

**Mecanismos de Incentivo à Cooperação em Redes
Peer-to-Peer para Distribuição de Fluxos de Vídeo**

Daniel Antonio Garcia Manzato

Dissertação de Mestrado

Mecanismos de Incentivo à Cooperação em Redes *Peer-to-Peer* para Distribuição de Fluxos de Vídeo

Daniel Antonio Garcia Manzato

Fevereiro de 2006

Banca Examinadora:

- Prof. Dr. Nelson Luis Saldanha da Fonseca
Instituto de Computação, UNICAMP (Orientador)
- Prof. Dr. Célio Vinicius Neves de Albuquerque
Instituto de Computação, UFF
- Profa. Dra. Islene Calciolari Garcia
Instituto de Computação, UNICAMP
- Prof. Dr. Edmundo Roberto Mauro Madeira
Instituto de Computação, UNICAMP

| | |
|--------------|-------------------------------------|
| UNIVERSIDADE | BC |
| Nº CHAMADA: | T/UNICAMP |
| | M319m |
| V. | Ed. |
| TOMBO BC/ | 72742 |
| PROC. | 16.145-07 |
| C | <input type="checkbox"/> |
| D | <input checked="" type="checkbox"/> |
| PREÇO | 24,00 |
| DATA | 30/05/07 |
| BIB-ID | 412214 |

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**
Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Manzato, Daniel Antonio Garcia.

M319m Mecanismos de incentivo à cooperação em redes peer-to-peer para distribuição de fluxos de vídeo / Daniel Antonio Garcia Manzato -- Campinas, [S.P. :s.n.], 2006.

Orientador: Nelson Luis Saldanha da Fonseca.

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas multimídia. 2. Redes de computação. 3. Simulação por computador. I. Fonseca, Nelson Luis Saldanha da. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: "Incentive mechanisms for cooperation in peer-to-peer media streaming networks"

Palavras-chave em inglês (Keywords): 1. Multimedia systems. 2. Computer networks. 3. Computer simulation.

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Nelson Luis Saldanha da Fonseca (IC-UNICAMP)
Prof. Dr. Célio Vinícius Neves de Albuquerque (IC-UFF)
Prof. Dra. Islene Calciolari Garcia (IC-UNICAMP)
Prof. Dr. Edmundo Roberto Mauro Madeira (IC-UNICAMP)

Data da defesa: 21/02/2006

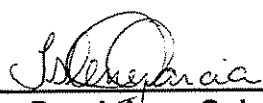
Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

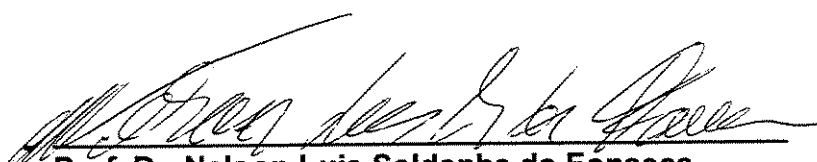
Tese defendida e aprovada em 21 de fevereiro de 2006, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Célio Vinicius Neves de Albuquerque
IC / UFF.



Profa. Dra. Islene Calciolari Garcia
IC / UNICAMP.



Prof. Dr. Nelson Luis Saldanha da Fonseca
IC / UNICAMP.

2007 2 23 108

Mecanismos de Incentivo à Cooperação em Redes *Peer-to-Peer* para Distribuição de Fluxos de Vídeo

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Daniel Antonio Garcia Manzato e aprovada pela Banca Examinadora.

Campinas, 21 de fevereiro de 2006.

Prof. Dr. Nelson Luis Saldanha da Fonseca
Instituto de Computação, UNICAMP
(Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

© Daniel Antonio Garcia Manzato, 2006.
Todos os direitos reservados.

Resumo

Os serviços de vídeo sob demanda estão ganhando popularidade com a disseminação das redes de banda larga. Apesar dos esforços realizados nos últimos anos sobre a arquitetura cliente-servidor, ela ainda se mostra bastante ineficiente em termos de escalabilidade para este tipo de serviço, uma vez que gera uma grande quantidade de tráfego de rede, processamento e consumo de memória de forma centralizada nos servidores. As redes *peer-to-peer* contornam este problema, explorando o crescente deslocamento de gargalo de recursos dos clientes para os servidores, inclusive de largura de banda. Conseqüentemente, elas são cada vez mais consideradas para a provisão de vídeo, podendo complementar ou substituir a arquitetura cliente-servidor. Entretanto, a distribuição de vídeo em redes *peer-to-peer* também apresenta desafios. Um dos problemas é o comportamento egoísta dos usuários, que procuram tirar o máximo de proveito dos recursos da rede, cooperando com o mínimo na distribuição dos fluxos. Um outro desafio é o curto tempo de permanência dos clientes no sistema, o que demanda esforços para se contornar as interrupções causadas nas entregas de fluxos com sensibilidade temporal quando as desconexões ocorrem. Uma possível solução para estes desafios é o uso de mecanismos de incentivo que estimulem os clientes a cooperarem e também a permanecerem conectados durante intervalos de tempo maiores, de forma a reduzir os problemas de desconexão. O presente trabalho propõe e avalia um novo mecanismo de incentivo para sistemas *peer-to-peer* de fluxo de mídia ao vivo (com distribuição sincronizada), a fim de tornar o sistema escalável, maximizar a qualidade recebida, bem como amenizar os problemas decorrentes das rupturas em árvores de distribuição.

Abstract

Video on demand services have become increasingly popular due to the development of new Internet access technologies such as DSLs and HFCs. In spite of the progress of the client-server architecture in the past years, it still has poor scalability, since in this architecture a huge amount of network traffic, processing and memory consumption is generated at the servers. Peer-to-peer networks provide scalability to video on demand systems, exploiting the increasing shift of abundant resources from servers to clients, including network bandwidth. As a consequence, they have been increasingly used for video on demand services, either completely replacing or simply complementing the client-server architecture. However, the use of peer-to-peer networks for video on demand services has also its own challenges. One of the existent problems is the selfish behavior of non-cooperative users, who try to maximize their utility by consuming as much resources as possible and cooperating the least for the service. Another existent problem is the short time users stay connected to the system, which causes frequent disruptions on the delivery of time-constrained content due to client disconnections. Such type of system requires incentive mechanisms that stimulate clients not only to cooperate, but also to stay connected longer. In this work, we propose and evaluate a new incentive mechanism for peer-to-peer systems with live media streaming (synchronized distribution), aiming to increase scalability and the quality of reception as well as to decrease the number of disruptions.

À minha mãe Lucia e ao meu pai Valdomiro.

Agradecimentos

Agradeço, primeiramente, a Deus, causa primária de todas as coisas. Sou muito grato a Ele pelo dom da vida, pelas inúmeras razões que tenho para sorrir, pelas dificuldades que me fazem evoluir, pela proteção constante que recebo e por Sua bondade infinita em permitir tantas coisas boas em meu caminho, apesar de minhas imperfeições.

Agradeço, com muito amor, aos meus queridos pais, por todo o apoio recebido em mais esta etapa de minha vida. Devo a eles tudo o que sou, o que sei e cada conquista realizada, graças à educação, orientação e incentivos que me deram. À minha mãe Lucia e ao meu pai Valdomiro dedico este trabalho, na esperança de conseguir retribuir um pouquinho o universo de coisas boas que eles sempre me proporcionaram.

Ainda em família, agradeço aos meus queridos irmãos, Gustavo e Marcelo, por sempre terem sido motivo de orgulho e de alegria para mim. Tive a felicidade de ter meus irmãos como melhores amigos, podendo contar com suas amizades, cumplicidades e dedicações sem limites. À minha querida tia Josefa, a quem considero como segunda mãe e a quem tenho como exemplo de vida. Aos meus queridos avós padrinhos, vovó Maria e vovô Henrique, por todo o carinho e amor que sempre me dedicaram. Aos meus queridos avós em memória, vovó Gema e vovô José, pela boa lembrança que tenho dos momentos que passamos juntos.

Agradeço, em especial, ao meu orientador, Prof. Nelson Fonseca, cuja ajuda foi fundamental em todas as fases deste trabalho. Agradeço por compreender minhas limitações e, mesmo assim, acreditar no meu trabalho. E também por todos os incentivos recebidos, orientações, cobranças interessadas, atendimentos fora de hora e críticas construtivas, que só me fizeram crescer.

Agradeço aos meus professores da USP de São Carlos, pela formação acadêmica recebida. Em particular, ao Prof. Edson dos Santos Moreira, Profa. Maria da Graça Campos Pimentel e Profa. Maria das Graças Volpe Nunes, pela amizade e também por terem me recomendado nos processos de seleção do mestrado, em 2003, e do doutorado, em 2005, do Instituto de Computação da Unicamp. Ao Prof. Edson, que foi meu orientador de iniciação científica na USP, de 1994 a 1998, agradeço ainda as conversas que tivemos em São Paulo em 2002, que me ajudaram na decisão de voltar para a universidade.

Agradeço ao Prof. Edmundo Roberto Mauro Madeira e Profa. Célia Picinin de Mello, do

Instituto de Computação da Unicamp, com quem fiz disciplinas no mestrado e que também me recomendaram no processo de seleção do doutorado.

Agradeço aos amigos Sr. Alberto Dabus, Alberto Dabus Filho e Alexandre Dabus, pelas inúmeras conversas que me ajudaram a direcionar minha carreira, pelas oportunidades de aprendizado pessoal e profissional, e também pela amizade, consideração e apoio constante.

Agradeço aos amigos Yurion Francisco Munhoz e Roger Carneiro Soares, por diversas ajudas recebidas, principalmente no período do mestrado, e também pela amizade antiga e inestimáveis confiança e consideração.

Agradeço aos amigos de república em Campinas, Arthur, Cláudio, Éric e Felipe, pelas conversas amigas, pela paciência que tiveram comigo nos momentos difíceis, por se preocuparem com o andamento do meu mestrado, pela amizade que se solidificou neste período e pela constante prestimosidade que temos em casa.

Agradeço ao pessoal do Laboratório de Sistemas Distribuídos, Sala 92 de doutorado e aos funcionários do Instituto, pela convivência agradável e inúmeros favores. Por auxílios em específico, agradeço, em particular, a André Santanchè, Luciano Digiampietri, Ulisses Silva, Rodrigo Tomita, Marcelo Uesono, Carlos Senna, Patrick Brito, Gilberto Pastorello, Leonardo Tizzei, Nielsen Simões, André Drummond, Lásaro Camargos, Augusto Devegili, Flávio Luzia, Carlos Frolidi, Rodrigo Canto e José Furlan.

Agradeço aos amigos do UOL, onde trabalhei de 1999 a 2003, pelo apoio que me deram quando decidi deixar a empresa para voltar à academia. Agradeço a Paulo Eduardo Macagnani, Alexandre Rodrigues dos Santos e Carlos Freitas, três grandes chefes que tive nas empresas onde trabalhei, pelo muito que me ensinaram no começo de minha carreira. Eu gostaria que eles soubessem o quanto admiro seus trabalhos, incluindo suas competências técnicas e seus talentos em lidar com pessoas.

É bem provável que eu tenha esquecido, injustamente, de mencionar alguns nomes importantes nestes agradecimentos, afinal, não é fácil de se concluir uma dissertação de mestrado com a mente ilesa. Peço a essas pessoas, portanto, que relevem a minha falha, e que aceitem meus sinceros agradecimentos da mesma forma. Obrigado!

Sumário

| | |
|---|-------------|
| Resumo | xi |
| Abstract | xiii |
| Agradecimentos | xvii |
| 1 Introdução | 1 |
| 2 Redes <i>Peer-to-Peer</i> de Fluxo de Mídia | 5 |
| 2.1 Vídeo sob Demanda e a Arquitetura Cliente-Servidor | 5 |
| 2.2 Redes <i>Peer-to-Peer</i> de Compartilhamento de Arquivos | 7 |
| 2.3 Redes <i>Peer-to-Peer</i> de Fluxo de Mídia | 8 |
| 2.3.1 Codificação em Múltiplos Descritores | 9 |
| 2.3.2 O Sistema CoopNet | 11 |
| 2.3.3 O Sistema SplitStream | 14 |
| 2.4 Considerações Finais | 16 |
| 3 Mecanismos de Incentivo à Cooperação | 17 |
| 3.1 Comportamentos Não-Cooperativos | 19 |
| 3.2 Padrões de Incentivo | 22 |
| 3.2.1 Padrões Baseados em Confiança | 22 |
| 3.2.2 Padrões Baseados em Comércio | 27 |
| 3.3 Mecanismos de Incentivo em Sistemas Autônomos | 35 |
| 3.3.1 Altruísmo em Redes <i>Peer-to-Peer</i> de Fluxo de Mídia | 35 |
| 3.3.2 <i>Multicast</i> com Incentivo em Redes <i>Peer-to-Peer</i> de Fluxo de Mídia | 37 |
| 3.3.3 Seleção de Clientes com Diferenciação de Serviço | 38 |
| 3.3.4 Incentivos em BitTorrent | 39 |
| 3.3.5 Comércio de Selos, Fichas e Confiabilidades | 40 |
| 3.3.6 Mobilidade em Redes sem Fio <i>Ad Hoc</i> com Incentivos | 41 |
| 3.4 Considerações Finais | 42 |

| | | |
|----------|--|------------|
| 4 | Uma Proposta de Mecanismo de Incentivo | 45 |
| 4.1 | Proposta de um Novo Mecanismo de Incentivo | 45 |
| 4.2 | Integração em CoopNet | 50 |
| 4.3 | Considerações Finais | 51 |
| 5 | Um Simulador de Redes <i>Peer-to-Peer</i> de Fluxo de Mídia ao Vivo | 53 |
| 5.1 | Arquitetura | 53 |
| 5.2 | Configuração | 56 |
| 5.2.1 | Variáveis Aleatórias | 57 |
| 5.2.2 | Parâmetros para Configuração do Simulador | 64 |
| 5.3 | Ferramentas Utilizadas | 76 |
| 5.4 | Considerações Finais | 77 |
| 6 | Avaliação do Mecanismo Proposto | 79 |
| 6.1 | Métricas Analisadas | 79 |
| 6.1.1 | Qualidade Média de Fluxo Recebida | 80 |
| 6.1.2 | Capacidade de Admissão do Sistema | 80 |
| 6.1.3 | Probabilidade de Bloqueio do Sistema | 81 |
| 6.1.4 | Taxa de Rupturas em Árvores | 81 |
| 6.2 | Determinação dos Parâmetros do Mecanismo Proposto | 82 |
| 6.2.1 | Limites Inferior e Superior da Taxa de Cooperação | 82 |
| 6.2.2 | Tempo Requerido de Cooperador Estável | 88 |
| 6.3 | Resultados dos Experimentos de Simulação | 95 |
| 6.3.1 | Experimentos Realizados | 95 |
| 6.3.2 | Benefícios do Incentivo Primário do Mecanismo Proposto | 97 |
| 6.3.3 | Benefícios do Incentivo Secundário do Mecanismo Proposto | 105 |
| 6.4 | Considerações Finais | 112 |
| 7 | Conclusão | 115 |
| 7.1 | Contribuições | 116 |
| 7.2 | Trabalhos Futuros | 117 |
| | Bibliografia | 120 |

Lista de Tabelas

| | | |
|-----|---|----|
| 4.1 | Variáveis do mecanismo de incentivo proposto. | 48 |
| 5.1 | Classes de conexão criadas no simulador desenvolvido e suas respectivas propriedades. | 63 |
| 6.1 | Variação dos parâmetros <i>LimiteInferiorTaxaCooperacao</i> e <i>LimiteSuperiorTaxaCooperacao</i> para os diferentes experimentos realizados no cenário <i>5mLIS</i> | 83 |
| 6.2 | Valores selecionados no cenário <i>5mLIS</i> para os parâmetros <i>LimiteInferiorTaxaCooperacao</i> e <i>LimiteSuperiorTaxaCooperacao</i> | 90 |
| 6.3 | Variação dos parâmetros <i>LimiteInferiorTaxaCooperacao</i> , <i>LimiteSuperiorTaxaCooperacao</i> e <i>TempoRequeridoCooperadorEstavel</i> para os diferentes experimentos realizados no cenário <i>24hTRCELIS</i> | 92 |
| 6.4 | Valores selecionados no cenário <i>24hTRCELIS</i> para o parâmetro <i>TempoRequeridoCooperadorEstavel</i> , para cada par de <i>LimiteInferiorTaxaCooperacao</i> e <i>LimiteSuperiorTaxaCooperacao</i> escolhido anteriormente. | 94 |
| 6.5 | Diferentes experimentos realizados nos cenários <i>24h</i> e <i>5mFC</i> , pertencentes aos grupos (1) sem incentivo, (2) escambo e (3) reputação. | 97 |

Lista de Figuras

| | | |
|------|---|-----|
| 3.1 | Taxonomia dos comportamentos não-cooperativos. | 19 |
| 3.2 | Terminologia de uma transação de cooperação. | 20 |
| 3.3 | Taxonomia dos padrões de incentivo. | 23 |
| 4.1 | Determinação dos parâmetros R_i , TC_i , t_i e t_* do mecanismo de incentivo proposto. | 49 |
| 4.2 | Determinação dos parâmetros B_{Ei} e B_{Si} do mecanismo de incentivo proposto, baseados em TC_i | 49 |
| 5.1 | Arquitetura do simulador desenvolvido. | 54 |
| 6.1 | Médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário $5mLIS$ (gráfico de barras). | 84 |
| 6.2 | Médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário $5mLIS$ (gráfico de superfície). | 85 |
| 6.3 | Médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário $5mLIS$ (gráfico de projeção da superfície). | 85 |
| 6.4 | Médias das capacidades de admissão do sistema, para cada experimento do cenário $5mLIS$ (gráfico de barras). | 88 |
| 6.5 | Médias das capacidades de admissão do sistema, para cada experimento do cenário $5mLIS$ (gráfico de superfície). | 89 |
| 6.6 | Médias das capacidades de admissão do sistema, para cada experimento do cenário $5mLIS$ (gráfico de projeção da superfície). | 89 |
| 6.7 | Médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário $24hTRCELIS$ | 93 |
| 6.8 | Medianas das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário $24hTRCELIS$ | 94 |
| 6.9 | Médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário $24h$ | 99 |
| 6.10 | Médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário $5mFC$ | 100 |

| | | |
|------|--|-----|
| 6.11 | Probabilidades de bloqueio do sistema, para cada experimento do cenário <i>24h</i> | 101 |
| 6.12 | Probabilidades de bloqueio do sistema, para cada experimento do cenário <i>5mFC</i> | 102 |
| 6.13 | Médias das capacidades de admissão do sistema, para cada experimento do cenário <i>24h</i> | 103 |
| 6.14 | Médias das capacidades de admissão do sistema, para cada experimento do cenário <i>5mFC</i> | 104 |
| 6.15 | Médias das taxas de rupturas em árvores sofridas pelos clientes, para cada experimento do cenário <i>24h</i> | 106 |
| 6.16 | Médias das taxas de rupturas em árvores sofridas pelos clientes, para cada experimento do cenário <i>5mFC</i> | 107 |
| 6.17 | Médias das taxas de rupturas em árvores sofridas pelos clientes, para cada experimento do cenário <i>5mFCSC</i> | 110 |
| 6.18 | Médias das taxas de rupturas em árvores sofridas pelos clientes, para cada experimento do cenário <i>5mFC32A</i> | 112 |
| 6.19 | Médias das taxas de rupturas em árvores sofridas pelos clientes, para cada experimento do cenário <i>5mFCSC32A</i> | 113 |

Capítulo 1

Introdução

Os serviços de vídeo sob demanda (do Inglês, *video on demand* ou VoD) estão ganhando cada vez mais popularidade com a disseminação das redes de banda larga. As aplicações nesta área incluem distribuição de noticiários, ensino à distância, bibliotecas digitais e distribuição de vídeos de entretenimento. O desenvolvimento de novas tecnologias de acesso à Internet, como DSLs e HFCs, tem viabilizado este tipo de sistema. Com essas tecnologias, o gargalo de largura de banda de rede, que tipicamente estava no lado dos clientes, foi transferido para o lado dos servidores.

A arquitetura tradicional cliente-servidor, utilizada na distribuição de fluxos nos sistemas de vídeo sob demanda, tem se mostrado ineficiente em termos de escalabilidade. Apesar dos esforços realizados nos últimos anos, ela ainda gera uma grande quantidade de tráfego de rede, processamento e consumo de memória de forma centralizada no lado dos servidores. A consequência disso é que o alto custo demandado por uma grande quantidade de clientes utilizando o serviço simultaneamente torna o serviço viável apenas a grandes sites comerciais.

Uma grande parte dos trabalhos realizados em cima da arquitetura cliente-servidor explora *multicast* em nível de rede, com o intuito de aliviar a sobrecarga nos servidores [24, 46, 50, 29]. É sabido, no entanto, que a sua utilização apenas transfere esta sobrecarga ao roteador, que fica incumbido de transmitir o fluxo único recebido do servidor a todos os endereços cadastrados no grupo *multicast*. Por se temer sobrecarga nos roteadores, é comum a desabilitação deste tipo de serviço na maioria das redes, principalmente nas que abrangem diferentes domínios [32, 59, 62].

O avanço tecnológico reduziu bastante a distância entre os computadores pessoais e os servidores de conteúdo das redes. Este fato, juntamente com a disseminação das redes de banda larga, sugere a abundância – e muitas vezes a ociosidade – de recursos (largura de banda, processamento e memória) nas estações clientes. Recursos estes que são justamente o gargalo no lado dos servidores imposto pelas arquiteturas cliente-servidor.

As redes *peer-to-peer* têm como característica explorar esses recursos ociosos nas estações clientes. Trata-se de uma organização de serviços na qual cada estação armazena o seu próprio

conteúdo e o compartilha com as demais. Esta abordagem é altamente escalável, permite a implementação de *multicast* em nível de aplicação e apresenta custo reduzido quando comparada às arquiteturas cliente-servidor. Sendo assim, as redes *peer-to-peer* estão sendo cada vez mais consideradas para a provisão de vídeo sob demanda, podendo complementar ou substituir a arquitetura cliente-servidor.

Em redes *peer-to-peer* para serviços de fluxo de mídia (do Inglês, *streaming media*), pode-se classificar os serviços em dois tipos: fluxo ao vivo e fluxo sob demanda. No primeiro tipo, o conteúdo é distribuído de forma sincronizada entre clientes, muitas vezes através de árvores que implementam *multicast* em nível de aplicação. No segundo tipo, o conteúdo é distribuído sob demanda e de forma assíncrona, podendo diferentes clientes servir diferentes partes de um mesmo fluxo para uma determinada requisição.

A distribuição de vídeo em redes *peer-to-peer* apresenta, também, desafios. Um dos problemas encontrados é o comportamento egoísta dos usuários, que procuram tirar o máximo de proveito dos recursos da rede, cooperando com o mínimo na distribuição dos fluxos. Estudos realizados em redes *peer-to-peer* para serviços de compartilhamento de arquivos (do Inglês, *file sharing*), contexto para o qual elas foram originalmente concebidas, mostraram que menos de 30% de clientes altruístas sustentam a maioria de todas as requisições da rede [1]. No contexto de serviços de fluxo de mídia, nos quais há sensibilidade temporal do conteúdo distribuído, níveis maiores de colaboração são imprescindíveis para viabilizar o sistema [27, 26, 28].

Um outro problema encontrado é o curto tempo de permanência dos clientes na rede, o que demanda esforços para se contornar as interrupções causadas nas entregas de fluxo com sensibilidade temporal quando as desconexões ocorrem [45]. Tal problema é particularmente importante em *multicast* na camada de aplicação, pois dependendo da posição de um nó nas árvores de distribuição, diversos clientes podem ser afetados quando ele se desconecta da rede [43]. Faz-se necessário, então, o uso de mecanismos de incentivo que estimulem os clientes a cooperarem e também a permanecerem conectados durante intervalos de tempo maiores, de forma a reduzir os problemas de desconexão.

Este trabalho tem como objetivo propor e avaliar um novo mecanismo de incentivo para sistemas *peer-to-peer* de fluxo de mídia ao vivo, a fim de tornar o sistema escalável, maximizar a qualidade recebida, bem como amenizar os problemas decorrentes das rupturas em árvores de distribuição. Dentre os sistemas existentes para este tipo de serviço, considerou-se, em particular, o descrito em [44, 43], chamado CoopNet. Apesar de se ter trabalhado sobre este sistema em específico, o novo mecanismo proposto pode ser aplicado, de forma geral, a qualquer outro sistema que considere distribuição sincronizada de fluxo através de múltiplas árvores de distribuição, como também ocorre, por exemplo, em SplitStream [11].

Esta dissertação está organizada da seguinte maneira. No Capítulo 2, discute-se alguns sistemas *peer-to-peer* existentes, com enfoque para a utilização dos mesmos em serviços de vídeo sob demanda. Comenta-se, inicialmente, algumas frentes de pesquisa para serviços de

vídeo ainda sobre a arquitetura cliente-servidor, de forma a contrastar suas limitações com os sistemas *peer-to-peer*. Em seguida, descreve-se as diferentes abordagens utilizadas, ao longo do tempo, para as redes *peer-to-peer* de compartilhamento de arquivos. Por fim, estuda-se sistemas *peer-to-peer* para serviços de fluxo de mídia, que constituem o principal interesse do presente trabalho.

No Capítulo 3, realiza-se um estudo sobre mecanismos de incentivo à cooperação, com enfoque para o projeto dos mesmos. Explica-se, inicialmente, os diferentes tipos de comportamentos não-cooperativos encontrados. Em seguida, os padrões de incentivo existentes são analisados, considerando-se propriedades como tipo de remuneração, eficácia contra comportamentos não-cooperativos, anonimato e escalabilidade. Por fim, uma seleção de trabalhos relacionados a esquemas de incentivo é apresentada, comparando-os aos padrões vistos anteriormente.

No Capítulo 4, introduz-se o novo mecanismo de incentivo proposto no presente trabalho. Apresenta-se, inicialmente, a sua proposta, com definição, variáveis e algoritmos. Em seguida, descreve-se uma forma de agregá-lo ao sistema CoopNet, através de modificações no protocolo de adaptação de banda apresentado em [44].

No Capítulo 5, apresenta-se o simulador desenvolvido para a avaliação do mecanismo de incentivo proposto. Discute-se, inicialmente, a sua arquitetura. Em seguida, as suas configurações são descritas, incluindo as variáveis aleatórias usadas e os parâmetros de configuração criados. Por fim, menciona-se as ferramentas utilizadas no desenvolvimento.

No Capítulo 6, discute-se os resultados obtidos nas simulações executadas. Descreve-se, inicialmente, as métricas coletadas e analisadas. Em seguida, estuda-se o procedimento realizado para a determinação experimental dos parâmetros responsáveis pelo funcionamento do mecanismo de incentivo proposto. Na seqüência, avalia-se o sistema com diferentes configurações de incentivo, tanto em condições normais de tráfego quanto em situações de *flash crowd*. Por fim, analisa-se os benefícios obtidos e os custos incorridos por cada estratégia adotada no mecanismo proposto.

No Capítulo 7, finalmente, encerra-se o presente trabalho com as conclusões finais. Discute-se, inicialmente, as contribuições alcançadas. Por fim, menciona-se possíveis trabalhos futuros a serem considerados.

Capítulo 2

Redes *Peer-to-Peer* de Fluxo de Mídia

Neste capítulo, discute-se alguns sistemas *peer-to-peer* existentes, com enfoque para a utilização dos mesmos em serviços de vídeo sob demanda. Na Seção 2.1, comenta-se algumas frentes de pesquisa para serviços de vídeo ainda sobre a arquitetura cliente-servidor, de forma a contrastar suas limitações com os sistemas *peer-to-peer*. Na Seção 2.2, descreve-se as diferentes abordagens utilizadas, ao longo do tempo, para as redes *peer-to-peer* de compartilhamento de arquivos. Na Seção 2.3, estuda-se os sistemas *peer-to-peer* para serviços de fluxo de mídia, que constituem o principal interesse do presente trabalho. Finalmente, na Seção 2.4, o capítulo é encerrado com algumas considerações finais.

2.1 Vídeo sob Demanda e a Arquitetura Cliente-Servidor

Um sistema típico de vídeo sob demanda consiste em um conjunto de servidores centralizados e clientes geograficamente distribuídos, conectados através de redes rápidas. Um grande número de arquivos de vídeo é armazenado nos servidores e reproduzido nos clientes. Antes de aceitar a requisição de um cliente, o servidor precisa reservar uma capacidade suficiente de processamento, memória e largura de banda de rede para o envio do fluxo, de forma que a continuidade da reprodução do vídeo seja garantida. Devido às altas exigências de largura de banda impostas pelos fluxos de vídeo, este recurso no servidor é considerado o mais caro neste tipo de serviço [46]. Dessa maneira, uma parte crítica no projeto de um sistema de vídeo sob demanda é a minimização da largura de banda de rede requerida no servidor.

Como conseqüência, a arquitetura cliente-servidor apresenta duas grandes limitações. Primeiro, ela não é escalável para um grande número de clientes. Segundo, a qualidade do fluxo transmitido é limitada por esse gargalo. Visando amenizar essas limitações, vários trabalhos exploram o uso de caches em servidores *proxy* para os fluxos multimídia [48, 16, 46, 63, 59, 62]. Através do uso desta técnica para fluxos populares, por exemplo, pode-se reduzir a carga da rede aproximando o conteúdo dos clientes. Com isso, minimiza-se tanto o gargalo no servidor

quanto a latência de recepção, bem como maximiza-se as qualidades de fluxo recebidas.

A latência de recepção, por sua vez, também é uma questão importante em sistemas de vídeo sob demanda. O serviço é considerado “sob demanda” porque se espera que os vídeos cheguem aos clientes tão logo tenham sido requisitados no servidor. Um dos principais desafios no projeto deste tipo de sistema é como fazer o gerenciamento de um grande número de clientes mantendo um custo de operação baixo e, ao mesmo tempo, provendo uma latência de recepção aceitável [24].

As técnicas de distribuição de conteúdo em sistemas de vídeo sob demanda podem ser classificadas em duas categorias: broadcast periódico e *multicast* escalonado. A primeira categoria é também conhecida como abordagem centralizada nos dados ou pró-ativa. Neste caso, os canais do servidor são dedicados a determinados objetos de vídeo que são transmitidos repetidamente em broadcast, segundo as técnicas de distribuição existentes (como, por exemplo, *Staggered*, *Pyramid*, *Permutation-based pyramid*, *Skyscraper*, *Harmonic* e *Catching*). A segunda categoria, também conhecida como abordagem centralizada no usuário ou reativa, dedica os canais do servidor a usuários individuais. Quando um canal do servidor está livre, um conjunto de requisições é servido simultaneamente, segundo as técnicas de escalonamento existentes (como, por exemplo, *Batching*, *Piggybacking*, *Patching* e *Multicast com caching*).

A replicação de servidores é uma outra estratégia que busca minimizar o problema de escalabilidade dos serviços de vídeo sob demanda, apesar de seu custo em prover servidores adicionais. Neste esquema, os servidores podem possuir as mesmas cópias dos vídeos disponibilizados [24], possuir cópias complementares [4] ou ainda assumir uma configuração dinâmica na qual os vídeos são replicados conforme suas demandas e a posição geográfica das máquinas [20]. Em todos esses casos, aspectos de seleção de servidores são levantados [24, 3].

Vale ainda ressaltar que na área de vídeo sob demanda também existem pesquisas sobre protocolos, que buscam promover sincronizações – através de sinais (mensagens) – entre servidores centrais de broadcast e servidores locais de anúncio. Com isso, oferece-se uma forma automática de se intercalar a distribuição de conteúdo por mais de um tipo de servidor [8].

Nota-se, no entanto, que apesar dos diversos esforços realizados na área de vídeo sob demanda sobre a arquitetura cliente-servidor, o problema de escalabilidade ainda persiste, sendo apenas minimizado. Isto ocorre devido à utilização de recursos de forma centralizada nos servidores, o que naturalmente impõe gargalos ao sistema. Uma abordagem distribuída faz-se portanto necessária, de forma a aproveitar melhor os recursos abundantes e ociosos das estações clientes. Esta é a principal motivação para a utilização de redes *peer-to-peer* nos serviços de vídeo sob demanda.

2.2 Redes Peer-to-Peer de Compartilhamento de Arquivos

As redes *peer-to-peer* consistem em uma organização de serviços na qual cada estação armazena o seu próprio conteúdo e o compartilha com as demais. Nelas, os dados são transferidos diretamente entre as estações clientes, sem passar por servidores. Dessa forma, elas exploram recursos como largura de banda de rede, armazenamento, memória e processamento de um grande número – algumas vezes milhões – de estações para a distribuição de conteúdo. Isso mostra o poder de escalabilidade deste tipo de organização [33].

Tendo sido concebidas inicialmente para serviços de compartilhamento de arquivos, as redes *peer-to-peer* passaram por diferentes gerações. A primeira foi implementada pelo sistema Napster [33], para o compartilhamento de músicas no formato MP3. Neste sistema, tem-se a presença de um diretório central, que mantém o cadastro de todos os arquivos disponíveis na rede, bem como dos endereços IP das estações que os disponibilizam. Apesar das estações consultarem um servidor para a localização do conteúdo, a transferência do mesmo, que é a operação mais custosa em termos de banda de rede, ocorre diretamente entre elas. A principal desvantagem desta abordagem centralizada é a existência de um ponto único de falha no sistema.

A segunda geração de redes *peer-to-peer* foi implementada pelo sistema Gnutella [22]. Nele, deixa-se de utilizar um servidor central, sendo a localização do conteúdo realizada através da técnica de inundação de consultas. Nesta técnica, cada estação dissemina requisições de busca a todas as suas vizinhas na rede *overlay*. Estas efetuam o repasse das consultas recebidas adiante, até que um limite de propagação (por exemplo, 8 saltos) tenha sido atingido ou, evidentemente, o conteúdo tenha sido encontrado. Um dos problemas existentes nesta abordagem é a necessidade de se ter nós *bootstrapping* para servirem como “portas de entrada” às estações que queiram se conectar na rede. Dado que se tem uma arquitetura totalmente descentralizada, sem a existência de nenhum servidor, um novo cliente precisa saber em quem se conectar primeiro na rede.

O sistema Kazaa [31] implementou a terceira geração de redes *peer-to-peer*. Nele, novamente não se tem um servidor central, porém, há uma hierarquia entre as estações clientes. As estações que apresentam maiores recursos em termos de conectividade são eleitas como “líderes de grupo”, passando a agrupar todas as requisições de seu grupo. Assim, quando uma estação procura um determinado arquivo, ela somente consulta a líder de seu grupo. Caso o arquivo não possa ser encontrado no grupo, a líder então repassa a consulta às demais líderes (de outros grupos), que ficam incumbidas de fazer a busca em seus próprios grupos. Esta abordagem é conhecida como semi-centralizada e também apresenta a necessidade de se ter nós *bootstrapping* na rede.

A quarta geração de redes *peer-to-peer* foi implementada pelos sistemas Pastry [49], CAN [47], Chord [54] e Tapestry [6]. Eles apresentam uma abordagem descentralizada porém es-

truturada, na qual as estações dividem um espaço de endereçamento global entre si, através da utilização de tabelas *hash* distribuídas. Neste esquema, identificadores únicos são atribuídos às estações e aos arquivos. Cada arquivo é então armazenado na estação cujo identificador seja mais próximo do seu identificador de arquivo, no espaço de endereçamento global onde estão todos os identificadores. Sendo assim, de maneira geral, dado um arquivo e uma função *hash*, sabe-se em qual estação encontrá-lo. Estes protocolos de busca e roteamento são melhor explicados na Subseção 2.3.3.

Um sistema interessante e recente de compartilhamento de arquivos que merece ser mencionado é o Bittorrent [7]. Ele é baseado na abordagem centralizada, uma vez que as estações compartilham pedaços de arquivos que são distribuídos diretamente entre elas, porém sob a coordenação de um *tracker*. Este é responsável por indicar às estações de quem obter determinados pedaços do arquivo. Além disso, controla-se o quanto se consome, em termos de banda, baseado no quanto se coopera. Assim, quanto maior a banda de saída disponibilizada para fins de cooperação, maior é a taxa de *download* obtida, o que significa que se consegue concluir a transferência mais rapidamente.

Não demorou muito para que outras finalidades, além de compartilhamento de arquivos, fossem atribuídas às redes *peer-to-peer*. A sua utilização na área de vídeo sob demanda é crescente, devido a seus altos níveis de escalabilidade, que tendem a suprir as limitações impostas pela arquitetura cliente-servidor. Um outro ponto interessante que as redes *peer-to-peer* podem superar a arquitetura cliente-servidor é a implementação de *multicast* em nível de aplicação. Conforme visto na seção anterior, diversas técnicas desenvolvidas para a provisão de vídeo baseiam-se em *multicast*, cujo uso no nível de rede é limitado ou pouco difundido. As redes *peer-to-peer* promovem naturalmente a implementação deste tipo de serviço na camada de aplicação, tirando a sua dependência da camada de rede.

Há ainda trabalhos que melhor exploram o uso de *multicast* na camada de aplicação, com relação à sua utilização na camada de rede. Nestes trabalhos, as estações, possuindo o conhecimento da topologia de rede virtual (que se forma sobre a topologia de rede real), têm condições de inferir possíveis locais de congestionamento ou desconexões inesperadas de clientes, tomando medidas apropriadas. Esta abordagem está descrita no protocolo de adaptação de heterogeneidade proposto em [44], estudado na Subseção 2.3.2.

2.3 Redes Peer-to-Peer de Fluxo de Mídia

Em redes *peer-to-peer* para serviços de fluxo de mídia, pode-se classificar os serviços em dois tipos: fluxo ao vivo e fluxo sob demanda. No primeiro tipo, o conteúdo é distribuído de forma sincronizada entre os clientes, muitas vezes através de árvores que implementam *multicast* em nível de aplicação. Sendo assim, cada cliente passa a receber o fluxo a partir do ponto de reprodução onde está (como ocorre com noticiários, por exemplo). No segundo tipo, o conteúdo

é distribuído sob demanda e de forma assíncrona, podendo diferentes clientes servir diferentes partes de um mesmo fluxo para uma determinada requisição. Neste caso, cada cliente assiste ao fluxo do início ao fim (como ocorre com filmes, por exemplo).

Em ambos os casos, um tipo particular de codificação de fluxo é comumente utilizado, visando dividir a sobrecarga de distribuição de conteúdo entre os diversos clientes da rede, bem como explorar redundância de dados e de caminhos de rede para se promover tolerância a falhas. Esta técnica de codificação, bem como dois sistemas de fluxo ao vivo que são relevantes ao presente trabalho são estudados a seguir.

2.3.1 Codificação em Múltiplos Descritores

A codificação em múltiplos descritores (do Inglês, *multiple description coding* ou MDC) é um método de codificação de áudio e/ou vídeo no qual um sinal é codificado em dois ou mais fluxos distintos, sendo cada fluxo chamado de descritor. Este método apresenta as seguintes propriedades:

- Cada descritor pode ser decodificado independentemente dos demais, gerando uma reprodução usável do sinal original;
- Os diferentes descritores contêm informações complementares, de forma que a qualidade do sinal reconstruído melhora com o aumento do número de descritores utilizados na decodificação.

Neste método, além de cada descritor ser independente um do outro, todos eles tipicamente possuem a mesma importância. Esta abordagem é oposta às tradicionais codificações em camadas, como MPEG-1/2/4, H.261/3 e alguns codificadores proprietários da Microsoft e Real Networks. Essas codificações essencialmente priorizam alguns dados, suportando um descarte seletivo quando necessário (os dados de complementação de qualidade podem ser perdidos ou descartados sem que o fluxo deixe de ser usável). No entanto, os dados da camada base são criticamente importantes – se eles forem perdidos, os demais se tornam inúteis. A codificação em múltiplos descritores supera essa limitação, garantindo uma reprodução usável do fluxo mediante a decodificação de qualquer subconjunto dos descritores e com qualidade proporcional ao número de descritores utilizados [43, 45].

Para alcançar essa flexibilidade, a codificação em múltiplos descritores incorre em uma modesta perda de desempenho se comparada à codificação em camadas. Esta, por sua vez, também incorre em uma pequena perda de desempenho se comparada à codificação em descritor único (do Inglês, *single description coding* ou SDC) [43, 45]. Nesta última, não se tem diferenciação ou priorização de nenhum subconjunto dos dados.

Quando conciliada à diversidade de caminhos (do Inglês, *path diversity*), a codificação em múltiplos descritores pode ser útil para se obter uma qualidade usável do fluxo sempre que

qualquer subconjunto dos descritores for recebido. Neste caso, cada caminho transporta um descritor diferente do mesmo fluxo e, dado que o uso de múltiplos caminhos reduz a probabilidade de se ter perdas simultâneas, pode-se melhorar as chances de recepção de um fluxo de boa qualidade na maior parte do tempo [4].

Diversos algoritmos para a codificação em múltiplos descritores têm sido desenvolvidos recentemente, provendo diferentes níveis de compressão ou tolerância a falhas. Algumas das metas buscadas por esses codificadores são [4]:

- Eficácia na compressão, buscando as propriedades da codificação em múltiplos descritores com um mínimo de acréscimo na largura de banda que os codificadores tradicionais em camadas proveriam;
- Habilidade de usar corretamente alguns descritores para o reparo de outros que estejam corrompidos. Com isso, uma qualidade usável do fluxo pode ser mantida mesmo quando existirem perdas em todos os descritores, desde que as mesmas não ocorram exatamente nos mesmos instantes de tempo;
- Habilidade de operar corretamente em caminhos de rede que possuam larguras de banda diferentes ou desbalanceadas;
- Compatibilidade com padrões existentes, como MPEG-4 e H.263, de forma que uma aplicação de reprodução convencional possa reproduzir separadamente cada descritor, como se ele fosse um fluxo único codificado pelos métodos tradicionais em camadas.

Um esquema simples de codificação em múltiplos descritores pode ser ilustrado da seguinte maneira. A seqüência de imagens de um vídeo é demultiplexada em M subseqüências, colocando-se cada M -ésima imagem, $m + iM$, $i = 0, 1, 2, \dots$, na m -ésima subseqüência, $m = 1, \dots, M$. As subseqüências são então independentemente codificadas para se formar os M descritores. Com isso, qualquer subconjunto destes M descritores pode ser decodificado e as seqüências de imagens remontadas, de forma a se reconstruir um fluxo de vídeo cuja taxa de quadros seja proporcional ao número de descritores utilizados na decodificação [45].

Técnicas mais sofisticadas de codificação em múltiplos descritores têm sido investigadas ao longo dos últimos anos. Um esquema interessante descrito em [43, 14, 45] é baseado em codificação em camadas, codificação Reed-Solomon, transmissão codificada em prioridades e alocação otimizada de bits. Neste esquema, o fluxo de vídeo e/ou áudio é particionado em grupos de quadros (do Inglês, *groups of frames* ou GOFs), tendo cada grupo uma duração aproximada de 1 segundo. Cada GOF é então codificado independentemente, protegido contra erros e o resultado subdividido em M pacotes. Se quaisquer $m \leq M$ desses pacotes forem decodificados, então os R_m primeiros bits do GOF original podem ser recuperados, ocasionando uma distorção $D(R_m)$, onde $0 = R_0 \leq R_1 \leq \dots \leq R_M$ e, conseqüentemente, $D(R_0) \geq D(R_1) \geq \dots \geq D(R_M)$.

Com isso, todos os M pacotes são igualmente importantes, sendo apenas o número deles que forem decodificados determinante na qualidade do GOF reconstruído. A distorção esperada do fluxo transmitido é dada por $\sum_{m=0}^M p(m)D(R_m)$, onde $p(m)$ é a probabilidade que m dos M pacotes sejam recebidos. Enviando-se o m -ésimo pacote de cada GOF no m -ésimo descritor, tem-se o fluxo original representado por M descritores. Cada descritor é constituído por uma seqüência de pacotes transmitidos na taxa de 1 pacote por GOF.

Em [14], os autores propuseram o conceito de codificação em múltiplos descritores em camadas (do Inglês, *layered multiple description coding* ou LMDC). Esta técnica concilia os benefícios da codificação em múltiplos descritores – como, por exemplo, o tratamento de tolerância a falhas nas transmissões dos fluxos – com os benefícios da codificação em camadas – como o tratamento da adaptabilidade de qualidade dos fluxos recebidos. Segundo os autores do referido trabalho, o provimento de diferentes qualidades de fluxo (por exemplo, para se tratar a heterogeneidade de diferentes clientes) através do oferecimento de diferentes números de descritores não é uma prática otimizada. A explicação é que existem dados redundantes em cada descritor, devendo esta redundância ser ajustada às condições de rede e aos diferentes clientes. É evidente que uma mesma configuração de redundância não poderia ser apropriada, ao mesmo tempo, para diferentes bandas de clientes heterogêneos. Por exemplo, um dado nível de redundância pode ser dispendioso e desnecessário a clientes com bandas largas, enquanto que insuficiente a clientes com bandas estreitas.

A idéia da codificação em múltiplos descritores em camadas é ter-se múltiplos descritores agrupados em diferentes camadas, cada qual com sua configuração específica de redundância. Dessa forma, pode-se otimizar a redundância das camadas base de acordo com os clientes que possuem bandas estreitas e otimizar a redundância das camadas de melhoria de acordo com os clientes que possuem bandas maiores. Sendo assim, clientes heterogêneos podem se adaptar a diferentes bandas de fluxo variando, primeiramente, o número de descritores na camada mais alta que esteja habilitado a participar; em seguida, partindo para camadas acima ou abaixo desta, caso o número atual de descritores obtidos seja, respectivamente, máximo ou mínimo naquela camada [14].

2.3.2 O Sistema CoopNet

O sistema CoopNet, proposto em [43, 45], segue uma abordagem que mescla aspectos da arquitetura tradicional cliente-servidor com aspectos das redes *peer-to-peer*. Assim como na arquitetura cliente-servidor, há um servidor central que provê todos os fluxos (pré-gravados ou ao vivo), além de servir como “porta de entrada” às novas estações que queiram se conectar na rede. Assim como nas redes *peer-to-peer*, há conexões diretamente entre os clientes que, utilizando os recursos ociosos que possuem, auxiliam o servidor na distribuição do conteúdo, provendo uma grande escalabilidade ao sistema. Trata-se de um sistema *peer-to-peer* de fluxo

de mídia ao vivo que complementa, ao invés de substituir, a arquitetura cliente-servidor.

Esta abordagem é viável e interessante uma vez que os fluxos de vídeo, demandando um grande espaço para armazenamento, tendem mesmo a ficar concentrados em *clusters* de servidores. Além disso, problemas típicos de localização de dados e de nós *bootstrapping* são resolvidos através da existência de uma entidade central. E ainda, esta entidade central pode também facilitar a aquisição e transmissão de programações ao vivo como, por exemplo, as programações de uma rádio on-line. É certo que a utilização de uma entidade central pode gerar questionamentos acerca da escalabilidade do sistema. No entanto, tem-se que a tarefa mais custosa do serviço – a transmissão do fluxo, que demanda grandes larguras de banda – é realizada de forma distribuída, contando com a cooperação dos clientes.

Assim, uma distribuição sincronizada de conteúdo aos clientes é realizada, através de *multicast* na camada de aplicação. Árvores de distribuição são formadas através de conexões *unicast*, estando o servidor em suas raízes. Cada cliente é cadastrado como nó nessas árvores, podendo ser admitido diretamente pelo servidor, caso este tenha recursos para isso, ou por outros nós que já estejam conectados no sistema, escolhidos aqueles que ainda tenham capacidade de admissão excedente. Cada cliente, então, repassa o fluxo recebido em uma determinada árvore de distribuição a todos os seus descendentes, até que o fluxo atinja os nós folhas daquela árvore. O grau de saída de cada nó do sistema é limitado pela banda de saída da respectiva entidade (servidor ou cliente). Em geral, o servidor apresenta uma banda de saída bem superior às disponibilizadas pelos clientes.

Neste tipo de sistema, os clientes tendem a permanecer na rede por um período curto de tempo (cerca de alguns minutos), inferior ao que se observa nos sistemas de compartilhamento de arquivos (cerca de uma hora, em média) [45]. Para se minimizar os problemas de rupturas em árvores decorrentes da constante entrada e saída de clientes no sistema, utiliza-se múltiplas árvores de distribuição e codificação em múltiplos descritores. Com isso, sendo cada descritor transportado por uma árvore de distribuição diferente, tem-se que o fluxo não é totalmente interrompido quando uma desconexão ocorre. Ao contrário disso, apenas um subconjunto dos descritores deixa temporariamente de ser entregue, ocasionando uma queda momentânea na qualidade de fluxo recebida.

Adotou-se uma sugestão feita em [11] que indica que cada cliente deve ser admitido como nó interior em uma única árvore, sendo nó folha nas $n - 1$ demais. Com isso, consegue-se construir árvores o mais curtas e disjuntas possível, diversificando os caminhos de rede para aumentar a tolerância a falhas, bem como melhor aproveitar as bandas de saída dos clientes (já que se um nó fosse folha em todas as n árvores, sua banda de saída não seria aproveitada).

O protocolo de admissão *top-down* proposto em [43] é descrito da seguinte forma. Quando um cliente se conecta no sistema, ele primeiramente contata o servidor, informando sua banda de saída. Assume-se que todos os clientes são homogêneos em suas bandas de entrada, ou seja, buscam ser admitidos em todas as árvores de distribuição. O servidor então decide em qual

das árvores o novo cliente será nó interior, escolhendo a árvore que possui o menor número de nós interiores. Com isso, procura-se balancear as capacidades de admissão das diferentes árvores. Na árvore escolhida, o servidor seleciona o nó mais próximo à raiz (possivelmente ela própria) que possa admitir o novo cliente, fazendo uma busca *top-down*. Caso, nesta busca, o servidor atinja um nó folha, então o novo cliente será admitido no lugar deste nó folha, e este será readmitido de forma análoga sob outro nó, possivelmente sob o próprio novo cliente, que passa a ser nó interior nesta árvore. Nas $n - 1$ árvores restantes, o novo cliente é admitido como nó folha, sob o nó mais próximo da raiz que possa admiti-lo. Neste caso, dado que o novo cliente é admitido nestas árvores como nó folha, não há vantagens em colocá-lo no lugar de um outro nó folha, caso todos os nós superiores percorridos estejam saturados.

Em [44], os autores evoluíram o sistema CoopNet para suportar heterogeneidade dos clientes e controle de congestionamento. Para viabilizar esta melhoria, utilizou-se a codificação em múltiplos descritores em camadas. Dessa forma, os clientes com bandas estreitas recebem apenas os descritores da camada base, enquanto que os demais recebem, adicionalmente, os descritores da camada de melhoria.

Além da aplicação de LMDC, uma contribuição adicional de [44] foi o protocolo de adaptação de banda proposto, que é implementado de forma integrada ao mecanismo de redundância de dados e de caminhos de rede. Nele, os nós pais e filhos atuam de forma conjunta na inferência dos locais de congestionamento (em seus enlaces de entrada, de saída ou em algum ponto intermediário do caminho de rede). Dependendo do local de congestionamento, um cliente pode optar por desconectar-se de alguns nós superiores, rejeitar nós inferiores ou ainda buscar novas rotas de rede conectando-se em novos nós superiores. De maneira geral, o protocolo de adaptação de banda opera da seguinte forma:

1. Descarte tráfego, de entrada ou de saída, quando houver congestionamento no respectivo enlace;
2. Adicione tráfego, de entrada ou de saída, após um determinado período de tempo sem congestionamento no respectivo enlace.

As variações de tráfego dos clientes neste tipo de sistema ocorrem através da obtenção ou repasse de um maior ou menor número de descritores. No caso do tráfego de entrada (banda consumida), o cliente inscreve-se em mais ou menos árvores para receber os respectivos descritores. No caso do tráfego de saída (banda cooperada), admite mais ou menos nós abaixo de si para repassar o descritor da árvore em que foi admitido como nó interior. A variação de tráfego para um cliente não poderia ocorrer, de outra forma, através da alteração da largura de banda de um descritor pois, neste caso, todos os nós da árvore que transporta este descritor também seriam afetados.

Para a escolha de quais descritores ou de quais clientes variar, considera-se a importância relativa de cada descritor de cada camada, bem como a estrutura das árvores de distribuição, de

maneira que o impacto nos nós inferiores ao cliente cujo tráfego é modificado seja minimizado. Por exemplo, um nó superior que descarta tráfego de saída escolhe primeiramente aqueles clientes que estão recebendo descritores da camada de melhoria mais alta; dentre estes, são selecionados aqueles que possuem o menor número de nós inferiores abaixo de si, preferencialmente nenhum. Por outro lado, um cliente que descarta tráfego de entrada escolhe primeiramente aqueles nós superiores de quem está recebendo descritores da camada de melhoria mais alta; dentre estes, são selecionados aqueles cujos descritores transmitidos são repassados ao menor número de nós inferiores do referido cliente, possivelmente a nenhum.

Apesar da evolução do sistema CoopNet descrita em [44], nenhum mecanismo de incentivo à cooperação foi proposto e sistematicamente avaliado.

2.3.3 O Sistema SplitStream

O sistema SplitStream, proposto em [11], também é um sistema *peer-to-peer* de fluxo de mídia ao vivo com distribuição sincronizada. A principal diferença entre ele e o sistema CoopNet está no gerenciamento distribuído das árvores de distribuição e protocolo de admissão. Considerando somente a distribuição de conteúdo, pode-se dizer que SplitStream substitui por completo a arquitetura cliente-servidor. Entretanto, o protocolo distribuído deixa às aplicações a tarefa de disponibilizar o conteúdo a ser servido, ou seja, apenas provê a infra-estrutura necessária à distribuição. Sendo assim, considerado que os fluxos de vídeo demandam um grande espaço de armazenamento, eles tendem a se aglomerar em estações mais capacitadas em termos de recursos, formando servidores de armazenamento na rede.

Em SplitStream, utiliza-se também múltiplas árvores de distribuição e codificação em múltiplos descritores. O enfoque do trabalho é equilibrar a tarefa de distribuição de conteúdo entre os clientes da rede *peer-to-peer*. Argumenta-se que em uma árvore balanceada de grau f e altura h o número de nós interiores é $\frac{f^h-1}{f-1}$ e o número de nós folhas é f^h , o que significa que o número de nós folhas cresce com f . Por exemplo, mais da metade dos clientes são nós folhas em uma árvore binária e cerca de 90% o são em uma árvore de grau 16. No segundo caso, toda a tarefa de distribuição de conteúdo é desempenhada por menos de 10% dos clientes [11]. A utilização de múltiplas árvores de distribuição e codificação em múltiplos descritores resolve esta questão, já que cada cliente é admitido como nó interior em pelo menos uma árvore. Além disso, se o fluxo não fosse subdividido em descritores de tamanhos menores, seria impraticável um cliente admitir vários outros abaixo de si, já que a banda de saída demandada seria muito grande.

Para se obter árvores o mais disjuntas e curtas possível, visando a promoção de tolerância a falhas, cada cliente é admitido como nó interior em apenas uma delas, sendo admitido como nó folha nas $n-1$ demais. Conforme mencionado na subseção anterior, esta idéia foi originalmente proposta em [11] e posteriormente considerada em [43].

Em SplitStream, assim como em CoopNet, as múltiplas árvores de distribuição e codificação em múltiplos descritores também são utilizadas para se acomodar a heterogeneidade dos clientes, apesar de não se usar LMDC. Não obstante, apenas são tratadas heterogeneidades estáticas, ou seja, variações de bandas de entrada e de saída causadas por enlaces distintos. As variações de banda dinâmicas, causadas por eventos de congestionamento, não são consideradas. Sendo assim, o número de descritores que um determinado cliente recebe em SplitStream é uma função direta da sua banda de entrada, que é constante.

Mencionou-se em [11] a possibilidade de se ter um mecanismo de incentivo à cooperação que garantisse que as bandas cooperadas fossem iguais ou superiores às bandas consumidas. Dado que não se tem variações dinâmicas de banda (congestionamentos), a limitação da banda consumida baseada na cooperada seria um procedimento realizado apenas no momento da admissão dos clientes. De qualquer forma, não se avaliou sistematicamente esta proposta. A utilização de mecanismos de incentivo eficazes em SplitStream foi deixada como trabalho futuro em [11].

Para o gerenciamento distribuído das árvores de distribuição, SplitStream utiliza os protocolos Pastry [49] e Scribe [12]. Pastry é um algoritmo de roteamento e busca para redes *peer-to-peer* estruturadas baseado em tabelas *hash* distribuídas, assim como CAN [47], Chord [54] e Tapestry [6]. Nesses algoritmos, cada cliente recebe um identificador único ao se conectar na rede. Baseado nesses identificadores, os clientes mantêm tabelas de roteamento que relacionam seus vizinhos próximos na rede *overlay*. Quando um conteúdo é disponibilizado na rede, cria-se um identificador único para o mesmo através de uma função *hash* sobre os dados (possivelmente sobre o nome), a qual é de conhecimento de todos os clientes. O conteúdo é então encaminhado, utilizando-se as tabelas de roteamento, ao cliente cujo identificador seja mais próximo ao identificador de conteúdo, passando por clientes intermediários. Estes últimos realizam o repasse dos dados, armazenando uma cópia local dos mesmos para se ter tolerância a falhas. Sendo assim, requisições posteriores àquele conteúdo são roteadas da mesma maneira, encerrando-se quando o objeto procurado for encontrado, seja em um cliente intermediário ou naquele responsável por seu armazenamento.

O protocolo Scribe constitui um sistema de comunicação de grupos implementado na camada de aplicação e sobre o protocolo Pastry. Nele, cria-se uma chave em Pastry para cada grupo, conhecida como identificador do grupo. Uma árvore *multicast* é então formada através das rotas determinadas pelo protocolo Pastry de cada membro do grupo ao cliente cujo identificador seja mais próximo ao identificador de grupo. Este último é o responsável, na rede Pastry, por aquele identificador de grupo; conseqüentemente, ele é considerado a raiz da árvore *multicast* formada para o grupo. Sendo assim, mensagens são distribuídas da raiz do grupo a todos os seus membros, utilizando-se roteamento por caminho reverso. Devido à natureza distribuída desta abordagem, consegue-se ter um desempenho eficiente mesmo com um grande número de grupos, constituídos por diversos membros. E ainda, consegue-se tratar eficientemente constan-

tes admissões e desconexões de membros nos diferentes grupos da rede [11].

Utilizando os protocolos Pastry e Scribe, projetou-se as principais operações de gerenciamento de árvores de distribuição do sistema SpliStream. Por exemplo, para a admissão de um cliente como nó interior em apenas uma árvore, usa-se prefixos distintos nos identificadores de grupos das diferentes árvores *multicast*. Com isso, somente os clientes que tenham identificadores com o mesmo prefixo de um grupo serão nós interiores na árvore *multicast* daquele grupo, o que ocorre naturalmente através do roteamento do protocolo Pastry. Um outro exemplo, para se implementar as restrições de banda de entrada de um cliente, inscreve-se o mesmo em um número limitado de árvores. Para se restringir a banda de saída deste cliente, usa-se um mecanismo específico existente em Scribe chamado *push-down*. Adicionalmente, para maximizar as chances de construção bem sucedida de um conjunto de árvores, cria-se um grupo adicional para agrupar todas as bandas excedentes de cada cliente. Tal grupo é utilizado quando um cliente não consegue ser admitido em uma árvore de distribuição, por falta de recursos.

2.4 Considerações Finais

Discutiu-se, neste capítulo, alguns sistemas *peer-to-peer* existentes, com enfoque para a utilização dos mesmos em serviços de vídeo sob demanda. Comentou-se, inicialmente, algumas frentes de pesquisa para serviços de vídeo ainda sobre a arquitetura cliente-servidor, de forma a contrastar suas limitações com os sistemas *peer-to-peer*. Em seguida, descreveu-se as diferentes abordagens utilizadas, ao longo do tempo, para as redes *peer-to-peer* de compartilhamento de arquivos. Por fim, estudou-se a codificação em múltiplos descritores e dois sistemas *peer-to-peer* para serviços de fluxo de mídia ao vivo com distribuição sincronizada, que constituem o principal interesse do presente trabalho.

Capítulo 3

Mecanismos de Incentivo à Cooperação

Com a predominância da arquitetura cliente-servidor na maioria dos sistemas de computação até pouco tempo atrás, não havia, até então, a preocupação de se garantir determinados padrões de comportamento das entidades constituintes, tanto dos clientes quanto dos servidores. Nesta arquitetura, enquanto os servidores já são intrinsecamente disponibilizados para atender requisições, não se espera que os clientes tenham qualquer tipo de prestimosidade entre si ou para com os próprios servidores que lhes atendem. Essa assimetria, peculiar deste tipo de arquitetura, garante que cada papel seja devidamente desempenhado pelos respectivos elementos integrantes, seja para servir ou para requisitar serviços.

Não obstante, arquiteturas descentralizadas têm emergido recentemente como solução para os conhecidos problemas de escalabilidade impostos pela arquitetura cliente-servidor. Nessas arquiteturas, os papéis de servir e ser servido nem sempre são assimétricos, ou seja, um mesmo participante pode ora cooperar e ora usufruir de cooperação alheia. Além disso, muitas vezes a figura de servidores ou autoridades centrais está ausente, o que implica numa autonomia das entidades em decidir os padrões de comportamento que irão apresentar, podendo atuar como egoístas, abusivas, altruístas ou, simplesmente, justas, ou seja, cooperando somente quando possível.

As redes *peer-to-peer*, *ad hoc* e sistemas multiagentes são exemplos nos quais os integrantes apresentam esse tipo de autonomia para optar por seus padrões de comportamento. Entretanto, a viabilidade desses sistemas está fortemente baseada na premissa de comportamentos cooperativos. Além disso, o conceito de sinergia é fundamental nesses sistemas, ou seja, espera-se explorar os recursos locais dos participantes de forma que a utilização conjunta dos mesmos produza um resultado melhor do que a soma das respectivas utilizações individuais [40]. Dessa forma, deixar a decisão de cooperar ou não sob responsabilidade única dos participantes e sem nenhuma influência ou incentivo externo é colocar em risco não somente a eficiência do sistema, mas também a sua própria viabilidade.

Para agravar o problema mencionado, a opção de se assumir um comportamento coope-

rativo incorre em custos, uma vez que recursos são consumidos – como banda de rede, armazenamento, processamento, memória ou, ainda, energia. Particularmente, em dispositivos móveis, comuns nas redes sem fio, tais recursos podem ser preciosos, o que aumenta ainda mais o desinteresse das entidades em cooperar de maneira altruísta. Estudos realizados em redes *peer-to-peer* de compartilhamento de arquivos mostram que apenas 30% dos usuários compartilham seus recursos e, ainda, que a maioria das requisições são servidas por apenas 1% dos principais clientes [1]. Tais níveis de cooperação inviabilizariam a maioria dos sistemas descentralizados mencionados, mas exerce um prejuízo particular às aplicações de distribuição de conteúdo com sensibilidade temporal [28, 26, 27], como é o caso das redes *peer-to-peer* de fluxo de mídia [25, 11, 45, 43, 44].

Dessa maneira, mecanismos de incentivo à cooperação vêm recebendo, ultimamente, uma atenção especial da literatura, numa mesma época onde os sistemas descentralizados ganham espaço sobre as tradicionais arquiteturas cliente-servidor. Em geral, esses mecanismos precisam ser projetados de forma a desestimular comportamentos não-cooperativos, levando-se em consideração a natural heterogeneidade dos clientes e também determinadas circunstâncias onde a cooperação não é possível, seja por uma razão estática – por exemplo, hardware limitado – ou por uma razão dinâmica – como congestionamento de rede. Em tais circunstâncias, o mecanismo de incentivo deve isentar o cliente de punições, o que nem sempre é trivial de ser implementado, conforme será abordado na Seção 3.2.

Este capítulo apresenta um estudo sobre mecanismos de incentivo à cooperação. Dado que o número de abordagens propostas na literatura é vasto, optou-se por apresentar, com maior enfoque, os diferentes *padrões de incentivo* utilizados em comum pelos diversos *esquemas de incentivo*. Um padrão de incentivo consiste em um conjunto de mecanismos abstratos que os esquemas de incentivo podem aplicar, com o intuito de estimular cooperação [40, 41]. Um esquema de incentivo pode utilizar mais de um padrão de incentivo, implementando os seus conceitos abstratos de forma a ser eficaz às peculiaridades da aplicação que o utilizará. Espera-se, com o estudo dos padrões de incentivo, prover informações para que uma maior abrangência de esquemas de incentivo existentes possa ser compreendida, bem como para a análise de futuras propostas que ainda estão por vir.

O restante deste capítulo está organizado da seguinte forma. Na Seção 3.1, os diferentes tipos de comportamentos não-cooperativos são discutidos, incluindo uma taxonomia proposta em [38, 39]. Na Seção 3.2, os padrões de incentivo mencionados são estudados, considerando-se propriedades como tipo de remuneração, eficácia contra comportamentos não-cooperativos, anonimato e escalabilidade. Na Seção 3.3, uma seleção de trabalhos relacionados a esquemas de incentivo é apresentada, comparando-os aos padrões vistos na seção anterior. Finalmente, na Seção 3.4, considerações finais são tecidas.

3.1 Comportamentos Não-Cooperativos

A compreensão dos diferentes tipos de comportamentos não-cooperativos é necessária para a correta análise dos padrões de incentivo existentes e também para a concepção de novos esquemas de incentivo. Esta seção está organizada de acordo com a taxonomia dos comportamentos não-cooperativos proposta em [38, 39], que pode ser vista na Figura 3.1.

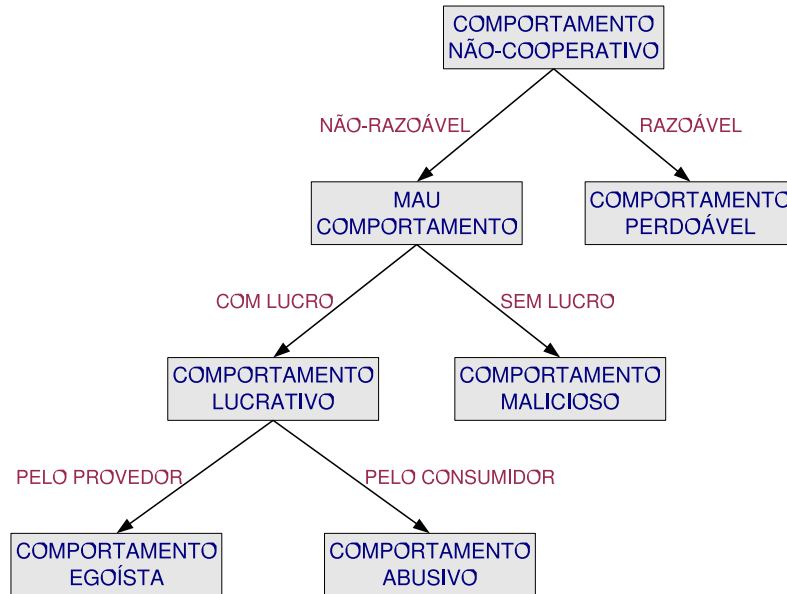


Figura 3.1: Taxonomia dos comportamentos não-cooperativos [38, 39].

De acordo com [38, 39], uma *transação* de cooperação é constituída por duas fases, a *negociação* e o *processamento*. Na primeira fase, negociação, as entidades entram em acordo acerca dos serviços prestados e respectivas remunerações. A importância dessa fase está na equalização das diferenças de recursos, padrões de acesso e assimetria de topologia entre as entidades. Por exemplo, uma entidade com sobrecarga de tráfego de rede ou com enlace limitado poderia apresentar um custo de roteamento mais elevado que as demais. Na segunda fase da transação de cooperação, processamento, ocorre a cooperação e a respectiva remuneração. Neste contexto, costuma-se denominar a entidade que coopera como *provedora* e a entidade que usufrui dos serviços da primeira como *consumidora*. Tais termos, bem como as operações da fase de processamento das transações de cooperação estão ilustrados na Figura 3.2 [40].

A cooperação é benéfica à entidade consumidora, que retribui o favor recebido à entidade provedora através de uma remuneração, de forma a incentivá-la a continuar tendo este comportamento cooperativo. Na ausência dessa remuneração, ou seja, sem um incentivo, a entidade provedora poderia se recusar a cooperar, o que configuraria um *comportamento egoísta*. Este

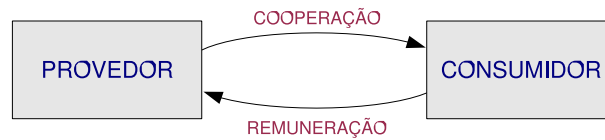


Figura 3.2: Terminologia de uma transação de cooperação [40].

tipo de *comportamento não-cooperativo* pode ser classificado como um *comportamento lucrativo*, uma vez que a entidade provedora se nega a cooperar com a consumidora não para prejudicá-la, mas sim para poupar seus recursos próprios *com lucro* – já que nada obteria em troca como remuneração. Além disso, este comportamento egoísta também pode ser classificado como sendo um *mau comportamento*, ou seja, caracteriza-se como *não-razoável*, dado que a entidade provedora tinha condições – recursos abundantes, por exemplo – de cooperar, ao contrário de uma entidade numa situação de contenção onde a economia de seus recursos fosse vital – como uma bateria de alimentação de um dispositivo móvel ao fim da carga, por exemplo. Como exemplos de comportamentos egoístas, pode-se citar um usuário que se nega a disponibilizar seus arquivos aos demais, numa rede *peer-to-peer* de compartilhamento de arquivos; um telespectador que não repassa fluxos de vídeo, numa rede *peer-to-peer* de fluxo de mídia; e um agente se recusando a repassar pacotes numa rede *ad hoc*.

Quando a entidade consumidora remunera a provedora de maneira insuficiente, ou ainda se aproveita de uma mesma remuneração efetuada para consumir mais recursos do que o acordado na fase de negociação da transação de cooperação, tem-se um *comportamento abusivo*. Assim como o comportamento egoísta, este pode ser classificado como lucrativo, já que a entidade consumidora não tem o objetivo de prejudicar a provedora simplesmente, mas sim obter vantagem sobre ela. O que diferencia um comportamento lucrativo de ser egoísta ou abusivo é o fato dele ser praticado *pelo provedor* ou *pelo consumidor*, respectivamente. O comportamento abusivo também pode ser classificado como um mau comportamento, caracterizando-se como não-razoável, por motivos análogos aos apresentados para o comportamento egoísta. Como exemplos de comportamentos abusivos, pode-se citar usuários que sempre efetuam *downloads* dos mesmos arquivos ao invés de armazená-los localmente, numa rede *peer-to-peer* de compartilhamento de arquivos; telespectadores que solicitam uma qualidade de vídeo superior à que seus periféricos possam reproduzir, numa rede *peer-to-peer* de fluxo de mídia; e um agente que envia pacotes desnecessários numa rede *ad hoc*, como anúncios excessivos dos seus serviços oferecidos.

Um mau comportamento pode ser *sem lucro*, ou seja, a entidade que o apresenta pode ter como objetivo somente prejudicar as demais, sem a obtenção de benefício próprio. Neste caso, tem-se um *comportamento malicioso*. De acordo com [38, 39], que categorizam os diferentes

tipos de comportamentos não-cooperativos em domínios ou camadas distintas – enlace, rede, transporte, descoberta e aplicação –, o comportamento malicioso somente se apresenta quando, apesar de não-lucrativo na camada em questão, faz-se lucrativo em camadas superiores. Por exemplo, em um esquema de incentivo baseado em reputação, a difusão de informações de difamação consumiria recursos da camada de rede, não sendo, portanto, um comportamento lucrativo neste domínio de cooperação. Não obstante, no domínio da camada de aplicação, este comportamento seria considerado lucrativo caso uma outra entidade provedora de serviços fosse excluída da rede em virtude de sua má reputação.

Considerando todos os domínios de cooperação juntos, o comportamento malicioso somente seria apresentado por entidades mal intencionadas, ou seja, aquelas com o objetivo não de tirar vantagens sobre as outras, baseadas em regras lógicas de maximização de seus benefícios, mas sim de degradar o desempenho do sistema como um todo ou o de algumas entidades específicas. Neste caso, este comportamento malicioso não cabe ser tratado pelos esquemas de incentivo, mas sim por mecanismos adicionais que detectem e punam as entidades responsáveis de forma eficaz. Como exemplo de comportamento malicioso, pode-se citar os ataques de negação de serviço (do Inglês, *denial of service*; tipo de ataque onde uma entidade torna outra indisponível na rede bombardeando-a com múltiplas requisições de serviço) nos diferentes tipos de rede existentes.

Finalmente, um comportamento não-cooperativo pode ser classificado como *razoável*, quando a entidade que o pratica de fato não apresenta condições de cooperar, devido à escassez de seus recursos. Tem-se, neste caso, um *comportamento perdoável*. A escassez de recursos pode ser permanente, quando a entidade apresenta limitações de recursos tais como poder computacional, memória, largura de banda ou energia; ou transiente, quando, por exemplo, a entidade experimenta situações de sobrecarga ou problemas de conectividade. Em ambos os casos, um mecanismo de incentivo ideal deveria poupar a entidade da obrigatoriedade de cooperação, ou seja, permitir que a mesma usufrua de cooperações alheias sem nada exigir em troca. Naturalmente, faz-se importante a diferenciação entre os comportamentos perdoáveis e os demais maus comportamentos, o que é um problema para os mecanismos de incentivo [38, 39]. Pode-se citar, como exemplos de comportamentos perdoáveis, uma entidade que se nega a cooperar porque se tornou o gargalo de roteamento numa rede *ad hoc*, ou um usuário que precisa limitar o número de conexões atendidas dado que disponibilizou uma versão de um arquivo popular numa rede *peer-to-peer*, tornando-se rapidamente provedor de muitas transferências – o que esgota a sua largura de banda.

Resta ainda mencionar que a utilização de um determinado mecanismo de incentivo pode induzir novos comportamentos não-cooperativos, além daqueles que ele combate. Por exemplo, em um cenário de remuneração por reputação, uma entidade poderia efetuar difamações de outras entidades na rede de forma a aumentar a sua reputação relativa, ou ainda conseguir excluir algumas entidades do conjunto de provedores que com ela competem. Ainda no mesmo cenário

de reputação, um conjunto de entidades poderia divulgar falsos elogios mútuos de forma que suas reputações aumentassem no grupo, de maneira desonesta com as demais. Tais comportamentos não-cooperativos não existiriam se o referido mecanismo de reputação não estivesse sendo utilizado.

Em um cenário de remuneração por pagamento, alguns comportamentos não-cooperativos também podem ser induzidos, principalmente quando a remuneração é relativa à escassez dos recursos da entidade provedora. Este tipo de remuneração é denominada flexível [38, 39] e é tratada na fase de negociação das transações de cooperação, conforme mencionado anteriormente. Neste cenário, uma entidade consumidora poderia impedir, de forma maliciosa, que uma determinada entidade provedora cooperasse com as demais, o que não consumiria seus recursos e, conseqüentemente, abaxaria o valor dos mesmos por se tornarem abundantes. Por outro lado, uma entidade provedora poderia conseguir elevar a sua remuneração tornando seus competidores indisponíveis como, por exemplo, através dos ataques de negação de serviço [38, 39]. Neste caso, tais ataques são considerados lucrativos ao invés de maliciosos, já que trazem benefícios às entidades que o praticam. Novamente, estes comportamentos não-cooperativos induzidos não existiriam se o mecanismo de incentivo não estivesse sendo aplicado.

3.2 Padrões de Incentivo

O estudo dos diferentes padrões de incentivo é importante para que uma maior abrangência de esquemas de incentivo possa ser compreendida, bem como para a análise de futuras propostas. Esta seção está organizada de acordo com a taxonomia dos padrões de incentivo proposta em [40, 41], que pode ser vista na Figura 3.3. Primeiramente, são discutidos os padrões baseados em confiança e, posteriormente, os baseados em comércio.

3.2.1 Padrões Baseados em Confiança

Nos padrões baseados em confiança, as entidades provedoras cooperam com as consumidoras porque nelas confiam, podendo esta confiança ser estática – não varia em função do tempo e em decorrência de seus comportamentos – ou dinâmica – caso contrário. Dessa forma, não ocorre uma remuneração explícita do consumidor ao provedor. Ao contrário, a retribuição da cooperação ocorre naturalmente porque: (1) as entidades constituem um grupo com os mesmos ideais ou (2) porque a reputação do provedor foi elevada e, como conseqüência, outras entidades com ele cooperarão posteriormente. Estes dois casos são apresentados a seguir.

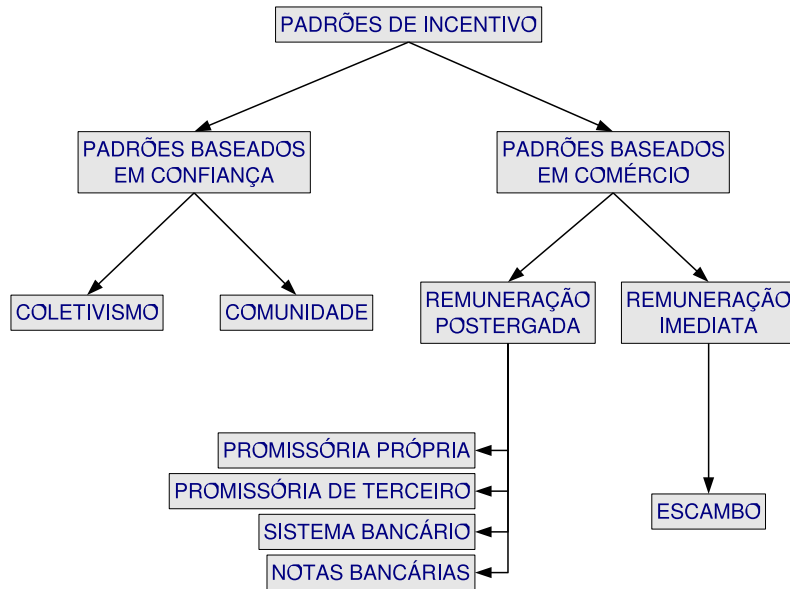


Figura 3.3: Taxonomia dos padrões de incentivo [40, 41].

Coletivismo

No padrão de incentivo coletivismo (do Inglês, *the collective pattern*), as entidades ajudam-se mutuamente porque compartilham os mesmos ideais, que são comuns ao grupo ou coletividade, não havendo interesses individuais. Dessa forma, quando o grupo ganha, todos ganham; quando o grupo perde, todos perdem. Decorre deste modelo que nenhuma entidade fica tentada a não cooperar com o intuito de tirar vantagem sobre as demais, uma vez que o grupo todo seria prejudicado – inclusive ela própria – e, além disso, nenhum ganho em termos de economia de recursos seria obtido individualmente.

Fazendo uma analogia, pode-se comparar este modelo a uma organização corporativa. Considerando que em tal cenário todos os lucros e prejuízos são igualmente divididos aos seus sócios, nenhum deles fica motivado a não colaborar com a produtividade do outro, dado que os interesses são comuns. Uma outra comparação, no campo da computação, é uma rede *ad hoc* de sensores. Neste ambiente, não faz sentido, por exemplo, um sensor cooperar menos que outro, dado que o objetivo de todos é o mesmo. A abstração de uma coletividade vista como uma única entidade possibilita que o padrão coletivismo seja combinado com os demais, apresentados em seguida. Pode-se, por exemplo, considerar cada entidade integrante dos demais modelos como sendo um conjunto de máquinas, ao invés de uma única.

Este padrão de incentivo pertence ao escopo dos padrões baseados em confiança, já que as entidades cooperam porque confiam-se mutuamente e incondicionalmente – por pertencerem à

mesma coletividade. Trata-se de uma confiança estática, ou seja, a reputação dos integrantes não varia em função do tempo e em decorrência de seus comportamentos. Por este motivo, é importante garantir que cada entidade membra da coletividade seja de fato uma integrante autêntica. Caso contrário, poder-se-ia ter, por exemplo, uma entidade de outra coletividade usufruindo dos recursos do grupo sem oferecer nada em troca – pois seus interesses e responsabilidades estariam focados em outro grupo.

Enquanto os demais padrões de incentivo apresentam sempre algum tipo de remuneração à entidade que cooperou – como reputação, moeda ou serviço em retorno –, o padrão de incentivo coletivismo é o único que dispensa este procedimento. Essa característica decorre da natureza coletiva do grupo e da ausência de interesses individuais. Como será visto posteriormente, nos demais padrões, essa remuneração, quando não consumida imediatamente, pode ficar armazenada em diferentes lugares, como no provedor, consumidor ou ainda em entidades terceiras.

Os três principais tipos de comportamentos não-cooperativos – egoísta, abusivo e perdoável – são naturalmente tratados neste padrão de incentivo, por suas características intrínsecas. Os comportamentos egoísta e abusivo não ocorrem devido à ausência de interesses individuais, enquanto que o comportamento perdoável é tolerado devido à confiança incondicional existente entre os integrantes do grupo. Este padrão de incentivo é o que melhor trata os comportamentos perdoáveis, isentando-os de punições.

Uma desvantagem deste padrão – e dos demais baseados em confiança – é a impossibilidade de manter o anonimato das entidades, uma vez que as mesmas precisam se identificar para provarem que são confiáveis. Dado que as entidades possuem confiabilidades estáticas neste modelo, são os consumidores que precisam abrir mão do anonimato, identificando-se como membro do grupo (as entidades provedoras não precisam se identificar uma vez que não têm suas reputações alteradas após suas cooperações). Dependendo da natureza da aplicação em questão, a característica de anonimato pode ser particularmente desejável. Como exemplo, pode-se citar um serviço de vídeo sob demanda cuja programação seja caracterizada como “adulta” onde, conseqüentemente, os telespectadores preferam não ser identificados.

Uma outra desvantagem deste padrão é a escalabilidade restrita. Neste contexto, entende-se escalabilidade como sendo a capacidade do padrão de incentivo de se manter eficiente e eficaz conforme o número de entidades participantes aumenta. Em geral, os padrões baseados em confiança não são muito escaláveis [40, 41]. Em específico, para este caso, uma coletividade relativamente grande poderá apresentar desafios para garantir que suas entidades sejam autênticas (não sejam exclusivamente de outros grupos), já que fica impraticável todas elas se conhecerem mutuamente em um grupo numeroso.

Comunidade

No padrão de incentivo comunidade (do Inglês, *the community pattern*), diferentemente do padrão anterior, as entidades cooperam umas com as outras por interesses individuais, não havendo o conceito de coletivismo. Ao cooperar, uma entidade aumenta a sua reputação naquela comunidade, sendo que esta reputação será necessária para que ela consiga usufruir de cooperações alheias posteriormente. Neste contexto, uma determinada entidade poderia se negar a cooperar com outra cuja reputação estivesse baixa ou, ainda, mais radicalmente, se a reputação desta fosse menor que a sua. Em outras palavras, a motivação para as entidades almejarem o aumento de suas reputações e, conseqüentemente, cooperarem mais, é estarem habilitadas a usufruírem das cooperações do maior número possível de integrantes do grupo.

Pode-se comparar este modelo de incentivo a um conjunto de estudantes de uma mesma turma. Aqueles mais prestativos sempre conseguirão auxílio dos companheiros quando precisarem, pois já o fizeram em ocasiões anteriores, passando a ter uma boa reputação no grupo. Já aqueles mais egocêntricos, que se negaram a prestar ajuda no passado, certamente terão dificuldades para conseguir o mesmo no futuro. Diversos sistemas de computação também utilizam este modelo, podendo algumas redes *peer-to-peer* de compartilhamento de arquivos serem citadas, como é o caso do Kazaa [31]. Nessas redes, geralmente, um usuário pode efetuar diversas transferências de arquivos de outros com reputações menores que a sua. Além disso, ao receber múltiplas requisições, um determinado usuário pode priorizar o atendimento daqueles com reputações maiores.

Assim como o coletivismo, este padrão também pertence ao escopo dos padrões baseados em confiança, dado que as entidades cooperam com aquelas que apresentam maiores confiabilidades, ou seja, boas reputações. Ao contrário do coletivismo, estas confiabilidades são dinâmicas, variando em função do tempo e em decorrência dos comportamentos apresentados. Conseqüentemente, faz-se necessário garantir que as entidades não burlem reputações de forma que não condigam com a realidade, como difamação de bons cooperadores ou elogio falso daqueles que não cooperam. Essa questão é particularmente delicada quando se combina este padrão com o coletivismo. Dado que cada entidade da comunidade é uma abstração de coletividade onde se pratica o coletivismo, integrantes desta última poderiam cooperar somente entre si, o que já é previsto no padrão coletivismo, elevando suas reputações na comunidade e, conseqüentemente, usufruindo de recursos externos ao grupo, sem nada prover em troca. Uma das formas de se contornar este problema seria atribuir reputações aos grupos ao invés de a cada integrante individual das coletividades [40, 41].

Sendo a remuneração deste padrão de incentivo a reputação, é interessante notar onde a mesma fica armazenada. Voltando à analogia dos estudantes de uma mesma turma, não faria sentido algum deles afirmar aos demais que a sua reputação fosse boa, ou seja, que ele próprio fosse uma pessoa prestativa. Dessa forma, neste modelo são os consumidores que precisam lembrar de quais provedores usufruíram cooperações e, ainda, o quanto cada um deles cooperou, de

forma a viabilizar uma quantificação da reputação individual de cada um. Normalmente estas reputações não são absolutas, mas sim relativas ao restante do grupo. Para que isso ocorra, o conhecimento global de todas as reputações é necessário, o que pode ser obtido pela propagação de tais informações na rede ou pelo armazenamento das mesmas em entidades centrais. Naturalmente esta última abordagem apresentaria desvantagens, por ser um ponto único de falha no sistema.

Os comportamentos não-cooperativos egoísta e abusivo são combatidos por este padrão de incentivo desde que as informações de reputações estejam disponíveis de forma comum a todas as entidades do grupo. Caso contrário, uma determinada entidade poderia se negar a cooperar com um determinado subconjunto da comunidade, cooperando apenas com aqueles nos quais tivesse interesses futuros. O comportamento abusivo, em particular, poderia ser combatido de maneira eficaz com o provedor diminuindo a reputação do consumidor caso este o apresentasse. Neste caso, as reputações seriam armazenadas ou modificadas pelos provedores também, além dos consumidores. Já o comportamento perdoável não é tão bem tratado como no coletivismo, onde a confiança era incondicional e estática. De qualquer maneira, mediante a dificuldade de se diferenciar uma não-cooperação como sendo comportamento egoísta ou perdoável, pode-se levar em conta o histórico da entidade, ou seja, aquelas que já possuem uma certa confiabilidade poderiam se negar a cooperar numa situação de sobrecarga, sem que isso tivesse um impacto negativo em sua reputação.

Conforme já mencionado, a impossibilidade de se manter o anonimato das entidades é uma desvantagem dos padrões baseados em confiança. No caso deste modelo, como a confiabilidade das entidades é dinâmica, tanto os consumidores quanto os provedores precisam se identificar. Os consumidores para provarem que são merecedores da cooperação que almejam, através de suas reputações, e os provedores para que possam ter suas reputações aumentadas, após suas colaborações se efetivarem.

Dado que este é um padrão baseado em confiança, também apresenta escalabilidade restrita. Em particular, a existência de muitas entidades no grupo pode dificultar a propagação das informações de reputação que, conforme visto, auxilia no combate dos maus comportamentos. Ainda que essas informações de reputação não fossem propagadas, a escalabilidade também poderia ficar comprometida, uma vez que dificilmente o mesmo par de entidades cooperaria entre si múltiplas vezes em um grupo numeroso, o que torna as informações locais inúteis. A utilização de entidades centrais para o armazenamento destas reputações, sugerida anteriormente, pode ainda impor gargalos de desempenho ao sistema, prejudicando ainda mais a escalabilidade em questão.

3.2.2 Padrões Baseados em Comércio

Nos padrões baseados em comércio, as entidades provedoras obtém uma remuneração explícita das consumidoras por suas cooperações. Esta remuneração consiste em uma retribuição de cooperação, que pode ocorrer imediatamente – enquanto a transação de cooperação ocorre – ou posteriormente – através de uma promessa, que será honrada pela própria entidade consumidora ou por uma entidade terceira em favor dela. Tais possibilidades são discutidas a seguir.

Escambo

No padrão de incentivo escambo (do Inglês, *the barter trade pattern*), as entidades cooperam de forma a realizarem uma troca direta de favores, troca esta que ocorre simultaneamente, ou seja, enquanto uma entidade está cooperando, está também usufruindo, ao mesmo tempo, da cooperação da entidade beneficiada por sua cooperação. Decorre que nenhuma das partes deixa obrigações pendentes após o término da transação de cooperação e, também, aquelas entidades que se negam a cooperar nada obtém das demais. Este é o único padrão de incentivo no qual os integrantes assumem papéis simétricos nas interações, atuando o consumidor como provedor e vice-versa, concomitantemente.

Um exemplo para se ilustrar este modelo são as feiras de escambo, popularmente conhecidas como “feiras do rolo”. Nestas feiras, as pessoas trocam objetos permanentemente, sem utilização de moeda, de modo que precisam dispor de algo que lhes pertence para a obtenção de um novo objeto almejado. Além disso, após uma determinada troca, nenhuma das partes deve nada à outra, pois já retribuiu aquilo que recebeu, naquele mesmo momento. Tipicamente comuns, nos dias atuais, são as feiras de trocas de livros, onde as pessoas realizam o escambo de livros já lidos por aqueles que gostariam de ler. No campo da computação, alguns sistemas podem ser citados, como é o caso do BitTorrent [15, 7], cujo mecanismo de incentivo será melhor descrito posteriormente. Nesta rede, quanto maior a banda de envio disponibilizada pelo cliente, maior é a taxa de recepção obtida, ou seja, quanto mais se coopera, mais se ganha.

Este padrão de incentivo pertence ao escopo dos padrões baseados em comércio, uma vez que as entidades cooperam para receber uma remuneração explícita em troca, ao contrário dos anteriores, nos quais se coopera por confiança. Este é o único padrão de incentivo que apresenta naturalmente a característica de persistência, ou seja, a remuneração acontece mesmo quando as entidades que participam de uma cooperação desconectam-se logo em seguida [40, 41]. Para que os demais padrões apresentem esta característica, é necessário a introdução de uma terceira entidade, por exemplo, um banco, que seja confiável e esteja sempre disponível a honrar as promessas feitas pelas entidades que a ele estejam relacionadas. Em contrapartida, a introdução desta terceira entidade pode violar uma outra característica que também é intrínseca ao padrão de incentivo escambo: a característica de localização, que é a ausência da necessidade de interação com entidades terceiras em operações de cooperação e remuneração [40, 41].

Dado que a remuneração das cooperações neste modelo de incentivo ocorre imediatamente e consiste de outra cooperação em troca, ela não precisa ser armazenada, como acontece nos demais padrões, exceto no coletivismo. Conseqüentemente, dificuldades como propagação de informações de reputação na rede, cálculos de reputações relativas, introdução de entidades centrais ou armazenamento distribuído de reputações não são necessárias. Este fato torna o padrão de incentivo escambo relativamente simples de ser implementado, ainda que muito poderoso.

Se por um lado os comportamentos não-cooperativos egoísta e abusivo são muito bem tratados neste padrão de incentivo, considerando a exigência de remuneração imediata, o mesmo não acontece com o comportamento perdoável. Devido à natureza simétrica do protocolo de cooperação, aquelas entidades que não cooperam, mesmo que justificadamente, não podem usufruir de nenhum benefício. Entretanto, esta limitação pode ser contornada combinando-se este padrão com o coletivismo, quando apropriado. Neste caso, para cada coletividade onde se tenha o coletivismo, a cooperação de algumas entidades integrantes com o meio externo (através do escambo) poderia compensar a falta de cooperação justificada de outras entidades da mesma coletividade.

Neste modelo de incentivo, ambas as entidades participantes de uma cooperação podem permanecer anônimas. Conforme explicado em [40, 41], confiança e, conseqüentemente, quebra de anonimato somente são necessárias quando a remuneração não for realizada simultaneamente. O termo confiança, neste contexto, não está restrito aos padrões de incentivo baseados em confiança. Por exemplo, se uma entidade consumidora remunerou uma provedora com uma nota promissória, cheque ou notas bancárias, padrões de incentivo a serem vistos mais adiante, a entidade credora deverá confiar que a devedora irá honrar a promessa no futuro, terá fundos em sua conta bancária ou ainda que as notas utilizadas sejam autênticas, respectivamente. O padrão escambo evita tais riscos com o uso de remuneração simultânea, bem como com a não utilização de promessas, ou seja, garantindo que o pagamento seja também uma cooperação em troca.

Uma outra grande vantagem deste padrão de incentivo é a sua escalabilidade, conseqüência direta da presença de acoplamento temporal entre cooperação inicial e respectiva remuneração. A desnecessidade de armazenamento de remuneração, a característica de localização e a desobligatoriedade de confiança mútua entre as entidades tornam este um dos padrões de incentivo mais escaláveis dentre os existentes. E ainda, considerando que uma entidade somente irá atuar como provedora se ela também estiver interessada na cooperação da entidade consumidora, quanto maior o grupo, maiores são as chances de se encontrar pares cujos interesses se casem.

Dentre as desvantagens deste padrão, pode-se citar a dificuldade de garantia de atomicidade das transações. Para que isto ocorra, é necessário que as partes permaneçam acessíveis durante toda a transação. Além disso, mecanismos adicionais podem ser precisos para se assegurar a remuneração em troca, mesmo ela sendo síncrona, evitando o comportamento abusivo [40, 41]. Finalmente, em muitos casos, a utilização deste padrão de incentivo não é viável, ou seja, a

utilização de remunerações postergadas é mais apropriada.

Promissória Própria e Promissória de Terceiro

Nos padrões de incentivo promissória própria e promissória de terceiro (do Inglês, *the bearer notes pattern* e *the bearer bills pattern*, respectivamente), as entidades também cooperam para obter uma remuneração explícita em troca. A diferença entre eles e o padrão escambo está no desacoplamento temporal entre cooperação e remuneração, o que permite que a entidade provedora usufrua de seu pagamento num momento posterior àquele em que cooperou. Neste contexto, uma nota promissória pode ser entendida como uma promessa de cooperação em favor de seu portador, a ser honrada sob demanda uma vez no futuro [40, 41]. No caso do padrão promissória própria, a entidade consumidora que emitiu a nota é quem deverá honrá-la posteriormente, ou seja, os papéis de emissor e devedor daquela nota são assumidos pela mesma entidade. No caso do padrão promissória de terceiro, uma outra entidade, sob ordem da entidade emissora, é quem deverá honrar a nota no futuro, ou seja, os papéis de emissor e devedor são assumidos por entidades diferentes. Em ambos os casos, a motivação para as entidades cooperarem atuando como provedoras é a obtenção das promissórias, que as tornam credoras no grupo e, portanto, capazes de usufruir de cooperações posteriormente, no momento em que desejarem.

Estes modelos de incentivo poderiam ser comparados a uma sociedade hipotética na qual não se utilizasse dinheiro como remuneração. Ao invés disso, toda vez que uma pessoa prestasse seus serviços a outrem, esta última daria a ela uma espécie de “vale-serviço” em retribuição, na sua especialidade. Por exemplo, ao consertar as instalações elétricas da residência de um médico, um eletricista poderia ganhar uma consulta médica para tratar de sua saúde no futuro. Isto nada mais seria que uma promissória própria, emitida e honrada pela mesma entidade – o médico – que usufruiu de uma determinada cooperação – o serviço do eletricista. De maneira similar, o referido médico poderia ainda ser casado com uma dentista, remunerando agora o eletricista com um tratamento odontológico, a ser realizado por sua cônjuge. Neste caso, ter-se-ia uma nota promissória de terceiro, emitida por uma entidade – o médico – e honrada por outra – a dentista –, para a remuneração da cooperação recebida.

No campo da computação, os modelos de incentivo promissória própria e promissória de terceiro também podem ser reconhecidos em alguns sistemas, como nas antigas BBS e nos atuais fóruns de IRC. Nestes sistemas, o servidor, ao receber um *upload* de um usuário, remunera-o liberando um crédito para *download*, que pode ser realizado em qualquer momento no futuro. Enquanto tal procedimento ilustra uma promissória própria (o servidor emite e honra a promessa), uma promissória de terceiro pode ser exemplificada nos mesmos sistemas com uma transferência de créditos da cota de um usuário a outro, operação normalmente permitida. Assim, o usuário solicita um *download* ao servidor, que coopera e, conseqüentemente, obtém sua remuneração através da dedução da cota daquele usuário. Ao invés deste último efetuar

um *upload* posteriormente, para honrar sua promessa de retribuição ao servidor, ele pede a um amigo, um outro usuário do sistema, para fazer tal contribuição e, em seguida, transferir o crédito obtido à sua conta, tirando-a do negativo. Tem-se, portanto, uma entidade como emissora da promissória (o usuário cuja cota de *download* foi deduzida), outra entidade como devedora (o usuário que efetivamente honra a promessa, efetuando o *upload*) e, finalmente, uma terceira entidade como credora (o servidor, que efetuou a cooperação inicial).

Da mesma maneira que o padrão de incentivo escambo, os padrões promissória própria e promissória de terceiro também pertencem ao escopo dos padrões baseados em comércio, pois as entidades recebem uma remuneração explícita por suas cooperações. Pode-se subdividir esta categoria em duas: de um lado, remuneração imediata, onde se posiciona o padrão escambo; de outro, remuneração postergada, onde se posicionam os padrões promissória própria, promissória de terceiro e os demais a serem ainda apresentados. Nesta segunda subcategoria, dado que a retribuição não ocorre de imediato e o que a garante no futuro é uma promessa, a qual, neste contexto, está sob a forma de promissórias, faz-se importante garantir a autenticidade destas promessas, de forma que entidades mal intencionadas não as falsifiquem para tirarem vantagem sobre as demais.

As promessas na forma de notas promissórias são os tipos de remuneração dos padrões de incentivo promissória própria e promissória de terceiro. Ao contrário do padrão comunidade, no qual a remuneração – na forma de reputação – é normalmente armazenada nos consumidores, nestes padrões a mesma é armazenada nos provedores, que podem ser vistos como portadores das promissórias. É interessante notar um novo conceito introduzido pelos presentes modelos de incentivo, o de transferência de remuneração entre as entidades, ou seja, após o recebimento de sua remuneração, uma entidade provedora poderia, eventualmente, repassá-la a outras entidades que viessem a cooperar com ela, evitando, assim, a emissão de uma nova promissória.

O comportamento não-cooperativo egoísta é combatido de maneira eficaz tanto no padrão promissória própria quanto no padrão promissória de terceiro, uma vez que as entidades não deixam de cooperar visando a obtenção das promissórias. Já o comportamento abusivo, enquanto ele é incondicionalmente evitado no padrão promissória própria, dado que a entidade que se torna devedora da promissória é a mesma que a emite, no padrão promissória de terceiro é necessário que a entidade que se torna devedora seja confiável para conter o abuso da emissora. Se esta premissa não fosse verdadeira, uma entidade poderia se sentir livre para emitir promissórias de terceiro às custas de outras entidades quaisquer, repassando a elas toda a sua responsabilidade de retribuição, o que provavelmente nunca seria honrado. Quanto ao comportamento perdoável, o padrão promissória própria não lida muito bem com ele, dado que a entidade que emitiu a promissória terá, ela mesma, que se tornar a devedora daquela promessa, estando ou não em condições de fazê-lo. Este padrão de incentivo, assim como o escambo, por motivos análogos, são os que pior lidam com o comportamento perdoável. Já o padrão promissória de terceiro, por outro lado, pode tratar melhor esta questão. Estando uma entidade

necessitada de cooperação mas impossibilitada de se tornar devedora de uma promessa – como numa situação de sobrecarga, por exemplo –, ela poderia convencer uma outra entidade a se tornar devedora por ela, emitindo, dessa forma, uma promissória de terceiro.

Com relação ao anonimato das entidades nos padrões promissória própria e promissória de terceiro, em geral tanto as emissoras quanto as devedoras das promissórias precisam se identificar para comprovarem suas confiabilidades [40, 41]. Sendo ambos os papéis assumidos pela mesma entidade no padrão promissória própria, a desvantagem da quebra de anonimato iguala-se ao que ocorre nos padrões baseados em confiança, ou seja, não há meios do consumidor permanecer anônimo. Já no caso do padrão promissória de terceiro, sendo o devedor da promissória suficientemente confiável, haveria uma possibilidade do emissor não se identificar, desde que fossem excluídas as chances de emissão de promissórias de terceiro falsas, ou seja, aquelas em que o devedor não autoriza a sua participação na transação do emissor. Neste contexto de confiança baseada em identidade, é interessante notar como os padrões promissória própria e comunidade levam o consumidor a querer trocar ou renovar a sua identidade – para se livrar das obrigações assumidas –, enquanto que, por outro lado, levam o provedor querer mantê-la – para receber as retribuições de suas cooperações já efetuadas.

Em decorrência da necessidade de confiança das entidades nos presentes padrões, tem-se que os mesmos apresentam restrições de escalabilidade, por motivos já explicados anteriormente. No entanto, supondo que estes padrões estivessem sendo aplicados em um ambiente isento de desonestidade, os problemas relacionados à comprovação de confiabilidades inexisteriam, tornando a escalabilidade destes padrões menos restrita que a dos padrões baseados em confiança. Nestes últimos, conforme já visto, tais informações são indispensáveis de qualquer maneira, pois determinam os níveis de cooperação entre as entidades.

Sistema Bancário

No padrão de incentivo sistema bancário (do Inglês, *the banking pattern*), assim como nos padrões promissória própria e promissória de terceiro, as entidades cooperam para obter uma remuneração explícita em troca, remuneração esta que será usufruída num momento posterior à cooperação. A diferença está no tipo de remuneração que, ao invés de ser uma promissória, é um cheque emitido. Pode-se, no entanto, entender um cheque, neste cenário, como sendo uma promissória de terceiro cujo devedor seja um banco [40, 41], ou seja, um cheque é um tipo especial de nota promissória de terceiro, a ser emitido pelas entidades consumidoras como remuneração às entidades provedoras após as respectivas cooperações. Como consequência, decorre que cada entidade, neste padrão de incentivo, possui uma conta bancária num banco associado. Sendo assim, o portador de um cheque apresenta-o ao seu banco para que sua conta seja creditada. Caso o referido banco seja diferente do banco do emissor do cheque, uma transação entre bancos se faz necessária. Em suma, a motivação para as entidades cooperarem é a obtenção de cheques como pagamentos que serão creditados em suas contas, habilitando-as

a pagar por cooperações futuras das quais venham a necessitar.

Pode-se comparar este modelo de incentivo à utilização de cheques no sistema bancário da sociedade atual. Cada pessoa possui uma ou mais contas bancárias, associadas a bancos específicos, e utiliza cheques para a remuneração de serviços que tenham usufruído. O portador de um cheque apresenta-o ao seu banco para que a sua conta seja creditada, ficando a cargo das diferentes instituições bancárias as devidas compensações monetárias entre elas. As pessoas, então, prestam seus serviços às outras para aumentarem suas contas e poderem pagar pelo que desejam, posteriormente.

No campo da computação, pode-se citar sistemas de redes *ad hoc* que utilizam mecanismos de incentivo baseados neste padrão, como é o caso de TerMiNodes e Sprite [38, 39]. Enquanto em TerMiNodes [9, 10] cada entidade possui um módulo de segurança que gerencia a sua própria conta bancária, sendo, portanto, este modelo de incentivo implementado de forma totalmente distribuída, em Sprite [64] há nós dedicados que atuam como bancos, centralizando o gerenciamento das contas que pertencem ao mesmo grupo e empregando transações inter-bancárias quando necessário.

Este padrão de incentivo, assim como os padrões promissória própria e promissória de terceiro, pertence ao escopo dos padrões baseados em comércio e também à categoria de remuneração postergada, onde há desacoplamento temporal entre cooperação e remuneração. Dessa forma, dado que o que garante a retribuição das cooperações no futuro são os cheques emitidos, é importante garantir a autenticidade dos mesmos, para que não haja falsificações. No entanto, sendo os bancos entidades bem conhecidas e estes, por suas vezes, conhecendo a identidade de seus clientes, tem-se que o risco de falsificação de cheques é menor que o de falsificação de promissórias.

O tipo de remuneração do presente padrão, já mencionado, é um cheque emitido. De forma similar aos demais padrões com remuneração postergada, os cheques também são armazenados nos provedores, até que sejam apresentados aos bancos. É importante notar, no caso deste padrão, a distinção entre o armazenamento dos cheques (promessas) e o armazenamento das contas bancárias, que cada entidade possui. Estas últimas, geralmente, são mantidas em bancos. Assim como no caso das promissórias, os cheques também podem ser transferíveis.

O comportamento não-cooperativo egoísta é combatido apropriadamente no padrão sistema bancário, dado que as entidades não deixam de cooperar com o objetivo de obterem cheques que elevam suas contas bancárias. O comportamento abusivo é também bem tratado: já que os bancos são confiáveis, sempre honram suas promessas e conhecem a identidade de seus correntistas, o que faz com que os abusos de entidades emissoras recaiam sobre elas mesmas, garantindo a não ocorrência deste tipo de mau comportamento. Quanto ao comportamento perdoável, este pode ser tratado de forma similar como o é no padrão promissória de terceiro: se uma entidade estiver precisando de uma cooperação, mas impossibilitada de cooperar em troca para “cobrir” a sua conta bancária, ela pode convencer uma outra entidade a fazer isso por

ela, posteriormente depositando o cheque resultante em sua conta.

De maneira análoga ao que ocorre nos padrões promissória própria e promissória de terceiro, tanto as entidades emissoras quanto as devedoras dos cheques precisam se identificar para comprovarem suas confiabilidades [40, 41]. Sendo, no entanto, as devedoras dos cheques os bancos, que não se importam em revelar suas identidades, tem-se que a questão do anonimato destas, pelo menos, não é mais um problema neste padrão de incentivo. Considerando, novamente, que os bancos são confiáveis e conhecem a identidade de seus clientes, há também uma possibilidade das entidades emissoras não precisarem se identificar, desde que sejam excluídas as chances de emissão de cheques falsos, ou seja, aqueles sem o consentimento do respectivo banco.

Uma vez que o conjunto de entidades devedoras que precisam comprovar suas confiabilidades aos provedores é restrito aos bancos e, portanto, consideravelmente reduzido, tem-se que este padrão de incentivo é mais escalável que os padrões promissória própria e promissória de terceiro. Não obstante, esta escalabilidade é limitada não somente pela acessibilidade dos bancos, mas também pelas transações entre eles que sejam necessárias. Neste contexto, a existência de um único banco poderia, por um lado, resolver as questões das interações inter-bancárias; por outro, poderia representar um ponto único de acesso e tornar-se, facilmente, um gargalo no sistema. Já a existência de múltiplos bancos, enquanto elimina a questão do ponto único de acesso, apresenta suas dificuldades nas transações inter-bancárias.

Notas Bancárias

No padrão de incentivo notas bancárias (do Inglês, *the banknotes pattern*), as entidades também cooperam para a obtenção de uma remuneração explícita em troca, que será usufruída num momento posterior à cooperação. A diferença entre ele e os demais padrões com remuneração postergada novamente está no tipo de remuneração que, neste caso, são as notas bancárias. Pode-se entender uma nota bancária, neste contexto, como sendo uma promissória própria pré-emitida por uma autoridade central [40, 41], ou seja, uma nota bancária é um tipo particular de promissória própria que é utilizada pelas entidades consumidoras para a remuneração das provedoras. Sendo assim, a motivação para as entidades cooperarem é a obtenção destas notas, que as habilitarão a pagar pelas cooperações futuras de que venham a necessitar.

Este modelo de incentivo pode ser comparado à utilização de cédulas bancárias na sociedade atual. Pessoas podem efetuar remunerações de serviços que tenham usufruído através de cédulas que estejam em seu poder. Dessa forma, as pessoas prestam seus serviços às outras para acumularem dinheiro em espécie, que será utilizado posteriormente para o que desejam. Na computação, pode-se citar os sistemas com mecanismo de pagamento baseado em trocas de fichas, sem a existência de bancos ou contas bancárias. Nestes sistemas, as entidades recebem uma ficha – que assume o papel de uma nota bancária – ao cooperarem. Posteriormente, esta ficha poderá ser trocada por serviços ou cooperações que a entidade deseje, sendo, portanto, a

motivação para que todas cooperem.

Assim como os padrões sistema bancário, promissória própria e promissória de terceiro, este também pertence ao escopo dos padrões baseados em comércio e com remuneração postergada. Uma vez que a promessa que assegura a retribuição futura da cooperação efetuada está sob a forma de notas bancárias, a garantia de genuinidade das mesmas é fundamental. Considerando que bancos não estão presentes nas transações de cooperação e remuneração, tem-se que o risco de falsificação é maior neste padrão do que no padrão sistema bancário, o que pode demandar a utilização de mecanismos adicionais de segurança, como uma infra-estrutura criptográfica. Entretanto, o poder computacional necessário para a implantação de tal infra-estrutura pode, eventualmente, não estar disponível, como em algumas redes *ad hoc* constituídas por entidades com recursos limitados.

O tipo de remuneração do presente padrão, já falado, é constituído por notas bancárias. Assim como nos demais padrões com remuneração postergada, estas são armazenadas nos provedores, que podem ser vistos como portadores das notas. Mais do que em qualquer outro padrão de incentivo, o conceito de transferência de remuneração se aplica neste modelo. Vale ainda recapitular que, nos padrões anteriores estudados, ou a remuneração era mantida no consumidor (no padrão comunidade), ou era inexistente (no padrão coletivismo), ou dispensava armazenamento por ser imediata (no padrão escambo).

Devido ao interesse das entidades em obterem as notas bancárias, de maneira que possam pagar pelas cooperações que desejem no futuro, o comportamento não-cooperativo egoísta é eficazmente combatido. O comportamento abusivo é também evitado, considerando que as entidades são diretamente tarifadas pelas cooperações de que usufruem. Quanto ao comportamento perdoável, ele pode ser tratado de forma análoga como ocorre nos padrões sistema bancário e promissória de terceiro: caso uma entidade esteja necessitada de cooperação, mas impossibilitada de fazer o mesmo para obter notas bancárias e efetuar aquele pagamento, ela poderia convencer uma outra entidade a fazer isso por ela, ou seja, assumir a responsabilidade de sua dívida.

Levando-se em conta, novamente, que neste padrão de incentivo as notas bancárias são pré-emitidas por uma autoridade central, que assume os papéis de emissora e devedora daquele tipo de promissória, o anonimato do consumidor fica garantido independentemente da confiabilidade de qualquer entidade envolvida na transação de cooperação, assim como somente ocorre no padrão escambo. Não obstante, considerando-se a possibilidade de falsificação das notas bancárias, pode ocorrer uma demanda pela confiabilidade do consumidor, o que acarretaria na quebra de seu anonimato, ou seja, ou as notas bancárias são infalsificáveis ou o anonimato do consumidor fica comprometido. De qualquer maneira, a confiabilidade da autoridade central que pré-emite as notas bancárias é fundamental, não sendo isto, no entanto, um problema.

Como a autoridade central não precisa estar acessível nas transações de cooperação e remuneração, diferentemente dos bancos no caso do padrão sistema bancário, tem-se que o presente

padrão é mais escalável do que aquele, evitando os problemas relacionados à acessibilidade dos bancos, já mencionados (disponibilidade e transações inter-bancárias). Além disso, segundo [40, 41], este é o padrão de incentivo mais escalável dentre todos os apresentados.

3.3 Mecanismos de Incentivo em Sistemas Autônomos

Diversos trabalhos relacionados a mecanismos de incentivo têm sido apresentados na literatura. Estes trabalhos discutem questões econômicas de redes *peer-to-peer*, *ad hoc* e sistemas multiagentes, levando em consideração a autonomia das entidades em decidir entre assumir um comportamento cooperativo ou não. Enquanto alguns propõem novos esquemas de incentivo, podendo ser mapeados aos padrões de incentivo já estudados, outros investigam questões decorrentes da utilização de esquemas existentes, como melhoria da qualidade de serviço, aumento da escalabilidade, mobilidade estratégica das entidades e novos comportamentos não-cooperativos induzidos. Esta seção tem como objetivo dar uma visão geral acerca das pesquisas realizadas na área de incentivo à cooperação, através da apresentação de alguns trabalhos selecionados.

3.3.1 Altruísmo em Redes *Peer-to-Peer* de Fluxo de Mídia

Em [30], um *framework* para a modelagem explícita de altruísmo foi proposto para o contexto de redes *peer-to-peer* de fluxo de mídia com *multicast* em nível de aplicação, como Coop-Net [45, 43, 44] e SplitStream [11]. Este *framework* considera um parâmetro de sistema K que relaciona as bandas consumida e cooperada pelos clientes, podendo variar de 1 a ∞ . Quando K assume o valor 1, cada cliente coopera exatamente com a mesma banda que consome; assumindo o valor 2, cada cliente coopera com o dobro da banda que consome, e assim por diante. O limite da banda cooperada é a capacidade da banda de saída do cliente, ou seja, é o menor dos seguintes valores: (1) a banda consumida vezes K e (2) a capacidade da banda de saída. K não pode ser menor que 1 pois, neste caso, o somatório de todas as bandas cooperadas seria inferior ao somatório de todas as bandas consumidas, ou seja, a demanda por banda seria maior que a oferta. Dessa forma, o sistema saturaria e começaria a bloquear novos clientes.

Nota-se claramente o uso do padrão de incentivo escambo neste contexto, ou seja, a remuneração da cooperação recebida é efetuada simultaneamente. Sendo assim, enquanto o cliente recebe o fluxo de mídia dos nós superiores da árvore *multicast*, repassa-o aos nós inferiores da mesma utilizando a sua banda de saída.

A principal idéia deste trabalho é distribuir a banda de saída excedente acumulada no sistema igualmente a todos os clientes conectados, melhorando suas qualidades de fluxo recebidas. Ou seja, dado que cada cliente já está recebendo a sua banda de direito, calculada pela razão entre a sua banda cooperada e o parâmetro K , distribui-se gradativamente a banda excedente, fazendo com que a banda obtida por cada cliente seja maximizada, caso esta ainda não esteja em seu

valor máximo. Este valor máximo é ou a banda do fluxo distribuído ou a capacidade da banda de entrada do cliente. Não é difícil observar que esta abordagem apenas é eficaz em um ambiente heterogêneo, ou seja, onde as bandas dos clientes sejam distintas, além da natural assimetria das bandas de entrada e de saída de cada cliente. Caso esta premissa não fosse verdadeira, ou haveria muita banda excedente – quando a maioria dos clientes pudesse contribuir o que o parâmetro K determina – ou não haveria capacidade excedente a ser distribuída – quando a maioria dos clientes possuísse bandas de saída limitadas.

Um protocolo distribuído foi proposto para o cálculo da banda excedente a ser redistribuída. A princípio, este cálculo demandaria um conhecimento global acerca da utilização das bandas de todos os clientes, para se determinar o montante excedente. No entanto, utilizando os conceitos de múltiplas árvores de distribuição e codificação em múltiplos descritores, apresentados em [42, 45, 14, 43, 44, 11], propôs-se a tentativa de admissão gradativa e periódica de cada cliente nas diferentes árvores de distribuição, até que a banda excedente se esgote. Neste caso, através de prioridades atribuídas a cada cliente admitido em cada árvore distinta, pode-se determinar a ordem das árvores que o cliente deixará de participar caso a banda excedente do sistema diminua. Conseqüentemente, quando a mesma se esgota, clientes são desconectados das árvores nas quais possuem prioridades menores para dar lugar àqueles com prioridades maiores. Este protocolo distribuído foi então avaliado, através de simulações, e comparado ao que seria um protocolo “ideal”, com conhecimento global.

Os resultados mostram que o protocolo distribuído pode atingir bons níveis de utilização da banda excedente, se comparado ao protocolo com conhecimento global. Além disso, comprovou-se a eficácia da alocação da banda recebida por cada cliente, no sentido da mesma ser concordante com a política de modelagem de altruísmo inicialmente proposta. Não obstante, também foi analisado o preço de implementação desta política, em termos de rupturas nas árvores de distribuição. Ou seja, cada vez que um determinado cliente é desconectado de uma árvore, ocorre uma interrupção na recepção do descritor relacionado, o que causa uma piora temporária na qualidade recebida, até que a árvore se restabeleça. Algumas rupturas são inevitáveis, como é o caso das decorrentes de chegadas e partidas de clientes na rede. Outras são conseqüentes da política proposta, como as decorrentes das desconexões de clientes com prioridades menores que outros. Dessa forma, concluiu-se que o custo da política proposta é o dobro do custo fundamental de manutenção das árvores de distribuição.

O que não fica claro, entretanto, é o quanto este custo adicional afeta a melhoria obtida pela banda excedente determinada pelo parâmetro K de altruísmo. Por um lado, foi afirmado que a taxa de rupturas pouco varia em função de K (apesar de que, quanto maior K , maior o grau de admissão dos nós, mais largas são as árvores e, conseqüentemente, menores são as taxas de rupturas). Por outro lado, mostrou-se que pequenos aumentos de K – como de 1 a 1.5 – trazem bons resultados em termos de aumento da banda excedente. Entretanto, falta ser avaliado o ganho relativo da política, ou seja, quantos descritores em média cada cliente recebe

a mais, considerando os adicionais propiciados pela banda excedente acumulada no sistema, e os a menos, em decorrência das perdas temporárias causadas pelas rupturas nas árvores de distribuição.

3.3.2 *Multicast* com Incentivo em Redes *Peer-to-Peer* de Fluxo de Mídia

Um outro esquema de incentivo para redes *peer-to-peer* de fluxo de mídia foi proposto em [37] e é baseado em reputação, ou seja, utiliza o padrão de incentivo comunidade. O trabalho foi apresentado sobre a arquitetura SplitStream [11] mas, segundo os autores, é genérico o suficiente para poder ser utilizado em praticamente qualquer sistema com *multicast* em nível de aplicação e baseado em árvores de distribuição. A única exigência da política é que as árvores sejam periodicamente reconstruídas, de forma a maximizar as chances de um nó superior com comportamento não-cooperativo se tornar nó inferior daqueles a quem recusou cooperação, podendo, dessa forma, ser retaliado. Essa abordagem constitui um modelo de ameaça baseado nas experiências que os nós tiveram uns dos outros no passado.

Para se identificar os nós não-cooperativos na árvore *multicast*, algumas abordagens foram propostas. A principal dificuldade são os falsos positivos, ou seja, aqueles nós que estão cooperando mas, eventualmente, recusam-se a servir determinadas requisições por estarem saturados – o que configura comportamentos perdoáveis, conforme visto anteriormente. Para se contornar este problema, utilizou-se uma combinação das abordagens propostas, já que cada abordagem, utilizada individualmente, não foi suficiente para reduzir o número de falsos positivos de maneira eficaz. Com relação ao custo de reconstrução periódica das árvores, estimou-se que este seria consideravelmente baixo, não inviabilizando a proposta. Os resultados mostram que, para uma pequena fração de usuários egoístas (5%), o sistema consegue identificar e puni-los com recusa de serviço, o que se torna um incentivo interessante à cooperação. Estudos com taxas maiores de usuários egoístas foram deixados como trabalhos futuros.

É interessante notar como a proposta contorna o problema da troca de identidade quando a reputação da entidade está ruim (esta questão já foi comentada anteriormente, na Seção 3.2). Sugere-se que, num contexto com várias árvores de distribuição, como ocorre em SplitStream, cada árvore seja reconstruída de cada vez, em intervalos de tempo pré-definidos. Adicionalmente, um novo nó ingressante na rede somente pode se inscrever em novas árvores de distribuição quando estas forem reconstruídas. Dessa forma, caso um determinado nó com má reputação optasse por se desconectar e reconectar para “limpar” o seu mau conceito, ele teria que aguardar um determinado período de tempo para voltar a ter a qualidade de vídeo que tinha até então. Naturalmente, esta política de punir recém-chegados também prejudica usuários cooperativos. Por exemplo, se alunos estivessem se conectando ao sistema para assistir a uma palestra, eles precisariam entrar com uma antecedência de k intervalos de tempo, onde k é o número de árvores de distribuição que o sistema utiliza e os intervalos são os tempos entre uma

reconstrução de árvore e outra. Neste caso, se o intervalo fosse 15 segundos e $k = 16$, por exemplo, então os alunos precisariam se conectar 4 minutos antes do início da palestra.

Uma desvantagem do esquema de incentivo proposto é o fato dele não considerar o quanto cada nó deve cooperar, ao contrário do esquema anterior, deixando de explorar a heterogeneidade dos clientes. Isso significa que dois usuários com capacidades de cooperação bem distintas (bandas de saída) podem obter a mesma qualidade de fluxo (banda de entrada) desde que somente não deixem de cooperar, o que poderia colocar em dúvida as questões de justiça da política. Conseqüentemente, este problema poderia induzir novos comportamentos não-cooperativos, como usuários declarando bandas de saída menores das que efetivamente possuem, para não consumir seus recursos além do estritamente essencial.

3.3.3 Seleção de Clientes com Diferenciação de Serviço

Mais uma proposta de mecanismo de incentivo para redes *peer-to-peer* de fluxo de mídia foi apresentada em [27, 26, 28] e é baseada em percentil. Ao contrário dos dois mecanismos apresentados anteriormente, projetados para fluxos ao vivo com distribuição sincronizada através de árvores *multicast*, este mecanismo foi proposto para fluxos sob demanda com distribuição – e, conseqüentemente, remuneração – assíncrona. Neste esquema, as entidades que cooperam são remuneradas com uma pontuação, que pode ser entendida como uma reputação. Esta pontuação é mapeada para um percentil, que indica a posição, em termos de cooperação, que o usuário está classificado com relação aos demais. Por exemplo, um usuário que apresenta um percentil de 90% somente coopera menos que 10% dos integrantes da rede. O percentil do usuário é então levado em consideração no momento em que ele solicita um fluxo: somente podem servi-lo aqueles clientes que possuem um percentil inferior ao seu. Nota-se, claramente, a utilização do padrão de incentivo comunidade neste esquema de incentivo.

Mostrou-se, para este tipo de aplicação, que a seleção de bons candidatos para servir uma determinada requisição é uma questão importante na melhoria da qualidade de fluxo obtida, uma métrica definida no contexto deste trabalho para avaliar o desempenho do sistema. Entende-se por bons candidatos aqueles que apresentam boas disponibilidades, taxas ofertadas e dinâmicas de caminho de rede do fornecedor ao consumidor do fluxo. Enquanto a seleção arbitrária de candidatos apresenta qualidade de serviço imprevisível, que se agrava com o número de fornecedores escolhidos, a seleção de bons candidatos provê qualidades boas e previsíveis, independentemente do número de fornecedores considerado. Dessa forma, a motivação para cooperação introduzida por este esquema é o aumento do percentil, que possibilita a entidade efetuar uma seleção de bons candidatos mais abrangente para servirem suas requisições, obtendo qualidades de serviço melhores e mais previsíveis do que as proporcionadas pela seleção arbitrária. Diferentemente da maioria dos mecanismos de incentivo baseados em reputação e projetados para redes *peer-to-peer* de compartilhamento de arquivos, a idéia deste não é o au-

mento da disponibilidade do conteúdo desejado, mas sim uma diferenciação de serviço que promova uma melhoria na qualidade do fluxo recebido.

Para cada entidade poder determinar o quanto de cooperação é interessante efetuar, uma função de utilização de recursos foi definida. Esta função leva em consideração a qualidade de fluxo obtida e o custo de cooperação, duas outras métricas também definidas neste trabalho. Dessa forma, através de simulações, comprovou-se que o mecanismo proposto provê qualidade de serviço quase ótima quando a rede não está congestionada. Quando comparado à sua não utilização, nota-se que a rede não pode atingir os mesmos níveis de qualidade de serviço somente com a colaboração de poucos usuários altruístas, como eventualmente ocorre nas redes *peer-to-peer* de compartilhamento de arquivos. Não obstante, também foi concluído que o mecanismo pode não ser eficiente quando a rede está totalmente congestionada ou muito ociosa, situações onde a qualidade obtida pode ser muito baixa ou quase ótima, respectivamente, independentemente da sua utilização. Uma outra conclusão é que o mecanismo reduz a quantidade de dados redundantes necessários para compensar a perda de pacotes, típica de uma sessão de fluxo de mídia e normalmente contornada por técnicas de codificação de fluxo, como FEC (do Inglês, *forward error correction*) e MDC (estudada na Subseção 2.3.1).

De forma a contornar a questão da troca injusta de identidades, almejada pelos usuários com percentis baixos decorrentes de seus comportamentos não-cooperativos, este mecanismo também não privilegia os usuários recém-chegados, assim como o mecanismo anterior. Logo, clientes recém-chegados são submetidos à seleção arbitrária de candidatos para que suas requisições sejam servidas, estando sujeitos a uma política de melhor esforço, dado que a qualidade a ser obtida é imprevisível e depende dos usuários altruístas no sistema.

3.3.4 Incentivos em BitTorrent

O mecanismo de incentivo utilizado no sistema BitTorrent [7] foi descrito em [15]. Trata-se de um esquema denominado *tit-for-tat*, que nada mais é que a aplicação direta do padrão de incentivo escambo, apresentado anteriormente, no contexto de redes *peer-to-peer* de compartilhamento de arquivos. Cada usuário consegue receber dados de um determinado arquivo almejado na mesma taxa em que envia (compartilha) dados já recebidos deste mesmo arquivo a outros usuários, simultaneamente. Em outras palavras, quanto maior for a banda de saída disponibilizada pelo usuário para fins de cooperação, maior será a taxa de recebimento alcançada, minimizando-se o tempo requerido para a transferência completa do arquivo.

Neste sistema, cada arquivo disponibilizado para transferência é segmentado em pedaços de tamanhos constantes, de forma que cada usuário possa obter um pedaço diferente e repassá-lo aos demais. Dessa forma, o servidor teoricamente precisaria transmitir o arquivo disponibilizado uma única vez. Na prática, eventualmente algum pedaço do arquivo pode se tornar indisponível na rede pela desconexão do cliente que o mantinha, sendo necessário uma outra

transferência daquele pedaço pelo servidor.

Cada arquivo disponibilizado também possui relacionado o endereço de um sistema coordenador (*tracker*), responsável por auxiliar os usuários a saberem quais clientes possuem quais partes do arquivo. Entretanto, a aplicação local é a que fica responsável pela logística das conexões. Verificou-se que grafos randômicos apresentam boas propriedades em termos de robustez. Para decidir quais pedaços do arquivo deverão ser obtidos primeiramente e, ainda, de quais usuários obtê-los, algumas políticas são definidas e utilizadas pelas aplicações locais, que tomam tais decisões interagindo entre si.

A aplicação também fica responsável por controlar a relação entre as taxas de recebimento e envio dos clientes, garantindo a eficácia do esquema *tit-for-tat*. Algumas estratégias determinam quando e quanto o cliente deve cooperar mais ou menos, baseadas na sua taxa de recebimento, de forma a maximizar a utilização de seus recursos. Dado que o consumo de recursos – largura de banda, principalmente – dos servidores pode ser consideravelmente minimizado, o sistema BitTorrent se tornou uma ferramenta bastante útil para a publicação de arquivos na Internet, principalmente para aplicações de código aberto que disponibilizam seus fontes e binários, como é o caso de algumas distribuições Linux.

3.3.5 Comércio de Selos, Fichas e Confiabilidades

Em [34], um esquema de incentivo interessante baseado em selos foi proposto como uma generalização dos mecanismos baseados em reputação (padrão de incentivo comunidade) e pagamento (padrões de incentivo baseados em remuneração postergada). Apresentou-se um estudo comparativo destes dois modelos, relacionando as propriedades e os tipos de economias e incentivos por eles induzidos. Foi demonstrado que a essência de ambos pode ser capturada por um esquema mais genérico, denominado comércio de selos.

Neste mecanismo, as entidades precisam obter, primeiramente, selos personalizados da entidade da qual desejam usufruir cooperação, antes de solicitá-la, para terem meios de efetuar o respectivo pagamento. Para conseguirem estes selos personalizados, elas comercializam selos próprios, emitidos por elas mesmas, ou selos de outras entidades que estejam em seus poderes, respeitando uma taxa de troca ou câmbio que indica o valor de cada tipo de selo. Estes valores são calculados por uma entidade central e de acordo com diferentes políticas, propostas e analisadas no trabalho. Independentemente da política, toda vez que uma entidade se nega a cooperar mediante a apresentação de um selo seu por outra entidade requisitante, o valor daquele tipo de selo é diminuído, ou seja, a confiabilidade ou reputação daquela entidade diminui. O valor de um selo também pode decrescer com o número de emissões feitas pela mesma entidade, ou seja, quanto maior for o número de selos de um mesmo tipo circulando na rede, menor será o seu valor.

Uma das questões discutidas neste trabalho é se a reputação das entidades pode ser con-

siderada uma moeda. Argumentou-se que, por um lado, sim, considerando que ela pode ser comercializada de forma a induzir uma economia; mas, por outro lado, não, considerando que ela não pode ser comprada ou vendida, mas sim ganhada ou perdida. Os selos, a princípio, lembram as promissórias próprias, vistas anteriormente, já que as cooperações são efetuadas mediante a apresentação deles às entidades que os emitiram. Entretanto, de outra forma, eles são pré-emitidos e posteriormente comercializados, sujeitos a preços variáveis baseados nas diferentes reputações existentes. Neste caso, lembram mais as notas bancárias, com a diferença de terem sido pré-emitidos pelas entidades, ao invés de por uma única autoridade central.

3.3.6 Mobilidade em Redes sem Fio *Ad Hoc* com Incentivos

O enfoque do trabalho apresentado em [19] não é uma proposta de um novo esquema de incentivo, mas sim o estudo da mobilidade estratégica de nós em redes sem fio *ad hoc* dado que mecanismos de incentivo existentes estejam sendo utilizados, como os de reputação (padrão de incentivo comunidade) e os de pagamento (padrões de incentivo baseados em remuneração postergada). Esta mobilidade é induzida pelo fato das entidades autônomas procurarem maximizar a utilização de seus recursos: dado que os mecanismos de incentivo obrigam-nas a cooperar de qualquer maneira, estratégias de movimentações geográficas são então assumidas de forma a priorizar seus interesses.

Por exemplo, com esquemas de reputação, as entidades procuram minimizar a quantidade de tráfego repassado – para economizar seus recursos, já que não se ganha uma remuneração explícita por cada repasse efetuado – e se tornam indiferentes ao número de saltos necessários para que seus pacotes alcancem seus destinos – não se cobra pelo número de saltos. Neste caso, elas procuram ficar mais nas bordas da topologia da rede, onde há menos tráfego a ser repassado, não importando que seus pacotes gastem muitos saltos para atingirem extremidades ou bordas opostas. Por outro lado, com esquemas de pagamento, as entidades procuram maximizar a quantidade de tráfego repassado, dado que se recebe por cada repasse efetuado, e minimizar o número de saltos necessários para que seus pacotes alcancem seus destinos, dado que se paga às demais entidades que efetuam os repasses de pacotes. Já neste caso, elas procuram ficar mais próximas ao centro da topologia da rede, onde há maior tráfego a ser repassado e menores distâncias – em termos de saltos – para os diferentes destinos de seus pacotes.

Através de modelos baseados em teoria dos jogos e simulações, comprovou-se que a mobilidade estratégica de usuários pode degradar o desempenho da rede, devido às topologias ineficientes que se formam nos pontos de equilíbrio dos jogos. De maneira geral, tratou-se de um problema causado por comportamentos não-cooperativos induzidos pelo mecanismo de incentivo utilizado. Com relação às topologias ineficientes formadas, constatou-se duas tendências: uma reunidora, que leva os nós a ficarem o mais próximos possível uns dos outros; e outra dispersiva, que os leva à formação de cadeias simples. Em ambos os casos, apesar das

topologias finais serem distintas, concluiu-se que a mobilidade estratégica, em geral, degrada a capacidade das redes sem fio *ad hoc*. Logo, futuros mecanismos de incentivo propostos para este contexto devem levar em consideração este fator, além de combaterem os comportamentos não-cooperativos já esperados, como o não-roteamento de pacotes em decorrência dos custos elevados de banda e energia.

3.4 Considerações Finais

Viu-se que, com a predominância da arquitetura cliente-servidor até pouco tempo atrás, não havia, até então, a preocupação de se garantir determinados padrões de comportamento das entidades constituintes, já que estas tinham papéis bem definidos na rede. No entanto, as arquiteturas descentralizadas têm emergido como forma de se solucionar os problemas de escalabilidade impostos pela arquitetura anterior. Neste novo contexto, as entidades podem tanto servir quanto serem servidas; além disso, são livres para decidir entre cooperar ou não, principalmente devido à ausência de servidores ou autoridades centrais. Esses fatos, juntamente com o custo de cooperação, que muitas vezes é alto devido ao consumo de recursos locais, levam as entidades a praticarem comportamentos não-cooperativos.

Por outro lado, a maioria dos sistemas descentralizados contam com tais cooperações para se tornarem eficientes e, principalmente, viáveis, sendo este fato particularmente importante nas aplicações que distribuem conteúdo com sensibilidade temporal, como é o caso dos fluxos de mídia. Dessa forma, mecanismos de incentivo à cooperação são necessários neste contexto, tendo recebido uma atenção especial da literatura recente. Tais mecanismos foram o objeto de estudo deste capítulo.

Estudou-se, primeiramente, os diferentes tipos de comportamentos não-cooperativos comuns nos sistemas descentralizados, como redes *peer-to-peer*, *ad hoc* e sistemas multiagentes. Em seguida, foram apresentados os padrões de incentivo existentes, suas características e de que forma combatem os comportamentos não-cooperativos, com o intuito de se prover informações para a compreensão de uma maior abrangência de esquemas de incentivo existentes, além de futuras propostas que ainda estão por vir. Dado que o número de trabalhos relacionados é vasto, incluindo as propostas de novos esquemas e os estudos de questões decorrentes da utilização de esquemas já conhecidos, realizou-se uma seleção dos mesmos, que foi apresentada na seqüência.

Nota-se que um bom mecanismo de incentivo é aquele que combate eficazmente os comportamentos não-cooperativos comuns, bem como aqueles induzidos por sua utilização. Além disso, idealmente, isenta os comportamentos perdoáveis de punição, questão esta que nem sempre pode ser tratada de forma trivial, de acordo com o estudo feito na Seção 3.2.

Entretanto, nenhum mecanismo de incentivo é ideal para todos os tipos de contextos, ou seja, é importante levar em consideração as características específicas de cada tipo de aplica-

ção no projeto do esquema de incentivo a ser empregado, ou seja, na escolha dos padrões de incentivo a serem utilizados. A Seção 3.3 ilustrou como diferentes esquemas, compostos por padrões distintos, podem ser pertinentes aos mais variados tipos de sistemas, como fluxo de mídia ao vivo, fluxo de mídia sob demanda e compartilhamento de arquivos. Finalmente, pode-se mencionar que há uma tendência de se combinar diferentes padrões de incentivo na definição de novos esquemas de incentivo, de forma a se reunir as vantagens e compensar as deficiências de cada um [40, 41].

Capítulo 4

Uma Proposta de Mecanismo de Incentivo

A partir do estudo realizado no Capítulo 3, sobre os padrões de incentivo à cooperação existentes, e no Capítulo 2, sobre os sistemas *peer-to-peer* de fluxo de mídia ao vivo, elaborou-se a proposta de um novo mecanismo de incentivo para este tipo de sistema. Considerou-se a recomendação mencionada em [40, 41] de se combinar diferentes padrões de incentivo na definição de novos mecanismos, de forma a se reunir as vantagens e compensar as deficiências de cada um. Foi levado também em consideração as características específicas do tipo de aplicação para a qual o mecanismo foi projetado, de maneira a se escolher os padrões de incentivo mais apropriados.

Mostra-se, adicionalmente, como o novo mecanismo de incentivo pode ser agregado, em particular, ao sistema CoopNet, descrito no Capítulo 2. Sugere-se, para isso, a modificação do protocolo de adaptação de banda apresentado em [44]. A principal vantagem desta abordagem é o reaproveitamento de toda a estrutura de suporte à heterogeneidade e controle de congestionamento já existente no referido trabalho.

O restante deste capítulo está organizado da seguinte forma. Na Seção 4.1, apresenta-se a proposta do novo mecanismo de incentivo, com sua definição, variáveis e algoritmos. Na Seção 4.2, descreve-se as modificações sugeridas no protocolo de adaptação de banda apresentado em [44], a fim de que o mecanismo proposto possa ser agregado ao sistema CoopNet. Finalmente, na Seção 4.3, o capítulo é encerrado com algumas considerações finais.

4.1 Proposta de um Novo Mecanismo de Incentivo

Nos sistemas *peer-to-peer* de fluxo de mídia ao vivo, a cooperação esperada dos clientes está no repasse dos fluxos nas árvores de distribuição, sendo, portanto, uma contribuição constituída primariamente de largura de banda e secundariamente de processamento e memória. Diferentemente do que acontece nas redes *peer-to-peer* de compartilhamento de arquivos, os interesses individuais de cada cliente não estão focados na disponibilidade de um grande número de ar-

quivos, mas sim na obtenção de um determinado fluxo que está sendo transmitido ao vivo na sua melhor qualidade possível [27]. Assim, um mecanismo de incentivo que explore esses interesses deve prover uma melhor qualidade aos clientes que cooperam mais, ou seja, àqueles que disponibilizam uma maior banda de saída para fins de repasse. Este incentivo deve ir até o ponto onde o cliente se torna satisfeito com a qualidade de fluxo recebida, a partir do qual nenhuma cooperação adicional se justifica – considerando um comportamento racional e não altruísta dos clientes.

Conforme visto no Capítulo 3, o padrão de incentivo escambo é o mais apropriado para serviços de natureza síncrona, como é o caso dos sistemas *peer-to-peer* de fluxo de mídia ao vivo, nos quais o cliente coopera enquanto estiver conectado à rede. Se comparado aos demais padrões de incentivo existentes – por exemplo, comunidade, sistema bancário e notas bancárias –, o padrão escambo possui grandes vantagens tais como anonimato, persistência, escalabilidade e localização. O que o inviabiliza, na maioria dos casos, é a necessidade de se efetuar as remunerações enquanto se está usufruindo das cooperações. Não obstante, essa restrição não é um empecilho para os sistemas em questão: o repasse de descritores, que é a cooperação esperada dos clientes, pode ser estimulado enquanto eles obtêm o fluxo distribuído. Encerrada a transmissão deste fluxo, nenhuma cooperação adicional é exigida, até mesmo porque ela não é mais necessária.

O mecanismo proposto no presente trabalho tem como incentivo primário o padrão escambo, de forma a aumentar a cooperação dos clientes e, conseqüentemente, a escalabilidade da rede. Decorre que, tendo o sistema mais recursos provenientes dos clientes, ele pode oferecer uma melhor qualidade de fluxo, pois haverá maior espaço para novas admissões nas árvores de distribuição. Assim, pretende-se aliviar o problema decorrente do comportamento egoísta dos usuários neste tipo de sistema.

Um outro problema encontrado nos sistemas *peer-to-peer* de fluxo de mídia ao vivo é o curto tempo de permanência dos clientes na rede, o que demanda esforços para se contornar as interrupções causadas nas entregas de fluxo com sensibilidade temporal quando as desconexões ocorrem [45]. Tal problema é particularmente importante nos sistemas em questão devido ao uso de *multicast* na camada de aplicação, pois dependendo da posição de um nó nas árvores de distribuição, diversos clientes podem ser afetados quando ele se desconecta da rede [43].

Para incentivar os clientes a ficarem mais tempo conectados na rede, deve-se oferecer algo que lhes interesse. O padrão escambo já oferece um fluxo de qualidade proporcional ao montante de largura de banda disponibilizada para fins de cooperação. Logo, uma opção seria amenizar as taxas de cooperação impostas pelo padrão escambo àqueles clientes que ficarem conectados por mais tempo. Entretanto, esta abordagem introduz um deficit de recursos no sistema. Para compensar este deficit, aumenta-se as taxas de cooperação daqueles clientes que acabaram de chegar ou que estão conectados há pouco tempo.

Assim, incluiu-se no mecanismo proposto um incentivo secundário, baseado em reputação,

que modifica as taxas de cooperação do padrão escambo no incentivo primário. As reputações dos clientes são diretamente proporcionais a seus tempos de permanência e suas taxas de cooperação são inversamente proporcionais a suas reputações. Com isso, espera-se amenizar o problema das rupturas em árvores, quando necessário, através do estímulo aos clientes a permanecerem mais tempo conectados no sistema. Neste esquema, a reputação máxima é a dos servidores, que estão sempre conectados na rede.

A seguir, define-se o mecanismo proposto. Seja R_i a reputação do cliente i e R_s a reputação do servidor. Tem-se que $0 \leq R_i \leq R_s = 1$.

Seja B_{S_i} a banda de saída do cliente i disponibilizada para fins de cooperação, B_{E_i} a sua banda de entrada utilizada para a recepção do fluxo na qualidade desejada e TC_i a sua taxa de cooperação, imposta pelo padrão escambo. Pode-se modelar a relação entre as bandas cooperada e consumida através da seguinte equação:

$$B_{S_i} = TC_i \cdot B_{E_i}.$$

O incentivo primário é implementado de forma que esta relação seja respeitada para cada cliente no sistema, ou seja, que não se consuma mais banda do que se tem direito, de acordo com a cooperação efetuada, e que não se coopere mais do que o necessário, de acordo com a qualidade desejada.

Para a implementação do incentivo secundário, basta variar as taxas de cooperação de cada cliente de acordo com suas reputações, o que é feito da seguinte forma. Se o cliente i acabou de se conectar, tem-se que $R_i = 0$ e, portanto, TC_i assume o seu valor máximo. Isso significa que a banda de saída a ser cooperada por este cliente, para uma determinada banda de entrada, é máxima. Por outro lado, após um determinado tempo de conexão, definido por um *threshold* TR (Tempo Requerido de Cooperador Estável), tem-se que $R_i = R_s = 1$, o que significa que o cliente i tornou-se um “cooperador estável” daquela rede e TC_i , por sua vez, assume o seu valor mínimo. Em outras palavras, este cliente “paga” o mínimo pelo serviço usufruído.

Enquanto o tempo de permanência do cliente i não ultrapassar TR , calcula-se R_i de forma proporcional ao cliente mais antigo da rede que ainda não tenha se tornado “cooperador estável”. Seja t_* o tempo de conexão deste último e t_i o tempo de conexão do cliente i . Dessa forma, a reputação do cliente i é calculada através da seguinte equação:

$$R_i = t_i / (t_* + 1).$$

Adicionou-se uma unidade ao denominador desta equação a fim de que R_i somente seja igual a R_s após $t_i \geq TR$. Seja LI_{TC} e LS_{TC} , respectivamente, os limites inferior e superior das taxas de cooperação praticadas no sistema. Pode-se calcular a taxa de cooperação do referido cliente através da seguinte equação:

$$TC_i = LS_{TC} - R_i(LS_{TC} - LI_{TC}).$$

O mecanismo de incentivo proposto no presente trabalho é constituído pelos incentivos primário e secundário descritos acima. A Tabela 4.1 resume todas as variáveis envolvidas neste contexto. A Figura 4.1 apresenta o algoritmo responsável pela determinação de R_i , TC_i , t_i e t_* . A Figura 4.2 apresenta o algoritmo responsável pelo balanceamento de B_{Ei} e B_{Si} , baseado em TC_i . Estes algoritmos são executados a cada chegada e partida de clientes, bem como periodicamente para os clientes já conectados no sistema. A atualização de t_* pode ocorrer sempre que um cliente se conecta, desconecta ou é promovido a “cooperador estável”. Nota-se, por fim, a necessidade de se determinar os parâmetros LI_{TC} , LS_{TC} e TR , de forma a calibrar o funcionamento do mecanismo proposto, o que será discutido no Capítulo 6.

| Nome | Descrição |
|-------------|---|
| B_{Ei} : | Banda de entrada (consumida) utilizada pelo cliente i para assistir aos seus fluxos na qualidade desejada. |
| B_{Si} : | Banda de saída (cooperada) disponibilizada pelo cliente i para fins de repasse de descritores no sistema. |
| R_s : | Reputação do servidor (= 1). |
| R_i : | Reputação do cliente i ($0 \leq R_i \leq R_s$). |
| TC_i : | Taxa de cooperação do cliente i . |
| LI_{TC} : | Limite inferior de TC . |
| LS_{TC} : | Limite superior de TC . |
| t_i : | Tempo de conexão do cliente i na rede. |
| t_* : | Tempo de conexão do cliente mais antigo da rede cuja $R_i < R_s$. |
| TR : | Tempo requerido de conexão para um cliente se tornar um “cooperador estável”, ou seja, $R_i = R_s$ e $TC_i = LI_{TC}$. |

Tabela 4.1: Variáveis do mecanismo de incentivo proposto.

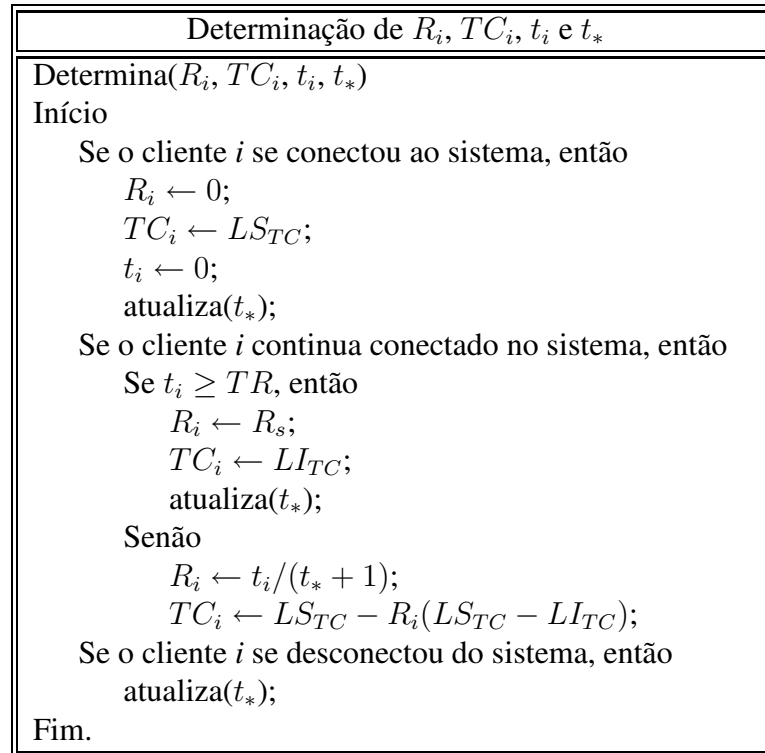


Figura 4.1: Determinação dos parâmetros R_i, TC_i, t_i e t_* do mecanismo de incentivo proposto.

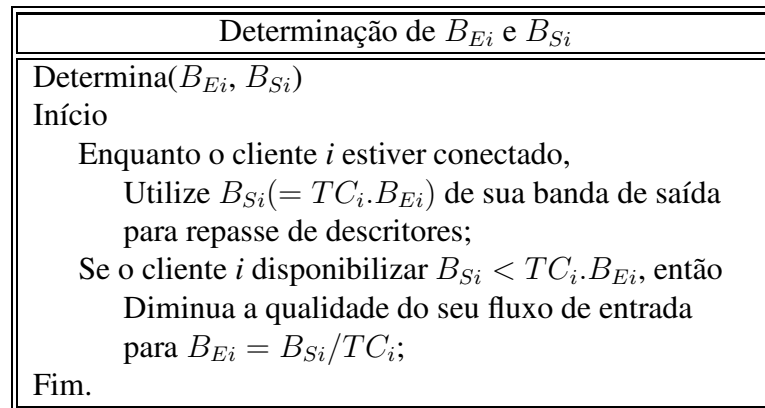


Figura 4.2: Determinação dos parâmetros B_{Ei} e B_{Si} do mecanismo de incentivo proposto, baseados em TC_i .

4.2 Integração em CoopNet

Para se agregar o mecanismo de incentivo proposto ao sistema CoopNet, modificou-se o protocolo de adaptação de banda apresentado em [44], de forma a se reaproveitar toda a estrutura de suporte à heterogeneidade e controle de congestionamento já existente no referido sistema. Em linhas gerais, o protocolo de adaptação de banda original opera da seguinte forma:

1. Descarte tráfego, de entrada ou de saída, quando houver congestionamento no respectivo enlace;
2. Adicione tráfego, de entrada ou de saída, após um determinado período de tempo sem congestionamento no respectivo enlace.

Após as modificações, o novo protocolo passou a operar da seguinte forma:

1. Descarte tráfego de entrada quando houver congestionamento no enlace de entrada *ou* quando $B_{Si} < TC_i \cdot B_{Ei}$;
2. Descarte tráfego de saída quando houver congestionamento no enlace de saída *ou* quando $B_{Si} > TC_i \cdot B_{Ei}$;
3. Adicione tráfego de entrada após um determinado período de tempo sem congestionamento no enlace de entrada *e* com $B_{Si} > TC_i \cdot B_{Ei}$;
4. Adicione tráfego de saída após um determinado período de tempo sem congestionamento no enlace de saída *e* com $B_{Si} < TC_i \cdot B_{Ei}$.

Cabe ressaltar ainda que o cliente *i* pode determinar a qualidade desejada de fluxo recebida ou a sua banda de saída a ser cooperada.

Os algoritmos das Figuras 4.1 e 4.2 são executados no servidor, determinando, periodicamente, para cada cliente, os resultados das inequações acima indicadas. Como o servidor já está envolvido em todas as operações de descarte e adição de tráfego do protocolo de adaptação original, a implementação das cláusulas adicionais descritas é simples e viável.

É interessante notar que as operações de adição e de descarte de tráfego fazem com que as bandas de consumo e de cooperação sejam variadas de forma discreta. O mecanismo de incentivo proposto, por sua vez, opera com variáveis contínuas. Fica, portanto, o novo protocolo de adaptação de banda também responsável por efetuar os devidos arredondamentos.

Por fim, reaproveitou-se as políticas do protocolo de adaptação de banda original que tomam as decisões a respeito de quais descritores adicionar ou descartar, além da inferência dos locais de congestionamento. Essas políticas, juntamente com o mecanismo de incentivo proposto, no novo protocolo de adaptação de banda modificado, escolhem de quais árvores um determinado cliente deve se desconectar ou, ainda, em quais nós um determinado cliente órfão deve se conectar, nas diferentes árvores de distribuição.

4.3 Considerações Finais

Neste capítulo, apresentou-se a proposta de um novo mecanismo de incentivo à cooperação para sistemas *peer-to-peer* de fluxo de mídia ao vivo (com distribuição sincronizada). Este mecanismo é composto por duas partes, o incentivo primário e o incentivo secundário. Espera-se que o incentivo primário aumente a escalabilidade do sistema, bem como maximize as qualidades de fluxo recebidas pelos clientes. Por outro lado, espera-se que o incentivo secundário amenize os problemas decorrentes das rupturas em árvores de distribuição, causadas pelo curto tempo de permanência dos clientes neste tipo de sistema. Mostrou-se, ainda, como o novo mecanismo proposto pode ser agregado, em particular, ao sistema CoopNet, discutido no Capítulo 2. Sugeriu-se, para isso, a modificação do protocolo de adaptação de banda apresentado em [44]. A principal vantagem desta abordagem é o reaproveitamento de toda a estrutura de suporte à heterogeneidade e controle de congestionamento já existente no referido trabalho.

Capítulo 5

Um Simulador de Redes *Peer-to-Peer* de Fluxo de Mídia ao Vivo

Uma contribuição adicional do presente trabalho é um simulador de eventos discretos para redes *peer-to-peer* de fluxo de mídia ao vivo, construído para a avaliação da eficácia do mecanismo de incentivo proposto. Considerou-se a proposta descrita em [43, 44], na qual múltiplas árvores de distribuição e codificação em múltiplos descritores são usadas como forma de se prover robustez e adaptabilidade à heterogeneidade dos clientes. Utilizou-se o protocolo de admissão *top-down* também proposto em [43, 44], e o novo protocolo de adaptação de banda, descrito na Seção 4.2, foi implementado.

O simulador foi desenvolvido usando-se o paradigma de orientação a objetos, o que facilita a sua modificação para ser reutilizado para outros tipos de sistemas relacionados. Por exemplo, no futuro pode-se facilmente trocar o mecanismo de incentivo implementado por um outro que se queira avaliar, reaproveitando todo o *framework* de simulação de eventos discretos e os controladores que implementam as operações do tipo de sistema considerado. Este simulador é descrito neste capítulo, incluindo sua arquitetura, configuração e ferramentas utilizadas no seu desenvolvimento.

O restante do capítulo está organizado da seguinte forma. Na Seção 5.1, apresenta-se a arquitetura do simulador. Na Seção 5.2, as configurações do mesmo são descritas, incluindo as variáveis aleatórias usadas e os parâmetros criados. Na Seção 5.3, as ferramentas utilizadas no desenvolvimento são mencionadas. Finalmente, na Seção 5.4, considerações finais são tecidas.

5.1 Arquitetura

Nesta seção, apresenta-se a arquitetura do simulador desenvolvido, a qual pode ser visualizada na Figura 5.1. Os três níveis mais abstratos existentes são: *apresentação*, *repositório* e *controlador*. A apresentação é responsável por gerenciar a saída de dados do simulador, ou seja, os

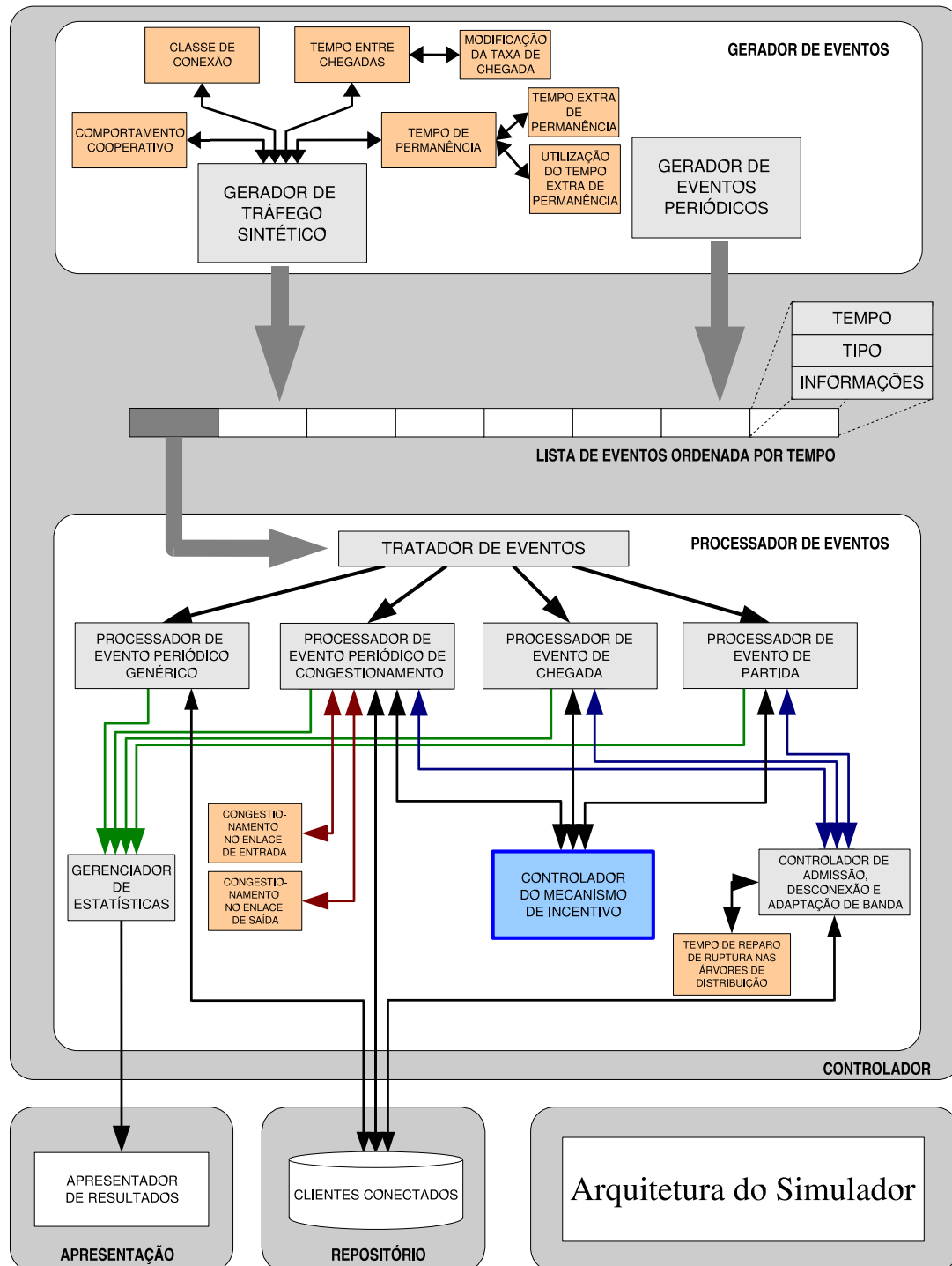


Figura 5.1: Arquitetura do simulador desenvolvido.

traces produzidos. Este gerenciamento inclui a seleção, formatação e apresentação das informações relevantes, incluindo as métricas de simulação coletadas, com base nas configurações de detalhamento dos *logs*.

O repositório é responsável pela manutenção de todas as informações relacionadas aos *clientes conectados* no sistema, tais como banda de entrada, banda de saída, tempo de chegada, tempo de permanência, tempo de ocorrência do último congestionamento, reputação e taxa de cooperação. Diversas estruturas de dados são utilizadas para a implementação de diferentes níveis de abstração como, por exemplo, “cliente”, “nó”, “árvore” e “floresta”. A representação das múltiplas árvores de distribuição, utilizadas no tipo de sistema simulado, também é mantida no repositório.

O controlador é responsável pela implementação da lógica do simulador. Ele é constituído por uma *lista de eventos ordenada por tempo* e dois subsistemas: *gerador de eventos* e *processador de eventos*. O gerador de eventos alimenta a lista de eventos com acessos intercalados ou não ordenados. Por exemplo, ele pode inserir, nesta ordem, o evento de chegada de um cliente no tempo 1, o evento de partida do mesmo cliente no tempo 10 e o evento de chegada de um outro cliente no tempo 2. O processador de eventos, diferentemente, apenas acessa a lista de eventos na sua primeira posição, retirando dela o evento mais antigo que ainda não tenha sido processado. Um parâmetro de configuração determina o número de eventos que são acumulados na lista antes de serem processados, permitindo um balanceamento entre desempenho e consumo de memória do simulador.

O gerador de eventos é constituído por duas entidades principais: *gerador de tráfego sintético* e *gerador de eventos periódicos*. O gerador de tráfego sintético é responsável pela produção de eventos de chegada e de partida de clientes. Para isso, ele usa geradores de números aleatórios que seguem distribuições probabilísticas específicas, a serem melhor explicadas na Subseção 5.2.1. Para cada cliente que chega, os geradores *classe de conexão* e *comportamento cooperativo* são consultados, respectivamente, para a determinação da classe de conexão e do tipo de comportamento assumido por aquele cliente. Os tempos entre chegadas são dados pelo gerador *tempo entre chegadas*, que por sua vez utiliza o gerador *modificação da taxa de chegada*. Os tempos de permanência dos clientes no sistema são definidos pelo gerador *tempo de permanência*, que utiliza os geradores *tempo extra de permanência* e *utilização do tempo extra de permanência*. Mais informações sobre estes geradores serão dadas na Seção 5.2.

O gerador de eventos periódicos é responsável pela produção de eventos periódicos. Um evento periódico é aquele que se repete em intervalos de tempo fixos, havendo dois tipos: o genérico e o de congestionamento. O genérico é aquele que dispara os processamentos periódicos genéricos como, por exemplo, a medição da capacidade de admissão de novos clientes no sistema e a contagem do número de clientes e de cooperadores estáveis existentes. O evento periódico de congestionamento dispara os processamentos periódicos de congestionamento, tais como os sorteios de ocorrência de congestionamento para cada cliente, os aumentos e diminui-

ções de tráfego dos clientes com base no novo protocolo de adaptação de banda e os cálculos do mecanismo de incentivo utilizado.

O processador de eventos é regido pelo *tratador de eventos*. Este é responsável por analisar o tipo do evento retirado da lista de eventos e delegar o seu tratamento ao processador de eventos responsável. Existem quatro tipos de processadores de eventos, *genérico*, *de congestionamento*, *de chegada* e *de partida*, que tratam os quatro respectivos tipos de eventos existentes. Todos eles enviam informações das operações processadas ao *gerenciador de estatísticas*. Este último é responsável por armazenar as contagens de diferentes variáveis, suas médias, medianas, variâncias, desvios padrão, valores mínimos e máximos, bem como os resultados para as diferentes métricas coletadas no simulador. O gerenciador de estatísticas envia as informações de saída das simulações ao *apresentador de resultados*, pertencente à abstração de apresentação.

Os processadores de eventos periódicos genérico e de congestionamento acessam diretamente o repositório de clientes conectados, já que diversas operações realizadas por eles envolvem o processamento de todos os clientes conectados. O processador de eventos periódicos de congestionamento, adicionalmente, consulta os geradores de números aleatórios *congestionamento no enlace de entrada* e *congestionamento no enlace de saída*. Estes determinam a ocorrência de congestionamento no respectivo enlace de cada cliente processado.

O *controlador de admissão, desconexão e adaptação de banda* é acessado pelos processadores de eventos de chegada e de partida, bem como pelo processador de evento periódico de congestionamento. Ele é responsável pelo tratamento das admissões e desconexões dos clientes, inserindo ou retirando-os do repositório de clientes conectados e, conseqüentemente, das árvores de distribuição. O controlador de admissão, desconexão e adaptação de banda também é responsável pelos aumentos e diminuições das bandas consumidas e cooperadas pelos clientes, com base nos cálculos do mecanismo de incentivo utilizado e ocorrências de congestionamento. Como ele ainda processa as rupturas ocorridas nas árvores de distribuição, consulta o gerador de números aleatórios *tempo de reparo de ruptura nas árvores de distribuição* para a determinação das durações das mesmas.

Finalmente, o *controlador do mecanismo de incentivo* é responsável pela realização dos cálculos do mecanismo de incentivo utilizado. Para isso, recebe informações dos processadores de eventos de chegada e de partida, bem como do processador de evento periódico de congestionamento. Além disso, fornece informações a estes três módulos acerca das bandas consumida e cooperada para cada cliente processado, de acordo com sua reputação e taxa de cooperação. Com isso, tem-se o mecanismo de incentivo proposto implementado no simulador.

5.2 Configuração

A seguir, descreve-se as variáveis aleatórias utilizadas e os parâmetros criados para configuração do simulador.

5.2.1 Variáveis Aleatórias

Nesta subseção, discute-se todas as variáveis aleatórias (distribuições probabilísticas) utilizadas no simulador.

Tempo entre Chegadas

A variável aleatória *TempoEntreChegadas* representa o tempo entre chegadas de novos clientes no sistema, sendo contínua.

Não se diferenciou o conceito de chegadas de diferentes clientes com o conceito de chegadas de diferentes requisições de um mesmo cliente (conceito de sessão). Esta diferenciação apenas é importante nas aplicações de fluxo sob demanda, nas quais um mesmo cliente pode permanecer no sistema requisitando diferentes fluxos. Nas aplicações de fluxo ao vivo, os clientes chegam para assistir a uma única programação existente, que está sendo transmitida de forma sincronizada. Quando um determinado cliente não deseja mais assistir àquela programação, ele se desconecta do sistema, o que acarreta na reorganização das árvores de distribuição. Caso o mesmo cliente opte por uma reconexão, a nova requisição resultante será tratada como se fosse a chegada de um novo cliente, por mais breve que tenha sido o tempo decorrido desde a sua última desconexão, uma vez que, de qualquer forma, novas posições nas árvores de distribuição terão que ser alocadas.

Em [53], os autores caracterizaram o processo de chegada de clientes em aplicações de fluxo ao vivo como sendo *Poisson* não estacionário, para grandes intervalos de tempo; para intervalos menores, concluiu-se que processos *Poisson* estacionários podem ser utilizados. Em [57], sugeriu-se que uma boa maneira de se implementar um gerador *Poisson* não estacionário é através da concatenação de múltiplos processos estacionários de menor duração, cada qual assumindo uma taxa média de chegada condizente com o número de clientes no sistema naquele intervalo de tempo. Com relação à duração dos intervalos estacionários, os autores concluíram que 15 minutos é um valor que melhor modela o processo não estacionário como um todo, após terem testado outros intervalos.

Um gerador de tempos entre chegadas de clientes, que segue as sugestões da literatura descritas acima, foi implementado no simulador desenvolvido. Processos de chegada *Poisson* estacionários são gerados a cada 15 minutos com uma determinada taxa de chegada. Esta taxa é variada a cada período estacionário decorrido, podendo ser aumentada ou diminuída com igual probabilidade (50%). Criou-se dois parâmetros de configuração, *LimiteInferiorTaxaChegada* e *LimiteSuperiorTaxaChegada*, que determinam os limites inferior e superior que a taxa de chegada pode assumir. O valor inicial da mesma é determinado como sendo a média aritmética simples entre estes dois limites. Um terceiro parâmetro, *ValorModificacaoTaxaChegada*, determina o valor a ser utilizado para se incrementar ou decrementar a taxa de chegada a cada período estacionário decorrido. Por fim, um quarto parâmetro, *Peri-*

odoReinicializacaoTaxaChegada, determina o tempo decorrido após o qual se realiza uma reinicialização da taxa de chegada de volta ao seu valor inicial.

A funcionalidade de reinicialização da taxa de chegada ao seu valor inicial após um determinado período de tempo foi introduzida para se aproximar o tráfego gerado das caracterizações feitas em [53, 57]. Nota-se, nestes dois trabalhos, que as variações de tráfego repetem-se a cada período de 24 horas, reproduzindo os padrões de acesso dos usuários em um dia típico de trabalho ou de final de semana, incluindo os picos de acesso (como horários de almoço e terminos de expediente) e períodos de baixo acesso (como de madrugada, após um determinado horário). Sendo assim, o valor 86400 segundos (24 horas) foi utilizado no parâmetro *PeriodoReinicializacaoTaxaChegada*.

Com relação aos limites inferior e superior da taxa de chegada, em [53] sugere-se taxas que variam entre 5 e 20 chegadas por minuto, com picos de até 40 chegadas por minuto (*flash crowds*), com base em um estudo feito com *traces* da rede Akamai. Em [57] foi sugerida uma taxa média de 16.71 chegadas por minuto. Já em [45], considerou-se um intenso *flash crowd* provocado pelo evento de 11 de setembro de 2001 nos EUA, constatando-se 84 clientes desconectando do sistema a cada segundo; assumindo que o sistema estava no seu estado estacionário (o período de tempo analisado foi em torno de uma hora), tem-se uma taxa média em torno de 80 chegadas por segundo. Baseado nestes valores da literatura, utilizou-se 0.083 chegadas por segundo (5 chegadas por minuto) na variável *LimiteInferiorTaxaChegada* e 0.333 chegadas por segundo (20 chegadas por minuto) na variável *LimiteSuperiorTaxaChegada*. Para a reprodução das situações de *flash crowd*, utilizou-se 80 chegadas por segundo em ambas as variáveis, o que fixou a taxa de chegada, tornando o período simulado estacionário. Tal fato não constitui um problema, dado que os períodos de *flash crowd* considerados são inferiores a 15 minutos.

Finalmente, para se calcular o parâmetro *ValorModificacaoTaxaChegada*, realizou-se o seguinte procedimento. Tem-se que 96 sorteios de aumento ou diminuição da taxa de chegada ocorrem no período de reinicialização da variável (um a cada 15 minutos no intervalo de 24 horas). Dessa forma, gerou-se 96 sorteios 10^6 vezes, usando 10^6 sementes diferentes. Para cada grupo de 96 sorteios (provenientes de uma mesma semente), registrou-se o máximo de repetições consecutivas de 0 ou 1 que ocorreram. Em seguida, calculou-se a média desses 10^6 valores máximos, o que resultou no valor 6. Isso significa que, na média, a taxa de chegada pode ser aumentada ou diminuída até 6 vezes consecutivas no período de reinicialização considerado. Sendo assim, para se obter uma maior abrangência dentro dos limites inferior e superior adotados, dividiu-se esta faixa de valores por 6: $(LimiteSuperiorTaxaChegada - LimiteInferiorTaxaChegada)/6$. Logo, utilizou-se o valor 0.0416 na variável *ValorModificacaoTaxaChegada*.

Tempo de Permanência

A variável aleatória *TempoPermanencia* representa o tempo de permanência dos clientes no sistema, sendo contínua.

Como não se diferenciou as chegadas de diferentes clientes com as chegadas de diferentes requisições de um mesmo cliente, também não se diferenciou os tempos de permanência de clientes com as durações de sessões de um mesmo cliente. Conforme mencionado na seção anterior, o conceito de sessão apenas é importante nas aplicações de fluxo sob demanda, sendo irrelevante nas de fluxo ao vivo. Conseqüentemente, no simulador desenvolvido, cada cliente apresenta uma única sessão durante a sua permanência no sistema, sendo o tempo de duração desta sessão equivalente ao seu tempo de permanência.

Em [53], observou-se que as durações de sessões nas aplicações de fluxo ao vivo seguem distribuições de cauda longa, sendo a *lognormal* predominante. Em [57], os autores caracterizaram dois tipos de tráfego para aplicações de fluxo ao vivo: *reality show*, e noticiários e esportes. Em ambas, a distribuição *lognormal* também foi a encontrada para os tempos de duração de sessões, com diferentes parâmetros estimados. Como o tráfego de noticiários e esportes é o que mais se aproxima da natureza do sistema simulado, utilizou-se, no simulador desenvolvido, os respectivos parâmetros de distribuição *lognormal* estimados: $\mu = 5.74$ e $\sigma = 2.01$. Foi utilizado, também, $\sigma = 1.44$, proveniente do tráfego de *reality show*, para um cenário específico no qual se considerou tempos de sessão mais curtos para o estudo das rupturas em árvores de distribuição.

Sendo assim, um gerador de tempos de permanência de clientes, que segue as constatações da literatura descritas acima, foi implementado no simulador desenvolvido. Como as distribuições de cauda longa geralmente são de difícil simulação, utilizou-se a biblioteca *GNU Scientific Library* [23] para a geração de valores segundo a distribuição *lognormal*. Em particular, usou-se a função *gsl_cdf_lognormal_Pinv*, que recebe um número aleatório uniformemente distribuído e os parâmetros μ e σ , retornando o número aleatório calculado pelo método da inversa da função distribuição acumulada.

Tempo Extra de Permanência

A variável aleatória *TempoExtraPermanencia* representa o tempo extra de permanência que os clientes ficam no sistema em decorrência do uso do incentivo secundário do mecanismo proposto, sendo contínua. O tempo sorteado por esta variável aleatória é somado ao tempo de permanência original, sorteado pela variável *TempoPermanencia*. Este tempo adicional somente é considerado quando o incentivo secundário estiver habilitado.

Como o incentivo ao aumento dos tempos de permanência dos clientes é uma característica peculiar do sistema simulado, não se encontrou nenhuma informação na literatura com relação à distribuição mais apropriada a ser utilizada. Dessa forma, procurando uma escolha que fosse

o mais neutra possível (não tendenciosa), optou-se por usar a mesma distribuição e os mesmos parâmetros da variável aleatória *TempoPermanencia*. Com isso, dobrou-se, na média, o tempo de permanência original.

Como a *lognormal* é uma distribuição de cauda longa, espera-se modelar, por consequência, aqueles poucos clientes que tendem a ficar muito mais tempo conectados no sistema por causa do incentivo secundário. Uma outra preocupação foi a modelagem daqueles clientes que nunca ficam conectados por mais tempo, independentemente do incentivo utilizado, por causa de limitações em suas conexões. Para isso, assumiu-se que os clientes pertencentes à classe de conexão discada sempre terão seus tempos extra de permanência iguais a zero.

Assim, implementou-se no simulador desenvolvido um gerador de tempos extra de permanência de clientes que utiliza a distribuição *lognormal* com os mesmos parâmetros usados para a variável aleatória *TempoPermanencia*. Em termos de arquitetura, o simulador foi projetado para permitir a definição dos parâmetros μ e σ desta *lognormal* separadamente para cada classe de conexão existente. Adicionalmente, todas as classes com $\mu = 0$ fazem com que os clientes pertencentes a elas nunca fiquem nenhum tempo extra de permanência. Dessa forma, para efetivar as configurações acima descritas, configurou-se valores iguais de μ e σ para todas as classes de conexão existentes, com exceção da discada, a qual foi configurada com $\mu = 0$.

Tempo de Reparo de Rupturas nas Árvores de Distribuição

A variável aleatória *TempoReparoRupturaArvore* representa o tempo que as árvores de distribuição levam para se reorganizarem, mediante a desconexão de algum nó interior. Em outras palavras, trata-se do tempo que um determinado cliente fica sem receber o descritor de uma árvore, quando algum outro cliente, inscrito nesta mesma árvore como nó interior e superior a ele, desconecta-se.

Em [43], usou-se tempos constantes de reparo de rupturas em árvores de 1, 5 e 10 segundos. Acredita-se que, na prática, tais intervalos de tempo sejam variáveis, pois dependem, pelo menos, das condições de rede para que as mensagens de reconexão trafeguem em tempo hábil. Como não se encontrou em outros trabalhos da literatura qual seria a melhor distribuição probabilística para modelar este caso, optou-se pela exponencial. Com relação à média da mesma, dada pelo parâmetro μ , utilizou-se o valor 5, que é o valor mediano das opções consideradas em [43] (1, 5 e 10 segundos).

Sendo assim, implementou-se no simulador desenvolvido um gerador de tempos de reparo de rupturas em árvores que utiliza a distribuição exponencial, com média igual a 5 segundos.

Congestionamento nos Enlaces de Entrada e de Saída

As variáveis aleatórias *CongestionamentoEntrada* e *CongestionamentoSaida* representam, respectivamente, a ocorrência de congestionamento nos enlaces de entrada e de saída dos

clientes, sendo discretas e tendo como espaço amostral 0 (não congestionado) e 1 (congestionado).

Foi implementado no simulador um gerador de números aleatórios discretos que pode ser instanciado com um determinado espaço amostral finito, tendo cada elemento do conjunto a sua respectiva probabilidade de ocorrência. Este gerador foi utilizado para estas duas variáveis aleatórias discretas, bem como para as demais a serem apresentadas ainda ao longo desta subseção.

Em [43], os autores analisaram o impacto que a perda de pacotes nos enlaces de saída dos clientes causava na distorção da imagem recebida. Três cenários foram considerados, sendo: (1) taxa de perda de 0.01 nos enlaces de saída de todos os clientes; (2) taxa de perda de 0.1 nos enlaces de saída de todos os clientes; (3) taxa de perda de 0.1 nos enlaces de saída de 10% de clientes escolhidos aleatoriamente. Não se modelou, de forma separada, a perda de pacotes nos enlaces de entrada dos clientes.

Tendo em vista esses valores da literatura e também o fato do simulador desenvolvido no presente trabalho modelar separadamente as perdas de pacotes nos enlaces de entrada e de saída dos clientes, adotou-se $P[\text{CongestionamentoEntrada} = 1] = P[\text{CongestionamentoSaida} = 1] = 0.05$. Este valor foi baseado no segundo cenário descrito acima, que é o mais pessimista dentre todos, dividindo-se a taxa de 0.1 por 2 (metade para cada um dos enlaces de entrada e de saída dos clientes). Sendo assim, cada cliente no sistema teve probabilidade de congestionamento em seus enlaces de entrada e de saída de 0.05.

Comportamento Cooperativo

A variável aleatória *ComportamentoCooperativo* representa a opção dos clientes entre assumir um comportamento cooperativo ou não, quando o mecanismo de incentivo proposto não estiver sendo utilizado (quando ele estiver, todos os clientes cooperam incondicionalmente, segundo a taxa de cooperação estipulada). Essa variável aleatória é discreta e tem como espaço amostral 0 (não coopera) e 1 (coopera). Os clientes sorteados com o valor 1 cooperam com toda a banda de saída que possuem, enquanto que os clientes sorteados com 0 não cooperam com absolutamente nada.

Em [1], estudos realizados na rede Gnutella [22] mostraram que apenas 30% dos usuários compartilham seus recursos e que a maioria das requisições são servidas por apenas 1% dos principais clientes. Apesar deste fato ter sido constatado no contexto de redes *peer-to-peer* de compartilhamento de arquivos, ele representa um perfil de comportamento altruísta em redes *peer-to-peer* em geral, que também pode ser utilizado no contexto de fluxos de mídia.

Como esta variável aleatória está diretamente ligada ao desempenho do sistema estudado quando nenhum mecanismo de incentivo é utilizado, variou-se outras percentagens de usuários cooperativos, além de 30%: 0%, 10%, 25%, 50%, 75% e 100%. A percentagem 0% representa o sistema degenerado à arquitetura cliente-servidor, ou seja, dado que nenhum cliente coopera,

apenas o servidor admite nós abaixo de si nas árvores de distribuição. A percentagem 100% representa um ambiente com altruísmo total, ou seja, o ponto máximo em termos de cooperação que uma rede *peer-to-peer* poderia atingir. Os demais valores foram distribuídos em incrementos de 25% dentro do intervalo, além do valor 10%, que também foi incluso.

Sendo assim, os seguintes valores foram utilizados para $P[\text{ComportamentoCooperativo} = 1]$: 0, 0.1, 0.25, 0.3, 0.5, 0.75 e 1. Uma nova instância do gerador de números aleatórios discretos foi utilizada com cada uma destas probabilidades de ocorrência do valor 1 (comportamento cooperativo), para a variável aleatória *ComportamentoCooperativo*.

Classe de Conexão

A variável aleatória *ClasseConexao* representa a classe de conexão dos clientes, sendo discreta e tendo como espaço amostral os identificadores das diferentes classes existentes no sistema. Essas classes são definidas no arquivo de configuração do simulador desenvolvido, com seus identificadores, probabilidades de ocorrência, bandas de entrada, bandas de saída e parâmetros μ e σ da distribuição *lognormal* utilizada para o sorteio do tempo extra de permanência dos clientes, quando o incentivo secundário do mecanismo proposto estiver sendo utilizado (conforme visto na subseção do tempo extra de permanência).

Em [55], as informações de acesso à Internet por banda larga no Brasil, no primeiro trimestre de 2005, indicam que 84.16% dos usuários utilizam a tecnologia ADSL, enquanto que 15.84% utilizam a tecnologia HFC (excluídas outras tecnologias menos comuns). Analisando os diferentes planos de assinatura para ADSL, oferecidos pelas principais operadoras instaladas no Brasil [52, 58, 56], pode-se identificar quatro planos comuns: (1) ADSL 150, com 150 Kbps de banda de entrada e 128 Kbps de banda de saída; (2) ADSL 350, com 350 Kbps de banda de entrada e 128 Kbps de banda de saída; (3) ADSL 600, com 600 Kbps de banda de entrada e 128 Kbps de banda de saída; e (4) ADSL 1M, com 1024 Kbps de banda de entrada e 128 Kbps de banda de saída. Da mesma forma, analisando os diferentes planos de assinatura para HFC, oferecidos pelas principais operadoras instaladas no Brasil [60, 2, 61], pode-se identificar quatro planos comuns: (1) HFC 150, com 150 Kbps de banda de entrada e de saída; (2) HFC 300, com 300 Kbps de banda de entrada e de saída; (3) HFC 600, com 600 Kbps de banda de entrada e de saída; e (4) HFC 1200, com 1200 Kbps de banda de entrada e 600 Kbps de banda de saída.

Duas informações não foram encontradas na literatura: a percentagem dos usuários de serviços de vídeo sob demanda que utilizam acesso discado e as distribuições para cada um dos planos identificados acima, tanto para ADSL quanto para HFC. Com relação à primeira informação não encontrada, assumiu-se que apenas 10% dos usuários utilizam acesso discado, dada a alta demanda por banda neste tipo de serviço e as limitações desta categoria de acesso. Conseqüentemente, considerou-se que 90% dos usuários de serviços de vídeo sob demanda são assinantes de acesso por banda larga. Com relação à segunda informação não encontrada, na ca-

tegoria ADSL assumiu-se 40% para os planos ADSL 150 e ADSL 350, 15% para o plano ADSL 600 e 5% para o plano ADSL 1M. A distribuição utilizada na categoria HFC foi a mesma: 40% para os planos HFC 150 e HFC 300, 15% para o plano HFC 600 e 5% para o plano HFC 1200.

A partir dos valores descritos acima, definiu-se então as classes de conexão, com suas respectivas probabilidades de ocorrência, da seguinte forma: 10% dos usuários utilizam acesso discado, enquanto que 90% utilizam banda larga. Destes 90%, 84.16% utilizam ADSL, o que equivale a 75.75% do total, enquanto que 15.84% utilizam HFC, o que equivale a 14.25% do total. Dos 75.75% de ADSL, 40% utilizam o plano ADSL 150, 40% utilizam o plano ADSL 350, 15% utilizam o plano ADSL 600 e 5% utilizam o plano ADSL 1M, o que equivale, respectivamente, às seguintes percentagens absolutas: 30.30%, 30.30%, 11.36% e 3.79%. Dos 15.84% que utilizam HFC, 40% utilizam o plano HFC 150, 40% utilizam o plano HFC 300, 15% utilizam o plano HFC 600 e 5% utilizam o plano HFC 1200, o que equivale, respectivamente, às seguintes percentagens absolutas: 5.7%, 5.7%, 2.14% e 0.71%. A Tabela 5.1 apresenta todas as classes de conexão criadas e suas respectivas propriedades, incluindo as probabilidades de ocorrência.

| Nome | Id | Probabilidade | B_E | B_S | μ | σ |
|----------------|----|---------------|-------|-------|-------|----------|
| <i>Dial up</i> | 1 | 0.1 | 56 | 56 | 0 | 0 |
| ADSL 150 | 2 | 0.303 | 150 | 128 | 5.74 | 2.01 |
| ADSL 350 | 3 | 0.303 | 350 | 128 | 5.74 | 2.01 |
| ADSL 600 | 4 | 0.1136 | 600 | 128 | 5.74 | 2.01 |
| ADSL 1M | 5 | 0.0379 | 1024 | 128 | 5.74 | 2.01 |
| HFC 150 | 6 | 0.057 | 150 | 150 | 5.74 | 2.01 |
| HFC 300 | 7 | 0.057 | 300 | 300 | 5.74 | 2.01 |
| HFC 600 | 8 | 0.0214 | 600 | 600 | 5.74 | 2.01 |
| HFC 1200 | 9 | 0.0071 | 1200 | 600 | 5.74 | 2.01 |

Tabela 5.1: Classes de conexão criadas no simulador desenvolvido e suas respectivas propriedades.

Uma nova instância do gerador de números aleatórios discretos foi utilizada com os identificadores das classes de conexão mencionadas acima e suas respectivas probabilidades de ocorrência, de forma a se modelar a heterogeneidade dos clientes.

Modificação da Taxa de Chegada

A variável aleatória *ModificacaoTaxaChegada* está relacionada à variável aleatória *TempoEntreChegadas*. Ela representa a opção de incremento ou decremento da taxa de chegada do

processo *Poisson* utilizado, de um período estacionário para outro. A variável aleatória *ModificacaoTaxaChegada* é discreta e tem como espaço amostral 0 (decrementa) e 1 (incrementa).

De forma a se estabelecer as mesmas probabilidades de incremento e de decremento, usou-se $P[\textit{ModificacaoTaxaChegada} = 0] = P[\textit{ModificacaoTaxaChegada} = 1] = 0.5$. Uma nova instância do gerador de números aleatórios discretos foi utilizada com estas probabilidades.

Utilização do Tempo Extra de Permanência

A variável aleatória *UtilizacaoTempoExtraPermanencia* está relacionada à variável aleatória *TempoExtraPermanencia*. Ela representa a opção dos clientes de classes com $\mu > 0$ de ficarem ou não mais tempo conectados no sistema, ou seja, somente os que forem sorteados por ela terão o tempo extra de permanência sorteado e considerado pela variável aleatória *TempoExtraPermanencia*, conforme explicado anteriormente. A variável aleatória *UtilizacaoTempoExtraPermanencia* é discreta e tem como espaço amostral 0 (cliente não fica mais tempo conectado) e 1 (cliente fica mais tempo conectado).

O propósito inicial desta variável aleatória era modelar a porcentagem de clientes que nunca ficam mais tempo conectados no sistema, independentemente do mecanismo de incentivo utilizado. Entretanto, criou-se a classe de conexão discada na qual os clientes nunca têm o tempo extra de permanência considerado ($\mu = 0$). Além disso, a própria dinâmica da segunda distribuição *lognormal* utilizada já modela, aproximadamente, os clientes que não ficam mais tempo no sistema, através dos sorteios de tempos curtos (que tendem a zero). Sendo assim, optou-se por utilizar apenas as seguintes probabilidades para a variável aleatória *UtilizacaoTempoExtraPermanencia*: $P[\textit{UtilizacaoTempoExtraPermanencia} = 1] = 1$, para as simulações nas quais o incentivo secundário está habilitado, e $P[\textit{UtilizacaoTempoExtraPermanencia} = 1] = 0$, caso contrário. Uma nova instância do gerador de números aleatórios discretos foi utilizada com estas probabilidades.

5.2.2 Parâmetros para Configuração do Simulador

Nesta subseção, os parâmetros criados para configuração do simulador são descritos. Todos eles podem ser definidos tanto em arquivo de configuração quanto em linha de comando. O simulador recebe como primeiro parâmetro da linha de comando o nome do arquivo de configuração utilizado. Os demais parâmetros da linha de comando (do segundo em diante) são opcionais e passados na forma “nome=valor”, sobrescrevendo os parâmetros de mesmo nome definidos no arquivo de configuração especificado. Com isso, criou-se um único arquivo de configuração base, utilizado para todas as simulações, tendo sido as variações de parâmetros específicas realizadas por linhas de comando, através de diversos *shell scripts* gerados para cada cenário.

Descrição de Configuração

O parâmetro *DescricaoConfiguracao* define um texto para permitir a identificação da configuração utilizada nos resultados de uma simulação. Dessa forma, toda vez que o simulador é executado, ele precede os resultados de saída com o texto deste parâmetro. Adicionalmente, todos os demais parâmetros definidos pelo arquivo de configuração ou pela linha de comando são também impressos, antes dos resultados de saída propriamente ditos. Exemplos: “Reputação, LI=1 LS=1 TR=2345”, “Sem Incentivo, 30% Cooperam” e “Escambo, TC=1”.

Tempo de Simulação

O parâmetro *TempoSimulacao* define o tempo a ser simulado, em segundos. Usou-se 86400 para cenários de 24 horas e 300 para cenários de 5 minutos.

Numero de Eventos no Buffer

O parâmetro *NumeroEventosBuffer* define o número máximo de eventos discretos de simulação que são armazenados em buffer antes de serem processados. Pode-se, através deste parâmetro, ajustar o balanço entre desempenho e consumo de memória do simulador. Usou-se 1024 para todas as simulações, tendo sido este um valor razoável encontrado.

Periodicidade de Processamento Periódico

O parâmetro *PeriodicidadeProcessamentoPeriodico* define a periodicidade, em segundos, na qual o simulador averigua se há processamentos periódicos pendentes. Um processamento periódico é aquele que se repete em intervalos de tempo fixos, como é o caso, por exemplo, das medições do número de clientes no sistema. Os processamentos periódicos implementados no simulador dividem-se em duas classes: os genéricos e os de congestionamento. Sendo assim, a cada *PeriodicidadeProcessamentoPeriodico* segundos, o simulador checa se as periodicidades de cada uma dessas classes foram atingidas, efetuando os respectivos processamentos. Utilizou-se, neste parâmetro, o valor 1 segundo para todas as simulações.

Periodicidade de Processamento Genérico

O parâmetro *PeriodicidadeProcessamentoGenerico* define a periodicidade, em segundos, da classe de processamentos periódicos genéricos. Dentre os processamentos existentes nesta classe, pode-se citar a medição da capacidade de admissão de novos clientes no sistema, a contagem do número de clientes e de cooperadores estáveis existentes, o cálculo da qualidade de fluxo recebida por cada cliente e as médias, desvios padrão e valores máximos e mínimos de todas essas variáveis.

De acordo com o que foi explicado para o parâmetro anterior, pode-se inferir que o valor do parâmetro *PeriodicidadeProcessamentoGenerico* deve ser um múltiplo do parâmetro *PeriodicidadeProcessamentoPeriodico*. Utilizou-se o valor 10 segundos para cenários de 24 horas e o valor 1 segundo para cenários de 5 minutos.

Tamanho do Fluxo

O parâmetro *TamanhoFluxo* define, em Kbps, o tamanho total do fluxo que é transmitido ao vivo no sistema, ou seja, a sua largura de banda. O tamanho dos descritores transmitidos em cada árvore de distribuição é calculado dividindo-se o valor deste parâmetro pelo número de árvores de distribuição utilizadas. Logo, o valor deste parâmetro deve incluir qualquer *overhead* gerado pela codificação em múltiplos descritores. Por exemplo, se o fluxo original tiver 120 Kbps, 16 árvores forem utilizadas e o tamanho de cada descritor após a codificação for 8 Kbps, então deve-se utilizar, neste parâmetro, o valor 128 Kbps, que já inclui os 8 Kbps adicionais de *overhead* gerado pela codificação (0.5 Kbps a mais por descritor).

Em [43, 45], usou-se um tamanho de fluxo de 160 Kbps, tendo sido as bandas de saída dos clientes também definidas com este mesmo valor. Como não se tratou a heterogeneidade dos clientes, todos eles apresentaram a mesma banda de saída, 160 Kbps. No simulador desenvolvido no presente trabalho, entretanto, a heterogeneidade dos clientes foi modelada, através das diferentes classes de conexão criadas. Como a maioria dos clientes (75.75%) pertence às classes ADSL, nas quais a banda de saída típica é 128 Kbps, este valor foi utilizado no parâmetro *TamanhoFluxo*, de forma a se manter o tamanho do fluxo equivalente às bandas de saída da maioria dos clientes, similar ao que foi feito em [43, 45]. O mesmo valor, 128 Kbps, foi utilizado para todas as simulações.

Número de Árvores de Distribuição

O parâmetro *NumeroArvoresDistribuicao* define o número de árvores de distribuição utilizadas no sistema. Baseado em [43, 45, 14], usou-se o valor 16 árvores para praticamente todas as simulações. Apenas para dois cenários usou-se o valor 32 árvores, de forma a se investigar os efeitos de diferentes profundidades das árvores nas rupturas contabilizadas.

Banda de Saída do Servidor

O parâmetro *BandaSaidaServidor* define, em Kbps, a banda de saída do servidor, de onde provêm todos descritores de fluxo, através das diferentes árvores de distribuição utilizadas. A banda definida por este parâmetro é igualmente distribuída para todas as árvores de distribuição. Sendo assim, por exemplo, se 16 árvores forem utilizadas e a banda do servidor for 2048 Kbps, então 128 Kbps será dedicado a cada árvore. Além disso, tem-se que o tamanho do descritor

transmitido em cada árvore equivale ao tamanho do fluxo original dividido pelo número de árvores utilizadas. Dessa forma, por exemplo, se o fluxo tiver 128 Kbps e 16 árvores forem utilizadas, então cada descritor terá 8 Kbps. Decorre que, com esses valores, o servidor admite 16 clientes abaixo de si em cada árvore. Generalizando, o servidor admite

$$\frac{BandaSaidaServidor/NumeroArvoresDistribuicao}{TamanhoFluxo/NumeroArvoresDistribuicao}$$

clientes abaixo de si em cada árvore de distribuição.

Adotou-se em [44, 43] a sugestão feita em [11] que cada cliente deve ser admitido como nó interior em uma única árvore, sendo nó folha nas $n - 1$ demais. O simulador implementado, por sua vez, também adotou essa sugestão. Sendo assim, suponha que em um sistema 16 árvores sejam utilizadas e que haja um perfeito balanceamento de nós interiores entre elas. Suponha agora que, em uma das árvores, os quinze primeiros clientes que se conectam no sistema sejam nela cadastrados como nós folhas (pior caso). Considerando a existência do balanceamento perfeito de nós interiores entre as árvores, certamente o décimo sexto cliente a chegar no sistema será cadastrado na referida árvore como nó interior. Mas, até então, esta árvore já teve que acomodar $NumeroArvoresDistribuicao - 1$ nós folhas iniciais com recursos apenas do nó raiz, que é o próprio servidor. Logo, tem-se que

$$\frac{BandaSaidaServidor}{TamanhoFluxo} \geq NumeroArvoresDistribuicao - 1,$$

ou seja,

$$BandaSaidaServidor \geq (NumeroArvoresDistribuicao - 1).TamanhoFluxo.$$

Portanto, considerando que o tamanho de fluxo utilizado foi 128 Kbps, usou-se, para cenários com 16 árvores de distribuição, $BandaSaidaServidor = 2048$. Para cenários com 32 árvores, usou-se $BandaSaidaServidor = 4096$.

Periodicidade de Processamento de Congestionamento

O parâmetro *PeriodicidadeProcessamentoCongestionamento* define a periodicidade, em segundos, da classe de processamentos periódicos de congestionamento. Dentre os processamentos existentes nesta classe, pode-se citar os sorteios de ocorrência de congestionamento para cada cliente, as promoções de clientes que se tornam cooperadores estáveis, os aumentos e diminuições de tráfego dos clientes com base no novo protocolo de adaptação de banda, os cálculos do mecanismo de incentivo utilizado, a desconexão de clientes que já estão há um certo tempo sem receber nenhum descritor, a atualização de t_* (tempo de conexão do cliente mais antigo que ainda não tenha se tornado cooperador estável) e o balanceamento dos nós interiores entre as árvores de distribuição (para equilibrar suas capacidades de admissão). Este último

procedimento, em particular, foi descrito em [43] como provável de ser necessário; na prática, verificou-se que ele é essencial, dado que é comum o esgotamento da capacidade de admissão de algumas árvores, enquanto que em outras sobra este tipo de recurso. Isso ocorre devido à desconexão de uma série de clientes que são nós interiores na mesma árvore. Sendo assim, o referido procedimento de balanceamento foi implementado no simulador.

O valor do parâmetro *PeriodicidadeProcessamentoCongestionamento* deve ser um múltiplo do parâmetro *PeriodicidadeProcessamentoPeriodico*, de acordo com a explicação dada para este último. Utilizou-se o valor 1 segundo, no parâmetro *PeriodicidadeProcessamentoCongestionamento*, para todas as simulações.

Semente da Variável Aleatória *CongestionamentoSaida*

O parâmetro *SementeCongestionamentoSaida* define a semente da variável aleatória *CongestionamentoSaida*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 50.

Probabilidade de Congestionamento nos Enlaces de Saída

O parâmetro *ProbabilidadeCongestionamentoSaida* define a probabilidade de ocorrência de congestionamento nos enlaces de saída dos clientes, ou seja, a ocorrência do valor 1 (congestionado) na variável aleatória *CongestionamentoSaida* ($P[\text{CongestionamentoSaida} = 1]$). Conforme explicado na Subseção 5.2.1, utilizou-se o valor 0.05, neste parâmetro, para todas as simulações.

Semente da Variável Aleatória *CongestionamentoEntrada*

O parâmetro *SementeCongestionamentoEntrada* define a semente da variável aleatória *CongestionamentoEntrada*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 100.

Probabilidade de Congestionamento nos Enlaces de Entrada

O parâmetro *ProbabilidadeCongestionamentoEntrada* define a probabilidade de ocorrência de congestionamento nos enlaces de entrada dos clientes, ou seja, a ocorrência do valor 1 (congestionado) na variável aleatória *CongestionamentoEntrada* ($P[\text{CongestionamentoEntrada} = 1]$). Conforme explicado na Subseção 5.2.1, utilizou-se o valor 0.05, neste parâmetro, para todas as simulações.

Período para Adição de Tráfego

O parâmetro *PeriodoAdicaoTrafego* define o período de tempo, em segundos, que se espera que um determinado cliente fique sem nenhuma ocorrência de congestionamento para então se realizar uma tentativa de aumento de tráfego com ele. Há um tratamento independente para os enlaces de entrada e de saída, ou seja, cada um deles possui o seu próprio contador de tempo, para cada cliente. O período de tempo definido por este parâmetro é o mesmo a que o novo protocolo de adaptação de banda, proposto na Seção 4.2, se refere. Usou-se o valor 5 segundos para todas as simulações.

Período sem Tráfego para Desconexão

O parâmetro *PeriodoSemTrafegoDesconexao* define o período de tempo, em segundos, que um determinado cliente pode permanecer conectado no sistema sem receber nenhum descritor, ou seja, sem estar cadastrado em nenhuma árvore de distribuição. Quando o cliente se conecta, se ele não puder ser admitido em nenhuma das árvores de distribuição, em virtude de uma sobrecarga do sistema, então ele é imediatamente desconectado. Entretanto, se ele for admitido em pelo menos uma árvore, mas posteriormente sair dela, em decorrência, por exemplo, de eventos de congestionamento, então ele poderá ficar, no máximo, *PeriodoSemTrafegoDesconexao* segundos nessa situação, tempo após o qual ele será desconectado.

Há uma relação de funcionalidade entre este parâmetro e o anterior. Se $PeriodoAdicaoTrafego < PeriodoSemTrafegoDesconexao$, poderia se pensar que o tráfego de um determinado cliente, não recebendo nenhum descritor, sempre seria aumentado antes de *PeriodoSemTrafegoDesconexao* segundos, considerando que não ocorre congestionamento sem nenhum tráfego (com zero descritores). Não obstante, o sistema pode estar sobrecarregado e constantemente rejeitando as tentativas de aumento de tráfego daquele cliente, o que é o caso mais comum. Por outro lado, se $PeriodoAdicaoTrafego > PeriodoSemTrafegoDesconexao$, então um determinado cliente somente pode aguardar *PeriodoAdicaoTrafego* segundos sem congestionamento para aumentar o seu tráfego se ele estiver recebendo pelo menos um descritor, caso contrário ele seria desconectado previamente após *PeriodoSemTrafegoDesconexao* segundos.

Utilizou-se, no parâmetro *PeriodoSemTrafegoDesconexao*, o valor 15 segundos para todas as simulações. Isto implicou na funcionalidade descrita para $PeriodoAdicaoTrafego < PeriodoSemTrafegoDesconexao$.

Nível de Log

O parâmetro *NivelLog* define o nível de detalhamento dos *traces* gerados pelo simulador. Para se implementar esta funcionalidade, desenvolveu-se uma infra-estrutura própria de log. Seis

níveis foram criados: (0) desligado; (1) informações críticas; (2) informações de alerta; (3) informações comuns; (4) informações de depuração; e (5) informações de depuração extensivas. Os níveis maiores incluem todas as informações dos níveis menores.

Os níveis 4 e 5 foram utilizados na fase de testes e depuração do simulador. Os níveis 0, 1 e 2 foram utilizados nos testes de estresse realizados, de forma a se otimizar as execuções através da diminuição da saída gerada. O nível 3 passou a ser adotado como padrão após o término do desenvolvimento, tendo sido utilizado para todas as simulações.

Arquivo de Log

O parâmetro *ArquivoLog* define o nome do arquivo de log gerado pelo simulador, de acordo com a configuração descrita para o parâmetro anterior. Caso nenhum nome seja especificado, então os *traces* são redirecionados para a saída padrão (*stdout*) do simulador. Usou-se, neste parâmetro, nomes intuitivos para se identificar cada experimento realizado, dentro dos diversos cenários considerados.

Semente da Variável Aleatória *TempoEntreChegadas*

O parâmetro *SementeTempoEntreChegadas* define a semente da variável aleatória *TempoEntreChegadas*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 150.

Período Estacionário da Taxa de Chegada

O parâmetro *PeriodoEstacionarioTaxaChegada* define a duração, em segundos, dos intervalos estacionários da variável aleatória *TempoEntreChegadas*. Tratam-se dos intervalos nos quais a taxa de chegada do processo *Poisson* utilizado não varia. Conforme foi explicado na Subseção 5.2.1, utilizou-se o período de 15 minutos, que equivale ao valor 900 segundos. Este valor foi usado, neste parâmetro, para todas as simulações.

Período para Reinicialização da Taxa de Chegada

O parâmetro *PeriodoReinicializacaoTaxaChegada* está relacionado à variável aleatória *TempoEntreChegadas*. Ele define o tempo decorrido esperado, em segundos, após o qual se realiza uma reinicialização da taxa de chegada do processo *Poisson* utilizado, de volta ao seu valor inicial. Utilizou-se o valor 86400 segundos (24 horas) para todas as simulações.

Limite Inferior da Taxa de Chegada

O parâmetro *LimiteInferiorTaxaChegada* também está relacionado à variável aleatória *TempoEntreChegadas*. Ele define, em chegadas por segundo, o menor valor que a taxa de chegada do processo *Poisson* utilizado pode atingir, através das variações realizadas de um período estacionário para outro. Conforme explicado na Subseção 5.2.1, utilizou-se o valor 0.083 chegadas por segundo (5 chegadas por minuto) para as situações normais de tráfego (sem *flash crowd*) e o valor 80 chegadas por segundo para as situações com *flash crowd*.

Limite Superior da Taxa de Chegada

O parâmetro *LimiteSuperiorTaxaChegada* também está relacionado à variável aleatória *TempoEntreChegadas*. Ele define, em chegadas por segundo, o maior valor que a taxa de chegada do processo *Poisson* utilizado pode atingir, através das variações realizadas de um período estacionário para outro. Conforme explicado na Subseção 5.2.1, utilizou-se o valor 0.333 chegadas por segundo (20 chegadas por minuto) para as situações normais de tráfego (sem *flash crowd*) e o valor 80 chegadas por segundo para as situações com *flash crowd*.

Valor de Modificação da Taxa de Chegada

O parâmetro *ValorModificacaoTaxaChegada* também está relacionado à variável aleatória *TempoEntreChegadas*. Ele define, em chegadas por segundo, o valor a ser utilizado para incrementar ou decrementar a taxa de chegada do processo *Poisson* utilizado, a cada período estacionário decorrido. Conforme explicado na Subseção 5.2.1, utilizou-se o valor 0.0416 chegadas por segundo para todas as simulações.

Semente da Variável Aleatória *ModificacaoTaxaChegada*

O parâmetro *SementeModificacaoTaxaChegada* define a semente da variável aleatória *ModificacaoTaxaChegada*, que está relacionada à variável aleatória *TempoEntreChegadas*, ambas descritas na Subseção 5.2.1. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 200.

Probabilidade de Incremento da Taxa de Chegada

O parâmetro *ProbabilidadeIncrementoTaxaChegada* define, para a variável aleatória *ModificacaoTaxaChegada*, a probabilidade de incremento da taxa de chegada do processo *Poisson* utilizado, a cada período estacionário decorrido. Conforme explicado na Subseção 5.2.1, utilizou-se o valor 0.5 para todas as simulações.

Semente da Variável Aleatória *TempoPermanencia*

O parâmetro *SementeTempoPermanencia* define a semente da variável aleatória *TempoPermanencia*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 250.

Parâmetro μ da Variável Aleatória *TempoPermanencia*

O parâmetro *ParametroMiTempoPermanencia* está relacionado à variável aleatória *TempoPermanencia*. Ele define o parâmetro μ da distribuição *lognormal* utilizada para modelar os tempos de permanência dos clientes no sistema. Conforme já explicado, usou-se o valor 5.74, neste parâmetro, para praticamente todas as simulações. A única exceção foi um cenário específico para o qual se considerou tempos de permanência mais curtos, de forma a se estudar as rupturas em árvores de distribuição. Esta variação particular, para a qual o parâmetro μ assumiu o valor 2.36, será explicada no Capítulo 6.

Parâmetro σ da Variável Aleatória *TempoPermanencia*

O parâmetro *ParametroSigmaTempoPermanencia* também está relacionado à variável aleatória *TempoPermanencia*. Ele define o parâmetro σ da distribuição *lognormal* utilizada para modelar os tempos de permanência dos clientes no sistema. Utilizou-se o valor 2.01, neste parâmetro, para praticamente todas as simulações. A única exceção, novamente, foi o cenário específico mencionado, para o qual o parâmetro σ assumiu o valor 1.44.

Semente da Variável Aleatória *ClasseConexao*

O parâmetro *SementeClasseConexao* define a semente da variável aleatória *ClasseConexao*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 300.

Definição de Classe de Conexão

O parâmetro *DefinicaoClasseConexao* define uma nova classe de conexão a ser utilizada pelo simulador. Trata-se de um parâmetro multivalorado, com os seguintes campos separados por vírgula: (1) identificador da classe; (2) probabilidade de ocorrência; (3) banda de entrada dos clientes; (4) banda de saída dos mesmos; (5) parâmetro μ da distribuição *lognormal* utilizada para se modelar os tempos extra de permanência dos clientes; (6) parâmetro σ da mesma; e (7) descrição da classe. Exemplos: “*DefinicaoClasseConexao* = 1, 0.1, 56, 56, 0, 0, *Dial_Up*” e “*DefinicaoClasseConexao* = 2, 0.303, 150, 128, 5.74, 2.01, *ADSL_150*”.

Este parâmetro, além de multivalorado, deve ser utilizado múltiplas vezes, tantas quantas forem as classes de conexão a serem criadas. Há duas ressalvas a serem respeitadas: cada classe deve ter um identificador único e o somatório das probabilidades de ocorrência de todas as classes criadas deve ser exatamente igual a 1. Utilizou-se este parâmetro nove vezes para se criar as nove classes de conexão descritas na Tabela 5.1, igualmente para todas as simulações.

Semente da Variável Aleatória *TempoExtraPermanencia*

O parâmetro *SementeTempoExtraPermanencia* define a semente da variável aleatória *TempoExtraPermanencia*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 350.

Semente da Variável Aleatória *UtilizacaoTempoExtraPermanencia*

O parâmetro *SementeUtilizacaoTempoExtraPermanencia* define a semente da variável aleatória *UtilizacaoTempoExtraPermanencia*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 400.

Probabilidade de Utilização do Tempo Extra de Permanência

O parâmetro *ProbabilidadeUtilizacaoTempoExtraPermanencia* define, para a variável aleatória *UtilizacaoTempoExtraPermanencia*, a probabilidade dos clientes com $\mu > 0$ terem o tempo extra de permanência considerado. Conforme explicado na Subseção 5.2.1, utilizou-se o valor 1 para as simulações nas quais o incentivo secundário estava habilitado, e o valor 0 para as simulações nas quais o incentivo secundário estava desabilitado.

Semente da Variável Aleatória *ComportamentoCooperativo*

O parâmetro *SementeComportamentoCooperativo* define a semente da variável aleatória *ComportamentoCooperativo*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 450.

Probabilidade de Comportamento Cooperativo

O parâmetro *ProbabilidadeComportamentoCooperativo* define, para a variável aleatória *ComportamentoCooperativo*, a probabilidade dos clientes assumirem um comportamento cooperativo. Este parâmetro somente foi diferente de 1 para as simulações nas quais o mecanismo de incentivo proposto não estava habilitado pois, caso contrário, todos os clientes cooperariam incondicionalmente. Quando ele foi variado, utilizou-se os seguintes valores de probabilidade: 0, 0.1, 0.25, 0.3, 0.5, 0.75 e 1.

Taxa de Cooperação

O parâmetro *TaxaCooperacao* define o esquema de incentivo utilizado pelo simulador. Ele pode assumir três possibilidades: o valor -1, que faz com que nenhum incentivo seja habilitado; o valor 0, que habilita os incentivos primário e secundário do mecanismo proposto, ou seja, taxas de cooperação variáveis entre os clientes; e valores maiores que 0, que habilitam somente o incentivo primário com a taxa de cooperação no valor estipulado, ou seja, a mesma taxa constante para todos os clientes, definida por este parâmetro. Quando se utiliza o valor 0, deve-se também especificar os parâmetros *LimiteInferiorTaxaCooperacao*, *LimiteSuperiorTaxaCooperacao* e *TempoRequeridoCooperadorEstavel*, descritos em seguida.

Nas análises comparativas apresentadas no Capítulo 6, utilizou-se os três casos descritos acima. Sem a utilização de nenhum incentivo (*TaxaCooperacao* = -1), definiu-se *ProbabilidadeUtilizacaoTempoExtraPermanencia* = 0 e variou-se o parâmetro *ProbabilidadeComportamentoCooperativo*. Com a utilização apenas do incentivo primário (*TaxaCooperacao* > 0), definiu-se *ProbabilidadeComportamentoCooperativo* = 1, *ProbabilidadeUtilizacaoTempoExtraPermanencia* = 0 e variou-se o parâmetro *TaxaCooperacao*. Com a utilização dos incentivos primário e secundário (*TaxaCooperacao* = 0), definiu-se *ProbabilidadeComportamentoCooperativo* = 1, *ProbabilidadeUtilizacaoTempoExtraPermanencia* = 1 e variou-se os parâmetros *LimiteInferiorTaxaCooperacao*, *LimiteSuperiorTaxaCooperacao* e *TempoRequeridoCooperadorEstavel*. Informações adicionais acerca das variações realizadas, com os respectivos valores utilizados para cada parâmetro, serão fornecidas no Capítulo 6.

Limite Inferior da Taxa de Cooperação

O parâmetro *LimiteInferiorTaxaCooperacao* define o limite inferior da taxa de cooperação utilizado pelo mecanismo de incentivo proposto. Ele corresponde à variável LI_{TC} descrita na Seção 4.1. Esta variável precisa ser determinada de forma a se calibrar o funcionamento do mecanismo proposto. A descrição de como este procedimento é realizado, bem como a apresentação dos valores selecionados para as simulações serão feitas no Capítulo 6.

Limite Superior da Taxa de Cooperação

O parâmetro *LimiteSuperiorTaxaCooperacao* define o limite superior da taxa de cooperação utilizado pelo mecanismo de incentivo proposto. Ele corresponde à variável LS_{TC} descrita na Seção 4.1. Esta variável também precisa ser determinada de forma a se calibrar o funcionamento do mecanismo proposto. A descrição de como este procedimento é realizado, bem como a apresentação dos valores selecionados para as simulações serão feitas no Capítulo 6.

Tempo Requerido de Cooperador Estável

O parâmetro *TempoRequeridoCooperadorEstavel* define, em segundos, o tempo requerido de cooperador estável utilizado pelo mecanismo de incentivo proposto. Ele corresponde à variável *TR* descrita na Seção 4.1. Esta variável também precisa ser determinada de forma a se calibrar o funcionamento do mecanismo proposto. A descrição de como este procedimento é realizado, bem como a apresentação dos valores selecionados para as simulações serão feitas no Capítulo 6.

Semente da Variável Aleatória *TempoReparoRupturaArvore*

O parâmetro *SementeTempoReparoRupturaArvore* define a semente da variável aleatória *TempoReparoRupturaArvore*. Como se executou simulações de uma única rodada, utilizou-se, neste parâmetro, somente o valor 500.

Parâmetro μ da Variável Aleatória *TempoReparoRupturaArvore*

O parâmetro *ParametroMiTempoReparoRupturaArvore* está relacionado à variável aleatória *TempoReparoRupturaArvore*. Ele define o parâmetro μ da distribuição exponencial utilizada para modelar os tempos de reparo de rupturas em árvores de distribuição. Conforme já mencionado, utilizou-se o valor 5, neste parâmetro, para todas as simulações.

Diretório de Saída

O parâmetro *DiretorioSaida* define o diretório no qual são gerados os arquivos de saída do simulador, a cada execução realizada. Caso já existam arquivos neste diretório com os mesmos nomes dos a serem gerados, então realiza-se uma concatenação dos novos dados nos arquivos já existentes, de forma a não se perder as informações anteriores. Utilizou-se diferentes nomes de diretório para cada experimento realizado, de maneira a se organizar os arquivos gerados.

Resultados da Métrica Qualidade Média de Fluxo Recebida

O parâmetro *ResultadosQualidadeMediaFluxoRecebida* define o nome do arquivo gerado com as informações da métrica qualidade média de fluxo recebida, descrita na Seção 6.1. Utilizou-se o nome “ResultadosQualidadeMédiaFluxoRecebida.txt” para todas as simulações, já que o parâmetro *DiretorioSaida* foi variado.

Resultados da Métrica Capacidade de Admissão do Sistema

O parâmetro *ResultadosCapacidadeAdmissaoSistema* define o nome do arquivo gerado com as informações da métrica capacidade de admissão do sistema, também descrita na Se-

ção 6.1. Utilizou-se o nome “ResultadosCapacidadeAdmissãoSistema.txt” para todas as simulações, já que o parâmetro *DiretorioSaida* foi variado.

Resultados da Métrica Probabilidade de Bloqueio

O parâmetro *ResultadosProbabilidadeBloqueio* define o nome do arquivo gerado com as informações da métrica probabilidade de bloqueio, também descrita na Seção 6.1. Utilizou-se o nome “ResultadosProbabilidadeBloqueio.txt” para todas as simulações, já que o parâmetro *DiretorioSaida* foi variado.

Resultados da Métrica Taxa de Rupturas em Árvores

O parâmetro *ResultadosTaxaRupturasArvores* define o nome do arquivo gerado com as informações da métrica taxa de rupturas em árvores, também descrita na Seção 6.1. Utilizou-se o nome “ResultadosTaxaRupturasÁrvores.txt” para todas as simulações, já que o parâmetro *DiretorioSaida* foi variado.

5.3 Ferramentas Utilizadas

Usou-se a linguagem de programação C++, com a biblioteca *Standard Template Library* [51, 35]. Esta biblioteca foi empregada principalmente para as estruturas de dados básicas, como vetores, listas, conjuntos e mapas. As estruturas de árvores utilizadas tiveram implementação própria, a partir das estruturas básicas. Criou-se diferentes níveis de abstração, como “cliente”, “nó”, “árvore” e “floresta”. Apesar do simulador ter sido desenvolvido no paradigma de orientação a objetos, tomou-se um cuidado especial com o uso de polimorfismo, dado os requisitos de desempenho deste tipo de aplicação e a sobrecarga gerada pelo uso de métodos “virtuais”, que demandam ligações dinâmicas em tempo de execução.

Usou-se a biblioteca *GNU Scientific Library* [23] para a geração de distribuições probabilísticas. O desenvolvimento do código e dos *shell scripts* foi realizado, em sua grande maioria, no editor de texto *NEdit* [36]. Em alguns momentos usou-se, também, para desenvolvimento de código, a plataforma *Eclipse* [18] com o *plugin CDT* [13]. Adicionalmente, usou-se o sistema *CVS* [17] para versionamento de código e de documentos, o que garantiu segurança e manutenção de histórico em todo o processo de desenvolvimento realizado.

Com relação aos *scripts* desenvolvidos, utilizou-se *Bash* [5]. Criou-se diferentes *scripts* para as execuções das simulações, considerando as diferentes variações de cenários e experimentos. Criou-se também diferentes *scripts* para o pós-processamento dos *traces* gerados pelo simulador, incluindo a geração automática dos gráficos. Usou-se, como ferramenta de geração de gráficos, a aplicação *Gnuplot* [21], tendo sido todos os parâmetros de configuração dos gráficos passados por linha de comando pelos *scripts* desenvolvidos.

5.4 Considerações Finais

Neste capítulo, apresentou-se o simulador desenvolvido como forma de se avaliar a eficácia do mecanismo de incentivo proposto. Trata-se de um simulador de eventos discretos para redes *peer-to-peer* de fluxo de mídia ao vivo, que constitui uma contribuição adicional do presente trabalho. Este simulador foi desenvolvido usando-se o paradigma de orientação a objetos, o que facilita a sua modificação para ser reutilizado para outros tipos de sistemas relacionados. Por exemplo, no futuro pode-se facilmente trocar o mecanismo de incentivo implementado por um outro que se queira avaliar, reaproveitando todo o *framework* de simulação de eventos discretos e os controladores que implementam as operações do tipo de sistema considerado.

O simulador utiliza a proposta descrita em [43, 44], na qual múltiplas árvores de distribuição e codificação em múltiplos descritores são utilizadas como forma de se prover robustez e adaptabilidade à heterogeneidade dos clientes. O protocolo de admissão *top-down* também proposto em [43, 44] e o novo protocolo de adaptação de banda, descrito na Seção 4.2, foram implementados.

Apresentou-se, inicialmente, a arquitetura do simulador. Em seguida, descreveu-se as variáveis aleatórias utilizadas, que correspondem às distribuições probabilísticas adotadas. Na seqüência, discutiu-se todos os parâmetros de simulação, que correspondem às entradas do simulador. Por fim, as ferramentas utilizadas no desenvolvimento foram mencionadas.

Capítulo 6

Avaliação do Mecanismo Proposto

Neste capítulo, apresenta-se a avaliação do mecanismo de incentivo proposto no Capítulo 4, realizada através de simulações de eventos discretos. Pretende-se investigar os possíveis benefícios obtidos tanto com o incentivo primário quanto com o incentivo secundário do mecanismo proposto. Tem-se como objetivo descobrir se a utilização desses incentivos é interessante no contexto do sistema estudado e sob quais condições. Para o caso do incentivo secundário, investiga-se para quais parâmetros ele apresenta seu melhor desempenho; para o caso do incentivo primário, investiga-se para qual taxa de cooperação ele é mais eficiente.

O restante deste capítulo está organizado da seguinte forma. Na Seção 6.1, descreve-se as métricas coletadas e analisadas nas simulações executadas. Na Seção 6.2, realiza-se um estudo sobre a determinação dos parâmetros *LimiteInferiorTaxaCooperacao*, *LimiteSuperiorTaxaCooperacao* e *TempoRequeridoCooperadorEstavel*, responsáveis pelo funcionamento do incentivo secundário. Após a seleção dos melhores valores para esses parâmetros, o sistema será avaliado, na Seção 6.3, tanto em condições normais de tráfego quanto em situações de *flash crowd*. O desempenho do mecanismo será comparado para os seguintes cenários: (1) sem incentivo; (2) somente com o incentivo primário; e (3) com os incentivos primário e secundário. Finalmente, na Seção 6.4, conclusões parciais são apresentadas.

6.1 Métricas Analisadas

Nesta seção, descreve-se as métricas coletadas e analisadas nas simulações executadas. Através delas, avalia-se a eficácia do mecanismo de incentivo proposto em tornar o sistema escalável, maximizar a qualidade recebida e amenizar os problemas decorrentes das rupturas em árvores de distribuição.

6.1.1 Qualidade Média de Fluxo Recebida

A qualidade de fluxo recebida por um cliente em um determinado instante de tempo corresponde ao número de descritores recebidos ou, em outras palavras, em quantas árvores de distribuição distintas ele está inscrito. A qualidade média de fluxo recebida por este cliente, durante o período que ele permaneceu conectado no sistema, corresponde à média aritmética simples das referidas medições instantâneas, feitas periodicamente enquanto ele estiver conectado no sistema. A métrica qualidade média de fluxo recebida representa a média aritmética simples destas qualidades médias de todos os clientes que forem servidos durante o período de simulação.

Por exemplo, suponha que um determinado cliente tenha permanecido no sistema por 100 segundos, apresentando as seguintes medições instantâneas de qualidade de fluxo recebida (com intervalos de tempo de 10 segundos entre elas): 5, 6, 7, 7, 8, 6, 5, 5, 7 e 8. Logo, a qualidade média de fluxo recebida por este cliente durante a sua permanência no sistema foi 6.4. Suponha agora que, além deste cliente, outros nove também tenham passado pelo sistema durante o período de simulação, apresentando as seguintes qualidades médias de fluxo recebidas: 6.1, 6.2, 6.3, 6.5, 6.6, 6.7, 6.8, 6.9 e 7. Sendo assim, a métrica qualidade média de fluxo recebida marcaria, para o período de simulação executado, o valor 6.55.

6.1.2 Capacidade de Admissão do Sistema

A capacidade de admissão do sistema em um determinado instante de tempo corresponde ao somatório dos números de novos clientes que podem ser admitidos em cada uma das árvores de distribuição existentes. Calcula-se esta medida instantânea do sistema periodicamente, de acordo com os intervalos de processamentos periódicos, descritos no Capítulo 5. A métrica capacidade de admissão do sistema representa a média aritmética simples destas medidas instantâneas e periódicas, coletadas durante o período de simulação.

Para se determinar o número de novos clientes que podem ser admitidos em uma dada árvore de distribuição, calcula-se quantos novos nós podem ser acomodados abaixo de cada nó interior daquela árvore, levando-se em conta a banda de saída cooperada do admissor, a utilização da mesma (quantos nós filhos já possui) e o tamanho do descritor que trafega naquela árvore. Um nó interior somente pode admitir um outro nó se a soma de sua banda de saída utilizada com o tamanho do descritor a ser servido não ultrapassar a sua banda de saída total disponibilizada para cooperação.

Por exemplo, suponha que o sistema tenha quatro árvores de distribuição e que, em um determinado instante de tempo, cada uma delas possa admitir 100, 200, 300 e 400 novos clientes. Logo, a capacidade de admissão do sistema naquele instante de tempo é 1000. Suponha agora que o tempo de simulação executado seja 100 segundos, e que a referida medição instantânea e periódica ocorra a cada 10 segundos, apresentando os seguintes valores: 1000, 1010, 1020,

1030, 1040, 1050, 1060, 1070, 1080 e 1090. Sendo assim, a métrica capacidade de admissão do sistema marcaria, para o período de simulação executado, o valor 1045.

6.1.3 Probabilidade de Bloqueio do Sistema

Um cliente, ao se conectar inicialmente no sistema, pode ser admitido em todas as árvores de distribuição, em apenas um subconjunto delas ou em nenhuma, dependendo de sua banda de entrada, de sua banda de saída disponibilizada para fins de cooperação e, mais importantemente, da capacidade de admissão do sistema. Quando ele for admitido em pelo menos uma das árvores existentes, entende-se que ele não foi bloqueado, apesar de passar a receber um fluxo na pior qualidade possível. Por outro lado, quando ele não for admitido em nenhuma das árvores existentes, tem-se que ele não receberá nenhum fluxo, devendo, portanto, ser desconectado. Neste caso, entende-se que ocorreu um bloqueio do referido cliente, causado pelo esgotamento da capacidade de admissão do sistema.

A métrica probabilidade de bloqueio do sistema representa a razão entre o número de clientes bloqueados e o número total de chegadas no sistema. Assim, ela indica a probabilidade de um cliente chegar no sistema e não ser admitido em nenhuma árvore de distribuição, sendo, conseqüentemente, desconectado.

Por exemplo, suponha que, durante um determinado período de simulação, 1000 clientes tenham chegado para se conectar no sistema. Suponha agora que, destes 1000 clientes, 100 deles tenham sido bloqueados. Sendo assim, a métrica probabilidade de bloqueio do sistema marcaria, para o período de simulação executado, o valor 0.1.

6.1.4 Taxa de Rupturas em Árvores

A taxa de rupturas em árvores sofrida por um cliente durante a sua permanência no sistema é calculada através da razão entre o número total de rupturas que o afetaram e o seu tempo de permanência. A métrica taxa de rupturas em árvores representa a média aritmética simples das taxas individuais de todos os clientes que forem servidos durante o período de simulação.

Para se determinar o número total de rupturas que tenham afetado um determinado cliente, mantém-se um registro dos tempos de início e término de cada ruptura ocorrida para cada cliente em cada árvore de distribuição. Dessa forma, toda vez que ocorre uma saída de nó interior de uma árvore, processa-se todos os nós que estão abaixo dele naquela árvore, até se atingir os nós folhas. Para cada nó processado, verifica-se se ele já não está passando por uma ruptura naquela árvore. Em caso afirmativo, ao invés de se contabilizar uma ruptura a mais para aquele cliente, apenas se atualiza o tempo de término da ruptura já existente, baseado no tempo de término da ruptura mais recente. Em caso negativo, contabiliza-se aquela nova ruptura para o referido cliente, armazenando-se os tempos de início e término da mesma para a análise de futuras ocorrências na mesma árvore.

Por exemplo, suponha que um determinado cliente tenha permanecido no sistema por 100 segundos, sendo afetado por um total de 300 rupturas. Logo, a taxa de rupturas em árvores sofrida por este cliente durante a sua permanência no sistema foi 3. Suponha agora que, além deste cliente, outros nove também tenham passado pelo sistema durante o período de simulação, sofrendo as seguintes taxas de rupturas em árvores: 2.5, 2.6, 2.7, 2.8, 2.9, 3.1, 3.2, 3.3 e 3.4. Sendo assim, a métrica taxa de rupturas em árvores marcaria, para o período de simulação executado, o valor 2.95.

6.2 Determinação dos Parâmetros do Mecanismo Proposto

Nesta seção, apresenta-se um estudo sobre a determinação dos parâmetros do incentivo secundário do mecanismo proposto. Esta determinação foi realizada experimentalmente e teve como objetivo proporcionar um entendimento inicial do algoritmo proposto. Pretende-se, futuramente, implementar uma determinação dinâmica dos referidos parâmetros, de forma que o sistema se ajuste automaticamente a diferentes situações.

Naturalmente, em todos os experimentos considerados nesta seção, o mecanismo é analisado de forma completa, ou seja, com seus incentivos primário e secundário. A análise comparativa de experimentos que envolvem apenas o incentivo primário ou, ainda, a não utilização de nenhum mecanismo será realizada após este ajuste de parâmetros.

São considerados, primeiramente, os parâmetros *LimiteInferiorTaxaCooperacao* e *LimiteSuperiorTaxaCooperacao*. Em seguida, o parâmetro *TempoRequeridoCooperadorEstavel* é avaliado.

6.2.1 Limites Inferior e Superior da Taxa de Cooperação

Para o ajuste dos parâmetros *LimiteInferiorTaxaCooperacao* e *LimiteSuperiorTaxaCooperacao*, detalha-se, primeiramente, os experimentos realizados, com suas configurações e variações específicas. Em seguida, a análise das métricas consideradas é realizada.

Experimentos Realizados

Para se determinar os parâmetros *LimiteInferiorTaxaCooperacao* e *LimiteSuperiorTaxaCooperacao*, o cenário *5mLIS* foi criado. Nele, o sistema é submetido a uma situação de *flash crowd* intensa durante os cinco primeiros minutos de simulação, na qual o número de clientes admitidos cresce rapidamente. O objetivo é encontrar os melhores valores para os dois parâmetros em questão de forma que o sistema torne-se escalável com o aumento do número de clientes e, principalmente, que as qualidades de fluxo recebidas sejam maximizadas. As métricas qualidade média de fluxo recebida e capacidade de admissão do sistema são investigadas.

A determinação do tempo de simulação em cinco minutos foi realizada levando-se em consideração dois fatores. Primeiramente, dado que altas taxas de chegada foram utilizadas, justamente com o intuito de se elevar o número de clientes conectados rapidamente, o tempo de processamento demandado para simulações com períodos maiores inviabilizaria a análise de diversas combinações, necessárias para a melhor aproximação dos parâmetros a serem encontrados. Em segundo lugar, um período de simulação curto torna irrelevante a influência do parâmetro *TempoRequeridoCooperadorEstavel*, que será discutido na Subseção 6.2.2. Conforme será visto, este parâmetro apresenta dependência com os dois aqui estudados, devendo ser determinado somente após a definição destes.

Os valores utilizados para *LimiteInferiorTaxaCooperacao* e *LimiteSuperiorTaxaCooperacao*, nas diferentes combinações do cenário *5mLIS*, estão no intervalo $[0.2, 2.0]$, ou seja, banda cooperada equivalente a valores entre 20% e 200% da banda consumida. Outros valores foram inicialmente testados, porém nota-se que o desempenho das métricas analisadas fica muito insatisfatório fora desse intervalo. Em particular, constata-se que quanto mais distantes de 1 (banda cooperada equivalente a 100% da banda consumida) são os valores considerados, piores são os resultados obtidos. Dada a necessidade de se discretizar os diferentes valores utilizados e também as restrições de número de combinações possíveis consideradas, optou-se por tomar elementos do referido intervalo em incrementos de 0.2. As combinações escolhidas podem ser vistas na Tabela 6.1.

| e | LI | LS | e | LI | LS | e | LI | LS | e | LI | LS |
|----|-----|-----|----|-----|-----|----|-----|-----|----|-----|-----|
| 1 | 0.2 | 0.2 | 15 | 0.4 | 1.2 | 29 | 0.8 | 1 | 43 | 1.2 | 1.6 |
| 2 | 0.2 | 0.4 | 16 | 0.4 | 1.4 | 30 | 0.8 | 1.2 | 44 | 1.2 | 1.8 |
| 3 | 0.2 | 0.6 | 17 | 0.4 | 1.6 | 31 | 0.8 | 1.4 | 45 | 1.2 | 2 |
| 4 | 0.2 | 0.8 | 18 | 0.4 | 1.8 | 32 | 0.8 | 1.6 | 46 | 1.4 | 1.4 |
| 5 | 0.2 | 1 | 19 | 0.4 | 2 | 33 | 0.8 | 1.8 | 47 | 1.4 | 1.6 |
| 6 | 0.2 | 1.2 | 20 | 0.6 | 0.6 | 34 | 0.8 | 2 | 48 | 1.4 | 1.8 |
| 7 | 0.2 | 1.4 | 21 | 0.6 | 0.8 | 35 | 1 | 1 | 49 | 1.4 | 2 |
| 8 | 0.2 | 1.6 | 22 | 0.6 | 1 | 36 | 1 | 1.2 | 50 | 1.6 | 1.6 |
| 9 | 0.2 | 1.8 | 23 | 0.6 | 1.2 | 37 | 1 | 1.4 | 51 | 1.6 | 1.8 |
| 10 | 0.2 | 2 | 24 | 0.6 | 1.4 | 38 | 1 | 1.6 | 52 | 1.6 | 2 |
| 11 | 0.4 | 0.4 | 25 | 0.6 | 1.6 | 39 | 1 | 1.8 | 53 | 1.8 | 1.8 |
| 12 | 0.4 | 0.6 | 26 | 0.6 | 1.8 | 40 | 1 | 2 | 54 | 1.8 | 2 |
| 13 | 0.4 | 0.8 | 27 | 0.6 | 2 | 41 | 1.2 | 1.2 | 55 | 2 | 2 |
| 14 | 0.4 | 1 | 28 | 0.8 | 0.8 | 42 | 1.2 | 1.4 | | | |

Tabela 6.1: Variação dos parâmetros *LimiteInferiorTaxaCooperacao* (LI) e *LimiteSuperiorTaxaCooperacao* (LS) para os diferentes experimentos (e) realizados no cenário *5mLIS*.

Análise das Métricas Consideradas

As Figuras 6.1 a 6.3 apresentam as médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário *5mLIS*, sendo a primeira em gráfico de barras, a segunda em gráfico de superfície e a terceira em gráfico de projeção da superfície. Analisando esses gráficos, três fatos podem ser constatados. O primeiro é que a qualidade recebida é máxima quando a taxa de cooperação é fixada em 1 (experimento 35 da Figura 6.1). O segundo é que quando a taxa de cooperação varia, os limites inferiores e superiores que maximizam as qualidades recebidas são aqueles cujas médias LS mais se aproximam de 1 (0.2 e 1.8, 0.4 e 1.6, 0.6 e 1.4, e assim por diante), o que pode ser melhor visualizado nas Figuras 6.2 e 6.3, nas proximidades das respectivas coordenadas. O terceiro é que, dos pares citados, aqueles que maximizam as qualidades recebidas são os que permitem uma menor variação das taxas de cooperação, ou seja, apresentam uma menor distância entre si (0.8 e 1.2, por exemplo).

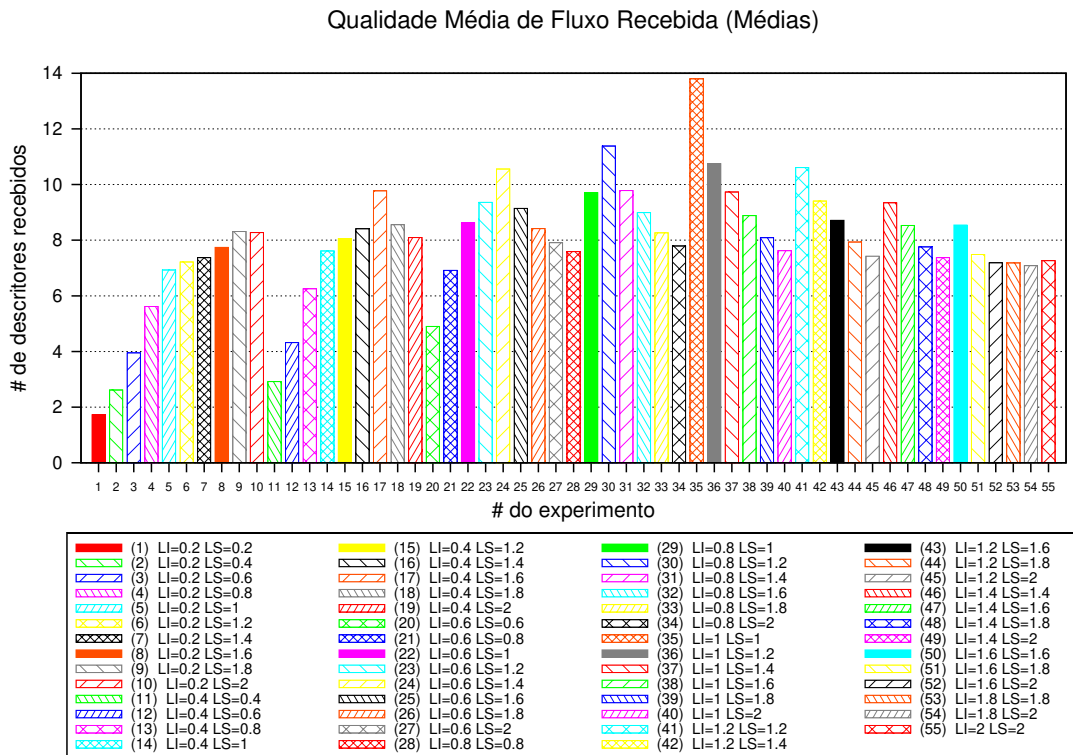


Figura 6.1: Médias das qualidades médias de fluxo recebidas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mLIS* (gráfico de barras).

A explicação para o primeiro fato, da taxa de cooperação fixada em 1 maximizar as qualidades recebidas, pode ser dada de forma intuitiva. Com taxas menores que 1, a oferta de bandas

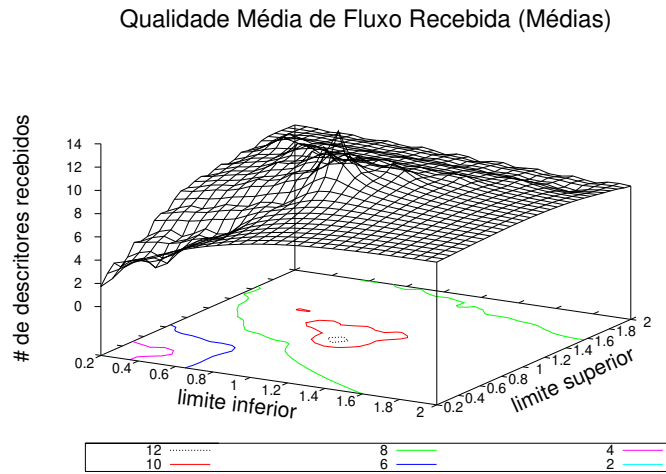


Figura 6.2: Médias das qualidades médias de fluxo recebidas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mLIS* (gráfico de superfície).

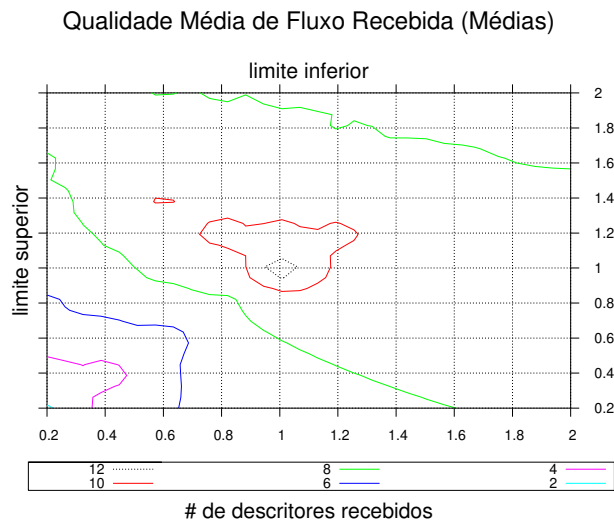


Figura 6.3: Médias das qualidades médias de fluxo recebidas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mLIS* (gráfico de projeção da superfície).

cooperadas é menor do que a respectiva demanda, ou seja, os clientes consomem mais banda do que cooperam em troca. Neste caso, falta recursos no sistema para atender todas as requisições e, conseqüentemente, os clientes são admitidos em um número menor de árvores do que suas bandas de entrada permitem, degradando a qualidade recebida. Por outro lado, com taxas maiores que 1, tem-se uma oferta maior que a demanda. Não obstante, apesar de sobraem recursos no sistema, os clientes muitas vezes não conseguem obter o número descritores que suas bandas de entrada permitem porque não possuem bandas de saída suficientes para “pagarem” a taxa de cooperação estipulada, ou seja, exige-se mais dos clientes do que o necessário, em termos de cooperação.

Pela mesma razão, pode-se explicar o segundo fato constatado, dos limites inferiores e superiores cujas médias mais se aproximam de 1 serem os que maximizam as qualidades recebidas. Enquanto aproximadamente metade dos clientes conectados cooperam com taxas inferiores a 1, a outra metade compensa este deficit cooperando com taxas superiores a 1, garantindo um equilíbrio, na média, onde não falta e nem sobra recursos no sistema.

Com relação ao terceiro fato observado, dos pares de limites inferiores e superiores que maximizam as qualidades recebidas serem os que permitem uma menor variação das taxas de cooperação, acredita-se que isso ocorra devido à discretização das quantidades de bandas consumidas e cooperadas pelos clientes no cálculo das mesmas, realizado pelo mecanismo de incentivo proposto. Conforme visto no Capítulo 4, as taxas de cooperação são determinadas em valores contínuos, contidos no intervalo compreendido entre os limites inferior e superior, através da utilização de reputações baseadas em tempos de permanência que, por suas vezes, também são contínuas. Logo, ao se multiplicar a taxa de cooperação contínua pela banda consumida – que é discreta, em termos de número de descritores recebidos –, obtém-se a banda a ser cooperada em valor contínuo, que precisa ser discretizada. O mesmo ocorre ao se dividir a banda cooperada pela taxa de cooperação, para se determinar a banda que pode ser consumida. Dessa maneira, há sempre um resíduo de banda consumida ou cooperada que não é utilizado por não ser suficientemente grande para acomodar um novo descritor (servido ou recebido), prejudicando o desempenho da métrica em questão. Além disso, quanto maior for a faixa de variação permitida à taxa de cooperação, um número maior de descritores pode ser afetado por este arredondamento, o que explica a degradação da qualidade recebida conforme a distância entre os limites inferior e superior aumenta.

Entende-se esse prejuízo como sendo o custo de utilização do mecanismo proposto. Por esse motivo, considera-se posteriormente, em todas as análises comparativas, a utilização do incentivo secundário com a combinação $LimiteInferiorTaxaCooperacao = 1$ e $LimiteSuperiorTaxaCooperacao = 1$. Apesar não prover, na prática, um incentivo para os clientes ficarem mais tempos conectados, já que a taxa de cooperação não varia em função do tempo de permanência, este caso representa o desempenho máximo que o mecanismo proposto poderia atingir, dando uma idéia do quanto se poderia ganhar, idealmente, sem a dedução do seu custo

de utilização.

Poder-se-ia minimizar esse custo de utilização com o aumento do número de árvores de distribuição utilizadas. Dessa forma, o fluxo original seria codificado em mais descritores de tamanhos menores, o que reduziria a distância entre os valores discretos a serem adotados para as bandas consumidas e cooperadas. De fato, testes realizados com 4, 16 e 32 árvores comprovam essa idéia. Ainda assim, optou-se por continuar com 16 árvores ao invés de 32, pelos seguintes motivos. Primeiramente, o mecanismo proposto deve ser avaliado com valores típicos sugeridos pela literatura [44, 43, 14, 45], ainda que sejam os piores casos. Em segundo lugar, com 16 árvores é provável que estas sejam mais estreitas e profundas do que seriam com 32, com um mesmo número de clientes; este fato propicia um melhor estudo das alterações nas taxas de rupturas quando se varia os tempos de permanência dos clientes.

Com relação ao estudo da escalabilidade do sistema conforme o número de clientes aumenta, utilizou-se a métrica capacidade de admissão do sistema. Esta pode ser entendida como uma medida dos recursos disponíveis no sistema para atender novos clientes. Caso ela aumente excessivamente, tem-se que recursos (dos clientes) estão sendo mais coletados do que utilizados. Por outro lado, caso ela tenda a zero, tem-se que o sistema não foi escalável com o aumento do número de clientes. Dessa forma, pode-se dizer que os valores ideais para essa métrica são aqueles que praticamente permanecem constantes conforme o número de clientes atendidos aumenta.

As Figuras 6.4 a 6.6 apresentam as médias das capacidades de admissão do sistema, para cada experimento do cenário *5mLIS*, sendo a primeira em gráfico de barras, a segunda em gráfico de superfície e a terceira em gráfico de projeção da superfície. Através da análise desses gráficos, pode-se comprovar o que foi afirmado. Os pontos onde as qualidades recebidas são maximizadas são justamente aqueles onde as capacidades de admissão deixam de ser nulas e tendem a aumentar (0.2 e 1.8, 0.4 e 1.6, 0.6 e 1.4, 0.8 e 1.2, e 1 e 1), o que pode ser melhor visualizado nas Figuras 6.5 e 6.6, nas proximidades das respectivas coordenadas. Na Figura 6.4, este fato também pode ser constatado comparando-se os experimentos adjacentes 9 e 10, 17 e 18, 24 e 25, 30 e 31, e 35 e 36. Nesses pontos, atinge-se um equilíbrio entre a demanda e a oferta de recursos no sistema, não se exigindo dos clientes nada além do que o necessário, em termos de cooperação.

Resta dizer que, sendo o ponto $LimiteInferiorTaxaCooperacao = 1$ e $LimiteSuperiorTaxaCooperacao = 1$ particularmente interessante, pelos motivos já mencionados, optou-se por também investigar suas redondezas. Em particular, considera-se experimentos com $LimiteInferiorTaxaCooperacao = 1$ fixo, variando $LimiteSuperiorTaxaCooperacao$ com valores próximos, de até 1.5 no máximo. Pretende-se, com esses experimentos, analisar os efeitos do custo de utilização do mecanismo proposto (discretização de valores das bandas consumidas e cooperadas) sem nenhum deficit de recursos ocorrendo (todos os clientes cooperando com pelo menos o que consomem).

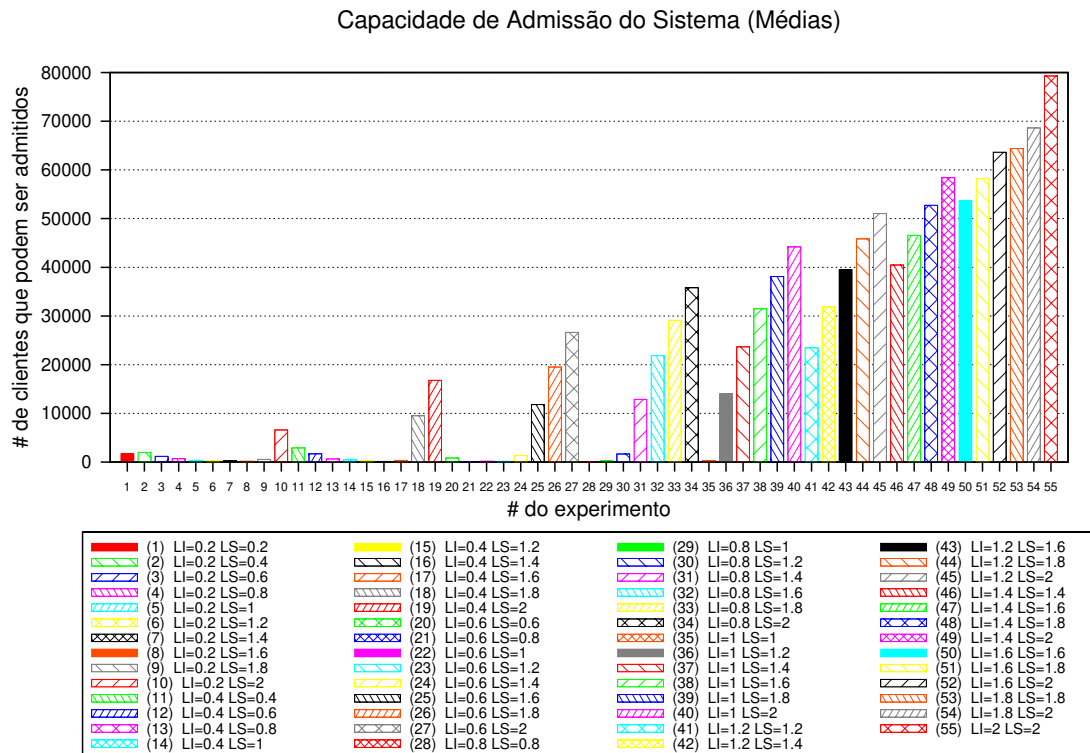


Figura 6.4: Médias das capacidades de admissão do sistema, para cada experimento do cenário *5mLIS* (gráfico de barras).

A Tabela 6.2 descreve os valores seleccionados nesta subsecção para os parâmetros *LimiteInferiorTaxaCooperacao* e *LimiteSuperiorTaxaCooperacao*.

6.2.2 Tempo Requerido de Cooperador Estável

Para o ajuste do parâmetro *TempoRequeridoCooperadorEstavel*, detalha-se, primeiramente, os experimentos realizados, com suas configurações e variações específicas. Em seguida, a análise das métricas consideradas é realizada.

Experimentos Realizados

O cenário *24hTRCELIS* foi criado para a determinação do parâmetro *TempoRequeridoCooperadorEstavel*. Neste cenário, o sistema é submetido a condições normais de tráfego (sem *flash crowd*) durante vinte e quatro horas, mantendo o número de clientes praticamente estável após o equilíbrio. Este período de tempo é o mesmo a ser utilizado nas análises comparativas que

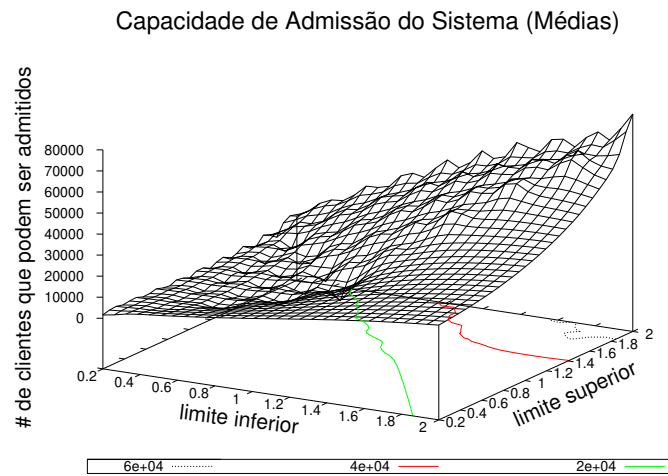


Figura 6.5: Médias das capacidades de admissão do sistema, para cada experimento do cenário *5mLIS* (gráfico de superfície).

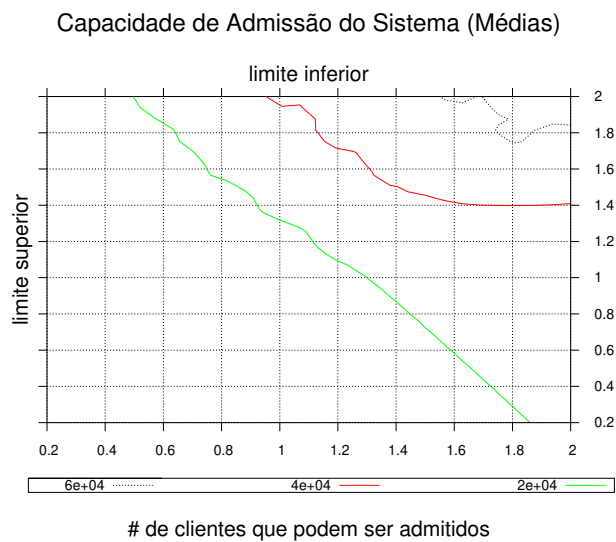


Figura 6.6: Médias das capacidades de admissão do sistema, para cada experimento do cenário *5mLIS* (gráfico de projeção da superfície).

| n | LI | LS | n | LI | LS |
|---|------|------|----|----|------|
| 1 | 0.5 | 1.5 | 8 | 1 | 1.05 |
| 2 | 0.6 | 1.4 | 9 | 1 | 1.1 |
| 3 | 0.7 | 1.3 | 10 | 1 | 1.2 |
| 4 | 0.8 | 1.2 | 11 | 1 | 1.3 |
| 5 | 0.9 | 1.1 | 12 | 1 | 1.4 |
| 6 | 0.95 | 1.05 | 13 | 1 | 1.5 |
| 7 | 1 | 1 | | | |

Tabela 6.2: Valores selecionados no cenário *5mLIS* para os parâmetros *LimiteInferiorTaxaCooperacao* (LI) e *LimiteSuperiorTaxaCooperacao* (LS).

serão apresentadas posteriormente. Escolheu-se este período de tempo por abranger todas as fases de um dia típico de trabalho ou de final de semana dos usuários, incluindo os picos de acesso (como horários de almoço e términos de expediente) e períodos de baixo acesso (como de madrugada, após um determinado horário).

Segundo [57, 53], os padrões de acesso repetem-se dia a dia, em períodos de vinte e quatro horas. Conforme já mencionado, as variações de acesso foram reproduzidas por processos de chegada *Poisson* não estacionários, tendo suas taxas variantes reinicializadas após vinte e quatro horas decorridas de simulação. É importante diferenciar uma situação de *flash crowd* dos picos de acesso aqui mencionados. Enquanto o primeiro apresenta uma elevação exagerada da taxa de chegada em virtude de uma situação atípica (como a transmissão de uma notícia importante), os segundos representam aumentos moderados e gradativos decorrentes das rotinas diárias dos usuários.

O objetivo do presente cenário é encontrar os melhores valores para o parâmetro *TempoRequeridoCooperadorEstavel* de forma que as qualidades recebidas sejam maximizadas. A escalabilidade do sistema não é o principal fator a ser observado, dado que o número de clientes não é monotonicamente crescente, como no cenário anterior. Conseqüentemente, somente a métrica qualidade média de fluxo recebida é avaliada.

Observa-se que o parâmetro *TempoRequeridoCooperadorEstavel* apresenta dependências com os parâmetros *LimiteInferiorTaxaCooperacao* e *LimiteSuperiorTaxaCooperacao*. Dessa forma, para cada par de valores adotados para estes últimos, indicados na Tabela 6.2, as mesmas variações de *TempoRequeridoCooperadorEstavel* são consideradas.

O parâmetro *TempoRequeridoCooperadorEstavel* permite que um ajuste fino seja feito para cada par de limites inferior e superior adotado, adequando suas ações em períodos de tempo mais longos. Se, por um lado, tais períodos não foram considerados nas simulações do cenário anterior, justamente para se anular o efeito deste parâmetro, por outro, eles são essenciais no

presente cenário, para a determinação do mesmo. Períodos de tempo curtos nunca disparariam a promoção de cooperadores estáveis, pois o referido parâmetro seria maior que o tempo de simulação.

Os valores utilizados para o parâmetro *TempoRequeridoCooperadorEstavel* no presente cenário são baseados na média dos tempos de permanência dos clientes, já que estas duas variáveis estão relacionadas. Dado que esta última segue uma distribuição *lognormal* com parâmetros $\mu = 5.74$ e $\sigma = 2.01$ (Capítulo 5), tem-se que a sua média é 2345.02 segundos ($m = e^{\mu + \sigma^2/2}$). Assim, utiliza-se os tempos discretos 2345 segundos (1×2345), 4690 segundos (2×2345), 7035 segundos (3×2345), ..., 35175 segundos (15×2345) como valores para o parâmetro *TempoRequeridoCooperadorEstavel*. Tempos menores que 2345 segundos e maiores que 35175 segundos foram avaliados, porém constata-se que aqueles que maximizam a métrica analisada estão dentro deste intervalo. Adicionalmente, verifica-se que as distâncias entre os valores discretos são suficientes para uma aproximação razoável do valor ideal. Com essas combinações, o número de experimentos no cenário *24hTRCELIS* já cresce muito. Caso uma menor distância entre os valores fosse considerada, o tempo de simulação demandado para todas as combinações resultantes inviabilizaria a determinação do valor ideal para o parâmetro em questão. Os parâmetros para cada experimento deste cenário podem ser vistos na Tabela 6.3.

Análise das Métricas Consideradas

As Figuras 6.7 e 6.8 apresentam, respectivamente, as médias e as medianas das qualidades médias de fluxo recebidas pelos clientes, para cada experimento do cenário *24hTRCELIS*. Analisando esses gráficos, pode-se constatar alguns fatos interessantes. Primeiramente, para os casos onde não há variação da taxa de cooperação (limites inferior e superior iguais), o parâmetro *TempoRequeridoCooperadorEstavel* não faz diferença na métrica analisada, o que é evidente (experimentos de 91 a 105). Para os casos onde o limite inferior é fixado em 1 e o superior variado (sobra recursos no sistema), os valores de *TempoRequeridoCooperadorEstavel* que maximizam a métrica em questão são os menores (experimentos de 106 a 195). Isso ocorre porque o aumento do número de cooperadores estáveis não traz prejuízos ao sistema, em termos de deficit de recursos; ao contrário, alivia o excesso de cooperação exigido por esses limites, promovendo mais clientes à taxa mínima (limite inferior) e não rebaixando tanto a reputação daqueles que acabam de se conectar no sistema.

Já para os casos onde a taxa de cooperação varia, mas tem como média o valor 1 (limites inferiores e superiores equidistantes de 1), nota-se que diferentes valores de *TempoRequeridoCooperadorEstavel* maximizam a métrica em questão (experimentos de 1 a 90). Nestes casos, quanto maior é a variação permitida da taxa de cooperação (*LimiteInferiorTaxaCooperacao* e *LimiteSuperiorTaxaCooperacao* mais distantes entre si), maiores são os valores ideais do parâmetro em questão, ou seja, números menores de cooperadores estáveis são mais adequados para compensar esta variação, uma vez que aumentam os recursos no sistema.

| TR | LI=0.5 LS=1.5 | LI=0.6 LS=1.4 | LI=0.7 LS=1.3 | LI=0.8 LS=1.2 | LI=0.9 LS=1.1 | LI=0.95 LS=1.05 | LI=1 LS=1 |
|-------|------------------|------------------|------------------|------------------|------------------|--------------------|--------------|
| 2345 | 1 | 16 | 31 | 46 | 61 | 76 | 91 |
| 4690 | 2 | 17 | 32 | 47 | 62 | 77 | 92 |
| 7035 | 3 | 18 | 33 | 48 | 63 | 78 | 93 |
| 9380 | 4 | 19 | 34 | 49 | 64 | 79 | 94 |
| 11725 | 5 | 20 | 35 | 50 | 65 | 80 | 95 |
| 14070 | 6 | 21 | 36 | 51 | 66 | 81 | 96 |
| 16415 | 7 | 22 | 37 | 52 | 67 | 82 | 97 |
| 18760 | 8 | 23 | 38 | 53 | 68 | 83 | 98 |
| 21105 | 9 | 24 | 39 | 54 | 69 | 84 | 99 |
| 23450 | 10 | 25 | 40 | 55 | 70 | 85 | 100 |
| 25795 | 11 | 26 | 41 | 56 | 71 | 86 | 101 |
| 28140 | 12 | 27 | 42 | 57 | 72 | 87 | 102 |
| 30485 | 13 | 28 | 43 | 58 | 73 | 88 | 103 |
| 32830 | 14 | 29 | 44 | 59 | 74 | 89 | 104 |
| 35175 | 15 | 30 | 45 | 60 | 75 | 90 | 105 |

| TR | LI=1 LS=1.05 | LI=1 LS=1.1 | LI=1 LS=1.2 | LI=1 LS=1.3 | LI=1 LS=1.4 | LI=1 LS=1.5 |
|-------|-----------------|----------------|----------------|----------------|----------------|----------------|
| 2345 | 106 | 121 | 136 | 151 | 166 | 181 |
| 4690 | 107 | 122 | 137 | 152 | 167 | 182 |
| 7035 | 108 | 123 | 138 | 153 | 168 | 183 |
| 9380 | 109 | 124 | 139 | 154 | 169 | 184 |
| 11725 | 110 | 125 | 140 | 155 | 170 | 185 |
| 14070 | 111 | 126 | 141 | 156 | 171 | 186 |
| 16415 | 112 | 127 | 142 | 157 | 172 | 187 |
| 18760 | 113 | 128 | 143 | 158 | 173 | 188 |
| 21105 | 114 | 129 | 144 | 159 | 174 | 189 |
| 23450 | 115 | 130 | 145 | 160 | 175 | 190 |
| 25795 | 116 | 131 | 146 | 161 | 176 | 191 |
| 28140 | 117 | 132 | 147 | 162 | 177 | 192 |
| 30485 | 118 | 133 | 148 | 163 | 178 | 193 |
| 32830 | 119 | 134 | 149 | 164 | 179 | 194 |
| 35175 | 120 | 135 | 150 | 165 | 180 | 195 |

Tabela 6.3: Variação dos parâmetros *LimiteInferiorTaxaCooperacao* (LI), *LimiteSuperiorTaxaCooperacao* (LS) e *TempoRequeridoCooperadorEstavel* (TR) para os diferentes experimentos realizados no cenário *24hTRCELIS*. Cada célula corresponde ao número do experimento, definido em função dos valores de LI e LS (coluna) e TR (linha).

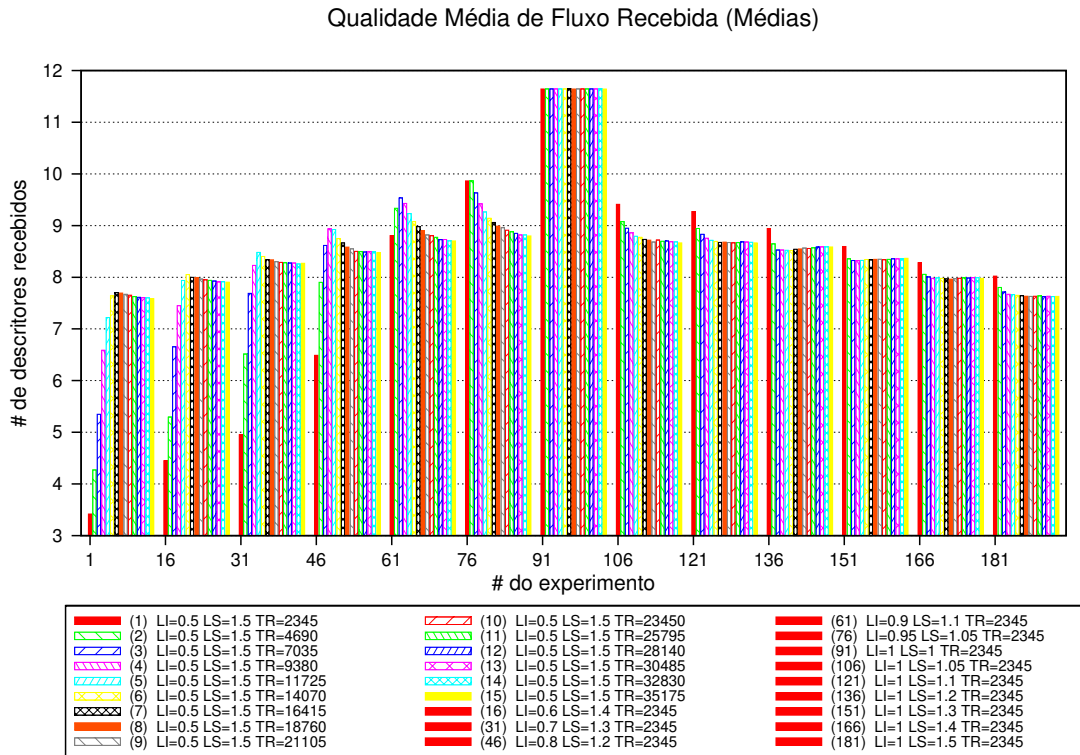


Figura 6.7: Médias das qualidades médias de fluxo recebidas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *24hTRCELIS*. As legendas que foram omitidas, por questões de espaço, podem ser encontradas na Tabela 6.3.

Percebe-se, no presente cenário, as mesmas tendências constatadas no cenário anterior com relação à oferta e à demanda de recursos no sistema. Tempos requeridos menores que os ideais elevam o número de cooperadores estáveis, diminuindo os recursos disponíveis no sistema, o que faz com que as qualidades recebidas sejam inferiores – uma vez que a oferta fica menor que a demanda. Por outro lado, tempos requeridos maiores que os ideais diminuem o número de cooperadores estáveis, aumentando os recursos disponíveis no sistema além do necessário, o que também faz com que as qualidades recebidas sejam inferiores – já que os clientes não conseguem “pagar” pela qualidade desejada segundo as taxas de cooperação estipuladas. Conforme mencionado, é interessante notar como o parâmetro *TempoRequeridoCooperadorEstavel* provê uma forma de se fazer um ajuste fino em um período longo de simulação para um dado par de limites inferior e superior que foram, por suas vezes, previamente escolhidos de forma análoga.

Finalmente, os valores selecionados nesta subseção para o parâmetro *TempoRequeridoCooperadorEstavel*, para cada par de limites inferior e superior indicado na Tabela 6.2, podem ser visualizados na Tabela 6.4.

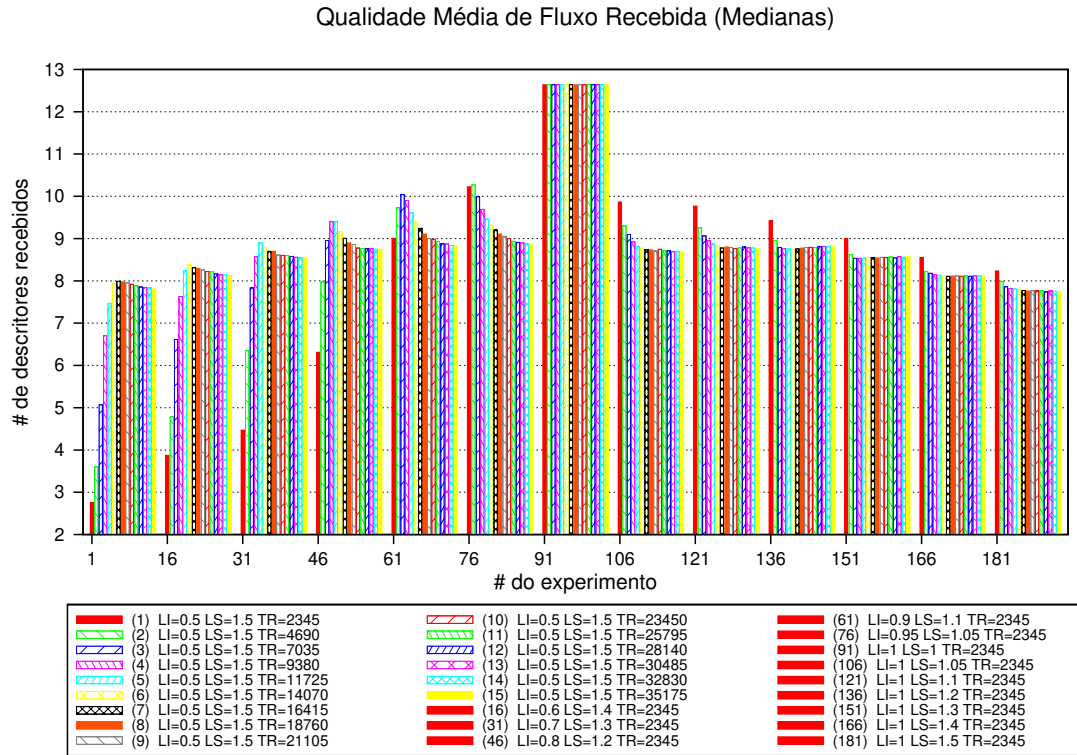


Figura 6.8: Medianas das qualidades médias de fluxo recebidas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *24hTRCELIS*. As legendas que foram omitidas, por questões de espaço, podem ser encontradas na Tabela 6.3.

| n | LI | LS | TR | n | LI | LS | TR |
|---|------|------|-------|----|----|------|------|
| 1 | 0.5 | 1.5 | 16415 | 8 | 1 | 1.05 | 2345 |
| 2 | 0.6 | 1.4 | 14070 | 9 | 1 | 1.1 | 2345 |
| 3 | 0.7 | 1.3 | 11725 | 10 | 1 | 1.2 | 2345 |
| 4 | 0.8 | 1.2 | 9380 | 11 | 1 | 1.3 | 2345 |
| 5 | 0.9 | 1.1 | 7035 | 12 | 1 | 1.4 | 2345 |
| 6 | 0.95 | 1.05 | 4690 | 13 | 1 | 1.5 | 2345 |
| 7 | 1 | 1 | 2345 | | | | |

Tabela 6.4: Valores seleccionados no cenário *24hTRCELIS* para o parâmetro *TempoRequeridoCooperadorEstavel* (TR), para cada par de *LimiteInferiorTaxaCooperacao* (LI) e *LimiteSuperiorTaxaCooperacao* (LS) escolhido anteriormente.

6.3 Resultados dos Experimentos de Simulação

Nesta seção, detalha-se, primeiramente, os cenários e experimentos realizados para a avaliação do mecanismo de incentivo proposto, com suas configurações e variações específicas. Em seguida, os benefícios dos incentivos primário e secundário são discutidos, cada qual em sua subseção.

6.3.1 Experimentos Realizados

Para a avaliação do mecanismo de incentivo proposto, dois cenários foram criados: *24h* e *5mFC*. No primeiro, assim como no cenário *24hTRCELIS*, utilizado anteriormente, o sistema é submetido a condições normais de tráfego (sem *flash crowd*) durante vinte e quatro horas, mantendo o número de clientes praticamente estável após o equilíbrio. As explicações para a escolha deste período de tempo, bem como para a forma de reprodução das diferentes variações de tráfego de acesso foram dadas na Subseção 6.2.2. No segundo cenário, assim como no *5mLIS*, descrito anteriormente, o sistema é submetido a uma situação de *flash crowd* intensa durante os cinco primeiros minutos de simulação, na qual o número de clientes admitidos cresce rapidamente. As explicações para a escolha deste período de tempo também foram dadas anteriormente, na Subseção 6.2.1.

Além dos tempos de simulação e variações de tráfego, os demais parâmetros utilizados nos cenários *24hTRCELIS* e *5mLIS* também foram igualmente reproduzidos nos cenários *24h* e *5mFC*, respectivamente. A diferença entre os dois cenários anteriores e os dois introduzidos nesta subseção está nos experimentos considerados: enquanto em *24hTRCELIS* e *5mLIS* utiliza-se o mecanismo proposto somente de forma completa (com seus incentivos primário e secundário habilitados simultaneamente), de maneira a se encontrar os melhores parâmetros a serem utilizados, em *24h* e *5mFC* realiza-se uma análise comparativa entre os seguintes grupos de experimentos: (1) não utilização de nenhum mecanismo de incentivo; (2) utilização somente do incentivo primário; e (3) utilização dos incentivos primário e secundário simultaneamente, com os melhores parâmetros escolhidos previamente.

O principal objetivo dos cenários *24h* e *5mFC* é avaliar os benefícios proporcionados ao sistema pelo mecanismo de incentivo proposto, tanto em situações de *flash crowd* quanto em condições normais de tráfego. Para isso, as seguintes métricas são consideradas: qualidade média de fluxo recebida, capacidade de admissão do sistema, probabilidade de bloqueio do sistema e taxa de rupturas em árvores.

Para cada um dos três grupos mencionados, incluiu-se experimentos com diferentes variações. No primeiro grupo, variou-se a percentagem de clientes que cooperam, incluindo os valores 0%, 10%, 25%, 30%, 50%, 75% e 100%. Neste caso, ou os clientes cooperam com toda a banda de saída que possuem (os selecionados pela percentagem em questão), ou não cooperam com absolutamente nada, somente consumindo recursos do sistema. Os valores 0%,

100% e 30% são particularmente interessantes. O primeiro representa o sistema degenerado à arquitetura cliente-servidor, ou seja, dado que nenhum cliente coopera, apenas o servidor admite nós abaixo de si nas árvores de distribuição. O segundo valor representa um ambiente com altruísmo total, ou seja, o ponto máximo em termos de cooperação que uma rede *peer-to-peer* poderia chegar. O terceiro valor representa a percentagem real de cooperação, constatada neste tipo de sistema pela literatura [1]. Os demais valores foram distribuídos em incrementos de 25% dentro do intervalo, além do valor 10% que também foi incluso.

No segundo grupo de experimentos, que envolve a utilização somente do incentivo primário, variou-se a taxa de cooperação em torno do valor 1 que, conforme visto anteriormente, é o ponto de equilíbrio entre a oferta e a demanda de recursos no sistema. É importante observar que, neste grupo, em virtude da utilização do padrão de incentivo escambo, tem-se que todos os clientes cooperam na taxa estipulada, ao contrário do grupo anterior. Logo, não há motivos para se considerar o valor de taxa de cooperação 0, o qual representaria 100% dos clientes não cooperando com nada – a mesma situação do experimento de valor 0% do grupo anterior. Resta ainda dizer que, uma vez que o incentivo secundário não está sendo utilizado neste grupo, tem-se que as taxas de cooperação são constantes e iguais para todos os clientes, independentemente de seus tempos de permanência. Assim, em intervalos de 0.25 investiga-se as redondezas da taxa de cooperação 1: 0.5, 0.75, 1.25 e 1.5. Adicionalmente, foram incluídos os valores 2 (extremo máximo do intervalo considerado na Subseção 6.2.1) e 0.3 (para se comparar com o experimento de valor 30% do primeiro grupo, que representa um valor real da literatura). Neste último, a idéia é pesquisar qual dos seguintes casos é melhor de se ter: 30% dos clientes cooperando com 100% de suas bandas de saída (primeiro grupo) ou 100% dos clientes cooperando com 30% das bandas que consomem (segundo grupo).

No terceiro grupo de experimentos, que considera a utilização dos incentivos primário e secundário simultaneamente, variou-se os parâmetros *LimiteInferiorTaxaCooperacao*, *LimiteSuperiorTaxaCooperacao* e *TempoRequeridoCooperadorEstavel*, de acordo com os valores ideais indicados na Tabela 6.4. Vale lembrar que, apesar de todos os clientes cooperarem incondicionalmente em decorrência do uso do padrão de incentivo escambo (presente no incentivo primário), neste grupo as taxas de cooperação são variáveis e diferentes entre todos eles, sendo calculadas de acordo com suas reputações (incentivo secundário). Conseqüentemente, com a motivação de aumentarem suas reputações, todos os clientes, com exceção daqueles que pertencem à classe de conexão discada, ficam mais tempo conectados no sistema (Subseção 5.2.1).

Conforme mencionado na Subseção 6.2.1, apesar da combinação *LimiteInferiorTaxaCooperacao* = 1 e *LimiteSuperiorTaxaCooperacao* = 1 não prover, na prática, um incentivo para os clientes ficarem mais tempos conectados, já que a taxa de cooperação não varia em função do tempo de permanência, ela representa, na teoria, o desempenho máximo que o mecanismo proposto poderia atingir, fazendo com que os clientes tenham o tempo extra de

permanência somado a seus tempos originais mas, idealmente, sem a queda nas qualidades recebidas, que representa o custo de utilização do mecanismo. Dessa maneira, há um experimento no terceiro grupo que representa esta configuração. A diferença entre ele e o experimento do segundo grupo cuja taxa de cooperação possui o valor 1 é somente o tempo extra de permanência.

Finalmente, as configurações dos experimentos de cada um dos três grupos descritos, a serem utilizados igualmente nos cenários *24h* e *5mFC*, podem ser visualizadas na Tabela 6.5.

| Primeiro Grupo | | | Terceiro Grupo | | | | |
|----------------|---------------|----------|----------------|-----------|------|------|-------|
| e | Incentivo | Cooperam | e | Incentivo | LI | LS | TR |
| 1 | Sem Incentivo | 0% | 15 | Reputação | 0.5 | 1.5 | 16415 |
| 2 | Sem Incentivo | 10% | 16 | Reputação | 0.6 | 1.4 | 14070 |
| 3 | Sem Incentivo | 25% | 17 | Reputação | 0.7 | 1.3 | 11725 |
| 4 | Sem Incentivo | 30% | 18 | Reputação | 0.8 | 1.2 | 9380 |
| 5 | Sem Incentivo | 50% | 19 | Reputação | 0.9 | 1.1 | 7035 |
| 6 | Sem Incentivo | 75% | 20 | Reputação | 0.95 | 1.05 | 4690 |
| 7 | Sem Incentivo | 100% | 21 | Reputação | 1 | 1 | 2345 |
| Segundo Grupo | | | 22 | Reputação | 1 | 1.05 | 2345 |
| e | Incentivo | TC | 23 | Reputação | 1 | 1.1 | 2345 |
| 8 | Escambo | 0.3 | 24 | Reputação | 1 | 1.2 | 2345 |
| 9 | Escambo | 0.5 | 25 | Reputação | 1 | 1.3 | 2345 |
| 10 | Escambo | 0.75 | 26 | Reputação | 1 | 1.4 | 2345 |
| 11 | Escambo | 1 | 27 | Reputação | 1 | 1.5 | 2345 |
| 12 | Escambo | 1.25 | | | | | |
| 13 | Escambo | 1.5 | | | | | |
| 14 | Escambo | 2 | | | | | |

Tabela 6.5: Diferentes experimentos (e) realizados nos cenários *24h* e *5mFC*, pertencentes aos grupos (1) sem incentivo, variando a percentagem dos clientes que cooperam; (2) escambo, variando a taxa de cooperação (TC); e (3) reputação, variando os parâmetros *LimiteInferiorTaxaCooperacao* (LI), *LimiteSuperiorTaxaCooperacao* (LS) e *TempoRequeridoCooperadorEstavel* (TR).

6.3.2 Benefícios do Incentivo Primário do Mecanismo Proposto

O principal objetivo do incentivo primário do mecanismo proposto é proporcionar um aumento de escalabilidade à rede *peer-to-peer* a partir da obrigatoriedade de cooperação dos clientes, de

forma a viabilizar os serviços de vídeo sob demanda neste ambiente. Além disso, a maximização da qualidade recebida é desejável, já que evidencia que os recursos acumulados no sistema, a partir das cooperações, estão sendo aproveitados eficientemente. Para se constatar estes dois benefícios, escalabilidade e qualidade recebida, a análise de três métricas é realizada: qualidade média de fluxo recebida, capacidade de admissão do sistema e probabilidade de bloqueio do sistema.

A métrica probabilidade de bloqueio não precisou ser considerada na fase de ajuste de parâmetros porque para todos os experimentos daqueles cenários apresentou o valor 0 (nenhum cliente bloqueado). Não obstante, para alguns experimentos dos cenários *24h* e *5mFC* ela foi diferente de 0, o que significa que a escassez de recursos não afetou somente a qualidade recebida, mas também a admissão de novos clientes. Dessa forma, por exemplo, um experimento que apresentasse bons valores de qualidade recebida e capacidade de admissão poderia, mesmo assim, ser considerado insatisfatório, caso tivesse tido probabilidade de bloqueio não nula – o que significa que deixou de atender um certo número de clientes. Conforme visto na Subseção 6.2.1, os valores ideais para a métrica capacidade de admissão do sistema são aqueles próximos de 0. Entretanto, como se pode ver, esta regra não vale para os casos onde a métrica probabilidade de bloqueio do sistema tiver sido maior que 0. Portanto, esta métrica deve ser considerada nesta subseção.

As Figuras 6.9 e 6.10 apresentam, respectivamente para os cenários *24h* e *5mFC*, as médias das qualidades médias de fluxo recebidas pelos clientes, para cada experimento destes cenários. Analisando inicialmente o primeiro grupo de experimentos (de 1 a 7), alguns fatos podem ser observados. Em primeiro lugar, a qualidade recebida é diretamente proporcional à percentagem de clientes que cooperam. Como o padrão escambo não está sendo utilizado neste grupo, não há uma amarração entre as bandas consumida e cooperada, o que significa que nenhum cliente deixa de obter uma melhor qualidade de fluxo (banda consumida) caso não tenha recursos suficientes para oferecer (banda cooperada).

Em segundo lugar, pode-se constatar que a percentagem real de altruísmo sugerida pela literatura (30%) não provê um nível satisfatório de qualidade recebida: cerca de 6 descritores em 16. Para se obter uma qualidade superior a pelo menos a metade da máxima disponível (8 descritores em 16), seria necessário que pelo menos 50% dos clientes cooperassem, o que não ocorre na prática. Na situação hipotética de 100% dos clientes cooperando com suas bandas de saída integralmente, ter-se-ia uma qualidade média de cerca de 15 descritores (experimento 7) e, finalmente, com nenhum cliente cooperando (experimento 1), o sistema degeneraria para a arquitetura cliente-servidor, apresentando uma qualidade média de cerca de 2 descritores.

Analisando agora o segundo grupo de experimentos (de 8 a 14), outras constatações podem ser feitas. Primeiramente, o valor da taxa de cooperação que maximiza a qualidade recebida é 1. Este fato comprova novamente a discussão realizada anteriormente sobre este ser o ponto de equilíbrio entre a oferta e a demanda de recursos no sistema. Em segundo lugar, enquanto

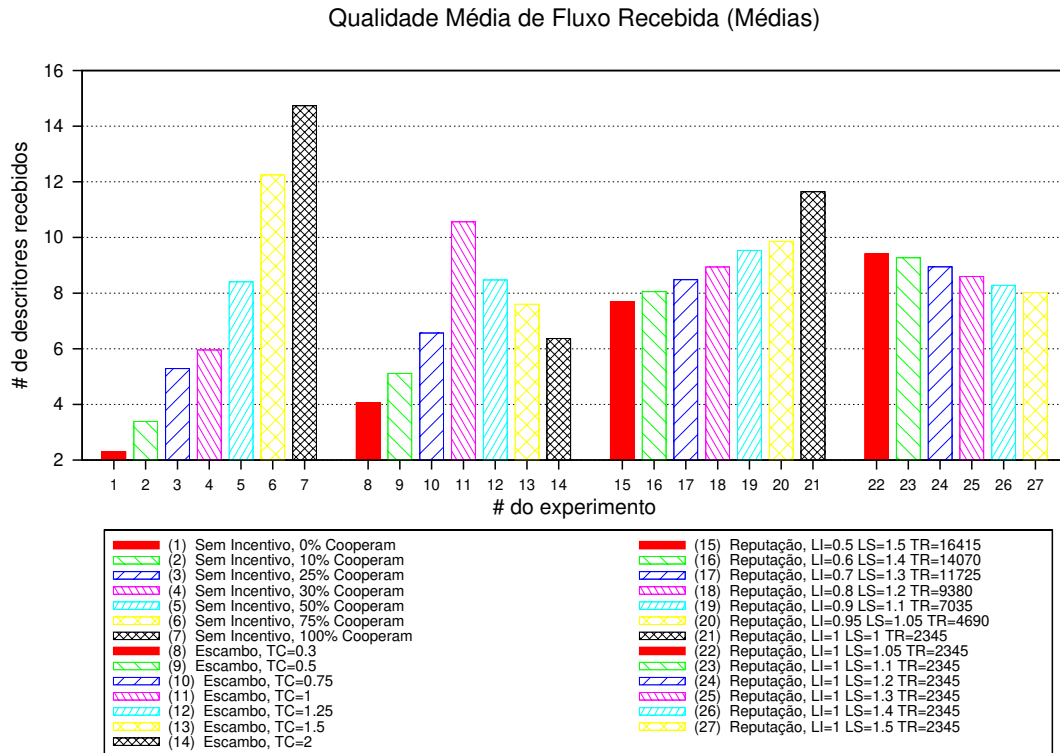


Figura 6.9: Médias das qualidades médias de fluxo recebidas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *24h*.

o primeiro grupo de experimentos apresenta resultados similares para os cenários *24h* e *5mFC*, o segundo grupo apresenta resultados notavelmente diferentes. Pode-se, na verdade, observar um aumento geral das qualidades recebidas nos experimentos de 8 a 27 (segundo e terceiro grupos), do cenário *24h* para o cenário *5mFC*. Este fato é uma evidência de que o padrão escambo, utilizado em todos esses experimentos, confere uma maior escalabilidade ao sistema, ou seja, quanto maior o número de clientes conectados, maior a qualidade recebida por eles.

Em terceiro lugar, a qualidade recebida com a utilização de somente o incentivo primário e com taxa de cooperação 1 (experimento 11) é cerca de 11 no primeiro cenário e aproximadamente 14 no segundo, significando uma melhoria de qualidade de quase o dobro, no primeiro caso, e de mais que o dobro, no segundo caso, com relação à recebida com 30% de clientes cooperando sem nenhum incentivo (experimento 4). Em quarto lugar, nos casos de *flash crowd* onde os benefícios do incentivo primário são maximizados (cenário *5mFC*), tem-se que a qualidade recebida no experimento 11 se aproxima muito da recebida no experimento 7 (100% de clientes cooperando sem nenhum incentivo, o que é uma situação hipotética), ou seja, 14 descritores no primeiro e cerca de 15 no segundo.

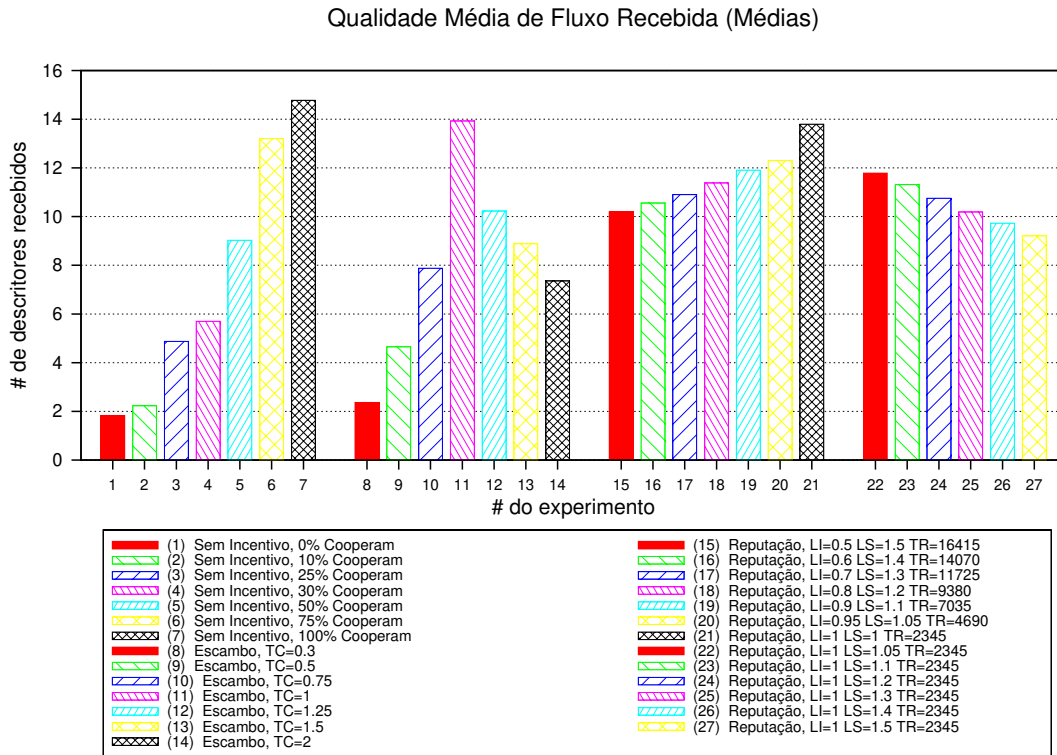


Figura 6.10: Médias das qualidades médias de fluxo recebidas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mFC*.

Analisando, finalmente, o terceiro grupo de experimentos (de 15 a 27), pode-se notar o seguinte. Em primeiro lugar, confirmando o que foi falado na Subseção 6.2.1, nota-se que quanto maior a variação permitida para a taxa de cooperação (limites inferiores e superiores mais distantes entre si), menor é a qualidade recebida pelos clientes, o que pode ser entendido como o custo de utilização do mecanismo proposto. Comparando o caso de maior variação (experimento 15) com a situação teórica na qual o incentivo secundário ocorre mesmo sem nenhuma variação (experimento 21), tem-se que este custo é de cerca de 4 descritores em 16. Mesmo com uma pequena variação permitida para as taxas de cooperação (experimentos 20 e 22), ocorre, de imediato, uma queda na qualidade recebida que corresponde a cerca de 50% do custo mencionado (aproximadamente 2 descritores em 4), ou seja, a escolha dos limites inferior e superior podem variar o custo de utilização do mecanismo de 2 a 4 descritores.

Em segundo lugar, comparando o caso do experimento 21 com o do experimento 11 (a única diferença entre eles é o aumento do tempo de permanência dos clientes), nota-se que as qualidades recebidas ficaram próximas. Isso significa que o aumento dos tempos de permanência dos clientes teve pouco efeito sobre a métrica em questão. Logo, outras métricas devem ser ana-

lisadas para se determinar se os benefícios obtidos com isso compensam o custo de utilização mencionado.

Em terceiro lugar, o propósito de se analisar os casos com o limite inferior de pelo menos 1 e o superior variando (experimentos de 22 a 27) era, recordando novamente da Subseção 6.2.1, verificar o custo de utilização do mecanismo sem a ocorrência de nenhum deficit de recursos no sistema (já que todos os clientes cooperam com no mínimo o que consomem). Verificando os resultados obtidos nesses experimentos, pode-se observar que a queda na qualidade recebida ocorre da mesma forma. Essa constatação, juntamente com o fato da variação permitida para a taxa de cooperação nos referidos experimentos ser menor do que a permitida nos experimentos de 15 a 21 tornam essas configurações de parâmetros desinteressantes de serem utilizadas.

Para se completar o estudo dos benefícios obtidos com o incentivo primário do mecanismo proposto, as métricas probabilidade de bloqueio e capacidade de admissão do sistema são analisadas. Primeiramente, as Figuras 6.11 e 6.12 apresentam, respectivamente para os cenários *24h* e *5mFC*, as probabilidades de bloqueio do sistema, para cada experimento destes cenários.

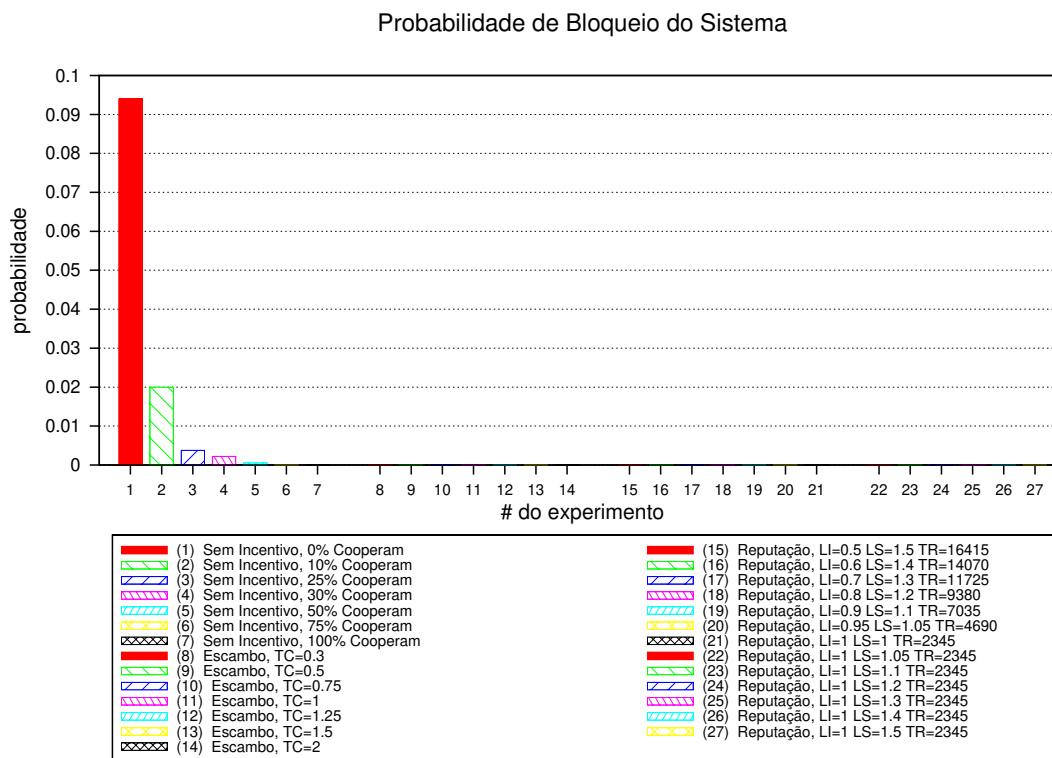


Figura 6.11: Probabilidades de bloqueio do sistema, para cada experimento do cenário *24h*.

O principal fato a ser observado nesses gráficos é que, ao contrário do que ocorre no primeiro grupo de experimentos (de 1 a 7), todos os clientes são admitidos no segundo e no terceiro

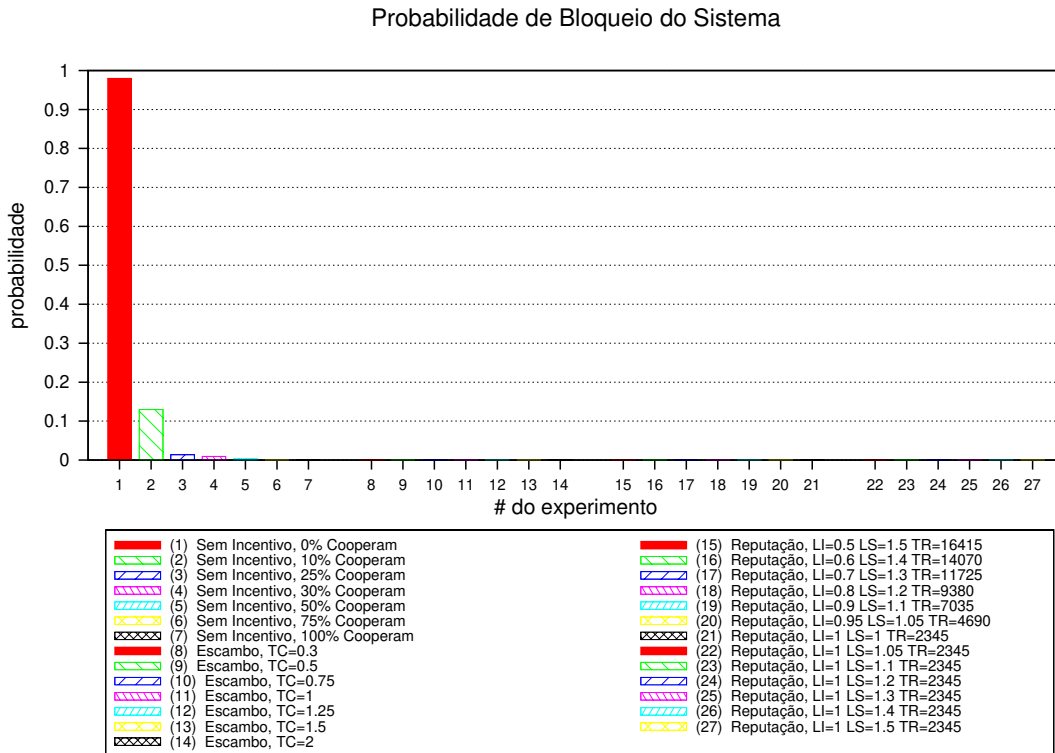


Figura 6.12: Probabilidades de bloqueio do sistema, para cada experimento do cenário *5mFC*.

grupo (de 8 a 27). Considerando que no cenário *5mFC* a taxa de chegada é elevada, representando uma situação de *flash crowd*, a ausência de qualquer bloqueio nos grupos onde o incentivo primário do mecanismo proposto está habilitado evidencia que o mesmo de fato aumenta a escalabilidade do sistema. No primeiro grupo, somente não ocorre bloqueio quando a percentagem de clientes que cooperam é superior a 50% (experimentos 6 e 7), situações que não condizem com a realidade, segundo a literatura. Para os casos nos quais ocorre bloqueio (experimentos de 1 a 5), pode-se notar a diferença entre as taxas de chegada de ambos os cenários pela escala das ordenadas dos gráficos: as probabilidades de bloqueio são cerca de 10 vezes maiores no segundo cenário.

Outro ponto a ser observado nesses gráficos é com relação à comparação entre os seguintes casos: (1) 30% dos clientes cooperando com 100% dos seus recursos (experimento 4) e (2) 100% dos clientes cooperando com 30% da banda consumida (experimento 8). Caso somente a métrica qualidade média de fluxo recebida fosse levada em consideração nesta análise, o primeiro caso seria o melhor, já que apresenta valores maiores. Não obstante, considerando que a métrica probabilidade de bloqueio indica que nem todos os clientes são atendidos no primeiro caso, o segundo passa a ser o mais interessante, ou seja, apesar dos clientes receberem qualida-

des inferiores neste último, todos eles são atendidos, o que reforça o benefício proporcionado pelo incentivo primário em termos de escalabilidade do sistema, mesmo com baixas taxas de cooperação. Esta conclusão está de acordo com a apresentada em [28, 26, 27], na qual se menciona que baixos níveis de cooperação (30% de clientes cooperando) inviabilizam a maioria dos sistemas *peer-to-peer*, exercendo um prejuízo particular às aplicações de distribuição de conteúdo com sensibilidade temporal.

Finalmente, as Figuras 6.13 e 6.14 apresentam, respectivamente para os cenários *24h* e *5mFC*, as médias das capacidades de admissão do sistema, para cada experimento destes cenários. Analisando inicialmente o primeiro grupo de experimentos (de 1 a 7), tem-se novos fatos interessantes. Primeiramente, os experimentos de 1 a 5, que apresentam probabilidades de bloqueio não nulas, não podem ter a métrica capacidade de admissão levada em consideração, já que não admitem todos os clientes que chegam, conforme mencionado anteriormente. Dessa forma, pode-se dizer que falta recursos no sistema nestes experimentos, o que evidencia a falta de escalabilidade com os respectivos níveis de cooperação.

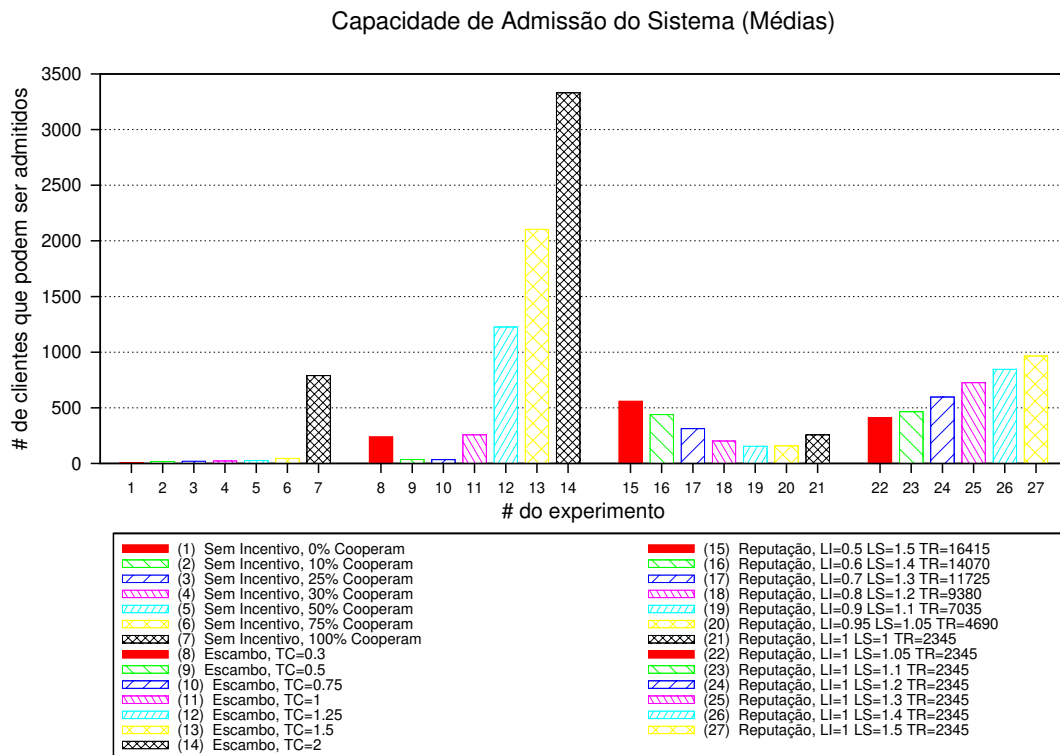


Figura 6.13: Médias das capacidades de admissão do sistema, para cada experimento do cenário *24h*.

Já o experimento 6, que não apresenta bloqueio, tem a métrica capacidade de admissão com

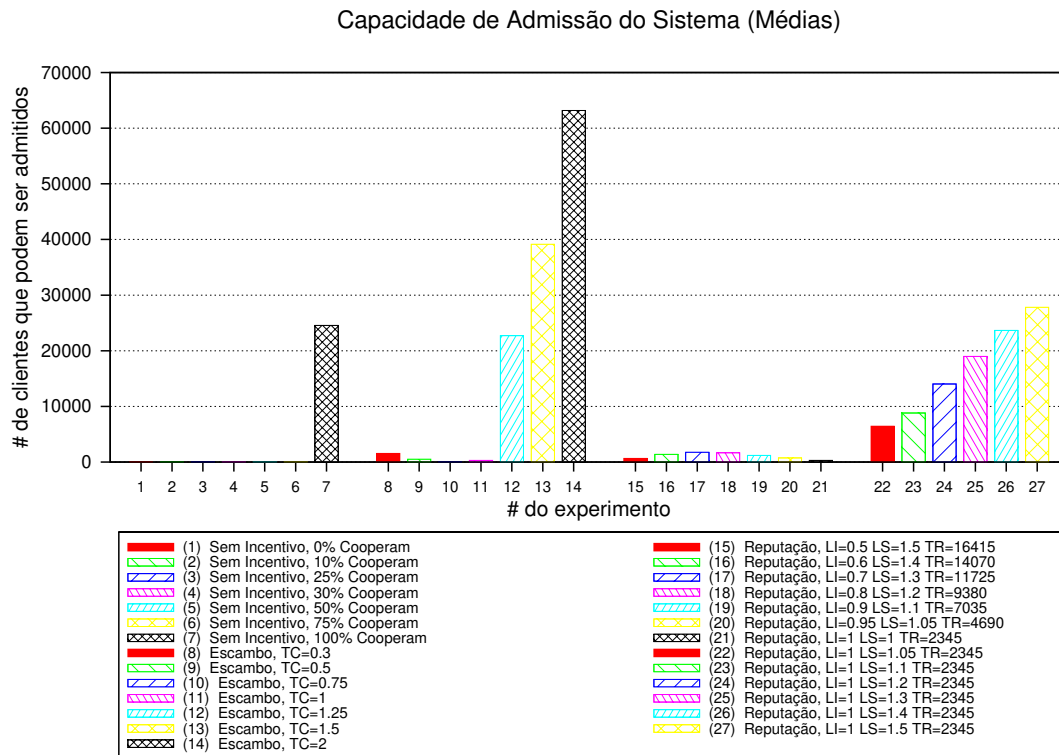


Figura 6.14: Médias das capacidades de admissão do sistema, para cada experimento do cenário *5mFC*.

valores muito baixos, no cenário *24h*, e praticamente zerados, no cenário *5mFC*. Isso significa que ou todos os recursos do sistema são utilizados eficientemente, sem falta ou excesso, ou falta recursos. Nas Figuras 6.9 e 6.10, pode ser observado, a partir da comparação dos experimentos 6 e 7, que as qualidades recebidas no primeiro poderiam ser maiores se a percentagem de cooperação fosse superior, como ocorreu no segundo. Dessa forma, pode-se concluir que falta recursos no experimento 6 e, conseqüentemente, que somente com 100% de clientes cooperando no primeiro grupo é que se tem sobra de recursos no sistema. Assim, para atender o tráfego gerado em ambos os cenários sem nenhum mecanismo de incentivo (primeiro grupo de experimentos), precisar-se-ia de uma percentagem de clientes que cooperam superior a 75% e inferior 100%, a qual não condiz com a realidade, de qualquer forma. Esta conclusão justifica a necessidade de se utilizar pelo menos o incentivo primário do mecanismo proposto, como forma de se aumentar a escalabilidade da rede *peer-to-peer*.

Analisando agora o segundo grupo de experimentos (de 8 a 14), pode-se novamente comprovar a discussão da taxa de cooperação com valor 1 no ponto de equilíbrio entre a demanda e a oferta de recursos no sistema. Neste ponto (experimento 11), a capacidade de admissão

é quase zero, tendo valores que tendem a aumentar com taxas de cooperação maiores que 1 (experimentos de 12 a 14), e valores que tendem a zerar com taxas de cooperação menores que 1 (experimentos de 8 a 10). Em particular, o experimento 8, em ambos os cenários, e o experimento 9, apenas no cenário *5mFC*, apresentam capacidades de admissão superiores àquela do experimento 11. Voltando novamente às Figuras 6.9 e 6.10, pode-se constatar que este excedente de recursos aparente ocorreu em virtude das baixas qualidades recebidas, o que também evidencia falta de recursos, ou seja, ausência de escalabilidade do sistema. Pode-se concluir, portanto, que com a utilização de apenas o incentivo primário do mecanismo proposto (segundo grupo de experimentos), a melhor taxa de cooperação a ser adotada é a de valor 1.

Analisando, finalmente, o terceiro grupo de experimentos (de 15 a 27), novas observações podem ser tecidas. Inicialmente, os experimentos de 22 a 27 (que fixam o limite inferior da taxa de cooperação em 1 e variam o superior de forma a se analisar o sistema sem a ocorrência de nenhum deficit de recursos) apresentam sobra de capacidade de admissão, o que significa que se exige dos clientes mais cooperação do que o necessário. Conforme mencionado, não se obteve benefícios com esta abordagem em termos de diminuição do custo de utilização do mecanismo proposto. Dessa forma, estes resultados somente reafirmam o que já foi concluído: que estas configurações de limites inferiores e superiores não são interessantes de serem utilizadas.

Nos demais experimentos do terceiro grupo (de 15 a 21), nos quais os limites inferiores e superiores são equidistantes do ponto de equilíbrio 1, pode-se constatar que ocorre uma utilização eficiente dos recursos acumulados no sistema. Além das qualidades recebidas serem boas (Figuras 6.9 e 6.10), aparentemente não sobra e nem falta recursos na maioria dos casos, o que pode ser notado a partir da comparação destes experimentos com o experimento 11 (utilização ideal dos recursos), tanto no cenário *24h* quanto no cenário *5mFC* (não obstante a utilização no segundo ser um pouco melhor).

Apesar desta utilização eficiente de recursos, há no terceiro grupo de experimentos o custo de utilização do mecanismo proposto, que reduz a qualidade recebida de 2 a 4 descritores. Além disso, outras métricas devem ser analisadas para se investigar os benefícios obtidos com o aumento dos tempos de permanência dos clientes. Pode-se concluir apenas que a utilização do incentivo primário é essencial no contexto da aplicação estudada, considerando os valores reais de altruísmo, fornecidos pela literatura, quando nenhum mecanismo de incentivo é utilizado.

6.3.3 Benefícios do Incentivo Secundário do Mecanismo Proposto

O principal objetivo do incentivo secundário do mecanismo proposto é estimular um aumento dos tempos de permanência dos clientes na rede *peer-to-peer*, a partir da diminuição gradativa das taxas de cooperação, de forma a minimizar as taxas de rupturas nas árvores de distribuição, que causam interrupções temporárias no recebimento dos descritores. A vantagem desta melhoria está na obtenção de fluxos com qualidades mais estáveis, ou seja, com número de descritores

recebidos o mais constante possível na maior parte do tempo. Para se comprovar este benefício, a análise da métrica taxa de rupturas em árvores é feita.

As Figuras 6.15 e 6.16 apresentam, respectivamente para os cenários *24h* e *5mFC*, as médias das taxas de rupturas em árvores sofridas pelos clientes, para cada experimento destes cenários. Analisando inicialmente o cenário *24h*, nota-se que os valores obtidos nas situações normais de tráfego (sem *flash crowd*) não são muito diferentes entre os experimentos considerados, ou seja, com poucos clientes chegando e saindo do sistema, o aumento dos tempos de permanência deles não afeta muito as rupturas em árvores de distribuição. Pode-se ter uma idéia deste baixo ganho obtido comparando os experimentos 11 e 21, que apenas têm este aumento de tempo de permanência como diferença entre si (ambos possuem a taxa de cooperação fixada em 1).

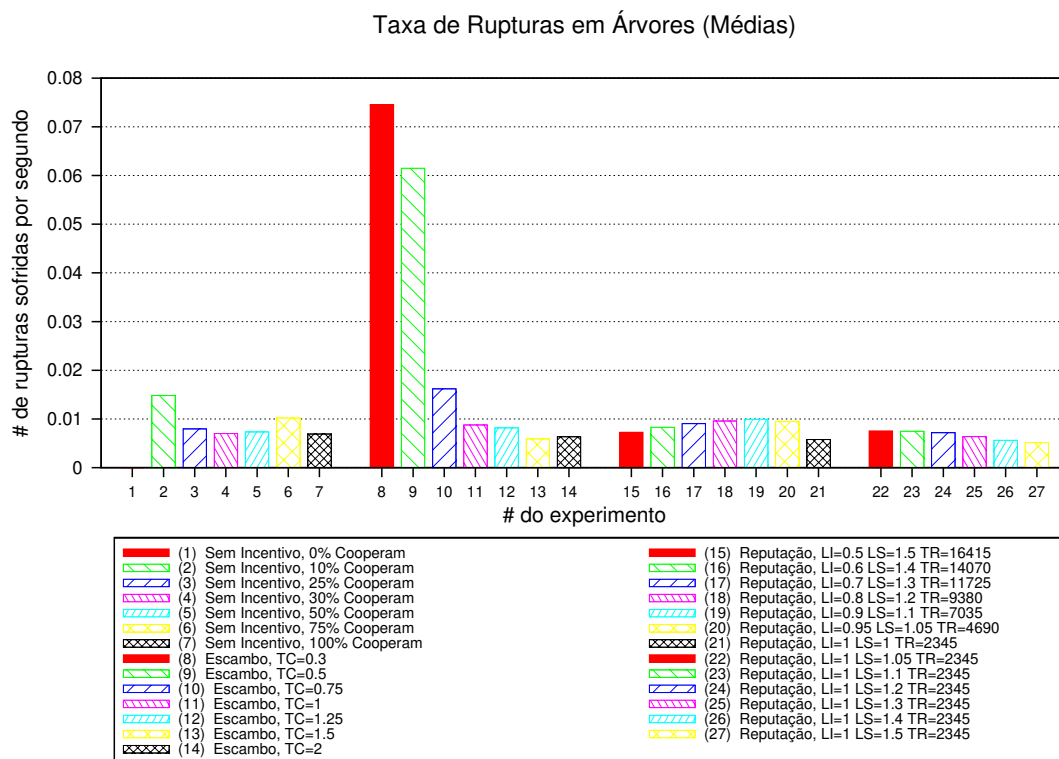


Figura 6.15: Médias das taxas de rupturas em árvores sofridas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *24h*.

Em ambos os cenários, os experimentos 8 e 9 apresentam taxas de rupturas mais altas em virtude de suas baixas taxas de cooperação. Devido a estas últimas, cada cliente admite menos nós abaixo de si, o que torna as árvores mais estreitas e profundas; conseqüentemente, a taxa de rupturas aumenta (as chances de um determinado cliente sofrer rupturas aumentam, pois basta qualquer nó superior a ele se desconectar do sistema para que isto ocorra). Por outro

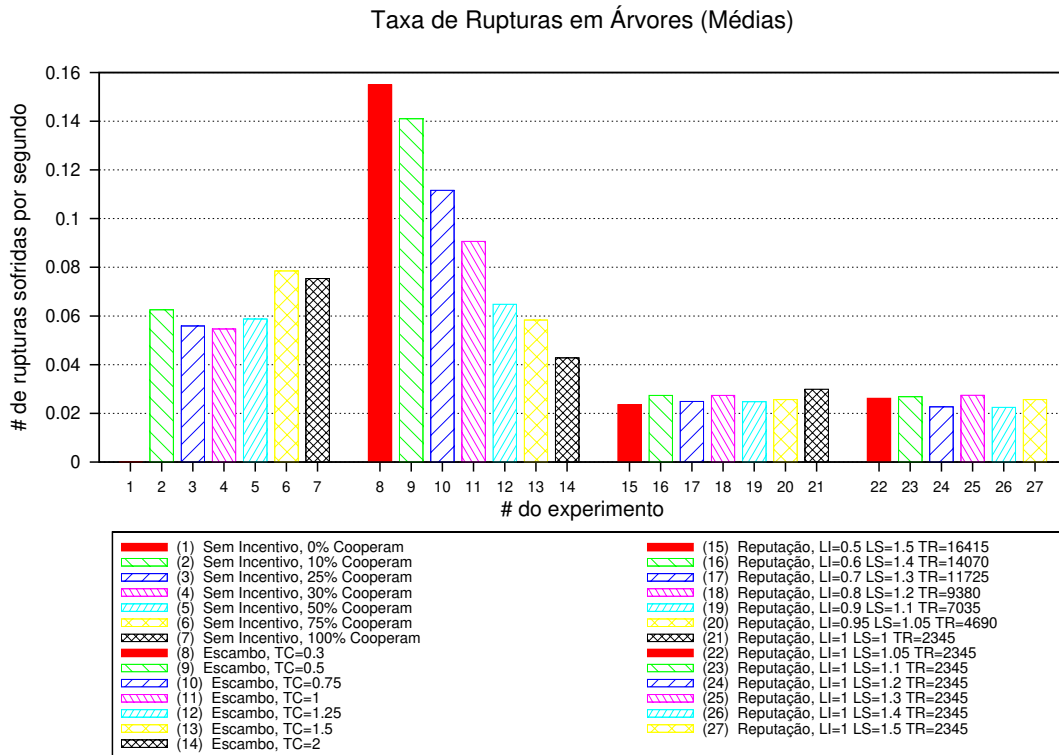


Figura 6.16: Médias das taxas de rupturas em árvores sofridas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mFC*.

lado, nenhuma ruptura em árvore é contabilizada no experimento 1. Isso ocorre porque com 0% de clientes cooperando, apenas o servidor admite nós abaixo de si, e este fica disponível durante todo o período de simulação (não ocorrendo nenhuma ruptura). Tem-se, portanto, que esta métrica não é relevante quando o sistema degenera para a arquitetura cliente-servidor.

Analisando agora somente o cenário *5mFC*, percebe-se que as situações de *flash crowd* têm grande impacto na métrica em questão. Diferentemente do que ocorre no cenário *24h*, nota-se aqui uma sensível queda na taxa de rupturas, quando o tempo de permanência é aumentado (experimentos de 15 a 27). Em particular, comparando novamente os experimentos 11 e 21, pode-se constatar uma queda de 67% na taxa de rupturas, ou seja, de 0.09 (uma ruptura a cada 11 segundos) para 0.03 (uma ruptura a cada 33 segundos). Além disso, comparando o experimento 21 com os demais do terceiro grupo (de 15 a 27), tem-se que, apesar das diferenças nas qualidades recebidas, conforme visto na Figura 6.10, a taxa de rupturas pouco varia em função dos diferentes limites inferiores e superiores adotados. Dessa forma, pode-se afirmar que todos os experimentos deste grupo apresentam benefícios similares, devido ao aumento dos tempos de permanência dos clientes.

Observando os experimentos do segundo grupo (de 8 a 14), nos quais não se tem o tempo extra de permanência, nota-se que a métrica em questão apresenta valores inversamente proporcionais às taxas de cooperação. Pode-se explicar este fato da mesma forma que se analisou a discrepância dos experimentos 8 e 9 em ambos os cenários: quanto menores as taxas de cooperação, tem-se árvores mais estreitas e profundas, o que aumenta as taxas de rupturas. Por este motivo, usa-se apenas o experimento 11 do segundo grupo como ponto de referência para se comparar o ganho obtido com os experimentos do terceiro grupo: ele é o que maximiza as qualidades recebidas, bem como o que apresenta árvores com larguras e profundidades equivalentes às formadas pelos experimentos do terceiro grupo, nos quais as médias das taxas de cooperação (dadas pelos limites inferiores e superiores) se aproximam de 1.

Analisando os experimentos do primeiro grupo (de 1 a 7), pode-se constatar mais alguns fatos. Primeiramente, os experimentos de 2 a 5 apresentam taxas de rupturas menores que os de 6 a 7 devido aos bloqueios neles ocorridos, o que diminui o número de clientes conectados e, conseqüentemente, o tamanho das árvores. Logo, os experimentos 6 e 7 são dois bons representantes para a métrica em questão no primeiro grupo. Nota-se que as taxas de rupturas nestes dois casos estão próximas daquela constatada no experimento 11. Uma possível explicação para o fato delas serem um pouco menores é a seguinte: sem o incentivo primário do mecanismo proposto, com taxa de cooperação 1, alguns clientes cooperam com mais de 128 Kbps, que é o tamanho total do fluxo distribuído; conseqüentemente, árvores mais largas e menos profundas se formam nestes casos, diminuindo um pouco a taxa de rupturas. De qualquer forma, o ganho na minimização da taxa de rupturas é notável para o terceiro grupo, o que se observa tanto em comparação aos experimentos 6 e 7, do primeiro grupo, quanto em comparação ao experimento 11, do segundo grupo.

Resta a seguinte dúvida: poderiam os resultados obtidos no gráfico da Figura 6.16 terem sido favorecidos pelo curto tempo de simulação executado (cinco minutos)? Conforme mencionou-se anteriormente, a execução das simulações com altas taxas de chegada (como as que se tem nas situações de *flash crowd*) e em longos períodos de tempo inviabilizaria o estudo de diversas combinações de experimentos em tempo hábil, devido ao tempo de processamento demandado. Conseqüentemente, optou-se por reduzir o tempo de simulação para cinco minutos, o que foi realizado nos cenários com prefixo *5m*. Não obstante, tem-se que o tempo médio de permanência dos clientes no sistema é 2345 segundos (distribuição *lognormal* com parâmetros $\mu = 5.74$ e $\sigma = 2.01$). E ainda, quando os clientes ficam mais tempo conectados – em decorrência do uso do incentivo secundário –, tem-se um tempo extra de permanência que é somado ao tempo original. Este tempo adicional também segue uma *lognormal* com os mesmos parâmetros, tendo a mesma média. O tempo original já é bem superior aos cinco minutos de simulação, mas o tempo extra de permanência poderia fazer com que uma porção ainda maior de clientes não desconectasse até o término do período. Dessa forma, as rupturas relacionadas a essas desconexões não seriam contabilizadas, afetando a métrica estudada.

Para se investigar essa tendência, criou-se um cenário hipotético (cujos parâmetros não mais condizem com a realidade), mas que poderia desmascarar o efeito descrito. A única modificação realizada neste novo cenário, em comparação com o *5mFC*, foi com relação aos tempos de permanência dos clientes (manteve-se todos os demais parâmetros, inclusive a taxa de chegada e o tempo de simulação). Dessa maneira, os parâmetros das distribuições *lognormal* – a do tempo original e a do tempo extra a ser adicionado – foram modificados, de forma que suas novas médias fossem 30 segundos (um décimo do tempo de simulação considerado). Com isso, espera-se que a maior parte dos clientes, com o tempo extra de permanência utilizado ou não, possa desconectar antes do término da simulação, sendo as rupturas decorrentes de suas saídas contabilizadas. Para isso, fixou-se $\sigma = 1.44$ e calculou-se o novo μ de forma que a média fosse 30 segundos, sendo obtido $\mu = 2.36$, ou seja, o novo cenário com sessões curtas, denominado *5mFCSC*, apenas difere de *5mFC* em $\mu = 2.36$ e $\sigma = 1.44$ para ambas as distribuições *lognormal* utilizadas. O valor $\sigma = 1.44$ também foi obtido de [57], só que dessa vez considerando programações ao vivo de *reality show* ao invés de noticiários e esportes, já que este outro tipo de programação apresenta sessões mais curtas.

Surpreendentemente, as métricas analisadas anteriormente, qualidade média de fluxo recebida, capacidade de admissão do sistema e probabilidade de bloqueio do sistema são, no novo cenário, muito parecidas com o que se tem no cenário *5mFC*. Já a métrica taxa de rupturas em árvores sofre alterações, conforme o esperado. A Figura 6.17 apresenta as novas médias desta métrica, para cada experimento do cenário *5mFCSC*. Comparando a escala das ordenadas deste gráfico com a do gráfico anterior (cenário *5mFC*), pode-se constatar que as taxas de rupturas aumentam bastante, em decorrência dos novos tempos de permanência dos clientes.

O primeiro ponto a ser observado, a partir desse gráfico, é que de fato a diminuição das rupturas em árvores, nos experimentos de 15 a 27, é mais discreta em comparação à do cenário anterior. Não obstante, ela ainda é significativa: 45%, contra os 67% constatados anteriormente, utilizando novamente os experimentos 11 e 21 como base de cálculo. Intuitivamente, esta nova porcentagem é mais condizente com o aumento dos tempos de permanência provocado no terceiro grupo de experimentos: dobra-se o mesmo na média, ao se utilizar a segunda *lognormal* com os mesmos parâmetros da primeira; conseqüentemente, a taxa de rupturas cai quase pela metade. Uma possível explicação para esta queda não ter sido exatamente a metade é a existência da classe de conexão discada, cujos clientes nunca ficam mais tempo conectados, ou seja, são menos de 100% dos clientes que têm seus tempos de permanência dobrados, na média, para os experimentos do terceiro grupo.

Outro ponto a ser observado é que existe uma pequena variação entre as taxas de rupturas dos experimentos do terceiro grupo (de 15 a 27), ao contrário do que ocorre no cenário anterior, onde estes valores são praticamente os mesmos entre si. Apesar desta variação ser pequena, ela reflete as diferentes qualidades recebidas em cada experimento, como conseqüência do custo de utilização do mecanismo proposto, ou seja, quanto maiores as qualidades recebidas, mais

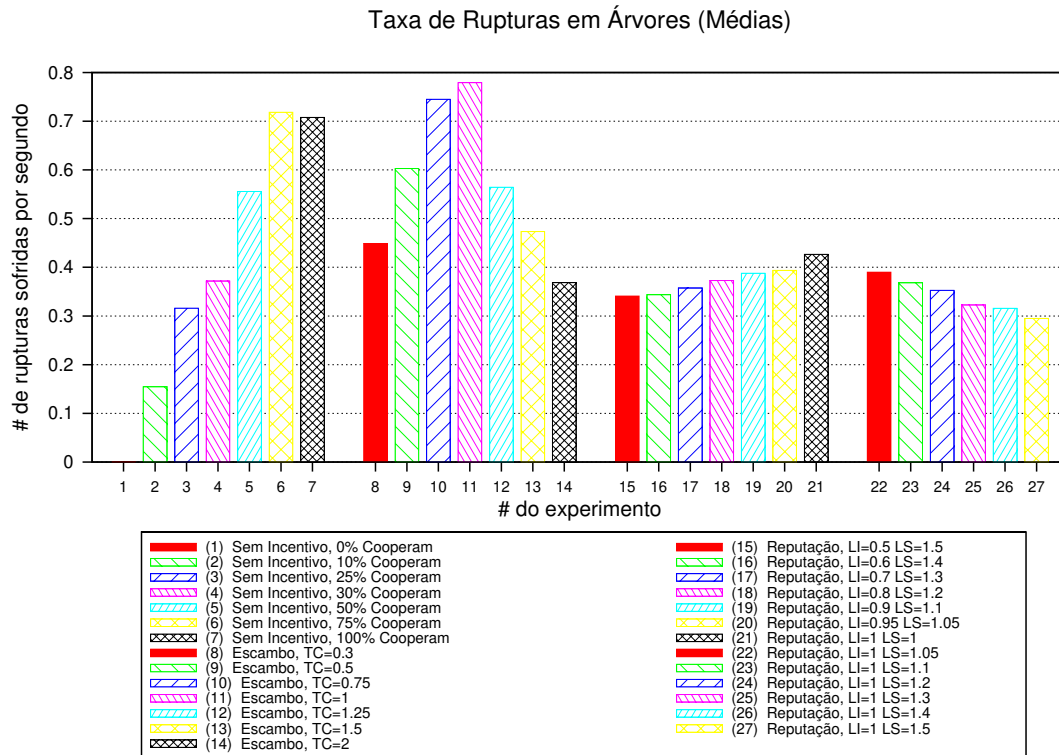


Figura 6.17: Médias das taxas de rupturas em árvores sofridas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mFCSC*.

clientes admite-se nas diferentes árvores; estas, sendo maiores, geram mais ocorrências de rupturas. De forma conservadora, utiliza-se o experimento 21 como base de cálculo, já que este é o experimento que apresenta a menor queda na taxa de rupturas.

Um terceiro ponto a ser notado, agora no segundo grupo de experimentos (de 8 a 14), é que as rupturas não são mais inversamente proporcionais à taxa de cooperação. Taxas de cooperação menores que 1 geram menos rupturas que com este valor. Acredita-se que isso seja um efeito direto da diminuição dos tempos de permanência. Ficando os clientes menos tempo conectados, tem-se que as árvores não crescem tanto a ponto de ficarem tão profundas em virtude de suas pequenas larguras (por causa das taxas de cooperação).

Apesar das diferenças discutidas até aqui, as taxas de rupturas em árvores dos experimentos 6 e 7 (primeiro grupo) e a do experimento 11 (segundo grupo) continuam próximas, assim como foi observa-se no cenário anterior. Dessa forma, a queda de 45% constatada no presente cenário pode ser entendida como o ganho aproximado dos experimentos com tempo de permanência aumentado (de 15 a 27) sobre todos os demais, tanto os do primeiro quanto os do segundo grupo (de 1 a 14). E ainda, apesar de ser obtida em um cenário hipotético, ela será preferida à

anterior (67%), uma vez que se faz mais pessimista e, portanto, não tendenciosa. Logo, pode-se dizer que a queda constatada na taxa de rupturas em árvores é de pelo menos 45%.

A alteração realizada nos tempos de permanência do cenário *5mFC*, que gerou o cenário hipotético *5mFCSC*, introduz um novo fato: antes dela, os números médios de clientes no sistema para os casos com e sem o tempo extra de permanência eram, respectivamente, 11000 e 9000, aproximadamente; após a alteração, estas médias caíram, respectivamente, para 3250 e 1750, aproximadamente. Essa diferença pode ser compreendida intuitivamente: ficando os clientes menos tempo conectados, o sistema retém um menor número deles no ponto de equilíbrio; logo, as médias são menores. A questão é que, com o número de árvores utilizadas, 16, esses dois conjuntos de valores podem induzir diferentes profundidades de árvores, possivelmente alterando a métrica analisada.

Para se ter uma idéia das profundidades induzidas por diferentes números de clientes, um cálculo aproximado pode ser feito, não considerando a heterogeneidade dos mesmos. Pode-se assumir que todos eles admitem igualmente 16 nós abaixo de si, nas árvores onde são nós interiores. Sabe-se que cada um é nó interior em uma única árvore e nó folha nas $n - 1$ demais árvores; dessa forma, consumindo a banda máxima de fluxo, que é 16 descritores, devem admitir 16 nós filhos naquela árvore em que são nós interiores, considerando uma taxa de cooperação igual a 1. Uma árvore completa de grau 16 tem 17 nós até o primeiro nível ($16^0 + 16^1$), 273 até o segundo ($16^0 + 16^1 + 16^2$), 4369 até o terceiro ($16^0 + 16^1 + 16^2 + 16^3$) e 69905 até o quarto ($16^0 + 16^1 + 16^2 + 16^3 + 16^4$), ou seja, com 11000 e 9000 clientes, tem-se árvores com profundidades de até 4; já com 3250 e 1750 clientes, as profundidades estimadas não passam de 3.

Para se investigar os efeitos de diferentes profundidades induzidas na métrica taxa de rupturas em árvores, criou-se os cenários *5mFC32A* e *5mFCSC32A*, que são exatamente os cenários *5mFC* e *5mFCSC* com 32 árvores de distribuição, ao invés de 16. Uma árvore completa de grau 32 tem 33 nós até o primeiro nível ($32^0 + 32^1$), 1057 até o segundo ($32^0 + 32^1 + 32^2$) e 33825 até o terceiro ($32^0 + 32^1 + 32^2 + 32^3$). Dessa forma, tanto 11000 e 9000 quanto 3250 e 1750 clientes induzirão árvores com profundidades estimadas de até 3. Com isso, pode-se ter uma idéia se a diferença nas profundidades das árvores, provocada pela modificação nos tempos de permanência dos clientes nos cenários *5mFC* e *5mFCSC*, afeta as conclusões tomadas até então.

As Figuras 6.18 e 6.19 apresentam, respectivamente para os cenários *5mFC32A* e *5mFCSC32A*, as médias das taxas de rupturas em árvores sofridas pelos clientes, para cada experimento destes cenários. Comparando os experimentos 11 e 21 de cada cenário, enquanto a queda no primeiro é de 69%, no segundo é de 43%. Estas percentagens são praticamente as mesmas constatadas nos cenários *5mFC* e *5mFCSC*, ou seja, 67% e 45%. Além disso, os valores desta métrica para os experimentos 6 e 7 (primeiro grupo) e 11 (segundo grupo) são próximos, indicando que o ganho de 43% obtido no segundo cenário com o aumento dos tempos de permanência (terceiro grupo) pode ser considerado como relativo a todos os demais experi-

mentos (conforme já mencionado, utiliza-se o experimento 11 como ponto de referência para o segundo grupo e os experimentos 6 e 7 como pontos de referência para o primeiro grupo).

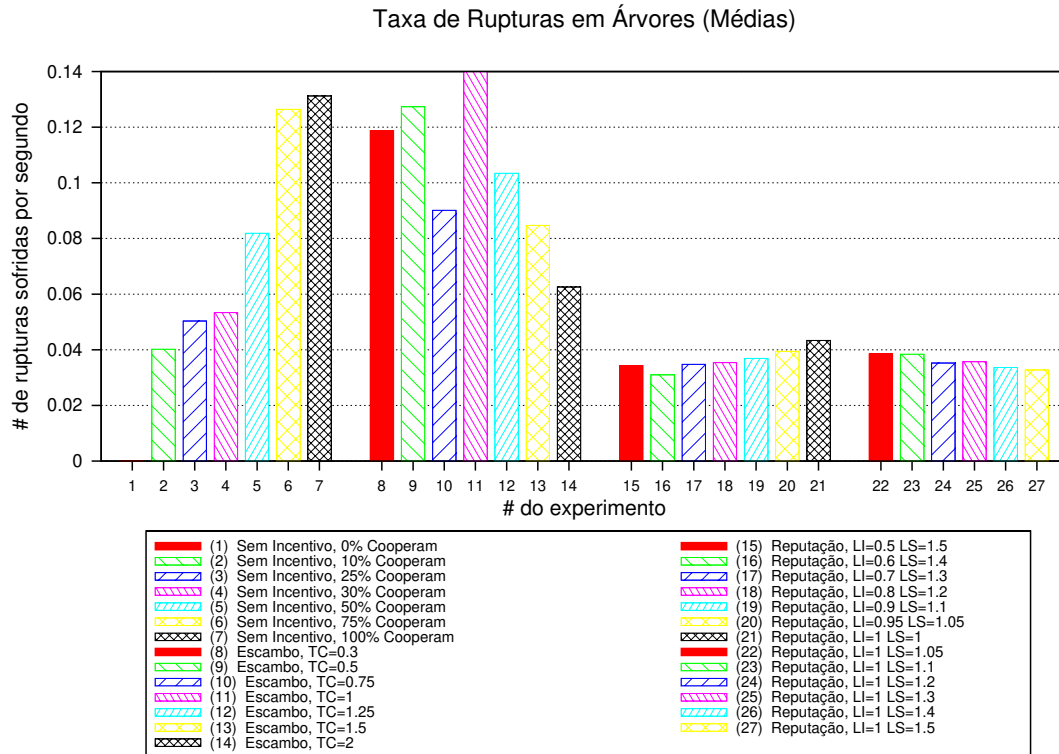


Figura 6.18: Médias das taxas de rupturas em árvores sofridas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mFC32A*.

As métricas qualidade média de fluxo recebida, capacidade de admissão do sistema e probabilidade de bloqueio do sistema também apresentam resultados similares com 32 árvores. Logo, pode-se concluir que a diferença nas profundidades das árvores, causada pelas modificações nos tempos de permanência dos clientes nos cenários anteriores, não afeta os resultados obtidos com relação à queda constatada do número de rupturas. De maneira geral, pode-se dizer que, dobrando o tempo de permanência dos clientes, na média, consegue-se reduzir o número de rupturas em árvores aproximadamente pela metade.

6.4 Considerações Finais

Neste capítulo, descreveu-se os resultados obtidos nas simulações executadas para a avaliação do mecanismo de incentivo proposto no Capítulo 4. Primeiramente, apresentou-se todas

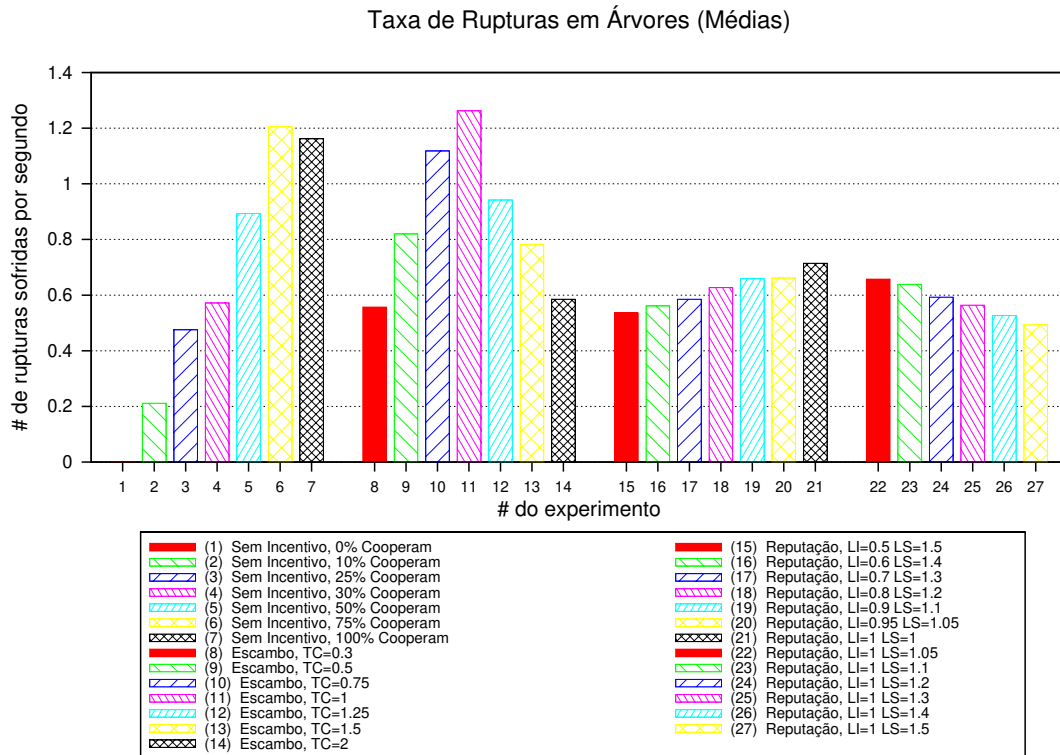


Figura 6.19: Médias das taxas de rupturas em árvores sofridas pelos clientes durante suas permanências no sistema, para cada experimento do cenário *5mFCSC32A*.

as métricas coletadas e analisadas. Em seguida, realizou-se um estudo sobre a determinação dos parâmetros *LimiteInferiorTaxaCooperacao*, *LimiteSuperiorTaxaCooperacao* e *TempoRequeridoCooperadorEstavel*, responsáveis pelo funcionamento do incentivo secundário. Após a seleção dos melhores valores, o sistema foi avaliado tanto em condições normais de tráfego quanto em situações de *flash crowd*. Nesses cenários, comparou-se o desempenho de diferentes métricas para os seguintes grupos de experimentos: sem incentivo, somente com o incentivo primário e com os incentivos primário e secundário. Em particular, analisou-se os benefícios obtidos a partir da utilização de cada um dos incentivos.

Foi mencionado que, segundo a literatura, apenas 30% dos clientes cooperam quando nenhum mecanismo de incentivo é utilizado. Conforme apontado pelos resultados das simulações, tal nível de altruísmo não é suficiente para se prover uma qualidade de fluxo de nem a metade da ideal. Além disso, as probabilidades de bloqueio mostraram que nem todos os clientes que chegam são atendidos, mesmo no cenário sem *flash crowd*, o que revela a falta de escalabilidade do sistema quando poucos cooperam. Para se obter uma qualidade de serviço aceitável, seria necessário que pelo menos 50% dos clientes cooperassem sem incentivo, o que não condiz com

a realidade [1].

Já com a utilização do incentivo primário, todos os clientes cooperam e, conseqüentemente, nenhum bloqueio ocorreu, bem como as qualidades recebidas foram maximizadas. Conseguiu-se níveis de qualidade de quase o dobro, para os casos sem *flash crowd*, e de mais que o dobro, para os casos com *flash crowd*. Nestes últimos, o mecanismo mostrou-se mais eficiente. Dessa forma, com relação ao incentivo primário, conclui-se que o seu uso se faz essencial para viabilizar os serviços de vídeo sob demanda nas redes *peer-to-peer*, de forma que a escalabilidade do sistema seja aumentada e as qualidades recebidas melhoradas. Neste contexto, quando todos os clientes cooperam, deve-se exigir deles exatamente a mesma quantidade de banda que consomem, de forma que a oferta e a demanda de recursos na rede fiquem em equilíbrio (taxa de cooperação igual a 1).

Com relação ao incentivo secundário, constatou-se, primeiramente, que ele apresenta um custo de utilização na qualidade recebida, que varia de 2 a 4 descritores em 16. Essa variação depende da escolha dos parâmetros que maximizam o seu desempenho, listados na Tabela 6.4. Em seguida, mostrou-se que os benefícios obtidos com a sua utilização, em termos da diminuição das taxas de rupturas em árvores, são notáveis apenas nas situações de *flash crowd*, nas quais as taxas de chegada são altas. Nestes casos, conseguiu-se reduzir as taxas de rupturas em pelo menos 45%, dobrando-se os tempos de permanência dos clientes na média. De forma geral, conclui-se que a utilização do incentivo secundário somente é vantajosa quando as taxas de chegada forem suficientemente altas a ponto do ganho obtido compensar o custo incorrido, como foi o caso do cenário com *flash crowd*. Assim, quando as rupturas em árvores forem numerosas o suficiente para provocarem distorções na reprodução do fluxo recebido, então o incentivo secundário oferecerá uma boa solução ao problema, fazendo com que os clientes recebam um fluxo que, apesar de ter menor qualidade (2 a 4 descritores a menos), seja mais estável.

Capítulo 7

Conclusão

As redes *peer-to-peer* são cada vez mais consideradas nos serviços de vídeo sob demanda. Elas contornam o problema de escalabilidade imposto pela arquitetura cliente-servidor, uma vez que exploram os recursos computacionais das estações clientes. Estes recursos – largura de banda, processamento e memória – são cada vez mais abundantes nas estações clientes em decorrência do aparecimento de novas tecnologias de acesso à Internet, tais como DSLs e HFCs, bem como em decorrência do desenvolvimento tecnológico, que reduziu a distância em termos de hardware entre os servidores de conteúdo e os computadores pessoais. Conseqüentemente, as redes *peer-to-peer* apresentam uma solução de baixo custo para a provisão de vídeo, se comparadas à arquitetura cliente-servidor. Além disso, permitem a implementação de *multicast* na camada de aplicação, o que é uma característica desejável por dois motivos: (1) *multicast* na camada de rede não é suficientemente difundido, principalmente nas redes que abrangem diferentes domínios; e (2) serviços que demandam muita banda de rede, como é caso dos de vídeo sob demanda, podem se beneficiar com o uso de *multicast* em distribuições sincronizadas.

Não obstante, a distribuição de vídeo em redes *peer-to-peer* também apresenta desafios. Um dos problemas encontrados é o comportamento egoísta dos usuários, que procuram tirar o máximo de proveito dos recursos da rede, cooperando com o mínimo na distribuição dos fluxos. Para se ter uma idéia, estudos realizados em redes *peer-to-peer* para serviços de compartilhamento de arquivos mostraram que menos de 30% de clientes altruístas sustentam a maioria de todas as requisições da rede. Um outro problema encontrado é o curto tempo de permanência dos clientes no sistema, o que demanda esforços para se contornar as interrupções causadas nas entregas de fluxos com sensibilidade temporal quando as desconexões ocorrem. Tal problema é particularmente importante em *multicast* na camada de aplicação, pois dependendo da posição de um nó nas árvores de distribuição, diversos clientes podem ser afetados quando ele se desconecta da rede. Faz-se necessário, então, o uso de mecanismos de incentivo que estimulem os clientes a cooperarem e também a permanecerem conectados durante intervalos de tempo maiores, de forma a reduzir os problemas de desconexão.

O objetivo do trabalho descrito nesta dissertação foi propor e avaliar um novo mecanismo de incentivo para sistemas *peer-to-peer* de fluxo de mídia ao vivo (com distribuição sincronizada), a fim de tornar o sistema escalável, maximizar a qualidade recebida, bem como amenizar os problemas decorrentes das rupturas em árvores de distribuição. A seguir, descreve-se as principais contribuições alcançadas e algumas direções de trabalhos futuros identificadas.

7.1 Contribuições

Dentre as principais contribuições deste trabalho, pode-se destacar:

- Uma revisão bibliográfica sobre mecanismos de incentivo à cooperação, com enfoque para o projeto dos mesmos;
- Proposta e avaliação de um novo mecanismo de incentivo para sistemas *peer-to-peer* de fluxo de mídia ao vivo (com distribuição sincronizada);
- Proposta de integração do novo mecanismo de incentivo com o sistema CoopNet, através da modificação do protocolo de adaptação de banda descrito em [44];
- Desenvolvimento de um simulador de eventos discretos para redes *peer-to-peer* de fluxo de mídia ao vivo.

O mecanismo de incentivo proposto foi elaborado a partir do estudo realizado no Capítulo 3, sobre os padrões de incentivo à cooperação existentes, e no Capítulo 2, sobre os sistemas *peer-to-peer* de fluxo de mídia ao vivo. Considerou-se duas recomendações mencionadas no Capítulo 3: (1) combinar diferentes padrões de incentivo na definição de novos mecanismos, de forma a reunir as vantagens e compensar as deficiências de cada um; e (2) levar em consideração as características específicas do tipo de aplicação para a qual o mecanismo é projetado, de maneira a escolher os padrões de incentivo mais apropriados.

Sendo assim, o mecanismo proposto é composto por duas partes, o incentivo primário e o secundário. O incentivo primário buscou aumentar a escalabilidade do sistema, bem como maximizar as qualidades de fluxo recebidas pelos clientes. O incentivo secundário buscou amenizar os problemas decorrentes das rupturas em árvores de distribuição, causadas pelo curto tempo de permanência dos clientes no sistema. Para a integração do mecanismo proposto com o sistema CoopNet, sugeriu-se uma modificação no protocolo de adaptação de banda apresentado em [44], a qual foi implementada no simulador desenvolvido.

O simulador foi projetado de forma a avaliar a eficácia do mecanismo proposto. Trata-se de um simulador de eventos discretos para redes *peer-to-peer* de fluxo de mídia ao vivo. Considerou-se a proposta descrita em [43, 44], na qual múltiplas árvores de distribuição e codificação em múltiplos descritores são usadas como forma de se prover robustez e adaptabilidade

à heterogeneidade dos clientes. Utilizou-se o protocolo de admissão *top-down* também proposto em [43, 44], bem como o novo protocolo de adaptação de banda descrito na Seção 4.2. O simulador foi desenvolvido no paradigma de orientação a objetos, o que facilita a sua modificação para ser reutilizado para outros tipos de sistemas relacionados.

A avaliação do mecanismo proposto demonstrou que a percentagem de clientes cooperativos indicada pela literatura quando nenhum incentivo é utilizado (30%) não é suficiente para se prover uma qualidade de serviço aceitável. Além disso, viu-se que, neste caso, nem todos os clientes que chegam são atendidos, o que revela a falta de escalabilidade do sistema quando poucos cooperam. Com a utilização do incentivo primário, no entanto, todos os clientes cooperaram e, conseqüentemente, nenhum bloqueio ocorreu, bem como as qualidades recebidas foram maximizadas. Dessa forma, com relação ao incentivo primário, conclui-se que o seu uso se faz essencial para viabilizar os serviços de vídeo sob demanda em redes *peer-to-peer*, de forma que a escalabilidade do sistema seja aumentada e as qualidades recebidas melhoradas.

Com relação ao incentivo secundário, constatou-se, primeiramente, que ele apresenta um custo de utilização na qualidade recebida. Em seguida, verificou-se que os benefícios obtidos com ele, em termos da diminuição das taxas de rupturas em árvores, são notáveis apenas nas situações de *flash crowd*, nas quais as taxas de chegada são altas. Sendo assim, conclui-se que o seu uso somente é vantajoso quando as taxas de chegada forem suficientemente altas a ponto do ganho obtido compensar o custo incorrido, ou seja, quando as rupturas em árvores forem numerosas o suficiente para provocarem distorções na reprodução do fluxo recebido, então o incentivo secundário oferecerá uma boa solução ao problema, fazendo com que os clientes recebam um fluxo que, apesar de ter menor qualidade, seja mais estável.

Uma outra possível utilização para o incentivo secundário seria o propósito econômico de se manter os clientes mais tempo conectados no sistema. Assim como ocorre na televisão analógica gratuita, quem paga os custos de distribuição são os anunciantes, que sempre almejam uma maior audiência. Da mesma forma, em um sistema de distribuição de fluxo digital como o estudado, poder-se-ia, por um lado, ter os anunciantes arcando com os servidores centrais, de onde o serviço e as árvores de distribuição se originam; por outro, os clientes da rede *peer-to-peer* cooperando sempre, a partir do incentivo primário, e permanecendo mais tempo conectados, a partir do incentivo secundário. Enquanto o primeiro fator proporcionaria uma maior escalabilidade ao sistema, de maneira que mais clientes pudessem ser atendidos, o segundo satisfaria os interesses econômicos dos anunciantes, que teriam uma maior audiência e, conseqüentemente, ficariam mais motivados a continuar mantendo o serviço.

7.2 Trabalhos Futuros

Com a conclusão do presente trabalho, algumas direções de trabalhos futuros foram identificadas. Um primeiro ponto a ser aprimorado no mecanismo de incentivo proposto é a determina-

ção dinâmica dos parâmetros *LimiteInferiorTaxaCooperacao*, *LimiteSuperiorTaxaCooperacao* e *TempoRequeridoCooperadorEstavel*, de forma que o sistema se ajuste automaticamente a diferentes situações. A análise das métricas qualidade média de fluxo recebida e capacidade de admissão do sistema, feita no procedimento manual descrito, pode ser implementada através de uma monitoração constante das mesmas, permitindo um ajuste automático dos referidos parâmetros de forma a se manter o equilíbrio entre a demanda e a oferta de recursos no sistema.

Uma outra questão a ser investigada no mecanismo de incentivo proposto é a possibilidade de redução do seu custo de utilização. Conforme visto no Capítulo 6, este custo varia de 2 a 4 descritores em 16, dependendo da escolha dos parâmetros que maximizam o seu desempenho. Acredita-se que o referido custo ocorra devido à discretização das bandas cooperadas e consumidas no cálculo das mesmas, realizado pelo mecanismo proposto. Conseqüentemente, há sempre um resíduo de banda que não é utilizado por não ser suficientemente grande para acomodar um novo descritor (servido ou recebido). Além disso, quanto maior for a faixa de variação permitida à taxa de cooperação, um número maior de descritores pode ser afetado por este arredondamento. Uma boa tentativa para se amenizar este problema – e, conseqüentemente, reduzir o custo de utilização do mecanismo – seria “afrouxar” a amarração entre as bandas cooperada e consumida de cada cliente i , introduzindo uma folga de k Kbps na relação

$$B_{Si} = TC_i \cdot B_{Ei},$$

onde k é o tamanho do descritor utilizado. Dessa forma, o protocolo de adaptação de banda modificado, descrito na Seção 4.2, ficaria da seguinte maneira:

1. Descarte tráfego de entrada quando houver congestionamento no enlace de entrada *ou* quando $B_{Si} < TC_i \cdot B_{Ei} - k$;
2. Descarte tráfego de saída quando houver congestionamento no enlace de saída *ou* quando $B_{Si} > TC_i \cdot B_{Ei} + k$;
3. Adicione tráfego de entrada após um determinado período de tempo sem congestionamento no enlace de entrada *e* com $B_{Si} > TC_i \cdot B_{Ei}$;
4. Adicione tráfego de saída após um determinado período de tempo sem congestionamento no enlace de saída *e* com $B_{Si} < TC_i \cdot B_{Ei}$.

É importante notar que somente as duas primeiras operações sofrem alterações, com a utilização de k . A conseqüência disso é que o descarte de descritores fica mais seletivo, preservando bandas cooperadas e consumidas mais elevadas, enquanto que a adição de novos descritores continua como está atualmente, facilitando o aumento das mesmas.

No simulador desenvolvido, uma necessidade imediata de melhoria é a otimização de todos os métodos implementados, de forma a se diminuir o tempo de execução das simulações. Apesar de se ter utilizado a biblioteca *Standard Template Library* [51, 35] para as estruturas de dados básicas, outras estruturas tiveram implementações próprias, como foi o caso das árvores utilizadas, podendo, portanto, ser otimizadas. Além disso, procedimentos como a medição da capacidade de admissão de novos clientes no sistema, os processamentos de congestionamento nos clientes, os aumentos e diminuições de tráfego dos clientes com base no novo protocolo de adaptação de banda, os cálculos do mecanismo de incentivo utilizado e o balanceamento dos nós interiores entre as árvores de distribuição (para equilibrar suas capacidades de admissão) apresentam várias operações não triviais, podendo, também, ser otimizados. O uso de ferramentas de otimização de código seria uma boa opção, visando identificar quais métodos são os mais custosos à aplicação.

Como consequência da melhoria anterior, alguns cenários que não puderam ser simulados, por causa do longo tempo de simulação demandado, poderiam ser executados, como é o caso daquele substituído pelo cenário hipotético descrito na Subseção 6.3.3. Neste caso, ao invés de se diminuir os tempos de permanência dos clientes, como foi feito, poder-se-ia mantê-los inalterados e aumentar o tempo total de simulação executado, reproduzindo-se mais fielmente, dessa forma, a realidade. Adicionalmente, com simulações mais ágeis, poder-se-ia variar diversos outros parâmetros de simulação que foram fixados, como é o caso de *TamanhoFluxo*, *ProbabilidadeCongestionamentoSaida*, *ProbabilidadeCongestionamentoEntrada* e *ParametroMiTempoReparoRupturaArvore*, visando um estudo mais completo do mecanismo de incentivo proposto.

Por fim, com a conclusão do simulador desenvolvido, outros mecanismos de incentivo para aplicações relacionadas podem ser propostos e avaliados de forma muito mais rápida, reaproveitando-se todo o *framework* de simulação de eventos discretos e os controladores que implementam as operações do tipo de sistema considerado. Da mesma forma, a avaliação comparativa de propostas já existentes também pode ser realizada de maneira mais ágil. Dado que a principal dificuldade do presente trabalho foi justamente o desenvolvimento do simulador em questão, tem-se que o mesmo em si constitui uma contribuição, passível de ser explorada em uma grande gama de trabalhos futuros. Como exemplo de reaproveitamento direto desta contribuição, poder-se-ia avaliar comparativamente os dois primeiros trabalhos descritos na Seção 3.3, o que demandaria somente uma codificação relativamente simples.

Referências Bibliográficas

- [1] E. Adar e B. A. Huberman. Free riding on gnutella. *First Monday*, 5(10), Outubro 2000.
- [2] Ajato (TVA). <http://www.ajato.com.br/>, Dezembro 2005. (Data de Acesso).
- [3] M. Andrews, F. B. Shepherd, A. Srinivasan, P. Winkler, e F. Zane. Clustering and server selection using passive monitoring. Em *Proceedings of IEEE INFOCOM*, Junho 2002.
- [4] J. G. Apostolopoulos, T. Wong, S. J. Wee, e D. Tan. On multiple description streaming with content delivery networks. Em *Proceedings of IEEE INFOCOM*, volume 3, páginas 1736–1745, Junho 2002.
- [5] Bash. <http://www.gnu.org/software/bash/>, Janeiro 2006. (Data de Acesso).
- [6] J. D. K. Ben Y. Zhao e A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Relatório Técnico UCB/CSD-01-1141, EECS Department, University of California, Berkeley, U. C. Berkeley, Abril 2001.
- [7] Bittorrent. <http://www.bittorrent.com/>, Outubro 2005. (Data de Acesso).
- [8] J. Brassil e H. Schulzrinne. Enhancing internet streaming media with cueing protocols. Em *Proceedings of IEEE INFOCOM*, volume 1, páginas 95–103, Abril 2001.
- [9] L. Buttyán e J.-P. Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks. Relatório Técnico DSC/2001, EPFL, 2001.
- [10] L. Buttyán e J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5):579–592, 2003.
- [11] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, e A. Singh. Splitstream: High-bandwidth multicast in a cooperative environment. Em *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, páginas 298–313. ACM Press, Outubro 2003.

- [12] M. Castro, P. Druschel, A.-M. Kermarrec, e A. Rowstron. Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communication (JSAC)*, 20(8):1489–1499, Outubro 2002.
- [13] Eclipse C/C++ development tools. <http://www.eclipse.org/cdt/>, Janeiro 2006. (Data de Acesso).
- [14] P. A. Chou, H. J. Wang, e V. N. Padmanabhan. Layered multiple description coding. Em *Proceedings of the Packet Video Workshop*, Abril 2003.
- [15] B. Cohen. Incentives build robustness in bittorrent. Em *Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems*, Junho 2003.
- [16] R. Cohen, L. Katzir, e D. Raz. Scheduling algorithms for a cache pre-filling content distribution network. Em *Proceedings of IEEE INFOCOM*, Junho 2002.
- [17] Concurrent versions system. <http://www.nongnu.org/cvs/>, Janeiro 2006. (Data de Acesso).
- [18] Eclipse. <http://www.eclipse.org/>, Janeiro 2006. (Data de Acesso).
- [19] D. R. Figueiredo, M. Garetto, e D. Towsley. Exploiting mobility in ad-hoc wireless networks with incentives. Relatório Técnico 04-66, UM-CS, Julho 2004.
- [20] Z. Ge, P. Ji, e P. J. Shenoy. A demand adaptive and locality aware (DALA) streaming media server cluster architecture. Em *Proceedings of the 12th international workshop on network and operating systems support for digital audio and video (NOSSDAV)*, páginas 139–146. ACM Press, 2002.
- [21] Gnuplot. <http://www.gnuplot.info/>, Janeiro 2006. (Data de Acesso).
- [22] Gnutella. <http://www.gnutella.com/>, Dezembro 2005. (Data de Acesso).
- [23] Gnu scientific library. <http://www.gnu.org/software/gsl/>, 2005. (Data de Acesso).
- [24] M. Guo, M. H. Ammar, e E. W. Zegura. Selecting among replicated batching video-on-demand servers. Em *Proceedings of the 12th international workshop on network and operating systems support for digital audio and video (NOSSDAV)*, páginas 155–163. ACM Press, Maio 2002.
- [25] Y. Guo, K. Suh, J. F. Kurose, e D. F. Towsley. P2cast: peer-to-peer patching scheme for vod service. Em *Proceedings of the twelfth international conference on World Wide Web*, páginas 301–309. ACM Press, Maio 2003.

- [26] A. Habib e J. Chuang. Incentive mechanism for peer-to-peer media streaming. Em *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS'04)*, páginas 171–180, Junho 2004.
- [27] A. Habib e J. Chuang. Service differentiated peer selection: An incentive mechanism for peer-to-peer media streaming. *IEEE Transactions on Multimedia*, Julho 2005. (Data de Submissão).
- [28] A. Habib, J. Chuang, e M. M. Hefeeda. Do we need incentive mechanisms for peer-to-peer media streaming. *IEEE Communications*, Dezembro 2003. (Data de Submissão).
- [29] A. Hu. Video-on-demand broadcasting protocols: a comprehensive study. Em *Proceedings of IEEE INFOCOM*, volume 1, páginas 508–517, Abril 2001.
- [30] Y. hua Chu e H. Zhang. Considering altruism in peer-to-peer internet streaming broadcast. Em *Proceedings of the 14th international workshop on network and operating systems support for digital audio and video (NOSSDAV '04)*, páginas 10–15, New York, NY, USA, 2004. ACM Press.
- [31] Kazaa. <http://www.kazaa.com/>, Outubro 2005. (Data de Acesso).
- [32] F. Kon, R. H. Campbell, e K. Nahrstedt. Using dynamic configuration to manage a scalable multimedia distribution system. *Computer Communications Journal (Special Issue on QoS-Sensitive Distributed Systems and Applications)*, 24(1):105–123, Janeiro 2001. Elsevier Science Publisher.
- [33] J. F. Kurose e K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2. edição, 2002.
- [34] T. Moreton e A. Twigg. Trading in trust, tokens, and stamps. Em *Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems*, 2003.
- [35] D. R. Musser, G. J. Derge, e A. Saini. *STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library*. Addison-Wesley Professional, 2. edição, 2001.
- [36] Nedit. <http://www.nedit.org/>, Janeiro 2006. (Data de Acesso).
- [37] T.-W. J. Ngan, D. S. Wallach, e P. Druschel. Incentives-compatible peer-to-peer multicast. Em *Proceedings of the 2nd Workshop on Economics of Peer-to-Peer Systems*, 2004.

- [38] P. Obreiter, B. König-Ries, e M. Klein. Stimulating cooperative behavior of autonomous devices – an analysis of requirements and existing approaches. Em *Proceedings of the Second International Workshop on Wireless Information Systems (WIS2003)*, páginas 71–82, 2003.
- [39] P. Obreiter, B. König-Ries, e M. Klein. Stimulating cooperative behavior of autonomous devices – an analysis of requirements and existing approaches. Relatório Técnico 2003-1, Universität Karlsruhe, Faculty of Informatics, Janeiro 2003.
- [40] P. Obreiter e J. Nimis. A taxonomy of incentive patterns – the design space of incentives for cooperation. Em *Proceedings of the Second International Workshop on Agents and Peer-to-Peer Computing (AP2PC'03)*, Springer LNCS 2872, páginas 89–100, Julho 2003.
- [41] P. Obreiter e J. Nimis. A taxonomy of incentive patterns – the design space of incentives for cooperation. Relatório Técnico 2003-9, Universität Karlsruhe, Faculty of Informatics, Maio 2003.
- [42] V. N. Padmanabhan e K. Sripanidkulchai. The case for cooperative networking. Em *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS)*, páginas 178–190. Springer-Verlag, Março 2002.
- [43] V. N. Padmanabhan, H. J. Wang, e P. A. Chou. Resilient peer-to-peer streaming. Em *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP)*, páginas 16–27. IEEE Computer Society, Novembro 2003.
- [44] V. N. Padmanabhan, H. J. Wang, e P. A. Chou. Supporting heterogeneity and congestion control in peer-to-peer multicast streaming. Em *Proceedings of the Third International Workshop on Peer-to-Peer Systems (IPTPS)*, páginas 54–63. Springer, Fevereiro 2004.
- [45] V. N. Padmanabhan, H. J. Wang, P. A. Chou, e K. Sripanidkulchai. Distributing streaming media content using cooperative networking. Em *Proceedings of the 12th international workshop on network and operating systems support for digital audio and video (NOSS-DAV)*, páginas 177–186. ACM Press, Maio 2002.
- [46] S. Ramesh, I. Rhee, e K. Guo. Multicast with cache (mcache): An adaptive zero delay video-on-demand service. Em *Proceedings of IEEE INFOCOM*, volume 1, páginas 85–94, Abril 2001.
- [47] S. Ratnasamy, P. Francis, M. Handley, R. Karp, e S. Schenker. A scalable content-addressable network. Em *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01)*, páginas 161–172, New York, NY, USA, 2001. ACM Press.

- [48] R. Rejaie e J. Kangasharju. Mocha: a quality adaptive multimedia proxy cache for internet streaming. Em *Proceedings of NOSSDAV*, páginas 3–10, Junho 2001.
- [49] A. Rowstron e P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. Em *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, páginas 329–350, Novembro 2001.
- [50] S. D. Servetto, R. Puri, J.-P. Wagner, P. Scholtes, e M. Vetterli. Video multicast in (large) local area networks. Em *Proceedings of IEEE INFOCOM*, Junho 2002.
- [51] I. Silicon Graphics Computer Systems. Standard template library programmer’s guide. <http://www.sgi.com/tech/stl/>, Janeiro 2006. (Data de Acesso).
- [52] Speedy (Telefônica). <http://www.speedy.com.br/>, Dezembro 2005. (Data de Acesso).
- [53] K. Sripanidkulchai, B. Maggs, e H. Zhang. An analysis of live streaming workloads on the internet. Em *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (IMC '04)*, páginas 41–54. ACM Press, 2004.
- [54] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, e H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. Em *Proceedings of ACM SIGCOMM '01 Conference*, páginas 149–160, Agosto 2001.
- [55] Teleco. <http://www.teleco.com.br/>, Dezembro 2005. (Data de Acesso).
- [56] Turbo (Brasil Telecom). <http://www.brasiltelecom.com.br/turbo>, Dezembro 2005. (Data de Acesso).
- [57] E. Veloso, V. Almeida, W. Meira, A. Bestavros, e S. Jin. A hierarchical characterization of a live streaming media workload. Em *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement (IMW '02)*, páginas 117–130. ACM Press, 2002.
- [58] Velox (Telemar). <http://www.velox.com.br/>, Dezembro 2005. (Data de Acesso).
- [59] C. Venkatramani, O. Verscheure, P. Frossard, e K.-W. Lee. Optimal proxy management for multimedia streaming in content distribution networks. Em *Proceedings of the 12th international workshop on network and operating systems support for digital audio and vide (NOSSDAV)*, páginas 147–154. ACM Press, Maio 2002.
- [60] Virtua (NET). <http://www.virtua.com.br/>, Dezembro 2005. (Data de Acesso).
- [61] Vivax. <http://www.vivax.com.br/>, Dezembro 2005. (Data de Acesso).

- [62] B. Wang, S. Sen, M. Adler, e D. F. Towsley. Optimal proxy cache allocation for efficient streaming media distribution. Em *Proceedings of IEEE INFOCOM*, páginas 1726–1735, Junho 2002.
- [63] K.-L. Wu, P. S. Yu, e J. L. Wolf. Segmentation of multimedia streams for proxy caching. *IEEE Transactions on Multimedia*, 6(5):770–780, Outubro 2004.
- [64] S. Zhong, J. Chen, e Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. Em *Proceedings of IEEE INFOCOM*, 2003.