

# Gerenciamento de Fila com Operação XOR em Redes de Comunicação Cooperativa

Hyago V. L. B. Silva; Luiz Renault Leite Rodrigues

Instituto Nacional de Telecomunicações - INATEL

Santa Rita do Sapucaí - MG

hyago.silva@mtel.inatel.br; luiz.rodrigues@dtel.inatel.br

**Resumo**—Este relatório explora o desempenho de um tipo de fila com aplicação em um nó retransmissor (relay) em uma rede de comunicação cooperativa. O relay utiliza uma estratégia de transmissão diferenciada onde, se houver mais de um pacote na fila, realiza a combinação de dois pacotes e transmite o resultado. Um exemplo de aplicação deste esquema de transmissão e sua eficiência sob o ponto de vista da capacidade da fila são apresentados em comparação com a transmissão direta simples. Foram realizadas simulações utilizando a linguagem Python no ambiente Google Colab para modelar o comportamento das filas e estimar as estatísticas usadas na comparação. Os resultados mostram que o modelo de fila analisado pode reduzir a sobrecarga do buffer de armazenamento.

**Index Terms**—filas, XOR, simulação

## I. INTRODUÇÃO

Redes de comunicação cooperativa têm se tornado uma área de crescente interesse devido à sua capacidade de melhorar a eficiência de transmissão e sua robustez. Nestas redes, nós retransmissores (relays) desempenham um papel crucial permitindo a aplicação da cobertura, especialmente em cenários onde a comunicação direta entre a fonte e o destino não é confiável ou eficiente [1].

Tradicionalmente, os relays retransmitem pacotes sem alteração à medida que são recebidos, o que pode sobrecarregar a rede. A figura 2 exemplifica o uso de uma estação retransmissora para aumentar o alcance e permitir que os terminais Alice e Bob se comuniquem. Neste caso, para uma troca bidirecional de pacotes entre os terminais, são necessárias quatro transmissões, ao invés de duas.

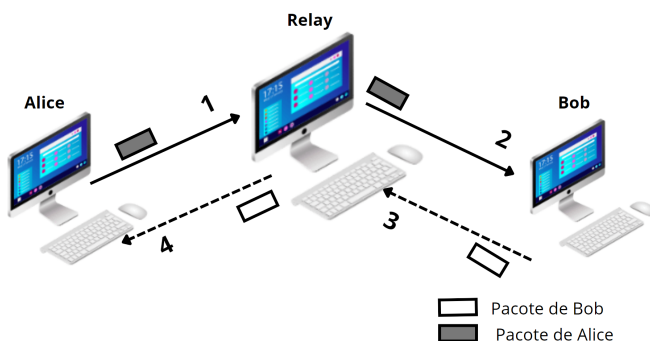


Figura 1: Exemplo de uso de retransmissor.

No entanto, diversas estratégias mais avançadas podem ser empregadas para melhorar a eficiência da transmissão. Uma dessas estratégias é o envio pela estação relay de pacotes combinados por alguma técnica que permita sua separação no nó de destino, diminuindo o tempo de transmissão [2]. Neste trabalho o autor propõe que a estação retransmissora envie apenas um pacote, resultado da combinação dos pacotes recebidos de Alice e Bob, que podem ser descombinados ao chegar nos destinos, recuperando o pacote de interesse, transmitido pelo outro terminal. Este esquema pode ser observado na figura ??, que mostra que são necessárias apenas três transmissões para que a mesma comunicação ocorra.

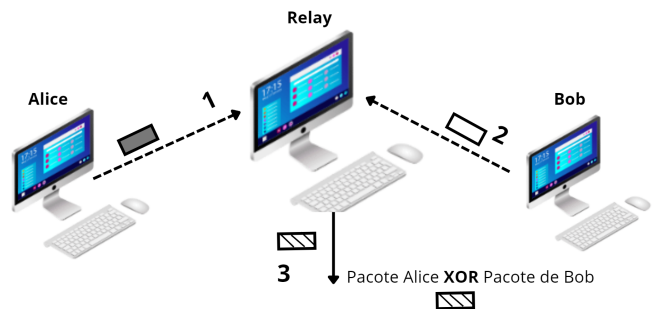


Figura 2: Exemplo de uso de retransmissor com envio de pacotes combinados.

De forma similar, um nó concentrador pode-se utilizar de múltiplos transmissores cooperativos empregando a mesma técnica de combinação de pacotes, podendo levar ao aumento da taxa, balanceado com o aumento da robustez de transmissão, duas melhorias extremamente relevantes em sistemas de comunicações [3], [1].

A figura 3 ilustra o uso de três transmissores para enviar simultaneamente dois pacotes distintos e uma combinação dos dois, chamada de paridade. Dessa forma, se um dos três elementos transmitidos for perdido e não recebido no destino, ele pode ser reconstruído através da operação inversa de combinação. Com isso, a taxa de transmissão é dobrada e é adicionada robustez ao sistema.

Para gerenciar estes fluxos diferenciados de pacotes, diversos tipos de fila podem ser utilizados. Este trabalho tem como objetivo explorar um destes tipos de fila, que apesar de possuir grande simplicidade de implementação, possibilita

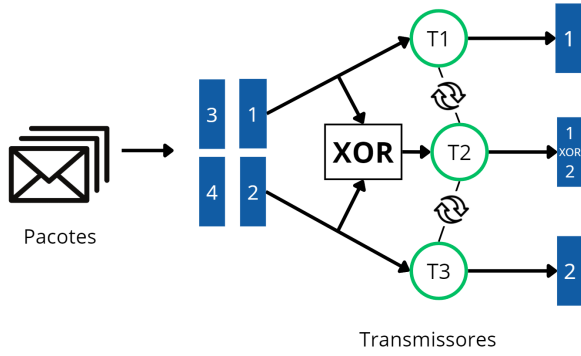


Figura 3: Exemplo de uso de múltiplos transmissores originais e combinados.

a combinação de pacotes como necessária acima. A fila analisada utiliza a operação lógica Ou-Exclusivo (XOR) bit a bit entre dois pacotes, conforme proposto em [2]. O funcionamento desta fila será detalhado adiante.

A lógica por trás da operação XOR é baseada no princípio de que a combinação de dois pacotes pode ser revertida no receptor final, desde que um dos pacotes originais e a combinação sejam conhecidos. Isso não apenas melhora a eficiência de transmissão e adiciona uma camada de robustez à comunicação, mas também dificulta a interceptação dos dados transmitidos e aumenta sua segurança [4]. No contexto de redes cooperativas, onde a confiabilidade e a eficiência são primordiais, essa técnica se mostra particularmente eficaz [5]. Neste estudo, é investigado o impacto da estratégia de combinação de pacotes utilizando a operação XOR no desempenho da fila do relay.

O restante deste relatório está organizado da seguinte forma: a seção II trás a modelagem da fila analisada, definindo as variáveis de estado, entidades e eventos do sistema, e a metodologia utilizada para simulação. A seção III apresenta os gráficos de estatísticas de desempenho da fila em alguns casos de uso selecionados, assim como as compara com as de outros tipos de filas tradicionais. Por fim, a seção IV avalia os resultados obtidos e apresenta sugestões para trabalhos futuros.

## II. MODELAGEM DA FILA

A fila analisada é composta de uma interface de chegada de pacotes, um buffer com capacidade para armazenar  $N$  pacotes, incluindo o que está em transmissão, e uma interface de transmissão de pacotes. Pacotes chegam ao relay com uma taxa média  $\lambda$  pacotes/segundo com distribuição arbitrária e são transmitidos a uma taxa média  $\mu$  pacotes/segundo, também com distribuição arbitrária. A política de serviço, ou transmissão do relay é a seguinte:

- Se houver um único pacote na fila, ele é transmitido imediatamente.
- Se houver dois ou mais pacotes na fila, o relay realiza uma operação XOR entre os dois primeiros pacotes e

transmite o pacote resultante, removendo os dois pacotes da fila.

Uma outra forma de interpretar a política de serviço é que sempre são retirados da fila dois pacotes que após a realização do XOR são transmitidos como um único pacote combinado. E no caso de haver apenas um pacote na fila, o outro é considerado um pacote nulo que, para fins da operação XOR, não altera o pacote original.

A fila pode ser classificada como  $G/G/1/N/\infty/FCFSXOR$  de acordo com a notação de Kendall [6], onde FCFSXOR representa a política de serviço descrita acima.

### A. Variáveis de Estado do Sistema

A representação do estado do sistema emprega dois valores, e é descrita como  $K_T$ , onde  $K = 1, 2, \dots, N$  representa o número de pacotes dentro do sistema e  $T = 0, 1, 2$  o número de pacotes sendo transmitidos.

A taxa de chegada de um pacote enquanto no estado  $K_T$  é  $\lambda_T^K$ . Uma chegada de pacote neste sistema representa um novo pacote a ser tratado pelo relay e efetivamente incrementa a quantidade  $K$  de pacotes no sistema.

Por outro lado, a taxa de partida é denotada por  $\mu_T^K$ . A partida neste sistema representa o término da transmissão de um pacote individual ou dois pacotes combinados, e efetivamente subtrai esta quantidade do número de pacotes  $K$  dentro do sistema.

Desta forma, o sistema é modelado permitindo processos de chegada e partida arbitrários.

O diagrama de transição de estados é mostrado na figura 4.

### B. Entidades

O sistema é composto pelas seguintes entidades:

- **Receptor:** recebe os pacotes a serem transmitidos;
- **Fila ou Buffer:** armazena os pacotes que estão aguardando a transmissão;
- **Combinador XOR:** combina dois pacotes para serem transmitidos;
- **Transmissor:** realiza a transmissão dos pacotes individuais ou combinados.

Para fins de simplificação o Combinador XOR e o Transmissor poderão ser agrupados representando a entidade Servidor. Os elementos do sistema estão representados na figura 5.

### C. Eventos

O fluxograma da figura 6 representa o evento de chegada do pacote e demonstra o como é a lógica da entrada do pacote na fila, se é encaminhado para ser transmitido ou é descartado.

Nele pode-se observar que na chegada de um novo pacote:

- Se a fila está cheia, ele é descartado;
- Se a fila contém pacotes e não está cheia, o pacote é colocado na fila;
- Se não há nenhum pacote na fila, ele é encaminhado para transmissão, sendo agendado um novo evento de partida
- Sempre são coletadas estatísticas.
- O controle é retornado para a rotina de avanço temporal da simulação.

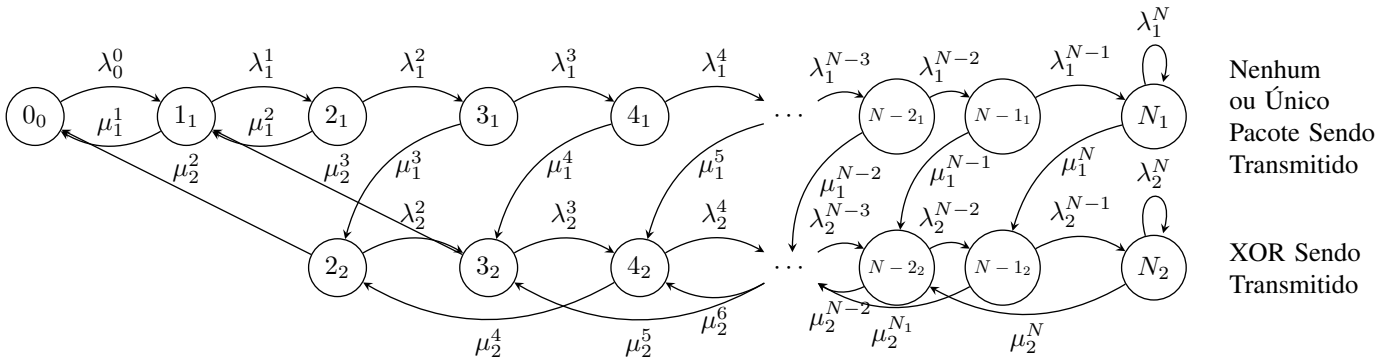


Figura 4: Diagrama de transição de estados

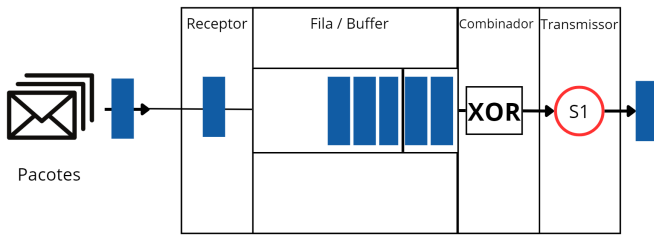


Figura 5: Diagrama dos elementos do sistema.

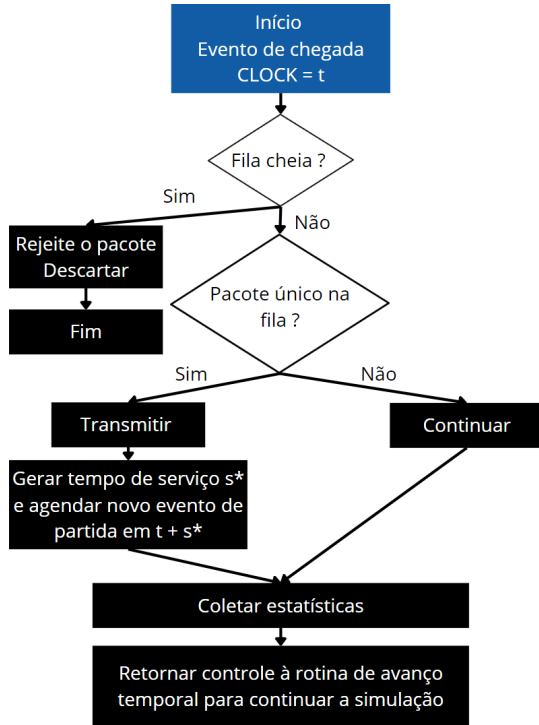


Figura 6: Evento de chegada

A figura 7, mostra o fluxograma do evento de partida do pacote em serviço e a lógica de início de um novo serviço.

Neste outro fluxograma, pode-se observar que na partida de um novo pacote / conclusão de transmissão:

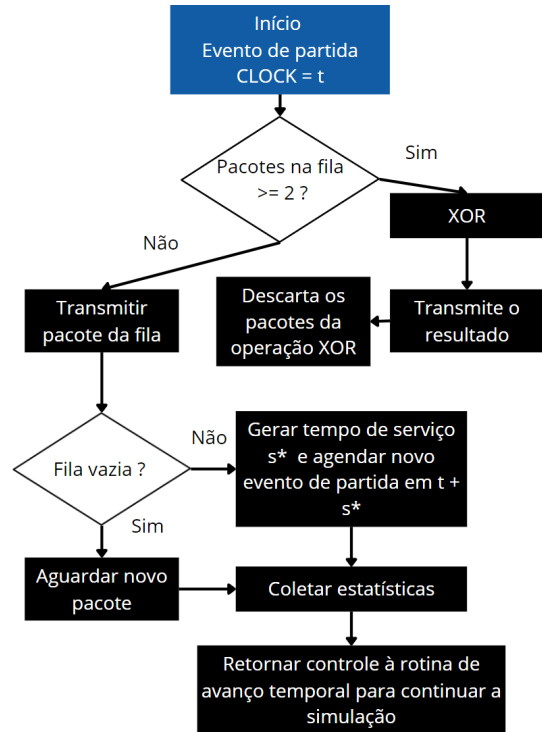


Figura 7: Evento de partida

- Se a fila está vazia, retorna o controle da rotina de avanço temporal da simulação.
- Se a fila possui um pacote, ele é retirado para ser transmitido e um novo evento de partida é agendado.
- Se a fila possui mais de um pacote, são retirados dois pacotes da fila, combinados através da operação XOR e um novo evento de partida é agendado.
- Sempre são coletadas estatísticas.

#### D. Simulação

Para analisar o desempenho do sistema e comparar com o de outros tipos de fila foram empregadas simulações utilizando a linguagem de programação Python no ambiente Google Colab. A implementação foi realizada visando ser o mais genérica possível, permitindo o uso de distribuições arbitrárias

de chegada e de partida, assim como a ativação paramétrica da função de combinação e definição de outros parâmetros do sistema e da simulação, como os descritos abaixo.

**Parâmetros do sistema e da simulação:**

- $TS$  : Tempo máximo de simulação
- $Nq$  : Tamanho da fila / buffer
- $lambda1$  : Taxa média de chegada [Pacotes/seg]
- $mu1$  : Taxa média de partida [Pacotes/seg]
- $chegada$  : Função que define o processo de chegada
- $partida$  : Função que define o processo de partida
- $xor$  : Parâmetro que define o número máximo de pacotes combinados

**Variáveis de Saída**

- $ts$  : Vetor dos tempos de permanência no sistema de cada pacote
- $tq \leftarrow np.mean(ts)$  : Tempo médio no sistema
- $tw \leftarrow tq - \frac{1}{mu1}$  : Tempo na fila
- $rho \leftarrow \frac{lambda1}{mu1}$  : Fator de utilização
- $b$  : Número de bloqueios
- $k1$  : Número de chegadas
- $k2$  : Número de serviços
- $k$  : Número de partidas
- $pb \leftarrow \frac{b}{k1}$  : Probabilidade de bloqueio
- $eq \leftarrow (1 - pb) \times lambda1 \times tq$  : Número médio de pacotes no sistema
- $ew \leftarrow (1 - pb) \times lambda1 \times tw$  : Número médio de pacotes na fila

A simulação inicia com o buffer e servidor vazios, e com o evento de chegada. A cada rodada da simulação, os três eventos de partida, chegada e serviço são realizados. A condição da realização dos eventos de partida e chegada é que o instante de tempo de simulação seja igual ao tempo agendado para estes eventos, tendo o evento de partida prioridade sobre o de chegada. A condição de realização do evento de serviço é testada em todas as iterações e depende somente de existirem pacotes na fila e o servidor estar vazio. Dessa forma são evitadas repetições de código de serviço nas rotinas de eventos de chegada e partida. As iterações de simulação são realizadas até que o tempo de simulação seja igual ao tempo máximo dado.

O pseudo-código do programa de simulação pode ser visto a seguir.

A validação do programa de simulação foi realizada utilizando processos de chegada e saída markovianos, tamanho fila infinita e parâmetro  $xor = 1$ , ou seja, uma fila do tipo M/M/1/∞/∞/FCFS, rotulada como "MM1" e M/M/1/5/∞/FCFS, rotulada como "MM1 Nq=5". Os resultados obtidos foram então comparados com os obtidos analiticamente para esta topologia, como ilustrado nas figuras 8 e 9.

### III. RESULTADOS

Para avaliar o desempenho do sistema XOR foram considerados processos markovianos de chegada e de partida, sem perda de generalidade, para que as comparações fossem

---

#### Algorithm 1 Simulação

---

**function** CHEGADA((void))

**return**  $\leftarrow np.random.exponential(1/lambda1)$

**end function**

**function** PARTIDA((void))

**return**  $\leftarrow np.random.exponential(1/mu1)$

**end function**

**while**  $t < TS$  **do**

**if**  $ta \leftarrow 0$  **then**

$k1 \leftarrow k1 + 1$

$ta \leftarrow np.append(tp, t)$

$ts = np.append(ts, (tp[k - 1] - tc[k - 1]))$

$ls = ls - 1$

$td = np.inf$

$print(f"PartidaF(lq)S(ls)T(t)")$

**end if**

**if**  $ls == 0$  **and**  $lq > 0$  **then**

**while**  $lq > 0$  **and**  $ls < xor$  **do**

$lq \leftarrow lq - 1$

$k2 \leftarrow k2 + 1$

$ls \leftarrow ls + 1$

**end while**

$td = t + partida()$

$print("SF(\{lq\})S(\{ls\})T(\{t\})")$

**end if**

$t \leftarrow \min(ta, td)$

**end while**

$tq \leftarrow \text{mean}(ts)$

$tw \leftarrow tq - \frac{1}{mu1}$

$eq \leftarrow lambda1 \times \text{mean}(ts)$

$ew \leftarrow lambda1 \times tw$

$rho \leftarrow \frac{lambda1}{mu1}$

$pb \leftarrow \frac{b}{k1}$

---

realizadas com filas do tipo MM1, inicialmente, e posteriormente com tamanhos de fila  $Nq = \{5, 10, 50\}$ . A partir deste ponto, todos os resultados apresentados para fila MM1 serão analíticos, enquanto para a fila XOR, os das simulações.

Os parâmetros usados nas simulações foram  $lambda1 = 1$ ,  $TS = 20000$  e  $xor = 2$ . O parâmetro  $\mu1$  foi calculado a partir do fator de utilização.

#### A. Análise de Desempenho

Para avaliar o desempenho, medimos o tempo médio que um pacote passa na fila (tempo de espera) e a taxa de utilização do buffer, que como  $lambda1 = 1$  são iguais. Variamos a taxa de serviço  $\mu1$  para observar como diferentes condições de rede afetam o desempenho. Veja a diferença ao utilizar fila MM1 e a XOR na figura 10, que mostra numero médio de pacotes na fila, pelo fator de utilização.

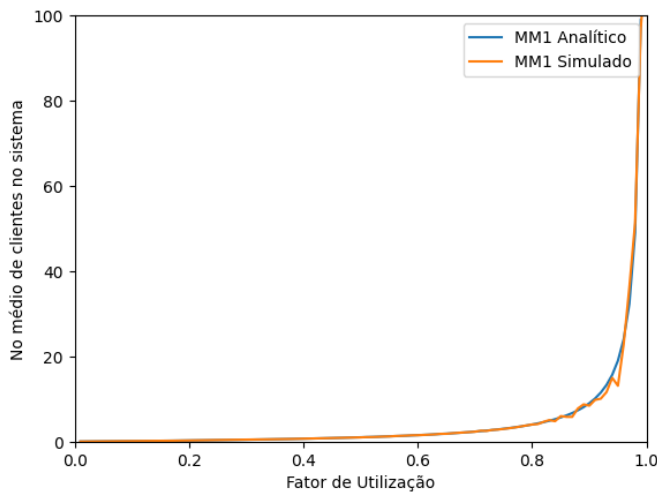


Figura 8: Resultado da validação da simulação para número médio de pacotes no sistema.

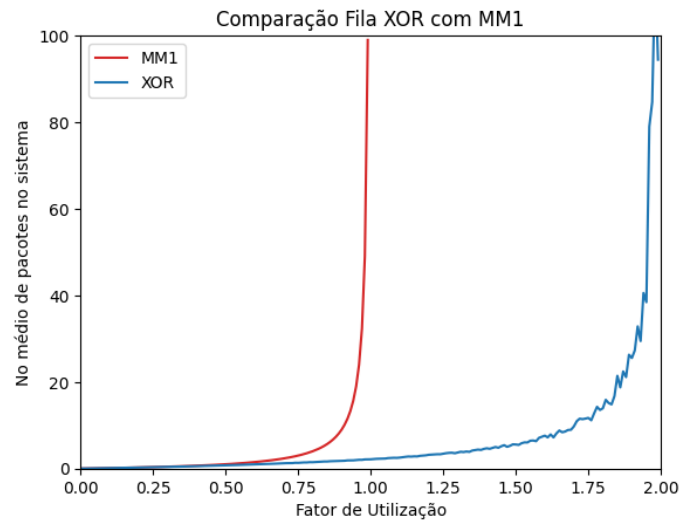


Figura 10: Comparação M/M/1 com XOR

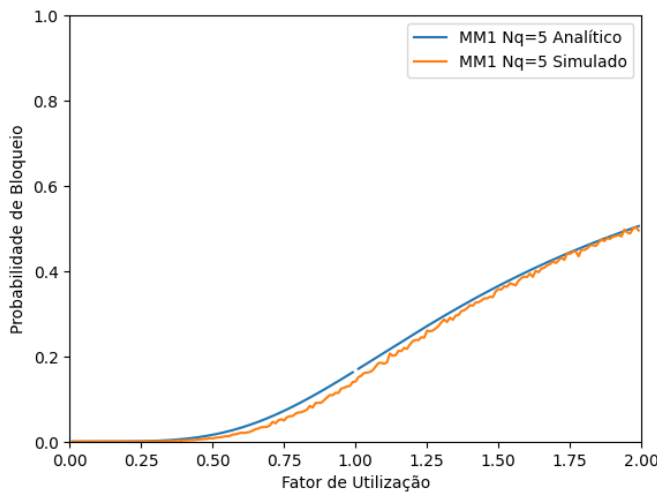


Figura 9: Resultado da validação da simulação para probabilidade de bloqueio.

de pacotes. Já o sistema MM1 não possui esta característica, sendo capaz de transmitir apenas um pacote por vez.

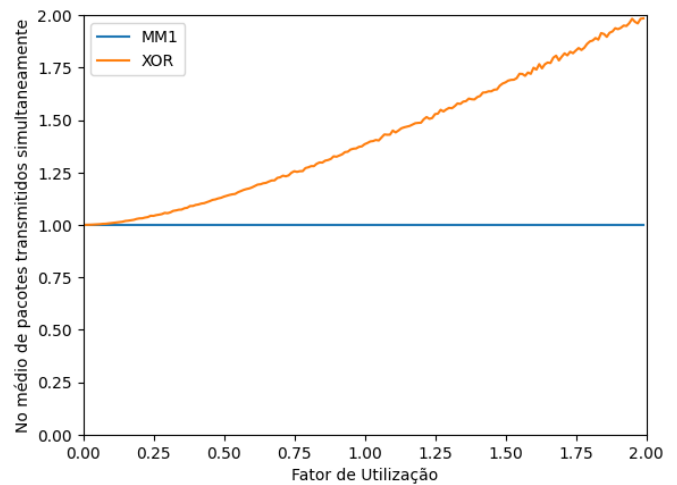


Figura 11: Número médio de pacotes simultâneos sendo transferidos.

Os resultados da simulação mostram que a utilização de operações XOR pode reduzir significativamente o tempo de espera médio dos pacotes na fila em comparação com a transmissão direta. Esta redução é mais pronunciada em cenários onde a taxa de chegada de pacotes é alta e a capacidade do buffer é limitada.

Ressalta-se que é possível utilizar a fila XOR mesmo em situações onde a taxa de chegada é maior do que a taxa de partida.

Abaixo estão alguns gráficos que mostram o desempenho do sistema sob diferentes condições.

A figura 11 mostra o número de pacotes simultâneos sendo transferidos pelos sistemas MM1 e XOR. Observa-se que com o aumento do fator de utilização, o sistema XOR passa a transmitir mais pacotes combinados, o que de certa forma age como um aumento da taxa de partida, aliviando o buffer

O desempenho em relação à probabilidade de bloqueio por fator de utilização é mostrado na figura 12. Nela pode-se observar que a probabilidade de bloqueio para o sistema XOR é muito menor do que para o sistema MM1. Este resultado está coerente com o mostrado anteriormente sobre o número médio de pacotes no sistema e evidencia que a fila XOR requer um tamanho de buffer menor que da fila MM1 para resultar em um mesmo desempenho. Para um fator de utilização de aproximadamente 1, por exemplo, o desempenho da fila XOR com buffer de tamanho 5 é muito próximo ao da fila MM1 com buffer de tamanho 50.

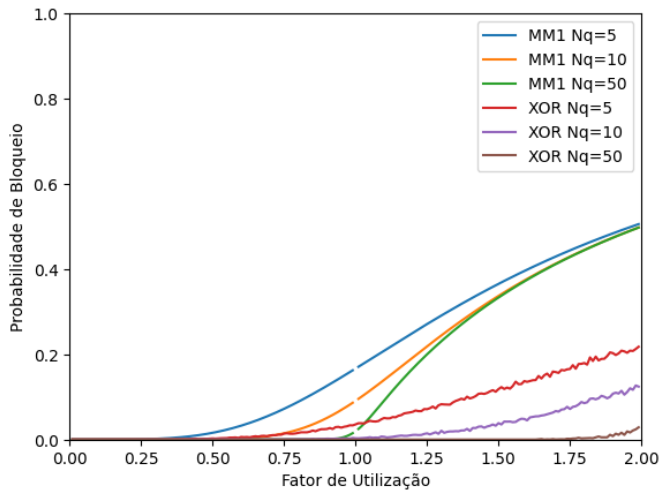


Figura 12: Comparação da probabilidade de bloqueio por fator de utilização para tamanhos de buffer de 5, 10 e 50.

#### IV. CONCLUSÃO

Este estudo demonstrou que o uso de operações XOR no gerenciamento de fila de um nó relay em uma rede de comunicação cooperativa pode trazer benefícios significativos em termos de eficiência de transmissão e redução de sobrecarga da fila.

Os resultados indicam que a operação XOR no relay pode ser uma estratégia eficaz para melhorar a eficiência da transmissão e reduzir a sobrecarga da fila. No entanto, a implementação prática desta estratégia deve considerar a complexidade adicional introduzida pela necessidade das coordenações extras para garantir que o receptor recupere as mensagens originais.

Futuras pesquisas podem explorar otimizações adicionais e a aplicação desta estratégia em diferentes tipos de redes.

#### REFERÊNCIAS

- [1] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, "Coopmac: A cooperative mac for wireless lans," *IEEE journal on selected areas in Communications*, vol. 25, no. 2, pp. 340–354, 2007.
- [2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.
- [3] M. Uysal, *Cooperative communications for improved wireless network transmission: framework for virtual antenna array applications: framework for virtual antenna array applications*. IGI Global, 2009.
- [4] E. C. M. Ishikawa *et al.*, "Um modelo computacional para o funcionamento da assinatura digital no sistema de informatização processual," 2003.
- [5] R. R. d. Oliveira *et al.*, "Metodologia de reconfiguração de hardware utilizando o sinal de tv digital," 2015.
- [6] D. G. Kendall, "Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain," *The Annals of Mathematical Statistics*, vol. 24, no. 3, pp. 338 – 354, 1953.