

3

Simulação de Redes *Manhattan Street* com Roteamento por Deflexão de Pacotes

Ao longo deste capítulo, serão apresentados alguns conceitos básicos sobre simulação de redes; uma descrição do *software* que foi utilizado para realizar as simulações; uma comparação entre os resultados obtidos através da simulação e os obtidos através da metodologia analítica desenvolvida no capítulo anterior; uma introdução sobre os protocolos UDP e TCP; os resultados das simulações; e uma análise do efeito da latência na vazão do protocolo TCP.

3.1. Introdução

Sempre é válido estudar redes existentes e futuras, de modo que se possa melhorar seu desempenho. No entanto, geralmente, se faz necessário utilizar um modelo para que se possa realizar o estudo, já que a experimentação pode ser muito cara ou até impossível de se realizar.

No caso de redes simples, pode-se, muitas vezes, utilizar um modelo matemático que represente a rede e, assim, descrevê-la utilizando métodos matemáticos tais como álgebra, cálculo, teoria da probabilidade, etc. Métodos analíticos para estudar o desempenho de filas têm sido utilizados há anos. No entanto, conforme a topologia e os protocolos utilizados na rede se tornam mais complexos, os modelos analíticos se tornam também muito complexos e inadequados. Além disso, modelos matemáticos apresentam algumas desvantagens. Apenas resultados estacionários podem ser analisados e é difícil obter medidas de desempenho que não sejam valores médios [28].

Devido a estas desvantagens e ao aumento do poder de processamento dos computadores atuais, a utilização de simuladores para a análise de desempenho de redes tem aumentado consideravelmente. Simulações podem ser aplicadas para a análise de redes de praticamente qualquer nível de complexidade, o fator limitante neste caso será a capacidade computacional disponível.

3.1.1. Objetivos de Simulação de Redes

A seguir serão apresentadas algumas vantagens de se utilizar simulação no projeto e na análise do desempenho de redes [28]:

- Avaliação do impacto no sistema como um todo, ao realizarem-se alterações locais;
- Podem-se realizar estudos que venham a melhorar o desempenho da rede com custo relativamente baixo;
- Garantir que o objetivo da rede será alcançado, antes de ser feita a compra ou instalação de qualquer equipamento;
- Identificação de gargalos antes da implementação do sistema.

A simulação pode responder uma série de perguntas, como por exemplo:

- Como será o comportamento da rede quando o tráfego aumentar?
- Quantos enlaces e portas nos *switches*/roteadores são necessários? Qual deve ser a velocidade destas portas? Que tamanho as filas devem ter para que a taxa de pacotes perdidos seja aceitável?
- Qual será o impacto no caso de falha em um enlace?
- Qual protocolo otimizaria o desempenho da rede?

A simulação pode fornecer uma série de parâmetros que são normalmente utilizados na análise do desempenho da rede, como por exemplo [28]:

- Vazão;
- Latência;
- Número médio de pacotes nas filas de acesso a rede;
- Tempo de espera nas filas de acesso a rede;
- Carga nos enlaces.

3.1.2. Desenvolvendo Modelos de Simulação Válidos e Confiáveis

O primeiro objetivo de toda simulação deve ser que o modelo simulado represente a rede sob estudo da melhor forma possível. Assim, as conclusões

obtidas pelas simulações devem ser bastante similares aos resultados obtidos na rede real. Se este objetivo for alcançado, pode-se dizer então que o modelo é válido.

Outro objetivo, não menos importante, diz respeito à confiabilidade dos resultados obtidos. É muito importante que os resultados obtidos de um modelo sejam confiáveis, pois do contrário não seria útil para o processo decisório, mesmo se o modelo for válido. Em geral o que torna os resultados de uma simulação confiáveis não é o grau de detalhamento do modelo e sim o número de vezes que o modelo é simulado.

Por exemplo, supondo que deseja-se simular o tempo de espera em uma fila de banco. Cria-se então um modelo extremamente simples, onde para qualquer entrada o tempo de espera seja igual a 10 minutos. O modelo é então simulado e o resultado obtido para qualquer situação é igual a 10 minutos. Pode-se notar que o resultado é extremamente confiável, pois não importa no número de vezes que seja simulado, sempre dará o mesmo resultado com variância zero. No entanto o modelo não é válido. Supondo agora que um modelo altamente sofisticado seja criado. Simula-se então uma vez, e o resultado obtido é de 5 minutos. Simula-se novamente e o resultado obtido é de 15 minutos. Como os resultados obtidos foram diferentes, ainda é cedo para afirmar qual o tempo médio na fila pois com apenas dois resultados a variância ainda é alta. Assim apesar do modelo ser válido o resultado não é confiável.

A seguir, serão apresentadas idéias e técnicas importantes para decidir o nível de detalhamento apropriado para validar o modelo simulado e obter resultados confiáveis [28]:

- Definir no início do estudo os pontos a serem estudados e os parâmetros que serão utilizados para avaliar o desempenho. Devem-se considerar as restrições de tempo e recursos disponíveis;
- Reunir informações sobre a topologia e protocolos que serão utilizados na rede em estudo;
- Realizar uma comparação conceitual entre o modelo e a rede em estudo;
- Testar e depurar o programa. Uma ferramenta gráfica que indique a trajetória dos pacotes pode facilitar este trabalho;

- Comparar resultados obtidos através da simulação com resultados empíricos. Como muitas vezes a rede simulada ainda não existe, pode-se tentar validar a simulação comparando seus resultados com resultados apresentados na literatura e/ou com resultados obtidos através de modelos matemáticos válidos;
- Realizar simulações durante um intervalo de tempo que garanta a confiabilidade nos resultados.

3.1.3. **Softwares para Simulação de Redes**

Uma das maiores dificuldades de se realizar simulações consiste em transformar o modelo da rede em estudo em um programa de computador. Existem, no entanto, programas disponíveis que já fornecem uma base com os principais padrões e protocolos. Assim, muitas vezes, são necessárias apenas modificações de pequena escala para que o programa simule a rede em estudo.

No presente trabalho utilizou-se o *software* de simulação NS-2 [29]. O NS-2 é um *software* de simulação amplamente utilizado em pesquisas acadêmicas, principalmente por ter seu código-fonte aberto e de distribuição gratuita. O NS-2 é um simulador de eventos discretos e orientado a objetos. Este *software* é ideal para simulação de protocolos que constituem a Internet, ou seja, redes que utilizam IP, TCP, UDP, protocolos de roteamento e *multicast*, etc.

A biblioteca de protocolos e mecanismos implementados no NS-2 é bastante vasta, abrangendo implementação dos protocolos TCP, UDP e IP, além de disciplinas de serviços, como WFQ (*Weight Fair Queueing*), protocolos para redes móveis, como o IP móvel, tecnologias para redes sem fio locais e de longa distância, como 802.11, *BlueTooth*, GPRS (*General Packet Radio Service*) e *DiffServ* (*Differentiated Services*).

Existem também bibliotecas de funções para a geração de alguns tipos de tráfego, tais como: CBR (*Constant Bit Rate*) utilizado para simular tráfego constante e voz, ON-OFF para tráfego em rajada e voz comprimida, FTP (*File Transfer Protocol*) para simular tráfego de transferência de arquivos e VBR (*Variable Bit Rate*) para tráfego com taxa de dados variável.

O NS-2 não fornece estatísticas de simulação de modo automático. Ele possui dois tipos de arquivos de saída, um para ser visualizado com o *NAM* (*Network Animator*) em forma de animação e outro que contém todos os eventos da simulação em ordem cronológica. Pode-se então, extrair as estatísticas da simulação com base nos eventos listados neste arquivo.

3.2. Simulações

Para realizar as simulações desejadas, alterou-se o código fonte do NS-2 de modo que este pudesse simular a rede em estudo.

O NS-2 foi desenvolvido para simular redes *store and forward*, ou seja, ao chegar em um nó intermediário da rede o pacote é armazenado em uma fila (mesmo que a rede esteja totalmente vazia), seus endereços origem e destino são lidos, sua melhor rota é calculada e, então, o pacote é encaminhado para a saída mais adequada.

De modo a permitir que o NS-2 simulasse redes *cut-through*, sem filas nos nós, alterou-se no código fonte do NS-2 o atraso que cada nó insere na transmissão dos pacotes. Desta forma, os nós passaram a não introduzir retardo no tempo de propagação dos pacotes. O código original do NS-2 não possui o roteamento por deflexão, o roteamento é feito pelo OSPF (*Open Shortest Path First*), assim foi necessário alterar no código fonte o mecanismo de roteamento.

Como o NS-2 é um simulador de eventos, os eventos são agendados em uma lista ordenada pelo tempo. Conforme o NS-2 vai realizando a simulação, vai retirando os eventos da lista de eventos e agendando novos eventos resultantes dos que foram retirados da lista. Para o NS-2, não existe tempo contínuo, o seu relógio vai saltando de acordo com os tempos dos eventos da lista.

Como os nós simulados no presente trabalho podem inserir novos pacotes desde que não interfiram com os pacotes que já estão na rede, fez-se necessário alterar o mecanismo de agendamento de eventos do NS-2. Foi criada uma função que inseria vários eventos em intervalos de tempo muito menores que a duração de um pacote. Estes novos eventos “avisam” a um determinado nó se ele pode inserir um novo pacote na rede. Como este mecanismo insere muitos eventos, o

tempo de simulação se tornou muito elevado e cada simulação realizada no presente trabalho levou alguns dias para ser realizada.

Cada nó da rede foi subdividido em módulos e cada um destes módulos apresenta uma funcionalidade, como pode ser observado na Figura 3.1. A função do “leitor de cabeçalhos de RF” é de informar ao controle os endereços de origem e destino dos pacotes que atingem o nó. Utilizando estas informações o “controle” configura a matriz de chaveamento totalmente óptica e libera ou bloqueia a fila de inserção de pacotes na rede. Deste modo, os pacotes são roteados e inseridos corretamente. Este procedimento corresponde ao método de roteamento dos nós 2x2 OPS [18], [19].

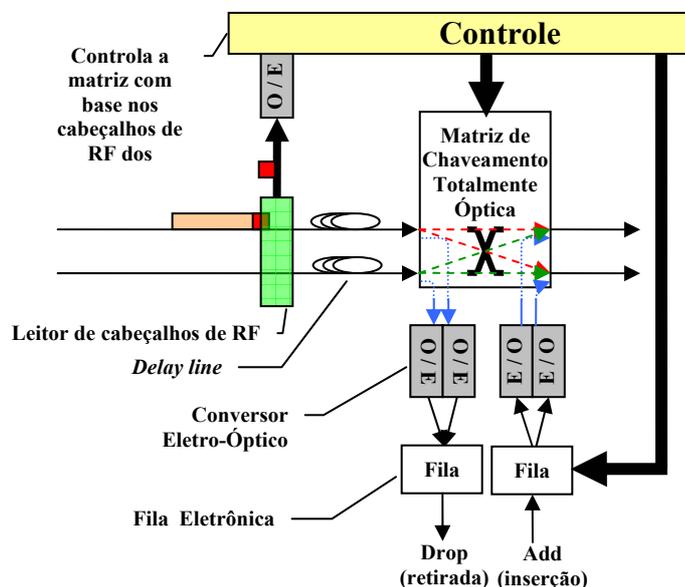


Figura 3.1: Detalhes dos nós da rede.

A Figura 3.2 foi obtida a partir de uma das simulações. Os círculos representam os nós, as linhas pretas os enlaces e as linhas e pontos coloridos os pacotes. Pode-se notar que existem 9 grupos de nós, cada um desses grupos representa um único nó. A funcionalidade de cada nó do grupo está sendo indicada na figura.

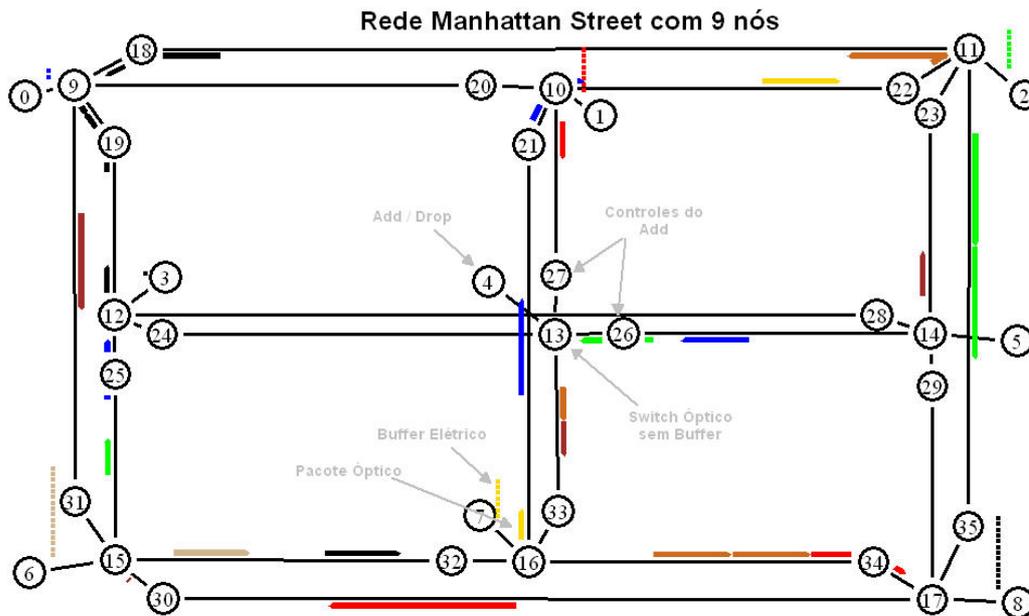


Figura 3.2: Exemplo da rede simulada.

A rede foi simulada com todos os enlaces com 10km de comprimento ($delay = 50\mu s$), operando a uma taxa de 1Gbps. Cada nó da rede gera tráfego para todos os outros, sendo que cada um destes tráfegos possui taxas e instantes iniciais ligeiramente distintos. Variou-se o tamanho das filas nos nós de inserção para analisar possíveis melhorias no sistema.

Para garantir a confiabilidade nos resultados da simulação, foram realizadas diversas simulações estatisticamente independentes e utilizou-se a média dos resultados obtidos. O número de amostras necessárias foi calculado para obter um nível de confiança de 95% com erro menor do que 1% que foi calculado conforme descrito em [8].

3.2.1. Validação do Modelo

Como a rede em estudo ainda não existe, os resultados do modelo simulado foram comparados com os resultados obtidos através da metodologia analítica descrita no capítulo anterior.

Como a metodologia analítica foi desenvolvida a partir de trabalhos apresentados há alguns anos por autores consagrados na área, a concordância dos

resultados obtidos nas simulações com os obtidos pela metodologia analítica, indica que o modelo simulado é válido.

A seguir, serão apresentados os resultados das simulações e as comparações com os resultados da metodologia analítica.

Vazão da Rede x Carga Oferecida

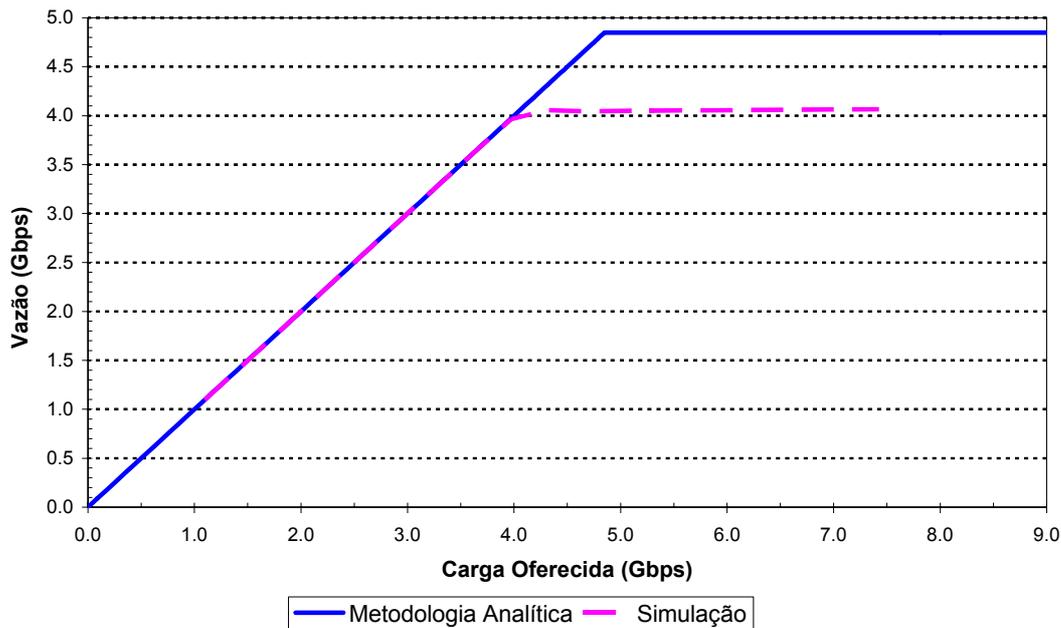


Figura 3.3: Comparação entre a vazão da simulação e da metodologia analítica vs. a carga oferecida, para uma rede MS com 9 nós.

A Figura 3.3 e a Figura 3.4 comparam, respectivamente os resultados obtidos pela metodologia analítica e pela simulação da vazão e número médio de *hops* percorrido pelos pacotes,. Pode-se notar a vazão obtida que pela metodologia analítica é maior do que a obtida pela simulação. Este resultado era esperado, pois em geral modelos matemáticos simplificam o problema, ou seja, o modelo analítico tem menos restrições que a rede real, conseqüentemente, mais pacotes podem trafegar pela rede, aumentando assim a vazão.

Com mais pacotes na rede, o número médio de deflexões aumenta, fazendo com que o número médio de *hops* também aumente, gerando assim uma grande diferença entre os resultados obtidos pela metodologia analítica e pela simulação da Figura 3.4.

Impondo uma restrição de vazão em torno de 4Gbps (o ponto onde a simulação satura na Figura 3.3) para a metodologia analítica, podemos notar que a resposta desta seria bem parecida com o resultado simulado.

Número Médio de Hops x Carga no Enlace

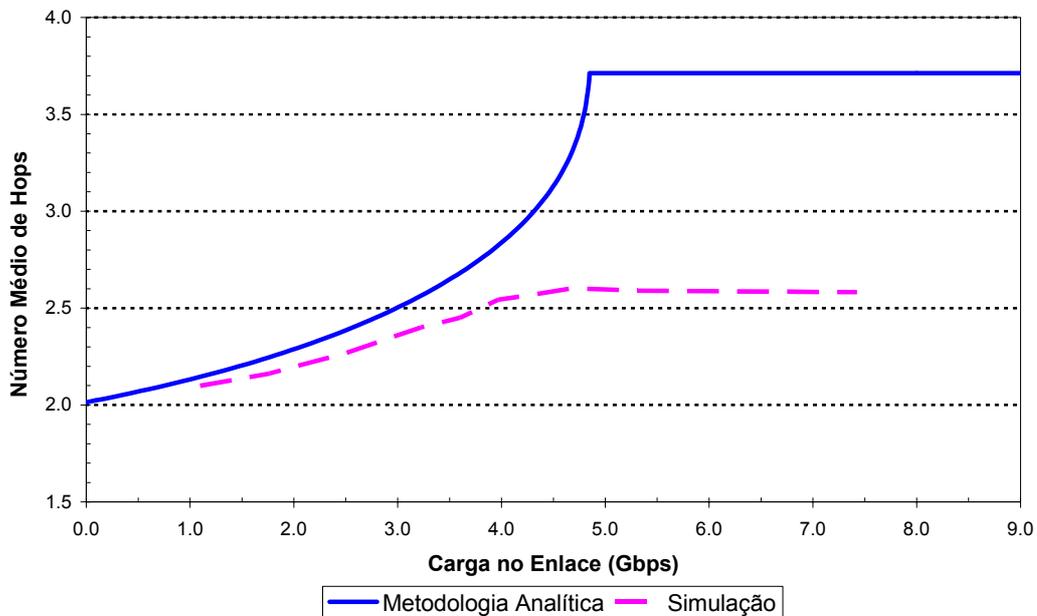


Figura 3.4: Comparação entre o número médio de hops da simulação e da metodologia analítica vs. a carga oferecida, para uma rede MS com 9 nós.

Com base na Figura 3.5 e na Figura 3.6 pode-se concluir que, como os resultados foram próximos, os modelos são válidos.

Vazão da Rede x Carga Oferecida

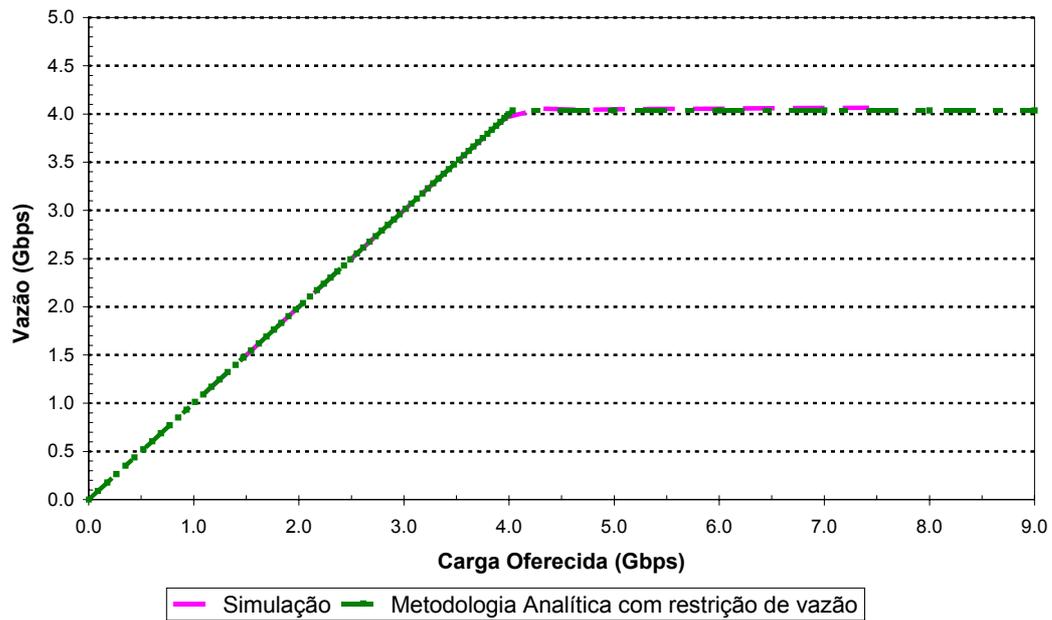


Figura 3.5: Comparação entre a vazão da simulação e da metodologia analítica com restrição vs. a carga oferecida, para uma rede MS com 9 nós.

Número Médio de Hops x Carga no Enlace

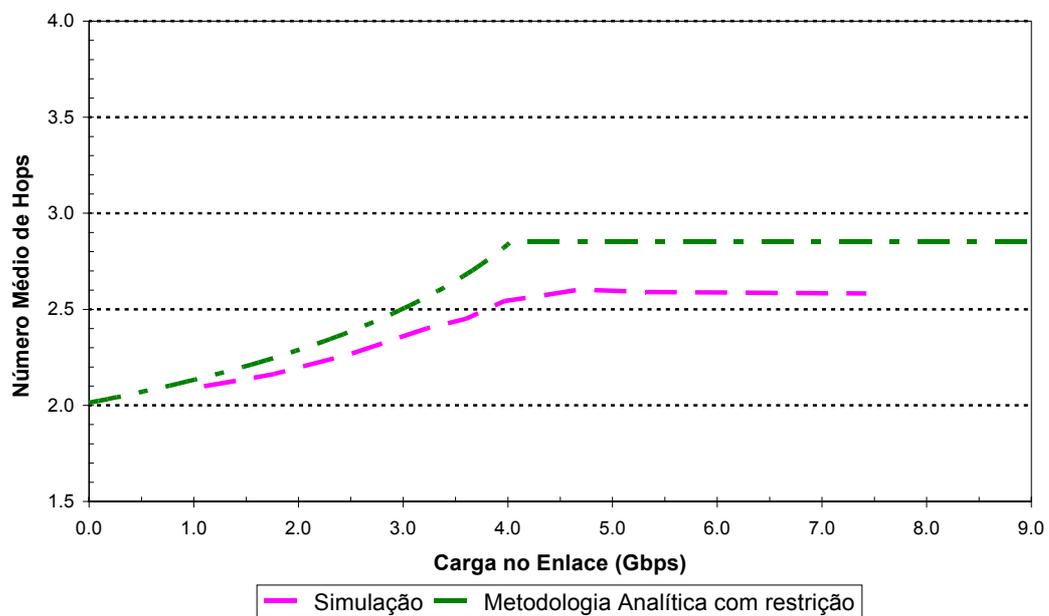


Figura 3.6: Comparação entre o número médio de hops da simulação e da metodologia analítica com restrição vs. a carga oferecida, para uma rede MS com 9 nós.

3.3. UDP e TCP

De modo a tornar claras algumas funcionalidades do UDP e do TCP, serão introduzidas, nesta sessão, algumas características destes protocolos [30].

O UDP (*User Datagram Protocol*) é um protocolo de camada 4, não orientado a conexão, que permite que a aplicação encapsule o pacote IP e o envie, sem ter que estabelecer uma conexão.

O TCP (*Transmission Control Protocol*) é um protocolo de camada 4 que fornece serviço fim-a-fim confiável, orientado a conexão e bidirecional. Este protocolo possui funcionalidades para controle de erro e prevenção de congestionamento.

Congestionamentos ocorrem em redes de pacotes quando a carga inserida na rede é maior do que esta pode suportar. Normalmente os protocolos das camadas 1, 2 e 3 não possuem nenhuma funcionalidade que faça o controle de congestionamento da rede. O protocolo TCP realiza este controle, a sua taxa de transmissão é definida pelas condições da rede e por isso é dito “compreensivo”. Já o UDP não realiza nenhum tipo de controle de congestionamento, ou seja, este protocolo transmite sempre a uma taxa que é definida pela aplicação. Assim, este protocolo é dito “não compreensivo”.

O protocolo TCP utiliza alguns métodos para estimar o nível de congestionamento da rede. O controle de congestionamento do TCP é feito basicamente por quatro algoritmos: *slow start*, *congestion avoidance*, *fast recovery* e *fast retransmit*.

O *slow start* é um mecanismo que controla o comportamento transitório da conexão levando-a ao ponto de equilíbrio de forma rápida, porém sem causar congestionamento na rede. Este algoritmo utiliza uma janela de transmissão para controlar a taxa de transmissão de pacotes, iniciando a conexão com uma janela pequena e aumentando-a exponencialmente até que a capacidade máxima de rede seja atingida. Este mecanismo é utilizado tanto no início de novas conexões como no reinício, quando o congestionamento já existe.

O algoritmo *slow start* funciona da seguinte maneira:

- No início ou reinício da conexão, a janela do TCP (CWND - *Congestion Window*) recebe o valor de 1 pacote;

- A cada novo pacote de ACK recebido pelo transmissor, a CWND é incrementada de 1 pacote.

O processo de aumentar a CWND de um pacote a cada ACK recebido gera um crescimento exponencial da variável, já que, caso todos os pacotes da janela sejam transmitidos com sucesso, a CWND dobra de tamanho. A CWND vai aumentando de tamanho até que a capacidade máxima da rede é atingida, quando o transmissor transmite a uma taxa maior do que a rede pode suportar, ocorre um *time-out* e o algoritmo reinicia. O *slow start* permite que uma conexão iniciada ou reiniciada após um *time-out* atinja rapidamente seu ponto de equilíbrio. Outros algoritmos tratam da adaptação do transmissor às limitações de banda-passante da rede.

Como o *slow start* incrementa a janela de transmissão exponencialmente, em um determinado momento a taxa de inserção de pacotes na rede será maior do que a rede pode suportar. Neste ponto, o algoritmo de *Congestion Avoidance* é utilizado. Este algoritmo incrementa a CWND de uma forma mais conservadora, funcionando da seguinte maneira:

- Quando ocorre um *time-out* a CWND é reduzida para sua metade;
- A cada novo pacote de ACK recebido pelo transmissor, a CWND é incrementada de $1/CWND$.

Na prática, o *slow start* e o *congestion avoidance* são implementados em conjunto. Para isto, basta adicionar a variável SSTHRESH (*slow start threshold*) que funciona como um limite para o *slow start* e determina qual dos algoritmos está ativo. Os dois algoritmos em conjunto funcionam da seguinte maneira:

- Quando ocorre um *time-out* $SSTHRESH = CWND/2$;
- Se $CWND < SSTHRESH$, a janela é manipulada pelo *slow start*;
- Se $CWND > SSTHRESH$, a janela é manipulada pelo *congestion avoidance*.

O *slow start* inicia aumentando a janela até ocorrer um *time-out*. Neste momento o SSTHRESH passa a valer $CWND/2$. A partir daí, caso o limiar seja atingido, o *congestion avoidance* aumenta a janela lentamente tentando atingir uma taxa de transmissão maior. Em caso de perda de pacote, o SSTHRESH cai para a metade de seu valor.

O *Fast Retransmit* visa apressar a retransmissão de pacotes quando a rede se encontra em uma situação de congestionamento moderado. Como a rede IP não é orientada a conexão, para cada pacote que atinge um nó a sua rota é calculada. Deste modo, é possível que alguns pacotes (incluindo os pacotes de ACK) sejam recebidos fora da ordem em que foram enviados. Assim, o TCP pode não saber se um pacote foi perdido ou se simplesmente um pacote que foi transmitido depois chegou antes, ou seja, chegam ACKs duplicados. Assume-se que, no caso de pacotes enviados fora de ordem, serão recebidos no máximo dois ACKs duplicados até que o pacote que chegou fora de ordem seja processado e um novo ACK seja gerado. Logo, caso cheguem três ACKs duplicados o TCP assume que um pacote foi perdido, retransmitindo-o. Assim evita-se que se tenha que esperar o *time-out*, e a retransmissão é feita mais rapidamente.

O *fast recovery* é implementado em conjunto com o *fast retransmit*, para o caso de necessidade de retransmissão por recebimento de ACK duplicado.

As primeiras implementações do TCP não possuíam nenhum algoritmo de controle de congestionamento. Simplesmente, utilizava-se a janela do receptor para transmissão e um *time-out* de retransmissão. As retransmissões eram feitas de acordo com a estratégia “*go-back-n*”, isto é, todos os pacotes transmitidos após um pacote dado como perdido eram retransmitidos o que era bem menos eficiente. As primeiras implementações também não utilizavam nenhuma informação sobre o estado da rede.

O Tahoe foi à primeira implementação do TCP a incluir o controle de congestionamento. Esta implementação utiliza o *slow start*, *congestion avoidance* e *fast recovery*.

No presente trabalho o Tahoe, foi à implementação do TCP utilizada nas simulações que utilizam este protocolo.

3.4. Resultados

Nesta sessão serão apresentados os resultados das simulações realizadas. Para analisar o desempenho da rede, foram realizadas simulações com os protocolos UDP e TCP, com sinalização na camada física.

Diversos cenários foram simulados variando-se a carga oferecida, tamanho das filas elétricas; utilizando nós com ou sem *buffer* óptico.

3.4.1. Simulação MS com Nove Nós sem *Buffer* Óptico Utilizando UDP e Sinalização na Camada Física

Nesta sessão serão apresentados os resultados das simulações de redes MS com 9 nós utilizando o protocolo UDP e tráfego CBR. Em todas as simulações utilizaram-se *links* de 1Gbps com 10km de comprimento (atraso de 50µs). A rede simulada utiliza nós com a arquitetura ilustrada na Figura 3.1.

Foram realizadas diversas simulações, variando-se o tamanho das filas elétricas de inserção de pacotes na rede. Os tamanhos de filas utilizados foram 2KB, 10KB, 100KB, 200KB e 300KB. Nas simulações, os pacotes possuem tamanho fixo de 1625 bytes (1500bytes de *payload* + 125bytes de sinalização física). Todos os nós geram tráfego para todos os outros, com taxas e instantes iniciais ligeiramente distintos, de modo que a rede não fique sobrecarregada devido a uma coincidência nos instantes iniciais dos pacotes.

Diversas simulações do mesmo “cenário” foram realizadas de modo a aumentar a confiança nos resultados obtidos.

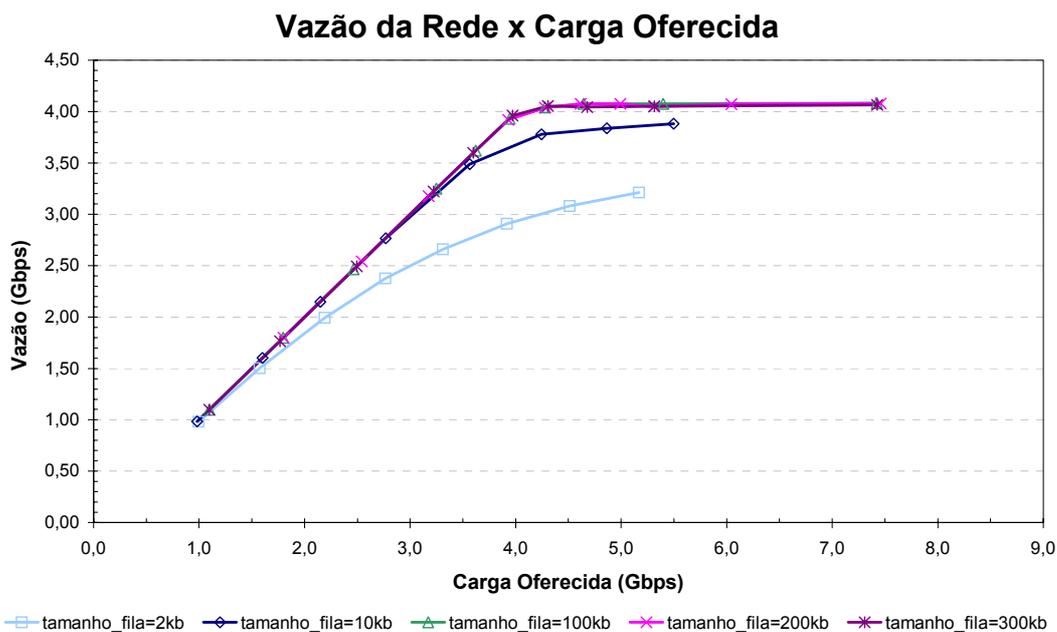


Figura 3.7: Vazão da rede em Gbps em função da carga oferecida.

A Figura 3.7 ilustra a vazão da rede em função da carga oferecida. Pode-se notar que a vazão da rede com filas de 2KB é bem inferior à das demais. No entanto, para filas maiores que 10KB, o ganho da vazão não é muito significativo.

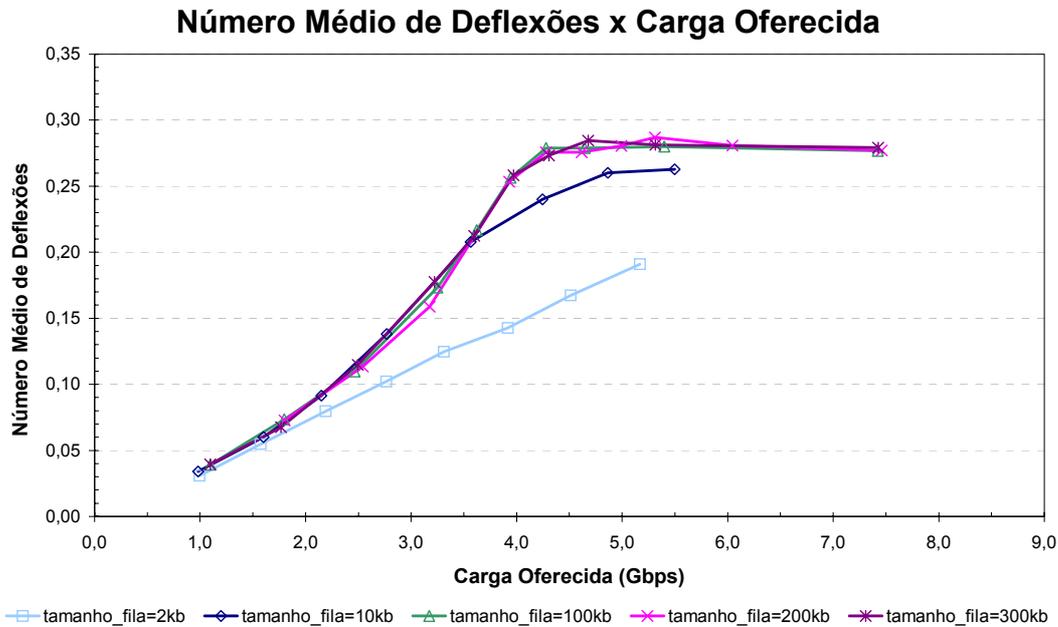


Figura 3.8: Média do número de deflexões sofridas por cada pacote em função da carga oferecida.

A Figura 3.8 ilustra o número médio de deflexões que cada pacote sofreu ao longo de sua trajetória na rede, em função da carga oferecida. Pode-se notar que, como era esperado, o número médio de deflexões aumenta conforme mais pacotes são inseridos na rede, pois quanto mais pacotes, maior a probabilidade de ocorrer uma disputa.

A Figura 3.9 ilustra a variância do número de deflexões que cada pacote sofreu ao longo de sua trajetória na rede, em função da carga oferecida. Pode-se notar que a variância do número de deflexões aumenta conforme mais pacotes são inseridos na rede, ou seja, quanto maior a carga na rede maior é a diferença nos tempos de propagação dos pacotes.

Como a média e variância do número de deflexões aumentam conforme mais pacotes são inseridos na rede, conseqüentemente a média e variância do

número de *hops* e do tempo de propagação também irão aumentar (Figura 3.10, Figura 3.11, Figura 3.12 e Figura 3.13).

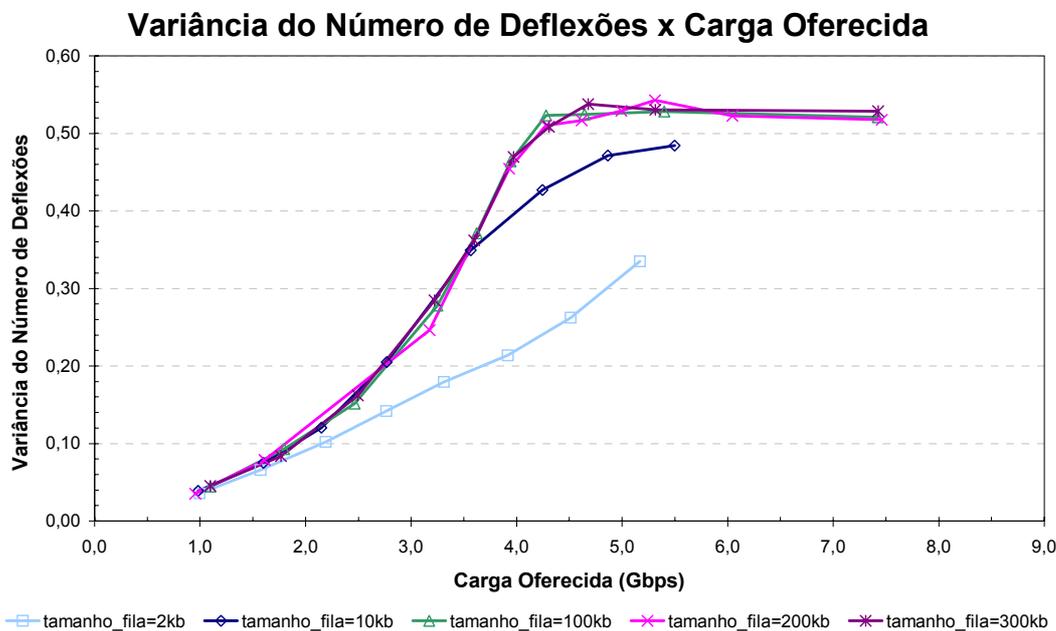


Figura 3.9: Variância do número de deflexões sofridas por cada pacote em função da carga oferecida.

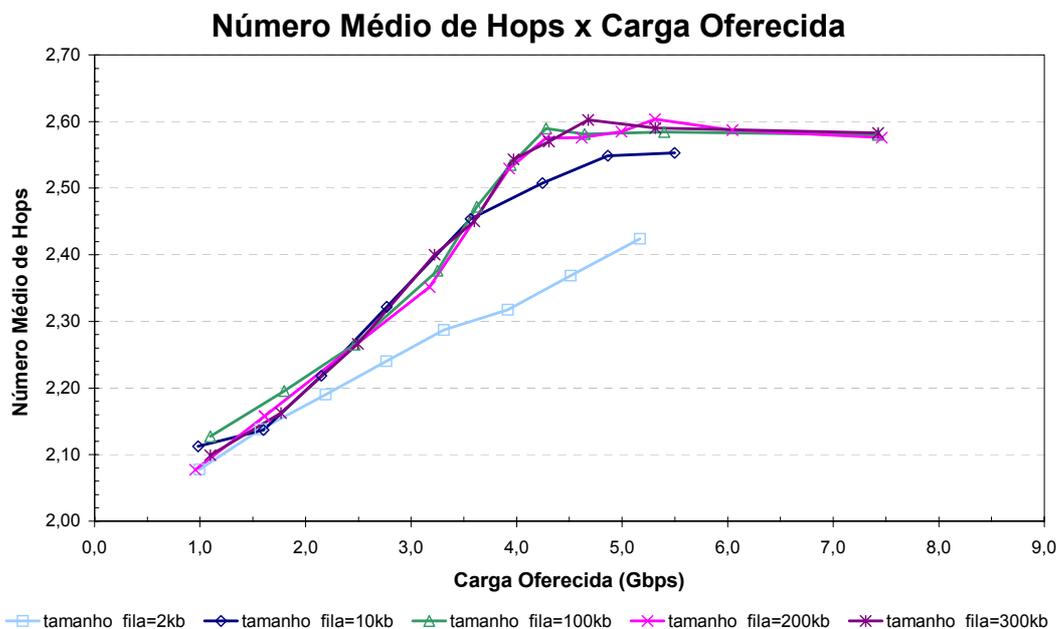


Figura 3.10: Média do número de hops percorridos por cada pacote em função da carga oferecida.

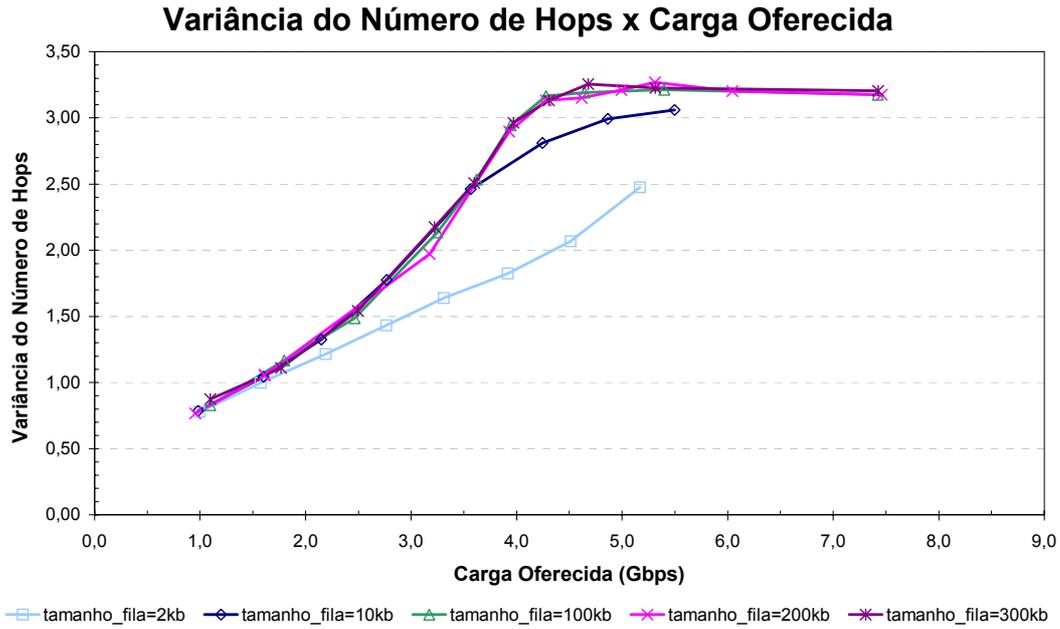


Figura 3.11: Variância do número de hops percorridos por cada pacote em função da carga oferecida.

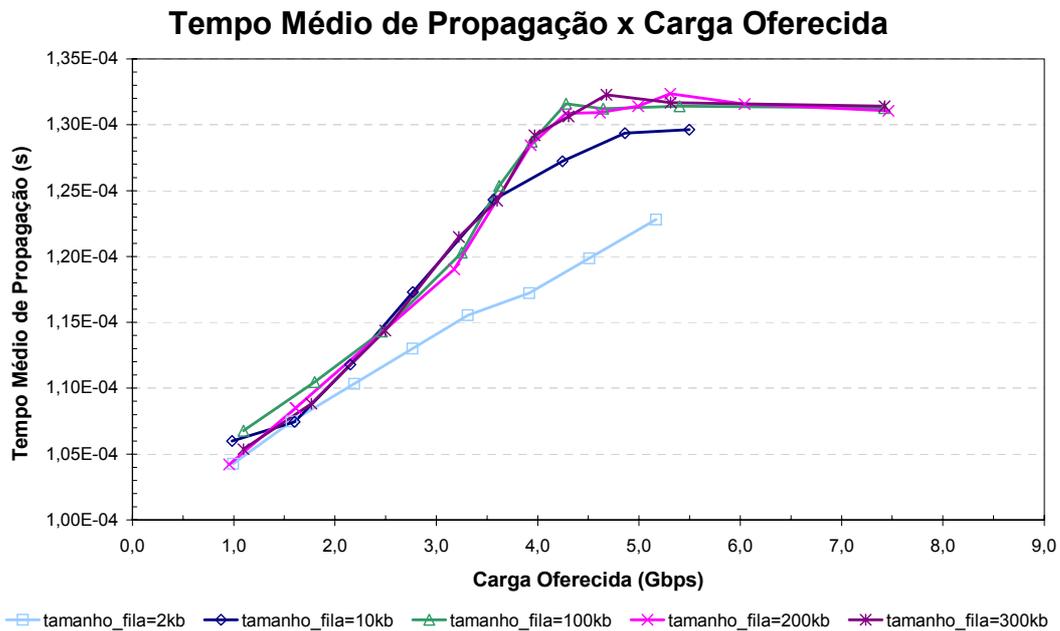


Figura 3.12: Tempo médio de propagação em função da carga oferecida.

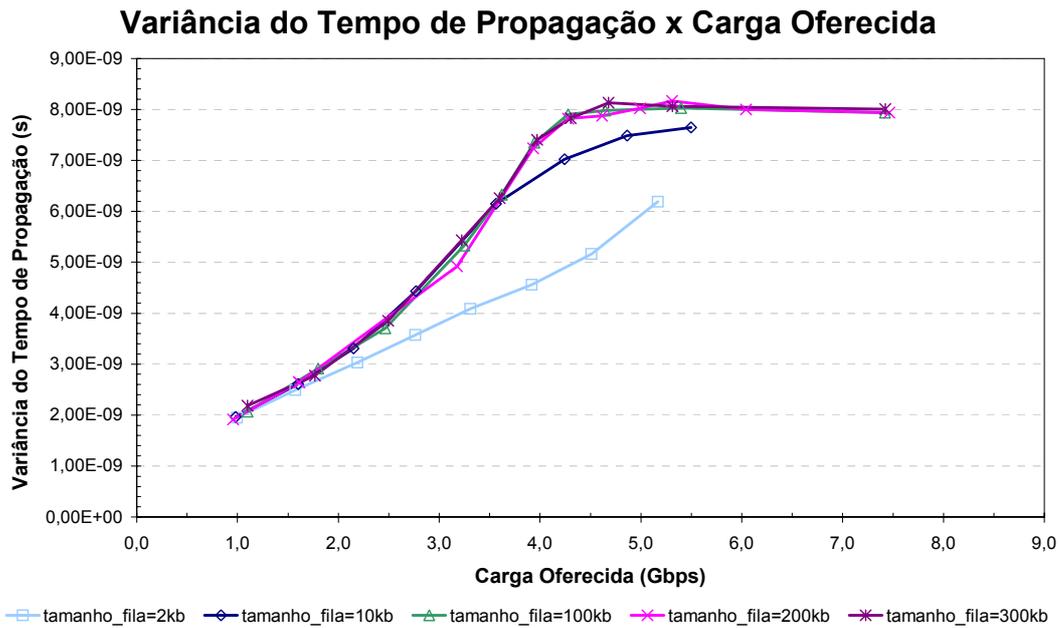


Figura 3.13: Variância do tempo de propagação em função da carga oferecida.

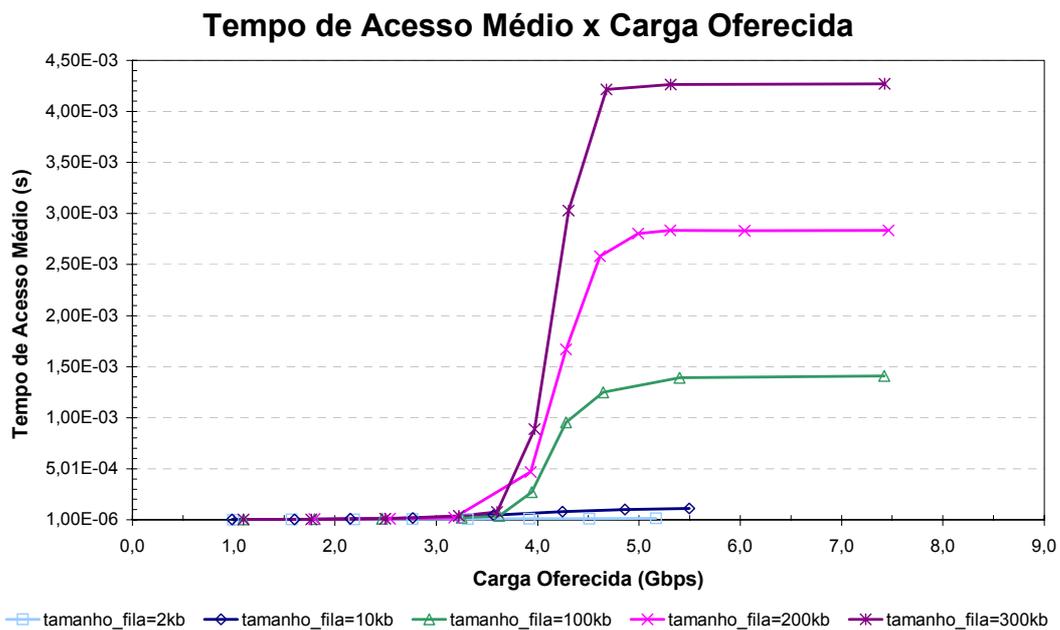


Figura 3.14: Tempo de acesso médio em função da carga oferecida.

A Figura 3.14 ilustra o tempo médio que cada pacote permanece na fila até ser inserido na rede em função da carga oferecida. Pode-se notar que o tempo de acesso médio da rede, antes da saturação (carga oferecida < 3,5 Gbps), é da ordem de μs para todas os tamanhos de fila simulados. Após a saturação

(carga oferecida > 3,5 Gbps), o tempo de acesso varia de acordo com o tamanho da fila.

A Figura 3.15 ilustra a variância do tempo de acesso à rede em função da carga oferecida. Pode-se notar que existe um ponto onde a rede fica mais instável (carga oferecida ≈ 4 Gbps), ou seja, um pouco antes da rede saturar, a variância tem um pico.

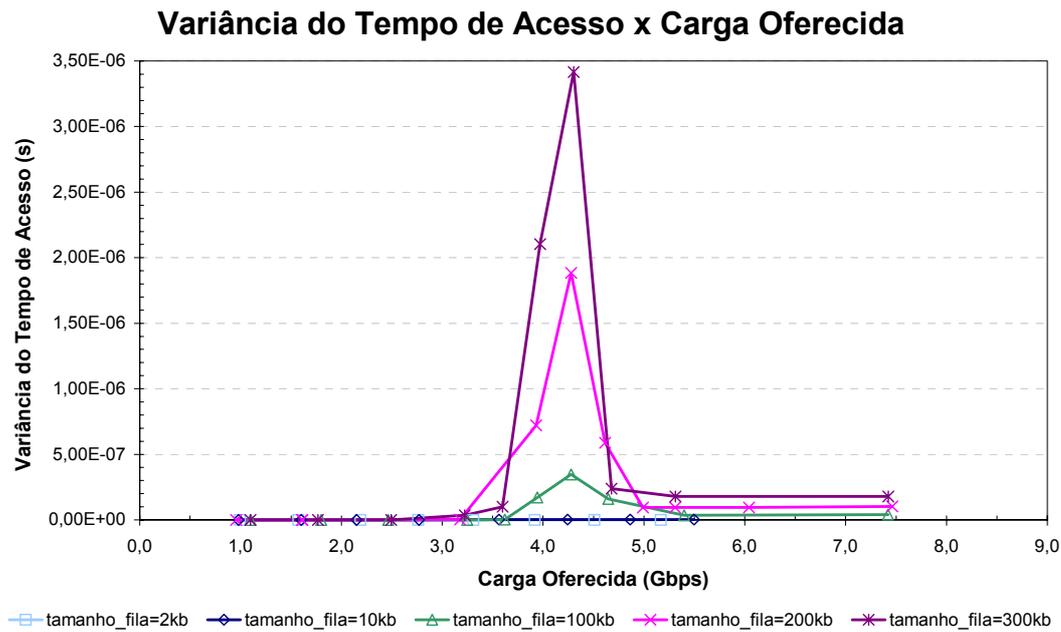


Figura 3.15: Variância do tempo de acesso em função da carga oferecida.

A Figura 3.16 ilustra a taxa de pacotes perdidos em função da carga oferecida. Nota-se que as redes cujos nós utilizam filas de 2KB apresentam desempenho inaceitável; redes cujos nós utilizam filas de 10KB apresentam desempenho aceitável até uma carga oferecida de 1,5 Gbps; redes cujos nós utilizam filas maiores do que 100KB apresentam desempenho aceitável até o ponto onde a rede satura.

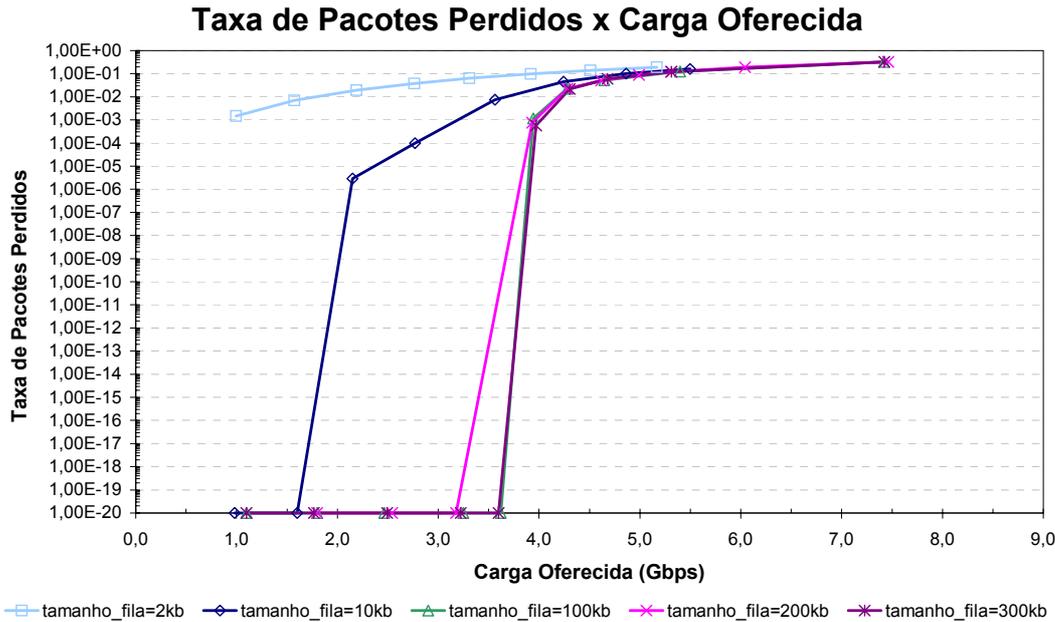


Figura 3.16: Taxa de pacotes perdidos em função da carga oferecida

3.4.2.

Simulação MS com Nove Nós com *Buffer* Óptico Utilizando UDP e Sinalização na Camada Física

Buffers ópticos consistem em um ou mais FDL (*Fiber Delay Line*), ligando a saída do *switch* com a sua entrada, como pode ser visto na Figura 3.17.

Redes *Manhattan Street* utilizando roteamento por deflexão foram projetadas para não precisar de *buffers* ópticos, já que em caso de disputa, ocorre uma deflexão. No entanto, a utilização de *buffers* ópticos melhora o desempenho da rede.

Nesta sessão são apresentados os resultados das simulações de redes MS com 9 nós utilizando o protocolo UDP e tráfego CBR (*Constant Bit Rate*). Em todas as simulações utilizaram-se enlaces de 1Gbps com 10km de comprimento (atraso de 50 μ s). A rede simulada utiliza nós com a arquitetura ilustrada na Figura 3.17.

Foram feitas diversas simulações, variando-se o comprimento e o número de FDL. Filas eletrônicas com 200KB foram utilizadas nos *Adds*, pois apresentaram bom desempenho nas simulações apresentadas anteriormente. Nas simulações, os pacotes possuem tamanho fixo de 1625 bytes (1500bytes de *payload* + 125bytes

de sinalização física). Todos os nós geram tráfego para todos os outros, com taxas e instantes iniciais ligeiramente distintos.

Diversas simulações do mesmo “cenário” foram realizadas de modo a aumentar a confiança nos resultados obtidos.

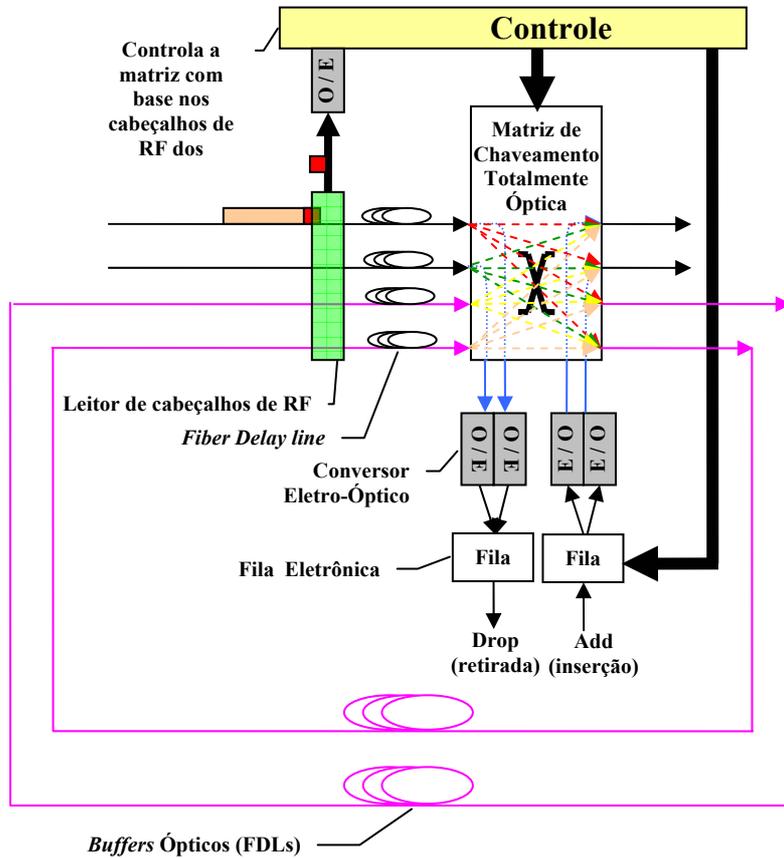


Figura 3.17: Detalhes dos nós da rede.

A Figura 3.18 ilustra a vazão da rede em função da carga oferecida. Pode-se notar que redes cujos nós possuem *buffers* ópticos (FDL) apresentam ganho significativo na vazão em relação à vazão de redes com nós sem *buffers* ópticos (comparar com a Figura 3.7). A maior vazão foi obtida nas redes cujos nós utilizavam 1 ou 2 FDLs de comprimento igual à duração de 1 pacote (aproximadamente 13 μ s). Verificou-se que redes com 1 FDL de comprimento igual à duração de 2 pacotes (aproximadamente 26 μ s) apresentam pior desempenho, mas mesmo assim melhor do que redes cujos nós não possuem *buffers* ópticos (FDLs).

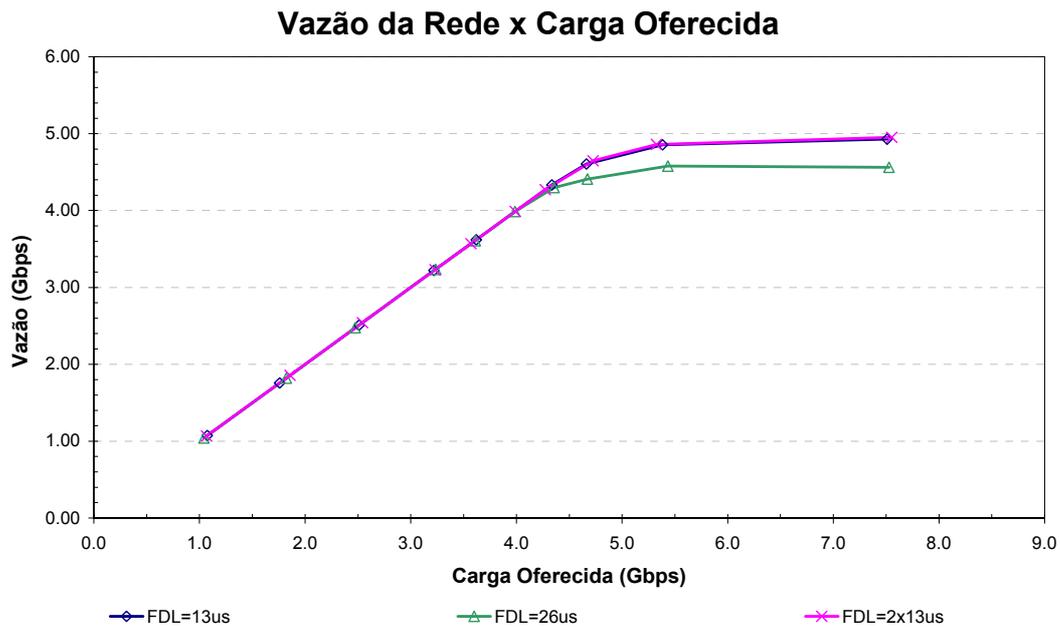


Figura 3.18: Vazão da rede em Gbps em função da carga oferecida.

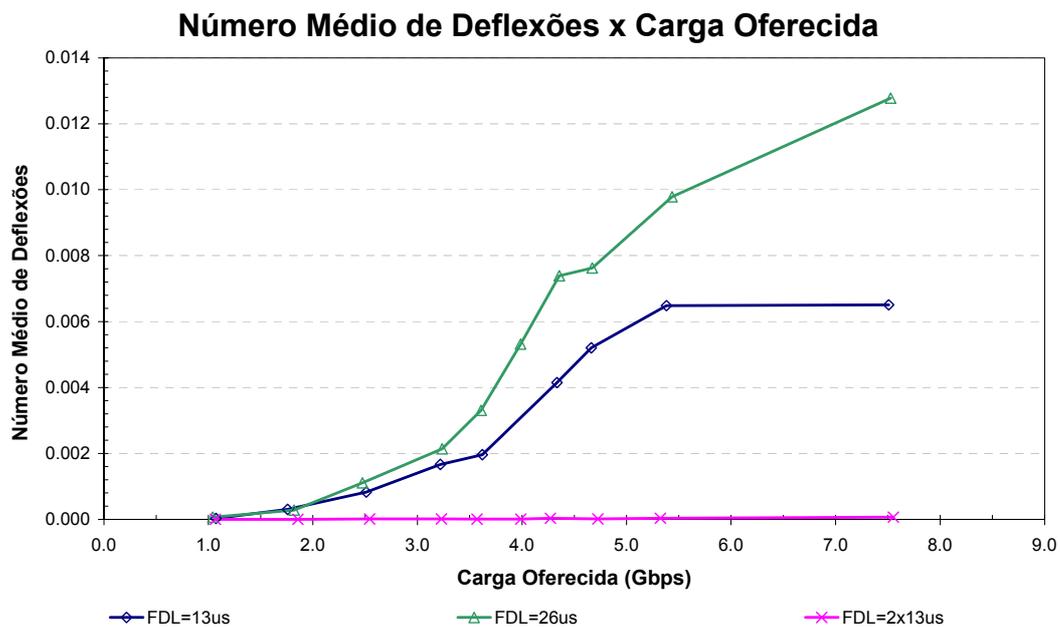


Figura 3.19: Média do número de deflexões sofridas por cada pacote em função da carga oferecida.

A Figura 3.19 ilustra o número médio de deflexões que cada pacote sofreu ao longo de sua trajetória na rede, em função da carga oferecida. Pode-se notar que o número médio de deflexões é muito pequeno em redes com *buffers* ópticos.

Para redes com dois *buffers* ópticos, o número médio de deflexões é praticamente zero.

A Figura 3.20 ilustra a variância do número de deflexões que cada pacote sofreu ao longo de sua trajetória na rede em função da carga oferecida. Pode-se notar que a variância do número de deflexões em redes cujos nós possuem *buffers* ópticos é quase nula já que o número de deflexões é bem reduzido. Assim, os tempos de propagação dos pacotes não variam muito.

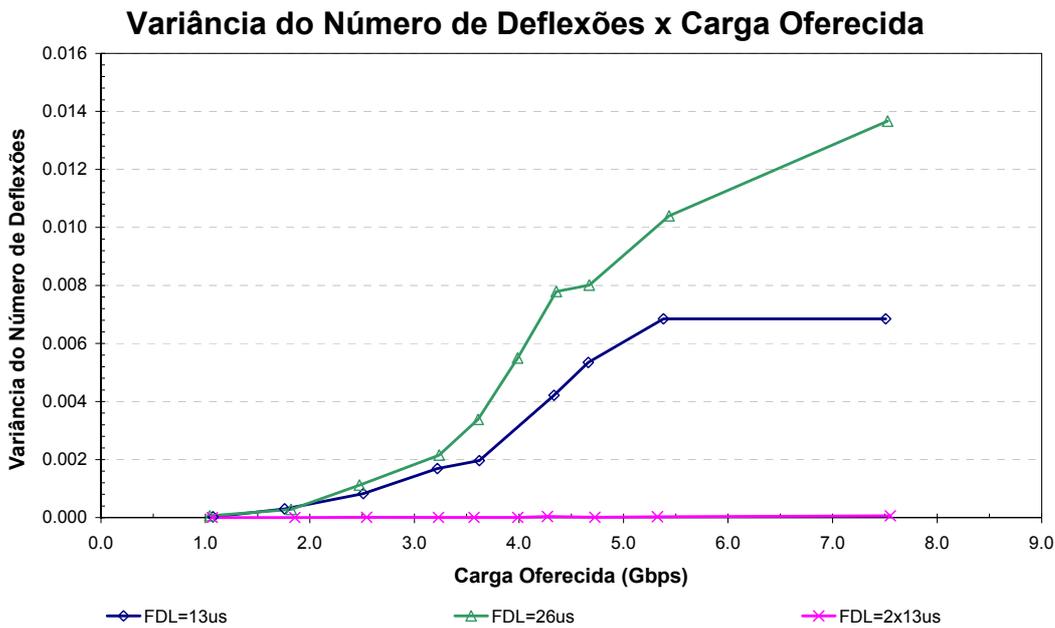


Figura 3.20: Variância do número de deflexões sofridas por cada pacote em função da carga oferecida.

Como a média e variância do número de deflexões são muito baixas em redes cujos nós utilizam *buffers* ópticos, a média e variância do número de *hops* também serão. No entanto, a média do tempo de propagação aumenta conforme a carga oferecida aumenta, pois mais pacotes passam a utilizar os *buffers* ópticos. Assim, dentre as redes com FDLs, a rede com pior desempenho (maior tempo de propagação) será a com FDLs de 26 μ s. As redes cujos nós utilizam 1 FDL de 13 μ s apresentaram os menores tempos de propagação, para carga oferecida inferior a 3,5Gbps. As redes cujos nós utilizam 2 FDL de 13 μ s apresentaram os menores tempos de propagação, para carga oferecida superior a 3,5Gbps. (Figura 3.21, Figura 3.22, Figura 3.23 e Figura 3.24).

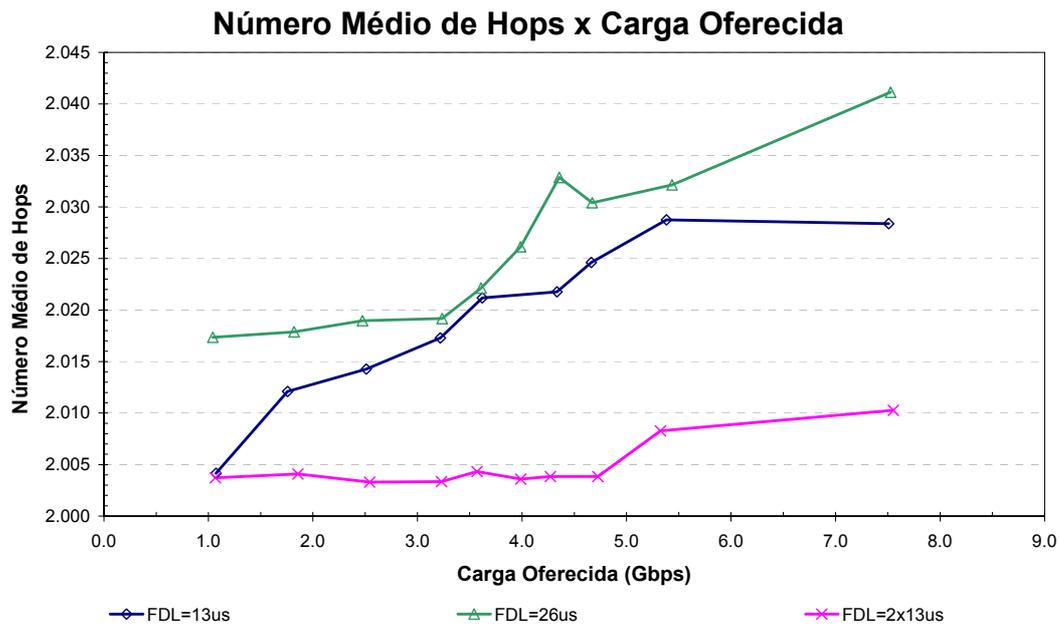


Figura 3.21: Média do número de *hops* percorridos por cada pacote em função da carga oferecida.

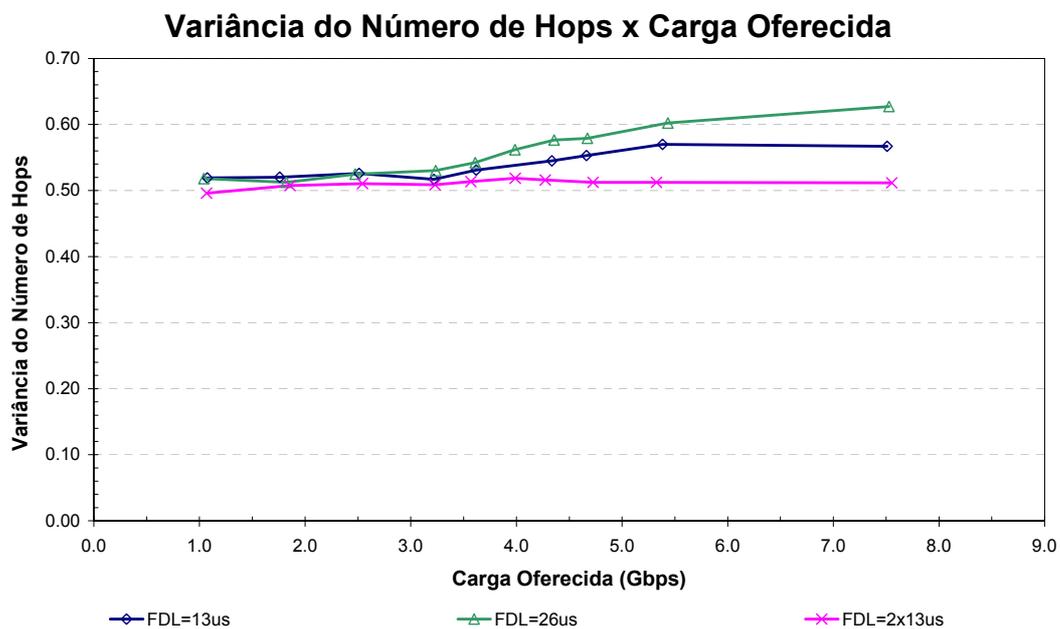


Figura 3.22: Variância do número de *hops* percorridos por cada pacote em função da carga oferecida.

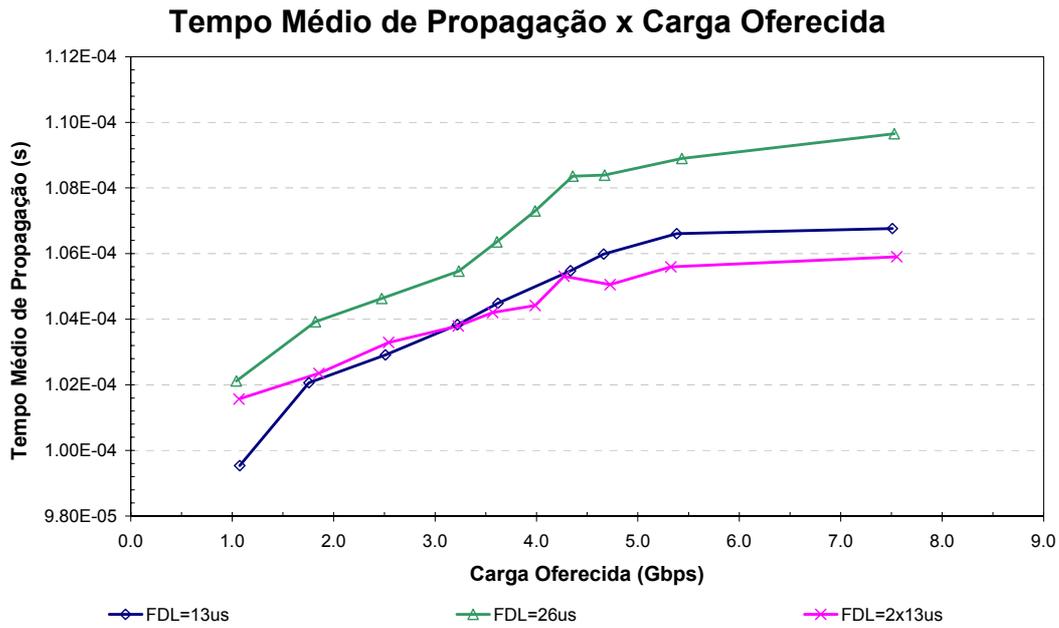


Figura 3.23: Tempo médio de propagação em função da carga oferecida.

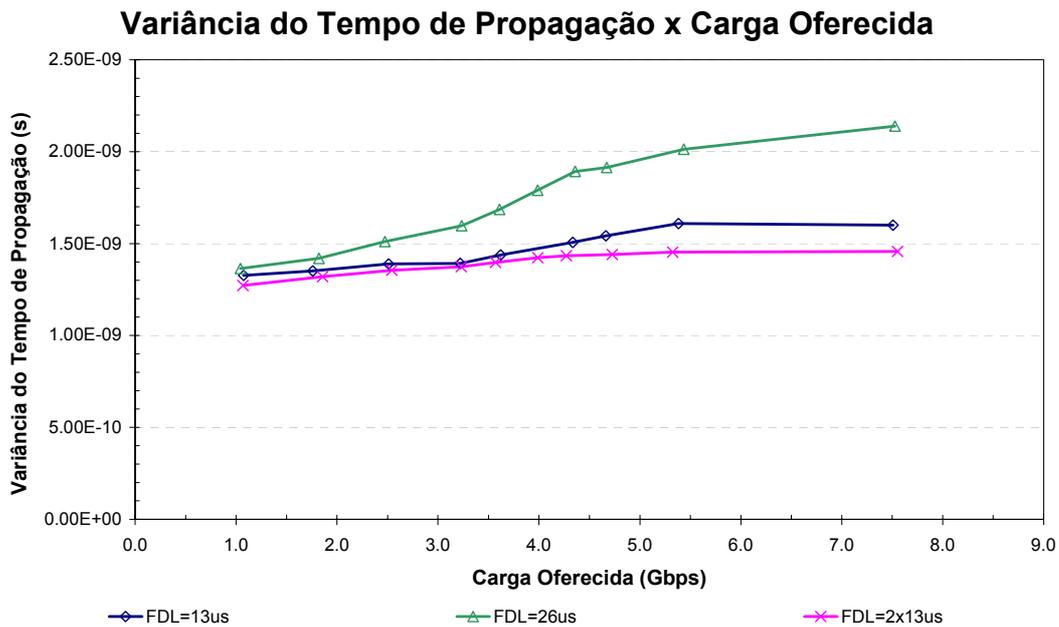


Figura 3.24: Variância do tempo de propagação em função da carga oferecida.

A Figura 3.25 ilustra o tempo médio que cada pacote permanece na fila até ser inserido na rede em função da carga oferecida. Pode-se notar que o tempo de acesso médio da rede, antes da saturação (carga oferecida < 4,5 Gbps), é da ordem de μ s para todas as filas simuladas. Após a saturação

(carga oferecida > 4,5 Gbps), o tempo de acesso varia de acordo com o tamanho da fila.

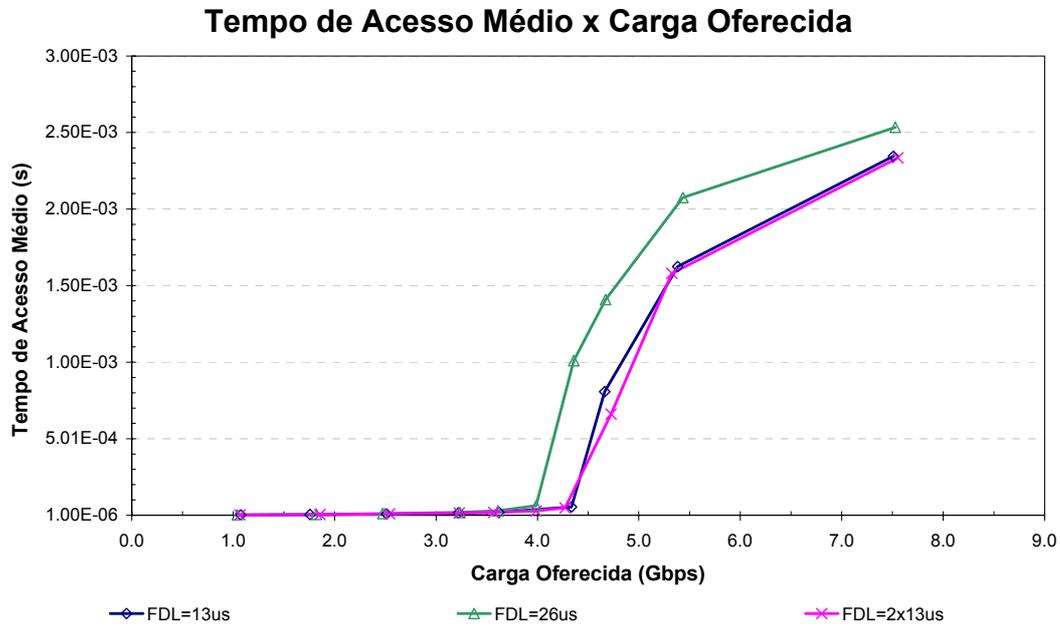


Figura 3.25: Tempo de acesso médio em função da carga oferecida.

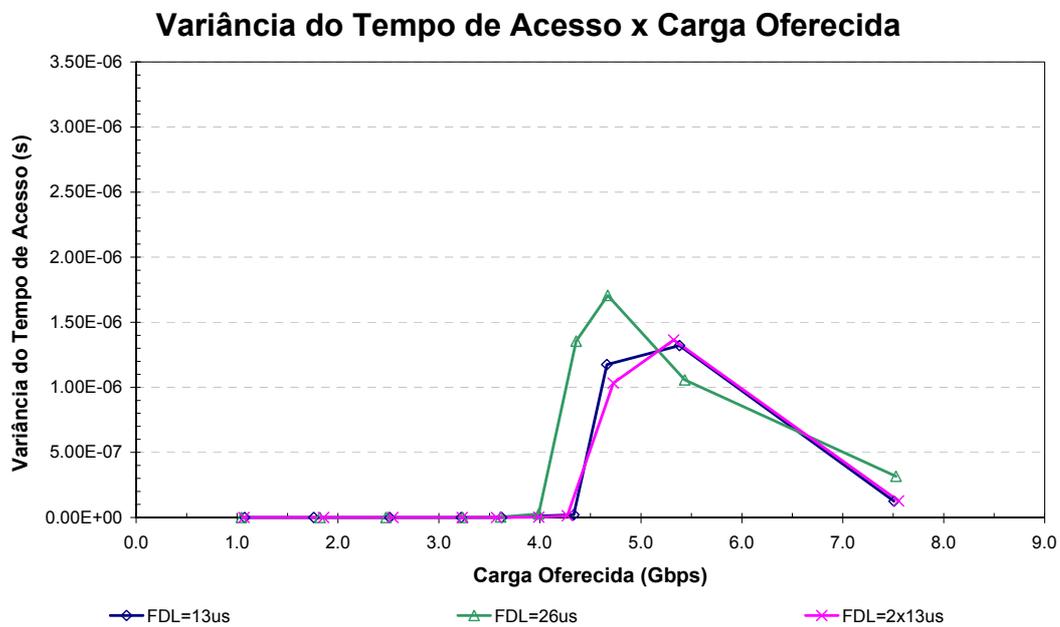


Figura 3.26: Variância do tempo de acesso em função da carga oferecida.

A Figura 3.26 ilustra a variância do tempo de acesso à rede em função da carga oferecida. Pode-se notar que existe um ponto onde a rede fica mais instável, um pouco antes da rede saturar a variância tem um pico.

A Figura 3.27 ilustra a taxa de pacotes perdidos em função da carga oferecida. Nota-se os melhores desempenhos são das redes cujos nós utilizam *buffers* ópticos, principalmente para nós com 1 FDL de 13 μ s e com 2 FDLs de 13 μ s.

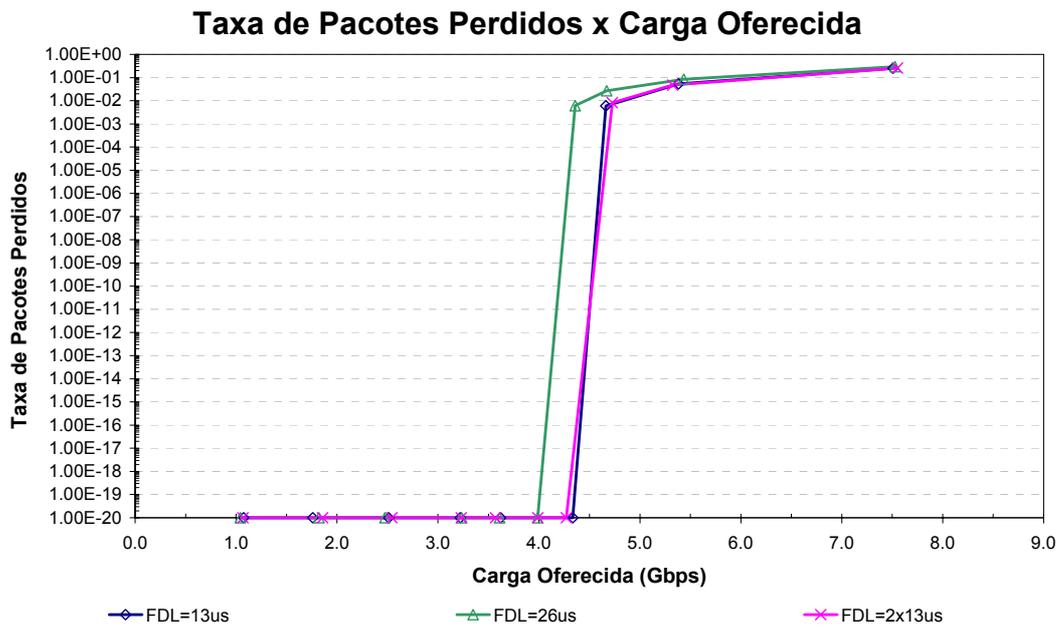


Figura 3.27: Taxa de pacotes perdidos em função da carga oferecida.

3.4.3. Simulação MS com Nove Nós sem *Buffer* Óptico Utilizando TCP e Sinalização na Camada Física

Nesta sessão serão apresentados os resultados da simulação de uma rede MS com 9 nós utilizando o protocolo TCP e tráfego CBR. Utilizam-se enlaces de 1Gbps com 10km de comprimento (atraso de 50 μ s). A rede simulada utiliza nós com a arquitetura ilustrada na Figura 3.1, onde as filas eletrônicas possuíam 200KB.

Nas simulações, os pacotes possuem tamanho fixo de 1625 bytes (1500bytes de *payload* + 125bytes de sinalização física) para os pacotes de dados e 40bytes para os pacotes de *acknowledgment*. Utilizou-se o TCP *window size* de 20

pacotes. Como as funcionalidades de *ack* seriam facilitadas entre nós adjacentes, este tráfego foi inicialmente desconsiderado nas simulações. Nos cálculos finais, como o percentual de tráfego adjacente é conhecido, foi possível estimar-se o tráfego total e acrescentá-lo aos resultados das simulações, sendo possível assim comparar os resultados do TCP com os do UDP.

Diversas simulações do mesmo “cenário” foram realizadas de modo a aumentar a confiança nos resultados obtidos.

A Figura 3.28 ilustra a vazão da rede em função da carga oferecida. Pode-se notar que a vazão da rede utilizando o protocolo TCP é inferior à vazão da rede utilizando o protocolo UDP, pois o protocolo TCP utiliza pacotes de *acknowledgment*, que ocupam a banda da rede. Além disso, o protocolo TCP espera os pacotes de *ack* para enviar novos pacotes e, em caso de congestionamento na rede, o protocolo TCP reduz o tamanho da janela de transmissão, diminuindo ainda mais a vazão.

PUC-Rio - Certificação Digital Nº 0310495/CA

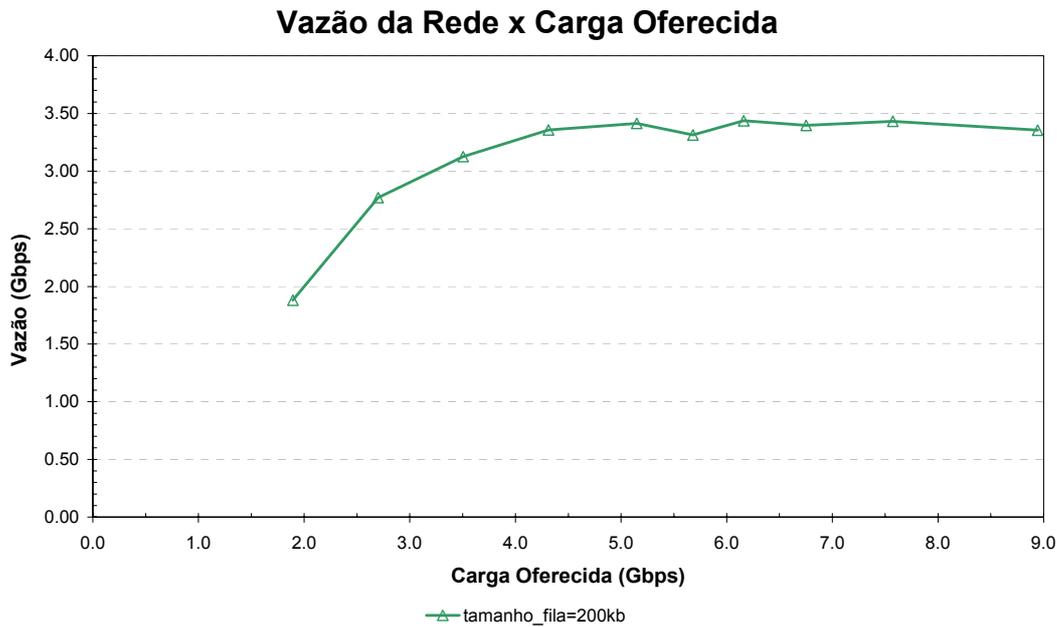


Figura 3.28: Vazão da rede em Gbps em função da carga oferecida.

A Figura 3.19 ilustra o número médio de deflexões que cada pacote sofreu ao longo de sua trajetória na rede, em função da carga oferecida. Pode-se notar que, o número médio de deflexões quando o protocolo TCP é utilizado, é elevado, pois este protocolo utiliza pacotes de *acknowledgment*, ou seja, existem mais

pacotes trafegando na rede quando é utilizado o TCP do que quando é utilizado o UDP. Assim a probabilidade de ocorrer uma deflexão é maior utilizando o TCP do que utilizando o UDP (comparar com a Figura 3.8).

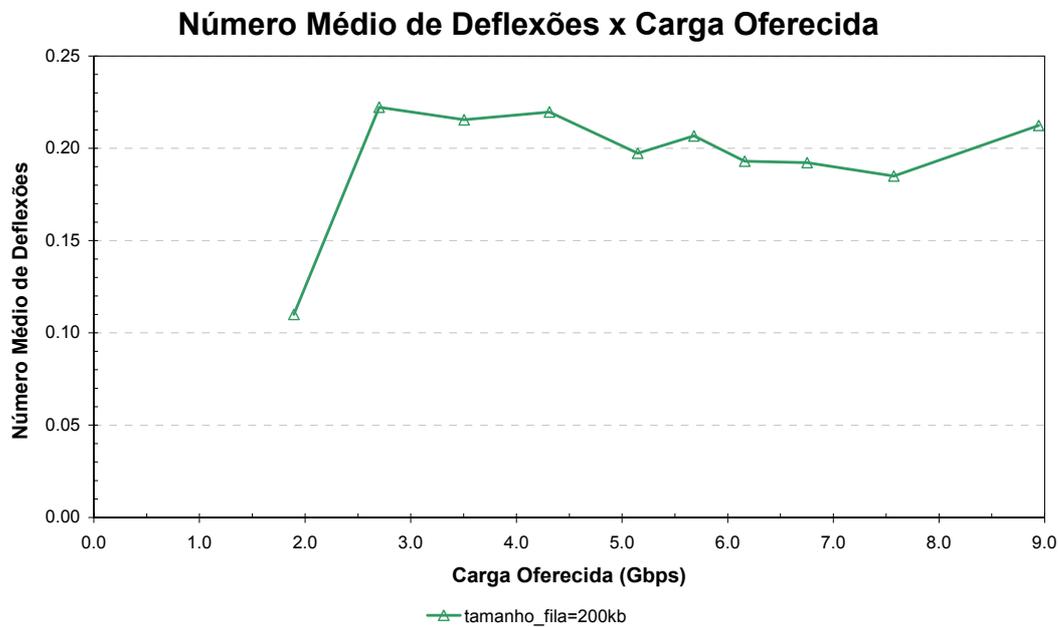


Figura 3.29: Média do número de deflexões sofridas por cada pacote em função da carga oferecida.

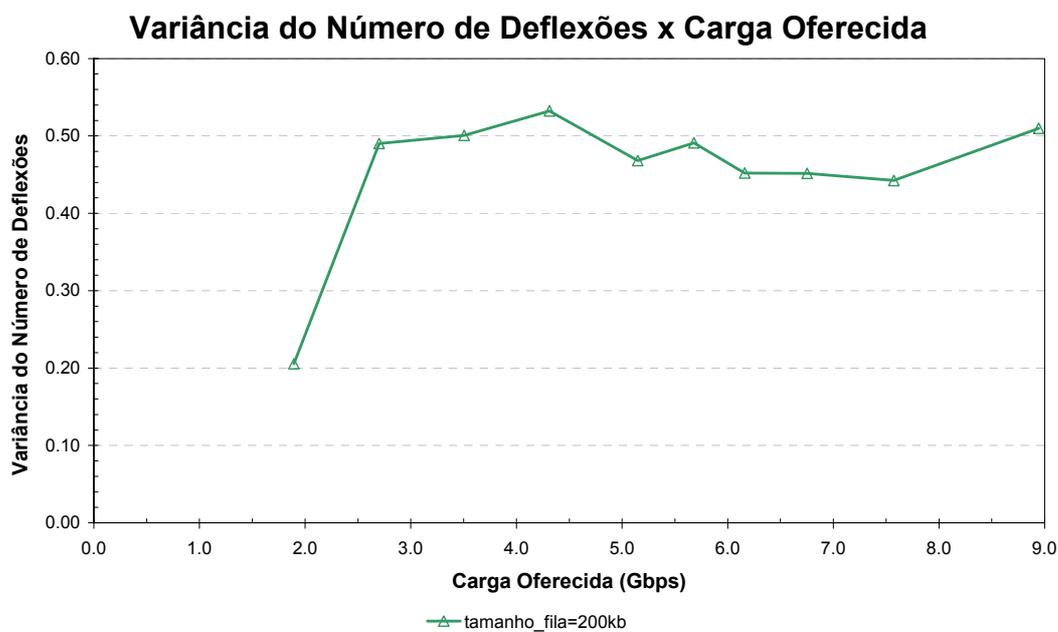


Figura 3.30: Variância do número de deflexões sofridas por cada pacote em função da carga oferecida.

A Figura 3.30 ilustra a variância do número de deflexões que cada pacote sofreu ao longo de sua trajetória na rede, em função da carga oferecida. Pode-se notar que até o ponto de saturação da rede utilizando o protocolo TCP (carga oferecida < 3,5 Gbps), a variância do número de deflexões é maior utilizando o protocolo TCP, pois como existem mais pacotes trafegando na rede do que quando é utilizado o UDP, as “colisões” tendem a ser mais aleatórias (comparar com a Figura 3.9).

Como a média e variância do número de deflexões aumentam conforme mais pacotes são inseridos na rede, conseqüentemente, a média e variância do número de *hops* e do tempo de propagação também irão aumentar (Figura 3.31, Figura 3.32, Figura 3.33 e Figura 3.34).

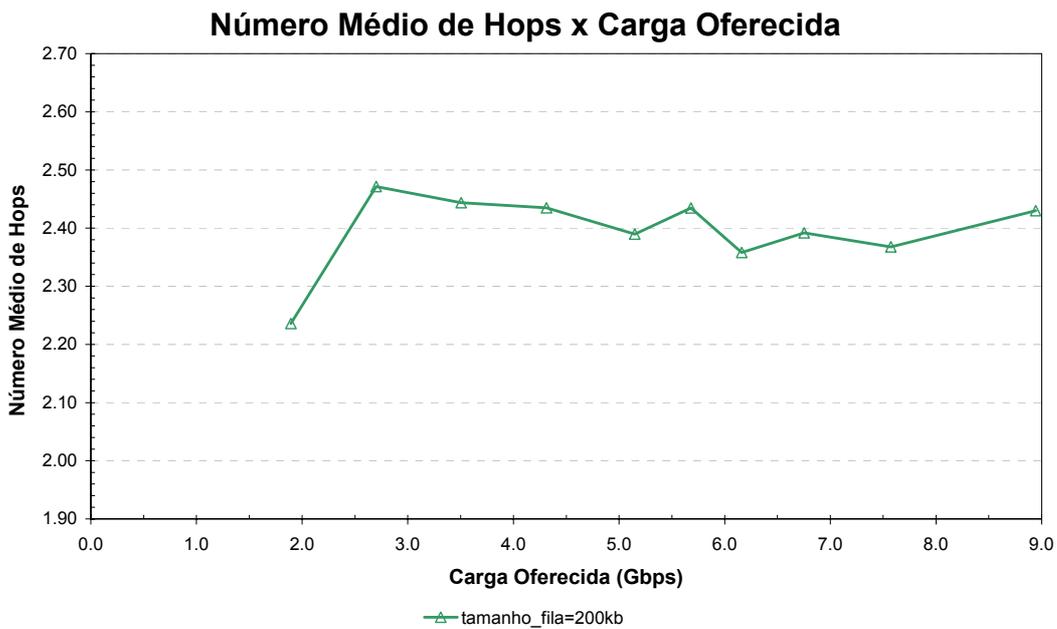


Figura 3.31: Média do número de *hops* percorridos por cada pacote em função da carga oferecida.

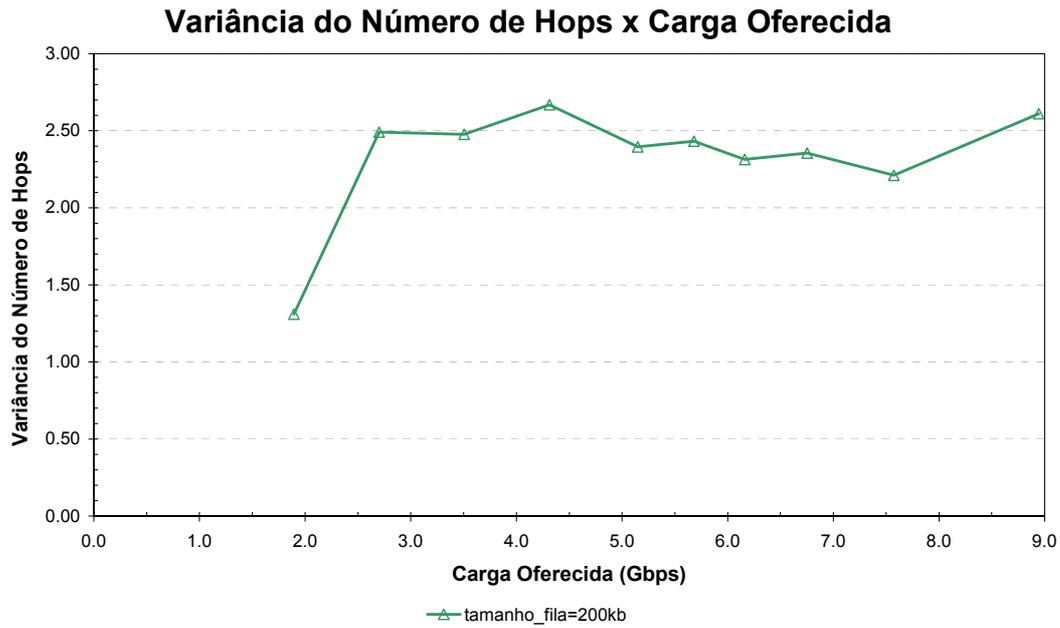


Figura 3.32: Variância do número de hops percorridos por cada pacote em função da carga oferecida.

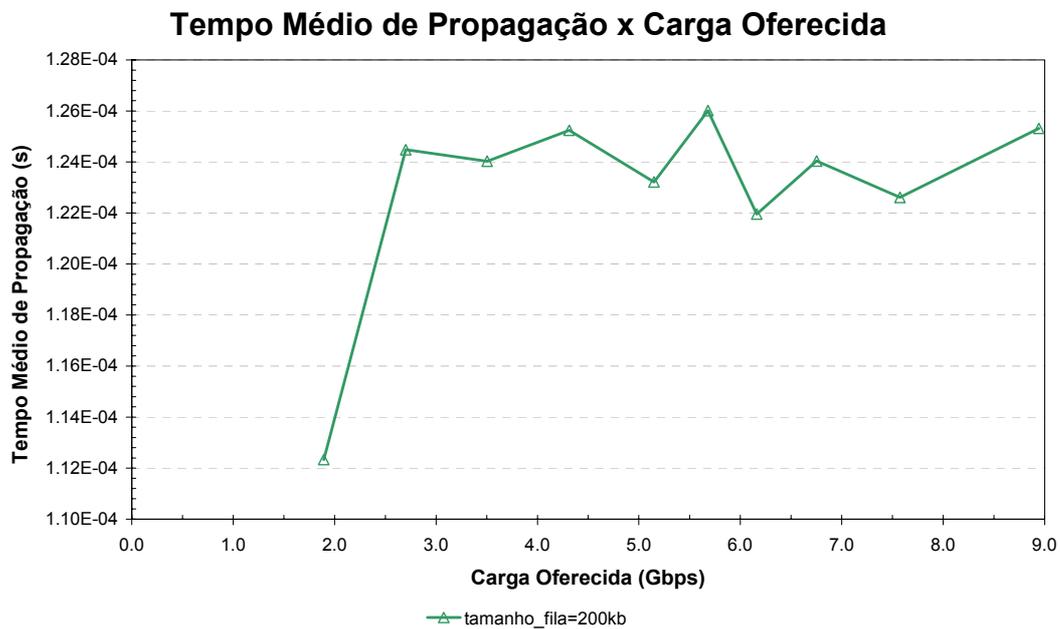


Figura 3.33: Tempo médio de propagação em função da carga oferecida.

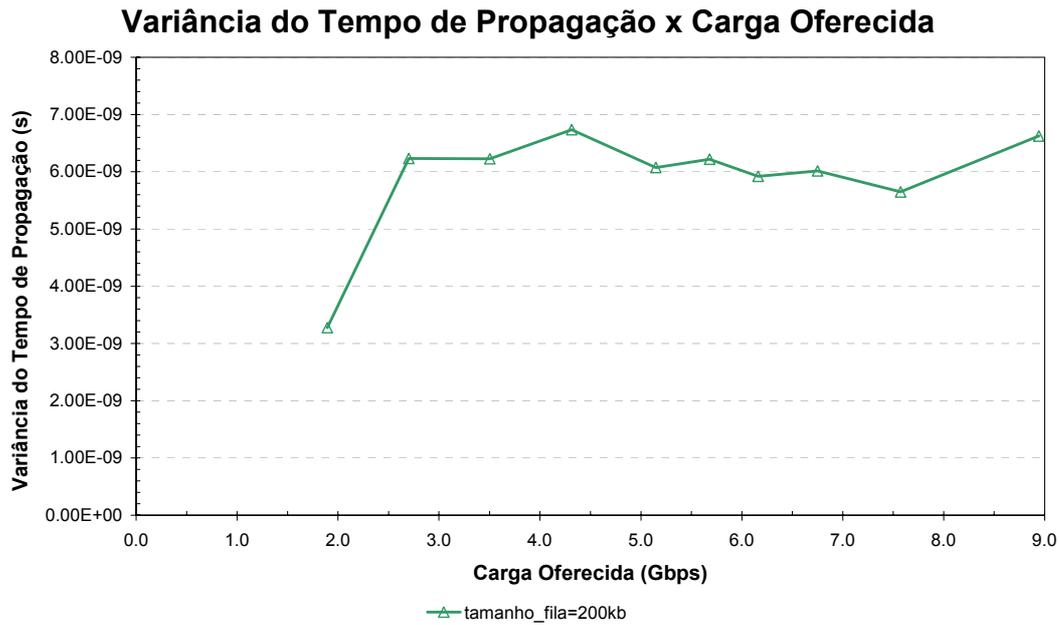


Figura 3.34: Variância do tempo de propagação em função da carga oferecida.

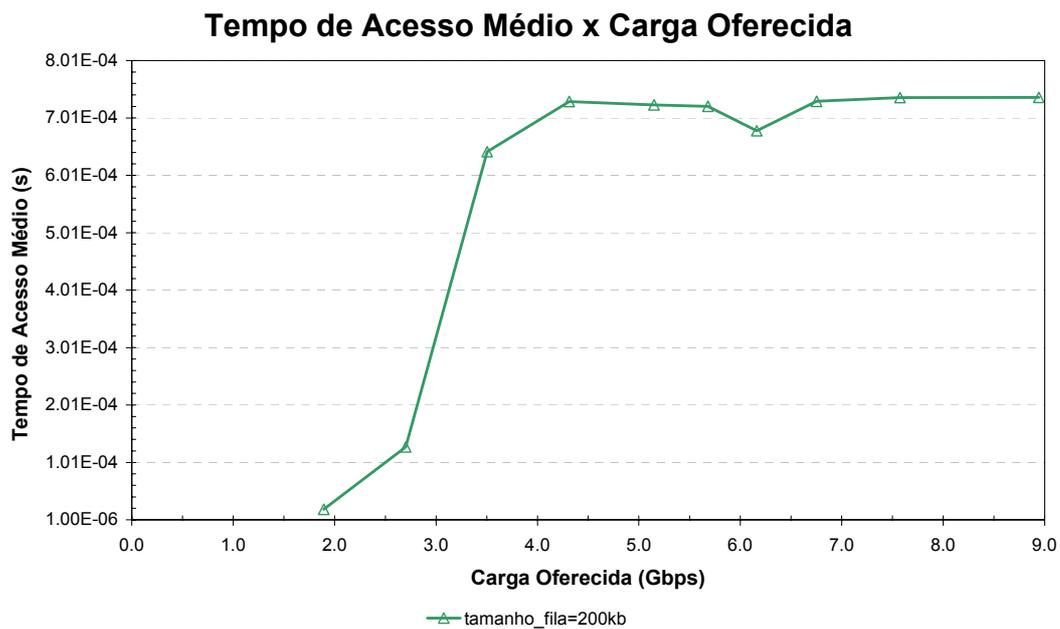


Figura 3.35: Tempo de acesso médio em função da carga oferecida.

A Figura 3.35 ilustra o tempo médio que cada pacote fica na fila até ser inserido na rede em função da carga oferecida. Pode-se notar que o tempo de acesso médio da rede, antes da saturação (carga oferecida < 3,5 Gbps), é menor para a rede MS com 9 nós utilizando o protocolo TCP (comparar com a Figura 3.14) pois os pacotes só são gerados pelas camadas superiores ao receber o pacote

de *ack*, logo as filas permanecem com poucos pacotes e conseqüentemente o tempo médio de acesso é baixo.

A Figura 3.36 ilustra a variância do tempo de acesso à rede em função da carga oferecida. Pode-se notar que existe um ponto onde a rede fica mais instável (carga oferecida $\approx 3,5$ Gbps), ou seja, um pouco antes da rede saturar a variância tem um pico.

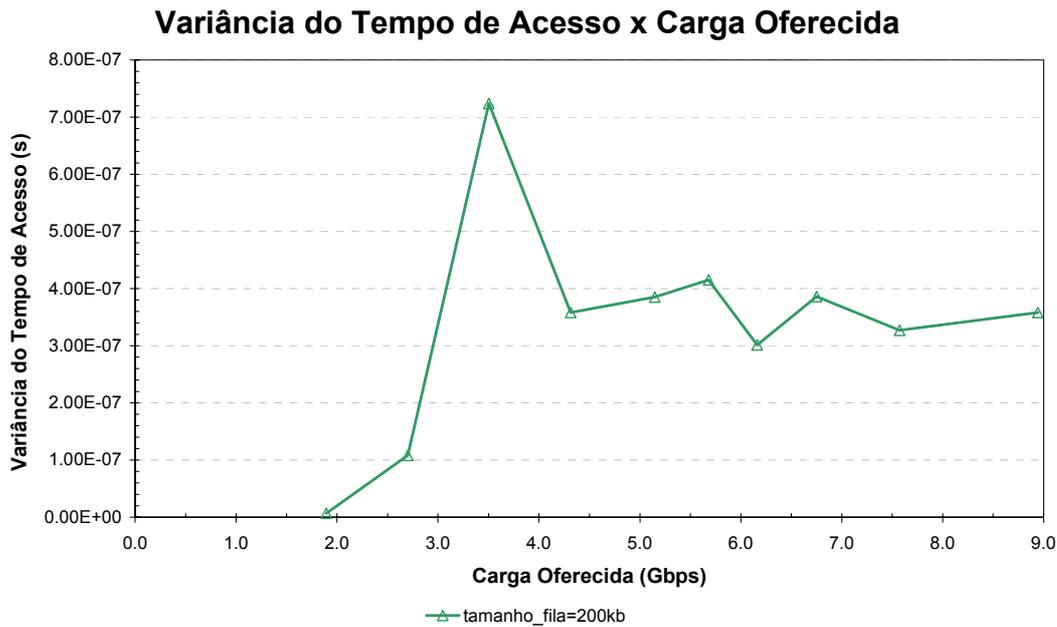


Figura 3.36: Variância do tempo de acesso em função da carga oferecida.

A Figura 3.37 ilustra a taxa de pacotes perdidos em função da carga oferecida. Nota-se que o resultado da simulação da rede utilizando o protocolo TCP indica que só existe perda de pacotes no ponto onde a variância é máxima. Isto acontece porque os pacotes só são gerados pelas camadas superiores ao receber o pacote de *ack*, logo as filas permanecem com poucos pacotes e conseqüentemente poucos pacotes são perdidos, no entanto, no ponto onde a variância tem um pico o número de pacotes nas filas varia muito, assim pode ocorrer eventuais perdas de pacotes.

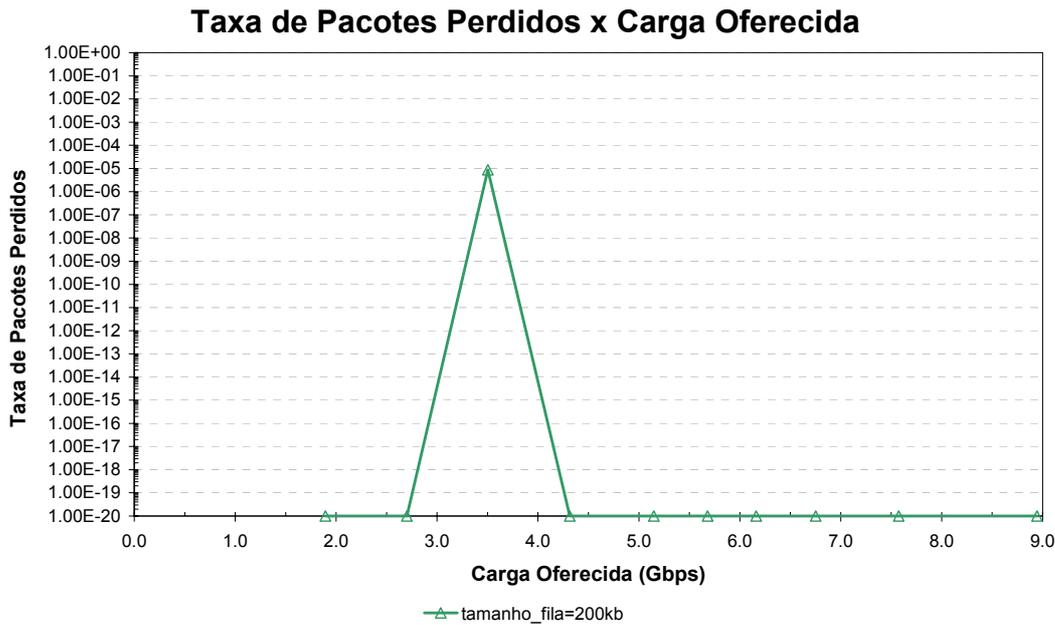


Figura 3.37: Taxa de pacotes perdidos em função da carga oferecida

3.4.4. Simulação MS com Nove Nós com *Buffer* Óptico Utilizando TCP e Sinalização na Camada Física

Nesta sessão serão apresentados os resultados das simulações de redes MS com 9 nós utilizando o protocolo TCP e tráfego CBR. Utilizaram-se enlaces de 1Gbps com 10km de comprimento (atraso de 50 μ s). A rede simulada utiliza nós com a arquitetura ilustrada na Figura 3.17.

Buffers ópticos utilizando um ou dois FDLs de 13 μ s e filas eletrônicas de 200KB foram utilizadas, por apresentarem bom desempenho nas simulações apresentadas anteriormente. Nas simulações, os pacotes de dados possuem 1625 bytes (1500bytes de *payload* + 125bytes de sinalização física) e os pacotes de *ack* de 40bytes. Como as funcionalidades de *ack* seriam facilitadas entre nós adjacentes, este tráfego foi inicialmente desconsiderado nas simulações. Nos cálculos finais, como o percentual de tráfego adjacente é conhecido, foi possível estimar-se o tráfego total e acrescentá-lo aos resultados das simulações.

Diversas simulações do mesmo “cenário” foram realizadas de modo a aumentar a confiança nos resultados obtidos.

A Figura 3.38 ilustra a vazão da rede em função da carga oferecida. Pode-se notar que o ganho da vazão é significativo para redes com nós com *buffers* ópticos (FDLs). Os resultados indicam que a utilização de *buffers* ópticos aumenta a vazão da rede utilizando o protocolo TCP (comparar com Figura 3.28). Diferente do UDP, no TCP quando os nós possuem dois *buffers* ópticos a vazão da rede é maior do que quando os nós possuem apenas um (comparar com Figura 3.18).

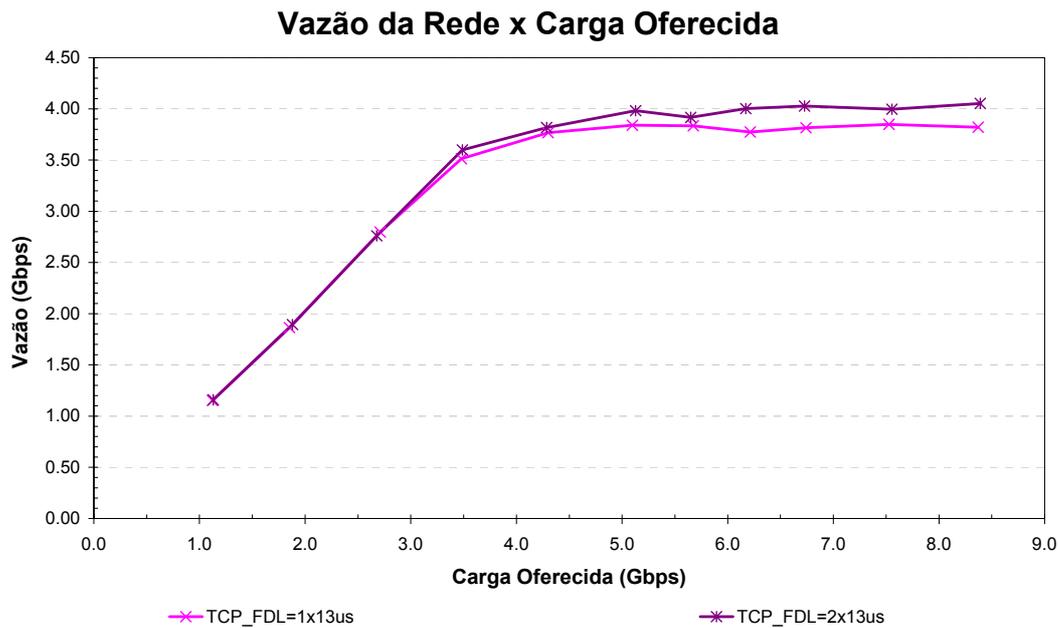


Figura 3.38: Vazão da rede em Gbps em função da carga oferecida.

A Figura 3.39 ilustra o número médio de deflexões que cada pacote sofreu ao longo de sua trajetória na rede, em função da carga oferecida. Pode-se notar que o número médio de deflexões, quando os nós utilizam *buffers* ópticos, é muito menor (comparar com a Figura 3.29). No entanto, a partir da carga oferecida de 2,5 Gbps, um *buffer* óptico passa a não ser suficiente e o número médio de deflexões sobe rapidamente. Quando os nós possuem dois *buffers* ópticos o número médio de deflexões só começa a aumentar a partir de 3,5 Gbps e sobe mais lentamente do que quando os nós possuem apenas um *buffer* óptico.

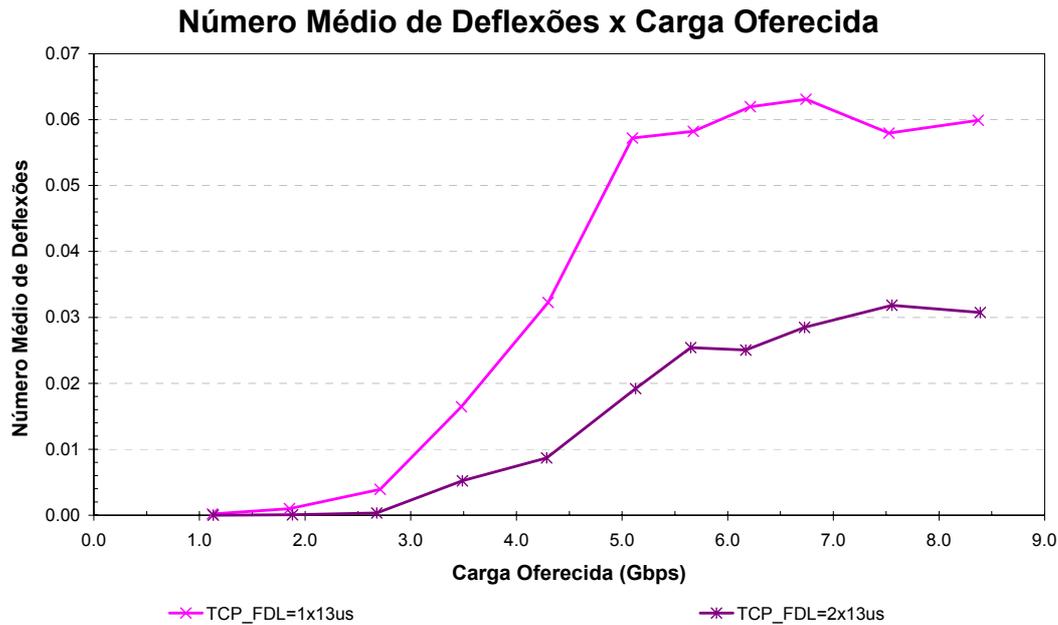


Figura 3.39: Média do número de deflexões sofridas por cada pacote em função da carga oferecida.

A Figura 3.40 ilustra a variância do número de deflexões que cada pacote sofreu ao longo de sua trajetória na rede em função da carga oferecida. Pode-se notar que a variância do número de deflexões até o ponto de saturação (carga oferecida < 3,5 Gbps) em redes cujos nós possuem *buffers* ópticos e utilizam o protocolo TCP é quase nula já que o número de deflexões é bem reduzido. Assim, o tempo de propagação dos pacotes não varia muito. No entanto após o ponto de saturação (carga oferecida > 3,5 Gbps) o número de deflexões aumenta e conseqüentemente sua variância também sobe. No entanto o valor é bem inferior ao apresentado em redes sem *buffers* ópticos (comparar com a Figura 3.30).

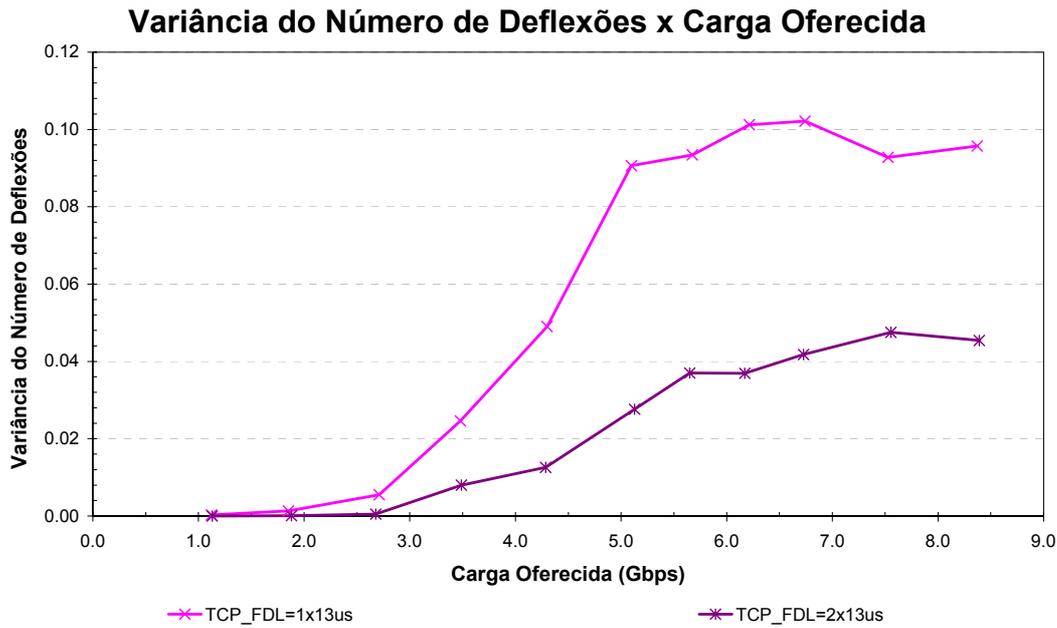


Figura 3.40: Variância do número de deflexões sofridas por cada pacote em função da carga oferecida.

Como a média e variância do número de deflexões permanecem praticamente constantes até o ponto de saturação (carga oferecida < 3,5 Gbps) em redes utilizando o protocolo TCP e cujos nós possuem *buffers* ópticos, a média e variância do número de *hops* também permanecem. No entanto, a média do tempo de propagação aumenta mesmo antes do ponto de saturação, pois os *buffers* ópticos já estão sendo utilizados. A partir de certo ponto (carga oferecida \approx 3 Gbps) um buffer óptico passa a não ser suficiente e começa a haver deflexões; conseqüentemente, o número médio de *hops* e o tempo de propagação passam a crescer mais rápido. Com dois *buffers* ópticos em cada nó da rede, o número médio de deflexões fica bastante reduzido, assim o número médio de *hops* não aumenta. No entanto, o tempo de propagação aumenta devido à maior utilização dos *buffers* ópticos (Figura 3.41, Figura 3.42, Figura 3.43 e Figura 3.44).

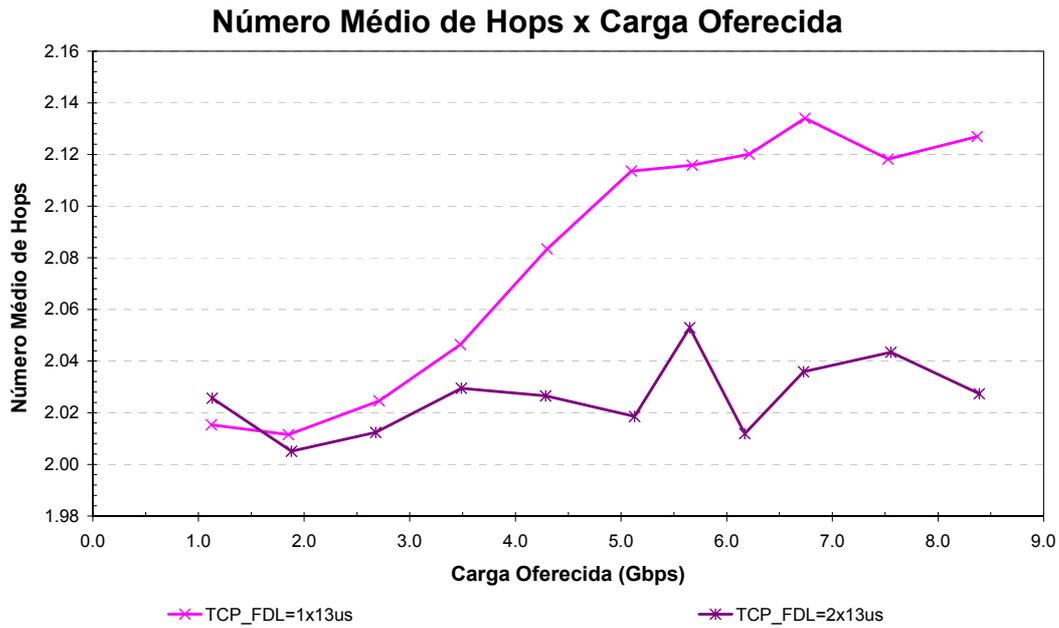


Figura 3.41: Média do número de *hops* percorridos por cada pacote em função da carga oferecida.

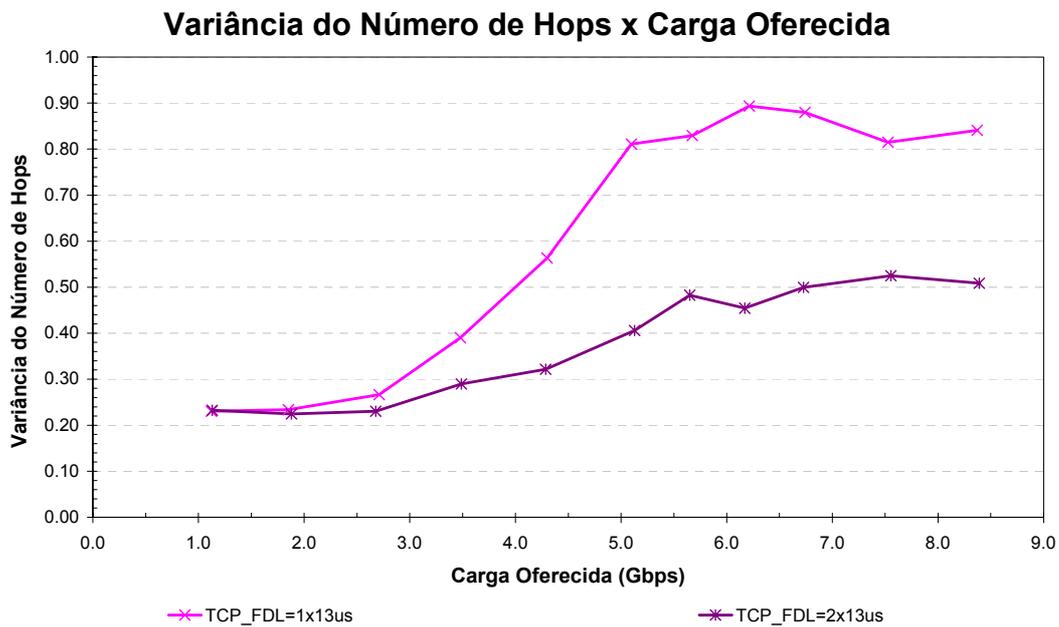


Figura 3.42: Variância do número de *hops* percorridos por cada pacote em função da carga oferecida.

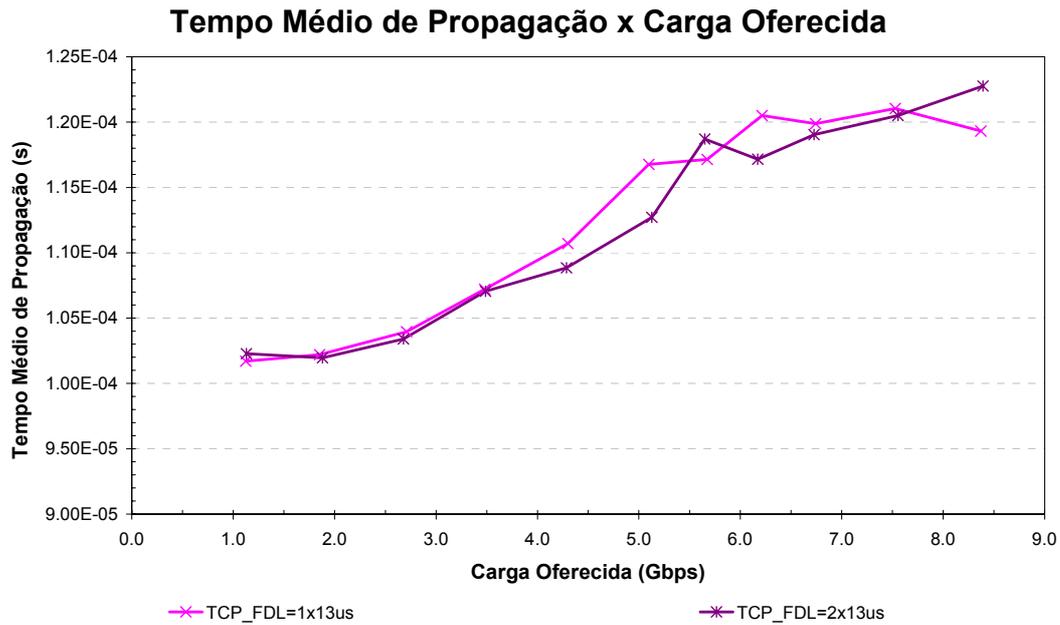


Figura 3.43: Tempo médio de propagação em função da carga oferecida.

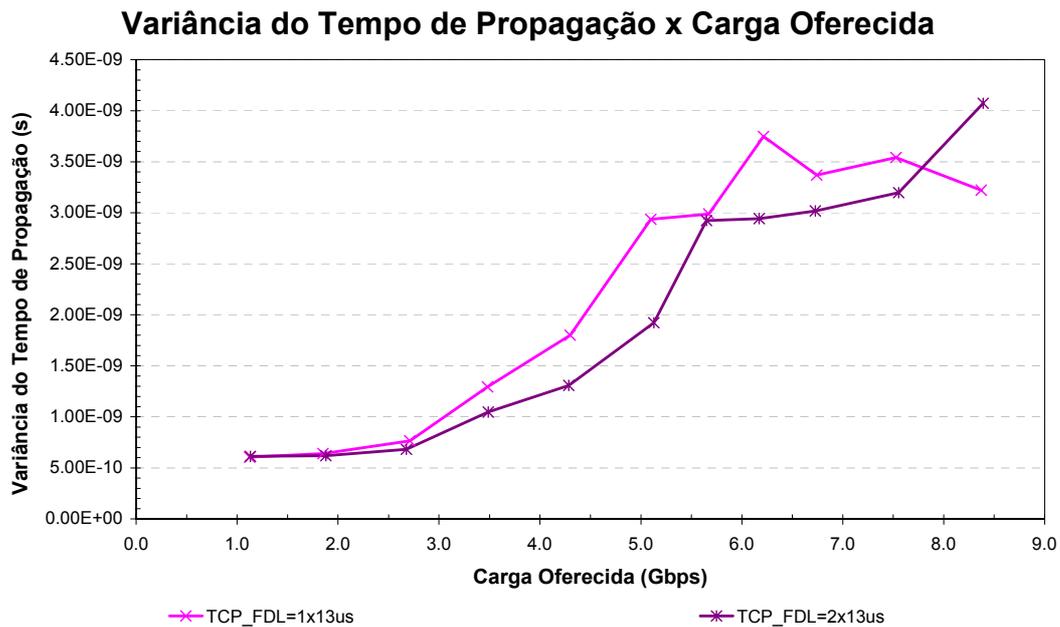


Figura 3.44: Variância do tempo de propagação em função da carga oferecida.

A Figura 3.45 mostra um resultado interessante. Como a utilização de *buffers* ópticos diminui o tempo médio de propagação, os pacotes de *ack* chegam mais rapidamente, assim, novos pacotes são gerados e inseridos nas filas eletrônicas. Então, diferente do que acontece com o UDP (ver Figura 3.25 e

Figura 3.35), no TCP o tempo médio de acesso à rede é maior quando *buffers* ópticos são utilizados.

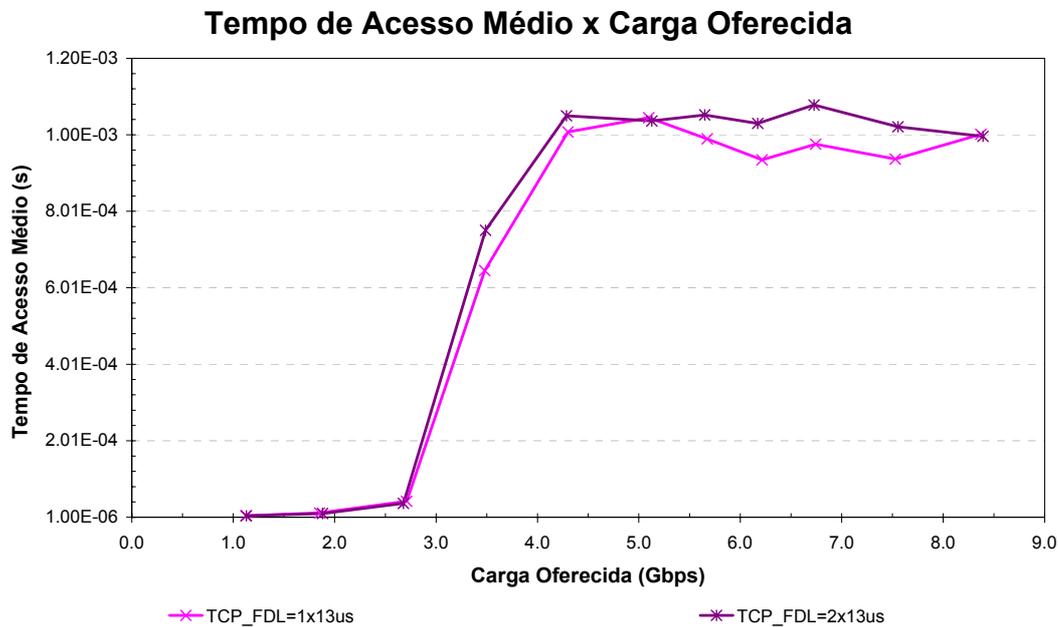


Figura 3.45: Tempo de acesso médio em função da carga oferecida.

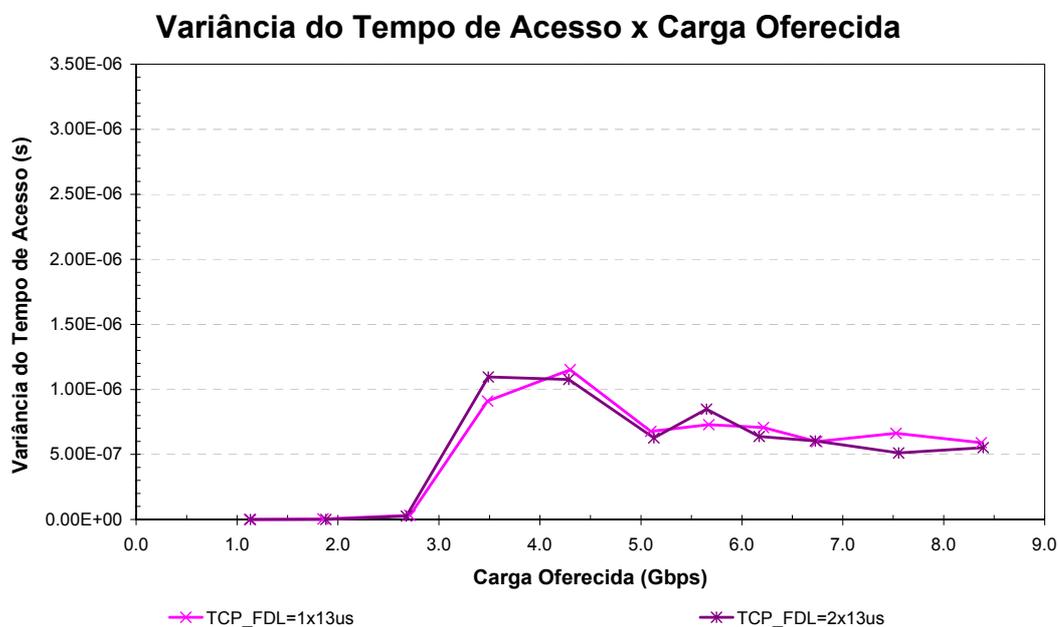


Figura 3.46: Variância do tempo de acesso em função da carga oferecida.

A Figura 3.46 ilustra a variância do tempo de acesso à rede em função da carga oferecida. Pode-se notar nos resultados das simulações do TCP que a

variância após a saturação não retorna para o patamar de antes da saturação, indicando assim que, quando a rede fica saturada, o número de pacotes nas filas varia mais do que quando a rede ainda não saturou. Este gráfico também mostra que a variância é maior no TCP quando *buffers* ópticos são utilizados.

A Figura 3.47 ilustra a taxa de pacotes perdidos em função da carga oferecida. Nota-se que como as filas ficam mais cheias quando *buffers* ópticos são utilizados com o protocolo TCP do que quando não são utilizados, a taxa de pacotes perdidos é maior no TCP com *buffers* óticos do que no TCP sem.

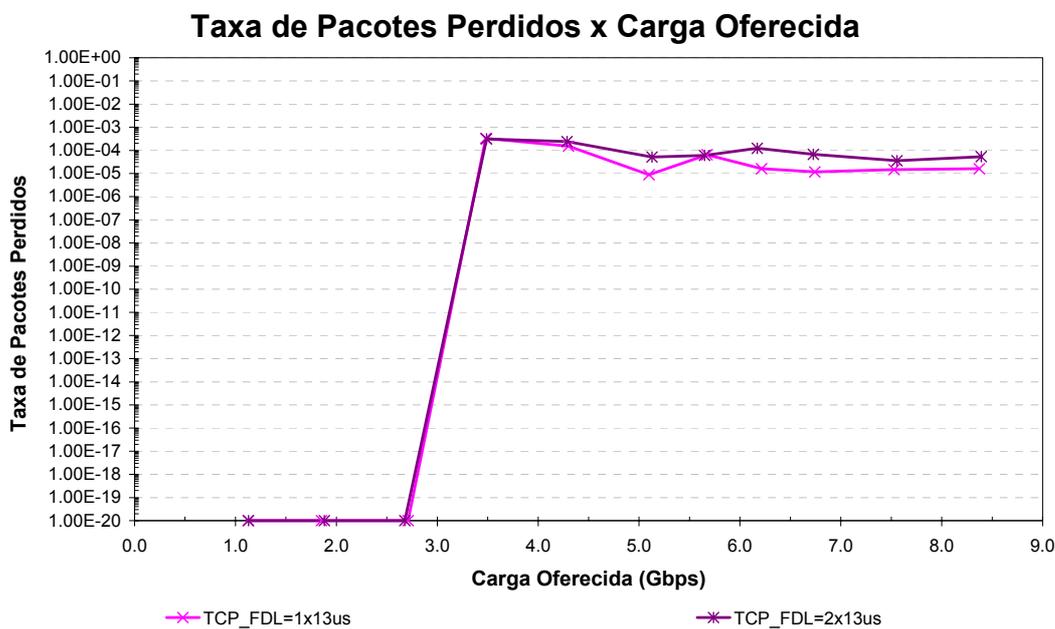


Figura 3.47: Taxa de pacotes perdidos em função da carga oferecida

3.4.5. Análise do Efeito da Latência na Vazão do TCP

Na sessão anterior, foram apresentados resultados de simulações utilizando o TCP com janela de tamanho igual a 20 pacotes. Surge então a dúvida: a vazão obtida foi limitada pela topologia da rede ou pelo protocolo TCP? Para responder esta questão, um enlace de 1 Gbps foi simulado. Variou-se o comprimento do enlace e o tamanho da janela do TCP. Os resultados obtidos são mostrados na Figura 3.48.

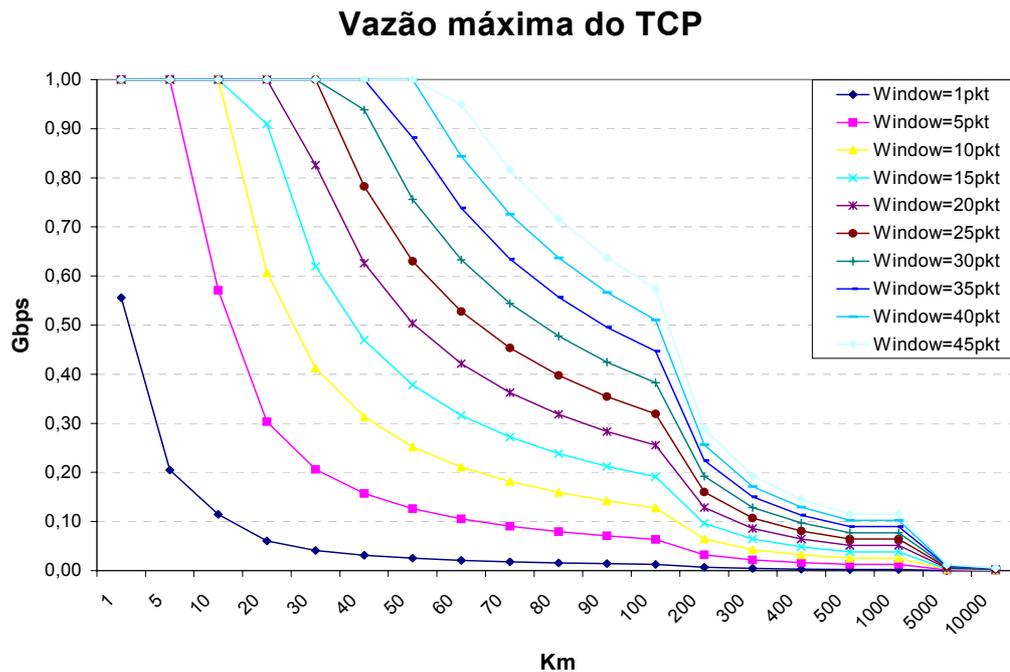


Figura 3.48: Vazão máxima do TCP em função da distância e do tamanho da janela

Pode-se notar que, para um enlace de 20 km de comprimento, uma janela de 20 pacotes é suficiente. No entanto, para um enlace de 30 km a vazão já cai para aproximadamente 800 Mbps. No entanto, como os nós possuem uma conexão TCP para cada outro nó na rede, a taxa máxima utilizada em cada uma destas conexões é 125 Mbps.

Logo, pode-se concluir que, para redes MS de 9 nós com enlaces de 10 km, a janela de 20 pacotes é suficiente. Quando as distâncias envolvidas são ampliadas, a vazão é totalmente dependente do tamanho da janela.

3.5. Resumo

Este capítulo apresentou alguns conceitos básicos sobre simulação de redes, descrevendo os objetivos de uma simulação e como obter modelos de simulação válidos e resultados confiáveis; uma descrição do NS-2 que foi utilizado para realizar as simulações; uma comparação entre os resultados obtidos através da simulação e os obtidos através da metodologia analítica desenvolvida no capítulo anterior, indicando que os modelos apresentaram resultados semelhantes; uma introdução sobre os protocolos UDP e TCP; os resultados das simulações das

redes MS com 9 nós, com ou sem *buffers* ópticos, utilizando o protocolo UDP ou TCP; e uma análise do efeito da latência na vazão do protocolo TCP, indicando que para as configurações utilizadas, a latência não interferiu no desempenho do TCP.