Estabelecimento e utilização de uma plataforma DiffServ gerida por um *Bandwidth Broker*

Óscar Gama, Paulo Carvalho, Solange Lima
osg@di.uminho.pt, {paulo, solange}@uminho.pt

Universidade do Minho

Resumo

O *Bandwidth Broker* (BB) é uma entidade central que num domínio de Serviços Diferenciados realiza o controlo de admissão através da gestão e supervisão global dos recursos disponíveis, podendo também controlar e regular o tráfego aí existente. Este artigo descreve o estabelecimento de uma plataforma experimental gerido por um BB e apresenta um conjunto de testes, realizados com tráfego UDP e TCP, que permitirá avaliar a operacionalidade da plataforma e a importância da utilização do BB na manutenção da QoS requerida pelas aplicações.

1. Introdução

Hoje em dia existe uma grande diversidade de aplicações necessitando de melhores níveis de serviço do que o melhor−esforço oferecido pela rede IP. Por exemplo, os serviços de difusão de vídeo e de telefonia IP, para que sejam úteis, necessitam que a rede disponha de qualidade de serviço adequada para proporcionar, se possível, largura de banda garantida, perda de pacotes limitada, atraso e *jitter* reduzidos.
A Qualidade de Serviço (QoS, *Quality of Service*) traduz o nível de desempenho que uma rede deve oferecer a uma dada aplicação, a fim de esta apresentar os padrões de qualidade convenientes e esperados. Do ponto de vista do ISP, a QoS pode ser definida como a capacidade da sua infra−estrutura de comunicação fornecer um serviço não padronizado a um subconjunto de tráfego, agregado ou não. Na perspectiva do utilizador, pode ser definido como o efeito colectivo do desempenho apresentado por um serviço e que determina o grau de satisfação do cliente no uso deste. A QoS pode ser expressa objectivamente através de vários parâmetros. Os mais importantes são normalmente a largura de banda oferecida, a latência da comunicação (atraso), a variação do atraso (*jitter*), a taxa de perda de pacotes e o débito efectivo recebido (*throughput*).
No modelo DiffServ [Blake98], apontado como o mais promissor para providenciar QoS na Internet, o tráfego de aplicações com requisitos de QoS idênticos é agregado numa classe. As classes são depois tratadas de forma diferenciada pelos *routers* dos domínios a fim de prestar−lhes o serviço pretendido. Portanto, não há um tratamento individual para cada fluxo, mas sim um tratamento agregado para todos os fluxos pertencentes à mesma classe. O tráfego de uma classe de serviço apresenta um comportamento de encaminhamento similar nos vários nós, ou seja, apresenta o mesmo PHB (*Per−Hop Behaviour*). O grupo IETF DiffServ não pretende definir um modelo de QoS fim−a−fim, mas apenas definir os PHBs de um número pequeno de classes de tráfego. Existem dois PHBs normalizados: o EF (*Expedited Forwarding*) [Davie02] e o AF (*Assured Forwarding*) [Heinanen99]. O primeiro é usado quando se pretende um serviço com elevada fiabilidade (baixa probabilidade de perdas, atraso e *jitter* reduzidos, largura de banda garantida). O segundo não garante atraso nem *jitter* reduzidos, mas oferece um serviço onde o tráfego conforme tem maior probabilidade de atingir o destino do que o tráfego não−conforme.
Numa rede com QoS torna−se conveniente reservar/definir previamente nesta uma certa quantidade de recursos, a fim de prover ao cliente um serviço com QoS. Ora, como estes recursos são necessariamente finitos, torna−se importante que existam mecanismos que controlem a admissão de novos fluxos de tráfego à rede por forma que a QoS garantida às aplicações existentes não seja afectada negativamente pela presença de uma nova aplicação. Para tal são necessários mecanismos de controlo de tráfego e, em particular, de controlo de admissão.
Várias metodologias, distribuídas e centralizadas, têm sido propostas para realizar a gestão e o controlo de admissão num domínio. O PBAC (*Parameter−Based Admission Control*) caracteriza−se por um pedido explícito de reserva de recursos à rede, o MBAC (*Measurement*−BAC) realiza o controlo com base em estimativas indicativas do grau de desempenho e/ou congestão da rede, o EMBAC (*End−to−end* MBAC) estima indirectamente o grau de congestão da rede usando tráfego de prova entre as máquinas origem e destino [Elek00].

Na concepção centralizada destaca−se o *Bandwidth Broker* (BB) [Jacobson99], entidade que numa rede DiffServ realiza o controlo de admissão através da gestão e supervisão global dos recursos disponíveis, podendo também controlar e regular o tráfego existente no seu domínio.
Este artigo descreve o estabelecimento de uma plataforma DiffServ experimental gerida por um BB e aborda as ferramentas que permitiram a sua utilização. Depois avalia−se, através de um conjunto de testes experimentais usando tráfego UDP e TCP, a sua operacionalidade e a relevância que a utilização do BB desempenha na manutenção da QoS requerida pelas aplicações.

2. *Bandwidth Broker*

Numa rede DiffServ gerida por um BB, os clientes reservam recursos efectuando SLAs e RARs. O SLA (*Service Level Agreement*) é um contrato de serviço administrativo realizado entre um cliente e o fornecedor desse serviço (ISP), ou entre ISPs, especificando o serviço a receber. O SLA é depois traduzido num conjunto de especificações técnicas, designado SLS (*Service Level Specification*), e que o BB deve implementar no seu domínio a fim de garantir o SLA acordado. Uma SLS descreve os recursos necessários e o PHB a aplicar nos *routers* por forma a implementar o serviço acordado no SLA. Os clientes que pretendam usar o serviço definido no SLA efectuam reservas de recursos ao BB através de pedidos RAR (*Resource Allocation Request*).
A arquitectura apresentada pelos BBs pode ser diversa. No entanto, nestes podem−se encontrar normalmente os componentes típicos ilustrados na Figura 1: o servidor, a base de dados, o cliente que efectua pedidos SLA, o cliente que efectua pedidos RAR e o cliente que configura os *routers*.

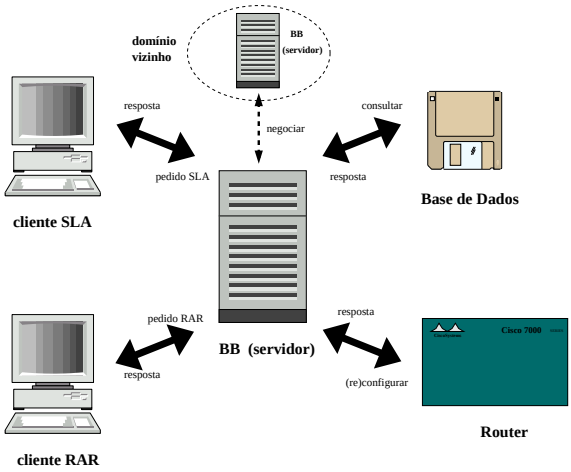


Figura 1 − Arquitectura típica simplificada de um BB

O servidor é responsável pela concessão ou rejeição dos recursos pedidos pelos clientes internos e externos ao domínio, pela monitorização dos recursos livres no seu domínio e pela negociação de SLAs com os BB dos domínios vizinhos. O servidor escuta continuamente eventuais pedidos de clientes. Após um cliente ter sido autenticado e concedido acesso a um serviço, o servidor atribui−lhe os recursos devidos, actualiza a base de dados e depois transmite os comandos de configuração aos *routers* apropriados do domínio. Os cliente efectuam SLAs e RARs ao servidor do BB através de um protocolo de sinalização (e.g., BBTP, RSVP, etc.).
A base de dados do BB é o repositório central que é usado pelo servidor para registar ou consultar SLAs, recursos consumidos, políticas de permissão, mapeamentos do DSCP, etc..
O cliente de configuração dos *routers* permite que estes forneçam o serviço desejado às aplicações após a aceitação de um novo pedido RAR. Aqui há necessidade de distinguir os *routers* internos dos *routers* fronteira, pois enquanto que o tráfego proveniente daqueles pode ser considerado seguro por provir do domínio local, o tráfego proveniente de domínios exteriores terá que ser policiado nos *routers* ingresso, com o *dropper* a fazer eventuais descartes aos pacotes não−conformes. Nos *routers* egresso o tráfego poderá ser regulado através do *shaper*, para que seja entregue ao domínio seguinte em conformidade com o SLA acordado entre domínios.

Por forma a garantir ao cliente um serviço de QoS fim−a−fim, um BB poderá dialogar com os BBs dos domínios vizinhos. Os domínios grandes podem ser divididos em zonas e os BBs podem gerir essas zonas segundo uma estrutura hierárquica. Contudo, apenas o BB representante de um domínio pode contactar o do domínio vizinho.
Em termos conceptuais, o BB introduz no modelo DiffServ um plano superior de controlo e gestão, que permite libertar os *routers* destas funções. Esta é uma das grandes vantagens oferecidas pelo BB, pois permite que sejam desenvolvidos algoritmos e políticas elaboradas de controlo e gestão, sem afectar o desempenho dos *routers*.
Atendendo ao caracter centralizado da solução e à possível sobrecarga (CPU, memória) no processamento dos pedidos RAR, pode−se argumentar que o uso de um BB pode tornar a rede não escalável a partir duma certa dimensão. No entanto, esta questão pode ser abordada usando um BB com *hardware* mais rápido, um BB hierárquico [Zhang02], através do processamento dos pedidos RAR em paralelo e/ou agregar estes pedidos para reduzir a latência de sinalização ao BB. Uma conjugação destas abordagens permitirá reduzir o problema em causa.

3. Descrição da plataforma de testes

A plataforma estabelecida para testes DiffServ foi implementada em ambiente Linux e é composta por uma rede Ethernet 10BaseT com três estações (duas emissoras e uma receptora de tráfego) e um *router*.
No *router* instalou−se o BB kubb [Rao99], que só permite comunicações intradomínio. Do conjunto de implementações gratuitas disponíveis para PCs seleccionou−se este BB porque está de acordo com a actividade de normalização Qbone, no sentido em que, face a um pedido RAR de um cliente, é capaz de: responder ao cliente após o estabelecimento dos parâmetros de QoS; rejeitar sobre−alocação de largura de banda; configurar os *routers* DiffServ e libertar os recursos reservados quando não são necessários. A implementação contém todos os componentes necessários à sua instalação e usa *software* de base de dados publicamente disponível (mySQL). Além disso, o código C está suficientemente comentado e pode ser usado livremente para fins educacionais.
No *router* implementaram−se os PHBs EF, AF1 e BE, através dos quais se estabeleceram três classes de serviço diferenciadas. Para tal utilizou−se a disciplina CBQ [Floyd95], tendo−se respectivamente atribuído 50%, 40% e 10% da largura de banda total às classes EF, AF e BE (em modo *borrow* [1]) e uma prioridade decrescente. Adicionalmente, a classe AF utiliza a disciplina GRED [2] e as classes EF e BE usam a disciplina por defeito (FIFO).

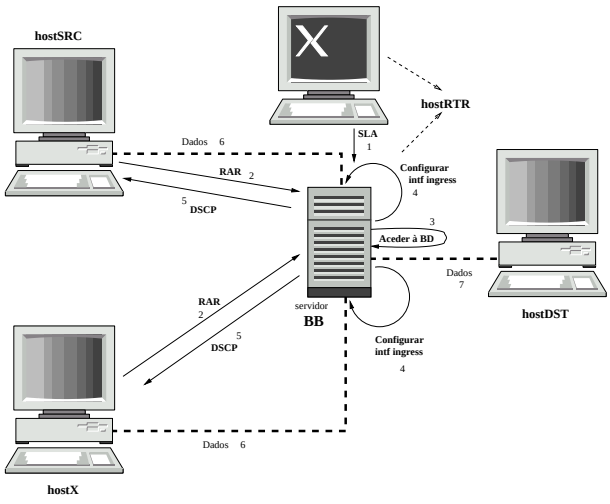
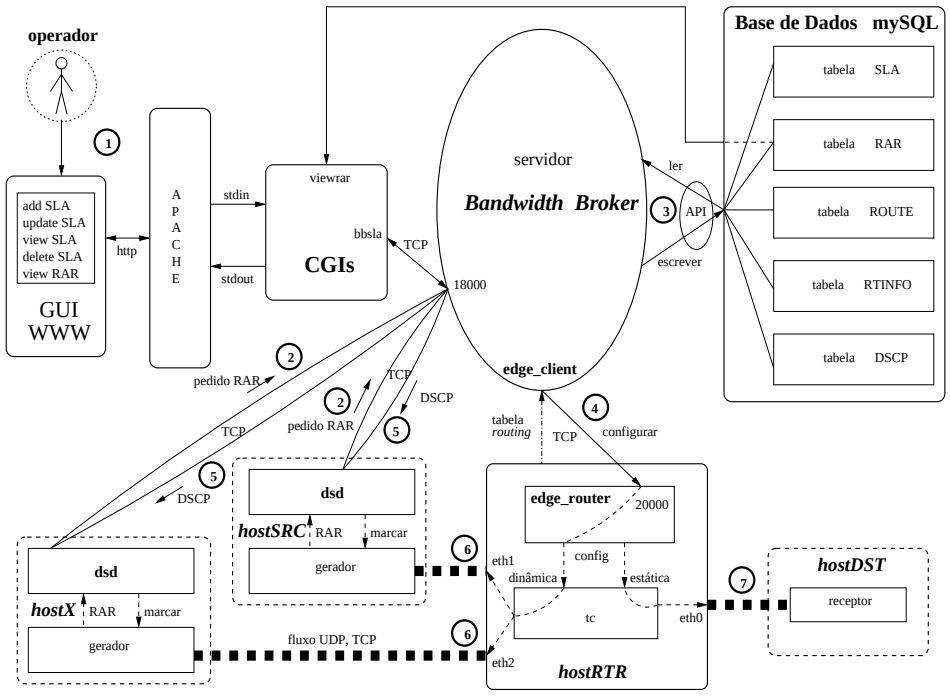


Figura 2 − Interacções globais entre os componentes do domínio

As Figuras 2 e 3 mostram as interacções entre os diversos componentes que constituem a plataforma. A primeira mostra as interacções globais com a sequência numerada das operações e a segunda ilustra−as em maior detalhe. A seguir descreve−se sumariamente a funcionalidade do domínio estabelecido.

1 Este modo significa que as classes CBQ podem exceder o débito que lhes foi atribuído caso se verifiquem as regras de partilha de ligação.
2 No controlo de tráfego Linux, esta é a disciplina mais apropriada para descartar pacotes de uma classe segundo as precedências de rejeição.

Os acordos SLA são efectuados estaticamente, isto é, são adicionados à base de dados do BB pelo administrador (*1*). O SLA é depois mapeado directamente na especificação SLS respectiva. O SLA permite definir os recursos para um serviço requisitado por um cliente. No BB kubb a definição restringe−se apenas à largura de banda, a qual é destinada a um cliente (utilizador, organização) e não a um fluxo particular. Compete depois ao cliente reservar para os seus fluxos individuais porções do recurso global acordado no SLA, através de pedidos RAR efectuados ao BB por meio do agente dsd (*2*). Todas as máquinas pertencentes ao referido cliente deverão aceder ao domínio através de um único *router* folha. É este *router* que o BB vai configurar a partir do RAR recebido.
O BB disponibiliza uma interface gráfica WWW através da qual um operador de rede pode interagir duma forma amigável com a tabela SLA para especificar os requisitos dos clientes. É possível criar, consultar, alterar e remover um SLA ou consultar os RARs existentes. Qualquer uma destas acções, excepto a última, exige interacção com a tabela SLA existente na base de dados. Contudo as CGIs associadas não contactam directamente o servidor mySQL. As CGIs devem comunicar com o servidor do BB, o qual se encarrega de contactar o servidor mySQL. Este devolverá então o resultado da operação à CGI através do servidor do BB. As CGIs comunicam com o servidor do BB através do cliente bbsla. A CGI usada para listar a tabela RAR completa é a única que contacta directamente o servidor mySQL, sem qualquer intervenção do BB já que é uma operação passiva. Esta CGI usa o cliente viewrar para executar a tarefa.
Os parâmetros envolvidos num acordo SLA podem ser divididos em quatro grupos: Identificação do cliente: *Customer ID* (identificação numérica); Tipo de serviço: *Service Type* (EF ou AF); Parâmetros do serviço escolhido: *Rate* (largura de banda máxima agregada), *Units* (unidade especificada para o *Rate*), *Class* (classe pretendida do serviço AF); Validade do contrato de serviço: *Start Date* (mês:dia:ano) e respectivo *Time* (h:min) do início do SLA, *End Date* e respectivo *Time* do término do SLA.



os componentes envolvidos com traço exterior contínuo encontram−se instalados no *router*

Figura 3 − Detalhe das interacções entre os componentes do domínio DiffServ

Recebido um RAR, o BB compara a largura de banda pedida com a diferença entre a largura de banda total disponível no respectivo SLA e a que é consumida pelos RARs aceites (pertencentes ao mesmo SLA), para o período temporal definido no novo RAR (*3*). Caso haja largura de banda disponível, o BB aceita o pedido e regista−o na tabela RAR da base de dados, configura o *router* folha (*4*) para policiar, remarcar e regular o tráfego da aplicação que efectuou o RAR e, após consultar a tabela DSCP, devolve o identificador da classe ao agente dsd (*5*) para marcação do tráfego (*6*). Caso contrário, o BB declinará o pedido RAR. A equação seguinte traduz matematicamente o critério de admissão definido:

$$\left[LB_{\text{rar } p,\Delta T} \leq LB_{\text{total}} - \sum_{i=1}^{n} LB_{\text{rar } i,\Delta T} \right]_{\text{sla } q} \qquad \text{(Eq. 1)}$$

em que p é o novo pedido RAR efectuado ao SLA q para o intervalo de tempo ΔT, e n o número de RARs já aceites para esse período temporal.
Uma vez terminado o RAR, por ter expirado a duração deste ou por vontade própria do cliente, porque terminou a geração de tráfego, a respectiva entrada é retirada da tabela RAR pelo BB.
A tabela RTINFO reflete a topologia do domínio indicando todos os *routers* folha existentes. Após receber um pedido RAR, o BB consulta−a para saber a que *router* folha deve pedir a tabela de encaminhamento. Recebida esta tabela, o BB extrai−lhe a entrada que permite alcançar a rede destino e adiciona−a à tabela ROUTE. A partir desta tabela, o BB determina qual a interface do *router* folha que deve configurar dinamicamente.
Os *routers* folha correm o *daemon* edge_router, ficando à espera de receber os pedidos feitos pelo BB através do cliente edge_client, o qual se encontra embebido no código do servidor do BB.

3.1 *Funcionalidades acrescidas ao BB*

O BB original apenas configura as interfaces egresso dos *routers* folha. Tal estratégia permite policiar o tráfego mas não remarcá−lo, porque os mecanismos de controlo de tráfego do *kernel* Linux não possibilitam realizar ambas funções numa só interface. Assim é necessário que o tráfego seja classificado e policiado na interface ingresso para ser depois eventualmente remarcado na egresso. Interessa portanto ao BB saber qual é a interface ingresso do *router* folha e não a egresso, pois a primeira é configurada dinamicamente de acordo com os RARs efectuados pelos clientes, ao passo que a última é configurada estaticamente quando se arranca o *daemon* edge_router. Este *daemon* recebe os parâmetros passados pelo BB e depois configura a interface ingresso do *router* para classificar e policiar o tráfego emitido pelo cliente. O código do BB foi alterado para permitir esta funcionalidade.
Uma outra limitação identificada no BB original prende−se com o processo de admissões de novos RARs, efectuado segundo a Eq. 1. Tal critério, além de poder ser bastante penalizador para as aplicações que efectuem pedidos RAR, poderá também ser pouco eficiente em termos de recursos aproveitados. A fim de melhorar esta questão, o código do BB foi alterado por forma a permitir que as aplicações possam efectuar pedidos RAR com uma margem de tolerância especificada no próprio pedido. Caso a largura de banda disponível pelo BB esteja dentro da tolerância, o pedido será aceite. Caso contrário, o BB analisa o tipo do RAR e determina o critério a aplicar nesta situação. Consegue−se assim quebrar a rigidez do controlo de admissão original usado pelo BB e melhorar o aproveitamento dos recursos disponíveis.
A semântica do RAR foi assim extendida através da introdução de dois novos campos: TOL e MODE. O campo TOL define, em termos percentuais, a tolerância k admissível para a largura de banda requerida no pedido RAR, sendo este aceite se e só se:

$$\left[(1-k) \cdot LB_{\text{rar } p,\Delta T} \leq LB_{\text{total}} - \sum_{i=1}^{n} LB_{\text{rar } i,\Delta T} \right]_{\text{sla } q} \qquad \text{(Eq. 2)}$$

O campo MODE define o tipo de comportamento que o BB deverá ter face a um novo RAR − pedido *Hard* (H) ou *Soft* (S) − que não possa ser aceite por insuficiência de recursos disponíveis. O modo H indica que o BB deverá tentar entregar à aplicação a largura de banda pedida no RAR dentro da tolerância especificada. Caso contrário, o pedido é rejeitado e não será reservada qualquer largura de banda à aplicação. No modo S, se o pedido não for satisfeito, o BB não reservará largura de banda, mas a aplicação poderá enviar o tráfego como BE. Ou seja,

$$\text{se} \left(\left[(1-k) \cdot LB_{\text{rar } p,\Delta T} > LB_{\text{total}} - \sum_{i=1}^{n} LB_{\text{rar } i,\Delta T} \right]_{\text{sla } q} \right) \qquad \text{(Eq. 3)}$$

então {se [MODE=H] *pedido rejeitado;*
senão se [MODE=S] *pedido aceite como BE;*}

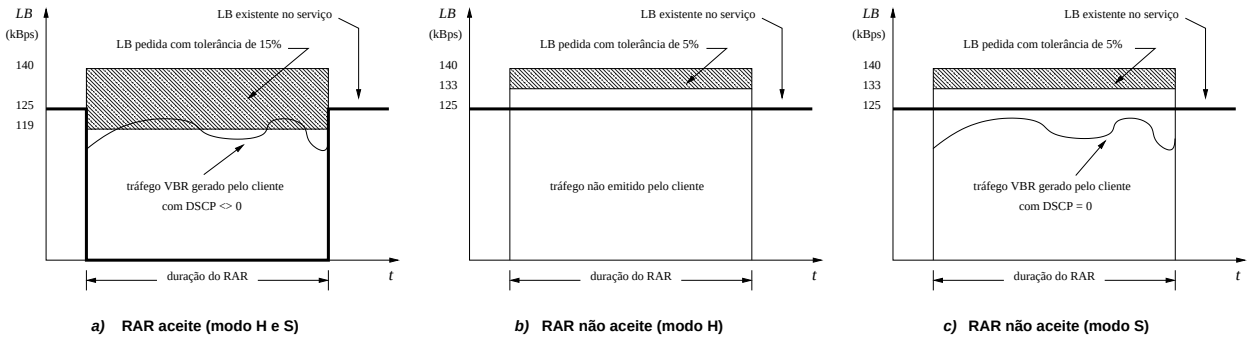


Figura 4 − Ilustração dos modos H e S em pedidos RAR.

A Figura 4 exemplifica graficamente a utilização dos modos H e S em conjunção com a tolerância.

3.2 *Geração de tráfego e sincronização temporal*

Os testes efectuados na plataforma realizaram−se usando tráfego UDP (CBR, Poisson) e TCP.
Para gerador de tráfego TCP optou−se por uma versão modificada do ttcp [Slattery91], designada ttcpds, o qual já vem incluído no pacote de distribuição do BB. O ttcpds é uma aplicação que, para além de possuir todas as características e opções inerentes ao ttcp, permite especificar pedidos RAR.
A fim de se efectuarem análises temporais ao tráfego, o código do ttcpds foi alterado por forma a que cada pacote fosse marcado com uma estampilha temporal na origem e outra na recepção.
Como gerador de tráfego UDP escolheu−se o rude [Laine99], o qual teve que ser modificado para permitir realizar pedidos RAR (no instante da emissão do tráfego) e gerar tráfego Poisson. O crude (receptor de tráfego) foi alterado por forma a garantir a captura de todos os pacotes e incluir o valor TOS no ficheiro de resultados.
Dado ser necessário calcular estatísticas temporais (atraso, *jitter*, etc.) e também porque o BB controla as datas dos fluxos admitidos, torna−se necessário que todas as máquinas estejam bem sincronizadas no tempo. Para tal utilizou−se o protocolo NTP [Mills92].

4. Testes e análise dos resultados

Uma vez estabelecida a plataforma experimental, planeou−se um conjunto de testes com o intuito de se analisar a maior ou menor relevância que a utilização do BB pode desempenhar na manutenção da QoS oferecida por um dado serviço. Os testes também permitirão avaliar a correcta operacionalidade da plataforma e dos serviços nela implantados.
Numa rede DiffServ, o conjunto de fluxos gerados pelas aplicações com requisitos de QoS idênticas é agregado e submetido a uma determinada classe de serviço. Ora da perspectiva do utilizador, o que se pretende é que o tráfego enviado pelas suas aplicações chegue ao destino dentro das especificações requeridas, independentemente do modelo de serviço de rede adoptado. É, pois, interessante analisar o impacto que sofre a QoS de um fluxo individual devido ao processo de agregação a uma classe de serviço EF, já que esta é a que apresenta menor tolerância à degradação da QoS perante as restantes classes normalizadas.
Os parâmetros de QoS analisados serão o atraso médio e máximo, o *jitter* médio e máximo, o débito recebido e a perda de pacotes.

4.1 *Cenário de testes*

A Figura 5 esquematiza o cenário de testes usado. Nele, um cliente pertencente ao domínio necessita de 620 kBps (50%) [3] de largura de banda para enviar tráfego EF [4] através de duas aplicações, cada uma residente numa máquina distinta. Efectuado o respectivo SLA ao BB, e uma vez aceite, o cliente decide dividir equitativamente essa largura de banda pelas referidas aplicações. Para tal, executa dois pedidos

3 Para se ter uma noção melhor das cargas envolvidas, é indicada a percentagem relativa à capacidade máxima da ligação física (1250kBps).
4 O tráfego pertencente à classe EF (i.e., o tráfego agregado marcado com o código do PHB EF) será futuramente referido como tráfego EF. O pertencente às classes AF1 e BE será designado respectivamente por tráfego AF e BE .

RAR ao BB, reservando 310 kBps (25%) para cada aplicação. Suponha−se ainda que o cliente efectuou um SLA especificando 500 kBps (40%) para tráfego AF e que foi aceite pelo BB. Na figura indicam−se também as larguras de banda máximas atribuídas às classes EF, AF e BE.

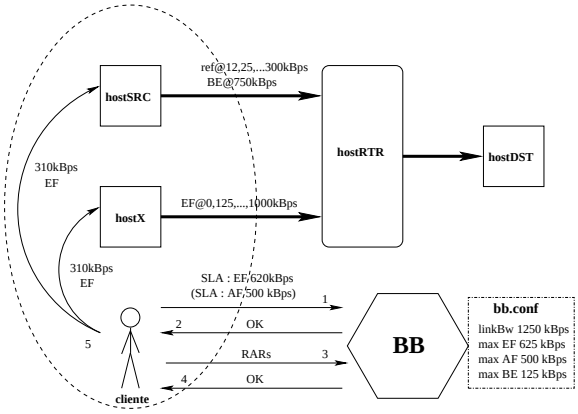


Figura 5 − Representação esquemática do cenário de testes

A Tabela 1 apresenta o conjunto de testes realizados com tráfego UDP e TCP e os objectivos dos mesmos. Os resultados obtidos são analisados seguidamente.

fonte	**DSCP do fluxo ou tráfego referência**	**tráfego agregado presente**			**objectivo do teste**
tráfego UDP					
CBR, Poisson	EF	EF		BE	Impacto da agregação na classe EF perante as situações de concorrência indicadas
		EF	AF	BE	
		EF		TCP BE	
tráfego TCP					
*	EF	EF			Impacto da agregação na classe EF

Tabela 1 − Testes a realizar (com e sem BB).

4.2 *Tráfego UDP EF*

Dois conjuntos de testes com tráfego UDP foram realizados no cenário apresentado. No primeiro, a máquina hostSRC envia um fluxo de referência emitido a uma taxa de transmissão crescente (12, 25, ..., 300 kBps (24%)) e policiado a 310 kBps (25%). Cada fluxo vai ser agregado ao tráfego EF emitido pela máquina hostX, enviado a 125 (10%), 250 (20%), 500 (40%), 750 (60%) e 1000 (80%) kBps, mas sempre policiado a 310 kBps (25%). Os policiamentos referidos são independentes. Está−se, portanto, em presença duma situação em que o domínio é gerido pelo BB. O segundo conjunto de testes é idêntico ao primeiro mas sem a presença do BB, e como tal o tráfego não é sujeito a nenhuma acção de controlo. Em ambos utiliza−se exclusivamente tráfego UDP do tipo CBR ou Poisson, e a máquina hostSRC gera tráfego BE a 750 kBps (60%) não sujeito a policiamento. A Figura 6 apresenta a carga do tráfego entregue às classes da disciplina CBQ do *router*. Repare−se que, numa infra−estrutura de comunicações real, a carga do tráfego EF representa normalmente uma pequena parcela face ao volume de tráfego BE existente. No cenário apresentado optou−se pela presença de cargas de tráfego EF mais significativas a fim de evidenciar melhor o impacto que poderá ocorrer sobre esta classe de serviço.
A Figura 7 apresenta as curvas do atraso e *jitter* médio obtidas com tráfego CBR nos casos em que o domínio é gerido (*a*), e não (*b*), pelo BB. Os gráficos indicam no eixo das abcissas a taxa com que o fluxo de referência foi emitido da máquina hostSRC, e apresentam uma curva referente a cada taxa com que o tráfego EF foi transmitido da máquina hostX: 0, 125, 250, 500, 750 e 1000 kBps. O eixo das

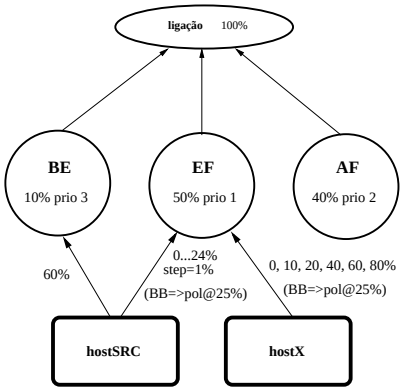
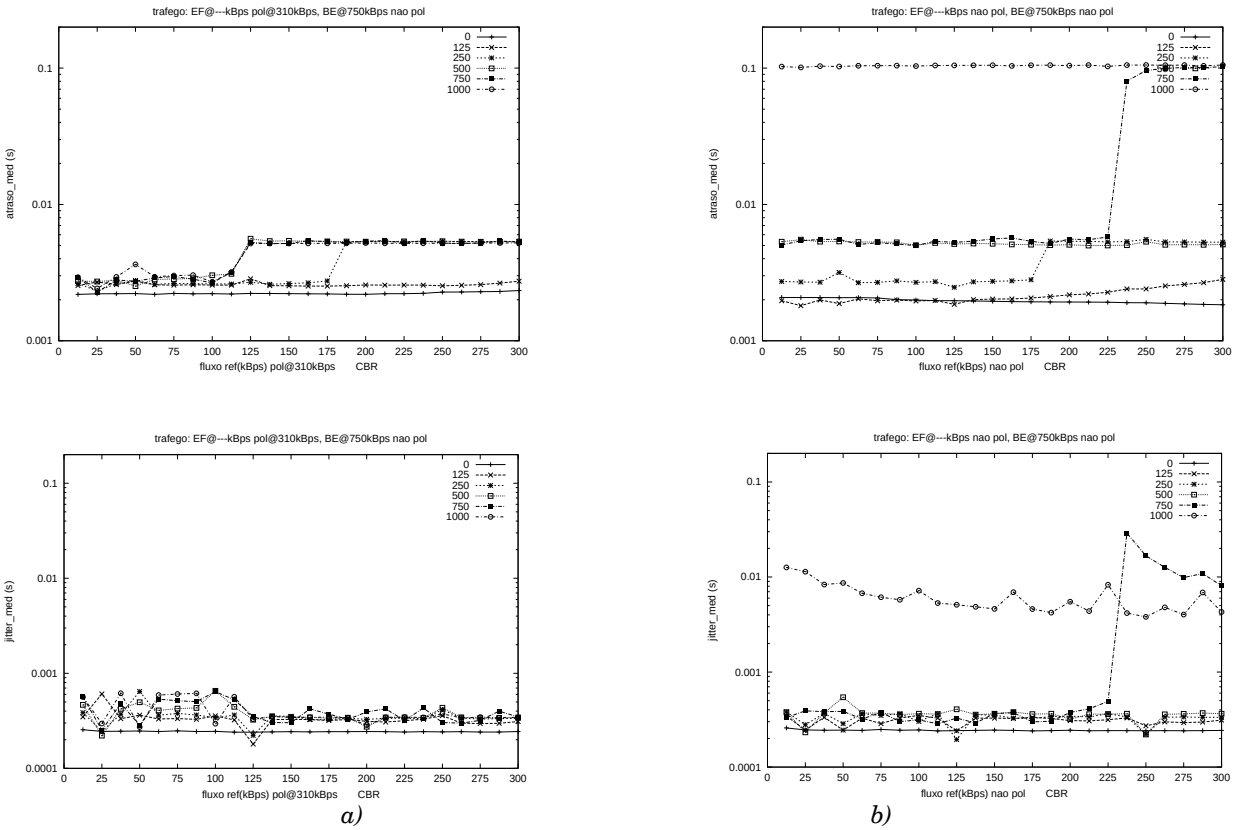


Figura 6 − Tráfego envolvido no teste presente

ordenadas apresenta o atraso/*jitter* médio que o fluxo de referência sofreu.
Observando as curvas 0, 125 (10%) e 250 kBps (20%), não se notam diferenças significativas entre usar ou não o BB, porque quer o fluxo de referência quer o tráfego EF estão em conformidade com os RARs.
Na curva 250 kBps (20%), nota−se que o *router* congestiona [5] quando a máquina hostSRC emite tráfego a uma taxa superior a 175 kBps (14%), ocorrendo então um incremento no atraso médio de cerca 3 ms. Isto deve−se ao facto da fila de espera associada à classe BE crescer até à saturação, tendo−se verificado degradação e perda de pacotes no tráfego BE justamente a partir daquela taxa. A fila de espera da classe EF mantém um mínimo de pacotes por ser de maior prioridade. Porém o escalonador geral, quando atende a classe BE saturada, despacha um número maior de pacotes BE para o *buffer* de transmissão do que quando esta classe não está saturada. Assim, os pacotes EF vão encontrar neste *buffer* mais pacotes BE à espera de serem despachados, e, portanto, sofrem um maior atraso.

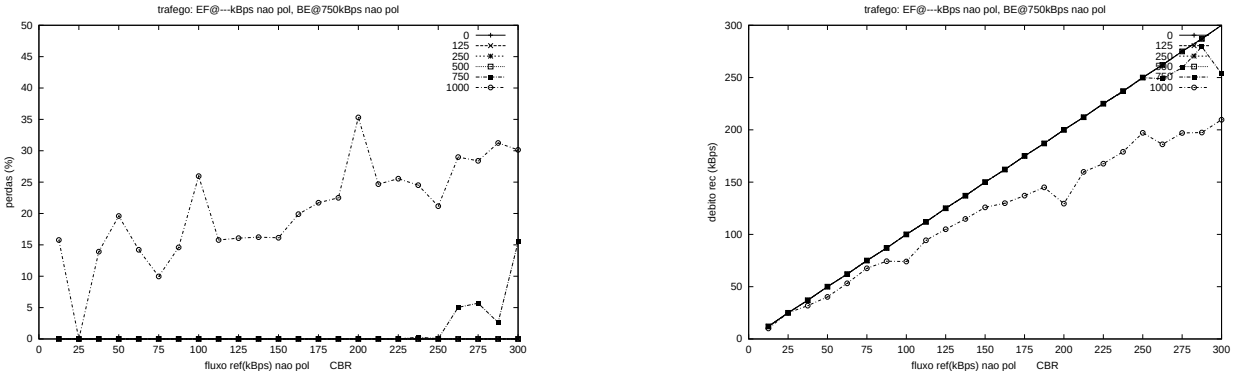


a) *b)*

Figura 7 − Atraso/*jitter* médio obtido *a)* com BB e *b)* sem BB

5 Determinou−se experimentalmente, através do gerador ttcp, que a taxa de transmissão máxima para tráfego UDP é igual a 1175 kBps (94%). Portanto, o *router* congestiona quando a sua carga atinge este valor.

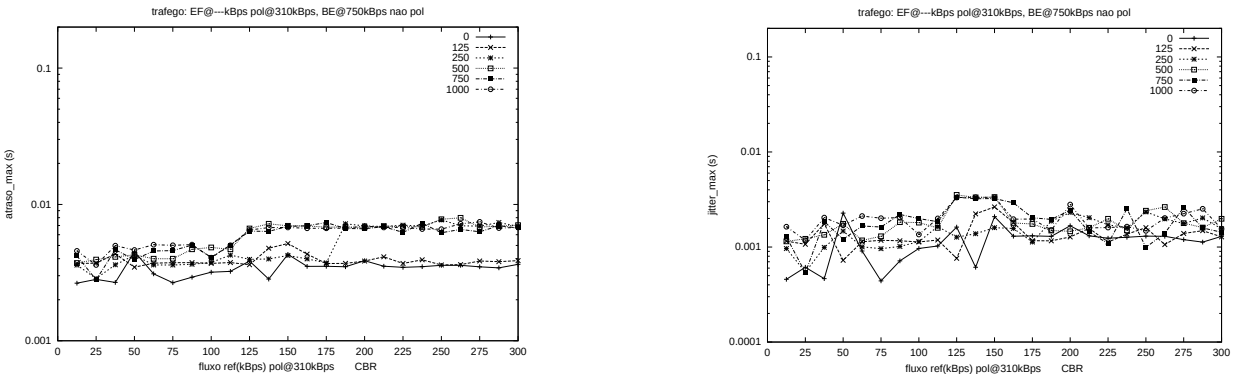
Para o tráfego emitido da máquina hostX a uma taxa igual ou superior a 500 kBps (40%) *sem a presença do BB*, o *router* está sempre congestionado, independentemente da taxa do fluxo de referência gerado. Comparando com a situação em que se usou o BB, nota−se que o fluxo de referência agregado não é muito prejudicado enquanto a carga do tráfego EF no *router* for inferior a aproximadamente 1000 kBps. Isto acontece porque as classes CBQ foram implementadas em modo *borrow*, e portanto os recursos disponíveis nas outras classes vão ser aproveitadas pelas classes activas em sobrecarga. Ora como não existe tráfego AF, o escalonador de partilha de ligação vai permitir que a classe EF utilize totalmente os recursos disponíveis na classe AF. Tal facto traduz−se na inexistência de perda de pacotes para o tráfego EF, como ilustra a Figura 8. Os recursos atribuídos à classe BE são usados totalmente, e como tal indisponibilizados à classe EF. (Sem a presença do BB e com tráfego AF emitido a 400 kBps e tráfego BE gerado a 350 kBps, observou−se que o escalonador de partilha de ligação esgotou a atribuição dos recursos disponíveis à classe EF quando a carga deste tráfego no *router* excedeu 750 kBps.)
O fluxo de referência começa a ser muito afectado quando a carga do tráfego EF no *router* ultrapassa 1000 kBps (80%). Ocorre, então, perda de pacotes e o atraso e *jitter* médio aumentam mais de vinte vezes. Isto deve−se ao facto da fila de espera associada à classe EF começar a crescer até ficar completamente saturada por esgotamento dos recursos não utilizados pela classe AF, originando um atraso elevado, perda de pacotes e decaimento do débito recebido no destino. Com efeito, um pacote que chegue à referida fila só será enviado quando todos aqueles que estiverem à sua frente forem despachados. Assim, o atraso será aproximadamente igual ao tempo necessário para escoar completamente a fila de espera.



a) *b)*

Figura 8 − *a)* perda de pacotes e *b)* débito recebido sem BB

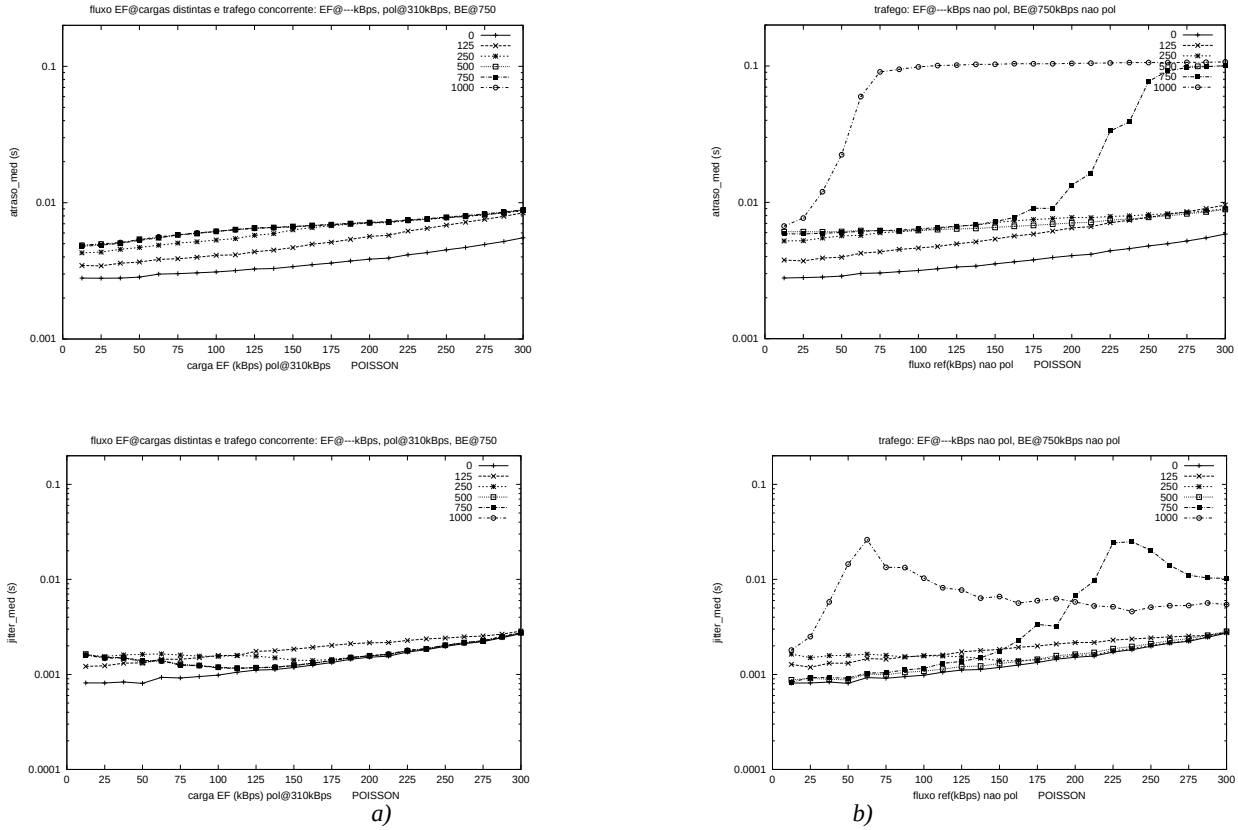
Caso se efectue policiamento ao tráfego por acção do BB, a Figura 9 evidencia que o fluxo de referência apresenta valores para o atraso e *jitter* máximo sempre bem limitados, mesmo que seja agregado a tráfego EF chegado ao *router* em não conformidade (descartado pelo policiador).
Quando o BB não está presente, o atraso e *jitter* máximo atingem valores da ordem de 100 ms quando a carga do tráfego EF no *router* alcança 1000 kBps (80%).



a) *b)*

Figura 9 − *a)* atraso máximo e *b) jitter* máximo com BB

Usando tráfego Poisson verifica−se que as curvas do atraso e *jitter* médio passaram a ter um comportamento curvilíneo ascendente, em vez do comportamento linear evidenciado com tráfego CBR, como mostram as Figuras 10, 11, 12.
Com o *router* não congestionado, nota−se um ligeiro aumento gradual da taxa de variação do atraso à medida que a taxa de transmissão aumenta. A razão de isto acontecer pode ser devido a existir um aumento gradual do número de pacotes na fila de espera da classe EF à medida que a taxa de transmissão sobe, à natureza estatística do tráfego, e/ou aumento gradual do número de pacotes EF acumulados no *buffer* de transmissão porque já existem pacotes BE aguardando a vez de serem transmitidos.
Quando o *router* entra em congestão, a curva do atraso apresenta, não um incremento brusco como acontecia com tráfego CBR, mas um abrandamento no gradiente.

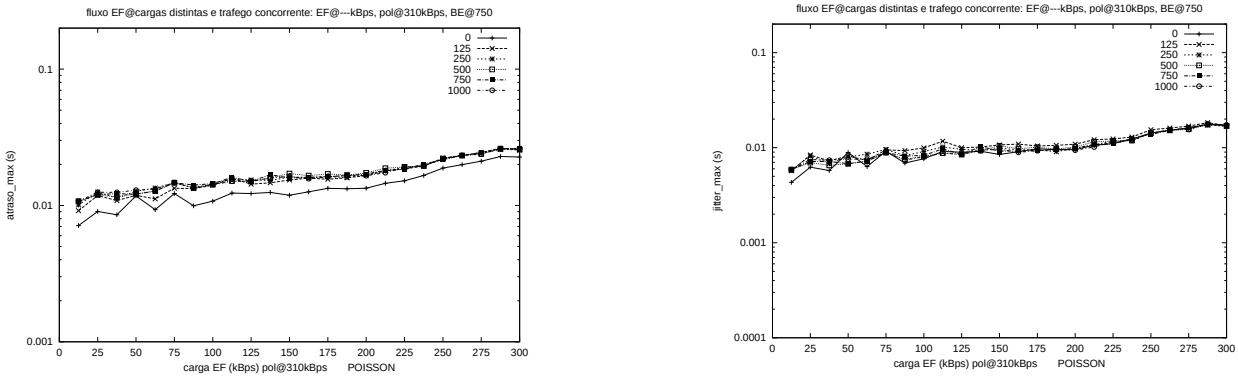


a) *b)*

Figura 10 − Atraso/*jitter* médio obtido *a)* com BB e *b)* sem BB

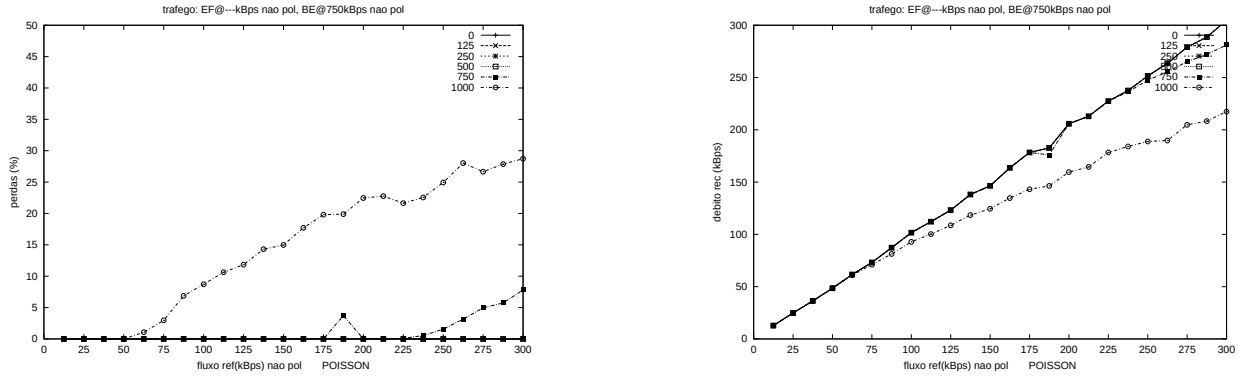
Caso não se utilize o BB, verifica−se que, quando a carga do tráfego EF no *router* ultrapassa 925 kBps (74%), o escalonador de partilha de ligação começa a ter dificuldades em atribuir os recursos não usados pela classe EF em sobrecarga (o tráfego de referência começa a ser afectado, sofrendo perda de pacotes e atraso e *jitter* crescentes), esgotando−os completamente quando a carga total EF recebida atinge a taxa de 1025 kBps (82%). Enquanto a referida carga é inferior a 925 kBps (74%), o fluxo de referência não é muito afectado pela agregação.
Comparativamente ao tráfego CBR, observa−se que o comportamento global é semelhante, mas agora as variações ocorrem mais gradualmente e com um maior período de pré−congestão.
Viu−se como um fluxo de referência é afectado pela agregação de tráfego EF na presença de tráfego UDP BE. Ora a situação mais real é o tráfego BE ser do tipo TCP, e não UDP, pois na rede Internet o tráfego TCP é predominante. É, portanto, conveniente ver o que acontece ao tráfego UDP EF quando se encontra em presença de tráfego TCP BE concorrente.
Para tal determinaram−se o atraso e *jitter* médio *com o BB presente*. A máquina hostSRC envia tráfego TCP BE não sujeito a policiamento e fluxos de referência CBR que vão ser agregados ao tráfego UDP EF gerado pela máquina hostX, sendo estes policiados a 310 kBps (25%). Notou−se uma degradação sempre superior a quatro vezes nos valores obtidos para os diversos parâmetros quando o tráfego BE passou de UDP para TCP. Todas as curvas registaram um atraso médio de cerca 20 ms e um *jitter* médio

de 9 ms, o atraso máximo registado foi sempre superior a 40 ms e o *jitter* máximo foi sempre superior a 20 ms. Isto acontece porque nos intervalos existentes entre os pacotes do tráfego CBR, o fluxo TCP BE aproveita para enviar o maior número de pacotes possíveis, os quais se acumularão no *buffer* de transmissão da interface egresso do *router*. Quando chega a altura de enviar um pacote do tráfego CBR, este vai encontrar alguns pacotes BE ainda por despachar neste *buffer*, e como tal terá que aguardar até que sejam todos despachados, causando−lhe um aumento no atraso. O fluxo de referência só sofreu perda de pacotes e redução do débito recebido quando foi agregado ao tráfego EF emitido a 1000 kBps.
Quando o BB passou a policiar o tráfego BE com uma taxa igual à largura de banda atribuída para esta classe de serviço verificou−se que o tráfego BE deixou praticamente de influenciar o tráfego EF. Em todas as condições de teste registaram−se um atraso médio de cerca 2 ms, um atraso máximo inferior a 10 ms, um *jitter* médio inferior a 1 ms e um *jitter* máximo inferior a 8 ms. Estes resultados mantiveram−se válidos usando tráfego EF do tipo Poisson.



a) *b)*

Figura 11 − *a)* atraso máximo e *b) jitter* máximo com BB



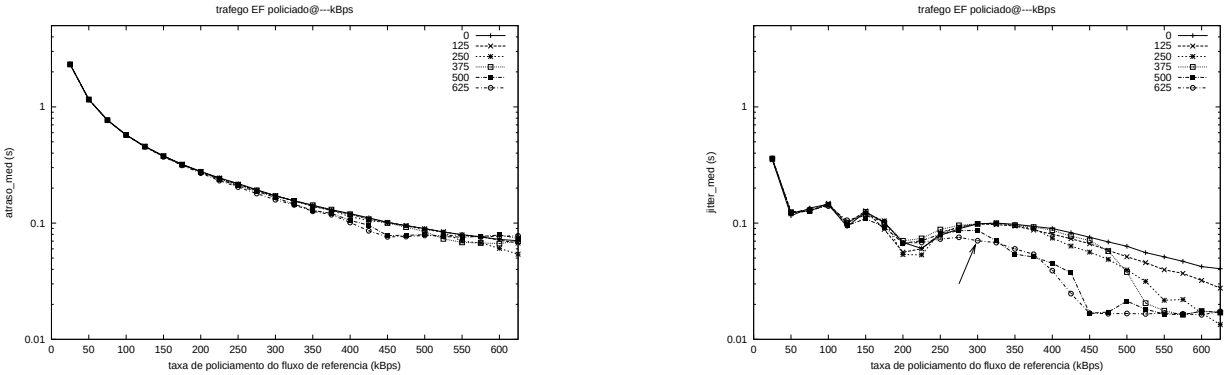
a) *b)*

Figura 12 − *a)* perda de pacotes e *b)* débito recebido sem BB

4.3 *Tráfego TCP EF*

Em virtude do protocolo TCP tentar sempre adaptar−se aos recursos de rede existentes, deixa de fazer sentido falar−se em fluxos TCP do tipo CBR ou Poisson. Como tal, o modelo de testes aplicado ao tráfego TCP terá necessariamente que ser distinto do usado para tráfego UDP. Assim, foi planeado um conjunto de testes com o intuito de se estudar como é que o policiamento imposto pelo BB afecta a QoS de um determinado tráfego TCP.
Suponha−se que um cliente especifica ao administrador do domínio um SLA requisitando toda a largura de banda disponível para tráfego EF, ou seja, 1250 kBps (100%). Apesar de algo irrealista, esta situação é sustentável porque na plataforma irá circular apenas tráfego EF e, portanto, o escalonador de partilha de ligação da disciplina CBQ atribuirá à classe EF os recursos das restantes classes. Aceite o pedido, a

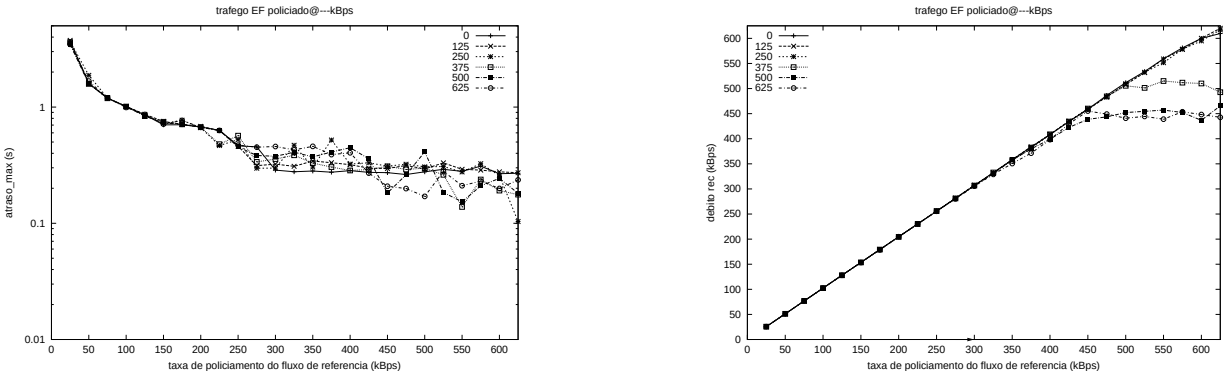
máquina hostSRC gera um fluxo de referência que irá ser agregado ao tráfego EF emitido pela máquina hostX. Contudo este fluxo deve primeiramente efectuar um RAR ao BB, pedindo uma porção de largura de banda ao SLA respectivo. Pretende−se estudar o impacto na QoS do fluxo de referência agregado ao tráfego EF em função dos pedidos RAR efectuados ao BB.
As Figuras 13 e 14, mostrando os resultados obtidos, devem ser interpretados duma forma completamente distinta dos apresentados para tráfego UDP. No eixo das abcissas são apresentadas as taxas de policiamento a que foi sujeito o fluxo de referência. O eixo das ordenadas apresenta os valores dos parâmetros de QoS com que este fluxo chegou ao destino. Existe uma curva para cada taxa de policiamento a que foi sujeito o tráfego EF. Estas taxas são 0, 125 (10%), 250 (20%), 375 (30%), 500 (40%) e 625 (50%) kBps. Por exemplo, o ponto assinalado com uma seta, no gráfico do *jitter* médio (Figura 13), significa que quando a máquina hostSRC envia um fluxo de referência policiado a 300 kBps (24%) e a máquina hostX envia tráfego EF policiado a 625 kBps (50%), após a agregação o fluxo de referência chega ao destino com um *jitter* médio de cerca 70 ms.



a) *b)*

Figura 13 − *a)* atraso médio e *b) jitter* médio

Observando os gráficos apresentados, a primeira evidência são os elevados valores temporais registados. Verifica−se também que os atrasos médio e máximo vão diminuindo à medida que o policiamento vai sendo feito a taxas mais elevadas. Ou seja, quando maior for a largura de banda oferecida ao tráfego TCP EF, menor é o grau de ocupação da fila de espera associada à classe EF. O gráfico do *jitter* máximo obtido é semelhante ao gráfico do atraso máximo apresentado.
Como já foi referido, usando a ferramenta ttcp determinou−se que a taxa de transmissão máxima de datagramas IP que é possível ter numa ligação UDP é 1175 kBps (94%). Quando se envia tráfego TCP verifica−se que esta taxa desce para cerca de 830 kBps (66%). Tal facto não é de estranhar se se tiver em conta que o protocolo TCP é fiável e orientado à conexão. Como tal, utiliza mecanismos de controlo de erros e de fluxo, os quais farão com que a taxa de transmissão máxima seja inferior à obtida usando tráfego UDP. Daqui a razão de não ser apresentado o gráfico relativo à perda, pois esta será sempre nula usando o protocolo TCP.



a) *b)*

Figura 14 − *a)* atraso máximo e *b)* débito recebido

Analisando o gráfico referente ao débito recebido, verifica−se que, enquanto a soma das taxas de policiamento (e portanto da taxa a que o tráfego agregado é emitido através da interface egresso do *router*) for inferior a 875 kBps (70%), o fluxo de referência chega ao destino à taxa imposta pelo policiamento. A partir desse valor a ligação TCP satura e o fluxo de referência chega ao destino a uma taxa limitada a 500 kBps (40%), apesar do policiamento poder ser feito a taxas superiores. Repare−se que, nesta situação, o fluxo de referência é afectado pela agregação ao tráfego EF, pois o policiador permite que este fluxo seja transmitido acima de 500 kBps (40%), mas tal não acontece porque a carga do tráfego EF não o permite.
A Tabela 2 mostra os valores obtidos quando o tráfego gerado pelas máquinas não é policiado pelo BB.

	atraso_med	**atraso_max**	***jitter*_med**	***jitter*_max**	**débito_rec**
fluxo ref.	0.076572 s	0.129099 s	0.016692 s	0.074563 s	451.6 kBps (36%)
tráfego EF	0.069736 s	0.186360 s	0.021079 s	0.116624 s	443.6 kBps (35%)

Tabela 5.2 − Resultados obtidos sem BB.

5. Conclusões

Neste artigo foi descrito o estabelecimento de uma plataforma experimental DiffServ e o conjunto de testes realizados que permitiram avaliar a sua operacionalidade, e estudar o papel do BB sobre o impacto da agregação na QoS de um fluxo de referência EF, sob a presença de tráfego BE (e AF).
Conclui−se que na presença do BB, o comportamento de um fluxo de referência mantém−se controlado e constante quando é agregado a tráfego UDP EF chegado ao *router* em não conformidade. Sem a presença do BB, um fluxo de referência é pouco afectado pela agregação a tráfego UDP EF mal comportado se o *router* não estiver congestionado e/ou se o escalonador de partilha de ligação encontrar recursos livres nas restantes classes para atribuir à classe EF. Senão, o referido fluxo poderá ser penalizado, tanto mais quanto menos recursos livres existirem. Também se observou que o tráfego EF pode ser bastante afectado caso o BB não policie o tráfego TCP BE a uma taxa igual à largura de banda atribuída para esta classe de serviço.
Na agregação a tráfego TCP EF, verificou−se que o BB garante uma dada largura de banda a um fluxo de referência, mas sem grandes exigências em termos de atraso e *jitter*. Para tal, o BB deve apenas admitir nos SLAs uma largura de banda igual à taxa de transmissão máxima que a ligação possibilita ao tráfego TCP.

Referências

[Blake98] S. Blake *et al.*, "*An architecture for Differenciated Services*", RFC 2475, IETF, Dez. 1998.
[Davie02] B. Davie *et al.*, "*An Expedited Forwarding PHB*", RFC 3246, IETF, Março 2002.
[Elek00] V. Elek *et al.*, "*Admission control based on end−to−end measurements*", *Proceedings of* IEEE INFOCOM 2000, pp. 623−630, Março 2000.
[Floyd95] S. Floyd, V. Jacobson, "*Link−sharing and resource management models for packet networks*", IEEE/ACM *Transactions on Networking*, vol. 3 nº 4, pp. 365−386, Agosto 1995.
[Heinanen99] J. Heinanen, *et al.*, "*Assured Forwarding PHB Group*", RFC 2597, IETF, Junho 1999.
[Jacobson99] V. Jacobson *et al.*, "*A two−bit Differentiated Services architecture for the Internet*", RFC 2638, IETF, Julho 1999.
[Laine99] J. Laine, S. Saaristo, R. Prior, "rude & crude", <http://rude/sourceforge/net>, 1999.
[Mills92] D. L. Mills, "*Network Time Protocol (version 3), specification, implementation and analysis*", RFC 1305, IETF, Março 1992.
[Rao99] K.D. Rao, D. Sreekantan, "*Differentiated Services − Implementation of a BB for resource management in DiffServ*", ITTC − *University of Kansas*, <http://www.ittc.ku.edu/~kdrao/845>, 1999.
[Slattery91] T. Slattery, M. Muuss, "ttcp: *test* TCP", <ftp://ftp.sgi.com/sgi/src/ttcp>, Outubro 1991.
[Zhang02] L. Zhang, Y. T. Hou, "*On scalable network resource management using Bandwidth Brokers*", *Proceedings of the Network Operations and Management Symposium*, Abril 2002.