buffer que comporta o pacote por inteiro (JARGER; PEH, 2009). O pacote só é repassado para o próximo roteador se ele estiver totalmente armazenado no buffer atual e se o buffer do próximo roteador já puder comportá-lo (DALLY; TOWLES, 2003).

O buffer precisa ter grande armazenamento, pois, ele deve ser dimensionado com base no maior pacote da rede, no caso de um tráfego de pacotes de tamanhos diferentes. Com isso pode haver desperdícios e a pode ser mais um parâmetro para o aumento da latência.

### 2.2.7.4 *Virtual Cut Through*

Nessa técnica o pacote avança assim que o próximo buffer puder comportar todo o conteúdo do buffer atual. Inclusive, o pacote não precisa estar inteiramente no buffer atual (KERMANI; KLEINROCK, 1979).

Com relação à técnica store and forward, ocorre uma redução de latência mas os buffers tem requisitos semelhantes.

### 2.2.7.5 *Wormhole*

Essa técnica permite que os flits do pacote avancem assim que houver algum espaço livre no buffer do roteador seguinte, não sendo preciso alocar totalmente o conteúdo do roteador atual ou que o pacote esteja integralmente nesse buffer atual para prosseguir. Essa característica permite buffers menores, comparados às técnicas anteriores. Os flits avançam de forma sequencial, resultando na possibilidade do pacote ficar distribuído pela rede (DALLY; SEITZ, 1986).

No caso de recurso ocupado, o flit fica no aguardo da liberação desse recurso para poder prosseguir na transmissão. Isso implica no canal, reservado por esse flit, permanecer ocupado (JARGER; PEH, 2009). Isso faz com que a técnica também seja passível de deadlocks.

Wormhole é uma das técnicas mais utilizadas em NoCs, fornecendo um baixo consumo de potência e menor área (AGARWAL; ISKANDER; SHANKAR, 2009).

## 2.2.8 *Arbitragem*

É o mecanismo que resolve a concorrência de recurso, no caso de flits de portas de entrada diferentes solicitarem uma mesma porta de saída. Nas subseções seguintes são apresentados os mecanismos de arbitragem Round Robin e First Come First Serve.

### 2.2.8.1 *Round Robin*

Esse mecanismo consiste de um árbitro circular, o qual dispõe a prioridade para todas as portas de entrada de forma igualitária, sendo considerado fortemente justo (DALLY; TOWLES, 2003). Em uma sequência de portas, que seguem uma ordem de prioridade, a



primeira porta terá prioridade sobre as outras e quando atendida passará a ter a menor prioridade diante das demais. Então, a segunda porta, agora, passa a ter a maior prioridade e assim por diante (JERGER; PEH, 2009).

#### 2.2.8.2 *First Come First Serve*

Nesse mecanismo ocorre um enfileiramento de prioridades segundo a chegada das requisições de recurso. Ou seja, comparando com estrutura de dados, se assemelha a uma fila, na qual o primeiro que chega é o primeiro a ser atendido (JERGER; PEH, 2009).

### 2.2.9 *Roteamento*

O algoritmo de roteamento é responsável por determinar por onde a mensagem deve continuar sua transmissão pela rede até atingir seu nodo de destino. Um algoritmo de roteamento ideal deve conseguir evitar deadlocks em qualquer situação. Um deadlock é a situação em que ocorre dependência cíclica nos recursos da rede. Por exemplo, um pacote que tem links reservados e está em espera da liberação de um novo link, sendo que este novo link está ocupado por outro pacote que precisa de um desses recursos ocupados pelo primeiro pacote.

#### 2.2.9.1 *Algoritmos Determinísticos*

Algoritmos determinísticos são aqueles em que já se tem conhecimento de qualquer caminho origem-destino. Qualquer que seja o par origem-destino, sempre resultará em somente um caminho possível para esse par.

Um algoritmo determinístico muito usado é o DOR (Dimension Order) XY. Nesse algoritmo é verificada a posição do nodo destino e comparada com a posição do nodo atual toda vez que um pacote avança na rede.

Considerando que cada roteador está posicionado segundo um par coordenado (X,Y), o avanço na rede se dá primeiro na direção do eixo X e depois na direção do eixo Y. Nesse procedimento ocorre a comparação da posição X do par coordenado do nodo destino com a posição X do nodo atual. No momento em que esses valores se igualam, o pacote pode trafegar no sentido de Y até igualar a posição Y do nodo atual com a posição Y do nodo destino.

Algoritmos DOR não são indicados para uso em redes irregulares, pois, não são adaptativos. Ou seja, não conseguiriam adaptar-se a um caminho que tivesse uma falha no seu percurso. Portanto, para esses casos, são utilizados algoritmos adaptativos.

#### 2.2.9.2 *Algoritmos Adaptativos*

Algoritmos que permitem a adaptabilidade devido a falhas ou tráfego na rede são denominados algoritmos adaptativos. Esses algoritmos também podem ser classificados desta

forma devido a características como: número de caminhos, progressividade e minimalidade (ZEFERINO, 2003).

No que diz respeito ao número de caminhos, ele é considerado completo quando se pode utilizar todos os caminhos disponíveis na rede ou parcial quando apenas parte do total de caminhos pode ser usada.

Quanto à progressividade, o algoritmo pode ser progressivo, quando o pacote avança na rede reservando os links a cada passo do roteamento, ou regressivo, quando existe a possibilidade do pacote retornar pela rede desalocando os recursos previamente alocados.

Com relação à minimalidade, o algoritmo pode ser mínimo ao usar somente os caminhos mínimos que são os menores caminhos existentes entre uma origem e um destino. Como também pode ser não mínimo, onde se é usado qualquer caminho existente entre uma origem e um destino.

### 2.2.9.3 Algoritmos para Redes Irregulares

Em termos de topologias irregulares, foram surgindo aplicações que necessitavam de um certo grau de tolerância a falhas independentemente da distribuição de falhas (FLICH et al., 2012). A seguir são apresentados alguns algoritmos que suprem essa necessidade.

#### 2.2.9.3.1 UP\*/DOWN\*

Esse algoritmo monta uma árvore de dispersão usando um search breadth-first (BFS) a partir do nodo raiz. Então, um grafo adicional é construído com indicações de direções up ou down para cada link baseando-se na estrutura de camadas da árvore (FLICH et al, 2012).

As direções indicadas são baseadas na nas direções dos links. Os que vão em direção à raiz da árvore são indicados como up e os que vão no sentido contrário são indicados como down. Os caminhos só podem ser escolhidos se passarem por uma sequência de links up seguidos de uma sequência de links down.

O deadlock é evitado pela proibição de uma mensagem seguirem por um link down e em seguida seguirem por um link up (FLICH et al., 2012).

#### 2.2.9.3.2 Segment-based Routing

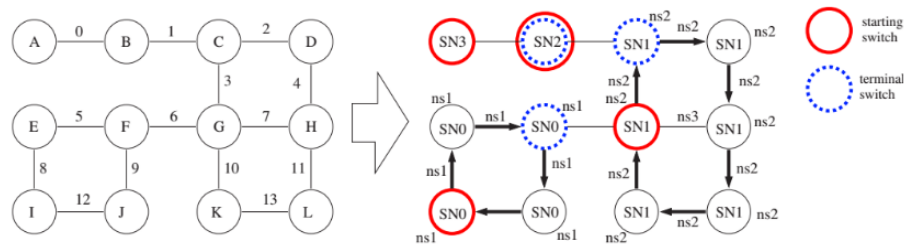
No algoritmo de criação de algoritmos livres de deadlock baseado em segmentos, do inglês Segment-based Routing (SBR), a rede é dividida em subredes e, então em segmentos. Um segmento é composto por, pelo menos, um roteador e dois links, com exceção do segmento unitário.

O algoritmo inicia a segmentação a partir de um segmento inicial, chamado de segmento start, que é circular. Em seguida todos os outros segmentos iniciam em um roteador já pertencente a um segmento e terminam em outro roteador também pertencente a um segmento. Existem três tipos de segmentos: start, regular e unitário. O segmento unitário acontece quando

um link está localizado entre dois roteadores já pertencentes a algum segmento (MEIJA et al., 2006). Cada roteador e link pertence somente a um segmento na rede.

No SBR os roteadores podem ser classificados como starting ou terminal. Um roteador é dito starting quando é o primeiro roteador de uma subrede que inicia um segmento. O roteador terminal é aquele em que não é mais formado nenhum segmento a partir de, pelo menos, um dos seus links (MEIJA et al., 2006). A Figura 9 exemplifica os roteadores starting e terminal.

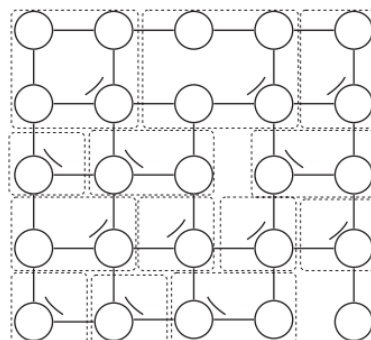
Figura 9 – Criação de segmentos e definição dos roteadores starting e terminal



Fonte: (MEIJA et al., 2006).

Visto que os segmentos são independentes, para cada segmento criado é atribuída uma única restrição bidirecional em qualquer roteador do segmento. Para o segmento start existe a exceção de que a restrição não pode ser inserida no primeiro roteador do segmento. No caso do segmento unitário, o link pertencente a ele se torna inválido, pois, não é permitido o tráfego por meio deste link. Portanto, são atribuídas restrições nos roteadores ligados a esse link em todas as direções que passam por esse link. A figura 9 exemplifica uma rede 5x5 segmentada com suas restrições postas.

Figura 10 – Topologia irregular 5x5 com segmentos e restrições



Fonte: (MEIJA et al., 2006).

Essas restrições garantem que não haverá caminhos cíclicos na rede. Portanto, deixando o algoritmo livre de deadlocks. Visto que não é um algoritmo simples de se implementar, é interessante implementá-lo por meio de tabelas de roteamento.

#### 2.2.9.4 Implementação dos Algoritmos

Esses algoritmos podem ser implementados basicamente por meio de tabelas de roteamento ou por lógica de roteamento. Na implementação por lógica de roteamento são usados circuitos que contêm o algoritmo de decisão de roteamento e geralmente são circuitos simples e rápidos (DALLY; TOWLES, 2003). Comumente somente algoritmos simples utilizam essa forma de implementação.

No caso das tabelas de roteamento, a tabela armazena todos os caminhos de roteamento e no momento de rotear um pacote a tabela é indexada pelo destino deste pacote (DALLY; TOWLES, 2003). Essas tabelas podem ser distribuídas de forma que cada roteador tem a sua tabela ou constarem somente na fonte, onde na origem do envio, todo o caminho é definido e armazenado no cabeçalho do pacote (BOLTIN et al., 2007b).

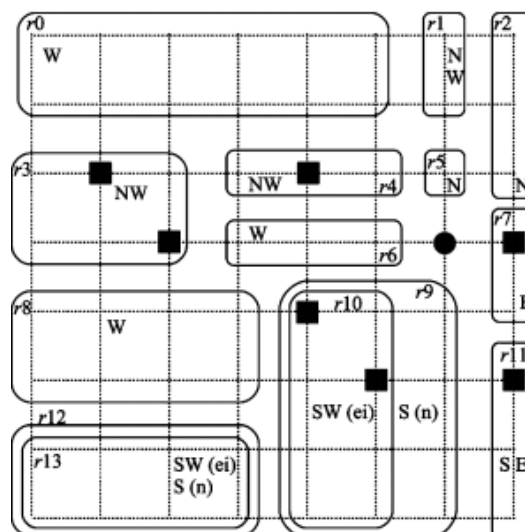
Quanto maior o tamanho da rede, maior será a tabela de roteamento. Portanto, o consumo e o tempo de consulta na tabela aumentam, também, com o aumento do tamanho da NoC.

##### 2.2.9.4.1 Region-based Routing

O roteamento baseado em regiões, do inglês Region-based Routing (RBR), é um mecanismo de implementação de algoritmos que resulta em tabelas de roteamento. Tem como objetivo minimizar a tabela de roteamento através da redução de redundância. Os destinos são agrupados nas chamadas regiões de acordo com as opções de roteamento (FLICH et al., 2007).

A Figura 11 ilustra o mecanismo aplicado, onde o roteador sinalizado (círculo preto) tem algumas de suas regiões exibidas e por quais portas é possível acessar essas regiões.

Figura 11 – Exemplo de regiões de um roteador específico



Fonte: (MEIJA et al., 2009).

A possibilidade de uma região ser acessada por mais de uma porta de saída do roteador garante a adaptatividade do algoritmo implementado. Por exemplo, na figura 10, a região r4 pode ser acessada pela porta W ou S. Com isso, temos mais opções de caminhos que possibilitam contornar possíveis falhas ou caminhos congestionados. Isso torna a rede mais confiável e pode diminuir atrasos.

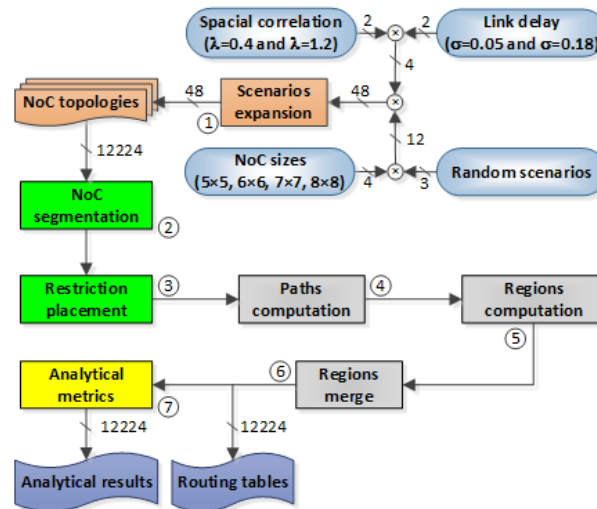
Visto que o algoritmo é livre de deadlocks, o acesso às regiões não implicará em rotas cíclicas. Pois o RBR vai obedecer as regras impostas pelo SBR. As regiões são definidas por dois roteadores: o superior direito e o inferior esquerdo. Assim resultando numa região retangular. Também há a possibilidade de uma região ser composta somente por um único roteador.



### 3 MATERIAIS E MÉTODOS

Este trabalho se baseia no processo de obtenção das métricas analíticas por meio do SBR e RBR, bem como na ferramenta gráfica NoIA (Network-on-Chip Interactive Analysis) que promove uma avaliação visual do resultado dos algoritmos aplicados.

Figura 12 – Passo a passo da geração de resultados analíticos e tabela de roteamento



Fonte: Elaborada pelo autor.

Neste capítulo serão apresentados cada um dos passos exibidos na figura 11 que ilustra a estratégia utilizada para obtenção das métricas analíticas. Inicialmente é apresentada a forma de geração das topologias, seguida da geração do algoritmo de roteamento e do mecanismo de roteamento. Então são apresentadas as funcionalidades da interface gráfica desenvolvida e a geração das métricas analíticas.

#### 3.1 Geração de Topologias

Neste trabalho é usado um modelo de falhas para a análise das métricas analíticas. Esse modelo se baseia no crescimento da tecnologia deep-submicron e diminuição da área de silício usada. Esse modelo de variabilidade de fabricação de CMOS, do inglês Complementary Mosfet Oxide Silicon, é descrito em (Hargreaves; Hult; Reda, 2008) e é usado para gerar cenários de falhas sintéticos.

O modelo de falhas é fundamentado em dois parâmetros: o atraso ( $\sigma$ ) e a variabilidade de correlação espacial ( $\lambda$ ). De acordo com o (ITRS 2007), as tecnologias CMOS de 65nm e 22nm são modelados usando  $\sigma=0,05$  e  $\sigma=0,18$ , respectivamente. Para a correlação típica em processos de fabricação (Hargreaves; Hult; Reda, 2008), foram escolhidos  $\lambda=0,4$  como alta força de variabilidade e  $\lambda=1,2$  como baixa força de variabilidade.

Com relação à dimensão da topologia do tipo malha foram usados quatro dimensões: 5x5, 6x6, 7x7 e 8x8. Combinados com a correlação espacial e o atraso geram 16 cenários de falha. Para alcançar um melhor nível de aleatoriedade do modelo, cada cenário foi gerado 3 vezes resultando em 48 cenários. A Tabela 1 mostra a expansão desses cenários que resultou em 12224 cenários.

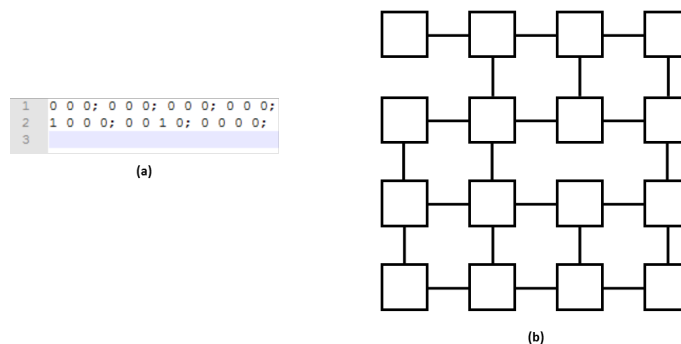
Tabela 1 – Expansão dos cenários de falha para 48 topologias de redes em malha irregulares

NoC sizes	Distribution fault		Amount of fault channels (3 samples)	Amount of scenarios expanded (3 samples)	Total amount of scenarios
	$\sigma$	$\lambda$			
5x5	0.05	1.2	4, 4, 4	15, 15, 15	45
	0.18	1.2	4, 4, 4	15, 15, 15	45
	0.05	0.4	6, 6, 6	63, 63, 63	189
	0.18	0.4	6, 6, 6	63, 63, 63	189
6x6	0.05	1.2	5, 5, 5	31, 31, 31	93
	0.18	1.2	5, 5, 5	31, 31, 31	93
	0.05	0.4	7, 7, 7	127, 127, 127	381
	0.18	0.4	7, 7, 7	127, 127, 127	381
7x7	0.05	1.2	5, 5, 5	31, 31, 31	93
	0.18	1.2	5, 5, 5	31, 31, 31	93
	0.05	0.4	9, 9, 9	511, 511, 511	1533
	0.18	0.4	8, 8, 8	255, 255, 255	765
8x8	0.05	1.2	4, 4, 4	15, 15, 15	45
	0.18	1.2	5, 5, 5	31, 31, 31	93
	0.05	0.4	11, 11, 10	2047, 2047, 1023	5117
	0.18	0.4	10, 10, 10	1023, 1023, 1023	3069
Total			48 NoC topologies	12224 scenarios	12224

Fonte: (SILVEIRA, 2015b).

Cada cenário é representado por um arquivo de texto que indica o valor 0 quando um link não é falho e 1 quando o link é falho. A primeira linha do arquivo representa os links horizontais e a segunda linha representa os links verticais. A Figura 13 exemplifica um arquivo de topologia e a sua redetendo os links como as linhas e os roteadores são os quadrados.

Figura 13 – (a) padrão do arquivo de topologia. (b) topologia referente ao arquivo



Fonte: Elaborada pelo autor.



## 3.2 Algoritmos Implementados

Nesta seção são detalhados o algoritmo SBR, a computação de caminhos e o mecanismo RBR implementados na ferramenta. Seus passos são representados pelos 2 e 3 (SBR), 4 (computação de caminhos) e 5 e 6 (RBR) da Figura 12, respectivamente.

Quando a topologia de entrada (arquivos de topologia) é lida, é gerado um grafo no qual as arestas representam os links e os vértices representam os nodos da topologia.

### 3.2.1 SBR

No SBR ocorre a segmentação da rede e então impõe restrições em cada segmento. Dessa forma, as restrições impedem que ocorra um fluxo cíclico de pacotes que resultaria em um deadlock (MEIJA et al., 2006).

Na implementação do algoritmo SBR os roteadores podem ser classificados como starting ou terminal e cada subrede só pode conter apenas um roteador starting. A Figura 9 exemplifica os termos starting e terminal.

Os roteadores e links variam entre três estados na rede: nvisited, visited e tvisited. Todos os roteadores e links iniciam o processo de segmentação no estado nvisited que indica que o elemento em questão (roteador ou link) não foi visitado pelo procedimento de segmentação.

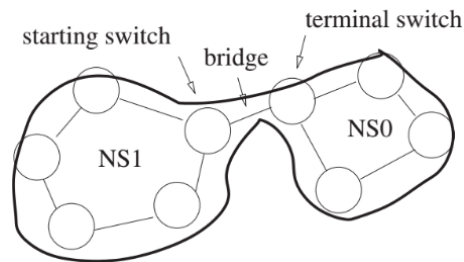
No início desse procedimento um roteador é determinado como starting e tvisited. Preferencialmente este roteador está posicionado nas extremidades da topologia, mas a seleção do roteador é randômica. A partir desse roteador ocorre a tentativa de formação de um segmento circular, incluindo links e roteadores não visitados (nvisited). Esses links e roteadores incluídos passam para o estado de temporariamente visitados (tvisited), enquanto ocorre a tentativa de formação do segmento. Esse segmento deve iniciar e terminar no roteador inicial.

Caso esse passo seja concluído, é formado o segmento inicial da rede (start) e os links e roteadores envolvidos passam para o estado visited. Caso contrário, se em algum dos links desse roteador não for possível efetuar o segmento start, o roteador é classificado, também, como terminal e os links e roteadores que anteriormente estavam no estado tvisited retornam para o estado nvisited.

Caso nenhum dos seus links permitam a criação do segmento start, esse roteador pertencerá a uma subrede e um roteador adjacente será selecionado para ser o starting da outra nova subrede e, assim, iniciar novamente a tentativa de criação de um segmento start. Nesse caso, o link que conecta essas duas subredes é chamado de bridge. A Figura 14 ilustra esse caso.

Considerando que o segmento start foi criado, é selecionado um roteador desse segmento que tenha um dos seus links marcados como nvisited. A partir dele é iniciada a tentativa de criação de um segmento regular em que os roteadores e links adicionados passam, também, para o estado tvisited e seguem os casos de alteração de estado do segmento start anteriormente descrito.

Figura 14 – Exemplo de um link bridge

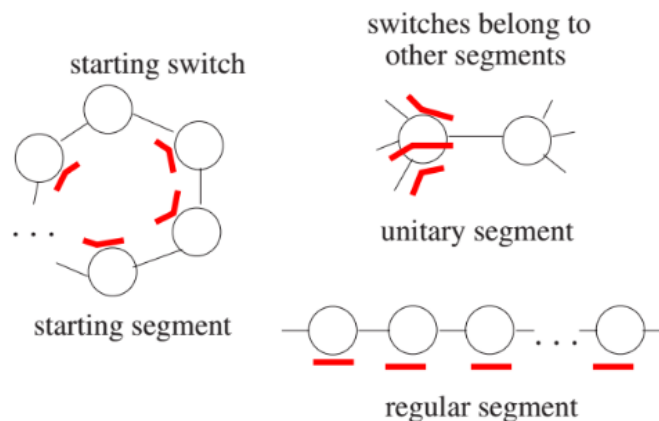


Fonte: (MEIJA et al., 2006).

Nesse procedimento de criação de segmento pode ocorrer o caso de um link não visitado (nvisited) estar situado entre dois roteadores já visitados (visited). Então, nesse caso, ocorre a criação de um segmento unitário.

Visto que os segmentos são disjuntos, cada um terá sua restrição imposta de forma independente. No segmento start a restrição pode ser inserida em qualquer roteador, com exceção do roteador starting. Em um segmento regular a restrição pode ser inserida em qualquer roteador do segmento. Já no segmento unitário são definidas restrições que proíbem qualquer caminho passar pelo link desse segmento (MEIJA et al., 2006). Portanto, esse link se torna inválido para a rede. A Figura 15 ilustra algumas possibilidades de restrições em cada tipo de segmento.

Figura 15 – Algumas possibilidades de posicionamento de restrições



Fonte: (MEIJA et al., 2006).

Concluídos esses passos, temos a rede dividida em subredes compostas por segmentos. Em cada segmento existirá apenas uma restrição em somente um dos seus roteadores. Assim, tráfegos cíclicos são proibidos e o algoritmo se torna livre de deadlocks.

### 3.2.2 Computação de Caminhos

Para a definição dos caminhos possíveis para cada par de roteadores origem-destino foi implementado um algoritmo de busca de caminhos. O algoritmo obedece as restrições

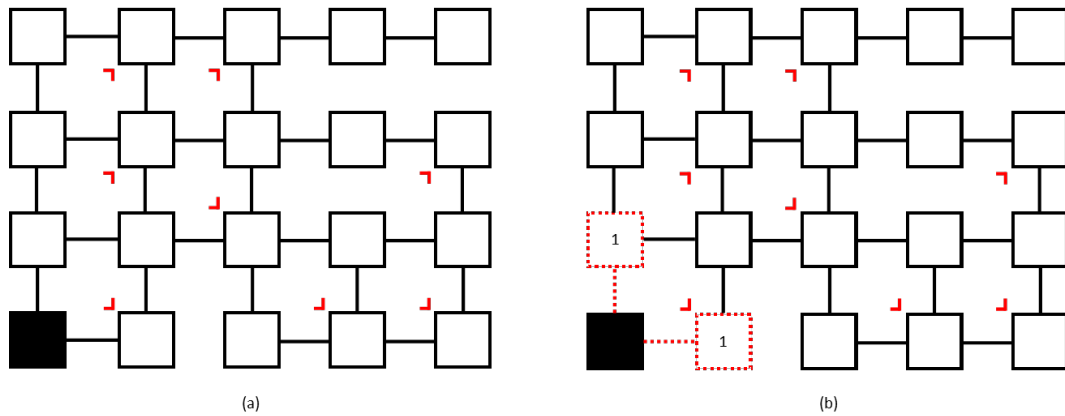
impostas no grafo. Como base são usadas duas listas nesse algoritmo: a lista de caminhos e a lista de pares origem-destino. Na lista de pares cada elemento é somente a identificação do roteador de origem e do roteador de destino cujos caminhos já foram computados.

Inicialmente são computados todos os caminhos que contêm somente um hop (saltos entre roteadores num caminho). Para cada roteador da rede são observados todos os seus links resultando na adição de um novo caminho na lista de caminhos e o seu par origem-destino é adicionado na lista de pares.

Nesse caso inicial, a única restrição que poderia proibir esses caminhos de um hop seriam as restrições de segmentos unitários (proíbe todos os fluxos, inclusive proveniente da porta local).

As Figuras 16 e 17 mostram uma topologia irregular com as restrições posicionadas e com o roteador (sinalizado em preto) adotado como origem para a construção dos caminhos. Os caminhos são os links e roteadores de borda pontilhados.

Figura 16 – (a) roteador selecionado para computar os caminhos. (b) caminhos de um hop computados



Fonte: Elaborada pelo autor.

Após essa computação inicial, é verificado o número total de pares origem-destino na topologia em questão. Esse valor é obtido através da Equação 3.1.

$$P = x \times y \times ((x \times y) - 1) \quad (3.1)$$

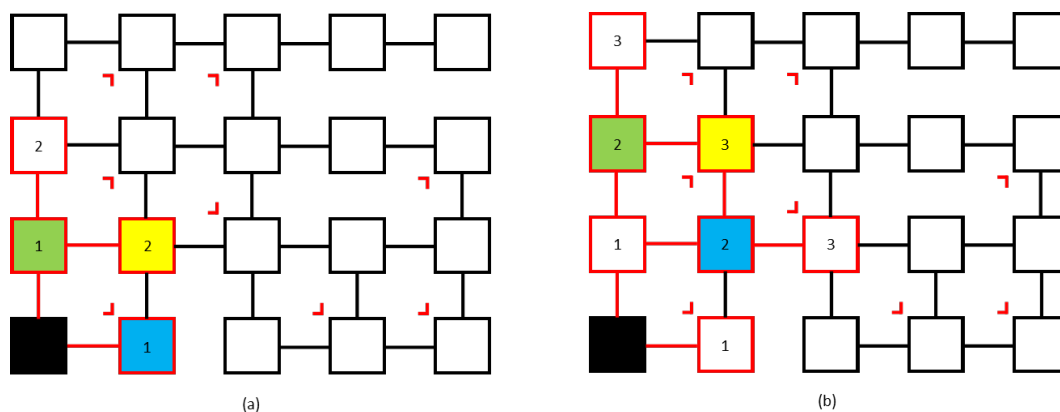
Onde  $x$  é a dimensão da topologia no eixo  $x$  (na Figura 16 tem valor 5) e  $y$  é a dimensão no eixo  $y$  (na Figura 16 tem valor 4).  $P$  é usado na trecho de busca de caminhos com mais de um hop, sendo o critério de parada quando o total de pares origem-destino se iguala a ele. Ou seja, quando forem encontrados todos os caminhos para todos os pares possíveis.

Para garantir somente caminhos mínimos, ocorre a adição de caminhos por tamanho. Para cada roteador destino de cada caminho, a cada iteração, os caminhos feitos pelos seus links são adicionados como um caminho novo com um hop a mais, exceto o link por onde o caminho veio a este roteador.

No trecho de adição de um par origem-destino à lista de pares ocorre a verificação da existência desse par na lista. Se esse par existir na lista, o caminho em questão é não mínimo e é descartado, inclusive, da lista de caminhos. Caso contrário, o caminho é mínimo e é adicionado na lista de caminhos.

A Figura 17 exemplifica esse passo a passo.

Figura 17 – (a) novos caminhos de dois hops. (b) novos caminhos de três hops



Fonte: Elaborada pelo autor.

A lista de pares só é alterada após a computação de todos os caminhos de mesma quantidade de hops. No caso da Figura 17(a), todos os caminhos de 2 hops foram computados para o roteador sinalizado em preto, então, são incluídos os pares origem-destino na lista de pares e se inicia outra iteração (agora para 3 hops, como na Figura 17(b)). Portanto, isso permite que existam mais de um caminho mínimo computado e adicionado para um mesmo par origem-destino.

Na Figura 17(a) a restrição no roteador azul impediu que um caminho fosse formado. Entretanto, o roteador que seria o destino desse caminho (amarelo) pode ser alcançado pelo outro caminho formado até ele (através do roteador verde). Os números internos nos roteadores indicam quantos hops são necessários para que um pacote que saiu da origem (roteador preto) alcance o destino.

Na Figura 17(b) temos o roteador de destino amarelo que tem duas alternativas de acesso: através do roteador azul e do verde. Esse é um caso de um par origem-destino com mais de um caminho mínimo. Também poderia acontecer o caso de o caminho encontrado para um par origem-destino não ser o mínimo por conta de restrições ou falhas na rede e de já haver um caminho mínimo para esse par. Por fim, caminhos de mais hops serão computados e todas as origens terão todos os caminhos computados para todos os destinos.

### 3.2.3 RBR

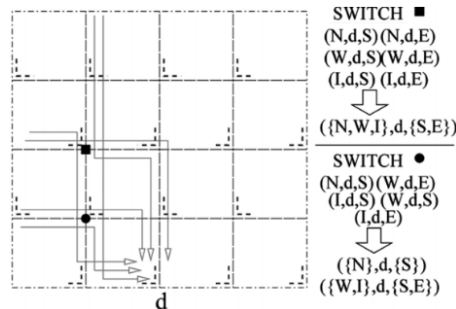
Inicialmente o mecanismo RBR implementado recebe o conjunto de caminhos e dela extrai as opções de roteamento para cada vértice do grafo. Uma opção de roteamento

indica por qual porta de saída um pacote deve usar para atingir um destino, com base na porta de entrada. Definido pela tupla (ip, dst, op) na qual ip é uma porta de entrada, dst é um destino e op é uma porta de saída.

Cada roteador tem seu conjunto de opções de roteamento que indicam por qual porta um pacote deve sair com base no seu destino e por qual porta ele entrou. As tuplas referentes a um mesmo roteador podem ser agrupadas por portas de saída, de entrada ou ambos. No agrupamento por portas de entrada, é necessário que as tuplas tenham as mesmas portas de saída para um mesmo destino. Da mesma forma é o agrupamento das portas de saída: precisam ter o mesmo destino e as mesmas portas de entrada.

A Figura 18 exemplifica o agrupamento de opções de roteamento.

Figura 18 – Agrupamento de opções de roteamento



Fonte: (MEIJA et al., 2009).

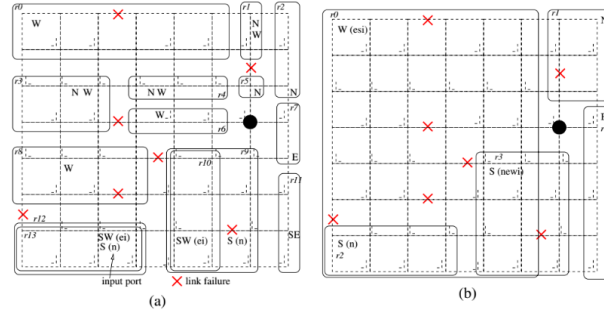
Após o agrupamento das opções de roteamento ocorre a formação das regiões que são definidas cada uma pela tupla ( $\{ \text{lista ip} \}, \text{Rsup}; \text{Rinf}, \{ \text{lista op} \}$ ). As opções de roteamento que têm o mesmo conjunto de portas de entrada (lista ip) e portas de saída (lista op) formam uma nova região com os destinos dessas opções de roteamento. Uma região é definida por um roteador superior esquerdo (Rsup) e um inferior direito (Rinf), formando, assim, uma região retangular.

Portanto, cada roteador vai ter um conjunto de regiões que agrupam todos os seus destinos existentes na rede. Na ocorrência da criação de regiões desnecessárias ocorre o agrupamento de regiões onde, para agrupar regiões, as portas de saída de uma região deve ser um subconjunto das portas de saída de outra. Bem como duas ou mais regiões podem juntar-se e formar uma única região (FLICH et al., 2007).

A Figura 19 exemplifica o resultado do agrupamento de regiões de um roteador específico.

Após o agrupamento das regiões de todos os roteadores, a fim de reduzir a tabela de roteamento, cada vértice terá o seu grupo de regiões formadas.

Figura 19 – (a) regiões formadas. (b) regiões resultantes após o agrupamento



Fonte: (FLICH et al., 2007).

### 3.2.4 Métricas

Concluídos os procedimentos anteriores, são coletadas as métricas analíticas: LW, STD LW, RMAX, UnitSeg e ARD (passo 7 da Figura 12).

A métrica ARD, do inglês Average Routing Distance, é a soma do comprimento dos caminhos em hops de todos os caminhos dividido pela quantidade de caminhos. A Equação 3.2 define essa métrica. LW, do inglês Link Weight, é a média de todos os pesos dos links (FLICH et al., 2012). O peso de um link é definido como a quantidade de caminhos que passam por ele. A Equação 3.3 define o LW.

$$ARD = \frac{1}{C} \sum_{c=0}^C h_c \quad (3.2)$$

Onde  $C$  é o total de caminhos existentes na rede e  $h_c$  é o número de hops no caminho  $c$ .

$$LW = \frac{1}{L} \sum_{l=0}^L p_l \quad (3.3)$$

Onde  $L$  é o total de links da rede e  $p_l$  é o peso do link  $l$ . O STD LW, do inglês Standard Deviation of LW, é o desvio padrão de LW.

RMAX informa a quantidade máxima de regiões presente nos roteadores da rede e UnitSeg é a quantidade de segmentos unitários presentes na segmentação da rede (FLICH et al., 2012).

### 3.2.5 Interface Gráfica

A ferramenta desenvolvida tem como objetivo fornecer um resultado visual referente ao algoritmo de roteamento executado nos passos 2 e 3 da Figura 12 e ao mecanismo de implementação por tabelas executado nos passos 5 e 6 da Figura 12. Bem como exibir as métricas analíticas.

Os requisitos funcionais do software foram: ler um arquivo de topologia no padrão definido; executar os algoritmos implementados; extrair as métricas analíticas; gerar a tabela de

roteamento; e desenhar a topologia com suas características e resultados (segmentos, restrições, regiões, peso de link, links unitários, links bridge e exibir links falhos). A execução do software pode ser por meio de linha de comando ou utilizando a interface gráfica. No caso da execução por linha de comando, o usuário deve indicar o caminho para o arquivo de topologia. Na execução por interface gráfica ocorre a solicitação de indicação do arquivo de topologia, então, ao confirmar o arquivo, ocorre todo o processo dos passos 2 ao 7 da Figura 12.

Após esse momento, o software se baseia no grafo para desenhar a topologia e a janela principal é dimensionada proporcionalmente à área ocupada pelo desenho. Nessa janela existe uma barra de opções de: exibição dos resultados visuais da topologia, exibir características da rede, opções de salvar a imagem desenhada (no formato PNG – Portable Network Graphics) e de processar um novo arquivo de topologia.

No caso das opções de exibição de resultados visuais, o desenho é atualizado com a exibição da opção desejada. Somente a opção que exibe regiões requer que o usuário especifique o roteador, visto que cada roteador tem o seu conjunto de regiões. Uma janela secundária é criada quando a opção de exibir características da rede é escolhida. Então, os resultados analíticos são exibidos juntamente com o nome do arquivo da topologia em exibição.





## **4 DISCUSSÃO DOS RESULTADOS**



## **5 CONCLUSÃO**