ORÊNCIO COUTINHO JÚNIOR
IGOR MIRANDA
RICARDO ARRUDA

Avaliando Desempenho de Políticas de Fila DropTail e SFQ diante de um ataque Distributed Denial of Service

UNAMA
Belém, Novembro de 2002.

UNIVERSIDADE DA AMAZÔNIA
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CCET
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Avaliando Desempenho de Políticas de Fila DropTail e SFQ diante de um ataque Distributed Denial of Service

por

IGOR MIRANDA
ORÊNCIO COUTINHO JÚNIOR
RICARDO ARRUDA

TCC apresentado como exigência parcial para a obtenção do título de Bacharelado em Ciência da Computação na Universidade da Amazônia, elaboração sob orientação do Prof. Mauro Margalho Coutinho.

UNAMA
Belém, Novembro de 2002.

Prof. 1 Banca

Prof.2 Banca

Prof.3 Banca

Data____/____/_____

Resumo

Atualmente é muito comum a prática de ataques hackes, os quais têm como objetivo simplesmente causar caos em sistemas de grande porte. Um dos grandes BackBones do Brasil, a RNP, até hoje, não teve sua segurança posta em evidência. O trabalhão proposto tem por objetivo avaliar o desempenho de simulações feitas com a ferramenta NS (Network Simulator). Utilizar-se-á como cenário o backbone da RNP onde serão executados ataques simulados. Serão avaliados algumas falhas de segurança e sugestões de procedimentos para minimizar tais falhas serão propostas.
Os resultados serão avaliados com base em gráficos obtidos a partir da evolução do quadro de simulação.

Abstract

Currently it is very common the pratice of hackers attacks, which has as objective simply to cause chaos in Large Systems. One of the great backbones of Brazil, RNP, until today, has not been its security dispatches posted in invidence. The considered work has for objective to evaluate the performance of simulations made with tool NS (Network Simulator). RNP's backbone will de used as a scenario where simulated attacks will be executed. Some security failures will be evaluated and procedure suggestions will be proposed to minimize such failures. The results will be evaluated based on graphics obtained from the evolution of simulation picture.

Sumário

Lista de Figuras

Lista de Gráficos

Lista de Siglas

CAE – Centro de Atendimento a Emergências

CAIS – Centro de Atendimento a Incidentes de Segurança

CBR – Constant Bit Rate

CRC – Cyclic Redundancy check

DoS – Denial of Serviçe

DdoS – Distribued Denial of Serviçe

FIFO – First In First Out

IEEE - Internet

LAN – Local Area Network

IP – Internet Protocol

ISP – Internet service Provider

NS – Network Simulator

RNP – Rede Nacional de Pesquisa

SFQ – Stocasthic Fair Queue

TCP – Protocol Control Transport

UCP – Unidade Central de Processamento

UDP –

VINT – Virtual Internetwork Testbed

WAN – Wide Area Network

1. O Estado da Arte em Segurança de Sistemas

Inicialmente mostrar-se-á um apanhado geral das expectativas mediante o desenvolvimento do trabalho referente aos problemas de segurança e confiabilidade encontrados em algumas “WANS” (Wide Area Network), explicitando sucintamente os tópicos que serão abordados no decorrer do trabalho, explanando algumas ferramentas e softwares que foram usadas para auxiliar na elaboração de simulações e geração de gráficos.

1.1 Visão geral sobre segurança em WANS

As redes de longa distância, também chamadas de WANS (Wide Area Network) são redes geograficamente distribuídas que surgiram da necessidade de se compartilhar recursos especializados. Por terem um custo bastante elevado, tais redes, em geral são públicas. Geralmente cobrem uma grande área geográfica, no caso da RNP (Rede Nacional de Pesquisa) cobre um país inteiro. As WANS são na realidade constituídas por múltiplas redes interligadas, por exemplo, redes locais e outras redes de longa distância ou metropolitanas. O exemplo mais divulgado é a Internet. Dada a sua dimensão e, uma vez que engloba diversas redes locais (LANS) e de longa distância (WANS), as tecnologias usadas para a transmissão dos dados são as mais diversas. Contudo, para que a troca de informação se processe, é necessário um elo comum que assente sobre essa tecnologia heterogênea. Esse elo comum é o protocolo de rede.

A interligação de redes de diferentes tecnologias é assegurada por dispositivos conhecidos por "routers" (roteadores). Um roteador possui tipicamente ligação física a duas ou mais redes, recebendo dados de uma rede para os transmitir a outra. Um exemplo típico é a ligação de uma rede “Ethernet” (802.3 – IEEE) a uma rede ponto-a-ponto. Por exemplo, quando um usuário particular estabelece uma ligação telefônica com um fornecedor de serviços Internet (ISP – Internet service Provider) podemos considerar que a parte da rede telefônica que é usada passa a fazer parte da grande rede WAN, que é a Internet.

1.1.1 Técnicas de Ataque

O aspecto “segurança” sempre tem sido colocado em evidência quando se fala em redes. O trabalho proposto tem como finalidade analizar os resultados obtidos mediante uma simulação de uma falha de serviço, gerada em uma video conferência, utilizando-se da modelagem do backbone da RNP, oriunda de ataques externos. As simulações foram geradas no NS (Network Simulator) e visualizadas no NAM (ferramenta de trabalho utilizada – Network AniMator). Entre as diversas formas de ataque que se utiliza hoje em dia, uma das mais comuns é o de negação de serviço, chamado de DoS

(Denial Of Service) e os ataques de DdoS (Distributed Denial of Service) que atualmente estão incomodando administradores de redes WANS.

BackDoors e Trojans - A definição de Trojans e Backdoors por vezes se confundem. Os ***Backdoors***, também conhecidos como "*Cavalos de Tróia*", são programas que quando executados, permitem a terceiros a execução de "processos" remotos (***invasão do sistema***) no computador em questão (denominado com "*vítima*"). O termo ***Trojan*** é mais empregado a programas cujo código serve apenas para danificar arquivos no sistema – portanto, deve-se pensar duas vezes quando se quiser testar um programa que esteja definido como *Trojan*. *Os principais Backdoors disponíveis na Internet, atualmente, são Backorifice, BO, SubSeven, NetBus e WinCrash*

Buffer Overflow – O buffer overflow é uma falha de programação que faz com que haja um "transbordamento" da área de memória de uma determinada variável sobre a área reservada para outras variáveis, ou sobre a área de memória que contém código executável. Softwares podem ser derrubados ou serem forçados a executar outras funções (código arbitrário), esta técnica atinge todos os tipos de software, sistemas operacionais, Serviços (ex: Web Servers) e aplicativos (scripts CGI). Os "piratas de rede", popularmente conhecidos como **"**hackers", desenvolvem programas para explorar um buffer overflow (exploits – programas desenvolvidos por "hackers" para explorar vulnerabilidades) em um determinado software/versão. Estes programas são extremamente sofisticados, exigindo que se mescle em tempo de execução dois códigos de máquina. Uma vez desenvolvido este exploits e divulgados na Internet, qualquer hacker pode se utilizar dos mesmos para fazer um ataque contra um servidor que utilize o software com problema de "buffer overflow". Este problema ocorre quando o programador esquece de validar se os dados de entrada, recebidos por uma variável que esta dentro dos limites máximos de tamanho reservado para aquela variável. Um "hacker" (pirata) pode utilizar uma falha de buffer overflow para paralisar um programa, forçando-o a executar instruções ilegais, ou então danificando a sua área de dados a tal ponto que o estado do programa fique tão inconsistente a ponto de causar um auto cancelamento. Desta forma, o buffer overflow pode ser utilizado para causar negação de serviço, indisponibilizando o serviço prestado por um programa que foi

paralisado. O “buffer overflow” também pode ser utilizado para fazer com que um programa passe a executar uma seqüência de código determinada pelo “hacker”. Neste caso, os dados que serão enviados contém uma seqüência de instruções de máquina que irão substituir o código de máquina original. Isto permite ao buffer overflow criar uma porta de entrada.

Exemplo de um Código que gera Buffer Overflow: [LV-02;LV01]

```
#include <stdio.h>
char s1[10];
char s2[10];

int main()
{
printf (“\ns 2antes = %s” , s2);
printf(“\ns2 depois = %s”,s2);
}
```

Trata-se de um programa extremamente simples, que lê um string de caracteres (s1), e em seguida exibe um outro string de caracteres (s2), o qual não recebeu nenhum valor. O resultado esperado do programa apresentado na figura 1.1 é exibir s2 como um string vazio (nenhum caracter). Contudo, em nenhum instante este programa valida se à quantidade de caracteres lido em s1 ultrapassa o tamanho da variável. Isto abre condições para um buffer overflow.

```
8:    printf("\ns2 antes = %s",s2);
00401028   push        offset _s2 (004237d0)
0040102D   push        offset string "\ns2 antes = %s" (00420040)
00401032   call        printf (00401160)
00401037   add         esp,8
9:    printf("\nDigite s1: ");
0040103A   push        offset string "\nDigite s1: " (00420030)
0040103F   call        printf (00401160)
00401044   add         esp,4
10:   gets(s1);
00401047   push        offset _s1 (004237b0)   s1
0040104C   call        gets (00401090)
00401051   add         esp,4
11:   printf("\ns2 depois = %s",s2);
00401054   push        offset _s2 (004237d0)   s2
00401059   push        offset string "\ns2 depois = %s" (0042001c)
0040105E   call        printf (00401160)
00401063   add         esp,8
12:
13:   }
```

Fig. 1.1 – Código de máquina de Buffer Overflow

Na figura 1.1 podemos ver o mesmo programa com instruções em código de máquina.

IP-Spoofing - Cada máquina que está se comunicando na Internet tem um identificador único: o endereço IP (Internet Protocol) que é similar ao número da placa de um carro. Contudo, este endereço pode ser forjado, permitindo que uma máquina utilize o endereço de uma outra máquina. Esta técnica é denominda IP spoofing. Utilizando o IP Spoofing o atacante pode esconder a sua verdadeira identidade (IP de origem). O IP spoofing faz com que os roteadores/servidores fiquem sobrecarregados ao ter de responder mensagens com endereços IPs falsos. Se isto for aplicado em grande escala, possibilita ataques de negação de serviço (DoS). Para construir um programa com IP spoofing, o “hacker” ignora a camada de protocolo IP do sistema operacional e gera os seus próprios pacotes. Esta técnica dá ao hacker um controle total sobre os pacotes IP. Quando este programa com IP Spoofing for executado, permitirá a escolha dos endereços IP de origem. O programa pode permitir ao usuário especificar um endereço IP, um grupo de endereços IP, ou ainda solicitar ao programa que gere endereços IP aleatoriamente. Quando um servidor/roteador tenta responder a um pacote com endereço falso (endereço spoofado), poderá acontecer duas coisas:

- Se existir uma máquina ativa com o endereço IP indicado, esta máquina responderá que não está interessada em receber dados deste servidor, uma vez que ela não solicitou nenhuma comunicação com este host.
- Se não existir nenhuma máquina com o endereço IP indicado, o servidor/roteador acabará recebendo da rede uma mensagem de "Host Unreacheable" (rede inalcançável).

Tanto um caso quanto o outro fazem com que haja um consumo de UCP (Unidade Central de Processamento) e de banda neste servidor. Este é um dos artifícios empregados nos ataques de denial of service. Neste ataques o servidor recebe milhares de pacotes com IP Spoofing e o resultado do tratamento a estes pacotes gera uma sobrecarga no host, ocasionando uma negação de serviço.

Sniffer - Computadores em uma rede local (Ethernet) compartilham um meio físico. Normalmente, uma placa de rede com os pacotes destinados a ela, e descartam os demais. Um programa Sniffer coloca a placa de rede em modo promíscuo, possibilitando que um computador receba todos os pacotes que circulam no segmento de rede (dominio de colisão) a que pertence. Isto possibilita ao hacker obter informações privilegiadas (ex: senhas que circulam sem criptografia) que facilitem um ataque.

ICMP Flood - Muitas pessoas acreditam que o ICMP (Internet Control Message Protocol – Protocolo de Controle de Mensagens) seja apenas um protocolo similar ao Ping. Porém, ele é bem mais sério do que parece. Tanto o TCP (Transmission Control Protocol) como o UDP (User Datagram Protocol) são pacotes gerados por programas do tipo FTP (File Transfer Protocol), Telnet e os outros. Porém, o ICMP é gerado diretamente pelo Kernel (Núcleo do sistema) e tem a função de nomear erros e controlar o fluxo de informações entre dois hosts diferentes. Assim como no Ping comum, o ICMP Flood pode ser usado para derrubar estações e servidores. Está técnica é conhecida como Flood, que consiste em enviar um grupo de dados para portas que são pouco usadas pelo computador. Ou seja, o ICMP Flood é o ato de enviar o máximo de pacotes no menor espaço de tempo possível a fim de tornar a conexão de um usuário tão lenta a ponto de desconectá-lo da rede. O ataque ICMP Flood pode ser dividido em duas categorias: usuários de modem e usuários de rede. Um usuário que esteja conectado via modem a 14.400 bps, dificilmente conseguirá atacar alguém com um ICMP Flood, já que não há velocidade suficiente para o envio de pacotes. Se você pensar bem, o ICMP Flood funciona mais ou menos como um cabo de guerra. Quem tem a maior força, que no caso é a largura de banda, acaba derrubando a outra conexão.

Seguindo essa lógica, um usuário em uma conexão do tipo T3 (Conexões de banda larga que atingem 55Mbps), derruba uma do tipo T1 (Conexões de banda larga que atingem 1.544Mbps), que derruba um modem de 56 Kbps, que derruba um de 28,8 Kbps e assim sucessivamente. [LV-02]

DoS e DDoS – Presenciamos atualmente um crescimento muito grande de ataques do tipo DoS (Denial of Service), isto é, negação de serviço que por sua vez é um ataque geralmente usado por script kiddies (são receitas de bolo deixadas na Internet para que adolescentes curiosos as executem, dando início a um procedimento de invasão) ou até mesmo por “crackers” (denominação da aos piratas, hackers, que fazem algazarras em sistemas) que não querem ter muito trabalho ou não têm habilidade suficiente para realizar um ataque mais elaborado para "derrubar" um alvo.

O DoS consiste em enviar uma quantidade de mensagens, para um determinado alvo, maior do que a quantidade que ele pode suportar. Este tipo de ataque ficou ainda mais conhecido quando alguns hackers usaram o método para derrubar sites de grande porte como o do Yahoo e do eBay, em fevereiro 1999.

Na prática, o programa mais usado para realizar este serviço é o "ping", que vem em todas as distribuições do Unix, do Linux e nas versões mais modernas do Windows. As mensagens são enviadas pela Internet por meio de pacotes (de bits), o que o ping faz é simplesmente enviar constantemente uma quantidade alta de pacotes para que possa sobrecarregar o seu alvo e assim parar os seus serviços por falta de recurso. Em essência, o comando ping seria usado para verificar se um determinado computador "está vivo", ou seja, está com uma conexão ativa na rede. O ping envia uma baixa quantidade de bytes (pacotes de 32 bytes em windows e 64 bytes em unix) para se comunicar com outro computador. Se este estiver ativo, retornará uma mensagem dizendo que os pacotes foram recebidos com sucesso. O que os hackers fazem é especificar o tamanho dos pacotes que serão enviados e adicionar uma instrução para que ele mande estes pacotes continuamente. Conforme mostrado na figura 1.2

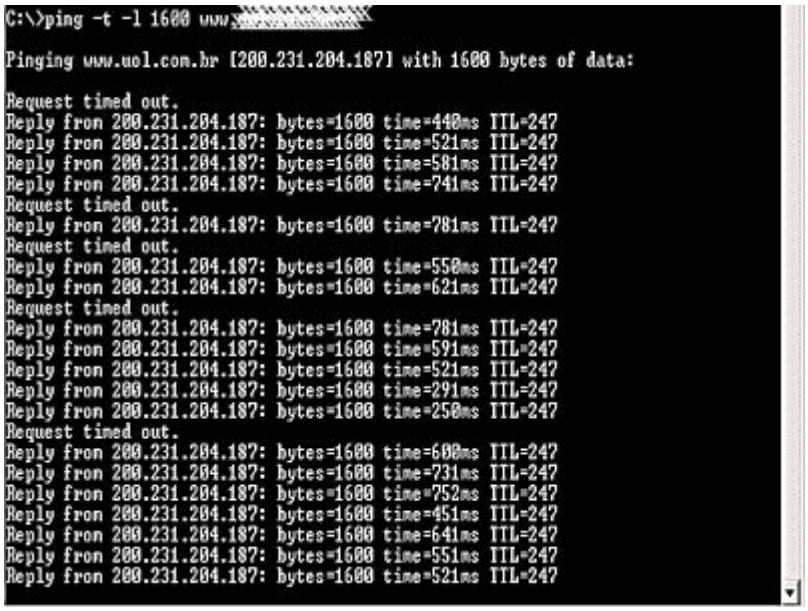


Fig. 1.2 – Exemplo do comando ping

Certamente, realizando-se este procedimento sozinho, percebe-se que nada de grave acontecerá com o "alvo" atacado. É agora que falaremos do DDoS (Distributed Denial of Service). Ataques distribuídos, semelhantes aos conceitos de sistemas distribuídos, são ataques que podem ser efetuados a partir de diversos computadores de forma bastante simples. Neste tipo de ataque é feita uma sobrecarga ou inundação de pacotes, formando uma quantidade de dados global maior que uma rede ou um host pode suportar. Isto, por conseqüência, pode demonstrar a quantidade de tráfego que um host qualquer pode suportar utilizando esta metodologia como uma ferramenta para teste de desempenho, ou então tornar instável a rede e prejudicar os serviços oferecidos por ela. Basicamente o ataque caracteriza-se por explorar vulnerabilidades e através disto obter acesso privilegiado a máquinas que preferencialmente operem em redes de banda larga. Os sistemas operacionais preferidos para utilização são o Solaris e Linux devido à existência de rootkits e sniffers, que são programas geralmente criados pelos próprios “hackers” para facilitar a entrada em sistemas. Após a invasão é criada uma lista dos IPs das máquinas exploradas para formar a rede de ataque. Neste ponto, cada uma das máquinas listadas já possui instalado software necessário para efetuar o ataque propriamente dito. Em um próximo passo são escolhidas as máquinas que serão mestres (máquinas que recebem os comandos de ataque e

comandam os agentes) e as que serão agentes (máquinas que efetivamente concretizam o ataque). Nos agentes é instalado e executado o software necessário e estes passam a anunciar ao mestre a sua presença. Assim, para efetuar o ataque basta que o mestre forneça o IP a ser atacado e o tempo de ataque e todos os agentes que entrarão em atividade. Como conseqüência, pode-se saturar o link ou paralisar os serviços oferecidos pela vítima. Veja na figura 1.3 a ilustração de uma rede de ataque: [LV-02]

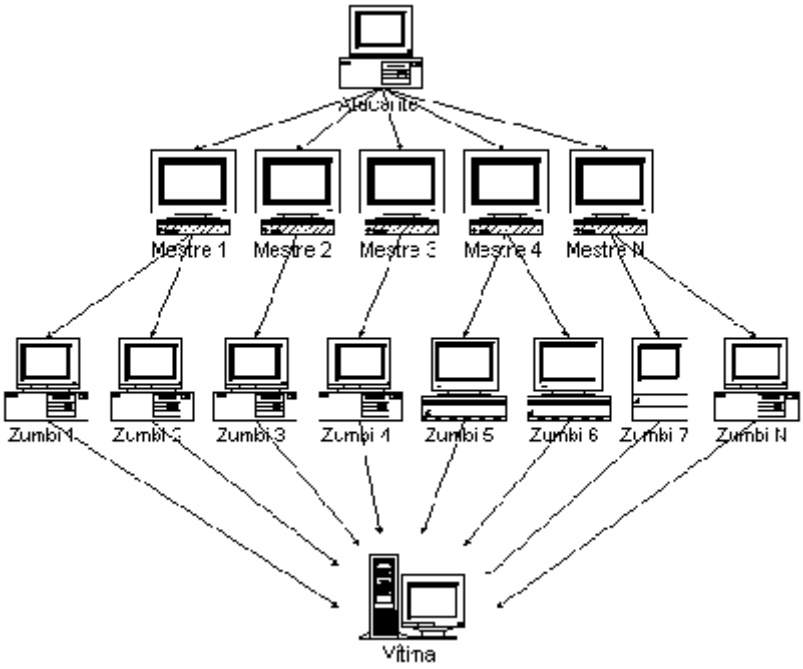


Fig. 1.3 – Rede de ataque DDoS

Algumas anomalias podem sinalizar a ocorrência deste tipo de ataque. Abaixo, algumas delas serão abordadas:

- **Excesso de tráfego**: A banda utilizada excede o máximo, ultrapassando o número de acessos esperados.
- **Pacotes UDP e ICMP de tamanho acima do normal**: Geralmente as sessões UDP utilizam pacotes pequenos de dados dificilmente maiores que 10 bytes (payload). As mensagens ICMP não excedem a faixa entre 64 e 128 bytes. Pacotes cujo tamanho seja superior a esses números são considerados suspeitos de conterem mensagens de controle, destinadas a cada um dos agentes que está participando do ataque. Apesar do conteúdo dos pacotes estar cifrado, o endereço do destino é verdadeiro, desta forma pode-se localizar um dos agentes que estão realizando o ataque baseado no seu fluxo de mensagens.

- **Pacotes TCP e UDP não fazem parte de uma conexão**: Alguns tipos de DDoS utilizam aleatoriamente vários protocolos (incluindo protocolos orientados a conexão) para enviar dados sobre canais não orientados a conexão. Isto pode ser detectado utilizando-se “firewalls “ que mantenham o estado das conexões (statefull-firewalls). Outro ponto importante é que estes pacotes costumam destinar-se a portas acima de 1024.

- **Os tipos de pacotes devem ser analisados:** Quando os dados de pacotes recebidos estiverem estritamente em formato binário e seu destino for diferente do das portas de ftp ou http, estes devem ser descartados.

Os ataques DoS atuam nas fraquezas dos protocolos TCP/IP, sendo possível de ser implemetado em qualquer dispositivo que utilize este protocolo, como servidores, clientes e roteadores. Um exemplo comum de ataque DoS é “TCP SYN”. De forma simplificada, toda vez que um cliente deseja iniciar uma conexão com um servidor, envia um pedido, utilizando o bit de SYN (sincronização) do pacote TCP. O servidor recebe o pacote e devolve um ACK para iniciar a conexão. A cada pedido gerado pelo cliente, o servidor aloca recursos internos, como memória, para poder atender a solicitação. Se um número muito grande de solicitações foram feitas, o servidor pode simplesmente parar ou ficar tão sobrecarregado que recuse novas conexões. Geralmente, o endereço de origem do cliente que solicita a conexão é forjado (spoofed) para tornar o ataque ainda mais efetivo e encobrir o seu autor. Além do TCP SYN Attack, existem vários outros ataques conhecidos do tipo DoS, como UDF Flood Attack (excesso de requisições UDP em um curto intervalo de tempo), Smurfing (Técnica de Multiplicação de pacotes que utiliza BroadCast como fonte). Existe uma série de utilitários disponíveis na Internet que facilitam estes ataques, tornado-os bastante simples de serem implementados, inclusive por amadores. Um ataque Distributed DoS nada mais é que um ataque DoS em larga escala, utilizando uma dezena, centena ou milhares de sistemas ao mesmo tempo no ataque de um ou mais alvos. Este tipo de ataque é considerado como sendo de alto risco e de difícil defesa. Durante o ano de 1999, foram encontradas

algumas ferramentas que automatizam os ataques DDoS, como TFN (Tribe Flood Network), Trinoo, stacheldraht (arame farpado em alemão) e o TFN2000 (TFN2K). Em agosto de 1999, as ferramentas TFN e Trinoo foram infiltradas em 27 máquinas da Universidade de Washington e utilizada em um ataque de três dias a Universidade de Minnesota, juntamente com mais outras 200 máquinas infectadas. Enquanto o Trinoo foi utilizado para múltiplos ataques DoS do tipo UDP Flood, o TFN gerava diferentes tipos de ataques DoS, inclusive com endereços forjados (IP spoofing). [LV-04]

Recentemente foram encontradas versões de ferramentas DDoS para ambientes Windows, tornando qualquer usuário da Internet um potencial cliente. Geralmente os clientes invadidos são escolhidos em função de sua capacidade de comunicação. Quanto maior a conexão de rede, maior será o volume de pacotes gerados na vítima. Isto faz com que os grandes provedores de acesso sejam um alvo preferencial para servir de cliente. Como os sistemas comprometidos estão sob controle externo, existe uma grande dificuldade do administrador do site atacado responder ao DDoS.

Estamos assistindo apenas ao começo de uma série de ataques que devem crescer em número e poder de ação. Cabe aos responsáveis pela segurança, estarem atualizados sobre as novas modalidades de ataques e como responder a eles.

1.1.2 Vulnerabilidades

A conexão de computadores em redes criou inúmeras portas de acesso aos sistemas computacionais. Desta forma torna-se realmente fácil encontrar um acesso que esteja desprotegido. Quanto mais se aumenta a complexidade das redes, quanto mais recursos são disponibilizados aos usuários, quanto mais informação é requerida por este usuário, mais difícil se torna garantir a segurança dos sistemas de informação. Este problema é agravado cada vez mais pela pouca relevância dada ao assunto em relação a avassaladora necessidade de novidades despejada nos usuários individuais que transportam as condições de insegurança vividas na computação pessoal para a computação corporativa, quando na realidade o caminho inverso é que deveria ser percorrido, com os usuários de computação

doméstica levando consigo todos os procedimentos de segurança adotados nas corporações.

Como as organizações, sejam elas públicas ou privadas, perceberam que se tornaram vulneráveis, tem-se procurado, em alguns casos, recuperar o tempo perdido implementando metodologias e ferramentas de segurança, sendo que o grande dilema desta questão é a criação de um ambiente controlado e confiável, mas que não tire do usuário a agilidade proporcionada pela micro informática nos últimos anos.

A tendência de “esquecimento” dos procedimentos de segurança, até que ocorra algum problema grave, é muito comum nos ambientes denominados “cliente-servidor”. Para tanto deve-se adotar políticas de segurança que determinem quais itens devem merecer atenção e com quais custos, sendo que de qualquer maneira vale a premissa: “Um ambiente totalmente seguro depende da aplicação de recursos ilimitados”.

Um diagnóstico simples para o problema pode ser feito observando as ocorrências de vírus. Se os computadores e as redes da organização sofrem de infecções virais, com certeza as informações presentes nesta rede, sejam públicas ou confidenciais estão sujeitas e vulneráveis a vazamentos, alterações indevidas e perdas. Da mesma forma a não existência de vírus garante que estas mesmas informações estão bem protegidas e invulneráveis.

Em geral, sistemas inseguros existem por três motivos: por desconhecimento (na maioria das vezes extremamente conveniente), por negligência ou por uma decisão dos níveis estratégicos das organizações em não adotar a segurança. É preciso conhecer os riscos, saber quais as consequências da falta de segurança, identificar os pontos vulneráveis e determinar uma solução adequada para a organização. O primeiro passo para isso é avaliar o valor do bem e/ou recurso a ser protegido e sua importância para a organização, o que ajuda a definir quanto vale a pena gastar com proteção. A análise do problema deve abordar três aspectos fundamentais: confidencialidade, integridade e disponibilidade, sendo que ninguém melhor que o proprietário da informação para determinar esta relevância.

Os conceitos de confidencialidade, integridade e disponibilidade dizem respeito, respectivamente, a quanto da informação deve ser limitada ou restringida, a correção e certeza que a informação é realmente a verdadeira e à possibilidade de utilização da informação no tempo e local requerido pelos usuários.

Alguns conceitos básicos devem ser internalizados para o desenvolvimento de todo um projeto. Um exemplo que ilustra a importância desta conceituação é relatado a seguir: Um advogado de uma grande organização afirma que: “Minha informação não precisa de segurança, pois ela é pública”. A área responsável pela segurança das informações tem o seguinte argumento: “É pública, mas não pode ser alterada indevidamente”. Este diálogo faz com que o usuário perceba que a confidencialidade é apenas um dos aspectos da proteção de dados. Com uma visão mais ampla do problema, usuários que imaginam não ter responsabilidades no quesito segurança das informações passam a ter uma consciência diferente em relação ao tema.

Um projeto de segurança sempre depende das características de cada organização, como seu ramo de negócios, o grau de importância das informações, o grau de dependência da empresa em relação aos seus computadores, dentre outras. As ações práticas podem ir desde a instalação de um sistema simples, que solicita senha para utilizar o microcomputador, até o uso de equipamentos onde apenas algumas aplicações são executadas e todas as operações são monitoradas.

Atualmente os ataques chamados de DoS (Denial Of Service) e os ataques de DDoS (Distributed Denial of Service) estão incomodando administradores de grandes redes WANS.

A segurança possui muitas fases e uma das mais importantes é a capacidade de controlar o fluxo de pacotes em uma rede, com o objetivo de protege-las de falhas, degradação dos serviços, roubo ou comprometimento dos dados resultantes de uma ação intencional ou de um erro provocado por usuários. Mas uma solução efetiva de segurança não deve ser baseada somente em recursos técnicos, deve-se elaborar uma política de segurança de forma a definir-se as diretrizes de segurança da instituição. Para tanto,

existem práticas de segurança, que podem auxiliar no processo de elaboração da política de defesa.

Antes de prosseguir este trabalho falaremos um pouco sobre roteamento, já que nossas simulações de ataques irão visar justamente congestionar o fluxo de dados nos roteadores envolvidos na suposta vídeo conferência proposta.

O controle de acesso é a forma pela qual pode-se controlar quem tem acesso aos servidores da rede e a quais serviços pode-se utilizar uma vez possuindo acesso aos mesmos:

- **A**utenticar - A autenticação é o método de identificação dos usuários que podem utilizar os recursos da rede;
- **A**utorizar - A autorização é o método de controle de acesso remoto;
- **A**uditar - A auditoria é o método de coletar as informações sobre os acessos, utilização dos recursos, tentativas de acesso falhos, horário de início de término de determinadas transações, número de pacotes enviados por protocolo entre outras;

Neste tópico iremos dar ênfase a AUTORIZAÇÃO de dados. Para tanto, iremos tratar dos recursos de controle de acesso que podem ser implementados em roteadores , embora os conceitos possam ser aplicados a outros elementos de controle de acesso. Deve-se ter em mente que um sistema efetivo de segurança não deve ser baseado somente em regras nos roteadores, deve-se utilizar outros elementos como: Firewall, ferramentas de controle de acesso (ssh por exemplo), segurança dos hosts, preservação da analise dos logs e políticas de segurança utilizadas.

Uma vez que o roteador sabe qual é ou quais são os tipos de interfaces (Ethernet, Token Ring, FDDI, X.25, Frame Relay...), o mesmo pode verificar o formato dos frames que chegam e montar os frames de saída. Além disso, o roteador pode verificar a integridade dos dados que chegam, pois, como o mesmo conhece o tipo de interface, pode calcular o *cyclic redundancy check* (CRC); da mesma forma, o roteador pode calcular o CRC dos *frames* de saída. Caso as tabelas de roteamento possuam apenas rotas estáticas, estas tabelas não serão trocadas com outros roteadores. O *cache* ARP representa uma área da memória onde são armazenadas as relações entre o endereço IP e seu endereço físico (o endereço MAC da camada 2). [LV-04]

Os dados que são recebidos ou preparados para transmissão podem entrar em filas de prioridades, onde o tráfego de baixa prioridade é atrasado em favor do processamento do tráfego de alta prioridade. Se o modelo do roteador suportar priorização de tráfego, certos parâmetros de configuração podem ser informados ao roteador para indicar como realizar esta priorização. As informações sobre o fluxo dos dados como localização e *status* dos pacotes são armazenadas na *Fila de espera*. As entradas das tabelas de roteamento informam a interface de destino para o qual determinados pacotes devem ser roteados. Se o destino for uma LAN e for necessária a resolução de endereço, o roteador procura o endereço MAC inicialmente no *cache* ARP. Caso o endereço não seja encontrado no *cache* ARP, o roteador monta um pacote ARP para descobrir o endereço MAC.

Uma vez que o endereço de destino e o método de encapsulamento estão determinados, os pacotes são enviados para a porta da interface. Dependendo do volume de tráfego, novamente o pacote pode entrar em uma fila de prioridade, até que possa ser enviado. Umas das formas mais freqüentes de se tirar de serviço um roteador, é justamente congestionar suas filas de modo que ele não consiga suprir a quantidade de requisições.[LV-04]

1.1.3 Segurança na RNP

A Rede Nacional de Pesquisa (RNP) está constituída nos 27 estados brasileiros, conectando milhares de computadores em mais de 800 instituições em todo o país. Com o crescimento da Internet e o conseqüente crescimento de problemas de segurança na rede, foi criado, em maio de 1997, o CAE - Centro de Atendimento a Emergências, que mais tarde mudaria o nome para CAIS (Centro de Atendimento a Incidentes de Seguranca).

"Este grupo tem como missão registrar e acompanhar incidentes de segurança que envolvem redes conectadas ao backbone da RNP, incluindo o auxílio à identificação de invasões e reparo de danos causados por invasores. Além do seu trabalho de resposta a incidentes de segurança, o CAIS tem também por objetivo desempenhar um papel preventivo, disseminando informações, alertas e recomendações na área de segurança

em sistemas e redes", diz a Gerente do CAIS, Liliana Velasquez Solha. A equipe do CAIS conta, atualmente, com quatro analistas de segurança de forma integral e exclusiva. Para se ter uma idéia da importância do trabalho realizado por estes profissionais, são reportados mensalmente cerca de 850 incidentes de segurança, no entanto, no último mês de maio foram notificados em torno de 1.400 incidentes. Este aumento deveu-se basicamente à propagação do worm Spike. SQL,

Como verificamos na figura 1.4, podemos observar que a RNP possui velocidades consideráveis de links, o que a torna visível para os “hackers”, pois estes podem vir a usar estes links, através de invasões, para proporcionar outros ataques de negação de serviço, por exemplo .[ST-01]

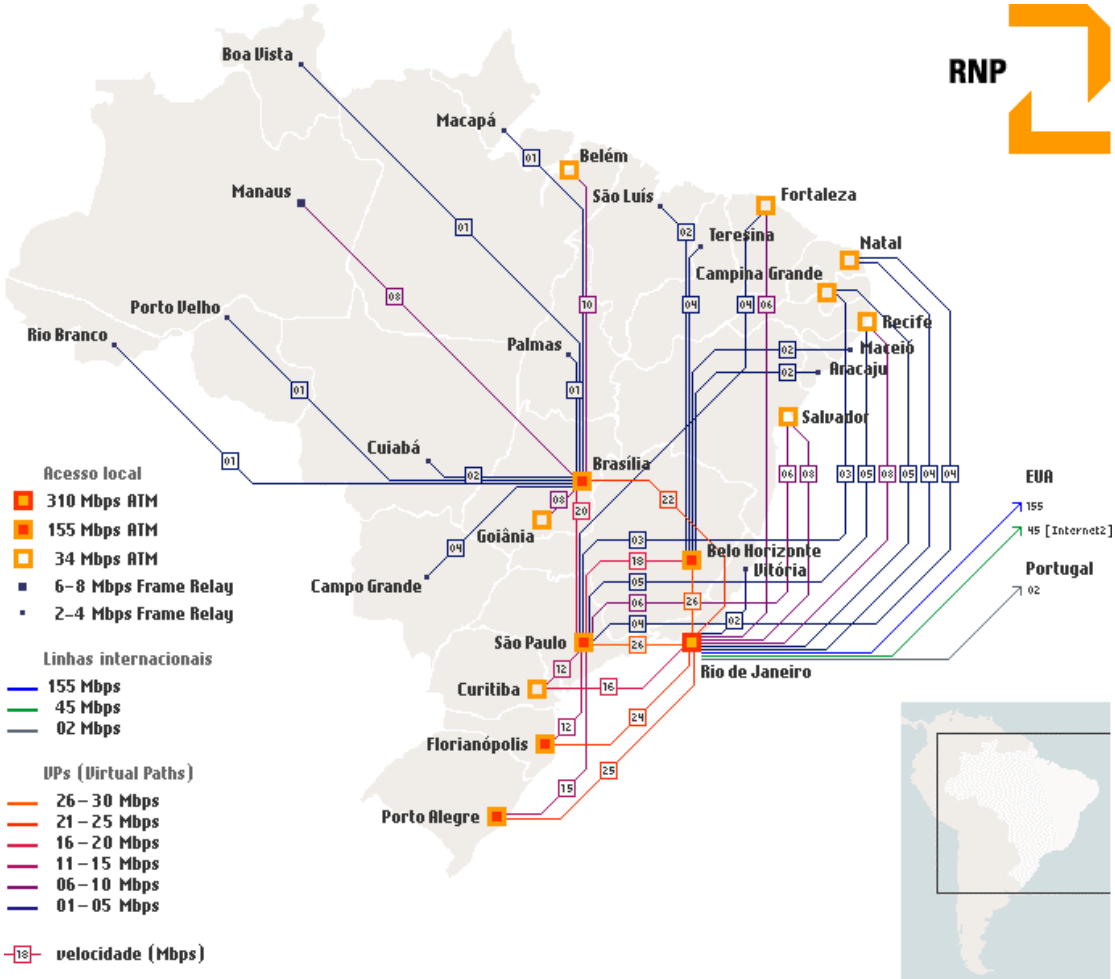


Fig. 1.4 – Backbone da RNP[ST-01]

1.1.4 Tipos de Ataques mais Freqüentes

Um tipo de ataque é o "spoofing". Neste caso, pacotes expúrios com informações erradas sobre roteamento são enviados a um ou mais roteadores fazendo com que eles roteem errado. Este ataque difere do ataque por flood ataque somente no propósito do roteamento. No ataque por flood, o objetivo era deixar o roteador inutilizável, um estado facilmente detectado pelos usuários da rede. No "spoofing" os pacotes são enviados a algum host onde eles podem ser monitorados e depois reenviados a seu destino correto, tendo seu conteúdo alterado ou não.

A solução para a maioria desses casos é proteger os pacotes de atualização de roteamento dos protocolos como RIP-2 e OSPF. Existem 3 níveis de proteção: senha textual, checksum criptografado e criptografia. Senhas textuais oferecem proteção mínima contra intrusos que não têm acesso à rede física. Também protegem roteadores mal configurados (ou seja, roteadores ainda não configurados e que não deveriam rotear pacotes). A vantagem de senhas é que elas têm um baixo overhead, tanto em tempo de transmissão como de CPU. Checksums protegem contra a injeção de pacotes daninhos, até mesmo se o intruso tem acesso à rede física. Combinado com um número de seqüência, ou outro identificador único, o checksum pode também detectar ataques de retransmissão, onde uma informação mais antiga está sendo retransmitida, seja por um intruso ou roteador mal configurado. O melhor é prover criptografia completa, de atualizações de roteamento seqüenciados ou univocamente identificados. Isso impede um intruso de determinar a topologia da rede. A desvantagem da criptografia é o overhead envolvido no processamento.

Tanto RIP-2 (RFC 1723) como OSPF (RFC 1583) suportam senhas textuais nas suas especificações básicas de projeto. Existem extensões para cada protocolo suportar o algoritmo de criptografia MD5.

Infelizmente não há proteção adequada a um ataque por "flooding" ou contra um host ou roteador que esteja inundando a rede, mas felizmente, esse ataque é óbvio quando ocorre e, geralmente, pode ser terminado de forma relativamente fácil. [LV-01]

2. Contextualização do Ambiente de Testes

Neste capítulo, detalharemos o cenário onde se passará a vídeo conferência, bem como as especificações correntes usadas para efetuar as simulações e obter os resultados.

2.1 Situação

Este trabalho pressupõe a realização de um vídeo conferência entre importantes pólos políticos de Manaus, Rio Branco, Palmas e Porto Velho. Um evento ocorrerá utilizando-se da infra-estrutura da Internet e será gerenciado por um refletor multicast da VCON (empresa mexicana de vídeo conferência) localizado em Brasília. Nesse dia, um grupo de hackers lançará um ataque sincronizado, feito por um worm, dos seguintes pontos em direção a Brasília: Belém, Manaus, Campo Grande e Cuiabá. Esse ataque, com conotação de protesto político, terá como finalidade tornar indisponível a rede gerando um super congestionamento e provocando uma negação de serviço (DoS). Considerar-se-á que os hackers descobriram que as portas que aceitam tráfego CBR (Constant Bit Rate) estarão abertas durante o evento e se valerão delas e de outras para lançar o ataque. O objetivo do trabalho é avaliar, via simulações, o desempenho de políticas de fila durante o ataque e sugerir alternativas para minimizar seu impacto.

2.2 Cenário

A figura 2.1 nos mostra como está disposto no mapa do Brasil o ataque que será simulado com os nós de ataque e os que serão atacados.

Fig. 2.1 – Cenário de Ataque

2.2.1 Tráfego

A figura 2.2 nos mostra os enlaces envolvidos na vídeo conferência e seus respectivos parâmetros:

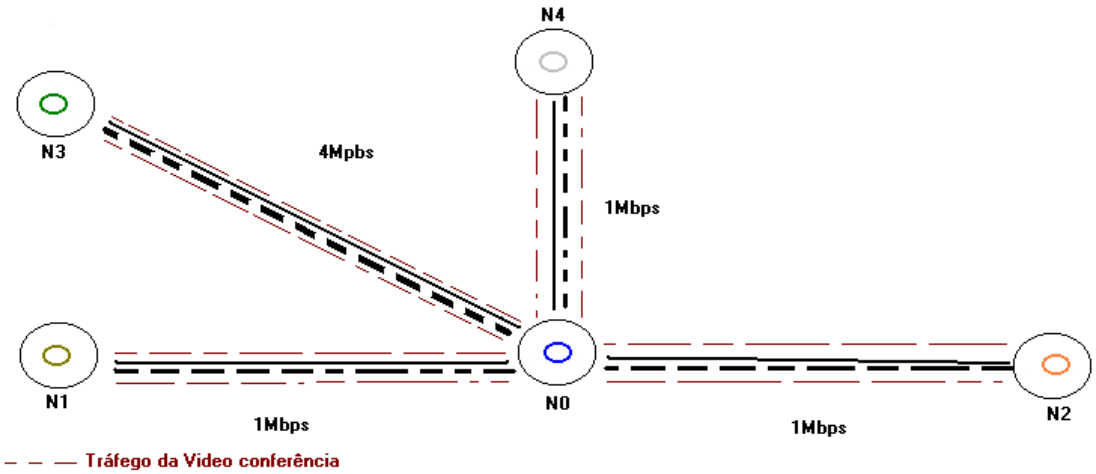


Fig. 2.2 – Tráfego da vídeo conferência

Os testes utilizam atraso de 20ms em todos os links. Serão avaliadas ação e reação do roteador de Brasília (descartando pacotes) diante de duas políticas de fila FIFO (Droptail), e SFQ (Stocasthic Fair Queue). No momento “0” inicia-se a simulação, no momento “0.1” se inicia a vídeo conferência, no momento “0.5” Manaus inicia o ataque a vídeo conferência. Em seguida, no momento “1”, Cuiabá e Campo Grande iniciam simultaneamente seus ataques e, finalmente, no momento “2”, Belém dispara seus ataques, o que sobrecarrega o roteador de Brasília e inviabiliza a vídeo conferência.

2.4 Ferramentas Utilizadas (NS – Network Simulator)

O NS (Network Simulator) é a implementação de uma máquina de simulação extensível orientada a eventos discretos (Open Source) originado a partir do projeto VINT (Virtual Internetwork Testbed) que possui interligações em ambientes multi-protocolos com o objetivo centrado em pesquisas em redes. Ele fornece um suporte substancial para simulações de TCP, roteamento, e protocolos multicast. Seu desenvolvimento é um trabalho de cooperação com o VINT project com a participação de allgumas entidades com o a AT&T Research, USC/ISI, Xerox, PARC e Eth Tik.

O Simulador funciona baseado em scripts em TCL, porém seu núcleo foi desenvolvido em C++ e grande parcela dos módulos de suporte a tecnologias mais específicas em Otcl – Versão orientada a objetos da linguagem TCL. O seu núcleo pode ser modificado, ou personalizado, de acordo com as necessidades de cada usuário. O NS utiliza para gerar seus gráficos e simulações linguagem de programação TCL. Sua inicialização é através do comando ‘ns <tclscrilpt>’,

+- onde <tclscrilpt> é o nome do script tcl que define o cenário da simulação. É possível também iniciar o NS sem nenhum argumento e adicionar os comandos Tcl no shell Tcl. Tudo no NS depende de um script Tcl. O script pode criar alguma saída na shell, pode escrever um arquivo de rota ou pode inicializar o nam para visualização da simulação.

Basicamente, para se dar inicio a uma simulação como a que foi proposta neste trabalho, necessitamos analisar alguns fatores de configuração de entrada para o simulador: [ST-02]

- Topologia de Rede;
- Modelos de tráfego;
- Geração de testes;
- Avaliação dos resultados;

Utilizando o NS tivemos alguns atributos favoráveis pois o mesmo nos proporciona métodos de geração de topologias.

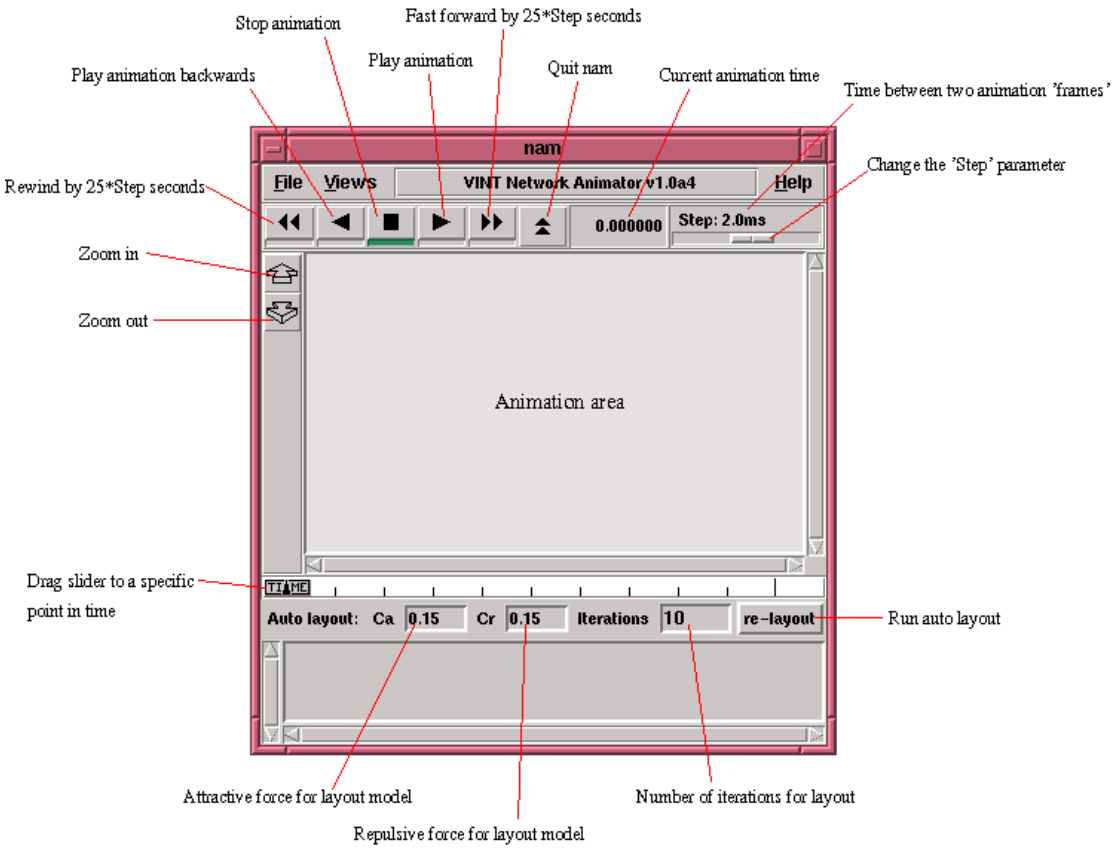


Fig. 2.3 – Network Simulator

2.4.1 Especificações técnicas - NS

O NS possui uma arquitetura de fina granularidade dos objetos definidos em uma estrutura de macro-objetos (Otcl) que permite ao projetista trabalhar no nível mais alto de abstração suportando a flexibilidade desejada, podendo trabalhar em alto nível (simplesmente criando e configurando macro-objetos) em nível médio (modificando o comportamento de um macro-objeto em uma subclasse derivada) ou em baixo nível (introduzindo outros macro-objetos ou objetos híbridos no cenário do NS).

3. Avaliação de Desempenho

Neste capítulo será mostrado os gráficos gerados a partir da simulação para que possamos, em cima destes, fazer as devidas análises bem como sugerir propostas para minimizar os efeitos do ataque.

3.1 Gráficos

Seguem os gráficos gerados, em Excel, em função do tempo e pacoteamento em cima de três diferentes políticas de fila de roteamento.

3.2 DropTail

O algoritmo DropTail é um algoritmo simples de controle de congestionamento. O algoritmo armazena os pacotes na ordem em que eles chegam, e, assim que a rede permitir, envia-os nesta mesma ordem. Nenhuma decisão é feita sobre a prioridade dos pacotes, a ordem de chegada é o determinante do fluxo. No caso da fila atingir seu limite, isto é, a rede congestionar, o algoritmo pode descartar os pacotes que chegarem a partir de então (DropTail). O algoritmo não garante limites quanto à vazão, atraso ou perda de pacotes. É simplesmente o algoritmo padrão de muitas implementações. Na figura 3.2 observamos a disposição da simulação.

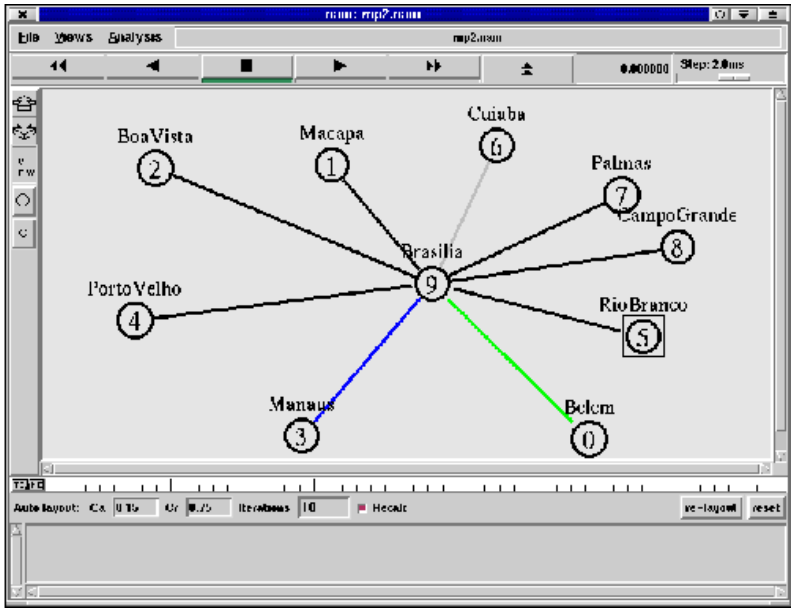


Figura 3.2 – Disposição da Simulação

Na figura 3.3 observamos o início da vídeo conferência.

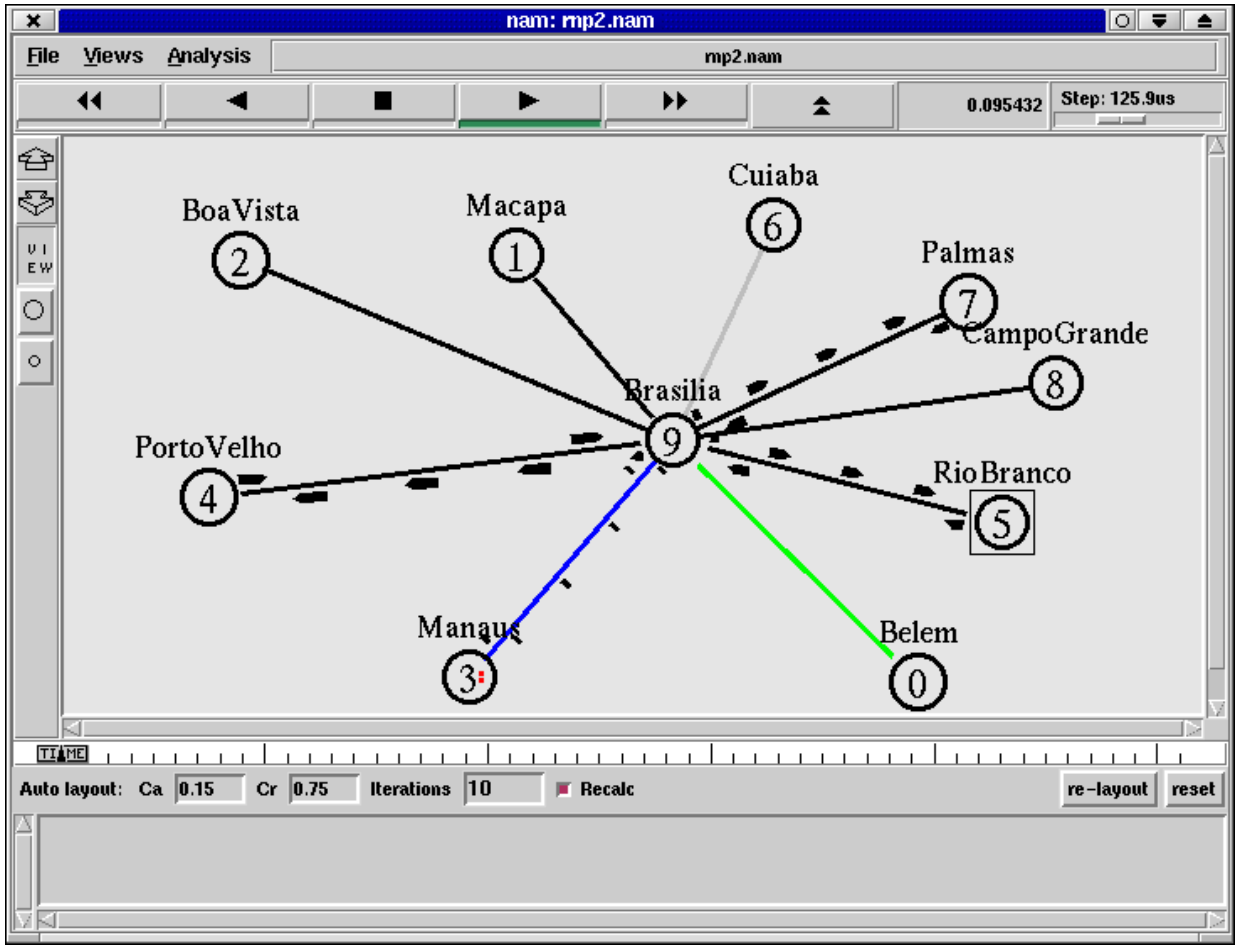


Fig. 3.3 – Início da Simulação

Na figura 3.4 observamos que no momento 0.5 da simulação que Manaus inicia o seu ataque (representado pelas células vermelhas) a vídeo conferência .

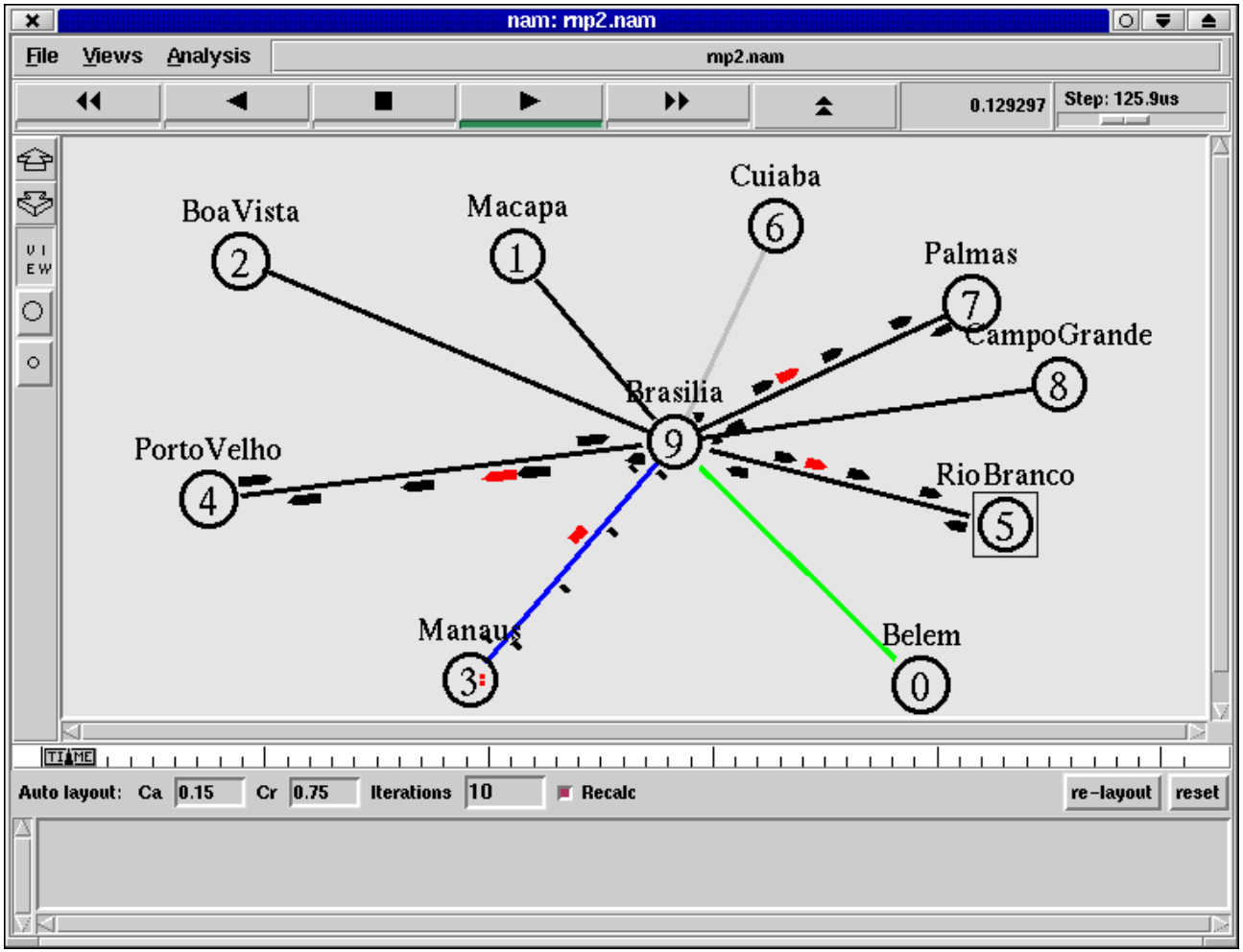


Fig. 3.4 – Início do ataque de Manaus

Na figura 3.5 observamos que no momento 1 da simulação os nós Campo Grande e Cuiabá também iniciam seus ataques espalhando ainda mais pacotes de ping-flood por toda a vídeo conferência já acarretando algum delay na mesma.

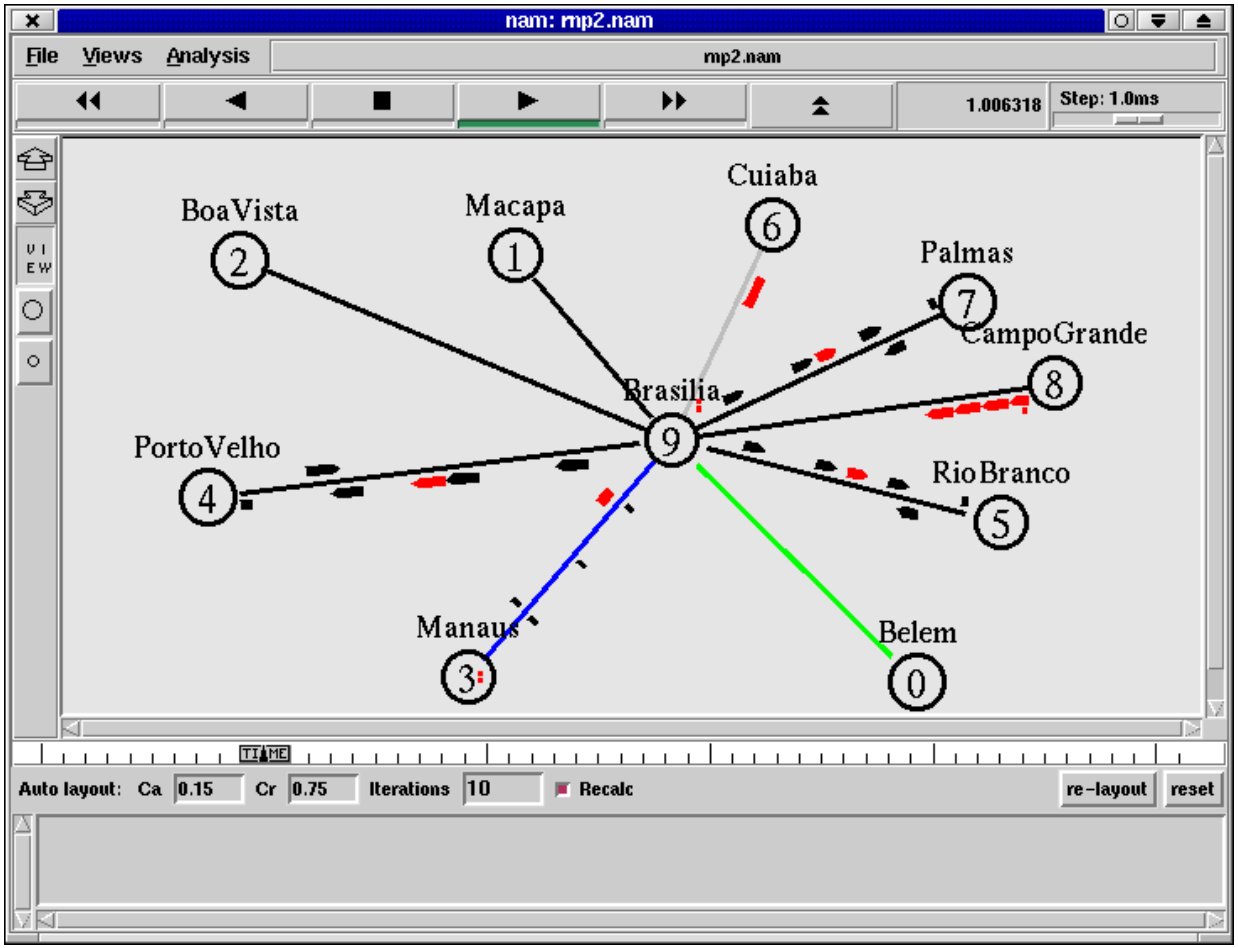


Fig. 3.5 – Cuiabá e Campo Grande atacam

Nas figuras 3.6, 3.7 e 3.8 observamos que quando o nó Belém inicia o ataque, a vídeo conferência começa a sofrer graves conseqüências, pois começamos a notar a formação de filas no roteador em Brasília até que, no momento 2.8 da simulação, observamos o descarte de pacotes (Fig. 3.7) inviabilizando a vídeo conferência.

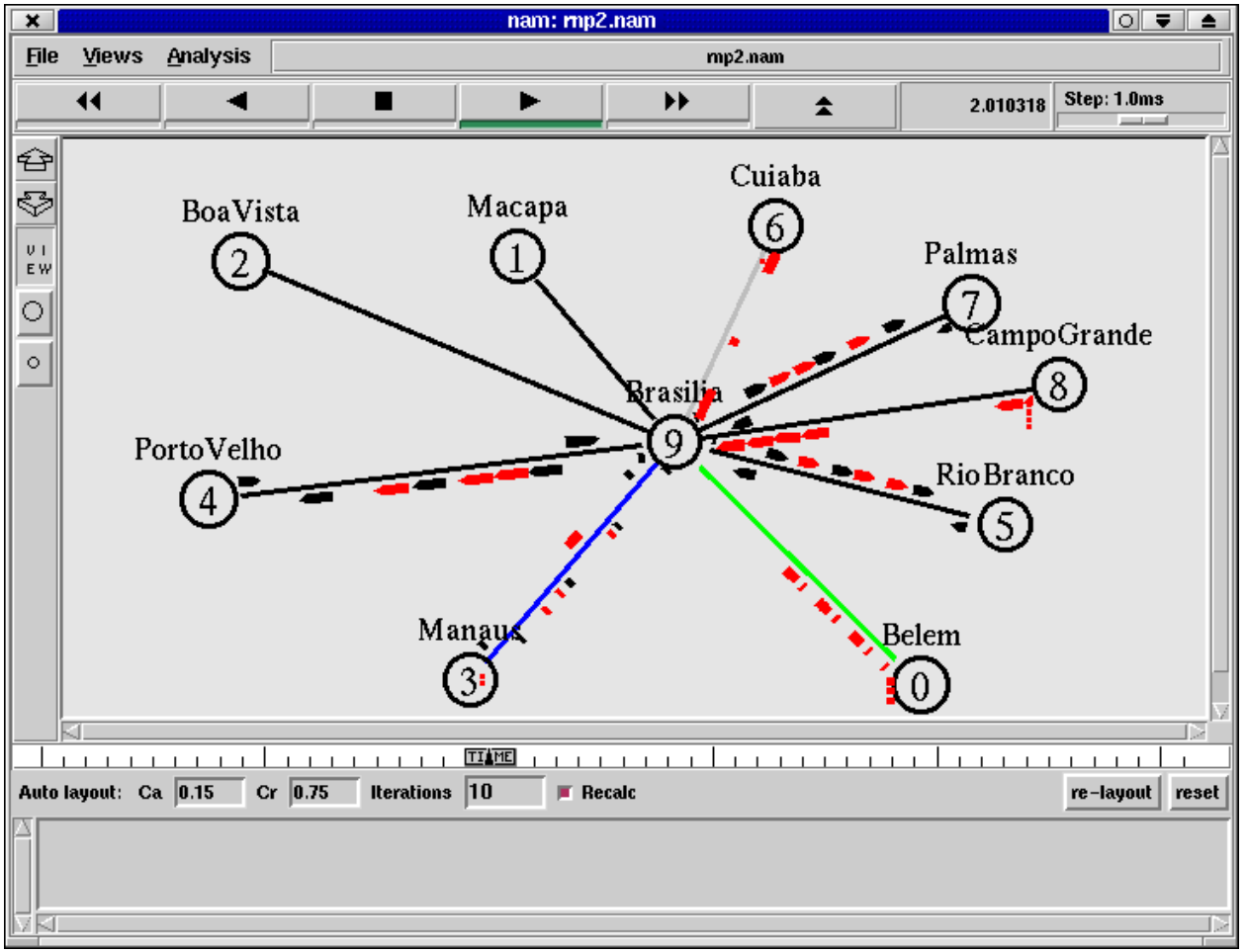


FIG. 3.6 – Início do ataque de Belém

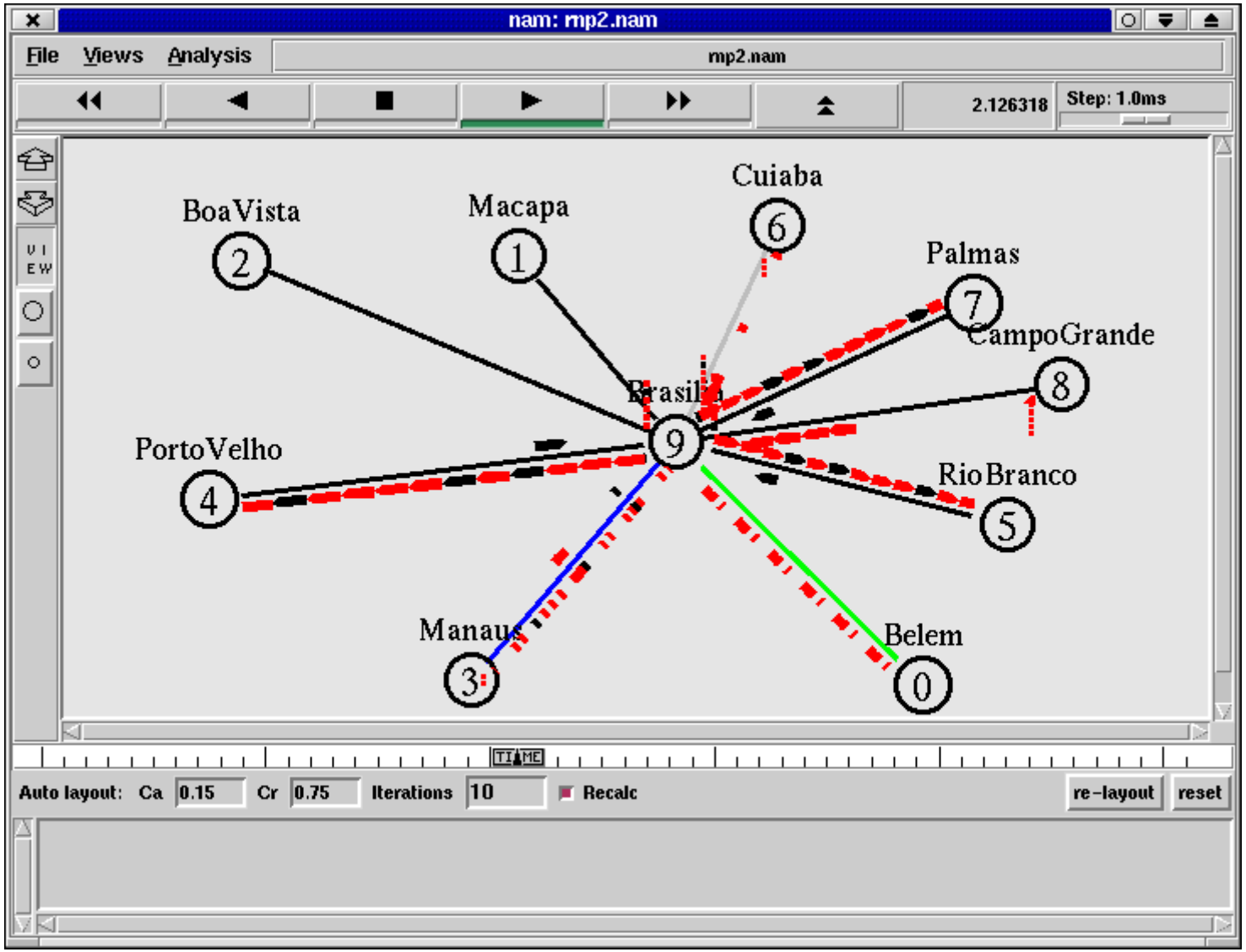


Fig. 3.7 – Contaminação de pacotes Ping-Flood na Vídeo Conferência

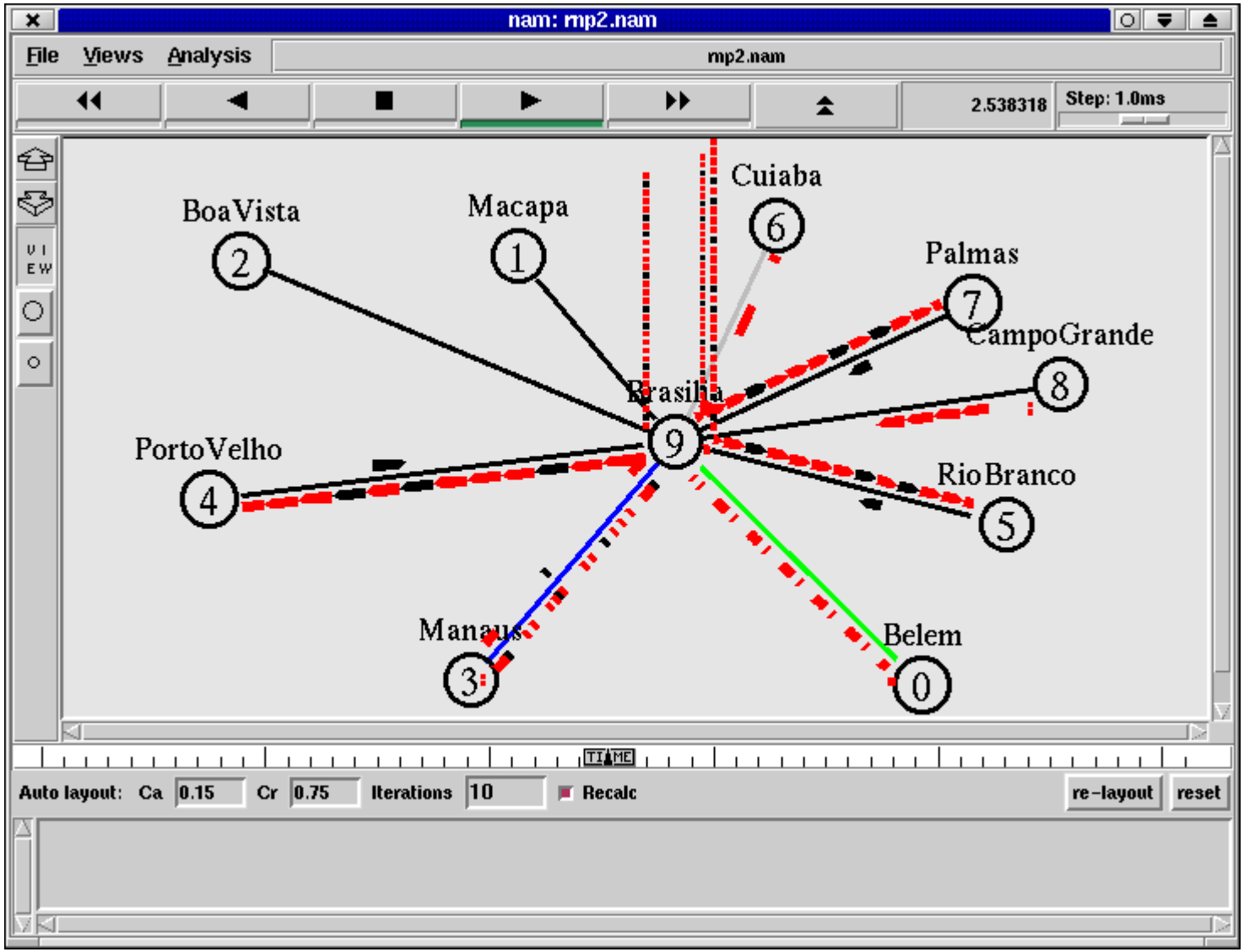


Fig. 3.8 – Criação de Filas no roteador de Brasília

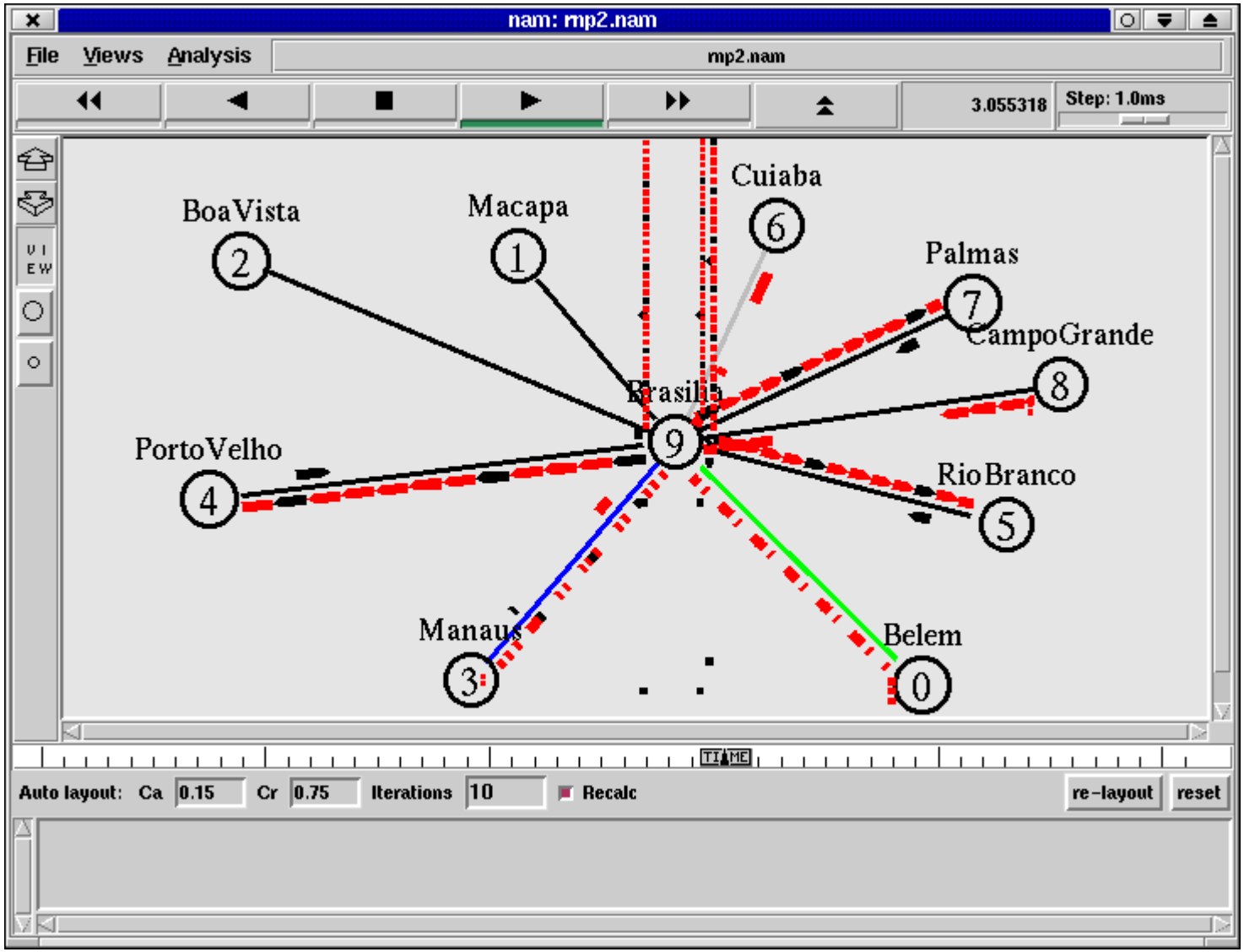


Fig.. 3.9 – Descarte de pacotes da Vídeo Conferência, inviabilizando a mesma

3.3 – SFQ

Proposta por McKenney em 1991. SFQ não é determinística, mas funciona bem (em média). Seus principais benefícios são que ela requer pouco uso da CPU e da memória. É menos precisa que outras implementações de Fair Queueing, mas requer menos cálculos. SFQ usa hashing para mapear os pacotes que chegam no roteador para uma fila FIFO. Entretanto, ao invés de manter uma fila para cada fluxo possível (o que seria intuitivo), o algoritmo utiliza menos filas do que o número de fluxos possíveis. Todos os fluxos que são mapeados para o mesmo valor da tabela hash são tratados de maneira equivalente, o que simplifica a computação, mas implica que esses fluxos que "colidem" na tabela são tratados "injustamente." Se tamanho do índice de hashing for suficientemente maior do que o número de fluxos ativos, a probabilidade de "injustiça" fica muito reduzida. As filas são servidas seguindo um algoritmo de round robin, sem considerar os tamanhos dos pacotes.

Na figura 3.10 observamos a diferença nas políticas de fila. Quando a política usada é a SFQ, além das filas formadas no roteador serem menores, os pacotes que são descartados são os pacotes de flood continuando, assim, a rotear os pacotes de vídeo conferência, sem afetá-la.

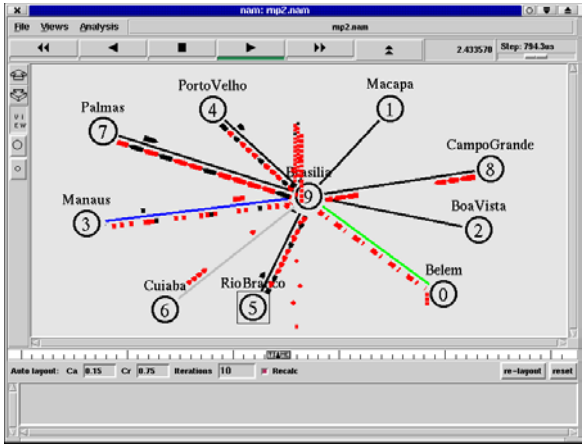
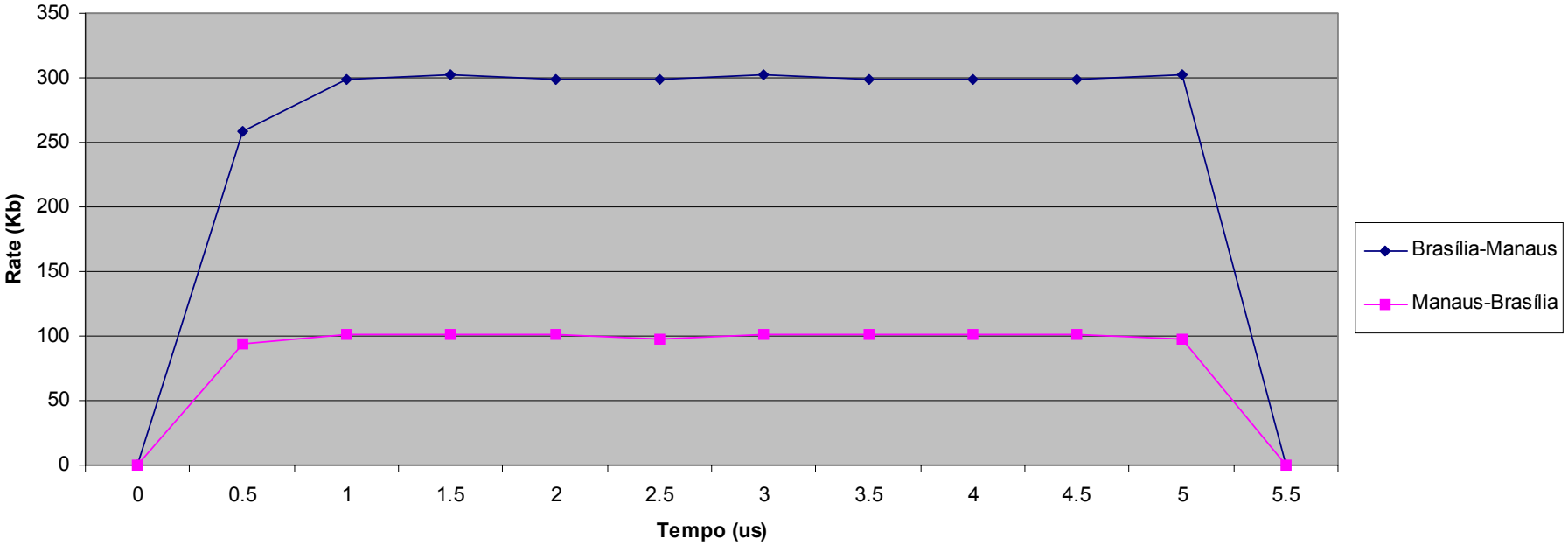


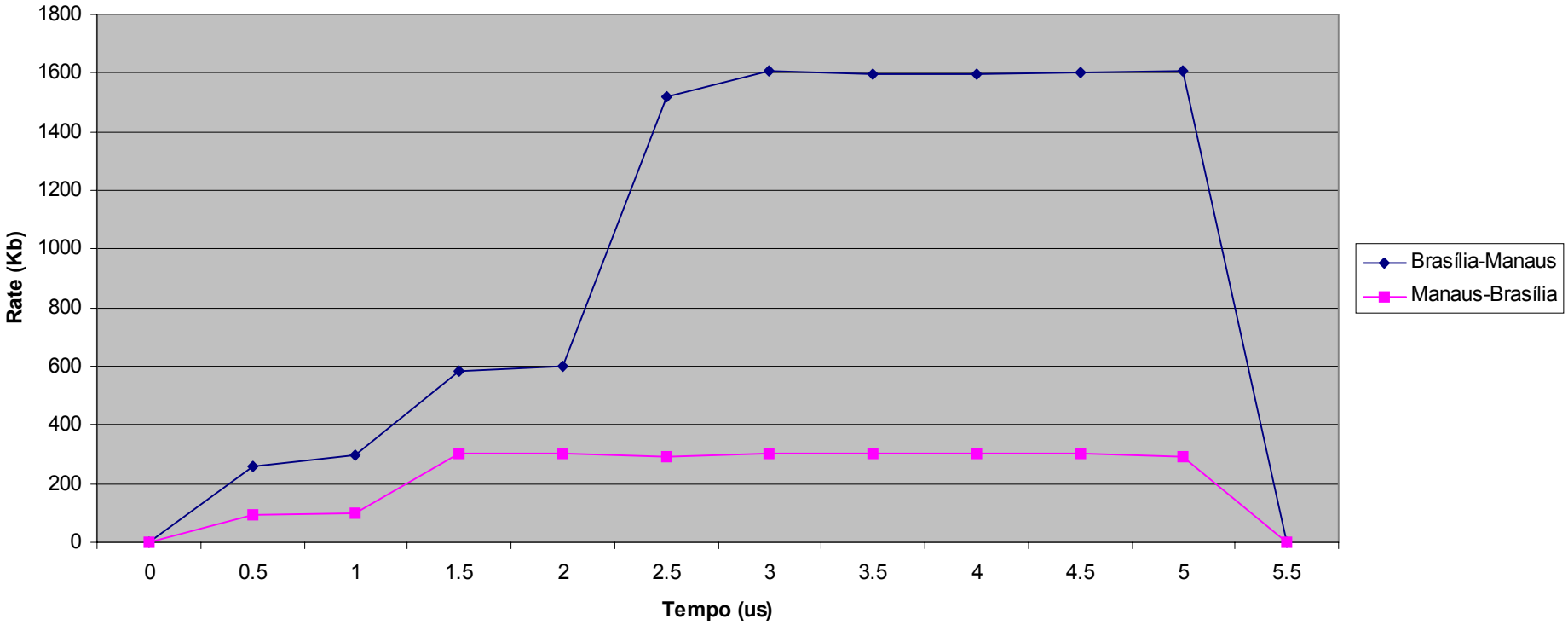
Fig. 3.10 – Fila SFQ

3.4 – DropTail

No gráfico 3.2.1 temos o gráfico mostrando o comportamento de Brasília e Manaus durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.2.2, vemos o mesmo gráfico porém com o ataque.

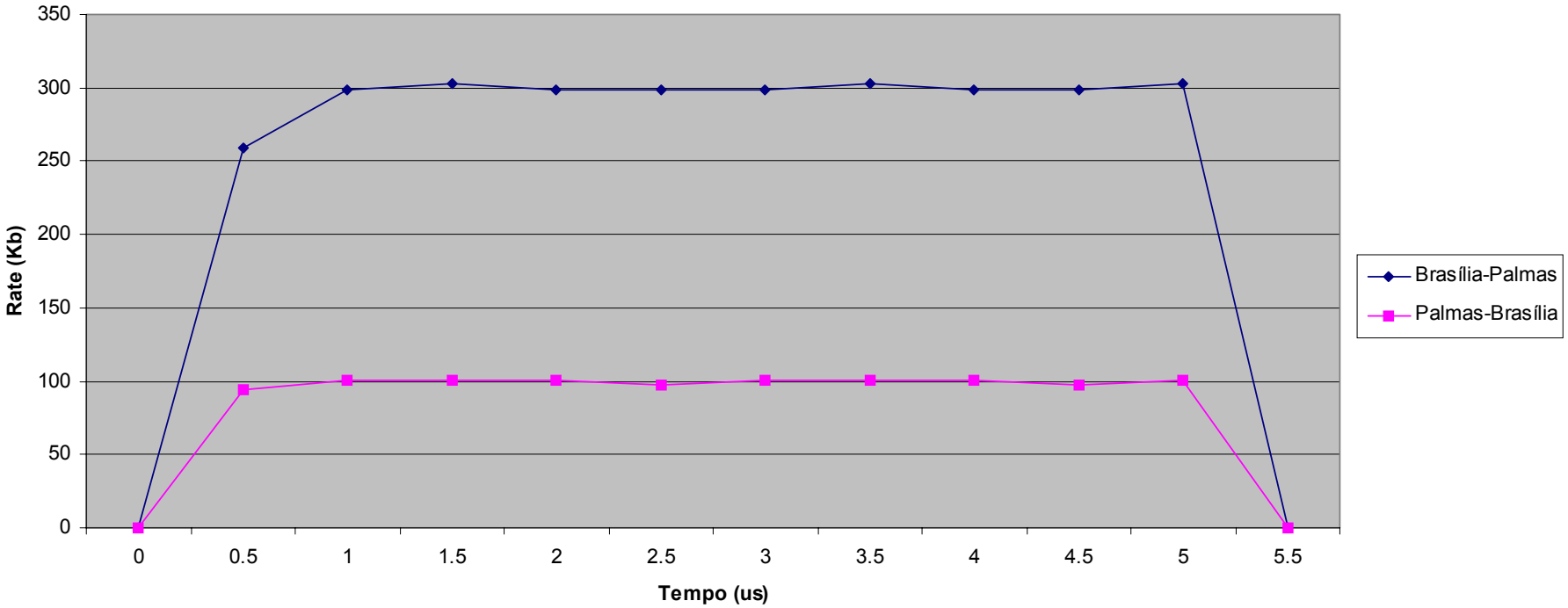


Graf. 3.2.1 – Comportamento de Brasília – Manaus sem os ataques

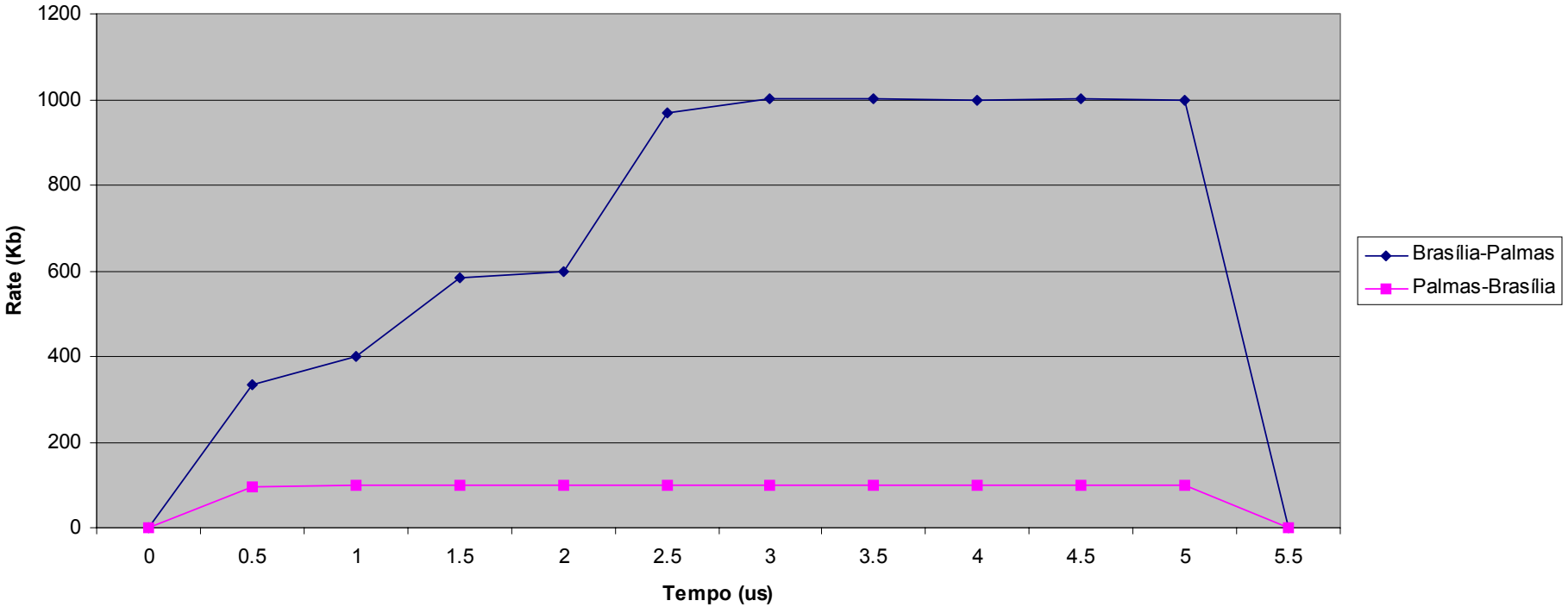


Graf.3.2.2- Comportamento de Brasília – Manaus com os ataques

No gráfico 3.2.3 temos o comportamento de Brasília - Palmas durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.2.4, vemos o mesmo gráfico, porém com o ataque.

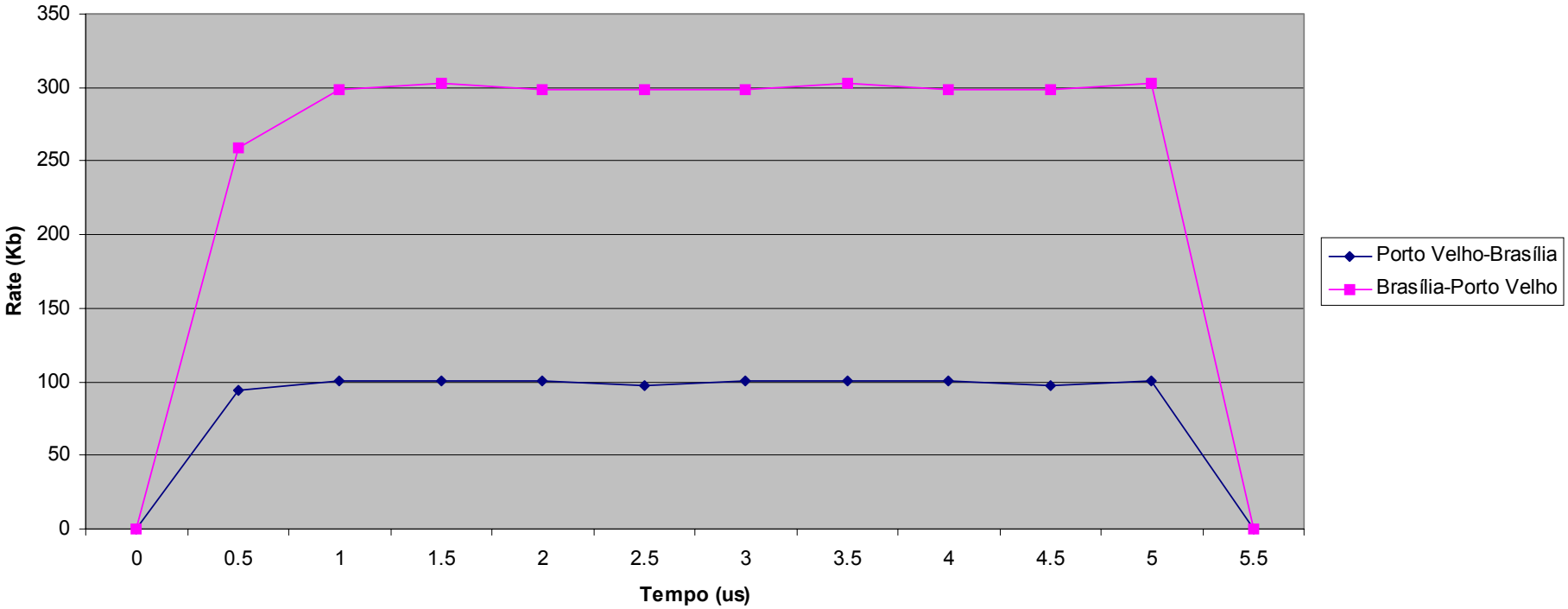


Graf. 3.2.3 - Comportamento de Brasília – Palmas sem os ataques

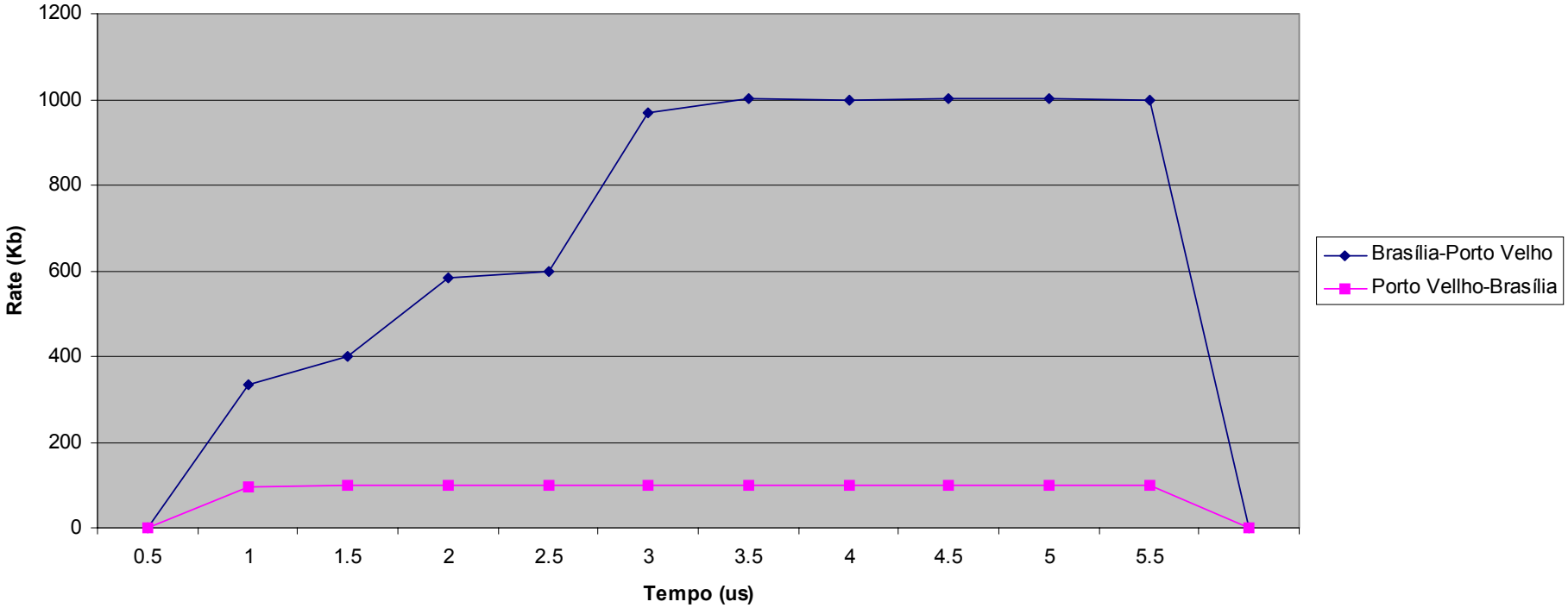


Graf. 3.2.4 - Comportamento de Brasília – Palmas com os ataques

No gráfico 3.2.5 temos o comportamento de Brasília – Porto Velho durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.2.6, vemos o mesmo gráfico agora, com o ataque.

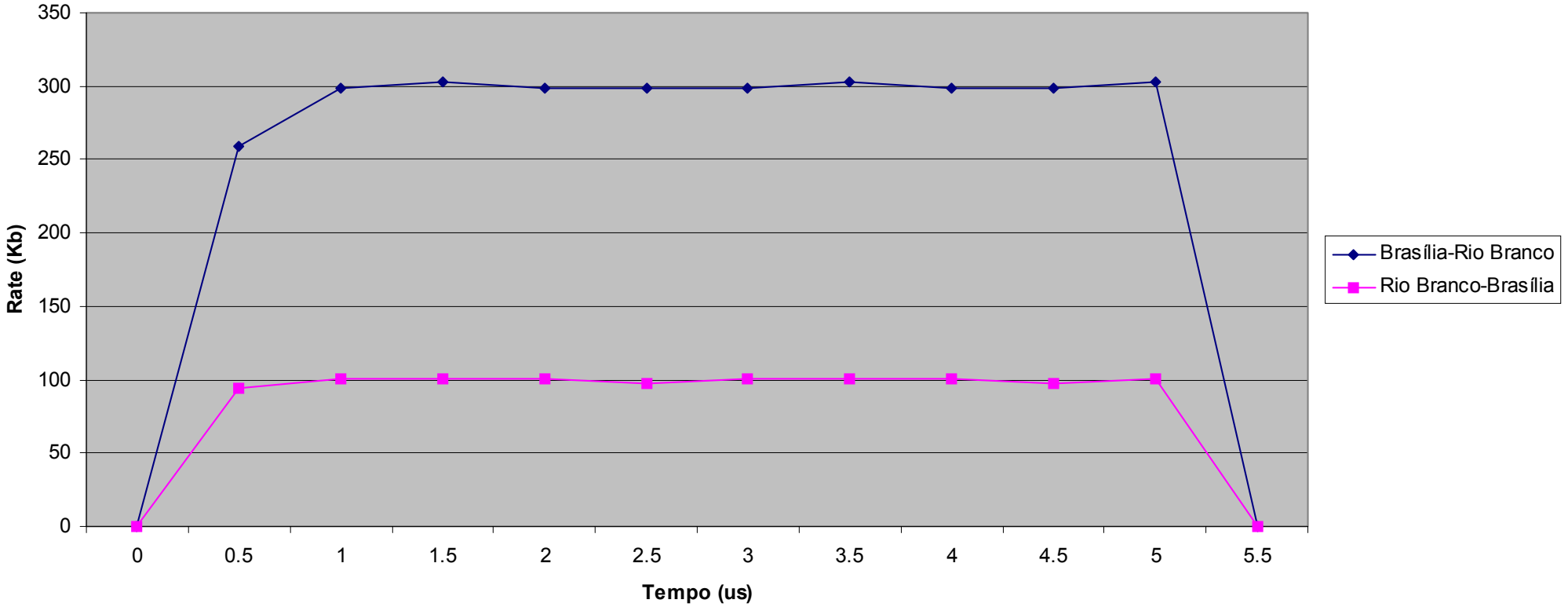


Graf.3.2.5 – Comportamento Brasília-Porto Vellho sem os ataques

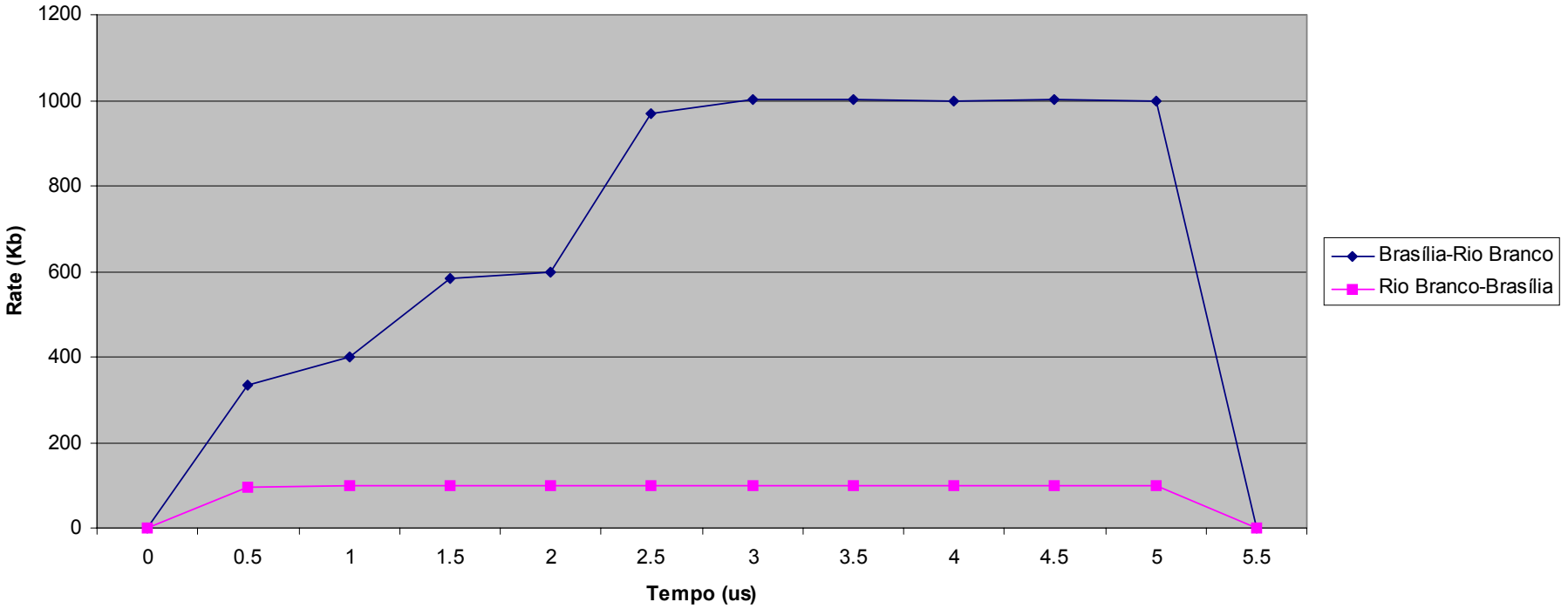


Graf.3.2.6 – Comportamento Brasília-Porto Velho com os ataques

No gráfico 3.2.7 temos o gráfico mostrando o comportamento de Brasília – Rio Branco durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.2.8, vemos o mesmo gráfico agora, com o ataque.



Graf.3.2.7 – Comportamento Brasília-Rio Branco sem os ataques



Graf. 3.2.8 – Comportamento Brasília-Rio Branco com os ataques

Na figura 3.2.9 temos o gráfico gerado com a vídeo conferência utilizando-se da política de fila DropTail com todos os nós juntos e sem os ataques simulados, onde podemos observar que antes da vídeo conferência as larguras de banda permanecem instáveis, ocorrendo apenas um aumento de tráfego com o início da mesma.

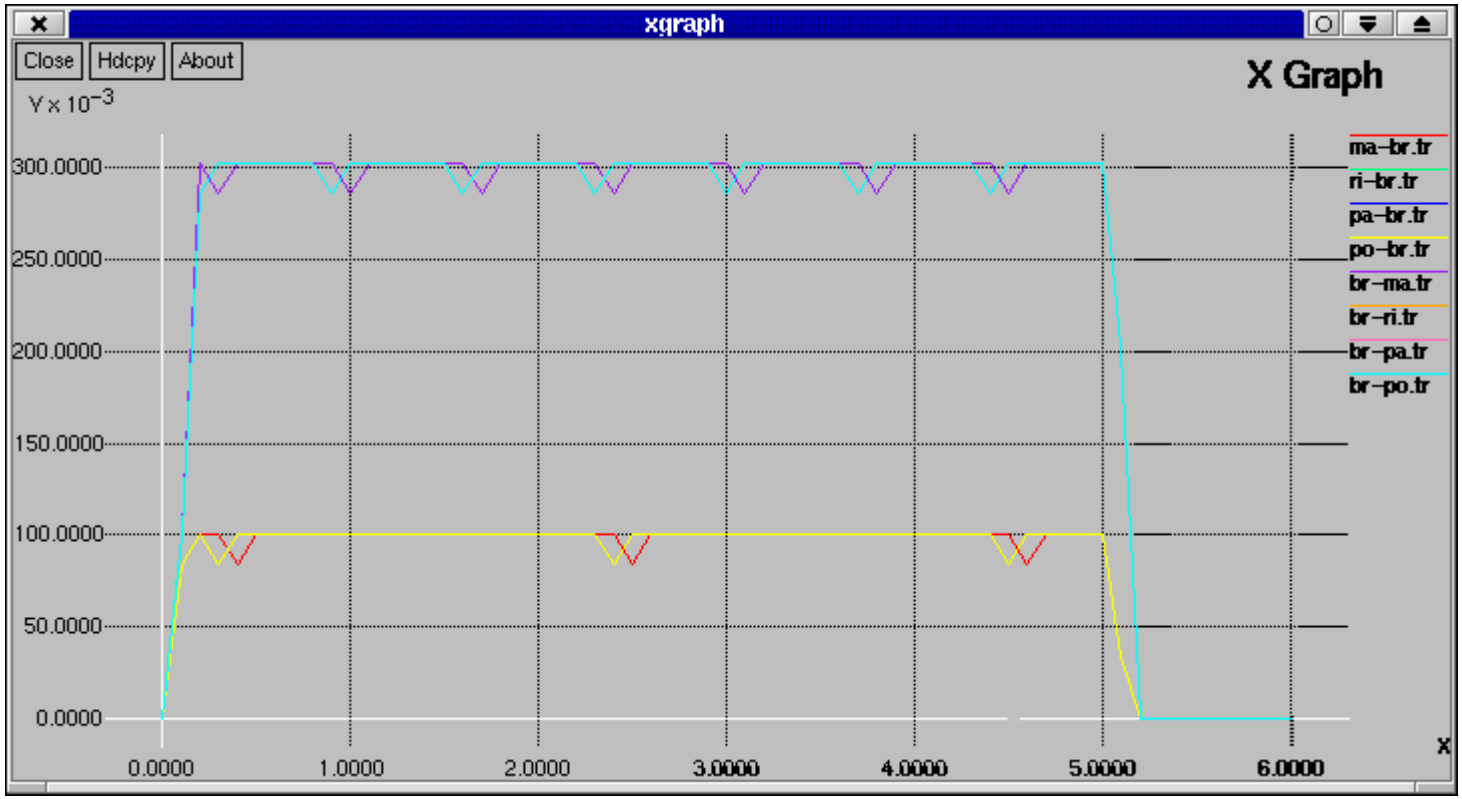


Fig. 3.2.9 – Gráfico DropTail sem ataque

Na figura 3.2.2 observamos a disposição do gráfico da vídeo conferência com todos os nós utilizando-se da política de filas DropTail, porém com os ataques simulados, observando que, mesmo depois do ataque, o tráfego na banda aumenta, porém permanece instável de acordo com a largura de banda de cada nó.

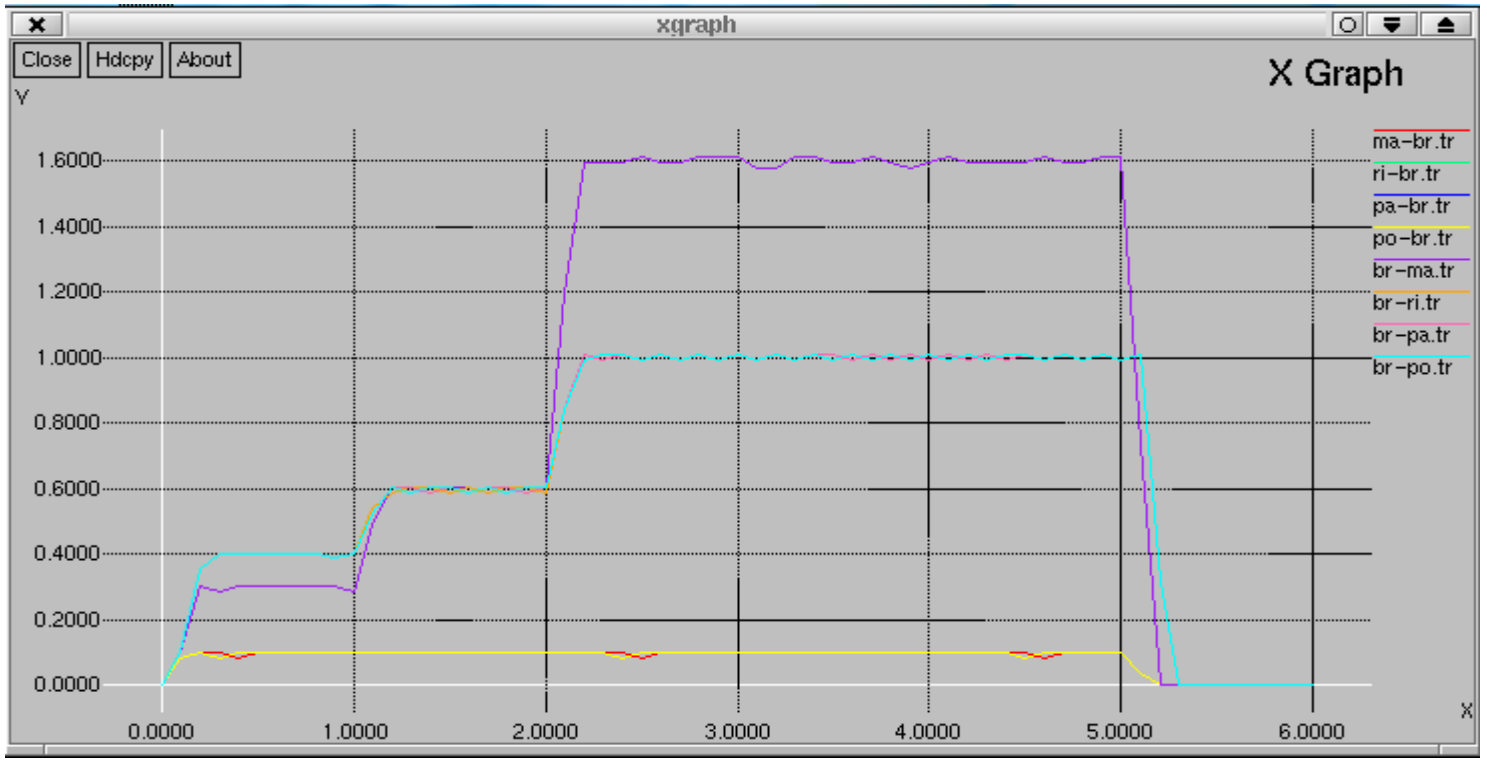
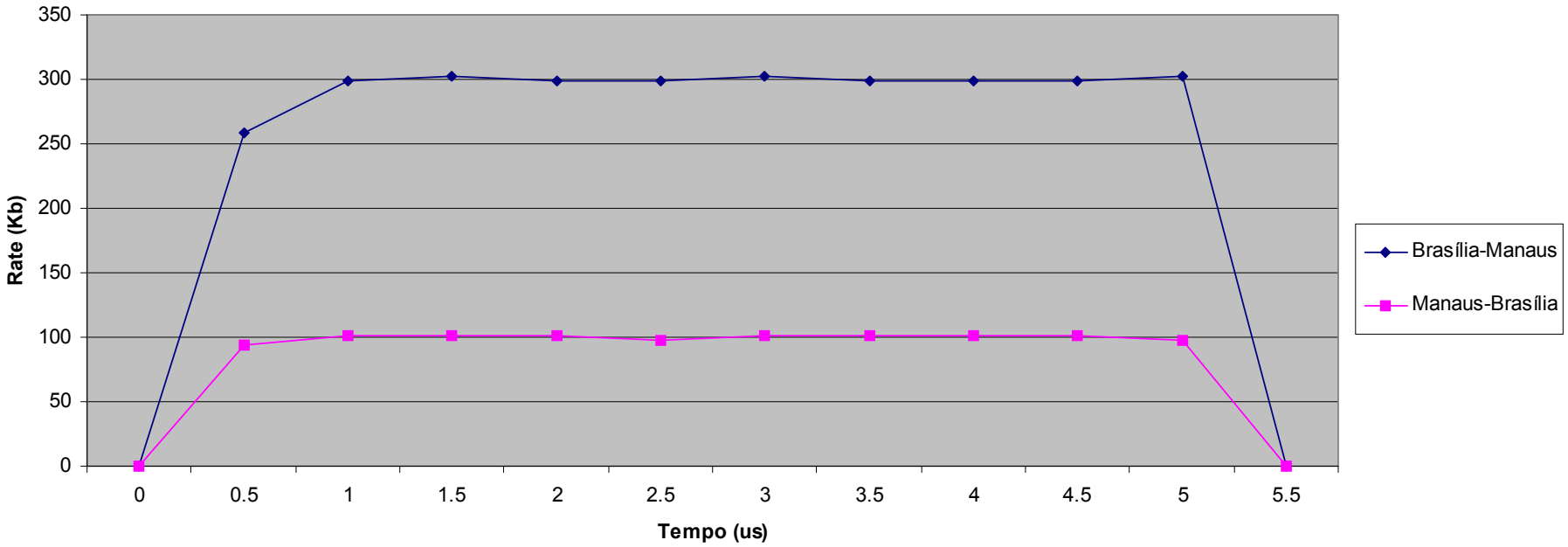


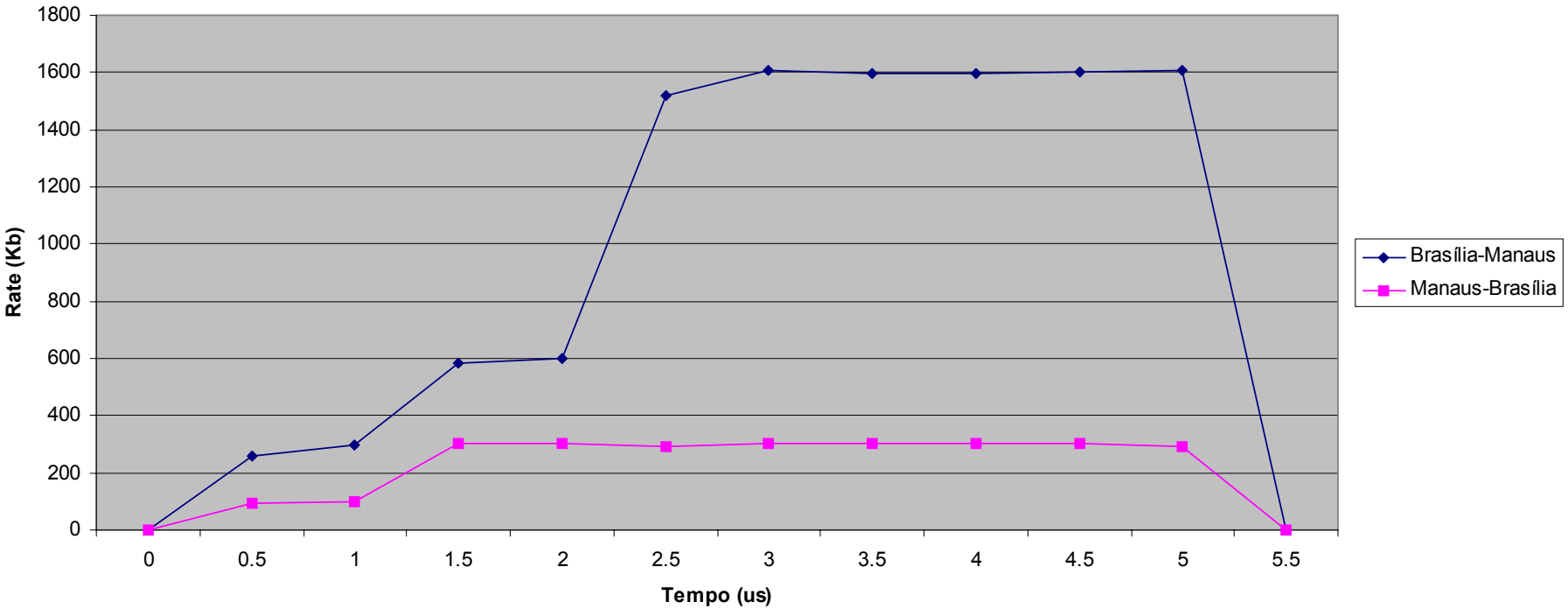
Fig. 3.2.10 – Gráfico DropTail com ataques

3.5 – SFQ

No gráfico 3.3.1 temos o comportamento de Brasília e Manaus durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.3.2, vemos o mesmo gráfico, agora, com o ataque.

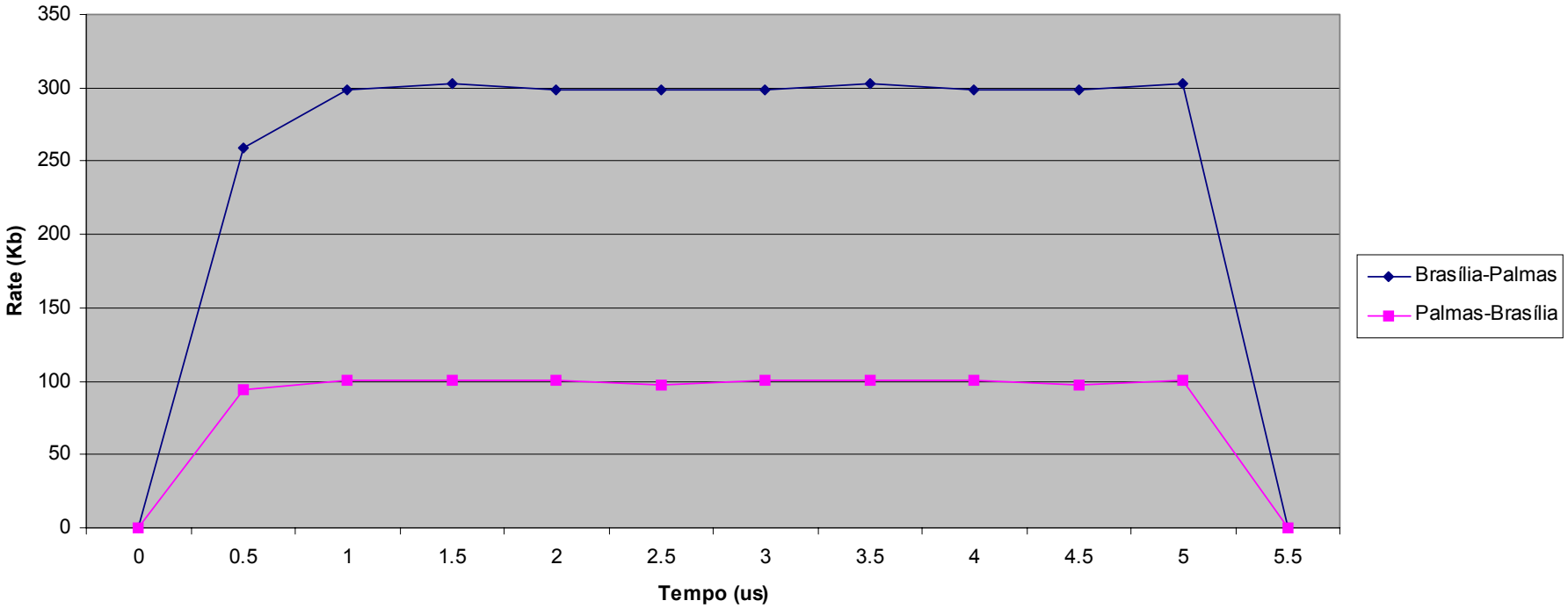


Graf. 3.3.1 – Comportamento de Brasília – Manaus sem os ataques

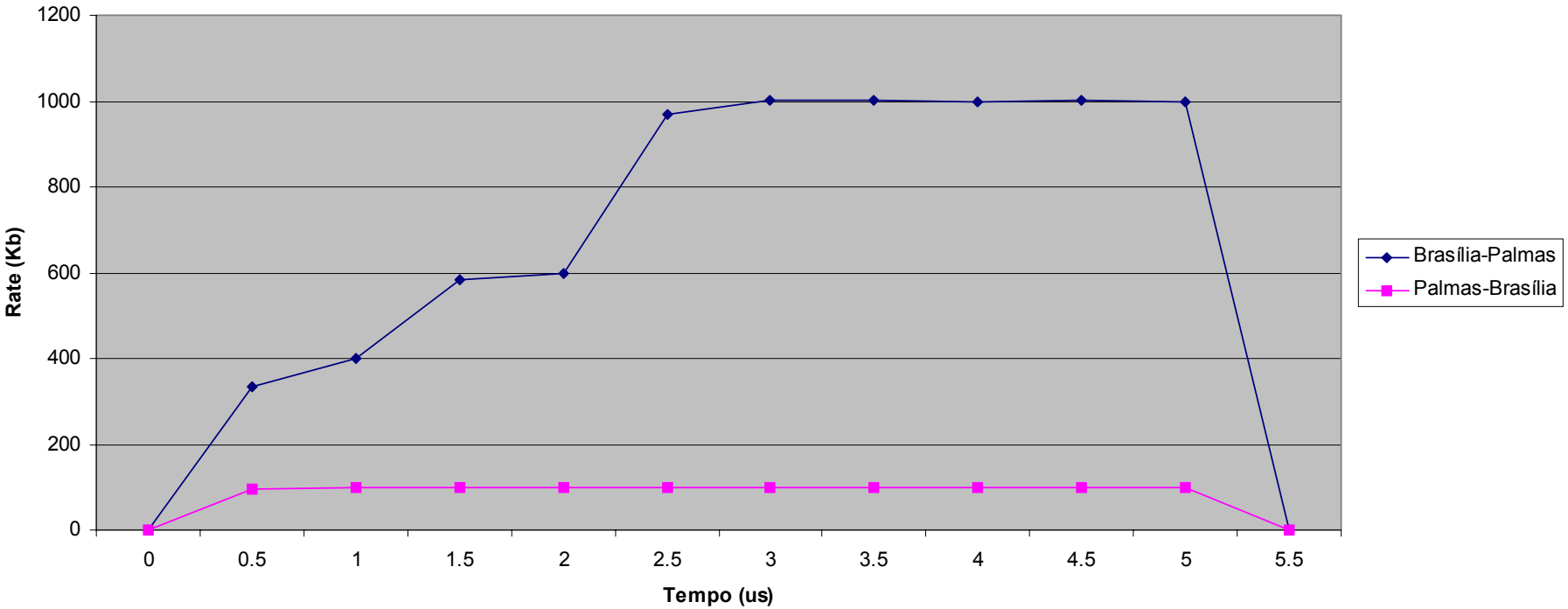


Graf.3.3.2- Comportamento de Brasília – Manaus com os ataques

No gráfico 3.3.3 temos o comportamento de Brasília - Palmas durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.3.4, vemos o mesmo gráfico, agora, com o ataque.

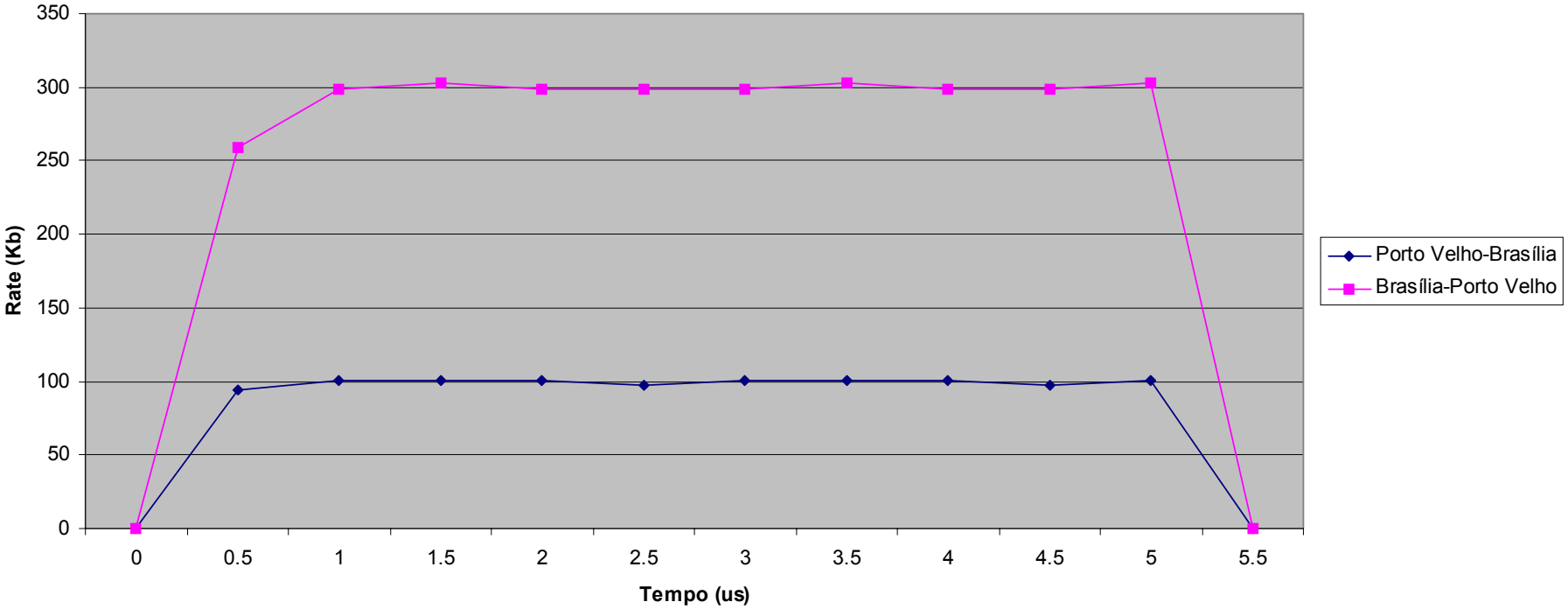


Graf. 3.3.3 - Comportamento de Brasília – Palmas sem os ataques

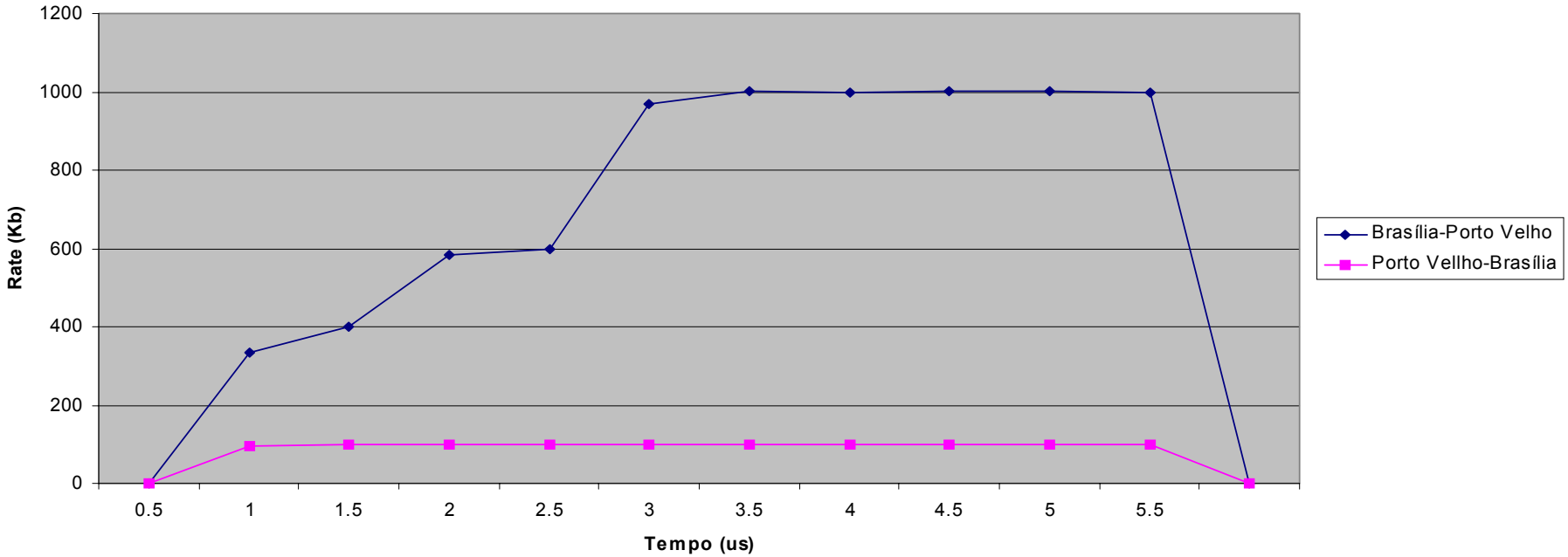


Graf. 3.3.3 - Comportamento de Brasília – Palmas com os ataques

No gráfico 3.3.5 temos o comportamento de Brasília – Porto Velho durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.3.6, vemos o mesmo gráfico, agora, com o ataque.

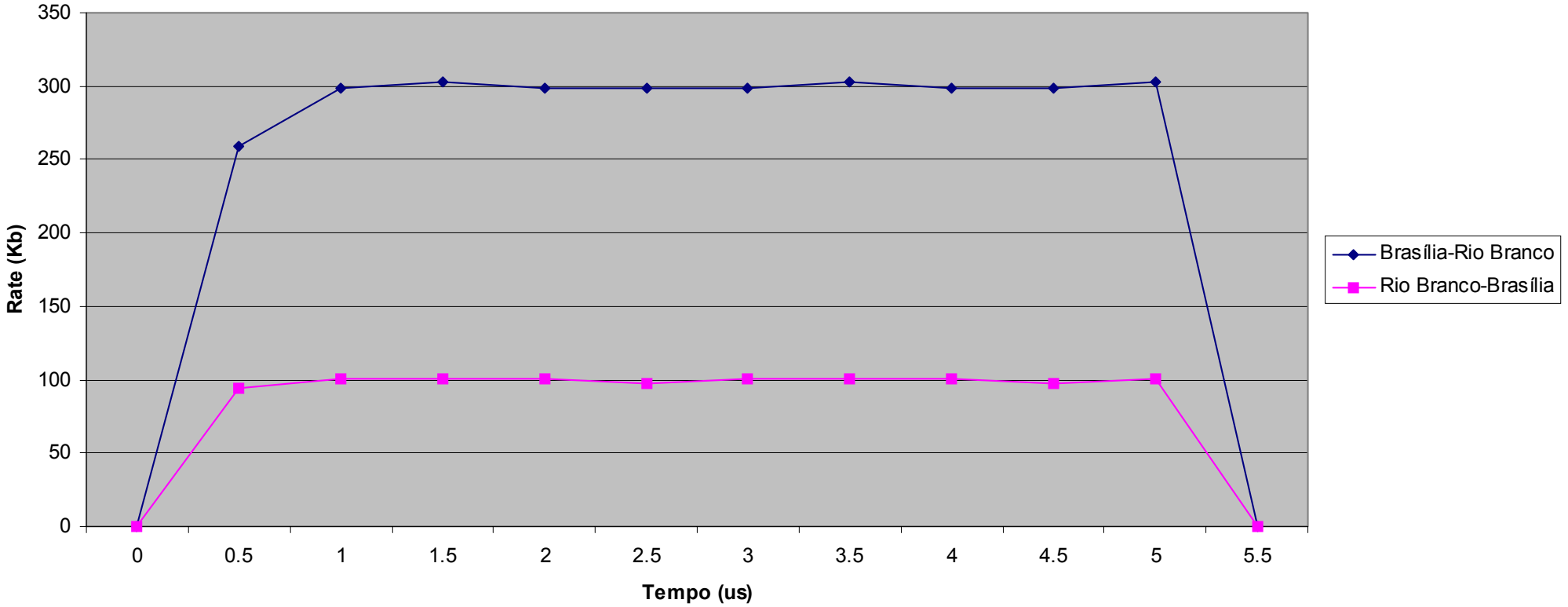


Graf.3.3.5 – Comportamento Brasília-Porto Velho sem os ataques

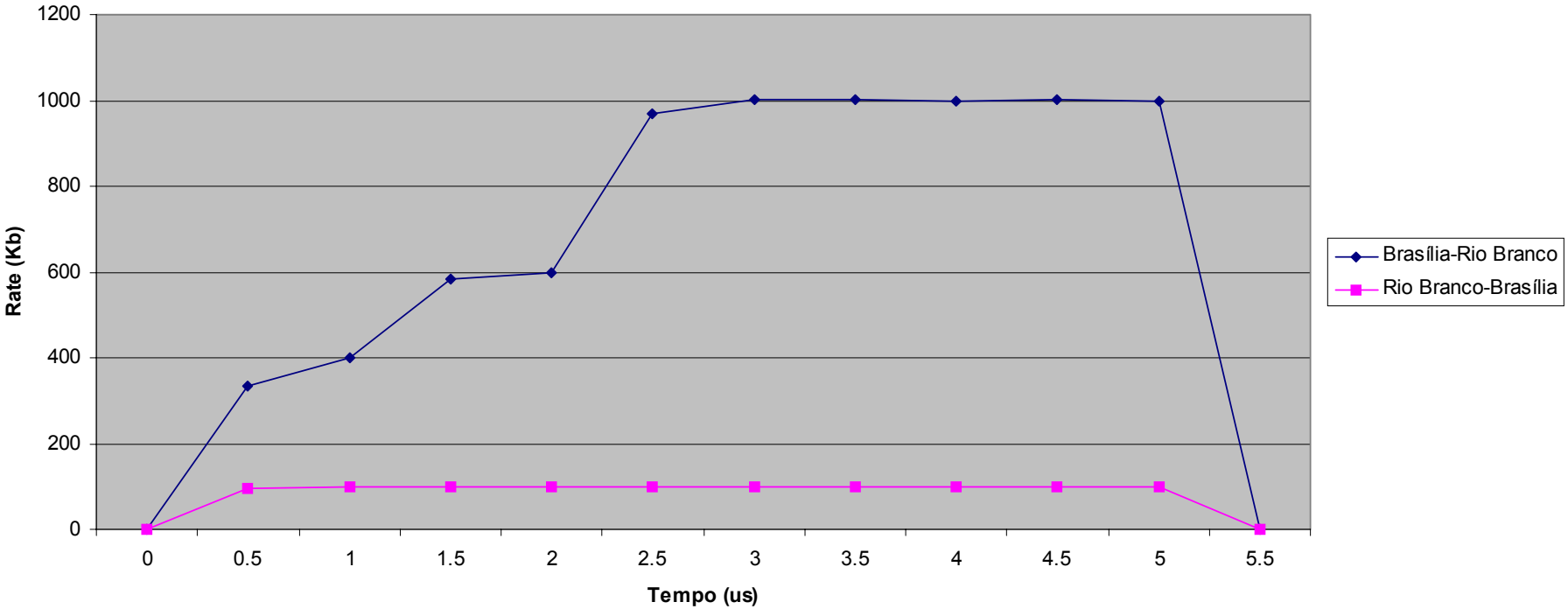


Graf.3.3.6 – Comportamento Brasília-Porto Velho com os ataques

No gráfico 3.3.7 temos o comportamento de Brasília – Rio Branco durante a simulação da vídeo conferência, sem ataque, e, logo em seguida, no gráfico 3.2.8, vemos o mesmo gráfico, agora, com o ataque.



Graf.3.3.7 – Comportamento Brasília-Rio Branco sem os ataques



Graf. 3.3.8 – Comportamento Brasília-Rio Branco com os ataques

Na figura 3.3.9 temos o gráfico gerado com a vídeo conferência com todos os nós utilizando-se da política de fila SFQ sem os ataques simulados. Observamos uma estabilidade na banda do gráfico.

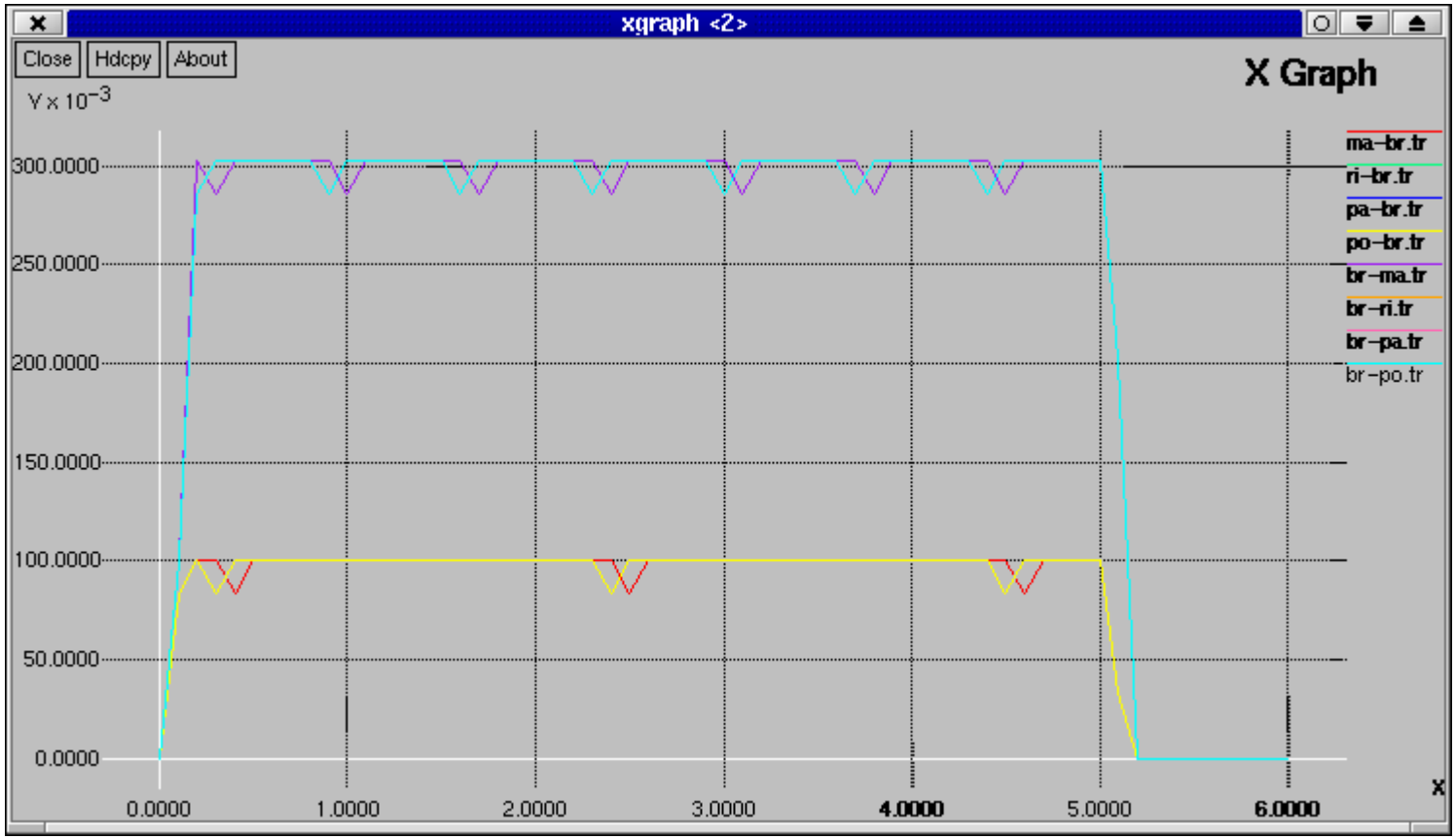


Fig. 3.3.9 – Gráfico com a política SFP sem ataques

Na figura 3.3.10 Observamos a disposição do gráfico da vídeo conferência com todos os nós utilizando-se da política de filas SFQ, agora sendo com os ataques simulados. Com esses ataques, percebemos claramente a oscilação da banda passante, como nos mostra o gráfico.

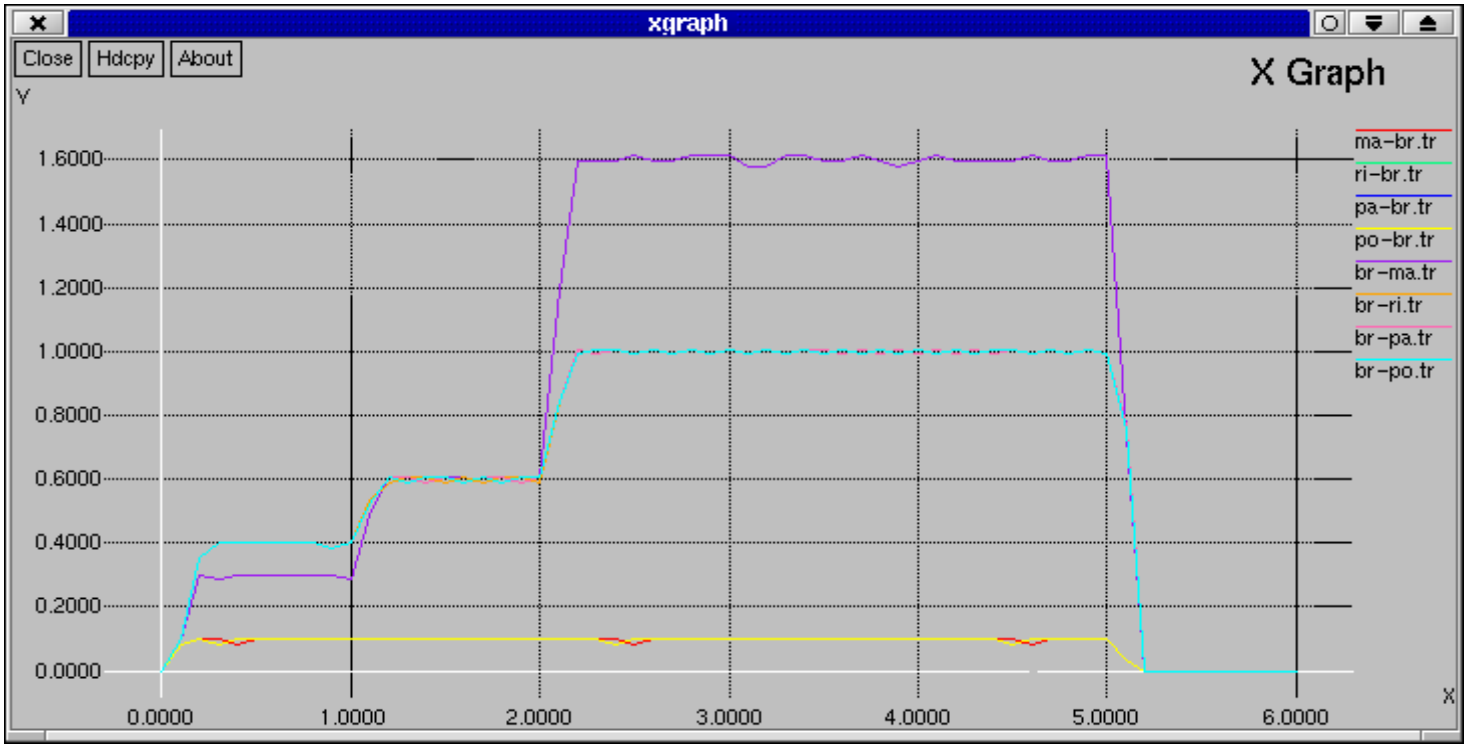


Fig. 3.3.10 – Gráfico SFQ com ataques

3.6 – Análise dos Resultados

Analisando os gráficos oriundos das simulações feitas utilizando as duas políticas de filas, DopTail e SFQ, observamos que a rede se comporta de forma semelhante em ambas situações, sendo observadas diferenças na organização e gerenciamento de filas do roteador principal em Brasília.

Na política DropTail observamos que o roteador de Brasília, quando foi saturado de pacotes decorrentes do ataque, começou a criar filas e a descartar pacotes de tráfego da vídeo conferência, prejudicando assim sua continuação pondo em questão o aspecto “prioridade de pacotes” .

Na política SFQ observamos uma melhora considerável no que diz respeito a priorização de pacotes, pois quando o roteador foi saturado de pacotes de ataques, as filas criadas foram bem menores e o roteador iniciou uma operação de descarte de pacotes do tipo ping-flood (pacotes do ataque), o que nos leva a indicar tal política para tal evento. Neste caso a vídeo conferência teve sua continuação um pouco prejudicada no que diz respeito a delay, porém não se observou nenhum tipo de perda de dados, ao contrário da política DropTail, os algorítimos estocásticos usados na política SFQ priorizaram os pacotes da vídeo conferência, descartando, assim, os demais.

4. Sugestões para minimizar os problemas de segurança

A seguir, sugerimos uma série de métodos que podem ser tomados para reduzir significativamente os problemas de segurança:

- Utilizar políticas de priorização de pacotes.
- Utilizar filtros anti-flood.
- Utilizar sistemas operacionais seguros e sempre mantê-los atualizados contra falhas de segurança.
- Gerenciar e analisar com prioridade máxima os logs do sistema.
- Manter uma equipe capaz de detectar e solucionar falhas de segurança.
- Utilização de Firewall
- Adotar uma política de Segurança
- Utilizar computadores que sirvam como Firewalls para filtrar a entrada de dados de determinadas portas, como as que aceitam tráfego CBR por exemplo
- Manter profissionais sempre atualizados em relação as novas falhas e correções das mesmas lançadas na internet
- Manter informações como IP's, datas e horas de determinados eventos no mais total sigilo.
- Evitar que muitas pessoas tenham acesso ás máquinas que foram tidas como servidores e firewall.
- Minimizar a quantidade de pessoas com acesso as senhas dos sistemas

5. Conclusão

Com o avanço da tecnologia e a diversificação dos meios de comunicação, é comum que, a cada dia, que passe novas políticas de segurança sejam criadas e, junto com elas, é normal que novas falhas sejam encontradas, pois os temidos “piratas de rede”, popularmente conhecidos como “hackers”, estão sempre procurando novas falhas para mostrar seus conhecimentos.

No trabalho mostrado observamos o comparativo, através de simulações, de duas das mais conhecidas políticas de filas usadas, a DropTail (FIFO) e a SFQ, reportando um cenário fictício de simulação, porém, utilizando-se de dados reais obtidos do mapa do BackBone da RNP, que foi a rede em cima da qual foi feita a simulação.

No intuito de sempre procurar melhores soluções indicamos também para estudos futuros simulações similares às feitas neste trabalho, porém explorando outras políticas de fila como WFK, Blue e Red.

6. Referências bibliográficas

COMER, Douglas. **Redes, Computadores, Internet**. Bookman Companhia. [LV-05]

KNIGHTMARE, Fiery. **Secrets Of a Super Harker**. Loopanics Unlimited. [LV-03]

MENDES, Wayne Rocha**. Submundo Harcker do Linux.** Ciência Moderna. [LV-01]

OLIVEIRA, Wilson José de**. Harker – Invasão e Proteção.** Visual Books Editora. [LV-02]

YOSVER, Elda Raquel. **Networkin RedesLan/Wan**. Brasport. [LV-04]

[ST-01] – RNP **– http://www.rnp.br**

[ST-02] – NS – **http://www.isi.edu/nsnam/**

Anexo I – Código Fonte da Simulação Drop tail sem ataques

Anexo II – Código Fonte da Simulação Drop tail com ataques

Anexo III – Código Fonte da Simulação SFQ sem ataques

Anexo IV – Código Fonte da Simulação SFQ com ataques