

# Projeto de Graduação



20 de Dezembro de 2011

## **SIMULAÇÃO DE REDES DE COMPUTADORES COM OPNET IT GURU® ACADEMIC EDITION**

José Roberto Teixeira Dias Filho



[www.ele.puc-rio.br](http://www.ele.puc-rio.br)

Projeto de Graduação



## **SIMULAÇÃO DE REDES DE COMPUTADORES COM OPNET IT GURU® ACADEMIC EDITION**

**Aluno: José Roberto Teixeira Dias Filho**

**Orientador: Marco Antônio Grivet**

Trabalho apresentado com requisito parcial à conclusão do curso de Engenharia Elétrica na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

## **Agradecimentos**

Gostaria de agradecer:

A Deus, que sempre esteve ao meu lado, me iluminando, dando forças e saúde nessa longa e difícil caminhada,

Aos familiares, minha namorada e meus amigos, pelo incansável apoio e paciência,

Ao meu orientador Marco Antônio Grivet, pelo incentivo e suporte para este projeto,

A todos os professores da PUC-Rio que me incentivaram durante o curso, sendo importantes tanto na minha vida acadêmica quanto no desenvolvimento deste trabalho,

À PUC-Rio, por fornecer uma excelente qualidade de ensino e estrutura.

## Resumo

O OPNET IT Guru® Academic Edition é, sem sombra de dúvidas, o mais versátil simulador de redes de telecomunicações/computadores. No entanto, o seu uso não é trivial, principalmente no que tange ao levantamento de estatísticas de interesse. A ideia do projeto é simular uma rede MPLS Backbone e averiguar os seus limites.

A partir da utilização deste *software* para simular redes de computadores, podemos obter diversos tipos de estatísticas, sendo possível realizar as devidas mudanças a fim de otimizar custos, performances e disponibilidade de redes e, posteriormente, termos uma rede mais eficaz.

Sendo assim, se faz necessário uma breve introdução sobre o *software* a ser utilizado (OPNET IT Guru® Academic Edition) assim como o tipo de rede que iremos simular utilizando o protocolo MPLS.

**Palavras-chave:** redes; Internet; OPNET; MPLS; QoS;

**Sumário**

|           |   |                  |
|-----------|---|------------------|
| <b>1.</b> | <b><i>Introdução .....</i></b>                                    | <b><i>1</i></b>  |
| a.        | <b>Protocolo MPLS (Multiprotocol Label Switching) .....</b>       | <b>1</b>         |
| i.        | <b>Motivação .....</b>  | <b>1</b>         |
| ii.       | <b>Surgimento .....</b>   | <b>1</b>         |
| iii.      | <b>Funcionamento básico .....</b>                                 | <b>1</b>         |
| iv.       | <b>MPLS e DiffServ (Differentiated Services) .....</b>            | <b>3</b>         |
| v.        | <b>Vantagens do uso do MPLS .....</b>                             | <b>4</b>         |
| b.        | <b>Software OPNET IT Guru® Academic Edition .....</b>             | <b>5</b>         |
| i.        | <b>Apresentação .....</b>   | <b>5</b>         |
| ii.       | <b>Principais editores .....</b>                                  | <b>5</b>         |
| iii.      | <b>Modelagem e simulação .....</b>                                | <b>5</b>         |
| iv.       | <b>QoS no OPNET .....</b>   | <b>6</b>         |
| <b>2.</b> | <b><i>Construindo a Rede de Computadores .....</i></b>            | <b><i>7</i></b>  |
| a.        | <b>Definindo serviços .....</b>                                   | <b>8</b>         |
| b.        | <b>Definindo os perfis de usuários .....</b>                      | <b>9</b>         |
| <b>3.</b> | <b><i>Inserindo Qualidade de Serviço na Rede .....</i></b>        | <b><i>10</i></b> |
| a.        | <b>Estruturando a priorização de pacotes nas aplicações .....</b> | <b>10</b>        |
| b.        | <b>Configurando os roteadores .....</b>                           | <b>11</b>        |
| <b>4.</b> | <b><i>Coleta de Estatísticas e Simulação .....</i></b>            | <b><i>13</i></b> |
| a.        | <b>Coleta de estatísticas .....</b>                               | <b>13</b>        |
| b.        | <b>Simulação da rede .....</b>                                    | <b>16</b>        |
| <b>5.</b> | <b><i>Resultados Obtidos Graficamente e Comentários .....</i></b> | <b><i>18</i></b> |
| <b>6.</b> | <b><i>Conclusão .....</i></b>                                     | <b><i>36</i></b> |
| <b>7.</b> | <b><i>Referências Bibliográficas .....</i></b>                    | <b><i>37</i></b> |

## 1. Introdução

Inicialmente, iremos apresentar a tecnologia do protocolo MPLS (localizada numa camada intermediária, entre a camada de enlace e a camada de rede), da motivação para seu desenvolvimento até seu funcionamento e vantagens do seu uso. Em seguida, faremos uma breve apresentação sobre o *software* a ser utilizado para simular a rede proposta, abordando suas principais características e funcionalidades.

### a. Protocolo MPLS (Multiprotocol Label Switching)

#### i. Motivação

Com o desenvolvimento da Internet, demanda por serviços multimídia vem crescendo e tornando-se cada vez mais popular. Tais serviços geram uma intensidade de tráfego na rede muito grande, demandando altas taxas de transmissão. Porém, são sensíveis ao atraso e à variação de atraso experimentados na rede.

A Internet consiste em redes IP não orientadas à conexão, baseadas em datagramas. No entanto, quando tratamos de tráfego em tempo real interativo, os roteadores não conseguem oferecer garantias de qualidade de serviço (*Quality of Service – QoS*), visto que os roteadores não reservam recursos nem realizam roteamento baseado nos requisitos de QoS.

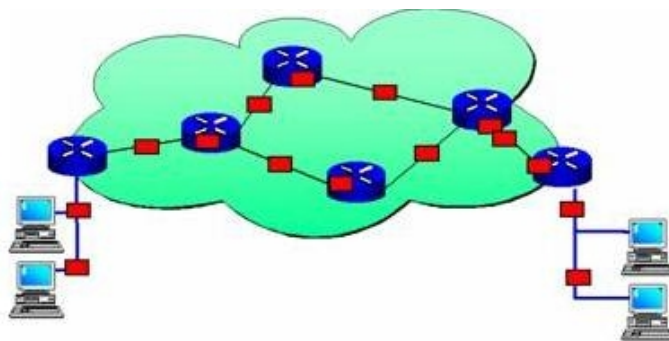


Figura 1 – Rede IP convencional

#### ii. Surgimento

O surgimento da tecnologia MPLS veio como solução para o problema citado no item anterior, a partir da criação das redes orientadas à conexão, baseadas em circuitos virtuais: as redes ATM (*Asynchronous Transfer Mode*), que utilizam comutadores ao invés de roteadores.

Esta tecnologia se propõe, de forma inteligente, a concatenar as vantagens do roteamento com a eficiência e a reserva de recursos presentes nas redes de comutação de pacotes baseadas em circuitos virtuais.

#### iii. Funcionamento básico

Baseado no conceito de *Label Switching* ("Rótulos de roteamento", em tradução livre), cada pacote recebe um *label* pequeno e fixo colocado no cabeçalho dos pacotes MPLS, que informa aos nós de comutação, como os dados devem ser comutados na rede. Os *labels* são os próprios Identificadores de Canal Virtual (*Virtual Channel Identifier*) e de Caminho Virtual (*Virtual Channel Identifier*).

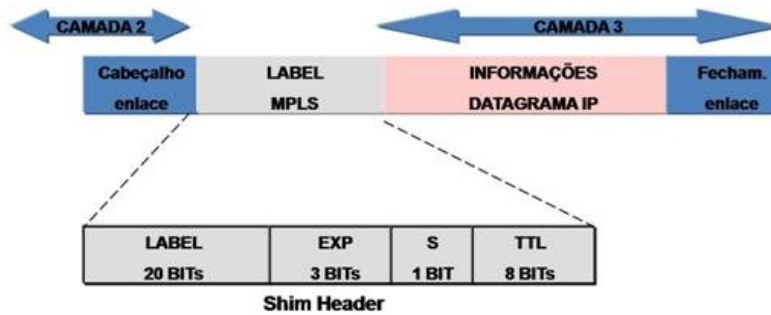


Figura 2 – Formato dos rótulos

**Label** – identificação do rótulo;

**Exp ou CoS (Class of Service)** – usados para alterar os algoritmos de enfileiramento e descarte, possibilitando dar prioridade a certos pacotes; especifica as classes de serviço num LSP (*Label Switch Path*), ou seja, determina o caminho do pacote dentro da nuvem MPLS;

**Stack** – permite a criação de uma pilha hierárquica de rótulos;

**TTL (Time to Live)** – tempo de vida do IP convencional.

Os rótulos só são válidos entre roteadores adjacentes.

À priori, o MPLS foi desenvolvido para ser usado conjuntamente com redes ATM. Entretanto, foi possível dispensar o plano de controle do ATM, aproveitando-se somente a comutação de células, feita em *hardware*.

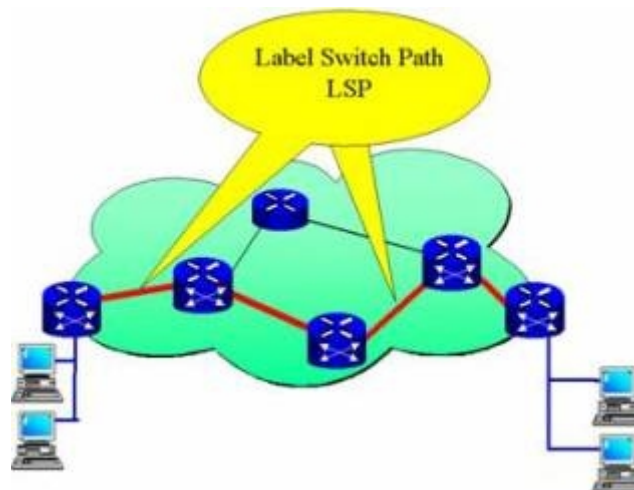


Figura 3 – LSP (*Label Switch Path*)

No MPLS, existem dois tipos de roteadores: o LER (*Label Edge Routers*) e o LSR (*Label Switch Routers*). Os LER's estão localizados na borda da rede e tem como principal função classificar e escolher os *labels* adequados para cada fluxo de pacotes na entrada da rede e remover os *labels* na saída da rede. Eles também convertem pacotes IP em pacotes MPLS. Os LSR's estão localizados no núcleo da rede e são comutadores de alta velocidade, tendo como objetivo encaminhar pacotes rapidamente.

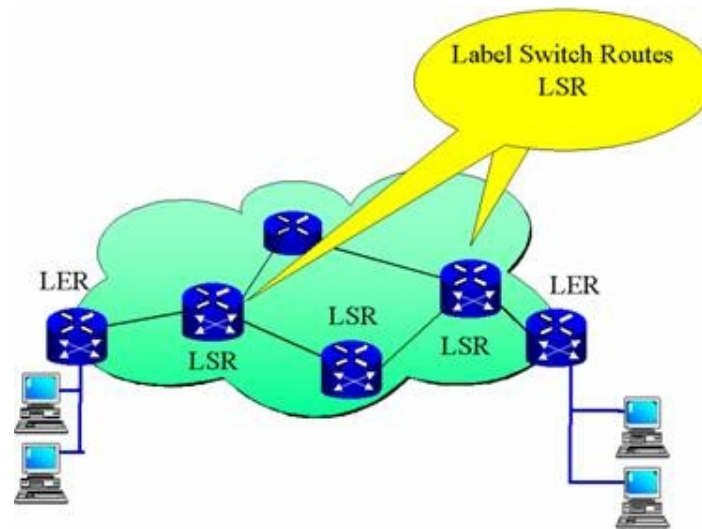


Figura 4 – LER (Label Edge Routers) e LSR (Label Switch Routers)

Ao chegar um pacote num LER de entrada, este analisa o conteúdo do datagrama IP e configura o cabeçalho adequado. Quando este pacote MPLS chega num LSR, este examina o cabeçalho do pacote MPLS. De acordo com o cabeçalho e a interface em que o pacote chegou, O LSR consulta uma tabela de encaminhamento e determina qual é o label da interface de saída. Conhecendo a saída, o LSR substitui o label no cabeçalho do pacote MPLS e o envia para o próximo LSR. Sendo assim, podemos concluir que um LSP é um caminho definido através de uma sequência de labels, entre dois LER's.

Sabendo que todos os pacotes pertencentes a um LSP seguem o mesmo caminho pré-definido, precisamos classificar esses caminhos. Para tal, faz-se uso das FEC's (*Forwarding Equivalence Class*), cujos pacotes que constam nelas recebem os mesmos tratamentos, podendo haver várias FEC's para classificar o tráfego de um LSP. Resumindo, uma FEC define um conjunto de regras de classificação do tráfego de um dado LSP.

Poderíamos nos perguntar o que aconteceria caso um LSP transportasse tráfego classificado por mais de uma FEC. É simples: o campo CoS do rótulo do pacote MPLS é utilizado para identificar à qual FEC pertence o pacote. Como este campo possui três bits, podemos definir oito FEC's num LSP.

#### iv. MPLS e DiffServ (Differentiated Services)

Com a finalidade de melhorar o suporte à QoS, que será abordado mais profundamente à frente, numa rede multimídia IP, foi implementado conjuntamente o MPLS e o *Diffserv*. Este último propõe uma estrutura de priorização de pacotes, na qual são classificados de acordo com o tipo de dados que levam. A classificação de tráfego é feita através do DSCP (*Differentiated Service Code Point*).

No *Diffserv*, o campo ToS dos pacotes IP é redefinido como DS (1 byte). Este campo possui os seis bits do DSCP, que são utilizados para definir a que classe de serviço (PHB – *Per Hop Behaviour*) pertence um determinado pacote IP. Cada DSCP especifica um PHB.



Todos os pacotes com o mesmo PHB são tratados da mesma forma num roteador da rede. O IETF (*Internet Engineering Task Force*) os seguintes tipos de PHB:

**EF – Expedited Forwarding:** maior prioridade. Tráfegos que exigem baixa perda de pacotes, atraso e variação de atraso limitados.

**AF – Assured Forwarding:** prioridade intermediária. Tráfegos com diferentes exigências de perda, mas que toleram erros.

**Best Effort:** prioridade baixa. Está presente na Internet, mas não provê QoS.

Caso não haja espaço suficiente numa área de armazenamento (chamada de *buffer* ou fila), pode haver perdas consideráveis de pacotes. De modo a controlar o tráfego total transportado numa rede, há os seguintes métodos de escalonamento dos sistemas de filas: FIFO (*First-In-First-Out*), PQ (*Priority queuing*) e WFQ (*Weighted Fair queuing*).

O método FIFO é o mais simples e comumente utilizado. Todos os pacotes são tratados igualmente, inseridos numa única fila e servidos na mesma ordem que chegam à fila.

Já o método PQ permite que sejam atribuídos até quatro níveis de prioridades aos diferentes tráfegos gerados na rede: alta, média, baixa e normal. Pacotes não marcados são tratados sem nenhuma prioridade, sendo classificado como *Best Effort*.

Por fim, o método WFQ, que consiste na diferenciação de fluxos por meio de um peso atribuído a cada um. Esse peso serve para calcular o tempo associado a cada pacote. Podendo suportar pacotes de tamanhos variáveis, fluxos com pacotes maiores não alocam mais banda que os fluxos de pacotes menores.

Também é possível no *Diffserv* classificar o tráfego nos LER's, a fim de configurar o campo ToS, ou que o *software* nos terminais da rede façam isto, antes que o tráfego atinja os roteadores.

A principal forma de integração entre o MPLS e o *Diffserv* é dada pela utilização do campo DSCP como regra de construção de FEC's, fazendo com que o tráfego *Diffserv* seja mapeado para dentro de uma FEC de um LSP. No entanto, os LER's da rede devem possuir tabelas de mapeamento PHB, com o campo CoS bem definido, permitindo o mapeamento de uma classe *Diffserv* numa classe MPLS de um dado LSP. Desta forma, é possível implementar mecanismos de priorização de tráfego nos LSR's da rede baseados no campo CoS do MPLS e ToS do IP (*Diffserv*).

## v. Vantagens do uso do MPLS

Uso de comutadores no lugar de roteadores: Além de serem mais baratos, os comutadores operam em velocidades superiores e realizam pesquisa e troca de rótulos.

Análise única de pacotes: Os pacotes que entram na rede MPLS são analisados apenas uma vez. O roteador de ingresso pode utilizar qualquer informação sobre o pacote para determinar a que FEC ele pertence, possibilitando a criação de classes de serviço para diferenciar pacotes, não deixando sobrecarregar rotas congestionadas. Pode-se, por exemplo, para pacotes de maior prioridade, escolher caminhos mais rápidos, de custo mais elevado.

Diferenciação de rótulos: Pode-se rotular diferenciadamente pacotes que entram por roteadores ou, até mesmo, interfaces diferentes, permitindo a criação de VPN's (*Virtual Private Networks*).

Processamento de pacotes: Visto que a parte pesada do processamento de pacotes é feita nos LER's, o núcleo da rede pode operar com mais folga, proporcionando uma elevada taxa de pacotes por segundo no núcleo da rede se comparada com a taxa de pacotes por segundo nas bordas.

## b. Software OPNET IT Guru® Academic Edition

### i. Apresentação

O *software* OPNET IT Guru® Academic Edition é uma ferramenta computacional utilizada para modelar redes num ambiente virtual, analisar e prever seu desempenho, incluindo aplicações, usuários e tecnologias de redes e protocolos.

### ii. Principais editores

**Editor de projeto:** É a área de trabalho para criação de um modelo de rede que possua, por exemplo, roteadores, *switches*, estações de trabalho, servidores e etc. Nesta área são criados os cenários da rede a ser simulada, sendo possível escolher as estatísticas a serem analisadas, rodar a simulação e apresentar os resultados a partir das bibliotecas existentes. Também é possível criar qualquer recurso para simular uma rede específica, fazendo uso dos editores auxiliares. É denominado “nó” cada elemento existente no editor de projeto.

**Editor de nó:** Define o comportamento de cada componente da rede. Diversos módulos definem tais comportamentos e modelam aspectos interno do comportamento do nó. Os módulos são conectados através de *packet streams* (feixe de pacotes) ou *statistics wires* (ligações para estatísticas).

**Editor de modelos de processos:** Responsável pelo controle da funcionalidade das camadas mais baixas dos modelos de nó. São representados por uma quantidade finita de máquinas de estado e linhas que representam as transições entre estados.

**Editor de modelo de enlaces:** Nesse editor, são criados os enlaces (*links*) utilizados para conectar os nós da rede. Cada novo enlace pode possuir diferentes interfaces para definir atributos e diferentes tipos de representação.

**Editor de rota:** Nesse editor, são gerados os elementos que definem uma rota (*path*) de tráfego. Estes elementos permitem criar um componente de rota em vários modelos de protocolo, utilizando conexões lógicas ou circuitos virtuais (*Ethernet*, *FDDI*, *Frame Relay*, *MPLS*, *ATM*, *SONET* e etc).

**Editor de formato de pacote:** Permite a configuração dos campos dos pacotes. O formato do pacote é representado por caixas coloridas e o tamanho destas é proporcional ao número de *bits* que compõe o campo.

### iii. Modelagem e simulação

Para podermos evitar erros potenciais no planejamento e implementação de redes, a tecnologia de modelagem e simulação deve crescer e se desenvolver cada vez mais. Quando simulamos uma rede de computador, podemos compará-la ao funcionamento real, visto que o simulador nos proporciona as mesmas condições do seu funcionamento.

A facilidade de modelarmos e simularmos os mais variados tipos de cenários nos propicia vantagens consideráveis, uma vez que podemos reiniciar a simulação diversas vezes, e estudar vários cenários diferentes.

Para avaliar o desempenho de rede as tarefas tornam-se complexas e trabalhosas, sendo necessário realizar um levantamento de dados estatísticos sobre a rede com a finalidade de que os resultados sejam confiáveis. Abaixo, seguem os modelos para representar e medir o desempenho do sistema:

**Modelos com redes reais:** Possuem alto custo na implementação dos equipamentos, mas em contrapartida oferecem uma visão ampla da rede, permitindo uma análise real. Outras desvantagens são: o tempo gasto para a instalação dos componentes, a preparação do ambiente de avaliação e o desenvolvimento de técnicas para geração e coleta das estatísticas.

**Modelos analíticos:** Suas bases são modelos matemáticos, oferecendo resultados bastante confiáveis. Entretanto, há restrições quando modelamos uma rede de comunicação, apresentando as seguintes desvantagens: excessivo número de variáveis a serem consideradas (a fim de manter os modelos tratáveis) e as inúmeras simplificações feitas ao longo da modelagem, conduzindo para conclusões restritas.

**Modelo computacional:** Desenvolvimento de um programa representando uma parte, ou a rede real como um todo, de maneira que experimentos no modelo representem uma previsão real do funcionamento da rede. Tal modelo é largamente utilizado nas mais diversas áreas da engenharia. Esse modelo é mais econômico que testar a rede real, evitando gastos na compra de equipamentos desnecessários.

#### iv. QoS no OPNET

A QoS refere-se à capacidade de criar caminhos com garantias de banda, atraso de pacotes, *jitter* (parâmetro de variação de atraso de pacotes), perda de pacotes entre dois pontos da rede. A introdução de QoS nas redes atuais de Internet é fornecer gerenciamento dos parâmetros de qualidade e controle, além do serviço de *Best Effort* da Internet.

Além de aumentar a velocidade de processamento nos nós comutadores, o maior desafio da Internet é suportar diversas aplicações a fim de garantir um certo nível de QoS para funcionar corretamente, mesmo quando a rede estiver congestionada. Neste sentido, a IETF possui alguns grupos de trabalho que têm como objetivo estender a arquitetura atual da Internet para prover diferentes níveis de QoS para os usuários: *Intserv* (Serviços Integrados), *RSVP* (*Resource reSerVation Protocol*) e *Diffserv*, este último já citado anteriormente.

**Serviços Integrados:** descreve as primeiras funções de QoS que devem estar presentes em todas as estações e roteadores, em especial o controle de admissão, a alocação de recursos e a classificação e escalonamento de pacotes. O *Intserv* visa fornecer, numa rede comutada de pacotes (por exemplo, a Internet), um serviço cuja QoS é determinada por uma das classes do tipo, Serviço Garantido (GS – *Guaranteed Service*) e Serviço de Carga Controlada (CL – *Controlled Load Service*).

Antes da transmissão de dados ser iniciada, as aplicações precisam encontrar um caminho até o receptor que satisfaça as exigências de QoS, reservando ao longo do caminho, os recursos necessários. Com o objetivo de atingir uma dada QoS, tais aplicações utilizam o protocolo RSVP, que nada mais é do que um protocolo de controle e sinalização que atua na camada de rede, sendo responsável por reservar caminhos e recursos na sub-rede de comunicação.

→ **GS – Serviço Garantido (RFC 2210):** O atraso fim-a-fim tem um limite superior rígido e garante a disponibilidade de largura de banda. Destinado à aplicações que possuem requisitos estritos de tempo real (voz e vídeo), não tolerantes a atraso, *jitter* e perdas, necessitando de garantia firme de que um pacote não chegará ao destino com tempo superior ao especificado. Sendo assim, este serviço apresenta um alto nível de QoS na Internet.

→ **CL – Serviço de Carga Controlada (RFC 2211):** Suas garantias não são tão rígidas quanto o Serviço Garantido, todavia, fornece um serviço equivalente ao *Best Effort* numa rede pouco utilizada com quase nenhuma perda ou atraso. Quando há sobrecarga, esse serviço compartilhará a largura de banda entre múltiplos fluxos, de maneira controlada, garantindo um serviço melhor que o usual. No entanto, esse serviço não oferece garantias de atraso máximo, apenas um limiar probabilístico, não podendo também assegurar que pacotes não sejam perdidos.

## 2. Construindo a Rede de Computadores

A rede de computadores criada é tipicamente de uma empresa prestadora de serviços de telecomunicações, possuindo diversos clientes espalhados por uma dada região. No caso a ser estudado, temos 280 usuários divididos conforme a figura abaixo, onde em cada localidade existem LAN's (Local Area Network) conectados ao roteador mais próximo através de *links* tipo 100BaseT (Cabo Ethernet UTP – *Unshielded Twisted Pair* – Categoria 5). As ligações entre roteadores e servidores também são feitas com esses mesmos tipos de *links*.

Temos cinco roteadores do fabricante Juniper que irão fazer o encaminhamento dos pacotes de uma LAN para a outra. Há dois servidores de onde os usuários irão buscar as informações solicitadas. Essa rede criada é denominada WAN (Wide Area Network).

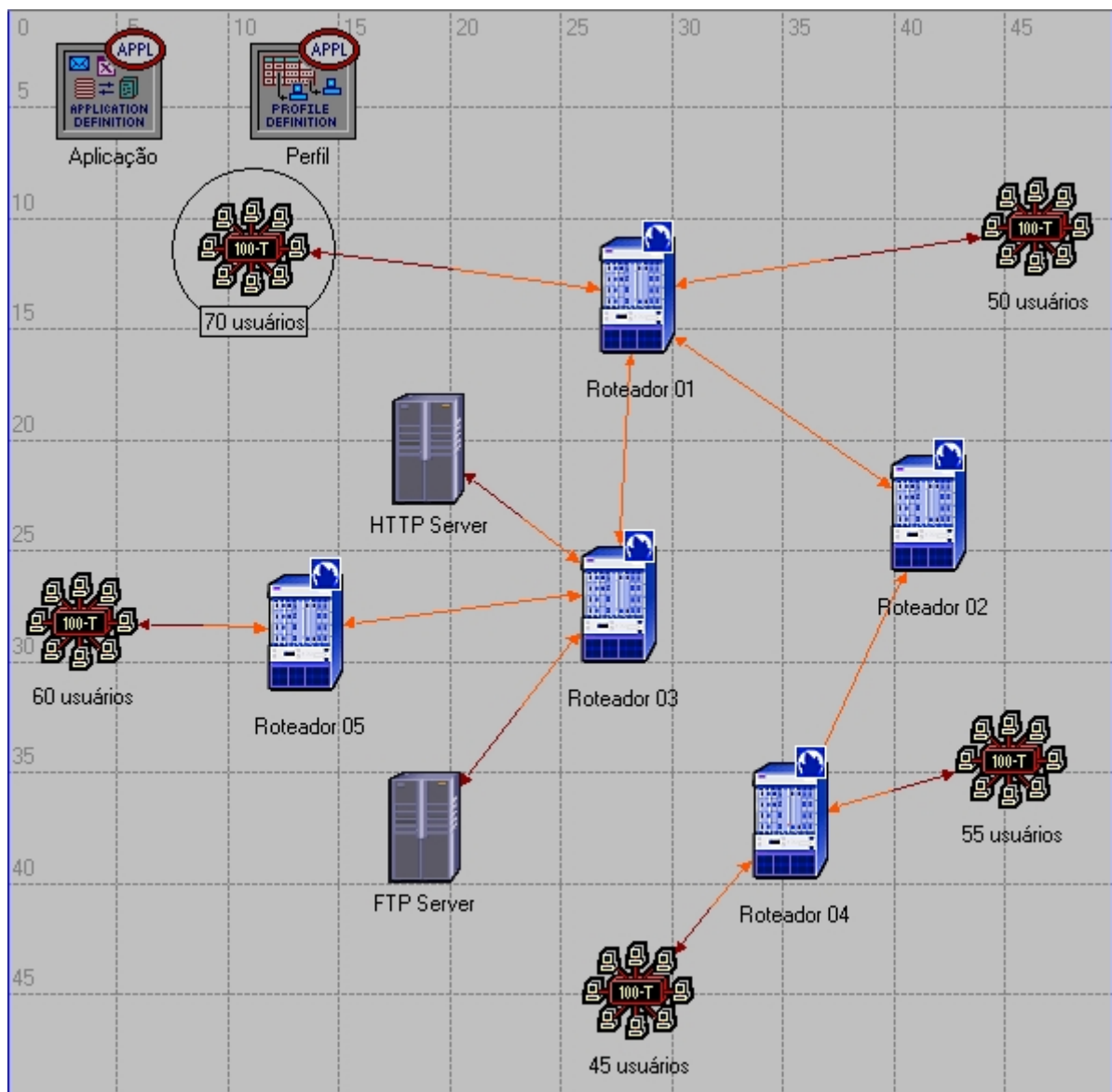


Figura 5 – A rede de computadores

## a. Definindo serviços

A caixa denominada "Aplicação" contém as informações sobre quais tipos de serviços são oferecidos aos usuários. São três tipos de serviços como mostrado abaixo:

- ➔ Transferência de arquivos: FTP (*File Transfer Protocol*) – As cargas de transferência podem ser do tipo: *High Load, Medium Load, Low Load*;
- ➔ Acesso à páginas da Internet: HTTP (*Hyper Text Transfer Protocol*) – Os tipos de acesso podem ser: *Searching, Image Browsing, Light Browsing, Heavy Browsing*;
- ➔ Comunicação telefônica via IP: VoIP (Voz sobre IP)

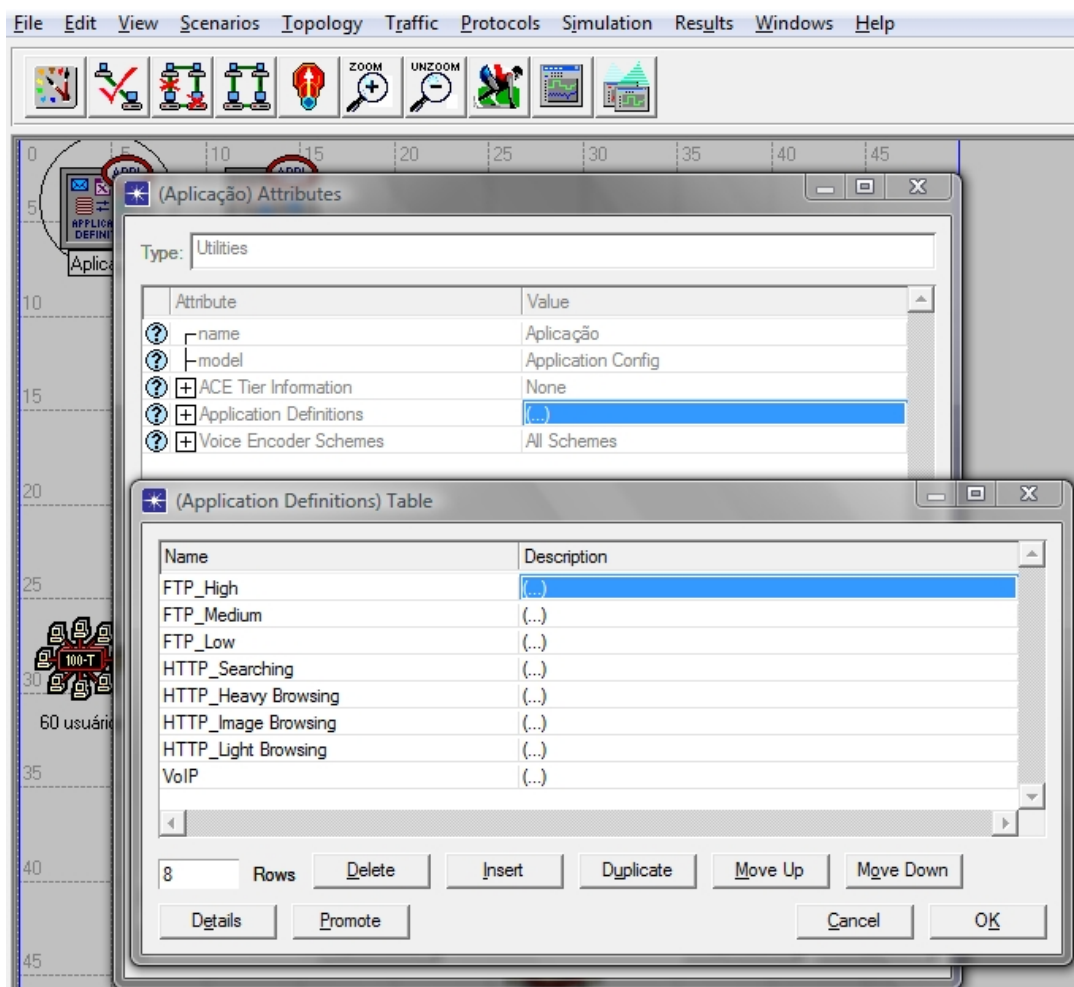
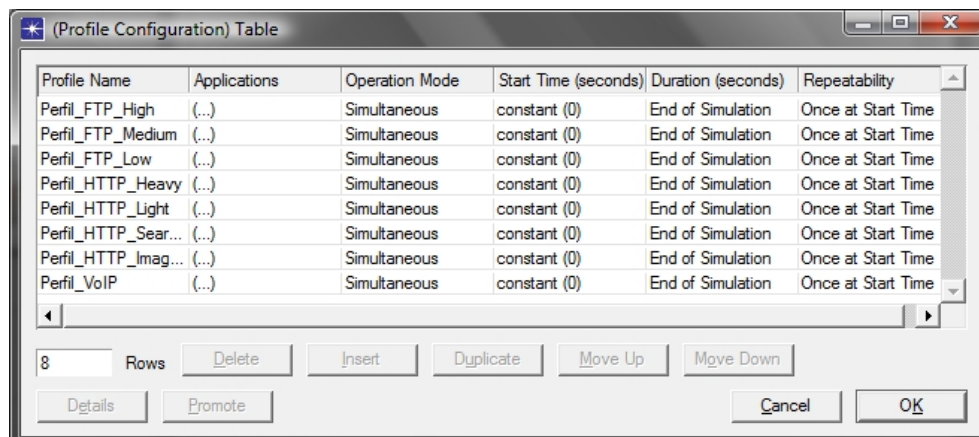


Figura 6 – Aplicações

## b. Definindo os perfis de usuários

Na caixa “Perfil” é onde são criados os perfis que serão atribuídos a cada uma das LAN’s. Para cada LAN precisamos definir quais aplicações (serviços) ela utilizará (FTP, HTTP, VoIP) e os seus respectivos perfis. Nessa rede, consideramos que todas as LAN’s estão utilizando todas as três aplicações.



| Profile Name        | Applications | Operation Mode | Start Time (seconds) | Duration (seconds) | Repeatability      |
|---------------------|--------------|----------------|----------------------|--------------------|--------------------|
| Perfil_FTP_High     | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |
| Perfil_FTP_Medium   | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |
| Perfil_FTP_Low      | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |
| Perfil_HTTP_Heavy   | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |
| Perfil_HTTP_Light   | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |
| Perfil_HTTP_Sear... | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |
| Perfil_HTTP_Imag... | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |
| Perfil_VoIP         | (...)        | Simultaneous   | constant (0)         | End of Simulation  | Once at Start Time |

Figura 7 – Perfis dos usuários

Os perfis de cada LAN seguem abaixo:

Perfil da LAN com 70 usuários:

→ *Perfil\_FTP\_High, Perfil\_HTTP\_Heavy, Perfil\_VoIP*

Perfil da LAN com 50 usuários:

→ *Perfil\_FTP\_High, Perfil\_HTTP\_Light, Perfil\_VoIP*

Perfil da LAN com 55 usuários:

→ *Perfil\_FTP\_Medium, Perfil\_HTTP\_Searching, Perfil\_VoIP*

Perfil da LAN com 45 usuários:

→ *Perfil\_FTP\_Medium, Perfil\_HTTP\_Searching, Perfil\_VoIP*

Perfil da LAN com 60 usuários:

→ *Perfil\_FTP\_Low, Perfil\_HTTP\_Image Browsing, Perfil\_VoIP*

Para que os usuários possam realizar transferências de arquivos e utilizarem páginas da Internet, eles devem acessar servidores que contenham tais informações. Para isto, há a necessidade dos dois servidores presentes no projeto: o “HTTP Server” e o “FTP Server”.

Em cada um deles é necessário realizar configurações sobre os serviços que irão suportar. No servidor HTTP, configuramos de modo a suportar os serviços “HTTP\_Searching”, “HTTP\_Image Browsing”, “HTTP\_Light Browsing” e “HTTP\_Heavy Browsing”. Por outro lado, o servidor FTP foi configurado para suportar os serviços “FTP\_High”, “FTP\_Medium” e “FTP\_Low”.



### 3. Inserindo Qualidade de Serviço na Rede

Com o objetivo de melhorarmos a qualidade de nossa rede, otimizando custos, performances e disponibilidade da rede, usaremos QoS em nossos roteadores.

Nota-se na figura abaixo, que foi inserido uma nova caixa chamada "QoS Config" onde podemos realizar as configurações necessárias para melhorarmos o desempenho da rede.

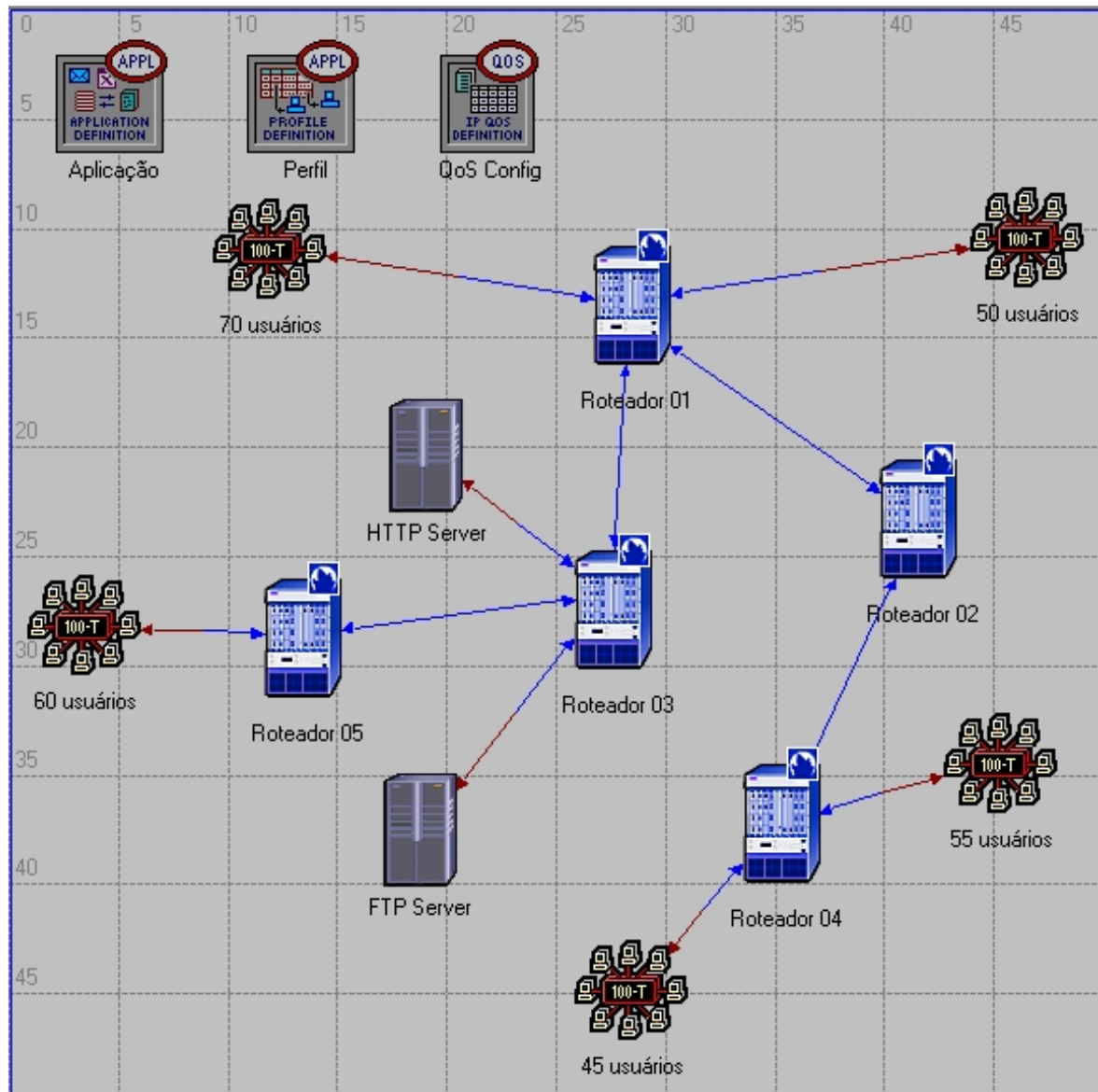


Figura 8 – Rede com QoS

#### a. Estruturando a priorização de pacotes nas aplicações

Antes de começarmos a configuração de QoS nos roteadores, devemos configurar as aplicações.

Na figura a seguir, podemos seguir os procedimentos adotados para realizar as alterações:

Entrando na “Aplicação”, devemos alterar o valor da linha “*Application Definitions*”. Na próxima tela, acessamos o campo “*Description*” de cada aplicação existente nela. Na janela seguinte, iremos alterar o campo “*Value*” da aplicação em questão. Abre-se outra janela (*FTP Table*), e nesta, mudamos no campo “*Value*” do atributo *Type of Service*. Então, configuramos as aplicações para usarem a estrutura de priorização de pacotes *Diffserv* (Serviços Diferenciados) e a classificação dos pacotes do tipo EF (*Expedited Forwarding*), ou seja, de prioridade máxima.

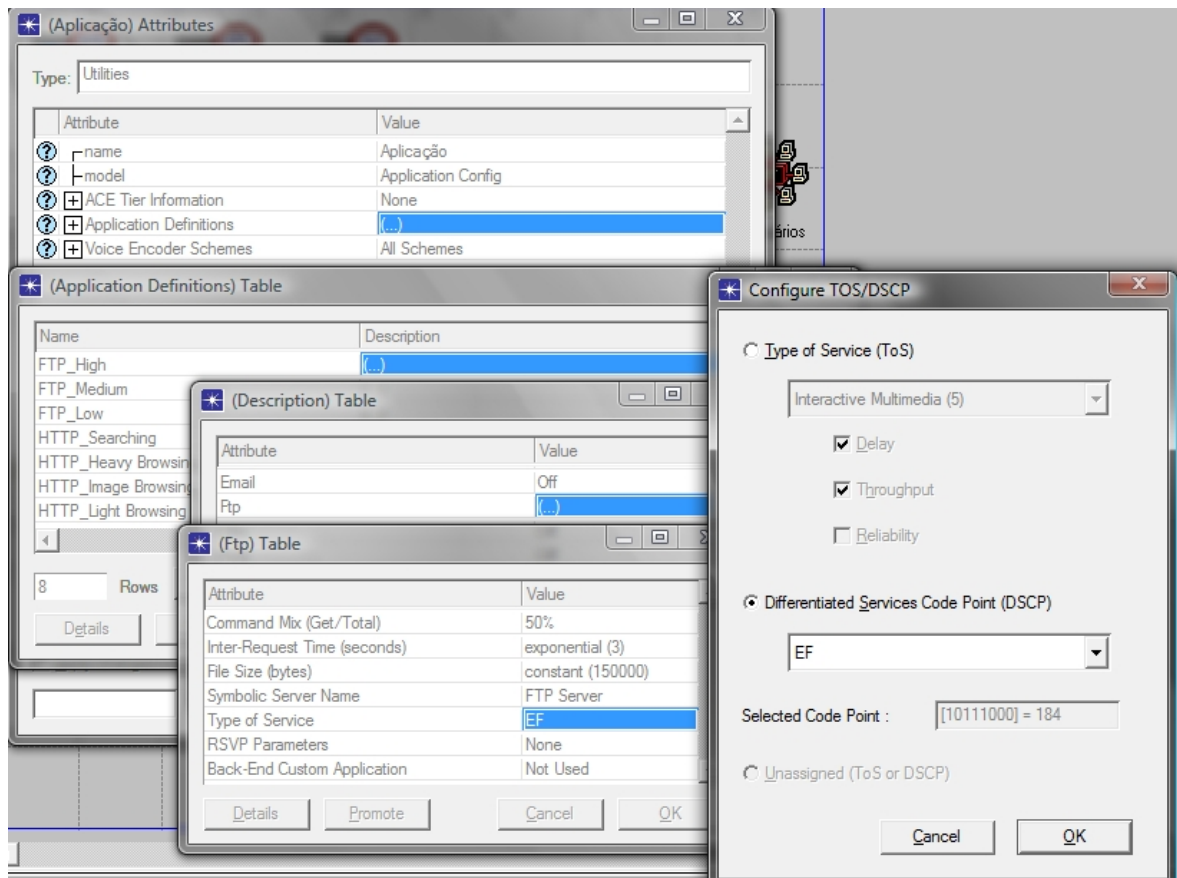


Figura 9 – Configuração de *Diffserv* nas aplicações

Agora, estamos aptos a configurar os roteadores utilizando QoS.

## b. Configurando os roteadores

Clicamos num dos roteadores com o botão direito do *mouse* e depois em “*Select Similiar Nodes*”, selecionando assim todos os roteadores presentes na rede. Em seguida, acessamos o menu no seguinte caminho:

**“Protocols” -> “IP” -> “QoS” -> “Configure QoS”**

Na figura a seguir, no campo “*QoS Scheme*”, temos as opções para os métodos de escalonamento dos sistemas de filas abordados anteriormente no capítulo 1: FIFO (*First-In-First-Out*), PQ (*Priority Queuing*) e WFQ (*Weighted Fair Queuing*).



Visto que não seria possível realizar as comparações necessárias utilizando apenas um dos métodos acima, resolvemos utilizar cada um desses métodos para analisarmos qual deles irá nos fornecer os melhores resultados para o desempenho da rede.

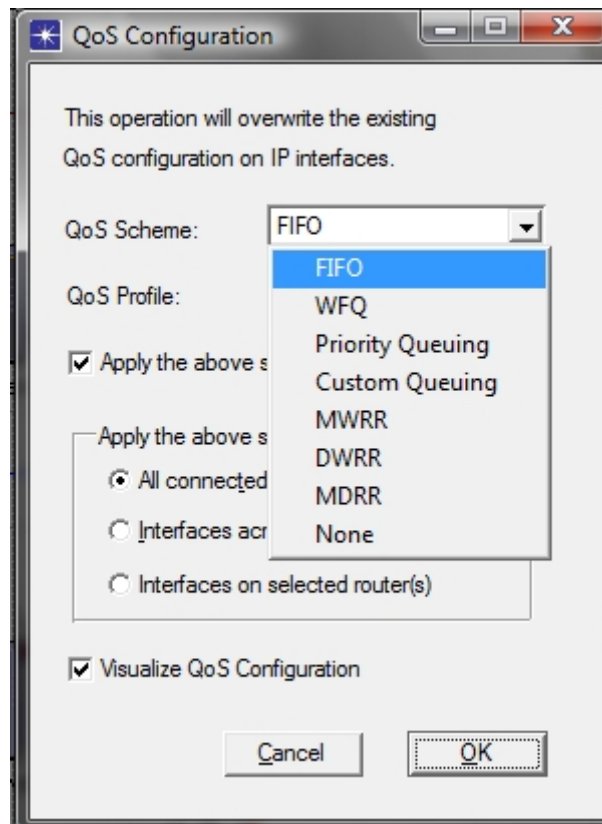


Figura 10 – Configuração da QoS

#### 4. Coleta de Estatísticas e Simulação

Conforme descrito anteriormente, foram criadas quatro redes idênticas: três delas utilizando QoS e os três métodos de escalonamento dos sistemas de filas abordados (FIFO, PQ e WFQ), e uma delas sem utilizar QoS.

Abaixo, seguem os nomes das redes que serão simuladas:

→ *rede\_FIFO*

→ *rede\_PQ*

→ *rede\_WFQ*

→ *rede\_Diffserv* (esta rede usa Serviços Diferenciados mas não utiliza QoS)

##### a. Coleta de estatísticas

Os gráficos a serem gerados e, posteriormente, analisados dependem única e exclusivamente do que o projetista está interessado em analisar no *software*. Portanto, selecionamos as características que julgamos necessárias para termos uma rede de qualidade levando-se em conta os serviços utilizados pelos usuários.

Para isto, foi preciso coletar as estatísticas de todas as cinco LAN's e roteadores, assim como dos dois servidores.

Nas LAN's, é importante coletarmos informações sobre cada uma das aplicações utilizadas: FTP, HTTP e VoIP, pois estes são os serviços que serão utilizados pelos usuários de acordo com seus perfis.

Na aplicação de transferência de arquivos (FTP), precisamos analisar o tempo de resposta de download (em segundos) e os tráfegos enviado e recebido (em pacotes por segundo).

Já na aplicação de acesso à página da Internet (HTTP), vamos analisar somente os tráfegos recebido e enviado (em pacotes por segundo).

Por fim, na aplicação de comunicação telefônica via IP (VoIP), analisaremos a variação do atraso de pacote (*jitter*), o atraso fim-a-fim do pacote (em segundos) e os tráfegos recebido e enviado (em pacotes por segundo).

A seguir, temos uma figura com as estatísticas que foram selecionadas.

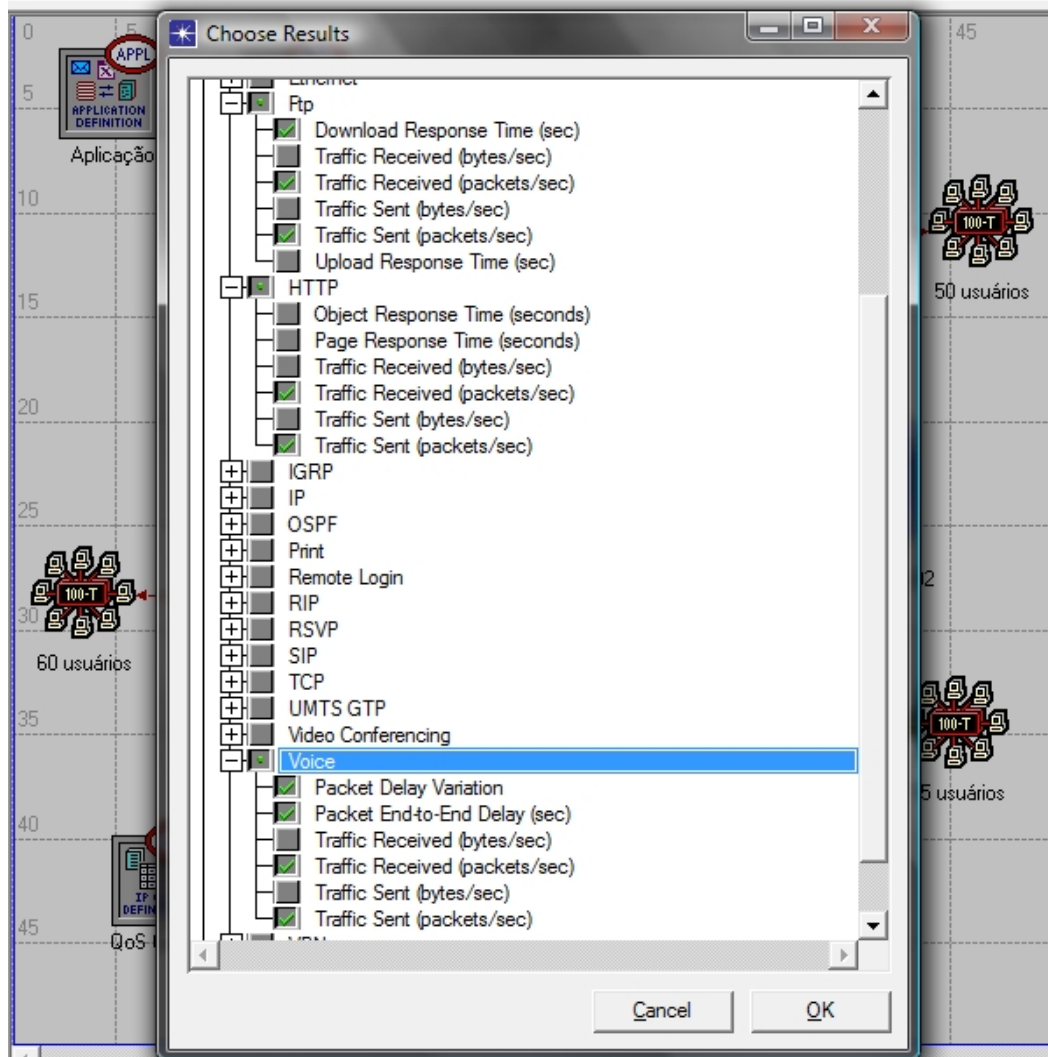


Figura 11 – Estatísticas coletadas dos usuários

Nos roteadores, iremos coletar apenas as informações referentes ao protocolo IP e na interface dos roteadores. Na interface, é onde ocorrem as filas de pacotes, tanto para entrar quanto para sair do roteador.

No protocolo IP, coletaremos informações sobre o tráfego enviado, o tráfego recebido e o tráfego perdido (todos em pacotes por segundo).

Já na interface dos roteadores, iremos coletar informações sobre o tráfego de pacotes perdidos que estão entrando e que estão saindo (em pacotes por segundo), o *buffer* usado (capacidade de armazenamento de pacotes) (em pacotes), a variação de atraso de fila (*jitter*) (em segundos), o atraso de fila (em segundos), o tráfego enviado, o tráfego recebido e o tráfego perdido (em pacotes por segundo). Na interface, se faz extremamente importante analisar o comportamento dos pacotes quando estão entrando ou saindo e a capacidade do *buffer*, pois são estes parâmetros que dirão como as filas estão se comportando (por exemplo, se há congestionamento de pacotes nos roteadores).

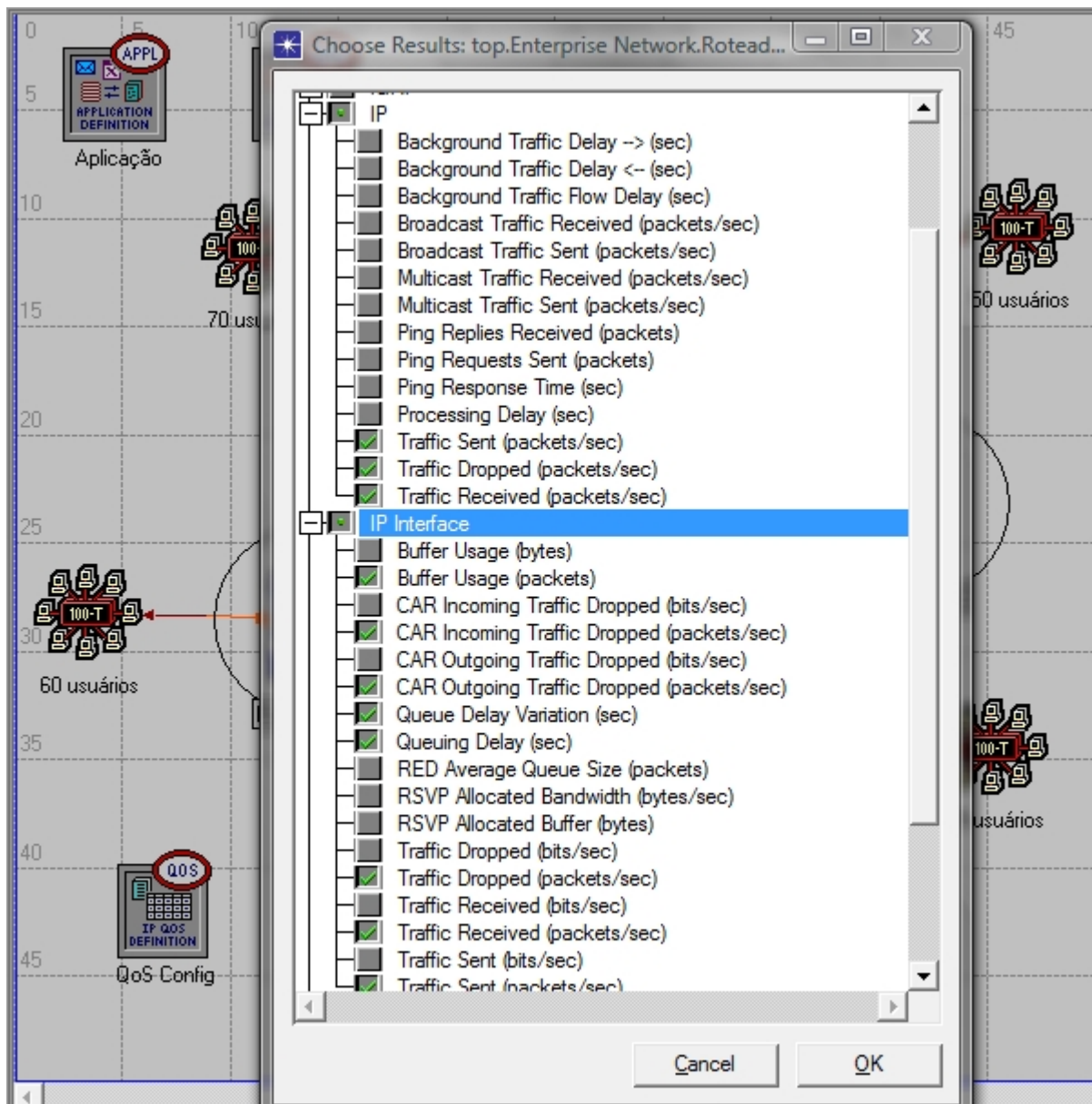


Figura 12 – Estatísticas coletadas nos roteadores

Para os servidores, basta coletarmos estatísticas dos tráfegos enviados, recebidos e perdidos (em pacotes por segundo) para cada um dos servidores e suas respectivas aplicações.

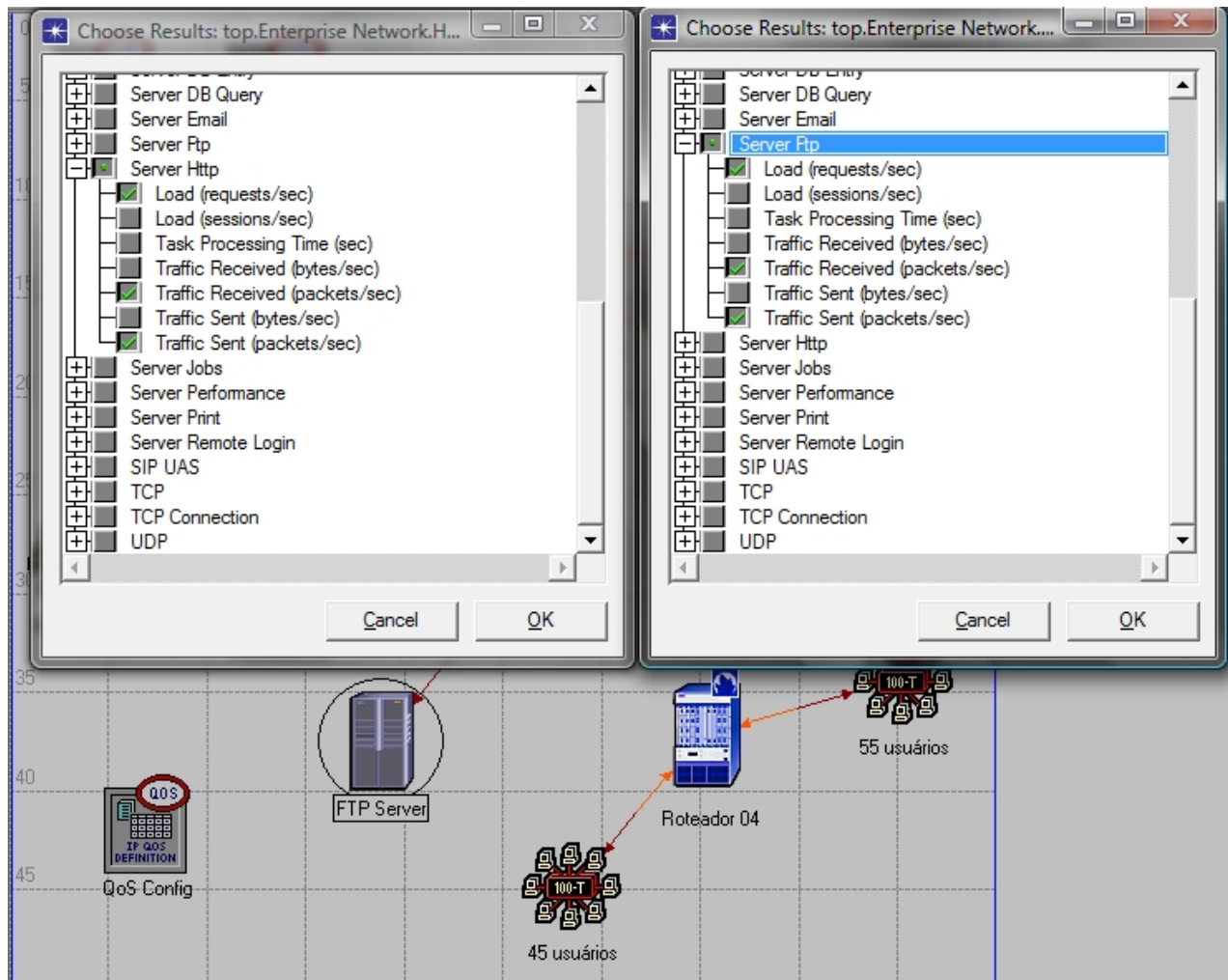


Figura 13 – Estatísticas coletadas nos servidores

## b. Simulação da rede

Abordamos anteriormente, a necessidade de criarmos quatro redes idênticas com algumas configurações diferentes a fim de verificarmos qual delas possui o melhor desempenho.

No OPNET, podemos simular estas redes simultaneamente de modo a comparar seus desempenhos através dos gráficos que serão obtidos, podendo ainda gerenciar os chamados “cenários” da forma que quisermos.

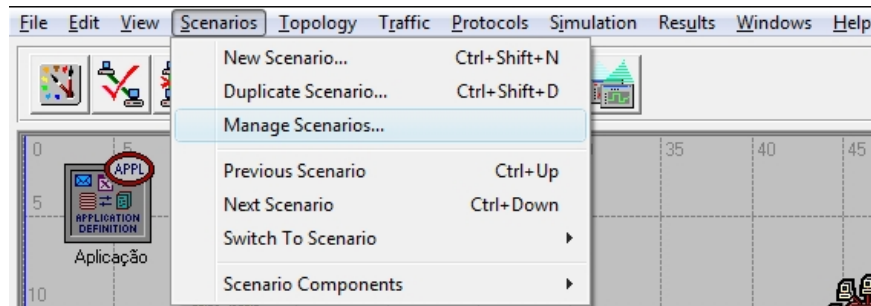


Figura 14 – Menu para gerenciar cenários diferentes

Acessando esse caminho, temos:

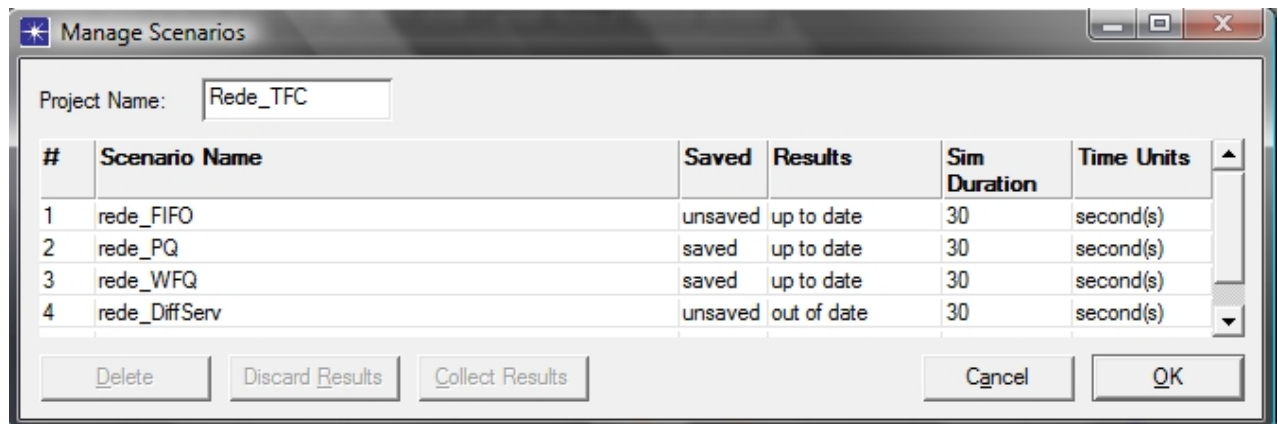


Figura 15 – Gerenciando cenários

Na primeira coluna, temos a quantidade de cenários que podem ser gerenciados para simulação.

Na segunda coluna, temos os nomes dos cenários (nossas redes).

Na quarta coluna, podemos escolher entre coletar ou descartar algum resultado.

Na quinta e sexta colunas, inserimos a duração e a unidade de tempo que a simulação deve possuir.

Em nosso projeto, foram coletados os resultados de todos os cenários, cada um tempo de simulação com duração de 30 segundos e em seguida demos OK.

Após essa etapa e alguns minutos, os resultados estarão disponíveis para análise e comparação.



## 5. Resultados Obtidos Graficamente e Comentários

Para visualizarmos os resultados obtidos e podermos compará-los, basta irmos ao menu do OPNET:

**Results → Compare Results**, e então visualizamos a seguinte tela:

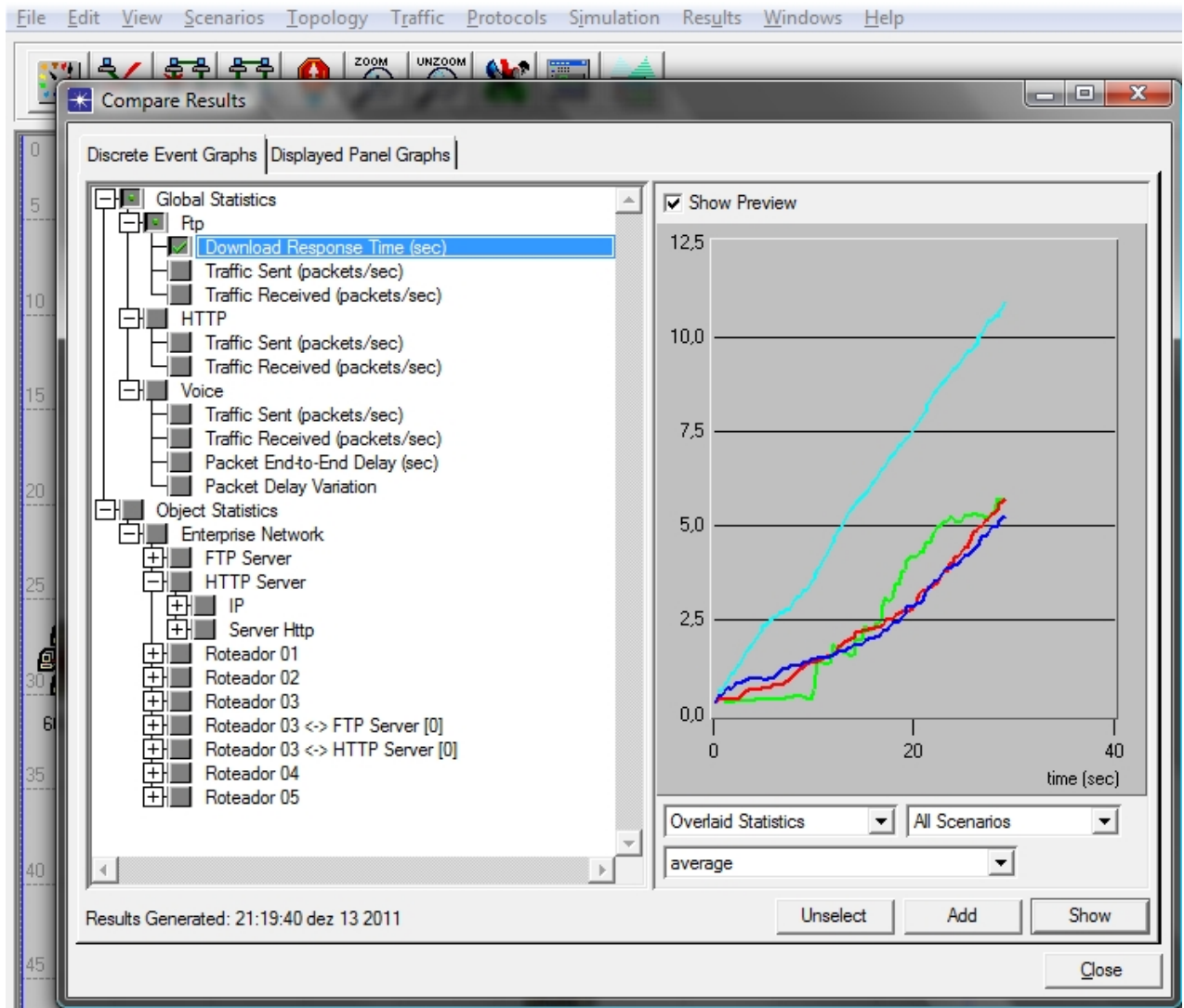


Figura 16 – Resultados obtidos

À esquerda da tela acima, podemos selecionar as estatísticas a serem visualizadas. Em “*Global Statistics*”, temos as estatísticas resultantes dos usuários. Em “*Object Statistics*”, podemos visualizar as estatísticas tanto dos roteadores quanto dos servidores.

À direita da tela acima, temos os gráficos obtidos e algumas opções de visualização:

→ No campo onde está escrito “*All Scenarios*”, podemos escolher a exibição de quantos cenários disponíveis quisermos;

→ No campo onde está escrito “*Overlaid Statistics*”, temos as opções de exibir os gráficos separadamente, sobrepostos ou individualmente;

→ No campo onde está escrito “*Average*”, escolhemos o tipo de valor a ser considerado. Na nossa rede, consideraremos as médias de cada estatística.

Resultados obtidos a partir das estatísticas dos serviços dos usuários:

→ **FTP:**

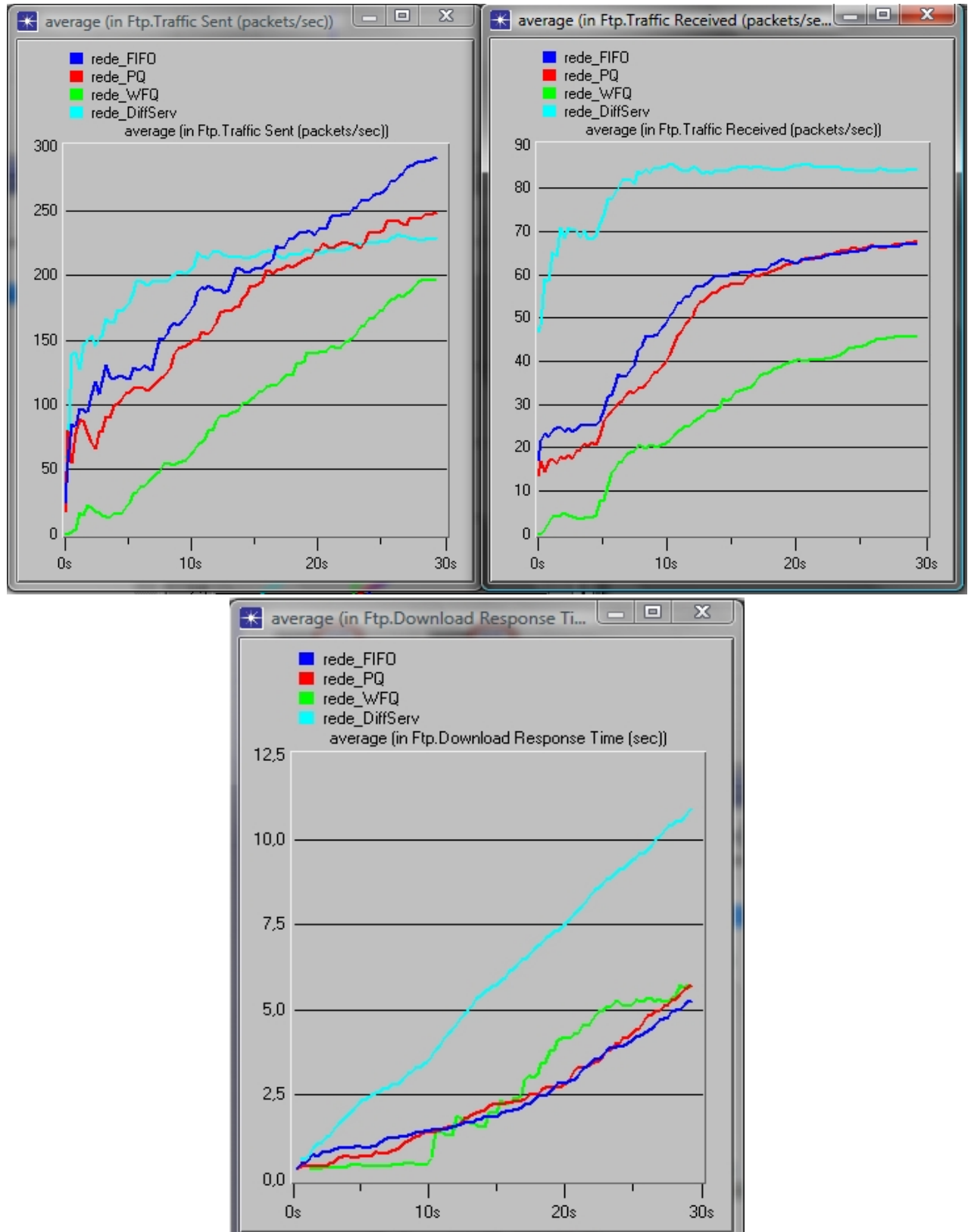


Figura 17 – Resultados FTP: *Traffic Sent*, *Traffic Received* e *Download Response Time* (packets/sec)



A rede que utiliza o método de escalonamento dos sistemas de filas FIFO possui um tráfego de pacotes consideravelmente maior que os outros durante o período de simulação.

No tráfego recebido, a rede que utiliza apenas *Diffserv* tem um melhor desempenho, pois está utilizando a estrutura de priorização de pacotes EF (*Expedited Forwarding*), ou seja, os pacotes possuem alta prioridade, admitindo baixas perdas de pacotes durante o tráfego.

No último gráfico, nota-se que a rede *Diffserv* o tempo de resposta de *download* muito pior que as outras. Isso se dá pelo fato de não possuir nenhum método de escalonamento de filas.

→ **HTTP:**

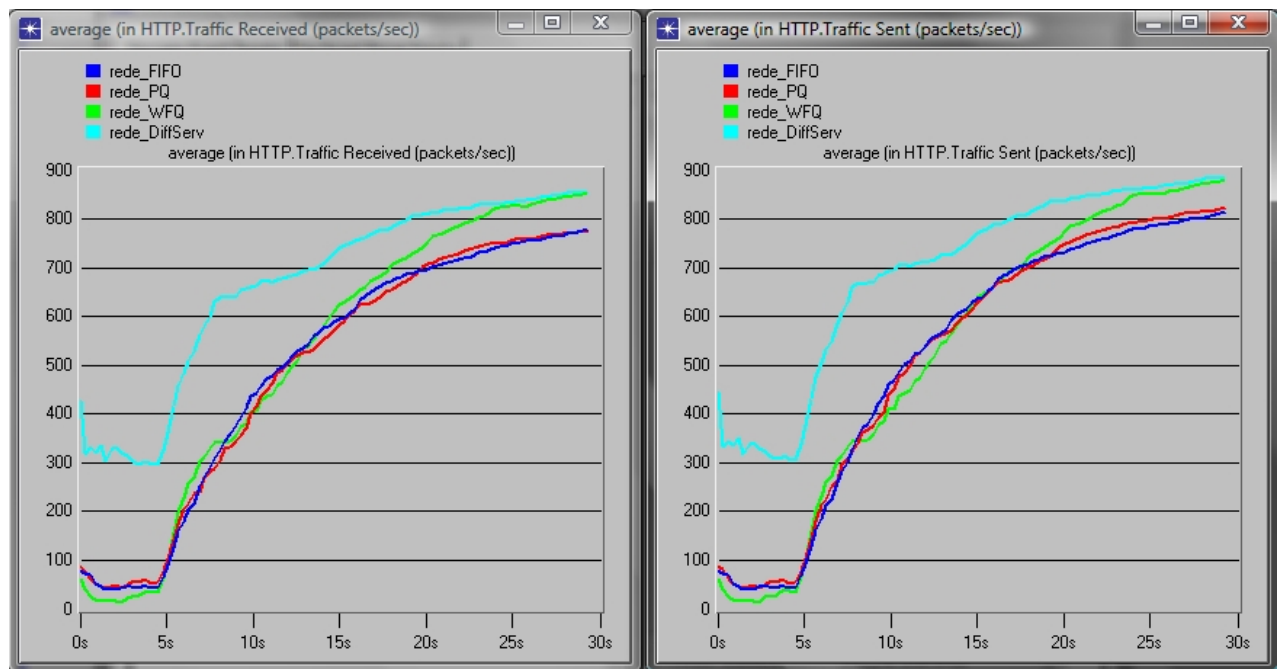


Figura 18 – Resultados HTTP: *Traffic Received* e *Traffic Sent*

Nesse tipo de aplicação (HTTP), não é possível gerar gráfico de perda de pacotes, mas podemos facilmente observar que houve perdas.

No tráfego recebido, temos valores de pacotes entre, aproximadamente, 780 (para as redes FIFO e PQ) e 840 (para as redes WFQ e Diffserv).

Já no tráfego enviado, os valores variam entre, aproximadamente, 810 (para as redes FIFO e PQ) e 880 (para as redes WFQ e Diffserv).

→ **VoIP:**

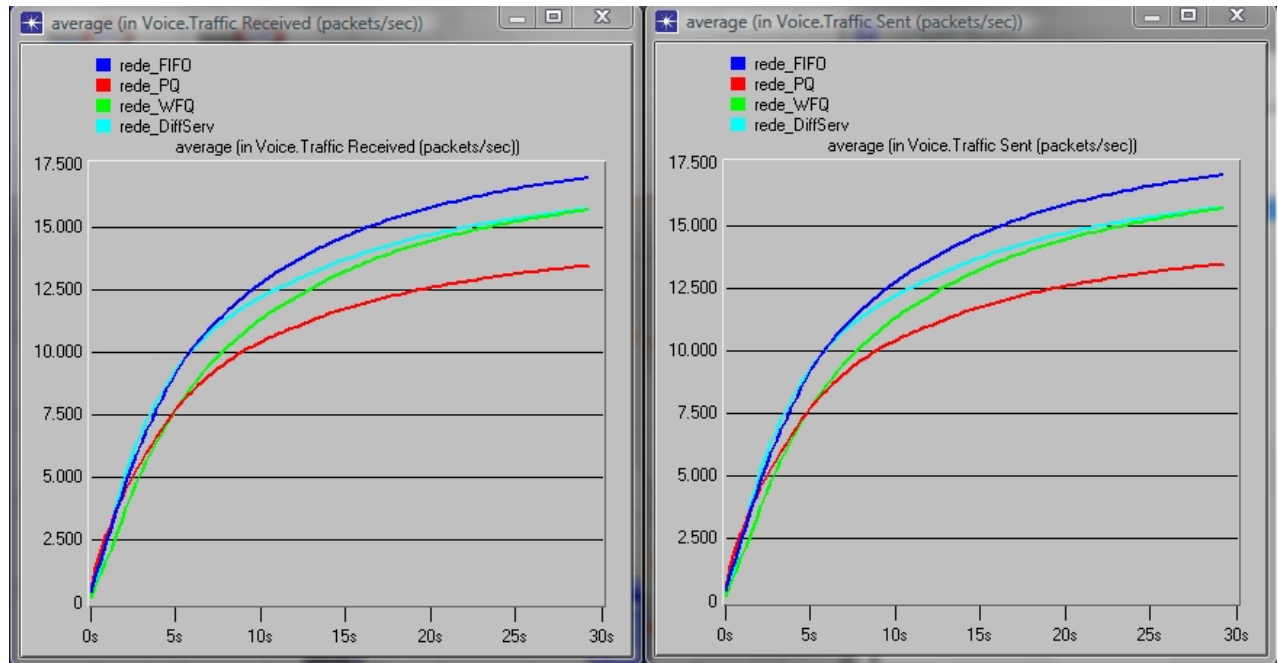


Figura 19 – Resultados VoIP: *Traffic Received* e *Traffic Sent*

Podemos notar que não houve perda de pacotes, pois os tráfegos enviado e recebido são iguais. Nota-se também que o desempenho da rede FIFO é mais eficiente.

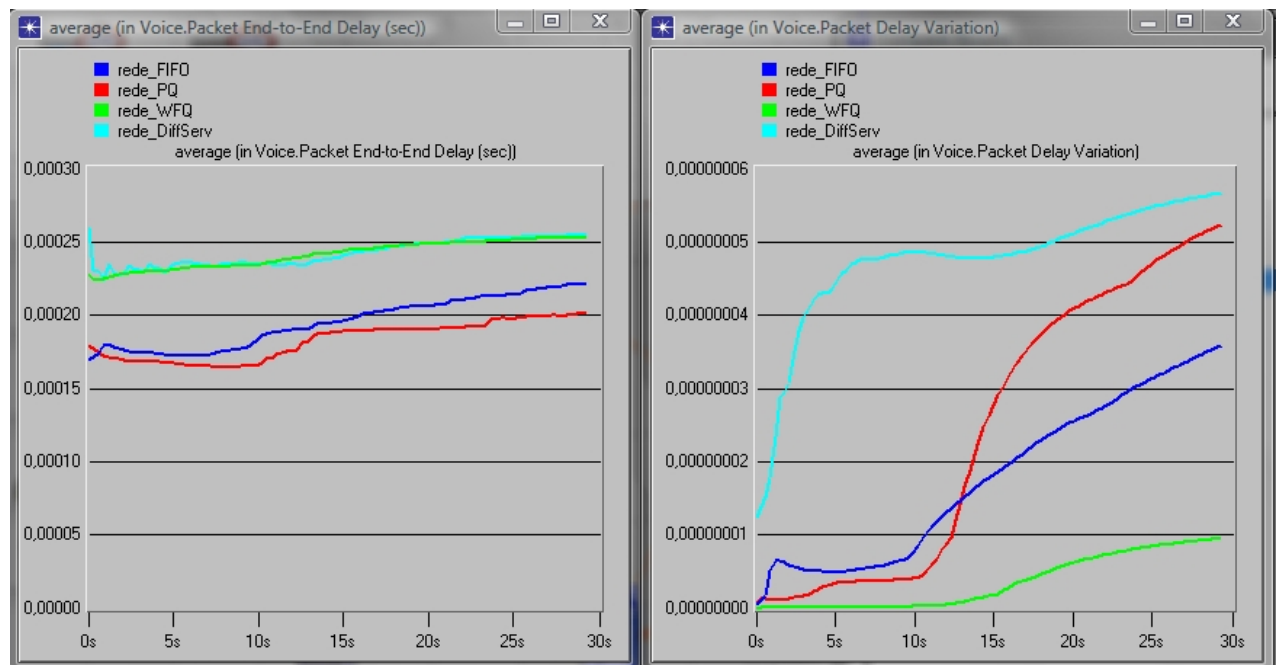


Figura 20 – Resultados VoIP: *Packet End-to-End Delay* e *Packet Delay Variation (jitter)*

Para ambos os casos, podemos observar que a inserção de QoS nas redes surtiu efeito: tanto o atraso de pacotes fim-a-fim quanto o *jitter* de pacotes são muito pequenos.

Porém, a fim de comparação, o atraso de pacotes fim-a-fim da rede PQ é a menor (o método PQ gera prioridades a certos pacotes, evitando assim atrasos), enquanto o *jitter* de pacotes é menor na rede WFQ.

Resultados obtidos a partir das estatísticas dos servidores:

→ **Servidor FTP:**

- Load (requests/sec)

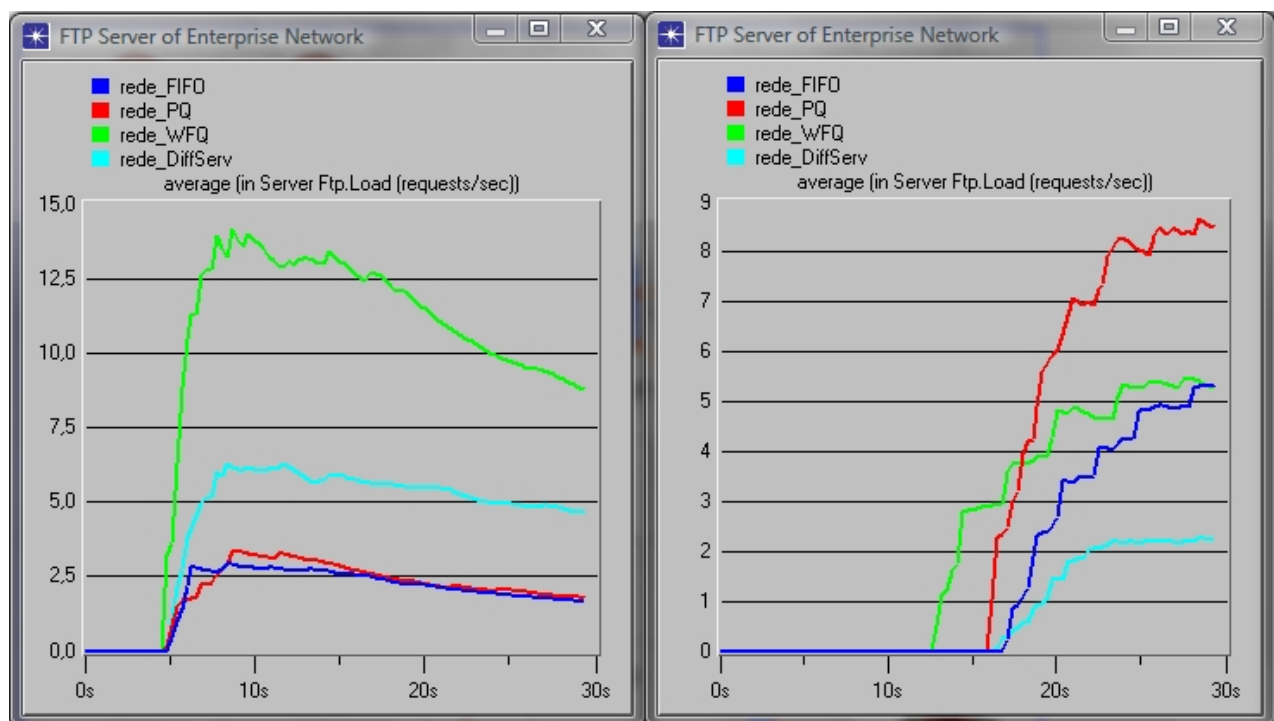


Figura 21 – Resultados Servidor FTP: Load (High e Medium)

O número de requisições de transferências de arquivos de grandes tamanhos é maior na rede WFQ uma vez que, neste método de escalonamento de filas, os pacotes de maiores portes recebem prioridade no fluxo.

Por outro lado, nota-se que, para transferências de arquivos de tamanho médio, o número de requisições de transferência de arquivos é maior para a rede PQ. Isso pode ser explicado pelo fato do método de escalonamento de filas PQ dar prioridade a pacotes de determinado tamanho.

Para o caso de requisições de transferências de arquivos de pequeno porte, não houve necessidade de priorização de pacotes. Sendo assim, a rede FIFO possui o maior número de requisições por segundo, já que, neste método de escalonamento de filas, os pacotes que entram primeiro no roteador, saem primeiro do roteador (*First-In-First-Out*).

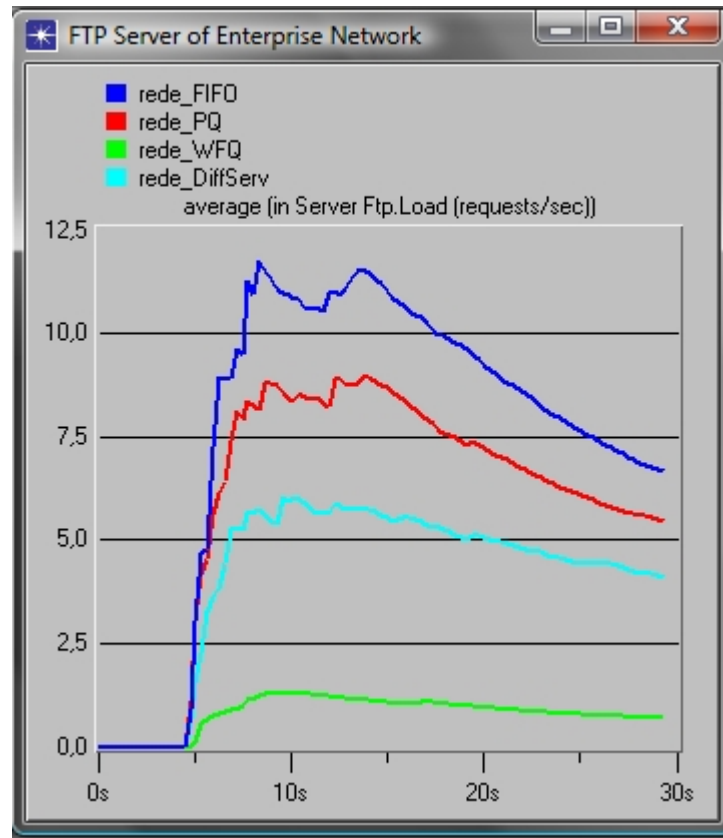


Figura 22 – Resultados Servidor FTP: Load (Low)

- Traffic Sent (packets/sec)

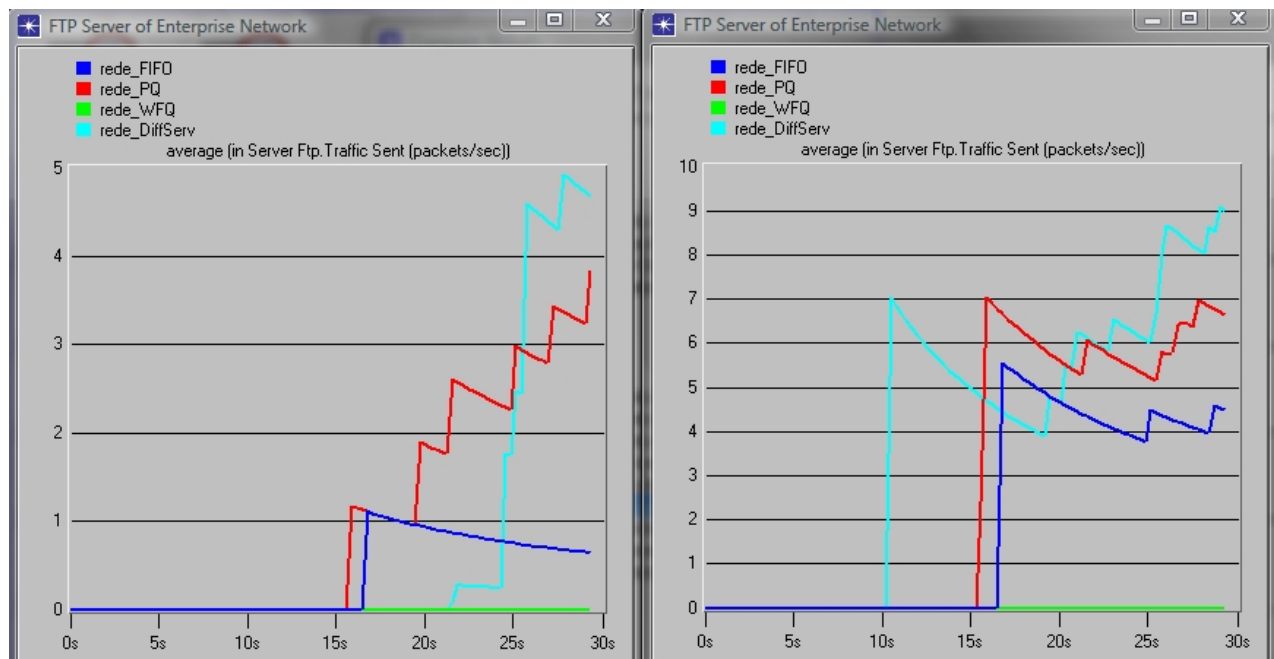


Figura 23 – Resultados Servidor FTP: Traffic Sent (High e Low)

No servidor FTP não houve tráfego de carga média. Entretanto, observa-se que o tráfego de grande tamanho só começa a fluir depois de 15 segundos e o de pequeno tamanho após 10 segundos. Podemos atribuir este fato à demora do tempo de resposta de download observado na Figura 17.

- *Traffic Received (requests/sec)*

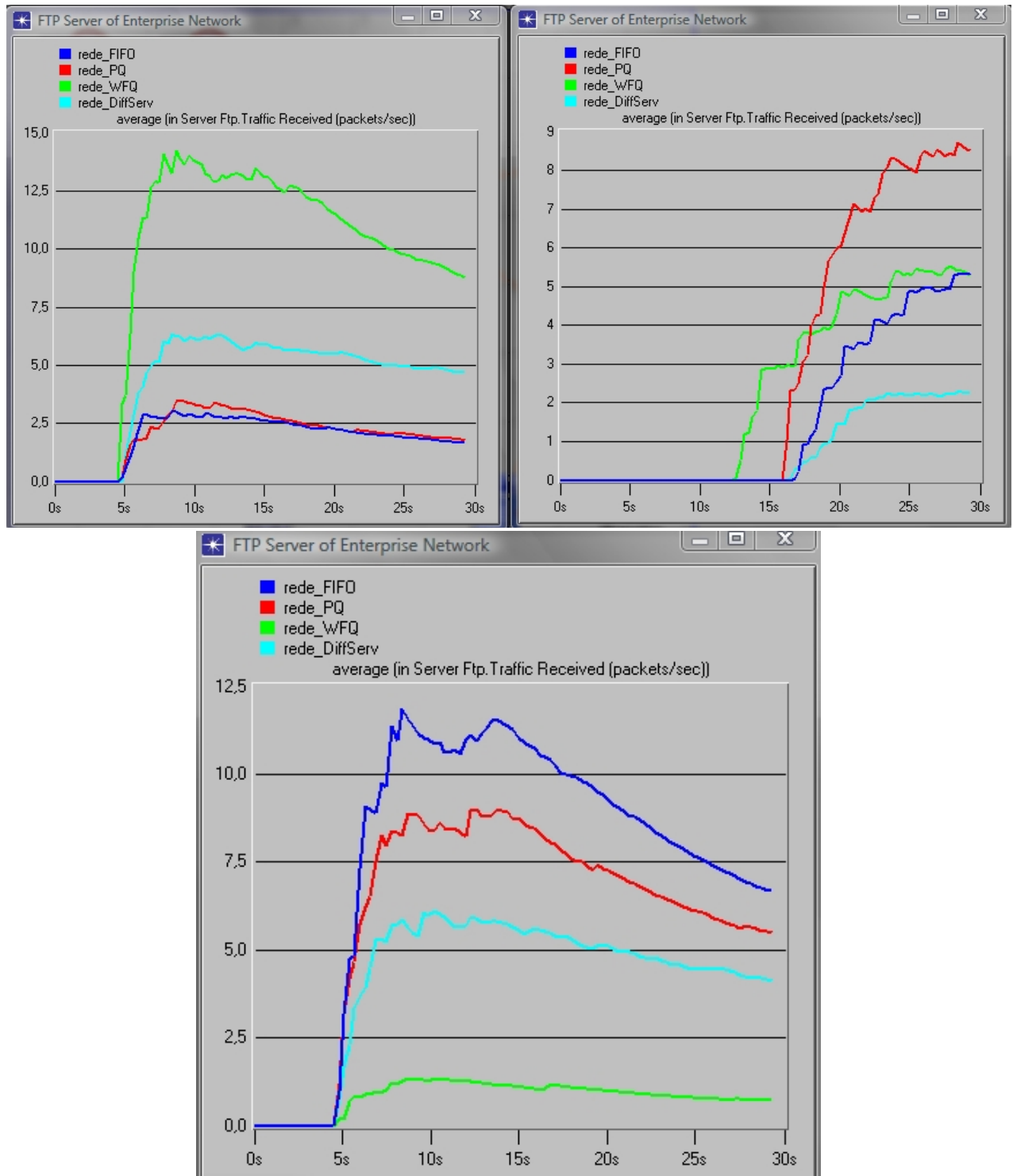


Figura 24 – Resultados Servidor FTP: *Traffic Received (High, Medium e Low)*

Como esperado e por coerência, os resultados obtidos na figura acima, são idênticos aos das Figuras 21 e 22. Sendo assim, os mesmos comentários realizados naquele tópico, servem para este também.

→ **Servidor HTTP:**

- Tráfego IP (*requests/sec*)

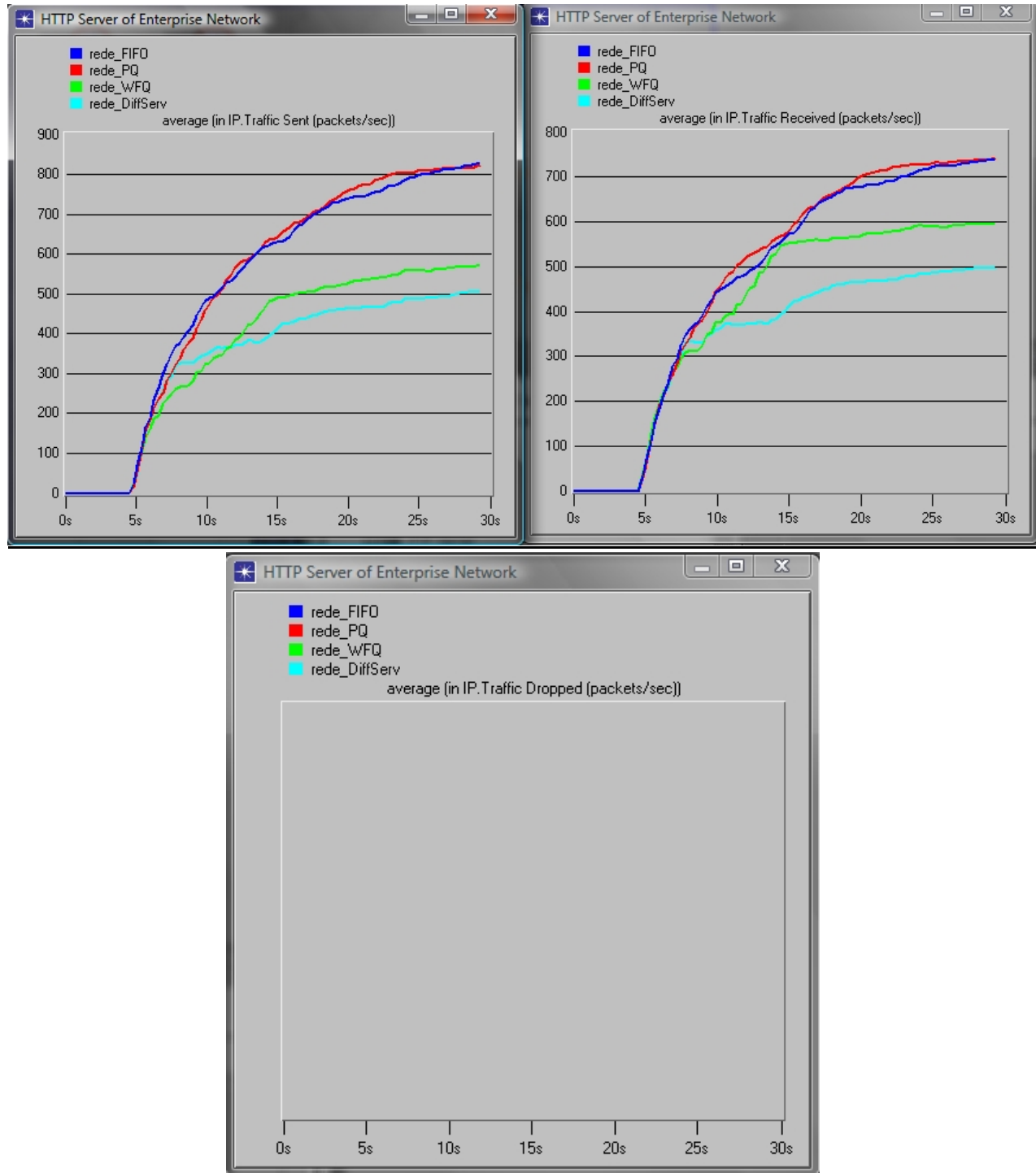


Figura 25 – Resultados Servidor HTTP: Tráfego IP (*Sent, Received e Dropped*)

Pode-se notar nos gráficos dos tráfegos enviado e recebido, que houve perda de pacotes. Porém, essas perdas não puderam ser detectadas na simulação. A explicação deste fato será dada mais adiante. Observa-se que as redes FIFO e PQ possuem aproximadamente as mesmas quantidades de pacotes enviadas e recebidas.

- Load (requests/sec)

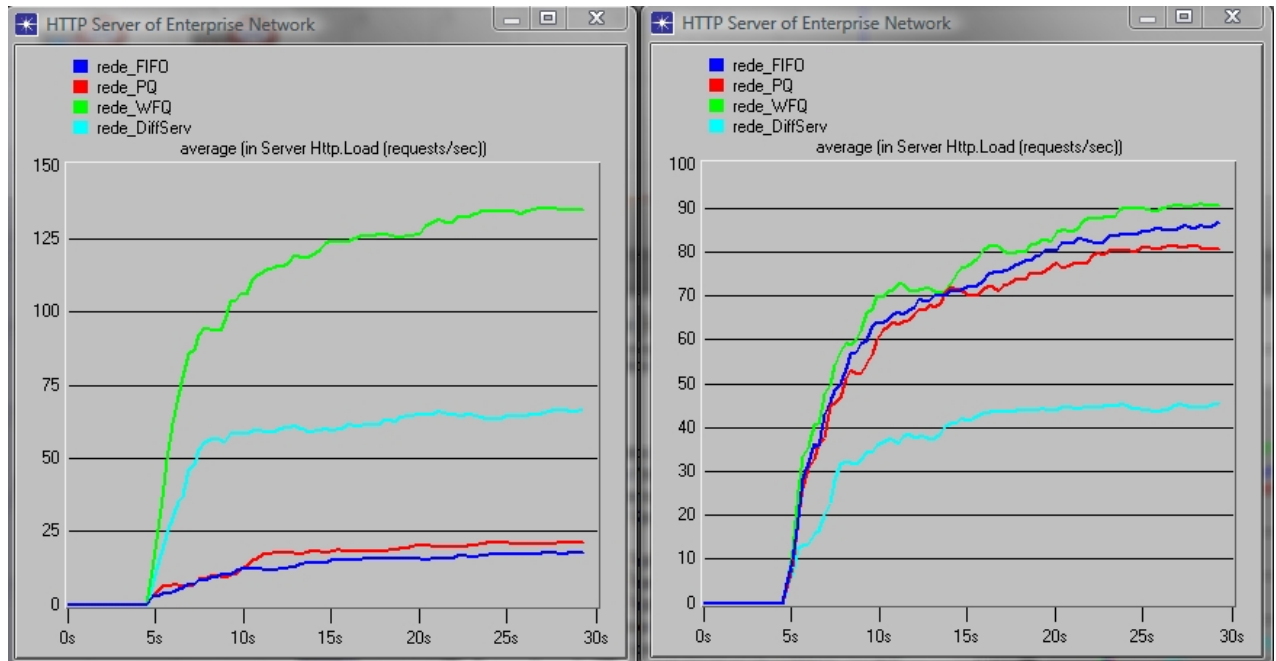


Figura 26 – Resultados Servidor HTTP: Load (Heavy Browsing e Light Browsing)

Para uma navegação “pesada” (*Heavy Browsing*) (sites muitos “detalhes”), a rede WFQ torna-se mais requisitada em função do seu método de escalonamento de filas WFQ, que dá prioridade ao fluxo considerado pesado. No caso de uma navegação “leve” (*Light Browsing*), todas as redes que usam algum método de escalonamento de filas têm bom desempenho, visto que, sendo a navegação “leve”, não há porquê priorizar certos fluxos.

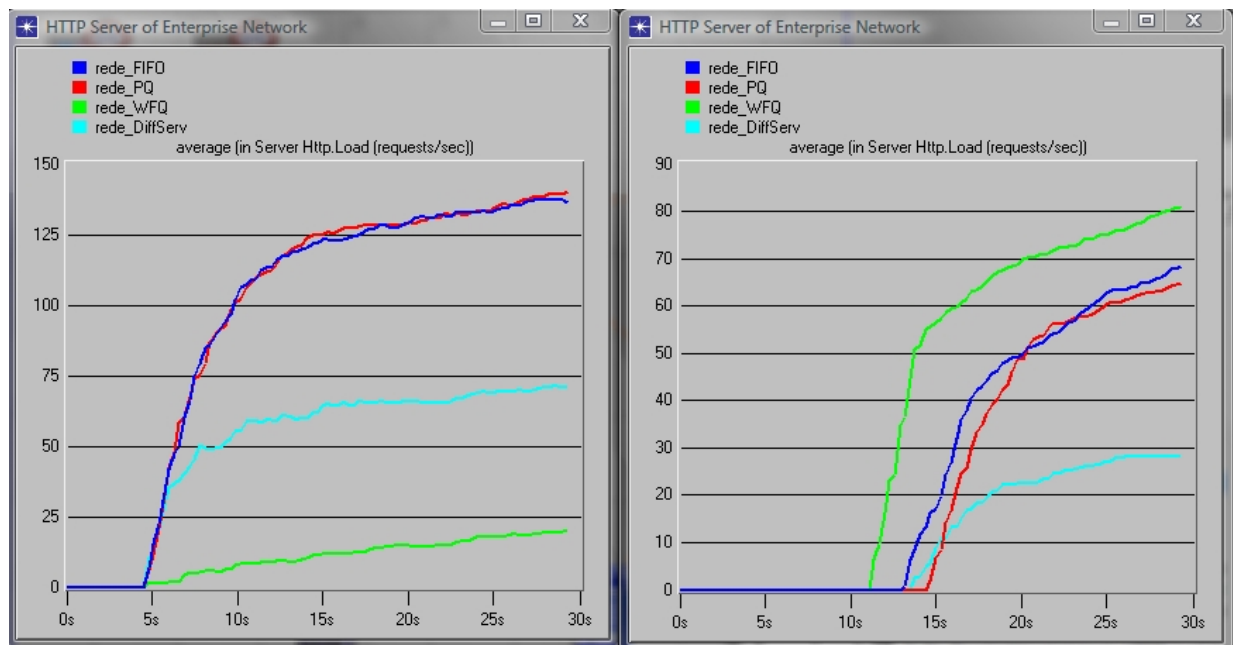


Figura 27 – Resultados Servidor HTTP: Load (Image Browsing e Searching)

Quando lidamos apenas com navegação de imagens, não precisamos priorizar pacotes. Então, a rede FIFO e PQ têm desempenhos ótimos e bem parecidos. Já no caso de uma navegação para pesquisa, o tempo entre a busca e a resposta sobre a informação requisitada é considerada demorada. Neste caso,



a rede WFQ tem melhor desempenho, pois a pesquisa é realizada em toda a Internet, sendo preciso priorizar o fluxo.

Uma vez que os resultados obtidos nas Figuras 26 e 27 são referentes à quantidade de tráfego enviado e recebido no servidor HTTP, veremos nas figuras seguintes (Figuras 28 à 31) que os gráficos são idênticos como esperado.

- *Traffic Sent (requests/sec)*

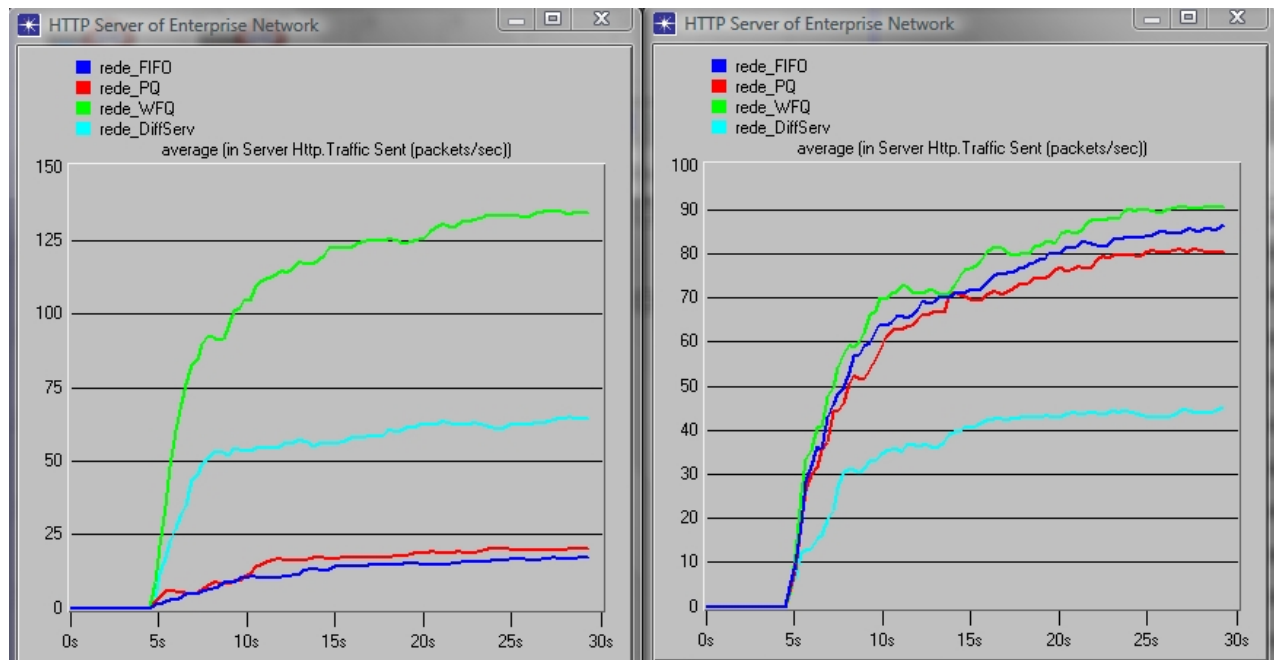


Figura 28 – Resultados Servidor HTTP: *Traffic Sent (Heavy Browsing e Light Browsing)*

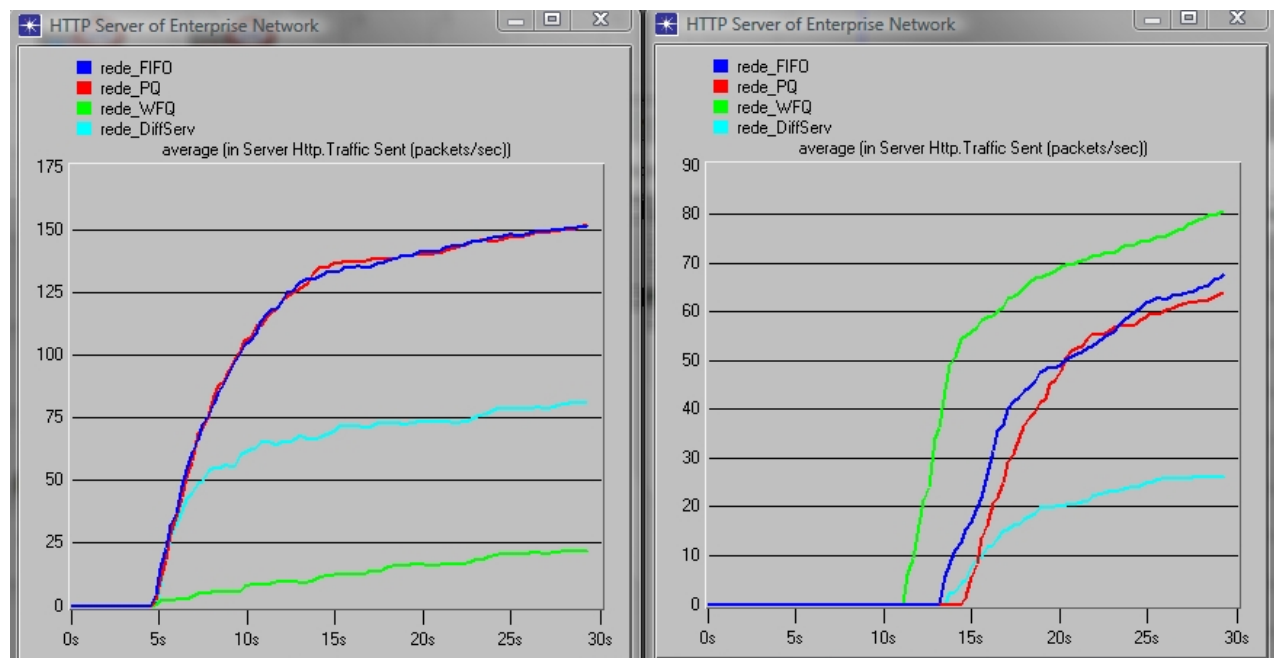


Figura 29 – Resultados Servidor HTTP: *Traffic Sent (Image Browsing e Searching)*



- *Traffic Received (requests/sec)*

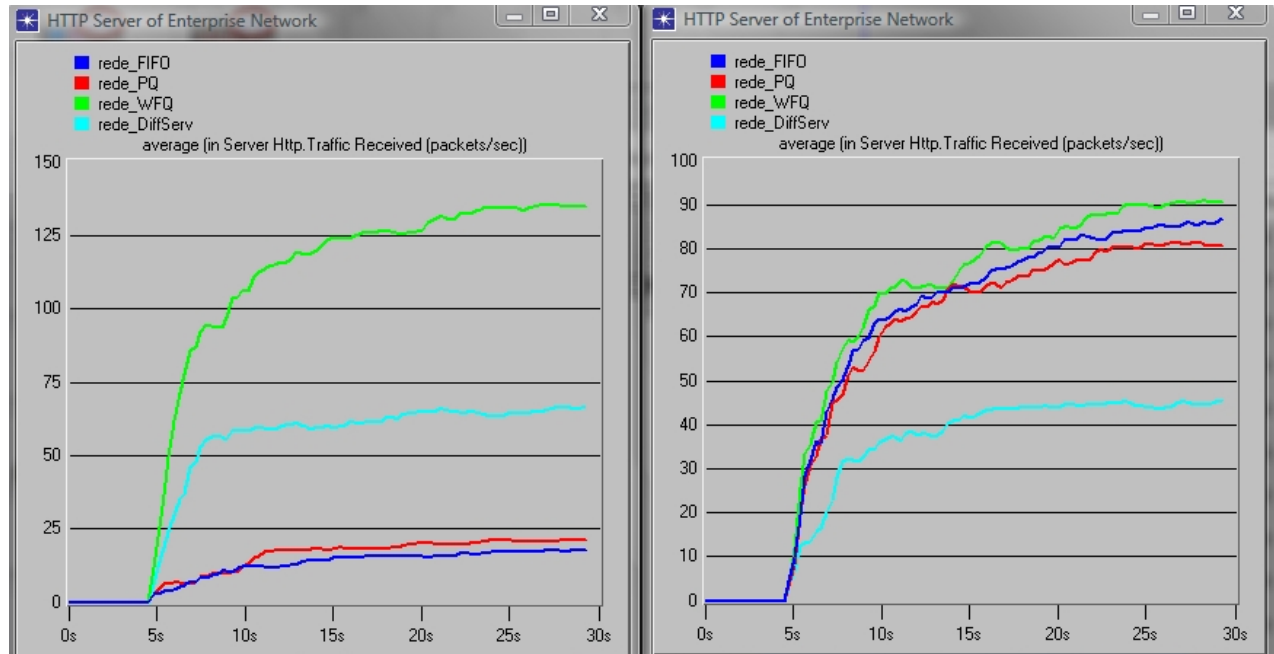


Figura 30 – Resultados Servidor HTTP: *Traffic Received (Heavy Browsing e Light Browsing)*

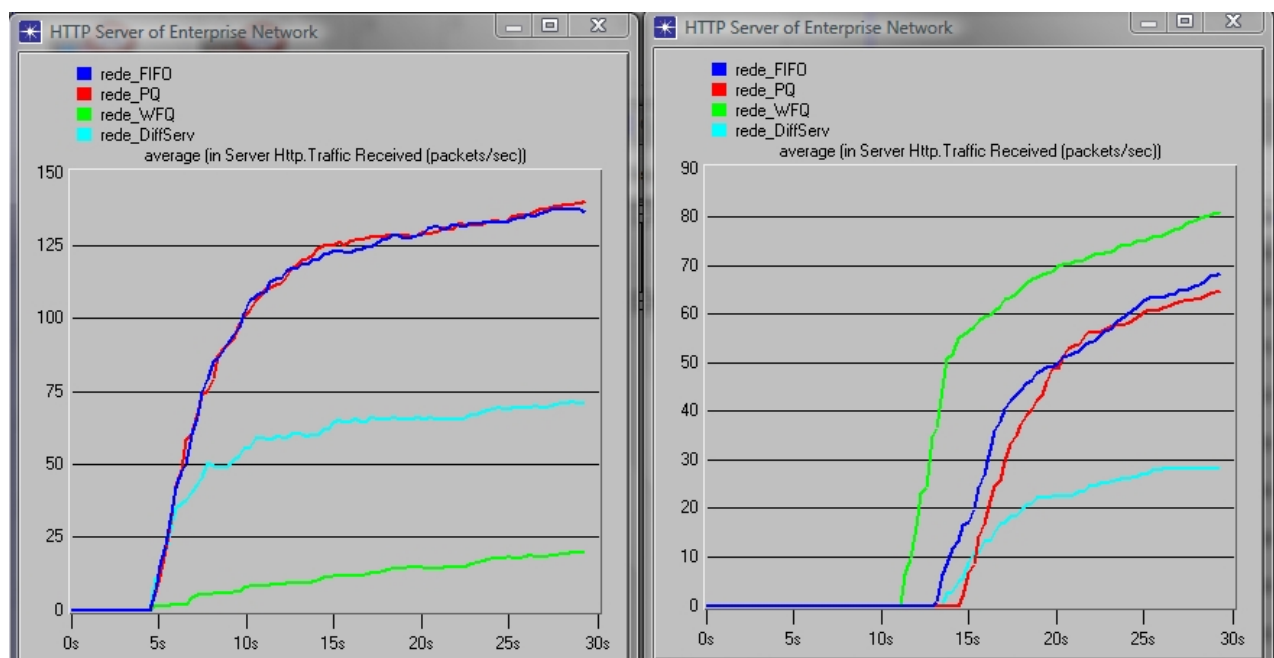


Figura 31 – Resultados Servidor HTTP: *Traffic Received (Image Browsing e Searching)*

Resultados obtidos a partir das estatísticas dos roteadores:

## → Roteador 01:

- Tráfego IP (*packets/sec*)

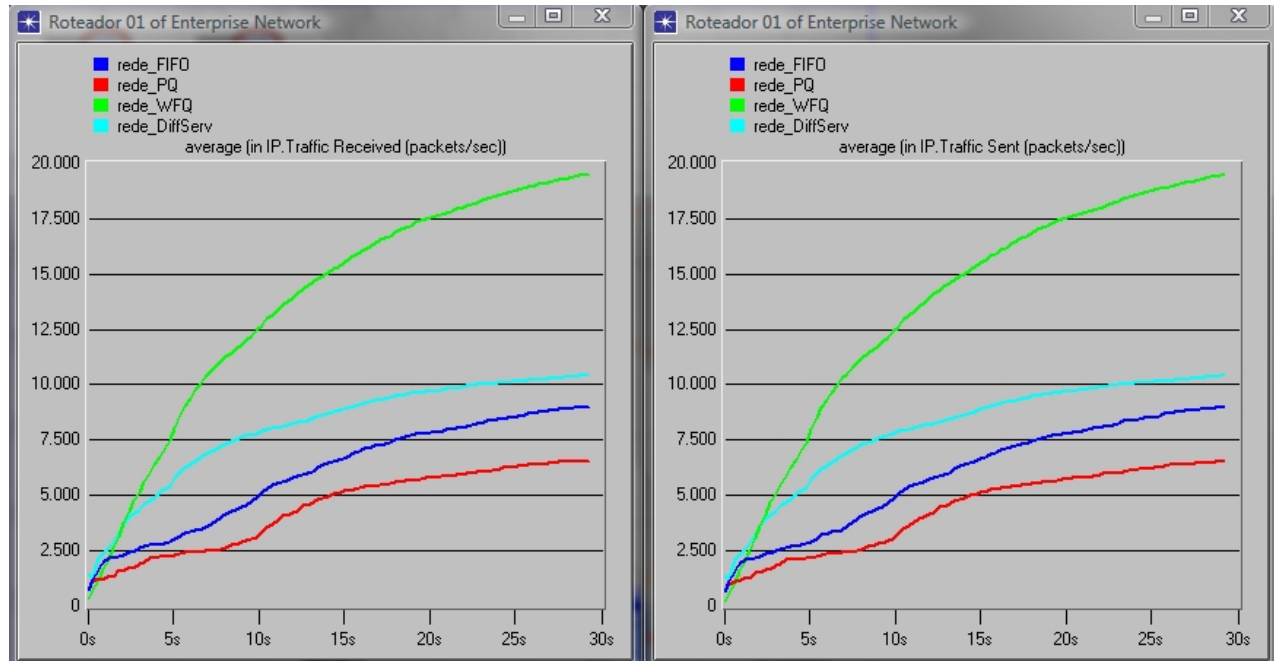


Figura 32 – Resultados obtidos no Roteador 01: Tráfego IP (*Received e Sent*)

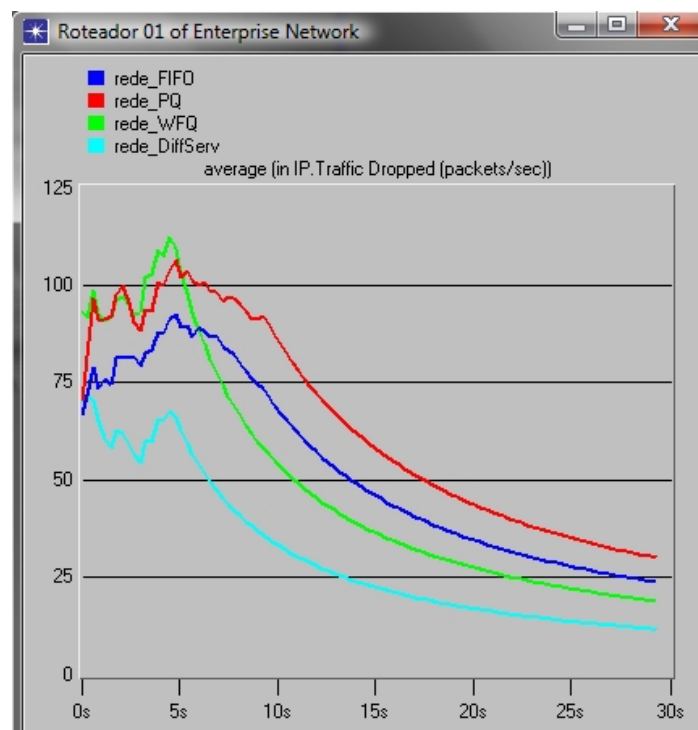


Figura 33 – Resultados obtidos no Roteador 01: Tráfego IP (*Dropped*)

A perda de pacotes ocorre principalmente nos primeiros segundos da simulação. Após isto, essa perda cai exponencialmente até o fim da simulação. Pode-se observar que o tráfego de pacotes enviados e recebidos aumenta no mesmo período que as perdas diminuem.

→ **Roteador 02:**

- Tráfego IP (*packets/sec*)

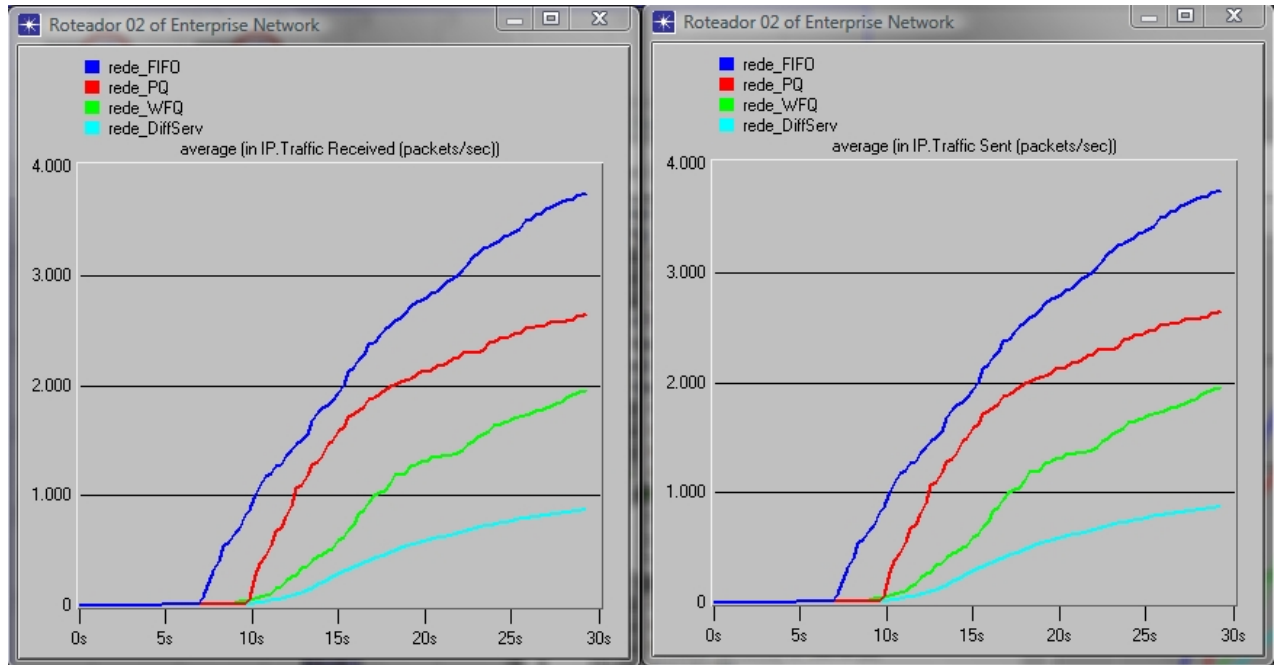


Figura 34 – Resultados obtidos no Roteador 02: Tráfego IP (*Received e Sent*)

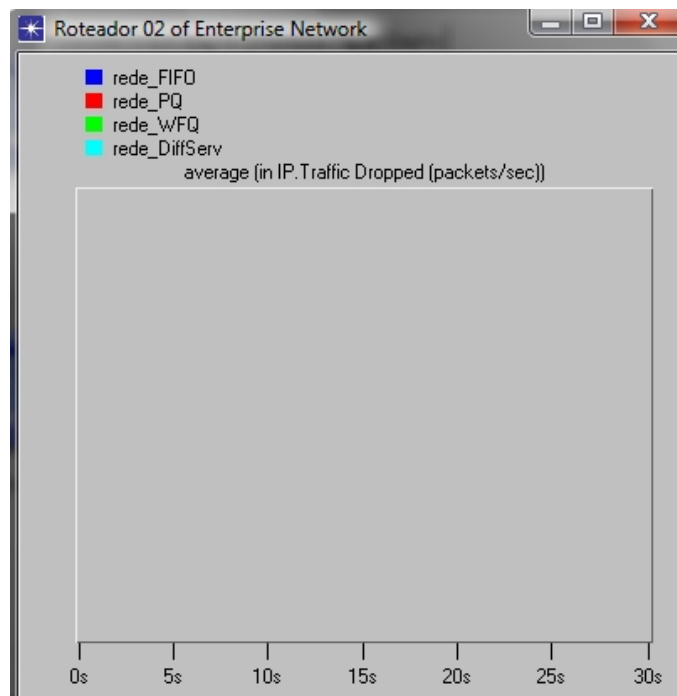


Figura 35 – Resultados obtidos no Roteador 02: Tráfego IP (*Dropped*)

O Roteador 02 tem um funcionamento praticamente perfeito: sendo um “roteador de passagem” entre os roteadores 01 e 04, todo tráfego que passa por ele é repassado para o roteador seguinte sem perda alguma.

→ **Roteador 03:**

- Tráfego IP (*packets/sec*)

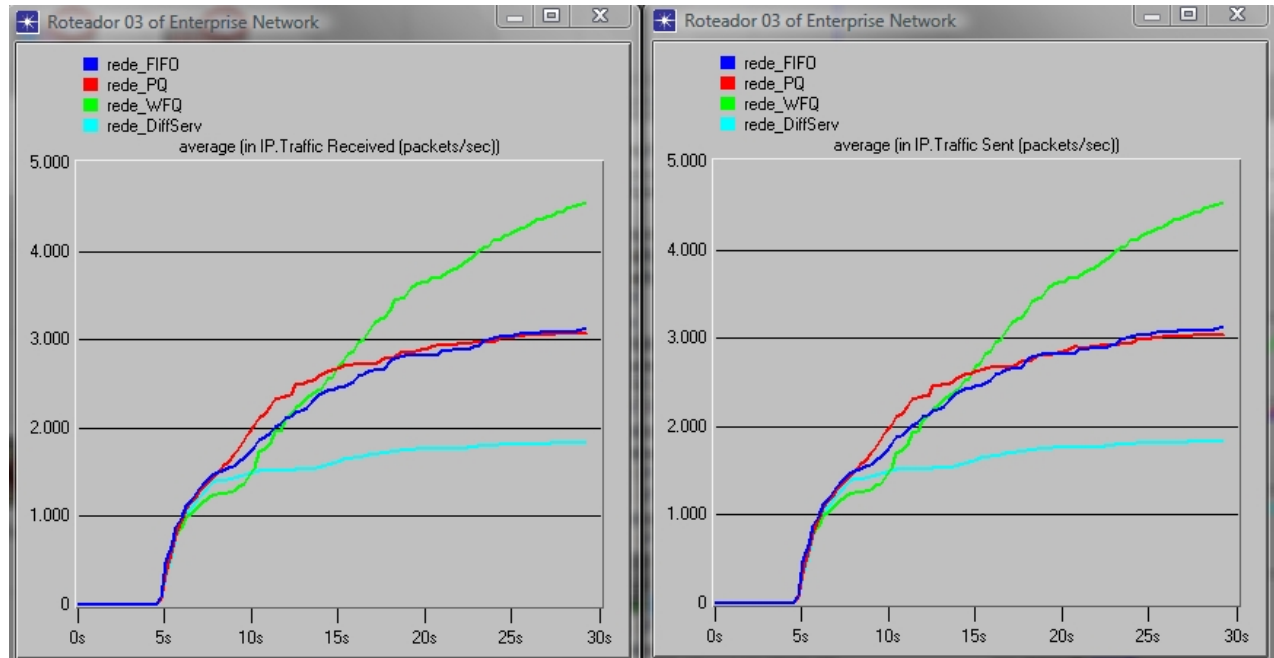


Figura 36 – Resultados obtidos no Roteador 03: Tráfego IP (*Received e Sent*)

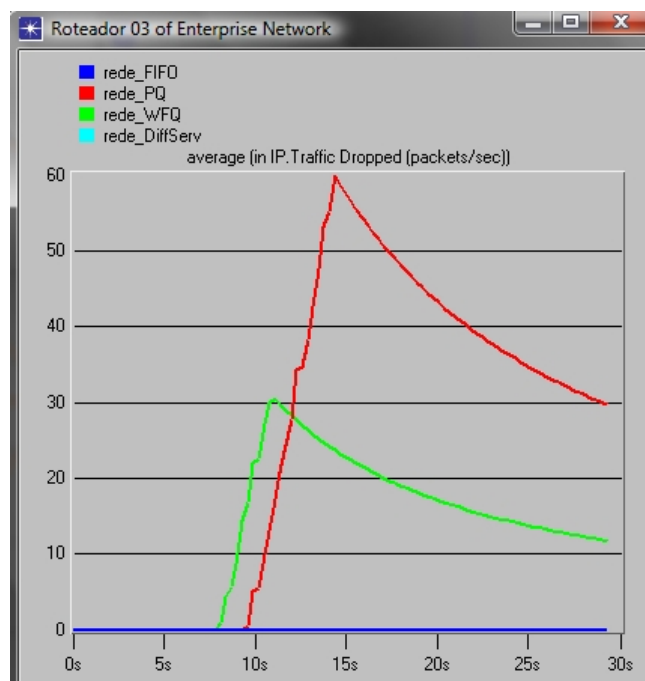


Figura 37 – Resultados obtidos no Roteador 03: Tráfego IP (*Dropped*)

Apesar do tráfego enviado e recebido da rede WFQ apresentar um maior fluxo de pacotes que as outras redes, possui uma perda de pacotes considerável. Sendo assim, em função do desempenho da rede FIFO ter um bom fluxo de pacotes, torna-se mais interessante sua utilização pelo fato de não ter perda de pacotes em todo seu processo.

→ **Roteador 03 (comunicação com o FTP Server):**

- *Throughput Point-to-point (packets/sec)*

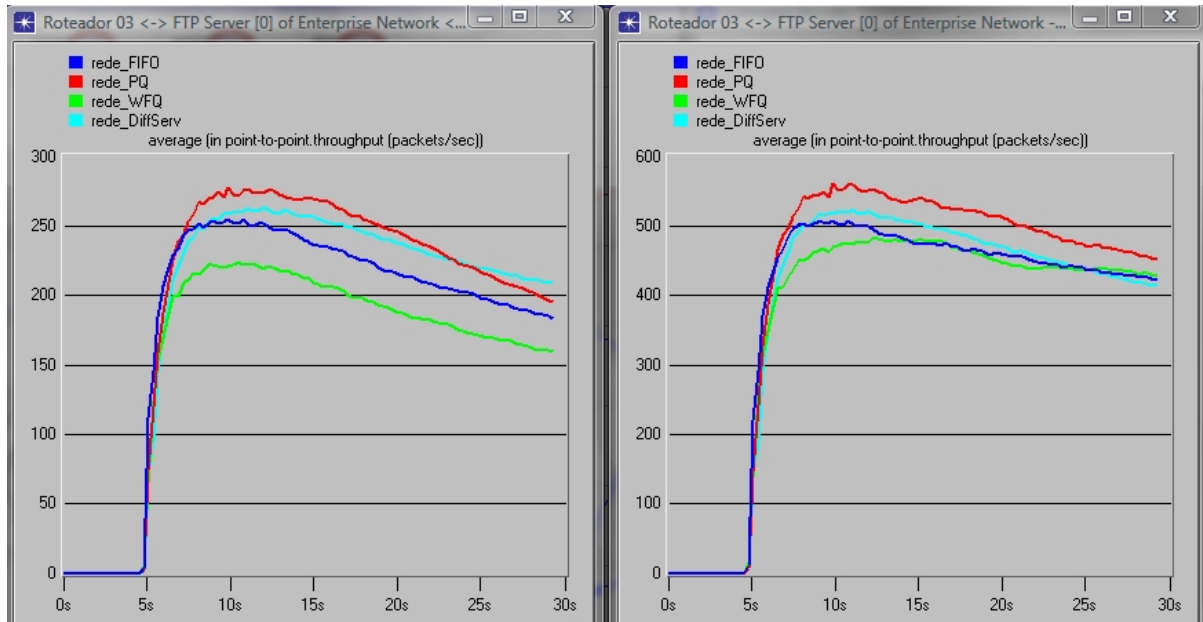


Figura 38 – Resultados obtidos na comunicação entre o Roteador 03 e o FTP Server: Rendimento (dos pacotes que estão entrando no Roteador 03 vindos do FTP Server e dos pacotes que estão saindo do Roteador 03 indo para o FTP Server)

O rendimento da rede PQ é melhor em ambos os casos pelo fato de priorizar certos pacotes nas filas existentes nos roteadores

→ **Roteador 03 (comunicação com o HTTP Server):**

- *Throughput Point-to-point (packets/sec)*

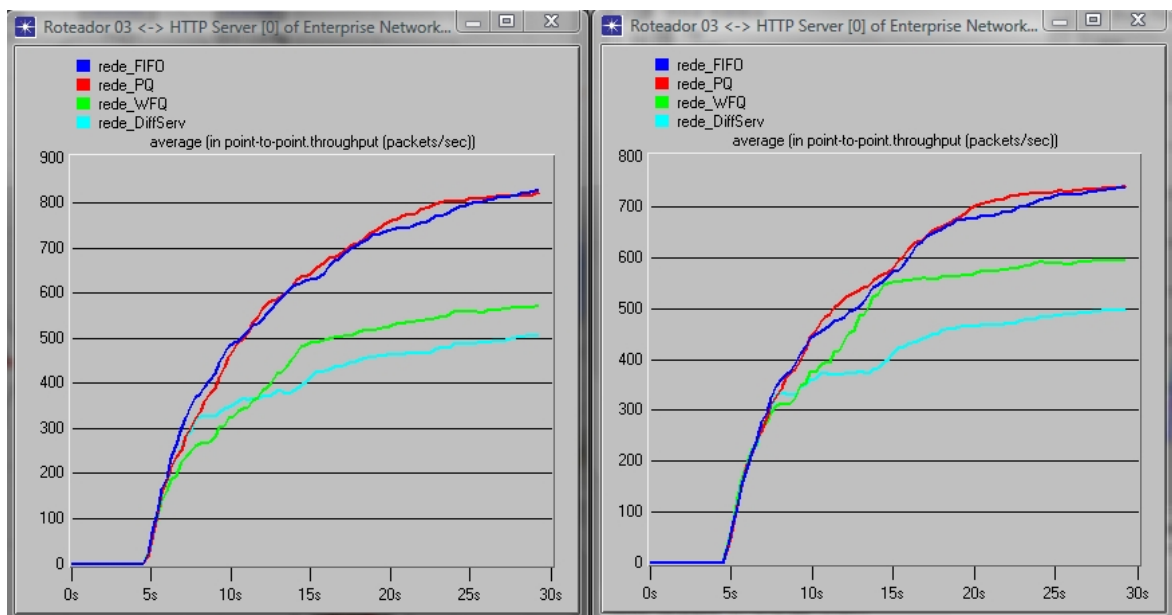


Figura 39 – Resultados obtidos na comunicação entre o Roteador 03 e o HTTP Server: Rendimento (dos pacotes que estão entrando no Roteador 03 vindos do HTTP Server e dos pacotes que estão saindo do Roteador 03 indo para o HTTP Server)

Tanto para entrada quanto para saída de pacotes no roteador 03, os melhores rendimentos são dados pelas redes que utilizam FIFO e PQ como método de escalonamento de filas. Os resultados acima são idênticos, como esperado, aos da Figura 25 (Servidor HTTP).

→ **Roteador 04:**

- Tráfego IP (*packets/sec*)

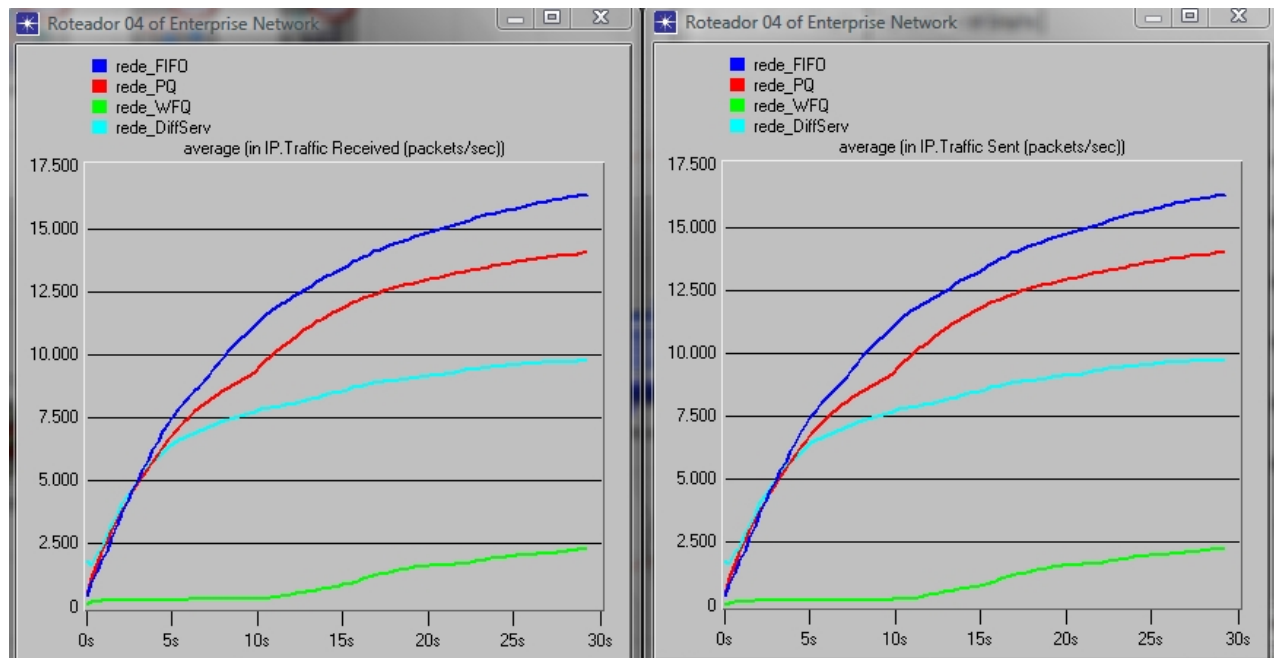


Figura 40 – Resultados obtidos no Roteador 04: Tráfego IP (*Received e Sent*)

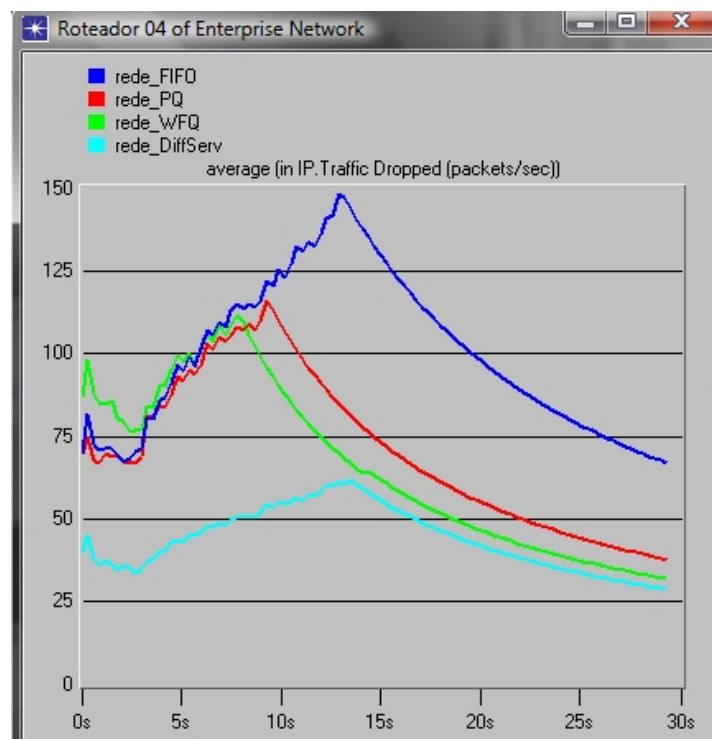


Figura 41 – Resultados obtidos no Roteador 04: Tráfego IP (*Dropped*)

O roteador 04 entrega às LAN's de 45 e 55 usuários serviços de VoIP, FTP (*Medium Load*) e HTTP (*Searching*). Como a rede FIFO envia e recebe mais pacotes, a probabilidade dela perder mais pacotes também é grande. Porém, a rede PQ possui quase o mesmo desempenho em relação a envio e recebimento de pacotes e suas perdas são bem menores.

## → Roteador 05:

- Tráfego IP (*packets/sec*)

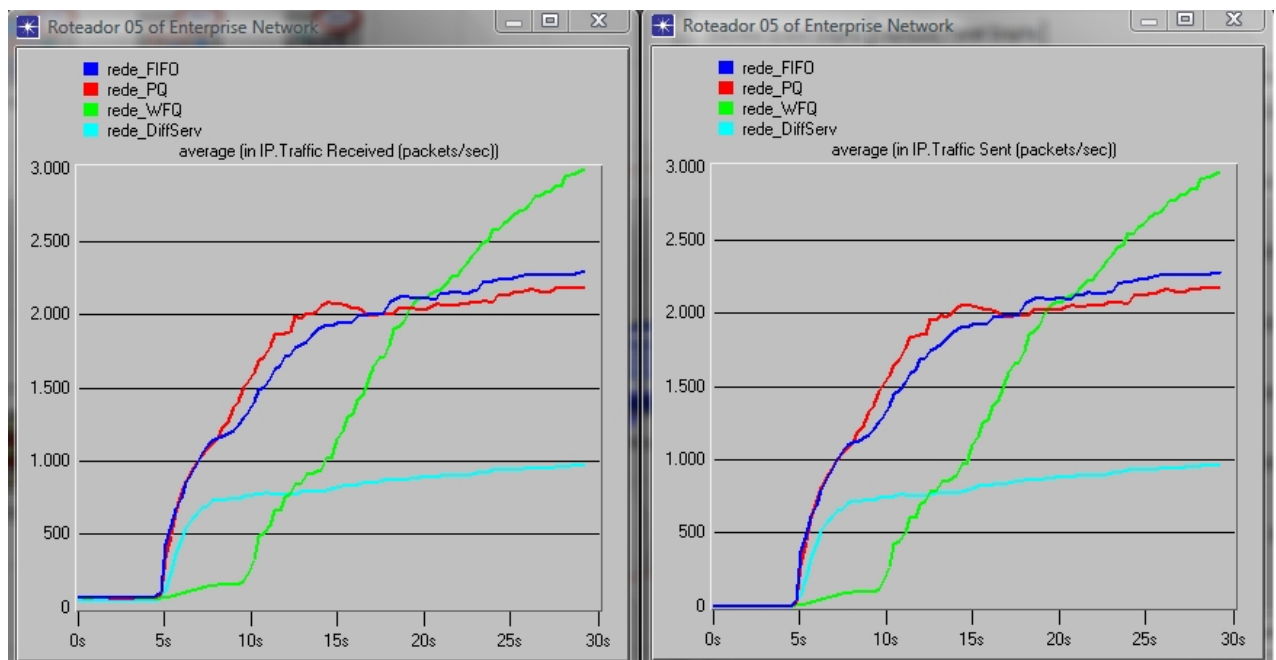


Figura 42 – Resultados obtidos no Roteador 05: Tráfego IP (*Received e Sent*)

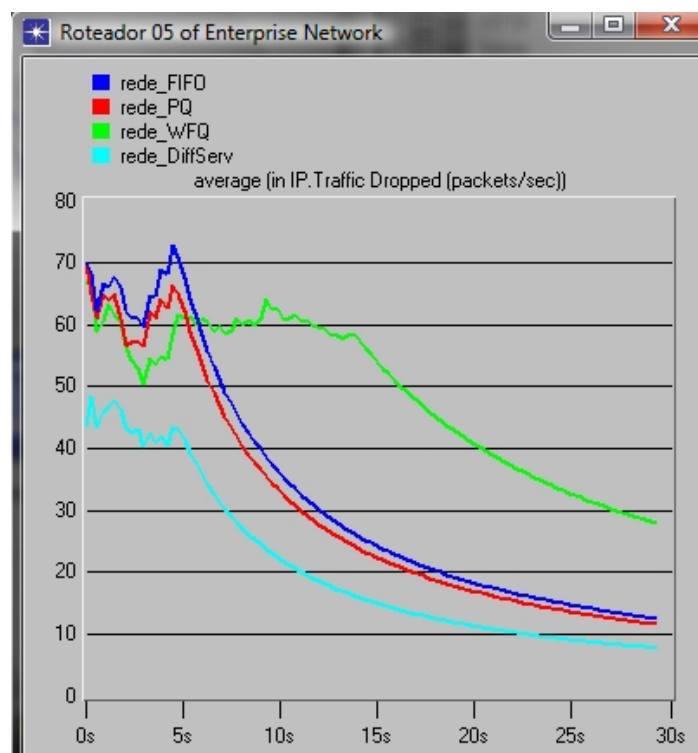


Figura 43 – Resultados obtidos no Roteador 05: Tráfego IP (*Dropped*)

O comportamento do roteador 05 é semelhante ao do roteador 01: o tráfego de pacotes enviados e recebidos começa a fluir após cinco segundos. Antes disto, existem perdas de pacotes, mas depois as perdas caem exponencialmente, estabilizando-se.



## 6. Conclusão

### • Sobre o projeto

Tendo o projeto sido criado apenas com o objetivo de verificar o comportamento de uma rede de computadores utilizando ou não qualidade de serviços nas aplicações suportadas, pudemos notar a necessidade de adicionar outras configurações nos elementos da rede.

Essas configurações foram realizadas, inicialmente, utilizando uma estrutura de priorização de pacotes nas aplicações chamada *Diffserv* (Serviços Diferenciados), classificando os pacotes como EF (*Expedited Forwarding*), ou seja, priorizando os pacotes. Posteriormente, fez-se necessário configurar os elementos de QoS: como as filas nos roteadores iriam se comportar. Esse comportamento é dado através de métodos de escalonamento dos sistemas de filas: FIFO (*First-In-First-Out*), PQ (*Priority Queuing*) e WFQ (*Weighted Fair Queuing*).

Entretanto, gostaríamos, inicialmente, comparar o comportamento de cada rede utilizando, em sua totalidade, somente um desses métodos de escalonamento de filas. Sendo assim, criamos quatro redes distintas a fim de compará-las.

Após simularmos todas as redes e feitos os devidos comentários, podemos notar que cada roteador tem um comportamento próprio, tendo seu desempenho muito dependente do que está acontecendo ao seu redor. As figuras dos gráficos obtidos mostram claramente esse aspecto. Os roteadores de borda têm comportamentos muitos parecidos entre si, sendo suas diferenças explicadas pelos perfis de usuários que estão conectados a eles.

### • Sobre o software

O *software* utilizado é excelente para simularmos qualquer tipo de redes de computadores. Podemos criar projetos utilizando fibras ópticas, cabos coaxiais e ethernet de todos os tipos. Uma grande quantidade de roteadores, *hubs*, *bridges* e *switches* com diversos tipos de configurações de portas e capacidade de transmissão.

Podemos utilizar todos os tipos de topologia de rede e configurá-las rápido e facilmente. Há opções de criar redes utilizando somente equipamento ATM, *Frame Relay* e etc. Sem contar os equipamentos de fornecedores (CISCO, Juniper, 3COM, NEC, Alcatel, Lucent, Nortel, HP, Foundry, Marconi, entre outros).

Porém, como comentado nos resultados obtidos, nem tudo ocorreu perfeitamente. A versão do *software* utilizada foi uma versão acadêmica, baixada diretamente do site do fabricante (*OPNET IT Guru Academic Edition*). Esse tipo de versão não nos oferece alguns recursos necessários, como por exemplo, o que mais prejudicou em algumas estatísticas: o tempo de duração da simulação.

Em função das limitações desta versão acadêmica, não foi possível criar a rede de melhor desempenho. A minha intenção era, após coletar e analisar os gráficos e as estatísticas, montar uma rede configurando os roteadores de acordo com as comparações realizadas anteriormente.

Depois de testar diversos tempos, verifiquei que o tempo máximo de duração das simulações é de 30 segundos. Portanto, não foi possível sobrecarregar a rede com muitos pacotes gerados pelas LAN's e, conseqüentemente, verificar mais claramente os possíveis congestionamentos e perdas de pacotes nos roteadores.

Até tentei, em vão, solicitar ao fabricante uma versão completa deste *software* por um período de experiência. Isto não foi possível, pois deveria pagar uma licença de alguns milhares de dólares.

Com a versão acadêmica já foi possível verificar muitos aspectos relevantes com relação à rede e à utilização de qualidade de serviço.

## 7. Referências Bibliográficas

TANENBAUM, Andrew S., *Redes de Computadores*. 4ª Edição. Rio de Janeiro: Elsevier, 2003. 945p.

ALECRIM, Paulo Dias de, *Simulação Computacional para Redes de Computadores*. Rio de Janeiro: Ciência Moderna Ltda., 2009. 253p.

IETF, "Multiprotocol Label Switching Architecture", RFC-3031, 2001. Acesso em: 24 de nov. 2011.

OPNET Technologies, Inc., "OPNET Modeler Brochure",  
<http://www.opnet.com/products/brochures/Modeler.pdf>. Acesso em: 26 de nov. 2011.

INSTITUTO SUPERIOR TÉCNICO, UNIVERSIDADE TÉCNICA DE LISBOA. Gestão de Redes e Sistemas Distribuídos. Disponível em: <https://dspace.ist.utl.pt/bitstream/2295/49532/1/grsd1.pdf>. Data de acesso: 26 de nov. 2011.

MARTIN SEEFELDER DE ASSIS, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. MPLS. Disponível em: [http://www.gta.ufrj.br/grad/01\\_2/mpls/](http://www.gta.ufrj.br/grad/01_2/mpls/). Data de acesso: 24 de nov. 2011.