

# Avaliação de topologias de rede utilizando o gem5

Evandro Fiorim Estima<sup>1</sup>, Diogo Alves de Almeida<sup>2</sup>, Cristofer Oswald<sup>3</sup>

<sup>1</sup>Departamento de Informática, Universidade Estadual de Maringá

**Resumo.** *Topologia de rede é o padrão de elementos interconectados dentro de uma rede. Uma topologia pode ser física, mapeando configurações de hardware, ou lógica, mapeando o caminho que os dados devem seguir para viajar dentro da rede. Este é o caso deste trabalho, que aborda mais especificamente os algoritmos Mesh e Octagon.*

**Abstract.** *Network topology is the interconnected pattern of network elements. A network topology may be physical, mapping hardware configuration, or logical, mapping the path that the data must take in order to travel around the network. This is this work's case, which will approach more specifically the Mesh and Octagon algorithms.*

## 1. Introdução

Quando se fala em topologia de rede, se fala do canal no qual o meio de rede está conectado aos computadores e outros componentes de uma rede de computadores. Essencialmente, é a estrutura topológica da rede, e pode ser descrito física ou logicamente.

Existem variadas topologias de rede diferentes, onde cada uma tem suas vantagens e desvantagens. Por exemplo a Mesh, que é de fácil implementação mas alto custo. Estas topologias não são muito rigorosas, ou seja, elas podem ser combinadas. Porém, cada topologia tem um padrão diferente, e podem usar métodos diferentes no hardware, logo, elas não são intercambiáveis. Dentre os tipos de topologias de rede estão a Mesh e a Octagon, que serão descritas neste trabalho.

A Seção 2 abordará o simulador gem5, explicando qual seu objetivo e quais suas funcionalidades. Durante o desenvolvimento deste trabalho, teremos duas configurações distintas, utilizando duas topologias diferentes (Mesh e Octagon), que serão descritas na seção 3. Na seção 4, serão expostos e discutidos os resultados das simulações, e por fim, na seção 5, é feita uma conclusão do trabalho.

## 2. O Simulador gem5

O simulador gem5 é uma plataforma modular usada para pesquisas de arquiteturas de sistemas de computadores, abrangendo arquiteturas no nível do sistema, além de processadores de microarquitetura.

Algumas de suas funcionalidades são

- Vários modelos de CPU que podem ser trocados; são quatro modelos de CPU baseados em interpretação: UM *one-CPI* CPU; um modelo detalhado de um CPU *in-order*, e um modelo detalhado de uma cpu *out-of-order*. Além desses, o gem5 possui uma CPU baseada em KVM que usa virtualização que acelera a simulação.
- UM modelo de GPU integrado que executa uma máquina ISA real e suporta memória virtual compartilhada com a CPU anfitriã.

- Um modelo GPU NoMali integrado que é compatível com os drivers GPU dos sistemas Linux e Android.
- Um sistema de memória dirigido por eventos. O gem5 possui um detalhado sistema de modelo de memórias, incluindo caches, crossbars, filtros snoop e um modelo DRAM rápido e preciso.
- Modo *Full-system*.
  - Alpha: O gem5 modela um sistema DEC Tsunami com detalhes suficientes para iniciar o Linux 2.4/2.6, FreeBSD, ou o L4Ka::Pistachio.
  - ARM: O gem5 consegue modelar até 64 cores (heterogêneos) de uma plataforma ARM *RealView*, e consegue fazer o *boot* no Linux e no Android com uma combinação de CPUs *in-order* e *out-of-order*. A implementação ARM suporta kernels e aplicações de 32 ou 64 bits.
  - SPARC: O simulador gem5 possui um único core de um processador UltraSPARC T1 com detalhes suficientes para iniciar o Solaris de uma maneira similar às ferramentas *Sun T1 Architecture simulator*.
  - x86: O simulador gem5 suporta plataformas PC padrão.

### 3. Arquiteturas simuladas

Nesta seção serão abordadas as topologias utilizadas para os experimentos deste trabalho.

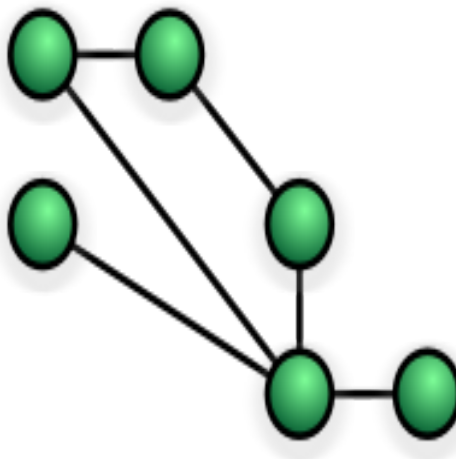
#### 3.1. Topologia Mesh

A topologia mesh foi desenvolvida há mais de 30 anos visando aplicações militares, e é composta de vários nós/roteadores, que passam a se comportar como uma única e grande rede, possibilitando que o cliente se conecte em qualquer um destes nós. Os nós têm a função de repetidores e cada nó está conectado a um ou mais dos outros nós. Desta maneira é possível transmitir mensagens de um nó a outro por diferentes caminhos. Já existem redes com cerca de 500 nós e mais de 400.000 usuários operando.

Apesar de redes do tipo mesh serem de fácil implantação e bastante tolerantes a falhas, possuem a desvantagem de possuir um alto custo.

O segredo do sistema mesh está no protocolo de roteamento, que avalia as diversas possibilidades de rotas de fluxo de dados baseado numa tabela dinâmica, onde é selecionada a rota mais eficiente a seguir para chegar ao seu objetivo, levando em conta a maior rapidez, com menor perda de pacotes, etc. Por exemplo, quando o nó que estava sendo utilizado pára de funcionar, o sistema se rearranja automaticamente, desviando o nó defeituoso, sem que usuário perceba.

Uma mesh pode ser completa ou parcialmente conectada. Um exemplo de uma rede mesh parcialmente conectada pode ser visto na figura 1.

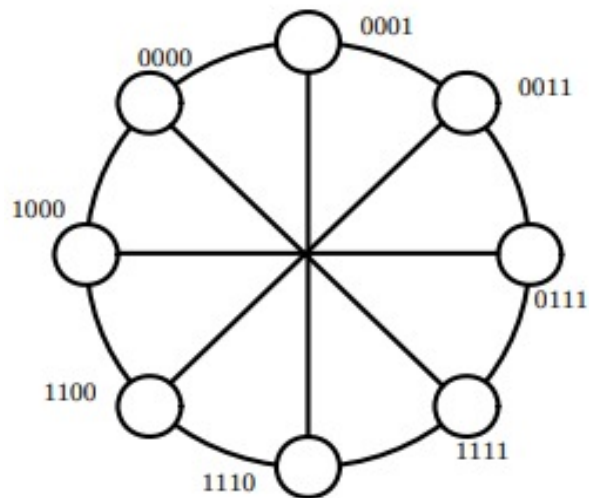


**Figure 1. Mesh parcialmente conectada.**

### **3.2. Topologia Octagon**

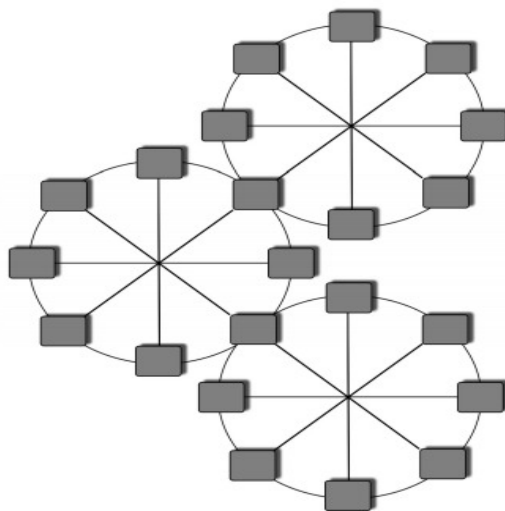
A rede Octagon foi desenvolvida pela STMicroelectronics para processadores de rede. Sua topologia é baseada em anel e sua configuração básica consiste de oito nós e doze links bidirecionais, permitindo que a comunicação entre qualquer par de nós seja realizada com no máximo dois saltos, passando por um único nó intermediário. Para isso, um algoritmo que sempre indica o caminho mínimo entre dois nós é utilizado.

Segundo [JUNIOR 2010], Os pacotes na rede Octagon podem ser de tamanhos fixos ou variáveis, fazendo com que essa rede possa operar tanto em modo packet quanto em modo circuit switching. Para permitir a operação no modo circuit switching um escalonador denominado best-fit foi desenvolvido. O escalonador consiste em um protocolo de comunicação orientado a conexão que pode acomodar simultaneamente diversas conexões que não se sobrepõem. Para isso, cada roteador possui três filas de requisições de saída, uma para cada porta do roteador. O escalonador verifica as primeiras posições de todas as filas e estabelece as conexões por ordem de espera. Todas as conexões possíveis de requisições nas primeiras posições das filas são estabelecidas. Quando um pacote termina de ser transportado o escalonador é reativado para verificar se existem novos pedidos de conexão. Essa estratégia permite melhorar o desempenho do sistema e utilização dos nós em relação a alguns protocolos de comunicação que bloqueiam o nó requerente até que seu pedido possa ser atendido. No entanto, cada nó da Octagon deve possuir uma fila grande o suficiente para evitar perdas de pacotes.



**Figure 2. Estrutura básica de uma rede octagon. Retirado de [Liu et al. 2015]**

Para conseguir maior escalabilidade, afim de possibilitar que mais núcleos sejam em uma rede Octagon, é possível interconectar vários anéis com a utilização de um nodo ponte, que não é ligado a nenhum núcleo, mas conecta anéis adjacentes, como é visto na Figura 3.



**Figure 3. Octagon com 3 níveis de anéis. Retirado de [JUNIOR 2010]**

#### **4. Resultados e discussão**

Nesta seção serão mostrados alguns resultados das simulações feitas. A latência do mesh foi 16% menor em relação ao octagon devido a quantidade de vazão por ciclo que o octagon teve a mais que o mesh, nesse caso temos que o octagon teve 25 vezes mais dados escritos e lidos fazendo com que a latência diminua. Os saltos (hops) foram 35% menor que o do mesh confirmado seu melhor desempenho. Além de que o octagon processou mais dados em menos tempo e como também tem mais ligações podemos ver que ele gastou 37% a mais de energia do que o mesh.

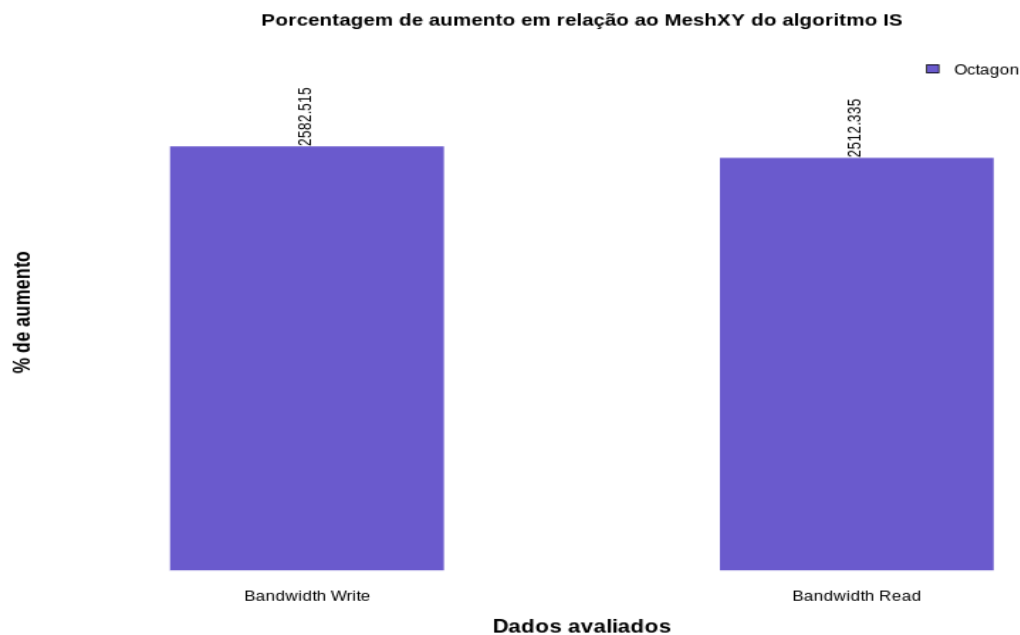


Figure 4. Porcentagem de aumento de vazão de leitura e escrita

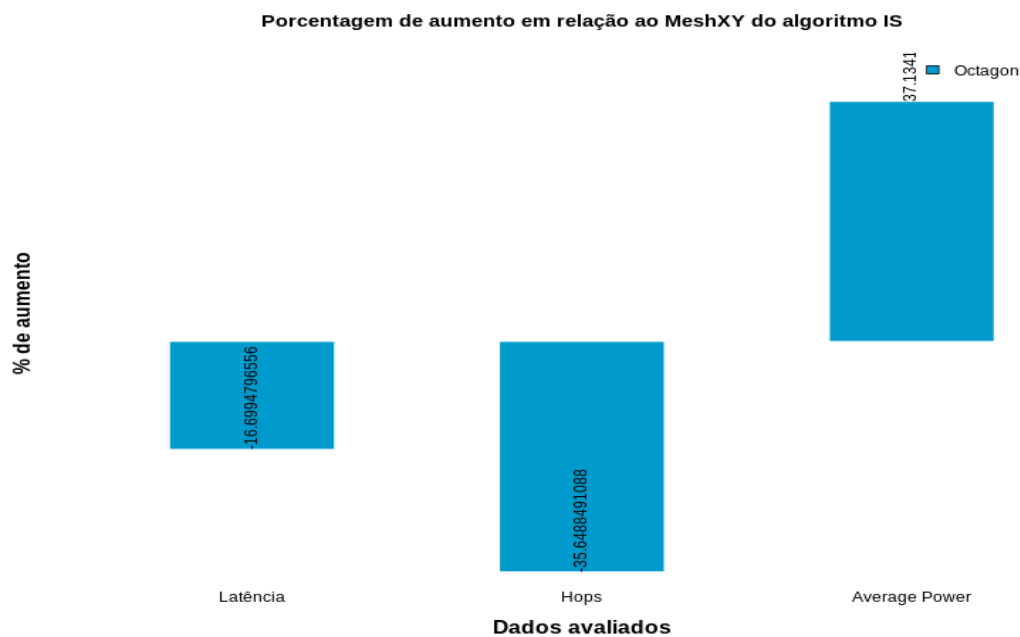


Figure 5. Porcentagem de aumento de latência, saltos e força média

## 5. Conclusão

As topologias de rede são usadas para mapear os caminhos que os dados devem seguir de forma mais eficiente, assim, o objetivo deste trabalho foi comparar duas topologias (Mesh

e Octagon) e verificar seus desempenhos.

Para atingir este objetivo, utilizamos o simulador gem5 no modo *system emulation*, rodando com cada topologia, os algoritmos IS, KM, Fast, FN, GF, LU e o TSP, retirados do CAPBenchmark.

Ao fim, pôde-se observar que por conta da maior vazão, (ou seja, mais dados sendo processados de uma vez) da topologia octagon, obteve-se um ganho de aproximadamente 16% na latência, e de aproximadamente 35% no número de saltos tomados. Logo, a topologia octagon se mostrou mais eficiente. No entanto, por conta do número elevado de operações, tivemos um aumento de cerca de 37% no consumo de energia.

## References

- JUNIOR, N. A. G. (2010). *Análise e simulação de topologias de redes em chip*. PhD thesis, Master's thesis, Universidade Estadual de Maringá.
- Liu, Y., Xing, L., and Zhou, X. (2015). Topology architecture and routing algorithms of octagon-connected torus interconnection network. *Telkomnika*, 13(1):305.