

Modelamento e Simulação de um Escalonador WF²Q Aplicado à Redes ATM

Antônio M. Alberti, Daniel Zaccarias, Samuel Penha e Leonardo de Souza Mendes

Departamento de Comunicações - Universidade Estadual de Campinas - UNICAMP - Campinas - SP - Brasil

Resumo—Algoritmos de escalonamento que aproximam o GPS (*Generalized Processor Sharing*) tem obtido bastante êxito em garantir o isolamento de QoS entre conexões em redes de serviços integrados, tais como redes ATM. Dentre estes algoritmos, o WF²Q (*Worst-case Fair Weighted Fair Queuing*) tem sido considerado um dos mais idênticos ao GPS. Assim sendo, este artigo apresenta o modelamento de um escalonador WF²Q para o ambiente de simulação SimNT 2.0. O objetivo do trabalho é expandir um conjunto de modelos no nível de células para a simulação integrada das funções de gerenciamento de tráfego ATM. Portanto, o modelo desenvolvido além de implementar eficientemente o algoritmo WF²Q, deve ainda interagir com os demais modelos deste conjunto, como por exemplo, com modelos de estruturas de filas. O artigo apresenta também a validação do modelo desenvolvido, bem como um exemplo de aplicação deste modelo no transporte de tráfego de vídeo codificado MPEG-4 sobre uma rede ATM.

Palavras-Chave—WF²Q, escalonador, modelamento, simulação e ATM.

Abstract—Scheduling algorithms that approximate GPS (*Generalized Processor Sharing*) have obtained great success in QoS isolation between integrated services network connections, such as ATM networks. Among these algorithms, WF²Q (*Worst-case Fair Weighted Fair Queuing*) has been considered one of the algorithms closest to GPS. Thus, this article presents the modeling of a WF²Q scheduler for the SimNT 2.0 simulation environment. The objective is to expand a set of models for integrated ATM traffic management functions simulation. Therefore, the developed model, besides implementing WF²Q algorithm, must also interact with other models of this set, such as, with queuing structure models. The paper also presents the developed model validation and its application to the simulation of MPEG-4 coded video traffic over an ATM network.

Index Terms—WF²Q, scheduler, modeling, simulation and ATM.

I. INTRODUÇÃO

Um dos aspectos mais importantes no projeto de redes de serviços integrados é a escolha da disciplina

de serviço¹ a ser utilizada em cada ponto de armazenamento de pacotes na rede a fim de selecionar a ordem apropriada de serviço de seus pacotes. Estas disciplinas afetam diretamente o desempenho da rede tanto em termos de atraso quanto em termos de perda de pacotes e de vazão, uma vez que são elas que priorizam e separam o tráfego de diversas conexões. Mais recentemente, com o surgimento de serviços que requerem níveis pré-determinados de qualidade de serviço (QoS – *Quality of Service*), a escolha dos algoritmos de escalonamento ganhou ainda mais importância, uma vez que estes se tornaram peças fundamentais para garantir o isolamento de QoS entre as várias conexões presentes na rede. Dentre as disciplinas de serviço existentes atualmente, as disciplinas que aproximam o GPS (*Generalized Processor Sharing*) [1] tem obtido bastante êxito em atender a tais pré-requisitos.

Em 1993, Parekh e Gallager [1] demonstraram que empregando servidores GPS nos comutadores da rede, garantias de atraso fim-a-fim poderiam ser providas para uma conexão cujo tráfego tivesse sido policiado por um policiador *Leaky Bucket* [2]. Parekh e Gallager também propuseram um algoritmo de aproximação para o GPS, uma vez que o GPS assume que um servidor pode servir várias conexões simultaneamente e que o tráfego destas conexões pode ser dividido infinitamente. Como isto não é realizável na prática, Parekh e Gallager propuseram uma aproximação baseada em pacotes para o GPS. Esta disciplina foi chamada de *Packet-by-Packet Generalized Processor Sharing* (PGPS) [1], e demonstrou-se idêntica à disciplina *Weighted Fair Queuing* (WFQ) proposta em 1989 por Demers, Keshav e Shenker [3]. Durante um certo período, considerou-se que tanto o PGPS, quanto o WFQ ofereciam serviços quase idênticos ao GPS, com uma diferença igual ao tamanho de um pacote. Entretanto, em 1996 Bennett e Zhang [4] demonstraram que podem haver grandes discrepâncias entre os serviços providos pelas disciplinas WFQ e PGPS com relação ao GPS. De fato, Bennett e Zhang provaram que as disciplinas WFQ e PGPS podem estar bastante distantes do GPS em termos do número de *bits* servidos para uma conexão. A fim de ultrapassar esta limitação, em 1996 Bennett e Zhang desenvolveram um novo algoritmo de aproximação para o GPS, que foi chamado de *Worst-case Fair Weighted Fair Queuing* (WF²Q). Em [4] os autores demonstraram

¹ Também conhecidas por algoritmos de escalonamento ou escalonador.

que o WF²Q prove serviços de forma quase idêntica ao GPS, agora sim com uma diferença máxima igual ao tamanho de um pacote. Eles também demonstraram que o WF²Q possui as mesmas propriedades de atraso máximo (*bounded-delay*) e de divisão justa de pior caso (*worst-case fairness*) que o GPS. Desde então, o WF²Q tem sido considerado um dos algoritmos que mais se aproxima do GPS.

Assim sendo, este artigo apresenta o desenvolvimento de um modelo do escalonador WF²Q para o ambiente de simulação SimNT 2.0 [5]. O SimNT 2.0 foi desenvolvido pelo Departamento de Comunicações (DECOM) da Faculdade de Engenharia Elétrica e de Computação (FEEC) da UNICAMP com o objetivo de permitir a análise integrada de redes de comunicações multiprotocolo, tais como redes ATM e TCP/IP [6]. O objetivo do trabalho é expandir um conjunto de modelos no nível de células [7] para a simulação integrada das funções de gerenciamento de tráfego ATM. Portanto, o modelo desenvolvido além de implementar eficientemente o algoritmo WF²Q, deve ainda interagir com os demais modelos deste conjunto, como por exemplo, com modelos de algoritmos de controle de admissão de conexões. O artigo apresenta também a validação do modelo desenvolvido, bem como um exemplo de aplicação deste modelo no transporte de tráfego de vídeo codificado MPEG-4 [8] sobre uma rede ATM. Embora o modelo do escalonador WF²Q tenha sido desenvolvido para o SimNT 2.0, os detalhes de implementação aqui apresentados poderão ser utilizados para desenvolver modelos de escalonadores WF²Q em outros simuladores ou até mesmo em aplicativos práticos.

O restante do artigo está dividido como segue: a seção II apresenta o funcionamento do escalonador WF²Q; a seção III apresenta o ambiente de simulação SimNT 2.0; a seção IV apresenta o modelo desenvolvido; a seção V apresenta a validação e um exemplo de aplicação do modelo desenvolvido; e finalmente, a seção VI discute algumas considerações finais.

II. ESCALONADOR WF²Q

O escalonador WF²Q é uma aproximação baseada em pacotes do GPS, que tem por objetivo emular este escalonador fluido o mais próximo possível. De acordo com Giroux e Ganti [2], o escalonador WF²Q é classificado como um escalonador com divisão justa e conservativa (*work-conserving fair-share scheduler*). Portanto, sempre que houver células aguardando para serem servidas, no mínimo uma célula será servida a cada *slot* de serviço do escalonador. Em outras palavras, o escalonador nunca é “desligado” enquanto existirem células aguardando para serem servidas.

Consideremos um conjunto de conexões B_{t_k} sendo servidas simultaneamente em um escalonador GPS no instante t_k . Para cada conexão i é atribuído um “peso” ϕ_i , que determina a fração de serviço que a conexão i receberá. Como no GPS cada conexão tem

exatamente um pacote sendo servido, a taxa de serviço instantânea para a conexão i é dada por:

$$r_i = \frac{\phi_i}{\sum_{j \in B_{t_k}} \phi_j} \cdot r \quad (1)$$

onde:

- B_{t_k} é o número de conexões sendo servidas simultaneamente no escalonador no instante t_k .
- ϕ_i é o peso da conexão i no escalonador GPS.
- ϕ_j é o peso da conexão j no escalonador GPS.
- r é a taxa total disponível no escalonador.

O funcionamento do escalonador WF²Q é bastante semelhante ao do escalonador PGPS (ou WFQ), porém com uma diferença na forma em que os pacotes a serem servidos são selecionados. Quando um escalonador PGPS escolhe um pacote para ser servido no instante t_s , ele seleciona, entre todos os pacotes presentes no servidor, o pacote que terminaria serviço primeiro no GPS. Já o escalonador WF²Q, ao invés de escolher o pacote que terminaria serviço primeiro no GPS dentre todos os pacotes presentes no servidor, escolhe somente entre os pacotes que já teriam iniciado serviço no GPS (pacotes elegíveis). Ou seja, no escalonador WF²Q, o pacote escolhido para ser servido no instante t_s é o pacote que terminaria serviço primeiro no GPS e que já teria começado a ser servido pelo GPS no instante t_s .

Tanto o PGPS quanto o WF²Q utilizam o conceito de tempos virtuais [1] para emular o funcionamento do GPS e a partir daí escolher a ordem de serviço dos pacotes presentes no escalonador. Consideremos a chegada de um pacote da conexão i no instante t_k com tamanho igual a $L_{t_k}^i$ bits. No momento da chegada deste pacote, um tempo virtual geral V_{t_k} é incrementado de acordo com a expressão:

$$V_{t_k} = V_{t_{k-1}} + \frac{(t_k - t_{k-1})}{\sum_{j \in B_{t_k}} \phi_j} \quad (2)$$

Onde:

- $V_{t_{k-1}}$ é o tempo virtual geral no instante t_{k-1} .
- B_{t_k} é o número de conexões sendo servidas simultaneamente no escalonador no instante t_k .

Uma vez atualizado o tempo virtual geral, é necessário calcular então os tempos virtuais de início de serviço ($S_{t_k}^i$) e de final de serviço ($F_{t_k}^i$) deste pacote no GPS emulado. O tempo virtual de início de

serviço $S_{t_k}^i$ é calculado de acordo com a expressão:

$$S_{t_k}^i = \max\{F_{t_{k-1}}^i, V_{t_k}\} \quad (3)$$

Onde:

- $F_{t_{k-1}}^i$ é o tempo virtual de final de serviço anterior.

Feito isso calcula-se o tempo virtual de final de serviço $F_{t_k}^i$ do pacote:

$$F_{t_k}^i = S_{t_k}^i + \frac{L_{t_k}^i}{\phi_i} \quad (4)$$

Tanto no escalonador PGPS quanto no escalonador WF²Q, a ordem de serviço dos pacotes é determinada em função dos tempos virtuais $S_{t_k}^i$ e $F_{t_k}^i$. Suponhamos que ambos os escalonadores queiram escolher no instante t_s um pacote para ser servido. Como vimos, no PGPS é servido por primeiro o pacote que terminaria serviço primeiro no GPS, ou seja o pacote que tiver o menor tempo virtual $F_{t_k}^i$, independente do instante de tempo t_s . Já no escalonador WF²Q, é servido por primeiro o pacote elegível que terminaria serviço primeiro no GPS. Ou seja, é servido por primeiro o pacote com menor tempo virtual $F_{t_k}^i$, cujo $S_{t_k}^i$ seja menor que o instante de tempo t_s .

Portanto, o escalonador WF²Q, ao invés de utilizar uma disciplina de seleção de pacotes “menor tempo virtual de final de serviço primeiro” (SFF – *Smallest virtual Finish time First*), utiliza uma disciplina “menor tempo virtual de final de serviço elegível primeiro” (SEFF – *Smallest Eligible virtual Finish time First*). Ou seja, o escalonador WF²Q escolhe entre todos os pacotes elegíveis, o pacote que tem o menor tempo virtual de final de serviço.

III. AMBIENTE DE SIMULAÇÃO

Nesta seção apresentaremos o ambiente de simulação SimNT 2.0 [5], que foi o ambiente escolhido para a implementação do nosso modelo de escalonador WF²Q. O SimNT 2.0 é baseado na técnica de simulação *event-driven* [9], e foi desenvolvido em C++ para o sistema operacional *Windows 9x/NT/2000*TM. A estrutura do SimNT 2.0 pode ser dividida em duas partes principais, como ilustra a Figura 1: programa executável e biblioteca de modelos. O programa executável é a parte principal do SimNT 2.0. Ele possui uma instância do *kernel* do simulador. A biblioteca de modelos é o conjunto de todos os modelos que podem ser utilizados para construir uma rede a ser simulada. No SimNT 2.0 tanto os modelos quanto o *kernel* são implementados como bibliotecas de ligação dinâmica (DLLs –

Dynamic Link Libraries) [10] do sistema operacional *Windows 9x/NT/2000*TM.

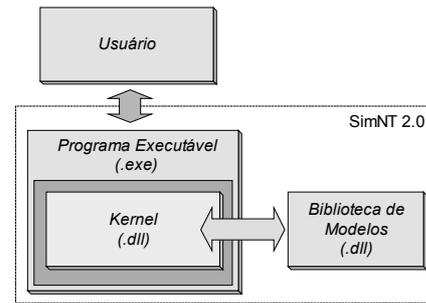


Figura 1. Estrutura do SimNT 2.0.

III.1. BIBLIOTECA DE MODELOS

O SimNT 2.0 permite que os seus usuários implementem e incorporem novos modelos à biblioteca do ambiente de simulação sem a necessidade de modificar o kernel do simulador. Além deste recurso, o SimNT 2.0 permite a construção hierárquica de modelos, ou seja a construção de modelos a partir do agrupamento de outros modelos. Para tanto, o SimNT 2.0 possui uma estrutura hierárquica de modelos, como ilustra a Figura 2.

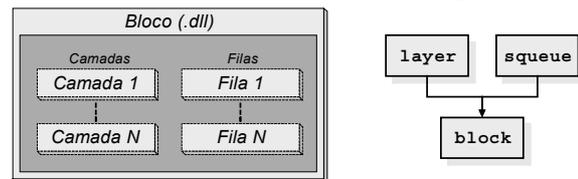


Figura 2. Estrutura hierárquica de modelos no SimNT 2.0.

Esta estrutura baseia-se em três classes base [11]: Block, Layer e Squeue. As classes Layer e Squeue constituem um primeiro nível desta estrutura hierárquica, enquanto a classe Block constitui um segundo nível. Portanto, no SimNT 2.0 os modelos são construídos a partir da derivação das classes base Block, Layer e Squeue, respectivamente. Assim sendo, um bloco pode ou não possuir várias instâncias de camadas e filas, que chamaremos de modelos internos de um bloco. É importante observar que apenas os blocos serão acrescentados a biblioteca de modelos do simulador no formato de DLLs do *Windows*.

No SimNT 2.0 parâmetros são utilizados para configurar ou modificar o comportamento dos modelos, podendo armazenar desde valores escalares, até valores vetoriais e matriciais.

IV. MODELO DESENVOLVIDO

Como vimos na seção II, o WF²Q serve por primeiro o pacote com menor tempo virtual $F_{t_k}^i$ cujo $S_{t_k}^i$ seja menor que o instante de tempo de serviço real t_s . Para implementarmos esta disciplina de seleção de pacotes, utilizamos estruturas de dados do tipo filas com prioridades [11]. Tais filas possuem desempenho ótimo de inserção e de retirada de objetos, e foram

implementadas utilizando árvores binárias balanceadas [11]. Duas filas com prioridades foram utilizadas: fila com prioridades para o $F_{t_k}^i$ e fila com prioridades para o $F_{t_k}^i$ temporária. Uma vez que nosso modelo de escalonador WF²Q foi desenvolvido voltado para a simulação de redes ATM, ambas as filas com prioridades são utilizadas para ordenar células ATM conforme os seus tempos virtuais de final de serviço $F_{t_k}^i$. A Figura 3 mostra a estrutura do modelo desenvolvido.

As células são armazenadas na fila com prioridades por $F_{t_k}^i$ onde aguardam para serem servidas. No momento em que uma célula é servida, é verificado se a célula no topo da fila é elegível. Se a célula escolhida for elegível, ela é servida. Caso contrário, ela é armazenada na fila com prioridades por $F_{t_k}^i$ temporária, e uma nova célula é verificada. O processo prossegue até que uma célula elegível seja escolhida ou até que a fila com prioridades fique vazia. Após uma célula ter sido escolhida, as células não elegíveis que se encontram na fila temporária são retornadas a fila principal. A seguir veremos em maiores detalhes o funcionamento do modelo desenvolvido.

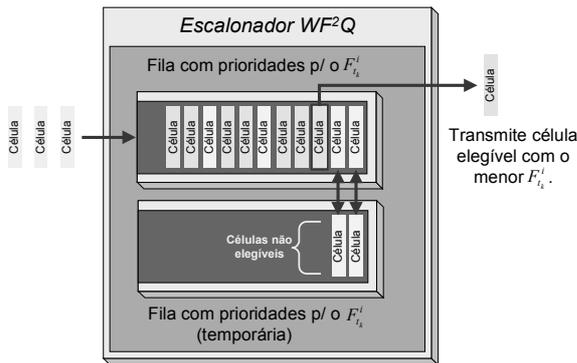


Figura 3. Estrutura do modelo do escalonador WF²Q.

IV.1. FUNCIONAMENTO DETALHADO

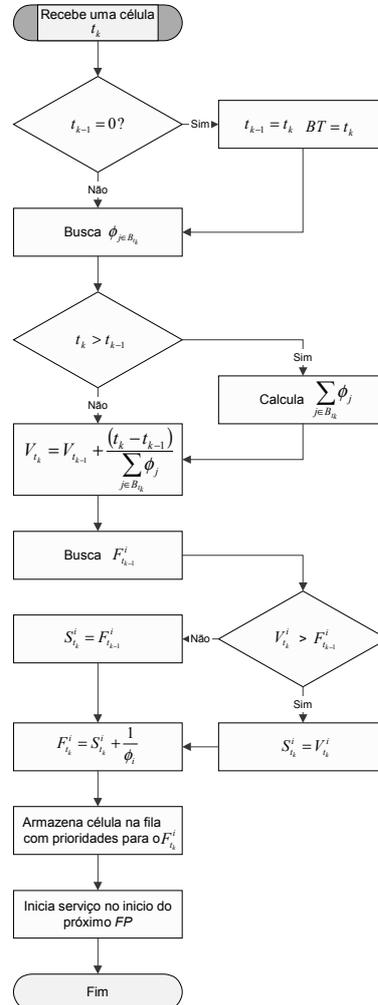
O modelo desenvolvido possui dois algoritmos principais: algoritmo de armazenamento de células ATM e algoritmo de serviço de células ATM.

IV.1.1. ALGORITMO DE ARMAZENAMENTO DE CÉLULAS ATM

A Figura 4 mostra o fluxograma do algoritmo de armazenamento de células ATM. Quando uma célula ATM é recebida (instante de tempo t_k ²), o modelo do escalonador verifica se o tempo de chegada da célula anterior (t_{k-1}) é igual a zero. Se esta variável for igual a zero, ela é configurada com o valor de t_k . Uma

² Quando a primeira célula ATM é recebida $k=1$, $F_{t_0}^i = 0$, $S_{t_0}^i = 0$ e $V_{t_0} = 0$.

variável chamada *BT (Begin Time)* é utilizada para marcar o instante de tempo em que pelo menos uma conexão está presente no escalonador. Em outras palavras, o *BT* marca o início de um período de atividade. Se t_{k-1} for igual a zero, a variável *BT* também é configurada com o valor de t_k .



- t_k - Tempo de chegada da célula atual
- t_{k-1} - Tempo de chegada da célula anterior
- ϕ_i - Peso para a conexão i
- $V_{t_{k-1}}$ - Tempo virtual para o instante t_{k-1}
- V_{t_k} - Tempo virtual para o instante t_k
- B_{t_k} - Conjunto de conexões ativas em t_k
- $F_{t_{k-1}}^i$ - Tempo virtual de final de serviço para a conexão i em t_{k-1}
- $F_{t_k}^i$ - Tempo virtual de final de serviço para a conexão i em t_k
- $S_{t_k}^i$ - Tempo virtual de início de serviço para a conexão i em t_k
- FP - Período de um frame de serviço
- BT - Início de um período de atividade no escalonador

Figura 4. Algoritmo de armazenamento de células ATM.

Ainda na Figura 4, o modelo do escalonador WF²Q busca o valor dos pesos para as conexões do conjunto B_{t_k} . Feito isso, é verificado se a variável t_k é maior que a variável t_{k-1} . Se esta condição for verdadeira, o somatório dos pesos de todas as conexões ativas é recalculado. Caso contrário, o mesmo valor obtido anteriormente é utilizado. Em seguida é atualizado o valor do tempo virtual do sistema (V_{t_k}). Também é lido o valor do tempo virtual

de final de serviço ($F_{t_{k-1}}^i$) para a conexão i no instante t_{k-1} .

Se o tempo virtual V_{t_k} for maior que o $F_{t_{k-1}}^i$, o valor do tempo virtual de início de serviço ($S_{t_k}^i$) para a conexão i é tomado como sendo igual ao V_{t_k} . Caso contrário, o valor do tempo virtual de início de serviço para a conexão i é tomado como sendo igual ao $F_{t_{k-1}}^i$. Então, o $F_{t_k}^i$ é atualizado para o valor do $S_{t_k}^i$ acrescido de $1/\phi_i$. A célula é então armazenada na fila com prioridades para o $F_{t_k}^i$, onde aguarda pelo serviço.

IV.1.2. ALGORITMO DE SERVIÇO DE CÉLULAS ATM

A Figura 5 mostra o fluxograma do algoritmo de serviço de células ATM. Quando uma primeira célula ATM é armazenada na fila com prioridades para o $F_{t_k}^i$, automaticamente o modelo do escalonador é “ligado” e o serviço de células começa no início do próximo *slot* de serviço. Uma única célula é servida no período de um *slot* de serviço (FP – *Frame Period*). Consideremos o caso em que uma célula ATM será servida no instante t_S . Inicialmente, é executado um laço na fila com prioridades para o $F_{t_k}^i$. O objetivo deste laço é encontrar a célula elegível com menor $F_{t_k}^i$. Para verificar a elegibilidade de uma célula, foi necessário desenvolver uma expressão que permitisse comparar o seu tempo virtual de início de serviço $S_{t_k}^i$ no sistema GPS com o tempo real deserviço t_S . A seguinte expressão foi desenvolvida para possibilitar tal comparação:

$$S_{t_k}^i \cdot FP + BT \leq t_S \quad (5)$$

Uma vez que o equacionamento do escalonador WF²Q foi desenvolvido considerando-se uma taxa de serviço normalizada de 1 célula por segundo, a multiplicação do tempo virtual de início de serviço $S_{t_k}^i$ pelo período de um *slot* de serviço ATM torna este tempo virtual compatível com a taxa de serviço FP do modelo do escalonador WF²Q. O acréscimo do tempo BT é necessário para reajustar o escalonador após um período de inatividade.

Portanto, a cada iteração do laço, o tempo virtual $S_{t_k}^i$ da célula que se encontra no topo da fila com prioridades para o $F_{t_k}^i$ é lido e a expressão (5) é utilizada para verificar a elegibilidade desta célula. Se a expressão (5) for verdadeira, a célula é considerada elegível, sendo portanto removida da fila $F_{t_k}^i$ e servida. Neste caso a execução do laço acaba.

Entretanto, se a expressão (5) não for verdadeira, a célula é considerada ilegível, sendo portanto removida da fila $F_{t_k}^i$ e armazenada na fila para o $F_{t_k}^i$ temporária, onde aguarda para ser re-inserida na fila $F_{t_k}^i$ principal. Neste caso, o laço prossegue verificando a elegibilidade das demais células armazenadas na fila $F_{t_k}^i$ principal até que uma célula elegível seja encontrada ou até que esta fila principal fique vazia.

Como vimos, o escalonador WF²Q é um escalonador com divisão justa e conservativa, portanto, se nenhuma célula da fila com prioridades $F_{t_k}^i$ principal for elegível para ser servida no instante t_S , o escalonador deverá escolher igual uma célula para ser servida. Neste caso será servida a célula com menor $F_{t_k}^i$ que se encontra na fila com prioridades temporária.

Depois de escolhida a célula a ser servida, é verificado se restaram células nas filas com prioridades principal e temporária. Se ambas as filas ficaram vazias, as variáveis k , t_{k-1} , V_{t_k} e $F_{t_{k-1}}^{j \in B_{t_k}}$ são reinicializadas e o escalonador é “desligado”. Caso contrário, é agendando o serviço de uma próxima célula ATM no instante $t_S + FP$. Neste ponto outro laço é executado. O objetivo é retornar as células ilegíveis não servidas de volta para a fila principal.

IV.2. ITERAÇÃO COM OUTROS MODELOS ATM

O modelo do escalonador WF²Q faz parte de um conjunto de modelos para a simulação de redes ATM no nível de células [7]. A arquitetura deste conjunto de modelos (veja Figura 6) é baseada em quatro blocos: aplicativo (*App* – *Application*), terminal faixa larga (BTE – *Broadband Terminal Equipment*), chaveador ATM (SW – *Switch*) e gerente (*Manager*). O aplicativo é um modelo dos clientes da rede ATM. Ele permite gerenciar conexões chaveadas ATM (SVCs – *Switched Virtual Connections*), configurar contratos de tráfego ATM e transmitir dados através da rede. O terminal faixa larga é um modelo de um dispositivo de borda da rede ATM, como por exemplo um cartão de interface com a rede (NICs – *Network Interface Cards*). O chaveador é um modelo de dispositivo de comutação de células da rede ATM. O gerente, como o próprio nome já diz, é um modelo de um gerente centralizado da rede. Ele é responsável pelo roteamento, estabelecimento e remoção de conexões da rede. A arquitetura do conjunto de modelos no nível de células ATM é implementada no simulador através de blocos, camadas e filas, que mais convenientemente resolvemos chamar de gerenciadores de tráfego. Nesta arquitetura, o modelo desenvolvido para o escalonador WF²Q é implementado como um gerenciador de tráfego, sendo portanto derivado da classe base *SQueue*. Assim sendo, o modelo do escalonador WF²Q foi

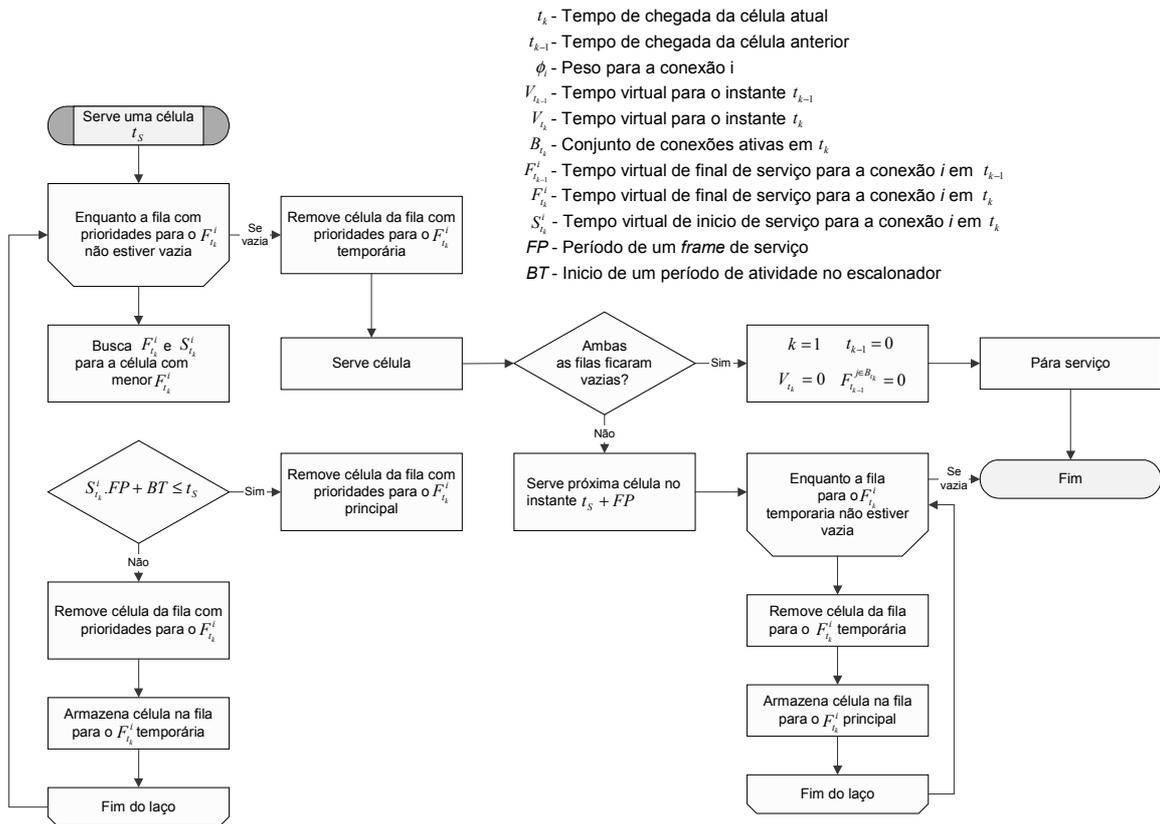


Figura 5. Algoritmo de serviço de células ATM.

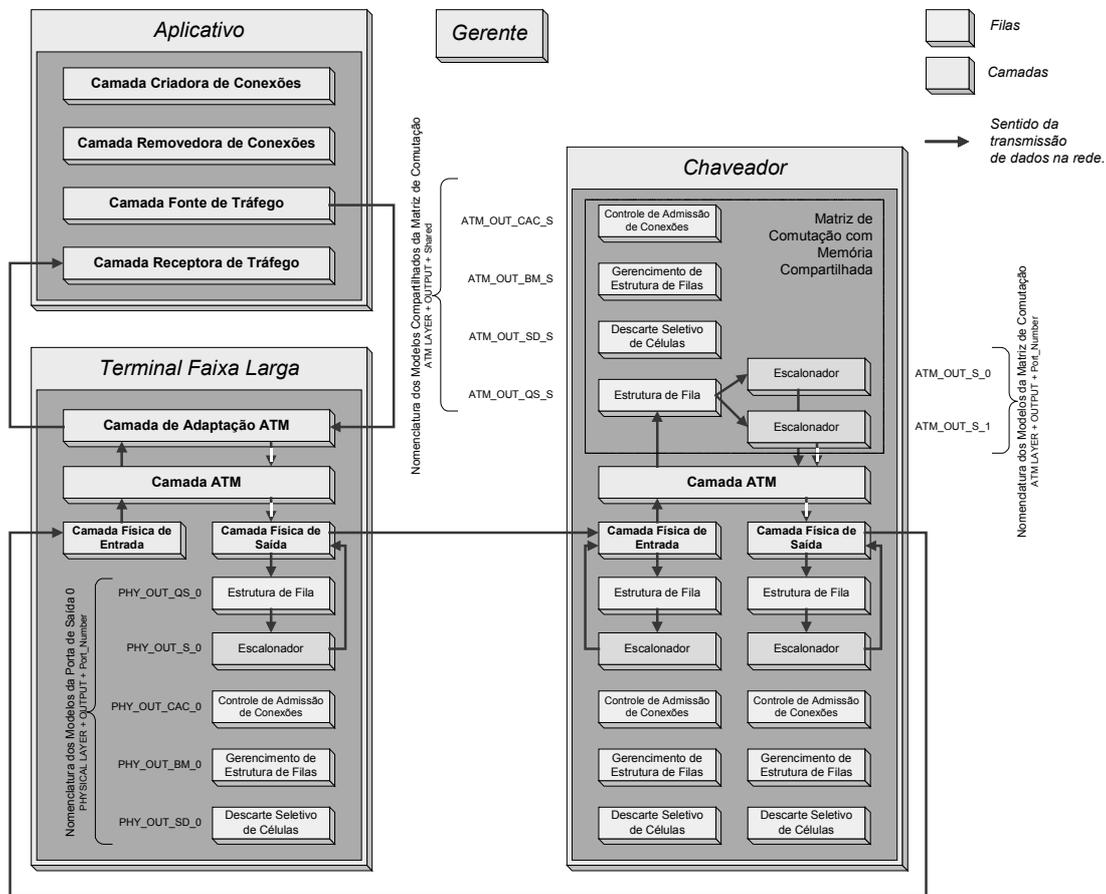


Figura 6. Arquitetura do conjunto de modelos no nível de células ATM.

desenvolvido como um modelo interno dos blocos terminal faixa larga e chaveador ATM, onde é utilizado para definir a ordem de serviço das células armazenadas nas estruturas de filas [2] associadas as camadas físicas e ATM destes modelos de equipamento. Cabe aqui uma observação: para que fosse possível trocar independentemente os modelos de escalonadores e os modelos de estruturas de filas dos equipamentos ATM, no conjunto de modelos no nível de células, as células ATM que são encaminhadas para os modelos de escalonadores também são encaminhadas para os modelos de estruturas de filas. Desta forma, o modelamento do serviço das células ATM é feito no escalonador e o modelamento da espera das células pelo serviço é feito na estrutura de filas.

A Figura 6 mostra também a nomenclatura deste conjunto de modelos. Observe que os modelos são denominados de acordo com o nome da camada a que estão associados, seguido por um indicador da posição da porta e do número da porta a que estão associados. Esta regra é válida para os modelos associados à camada física. Na camada ATM, existem alguns modelos que são compartilhados por todas as portas de saída. Para diferenciar estes modelos dos demais, substitui-se o número da porta associada por um "S", indicando que estes modelos são modelos compartilhados (*shared*).

IV.3. PARÂMETROS

O modelo do escalonador WF²Q possui os seguintes parâmetros:

- *Scheduler_Capacity* – Taxa de serviço em células por segundos.
- *Event_Sampling* – Estado individual das variáveis estatísticas a serem amostradas na amostragem por eventos. Neste tipo de amostragem, o valor das variáveis estatísticas é salvo para arquivo toda vez que ocorre um evento significativo, como por exemplo a chegada ou serviço de uma célula ATM.
- *Time_Sampling* – Estado individual das variáveis estatísticas a serem amostradas na amostragem por tempo. Neste tipo de amostragem, o valor das variáveis estatísticas é salvo de acordo com um intervalo de tempo pré-definido.
- *Interval_Between_Time_Samples* – Intervalo entre amostragens por tempo.

V. APLICANDO O MODELO DESENVOLVIDO NO TRANSPORTE DE TRÁFEGO MPEG-4 SOBRE UMA REDE ATM

Para demonstrarmos o modelo desenvolvido, simularemos o transporte de seqüências de tráfego MPEG-4 [8] sobre uma rede ATM. Para tanto, utilizaremos o simulador de redes de comunicação SimNT 2.0 [5] e o conjunto de modelos para a simulação de redes ATM no nível de células,

previamente apresentado em [7]. Utilizaremos também os resultados obtidos em [12].

V.1. MPEG-4

Atualmente, o MPEG-4 [8] está sendo considerado o padrão mais indicado para as aplicações de vídeo iterativo que utilizarão as futuras redes de telecomunicações faixa larga. Neste exemplo de aplicação, utilizaremos cinco seqüências de *frame size* MPEG-4 disponibilizadas pelo Grupo de Redes de Telecomunicações da Universidade Técnica de Berlin [13] para investigar o desempenho do modelo desenvolvido quando submetido ao tráfego de vídeo codificado MPEG-4. As seqüências utilizadas foram:

- Parque dos Dinossauros (*Jurassic Park*)
- Silêncio dos Inocentes (*Silence of The Lambs*)
- Programa de Entrevistas (*Boulevard Bio*)
- Corrida de Fórmula 1 (*Formula 1*)
- Desenho Animado (*Simpsons*)

Entretanto, antes de simular o transporte destas seqüências, foi necessário adaptá-las do nível de tamanho de *frames* MPEG (Fluxo de Vídeo Elementar) [14] para o nível de pacotes de fluxo de transporte MPEG (MPEG TS – *Transport Stream*). Esta adaptação foi necessária, porque de acordo com a estrutura do padrão MPEG os pacotes entregues à rede de transporte devem ser formatados como fluxos de transporte MPEG. Para resolver este problema, em [12] nós desenvolvemos um *software* baseado na especificação de Vídeo sobre Demanda v1.1 do ATM Fórum [15] e no esquema de encapsulamento de pacotes de fluxo elementar desenvolvido por Mehaoua

Descritores de Tráfego

Seqüência	Descritores			
	PCR (células/seg.)	CDVT (seg.)	SCR (células/seg.)	MBS (células)
Parque dos Dinossauros	1198.3763	0.000830	958.7010	51
Silêncio dos Inocentes	1600	0.000625	1280	107
Programa de Entrevistas	925	0.001081	740	41
Corrida de Fórmula 1	1000	0.001	800	66
Desenho Animado	3259.5052	0.000307	2607.6041	76

Tabela 1. Descritores de tráfego ATM utilizados.

et. al [16]. Em [12], nós também implementamos uma técnica para a estimativa de descritores de tráfego ATM visando o transporte de tráfego de vídeo codificado MPEG sobre ATM. Portanto, neste exemplo de aplicação utilizaremos os descritores ATM estimados de acordo com a técnica do *buffer virtual* apresentada em [12]. A Tabela 1 sumariza os descritores de tráfego estimados para as seqüências de tamanho de pacotes de transporte MPEG-4.

V.2. SIMULAÇÕES

A Figura 7 mostra a topologia da rede simulada. Fazem parte desta rede dez aplicativos fonte, dois terminais faixa larga, um chaveador ATM, um gerente e um aplicativo de destino. Cada um dos aplicativos fonte (App_0 até App_9) estabelece uma conexão chaveada até o aplicativo de destino (App_D). Os cinco primeiros aplicativos (App_0 até App_4) transmitem respectivamente as cinco seqüências de tráfego de pacotes MPEG-4 apresentadas anteriormente. Os cinco aplicativos restantes (App_5 até App_9), também transmitem estas mesmas seqüências. Todas as conexões utilizam a categoria de serviço nrt-VBR (*Non Real-time Variable Bit Rate*) e os descritores de tráfego ATM são configurados de acordo com a Tabela 1. O CLR utilizado em cada conexão é incrementado a partir de 1×10^{-9} (a conexão a partir do App_0 possui $CLR = 1 \times 10^{-9}$, enquanto conexão a partir do App_9 possui $CLR = 10 \times 10^{-9}$). Os enlaces entre o terminal BTE_0 e o chaveador SW_0 e entre o chaveador SW_0 e o terminal BTE_1 tem a capacidade de 2.048 Mbts/seg. A taxa de serviço da matriz de comutação em cada porta do chaveador SW_0 foi configurada para 149.76 Mbts/seg. A distância entre o BTE_0 e o SW_0 é de 10 metros, e entre o SW_0 e o BTE_1 é de 80 metros. A velocidade de propagação do sinal é de 200×10^6 metros/seg.

Tomando como referência a Figura 6, os seguintes modelos de gerenciadores de tráfego foram utilizados nos terminais BTE_0 e BTE_1, e no chaveador SW_0:

- **Estrutura de Filas** – Foi utilizado somente o modelo estrutura de filas com filas por conexão (*Per-VC Queuing Structure*) [2]. Este modelo mantém filas lógicas para cada conexão existente na rede. Estas filas são servidas por um modelo de escalonador associado.

- **Escalonador** – Foram utilizados dois modelos de escalonadores: escalonador FCFS – *First Coming First Served* e escalonador WF²Q. O modelo de escalonador FCFS serve as células na mesma ordem em que são recebidas.
- **Controle de Admissão de Conexões** – Foi utilizado o modelo controle de admissão de conexões com alocação baseada em critérios de banda e *buffer* efetivos. Este modelo foi desenvolvido a partir do trabalho de Elwalid et al. [17]. Ele aceita uma nova conexão se a banda e o *buffer* efetivos calculados para uma nova conexão forem menores que a largura de faixa disponível no escalonador e o espaço disponível na estrutura de filas, respectivamente. Neste caso, o modelo de controle de admissão com alocação baseada em critérios de banda e *buffer* efetivos ajusta o peso desta conexão de acordo com a expressão $\phi_i = EB/C$, onde *EB* é a banda efetiva calculada para a conexão *i* e *C* é a capacidade do escalonador.
- **Gerenciador de Estrutura de Filas** – Foi utilizado o modelo gerenciador de estrutura de filas com particionamento dinâmico (*Dynamic Partioning*). Este modelo foi desenvolvido a partir do trabalho de Krishnan et.al [18]. O gerenciador de estrutura de filas com particionamento dinâmico trabalha em parceria com o modelo de controle de admissão de conexões anterior. Ele mantém vários limiares dinâmicos (um para cada fila por conexão) que são utilizados para determinar se uma célula pode ser aceita ou não em uma estrutura de filas. Estes limiares são calculados a partir das alocações de banda e *buffer* efetivos feitos pelo controle de admissão de conexões.

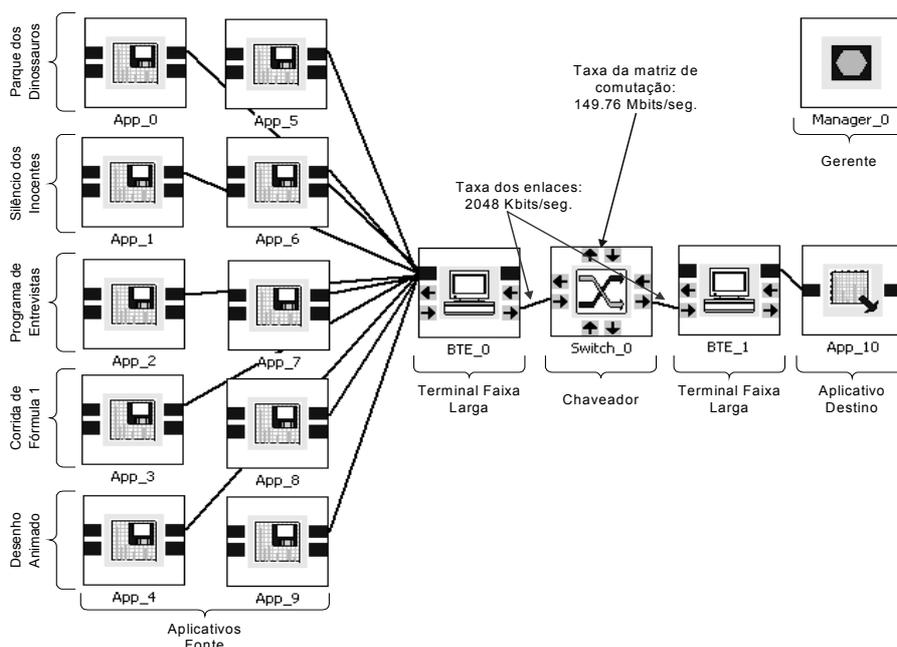


Figura 7. Topologia da rede simulada.

- **Descarte Seletivo de Células** – Foi utilizado o modelo descarte seletivo de células baseado no CLR (*Cell Loss Ratio*) [2]. Em caso de congestionamento, este modelo descarta as células das conexões com menor pré-requisito de CLR em prol das células das conexões com um maior pré-requisito de CLR.

Para avaliarmos o desempenho do modelo do escalonador WF²Q quando sujeito a um tráfego MPEG-4, realizamos 8 simulações para cada tipo de amostragem estatística, variando desde a capacidade das estruturas de filas (1000, 500, 100 e 50 células) até o modelo de escalonador utilizado. Ou seja, simulamos a rede da Figura 7 utilizando primeiro o escalonador FCFS e depois o escalonador WF²Q. A idéia é mostrar o desempenho do escalonador WF²Q tendo como referência o escalonador FCFS. A variação da capacidade das estruturas de filas revela a importância da ordem de serviço das células ATM na qualidade de serviço garantida para cada conexão.

V.3. RESULTADOS

Inicialmente é apresentada a validação do modelo desenvolvido, seguido pelos resultados das simulações da rede da Figura 7. Estes resultados foram divididos de acordo com os dois tipos de amostragem disponíveis (veja seção 0): amostragem por eventos e amostragem por tempo.

V.3.1. VALIDAÇÃO DO MODELO

A validação do modelo será feita a partir da análise da ordem de serviço obtida na estrutura de filas PHY_OUT_QS_0 do BTE_0 no intervalo de tempo de 0.04 até 0.043 segundos. A fim de facilitar a compreensão dos acontecimentos, a Tabela 2 mostra os pesos ϕ_i alocados pelo modelo de controle de admissão de conexões para cada conexão i , bem como o CLR definido para cada conexão. Observe que as conexões estão ordenadas na ordem decrescente de seus pesos ϕ_i . Na Figura 8 é mostrada a ocupação

obtida na estrutura de filas PHY_OUT_QS_0 durante o intervalo de tempo considerado. Na Tabela 3 é mostrada a evolução das variáveis do modelo neste mesmo intervalo de tempo.

No instante 0.04 segundos, 10 células ATM são armazenadas para serviço no escalonador WF²Q (uma de cada conexão). A primeira célula servida pertence a conexão 4. Esta conexão, junto com a conexão 9, é a conexão com maior peso no escalonador. Como ambas as conexões possuem o mesmo peso ϕ_i , o desempate é feito baseado na ordem de chegada das células. Em seguida são servidas as células da conexão 9 e da conexão 1, respectivamente. No instante 0.0407702310 ocorre a chegada de mais 2 células,

SEQUÊNCIA	i	ϕ_i	CLR
Desenho Animado	4	0.5398547053	5×10^{-9}
Desenho Animado	9	0.5398547053	10×10^{-9}
Silêncio dos Inocentes	1	0.2650000200	2×10^{-9}
Silêncio dos Inocentes	6	0.2650000200	7×10^{-9}
Parque dos Dinossauros	0	0.1984810867	1×10^{-9}
Parque dos Dinossauros	5	0.1984810867	6×10^{-9}
Corrida de Fórmula 1	3	0.1656250204	4×10^{-9}
Corrida de Fórmula 1	8	0.1656250204	9×10^{-9}
Programa de Entrevistas	2	0.1532031455	3×10^{-9}
Programa de Entrevistas	7	0.1532031455	8×10^{-9}

Tabela 2. Pesos e CLR para cada conexão.

ocorre a chegada de mais 2 células, uma da conexão 1 e outra da conexão 6. Apesar da chegada destas duas células, a próxima célula servida será a célula da conexão 6 que chegou no instante 0.04. Isto porque neste instante esta célula é que possui o menor $F_{t_k}^i$. Na sequência é servida a célula da conexão 0 que chegou no instante 0.04. Embora esta conexão tenha um peso menor do que a conexão 1, que teve uma nova célula recebida no instante 0.0407702310, ela possui um $F_{t_k}^i$ menor do que o desta conexão. No instante 0.041001, ocorre a chegada de mais 2 células,

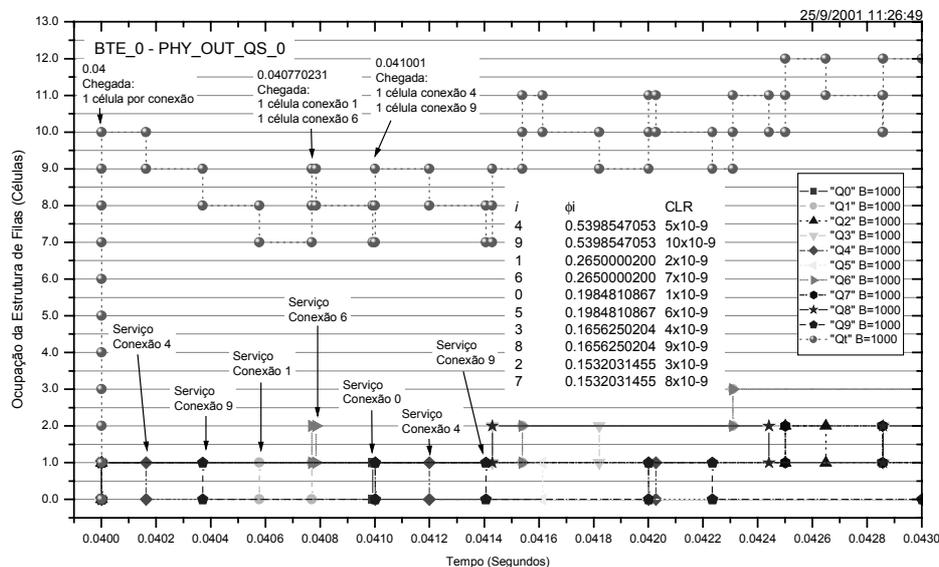


Figura 8. Ocupação da estrutura de filas do BTE_0 para o modelo de escalonador WF²Q.

uma da conexão 4 e outra da conexão 9. Neste caso, entretanto, devido ao peso destas conexões, ambas as células recém chegadas são servidas antes do que as demais células que aguardam por serviço no escalonador. Os acontecimentos restantes da Figura 8 podem ser compreendidos a partir da observação

do tempo de chegada t_k e do $F_{t_k}^i$ de cada célula na Tabela 3. Concluímos portanto, que os resultados apresentados nesta subseção estão de acordo com a referência [4], validando assim o modelo desenvolvido.

t_k	t_{k-1}	$\sum \phi_i$	V_{t_k}	$F_{t_{k-1}}^i$	$S_{t_k}^i$	ϕ_i	$F_{t_k}^i$	i	BT
0.0400010000	0.0400010000	0.1984810867	0	0	0	0.1984810867	5.0382634262	0	0.0400010000
0.0400010000	0.0400010000	0.4634811067	0	0	0	0.2650000200	3.7735846208	1	0.0400010000
0.0400010000	0.0400010000	0.6166842522	0	0	0	0.1532031455	6.5272811269	2	0.0400010000
0.0400010000	0.0400010000	0.7823092726	0	0	0	0.1656250204	6.0377351043	3	0.0400010000
0.0400010000	0.0400010000	1.3221639779	0	0	0	0.5398547053	1.8523502531	4	0.0400010000
0.0400010000	0.0400010000	1.5206450646	0	0	0	0.1984810867	5.0382634262	5	0.0400010000
0.0400010000	0.0400010000	1.7856450846	0	0	0	0.2650000200	3.7735846208	6	0.0400010000
0.0400010000	0.0400010000	1.9388482301	0	0	0	0.1532031455	6.5272811269	7	0.0400010000
0.0400010000	0.0400010000	2.1044732505	0	0	0	0.1656250204	6.0377351043	8	0.0400010000
0.0400010000	0.0400010000	2.6443279558	0	0	0	0.5398547053	1.8523502531	9	0.0400010000
0.0407702310	0.0400010000	1.5646185452	0.0004916412	3.7735846208	3.7735846208	0.2650000200	7.5471692417	1	0.0407702310
0.0407702310	0.0407702310	1.5646185452	0.0004916412	3.7735846208	3.7735846208	0.2650000200	7.5471692417	6	0.0407702310
0.0410010000	0.0407702310	1.9059921638	0.0006127168	1.8523502531	1.8523502531	0.5398547053	3.7047005063	4	0.0410010000
0.0410010000	0.0410010000	2.4458468691	0.0006127168	1.8523502531	1.8523502531	0.5398547053	3.7047005063	9	0.0410010000
0.0414295710	0.0410010000	1.3661374585	0.0009264268	6.0377351043	6.0377351043	0.1656250204	12.0754702087	3	0.0414295710
0.0414295710	0.0414295710	1.3661374585	0.0009264268	6.0377351043	6.0377351043	0.1656250204	12.0754702087	8	0.0414295710
0.0415394620	0.0414295710	1.3661374585	0.0010068660	7.5471692417	7.5471692417	0.2650000200	11.3207538625	1	0.0415394620
0.0415394620	0.0415394620	1.3661374585	0.0010068660	7.5471692417	7.5471692417	0.2650000200	11.3207538625	6	0.0415394620
0.0420010000	0.0415394620	1.7075110771	0.0012771647	3.7047005063	3.7047005063	0.5398547053	5.5570507594	4	0.0420010000
0.0420010000	0.0420010000	2.2473657824	0.0012771647	3.7047005063	3.7047005063	0.5398547053	5.5570507594	9	0.0420010000
0.0423086920	0.0420010000	1.1676563718	0.0015406771	11.3207538625	11.3207538625	0.2650000200	15.0943384834	1	0.0423086920
0.0423086920	0.0423086920	1.1676563718	0.0015406771	11.3207538625	11.3207538625	0.2650000200	15.0943384834	6	0.0423086920
0.0425010000	0.0423086920	1.1676563718	0.0017053728	6.5272811269	6.5272811269	0.1532031455	13.054522538	2	0.0425010000
0.0425010000	0.0425010000	1.1676563718	0.0017053728	6.5272811269	6.5272811269	0.1532031455	13.054522538	7	0.0425010000
0.0428581430	0.0425010000	1.1676563718	0.0020112359	12.0754702087	12.0754702087	0.1656250204	18.1132053130	3	0.0428581430
0.0428581430	0.0428581430	1.1676563718	0.0020112359	12.0754702087	12.0754702087	0.1656250204	18.1132053130	8	0.0428581430
0.0430010000	0.0428581430	1.7075110771	0.0020948998	5.5570507594	5.5570507594	0.5398547053	7.4094010126	4	0.0430010000
0.0430010000	0.0430010000	2.2473657824	0.0020948998	5.5570507594	5.5570507594	0.5398547053	7.4094010126	9	0.0430010000
0.0430779230	0.0430010000	1.7075110771	0.0021399496	15.0943384834	15.0943384834	0.2650000200	18.8679231042	1	0.0430779230
0.0430779230	0.0430779230	1.7075110771	0.0021399496	15.0943384834	15.0943384834	0.2650000200	18.8679231042	6	0.0430779230
0.0438471540	0.0430779230	1.1676563718	0.0027987316	18.8679231042	18.8679231042	0.2650000200	22.6415077251	1	0.0438471540
0.0438471540	0.0438471540	1.1676563718	0.0027987316	18.8679231042	18.8679231042	0.2650000200	22.6415077251	6	0.0438471540

Tabela 3. Evolução das variáveis do modelo durante a execução do algoritmo de armazenamento de células ATM.

V.3.2. AMOSTRAGEM POR EVENTOS

As simulações com amostragens por eventos tiveram uma duração de 0.25 segundos. Novamente, concentraremos nossa análise no BTE_0. Consideremos o intervalo de tempo entre os instantes 0.04 e 0.15 segundos, que é o intervalo de tempo em que acontece o congestionamento dos recursos da rede. Na Figura 9 mostramos o atraso médio das células de cada conexão neste intervalo. Podemos nitidamente ver que o escalonador WF²Q oferece um melhor isolamento de atraso entre as conexões do que o escalonador FCFS. E mais, células das conexões de maior peso (4 e 9) que no escalonador FCFS tiveram o maior atraso, no escalonador WF²Q tiveram o menor atraso dentre todas as conexões. Portanto, podemos observar que o escalonador WF²Q é capaz de manter garantias de qualidade de serviço para cada conexão da rede, independente da presença de outras conexões. Entretanto, a qualidade de serviço oferecida pelo escalonador WF²Q depende de uma configuração cuidadosa dos pesos ϕ_i para cada conexão.

A Figura 10 mostra o número de células perdidas na entrada da estrutura de filas PHY_OUT_QS_0 do BTE_0 e a Figura 11 mostra a taxa de perda de células obtida. A taxa de perda de células depende fundamentalmente do CLR configurado para cada conexão (veja Tabela 2), uma vez que o modelo de algoritmo de descarte seletivo utilizado descarta primeiro as células cuja conexão tem maior CLR configurado. A partir da Figura 11, e considerando o escalonador FCFS, podemos observar que quando a capacidade da estrutura de filas é igual a 100 células, são descartadas células das conexões 4, 9 e 6. Isto

ocorre porque a quantidade de células nas filas lógicas destas conexões neste instante de tempo é maior do que a de outras conexões, uma vez que o escalonador FCFS atende todas as conexões sem diferenciação.

Quando a capacidade da estrutura de filas é igual a 50 células, praticamente são descartadas células de todas as conexões, sendo que as conexões 9 e 8 são as mais prejudicadas. Por outro lado, quando se utiliza o escalonador WF²Q o cenário muda drasticamente. As filas para as conexões de maior peso (com exceção das filas para as conexões 1 e 6) mantêm-se com tamanhos máximos inferiores do que quando se usa o escalonador FCFS. Portanto o CLR observado para estas conexões quando se usa o escalonador WF²Q é bastante reduzido (para a conexão 9 reduziu-se de 13 células para 1 célula descartada). E mais, o número de conexões que tem células descartadas também diminui.

V.3.3. AMOSTRAGEM POR TEMPO

As simulações com amostragens por tempo tiveram uma duração de 60 segundos. Para as simulações com amostragem por tempo os gráficos das figuras a seguir têm como eixo horizontal a capacidade de armazenamento da estrutura de filas PHY_OUT_QS_0 do BTE_0. A Figura 12 mostra a ocupação média e o atraso médio das células nesta estrutura de filas. Podemos observar que após 60 segundos de simulação a ocupação média das filas para cada conexão no escalonador FCFS é muito mais próxima umas das outras do que no escalonador WF²Q, refletindo, portanto a diferenciação de serviços existente no escalonador WF²Q. A mesma observação é válida para o atraso médio. A Figura 13 mostra a

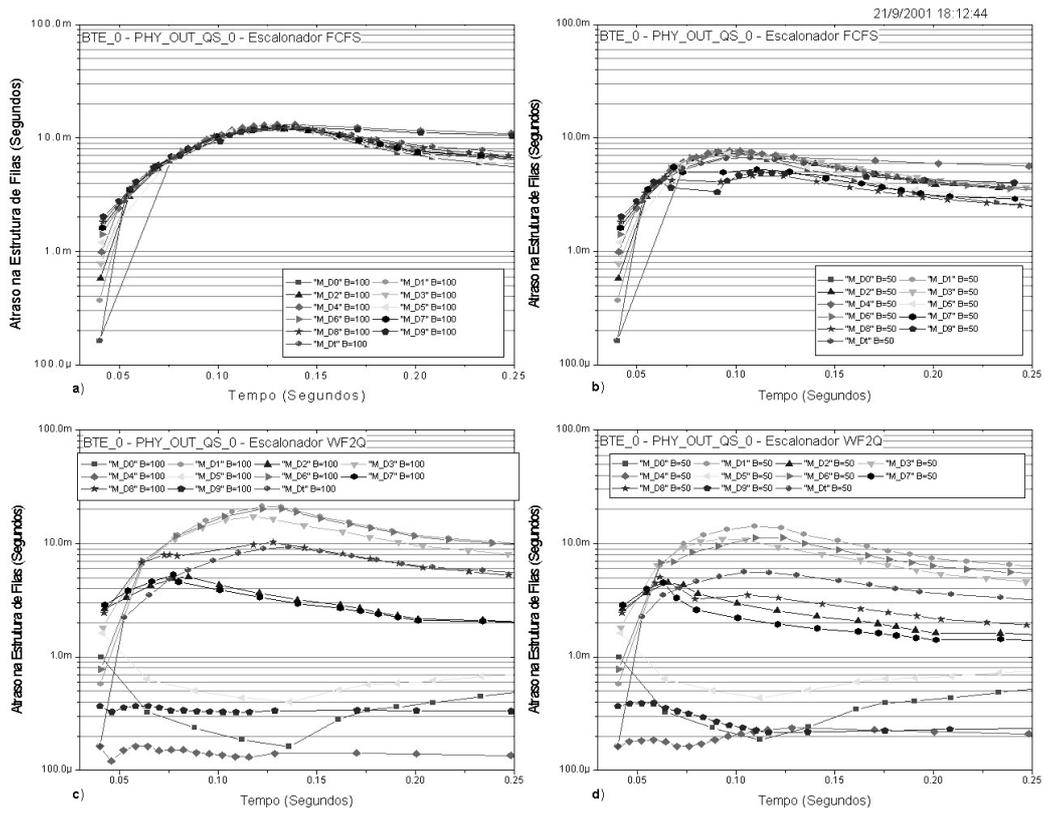


Figura 9. Atraso médio na estrutura de filas do BTE_0 com capacidades de 100 e 50 células para ambos os modelos de escalonadores.

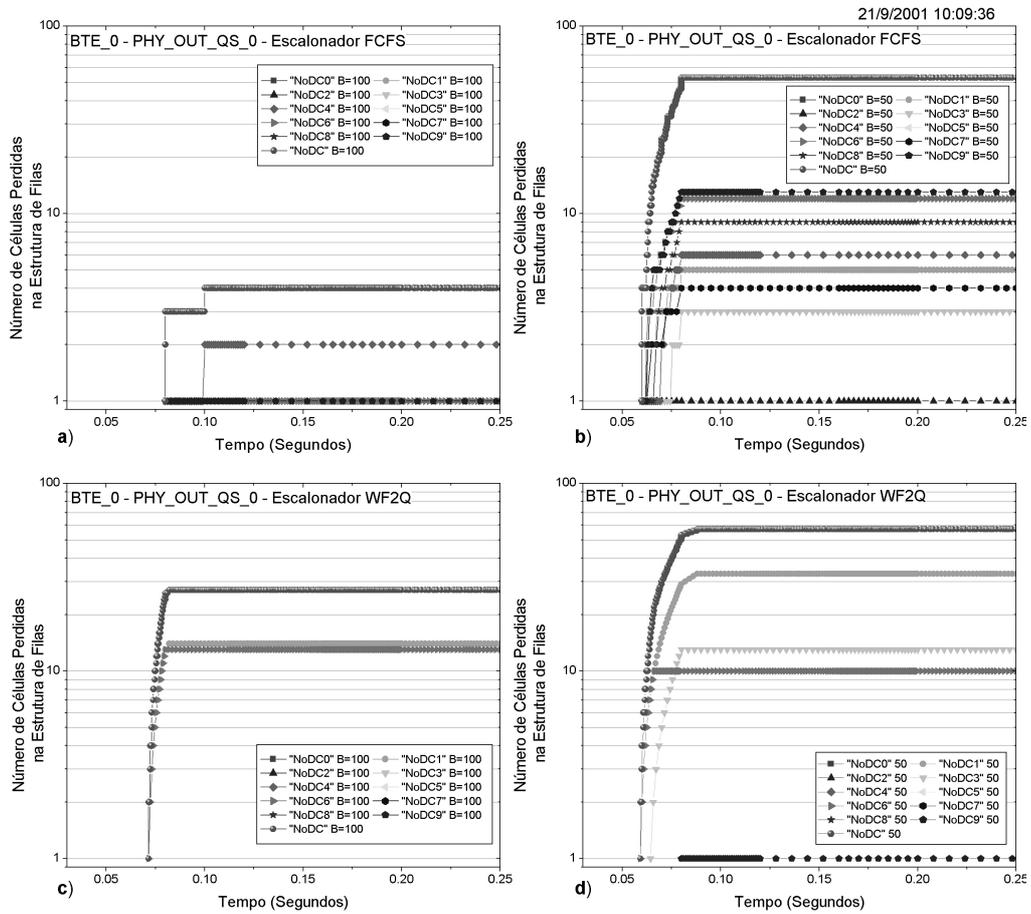


Figura 10. Número de células perdidas na estrutura de filas do BTE_0 com capacidades de 100 e 50 células para ambos os modelos de escalonadores.

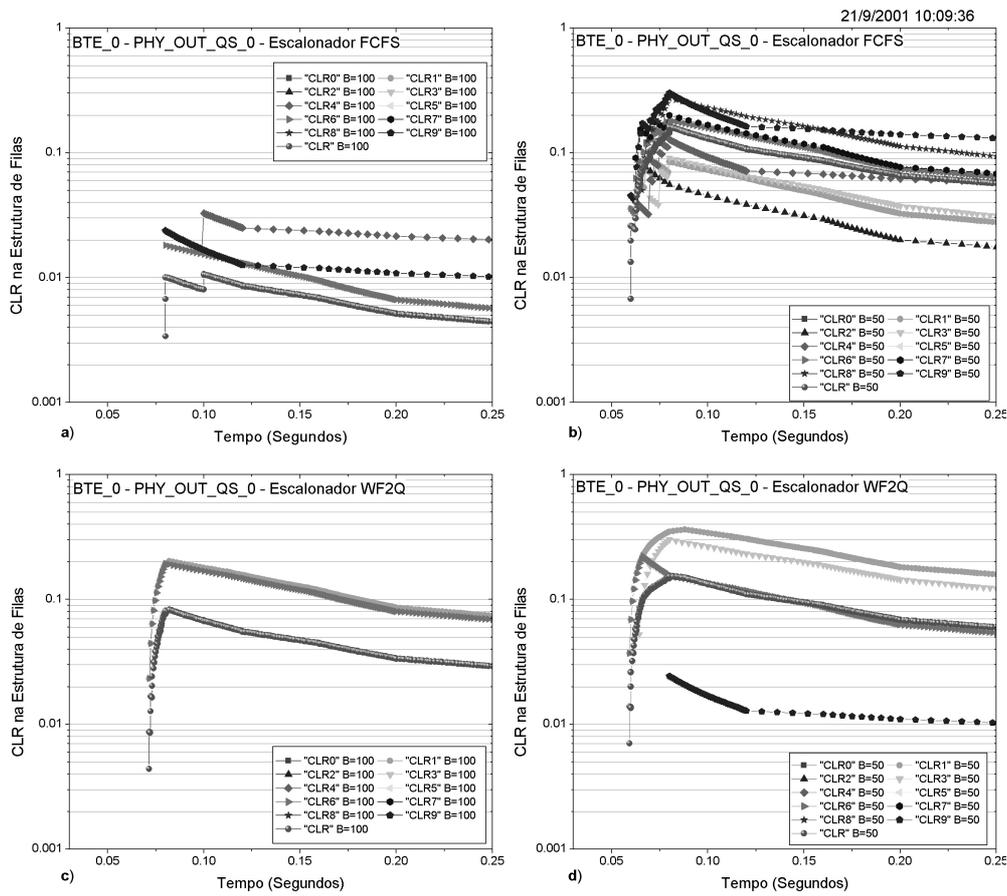


Figura 11. CLR na estrutura de filas do BTE_0 com capacidades de 100 e 50 células para ambos os modelos de escalonadores.

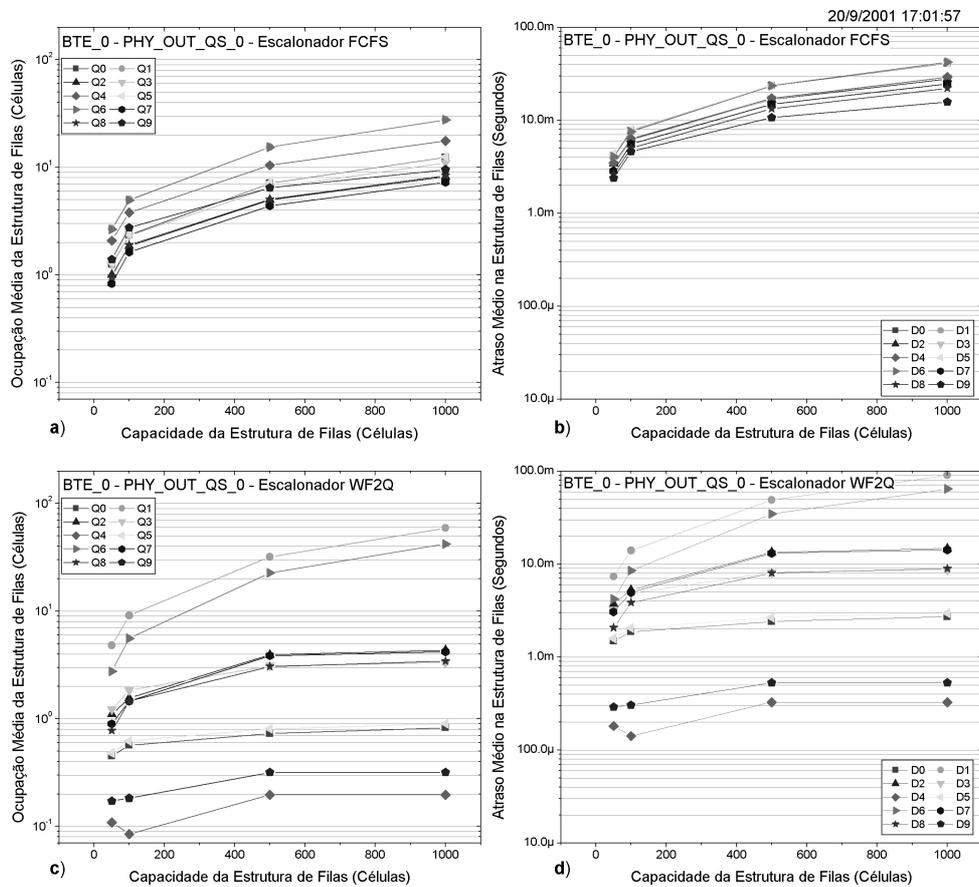


Figura 12. Ocupação média e atraso médio da estrutura de filas do BTE_0 para os dois modelos de escalonadores

taxa média de perda de células na estrutura de filas PHY_OUT_QS_0 e a utilização média dos escalonadores do BTE_0. Consideremos a Figura 13a para o escalonador FCFS. Podemos observar que para a capacidade de armazenamento de 1000 células, somente existem células perdidas nas conexões 1, 6, 0 e 5. Para a capacidade de 500 células, foram perdidas células em todas as conexões. Porém, as conexões 1, 6, 0 e 5 ainda foram as mais afetadas. Para a capacidade de 100 células, todas as conexões tiveram perdas no intervalo entre 10^{-1} e 9×10^{-3} . Finalmente, para a capacidade de 50 células, todas as conexões tiveram perdas no intervalo entre 6×10^{-2} e 3×10^{-2} . Para o escalonador WF²Q (Figura c) as conexões que tiveram maiores perdas considerando a capacidade de armazenamento de 1000 células também foram as conexões 1 e 6. Para a capacidade de 500 células, além das conexões 1 e 6, houveram perdas consideráveis nas conexões 3 e 8. Para a capacidade

de 100 células, todas as conexões tiveram perdas no intervalo entre 2×10^{-1} e 1×10^{-3} . Para a capacidade de 50 células, todas as conexões tiveram perdas no intervalo entre 2×10^{-1} e 1×10^{-2} . A partir destas observações podemos concluir que após 60 segundos de simulação o escalonador WF²Q proveu um melhor isolamento de QoS entre as conexões do que o escalonador FCFS. Este fato fica mais evidente a medida que o nível de congestionamento na rede aumenta. Outra conclusão importante, diz respeito ao CLR evidenciado pelas conexões 1 e 6, que foram as conexões mais afetadas em ambos os escalonadores. Novamente, suspeitamos que a principal razão para este fato foi a escolha inadequada dos pesos para estas conexões.

As figuras: Figura 13b e Figura 13d mostram que a utilização média do escalonador WF²Q foi ligeiramente superior do que a utilização média do escalonador FCFS.

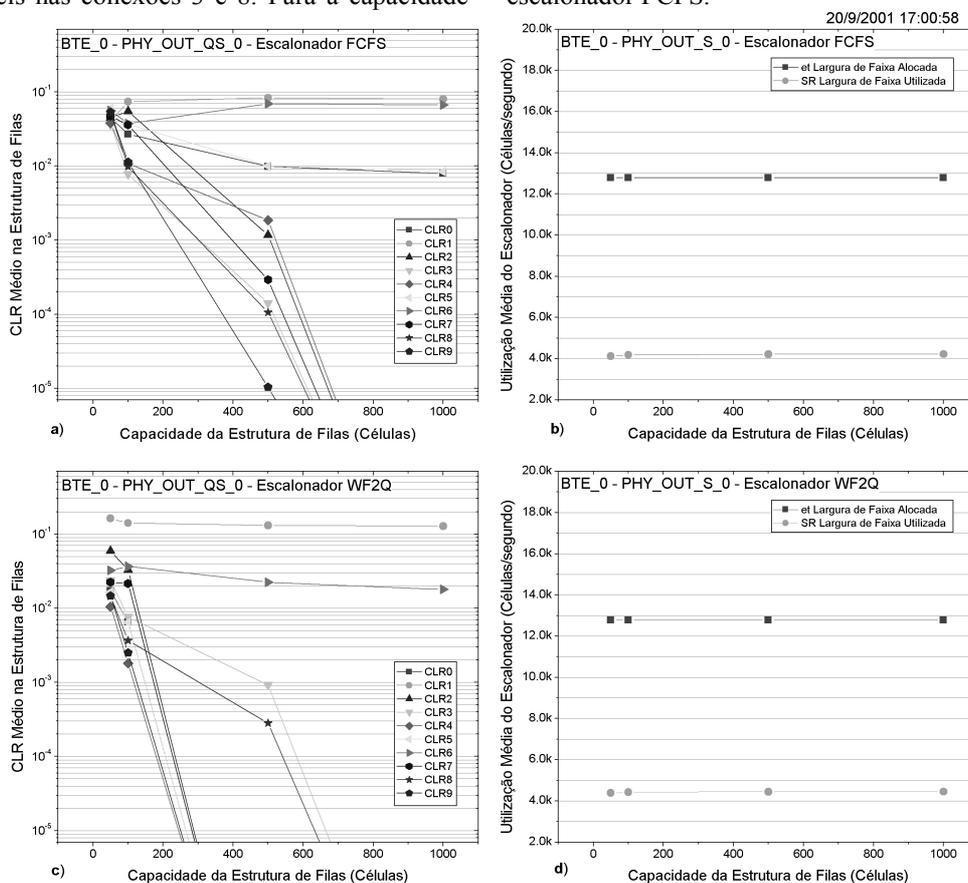


Figura 13. CLR médio na estrutura de filas do BTE_0 para os dois modelos de escalonadores e utilização média dos escalonadores.

VI. CONSIDERAÇÕES FINAIS

Neste artigo apresentamos o modelamento e a simulação de um escalonador WF²Q – *Worst-case Fair Weighted Fair Queuing*. Para tanto, utilizamos o ambiente de simulação de rede de comunicações SimNT 2.0 [5]. Embora nosso modelo de escalonador WF²Q tenha sido projetado para este ambiente de simulação, onde faz parte de um conjunto de modelos no nível de células para a simulação de redes ATM [7], os detalhes de implementação aqui apresentados

poderão ser utilizados para desenvolver modelos de escalonadores WF²Q em outros simuladores ou até mesmo em aplicativos reais.

Para demonstrarmos o modelo desenvolvido, simulamos o transporte de seqüências de tráfego de vídeo codificado MPEG-4 [8] sobre uma rede ATM. O desempenho do escalonador WF²Q foi analisado tomando-se como referência o desempenho de um escalonador FCFS. Nas simulações realizadas, confirmamos que o escalonador WF²Q é capaz de garantir o isolamento de QoS entre conexões em redes

ATM. Ou seja, o escalonador WF²Q é capaz de manter a qualidade de serviço para cada conexão da rede, independente das outras conexões. Entretanto, verificamos que a QoS oferecida pelo escalonador WF²Q depende diretamente de uma configuração cuidadosa dos pesos ϕ_i para cada conexão. Em trabalhos futuros pretendemos investigar a influência destes pesos ϕ_i no desempenho do escalonador WF²Q.

A validação do modelo apresentado foi feita através da análise da evolução das variáveis do escalonador durante a execução do algoritmo de armazenamento de células ATM. A partir desta análise (seção 0), concluímos que o funcionamento do modelo desenvolvido está de acordo com a seção 3 do artigo de Bennett e Zhang [4].

VII. AGRADECIMENTOS

Agradecemos à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo suporte a este trabalho e ao Grupo de Redes de Telecomunicações (TKN) da Universidade Técnica de Berlin (TU Berlin) por disponibilizar na Internet as seqüências de tráfego MPEG-4.

VIII. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] A. Parekh e R. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case”, *IEEE/ACM Transactions on Networking*, volume 1, número 3, junho de 1993.
- [2] N. Giroux e S. Ganti, “Quality of Service in ATM Networks: State-of-Art Traffic Management”, *Prentice Hall*, 1998.
- [3] A. Demers, S. Khesav e S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm”, *Journal of Internetworking Research and Experience*, outubro de 1990.
- [4] Jon Bennett e Hui Zhang, “WF2Q: Worst-case Fair Weighted Fair Queueing”, *IEEE Infocom* 1996.
- [5] Ernesto Andrade Neto e Antônio M. Alberti, “Hydragyrum – Ambiente de Simulações de Redes a Eventos Discretos”, Congresso SBrT 2000, setembro de 2000.
- [6] William Stallings, “High-Speed Networks TCP/IP and ATM Design Principles”, *Prentice Hall*, 1998.
- [7] Antônio M. Alberti e Ernesto Andrade Neto, “Simulação de Redes ATM no Nível de Células”, Congresso SBrT 2000, setembro de 2000.
- [8] Rob Koenen, “MPEG-4: Multimedia for Our Time”, *IEEE Spectrum*, volume 36, número 2, fevereiro de 1999.
- [9] Antônio M. Alberti, “SimATM: Um Ambiente para a Simulação de Redes ATM”, Tese de Mestrado, Orientador: Leonardo de Souza Mendes, UNICAMP – Universidade Estadual de Campinas, abril de 1998.
- [10] Charles Petzold, “Programando para Windows 95”, *Markron Books e Microsoft Press*, 1996.
- [11] Timothy Budd, “Classic Data Structures in C++”, *Addison Wesley*, 1994.
- [12] Antônio M. Alberti, “Estimativa de Descritores de Tráfego MPEG-4 para o Transporte em Redes ATM Utilizando a Técnica do Buffer Virtual”, Congresso SBrT 2001, setembro 2001.
- [13] Frank H.P. Fitzek e Martin Reisslein, “MPEG-4 and H.263 Video Traces for Network Performance Evaluation”, Tutorial Técnico, <http://www-tkn.ee.tu-berlin.de/research/trace/trace.html/>, outubro de 2000.
- [14] Michael Orzessek e Peter Sommer, “ATM & MPEG-2: Integrating Digital Video in Broadband Networks”, *Prentice Hall*, 1998.
- [15] ATM Forum, “Audiovisual Multimedia Services: Video on Demand Specification 1.1”, março de 1997.
- [16] Ahmed Mehaoua, Raouf Boutaba e Guy Pujolle, “An Adaptive and Selective Cell Drop Policy with Dynamic Data Partitioning for Best Effort Video over ATM”, *IEEE International Conference on Computer Networks (LCN'97)*, novembro de 1997.
- [17] Anwar Elwalid, Debasis Mitra e Robert Wentworth, “A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node”, *IEEE Journal on Selected Areas in Communications*, volume 13, número 6, agosto de 1995.
- [18] Santosh Krishnan, Abhijit K. Choudhury e Fabio Chiussi, “Dynamic Partitioning: A Mechanism for Shared Memory Management”, *IEEE Infocom* 1999.

SOBRE OS AUTORES

Antônio Marcos Alberti graduou-se em Engenharia Elétrica pela Universidade Federal de Santa Maria, RS, em 1996. Recebeu o título de mestre em Eletrônica e Comunicações pela Universidade Estadual de Campinas, SP, em 1998. Atualmente está concluindo o seu doutorado também em Eletrônica e Comunicações na UNICAMP, onde desenvolve trabalhos ligados ao modelamento e a simulação de redes de comunicações. **E-mail:** alberti@decom.fee.unicamp.br
Daniel Zaccarias é aluno de graduação do curso de Engenharia Elétrica da UNICAMP. **E-mail:** dzaccaris@fee.unicamp.br
Samuel Penha é aluno de graduação do curso de Engenharia Elétrica da UNICAMP. **E-mail:** spenha@fee.unicamp.br
Leonardo de Souza Mendes graduou-se em Engenharia Elétrica com Ênfase em Eletrônica pela Universidade Gama Filho, RJ, em 1985. Recebeu o título de mestre em Ciências em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio de Janeiro, RJ, em 1987. Recebeu o título de PhD em *Electrical Engineering*, pela *Syracuse University*, NY, USA, em 1991. Atualmente, é professor associado do Departamento de Comunicações da Faculdade de Engenharia Elétrica e de Computação da UNICAMP. **E-mail:** lmendes@decom.fee.unicamp.br

