

Análise de desempenho de redes p2p com protocolo “push/pull” para distribuição de vídeo na presença de nós não-cooperativos

Flávia Marinho de Lima¹, Alexandre Sztajnberg¹

¹Programa de Pós-Graduação em Eletrônica - Universidade do Estado do Rio de Janeiro
R. S. F. Xavier, 524/5º 22550-013 - Rio de Janeiro – RJ – Brasil

flaviamarinho@yahoo.com, alexszt@uerj.br

Abstract. *P2P (peer to peer) architectures are being used as an infrastructure for video stream distribution on the Internet. The idea is that the several nodes of the overlay network should cooperate distributing and forwarding video chunks, making available their local resources to the network. In this way, it is important to investigate what happens to the quality of service of the video distribution when the infrastructure provided by the P2P network is “contaminated” with free-riding nodes, which are not willing to cooperate, since the basis of this architecture is cooperation. In this work, we evaluate how the presence of uncooperative nodes can affect the quality of the video stream distribution on push-pull P2P networks. The evaluation was performed using the PeerSim simulator.*

Resumo. *A arquitetura P2P (peer-to-peer) vem sendo considerada como infraestrutura para a distribuição de fluxos de vídeo na Internet. A idéia é a de que os vários nós integrantes da rede sobreposta distribuem e encaminham pedaços de vídeo de forma cooperativa, dividindo as tarefas, e colocando à disposição da rede seus recursos locais. Dentro deste contexto, é importante investigar o que ocorre com a qualidade do serviço de distribuição de vídeo quando a infraestrutura provida pelas redes P2P é “contaminada” por nós que não estejam dispostos a cooperar, já que a base desta arquitetura é a cooperação. Neste trabalho, é feito um estudo para verificar como a presença de nós não-cooperativos pode afetar a qualidade da distribuição de fluxo de vídeo em redes P2P com protocolo push-pull. A avaliação foi realizada utilizando-se o simulador PeerSim.*

1. Introdução

A distribuição de vídeo pela Internet traz grandes desafios, pois é um serviço que requer (i) servidores com grande capacidade de processamento e armazenamento, e (ii) grande largura de banda, baixo retardo e variação de retardo (*jitter*) e poucas perdas. É também um serviço que apresenta requisitos de escalabilidade, pois se estima uma quantidade grande de usuários simultâneos, e flexibilidade, uma vez que estes usuários devem receber os fluxos de vídeo de acordo com sua capacidade de banda e recursos computacionais para tratamento e exibição dos mesmos. Tornar disponível uma solução que contemple todas estas características é importante para que se possa oferecer ao usuário um serviço de qualidade, mas não é tarefa trivial, dado que a arquitetura da Internet não foi concebida com esta finalidade.

Neste contexto, as redes par-a-par, ou *peer to peer* (P2P)¹, vem sendo consideradas como solução potencial para a distribuição de vídeo na Internet, devido às suas características de escalabilidade e distribuição de responsabilidades. A idéia básica é a de que os vários nós integrantes da rede P2P distribuem e encaminham pedaços de vídeo de forma cooperativa, dividindo as tarefas, e colocando à disposição da rede seus recursos locais. Com isso diminui-se a necessidade de canais reais com grandes bandas e servidores com grande capacidade de processamento e armazenamento. Nesta abordagem, quanto maior o número de nós, maior é a capacidade de distribuição de vídeo, pois mais recursos são compartilhados, sem a dependência de um único servidor ou federação de servidores, o que torna as redes P2P robustas para este serviço.

Entretanto, mesmo com características favoráveis, a distribuição de vídeo nas redes P2P também apresenta desafios. É frequente que alguns nós, geralmente nós com mais recursos que os demais, fiquem sobrecarregados porque nem todos os nós participantes querem cooperar. De certa forma, com a existência de muitos nós com comportamento não-cooperativo, as redes P2P passam a apresentar características de um sistema centralizado, podendo apresentar um desempenho ainda pior, pois os nós sobrecarregados não são necessariamente especializados para a atividade de armazenamento e distribuição de vídeo, tornando-se gargalos.

Estudos empíricos têm mostrado que nós que atuam como parasitas na rede, consumindo recursos, sem contribuir, são maioria em redes P2P voltadas para o compartilhamento de arquivos [Adar e Huberman, 2000] [Saroiu, 2003], prejudicando os usuários cooperativos. No caso da aplicação de distribuição de vídeo este comportamento é ainda mais nocivo, uma vez que os usuários não estão em busca apenas da disponibilidade do arquivo contendo o vídeo, mas também esperam um mínimo de qualidade na obtenção e exibição do mesmo.

Neste trabalho analisamos o impacto causado pela ação de nós com comportamento não-cooperativo, no desempenho da distribuição de fluxos de vídeo em redes P2P que usam o protocolo de difusão de pedaços *push/pull* proposto em [Cigno, 2008]. Numa primeira etapa, simulações são realizadas aumentando-se gradativamente o número de nós maliciosos na rede. Assim, é possível avaliar a influência da porcentagem de nós não-cooperativos, no sistema como um todo. Os resultados comprovam que o desempenho do protocolo *push/pull* diminui à medida que a porcentagem de nós maliciosos aumenta. Numa segunda etapa introduzimos no modelo a possibilidade de cada nó escolher os seus vizinhos na formação do grafo, baseado em uma “reputação” inicial, considerando se o nó candidato a vizinho é ou não cooperativo. Com isso, a escolha deixa de ser 100% aleatória, garantindo uma certa qualidade de vizinhança. Após as simulações, uma melhora no desempenho da rede, como um todo, é constatada e os nós cooperativos são em geral ativados mais rapidamente. Entretanto, os nós não-cooperativos ainda são beneficiados. Diante destes resultados identificamos a necessidade de criação de um mecanismo de incentivo à cooperação.

Este artigo está organizado da seguinte forma. A próxima seção discute os conceitos do protocolo *push/pull*. A Seção 3 discute trabalhos relacionados. Na Seção 4, apresentam-se o simulador PeerSim e as métricas usadas na simulação. A Seção 5 apresenta os resultados da primeira etapa da avaliação. Na Seção 6 o mecanismo de controle de admissão de vizinhos é apresentado e avaliado. A Seção 7 conclui o artigo.

¹ Redes sobrepostas (*overlay*), que executam sobre redes como a Internet, com nós virtuais, sendo interligados por canais de comunicação também virtuais.

2. Conceitos Básicos

Uma das formas de se transmitir arquivos ou fluxos de vídeo em redes P2P pela Internet requer a divisão do conteúdo em pedaços, ou *chunks*, a fim de transmiti-los de maneira independente uns dos outros. O *BitTorrent* [Hales, 2005] utiliza este método para compartilhamento de arquivos em redes P2P, e alguns sistemas de distribuição de vídeo, como o *CoolStreaming* [Xinyan, 2005], também utilizam este método, dividindo um fluxo de vídeo em vários pedaços para então transmiti-los.

2.1. Modelos de difusão de pedaços em redes P2P

Para os sistemas que utilizam a divisão em pedaços para a transmissão do conteúdo, o mecanismo de difusão dos pedaços entre os nós da rede P2P é ponto de relevância. Existem três modelos de difusão de pedaços que se destacam na literatura: o modelo *push*, o modelo *pull*, e o modelo baseado no estado de cada nó. Além disso, alguns trabalhos propõem modelos híbridos.

No modelo *push*, os pedaços são enviados de um nó pai para nós filhos, sem o nó pai questionar se os nós filhos precisam ou não daquele pedaço. Caso um nó filho, possua mais de um nó pai, fica fácil perceber que o mesmo pedaço pode chegar várias vezes ao mesmo nó, ao passo que um determinado pedaço pode demorar ou nunca chegar. O método *push* é, portanto, mais indicado em redes estruturadas, com topologia baseada em árvore [Tran, 2004] [Venkataraman, 2006] [Sung, 2006].

O modelo *pull* apresenta comportamento oposto ao do modelo *push*. Neste, um nó filho faz a requisição de um pedaço ao nó pai sem saber se ele o possui. Aqui não há problema de duplicação, mas existe o problema de *starvation*, pois um filho pode nunca encontrar um nó pai que possua o pedaço que ele esteja procurando. Em geral, o modelo *pull* está associado a sistemas não estruturados, onde um nó filho pode ser suprido por vários nós pai, diminuindo assim a probabilidade de ocorrer *starvation*. Até onde foi investigado, na prática é difícil existirem sistemas que utilizem exclusivamente o modelo *pull* para distribuição de vídeo.

No modelo baseado no estado [Pai, 2005] [Xinyan, 2005] [Pianese, 2006], os nós trocam informações entre si para saber o que cada nó vizinho possui, e assim pedir-lhe ou enviar-lhe um determinado pedaço. Cada nó possui o seu mapa de *buffer* que mantém um índice dos pedaços que o mesmo possui, e ainda uma lista dos pedaços que estão na eminência de serem reproduzidos. Os nós fazem a troca do mapa de *buffer* com os seus nós vizinhos. Como os nós têm consciência do que cada um possui, eles podem fazer a requisição de pedaços de maneira otimizada.

Alguns trabalhos propõem modelos híbridos para a distribuição de pedaços. Um desses modelos é o *Interleave* [Sanghavi, 2007] que combina os modelos *push* e *pull*, e adiciona um mecanismo de política de seleção de pedaços, sem nenhuma troca de informações entre os nós sobre os pedaços que cada um possui. Este modelo foi projetado e analisado principalmente para transferência de arquivos.

O trabalho [Cigno, 2008] analisou o desempenho do modelo *Interleave* em detalhe, e ampliou seu escopo para permitir redes P2P com comportamentos diversos e para comprovar se era possível transmitir vídeo com este modelo de difusão de pedaços. Os resultados mostraram que era possível não só ter bom desempenho para distribuição de arquivos, como também para a distribuição de vídeo sem manter quaisquer informações sobre os pedaços.

2.2. O modelo de difusão de pedaços *push/pull*

O modelo de rede P2P para distribuição de vídeo proposto em [Cigno, 2008] possui apenas uma única fonte, que divide o conteúdo do vídeo em pedaços para serem distribuídos entre os nós da rede sobreposta de forma independente. Os pedaços são gerados a uma taxa constante pela fonte e cada pedaço possui um identificador único que reflete a sua ordem de criação.

A rede sobreposta é do tipo não-estruturada, com topologia em malha e simétrica, ou seja, se um nó N é vizinho de V , então V é vizinho de N . Cada nó N da rede possui uma lista com a referência para todos os seus nós vizinhos e pode contatar ativamente estes nós e apenas eles.

Cada nó alterna seu estado entre os modos *push* e *pull*. Apenas o nó fonte não alterna o seu estado, ele permanece sempre com o estado *push*, pois o nó fonte não precisa requisitar pedaços, ele já os possui.

Durante o estado *push*, um nó N realiza as seguintes atividades:

1. seleciona um vizinho V aleatoriamente, e o pedaço K com o identificador mais alto, ou seja o k mais recente, entre os diversos pedaços recebidos dos seus vizinhos;
2. em seguida, envia uma mensagem ao vizinho V selecionado aleatoriamente, questionando se este deseja K ;
3. se o nó V não possui K e tiver banda de *download* disponível, o nó V envia uma mensagem a N aceitando a oferta, e o nó N envia o pedaço K ao nó V ;
4. caso contrário, o nó N aborta o processo.

Durante o estado *pull*, um nó N realiza as seguintes atividades:

1. envia uma mensagem de requisição de pedaço a um nó vizinho V aleatório, solicitando o pedaço K com o identificador mais baixo que não possua;
2. se o nó V possuir o pedaço K e tiver banda de *upload* disponível, o nó V envia uma mensagem ao nó N aceitando a requisição;
3. caso contrário, o nó V rejeita a requisição.

A política de seleção de pedaços é uma parte muito importante em sistemas de distribuição de vídeos, pois cada pedaço tem um tempo máximo de retardo tolerado para iniciar a sua reprodução. Se um pedaço de fluxo de vídeo chegar após o seu tempo de reprodução, haverá uma descontinuidade no vídeo. Algoritmos para selecionar os pedaços, tal qual o mais raro primeiro, não podem ser implementados apenas com modelos *push/pull*, pois, como visto, é necessário conhecimento prévio do que os outros nós possuem. O modelo sugerido por [Cigno, 2008] adotou a mesma política de seleção de pedaços empregada em [Sanghavi, 2007]:

1. Durante o estado *push*, o nó N envia o pedaço com o identificador mais alto entre os diversos pedaços recebidos via estado *push* de seus vizinhos.
2. Durante o estado *pull*, o nó N requisita o pedaço com o identificador mais baixo que não possua. Essa política ajuda o nó N a preencher buracos na sequência de pedaços.

É possível observar que essa política de seleção não necessita de um prévio conhecimento do que outro nó possui. Um nó muda de um estado para outro, ou depois de uma requisição ser aceita e o pedaço correspondente terminar de ser transferido, ou

depois de receber um número máximo de negações às suas requisições. Este comportamento mantém o sistema rodando suavemente, e evita *starvation*. Além disso, as informações trocadas pelos nós vizinhos são as mínimas possíveis, apenas mensagens de *keep-alive* e mensagens de requisições e respostas do protocolo *push/pull* são usadas.

A maior vantagem em se utilizar um protocolo de difusão baseado nos modelos *push/pull* é a sua simplicidade, relacionada ao fato deste poder trabalhar sem qualquer suposição em relação ao comportamento de cada nó. Nenhuma sinalização é necessária para coordenar os nós, fazendo com que este protocolo seja, em tese, adequado a redes muito dinâmicas, ou seja, com muitas entradas e saídas de nós. Por outro lado, protocolos baseados em estado buscam melhorar o desempenho e a eficiência, aumentando o risco de se tornarem frágeis e propensos a falhas em redes heterogêneas e dinâmicas, com nós com comportamento variável.

Para alguns casos, como transmissão de vídeo por difusão, onde, em geral, todos os usuários estão interessados no mesmo recurso em um espaço curto de tempo, talvez um protocolo mais simples seja suficiente para atender todos os requisitos do serviço de vídeo de maneira eficiente. Para outros casos, como transmissão de vídeo sob demanda, onde os usuários interessados no vídeo estão em diferentes momentos de reprodução do mesmo, provavelmente um protocolo baseado em estado seja mais eficaz, pois cada usuário saberá exatamente a quem recorrer para obter os pedaços de vídeo desejados, sem correr o risco de pedir para um usuário que não o tenha.

3. Trabalhos relacionados

Existem vários trabalhos relacionados ao uso de redes P2P para distribuição de vídeo [Tran, 2003] [Habib, 2004] [Xinyan, 2005] [Carlsson, 2007] [Cigno, 2008] [Moraes, 2008] [Moraes, 2009]. Nenhum deles, entretanto, faz um estudo específico sobre o comportamento de uma rede que utilize o protocolo de difusão *push/pull* face a presença de nós com comportamento não-cooperativo.

[Sanghavi, 2007] propôs o protocolo *Interleave* para difundir os pedaços entre os nós da rede. Este protocolo combina os modelos *push* e *pull*, alternando os estados dos nós entre um ou outro, e foi criado com o propósito de aumentar o desempenho de serviço de distribuição de arquivos. O autor também adotou uma política de seleção de pedaços, conforme mencionado na seção anterior. O trabalho comprovou que o protocolo podia ser utilizado satisfatoriamente para serviço de distribuição de arquivos.

Alguns trabalhos propõem a combinação dos modelos *push* e *pull*, utilizando o mecanismo *push* para distribuir rapidamente os pedaços, e o mecanismo *pull* para preencher os “buracos” no *buffer* de reprodução. Não há uma alternância entre os dois modelos, por exemplo, [Locher, 2007].

Mais próximo de nosso trabalho, [Cigno, 2008] fez um estudo do protocolo *push/pull* com a finalidade de avaliar o comportamento e desempenho do mesmo na tarefa de distribuição de fluxo de vídeo na Internet. Para atingir seus objetivos, variou parâmetros como: quantidade de nós, quantidade de vizinhos por nó, quantidade de pedaços a serem distribuídos e banda de *upload*. Os resultados foram comparados aos obtidos em [Sanghavi, 2007] e apresentaram indícios de que era possível ter bom desempenho para transmissão de vídeo utilizando um modelo de rede P2P que combinasse os modelos *push* e *pull*.

Até aqui as discussões apresentadas consideraram que todos os nós atuam de

forma cooperativa na rede. Grande parte dos sistemas P2P conta com isto, já que deveria ser do interesse do nó participante cooperar com os demais, pois a qualidade dos serviços que ele utiliza depende do desempenho da rede como um todo.

Estudos mostram que em sistemas de compartilhamento de arquivos, o tempo total para baixar arquivos e a taxa de falhas de toda a rede aumentam a medida que os nós deixam de compartilhar seus recursos [Adar e Huberman, 2000]. Em redes sem fio *ad hoc*, a latência de pacotes e a taxa de perda aumentam para todos, quando os nós se recusam a encaminhar pacotes de controle sobre o comportamento dos outros nós [Marti et al, 2000].

Seria natural raciocinar que é interessante para um determinado nó cooperar para maximizar os seus resultados. Mas, isso não é o que ocorre de fato. Estudos têm mostrado que nós que atuam como parasitas, consumindo recursos sem contribuir, são maioria em redes P2P para compartilhamento de arquivos [Adar e Huberman, 2000] [Sarioiu et al, 2003]. Não apenas usuários egoístas ou parasitas prejudicam o desempenho de uma rede P2P. Usuários mal intencionados, que querem realmente degradar o serviço oferecido pela rede, ou usuários mentirosos, que querem trapacear, também prejudicam o desempenho da rede.

Seria desejável que o desempenho das aplicações em redes P2P se mantivesse estável, mesmo diante de nós não-cooperativos. A infraestrutura provida e os protocolos utilizados deveriam ser robustos o suficiente para que as aplicações fossem minimamente afetadas por nós com comportamento malicioso. Alguns trabalhos relacionados ao uso de redes P2P para distribuição de vídeo abordam esse problema.

[Habib e Chuang, 2004] propõe um mecanismo de seleção de vizinhos baseado em um ranking para distribuição de vídeo na Internet com nós com interesses assimétricos. O mecanismo estimula o incentivo à cooperação através da diferenciação de serviços. Nós cooperativos da rede são recompensados com flexibilidade e escolha na seleção dos vizinhos, resultando em alta qualidade na reprodução do vídeo. Os nós não-cooperativos têm limitadas opções na seleção de vizinhos, obtendo uma baixa qualidade de vídeo. Os autores verificaram que o mecanismo pode prover qualidade de vídeo próxima ao ótimo, quando todos os nós são cooperativos, desde que as fontes de vídeo não estejam conectadas por enlaces no limite de sua banda. Os autores também afirmam que o mecanismo de incentivo proposto não é necessário quando a rede está totalmente disponível e não é efetiva quando o congestionamento é alto.

[Chu et al, 2004] propõe um modelo de taxaço, onde nós com mais recursos contribuem com mais banda no sistema. Nós com recursos limitados são subsidiados pelo sistema. O modelo é aplicado no contexto de distribuição de vídeo onde o distribuidor do vídeo impõe a taxaço para os demais nós para obter o máximo de cooperação de cada um. Este esquema é usado quando um número alto de usuários está interessado em uma mesma sessão de vídeo, e estes usuário estão dispostos a cooperar de maneira síncrona, mesmo que com alguma sobrecarga, para que a qualidade do vídeo recebido seja satisfatória.

[Hoong e Matsuo, 2008] propõe o protocolo PALMS (*P2P Unstructured Live Media Streaming*) que faz uso dos modelos *push*, *pull* e conhecimento do estado do mapa de *buffer* de cada nó para difusão dos pedaços de vídeo em conjunto com um mecanismo de incentivo à cooperação baseado na pontuação do nó. A pontuação é função da razão entre *bytes* transmitidos e *bytes* recebidos. A seleção do vizinho depende da pontuação do nó solicitante e do nó candidato a vizinho. Um nó solicitante

só pode selecionar nós com pontuações menores ou iguais à dele, assim cada nó fica estimulado a cooperar mais a fim de poder selecionar melhores parceiros para aumentar a qualidade do vídeo reproduzido. Com este mecanismo os autores obtiveram resultados satisfatórios, mesmo diante da existência de nós não-cooperativos na rede.

4. Ambiente da simulação

O ponto fundamental deste trabalho trata do estudo do comportamento de uma rede sobreposta em malha que utilize o método de difusão de pedaços baseado no modelo *push/pull*, diante de nós que podem ou não estarem dispostos a cooperar. O objetivo é mensurar o quanto a presença de nós maliciosos pode prejudicar o desempenho da rede P2P na distribuição de vídeo. Devido às dificuldades inerentes aos testes reais com muitos nós, optou-se por simular a rede sobreposta em um simulador de rede P2P.

4.1. O simulador PeerSim

As simulações foram realizadas no simulador *PeerSim* [PeerSim] utilizando o módulo P4S [P4S], desenvolvidos em Java. O *PeerSim* suporta dois modelos de simulação: orientado a ciclos e o orientado a eventos. Para este trabalho foi escolhido o modelo de simulação orientado a eventos, pois era preciso que o ambiente fosse o mais próximo possível da realidade.

Em nossas simulações utilizamos grafos simétricos. Para isso, utilizamos a classe *peersim.dynamics.WirekOutUnd*, a mesma utilizada pelo trabalho [Cigno, 2008] para distribuição de vídeo. Esta classe cria um grafo simétrico com cada nó possuindo um número fixo de vizinhos, e a escolha de cada vizinho é aleatória.

Para facilitar o estudo do protocolo de difusão baseado no modelo *push/pull*, utilizou-se o módulo P4S, criado para simular a distribuição de arquivos e a distribuição de vídeo em redes P2P em malha. O módulo P4S provê a classe *p4s.core.Alternate* que simula um protocolo de difusão que combina os modelos *push* e *pull*.

Para introduzir as características de nós não-cooperativos na primeira etapa da simulação foi necessário criar o código para nós com tal comportamento. Na segunda etapa introduzimos um controle na formação das vizinhanças, limitando a quantidade de nós não-cooperativos que um nó cooperativo poderia ter.

4.2. Infraestrutura da simulação

Para modelar um nó com comportamento não-cooperativo, modificamos a seqüência das atividades realizadas pelo nó no estado *push* porque de acordo com os resultados de [Cigno, 2008] o número de pedaços recebidos via *push* é ligeiramente maior do que recebido via *pull*. Com isso o nó não-cooperativo nesta simulação prejudica seus vizinhos porque não envia pedaços de forma voluntária aos mesmos, entretanto quando é solicitado a enviar um pedaço, o nó não-cooperativo age corretamente. Pode-se dizer então, que o nó não-cooperativo neste caso não é 100% prejudicial, não tem a finalidade de prejudicar totalmente a rede e sim de se beneficiar em relação aos outros nós, de certa forma age de maneira egoísta.

A diferença da seqüência original é a introdução de mais uma condição no estado *push*. O nó que vai enviar voluntariamente um pedaço deve, agora, além de possuir um número de conexões ativas menor que o número máximo de conexões possíveis e banda de *upload* disponível, ser também um nó cooperativo. Caso contrário,

sendo um nó não-cooperativo, o mesmo passa para o estado *pull* diretamente, sem enviar nenhum pedaço.

Na simulação a rede sobreposta é composta por um nó fonte que gera os pedaços a uma taxa constante. Os nós são inicializados ao mesmo tempo e seus vizinhos são escolhidos de forma aleatória e simétrica, conforme já explicado. Os nós entram na rede como nós passivos, ou seja, não podem solicitar nenhum pedaço a um de seus vizinhos até que fiquem no modo ativo, para que isso ocorra, o nó precisa receber o seu primeiro pedaço via método *push*. Uma vez aceitos na rede, os nós não saem desta, ou seja, não há *churn*.

Após a inicialização dos nós, o nó fonte começa a enviar pedaços para os seus vizinhos, ativando-os. A partir daí, outros nós também são ativados e portanto ficam aptos a pedir e enviar pedaços. A simulação chega ao fim quando todos os nós obtiverem todos os pedaços ou o tempo limite estiver esgotado.

Dois tipos de vídeo e respectivos parâmetros, utilizados em [Costa, 2004] e em [Moraes, 2009], foram adotados em nossas simulações. Um deles, obtido a partir da TV UOL - entretenimento de 5 min e o outro do servidor eTeach, Univ. de Wisconsin - educacional de 20 min. A Tabela 1 resume estes parâmetros.

Tabela 1. Parâmetros considerados em todas as simulações

	Entretenimento	Educativo
Número de nós participantes	200	200
Tamanho do vídeo (pedaços)	30	120
Tamanho do vídeo (minutos)	5	20
Taxa de transmissão do vídeo (Kb/s)	350	350
Duração do pedaço de vídeo (s)	10	10
Tamanho do pedaço do vídeo (kB)	437,5	437,5

Além dos parâmetros da Tabela 1, definiu-se a banda de *upload* 600 kb/s para todos os nós, e o número de vizinhos de cada nó igual a 12, essas escolhas foram baseadas nos resultados de [Cigno, 2008], onde a rede tornava-se estável. Foram realizadas 300 rodadas de simulação, variando a porcentagem de nós não-cooperativos na rede em 0%, 5%, 10%, 15% e 20%.

4.3. Métricas escolhidas

Quatro métricas foram escolhidas para verificar o desempenho do protocolo push/pull em presença de nós não-cooperativos durante a simulação:

Tempo máximo, tempo que o último nó a terminar de reproduzir o vídeo leva para obter todos os pedaços.

Qualidade do sistema de vídeo, definida em [Habib, 2004] como:

$$Q = \frac{\sum_{i=1}^T Z_i}{T} \quad (1)$$

onde T é o número total de pedaços envolvidos na distribuição de vídeo e Z_i é a variável que representa o fato do pedaço ter chegado ou não antes do seu tempo de reprodução.

Caso o pedaço tenha chegado antes do seu tempo de reprodução, a variável Z_i recebe 1, do contrário Z_i recebe 0, o que representa descontinuidade no vídeo.

Índice de continuidade, definido em [Moraes, 2009], representa o quanto o usuário obteve o vídeo de forma contínua, ou seja, sem interrupções. Quando um pedaço chega após o seu tempo de reprodução, significa que o usuário ficou provavelmente com a imagem do último quadro congelada até a chegada deste pedaço ou que um *timeout* ocorreu, gerando descontinuidade na reprodução do vídeo e desconforto ao usuário.

O tempo total em que a reprodução do vídeo fica parada aguardando o pedaço atrasado chegar para reproduzi-lo é chamado de *tempo de espera*. De posse do tempo de espera do nó é possível calcular o índice de continuidade do mesmo. Tem-se que:

$$c = \frac{tr - te}{tr} \quad (2)$$

onde t_r é o tempo total de reprodução do vídeo. Quanto maior o índice de continuidade, mais tempo em que o usuário assistiu ao vídeo sem interrupções. Um índice igual a 100% significa que não houve interrupções na reprodução de vídeo de um dado nó.

Atraso inicial, na reprodução do vídeo. A estratégia definida para o momento certo de inicializar a reprodução de um vídeo tem importância fundamental no sucesso da qualidade do vídeo ao longo da reprodução. [Carlsson, 2007] apresenta algumas políticas de como determinar o melhor momento para iniciar o vídeo.

O simulador P4S/PeerSim não possui um escalonador para a reprodução de vídeo. Como era importante visualizar a tendência do atraso inicial e da perda de continuidade da rede P2P em face a adição de nós não-cooperativos, definimos uma política simples para inicializar o vídeo: aguardar a chegada dos três primeiros pedaços, (0, 1 e 2). Esta política é aplicada aos arquivos de resultado das simulações. [Lima, 2010] oferece uma discussão ampla do problema e seus efeitos na distribuição de vídeo.

5. Avaliação do comportamento de nós maliciosos

Devido a limitação de espaço, apresentaremos apenas os gráficos referentes à simulação do cenário com 120 pedaços. As Figuras 1(a), 1(b), 2(a) e 2(b) apresentam os gráficos de desempenho para as métricas tempo máximo, índice de continuidade (ICO), qualidade do sistema de vídeo (Q) e atraso inicial, respectivamente.

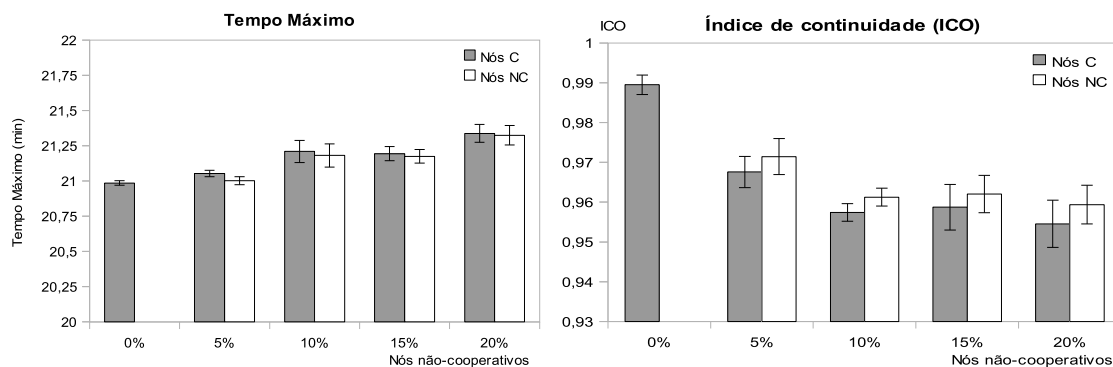


Figura 1. Desempenho: (a) tempo máximo e (b) índice de continuidade

O eixo y dos gráficos das figuras 1 e 2(a) representam as médias das métricas avaliadas obtidas pela rede com um intervalo de confiança de 90%, já o eixo x representa a porcentagem de nós não-cooperativos na rede sobreposta. Barras cinzas

correspondem ao desempenho dos nós cooperativos (nós C), barras brancas correspondem ao desempenho dos nós não-cooperativos (nós NC).

Observa-se, na Figura 1(a), que o tempo máximo dos nós cooperativos e não-cooperativos são muito próximos, com pequena vantagem para os nós não-cooperativos. O melhor desempenho dos nós da rede ocorre quando a quantidade de nós não-cooperativos é igual a zero, e o desempenho piora a medida que a percentagem de nós não-cooperativos na rede aumenta. No melhor caso, todos os nós terminaram em até 21 min e no pior caso em até 21 min e 21 seg, lembrando que o vídeo é de 20 min e que há um atraso inicial para a reprodução do vídeo.

A Figura 1(b) apresenta o desempenho da rede com relação a métrica ICO. Observa-se uma degradação do ICO à medida que a porcentagem de nós não-cooperativos aumenta. Assim como na métrica anterior, o melhor desempenho também foi para a rede sem nós não-cooperativos. O ICO para o melhor caso é de 0,99, ou seja, em 20 min de reprodução de vídeo houve em média 12 seg de interrupção. Já para o pior caso, que também corresponde a rede com 20% de nós não-cooperativos, o ICO é igual a 0,95, ou seja, média de 60 seg de interrupção. Para esta situação verifica-se uma diferença de 4 pontos percentuais.

O melhor desempenho da métrica ICO para os nós não-cooperativos era esperado, pois como os mesmos não perdem tempo no estado *push*, trocando de imediato para o estado *pull*, tendem a preencher mais rapidamente os “buracos” do seu *buffer* de reprodução, desde que, possuam nós cooperativos como seus vizinhos.

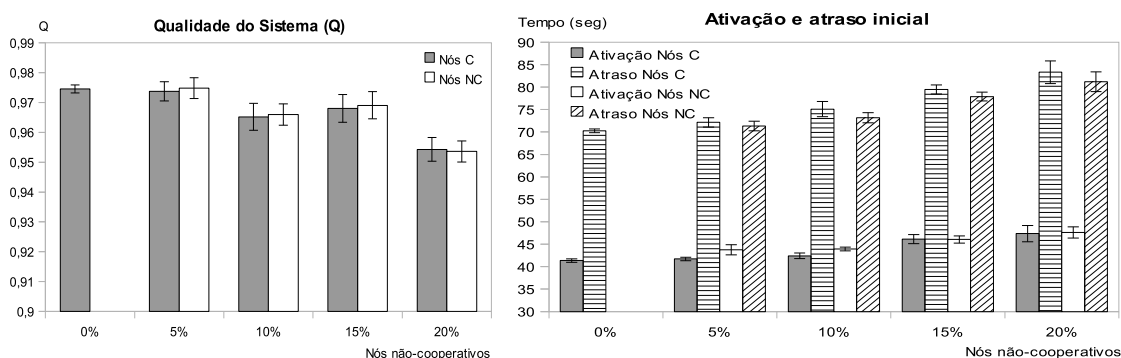


Figura 2. Desempenho: (a) qualidade do sistema de vídeo e (b) atraso inicial

A Figura 2(a) apresenta o desempenho da rede com relação a métrica qualidade (Q). Percebe-se que o comportamento dos nós não-cooperativos e cooperativos ficam praticamente iguais. Além disso, ocorre uma degradação da qualidade à medida que a porcentagem de nós não-cooperativos aumenta, conforme ocorreu com as métricas anteriores. O melhor Q é de 0,97, ou seja, de 120 pedaços apenas 3,6 pedaços chegaram atrasados em cada nó. No pior caso, novamente quando a rede possui 20% de nós não-cooperativos, o Q obtido é de 0,95, isto é, de 120 pedaços em média 6 pedaços chegaram atrasados em cada nó. Neste caso, houve uma perda de 2 pontos percentuais.

A Figura 2(b) apresenta o desempenho para o atraso inicial. Para analisar esta métrica, decidimos inserir também o comportamento do tempo de ativação dos nós, pois como discutido anteriormente, o nó só começa a participar da rede de forma efetiva apenas após receber o primeiro pedaço, a partir daí o nó pode solicitar e enviar pedaços. Por isso, acreditávamos que o tempo de ativação influenciava a métrica atraso inicial, só não sabíamos o quanto.

O gráfico da Figura 2(b) possui quatro grupos de barras: cinzas e brancas, correspondem ao desempenho dos nós cooperativos e não-cooperativos com relação à métrica ativação; barra hachuradas horizontalmente e hachuradas com ângulo de 45° correspondem ao desempenho dos nós cooperativos e não-cooperativos, com relação ao atraso inicial, respectivamente.

Percebe-se que o desempenho das redes com relação a esta métrica vai se degradando à medida que a porcentagem de nós não-cooperativos aumenta. As curvas dos nós cooperativos e não-cooperativo são muito similares. O melhor valor para tempo de ativação fica próximo a 40 seg. e o pior caso, com 20% de nós não-cooperativos, próximo a 50 seg, ou seja, uma diferença de 10 seg. O melhor valor para o atraso inicial fica próximo a 70 seg. e o pior valor próximo a 82 seg, uma diferença de 12 seg.

Pode ser notado que o tempo de ativação de um nó é responsável por mais da metade do atraso inicial do mesmo, o que é muito representativo. Seria de interesse estudar modificações no mecanismo de ativação do nó para otimizar o atraso inicial.

Vale a pena comentar outros dois pontos observados durante as simulações. Com valores mais altos para porcentagem de nós não-cooperativos, como 50% por exemplo, a simulação fica impraticável, pois vários nós sequer são ativados, além do desempenho das redes ficar bastante degradado. Além disso, em alguns casos, ocorreu de nós cooperativos não ativarem, o que é um ponto bastante negativo.

6. Controle de admissão

Na segunda parte das simulações introduziu-se um mecanismo de controle de admissão de vizinhos na formação do grafo, onde nós cooperativos possam selecionar vizinhos cooperativos e rejeitar vizinhos não-cooperativos. A idéia básica é a de criar um mecanismo que possibilite o nó fazer sua escolha baseada em uma reputação prévia do nó candidato a vizinho, neste caso, ou um nó cooperativo ou um nó não-cooperativo, tornando a seleção de vizinhos menos aleatória.

No simulador utilizamos o atributo criado anteriormente que identificava se o nó é cooperativo ou não-cooperativo. Para cada nó cooperativo foi criado um contador de vizinhos não-cooperativos utilizado na rotina de formação do grafo. Também foi definido um parâmetro para especificar a % máxima de nós não-cooperativos que um nó cooperativo está disposto a aceitar. Quando este contador atinge um determinado limiar, o nó cooperativo passa a rejeitar nós não-cooperativos como vizinhos. Repetimos as mesmas simulações, mantendo-se os parâmetros apresentados na Tabela 1 e as métricas usadas na primeira etapa para comparar os resultados.

6.1. Resultados

Para permitir a comparação com os resultados da primeira etapa, apenas apresentaremos os gráficos referentes à simulação com 15% de nós não-cooperativos na rede, com distribuição de 120 pedaços. O trabalho [Lima, 2010] apresenta os outros resultados.

O eixo y dos gráficos das Figuras de 3 a 4 representam as médias das métricas obtidas pela rede com um intervalo de confiança de 90%, já o eixo x representa a porcentagem máxima de nós não-cooperativos permitidos na lista de vizinhos de cada nó cooperativo.

Cada gráfico possui quatro grupos de blocos, o grupo de blocos cinzas corresponde ao desempenho dos nós cooperativos; já o grupo de blocos brancos

corresponde ao desempenho dos nós não-cooperativos ambos com o mecanismo (nós C e nós NC - CM); os grupos com barras hachuradas horizontalmente e com barras hachuradas com ângulo de 45 graus correspondem ao desempenho dos nós cooperativos e não-cooperativos respectivamente, sem o mecanismo (nós C e nós NC - SM).

A Figura 3(a) apresenta o desempenho da rede com relação ao tempo máximo. Observa-se que para uma configuração com 10% de vizinhos não-cooperativos, o tempo máximo para nós maliciosos aumentou em comparação com as simulações anteriores sob mesmas condições, ou seja, uma rede com 15% de nós maliciosos como um todo. O tempo máximo ficou próximo a $24,5 \pm 1,5$ min, ao passo que nas simulações sem mecanismo o valor era próximo a 21 min. Percebeu-se, não só neste cenário, mas nos demais também, que a variação foi pequena em termos gerais, entretanto com uma punição considerável para os nós não-cooperativos, para limiares baixo de admissão.

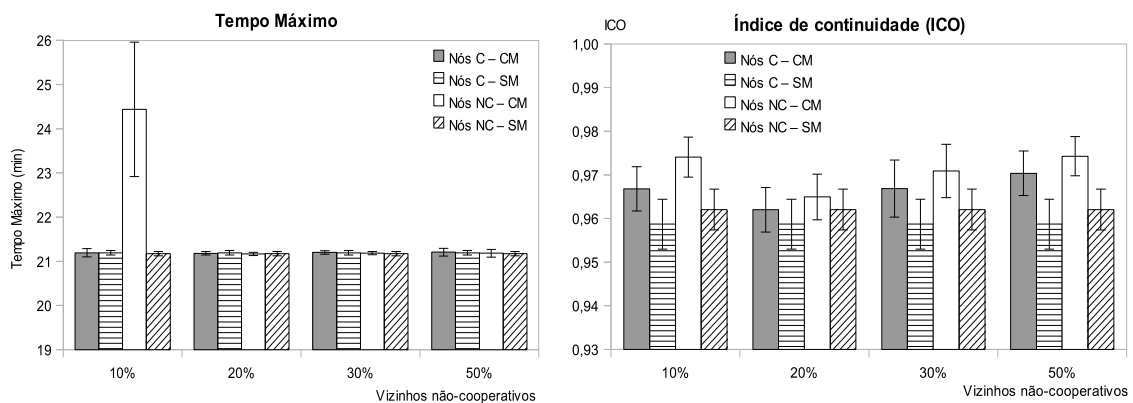


Figura 3. Desempenho: (a) tempo máximo e (b) índice de continuidade

No gráfico da Figura 3(b) podemos visualizar a curva da métrica ICO. Verificam-se que os nós não-cooperativos continuaram tendo melhor desempenho em comparação com nós cooperativos. Entretanto, podemos notar que a rede como um todo obteve uma melhora com a adição do mecanismo de admissão.

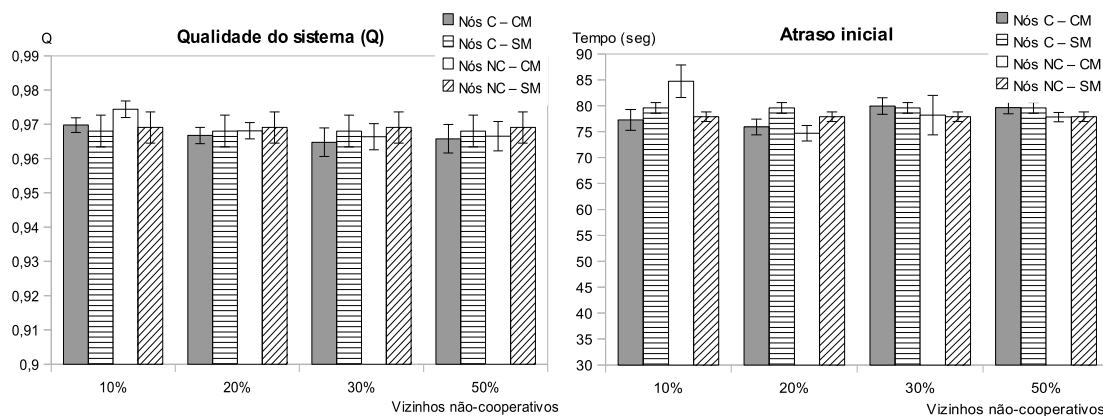


Figura 4. Desempenho: (a) qualidade do sistema de vídeo e (b) atraso inicial

A Figura 4(a) mostra que não houve melhora substancial na qualidade de vídeo com a adição do mecanismo de admissão. A Figura 4(b) apresenta o desempenho da rede sob o ponto de vista da métrica atraso inicial. Aqui notamos que há momentos em que os nós não-cooperativos são punidos e os nós cooperativos beneficiados.

Na Figura 4(b), para porcentagem de vizinhos igual a 10%, o atraso inicial dos nós cooperativos próximo a 77 ± 2 seg, e dos nós não-cooperativos próximo a 85 ± 3 seg. Sem o mecanismo, o atraso inicial para os nós cooperativos ficou próximo a 80 seg e para os nós não-cooperativos próximo a 78 seg, resultado que pode ser considerado uma melhora para os nós cooperativos e uma punição para os nós não-cooperativos.

7. Conclusão

O presente trabalho investigou como nós não-cooperativos afetam a qualidade da aplicação de distribuição de fluxos de vídeo em redes P2P que fazem uso de protocolos de difusão de pedaços *push/pull* [Cigno, 2008] onde co-existem nós cooperativos e não cooperativos. Utilizamos o simulador *PeerSim* e o módulo *P4S*, fazendo alterações e acréscimos necessários para incluir os aspectos sendo investigados.

Os resultados comprovaram que o desempenho da aplicação tende a diminuir à medida que a porcentagem de nós não-cooperativos aumenta. Observamos, também, que em geral os resultados dos nós não-cooperativos foram melhores do que os resultados dos nós cooperativos, ressaltando um outro lado do problema. Além disso, alguns nós cooperativos não foram ativados em algumas rodadas de simulação.

Com a adição de um mecanismo de seleção inicial dos vizinhos, o desempenho da aplicação melhorou para até 20% de nós não-cooperativos na lista de vizinhos dos nós cooperativos. A distribuição de nós não-cooperativos tornou-se mais homogênea, não sobrecarregando um nó cooperativo. Não se observou mais o problema de nós cooperativos não serem ativados. Este fenômeno ocorreu apenas com os nós não-cooperativos, o que funciona como um mecanismo de punição. Entretanto, o controle de admissão não conseguiu solucionar um problema: o fato de nós não-cooperativos ativados continuarem tendo melhores resultados do que nós cooperativos. Este problema está relacionado ao fato da formação do grafo ser estático. Desta forma, o bom desempenho dos nós fica condicionado à formação inicial do grafo.

Assim, estamos trabalhando em uma proposta de mecanismo dinâmico de atualização da lista de vizinhos de cada nó, a fim de introduzi-lo no simulador e estudar seus efeitos. Acreditamos que este mecanismo proporcionará o efeito de justiça desejado, introduzindo uma punição aos nós não-cooperativos que, ao longo da execução da aplicação, vão estar ligados a menos nós cooperativos, tendo a qualidade do vídeo recebido degradada. Isso serviria de incentivo para o nó não-cooperativo mudar seu comportamento.

Além do trabalho iniciado, um estudo do desempenho do protocolo *push/pull* na presença de nós não-cooperativos com gradações diferentes é considerado. Por exemplo, um nó poderia ser muito, pouco ou mediantemente não-cooperativo. Adicionalmente o comportamento do estado *pull* ou até mesmo os dois estados ao mesmo tempo poderiam ser modificados e avaliados. Neste último caso, um nó com esta característica seria muito nocivo à rede, já que não contribuiria de maneira alguma.

Por fim, um outro ponto de investigação é o tamanho ideal do pedaço para manter uma boa relação entre as métricas índice de continuidade e atraso inicial, pois pedaços pequenos podem acarretar em atraso inicial pequeno, mas desempenho pobre com várias pequenas interrupções. Já pedaços grandes podem acarretar em atraso inicial alto, mas um bom desempenho ao longo da reprodução.

Agradecimento. Os autores agradecem o apoio parcial da FAPERJ.

Referências

- ADAR, E., HUBERMAN, B.. “Free riding on Gnutella”. *First Monday*. Vol. 5, No.10. Outubro, 2000.
- CARLSSON, N., EAGER, D. “Peer-assisted on-demand streaming of stored media using BitTorrent-like protocols”. *IFIP-Tc6*, Maio, 2007.
- CIGNO, R., RUSSO, A., CARRA, D. “On Some Fundamental Properties of P2P Push/Pull Protocols”, *HUT-ICCE 2008*, pp. 67-73, Junho, 2008.
- COSTA, C., CUNHA, I., BORGES, A. “Analyzing client interactivity in streaming media”. *13th ACM Int. Conf on WWW*, Maio 17 - 20, 2004.
- HABIB, A., CHUANG, J., “Incentive mechanism for peer-to-peer media streaming”. *IWQOS 2004*, pp. 171-180, Junho, 2004.
- HALES, D. E., PATARIN, S. “Computacional sociology for system in the wild: the case of BitTorrent”, *IEEE Distributed Systems Online*, 2005.
- HOONG, P. K., MATSUO, H. “Push-Pull Incentive-based P2P Live Media Streaming System”. *WTOC 7*, 33-42, Fevereiro, 2008.
- LIMA, F. M, “Análise de desempenho de redes p2p com protocolo “push/pull” para distribuição de vídeo na presença de nós não-cooperativos”, *Dissert. de Mestrado, UERJ*, Agosto, 2010.
- LOCHER, T., MEIER, R., SCHMID, S. et all. “Push-to-Pull Peer-to-Peer Live Streaming”. *21st Int. Symp. on Distributed Computing (DISC)*. Setembro, 2007.
- MARTI, S., GIULI, T. J., LAI, K., BAKER, M. “Mitigating routing misbehavior in mobile ad hoc networks”. *6th International Conference on Mobile Computing and Networking*, 2000.
- MORAES, I., CAMPISTA, M., MOREIRA, et al. “Distribuição de Vídeo sobre Redes Par-a-Par: Arquiteturas, Mecanismos e Desafios”. *Minicurso, Cap. 3, SBRC*. Maio, 2008.
- MORAES, I. “Distribuição de vídeo sobre redes par-a-par”, *Tese de Doutorado, COPPE Elétrica, UFRJ*. Dezembro, 2009.
- P4S. “P4S: P2P 4 Streaming System”, <http://www.dit.unitn.it/networking/P4S-main.html>. [04/2010].
- PAI, V., KUMAR, K., TAMILMANI, K., et al. “Chainsaw: Eliminating Trees from Overlay Multicast”. *IPTPS*. February, 2005.
- PEERSIM. “PeerSim: A Peer-to-Peer Simulator”, <http://peersim.sourceforge.net/>. [04/2010]
- PIANESE, F., KELLER, J., BIRSACK, W. “PULSE, a Flexible P2P Live Streaming System”. *IEEE Global Internet Symposium*. Abril, 2006.
- SANGHAVI, B.S., HAJEK, B., MASSOULIE, L. “Gossiping with multiple messages”. *IEEE INFOCOM 2007*. Dezembro, 2007.
- SAROIU, S., GUMMADI, K. P., et al, “Measuring and analyzing the characteristics of Napster and Gnutella hosts”. *Multimedia Systems*, No.9, pp.170-184, Springer. Agosto, 2003.
- SUNG, Y., BISHOP, M., RAO, S. “Enabling Contribution Awareness in an Overlay Broadcasting System”. *SIGCOMM*. Setembro, 2006.
- TRAN, D. A., HUA, K. A., DO, T. T. “ZIGZAG: An efficient peer-to-peer scheme for media streaming”. *IEEE INFOCOM 2003*. Vol. 2, pp. 1283-1292, 2003.
- TRAN, D. A., HUA, K. A., DO, T. T. “A Peer-to-Peer Architecture for Media Streaming”. *IEEE JSAC*. Vol. 22, No.1, pp. 121-133, Janeiro 2004.
- VENKATARAMAN, V., FRANCIS, P. “Chunkyspread: Multi-tree Unstructured Peer-to-Peer Multicast”. *IPTPS*. Fevereiro, 2006.
- XINYAN, Z., JIANGCHUAN, L., et al., “CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming”. *INFOCOM*. Vol.3, pp.2102-2111, 2005.