

Arquitectura Escalável para o Suporte de QoS em Redes IP

Rui Prior¹, Susana Sargento²

¹LIACC — Universidade do Porto

²Instituto de Telecomunicações — Universidade de Aveiro

16 de Setembro de 2004

Resumo

Este artigo descreve a arquitectura SRBQ (Scalable Reservation-Based QoS), desenvolvida para o suporte de qualidade de serviço (QoS) extremo-a-extremo, e analisa o seu desempenho em termos de garantias de QoS e de escalabilidade. Esta arquitectura baseia-se num modelo de sinalização por fluxo escalável, usando diversas técnicas e algoritmos desenvolvidos com vista a minimizar a complexidade computacional, nomeadamente um mecanismo de comutação de etiquetas e uma implementação escalável de temporizadores. A arquitectura subjacente é baseada no modelo DiffServ, reservando recursos para agregados de fluxos nas redes de *core* e de acesso. Os resultados de simulação apresentados indicam que a arquitectura proposta suporta os dois modelos de serviço IntServ de forma escalável para redes de elevado débito através da minimização da carga de processamento em cada nó.

1 Introdução

Actualmente a Internet apenas suporta um serviço tipo melhor esforço (*best effort* — BE). Tradicionalmente utilizado no simples transporte de dados, este serviço é inadequado para suportar aplicações com requisitos de tempo real, nomeadamente de comunicações multimédia, actualmente em grande expansão.

Com o objectivo de introduzir Qualidade de Serviço (QoS) e diferenciação de serviços na Internet, o organismo de normalização IETF propôs duas arquitecturas. A arquitectura de Serviços Integrados (IntServ) [8] baseia-se em reservas por fluxo, efectuadas com recurso ao protocolo RSVP (*Resource ReSerVation Protocol*) [9], disponibilizando garantias estritas de QoS e uma utilização eficiente dos recursos. Esta arquitectura sofre, no entanto, de diversos problemas de escalabilidade, relacionados com a classificação e escalonamento de pacotes e com o estabelecimento de reservas por fluxo. A arquitectura de Serviços Diferenciados (DiffServ) [6] não sofre destes problemas de escalabilidade: nela os fluxos são agregados em classes de tráfego com características específicas, usando mecanismos simples e eficientes para a classificação e o escalonamento de pacotes. No entanto, a inexistência de mecanismos de controlo de admissão pode conduzir à degradação da QoS recebida pelos fluxos devido a excesso de tráfego nas ligações.

Visando colmatar os problemas acima mencionados, inúmeras propostas de arquitecturas de QoS têm surgido na literatura. Todas elas, no entanto, sofrem de um ou mais dos seguintes problemas: falta de suporte para garantias estritas de QoS, subutilização dos recursos de rede, limitações de escalabilidade devidas à complexidade dos algoritmos e mecanismos utilizados. A arquitectura SCORE (*Stateless Core*) [14], por exemplo, evita a manutenção de estado no interior da rede através do transporte de informação de estado no cabeçalho dos próprios pacotes, mas impõe o uso em todos os *routers* de uma disciplina de escalonamento cuja complexidade computacional é elevada. Na arquitectura *Egress Admission Control* [11], apenas os *routers* de saída efectuam controlo de admissão, com base em monitorização passiva; este modelo é, no entanto, incapaz de suportar garantias estritas de QoS. Os modelos baseados em *probing* [10, 13] dispensam o controlo por parte da rede, mas não disponibilizam garantias firmes de QoS, além de outros problemas. Em [5] é proposto um modelo para o suporte de IntServ em redes DiffServ, que dispensa o processamento da sinalização no interior do domínio; porém, este modelo sofre de uma atribuição sub-óptima de recursos e de um controlo de admissão impreciso. O modelo RSVP *Reservation Aggregation* (RSVPRAg) [4] baseia-se na agregação de reservas no interior dos domínios de *core* com vista a tornar-se escalável, mas implica um compromisso entre a utilização dos recursos da rede e a escalabilidade a nível de sinalização, conduzindo a uma sub-utilização dos recursos nas redes de elevado débito.

Em [12] foi por nós proposto um novo modelo, denominado Scalable Reservation-Based QoS (SRBQ), que disponibiliza suporte de QoS extremo-a-extremo sem os problemas das arquitecturas acima mencionadas. Este modelo não impõe o uso de disciplinas de escalonamento complexas, usa um mecanismo simples de classificação de pacotes baseado em agregados de tráfego, suporta classes de serviço com garantias suaves e estritas de QoS e permite uma boa utilização dos recursos de rede. Diversos mecanismos e algoritmos foram desenvolvidos com vista a aumentar a escalabilidade do modelo, nomeadamente um mecanismo de comutação de etiquetas que evita a procura complexa da informação do fluxo na tabela de reservas e uma implementação escalável dos temporizadores de expiração de reservas.

Este artigo contém uma descrição geral do modelo SRBQ, focando a arquitectura-base, o protocolo de sinalização e os mecanismos desenvolvidos com vista à melhoria da escalabilidade, bem como uma avaliação do modelo baseada em

diversas simulações, efectuadas recorrendo a uma implementação do modelo por nós efectuada no simulador ns-2. Os parâmetros usuais de qualidade de serviço (atraso, *jitter* e perdas) são analisados em diversos conjuntos de simulações, usando tanto fluxos “sintéticos” (com parâmetros bem definidos) como fluxos multimédia reais. Em termos de garantias de QoS, demonstra-se que as classes de serviço disponibilizadas pelo modelo SRBQ dão a cada fluxo garantias estritas e suaves de QoS, independentemente de outros fluxos na mesma e em diferentes classes. O comportamento da arquitectura em relação aos fluxos reais indicia o bom desempenho do modelo numa utilização prática.

A avaliação efectuada inclui uma comparação com o modelo de agregação de reservas RSVP (RSVPRAgg), igualmente implementado por nós no ns-2. Ambas as arquitecturas são escaláveis e apropriadas para o uso em redes de débito elevado e fica demonstrado que a qualidade de serviço fornecida pelo SRBQ é comparável, e em vários aspectos superior, à fornecida pelo RSVPRAgg, e com uma utilização de recursos substancialmente melhor.

O artigo está organizado da forma a seguir descrita. A secção 2 descreve a arquitectura SRBQ. Além da descrição geral do sistema são apresentados diversos aspectos particulares, nomeadamente o escalonamento, a formatação de tráfego e o policiamento, a comutação de etiquetas, o protocolo de sinalização e os temporizadores de reserva volátil. A secção 3 contém uma análise dos resultados obtidos por simulação do SRBQ em diferentes experiências. Na secção 4 é feita uma comparação com o modelo RSVPRAgg. Por fim, a secção 5 apresenta as principais conclusões, bem como alguns tópicos para trabalho futuro.

2 A Arquitectura SRBQ

A arquitectura SRBQ combina as garantias extremo-a-extremo, em termos de limites do atraso e perdas introduzidas pela rede, inerentes a uma aproximação baseada em sinalização para estabelecimento de reservas por fluxo sujeitas a controlo de admissão, com a eficiência e escalabilidade possibilitadas pela agregação de fluxos e por uma série de mecanismos e algoritmos desenvolvidos. As próximas subsecções descrevem diversos aspectos da arquitectura; uma descrição detalhada pode encontrar-se em [12].

2.1 Arquitectura Geral do Sistema

A arquitectura subjacente ao modelo proposto baseia-se no modelo DiffServ, podendo ambos os modelos coexistir numa mesma rede. Sobre esta base existe um sistema de reservas estabelecidas por meio de sinalização e sujeitas a controlo de admissão. A rede divide-se em domínios, consistindo em nós interiores (*core*) e de fronteira; os domínios de acesso incluem ainda nós de acesso. Os fluxos individuais são agregados segundo classes de tráfego, mapeadas em *Per-Hop Behaviours* (PHB) compatíveis com o modelo DiffServ, e a classificação de pacotes é feita por agregado com base no campo DSCP (*DiffServ Code Point*) do cabeçalho dos pacotes.

Além do serviço BE, o SRBQ suporta duas outras classes: a classe de Serviço Garantido (GS), caracterizada por garantias rígidas em termos de entrega de pacotes e pela limitação do atraso introduzido, que se baseia em princípios semelhantes ao PHB EF (*Expedited Forwarding*) do modelo DiffServ, e uma ou mais classes de Carga Controlada (CL), que emulam o comportamento de redes BE com pouca carga, baseada nos princípios do PHB AF (*Assured Forwarding*).

As reservas são unidireccionais, iniciadas pelo emissor e de estado volátil, sendo estabelecidas com recurso a um protocolo de sinalização. O protocolo funciona nó a nó, usando rotas previamente estabelecidas por um protocolo de encaminhamento, e foi concebido por forma a ser tão simples e eficiente quanto possível, pois é processado em todos os nós ao longo do percurso, incluindo nós interiores. A caracterização das reservas para fluxos da classe GS é de tipo *token bucket*. As reservas da classe CL são caracterizadas por 3 marcas de água (*water-marks*) de débito médio: pacotes que excedam as duas primeiras marcas vêem aumentada a sua probabilidade de perda; pacotes que excedam a terceira marca são eliminados.

O controlo de admissão, descrito detalhadamente em [12], efectua-se em todos os nós ao longo do percurso do fluxo. Um fluxo GS i , caracterizado pelo *token bucket* (r_i, b_i) é admitido numa ligação em que j fluxos GS estão activos se $r_i + \sum_j r_j \leq R_{GSmax}$ e $b_i + \sum_j b_j \leq B_{GSmax}$, em que R_{GSmax} e B_{GSmax} são, respectivamente, a largura de banda e o espaço na fila de espera atribuídos à classe GS. Um fluxo CL i é admitido se $r_{i,w} + \sum_j r_{j,w} \leq R_{max,w}, \forall w$, em que $R_{max,w}$ é o débito máximo configurado para a marca de água w . O controlo de admissão na classe GS é necessariamente do tipo PBAC (*Parameter-Based Admission Control*), ao passo que na classe CL pode ser do tipo PBAC ou MBAC (*Measurement-Based Admission Control*).

2.2 Escalonamento e filas de espera

O modelo mais simples de escalonamento das/nas filas de espera ilustra-se na figura 1.a, devendo existir em todos os nós da arquitectura. O escalonador principal é de prioridade estrita, com pelo menos 4 filas distintas. A fila de prioridade mais elevada é usada para o tráfego da classe GS, altamente sensível ao atraso. Interiormente à classe, os pacotes são servidos segundo uma disciplina FIFO (*First In First Out*). A prioridade imediatamente abaixo é usada para tráfego prioritário não sujeito a controlo de admissão, e é usada para sinalização, protocolos de encaminhamento e tráfego similar (Sig.). Na prioridade seguinte encontra-se a fila da classe CL, contendo 3 filas virtuais (VQ), uma para cada valor de precedência de

perda. A classe CL é servida internamente por uma disciplina de serviço da família GRED (*Generalized Random Early Detection*). O modelo suporta até um máximo de 4 classes de Carga Controlada, usando valores de DSCP de classes AF diferentes (desde que não sejam utilizadas pelo DiffServ, se os modelos coexistirem na mesma rede); neste caso, o bloco CL da figura 1.a substitui-se pelo ilustrado na figura 1.b, usando-se um algoritmo como o DWRR (*Deficit Weighted Round Robin*) para o escalonamento entre as diferentes filas da classe CL. Por fim, a fila com prioridade mais baixa destina-se ao tráfego BE, sendo servida internamente segundo uma disciplina FIFO, RED ou qualquer outra apropriada.



Figura 1: Modelo de escalonamento e filas de espera

2.3 Formatação de tráfego

Como se pode observar na figura 1.a, a classe de tráfego GS está sujeita a um formatador (*shaper*) de tráfego de tipo *token bucket*. Este formatador é parametrizado com os valores somados dos *token buckets* das reservas GS activas. Esta formatação é necessária para evitar a eliminação de pacotes por policiamento devido a *jitter* introduzido pela multiplexagem e desmultiplexagem de fluxos, e não diminui as garantias em relação ao atraso, uma vez que é efectuada com base no limite superior do tráfego do agregado [7].

Apesar de não estar sujeito a controlo de admissão, o tráfego de sinalização é também formatado por forma a evitar a inanição da classe CL. A formatação de tráfego na classe CL é opcional, sendo apenas necessária se o administrador da rede quiser garantir sempre algum débito para o tráfego BE. Ao contrário do formatador da classe GS, estes dois últimos são simples limitadores de débito *work-conserving*, estando apenas activos se as filas de classes inferiores tiverem pacotes a aguardar transmissão.

Além dos já referidos, existe ainda um outro tipo de formatador de tráfego no SRBQ. Nem todos os terminais são capazes de efectuar um escalonamento preciso dos fluxos GS segundo o *token bucket* reservado, e ainda que fossem seria inevitável a introdução de *jitter* pela multiplexagem de diferentes fluxos GS ou de fluxos GS com outros de diferentes classes, ou ainda pelo simples acesso ao meio. Este *jitter* conduziria a perdas no módulo de policiamento nos *routers* de acesso. Por este motivo, os fluxos GS são formatados, e não policiados, à entrada dos *routers* de acesso, garantindo a sua conformidade às reservas sem introdução de perdas devidas a *jitter*.

2.4 Policiamento

Ao contrário da classe GS, os fluxos CL são policiados à entrada dos *routers* de acesso. Existem duas razões para que isto aconteça: por um lado, a caracterização dos fluxos CL é mais permissiva em termos de *jitter*, pois baseia-se em marcas de água correspondentes a valores médios de débito; por outro lado a classe CL não dá garantias estritas de QoS, podendo ocorrer perdas desde que em valores pequenos. Este policiamento garante a conformidade dos fluxos CL às reservas, evitando a degradação de outros fluxos da mesma classe.

Os nós de fronteira também efectuem policiamento das classes GS e CL em cada interface de entrada, mas apenas por agregado (com base nos valores de DSCP). O policiamento por fluxo é desnecessário, pois já foi efectuada pelos *routers* de acesso. Com base no policiamento, os *routers* de fronteira fazem também remarcação de probabilidade de perdas na classe CL. Uma vez que é efectuada por agregado, esta remarcação deve ser *color-aware*, isto é, nenhum pacote pode ser remarcado para uma probabilidade de perdas menor, por forma a evitar a degradação injusta de fluxos CL “bem comportados”. As 3 marcas de água para policiamento de agregados são obtidas somando as marcas de água correspondentes dos fluxos CL individuais.

Nos nós interiores não é efectuada policiamento, uma vez que os *routers* de fronteira já o fizeram à entrada do domínio, garantindo assim a conformidade às reservas.

2.5 Comutação de etiquetas

Associados à manutenção de reservas por fluxo existem dois factores potencialmente limitadores da escalabilidade: a manutenção de informação de reservas e a complexidade do processamento. Em relação ao primeiro, se cada estrutura de informação de reserva ocupar 100 bytes, um *router* que suporte cem mil fluxos em simultâneo necessita de 10 Mbytes para

armazenar essa informação, quantidade de memória existente em praticamente todos os *routers* actuais. Assim, apenas o segundo factor pode verdadeiramente limitar a escalabilidade.

A procura numa tabela da informação armazenada sobre um fluxo, com base nos 5 campos que o identificam, é uma tarefa limitativa, particularmente em termos de pior caso. Esta tarefa é, normalmente, implementada recorrendo a tabelas de *hash*. No entanto, *hashing* perfeito para esses campos exigiria uma quantidade absurda de memória. Para permitir um acesso eficiente às estruturas de informação de reservas, desenvolveu-se um mecanismo de comutação de etiquetas que permite o acesso directo a essas estruturas sem necessidade de procura em tabelas de *hash*. As etiquetas são valores de 32 *bits* cujo significado é externamente opaco, mas que internamente podem ser índices de uma tabela ou os próprios endereços de memória das estruturas contendo a informação sobre as reservas. Cada reserva tem associadas três etiquetas: B, T e F. A etiqueta T (*this*), que pode ser implícita, é a etiqueta usada no próprio nó, mapeada directamente no endereço da estrutura da reserva; a etiqueta B (*backward*) é usada em mensagens para montante, e corresponde à etiqueta T do nó anterior; a etiqueta F (*forward*) corresponde à T do salto seguinte, sendo usada em mensagens para jusante.

As etiquetas são instaladas durante a iniciação da reserva e incluídas em todas as mensagens subsequentes. Como se verá, apenas a primeira mensagem não faz uso da etiqueta. No entanto, a reserva ainda não existe quando ela é processada, pelo que será atribuída uma nova estrutura para guardar a sua informação. Esta atribuição é trivial, consistindo em retirar o primeiro elemento de uma lista ligada.

O mecanismo de comutação de etiquetas pode igualmente ser vantajoso nas tarefas efectuadas por fluxo, tais como o policiamento nos *routers* de acesso. Pode igualmente ser usado na detecção eficiente de alterações de rota: uma diferença entre o salto seguinte nas tabelas de reserva e de encaminhamento significa que a rota foi alterada. Para obter estes benefícios, no entanto, seria necessário incluir a etiqueta em todos os pacotes. Em [12] são apresentadas propostas para a inclusão das etiquetas nos cabeçalhos dos pacotes, tanto em IPv4 como em IPv6. Note-se que, apesar destas vantagens, as etiquetas não são usadas na classificação de pacotes (embora o pudessem ser nos *routers* de acesso), que é efectuada com base em agregados, usando apenas o campo DS do cabeçalho IP.

Embora o uso de etiquetas com um protocolo de sinalização de estado volátil possa sugerir alguma semelhança com o MPLS (MultiProtocol Label Switching) com RSVP-TE (RSVP-Traffic Engineering) [3], na realidade trata-se de modelos muito diferentes. De facto, o modelo SRBQ é puramente de nível 3. A finalidade principal das etiquetas é otimizar o processamento da sinalização para torná-lo escalável, garantindo eficiência mesmo em termos de pior caso. Mesmo na versão estendida do protocolo, em que as etiquetas são utilizadas nos pacotes de dados, existe uma reutilização de campos do cabeçalho IP, evitando o *overhead* introduzido pelo cabeçalho de “nível 2.5” (*shim header*) utilizado no MPLS. Em termos de escalonamento de pacotes, o SRBQ aproxima-se muito mais do modelo *DiffServ*, utilizando apenas uma classe no escalonador por cada classe de tráfego na interface de saída, ao invés do MPLS, em que cada LSP (Label Switched Path) estabelecido usa uma classe de diferente no escalonador. Além disso, para a sinalização ser escalável os LSP funcionam como túneis de tráfego agregado cuja largura de banda é actualizada com uma frequência muito inferior à taxa de iniciação e terminação de fluxos individuais, conduzindo à sub-utilização de recursos de rede, à semelhança do que acontece no RSVPRAg (ver secção 4). Por fim, enquanto no MPLS existe, em princípio, classificação multi-campo à entrada de cada domínio, no SRBQ essa classificação é mais simples, baseando-se apenas na classe de tráfego, indicada pelo campo DS.

2.6 Protocolo de sinalização

Todos os nós na arquitectura SRBQ processam sinalização de reservas. O protocolo de sinalização funciona salto a salto, disponibilizando reservas unidireccionais, de estado volátil (*soft-state*) e iniciadas pelo emissor. O protocolo foi implementado como extensão ao RSVP por diversas razões: o RSVP é um protocolo bem conhecido e amplamente disponível; funciona igualmente nó a nó; também usa rotas previamente estabelecidas por um protocolo de encaminhamento; e o RSVP já implementa alguma da funcionalidade necessária.

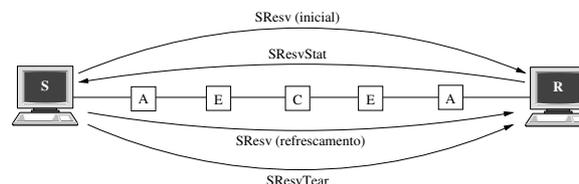


Figura 2: Mensagens de sinalização SRBQ

Apesar de implementado como uma extensão ao RSVP, este protocolo é muito mais escalável, pois (1) o acesso à informação do fluxo é directo, usando as etiquetas, (2) os temporizadores para expiração das reservas voláteis são implementados de forma muito eficiente e (3) usa uma identificação simples das reservas por forma a reduzir o tamanho e o tempo de processamento das mensagens de refrescamento e terminação de reservas. A extensão ao RSVP inclui três novos tipos de mensagem (figura 2): SRResv (Sender Reservation), usada para estabelecer, refrescar e alterar reservas

iniciadas pelo emissor; SResvStat (Sender Reservation Status), usada para confirmação de reservas e notificação de erros; e SResvTear (Sender Reservation Teardown), usada para terminar reservas explicitamente.

As mensagens SResv são normalmente originadas no emissor, podendo, no entanto, ser originadas em nós intermédios para efeitos de reparação local, por exemplo em resposta a uma alteração de rota. As mensagens SResv de iniciação incluem a identificação do fluxo, um objecto SRESV_PARAMS contendo a especificação da classe de serviço (através do PHB correspondente) e um valor de tempo para expirar a reserva, a quantificação da reserva, cujo formato depende da classe de serviço, e um objecto LABEL_SETUP, usado para instalar as etiquetas B, alterado em cada salto. Após receber uma mensagem SResv inicial, o pedido é sujeito a controlo de admissão. Se for aceite, o *router* actualiza a reserva agregada para a classe do fluxo, cria uma nova entrada na tabela de reservas, armazena a etiqueta B da reserva, actualiza o objecto LABEL_SETUP para incluir a sua própria etiqueta (T), e passa a mensagem ao nó seguinte. Se, algures ao longo do percurso, o controlo de admissão falhar, é enviada no sentido contrário uma mensagem SResvStat para indicar a falha. Esta mensagem já faz uso das etiquetas B instaladas para acesso directo às estruturas.

Quando a mensagem SResv atinge o receptor, todos os *routers* ao longo do percurso têm recursos reservados para o novo fluxo e todas as etiquetas para processamento de mensagens em sentido inverso (B) estão instaladas. O receptor confirma a reserva enviando uma mensagem SResvStat em direcção ao emissor. Esta mensagem inclui um objecto LABEL que permite um acesso directo à estrutura da reserva e um objecto LABEL_SETUP para instalar as etiquetas no outro sentido (F). Ambos os objectos são comutados em cada nó. A recepção desta confirmação despoleta, em cada nó, a activação da reserva nos módulos de policiamento e escalonamento. Se, pelo contrário, se tratasse de uma mensagem SResvStat de erro a reserva seria removida do módulo de controlo de admissão.

As mensagens SResvTear são normalmente originadas no emissor para terminar reservas explicitamente, mas podem igualmente ser originadas em nós intermédios em caso de alteração de rota para remover a reserva na rota antiga. Ela inclui o objecto LABEL para acesso directo às estruturas de reserva.

As reservas precisam de ser periodicamente refrescadas usando mensagens SResv de refrescamento. Em relação às SResv iniciais, as de refrescamento são mais simples, pois dispensam a maior parte dos objectos usados na iniciação da reserva, e incluem um objecto LABEL para um acesso directo às estruturas de reserva.

Uma descrição detalhada do protocolo de sinalização, incluindo a especificação de todos os objectos nas mensagens, pode encontrar-se em [12].

2.7 Temporizadores para estado volátil

No SRBQ as reservas são voláteis: se não forem refrescadas nem terminadas explicitamente o temporizador associado expira e são removidas. A implementação básica de temporizadores genéricos baseia-se em filas ordenadas de eventos: o processador espera até ao instante do primeiro evento, retira-o da lista, efectua o processamento e volta a esperar pelo evento seguinte. Apesar de a remoção de eventos ser trivial, a inserção em listas ordenadas é uma operação custosa, altamente dependente do número de eventos na lista. Pelo contrário, temporizadores de atraso fixo são de implementação simples e eficiente, consistindo numa única fila FIFO. Para conciliar as vantagens destes dois tipos de temporizadores, foi desenvolvido um algoritmo com inserção trivial e remoção de complexidade $O(1)$, disponibilizando 8 valores possíveis de atraso numa escala logarítmica (base 2) com uma gama de 1:128. A implementação baseia-se em 8 filas de eventos, cada uma com um atraso fixo e, portanto, servidas em disciplina FIFO. A inserção de um evento é trivial, consistindo em adicioná-lo à cauda da fila correspondente ao atraso desejado. A remoção consiste em escolher a fila cujo primeiro evento tem o tempo de expiração mais baixo e retirar esse primeiro evento dessa fila.

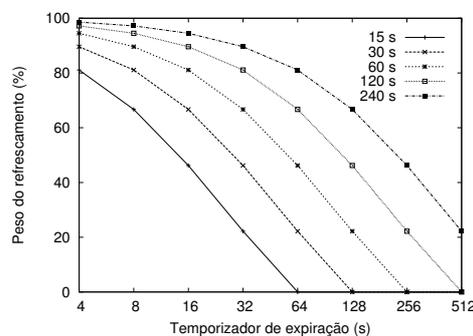


Figura 3: Peso relativo das mensagens de refrescamento

A existência de uma ampla gama de valores de temporização permite que as reservas de curta duração não fiquem pendentes na rede se algo de anormal ocorrer (tal como o bloqueio da aplicação ou uma mudança de rota não detectada) e que as de mais longa duração não geram demasiado tráfego de sinalização apenas para o seu refrescamento. A figura 3 mostra o peso relativo das mensagens de refrescamento no total de sinalização para fluxos com duração entre 15 s e 240 s, usando os oito valores possíveis de temporizador. A base de tempo é de 4 s, e as mensagens de refrescamento

são enviadas a uma taxa 4 vezes superior ao inverso do tempo de expiração, por forma a suportar algumas perdas. Como pode observar-se, o peso das mensagens de refrescamento pode, nestas condições, variar entre 0% e 98,6% do total da sinalização. Assim, as aplicações devem escolher um valor de temporizador equilibrado entre a geração de tráfego de sinalização e a recuperação rápida de condições de erro. Não sendo possível estimar *a priori* a duração do fluxo, a aplicação pode escolher um valor inicial baixo, incrementando-o posteriormente nas mensagens de refrescamento.

3 Avaliação do desempenho

A arquitectura SRBQ foi implementada no simulador ns-2 [1]. Esta secção contém uma análise dos resultados de simulação, visando essencialmente as garantias de QoS. Apesar de extremamente importante, a medição da eficiência em termos de processamento está fora do âmbito deste artigo, dada a inaptidão do ns-2 para avaliar tempos de processamento.

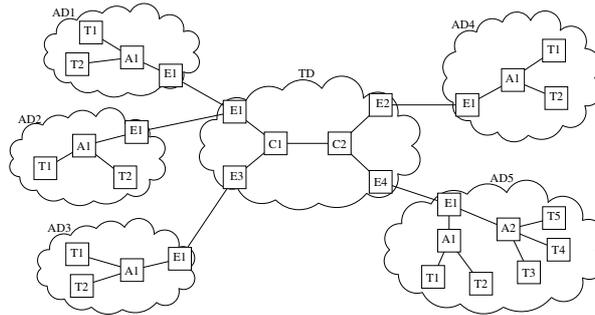


Figura 4: Topologia usada na avaliação do SRBQ

A figura 4 ilustra a topologia simulada. Ela inclui 1 domínio de trânsito (TD), contendo nós interiores (C) e de fronteira (E), e 5 de acesso (AD), contendo nós de fronteira (E), de acesso (A) e terminais (T). Cada terminal nos domínios de acesso simula um conjunto de terminais. O nó de fronteira E1 do domínio de trânsito está ligado a mais que um domínio de acesso por forma a verificar o correcto funcionamento do policiamento à entrada do domínio. A largura de banda das ligações no domínio de trânsito e nas interligações com os domínios de acesso é de 10 Mbps. O atraso de propagação é de 2 ms no domínio de trânsito e 1 ms nas interligações com os domínios de acesso.

No cenário simulado coexistem as classes GS, CL e BE. Em cada ligação, o débito atribuído à classe de sinalização é de 1 Mbps. Note-se que, apesar de este valor ser elevado, o excesso pode ser usado pela classe BE. O débito atribuído à classe GS é de 3 Mbps, e à classe CL é de 4 Mbps. O restante débito, assim como o não utilizado pelos fluxos de outras classes, é utilizado para tráfego BE. Cada terminal do lado esquerdo gera um conjunto de fluxos das 3 classes mencionadas, sendo o destino escolhido aleatoriamente entre todos os terminais do lado direito. O tráfego em cada classe consiste em fluxos de tipos diferentes.

O controlo de admissão na classe GS é PBAC; na classe CL são comparados resultados de simulações usando PBAC e MBAC. O algoritmo MBAC usado é uma adaptação do Measured Sum (MS) para 3 níveis de probabilidade de perda, em que o tráfego estimado para cada nível é adicionado ao estimado para os níveis de probabilidade de perda mais baixa. O factor de utilização alvo é de 95%.

Todas as simulações tiveram a duração de 5400 segundos simulados, sendo eliminados os dados relativos aos primeiros 1800 segundos. Todos os valores representam uma média de, no mínimo, 5 simulações com sementes de geração de números aleatórios diferentes.

3.1 QoS extremo-a-extremo

Esta subsecção apresenta os resultados de 3 experiências visando avaliar o desempenho em termos de QoS da arquitectura SRBQ. O conjunto de fluxos usado é o seguinte (tabela 1): a classe GS é composta por fluxos de débito constante (CBR64 e CBR256) e exponenciais *on-off*; a classe CL é composta por exponenciais *on-off* e Pareto *on-off*; a classe BE é composta por fluxos Pareto *on-off* e FTP. Os fluxos BE estão permanentemente activos (3 FTP e 2 Pareto por fonte), enquanto nas outras classes são iniciados segundo um processo de Poisson com um certo tempo médio entre geração de chamadas (TMEC), sendo a duração exponencialmente distribuída.

O maior valor médio de carga oferecida (CMO) nas classes GS e CL é, em termos de tráfego, 20% superior aos valores de débito atribuídos a essas classes, que, devido a diferentes misturas de fluxos, se traduzem em valores de reserva pedidos (RMO) com excessos de 26% (GS) e 42% (CL). Os valores apresentados na tabela correspondem à carga máxima (factor de carga 1.2), sendo os factores de carga inferiores obtidos por variação do TMEC.

Nos fluxos GS, a taxa reservada (Déb. res.) representa o débito do *token bucket*, e a rajada (Burst res.) a sua profundidade. Estes parâmetros incluem uma pequena margem para compensar erros de cálculo em vírgula flutuante. Nos fluxos CL, MA 1, Déb. res. e MA 3 representam as três marcas de água. A soma dos débitos dos fluxos em cada

Classe	Tipo	Déb. méd. (kbps)	Déb. pico (kbps)	Tam. pac. (bytes)	T. on (ms)	T. off (ms)	Déb. res. (kbps)	Burst res. (bytes)	MA 1 (kbps)	MA 3 (kbps)	TMEC (s)	Dur. méd. (s)	CMO (kbps)	RMO (kbps)
GS	CBR48	48	48	80	-	-	48.048	81	-	-	45	120	768	769
	CBR256	256	256	1000	-	-	256.256	1050	-	-	180	240	2048	2050
	Exp.	128	256	1000	200	200	160	5000	-	-	90	90	768	960
CL	Pareto	128	256	1000	200	200	150	-	64	256	38	120	2425	2842
	Exp.	128	256	1000	200	200	150	-	64	256	38	120	2425	2842
Fluxos simult.														
BE	FTP	-	-	1040	-	-	-	-	-	-	-	3 por terminal orig.	var.	N.A.
	Pareto	128	256	1000	200	200	-	-	-	-	2 por terminal orig.	2304	N.A.	

Tabela 1: Fluxos para os testes de QoS extremo-a-extremo

marca de água não pode exceder os limites de utilização, que foram configurados para 0.7, 1.0 e 1.7 vezes o débito atribuído à classe CL.

Com a primeira experiência pretende-se avaliar a QoS extremo-a-extremo nas classes GS e CL para valores diferentes de carga oferecida na classe CL. A figura 5 mostra o atraso médio e as perdas quando o factor de carga oferecida é 1.2 na classe GS e varia entre 0.6 e 1.2 na classe CL, combinando os resultados de simulações com PBAC e MBAC na classe CL. O atraso na classe GS é constante e muito pequeno, excepto nos fluxos exponenciais. Nestes é introduzido um atraso adicional e potencialmente elevado no formatador de entrada do *router* de acesso por transmitirem a um débito superior ao reservado por longos períodos. A responsabilidade é, todavia, apenas da aplicação por transmitir tráfego não conforme à reserva. O facto de o atraso ser muito baixo nos restantes fluxos GS mostra que não são afectados negativamente. Na classe CL o atraso é praticamente constante com PBAC, mas aumenta perceptivelmente com MBAC. O comportamento do *jitter* é semelhante ao do atraso.

As perdas de pacotes são nulas para os fluxos GS conformes; para os exponenciais são de cerca de 0.13%, e devem-se à limitação de espaço na fila de espera no formatador de entrada, pois em certos períodos estes fluxos excedem largamente o perfil de tráfego contratado. Nos fluxos CL as perdas aumentam com a carga oferecida, mas são sempre muito baixas (<0.03%) com PBAC. Com MBAC elas aumentam significativamente, chegando a atingir 0.25% nos fluxos Pareto, de cauda pesada. A utilização da classe GS manteve-se constante em 83% do débito atribuído. Na classe CL variou entre 60% e 78% (com PBAC) ou 83% (com MBAC). A maior utilização com MBAC é responsável pelos piores valores de QoS, pois é impossível fazer uma previsão exacta da ocupação da classe.

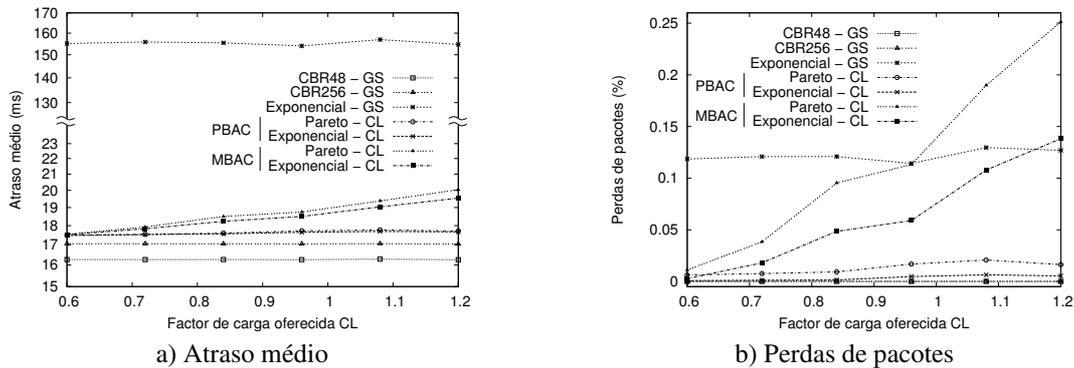


Figura 5: Atraso médio e perdas com variação da carga CL oferecida

Numa segunda experiência variou-se o factor de carga oferecida GS entre 0.6 e 1.2, mantendo-se fixo em 1.2 o factor de carga oferecida CL. Todos os resultados de QoS foram constantes, tanto na classe GS como na CL, não sendo afectados pela carga oferecida GS.

Os resultados destas experiências mostram que, apesar de baseado em agregados, o SRBQ suporta garantias suaves e estritas de QoS, obtendo uma independência total entre as classes de tráfego.

3.2 Independência de fluxos

Nesta subsecção avalia-se o desempenho da arquitectura na presença de fluxos que transmitem a taxas muito superiores às reservadas por períodos de tempo consideráveis. É igualmente avaliada a influência desses fluxos nos restantes, tanto na classe GS como na CL. De modo a proteger a rede de fluxos não-conformes, o *router* de acesso efectua formatação de tráfego por fluxo à entrada na classe GS. Este formatador absorve o *jitter* de multiplexagem do terminal e garante que o tráfego injectado na rede não excede os valores reservados através da absorção de rajadas da aplicação maiores que o *bucket* (de 5 pacotes, neste caso), protegendo assim os restantes fluxos GS. Os fluxos CL são policiados, e não formatados, à entrada do *router* de acesso. Assim, um fluxo CL não-conforme é penalizado em termos de probabilidade de perda, mas não é significativamente afectado em termos de atraso sofrido.

Nesta experiência, a carga média oferecida em ambas as classes excede em 23% o respectivo débito. Em cada classe existem 3 tipos de fluxos (tabela 2): fluxos CBR (CBR64), que são conformes às reservas; fluxos exponenciais *on-*

Classe	Tipo	Déb. méd. (kbps)	Déb. pico (kbps)	Tam. pac. (bytes)	T. on (ms)	T. off (ms)	Déb. res. (kbps)	Burst res. (bytes)	MA 1 (kbps)	MA 3 (kbps)	TMEC (s)	Dur. méd. (s)	CMO (kbps)	RMO (kbps)
GS	CBR64	64	64	500	-	-	64.064	501	-	-	75	240	1229	1230
	Exp1	128	256	1000	200	200	160	5000	-	-	75	120	1229	1536
	Exp2	128	var.	1000	var.	var.	160	5000	-	-	75	120	1229	1536
CL	CBR64	64	64	500	-	-	65	-	64	66	75	240	1229	1248
	Exp1	128	256	1000	200	200	150	-	64	256	50	120	1843	2160
	Exp2	128	var.	1000	var.	var.	150	-	64	256	50	120	1843	2160

Tabela 2: Fluxos para o teste de isolamento

off (Exp1) com um período médio de actividade de 50% do período médio de ciclo (tempos médios de actividade e inactividade de 200 ms), ligeiramente não-conformes; fluxos exponenciais *on-off* (Exp2) altamente não-conformes, pois transmitem a débitos muito superiores ao reservado por longos períodos de tempo. O tempo médio de actividade destes últimos varia entre 50% e 12.5% do tempo médio de ciclo, variando o débito de pico entre 256 kbps (com tempos médios de actividade e de inactividade de 200 ms) e 1024 kbps (tempo médio de actividade de 50 ms e de inactividade de 350 ms). Note-se que o tempo médio de ciclo se mantém constante (400 ms), assim como o débito médio. A elevada não-conformidade destes fluxos advém da grande diferença entre o débito de pico e o débito reservado.

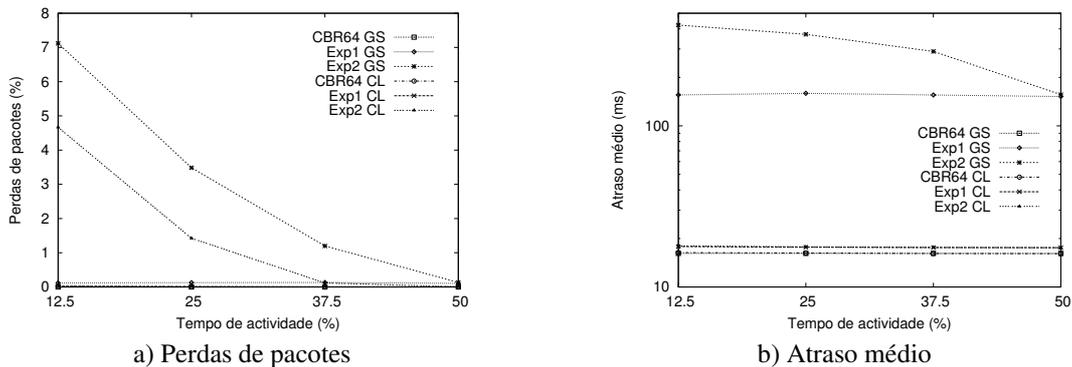


Figura 6: Atraso médio e perdas de pacotes com fluxos não-conformes

A figura 6 mostra as perdas e o atraso médio dos três tipos de fluxos com a variação do tempo médio de actividade dos fluxos altamente não-conformes (Exp2). Não existem perdas nos fluxos GS conformes (CBR64), e nos ligeiramente não-conformes (Exp1) as perdas são de cerca de 0.1%. Nos fluxos altamente não-conformes as perdas atingem 7.1% com tempos médios de actividade de 12.5%, caso em que a taxa de transmissão de pico é muito superior ao débito reservado. Os restantes fluxos não são, no entanto, afectados por estes.

A classe CL não dá as mesmas garantias absolutas da GS: as perdas variam entre 0.015% e 0.002% nos fluxos conformes e entre 0.024% e 0.006% nos ligeiramente não-conformes quando o tempo médio de ciclo dos fluxos não-conformes varia entre 12.5% e 50%. Note-se que mesmo os fluxos ligeiramente não-conformes sofrem valores muito baixos de perdas. Os fluxos altamente não-conformes sofrem perdas que variam entre 4.6% e 0.006%, sendo penalizados pela não-conformidade.

O atraso médio dos fluxos GS conformes é muito baixo, devendo-se essencialmente aos atrasos de transmissão e propagação. Os fluxos GS ligeiramente não-conformes têm um atraso significativamente maior, de 160 ms. Note-se que estes fluxos têm um débito de pico cerca de 100 kbps superior à reserva, pelo que vão sofrer atrasos (e pequenas quantidades de perdas) no formatador de entrada no *router* de acesso quando transmitem ao débito de pico por períodos mais prolongados. Como esperado, os fluxos altamente não-conformes sofrem atrasos maiores, que chegam aos 400 ms. As curvas de *jitter* (omitidas) têm um comportamento semelhante às de atraso. É de salientar que todos os fluxos GS são agregados, partilhando a mesma fila, internamente servida segundo uma disciplina FIFO, o que implica que os atrasos em filas de espera são partilhados por todos eles. Assim, os elevados valores de atraso dos fluxos não-conformes são infligidos no formatador de entrada. Esta penalização dos fluxos não-conformes, em termos de perdas e atrasos, visa proteger os restantes fluxos GS. Assim, os fluxos GS conformes mantêm baixos valores de atraso e perdas nulas independentemente dos não-conformes. O estabelecimento de reservas inadequadas ao tráfego transmitido é da exclusiva responsabilidade da aplicação.

Em relação ao atraso, o comportamento da classe CL é inteiramente diferente do da GS. Uma vez que não existe formatação, apenas policiamento, os fluxos não-conformes não são significativamente penalizados em termos de atraso: em média, ele é apenas 2.6% mais elevado com tempos de actividade de 12.5% que com tempos de actividade de 50%. No entanto, existe igualmente um pequeno aumento, de cerca de 1.5%, no atraso dos restantes fluxos CL.

Os resultados desta experiência mostram uma resposta do sistema adequada na presença de fluxos não-conformes, mantendo uma independência completa entre fluxos GS, esperada neste tipo de serviço. A classe CL é mais tolerante, penalizando os fluxos não-conformes apenas em termos de perdas, infligidas nos *routers* de acesso e de fronteira através da (re)marcação e do policiamento de pacotes. Estas perdas induzidas mantêm baixo o nível de congestionamento da

Classe	Tipo	Déb. méd. (kbps)	Tam. pac. (bytes)	Déb. res. (kbps)	Burst res. (bytes)	MA 1 (kbps)	MA 3 (kbps)	TMEC (s)	Dur. méd. (s)	CMO (kbps)	RMO (kbps)
GS	Vídeo	16	var.	17	4000	-	-	151	180	114	122
	Vídeo	64	var.	68	5000	-	-	151	180	458	486
	Vídeo	256	var.	272	8000	-	-	151	180	1831	1945
CL	Vídeo	16	var.	16.5	-	12	40	60	180	288	297
	Vídeo	64	var.	66	-	48	96	60	180	1152	1188
	Vídeo	256	var.	264	-	192	352	60	180	4608	4752

Tabela 3: Características dos fluxos multimédia reais

rede, protegendo (embora sem isolar completamente) os fluxos CL conformes. Esta classe é, portanto, mais adequada para fluxos mais variáveis e com requisitos de QoS menos estritos.

Os resultados de QoS obtidos não são extraordinários: o grande mérito do modelo SRBQ está na sua obtenção de forma escalável, numa arquitectura baseada em agregação, mantendo uma boa utilização dos recursos de rede.

3.3 Fluxos multimédia reais

As experiências anteriores basearam-se em fluxos sintéticos com características específicas para a avaliação de diferentes aspectos do desempenho da arquitectura. Por forma a avaliar o desempenho em condições normais de utilização foi efectuada uma outra experiência usando *traces* de pacotes de fluxos multimédia reais. Estas listas correspondem a *streams* de vídeo H.263 com débitos entre 16 kbps e 256 kbps, disponíveis em [2]. Nesta experiência foram atribuídos às classes GS e CL débitos de 2 Mbps e 5 Mbps, respectivamente. As marcas d'água da classe CL foram ajustadas para 0.8, 1.0 e 2.0 vezes o débito atribuído. A tabela 3 resume os parâmetros dos fluxos usados. Foram usados diversos fluxos de cada taxa de transmissão, sendo o ponto de início do fluxo aleatoriamente escolhido. Os fluxos são iniciados segundo um processo de Poisson, tendo a sua duração uma distribuição de Pareto.

A figura 7 ilustra o atraso médio e as perdas de pacotes com a variação da carga oferecida em cada classe entre 60% e 120% do débito respectivo. O atraso na classe GS não é afectado pela carga oferecida, equanto na classe CL exhibe uma ligeira tendência de crescimento. Note-se que o atraso fica sempre abaixo dos 20 ms em ambas as classes. As curvas de *jitter* têm um comportamento semelhante, mas com valores mais elevados na classe GS; estes valores devem-se à formatação no *router* de acesso, pois estes fluxos apresentam variações não negligenciáveis do débito instantâneo.

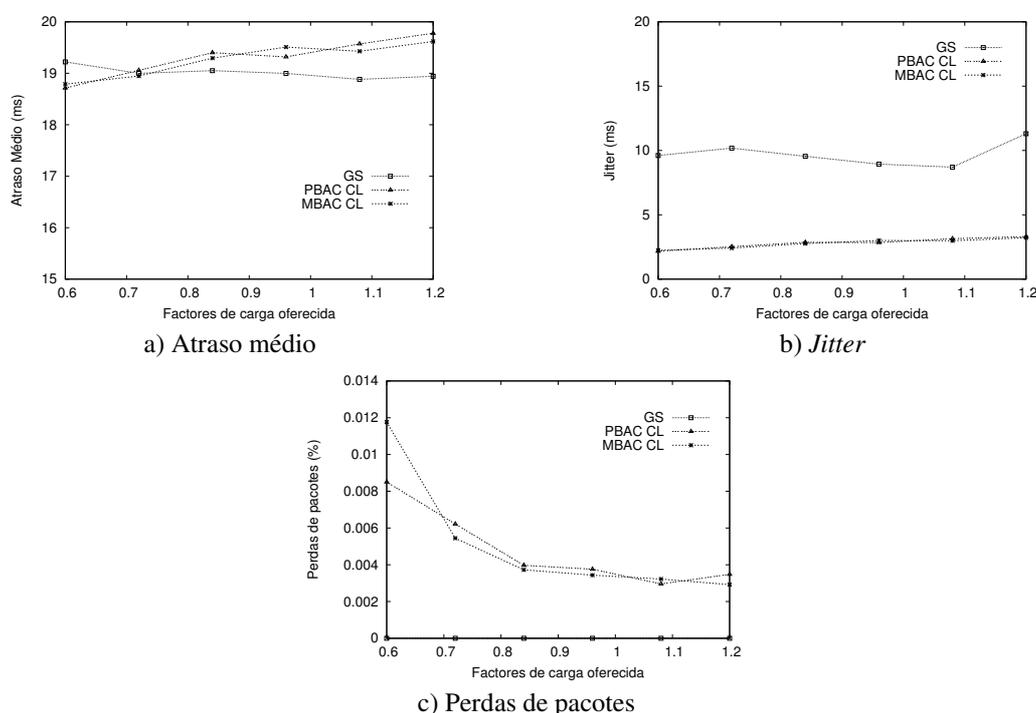


Figura 7: Resultados de QoS com fluxos multimédia reais

As perdas na classe GS são nulas, pois o formatador no *router* de acesso absorve as rajadas superiores ao tamanho do *bucket* reservado, traduzindo-as em atrasos em vez de perdas. Na classe CL elas têm um comportamento aparentemente contraditório, diminuindo com o aumento da carga oferecida. Existe, no entanto, uma explicação simples para este facto, resultante da combinação de diversos factores. Por um lado, os fluxos H.323 de débito mais baixo têm pacotes mais pequenos e maiores variações no débito instantâneo que os de débito mais elevado. Por outro lado, a segunda marca de água (com um factor de utilização alvo de 1) usada nas reservas é apenas 3% superior ao débito nominal dos fluxos. A

diferença absoluta entre o débito reservado e o débito nominal é, assim, maior nos fluxos de débito nominal mais elevado, o mesmo acontecendo em relação à terceira marca de água. Por fim, a remarcação efectuada nos *routers* de fronteira é feita com base em agregados. Estes 3 factores combinados implicam que quando existem mais fluxos de elevado débito na rede os de menor débito beneficiam do excedente de reserva dos primeiros, diminuindo a sua taxa de perdas. Este factor tem um peso muito maior que o congestionamento (mínimo) na rede.

Os resultados apresentados indicam que o modelo SRBQ cumpre os requisitos das classes de serviço suportadas com o uso de fluxos multimédia reais.

4 Comparação com o modelo de Agregação RSVP

Nesta secção efectua-se uma comparação do SRBQ com o modelo de agregação de reservas RSVP (RSVPRAg) [4] proposto pela IETF. Ao contrário do SRBQ, em que a escalabilidade da sinalização é obtida através do uso de técnicas e algoritmos computacionalmente eficientes, no RSVPRAg ela é obtida pela redução do número de mensagens de sinalização processadas. Este modelo define uma extensão do protocolo RSVP [9] que permite “esconder” a sinalização extremo-a-extremo no interior de regiões de agregação. No caso mais simples, os *routers* de fronteira reservam agregados entre a entrada e a saída do domínio, atribuindo a um agregado os fluxos que partilhem os nós de entrada e de saída e a classe de tráfego. Por forma a reduzir a sinalização processada no interior do domínio, as alterações às reservas dos agregados são efectuadas em valores muito superiores ao débito médio dos fluxos individuais, designados *bulks*. Quando um novo fluxo tenta a admissão na região de agregação, o *router* de fronteira verifica se existe largura de banda suficiente no agregado correspondente. Se existir, o fluxo é aceite sem sinalização no *core*. Se não existir, é feita uma tentativa de aumentar a largura de banda do agregado, sendo o fluxo aceite se este aumento for bem sucedido ou rejeitado no caso contrário.

No RSVPRAg os *routers* interiores apenas mantêm estado de agregados, enquanto no SRBQ mantêm de fluxos individuais. Esta não é, no entanto, uma limitação do SRBQ, como já se viu, pois a quantidade de memória existente em qualquer *router* comercializado actualmente é suficiente. O acesso a essas estruturas poderia ser penalizador, particularmente em termos de pior caso, se efectuado com base nos 5 campos que identificam o fluxo. No entanto, esse problema não existe no SRBQ, pois o acesso é directo com base nas etiquetas.

A escalabilidade dos mecanismos de classificação e de escalonamento a nível do *core* é semelhante nos dois modelos, sendo efectuados por agregados, com base no DSCP, em ambos os modelos. Na fronteira, a classificação é mais simples no SRBQ, pois apenas se baseia no DSCP, não sendo necessário mapear fluxos individuais em agregados.

Para efectuar uma comparação mais substanciada, foram efectuadas simulações no ns-2 usando as mesmas condições de tráfego oferecido e a mesma topologia nos dois modelos: as regiões de agregação e não-agregadas no RSVPRAg correspondem aos domínios de *core* e de acesso no SRBQ; os agregadores e desagregadores correspondem aos *routers* de fronteira. A topologia usada ilustra-se na figura 8. Com 3 *routers* de fronteira de cada lado, poderão existir até 9 agregados em simultâneo no ligação central.

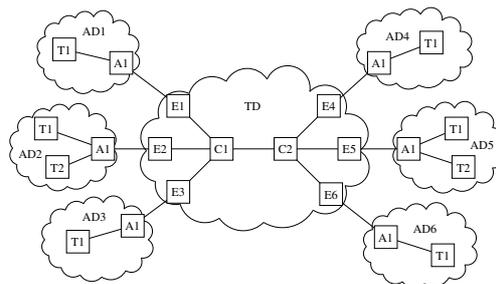


Figura 8: Topologia usada na comparação dos modelos

Nas simulações efectuadas foram usadas as classes BE e CL. Dadas as dificuldades no suporte da classe GS no modelo de agregação, esta classe não foi usada. A largura de banda atribuída em cada ligação à classe CL é de 7 Mbps. As restantes condições são semelhantes às das simulações anteriores. O tráfego de *background* na classe BE compõe-se de fluxos FTP e Pareto *on-off*. Na classe CL foram usados fluxos CBR, exponenciais *on-off* e Pareto *on-off*. A tabela 4 resume as características destes fluxos. No modelo RSVPRAg as reservas caracterizam-se por *token-buckets*, ao passo que no SRBQ se caracterizam pelas três marcas de água. O tempo médio entre geração de chamadas (TMEC) foi escolhido por forma a que exista, em média, a mesma largura de banda reservada para todos os tipos de fluxos. Este parâmetro foi ajustado simultaneamente em todos eles por forma a variar a carga oferecida entre 80% e 120% (factores de carga 0.8 a 1.2) da largura de banda atribuída à classe CL na ligação com maior tráfego. Os valores apresentados na tabela correspondem ao factor de carga máximo (1.2).

Tipo	Déb. méd. (kbps)	Déb. pico (kbps)	Tam. pac. (bytes)	T. on (ms)	T. off (ms)	Token bucket		Marcas d'água (kbps)			TMEC (s)	Dur. méd. (s)
						R (kbps)	B (bytes)	1	2	3		
CBR	48	-	500	-	-	48	1500	48	48.048	56	11	120
CBR	64	-	500	-	-	64	1500	64	64.064	72	14.6	120
Exp.	48	96	500	200	200	64	15000	32	64	96	14.6	120
Pareto	48	96	500	200	200	64	15000	32	64	96	14.6	120

Tabela 4: Características dos fluxos usados na comparação

A figura 9 mostra os resultados mais relevantes. No modelo de agregação, os atrasos e perdas apresentados correspondem a um *bulk* de 300 kbps, apenas cerca de cinco vezes superior às reservas. Como pode observar-se, no SRBQ todos os fluxos sofrem atrasos similares, o que é óbvio, pois partilham as filas de espera. O facto de no modelo de agregação RSVP os atrasos serem diferentes deve-se ao uso da disciplina de serviço WFQ (*Weighted Fair Queuing*) fora da região de agregação. Assim, os fluxos Pareto (uma distribuição de cauda pesada) sofrem atrasos médios consideravelmente superiores aos restantes fluxos.

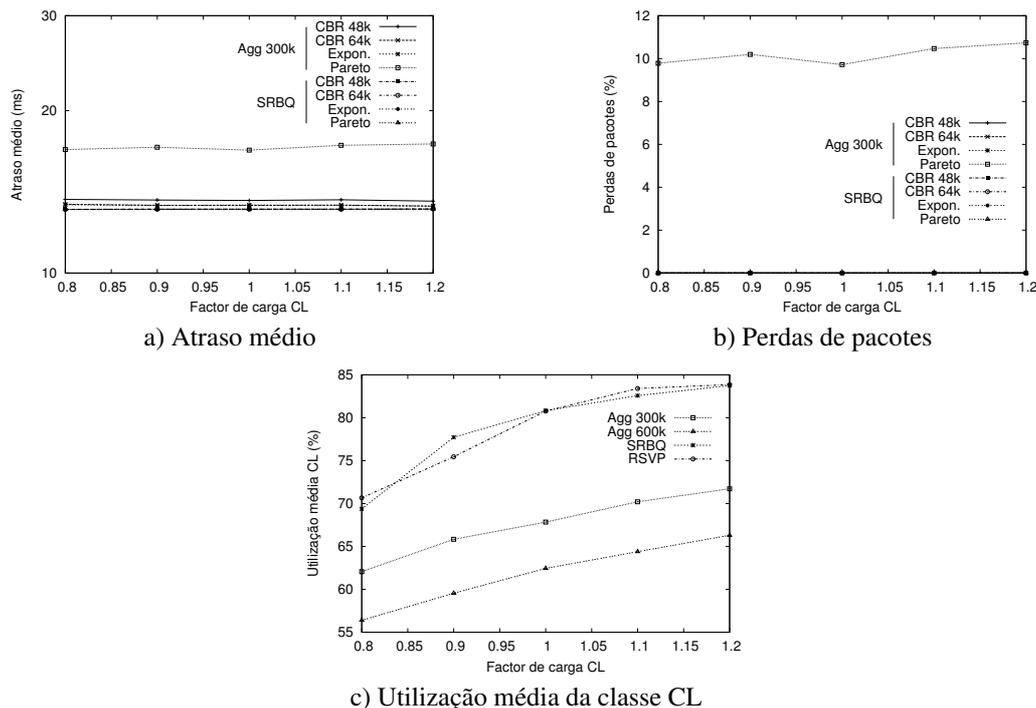


Figura 9: Resultados de QoS e utilização nos dois modelos

Relativamente a perdas, os fluxos Pareto, ao contrário dos restantes, sofrem perdas significativas, de cerca de 10%. Note-se que esta é uma distribuição de cauda pesada, com variância infinita, que não é, portanto, apropriada para uma caracterização *token bucket*. Assim, a menos que se usem valores de *bucket* demasiado elevados, as perdas são sempre significativas. Pelo contrário, a caracterização por marcas de água adequa-se muito melhor a estes fluxos, como se verifica pelos valores baixos de perdas obtidos para estes fluxos no SRBQ (menos de 0.003%). Para os fluxos exponenciais, as perdas são muito baixas no RSVPRAgg (cerca de 0.003%) e nulas no SRBQ. Nos fluxos CBR não existiram perdas em qualquer dos modelos.

A figura 9.c compara a utilização da classe CL na ligação central (comum a todos os fluxos) nos dois modelos, e ainda com o RSVP tradicional em condições similares. O SRBQ tem uma utilização muito semelhante à do RSVP, exibindo saturação à volta do factor de carga oferecida 1. No modelo de agregação, a utilização é substancialmente mais baixa, particularmente quando se usa um *bulk* de 600 kbps (ainda assim apenas cerca de 10 vezes superior às reservas por fluxo); além disso, não é visível saturação, pois ela inicia-se com factores de carga mais baixos e é bem mais gradual.

Dos resultados anteriores, podemos concluir que ambos os modelos têm um suporte adequado de QoS, excepto para os fluxos Pareto no modelo RSVPRAgg, dada a inadequada caracterização por *token bucket*. No entanto, a utilização é substancialmente melhor no modelo SRBQ.

5 Conclusões e trabalho futuro

Neste artigo foi apresentada a arquitectura SRBQ, que suporta QoS extremo-a-extremo em classes de serviço com garantias suaves e estritas. Apesar de baseada em reservas por fluxo, a carga em termos de processamento é muito reduzida quando comparada com o modelo RSVP/IntServ, pelo que, ao contrário deste, é escalável.

Em primeiro lugar, a classificação e o escalonamento dos pacotes são efectuados com base em agregados, correspondendo às classes de serviço, e não por fluxo. Assim, a complexidade destas tarefas é independente do número de fluxos: elas dependem apenas do número de classes, que é fixo e pequeno. Além disso, os algoritmos de escalonamento usados são simples e computacionalmente eficientes.

Em segundo lugar, os temporizadores de expiração são implementados de forma extremamente eficiente. A flexibilidade de temporizadores genéricos, que não era aqui necessária, foi trocada por uma implementação eficiente usando um pequeno número de valores discretos, permitindo um custo em termos de processamento muito baixo e constante. A escala logarítmica permite uma boa gama de valores para estes temporizadores.

Por fim, o mecanismo de comutação de etiquetas diminui o tempo de processamento das mensagens nos *routers*, principalmente em termos de pior caso, aumentando a escalabilidade do modelo. Sem este mecanismo, o simples acesso às estruturas de reserva seria dependente do número de fluxos, que pode ser enorme em *routers* interiores de alto débito. Usando as etiquetas, o acesso a essas estruturas é imediato.

A análise dos resultados obtidos por simulação indica que a arquitectura SRBQ suporta adequadamente ambas as classes de serviço IntServ. Os resultados obtidos com fluxos multimédia reais são indicadores adicionais de um bom desempenho em condições de utilização reais. Os resultados da comparação com o modelo RSVPRAgg mostram que o SRBQ tem um desempenho igual ou superior em termos de QoS e uma utilização de recursos apreciavelmente melhor.

O posterior desenvolvimento do trabalho aqui apresentado passará pela implementação de um protótipo que permita avaliar de forma quantitativa a escalabilidade do modelo, dada a inadequação do simulador ns-2 para este tipo de avaliação. A integração com cenários de mobilidade e redes sem fios é um outro desenvolvimento a efectuar. Deverá ainda ser avaliada a possibilidade de integração com outros modelos de QoS existentes e ainda com *multicast*. Um último tópico para trabalho futuro é a integração de mecanismos de contabilização e cobrança que permitam a um operador de rede obter o incentivo financeiro para a implementação do modelo.

Referências

- [1] The Network Simulator ns-2. <http://www.isi.edu/nsnam/ns/>.
- [2] Arizona State University. MPEG-4 and H.263 Video Traces for Network Performance Evaluation. <http://trace.eas.asu.edu/TRACE/trace.html>, 2004.
- [3] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209, Internet Engineering Task Force, December 2001.
- [4] F. Baker, C. Iturralde, F. Le Faucheur, and B. Davie. Aggregation of RSVP for IPv4 and IPv6 Reservations. RFC 3175, Internet Engineering Task Force, September 2001.
- [5] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine. A Framework for Integrated Services Operation over Diffserv Networks . RFC 2998, Internet Engineering Task Force, November 2000.
- [6] S. Blake, D. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC 2475, Internet Engineering Task Force, December 1998.
- [7] Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus*. Springer-Verlag, 2001.
- [8] R. Braden, D. Clarck, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633, Internet Engineering Task Force, June 1994.
- [9] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource Reservation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, Internet Engineering Task Force, September 1997.
- [10] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint admission control: Architectural issues and performance. In *Proceedings of ACM SIGCOMM 2000*, August 2000.
- [11] C. Cetinkaya and E. Knightly. Egress Admission Control. In *Proceedings of IEEE INFOCOM 2000*, March 2000.
- [12] R. Prior, S. Sargento, S. Crisóstomo, and P. Brandão. End-to-end Quality of Service with Scalable Reservations. In *Proceedings of the 11th International Conference on Telecommunication System, Modeling and Analysis*, October 2003.
- [13] S. Sargento, R. Valadas, and E. Knightly. Resource Stealing in Endpoint Controlled Multi-class Networks. In *Proceedings of IWDC 2001*, September 2001. Invited paper.
- [14] Ion Stoica. *Stateless Core: A Scalable Approach for Quality of Service in the Internet*. PhD thesis, Carnegie Mellon University, December 2000.