

AVALIAÇÃO DE QoS NUMA ARQUITECTURA *DIFFSERV*

José Faria¹, Alexandre Santos²
Universidade do Minho, Dep. De Informática
Campus de Gualtar
4710-057 Braga

¹{jose@di.uminho.pt}

²{alex@di.uminho.pt}

Palavras chave: Qualidade de Serviço, QoS, Diffserv, ALTQ.

Resumo¹

Estabeleceu-se uma plataforma de teste com um encaminhador que implementa o paradigma Diffserv. Este encaminhador revelou que é possível fornecer serviços com qualidade recorrendo á implementação de classes de tráfego, cada uma com características específicas, de forma independente e sem interferências.

I.Introdução

A Qualidade de Serviço² é um tópico muito controverso quando nos referimos a redes de dados, nos dias de hoje. Controverso porque o termo **QoS** significa diferentes comportamentos da rede consoante o que dela esperamos, aliás, o próprio termo é ambíguo. Para uns pode significar elevadas taxas de transferência, enquanto para outros, pode significar ter perdas ou atrasos dentro de um intervalo determinado.

O desenvolvimento de aplicações cada vez mais exigentes, em termos de **QoS**, está frequentemente acima das capacidades oferecidas pela, ainda actual, técnica do "melhor esforço" das redes **TCP/IP**.

Apesar disto já existem tecnologias de interligação, em rede, capazes de fornecer a **QoS** que os utilizadores/aplicações podem requisitar. Existem essencialmente duas abordagens: o modelo de serviços integrados, o **Intserv** e o modelo de serviços diferenciados, o **Diffserv**[1][2][3].

O objectivo principal deste trabalho consistiu na implementação de uma plataforma com um encaminhador de tráfego, implementando o Diffserv. Este encaminhador fazia a admissão, classificação e re-classificação de tráfego, segundo algumas regras, encaminhando-o posteriormente de forma independente e consertada.

1 Trabalho parcialmente financiado pelo Programa PRAXIS XXI do MCT,P/EEI/10168/1998

2 Na terminologia inglesa **Quality of Service**, frequentemente referido por **QoS**.

II. Qualidade de Serviço e Classes de Serviço

A Qualidade de serviço em redes de computadores consiste na implementação, nas actuais infra-estruturas de rede, via mecanismos ou modelos, do suporte á diferenciação de tráfego[4]. A palavra chave é "**Diferenciação**". Consiste na criação de classes de tráfego, atribuindo a cada uma um conjunto de características, de forma que cada classe obtém da rede um comportamento que a diferencia das outras.

É muito importante que estas características, conferidas a cada classe, se mantenham inalteradas e consistentes ao longo do tempo e da duração do serviço.

Estas características dizem respeito aos tempos de resposta fim-a-fim, tempos de latência, atrasos nas filas de espera, largura de banda disponível, etc. Algumas destas características são mais previsíveis do que outras, dependendo do tipo de tráfego, das disciplinas das filas de espera, do tamanho dos *buffers* do meio ou tecnologia empregue.

É provável que algumas pessoas estejam a falar de **QoS** mas o que na realidade estarão a pensar é em classes de serviço diferenciadas. Quando muito, estarão a referir-se a classes de serviço diferenciadas acompanhadas por um controlo de tráfego, monitorização e administração.

III. Avaliação de QoS numa Arquitectura *Diffserv*

Para testar e avaliar o paradigma *Diffserv* definiu-se e implementou-se uma arquitectura, representada na figura 1, cujo objectivo consistiu na análise da classificação, re-classificação e controlo de admissão de tráfego. Para auxiliar as conclusões sobre estes objectivos foram retirados os seguintes valores:

- número de pacotes recebidos por classe
- número de pacotes expedidos por classe
- número de pacotes que foram descartados por classe
- número de Bytes enviados por classe
- número de Bytes perdidos por classe
- número de pacotes na fila por cada classe

Estes dados foram recolhidos recorrendo a uma ferramenta desenvolvida por **Kenjiro Cho**[5], com módulos que podem ser incorporados no *Kernel*, neste caso do **FreeBSD**. Estes módulos vão desde a implementação de algoritmos de gestão e escalonamento em filas: **FIFO**, **CBQ**, **WFQ**, etc., até a módulos para implementar o acondicionamento do tráfego em classes segundo um conjunto de regras.

A.Implementação Diffserv

Para implementar o **Diffserv** foi instalado no *kernel* do encaminhador a ferramenta, já referida, de **Kenjiro Cho**. Esta ferramenta chamada **ALTQ**, esquematicamente representada na figura 2, permitiu-nos criar várias classes de tráfego com diferentes características.

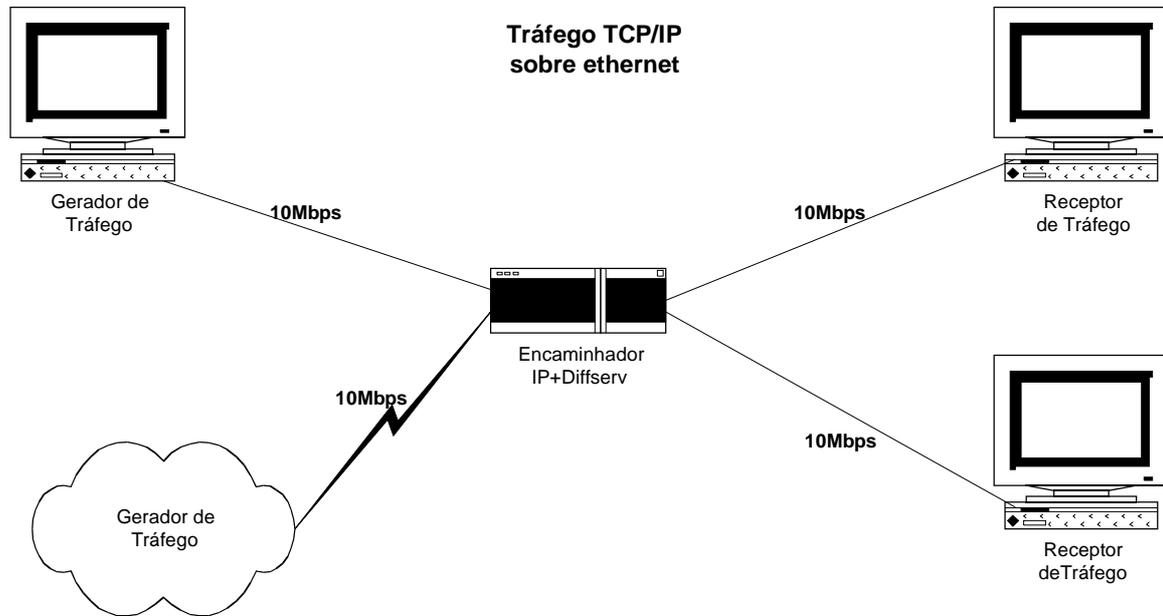


Fig. 1 – Arquitectura de Teste **Diffserv**

Neste teste estavam em jogo 3 interfaces: dois interfaces de chegada de tráfego e um terceiro para a saída de tráfego. Um pacote ao chegar ao interface de entrada era tratado seguindo o seguinte algoritmo:

- Chegada do pacote ao interface
- Inspecciona qual a porta destino/fonte e qual o endereço **IP** destino/fonte
- Consoante estes valores o pacote é etiquetado no cabeçalho, no campo **TOS/CS**³[6][7], com o **DSCP**⁴ adequado.
- O pacote é passado à função de saída (neste caso em estudo, ao **CBQ**)
- A função invocada olha para o campo **TOS/CS** e para o endereço/porta **IP** destino e vai colocar o pacote na fila adequada.
- Entretanto o algoritmo de gestão da fila vai expedindo os pacotes.

No âmbito do IETF⁵, e para o **Diffserv**, já foram definidos duas etiquetas para o **DSCP**[8]: **EF** (Expedited Forward) e o **AF** (Assured Forward).

3 IPv4 usa TOS (Type of Service). IPv6 usa CS(Classe of Service).

4 DiffServ CodePoint.

5 Internet Engineer Task Force.

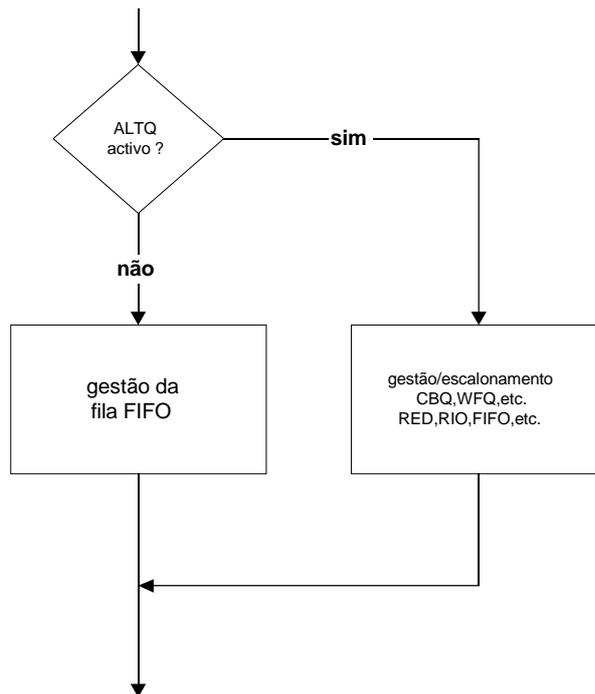


Fig. 2 – ALTQ no kernel

Convém referir que o encaminhador está na situação de fronteira e por isso faz também o controlo de admissão do tráfego além da sua etiquetagem.

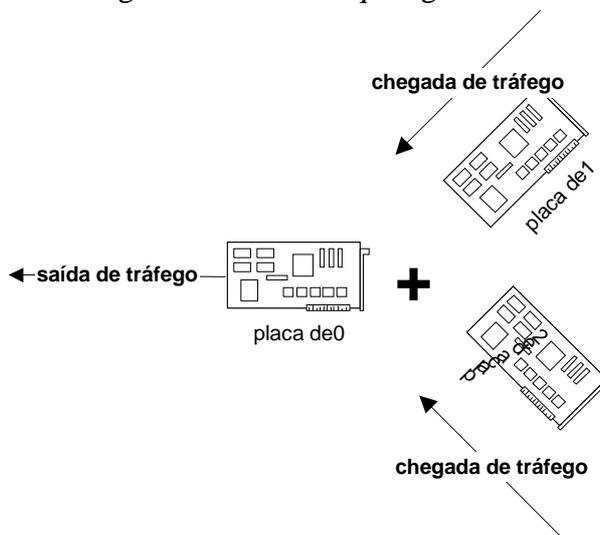


Fig. 3 – Interfaces de entrada/saída

O comportamento da ferramenta **ALTQ** é definido por um ficheiro de configuração e, como sugerido pela figura 3, o encaminhador possui 3 interface (*de0*, *de1* e *de2*) que foram configurados genericamente da seguinte forma:

- Todo o tráfego que chega pelo interface *de2* vai ser etiquetado com a etiqueta *0x00* (tráfego *Best-effort*). Isto é, até 10Mbps, o que excede este valor é descartado.
- O tráfego que chega pelo interface *de1* e com destino á porta 30000, será marcado com a etiqueta **EF** até 2Mbps. O excedente será reclassificado como *Best-effort*.

- O tráfego que chega no interface *de1* e com destino á porta 31000, será marcado com a etiqueta **AF11** até 1000Kbps, **AF12** de 1000 a 1800Kbps, **AF13** de 1800 a 2200Kbps e todo o excedente será descartado.
- Ao interface de0 foi associado o algoritmo **CBQ** criando as filas **be**, **ef**, **af11**, **af12** e **af13**. Neste ficheiro de configuração definiram-se os filtros que pegam nos pacotes, depois de marcados ou re-classificados e colocam-nos nas filas respectivas.

Pode-se ainda definir quais os pacotes que entram e como são etiquetados, realizando controlo de admissão. Em vez de colocar uma etiqueta podemos colocar a palavra **<drop>** que significa que nesse caso o pacote deve ser descartado.

IV. Testes

Os testes foram realizados tendo em vista avaliar o comportamento do encaminhador em cada classe e nas seguintes situações:

- abaixo da sua largura de banda disponível
- próximo da largura de banda disponível
- no limite da largura de banda disponível
- para além da largura de banda disponível

Os testes foram realizados em separados mas no fim foi realizado um teste em que o encaminhador tinha á entrada, em simultâneo, tráfego **EF**, **AF11** e *Best-effort*. Neste último teste foram sendo aumentadas as cargas do *Best-effort*, ao ponto de entrar em sobre-carga, e nesse altura interessou verificar se as outras classes se mostravam indiferentes e sem interferências. As tabelas 1, 2, 3 e 4 indicam os valores do tráfego gerado.

Teste n.	Tráfego Gerado	Tamanho do Pacote(bytes)	N.º de pacotes/segundo
1	1 Mbps	625	200
2	1,9 Mbps	1188	200
3	2 Mbps	1250	200
4	2,2 Mbps	1375	200

Tab. 1 – Testes para a classe **EF**

Teste n.	Tráfego Gerado	Tamanho do Pacote(bytes)	N.º de pacotes/segundo
1	500 Kbps	313	200
2	900 Kbps	563	200
3	1200 Kbps	750	200
4	1600 Kbps	1000	200
5	2000 Kbps	1250	200
6	2288 Kbps	1430	200

Tab. 2 – Testes para a classe **AF**

Teste n.	Tráfego Gerado	Tamanho do Pacote(bytes)	N.º de pacotes/segundo
1	4 Mbps	1250	400
2	5,6 Mbps	1250	560
3	5,8 Mbps	1250	580
4	8 Mbps	1250	800

Tab. 3 – Testes para a classe *Best-effort*

Depois destes testes isolados realizaram-se os seguintes testes com tráfego diferente em simultâneo.

Cenário	Classe	Tráfego Gerado	Tamanho dos pacotes (bytes)	N.º pacotes/segundo
0	EF	2000 Kbps	1250	200
	AF	1000 Kbps	625	200
1	EF	2000 Kbps	1250	200
	AF	1000 Kbps	625	200
	Best-effort	4000 Kbps	1250	400
2	EF	2000 Kbps	1250	200
	AF	1000 Kbps	625	200
	Best-effort	5600 Kbps	1250	560
3	EF	2000 Kbps	1250	200
	AF	1000 Kbps	625	200
	Best-effort	5800 Kbps	1250	580
4	EF	2000 Kbps	1250	200
	AF	1000 Kbps	625	200
	Best-effort	8000 Kbps	1250	800

Tab. 4 – Tráfego em concorrência

A.Análise dos Testes

A análise dos resultados assenta essencialmente em dois aspectos:

- Pacotes encaminhados: evidencia o facto dos dados serem transportados dentro de pacotes.
- Bits encaminhados: este tipo de análise é mais “fina” porque se analisa o número de bits, em concreto, que fora expedidos.

Os resultados recolhidos nos testes são apresentados pelas tabelas seguintes.

1) Classe Best-effort

O gráfico da figura 4 apresenta os resultados para o teste com o tráfego Best-effort e podemos observar o seguinte:

- Para 4 e 5,6 Mbps de carga o encaminhador não evidencia qualquer problema
- Para 5,8 Mbp, que coincide com a largura de banda disponível, existe já um pequena quantidade de pacotes que são descartados. Seria de esperar que todos os pacotes fossem encaminhados, o que não acontece. A explicação para o sucedido vai para a fonte de geração de tráfego que se supõe não ser uniforme. Isto é, gera 5,8 Mbps, mas pode em meio segundo gerar 5,6 e na outra metade de segundo gerar 6 Mbps, o que dá 5,6 Mbps em média. Admitimos um pequeno “burstiness”.
- Para um tráfego de 8 Mbps como seria de esperar, pela configuração, o excedente foi descartado. Embora o interface tivesse largura de banda disponível esta não foi usada porque estava atribuída às outras classes. Os 68% encaminhados correspondem aproximadamente a 5,4 Mbps o que fica aquém dos 5,8 Mbps da largura da banda. A explicação para esta discrepância vai também para a fonte de tráfego que não deve ser uniforme.

Teste n.	Pacotes encaminhados	Pacotes descartados
1	2000	0
2	2801	0
3	2809	91
4	2712	1289

Tab. 5 – Pacotes encaminhados *versus* descartados na classe Best-effort

Teste n.	Kbps gerados	Kbps encaminhados
1	4000	4000
2	5600	5601
3	5800	5619
4	8000	5424

Tab. 6 – Kbps encaminhados na classe Best-effort

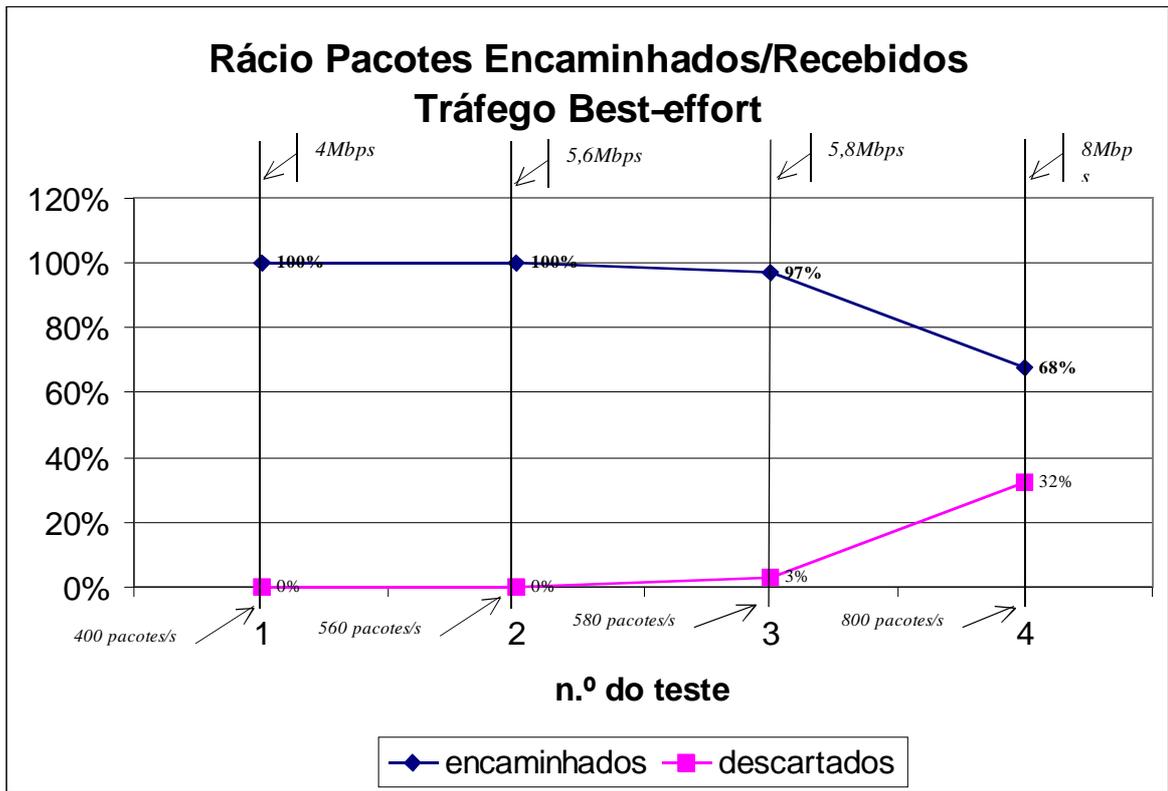


Fig. 4 – Gráfico pacotes para Best-effort

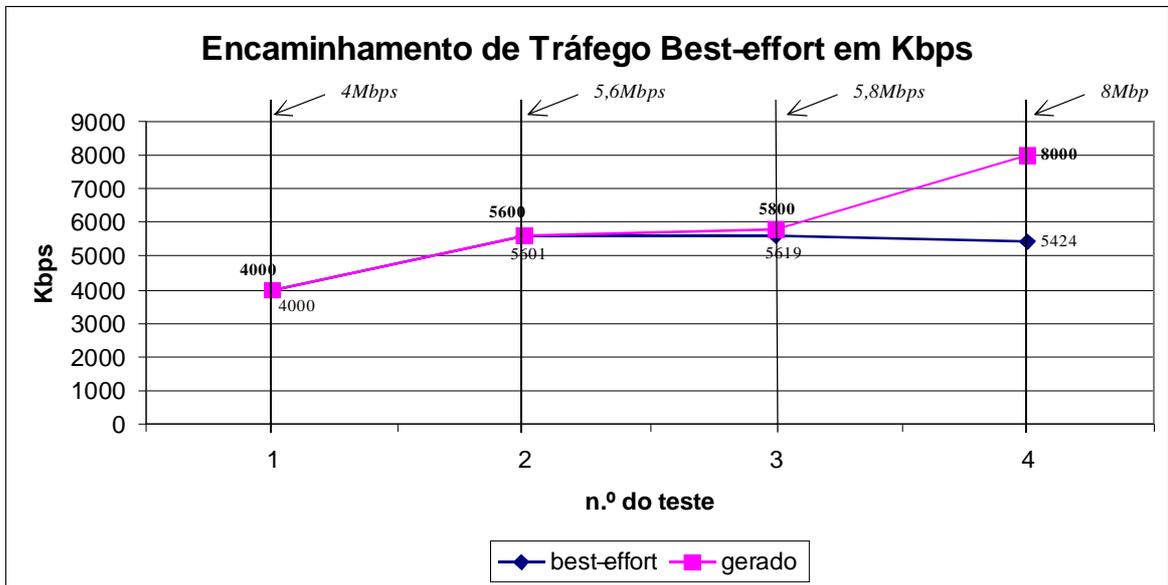


Fig. 5 – Gráfico Kbps para Best-effort

2) Classe EF

Os testes para esta classe tem comportamentos muito semelhantes ao tráfego *Best-effort*. As explicações para as pequenas diferenças julga-se serem também são as mesmas. O gráfico da figura 5 apresenta o comportamento para esta classe.

Convém referir que no caso desta classe, tinha sido definido que o tráfego que excede os 2 Mbps é re-classificado como *Best-effort*, por isso que no gráfico aparece pacotes a serem expedidos na classe *Best-effort*.

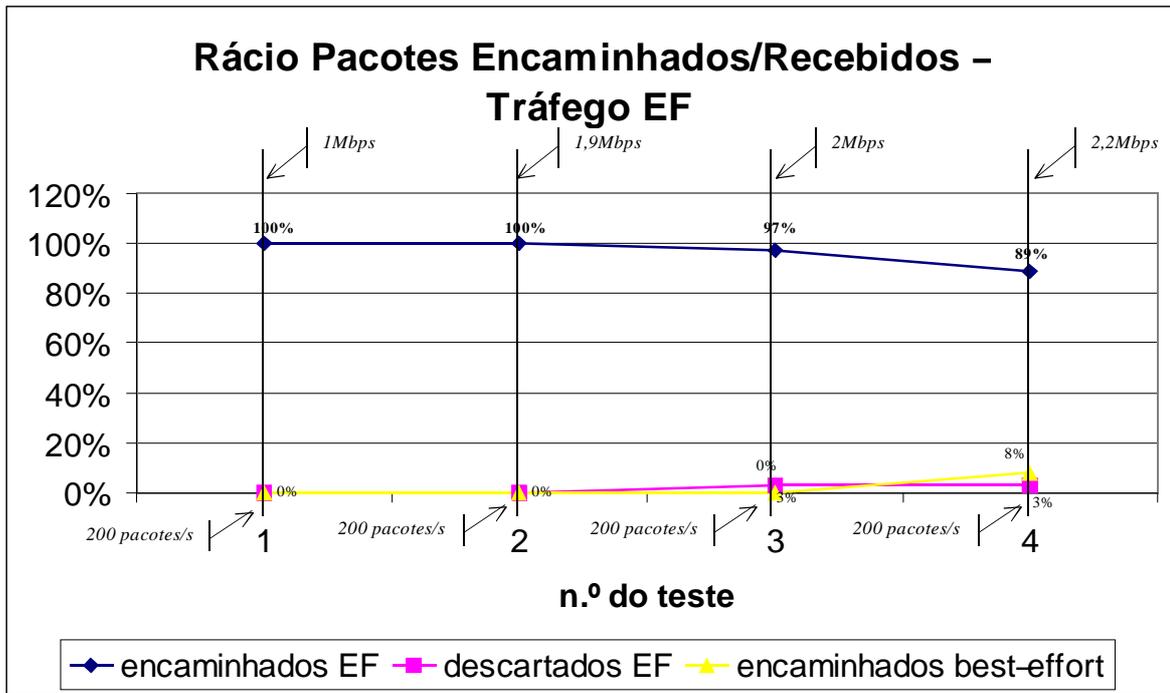


Fig. 6 – Gráfico de pacotes para EF

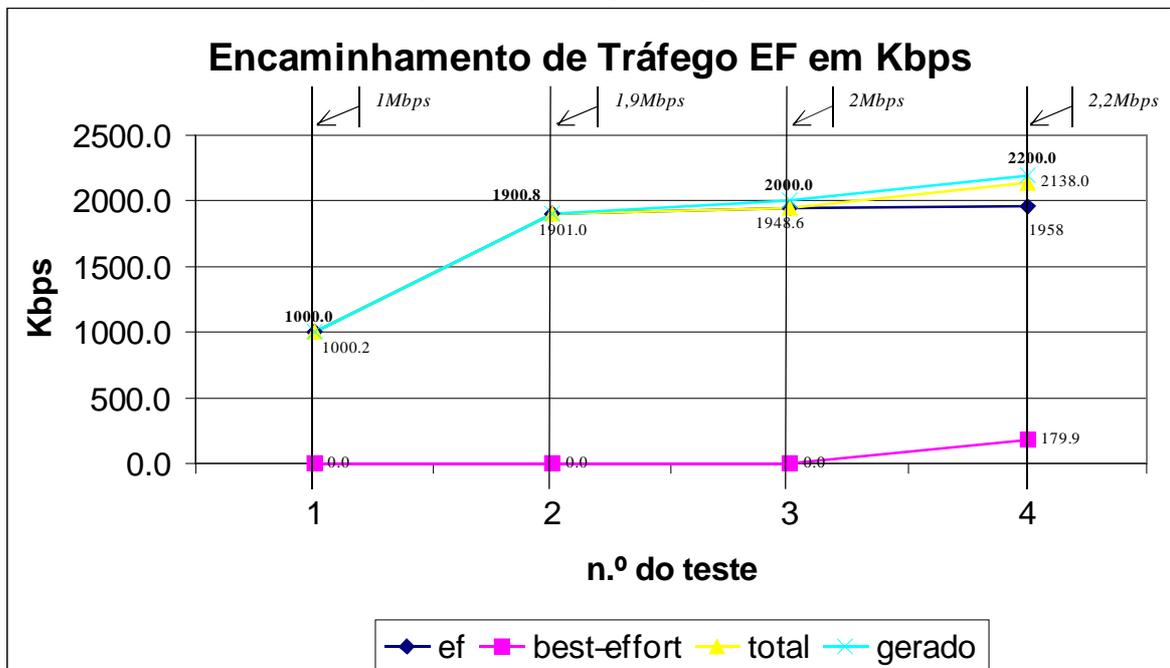


Fig. 7 – Gráfico Kbps para EF

Teste n.	Pacotes EF encaminhados	Pacotes EF descartados	Pacotes Best-effort encaminhados
1	1000	0	0
2	1000	0	0
3	974	26	0
4	890	28	82

Tab. 7 – pacotes encaminhados *versus* descartados para EF

Teste n.	Kbps gerados	Kbps EF encaminhados	Kbps Best-effort encaminhados	Kbps total
1	1000	1000	0	1000
2	1901	1901	0	1901
3	2000	1949	0	1949
4	2200	1958	180	2138

Tab. 8 – Kbps encaminhados para EF

3) Classe AF

A explicação e o comportamento desta classe é muito semelhante às anteriores. De referir que pela análise dos gráficos 8 e 9 se pode ver que conforme se vai aumentando a carga, as classes AF11, AF12 e AF13 vão enchendo ao ponto de estabilizarem a quantidade recolhida e encaminhada nesta sub-classe.

Teste n.	Pacotes AF11 encaminhados	Pacotes AF12 encaminhados	Pacotes AF13 encaminhados	Pacotes AF11 descartados	Pacotes AF12 descartados	Pacotes AF13 descartados
1	1000	0	0	0	0	0
2	1000	0	0	0	0	0
3	791	151	0	58	0	0
4	601	366	0	33	0	0
5	489	387	90	17	18	0
6	429	343	168	13	10	37

Tab. 9 – Pacotes encaminhados *versus* descartados para AF

Teste n.º	Kbps AF11 encaminhados	Kbps AF12 encaminhados	Kbps AF13 encaminhados	Kbps descartados	Kbps total	Kbps Gerados
1	501	0	0	0	501	501
2	901	0	0	0	901	901

Teste n.º	Kbps AF11 encaminhados	Kbps AF12 encaminhados	Kbps AF13 encaminhados	Kbps descartados	Kbps total	Kbps Gerados
3	948	181	0	70	1130	1200
4	961	586	0	53	1547	1600
5	978	774	180	69	1931	2000
6	980	786	385	137	2151	2200

Tab. 10 – Kbps encaminhados para a classe AF

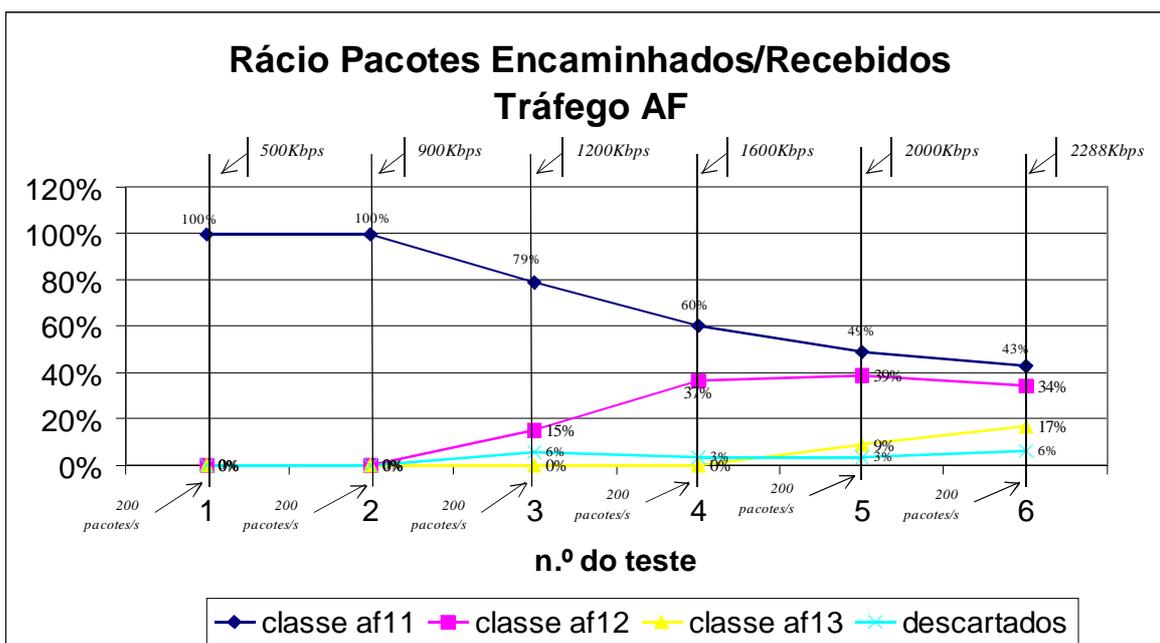


Fig. 8 – Gráfico pacotes para AF

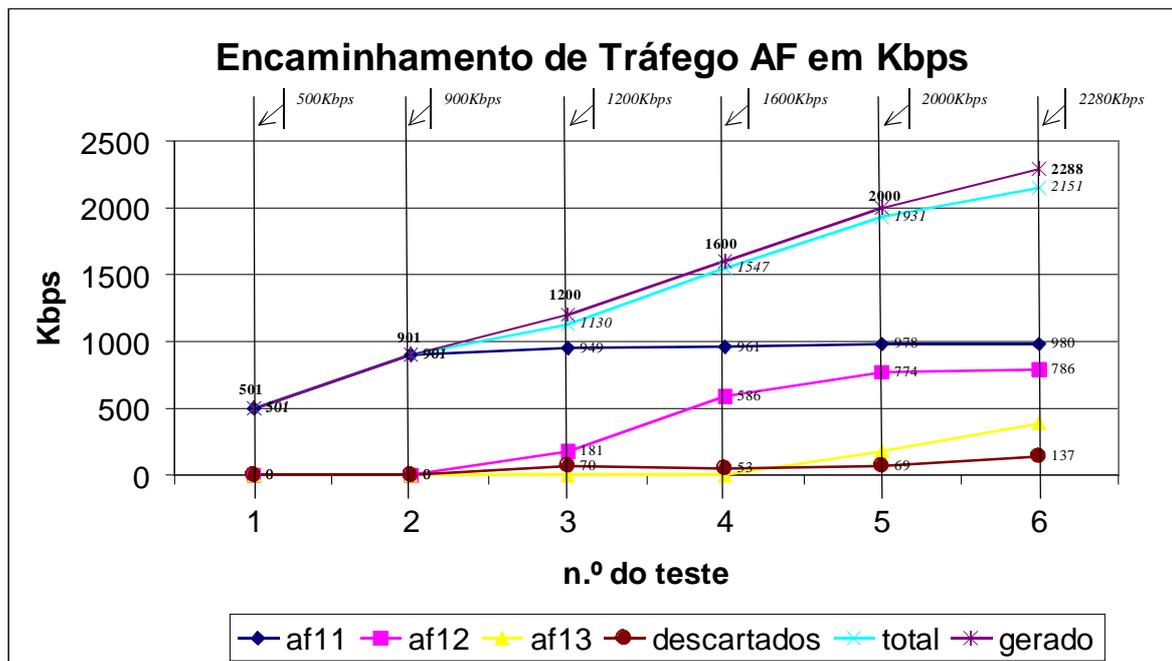


Fig. 9 – Gráfico Kbps para AF

4) Cenários com Tráfego concorrente

As análises anteriores servem de base para garantir que o comportamento nestes testes, para as classes *Best-effort*, **EF** e **AF** são coerentes, quer isoladamente quer em concorrência. Como já foi referido, nestes cenários pretende-se verificar se com uma classe em sobre-carga as outras se mantêm independentes e sem interferências.

O que se verificou foi que a carga do *Best-effort* foi aumentando ao ponto de entrar em congestão e as outras classes, como se pode ver pelo gráfico da figura 10, não sofreram com esse excesso de tráfego. A figura 12 exemplifica o caso de existirem ou não classes para o tráfego.

Cenário n.º	Pacotes AF11 encaminhados	Pacotes AF11 descartados	Pacotes EF encaminhados	Pacotes EF descartados	Pacotes Best-effort encaminhados	Pacotes Best-effort descartados
0	897	103	980	21	0	0
1	958	42	993	8	1999	0
2	958	42	989	11	2798	0
3	958	42	991	10	2860	39
4	954	46	988	12	2803	1192

Tab. 11 – pacotes encaminhados versus descartados nos cenários

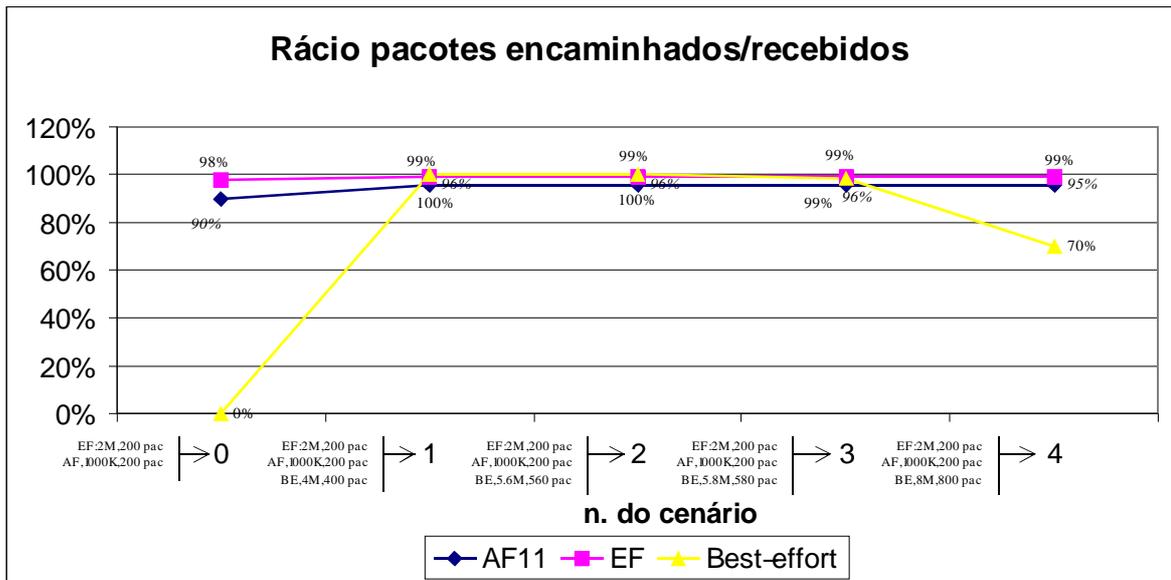


Fig. 10 – Gráfico para os cenários

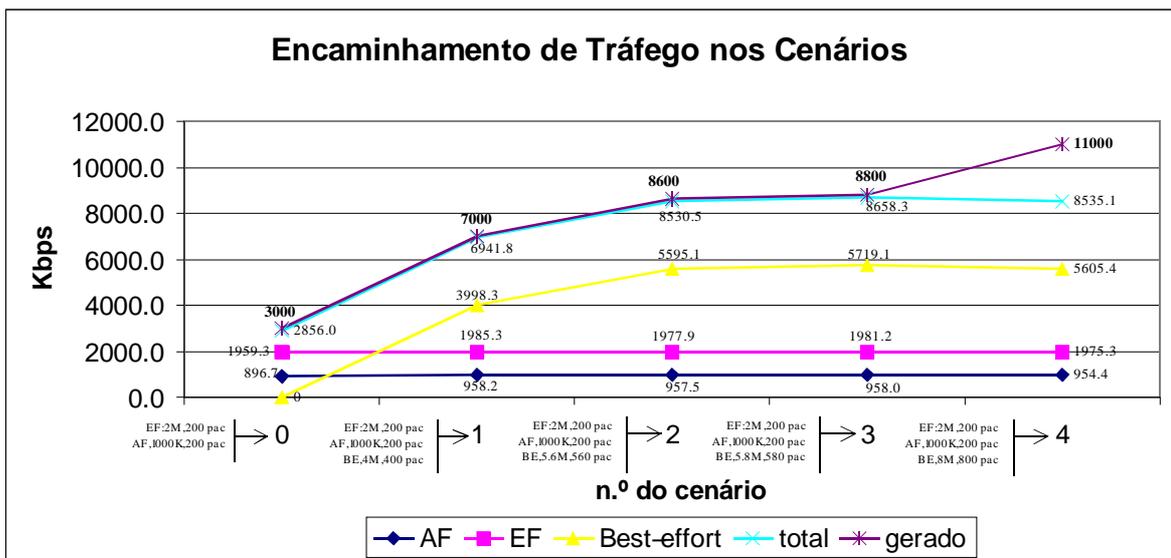


Fig. 11 – Gráfico Kbps para os cenários

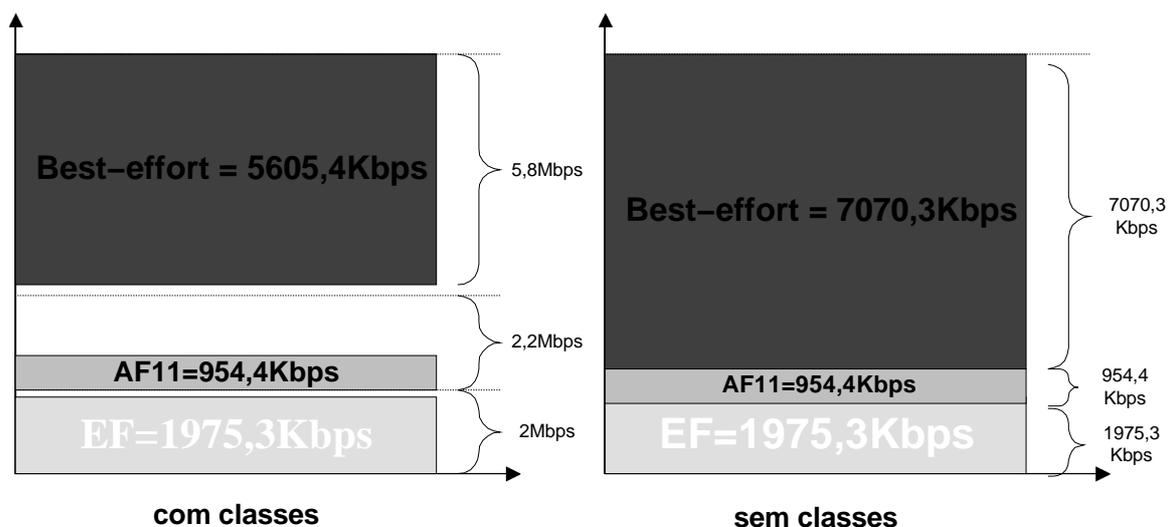


Fig. 12 – Largura de banda ocupada com ou sem classes

V. Conclusões

Actualmente os níveis de exigência em termos de redes e de serviços de rede são muito superiores ao que era há alguns anos. As aplicações evoluíram muito e criaram expectativas nos utilizadores de tal ordem que toda a tecnologia foi obrigada a acompanhar.

Estas necessidades de maior desempenho e com garantias de resultados levou ao aparecimento de novas tecnologias (B-ISDN e ATM por ex.).

A par destas novas tecnologias foram desenvolvidos e estudados novos paradigmas de rede, dos quais se destaca o **Diffserv**.

Após a realização dos testes e análise dos resultados ficou a ideia de que o **Diffserv** apresenta características que prometem cumprir pelo menos alguns dos objectivos na obtenção de **QoS**.

Os testes mostraram que é possível manter, num mesmo encaminhador, várias classes de tráfego e que estas se mantêm independentes quer na qualidade e quantidade umas das outras. O comportamento de uma não interfere nas características das outras.

Mas ao tráfego com origens, destinos, conteúdos, etc., diferentes, mas que num dado encaminhador **Diffserv** é conhecido por tráfego agregado, quando excede a largura de banda atribuída voltamos a ter o velho problema de escassez de largura de banda.

O encaminhador neste teste funcionou como encaminhador fronteira, que além de classificar e re-classificar tráfego, procedendo á sua etiquetagem, fazia também o controlo de admissão. Desta forma é possível filtrar os tráfegos á entrada e deixar passar apenas os autorizados.

As teorias apresentadas nos documentos do **Diffserv**, para classificação e controlo de admissão, no contexto construído e com os testes realizados, parecem ser válidas e funcionais.

VI.Referências

- [1] J. Wroclawki. The Use of RSVP with IETF Integrated Services. Technical Report, RFC2210 from IETF, 1997
- [2] P. Oechslin, W. Almesberger and J. Le boudec. Technical Report, RFC2170 from IETF, 1997
- [3] M. Carlson, E. Davies, Z. Wang, D. Black and S. Blake. An Architecture for Differentiated Services. Technical Report, RFC2475 from IETF, 1998
- [4] Paul Ferguson and Geoff Huston. Quality of Service – Delivering QoS on the Internet and in Corporate Networks. Wiley Computer Publishing, 1998
- [5] Kenjiro Cho, A Framework for Alternate Queueing: Towards Traffic Management by PC–Unix Based Routers. Technical Report, Sony Computer Science Laboratory, Inc., Tokyo, Japan, 1999
- [6] J. Postel. Internet protocol, std 5. Technical Report, RFC791 from IETF, 1981
- [7] S. Deering and R. Hinden. Internet protocol, version 6 (Ipv6) specification. Technical Report, RFC2460 from IETF, 1998
- [8] F. Baker, K. Nichols, S. Blake and D. Black. Definition of the Differentiated Services Field (ds field) in the IPv4 and IPv6 Headers. Technical Report, RFC2474 from IETF, 1998