PLATAFORMA DE CONFIGURAÇÃO E MONITORIZAÇÃO DE QOS NUMA REDE DIFFSERV

José Pedro Morais
Departamento de Optoelectrónica, INETI, Lisboa, Portugal
jmorais@dop.ineti.pt

Paulo Carvalho, Solange Lima
Departamento de Informática, Universidade do Minho,Gualtar,Braga, Portugal
{pmmc,solange}@di.uminho.pt

Palavras chave: QoS, Serviços Diferenciados, Monitorização, Métricas de QoS

Resumo: À medida que surgem novas aplicações cada vez mais complexas e exigentes, torna-se necessário dar um tratamento diferenciado ao tráfego que circula na Internet prestando uma qualidade de serviço (QoS) de rede apropriada. Neste artigo, inserido na área de monitorização de QoS em redes IP, propõe-se uma plataforma de configuração e monitorização de QoS integrada na ferramenta de simulação *Network Simulator* (NS) que permite parametrizar e testar vários mecanismos presentes numa rede de Serviços Diferenciados (*DiffServ*). Os resultados das principais métricas de QoS, fornecidas pela aplicação sob a forma de tabelas, gráficos e de uma visualização on-line, tornam possível reajustar e corrigir os valores de parametrização dos diversos mecanismos e melhorar o desempenho global da rede DiffServ numa perspectiva fim-a-fim.

1. INTRODUÇÃO

O constante aumento e diversidade de tráfego na Internet tem vindo a obrigar a um aumento dos largura de banda para se poder dar resposta a aplicações cada vez mais exigentes em termos de qualidade de serviço (QoS). Este aumento não consegue solucionar eficazmente o problema já que vai abrir caminho para o aparecimento de novas aplicações ainda mais pesadas e exigentes acabando por ocupar a largura de banda entretanto disponibilizada. Por forma a ultrapassar este problema, o *Internet Engineering Task Force* propôs as arquitecturas de Serviços Integrados *(IntServ)* [1] e Serviços Diferenciados *(DiffServ)* [2] por forma a permitir que tráfego de aplicações distintas possa ser enviado com a QoS necessária.

No modelo *DiffServ,* após agregação dos fluxos de tráfego em classes de serviço, é dado um tratamento diferenciado ao tráfego de rede de acordo com a sua sensibilidade a parâmetros, tais como, largura de banda, atraso, *jitter* e perda de pacotes. Através de classificação e marcação do tráfego, segundo determinados critérios, por exemplo, classes mais sensíveis a atrasos passam a ter um tratamento prioritário. Para que os objectivos de QoS sejam satisfeitos entre a fonte e o destino é necessário efectuar previamente a correcta parametrização dos componentes de uma rede DiffServ, tarefa essa nem sempre linear.

Nesse sentido, este trabalho propõe a construção de uma aplicação na qual é possível parametrizar e testar, de uma forma simples e integrada, os componentes de uma rede DiffServ numa perspectiva fim-a-fim. Os resultados obtidos para as métricas de QoS mais relevantes, tais como o atraso registado na entrega de pacotes, o tamanho máximo registado nas filas de espera, o número de pacotes perdidos, o *Instantaneous Packet Delay Variation (IPDV),* entre outros, são apresentados em tabelas e gráficos, sendo estas métricas obtidas na perspectiva do fluxo e da classe. Paralelamente é também possível apresentar um conjunto de resultados on-line reflectindo o estado da rede num determinado instante e ilustrando a sua evolução no tempo. Através dos resultados obtidos é possível reajustar e corrigir os valores de parametrização dos diversos mecanismos por forma a conseguir uma melhoria do desempenho global de rede, e verificar se um elemento de rede satisfaz os requisitos do tráfego e do serviço oferecido.

Para além da presente secção, de caracter introdutório, o artigo é composto por mais cinco secções. Na secção 2 são apresentados os principais componentes que definem uma rede *DiffServ*. Na secção 3 é feita uma abordagem sobre a monitorização de QoS sendo dada uma breve descrição das métricas de QoS utilizadas. Na secção 4 é feita a apresentação da aplicação desenvolvida que inclui a topologia da rede criada, as respectivas fontes de tráfego e os mecanismos suportados. Por fim, na secção 5, apresentam-se os resultados obtidos e, na secção 6, as conclusões e propostas para trabalho futuro.

2. COMPONENTES DE UMA REDE DIFFSERV

Esta secção inclui as definições necessárias para facilitar a compreensão dos diversos parâmetros e mecanismos de QoS normalmente associados a redes com serviços diferenciados, e que compõem a aplicação de monitorização de QoS desenvolvida.

2.1 Redes com serviços diferenciados

A arquitectura de serviços diferenciados surgiu para permitir fornecer de uma forma escalável QoS na Internet. A arquitectura prevê a divisão dos fluxos de dados em classes de serviço distintas recorrendo à etiquetagem de um código diferenciador em cada pacote. Cada nó da rede irá examinar esse código associado para determinar o tratamento a dar ao tráfego, ou seja, estabelecer o *Per-Hop Behavior (PHB)* adequado. A etiquetagem consiste na marcação de um campo no cabeçalho IP, com a designação de *Type Of Service (ToS)* no caso do IPv4 e *Differentiated Services Code Point (DSCP)* no caso do IPv6 [3], com um valor numérico que os routers usam para tomarem um determinado comportamento face ao pacote a enviar. Quando um pacote chega a um nó da rede é feita a sua diferenciação com base na marca, ou seja, o pacote será analisado e classificado sendo encaminhado para uma fila de espera e escalonado, de forma a cumprir os requisitos de QoS da classe a que pertence, ou então é descartado. Através desta técnica os routers deixam de ter filas únicas do tipo FIFO passando a ter várias que guardam os pacotes consoante a sua marcação. Um router sem capacidades DiffServ dará o encaminhamento por defeito ao pacote mantendo a marcação prévia tal como se encontrava. Uma grande vantagem do modelo DiffServ é o facto de ser escalável não sendo necessário manter informação de estado por fluxo. Relativamente às tarefas realizadas pelos diversos nós, os routers fronteira efectuam a classificação, condicionamento de tráfego, gestão de filas e escalonamento, enquanto que os routers que compõe o núcleo da rede não efectuam condicionamento de tráfego, tratando apenas cada pacote de acordo com o PHB [4][5] correspondente. A nível de um domínio existem propostas de diversos comportamentos a prestar *edge-to-edge*, denominados *Per-Domain-Behavior* (PDB)[6] .

2.2 Classificação de tráfego

Nos serviços diferenciados os nós fronteira são responsáveis por fazer a classificação e condicionamento de tráfego por forma a assegurar que este está de acordo com o *Service Level Agreement (SLA).* A arquitectura DiffServ prevê a existência dos módulos a seguir descritos.

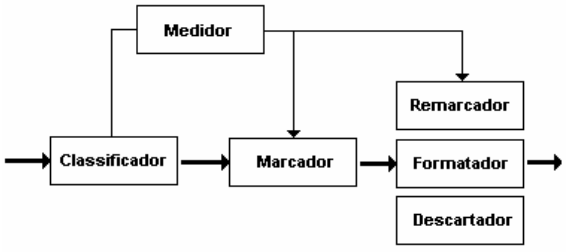

Figura 1 - Classificação e condicionamento de tráfego

2.2.1 Classificação

Em geral, nos routers fronteira a classificação é realizada por classificadores *MultiField (MF)* enquanto que no interior do domínio DiffServ é feita por classificadores de comportamento agregado – *Behavior Aggregate (BA).* As políticas de classificação podem especificar um conjunto de regras e os valores DSCP correspondentes para marcar os pacotes.

2.2.2 Condicionamento de Tráfego

O condicionamento de tráfego inclui funções de policiamento de tráfego para fazer cumprir o *Traffic Conditioning Agreement (TCA)* estabelecido entre os clientes e os fornecedores de serviços. O condicionamento de tráfego pode incluir diversas componentes tais como medidores, marcadores,

formatadores e descartadores. Segue-se uma breve descrição de cada um deles:

- **medidor**

o medidor (*meter*) mede as propriedades temporais de uma *stream* de tráfego seleccionada pelo classificador. É feita a distinção do tráfego que se encontra dentro (*in-profile*) e fora (*out-of-profile*) de um determinado perfil acordado, podendo estes valores vir a afectar o marcador, o descartador e o formatador;

- **marcador**

o marcador (*marker*) preenche ou altera o campo DS dos pacotes processados. A remarcação pode ser também necessária na fronteira de dois domínios administrativos que usem diferentes DSCPs;

- **formatador**

o formatador (*shaper*) atrasa os pacotes não conformes por forma a trazer a *stream* de tráfego à concordância com o perfil de tráfego acordado;

- **descartador**

o descartador (*dropper*) efectua o descarte de pacotes com base no estado do medidor e de regras pré-definidas, por exemplo o descarte de pacotes *out-of-profile*.

2.3 Per-Hop Behaviours

Os PHBs descrevem o comportamento de encaminhamento num router DiffServ a aplicar ao tráfego agregado com o mesmo DSCP, determinando a estratégia a seguir de alocação de recursos para a construção dos serviços diferenciados. A implementação dos PHBs é feita por mecanismos de gestão de filas de espera e de escalonadores apropriados. A Tabela 1 apresenta os códigos normalizados para os PHBs implementados.

PHB	Binário			Decimal
EF	101110			46
AF1x	001010	001100	001110	10,12,14
AF2x	010010	010100	010110	18,20,23
AF3x	011010	011100	011110	26,28,30
AF4x	100010	100100	100110	34,36,38
BE	000000			0

Tabela 1 – PHBs normalizados

Os PHBs EF [5] e AF [4] são os PHBs normalizados vocacionados para tráfego com diferentes tipos de exigências.

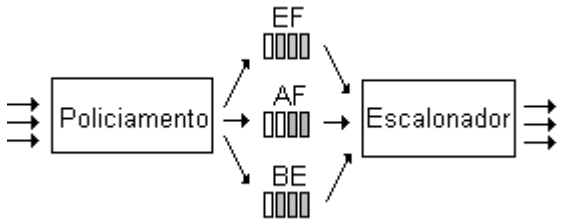

Figura 2 – Tratamento dos pacotes após marcação

- ***Expedited Forwarding***

Uma classificação tipo EF será adequada para aplicações que exijam um serviço com um atraso e *jitter* reduzidos e com uma largura de banda garantida podendo desta forma simular uma conexão ponto-a-ponto ou uma linha dedicada fim-a-fim. Neste tipo de encaminhamento a taxa de saída de um router DiffServ deve ser maior do que a taxa de entrada. Apesar das suas vantagens, os serviços baseados neste PHB poderão ter um elevado custo e um controlo muito estrito, com rejeição de tráfego *out-of-profile*.

- ***Assured Forwarding***

No caso do AF estão previstas quatro classes de serviço cada uma delas com três níveis de precedência para descarte de pacotes no caso de existir congestionamento. Cada classe de serviço pode permitir ainda que a sua largura de banda reservada possa ser utilizada no caso de não estar a ser utilizada. O AF apesar de não apresentar garantias de limite no atraso e no *jitter* permite diferentes níveis de garantia na expedição dos pacotes havendo, uma maior probabilidade que esta se realize no caso de se tratar de tráfego *in-profile* e menor probabilidade no caso de tráfego *out-of-profile*.

3. MONITORIZAÇÃO DE QOS

O processo de monitorização de redes com QoS é muito importante para que se faça a verificação de possíveis quebras de desempenho na transmissão de pacotes com efeitos imediatos de degradação sobre o contrato de QoS celebrado. Com vista a monitorizar este tipo de comportamento, e recorrendo à plataforma de simulação NS, foi desenvolvida uma aplicação cuja interface permite configurar os diversos componentes DiffServ e analisar os resultados obtidos. No estudo efectuado é realizada uma monitorização *off-line* e uma outra *on-line* pós-processamento. Segue-se uma breve descrição do ambiente de simulação utilizado, dos

diversos componentes passíveis de configurar na aplicação desenvolvida e das diversas métricas de QoS que integram os resultados.

3.1 Ambiente de simulação

- **Elementos do NS**

O NS é um simulador de redes orientado a eventos [7]. O simulador contempla a definição e uso de diferentes protocolos, tráfego *Constant Bit Rate (CBR)* ou *Variable Bit Rate (VBR),* mecanismos de gestão de filas de espera, escalonadores entre outras funcionalidades. O NS é um interpretador de *script* Tcl, orientado a objectos, contendo ainda uma biblioteca com objectos de escalonamento de eventos, componentes de rede e módulos de ajuda na configuração da rede. Um evento no NS tem um identificador por pacote que é único guardado simultaneamente com o instante de tempo. O escalonador de eventos, sabendo o tempo de escalonamento do objecto, coloca-o na fila de eventos que faz a chamada aos componentes de rede apropriados num tempo pré-definido.

- **Análise de Traces**

Os resultados da simulação são guardados em ficheiros designados por *Trace-Files.*

```
(...)
+ 2.118995 2 3 exp 300 - 0 2.0 8.0 1616 405
- 2.118995 2 3 exp 300 - 0 2.0 8.0 1616 405
- 2.119122 4 5 cbr 500 -46 0.0 6.0 1035 388
r 2.119219 2 3 exp 300 - 0 2.0 8.0 1606 404
+ 2.119219 3 4 exp 300 - 0 2.0 8.0 1606 404
d 2.119219 3 4 exp 300 - 0 2.0 8.0 1606 404
 (...)
```

Exemplo 1 - Ficheiro *trace-all* criado

O ficheiro *Trace-All* gerado guarda a informação referente a cada pacote num determinado instante, sendo o seu conteúdo (ver Exemplo 1) descrito na Tabela 2. Por defeito, o NS não permite analisar o campo DSCP necessário para efectuar monitorização por classe pelo que foi necessário proceder a essa alteração (ver secção 4).
Tal como se encontra ilustrado na Figura 4, o pacote poderá assumir vários comportamentos ao passar por um nó DiffServ:

1. Entra na fila de espera – *Enqueue* (+)
2. Sai da fila – *Dequeue* (-)
3. Prossegue para o nó seguinte – *Event receive* (r)
4. É feito o descarte – *Drop* (d)
5. Procede à remarcação

Campo	Símbolo	Descrição
1	+ - r d	Comportamento do pacote: + :: *enqueue* - :: *dequeue* r :: *event receive* d :: *drop*
2	n	Instante em segundos em que o pacote é analisado
3	0 ... 8	Local onde o pacote se encontra – origem
4	0 ... 8	Local onde o pacote se encontra – destino
5	cbr, pareto, exp	Tipo de tráfego
6	n	Tamanho do pacote
7	---	(não usados)
8	46, 10, 12, 14, 0	Código DSCP associado ao pacote
9	0 .. 2	Origem inicial do pacote
10	6 .. 8	Destino final do pacote
11	n	Sequência do pacote
12	n	Identificação única do pacote

Tabela 2 – Códigos associados ao ficheiro "*Trace-All*"

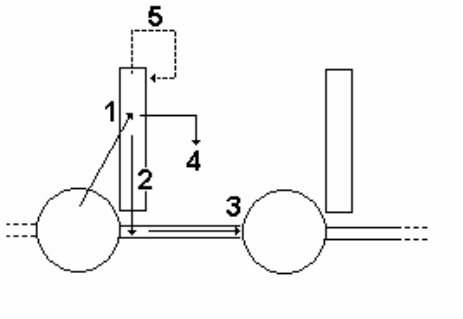

Figura 4 – Comportamento por nó de um pacote

Na análise de resultados realizada são pesquisados os campos 1, 2, 3, 4, 8 e 12 por forma a calcular as métricas de QoS por fluxo e por classe.

- **Visualização de resultados**

Como foi referido, após uma simulação ser realizada, o NS produz um ou mais ficheiros (*Trace-Files*) que contêm resultados detalhados da simulação. Estes valores são depois utilizados para análise ou como entrada para outro tipo de ferramentas como é o caso do *Network Animator (NAM)* [7] que possibilita a visualização animada da simulação gerada sendo possível observar o comportamento de cada pacote sobre cada nó da rede, bem como, verificar o comportamento das filas de espera configuradas.

Outro tipo de ferramentas, como o *AWK*, *Xgraph* [7] e *VisualTcl* [8], são também usadas para trabalhar os resultados gerados e mostrar os mesmos sob a forma de tabelas, gráficos e visualizações on-line pós-processamento.

3.2 Componentes configuráveis

3.2.1 Tráfego simulado

Dado se tratar de um modelo simulado é necessário injectar tráfego sintético nos nós fonte para que se possa proceder à monitorização do mesmo. Seguem-se as definições das respectivas fontes de tráfego que podem ser utilizadas na aplicação de monitorização desenvolvida.

- **Exponencial on/off**

O tráfego é gerado de acordo com uma distribuição exponencial *on/off*. Os pacotes são enviados a uma taxa fixa durante os períodos de tempo *on,* não sendo enviado tráfego durante os períodos *off*.

- **Pareto on/off**

Idêntico ao tráfego exponencial mas no qual os períodos o*n/off* em vez de terem distribuição exponencial seguem uma distribuição de Pareto.

- **CBR**

O tráfego *Constant Bit Rate (CBR)* é caracterizado por uma taxa de transmissão constante.

3.2.2 Policiamento

O policiamento define o conjunto de regras a aplicar nos nós fronteira ao tráfego que entra ou sai do domínio por forma a verificar a conformidade (*in* ou *out-of-profile*) com os parâmetros de serviço especificados. Todos os fluxos que tenham um determinado par origem-chegada são tratados como um fluxo agregado tendo cada um deles um tipo específico de policiamento, medição e DSCP associado. Dependendo do tipo de policiamento em uso e da sua parametrização, o tráfego *out-of-profile* pode ser atrasado, descartado ou remarcado com menor prioridade. Os parâmetros comuns utilizados para a configuração dos mecanismos de policiamento são: *Committed Information Rate (CIR)* e *Peak Information Rate (PIR)* em bits por segundo, *Committed Burst Size (CBS)*, *Excess Burst Size (EBS)* e *Peak Burst Size (PBS)* valores estes em *bytes*. Os mecanismos considerados na aplicação desenvolvida são os que a seguir se descrevem:

- **Token Bucket (TB)**

O algoritmo TB imprime um padrão de saída de acordo com a taxa de enchimento do balde e com o número de *tokens* que este contém, permitindo *bursts* de tráfego até um determinado limite [9].

- **TSW3CM**

O valor médio da precedência de descarte é usado probabilisticamente quando o CIR for excedido sendo a precedência de descarte mais baixa usada probabilisticamente quando o PIR é excedido [10].

- **srTCM**

A escolha das três precedências de descarte é feita a partir dos parâmetros CIR, CBS e EBS [11].

- **trTCM**

A escolha das três precedências de descarte é feita a partir dos parâmetros CIR, CBS, PIR e PBS [12].

A aplicação de monitorização desenvolvida permite configurar os parâmetros anteriormente referidos da seguinte forma :

	CIR	**CBS**	**EBS**	**PIR**	**PBS**
TB	▪	▪			
TSW3CM	▪			▪	
srTCM	▪	▪	▪		
trTCM	▪	▪		▪	▪

Tabela 3 – Parâmetros de policiamento

3.2.3 Gestão de filas de espera

A gestão de filas de espera é necessária para facilitar a diferenciação de tráfego consoante o seu grau de prioridade e prevenir situações de congestão. A escolha de um algoritmo adequado e tamanho máximo da fila são factores relevantes. Um tamanho de fila elevado pode provocar atrasos demasiado elevados enquanto que um tamanho reduzido pode resultar no descarte excessivo de tráfego. Seguem-se as descrições das técnicas utilizadas na gestão das filas de espera que se encontram disponíveis na aplicação desenvolvida.

- **DROP**

O DROP é um mecanismo elementar que monitoriza constantemente a fila de espera de modo a que a partir de um determinado nível de ocupação os novos pacotes são descartados. Este é um mecanismo ainda mais simples do que o *Random Early Detection (RED)* [13] onde os pacotes são descartados de forma aleatória e independente do seu estado de conformidade quando o tamanho da fila de espera excede um valor limite pré-definido.

- **RIO-C**

O *Random Early Detection with In and Out (RIO)* [14] inclui um algoritmo RED distinto para tráfego *in-profile* e outro para tráfego *out-of-profile*. No RIO-C (RIO-*coupled*) a probabilidade de descarte de um pacote *out-of-profile* é baseada no peso médio do comprimento de todas as filas de espera virtuais, enquanto que a probabilidade de descarte de um pacote *in-profile* é baseada somente na média do tamanho da fila de espera correspondente.

- **WRED**

O *Weighted Random Early Detection* (WRED) [15] é uma variante do RED. O WRED combina as funcionalidades do RED com a classificação de pacotes IP fazendo com que os de baixa prioridade sejam descartados com maior probabilidade do que os de alta prioridade, sendo todas as probabilidades baseadas apenas no tamanho de uma fila de espera única.

3.2.4 Escalonamento

Os algoritmos de escalonamento determinam as disciplinas de serviço dos pacotes que estão nas filas de espera, por forma a contribuir para a QoS pretendida. Na aplicação implementada é possível efectuar a configuração dos seguintes escalonadores:

- **RR**

o escalonador *Round-Robin* (RR) [16] serve rotativamente um pacote de cada fila não vazia;

- **Priority Queuing**

nesta disciplina a cada fila é atribuído um nível de prioridade, sendo despachados os pacotes de prioridade mais baixa se não existirem pacotes com uma prioridade mais alta [16]. Desta forma, quanto maior for o nível de prioridade atribuído menor será o atraso médio sofrido pelos pacotes;

- **WRR**

o escalonador *Weighted Round-Robin* (WRR) [17] serve as filas proporcionalmente aos pesos atribuídos na configuração das mesmas, e de forma rotativa.

3.3 Métricas de QoS disponibilizadas

As métricas de QoS são usadas para traduzir o nível de serviço oferecido pela rede. Estudos recentes [18,19,20] mostram as principais métricas que são objecto de monitorização. No nosso trabalho as métricas de QoS, calculadas por fluxo e por classe, são:

- **atraso fim-a-fim**

o atraso fim-a-fim - *One-Way Delay* (OWD) [21] - mede o tempo consumido pela rede para transportar um pacote desde a fonte até ao seu destino. Este tipo de métrica é de grande importância dado o elevado número de aplicações sensíveis a tempos;

- **jitter**

o jitter indica a variação no atraso relativamente ao atraso médio (ver secção 5.2).

- **variação instantânea no atraso**

a variação instantânea no atraso - *Instantaneous Packet Delay Variation (IPDV)* – segundo definição do IPPM [22], é obtida com base nos valores do OWD para pares de pacotes consecutivos. Tomando por Di o atraso verificado no pacote com índice i, o IPDV é definido como $| D_i - D_{i-1} |$;

- **throughput**

o débito fim-a-fim ou *throughput* indica o volume de tráfego transmitido por unidade de tempo de um nó da rede de origem até ao nó destino;

- **perda de pacotes**

a perda de pacotes é obtida pelo rácio de pacotes perdidos face aos pacotes enviados, e normalmente é expressa em percentagem. A perda de pacotes dá-se essencialmente por descarte de pacotes nas filas de espera quando estas ficam cheias ou por acção de um mecanismo de gestão de filas.

4. PLATAFORMA DE CONFIGURAÇÃO E MONITORIZAÇÃO

Para monitorizar e avaliar as métricas de QoS foi criado um modelo de rede e um interface que permite configurar e parametrizar os diversos componentes de rede, bem como, avaliar o desempenho obtido através de um conjunto de métricas.

4.1 Modelo de rede

O modelo de rede apresentado na Figura 5 é composto por nove nós sendo três de partida, três de chegada e três intermédios. Apesar de se tratar de um modelo simples permite testar e afinar o comportamento de uma rede com QoS. Torna-se

assim possível parametrizar a rede de acordo com os níveis de QoS a oferecer e verificar se o seu comportamento está de acordo com o pretendido, bem como, concluir sobre essa parametrização.
Existem três fontes de tráfego designadas por S1, S2 e S3 as quais enviam dados para D1, D2 e D3 respectivamente, interligados por dois routers fronteira e um router interior (core). Tanto a largura de banda como o tempo de propagação entre cada um dos nós são parametrizáveis. Todo o modelo foi construído em código TCL sobre a plataforma de simulação NS2. À semelhança da aplicação NAM que se baseia numa demonstração *on-line* assente num pós-processamento de um ficheiro de resultados previamente criado, neste estudo foi também incluída essa opção.

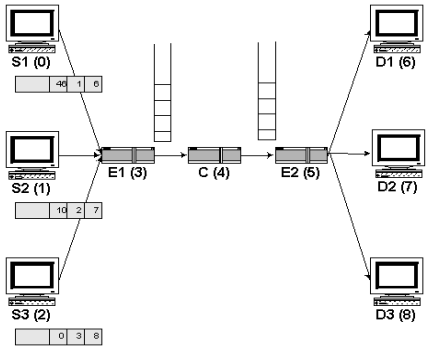

Figura 5 – Diagrama do modelo de rede implementado

Pretendeu-se criar um cenário onde fosse possível analisar o comportamento das classes EF, AF e BE em simultâneo e qual o comportamento dos routers fronteira ao processarem este tipo de pacotes. A Tabela 4 ilustra a forma como o tráfego é classificado por forma a cumprir este objectivo.

Origem	Destino	DSCP	Classe
S1	D1	46	EF
S2	D2	10	AF
S3	D3	0	BE

Tabela 4 – Classificação de tráfego

Apesar do NS permitir realizar análise ao nível de fluxo não permite obter directamente uma análise por classe. Assim foi necessário alterar o código fonte (trace.cc) para passar a incluir o DSCP associado a cada pacote enviado:

```
sprintf(wrk_, "%c %g %d %d %s %d %s %d %d.%d
               %d.%d %d %d",
        tt, Scheduler::instance().clock(),
        s, d, name, th-\>size(), flags,
        iph-\>prio()
```

Exemplo 2 – Código fonte do NS alterado

Posteriormente e recorrendo à pesquisa dos ficheiros gerados usando filtros apropriados (AWK) consegue-se extrair a informação necessária para analisar o comportamento por fluxo e por classe na rede implementada (ver Figura 6).

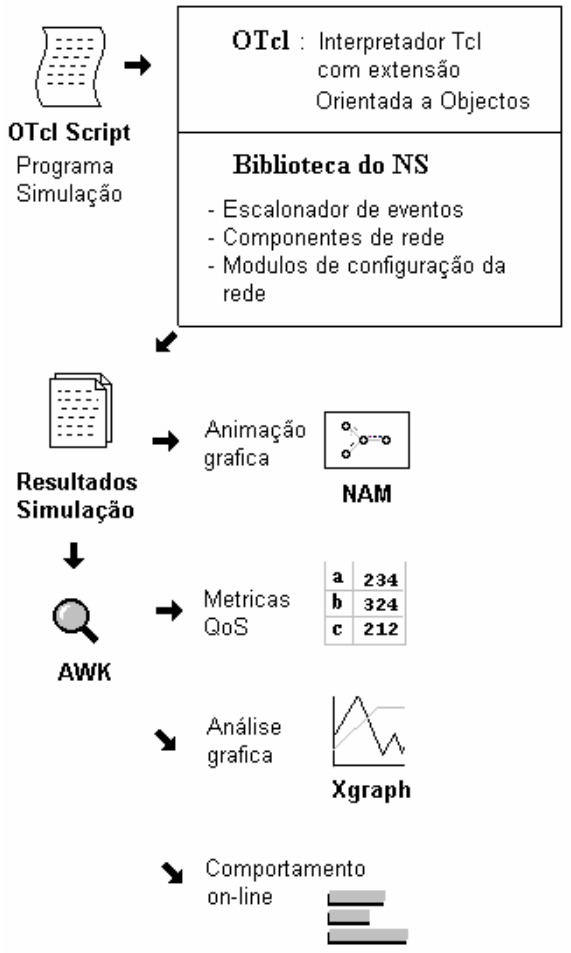

Figura 6 – Análise dos ficheiros *trace* gerados

4.2 Parametrização do modelo

Em seguida são explicados os vários quadros que compõem a aplicação desenvolvida com as respectivas parametrizações ao nível dos elementos de rede e mecanismos de QoS associados. A Figura 7 mostra o primeiro quadro que permite introduzir os parâmetros relativos ao protocolo de transporte, tamanho pacote e tipos de tráfego. Por fluxo é definido o protocolo a utilizar e o tráfego gerado, que poderá ser CBR, *Pareto* ou *Exponencial*. Segue-se a taxa de transmissão, o tamanho de cada pacote e três outras opções – *burst*, *idle* e *shape* – usadas nas distribuições de *Pareto* e *Exponencial*. É também

definido o tempo de geração de tráfego e a duração da simulação havendo uma opção para prolongar esta última em 0.5 segundos após ter terminado a geração de tráfego.

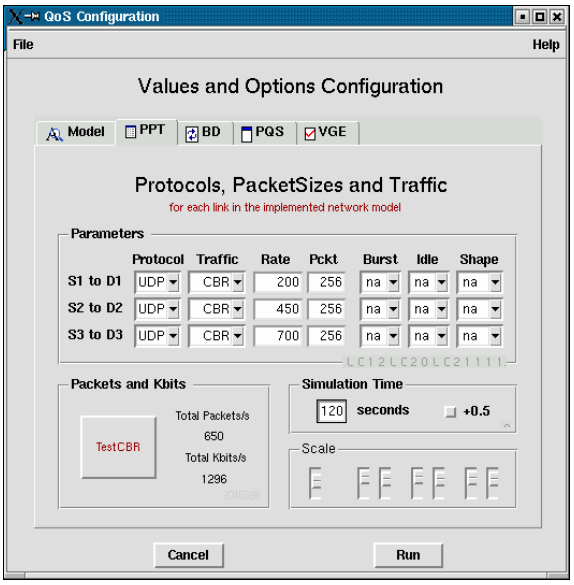

Figura 7 – Quadro da aplicação em "*Protocols, PacketSizes and Traffic"*

Segue-se outro quadro de configuração que permite parametrizar a largura de banda entre cada nó, bem como, o atraso entre os mesmos (em milisegundos).

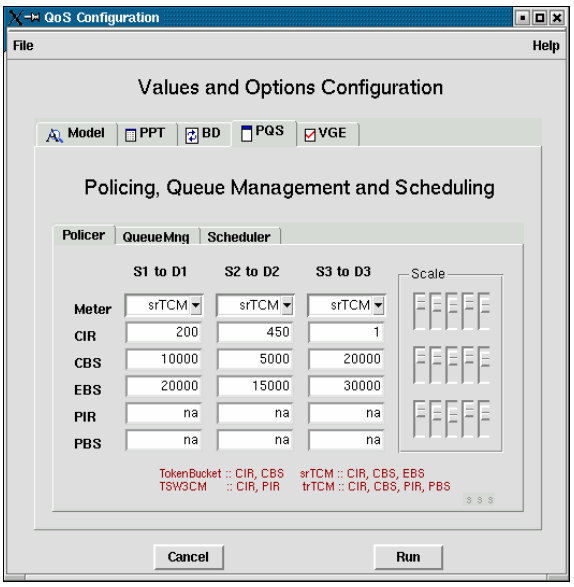

Figura 8 – Quadro da aplicação em "*Policing, QueueManagement and Scheduling*" -> "*Policer*"

Posteriormente surgem as opções de configuração do policiamento, fila de espera e escalonador tal como se encontra ilustrado na Figura 8. Para a primeira opção de parametrização pode ser escolhido o *Token Bucket*, TSW3CM, srTCM ou trTCM. Consoante o policiador escolhido é necessário introduzir os respectivos valores para parâmetros tais como CIR, CBS, EBS entre outros já descritos anteriormente.
No subquadro seguinte - "*Queue Management*" define-se o mecanismo de descarte a utilizar para as classes AF - DROP, RIO-C e WRED - e os parâmetros *minimum* e *maximum Threshold e Drop Probability.*
Finalmente, é permitida a configuração e parametrização do escalonador – RR, PQ, WRR – a aplicar às classes EF, AF e BE.
Ainda na parametrização da simulação existe o quadro "*Visualization and Graph Options"* onde se define através de três opções o que se pretende gerar e visualizar na simulação:

- **NAM**

a simulação NAM pode ou não ser criada no final da simulação tendo ainda por opção a apresentação de valores *on-line* dentro da mesma;

- **Graphs**

a visualização dos resultados poderá ser mostrada na janela de terminal de forma contínua ou ainda ser definido se se pretende uma análise *on-line*, pós-processamento;

- **Visualization**

relativamente à análise gráfica pode ser definida a opção de não gerar quaisquer tipo de gráficos ou de criar uma análise instantânea baseando-se em funções existentes no próprio NS, sendo por isso de rápido processamento mas com resultado limitados. É ainda possível efectuar uma análise estendida. Esta última opção faz o uso de uma análise aprofundada de vários ficheiros *trace* gerados, recorrendo ao AWK.

5. CÁLCULO E VISUALIZAÇÃO DE RESULTADOS

Nesta secção é descrito o tipo de resultados que podem ser alcançados através da aplicação de monitorização. Apesar do objectivo deste artigo não ser a análise de resultados obtidos, é avançado um exemplo como forma de demonstrar as capacidades oferecidas. Toma-se por base o modelo de rede definido na Figura 5, na presença de três classes de

tráfego e um período de simulação de 120 segundos. Entre os nós S1 e D1 é simulado uma aplicação cujo tráfego tem mais prioridade e maiores exigências de QoS do que o restante, pelo que é classificado como EF. Entre S2 e D2, o tráfego enviado é classificado como AF e entre S3 e D3 é gerado tráfego BE.
A capacidade dos *links* de acesso é de 10Mbps e o tempo de propagação de 5ms. No interior da rede foi definida uma largura de banda de 1Mbps e 1,5Mbps, com um atraso de 20ms entre os nós E1-C e C-E2, respectivamente. O perfil de tráfego gerado, tipo de policiamento e gestão de filas são mostrados na Tabela 5.

	EF	**AF**	**BE**
	S1 – D1	**S2 - D2**	**S3 - D3**
Protocolo	UDP	UDP	UDP
Tráfego	CBR / Pareto	CBR / Pareto	CBR / Pareto
T. Pacote (bits)	256	256	256
Débito CBR (Kbps)	200	450	500
Débito Pareto (no período on) (Kbps)	400	900	1000
Policiamento	srTCM	srTCM	srTCM
CIR (Kbps)	200	450	1
CBS (bytes)	10000	5000	20000
EBS (bytes)	20000	15000	30000
Gestão Fila E.	RIO-C	RIO-C	RIO-C
WRR	10	5	1

Tabela 5 – Parametrização das três classes existentes

Os fluxos de tráfego foram definidos como CBR por forma a avaliar mais facilmente a relação entre a largura de banda disponibilizada e a largura de banda utilizada. É também abordada uma simulação equivalente com tráfego Pareto, alterando o CIR do EF para 400 e AF para 900, mantendo os restantes parâmetros e usando os valores de *burst*, *idle* e *shape* de 1s, 1s e 1.5 respectivamente. A gestão de cada fila de espera é efectuada pelo RIO-C e o escalonamento por WRR com os pesos de 10, 5 e 1 para as classes EF, AF e BE respectivamente.
Após a definição destes parâmetros e executada a simulação são mostrados no interface de resultados os vários subquadros com os valores das principais métricas da QoS obtidas tanto ao nível de fluxo como ao nível de classe:

- PT - Packet Loss;
- OWD - One Way Delay;
- IPDV – Inst. Packet Delay Variation;

e ainda uma indicação do

- MQL - Maximum Queue Length.

Simultaneamente é possível visualizar os resultados anteriores sob a forma gráfica ou através de uma animação *on-line* pós-processamento:

- GG - Graphics Generation;
- OL - On-line Results.

Segue-se uma explicação de cada um destes resultados acompanhado com os valores obtidos respeitantes ao exemplo anteriormente definido.

5.1 Perda de pacotes

O subquadro "*Packet Loss*" inclui os resultados finais por fluxo e por classe para o tempo de simulação anteriormente estipulado. No primeiro caso são analisados os fluxos entre cada nó fonte e destino, bem como, entre os dois nós fronteira. Os valores obtidos são:

- rácio de pacotes perdidos;
- nº de pacotes enviados;
- nº de pacotes recebidos;
- nº de pacotes perdidos.

Quanto à análise por classe são conseguidos os mesmos valores que os anteriores mas desta vez para as classes EF, AF e BE entre os nós fronteira.
Como a Figura 9 indica, o tráfego classificado como EF e AF é totalmente encaminhado tendo por isso valores iguais para o número de pacotes *In* e *Out*. Para o PHB BE, verifica-se uma taxa de perda de 29.8 %.

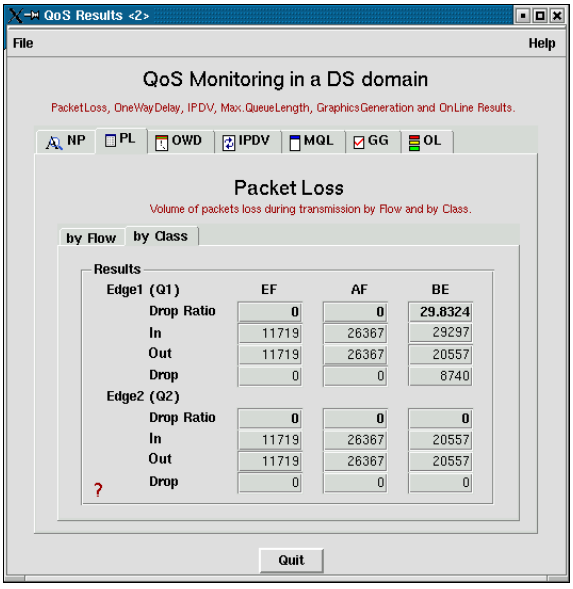

Figura 9 – Quadro da aplicação em *Results* - "*Packet Loss*" -> "*by Class*"

A Tabela 7 compara estes resultados com os correspondentes obtidos com tráfego Pareto.

	Taxa de Perda	
	Tráfego CBR	**Tráfego Pareto**
S1-D1	0.0 %	0.0 %
S2-D2	0.0 %	27.0 %
S3-D3	29.8 %	64.8 %

Tabela 7 – Comparação de perda de entrega de pacotes para tráfego CBR e Pareto

A aplicação possibilita também uma visualização da informação relativa a um determinado fluxo, sendo desta forma possível analisar o comportamento que um determinado pacote teve na rede. A obtenção destes resultados é feita através da análise do ficheiro "*trace*" ao longo do tempo de simulação, recorrendo ao AWK.

5.2 Atraso fim-a-fim

Segue-se a obtenção de métricas relacionadas com o "*One Way Delay"* sempre numa perspectiva de fluxo e de classe, obtendo-se os valores para o *Average Delay* e *Jitter*.
O Exemplo 3 ilustra o código utilizado para o cálculo do atraso fim-a-fim entre S1 e D1. Após registados os tempos em *ms* de saída e de chegada dos respectivos pacotes iniciados com o código "+" e terminados com o código "r", é calculado o atraso dos pacotes registado entre os dois nós.

```
if ( $3 == "0" && $1 == "+" ){
     tpartida[$12] = $2 ;}
if ( $4 == "6" && $1 == "r" ) {
     tchegada[$12] = $2 ;}
  (…)

for ( pktId=0; pktId <= lastPkt; pktId++ ) {
  if (chegada[pktId] != "" &&
      partida[pktId] != "" ) {
          atrasoActual =
          tchegada[pktId] – tpartida[pktId];
      (…)
     }
```

Exemplo 3 – Código para análise de fluxo e cálculo do atraso registado por pacote

Neste tipo de análise os pontos de medição podem estar localizados na fonte e destino, bem como, nos nós fronteira. O atraso médio e jitter são dados por:

$$\text{atrasoMedio} = \frac{\sum_{\text{pktId}=0}^{\text{lastReceivedPkt}} \text{tchegada}[\text{pktId}] - \text{tpartida}[\text{pktId}]}{\text{nPkts Recebidos}}$$

$$\text{jitter} = \left(\frac{\sum_{\text{pktId}=0}^{\text{lastReceivedPkt}} (\text{atraso} - \text{atrasoMedio})^2}{\text{nPkts Recebidos}} \right)^{1/2}$$

No caso de se tratar da análise feita por classe e não por fluxo, a filtragem é feita com base no valor do DSCP contido no ficheiro *trace*. O Exemplo 4 apresenta o tipo de filtragem utilizado para o tráfego classificado como EF, AF e BE relativamente à fila de espera existente no primeiro nó fronteira.

```
If ( $8 == "46" &&
      $3 == "3" && $4 == "4" && $1== "+" )
     (…)
If ( $8 == "10"||$8 == "12"||$8 == "14" &&
     (…)
If ( $8 == "0" &&
     (…)
```

Exemplo 4 – Código AWK para análise por classe

A Tabela 8 apresenta uma comparação de resultados entre o atraso médio fim-a-fim para os dois cenários de simulação anteriormente definidos.

	Atraso Médio	
	Teste CBR	**Teste Pareto**
S1-D1	55.3 ms	54.9 ms
S2-D2	55.6 ms	194.8 ms
S3-D3	342.3 ms	424.3 ms

Tabela 8 – Comparação entre o atraso médio fim-a-fim

5.3 IPDV

O Instantaneous Packet Delay Variation resulta no módulo da diferença entre o *atrasoActual* com o *atrasoAnterior.* São obtidos os seus valores médio e máximo tanto a nível do fluxo como da classe.

	IPDV Médio	
	Teste CBR	**Teste Pareto**
S1-D1	2 ms	1ms
S2-D2	1.4 ms	3ms
S3-D3	1.9 ms	10ms

Tabela 9 –IPDVs médios obtidos

	IPDV Máximo	
	Teste CBR	**Teste Pareto**
S1-D1	6.1 ms	11 ms
S2-D2	2.5 ms	206 ms
S3-D3	8.1 ms	1023 ms

Tabela 10 –IPDVs máximos obtidos

Ainda para o mesmos cenários de teste apresentam-se nas Tabelas 9 e 10 os valores obtidos por fluxo para o cálculo do IPDV médio e IPDV máximo.

5.4 Tamanho máximo das filas de espera

A aplicação permite ainda verificar o estado das filas de espera nos nós fronteira relativamente ao tamanho máximo registado em:

- bytes
- número de pacotes
- *throughput*

Este tipo de verificação, apesar de não ser considerada uma métrica de QoS, torna-se extremamente útil para determinar um dimensionamento adequado das filas de espera em cada nó.

5.5 Visualização Gráfica

Para além da obtenção dos resultados numéricos de métricas de QoS é ainda possível efectuar a análise gráfica da evolução que as mesmas tiveram durante o tempo de simulação. Através do comando *Xgraph* aplicado sobre os *traces* gerados obtém-se os gráficos reflectindo a evolução das perdas, atraso e *throughput* por fluxo e por classe. No caso do exemplo que tem vindo a ser seguido, é possível aprofundar a análise e visualizar o seu tipo de comportamento.

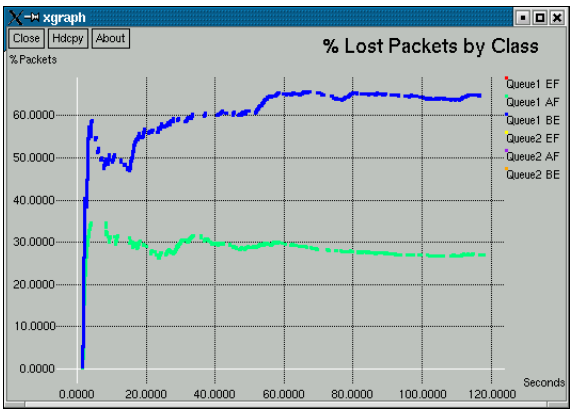

Figura 10 – Análise gráfica do *throughput* por classe relativa ao "teste Pareto"

O gráfico da Figura 10 mostra os valores de perdas verificados para as diversas classes durante o tempo total de simulação. O tráfego classificado como EF apresenta numero de perdas nulo seguindo-se o tráfego AF com uma percentagem média de perdas de 27% e BE com 64.8%. Este ultimo tem um valor muito alto já que depende da largura de banda que se encontre ainda disponível.

A Figura 11 mostra o gráfico dos diversos valores de atraso registados nas filas de espera. A fila 2 situada entre C-E2 apresenta valores nulos dada a largura de banda ser suficiente para não colocar os pacotes em buffer. O tráfego classificado como EF é o que apresenta o atraso mais baixo, seguindo-se AF e BE.

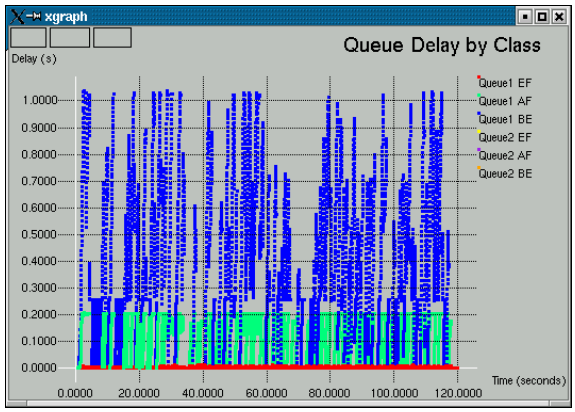

Figura 11 – Análise gráfica do atraso da fila de espera por classe relativa ao "Teste Pareto"

Para a parametrização efectuada, os resultados obtidos através das Figuras 10 e 11 mostram que os parâmetros de QoS configurados conduzem a baixas perdas e atrasos na entrega registando-se um aumento desses valores para classes de menor prioridade. Com esta aplicação de monitorização seria possível afinar a parametrização dos mecanismos de QoS por forma a ajustar os níveis de serviço e o desempenho global da rede.

5.6 Análise On-line

A criação de um interface de monitorização *on-line* tal como mostra a Figura 12 permite que através da leitura do ficheiro *trace* se possa observar dinamicamente o comportamento dos nós fronteira face ao número de pacotes recebidos, descartados e reencaminhados durante o intervalo de simulação, assim como, o *throughput* para os fluxos definidos.

Paralelamente, é possível observar uma animação gráfica da evolução de todos os elementos do modelo recorrendo ao comando NAM. Seguindo ainda este tipo de análise foram criadas mensagens de rodapé na janela que vão acompanhando a simulação, indicando por exemplo, a percentagem de perda de pacotes, o volume de tráfego entrado e

saído nos nós fronteira, ou o número de pacotes existentes nas filas de espera.

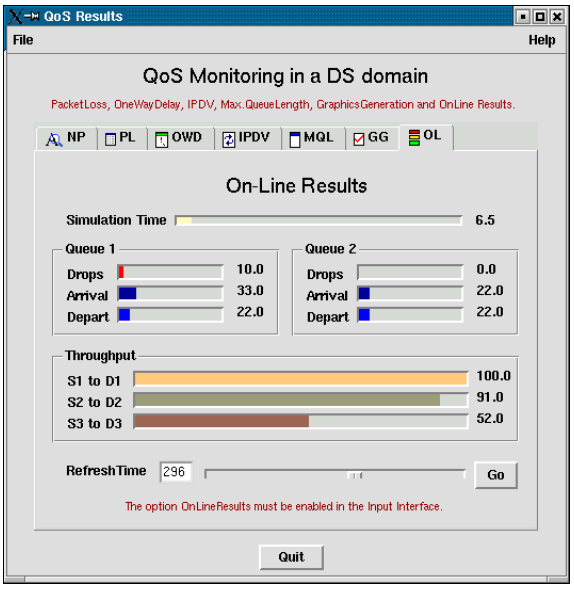

Figura 12 – Quadro da aplicação em Results - ”On-Line”

6. CONCLUSÕES

Neste trabalho foram abordados aspectos da gestão e monitorização da QoS em redes IP com serviços diferenciados e dos vários mecanismos que compõem esta arquitectura. Nesse sentido, foi desenvolvida uma aplicação cujo interface permite introduzir os valores de simulação referentes ao tipo de rede utilizada, policiamento a aplicar sobre o modelo implementado, gestão de filas de espera, escalonadores a utilizar, tempo total de simulação e tipo de resultados a obter. Os resultados disponibilizados permitem desta forma avaliar os níveis de serviço face à parametrização das fontes de tráfego e dos mecanismos que a controlam. A análise é efectuada tanto numa perspectiva de fluxo como de classe. É também disponibilizada uma visualização gráfica e dinâmica dos resultados.

A plataforma desenvolvida encontra aplicação no domínio da investigação e área académica permitindo de uma forma integrada afinar os diversos mecanismos de controlo de tráfego optimizando o desempenho global da rede e prevenindo eventuais falhas de QoS.

Uma maior flexibilização da plataforma, tanto a nível dos protocolos suportados como do modelo de rede subjacente, serão objecto de trabalho futuro.

REFERÊNCIAS

[1] R. Braden, D.Clark, S. Shenker, RFC 1633 – Integrated Services in the Internet Architecture: an Overview, Junho 1994.

[2] D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, RFC 2475 – An Architecture for Differentiated Services, Dezembro 1998.

[3] K. Nichols et al., Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474, IETF, Dezembro, 1998.

[4] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, RFC 2597 - Assured Forwarding PHB, Junho 1999.

[5] B. Davie et al. , An Expedited Forwarding PHB, RFC 3246, Março 2002.

[6] K. Nichols, Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification, RFC 3086, IETF, Abril, 2001.

[7] NAM, XGraph and NS Manuals, http://www.isi.edu/nsnam/.

[8] Visual TCL, http://vtcl.sourceforge.net/.

[9] S. Shenker and C. Partridge, "Specification of Guaranteed Quality of Service", RFC 2212, Septembro 1997.

[10] W. Fang e al., RFC 2859 – A Time Sliding Window Three Colour Marker, Junho 2000.

[11] J. Heinanen, R. Guerin, RFC 2697 – A Single Rate Three Color Marker, Setembro 1999.

[12] J. Heinanen, R. Guerin, RFC 2698 – A Two Rate Three Color Marker, Setembro 1999.

[13] Floyd and Jacobson, Random Early Detection - Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, v.1 n.4, Agosto 1993.

[14] D. Clark and W. Fang, Explicit allocation of best effort packet delivery service, IEEE/ACM Trans. on Networking, vol.6, pp.362-373, 1998.

[15] DiffServ Compliant Weighted RED, http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdswred.pdf .

[16] L. Kleinrock, Queueing Systems. Volume 1, New York: John Wiley, 1974.

[17] Planning for QoS - Queuing Techniques, http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/ciscoasu/class/qpm1_1/using_qo/c1plan.pdf.

[18] The Joint DANTE/TERENA Task Force TF-TANT, http://www.dante.net/tf-tant/.

[19] S. Radhakrishan, Internet Protocol QoS Page, “Implementations and Evaluation-Performance Measurements”, http://qos.ittc.ukans.edu .

[20] T. Ferrari, TF-TANT, “Differentiated Services: Experiment Report – phase2”, Maio 2000.

[21] G. Almes, S. Kalidindi, M. Zekauskas, “One-Way Delay Metric for IPPM”, RFC 2679, Set. 1999.

[22] C. Demichelis, P. Chimento, “IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)”, Novembro 2002.