Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Tese de Doutorado

Conciliando Satisfação e Paridade para Promover Reciprocidade em Federações de Provedores de Computação na Nuvem

Eduardo de Lucena Falcão

Campina Grande, Paraíba, Brasil
Abril de 2018

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Conciliando Satisfação e Paridade para Promover Reciprocidade em Federações de Provedores de Computação na Nuvem

Eduardo de Lucena Falcão

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Metodologia e Técnicas da Computação

Francisco Vilar Brasileiro e Andrey Elísio Monteiro Brito (Orientadores)

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

F178c Falcão, Eduardo de Lucena.

Conciliando satisfação e paridade para promover reciprocidade em federações de provedores de computação na nuvem / Eduardo de Lucena Falcão. – Campina Grande, 2018.

130 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2018.

"Orientação: Prof. Dr. Francisco Vilar Brasileiro, Prof. Dr. Andrey Elísio Monteiro Brito".

Referências.

1. Internet - Padrões de Rede. 2. Computação na Nuvem - Provedores. 3. Sistemas P2P. 4. Federação de Nuvem - Provedores - Mecanismo de Incentivo. I. Brasileiro, Francisco Vilar. II. Brito, Andrey Elísio Monteiro. III. Título.

CDU 004.738.5.057.2(043)

"CONCILIANDO SATISFAÇÃO E PARIDADE PARA PROMOVER RECIPROCIDADE EM FEDERAÇÕES DE PROVEDORES DE COMPUTAÇÃO NA NUVEM"

EDUARDO DE LUCENA FALCÃO

TESE APROVADA EM 24/04/2018

FRANCISCO VILAR BRASILEIRO, Ph.D, UFCG
Orientador(a)

ANDREY ELÍSIO MONTEIRO BRITO, Dr., UFCG
Orientador(a)

NAZARENO FERREIRA DE ANDRADE, Dr., UFCG
Examinador(a)

JOSÉ ANTÃO BELTRÃO MOURA, Ph.D, UFCG
Examinador(a)

LUCIANO PASCHOAL GASPARY, Dr., UFRGS
Examinador(a)

LUIZ FERNANDO BITTENCOURT, Dr., UNICAMP
Examinador(a)

CAMPINA GRANDE - PB

Resumo

Provedores privados de computação na nuvem poderiam obter considerável benefício mútuo ao operar suas infraestruturas de forma federada. Tal operação permite que a demanda excedente de um provedor possa ser atendida por outros provedores que estejam experimentando uma baixa demanda naquele mesmo instante. Sob uma ótica de mercado, federações com arquitetura descentralizada têm como principal desafio a promoção de cooperação entre indivíduos egoístas e racionais em um cenário que não dispõe de uma autoridade central e confiável. Este trabalho se concentra em arquiteturas de mercado descentralizado, baseadas em *mecanismos de reciprocidade*, para suporte a federações P2P (*Peer-to-Peer*) de provedores de computação na nuvem.

Nos mecanismos de reciprocidade, um indivíduo utiliza o histórico de comportamentos de cooperação dos demais indivíduos para doar seus recursos aos que forem mais recíprocos. A maioria dessas estratégias se atém à priorização dos participantes (*a quem devo doar?*) de acordo com métricas tais como reputação ou grau de reciprocidade. Essa estratégia é suficiente para promover cooperação e assegurar aos participantes cooperativos os melhores níveis possíveis de ***satisfação*** (percentual de requisições atendidas). Porém, em cenários com baixa contenção de recursos, a priorização por si só não é suficiente para evitar o aproveitamento por parte de indivíduos não-cooperativos e garantir ***paridade*** (percentual de recursos retribuídos). Neste sentido, este trabalho propõe que mecanismos de reciprocidade, especialmente aqueles baseados em reciprocidade direta, sejam estendidos com um ***laço de controle retroalimentado*** que regula a quantidade de recursos que cada nuvem deveria ofertar à federação. Quando cada participante cooperativo controla de maneira individual a quantidade de recursos ofertada à federação, tem-se como resultado um controle indireto da contenção de recursos, que por sua vez é mantida em um patamar que assegura aos participantes cooperativos níveis adequados de satisfação e paridade.

Por fim, é apresentada uma investigação acerca da utilização de uma forma mais limitada de reciprocidade indireta, a ***reciprocidade transitiva***, que pode ser utilizada conjuntamente com mecanismos de reciprocidade direta, para evitar impasses gerados pela assimetria de tempo/interesses, aquecendo a economia da federação e consequentemente provendo maiores níveis de paridade e satisfação aos nós cooperativos.

Abstract

Private cloud providers could obtain considerable benefits from operating their infrastructures within a federation. Such operation allows a provider's exceeding demand to be met by other providers experimenting a resource underutilization on the same moment. From a market perspective, federations with decentralized architecture have as main challenge the promotion of cooperation among rational selfish individuals in a context with no central trusted authorities. This work focuses in decentralized market architectures, based on *reciprocity mechanisms*, to support P2P (*Peer-to-Peer*) federations of cloud providers.

In reciprocity mechanisms, an individual uses the history of behaviors of other individuals, reflecting their cooperation levels, in order to prioritize the provision of resource to those shown to be the most reciprocative. Most of these strategies are restricted to the prioritization of participants (*to whom should I donate?*) according to metrics such as reputation or degree of reciprocity. This strategy is sufficient to promote cooperation and assure cooperative participants the best possible levels of ***satisfaction*** (percentage of requests met). However, in low resource contention scenarios, prioritization alone is not sufficient to avoid resource provision to non-cooperative individuals and thus to guarantee ***fairness*** (percentage of resources reciprocated). In this sense, this work proposes that mechanisms of reciprocity, especially those based on direct reciprocity, be extended with a feedback control loop that regulates the amount of resources that each cloud should offer to the federation. When each cooperative participant controls individually the amount of resources offered to the federation, there is an indirect control of resource contention, which in turn is kept at a level that ensures cooperative participants adequate levels of satisfaction and fairness.

Finally, an investigation on the use of a more limited form of indirect reciprocity is presented, the ***transitive reciprocity***, which can be used in conjunction with direct reciprocity mechanisms to avoid deadlocks generated by time/interest asymmetry, moving the federation's economy and consequently assuring higher levels of fairness and satisfaction to cooperative nodes.

Agradecimentos

A Deus, autor da vida, por minha saúde e por todas as graças recebidas nesse período do doutorado. A minha mãezinha do céu, Nossa Senhora, obrigado por toda intercessão e companhia nessa jornada.

Aos meus pais, Marcelo e Virgínia, por todo esforço empreendido em minha educação, pelos conselhos, por sempre acreditarem em mim e, acima de tudo, pelo amor sem medidas. Aos meus irmãos, Fábio e Rafael, por todo companheirismo, amor, e por todos os momentos de descontração durante esses 4 anos (onde também estendo os agradecimentos as minhas cunhadas, Camila e Thaís). Amo todos vocês!

À Jéssyca Fernanda, amor da minha vida, por todo seu amor e carinho que continuamente renovam minhas forças, mas sobretudo pela paciência nos maus momentos. Também estendo meus agradecimentos a Angelita, Fernando e Louise, que também são minha família.

Aos meus orientadores, Fubica e Andrey, por todos os ensinamentos a mim transmitidos que vão muito além de "federações de nuvens" e "Redes de Favores". Espero poder fazer por muitas pessoas o que vocês fizeram por mim.

A todos os meus amigos do Laboratório de Sistemas Distribuídos e membros do projeto Fogbow, especialmente a José Luis Vivas, Chico, Giovanni e Fábio, pelas contribuições diretas à minha tese. Também gostaria de agradecer aos amigos (professores e alunos) que fiz nos 3 anos que ensinei na UFPB Campus IV — experiência gratificante que me fez ter certeza de que o ensino também é minha vocação.

Além destes, também gostaria de agradecer todos os meus grandes amigos: Bruno Sales, Bruno Augusto, Diego, Erick, Ernando, Guilherme, Gustavo, Hermano, Igor, Ivan, Lucas Guedes, Lucas Vieira, Luiz, Pedro, Victor Martins, Victor Mesquita. Também aos meus irmãos em Cristo, em especial a Mabel e George, e Julianne e Henrique.

Por fim, meus agradecimentos a todo o povo brasileiro e à Capes pelo subsídio financeiro que possibilitou a realização deste trabalho.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrônimos

DFKL - *Data First, Key Later*

FD-NoF - *Fairness-Driven Network of Favors*

FM - *Fogbow Manager*

FR - *Fogbow Rendezvous*

IaaS - *Infrastructure as a Service*

LAN - *Local Area Network*

NoF - *Network of Favors*

P2P - *Peer-to-Peer*

SD-NoF - *Satisfaction-Driven Network of Favors*

TI - Tecnologia da Informação

VM - *Virtual Machine*

WAN - *Wide Area Network*

Símbolos

Formalização do Problema

$\mathbb{F}$ - conjunto de nós de uma federação

$\mathbb{F}_c$ - conjunto dos nós cooperativos de uma federação

$\upsilon(A, B, t)$ - quantidade de recursos que o nó *A* doou para o nó *B* no passo de tempo t

$\kappa(t)$ - contenção de recursos entre nós cooperativos no passo de tempo t

$\rho(A, t)$ - quantidade de recursos requisitados pelo nó A no passo de tempo t

$\sigma(A, t)$ - quantidade de recursos disponibilizados pelo nó A no passo de tempo t

$\gamma(A, B, t)$ - crédito que o nó B possui com o nó A no passo de tempo t

$\phi(A, B, t)$ - paridade relativa a recursos consumidos e providos entre os nós A e B no passo de tempo t

$\phi(A, t)$ - paridade relativa a recursos consumidos e providos entre o nó A e os demais nós da federação no passo de tempo t

$\psi(A, t)$ - satisfação do nó A com os demais nós da federação no passo de tempo t

Controlador de Capacidade

$\alpha(A, B, t)$ - quantidade de recursos que o nó A disponibiliza para o nó B no passo de tempo t

$\mathcal{L}_{min}$ - limiar mínimo de paridade desejada

$\mathcal{L}_{max}$ - limiar máximo de paridade desejada

$\mathcal{C}$ - capacidade total de recursos de um nó

Δ - fator de variação utilizado para decrementar ou incrementar $\alpha(A, B, t)$

Modelo de Simulação

$\mathcal{N}$ - quantidade de nós na federação

$\mathcal{F}$ - proporção de caronas na federação

$\mathcal{P}$ - probabilidade de um dado nó estar em estado consumidor

$\mathcal{D}$ - quantidade total de recursos demandada por um nó

$\mathcal{D}_{fed}$ - quantidade total de recursos demandada por um nó à federação

$\mathbb{E}[\kappa]$ - nível de contenção esperado

Validação

Orders: estado, membro solicitante e membro provedor

$O_{r=i}$ - número de *orders* no estado *OPEN* em que o requerente é P_i

$P_{r=i}$ - número de *orders* no estado *PENDING* em que o requerente é P_i

$F_{r=i \wedge p \neq i}$ - número de *orders* no estado *FULFILLED* em que o requerente é P_i e o provedor não é P_i

$F_{r=i \wedge p=i}$ - número de *orders* no estado *FULFILLED* em que o requerente é P_i e o provedor também é P_i

$F_{r \neq i \wedge p=i}$ - número de *orders* no estado *FULFILLED* em que o requerente não é P_i e o provedor é P_i

Variáveis registradas nos logs

$D_i^T[x]$ - demanda total de P_i

$D_i^F[x]$ - demanda de P_i à federação

$R_i^F[x]$ - total de recursos da federação recebidos por P_i

$O_i^F[x]$ - total de recursos ofertados por P_i à federação

$S_i^F[x]$ - total de recursos de fato providos por P_i à federação

$U_i^F[x]$ - demanda não atendida pela federação

$R_i^L[x]$ - total de recursos locais recebidos por P_i

Métricas de desempenho

$\phi_i[x]$ - paridade de P_i

$\psi_i[x]$ - satisfação de P_i

Cenários - carga de trabalho

μ - fração dos nós cooperativos que apenas doa recursos

λ - intervalo de tempo que um nó permanece em um dado estado (provedor ou consumidor)

Contenção

$\mathbb{E}_F[\kappa]$ - contenção esperada do experimento no Fogbow

$\mathbb{E}_S[\kappa]$ - contenção esperada do experimento no simulador

Transitividade

η - probabilidade de um nó assumir o mesmo estado (consumidor, ocioso ou provedor) do grupo ao qual pertence

Capítulo 1

Introdução

Este capítulo inicia apresentando a importância de um paradigma emergente no contexto de sistemas para compartilhamento de recursos computacionais: a federação de provedores de computação em nuvem. Em seguida, são explicados a importância dos mecanismos de incentivo à cooperação, mais especificamente os mecanismos baseados em reciprocidade, e o motivo pelo qual estes mecanismos não conseguem assegurar paridade em cenários de baixa contenção de recursos. Por fim, são apresentados os objetivos, metodologia, principais contribuições e a organização do trabalho.

1.1 Motivação e Relevância

Organizações cujo grau de demanda computacional apresenta alta variabilidade geralmente recorrem às nuvens públicas (cf. "*cloud bursting*" [66]) a fim de atender necessidades inesperadas ou de curto prazo. Com uma ágil alocação de recursos, tais organizações conseguem evitar possíveis falhas no cumprimento de acordos de nível de serviço, possibilitando a oferta da qualidade de serviço acordada, mesmo em momentos de pico. Por outro lado, recursos próprios podem se tornar ociosos durante períodos de baixa demanda e embora nesse caso os custos operacionais possam ser reduzidos, bastando simplesmente desligá-los, os custos de amortização decorrentes da compra e manutenção dos recursos de Tecnologia da Informação (TI) ainda constituem uma perda de eficiência para a organização.

Nestas circunstâncias, o *cloud bursting* por si só não é suficiente para atingir eficiência máxima de custo. Uma alternativa que os provedores de nuvem, especialmente os privados,

podem considerar vantajosa é a colaboração de forma voluntária em um ambiente federado através do escambo de recursos ociosos. Neste caso, assume-se que cada membro da federação atua tanto como provedor quanto como consumidor de recursos durante momentos de vale e de pico de demanda, respectivamente. Em particular, devido à quantidade geralmente limitada de recursos que possuem, provedores de computação na nuvem privados poderiam obter considerável benefício mútuo ao combinarem seus esforços em uma federação [41].

No que concerne à arquitetura, federações de nuvens, ou em uma perspectiva mais abrangente, sistemas para compartilhamento de recursos, podem ser categorizadas como centralizadas ou descentralizadas [43]. Em arquiteturas centralizadas, a alocação de recursos é normalmente realizada ou facilitada por uma entidade central confiável cujas responsabilidades podem incluir o registro de recursos disponíveis além das funções convencionais de um mercado. Naturalmente, por possuir conhecimento total do sistema (e.g., características específicas da oferta e demanda, além do grau de reciprocidade de cada participante), arquiteturas centralizadas conseguem combinar consumidores e provedores com mais facilidade, o que diminui a eficiência de nós não-cooperativos [78]. Por outro lado, topologias centralizadas não são escaláveis nem flexíveis, e podem enfrentar problemas para garantir alta disponibilidade e geo-replicação.

Quando o contexto são arquiteturas descentralizadas, os participantes precisam se comunicar e negociar diretamente entre si, sem mediadores. Em contraponto às topologias centralizadas, as vantagens de arquiteturas descentralizadas incluem escalabilidade, flexibilidade e extensibilidade, provendo mais liberdade aos participantes do sistema. Obviamente, elas são independentes de qualquer infraestrutura centralizada e autoridade confiável, o que facilita a adesão de novos membros. As desvantagens, por outro lado, incluem dificuldades na descoberta, roteamento, segurança, e o fato de que os participantes são em sua maioria desconhecidos uns dos outros e não podem ser considerados confiáveis ou cooperativos.

Da mesma forma, sob uma ótica de mercado, tais federações também podem ser categorizadas como centralizadas ou descentralizadas. Evidentemente, há uma estreita relação entre as arquiteturas e as estruturas de mercado de tais federações. Geralmente, não é interessante implementar um sistema de mercado centralizado em uma arquitetura descentralizada, e vice-versa, pois deste modo o sistema tende a incorporar as desvantagens das duas topologias. Este trabalho se concentra em arquiteturas de mercado descentralizado, mais es-

pecificamente, arquiteturas par-a-par (P2P, do inglês *Peer-to-Peer*) baseadas em *mecanismos de reciprocidade*, para suporte a federações de provedores de computação na nuvem.

No contexto de mercados descentralizados o principal desafio é a promoção de cooperação entre indivíduos egoístas e racionais em um cenário que não dispõe de uma autoridade central e confiável. Neste tipo de mercado, os participantes possuem apenas uma quantidade limitada de informação sobre o sistema, adquirindo noção do comportamento real de outros atores através de sua própria experiência e baseando-se em interações passadas para decidir até que ponto eles devem confiar em outros participantes. Por isso, é natural esperar que, a priori, os participantes prefiram agir como caronas – indivíduos que apenas consomem recursos do sistema – uma vez que este é o comportamento que inicialmente seria mais vantajoso para eles. Além disso, os participantes cooperativos podem sair da federação se estiverem insatisfeitos com os resultados de suas interações.

Portanto, alguma estratégia que maximize os benefícios de participar da federação deve ser considerada, a fim de mantê-la proveitosa para os participantes. Uma alternativa é a utilização de mecanismos baseados em reciprocidade, onde um indivíduo utiliza o histórico de comportamentos de cooperação dos demais indivíduos para doar seus recursos aos participantes que demonstrarem maior reciprocidade. Já existem na literatura inúmeras investigações acerca da eficiência de mecanismos de reciprocidade [7; 8; 9; 16; 20; 22; 24; 32; 33; 34; 44; 46; 47; 48; 49; 54; 55; 58; 62; 63; 64; 65; 67; 68; 73; 74; 76; 78; 80; 81; 82; 84; 87; 88; 90; 91; 92; 93] mas apenas uma minoria foca o contexto de sistemas para compartilhamento de recursos computacionais [7; 8; 9; 68; 76; 78; 92; 93], que estaria mais alinhado às federações de nuvens. A maioria dessas estratégias se atém à priorização dos participantes de acordo com métricas tais como reputação ou grau de reciprocidade. Naturalmente, todas as estratégias priorizam participantes colaborativos, e isto se torna um incentivo à cooperação. Porém, em cenários com baixa contenção de recursos, *i.e.*, cenários com excesso de recursos, a priorização por si só não é suficiente para evitar o aproveitamento por parte dos caronas.

Para melhor compreensão do problema, a seguir, são brevemente descritas as duas principais métricas, usadas ao longo deste trabalho, para medir a eficiência de mecanismos de reciprocidade: *satisfação* e *paridade*.

Em qualquer momento, a *satisfação* de um participante é definida pela razão entre a

quantidade total de recursos recebidos de outros membros da federação até um dado instante, e o montante total de recursos requisitados pelo participante a outros membros da federação até o mesmo dado instante.

A *paridade* da federação em relação a um participante, por outro lado, é definida, em um dado momento, pela razão entre a quantidade total de recursos recebidos por este participante a partir de outros membros da federação até um dado instante, e o montante total de recursos providos por este participante a outros membros da federação até o mesmo dado instante. De maneira similar, pode-se definir a paridade de um participante em relação a outro membro de modo individual, considerando apenas os recursos providos e consumidos deste membro.

Este trabalho propõe uma abordagem que garante que tanto os níveis de satisfação como os níveis de paridade sejam bons o suficiente para assegurar que a maioria dos participantes não deixará a federação e que os caronas serão isolados e mantidos com um nível muito baixo de satisfação.

Como mencionado anteriormente, a maioria dos trabalhos encontrados na literatura tratam de mecanismos de reciprocidade para compartilhamento de recursos computacionais no contexto de grades computacionais P2P oportunistas [7; 68; 76; 92; 93]. Geralmente, os recursos ofertados em grades computacionais P2P oportunistas têm baixo custo operacional, haja vista que parte desses custos é amortizada pela utilidade obtida com o uso dos recursos para sua função primária. Por este motivo, a maioria [7; 68; 76; 92; 93] das estratégias para promover cooperação nestes cenários prioriza a satisfação, ou seja, a quantidade de requisições feitas por um nó cooperativo que são atendidas. Neste sentido, os nós cooperativos oferecem todos os seus recursos ociosos na expectativa de acumular créditos (ou boa reputação) com outros participantes, créditos estes que podem no futuro ser convertidos na concessão de recursos. Tal estratégia garante os melhores níveis possíveis de satisfação aos participantes, qualquer que seja o nível de contenção de recursos. No entanto, ofertar sempre todos os recursos ociosos pode prejudicar a paridade dos nós cooperativos, já que em cenários de baixa contenção de recursos os caronas poderiam aumentar seu nível de satisfação consumindo os recursos excedentes do sistema. Porém, sempre que o custo de fornecimento dos recursos excedentes não for muito maior do que mantê-los ociosos, como é o caso das grades oportunistas, a paridade não será mais importante do que a satisfação.

Quando o contexto é uma federação P2P de provedores de computação na nuvem as prio-

ridades passam a ser diferentes. Greenberg *et. al.* (2009) estimam que o custo anual de 50 mil servidores, assumindo um preço considerável de 3 mil dólares para cada servidor, é de aproximadamente 52.5 milhões de dólares, acompanhados de 18.4 milhões de dólares anuais referentes à manutenção da infraestrutura (distribuição de energia e sistemas de arrefecimento) e 9.3 milhões anuais provenientes de custo energético. Esses custos envolvidos na aquisição e operacionalização de *data centers* tornam a paridade um fator extremamente relevante. A priorização de concessão de recursos aos nós cooperativos que é realizada pela maioria [24; 73; 74; 65; 80; 7; 76; 70; 62; 16; 90; 32; 67; 55; 22; 46; 64; 91; 48; 44; 49; 58; 68; 92; 93] dos mecanismos de incentivo à cooperação baseados em reciprocidade garante níveis adequados de paridade somente em cenários onde há contenção pelos recursos.

Diante do exposto, fica evidente que para sistemas P2P nos quais o custo do recurso compartilhado não seja marginal, a exemplo de federações de provedores de computação na nuvem, a paridade não pode ser negligenciada. Nesse contexto, um membro cooperativo só permanecerá na federação se ela for paritária, ou seja, se ao menos a maioria dos recursos por ele providos forem retribuídos dentro de um intervalo razoável de tempo. Neste sentido, este trabalho propõe que os mecanismos de reciprocidade sejam estendidos com alguma lógica que, principalmente em cenários de baixa contenção de recursos, evite que participantes cooperativos provejam recursos para caronas.

Este trabalho propõe que os participantes sejam equipados com um mecanismo baseado na Teoria do Controle Retroalimentado [31], que regula a quantidade de recursos que cada participante disponibiliza para o sistema. Tal mecanismo se baseia na paridade individual com os demais participantes para definir a quantidade de recursos que deve ser ofertada a cada um deles. Com isto, quando os nós cooperativos tentam melhorar seus níveis de paridade através do controlador, eles indiretamente regulam a contenção de recursos mantendo-a em um nível adequado, mesmo que este não seja seu objetivo principal, e por consequência diminuem a satisfação dos caronas com mais eficiência, principalmente quando o nível de contenção é considerado demasiadamente baixo.

1.2 Objetivos

O objetivo geral do trabalho é permitir que participantes de uma federação P2P de nuvens sejam capazes de conciliar seus níveis de satisfação e paridade, principalmente nos cenários nos quais a oferta de recursos seja muito maior do que a demanda, situação na qual os caronas conseguem consumir recursos com mais facilidade. O objetivo específico consiste em verificar se o mecanismo de controle da oferta proposto é efetivo para promover cooperação paritária em federações P2P, através das seguintes hipóteses:

- o mecanismo de controle da oferta garante altos níveis de paridade em diferentes cenários de contenção de recursos, principalmente nos cenários de baixa contenção;
- o mecanismo de controle da oferta garante altos níveis de satisfação em diferentes cenários de contenção de recursos, principalmente nos cenários de baixa contenção.

Outro objetivo deste trabalho é propor uma solução complementar a mecanismos de reciprocidade direta que permita participantes com assimetria de disponibilidades (os momentos em que diferentes membros estão dispostos a prover e consumir recursos, reciprocamente, são incompatíveis) ou assimetria de interesses (o tipo do recurso ofertado é diferente do tipo do recurso demandado) cooperarem naturalmente.

1.3 Metodologia

Para **conciliar satisfação e paridade** em federações P2P de provedores de computação na nuvem foram definidas as seguintes atividades:

1. conceber um mecanismo que controle a quantidade de recursos que cada participante deve disponibilizar para a federação;
2. modelar e desenvolver um simulador que permita verificar os níveis de paridade e satisfação dos participantes em federações P2P;
3. executar simulações com diferentes configurações (número de participantes, porcentagem dos participantes que são caronas, nível de oferta e demanda, frequência de consumo/doação) que permitam avaliar o desempenho do mecanismo proposto em diferentes cenários de contenção de recursos;

4. validar o mecanismo proposto:

    4.1. implementar o mecanismo de controle de capacidades em um *middleware* para federação de provedores de computação na nuvem;

    4.2. verificar o desempenho do mecanismo através de experimentos controlados (usando o *middleware*) e compará-lo com os resultados obtidos pelo simulador.

Para **permitir cooperação entre participantes com assimetria de disponibilidades ou interesses** em federações P2P de provedores de computação foram definidas as seguintes atividades:

1. conceber um mecanismo que contenha algum tipo de reciprocidade indireta para permitir cooperação entre indivíduos com alguma espécie de assimetria;

2. modelar e desenvolver um simulador que permita verificar os níveis de paridade e satisfação de participantes com alguma espécie de assimetria em federações P2P;

3. executar simulações que permitam avaliar o desempenho do mecanismo proposto em cenários com diferentes níveis de assimetria entre participantes.

1.4 Contribuições

Este trabalho apresenta um mecanismo que pode ser utilizado por provedores de computação na nuvem para controlar a quantidade de recursos a ser ofertada aos demais membros de uma federação de nuvens. No que concerne seu desempenho, o mecanismo proposto é avaliado em cenários com diferentes níveis de contenção de recursos, de onde verificou-se que o mesmo concilia os níveis de satisfação e de paridade entre recursos doados e recebidos, até mesmo em cenários de baixa contenção por recursos. Além disto, uma implementação desse mecanismo é realizada em um *middleware* para federação de nuvens, sugerindo decisões arquiteturais para sua implementação. A partir da avaliação do funcionamento do mecanismo no *middleware* também é possível compreender a complexidade natural decorrente da negociação entre consumidores e provedores, o que pode elevar o tempo de descoberta de provedores disponíveis a doar recursos.

No campo teórico-analítico, uma discussão sobre os principais aspectos dos mecanismos de incentivo à cooperação baseados em reciprocidade bem como uma apresentação de seu estado da arte são realizadas, o que permite o leitor compreender a lacuna existente nessa área.

Por fim, um tipo limitado de reciprocidade indireta, a reciprocidade transitiva, é apresentado como solução complementar a mecanismos de reciprocidade direta, com o objetivo de permitir cooperação entre participantes com assimetria de tempos ou interesses mas mantendo os benefícios da reciprocidade direta.

1.5 Organização do Documento

O restante deste documento encontra-se organizado da seguinte forma. No Capítulo 2 são apresentadas a fundamentação teórica, que traz os principais conceitos acerca dos mecanismos de reciprocidade, e um detalhamento do problema que é causado pela abundância de recursos, seguido por uma descrição formal do problema. Os trabalhos relacionados a mecanismos de reciprocidade direta e indireta são apresentados no Capítulo 3, estejam eles inseridos no contexto de compartilhamento de recursos computacionais ou no contexto de compartilhamento de arquivos, uma vez que mecanismos utilizados em sistemas de compartilhamento de arquivos possuem conceitos e técnicas que podem ser adaptados ao contexto de federações de nuvens. A seguir, no Capítulo 4 são apresentados os detalhes da solução proposta – o controlador de oferta de recursos, implementado através de um laço de controle retroalimentado. A avaliação da solução proposta é feita no Capítulo 5, onde os resultados das simulações e correspondentes análises são apresentados. O Capítulo 6 apresenta a validação da solução proposta, detalhando sua implementação em um *middleware* para federação de nuvens, e exibe uma comparação dos resultados obtidos através de experimentos utilizando o *middleware* com os resultados do simulador. Um estudo paralelo sobre a reciprocidade transitiva é apresentado no Capítulo 7, explicando como ela pode elevar os níveis de cooperação de sistemas P2P em determinados cenários. Por fim, o Capítulo 8 encerra este trabalho apresentando as principais conclusões e direções para trabalhos futuros.

Capítulo 2

Problema Investigado

Neste capítulo são apresentadas a fundamentação teórica e a formalização do problema. A seguir são detalhados os principais conceitos relativos a mecanismos de incentivo à cooperação baseados em reciprocidade, mecanismos estes que podem se tornar mais eficientes com a contribuição do presente trabalho. Os princípios da reciprocidade direta e indireta, detalhados neste capítulo, também podem auxiliar pesquisadores a discernirem com mais clareza qual tipo de reciprocidade é mais adequado para um determinado contexto. Por fim são apresentadas a definição do problema e as questões de pesquisa investigadas por este trabalho.

2.1 Fundamentação Teórica

Existem diferentes abordagens para promover cooperação em um sistema distribuído com topologia descentralizada. Nenhuma dessas abordagens é perfeita para promover cooperação de maneira eficiente em todos os contextos. Ao longo deste capítulo é possível entender, por exemplo, por que os mecanismos de incentivo à cooperação comumente aplicados em sistemas de compartilhamento de arquivos podem não prover os melhores resultados se aplicados a uma grade computacional oportunista ou a uma federação de provedores de computação na nuvem. Por este motivo, é importante conhecer as vantagens e desvantagens de cada mecanismo para compreender com mais facilidade em quais contextos eles deveriam ser utilizados.

De modo geral, há duas maneiras de fomentar cooperação em sistemas descentralizados,

através de um *mercado monetário* [59; 69] ou baseado em *princípios de reciprocidade* [47]. Em mercados monetários, entidades pagam para obter serviços e são pagas pelos serviços que proveem. Esses pagamentos são efetuados através de crédito ou dinheiro que são comumente mantidos em uma entidade centralizada e confiável — uma espécie de banco. Abordagens monetárias impõem a introdução de procedimentos complexos de gerenciamento e segurança, enquanto mecanismos baseados em reciprocidade podem ser mais facilmente implementados, por exemplo, através de anotações locais de crédito ou reputação.

A contribuição deste trabalho não se estende a mecanismos de incentivo à cooperação baseados em mercados monetários. A oferta de serviços mediante pagamento é suficiente para evitar os comportamentos egoístas dos caronas. Nesses casos, a abundância de recursos existentes em um sistema P2P não seria um problema pois a obrigatoriedade dos pagamentos impediria que os caronas consumissem os recursos excedentes sem uma contrapartida. Porém, quando se trata de mecanismos de reciprocidade, o excesso de recursos pode ser considerado um problema uma vez que os caronas conseguem consumir as sobras do sistema sem retribuí-las.

O produto deste trabalho, um controlador de oferta de recursos, afeta positivamente os mecanismos de incentivo à cooperação baseados em reciprocidade, principalmente reciprocidade direta. Tais mecanismos são explicados a seguir.

2.1.1 Mecanismos Baseados em Reciprocidade

Em mecanismos baseados em reciprocidade, um indivíduo utiliza o histórico de comportamentos de cooperação de outros indivíduos para, de acordo com o grau de reciprocidade, decidir quais devem ser priorizados [34]. Tais esquemas podem ser classificados como *reciprocidade direta* ou *reciprocidade indireta*.

A *reciprocidade direta* acontece quando dois indivíduos decidem se ajudar baseados em interações passadas entre eles. Mais especificamente, um indivíduo A decide (como) servir um indivíduo B baseado exclusivamente nos serviços que B prestou para A anteriormente. A baseia suas decisões utilizando apenas suas experiências anteriores, informações locais mantidas por A a respeito do comportamento dos demais indivíduos.

Essa troca de serviços pode se dar através de *permuta imediata* ou através de *permuta tardia* [47]. Na permuta imediata dois indivíduos proveem serviços ou recursos um para

o outro no mesmo tempo, ao passo em que na permuta tardia o serviço é provido como uma compensação de um serviço previamente recebido ou na expectativa de compensação futura. A principal vantagem da permuta imediata é a detecção e possibilidade de punição dos caronas em tempo real. A desvantagem da permuta imediata, em contrapartida, está na necessidade de que os indivíduos estejam aptos a oferecer conteúdos/serviços entre si de forma simultânea. Em contraponto à reciprocidade direta por permuta imediata, uma vantagem da reciprocidade por permuta tardia é a não necessidade de interesses mútuos em um mesmo momento.

O exemplo mais popular de permuta imediata é o mecanismo de incentivo à cooperação utilizado no BitTorrent [24], protocolo de comunicação para compartilhamento de arquivos em sistemas P2P. Um exemplo de reciprocidade direta por permuta tardia é a Rede de Favores [7], mecanismo de incentivo à cooperação utilizado em grades computacionais P2P oportunistas. No BitTorrent, dois participantes interagem quando, em um dado momento, cada um tiver conteúdo de interesse do outro. Nesses casos, cada participante é consumidor e provedor de recursos em um mesmo instante de tempo. A Rede de Favores, por sua vez, utiliza um sistema baseado em créditos para priorizar os nós solicitantes. Na Rede de Favores, sempre que um nó recebe/doa um recurso (ou favor, nesse contexto) ele anota uma certa quantidade de crédito/débito para o provedor/consumidor, e posteriormente utiliza essas informações para decidir a quem conceder favores no futuro.

A cooperação através de reciprocidade direta é perceptivelmente mais adequada para cenários onde indivíduos *interagem repetidamente* [17]. Na reciprocidade direta os indivíduos se baseiam exclusivamente em seu histórico local de informações para decidir até que ponto devem confiar em outros participantes. Deste modo, interações repetidas com indivíduos permitem que um participante construa um histórico sólido a respeito do comportamento de alguns indivíduos e isto o permite agir racionalmente, recompensando os indivíduos mais cooperativos e retaliando indivíduos que agem como caronas. Na Teoria dos Jogos, tais comportamentos têm sido extensivamente investigados no tópico "jogos repetidos", de onde muitos trabalhos extraíram e utilizaram a estratégia *tit-for-tat*[a] [12].

[a]*Tit-for-tat* pode ser traduzido para a língua portuguesa através da expressão "*olho por olho, dente por dente*", inspirado na lei de talião. O algoritmo rege que na primeira interação o jogador deve cooperar, e a partir da segunda interação ele deve agir exatamente como seu oponente agiu anteriormente [12].

Seguindo a linha de raciocínio anterior é possível afirmar que mecanismos de incentivo à cooperação baseados em reciprocidade direta são mais adequados para comunidades cujos participantes possuem *interesses simétricos* [62], *i.e.*, participantes que ofertam conteúdos ou serviços que interessam outros participantes. De modo geral, sistemas de compartilhamento de recursos computacionais, como grades computacionais e federações de provedores de computação na nuvem, possuem esta característica [8]. Os participantes ingressam na comunidade para ofertar seus recursos ociosos (*e.g.*, ciclos de processamento, largura de banda, espaço em disco) e consumir recursos ociosos da comunidade quando os recursos locais se esgotarem. O portfólio pequeno e bem definido de serviços ofertados converge para a simetria de interesses entre os participantes desses tipos de sistema. Esta simetria, por sua vez, aumenta as chances de haver encontros repetidos entre indivíduos e isto permite que o histórico local de interações que cada participante mantém seja uma boa fonte de informações para tomada de decisão.

O histórico privado de interações proposto pela reciprocidade direta tem como principal vantagem a *integridade das informações* a respeito dos comportamentos dos demais participantes. Obviamente, um indivíduo sempre confiará na base de informações que ele mesmo mantém. Esta é, portanto, a maneira mais simples de proteção contra indivíduos maliciosos que formam conluios para ludibriar participantes colaborativos [23]. Contudo, históricos privados também trazem consigo uma desvantagem: a base de informações que um único indivíduo mantém cresce lentamente, de acordo com suas experiências individuais. Deste modo, em sistemas de reciprocidade direta indivíduos recém-chegados tomam decisões de doações completamente "às cegas", tornando-os mais vulneráveis às ações dos caronas.

Outro fator importante é a facilidade com que nós recém-chegados conseguem, ao entrar no sistema, consumir serviços. Em inglês, o termo utilizado para representar e quantificar esta propriedade é *bootstrapping speed* [54]. Se os participantes do sistema usassem uma forma de reciprocidade direta mais estrita, ou seja, se eles só provessem um serviço como forma de retribuição, indivíduos recém-chegados teriam que primeiro ofertar algum serviço para depois estarem aptos a receber serviços. Porém, se a princípio todos os indivíduos fossem recém-chegados (*e.g.*, formação inicial do sistema), todos esperariam os outros indivíduos doarem primeiro, e esse impasse jamais seria resolvido. Por esta razão, é normal que os mecanismos de incentivo à cooperação baseados em reciprocidade direta adicionem

em suas estratégias algum nível de altruísmo[b]. Adicionalmente, o altruísmo é uma das principais maneiras que um indivíduo tem de ampliar seu círculo social, visando aumentar as chances de conhecer outros nós cooperativos e priorizá-los em detrimento de participantes desconhecidos.

É interessante ressaltar que a reciprocidade direta utiliza o altruísmo como uma alternativa para evitar situações de impasse e aumentar a velocidade de *bootstrapping* dos indivíduos, e consequentemente do sistema. O principal pilar da **reciprocidade direta** é, sem sombra de dúvidas, o *tit-for-tat*.

Na *reciprocidade indireta*, também conhecida por *reciprocidade generalizada* [86], um indivíduo decide interagir com outro indivíduo a partir de informações fornecidas por terceiros, e não mais baseado exclusivamente em suas próprias informações, como é o caso da reciprocidade direta. Na reciprocidade indireta, um indivíduo busca ajudar outros indivíduos cooperativos, independentemente de conhecê-los. Por esta razão é possível afirmar que um indivíduo fornece serviços de modo altruísta a outros participantes da comunidade como uma maneira de premiá-los por seus comportamentos cooperativos com outros indivíduos e, do mesmo modo, espera ser retribuído futuramente de maneira indireta por outros participantes com os quais possivelmente nunca interagiu. Dessa forma, a **reciprocidade indireta** tem como principal pilar o *altruísmo recíproco*.

A reciprocidade indireta pode ser classificada nos dois seguintes tipos [34]:

1. ***Pay-it-forward***: se *A* ajudar *B*, *B* irá ajudar *C*;

2. **Sistemas de reputação**: se *A* ajudar *B*, *C* irá ajudar *A*.

No *pay-it-forward*, quando *A* ajuda *B*, *B* se sente bem e retribui o favor que recebeu de *A* para um outro indivíduo, *C* neste caso. A ideia principal é que o favor deve ser passado adiante. As métricas utilizadas para que *B* escolha *C* podem variar nas diferentes implementações desse mecanismo, ou simplesmente inexistir, onde a escolha seria completamente aleatória. No mecanismo de *reputação*, *C* ajuda *A* porque soube que *A* ajudou *B*. *C* ajuda *A* porque na visão de *C*, *A* tem uma boa reputação e é recompensado justamente por isso.

A reciprocidade indireta por *pay-it-forward* pode se dar através de permuta tardia ou permuta imediata. Por outro lado, a reciprocidade indireta por reputação só acontece através

[b]Comportamento que beneficia outro indivíduo e incorre em um certo custo ao primeiro indivíduo.

de permuta tardia, uma vez que a reputação é construída a partir de comportamentos passados dos outros indivíduos. A Figura 2.1 ilustra, para fins de comparação, o funcionamento da reciprocidade direta e dos dois tipos de reciprocidade indireta.

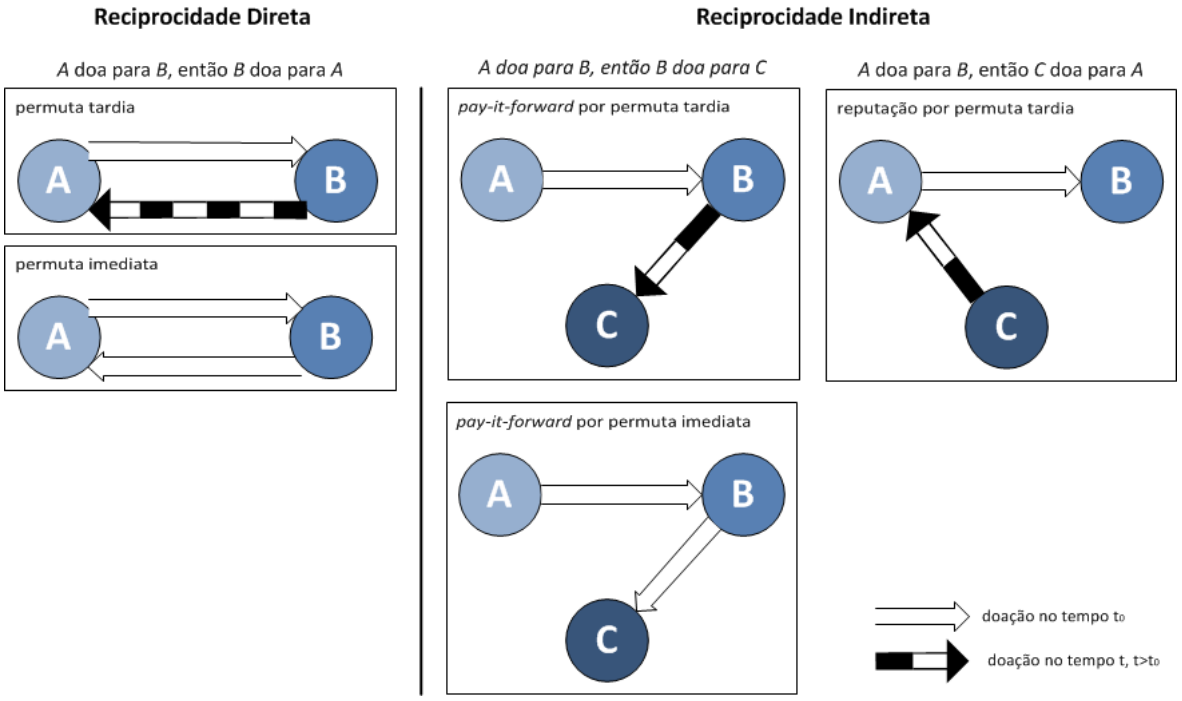


Figura 2.1: Tipos de mecanismos de incentivo à cooperação baseados em reciprocidade.

O *pay-it-forward* puro, *i.e.*, quando um indivíduo retribui esse favor a um terceiro baseado em nenhuma métrica ou de forma completamente aleatória, é um mecanismo de incentivo à cooperação que beira a utopia. Para que uma comunidade funcione com um mecanismo de reciprocidade baseado em *pay-it-forward* é preciso confiar que todos os indivíduos sejam cooperativos, já que sanções ou punições não são aplicáveis. Por este motivo, poucos trabalhos utilizam essa estratégia para promover cooperação.

Por outro lado, mecanismos de incentivo à cooperação baseados em sistemas de reputação são usados com frequência em sistemas P2P. Nesse tipo de sistema os indivíduos são incentivados a cooperar para que construam a imagem de um participante colaborativo e que agrega valor à comunidade, de modo que sejam retribuídos futuramente. Portanto, se *A* ajudar *B*, de maneira indireta *C* saberá que *A* é cooperativo e o premiará pela sua boa reputação com a comunidade.

A principal *desvantagem* desse tipo de reciprocidade é que um indivíduo baseará suas decisões de doações em informações fornecidas por terceiros, que podem manipular as in-

formações visando benefícios próprios. Esse é, segundo Ciccarelli e Cigno (2011), o tipo mais simples de *colusão*: os indivíduos em conluio sempre mentem a respeito dos demais participantes, reportam parceiros de colusão como sendo cooperativos e reportam indivíduos honestos como não cooperativos. Existem algumas maneiras de atenuar este problema, como por exemplo, confiar apenas nas informações providas por indivíduos cooperativos e que, portanto, transmitem um bom nível de confiança. Esta é a categoria mais simples de colusão, e apesar de existirem estratégias que atenuam os efeitos desse problema não se pode afirmar que elas o eliminam por completo.

Se por um lado, a maneira como as informações são coletadas em sistemas de reputação trazem a possibilidade de colusão como desvantagem, por outro lado, a *velocidade* com que elas são disseminadas pelos participantes do sistema é sua principal *vantagem*. Essa característica permite que os indivíduos recém chegados à comunidade obtenham de maneira rápida uma quantidade razoável de informações a respeito do comportamento de outros participantes, de modo que possam discernir com mais lucidez com quem deveriam cooperar. Adicionalmente, essa velocidade com que a informação é disseminada faz com que participantes de um sistema P2P regido por reputação possam também *interagir eventualmente*, ao invés de apenas interagir repetidamente, como requerido na reciprocidade direta. Desse modo, é possível afirmar que esse tipo de incentivo à cooperação é adequado também para comunidades onde os participantes possuam alto nível de *assimetria de interesses*, e não apenas interesses simétricos como na reciprocidade direta. Ora, se os encontros repetidos são desnecessários também não há obrigatoriedade de haver alta simetria de interesses.

Dois problemas fortemente relacionados que existem em ambas as modalidades de reciprocidade são: i) como tratar os indivíduos recém-chegados e ii) como atenuar as vantagens obtidas por *whitewashers*, *i.e.*, indivíduos que saem e voltam para a comunidade com uma nova identidade para se livrar das consequências de um passado em que agiram de maneira maliciosa ou não-cooperativa [47]. Se as regras para entrar no sistema forem muito restritivas os indivíduos podem ser desencorajados de entrar na comunidade, porém, se essas regras forem muito liberais, isso dará incentivo à prática de troca de identidades. Uma maneira de resolver esse problema é fixando uma taxa de entrada, ou prova de trabalho, para poder entrar no sistema [25; 26]. Outra alternativa que pode ser utilizada tanto na reciprocidade direta como na indireta é desconsiderar valores negativos para sistemas de crédito ou repu-

tação. Deste modo, não faria sentido para um indivíduo sair e voltar para a rede se esse fato não altera a forma como os demais participantes o valoram.

De modo geral, é possível afirmar que a principal vantagem da **reciprocidade indireta** é a *velocidade rápida* com que a base de informações dos comportamentos dos demais indivíduos cresce. A **reciprocidade direta**, por sua vez, tem como principal vantagem a *integridade* dessas informações. A Tabela 2.1 apresenta, a partir dessas duas vantagens, as principais características da reciprocidade direta e indireta, como forma de comparação rápida entre os dois tipos de mecanismo.

Característica	Reciprocidade Direta	Reciprocidade Indireta
base de informações - velocidade	lenta	rápida
base de informações - integridade [c]	garantida	não é garantida
oferta e demanda - serviço	simétrico	simétrico e assimétrico
frequência de interações	repetidas	repetidas e eventuais
velocidade de *bootstrapping*	lenta	rápida
colusão	impossível	possível
troca de identidades	possível	possível

Tabela 2.1: Características da reciprocidade direta e indireta.

A partir da Tabela 2.1 é possível perceber que a reciprocidade indireta leva vantagem em quatro características enquanto a reciprocidade direta leva vantagem em apenas duas. Contudo, esta simples constatação não pode ser usada para afirmar que a reciprocidade indireta é melhor do que a direta. O que deve servir como guia para esta escolha são as características do ambiente em que ela será aplicada.

Em *sistemas de compartilhamento de arquivos*, por exemplo, existe uma assimetria de interesses muito grande (diferentes participantes demandam e ofertam conteúdos diferentes) e por esta razão os participantes interagem diretamente com uma frequência muito baixa. Pela tabela apresentada anteriormente, acredita-se que a reciprocidade indireta é a melhor solução para esse cenário, embora vários trabalhos [24; 33; 63; 65; 73; 74; 80; 87; 82; 88] já tenham utilizado a reciprocidade direta para estes mesmos casos. Acredita-se que

[c] Aspectos de segurança (*e.g.*, *hackers* ou *malwares*) não são considerados.

eles preferiram a confiabilidade provida pela integridade das informações ao invés de uma evolução mais rápida do sistema. Por fim, um fato que não pode ser negligenciado é que mecanismos de reciprocidade direta são de implementação mais fácil do que os que se baseiam em reciprocidade indireta.

Outros exemplos são os *sistemas para compartilhamento de recursos computacionais* como grades computacionais P2P ou federações P2P de provedores de computação na nuvem. Nesses tipos de sistema a oferta e demanda de serviços possuem alto grau de simetria e isto leva os participantes a interagirem repetidamente. Nesse sentido, a reciprocidade direta seria uma alternativa interessante pelo fato de garantir a integridade das informações de comportamentos dos demais participantes, o que, de maneira extremamente simples, tornaria nula qualquer possibilidade de colusão por parte de indivíduos maliciosos. Contudo, a literatura contém alguns trabalhos [68; 92; 93] que utilizam sistemas de reputação para promover cooperação em grades computacionais. Provavelmente esses trabalhos priorizaram a velocidade de disseminação da informação dos comportamentos dos participantes e, portanto, a velocidade de "evolução do sistema" em detrimento da integridade dessas informações.

Em resumo, a decisão sempre recairá sobre a análise do custo e benefício entre integridade das informações e velocidade com que elas são disseminadas.

Para melhor entendimento de como este trabalho se posiciona em relação aos demais e como ele pode contribuir para aumentar a eficiência de mecanismos de incentivo à cooperação baseados em reciprocidade, primeiramente é importante entender o problema abordado com maior riqueza de detalhes. A seguir é detalhado o problema de como garantir paridade em cenários de baixa contenção de recursos.

2.2 O Paradoxo da Abundância de Recursos

De modo geral, os mecanismos de reciprocidade direta e indireta servem para ajudar um indivíduo a decidir *quais participantes devem ser* ***priorizados*** na concessão de recursos. A principal métrica para auxiliar essa escolha é o nível de cooperação/colaboração/reciprocidade de cada participante. Porém, a priorização por si só não é suficiente para assegurar paridade quando a contenção de recursos está muito baixa, isto é, quando a oferta é muito maior do que a demanda. Para entender este problema sem ambiguidades, a seguir é forma-

lizado o conceito de contenção de recursos.

Primeiramente é preciso formalizar a noção de recursos compartilhados entre os participantes (daqui por diante referidos como "nós"). Considere $\mathbb{F}$ como sendo o conjunto de nós em uma federação P2P. Assuma que $\upsilon : \mathbb{F} \times \mathbb{F} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ é uma função degrau que denota a quantidade de recursos compartilhados entre dois nós em um certo intervalo de tempo (considerado discreto). $\upsilon(A, B, t)$, por exemplo, denota a quantidade de recursos que *A* doou para *B* durante o passo de tempo *t*.

Contenção é uma métrica que caracteriza o grau de competição por recursos, e é definida pela razão entre a quantidade total de recursos requisitados por nós cooperativos à federação e a quantidade total de recursos ofertados à federação, em um dado tempo. Formalmente, a contenção pode ser representada por uma função degrau $\kappa : \mathbb{N} \to \mathbb{R}_{\geq 0}$, como explicado a seguir.

Considere $\mathbb{F}_c \subseteq \mathbb{F}$ como sendo o conjunto de nós cooperativos na federação, $\rho : \mathbb{F}_c \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ como sendo uma função degrau que denota a quantidade de recursos requisitados por um nó $A \in \mathbb{F}_c$ no passo de tempo t, e $\sigma : \mathbb{F}_c \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ como sendo uma função degrau que denota a quantidade de recursos que um nó $A \in \mathbb{F}_c$ oferta no passo de tempo t. A contenção de recursos no passo de tempo t pode ser definida por:

$$\kappa(t) = \frac{\sum_{P \in \mathbb{F}_c} \rho(P, t)}{\sum_{P \in \mathbb{F}_c} \sigma(P, t)}. \tag{2.1}$$

Para facilitar o entendimento do problema, considere uma federação com apenas seis nós, A, B, C, D, E e F, onde os nós B, C, D, E e F são priorizados por A em um dado tempo t através de um sistema de crédito[d], como apresentado na Tabela 2.2.

Para fins de explicação, considere a seguinte função de crédito $\gamma : \mathbb{F} \times \mathbb{F} \times \mathbb{N} \to \mathbb{R}$ como forma de priorização dos nós na Tabela 2.2. A função $\gamma(A, B, t)$ denota o crédito (ou saldo) de B na visão de A no início do passo de tempo t. A fórmula mais simples para esse cálculo é a diferença entre a quantidade total de recursos providos por B para A e quantidade total de recursos providos por A para B, até o passo de tempo $t - 1$:

$$\gamma(A, B, t) = \sum_{i=0}^{t-1} \upsilon(B, A, i) - \upsilon(A, B, i). \tag{2.2}$$

[d]Não faz diferença se a priorização é realizada via sistema de créditos ou reputação.

Nó P	$\sum_{i=0}^{t-1} \upsilon(P, A, i)$	$\sum_{i=0}^{t-1} \upsilon(A, P, i)$	$\gamma(A, P, t)$
D	100	80	**20**
E	100	85	**15**
B	1000	990	**10**
C	100	100	**0**
F	0	300	**-300**

Tabela 2.2: Tabela de priorização mantida por A no passo de tempo t.

Agora suponha que no passo de tempo t o nó A tenha capacidade excedente de 40 unidades de recurso que deseja prover à federação. Se nesse passo de tempo os demais nós estiverem requisitando cada um 5 unidades de recurso, então A alocaria 5 unidades para cada um deles. Note que F aparenta ser um carona uma vez que até o tempo t não retribuiu para A nenhum recurso recebido. Nesse caso, o nível de contenção da federação no passo de tempo t seria $\kappa(t) = 0.5$, o que significa que a oferta seria duas vezes maior do que a demanda dos nós cooperativos, e portanto, os caronas conseguiriam consumir com facilidade os recursos excedentes no sistema.

Suponha agora que ao invés de ofertar 40 unidades de recurso A disponibilizasse apenas 10 unidades de recurso para a federação. Nessa nova situação A doaria apenas 5 unidades de recurso para D e 5 unidades de recurso para E. Os nós B, C e F não receberiam recursos de A no passo de tempo t simplesmente porque não haveria recursos excedentes para tal. O nível de contenção para esse cenário seria $\kappa(t) = 2$, o que significa que a demanda dos nós cooperativos seria o dobro da quantidade de recursos disponível na federação.

Desses dois cenários é possível concluir que quando o nível de contenção por recursos estiver baixo ($\kappa(t) < 1$), um nó cooperativo poderá tomar más decisões de doações, provendo recursos a caronas, que por sua vez não retribuirão estes recursos no futuro. Por outro lado, quando o nível de contenção estiver intermediário ou alto ($\kappa(t) \geq 1$), os nós cooperativos tomarão boas decisões de doações, provendo recursos a nós cooperativos que os retribuirão no futuro.

Neste sentido, este trabalho propõe que os nós cooperativos sejam equipados com algum mecanismo que limite a quantidade de recursos ofertada para a federação, com o objetivo de obter níveis mais elevados de paridade, o que de modo indireto regularia os níveis de

contenção em um patamar intermediário ($\kappa(t) \approx 1$).

A seguir são formalizadas as métricas, paridade e satisfação, utilizadas para medir o desempenho dos nós, além de auxiliar na formalização do problema.

2.2.1 Formalização do Problema

Com o objetivo de avaliar o desempenho dos nós em diferentes cenários de contenção por recursos, foram definidas duas métricas, paridade e satisfação, brevemente descritas na Introdução. *Paridade* é uma medida do nível de reciprocidade dos serviços providos, seja em relação a apenas um nó específico ou ao sistema como um todo, enquanto *satisfação* representa a porcentagem de requisições atendidas.

A paridade entre dois nós é definida por uma função degrau $\phi : \mathbb{F}_c \times \mathbb{F} \times \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ denotando a razão entre a quantidade total de recursos que um nó cooperativo A consumiu de um nó B e a quantidade total de recursos que o mesmo nó A doou para B, até o passo de tempo t. Para os casos em que A tenha realizado pelo menos uma doação para B até passo de tempo t, essa função pode ser definida como:

$$\phi(A, B, t) = \frac{\sum_{i=0}^{t-1} \upsilon(B, A, i)}{\sum_{i=0}^{t-1} \upsilon(A, B, i)}. \tag{2.3}$$

Por conveniência, assume-se que $\upsilon(A, A, i) = 0, \forall A \in \mathbb{F} \wedge \forall t \in \mathbb{N}$. Se $\sum_{i=0}^{t-1} \upsilon(A, B, i) = 0$, então o valor de $\phi(A, B, t)$ é indefinido.

Quando medida em relação ao sistema como um todo, a paridade é definida pela função degrau $\phi : \mathbb{F}_c \times \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ que, sempre que um nó $A \in \mathbb{F}_c$ tenha provido algum recurso a algum outro nó, pode ser definida por:

$$\phi(A, t) = \frac{\sum_{P \in \mathbb{F}} \sum_{i=0}^{t-1} \upsilon(P, A, i)}{\sum_{P \in \mathbb{F}} \sum_{i=0}^{t-1} \upsilon(A, P, i)}. \tag{2.4}$$

Satisfação, por outro lado, é definida por uma função degrau $\psi : \mathbb{F} \times \mathbb{N} \rightarrow \mathbb{R}_{[0..1]}$ como a razão, para qualquer nó A, entre a quantidade total de recursos consumidos e a quantidade total de recursos requisitados, antes do passo de tempo t. Lembrando que $\rho(A, t)$ denota a quantidade total de recursos requisitados por A aos demais membros da federação durante o passo de tempo t, então a satisfação de A com o sistema no passo de tempo t é definida por:

$$\psi(A, t) = \frac{\sum_{P \in \mathbb{F}} \sum_{i=0}^{t-1} \upsilon(P, A, i)}{\sum_{i=0}^{t-1} \rho(A, i)}. \tag{2.5}$$

Note que a satisfação é inversamente proporcional à contenção, sendo uma função indireta da mesma. Obviamente, quanto maior o nível de contenção, menor será a probabilidade que um certo nó receberá o que pede em um dado momento, uma consequência do fato de que um serviço recebido por um certo nó é um serviço provido por outro nó. Portanto, é de comum interesse que os nós cooperativos ofertem a maior quantidade de recursos possível.

Comunidades em que os nós (sugeridos por seus respectivos mecanismos de incentivo) ofertarem todos os seus recursos ociosos ao sistema, terão de modo geral os maiores níveis de satisfação possíveis. Nesses casos a satisfação é maximizada sem considerar (ou dando pouca atenção) aos níveis de paridade alcançados. Isto é aceitável quando o custo de provisão dos recursos é negligenciável (*e.g.*, banda de rede em sistemas P2P para compartilhamento de arquivos). Contudo, se este não é o caso, a satisfação virá acompanhada de um certo custo. Se a utilidade marginal obtida a partir dos recursos recebidos for menor do que o custo marginal dos recursos providos (para poder recebê-los), o nó provavelmente estará tomando uma decisão ruim ao prover todos os seus recursos disponíveis.

Nesses casos, os nós estarão se deparando com uma situação em que a maximização dos níveis de satisfação deve ser restringida pelo custo de provisão de recursos por um nó A, ou em outras palavras, pela quantidade total de recursos que o nó A doa no passo de tempo t, *i.e.*, $\sum_{P \in \mathbb{F}} \upsilon(A, P, t)$. Uma vez que a quantidade de recursos doados por A é restringida pela quantidade de recursos que A oferta ao sistema no passo de tempo t, *i.e.*, $\sum_{P \in \mathbb{F}} \upsilon(A, P, t) \leq \sigma(A, t)$, reduzir a quantidade de recursos ofertados deve portanto levar a um aumento nos níveis de paridade, se a quantidade de recursos recebidos não for reduzida na mesma taxa.

A proposta deste trabalho é que os nós cooperativos sejam equipados com algum mecanismo que os permita regular a quantidade de recursos ofertados, impactando indiretamente o nível da contenção por recursos no sistema. Isto deve ser feito de tal maneira que um nó consiga obter níveis justos de paridade (próximo da unidade), sem afetar de maneira significativa seu nível de satisfação.

Embora possa ser aplicada em ambos os mecanismos de reciprocidade, essa ideia tende a beneficiar muito mais os mecanismos de reciprocidade direta, pois mesmo em cenários de alta contenção é possível que os nós cooperativos doem recursos a caronas. Na reciprocidade indireta essa possibilidade diminui consideravelmente uma vez que os nós têm uma visão mais abrangente do sistema a partir da disseminação das reputações dos demais participantes.

O problema descrito nesta seção levanta as seguintes questões de pesquisa:

1. **Como controlar a contenção de recursos em um sistema P2P completamente descentralizado?**

2. **O controle da contenção de recursos é suficiente para assegurar aos nós cooperativos altos níveis de paridade?**

3. **Quão intensamente são afetadas as satisfações dos nós cooperativos e do sistema como um todo diante da redução da oferta de recursos ao sistema?**

A partir da análise dessas questões foi concebida uma solução para o controle dinâmico da oferta de recursos que possibilite validar a tese defendida neste trabalho.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são descritos os trabalhos relacionados a mecanismos de incentivo à cooperação em sistemas P2P. As contribuições do presente trabalho se restringem apenas a **mecanismos de reciprocidade**, e portanto, este capítulo foca exclusivamente mecanismos de incentivo à cooperação baseados em reciprocidade direta e indireta. Esse levantamento do estado da arte tem como principal objetivo posicionar este trabalho em relação à literatura atual para que suas contribuições fiquem em evidência.

3.1 Mecanismos de reciprocidade

Mecanismos de incentivo têm sido utilizados nos mais diferentes contextos para avaliar o comportamento de sociedades humanas [13] e animais [53]. Na área da Ciência da Computação, mais especificamente no contexto de Sistemas Distribuídos, os principais cenários de utilização de mecanismos de incentivo baseados em reciprocidade são sistemas P2P para compartilhamento de arquivos [20; 22; 24; 32; 33; 44; 46; 48; 49; 55; 58; 63; 64; 65; 67; 73; 74; 80; 81; 82; 84; 87; 88; 90; 91] e sistemas P2P para compartilhamento de recursos computacionais [7; 68; 76; 78; 92; 93], a exemplo das grades computacionais oportunistas e federações de provedores de computação na nuvem.

A métrica mais importante, utilizada unanimemente em todos os mecanismos de reciprocidade, é a *satisfação*. O que este trabalho chama de satisfação outros trabalhos se referem como: taxa de transferência (do inglês *download rate*) [49; 65; 73; 74; 87], tempo de conclusão de transferência (do inglês *download completion time*) [22;

32; 83; 84], desempenho de transferência (do inglês *download performance*) [24; 33; 88], taxa de sucesso (do inglês *success rate*) [64; 70], alta utilização (do inglês *high utilization*) [80] e eficiência (do inglês *efficiency*) [81]. Como a maioria dos trabalhos correlatos estão inseridos no contexto de compartilhamento de arquivos, o termo por eles utilizado representa o tempo de transferência destes arquivos, que quanto menor for melhor será.

Outra métrica que alguns mecanismos de reciprocidade também consideram é a *paridade* [33; 49; 63; 73; 74; 78; 80; 81; 84; 88; 91]. A definição de paridade pode variar um pouco de acordo com o contexto. Em sistemas P2P para compartilhamento de arquivos, uma interação justa entre dois nós é aquela na qual as taxas de transferência são semelhantes. Nesse contexto, a paridade tende a ser uma métrica mais imediatista e geralmente não considera contribuições de longo prazo. Por outro lado, em sistemas P2P para compartilhamento de recursos computacionais, uma interação paritária é aquela na qual dois nós contribuíram um para o outro uma quantidade similar de recursos, o que considera todo o histórico de interações passadas.

Paridade e satisfação são dois conceitos interligados por uma relação de custo-benefício. Para aumentar seu nível de satisfação um nó tende a ofertar mais recursos e consequentemente passa a ser mais altruísta. Esse aumento da oferta é justificado pela expectativa de obter mais créditos ou boa reputação com os demais nós e, no futuro, ser priorizado em detrimento de outros solicitantes. Outra justificativa para esse comportamento é a expectativa de conhecer outros nós cooperativos e expandir sua rede de colaboração, o que aumentaria as chances de ter sua requisição atendida em um momento posterior [24]. Para que isso aconteça o nó precisa adotar um comportamento mais altruísta, ofertar recursos a nós desconhecidos, e consequentemente correr o risco de prover recursos aos caronas. Quando esse risco acontece de fato, o provedor terá sua paridade afetada, uma vez que esses recursos não serão retribuídos. Por essa razão, pode-se afirmar que paridade e satisfação são medidas inversamente proporcionais, e essa relação já foi provada através de modelos matemáticos e simulações por Fan, Lui e Chiu (2009).

O principal objetivo deste trabalho é o aperfeiçoamento dos mecanismos de incentivo à cooperação baseados em reciprocidade direta, mais especificamente para o contexto de compartilhamento de recursos computacionais, com o objetivo de permitir que os participantes obtenham níveis adequados de paridade e satisfação, qualquer que seja o nível de conten-

ção da comunidade por recursos. A escolha da reciprocidade direta nesse contexto pode ser justificada pela existência de interesses simétricos entre os nós, o que leva a interações repetidas e permite a utilização de histórias privadas, evitando de forma definitiva a colusão. A partir dos trabalhos correlatos apresentados nas próximas seções é possível perceber que a maioria dos mecanismos são destinados a compartilhamento de arquivos, e apesar de alguns resolverem o problema da falta de paridade em cenários de baixa contenção, a técnica por eles utilizada (criptografia) não pode ser adaptada para um contexto de compartilhamento de recursos computacionais. Adicionalmente, são apresentados alguns mecanismos baseados em reciprocidade indireta cujos fundamentos podem ser utilizados na reciprocidade direta para evitar possível assimetria de interesses ou disponibilidades sem permitir a colusão.

A seguir, são apresentados os mecanismos de incentivo à cooperação baseados em reciprocidade direta, e posteriormente os mecanismos de reciprocidade indireta. As informações buscadas em cada trabalho são:

1. **contexto de utilização:** compartilhamento de arquivos ou compartilhamento de recursos computacionais, ou ainda um contexto genérico (quando não é explicitado no trabalho mas percebe-se que poderia ser aplicado a qualquer contexto);
2. **tipo de reciprocidade:** direta, indireta, ou uma abordagem híbrida;
3. **estratégia de implementação da reciprocidade:** algoritmos e regras utilizados para implementação do mecanismo de reciprocidade;
4. **consequências:** vantagens e desvantagens ao se utilizar o tipo de reciprocidade e estratégias de implementação escolhidos;
5. **paridade e satisfação:** eficiência da abordagem escolhida na promoção de paridade e satisfação em contextos de escassez e, principalmente, abundância de recursos.

3.1.1 Reciprocidade Direta

Compartilhamento de Arquivos

A utilização de mecanismos de incentivo à cooperação baseados em reciprocidade direta no contexto de sistemas distribuídos ganhou mais atenção da comunidade acadêmica com o surgimento do *BitTorrent*. Criado em 2001 pelo desenvolvedor norte-americano Bram Cohen, o

BitTorrent é um protocolo de rede que permite que usuários realizem transferência de arquivos indexados na *web* [1]. Dotado de topologia descentralizada, os sistemas P2P que usam o BitTorrent necessitam de mecanismos de incentivo à cooperação para que os participantes não ajam como caronas. A partir da criação do BitTorrent vários trabalhos intensificaram a exploração do aspecto da cooperação em sistemas P2P para compartilhamento de arquivos.

Neste sentido, o próprio criador do protocolo propôs uma adaptação do tradicional algoritmo *tit-for-tat*, que utiliza as taxas de transferência dos participantes com quem estiver interagindo para decidir quais deles merecem prioridade [24]. Por padrão, o BitTorrent rege que um nó provedor de recursos (também chamado de semeador, do inglês *seeder*) escolha um número fixo (o padrão é 4) de outros nós consumidores para transferir dados. A cada 10 segundos, o nó semeador utiliza as taxas com que consegue consumir a partir dos demais nós com o intuito de priorizar os mais cooperativos, *i.e.*, os que mais transferiram dados para ele. Esse cálculo é realizado através da média móvel da taxa de transferência utilizando como histórico os últimos 20 segundos, para que se obtenha maior estabilidade. Por fim, a cada 30 segundos, o nó semeador escolhe outro participante de modo completamente aleatório e altruísta, sem considerar seu histórico de contribuições, com o objetivo de descobrir novos nós que sejam mais cooperativos. Esta técnica é chamada de seleção otimista (do inglês *optimistic unchoking*).

O mecanismo de incentivo proposto por **Cohen (2003)** é baseado em reciprocidade direta por permuta imediata. Isso significa que os dois indivíduos envolvidos numa interação devem possuir interesses simétricos em um mesmo instante de tempo. Para minimizar essa restrição imposta pelo tempo e pela simetria de interesses, o protocolo propõe que os arquivos sejam particionados em vários pedaços pequenos que seriam disseminados mais rapidamente. Adicionalmente, os consumidores escolhem qual pedaço do arquivo baixar *priorizando os mais raros* com o objetivo de aumentar as possibilidades de semear para outros nós e de minimizar as chances de futuramente não conseguir consumir as partes mais raras por indisponibilidade das mesmas nos outros nós.

Uma das vantagens de se implementar a reciprocidade direta por permuta imediata é que as informações dos comportamentos de outros nós são íntegras e o imediatismo do mecanismo permite identificar e punir caronas rapidamente – os nós com baixa taxa de transferência (potencialmente caronas) tendem a ser substituídos por outros candidatos. Como as

contribuições de longo prazo não são consideradas, os caronas não têm incentivo para trocar de identidade. Ao priorizar os nós mais cooperativos nos últimos 20 segundos e distribuir a capacidade de banda equanimemente entre os consumidores mais recíprocos, o mecanismo tem como objetivo diminuir os tempos de transferência dos arquivos.

A estratégia proposta por Cohen (2003) garante, de modo geral, bons níveis de satisfação, mas em alguns cenários apresenta vulnerabilidades para assegurar paridade. Nós com capacidade de *upload* muito alta, por exemplo, tendem a criar uma situação de baixa contenção de recursos, pois eles podem ter dificuldades para encontrar nós cooperativos suficientes para alocar toda sua banda, o que favorece os caronas [73] [74]. Tais cenários tendem a gerar desequilíbrio para esses nós provedores.

O trabalho de **Fan, Lui e Chiu (2009)** analisa detalhadamente os níveis de paridade e satisfação que o mecanismo proposto por Cohen (2003) proporciona aos nós. Os autores argumentam que o mecanismo proposto por Cohen (2003) busca promover paridade através do *tit-for-tat* e satisfação através do altruísmo implementado pela *seleção otimista*. Cohen (2003) sugere que os quatro nós mais cooperativos sejam escolhidos através das taxas de transferência do *tit-for-tat* e um nó seja escolhido de modo altruísta pela seleção otimista. Fan, Lui e Chiu (2009) argumentam que o mecanismo pode alcançar diferentes níveis de custo-benefício entre paridade e satisfação ao alterar a quantidade de nós selecionados via *tit-for-tat* e via seleção otimista.

Nesse sentido, Fan, Lui e Chiu (2009) enfatizam a configurabilidade do mecanismo e traçam uma análise a partir de simulações com diferentes configurações. O resultado é previsível: quanto mais nós forem selecionados segundo o *tit-for-tat* maior paridade é alcançada, e quanto mais nós forem selecionados segundo a seleção otimista mais satisfação é alcançada, e estas duas grandezas são inversamente proporcionais. A configuração padrão proposta por Cohen (2003), por exemplo, enfatiza mais a paridade do que a satisfação.

O trabalho de Fan, Lui e Chiu (2009) levanta a discussão de que para obter paridade um nó tende a sacrificar um pouco de sua satisfação. Contudo, este trabalho não propõe uma estratégia dinâmica que busque níveis adequados de paridade e satisfação. Seguindo a linha de raciocínio deles, uma abordagem dinâmica poderia, por exemplo, determinar dinamicamente a quantidade de nós selecionados segundo o *tit-for-tat* e segundo a seleção otimista com o objetivo de elevar os níveis de paridade e satisfação. Sem essa estratégia dinâmica,

Fan, Lui e Chiu (2009) apenas aguçam o entendimento do funcionamento do mecanismo proposto por Cohen (2003), mas não resolvem o problema levantado pelo presente trabalho – a dificuldade de promover paridade em cenários com abundância de recursos.

A estratégia adotada por **Piatek *et al.* (2007)** segue a linha de raciocínio de Fan, Lui e Chiu (2009) e aprimora o mecanismo de incentivo proposto por Cohen (2003) para uma abordagem dinâmica.

Piatek *et al.* (2007) analisaram rastros de nós em um ambiente real e detectaram que muitas contribuições altruístas poderiam ser eliminadas sem que seus níveis de satisfação fossem afetados. De forma análoga a este trabalho, eles argumentam que ofertar todos os recursos disponíveis de modo indiscriminado não é a maneira mais inteligente de otimizar o desempenho dos nós, que no caso deles foca na satisfação enquanto que este trabalho prioriza a paridade. Para isso, citam como exemplo a interação entre um nó com baixa capacidade de banda de 100kb/s com outro nó com alta capacidade de banda de 1000kb/s. Apesar do nó com menos capacidade contribuir com apenas 100kb/s é possível que o usuário com mais capacidade retribua com uma taxa desproporcional de 1000kb/s. Nesta situação específica, 90% de sua contribuição seria altruísta, e portanto o nó com maior capacidade poderia diminuir sua contribuição para apenas 100kb/s sem perder satisfação.

Piatek *et al.* (2007) desenvolveram um novo cliente BitTorrent, chamado *BitTyrant*, que explora as vulnerabilidades do mecanismo proposto por Cohen (2003) visando aumentar os níveis de satisfação dos nós. A ideia é descobrir nós com alta capacidade e explorar suas potencialidades. Para isto, o BitTyrant seleciona dinamicamente os nós a serem priorizados e também ajusta dinamicamente as taxas de transferência para os selecionados, ao invés de dividir sua capacidade de transferência igualmente como proposto por Cohen (2003).

O BitTyrant seleciona os nós para doação de acordo com a razão entre dados consumidos e dados providos a cada outro nó de forma individual, selecionando-os em ordem decrescente, respeitando o limite de transferência imposto pelo algoritmo de controle de taxa de transferência, até que sua capacidade de transferência se esgote. As taxas de transferência também são definidas de maneira individual e dinâmica, de modo que o nó provedor aloque apenas a quantidade de recursos necessária a seus consumidores com o objetivo de ser retribuído por estes através do rateio do *tit-for-tat*. Assim que o nó provedor entra no rateio do *tit-for-tat* e consegue receber dados, ele continuamente ajusta a quantidade de recursos

providos para se manter nesse rateio com o menor investimento possível. Quando houver reciprocidade com o nó consumidor durante um dado intervalo de *tit-for-tat*, o nó provedor diminui a taxa de transferência para este nó no intuito de economizar banda, uma vez que existe a possibilidade de que essa economia na taxa de envio não afete sua taxa de consumo. O objetivo é utilizar essa banda economizada para estabelecer outras interações. Quando não houver reciprocidade o nó provedor elevará a taxa de transferência com o objetivo de induzir reciprocidade por parte do nó consumidor.

A priorização é realizada a partir da razão entre dados consumidos e dados providos, ou seja, a partir da paridade individual com outros nós. Isso poderia sugerir que o BitTyrant também garante paridade, mas não é o caso. O modo como as taxas de envio são controladas favorece apenas a satisfação, através da exploração das fragilidades do mecanismo proposto por Cohen (2003). A estratégia de elevar as taxas de envio quando não há reciprocidade por parte do nó consumidor, por exemplo, poderia ser facilmente explorada por um nó carona que simplesmente nunca doa, o que afetaria a paridade do nó provedor. Esse comportamento egoísta e racional dos caronas é implementado no cliente BitTorrent chamado *BitThief* [65], explicado a seguir.

Locher *et al.* (2006) também propuseram uma estratégia que, de modo similar à estratégia de Piatek *et al.* (2007), explora o altruísmo inserido pela seleção otimista do algoritmo proposto por Cohen (2003). Para validar sua estratégia, Locher *et al.* (2006) programaram um cliente BitTorrent chamado *BitThief* que implementa o comportamento de nós caronas. Os nós que utilizam o BitThief se aproveitam do altruísmo dos demais participantes cooperativos para consumir recursos sem nunca os retribuírem.

Através do cliente BitThief, o nó sempre se apresenta como um usuário recém-chegado na rede, que não possui nenhum arquivo e portanto não pode contribuir. O cliente BitThief requisita a maior quantidade de nós possíveis ao rastreador, pois quanto mais nós ele explorar maior tende a ser sua satisfação. Esta simples abordagem é suficiente para garantir aos nós que usem o BitThief taxas de transferência até maiores do que os nós que seguem a implementação original de Cohen (2003). Uma outra forma de potencializar este ataque é utilizando múltiplas identidades (do inglês *sybil attacks*), através da exploração de ampla visão (*large view exploit*, em inglês) [83].

O BitThief é um cliente racional e egoísta que provê satisfação ao nó que o utilize mas

para isso afetando consideravelmente a paridade dos demais nós. A estratégia do BitTyrant, por exemplo, é extremamente ineficiente contra os nós que usem o BitThief, pois na tentativa de induzir reciprocidade o BitTyrant elevaria as taxas de transferência para os caronas. Contudo, há de se considerar que o desempenho garantido pelo BitThief incentiva os participantes do sistema adotarem esse comportamento egoísta, e quando isto acontece massivamente o sistema tende a entrar em colapso – quando a partir de um momento todos os provedores reduzem a oferta de recursos para zero e, portanto, nenhum consumidor consegue satisfazer sua demanda.

Levin *et al.* (2008) acreditam que a política de priorizar os pedaços mais raros do arquivo pode ser explorada para manter o interesse dos demais nós por um tempo prolongado e consequentemente elevar seus níveis de satisfação. Neste sentido, Levin *et al.* (2008) sugerem que um nó provedor deve oferecer os pedaços mais raros apenas quando o nó consumidor "perder interesse" nos pedaços ofertados. Eles argumentam que a estratégia de ofertar os pedaços mais comuns (ao invés dos mais raros) poderia elevar os níveis de satisfação dos nós provedores. Contudo, de modo semelhante ao BitThief, se essa estratégia é adotada pela maioria da comunidade a satisfação global do sistema é afetada, ou seja, os tempos de transferência dos arquivos de todos os participantes tendem a aumentar.

Adicionalmente, Levin *et al.* (2008) frisam que o mecanismo proposto por Cohen (2003) não usa o *tit-for-tat* (apesar de compartilhar ideias semelhantes) e propõe que uma modelagem através de leilões representaria melhor a estratégia – os nós que dão o maior lance são contemplados. Eles argumentam que um algoritmo de compartilhamento proporcional estaria mais alinhado com o *tit-for-tat*. Nesse algoritmo, apenas o primeiro bloco seria doado de forma altruísta, substituindo a seleção otimista (considerada por eles exageradamente altruísta), e a partir daí a taxa alocada para transferência seria proporcional à taxa recebida na interação do último intervalo, como rege o *tit-for-tat*. Essa estratégia foi implementada no cliente BitTorrent chamado *PropShare*.

Explorar o altruísmo de nós que usem o PropShare é muito mais complicado, uma vez que apenas um bloco é provido altruisticamente. Embora nenhum resultado do trabalho meça os níveis de paridade dos nós, acredita-se que a estratégia implementada no PropShare assegure níveis razoáveis de paridade já que se utiliza de uma implementação mais fiel do *tit-for-tat*. Por outro lado, para obter maiores níveis de paridade é provável que os níveis de

satisfação sejam afetados. Seria interessante investigar essa relação de custo e benefício.

Um detalhe interessante é que o trabalho explora a contenção de recursos sob a ótica da raridade das partes dos arquivos espalhados na comunidade com o objetivo de elevar os níveis de satisfação. Esta parte do trabalho, em contraponto ao PropShare, visa controlar indiretamente a contenção de recursos para regular principalmente os níveis de paridade.

O *Swift*, proposto por **Tamilmani *et al.* (2004)**, é um mecanismo de reciprocidade direta baseado em créditos, projetado para sistemas P2P de compartilhamento de arquivos. Tamilmani *et al.* (2004) mostram através de simulações que os nós que "se arriscam mais" contribuindo com mais banda para os demais participantes são aqueles que recebem mais benefícios em troca. Contudo, Tamilmani *et al.* (2004) não têm a cautela de ressalvar que a paridade desses nós é negligenciada, assim como afirmam Fan, Lui e Chiu (2009).

A estratégia do *Swift* se baseia na priorização dos nós e na definição dinâmica da taxa de transferência para cada nó selecionado. Quando um nó *A* recebe recursos de outro nó *B*, *A* aumenta o crédito de *B* proporcionalmente à contribuição de *B* e, de modo similar, *A* diminui este crédito quando prover recursos a *B*. A requisição de um nó só é satisfeita se ele possuir uma quantidade de crédito maior ou igual ao tamanho do pedaço do arquivo requisitado.

No *Swift*, a taxa de transferência é ajustada dinamicamente, e o algoritmo utilizado rebusca fundamentos utilizados por Cohen (2003) e por Fan, Lui e Chiu (2009). Ao utilizar o *Swift*, cada nó precisa definir o grau de provisão de recursos via reciprocidade direta e via altruísmo. Para tal, um nó deve escolher valores entre 0 e 1 que definam o desejo de prover recursos através de reciprocidade direta por permuta imediata (α), através de altruísmo com nós conhecidos (β) e através de altruísmo com nós desconhecidos (γ). Baseado nesses valores, um nó pode ser classificado como carona ($\alpha = \beta = \gamma = 0$), paranóico ($\alpha = 1$, $\beta = 0$, $\gamma = 0$) ou aventureiro ($\alpha = 1$, $0 \leq \beta \leq 1$, $\gamma = 1$). Os resultados das simulações apontam os nós aventureiros como os mais beneficiados em termos de satisfação.

A ideia de Fan, Lui e Chiu (2009) de balancear altruísmo e reciprocidade direta é expandida no Swift com a implementação do controle dinâmico e individual das taxas de transferência para os demais nós. O Swift também se assemelha ao BitTyrant no que concerne o controle dinâmico das taxas de transferência com o objetivo de promover satisfação, mas enquanto o BitTyrant implementa uma estratégia que busca exclusivamente explorar as vulnerabilidades do mecanismo proposto por Cohen (2003), o Swift utiliza uma estratégia mais

genérica que aparenta obter bons resultados em um sistema que os nós utilizem os mais variados clientes BitTorrent.

O Swift é um mecanismo de reciprocidade direta que utiliza a permuta tardia como estratégia de priorização e a permuta imediata para controlar dinamicamente as taxas de transferência para os outros nós. Os cenários simulados mostram que quanto maior o altruísmo mais satisfação os nós terão. Contudo, o trabalho não faz nenhuma avaliação dos níveis de paridade alcançados pelos nós. Acredita-se que essa avaliação pudesse trazer resultados interessantes, uma vez que a ideia do *Swift* de controlar dinamicamente as taxas de transferência é alinhada com a ideia proposta por este trabalho.

A proposta de **Sherman, Nieh e Stein (2009)**, o *FairTorrent*, compartilha algumas semelhanças com o Swift. Ambos se baseiam em um sistema de crédito, calculado pela diferença entre recursos recebidos e recursos providos, para que um nó provedor priorize os nós consumidores que possuam mais crédito. A principal diferença é que o Swift, além da priorização, controla dinamicamente as taxas de transferência dos nós selecionados, enquanto o FairTorrent doa o máximo que puder para o nó com maior prioridade, e procede provendo o máximo possível para os nós posteriormente selecionados até que se esgote sua capacidade.

Os objetivos do FairTorrent e Swift também são diferentes. O Swift tem como principal objetivo prover satisfação, ou seja, diminuir o tempo de transferência dos arquivos. A ideia do FairTorrent é doar primeiramente para os nós aos quais se deve mais com o objetivo de balancear os níveis de paridade. O algoritmo do FairTorrent busca assegurar que a taxa de *download* de um nó convirja rapidamente para um valor aproximado à sua taxa de *upload*. Isto garante que nós que desejem fazer *upload* com uma taxa maior também terão maiores taxas de *download*. Um ponto positivo do FairTorrent é sua implementação e configuração simples, que não requer ajuste de parâmetros razoavelmente complexos como o Swift.

O FairTorrent desencoraja a troca de identidades limitando a quantidade de dados que um nó recém-chegado pode receber sem retribuir; estratégia semelhante à do PropShare. O lado ruim dessa estratégia é que esse limite pode desencorajar a entrada de novos membros na comunidade.

O FairTorrent também é vulnerável em cenários com baixa contenção de recursos. Um nó com capacidade de *upload* muito alta pode doar a caronas se em determinado momento não existir nós cooperativos em estado consumidor com interesse em seus arquivos. Neste

sentido, o mecanismo de priorização do FairTorrent só garante paridade em cenários de contenção de recursos – problema abordado no presente trabalho.

O trabalho de Capota *et al.* (2011) tem ideias bastante alinhadas com as ideias do presente trabalho. Capota *et al.* (2011) analisaram a satisfação e paridade de mecanismos de reciprocidade em sistemas BitTorrent através da decomposição da provisão de recursos em duas etapas: priorização dos nós e alocação da quantidade de banda mais apropriada. Para a alocação de banda, os autores formalizam o problema como um desafio de maximização do fluxo em grafos (que representam a rede P2P) através da programação linear, e utilizam o algoritmo de otimização de paridade chamado *max-min fairness* [20]. Os resultados das simulações corroboram com a linha de investigação do presente trabalho, apontando que as estratégias de priorização apesar de simples são eficientes, mas a estratégia experimentada para alocação de banda tem um baixo desempenho quando comparada a algoritmos centralizados, apontando uma linha de investigação pouco explorada.

Outro conceito aplicado em alguns mecanismos de reciprocidade para sistemas de compartilhamento de arquivos é a *criptografia* [82; 88]. A ideia é utilizar a criptografia para forçar a reciprocidade ao invés de simplesmente esperar a decisão futura de retribuição por parte dos outros nós.

As estratégias de incentivo à cooperação propostas pelo ***Treat-Before-Trick*** **[82]** e ***Quota-Encryption*** **[88]** compartilham fundamentos semelhantes em relação à utilização da criptografia. Os arquivos são divididos e criptografados pelo nó provedor e a chave (simétrica) de criptografia só é fornecida aos consumidores mediante reciprocidade.

No Treat-Before-Trick, um nó deve enviar um arquivo completo e criptografado para estar apto a receber a chave de criptografia. No Quota-Encryption, é possível definir uma quota para especificar a quantidade de blocos criptografados trocados via *tit-for-tat* para que seja enviada a chave de criptografia. O Treat-Before-Trick chama este algoritmo de "dado antes, chave depois" (DFKL, do inglês *Data First, Key Later*). A estratégia utilizada em ambos é semelhante à proposta de Cohen (2003), reutiliza a ideia do *tit-for-tat* e seleção otimista acrescentando a criptografia dos arquivos.

A principal vantagem desses esquemas é que eles asseguram bons níveis de paridade a participantes de sistemas P2P para compartilhamento de arquivos. Por outro lado, para obter paridade todos os participantes devem abrir mão de um nível razoável de satisfa-

ção. Uma restrição é que esquemas baseados em criptografia só funcionam para sistemas de compartilhamento de arquivos. Por exemplo, não faz sentido e não é possível criptografar uma máquina virtual (no contexto de federações de nuvens) para liberar a chave apenas mediante reciprocidade. Talvez essa estratégia até funcione para grades computacionais, através da criptografia do resultado de um processamento, mas a entrega da chave mediante reciprocidade inviabilizaria o modelo de negócios uma vez que a provisão de recursos é mais demorada e acontece com menos frequência do que em sistemas de compartilhamento de arquivos. Mesmo no contexto de compartilhamento de arquivos, clientes BitTorrent que funcionam com criptografia requerem que os demais clientes estejam cientes da existência desse novo protocolo de segurança que impõe o envio e recebimento de chaves. Isso tira a liberdade individual de escolha de qual cliente BitTorrent os nós desejam utilizar. Por último, apesar de alguns autores [81; 82; 88] argumentarem que o processo de criptografia é extremamente leve, esta etapa adiciona sobrecarga ao mecanismo de incentivo.

A seguir, são apresentadas as abordagens baseadas em reciprocidade direta para promover cooperação em sistemas P2P para compartilhamento de recursos computacionais.

Compartilhamento de Recursos Computacionais

De acordo com Anderson (2007), o principal precursor de sistemas P2P para compartilhamento de recursos computacionais foram os projetos de recursos computacionais públicos (*public-resource computing*, em inglês) que surgiram em meados de 1990 com os projetos *Distributed.net* [2] e *GIMPS* [3]. Outro projeto de recurso computacional público bastante conhecido é o *SETI@home* [6], que em 2002 já se utilizava de mais de 1 milhão de computadores domésticos para processamento de sinais em busca de inteligência extraterrestre. Obviamente, esses projetos não carecem de mecanismos de incentivo à cooperação uma vez que os participantes doam seus ciclos ociosos de processamento de forma deliberadamente voluntária, sem esperar algo em troca.

No contexto de compartilhamento de recursos computacionais existem as grades computacionais e as federações de provedores de computação na nuvem. O surgimento da computação na nuvem foi impulsionado pela evolução das técnicas de virtualização que possibilitam customização e rápida alocação/desalocação dos recursos além de pagamento sob demanda.

A partir do surgimento desse novo paradigma, as grades computacionais estão paulatinamente dando lugar às federações de provedores de computação na nuvem, uma vez que este último provê maior liberdade aos usuários.

No que concerne as grades computacionais, a maioria das abordagens para promover cooperação são monetárias [45; 52; 59; 60; 61; 72; 89]. Em relação a federações de nuvens, um *survey* realizado por Assis e Bittencourt (2016) aponta que os trabalhos adotam abordagens monetárias [41; 69; 40], ou estratégias mais flexíveis para cada membro da federação [21; 56; 77]. Nesta última abordagem, os participantes geralmente utilizam arquivos de configuração para determinar políticas que auxiliam na decisão de quais membros da federação devem requisitar ou conceder recursos, pressupondo uma característica mais voluntária por parte dos membros dessas federações [10].

Ainda no âmbito de federações, trabalhos mais recentes [78; 39] utilizam mecanismos de credibilidade com topologia centralizada para realizar a combinação entre provedores e consumidores. Este componente central, por possuir todas as informações do sistema, consegue penalizar nós caronas com mais facilidade. A ideia é essencialmente semelhante à ideia dos mecanismos de reciprocidade indireta, pois também consegue combinar participantes que não cooperariam segundo a reciprocidade direta, mas possui a restrição de uma arquitetura centralizada.

Outro trabalho interessante, no contexto de mecanismos de incentivo à cooperação em federações de nuvens foi proposto por Khan *et. al.* (2015). A abordagem consiste em um *mecanismo de incentivo baseado em esforço*, onde o esforço é definido como a contribuição de um nó em relação a sua capacidade. Este mecanismo é inspirado no modelo econômico ParEcon (abreviação para "Economia Participativa") que foca no bem estar social ao considerar as desigualdades entre os participantes. Neste modelo, nós com diferentes capacidades contribuem com diferentes quantidades mas obtém a mesma recompensa se eles compartilharem o máximo que puderem. Para isto, Khan *et. al.* (2015) mencionam que o mecanismo utiliza um sistema de créditos gerenciado por super-nós, mas não mencionam como é possível certificar que os nós não estão mentindo sobre sua capacidade total.

Apenas dois trabalhos abordaram o problema da cooperação sob a ótica da reciprocidade direta, a **Rede de Favores [7; 9]** e o ***Weighted* [76]**.

A *Rede de Favores* e o *Weighted* são mecanismos baseados em reciprocidade direta que

foram projetados para grades computacionais P2P oportunistas. Ao contrário do Swift, e de modo similar ao FairTorrent, ambos se baseiam em reciprocidade direta por permuta tardia, modalidade mais apropriada para sistemas P2P nos quais os nós interagem entre si repetidamente, como é o caso das grades computacionais e federações de nuvens.

Na Rede de Favores e no Weighted, os créditos são computados de maneira semelhante ao Swift e FairTorrent, diferença entre recursos consumidos e doados, mudando apenas a forma como os recursos são contabilizados (poder computacional multiplicado pelo tempo de uso) [79]. A arquitetura proposta pelo Weighted é um pouco mais detalhista e organiza os nós em grupos controlados por super-nós (*superpeer*, em inglês) que, por possuir informações privilegiadas dos estados dos nós componentes do seu grupo, podem alocar as requisições nos nós pertencentes ao grupo de forma otimizada. A capacidade, distância e crédito dos demais super-nós são utilizados no processo de priorização para doação.

A Rede de Favores e o Weighted desencorajam a troca de identidades pois não permitem que os créditos dos demais nós assumam valores negativos, o que poderia incentivar uma reentrada no sistema para automaticamente limpar seus débitos. Adicionalmente, a função de crédito de ambos considera um termo histórico que diferencia nós recém-chegados de nós cooperativos previamente conhecidos.

A função primordial dos sistemas baseados em crédito, a exemplo da Rede de Favores, Weighted, Swift e FairTorrent, é a priorização dos nós mais cooperativos, uma espécie de recompensa por comportamentos colaborativos no passado. O modo como os recursos computacionais são providos aos consumidores é semelhante às estratégias do Swift e FairTorrent: o nó provedor doa todos os recursos disponíveis para o nó a quem deve mais, e se após essa doação ainda houver recursos ociosos, procede de maneira semelhante com os demais nós respeitando a ordem de prioridade.

A Rede de Favores e o Weighted focam em prover altos níveis de satisfação a seus nós. Nestes tipos de mecanismos, os níveis de paridade são afetados consideravelmente em cenários de baixa contenção por recursos, uma vez que apenas a priorização não é suficiente para impedir os nós caronas de consumirem recursos [27; 29; 30]. Este trabalho propõe como solução o controle dinâmico da oferta, de forma individual por cada nó, com o objetivo de controlar a contenção de recursos e evitar o aproveitamento dos caronas.

Na próxima seção são apresentados os mecanismos de incentivo à cooperação baseados

em reciprocidade indireta.

3.1.2 Reciprocidade Indireta

Compartilhamento de Arquivos

Com relação a mecanismos de incentivo à cooperação baseados em reciprocidade indireta as principais estratégias utilizam sistemas de reputação, sistemas de crédito transitivo ou o princípio de passar o favor adiante (do inglês *pay-it-forward*).

No contexto da utilização de criptografia para assegurar paridade, a seguir, são apresentados o *Dandelion* [84] e o *T-Chain* [81], baseados respectivamente em um *sistema de créditos global* e na ideia do *pay-it-forward*.

O Dandelion, criado por **Sirivianos, Yang e Jarecki (2009)**, permite compartilhamento de arquivos através do *tit-for-tat* aliado a um sistema de créditos global. O mecanismo utiliza a criptografia para contabilizar a quantidade de dados enviados de maneira acurada, evitando ataques de nós maliciosos. Sempre que um nó A enviar dados a outro nó B os dados são enviados criptografados. Para descriptografar este bloco de dados, o nó B deve requisitar a chave ao servidor, uma entidade central confiável. A requisição da chave é usada como garantia de que A transferiu dados para B, e portanto, neste momento, a entidade central computará um certo crédito para A e débito para B.

O sistema de créditos também serve para aumentar os níveis de satisfação quando existe assimetria de interesses ou de capacidades. Quando um nó não é capaz de retribuir o favor ao nó provedor, seja porque não possui banda suficiente para transferência ou pelo simples fato de não possuir o conteúdo que o outro nó deseja em troca, o nó provedor receberá créditos como recompensa. Os créditos podem ser utilizados para posterior consumo de recursos com nós desconhecidos via reciprocidade indireta.

Quando uma entidade centralizada é incluída no mecanismo, o desafio do algoritmo deixa de ser distribuído e passa a ser local, se torna apenas um problema de combinação otimizada entre nós consumidores e provedores. Essa simplificação não deveria justificar a desvantagem de introduzir um componente central no sistema, que limita a capacidade de escalabilidade e se constitui um ponto único de falha.

Shin *et al.* (2015) criaram o *Triangle Chaining* (*T-Chain*), um mecanismo que utiliza

criptografia e reciprocidades direta e indireta para garantir paridade aos nós em qualquer cenário de contenção de recursos. Neste mecanismo, a criptografia de pedaços dos arquivos serve para garantir reciprocidade e consequentemente paridade. O principal objetivo da reciprocidade indireta, implementada através de *pay-it-forward*, é atenuar os efeitos da assimetria de interesses que acontece com alta frequência em sistemas P2P para compartilhamento de arquivos.

No esquema proposto, um nó *A* envia um arquivo criptografado a um nó *B* e informa a quem *B* deve retribuir esse serviço para receber a chave. Se *A* e *B* possuem interesses simétricos, então *A* se auto-designa como receptor da retribuição de *B*. *B* só irá receber a chave de criptografia após retribuir diretamente para *A*. Se *A* e *B* não possuem interesses simétricos, *A* designa um outro nó *C* como receptor do arquivo transferido por *B*. *C* é escolhido pois *A* sabe que *B* possui algum arquivo de interesse de *C*. Como retribuição do serviço prestado por *A*, *B* envia parte do arquivo que interessa *C*, também criptografado. Neste ponto, *C* deve notificar *A* de que *B* "pagou" pelos serviços de *A*, e então *A* disponibilizará a chave para *B* descriptografar o pedaço provido por *A*. Adicionalmente, *B* informa um nó *D* ao qual *C* deve retribuir esse serviço, de modo que *C* se torne elegível para receber a chave para descriptografar o pedaço provido por *B*; e seguindo essa abordagem a cadeia de serviços se expandirá. Contudo, se *B* não conhecer um nó *D* que possua interesse em algum arquivo de *C*, então *B* deve enviar um arquivo não-criptografado a *C*, o que terminaria a cadeia de serviços.

A criptografia de arquivos aplicada no T-Chain é eficiente para assegurar paridade em quaisquer que sejam os cenários de contenção de recursos. Por outro lado, essa ideia só faz sentido em sistemas P2P para compartilhamento de arquivos, pois arquivos são criptografáveis. O *pay-it-forward* é uma boa alternativa para a assimetria de interesses, porém, dá brechas à colusão por parte de nós maliciosos. No exemplo anterior, *B* e *C* poderiam formar um conluio para que *A* liberasse a chave para *B* sem que *B* houvesse prestado qualquer serviço para *C*. Posteriormente, *B* e *C* alternariam e conseguiriam consumir recursos de outros nós.

No contexto de atenuar os efeitos da assimetria de interesses em sistemas P2P para compartilhamento de arquivos, mas ainda sem utilizar sistemas de reputação (*i.e.* informações de terceiros), podem ser citados o ***CompactPSH*** **[16]**, o ***PledgeRoute*** **[62]**, o ***Scrivener*** **[70]**, o ***Cluster Based Incentive Mechanism*** **(CBIM) [90]**, ***K-Cycle*** **[32]**, e o trabalho de **Me-**

nasché, Massoulié e Towsley (2010). Menasché, Massoulié e Towsley (2010) propõem converter uma interação que aconteceria via reciprocidade indireta em duas interações via reciprocidade direta. O CBIM e o K-Cycle organizam os nós com interesses assimétricos no que os autores chamam de *anéis* [90] ou *ciclos* [32] de permuta via escambo, forçando os nós que compõem o anel a contribuírem para poder consumirem. No Scrivener, PledgeRoute e CompactPSH a reciprocidade indireta é decomposta em duas etapas: *transferência de crédito* e reciprocidade direta. A ideia é fortalecer a economia do sistema uma vez que a transferência de créditos possibilita interações que não aconteceriam entre nós com interesses assimétricos.

No **Scrivener [70]**, os créditos locais entre vizinhos são usados para interagir com nós mais distantes sem que o nó consumidor aparente ser carona. Para isto, o Scrivener ajuda um nó consumidor a identificar um *caminho de crédito* para um nó com o arquivo desejado. Naturalmente, cada nó pertencente ao caminho de crédito deve possuir crédito com o nó sucessor. Uma vez que o caminho de crédito seja identificado os nós ajustam todos os créditos no caminho de tal modo que o nó provedor (no fim do caminho) deixe de dever a seu predecessor dessa rota e passe a dever exclusivamente ao nó consumidor no início do caminho. Uma série de débitos onde B deve a A, C deve a B, e assim por diante até que Z deva a seu predecessor, seriam todos substituídos por um único débito direto de Z para A. Neste momento, Z pode cancelar seu débito com A através de reciprocidade direta, provendo o arquivo desejado por A. Os autores chamam este esquema utilizado no Scrivener de *troca transitiva* (*transitive trading*, em inglês).

O **PledgeRoute [62]** também tem como objetivo permitir a transferência de crédito para possibilitar interações entre nós com assimetria de interesses. O mecanismo permite que nós contribuam para um conjunto de nós e transfiram essas contribuições para obter serviços de outros nós via reciprocidade direta. Visando prover maior satisfação (menores tempos de transferência), Landa *et al.* (2009) [62] modelaram a transitividade de contribuições como um problema de roteamento que busca os caminhos de contribuições que provejam maior vazão na interação entre os nós consumidor e provedor.

Bocek *el. al.* (2009) afirmam que o mecanismo utilizado no **CompactPSH [16]** é baseado em histórias privadas aliadas a histórias compartilhadas, o que poderia permitir colusão e trapaças através de histórias forjadas. Contudo, o autor não percebeu que apesar do Com-

pactPSH compartilhar histórias, os nós maliciosos não conseguem utilizar este compartilhamento de informações para benefícios próprios.

No CompactPSH, dois nós interagem via reciprocidade direta (*tit-for-tat*) quando possuem interesses simétricos. Diante da assimetria de interesses, quando por exemplo um nó *A* tem interesse nos conteúdos de *C* mas a recíproca não é verdadeira, *C* não interage diretamente com *A* pois *A* seria para *C* um carona. Neste sentido, a fim de viabilizar interação com *A*, o nó *C* provê para *A* uma lista de nós em débito com *C* de modo que o nó *A* possa utilizar em seu benefício o crédito que outro nó possui com *C*. Para tal, *A* verificará se existe nessa lista algum nó *B* em débito com *A*, de modo que *B* proveja um certificado que autorize *A* utilizar os créditos de *B* com *C*. Neste momento, caso *B* aceite a solicitação de *A* (e assim ele o fará somente se estiver em débito com *A*), *B* anotaria uma certa quantidade de créditos para *C*, podendo quitar ou não a dívida de *C* com *B*; *B* anotaria a mesma quantidade em forma de débito para *A*, podendo ou não quitar sua dívida com *A*; *A* anotaria a mesma quantidade de créditos para *B*, podendo ou não quitar a dívida de *B*; e *A* anotaria a mesma quantidade de débito para *C*, que só será quitado mediante prestação de serviços de *C* para *A*. De posse do certificado de *B*, *A* consegue interagir com *C* via reciprocidade direta, eliminando desse modo a assimetria de interesses.

O **CBIM [90]** difere de outros trabalhos que utilizam anéis de permuta via escambo pois ao invés de uma abordagem estática, seu algoritmo de formação do anel é dinâmico. Adicionalmente, o trabalho utiliza um mecanismo de identificação de caronas, mecanismo este que os autores dizem ser baseado em reputação mas na verdade não o é, uma vez que apenas informações locais são utilizadas para identificação dos caronas. Sempre que um anel for quebrado os integrantes deste anel são adicionados a uma lista local de nós suspeitos. Essa lista é utilizada para que um nó evite ingressar em outros anéis com nós suspeitos.

O ***K-Cycle* [32]** utiliza a mesma abordagem do CompactPSH em relação ao compartilhamento de listas que contêm nós com quem já interagiu, seus interesses, e conteúdos possuídos por eles para viabilizar a interação entre nós com interesses assimétricos. De maneira similar ao CBIM, a reciprocidade indireta só acontecerá se existir ciclos (ou anéis) de compartilhamento nos quais os nós interajam via permuta imediata.

Menasché, Massoulié e Towsley (2010) sugerem que ciclos que permitam interação via reciprocidade indireta sejam transformados em reciprocidade direta. Para isso um nó *A* re-

ceberia do rastreador um conjunto de informações que o permitiria identificar os interesses de um nó *C*, nó este que possui arquivos de interesse de *A*. A ideia é que *A* consiga por si só identificar a razão da assimetria de interesses com *C*, e eliminá-la através do consumo de conteúdos com um outro nó *B*, mesmo que esse conteúdo não seja do interesse de *A*. Posteriormente, *A* utilizaria esses recursos recém-consumidos para interagir via reciprocidade direta com *C*. Isso permite que *A* continue interagindo com *C* mesmo que *B* quebre esse ciclo. Em contrapartida, *A* irá gastar o dobro de banda que gastaria se estivesse interagindo com *C* sem precisar consumir recursos de *B*.

O Scrivener, PledgeRoute, e CBIM conseguem driblar a assimetria de interesses com um esquema que utiliza apenas informações locais, baseadas na experiência individual de cada nó. O CompactPSH e o K-Cycle utilizam história compartilhada mas de maneira limitada, apenas para permitir a transitividade de crédito, e esta abordagem não abre possibilidades de colusão. Em contraponto à estratégia da transitividade de créditos existem os sistemas de reputação, ou seja, sistemas nos quais um nó coleta a partir de outros nós (que julga serem confiáveis) informações acerca de comportamento dos demais nós. Essa abordagem também elimina a assimetria de interesses, pois não requer encontros repetidos e a construção de uma história privada, porém, dá oportunidades à colusão.

Sistemas de reputação diferem principalmente no modo como as reputações são calculadas e propagadas. De modo geral, cada nó avalia os nós de acordo com comportamentos passados e propaga essa avaliação quando requisitado por outros nós. Algumas vulnerabilidades decorrem do baixo custo para criar novas identidades, e isto pode incentivar a reentrada no sistema para se desvincular de maus comportamentos do passado (*whitewashing*), além de abrir possibilidade de colusão via ataques de identidade (*sybil attacks*, em inglês), quando nós fictícios são criados para se auto-avaliarem positivamente mesmo quando nenhum serviço foi prestado, e de forma semelhante também é possível fazer ataques de calúnia (*slander attacks*, em inglês), quando os nós agem coletivamente para difamar nós cooperativos.

A ideia do **Eigentrust [55]** é usar a confiança local que cada nó armazena em relação aos demais e agregar esses valores de forma global, para que todos os nós tenham uma visão semelhante da reputação dos nós do sistema. Basicamente, cada nó avalia as interações com um nó com nota 1 ou -1, e baseado nelas estabelece uma relação de confiança com cada nó com quem interagiu. Para convergir para um valor global, cada nó continuamente requisita as

opiniões dos nós mais confiáveis, e como todos os nós fazem isso com uma certa frequência as reputações armazenadas localmente convergem para um valor similar.

O mecanismo proposto pelo **ARA [48]** é dotado de um sistema global de créditos voláteis (expiram depois de um dado tempo) que pode ser auditado pelos nós interessados. Os nós interessados em um nó *A* são todos os nós que interagiram ou pretendem interagir com *A*. O sistema de créditos pode ser considerado global pois as contribuições feitas por um nó *A* para o nó *B* podem ser utilizadas por *A* como crédito para consumir recursos de um outro nó *C*. Para assegurar que os nós maliciosos não alterem as contribuições feitas por eles mesmos ou por outros nós, o ARA (*A Robust Audit*) se baseia em um esquema de auditoria de créditos que permite um nó checar a integridade das informações com outros nós para descobrir inconsistências.

Como o presente trabalho não foca mecanismos de reciprocidade indireta baseados em reputação, esta seção (3.1.2) se limita a prover apenas uma visão geral desta abordagem. A partir de uma revisão de literatura é possível perceber que os trabalhos que usam reputação abordam o tema de modo semelhante ao *EigenTrust* [55], tratando da agregação e disseminação das reputações [22; 46; 64; 91], ou modelam o problema como o *ARA* [48], que utiliza um sistema de crédito para representar a reputação [44; 49; 58].

Compartilhamento de Recursos Computacionais

No contexto de compartilhamento de recursos computacionais foram encontrados apenas mecanismos baseados em sistemas de reputação projetados para grades computacionais oportunistas [68; 92; 93].

3.2 Considerações

A partir da revisão de literatura realizada pode-se perceber que a maioria dos mecanismos de reciprocidade são utilizados em sistemas P2P para compartilhamento de arquivos. Apesar dos contextos serem diferentes, os fundamentos utilizados nesses mecanismos podem ser reutilizados ou adaptados para sistemas P2P para compartilhamento de recursos computacionais. Um exemplo disto é a utilização de créditos transitivos por mecanismos de reciproci-

dade direta com o objetivo de atenuar a assimetria de interesses sem dar brechas à colusão (problema recorrente em mecanismos de reciprocidade indireta). Até onde se sabe, esta ideia ainda não foi abordada no contexto de compartilhamento de recursos computacionais.

A Tabela 3.1 lista os trabalhos relacionados classificando-os por tipo de reciprocidade, recurso compartilhado, promoção de satisfação e de paridade em cenários de baixa e alta contenção por recursos.

Naturalmente, ao promover cooperação, todos os mecanismos buscam garantir a satisfação dos nós, ou seja, buscam garantir que os nós tenham suas requisições atendidas (e o mais breve possível). A paridade em cenários de alta contenção por recursos é alcançada como efeito colateral dos mecanismos de reciprocidade. Em cenários de alta contenção, a baixa capacidade ofertada aliada à alta demanda e à *priorização dos nós mais cooperativos* realizada pelo mecanismo é suficiente para evitar a provisão de recursos a caronas. Se os recursos são doados a nós cooperativos, a probabilidade de haver reciprocidade é alta e isto tende a assegurar bons níveis de paridade. Por outro lado, apenas a priorização de nós cooperativos não é suficiente para promover paridade em cenários de baixa contenção por recursos.

Quando visaram paridade, nenhum mecanismo percebeu a contenção de recursos como um fator chave e portanto não exploraram essa perspectiva. Mecanismos de criptografia [82; 88; 84; 81] resolvem a paridade, com o controle estrito da liberação de chaves, mas não são genéricos o suficiente para serem aplicados em qualquer contexto, como por exemplos grades computacionais ou federações de nuvens. O PropShare, o Swift, e o trabalho de Capota *et al.* (2011) são os únicos mecanismos que, de modo similar à nossa proposta, controlam a oferta de capacidade, porém, não se sabe a efetividade de seus mecanismos uma vez que eles não foram avaliados considerando diferentes níveis de contenção.

No próximo capítulo é apresentada a solução proposta: um mecanismo baseado na teoria do controle retroalimentado que, quando utilizado por mecanismos de reciprocidade direta, regula a oferta dos recursos e assegura paridade em cenários com diferentes níveis de contenção de recursos.

Trabalho	Reciprocidade	Recurso	Satisfação	Paridade (Contenção)	
				Baixa	Alta
Cohen [24]	direta	arquivo	X		X
BitTyrant [73; 74]	direta	arquivo	X		X
BitThief [65]	direta	arquivo	X		
PropShare [63]	direta	arquivo	X	X	X
Swift [87]	direta	arquivos	X	X	X
FairTorrent [80]	direta	arquivo	X		X
Capota *et al.* [20]	direta	arquivo	X	X	X
Treat-Before-Trick [82]	direta	arquivo	X	X	X
Quota-Encryption [88]	direta	arquivo	X	X	X
Rede de Favores [7]	direta	computacional	X		X
Weighted [76]	direta	computacional	X		X
Dandelion [84]	indireta	arquivo	X	X	X
T-Chain [81]	indireta	arquivo	X	X	X
Scrivener [70]	indireta	arquivo	X		X
PledgeRoute [62]	indireta	arquivo	X		X
CompactPSH [16]	indireta	arquivo	X		X
CBIM [90]	indireta	arquivo	X		X
K-Cycle [32]	indireta	arquivo	X		X
Menasché *et al.* [67]	indireta	arquivo	X		X
Eigentrust [55] e [22; 46; 64; 91]	indireta	arquivo	X		X
ARA [48] e [44; 49; 58]	indireta	arquivo	X		X
[68; 92; 93]	indireta	computacional	X		X
Transitive FD-NoF [1]	direta e indireta	computacional	X	X	X

[1] Mecanismo de reciprocidade resultante deste trabalho. [28; 30]

Tabela 3.1: Resumo dos mecanismos de reciprocidade classificados por tipo, contexto de utilização, promoção de satisfação e de paridade em cenários de baixa e alta contenção por recursos.

Capítulo 4

Solução Proposta

Uma das maneiras de evitar que caronas consumam recursos, o que afetaria a paridade de nós cooperativos, é regular a contenção de recursos mantendo-na em um nível intermediário, onde a quantidade de recursos ofertada seja semelhante à quantidade de recursos demandada por nós cooperativos. Contudo, a maioria dos mecanismos de reciprocidade foi projetada para arquiteturas completamente descentralizadas, e por esta razão não existe uma maneira simples de controlar diretamente os níveis de contenção de recursos no sistema.

A partir da definição de paridade (Equações 2.3 e 2.4, pg 20), pode-se perceber que o modo mais simples de aumentá-la é através da diminuição da quantidade de recursos ofertada ao sistema. Porém, embora paridade seja uma métrica importante, também é preciso considerar os níveis de satisfação, pois quanto menor a quantidade de recursos disponibilizada ao sistema, menor serão os níveis de satisfação de maneira geral, incluindo as satisfações dos nós cooperativos.

Este trabalho propõe que cada nó cooperativo deveria controlar de modo individual a capacidade de recursos ofertada ao sistema. Espera-se como resultado que a partir da ação coletiva dos nós cooperativos, quando todos eles controlam a quantidade de recursos ofertada com o objetivo de obter melhores níveis de paridade, a contenção de recursos seja indiretamente regulada para um dado nível que segregue os caronas.

Neste capítulo é apresentado o mecanismo proposto para conciliar satisfação e paridade em federações de provedores de nuvens. Esse mecanismo consiste em um *Laço de Controle Retroalimentado*, ou abreviadamente, *controlador*, com o objetivo de regular a quantidade de recursos que cada nó deveria ofertar aos demais nós. Este controlador é particularmente

importante para cenários com baixo nível de contenção de recursos, de modo que evite a oferta exagerada por parte de nós cooperativos. Uma implementação para este controlador é apresentada na próxima seção. Posteriormente é descrita a instanciação do controlador proposto em um mecanismo de incentivo à cooperação baseado em reciprocidade direta, a *Rede de Favores*.

4.1 O Controlador de Capacidade

A ideia de qualquer mecanismo baseado em *laços de controle retroalimentado* é monitorar periodicamente o sistema e utilizar as métricas coletadas para regular algumas configurações do sistema e alcançar determinados objetivos [50]. Neste trabalho, um nó cooperativo monitora sua paridade para regular a quantidade de recursos ofertada e alcançar níveis específicos de paridade.

Suponha que cada nó cooperativo tenha capacidade total de recursos $\mathcal{C} \in \mathbb{R}_{\geq 0}$. Considere que a quantidade de recursos que um nó $A \in \mathbb{F}_c$ disponibiliza para qualquer outro nó $B \in \mathbb{F}$ seja expressada por uma função degrau $\alpha : \mathbb{F}_c \times \mathbb{F} \times \mathbb{N} \rightarrow \mathbb{R}_{[0,\mathcal{C}]}$. Nesse sentido, o objetivo do mecanismo é sugerir para A a quantidade de recursos $\alpha(A, P, t), \forall P \in \mathbb{F} \wedge \forall t \in \mathbb{N}$ que deve ser ofertada para que A alcance os patamares desejados de paridade.

O controlador de capacidade proposto requer que cada nó cooperativo A defina dois valores, um limite mínimo $\mathcal{L}_{min} \in \mathbb{R}_{\geq 0}$ e um limite máximo $\mathcal{L}_{max} \in \mathbb{R}_{\geq 0}$, estabelecendo portanto um intervalo $[\mathcal{L}_{min}, \mathcal{L}_{max}]$ que denota os níveis desejados de paridade. O controlador funciona da seguinte maneira. Inicialmente, $\alpha(A, P, 0) = \mathcal{C}, \forall P \in \mathbb{F}$. Se em um dado passo de tempo $t, t > 0$, a paridade de A em relação a um nó B for inferior ao limite mínimo, *i.e.* $\phi(A, B, t) < \mathcal{L}_{min}$, o controlador de capacidade irá continuamente reduzir $\alpha(A, B, t)$ (enquanto $\alpha(A, B, t) > 0$) nos passos de tempo subsequentes até que $\phi(A, B, t)$ seja superior a $\mathcal{L}_{min}$; linhas 4 e 5 do Algoritmo 1. Por outro lado, se $\phi(A, B, t) > \mathcal{L}_{max}$, o controlador de capacidade ofertada aumentará o valor $\alpha(A, B, t)$ (enquanto $\alpha(A, B, t) < \mathcal{C}$) nos passos de tempo subsequentes até que $\phi(A, B, t)$ assuma um valor inferior a $\mathcal{L}_{max}$ – uma vez que o nível desejado de paridade tenha sido alcançado, o mecanismo tentará aumentar os níveis de satisfação; linhas 6 e 7 do Algoritmo 1. Finalmente, quando $\mathcal{L}_{min} \leq \phi(A, B, t) \leq \mathcal{L}_{max}$, o controlador de capacidade executará um algoritmo de subida de encosta (*hill climbing*, em

inglês) [51] que consiste em utilizar os valores mais recentes de paridade para decidir se a capacidade ofertada deve ser aumentada ou diminuída, para maximizar a paridade de A com B dentro do intervalo $[\mathcal{L}_{min}, \mathcal{L}_{max}]$; linhas 8 e 9 do Algoritmo 1.

A abordagem descrita não se aplica a um nó B para o qual A nunca doou recursos, uma vez que neste caso $\phi(A, B, t)$ é indefinida. Uma abordagem simples que A poderia seguir nestes casos seria sempre ofertar toda sua capacidade ($\mathcal{C}$). Contudo, um carona poderia se aproveitar de nós cooperativos ao mudar sua identidade depois de cada interação com A, fingindo ser um nó recém-chegado. Para lidar com esse problema, A determina a quantidade máxima de recursos oferecidos aos nós que tenham paridade indefinida, *i.e.*, nós com os quais A não tenha interagido anteriormente, rodando o controlador de capacidades em um contexto global, usando sua paridade em relação à federação como um todo; linhas 10 a 16 do Algoritmo 1. Note que isto só irá acontecer depois que A tiver feito sua primeira doação a algum nó. Portanto, o controlador de capacidade só é executado no começo do passo de tempo t se $\sum_{P \in \mathbb{F}} \sum_{i=0}^{t-1} \upsilon(A, P, i) > 0$, caso contrário $\alpha(A, P, t) = \mathcal{C}, \forall P \in \mathbb{F}$.

O pseudo-código apresentado pelo Algoritmo 1, executado por um nó cooperativo A no começo de um dado passo de tempo $t, t > 0$, quando A já tiver feito sua primeira doação à federação, descreve o controle de capacidade rodado por A; Δ é um parâmetro que define o quanto o montante máximo de recursos ofertado por A deve ser aumentado ou diminuído em cada passo de tempo.

Note que através deste mecanismo um nó cooperativo poderia momentaneamente se recusar a prover recursos à federação até que receba uma certa quantidade de recursos de outros nós. Em um caso extremo, quando o número de caronas for consideravelmente alto, o uso do algoritmo pelos nós cooperativos poderia levar a uma deserção definitiva da federação, que como resultado entraria em colapso.

Por fim, há de se considerar a versatilidade do controlador provida através da configuração de $\mathcal{L}_{min}$ e $\mathcal{L}_{max}$. Se algum nó em um dado momento mudar seu objetivo e desejar maximizar a satisfação, por exemplo, bastaria configurar $\mathcal{L}_{min} = \mathcal{L}_{max} = 0$ para anular o controlador e ofertar sempre toda capacidade máxima à federação.

Algoritmo 1 Algoritmo do controlador de capacidades para o nó A no início do passo de tempo $t, t > 0 \wedge \sum_{P \in \mathbb{F}} \sum_{i=0}^{t-1} \upsilon(A, P, i) > 0$.

1: $aumente \leftarrow falso$
2: **para cada** P $\in \mathbb{F}$ **faça**
3: **se** $\sum_{i=0}^{t-1} \upsilon(A, P, i) > 0$ **então** ▷ nós com paridade definida
4: **se** $\phi(A, P, t) < \mathcal{L}_{min}$ **então**
5: $aumente \leftarrow falso$
6: **senão se** $\phi(A, P, t) > \mathcal{L}_{max}$ **então**
7: $aumente \leftarrow verdadeiro$
8: **senão se** $\phi(A, P, t) \leq \phi(A, P, t-1)$ **então**
9: $aumente \leftarrow !aumente$
10: **fim se**
11: **senão** ▷ nós com paridade indefinida
12: **se** $\phi(A, t) < \mathcal{L}_{min}$ **então**
13: $aumente \leftarrow falso$
14: **senão se** $\phi(A, t) > \mathcal{L}_{max}$ **então**
15: $aumente \leftarrow verdadeiro$
16: **senão se** $\phi(A, t) \leq \phi(A, t-1)$ **então**
17: $aumente \leftarrow !aumente$
18: **fim se**
19: **fim se**
20: **se** $aumente$ **então**
21: $\alpha(A, P, t+1) \leftarrow min(\mathcal{C}, \alpha(A, P, t) + \Delta)$
22: **senão**
23: $\alpha(A, P, t+1) \leftarrow max(0, \alpha(A, P, t) - \Delta)$
24: **fim se**
25: **fim para**

4.2 A Rede de Favores

Para verificar a tese de que um controlador da quantidade de recursos ofertada é suficiente para conciliar satisfação e paridade em federações P2P através do controle da contenção de

recursos, é preciso instanciá-lo em um mecanismo de reciprocidade. Como o objetivo deste trabalho é garantir paridade em federações P2P de provedores de computação na nuvem, a reciprocidade direta por permuta tardia é a abordagem mais adequada. No contexto de mecanismos baseados em reciprocidade direta por permuta tardia para promover cooperação em sistemas P2P para compartilhamento de recursos computacionais, na literatura foram encontradas apenas a *Rede de Favores* e o *Weighted*, ambos para grades computacionais P2P oportunistas.

Os dois mecanismos são bastante semelhantes. Ambos priorizam os nós baseados em um sistema de créditos que não permite valores negativos e cuja função utiliza um termo histórico para diferenciar os nós recém-chegados de nós cooperativos já conhecidos. A Rede de Favores (NoF, do inglês *Network of Favors*) foi escolhida pois é mais genérica do que o *Weighted*, uma vez que este utiliza a noção de super-nós que é utilizada em uma proporção menor do que redes formadas apenas por nós, e portanto tende a ser mais útil e ter maior aplicabilidade em outros trabalhos.

Na NoF, um favor consiste na alocação de um recurso para um nó solicitante. Mecanismos de reciprocidade por permuta tardia se baseiam na expectativa de que os participantes irão eventualmente retribuir os favores recebidos.

O valor de um favor é associado com o tempo e poder de processamento fornecidos. Assume-se que cada nó mantém um registro local do valor total de todos os favores concedidos e recebidos de cada nó com quem interagiu no passado. Sempre que um nó concede ou recebe um favor, o valor balanceado de todos os favores trocados com o nó com quem colabora é atualizado. Com base nesses valores, cada nó calcula um valor de crédito local associado aos nós com quem colabora. Para um nó A, o crédito do nó B só aumenta se A receber algum favor de B. O crédito atual que o nó A tem relacionado a outros nós competindo por seus recursos é usado por A para decidir a qual dos nós solicitantes será fornecido o recurso. Com esta abordagem o nó A sempre priorizará a concessão de favores aos nós cooperativos. Isso garante que sempre que houver contenção por um recurso de propriedade de A, A priorizará os solicitantes com maior crédito.

Reutilizando as notações definidas na Seção 2.2 (pg 18) para formalização do problema, onde $v(A, B, t)$ representa o valor total dos favores que A doou para B no passo de tempo t, e $\gamma(A, B, t)$ é uma função que denota o crédito de B na visão de A no começo do passo de

tempo t, a definição mais simples para a função de crédito é $\gamma(A,B,t) = \sum_{i=0}^{t-1} \upsilon(B,A,i) - \upsilon(A,B,i)$, definida anteriormente na Equação 2.2.

Na NoF, colusão não é um problema uma vez que os créditos são mantidos localmente, de maneira autônoma por cada nó, o que evita que nós maliciosos consigam manipular seus valores de crédito. No entanto, um nó pode ser capaz de manipular seu crédito alterando sua identidade e aparecendo como um nó recém-chegado, sem interação prévia com os outros participantes da federação. Para resolver esse problema, Andrade *et al.* (2004) propuseram que a NoF deveria forçar o crédito de cada nó a ser não-negativo, da seguinte forma:

$$\gamma(A,B,t) = max\left\{0, \sum_{i=0}^{t-1} \upsilon(B,A,i) - \upsilon(A,B,i)\right\}. \tag{4.1}$$

Essa definição previne a priorização de nós que maliciosamente mudam suas identidades ao invés de nós cooperativos que tenham consumido mais recursos do que contribuído no passado. Com isso, um carona não teria incentivo em trocar sua identidade uma vez que isto não mudaria o modo como os outros nós o priorizariam, já que seu crédito será sempre zero na perspectiva de um nó cooperativo. Entretanto, um nó A não conseguiria distinguir um nó B carona que nunca doou recursos, de um nó cooperativo C que já doou para A no passado mas consumiu de A pelo menos a mesma quantidade, *i.e.*, $\upsilon(A,C,t) \geq \upsilon(C,A,t) \wedge \upsilon(C,A,t) > 0, \forall C \in \mathbb{F}_c \wedge \forall t \in \mathbb{N}$. Para distingui-los, Andrade *et al.* (2004) introduziram na definição da função de crédito $\gamma(A,B,t)$ um identificador histórico que leva em consideração a quantidade de doações que A recebeu de B no passado. Para evitar uma grande diferença entre os créditos de nós cooperativos já conhecidos e de nós recém-ingressados no sistema, o que complicaria a entrada desses novos participantes, a NoF usa uma função sub-linear de $\upsilon(B,A,t)$ como o identificador histórico na definição de $\gamma(A,B,t)$, que *e.g.* poderia ser:

$$\gamma(A,B,t) = max\left\{0, \sum_{i=0}^{t-1} \upsilon(B,A,i) - \upsilon(A,B,i) + \sqrt{\upsilon(B,A,i)}\right\} \tag{4.2}$$

ou

$$\gamma(A,B,t) = max\left\{0, \sum_{i=0}^{t-1} \upsilon(B,A,i) - \upsilon(A,B,i) + \log \upsilon(B,A,i)\right\}. \tag{4.3}$$

Na NoF, um nó sempre oferta todos os seus recursos ociosos para o sistema. O mecanismo de priorização estabelece que em caso de contenção pelos recursos de A, *i.e.*, quando dois ou mais nós desejam obter os recursos de A, A os listará em ordem decrescente de acordo com seus respectivos valores de crédito, e tentará doar todos os recursos pedidos pelo

primeiro nó da lista, que é o nó com maior crédito na lista de A. Os nós são servidos um por um de acordo com sua posição na lista até que todos os recursos sejam alocados, ou então até que todos os nós solicitantes tenham sido atendidos. Se dois ou mais nós tiverem a mesma quantidade de créditos, A irá compartilhar igualmente os recursos disponíveis entre eles até que os recursos se esgotem ou que ambos os nós tenham obtido a quantidade de recursos solicitada.

Para fins de ilustração, considere uma federação com apenas seis nós, A, B, C, D, E e F, onde os nós B, C, D, E e F são ordenados por A de acordo com seus créditos na visão de A no passo de tempo t, como mostrado na Tabela 4.1.

Nó P	$\sum_{i=0}^{t-1} \upsilon(P, A, i)$	$\sum_{i=0}^{t-1} \upsilon(A, P, i)$	$\gamma(A, P, t)$ via Equação 4.3
D	100	80	**22**
E	100	80	**22**
B	1000	990	**13**
C	0	300	**0**
F	100	102	**0**

Tabela 4.1: Exemplo da tabela NoF mantida por A no passo de tempo t.

Suponha que no passo de tempo t o nó A disponibilize 20 unidades de recurso para a federação. Se nesse momento todos os demais nós estiverem requisitando recursos, estando os nós C, D e E requisitando 5 unidades de recursos cada, e os nós B e F 4 unidades cada, então A alocaria 5 unidades para D e 5 unidades para E, e depois alocaria 4 unidades para B, e finalmente 3 unidades para C e 3 unidades para F.

Se o nó C for considerado carona, uma vez que até o passo de tempo t apenas consumiu recursos sem retribuir, a contenção de recursos estaria baixa com $\kappa < 1$. De maneira geral, no fim da tabela estarão misturados nós cooperativos que consumiram igualmente ou pouco mais do que doaram (*e.g.*, F) e nós caronas que apenas consumiram recursos (*e.g.*, C). Sempre que $\kappa < 1$, alguns dos nós (ou todos) que estão no fim da tabela NoF também conseguirão consumir recursos, e quando estes forem caronas a paridade de A será afetada.

Acredita-se que uma maneira de atenuar esta vulnerabilidade seria controlando a quantidade de recursos que um nó cooperativo deveria ofertar aos demais nós. Essa abordagem é descrita a seguir no contexto da Rede de Favores.

4.2.1 A Rede de Favores Dirigida a Paridade

Na NoF, um nó cooperativo sempre oferta todos os seus recursos ociosos ao sistema, sem considerar os níveis de paridade, com o objetivo de acumular mais créditos com outros participantes e portanto aumentar as chances de receber favores no futuro. Isto garante os melhores níveis possíveis de satisfação aos nós cooperativos, qualquer que seja o nível de contenção de recursos, e por este motivo, esta NoF é chamada de *NoF Dirigida a Satisfação* ou, abreviadamente, SD-NoF (do inglês *Satisfaction-Driven NoF*).

No entanto, ofertar sempre todos os recursos ociosos pode prejudicar a paridade dos nós cooperativos em cenários de baixa contenção de recursos. Nestes cenários o controlador de recursos poderia evitar que nós cooperativos provessem recursos a caronas. A *Rede de Favores Dirigida a Paridade* (ou simplesmente FD-NoF, do inglês *Fairness-Driven Network of Favors*) é basicamente o mecanismo da NoF executando concorrentemente com o controlador de oferta de recursos proposto por este trabalho.

Em suma, o controlador pode ser representado de modo simples por uma coluna a mais na tabela NoF, que traria a quantidade máxima de recursos que o nó cooperativo deveria ofertar a cada outro nó solicitante. Esses valores são atualizados dinamicamente de tempos em tempos de acordo com o Algoritmo 1 (pg 48). Na FD-NoF, a Tabela 4.1 da NoF seria aprimorada para a Tabela 4.2[a], apresentada a seguir.

Nó P	$\sum_{i=0}^{t-1} \upsilon(P, A, i)$	$\sum_{i=0}^{t-1} \upsilon(A, P, i)$	$\gamma(A, P, t)$ via Equação 4.3	$\alpha(A, P, t)$
D	100	80	22	$\mathcal{C}$
E	100	80	22	$\mathcal{C}$
B	1000	990	13	$\mathcal{C}$
C	0	300	0	0
F	100	102	0	$0.95 \cdot \mathcal{C}$

Tabela 4.2: Exemplo da tabela FD-NoF mantida por A no passo de tempo t.

Suponha que A tenha configurado o controlador com $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 1$ e $\Delta = 0.05 \cdot \mathcal{C}$. Uma vez que D, E e B são nós cooperativos nos quais os níveis de paridade de A com cada um deles é superior a 1, é natural que o controlador oferte todos os recursos

[a]Os valores de $\alpha(A, P, t)$ são meramente ilustrativos.

disponíveis ($\mathcal{C}$) para eles. F é um nó cooperativo cuja paridade na visão de A é ligeiramente inferior a 1, *i.e.* $\phi(A, F, t) < 1$. Nesta situação, o controlador estaria executando o algoritmo de subida de encosta, o que poderia culminar em $\alpha(A, F, t) = 0.95 \cdot \mathcal{C}$. Como C é um carona, e $\phi(A, C, t) = 0 < \mathcal{L}_{min}$, o controlador provavelmente reduziria $\alpha(A, C, t)$ para valores incrementais de t por 5% de $\mathcal{C}$ até que $\alpha(A, C, t) = 0$, e A deixaria de cooperar com C por falta de reciprocidade – mesmo que nesse dado momento A tivesse recursos disponíveis para doar para C.

4.3 Considerações

Neste capítulo foi descrita a proposta deste trabalho para assegurar aos nós cooperativos níveis adequados de satisfação e paridade, em cenários nos quais a oferta de recursos é maior do que a demanda dos nós cooperativos, o que consequentemente favorece os caronas. Foi concebido um algoritmo baseado na teoria do controle retroalimentado, que utiliza as medidas mais recentes de paridade para controlar a quantidade de recursos ofertada ao sistema, evitando uma oferta exagerada e consequentemente o aproveitamento por parte dos caronas. De modo geral, o controlador de recursos é genérico o suficiente para ser utilizado em conjunto com qualquer mecanismo de reciprocidade direta (ou até indireta, mas com menor impacto positivo).

Este trabalho reutilizou a noção da Rede de Favores, que quando instanciada juntamente com o controlador de recursos é chamada de Rede de Favores Dirigida a Paridade. O funcionamento da Rede de Favores Dirigida a Paridade foi ilustrado através de um exemplo de uma federação com seis participantes, dos quais apenas um é carona, e por isso, em um dado momento, o controlador de recursos reduz a oferta para esse nó para zero.

Para obter mais detalhes sobre o desempenho dos nós cooperativos em diferentes cenários, com diferentes níveis de contenção por recursos e diferentes proporções de caronas, este trabalho recorre a simulações, descritas e apresentadas no próximo capítulo.

Capítulo 5

Avaliação

Para avaliar o mecanismo proposto foi desenvolvido um simulador para um modelo simplificado de uma federação P2P de provedores de computação na nuvem. O objetivo é analisar o comportamento e desempenho dos participantes cooperativos e caronas em cenários com diferentes níveis de contenção de recursos. O simulador[a] pode ser utilizado para avaliar tanto a *Satisfaction-Driven NoF* como a *Fairness-Driven NoF*, de modo que seja possível tirar conclusões sobre as vantagens e desvantagens de cada abordagem para os nós cooperativos.

Neste capítulo são apresentados o modelo de simulação, cenários simulados e os resultados, seguidos por suas respectivas análises.

5.1 Modelo de Simulação

A federação $\mathbb{F}$ consiste em uma comunidade de $\mathcal{N}$ nós, com $(1 - \mathcal{F}) \cdot \mathcal{N}$ nós cooperativos e $\mathcal{F} \cdot \mathcal{N}$ caronas, $0 \leq \mathcal{F} < 1$. A simulação acontece em intervalos de tempo discretos, que correspondem aos passos de tempo das funções degrau definidas nas Seções 2.2, 4.1 e 4.2 (pg 18, 46 e 48). Em cada passo, cada participante pode estar ou em *estado consumidor* com probabilidade $\mathcal{P}$, ou em *estado provedor* com probabilidade $1 - \mathcal{P}$. Sempre que estiver em estado consumidor, cada nó envia primeiramente requisições para os nós com quem ele sabe que possui crédito, e se nenhum destes puder atender sua requisição, ele passa a enviar

[a]O código-fonte do simulador da SD-NoF e FD-NoF, bem como os dados de entrada e saída reportados na tese, estão disponíveis publicamente no sítio `https://github.com/eduardolfalcao/nof-simulator`.

requisições a nós desconhecidos, escolhidos de forma aleatória, com o intuito de estabelecer novos vínculos. Em cada passo de tempo, diante do recebimento das requisições dos nós consumidores, o simulador seleciona de forma aleatória um nó em estado provedor, e este nó prioriza a concessão de recursos para os nós consumidores que possuem mais crédito com ele, seguindo o protocolo estabelecido pela NoF. Deste modo, os nós provedores são sorteados até que toda a demanda dos nós consumidores seja atendida ou até que todos os nós provedores sejam consultados.

Assume-se que cada nó tem capacidade total de recursos $\mathcal{C}$. Quando estiver em estado provedor no passo de tempo t, um nó A irá prover até $\alpha(A, B, t)$ unidades de recurso para qualquer nó B que esteja em estado consumidor no mesmo passo de tempo t. Na simulação da SD-NoF, $\alpha(A, P, t) = \mathcal{C}, \forall P \in \mathbb{F} \wedge \forall t \in \mathbb{N}$. Por outro lado, na simulação da FD-NoF, $\alpha(A, P, t)$ pode variar de 0 a $\mathcal{C}$, de acordo com os resultados do algoritmo de controle de capacidade (Algoritmo 1, pg 48).

Um nó em estado consumidor em qualquer passo de tempo t irá requisitar $\mathcal{D}_{fed} + \mathcal{C}$ unidades de recursos para atender sua demanda local, das quais $\mathcal{C}$ unidades são provenientes de recursos locais e $\mathcal{D}_{fed}$ unidades são requisitadas à federação. Em cada passo de tempo, os nós cooperativos priorizam os nós requisitantes que tenham o maior crédito de acordo com o protocolo da NoF, como explicado na Seção 4.2 (pg 48).

Nessas simulações deseja-se avaliar os cenários que são mais favoráveis aos caronas. Portanto, para tornar ainda mais complicada a discriminação entre caronas e nós cooperativos que momentaneamente tenham consumido mais do que tenham doado, o identificador histórico da função de crédito é desconsiderado. Neste sentido, a NoF implementada no simulador utiliza a Equação 4.1 (pg 50) para o cálculo de $\gamma(A, B, t)$. Desse modo, um nó cooperativo A não consegue distinguir um nó carona C de um outro nó cooperativo B que até o passo de tempo t tenha consumido uma quantidade de recursos maior ou igual à quantidade que doou, *i.e.*, $\upsilon(A, B, t) \geq \upsilon(B, A, t)$, uma vez que $\gamma(A, B, t) = \gamma(A, C, t) = 0$. Isto retira a necessidade de investigar cenários nos quais os caronas inicialmente proveriam alguns recursos antes de se tornarem caronas, ou de algum outro modo alternariam entre o comportamento carona e cooperativo. Não obstante, alguns cenários simulados consideram nós cooperativos nos quais a proporção entre oferta e demanda pode variar amplamente.

5.2 Cenários

Os cenários são pensados levando em consideração a relação entre recursos ofertados e recursos demandados por nós cooperativos, uma vez que o objetivo é analisar o desempenho dos nós em federações com diferentes níveis de contenção por recursos. Lembrando que a contenção de recursos é calculada através da razão entre recursos requisitados e ofertados, considerando apenas os nós cooperativos, na simulação da SD-NoF, o valor esperado do nível de contenção em cada cenário é dado por:

$$\mathbb{E}[\kappa] = \frac{\mathcal{P} \cdot (1 - \mathcal{F}) \cdot \mathcal{N} \cdot \mathcal{D}_{fed}}{(1 - \mathcal{P}) \cdot (1 - \mathcal{F}) \cdot \mathcal{N} \cdot \mathcal{C}} = \frac{\mathcal{P} \cdot \mathcal{D}_{fed}}{(1 - \mathcal{P}) \cdot \mathcal{C}}. \tag{5.1}$$

Para fins de simplificação, mas sem perda de generalidade, é assumido que $\mathcal{C} = 1$, e $\mathcal{P}$ e $\mathcal{D}_{fed}$ são variados no intuito de conceber cenários com níveis de contenção baixo ($\mathbb{E}[\kappa] \in \{0.25, 0.5, 0.75\}$), moderado ($\mathbb{E}[\kappa] = 1$), alto ($\mathbb{E}[\kappa] = 2$) e muito alto ($\mathbb{E}[\kappa] = 4$).

Os mesmos cenários foram simulados na SD-NoF e FD-NoF, para permitir a comparação entre eles. Note, contudo, que na FD-NoF um nó cooperativo pode reduzir a quantidade de recursos ofertada a outros nós para valores inferiores a $\mathcal{C}$. Portanto, nas simulações da FD-NoF, o valor de $\mathbb{E}[\kappa]$ define apenas um limite inferior para o valor esperado da contenção que os nós irão experimentar no sistema durante a simulação. Porém, a notação $\mathbb{E}[\kappa]$ também será utilizada nas simulações da FD-NoF, simplesmente para identificar os diferentes níveis iniciais de contenção dos cenários analisados.

Também há de ser considerado o fato de que, dado um valor de $\mathcal{F}$, quanto maior o número de participantes da federação, menos benéfica será a situação para os caronas. Isto acontece porque quanto maior a quantidade de nós cooperativos, menor será o desvio do valor esperado da quantidade de nós provedores em cada passo de tempo, *i.e.*, $\mathcal{N} \cdot (1 - \mathcal{F}) \cdot (1 - \mathcal{P})$, e portanto menor será a probabilidade de que uma quantidade significativa de recursos seja disponibilizada para os caronas em um dado momento. Por isto, todos os cenários foram configurados com $\mathcal{N} = 200$.

Os cenários mais interessantes são aqueles nos quais o impacto dos caronas no sistema seja o maior possível. Portanto, nas simulações realizadas, os nós caronas são configurados com $\mathcal{P} = 1$ e $\mathcal{D}_{fed} = \infty$, *i.e.*, eles estão sempre tentando consumir a maior quantidade de recursos possível. Dois níveis de caronas foram considerados nas simulações: $\mathcal{F} = 0.25$ (moderado) e $\mathcal{F} = 0.75$ (muito alto). Adicionalmente, também foram simulados cenários

com $\mathcal{F} = 0$ para fins de comparação com os cenários nos quais $\mathcal{F} > 0$. Desta maneira, será possível compreender o impacto exato dos caronas na federação.

No que concerne a configuração do controlador de capacidade, foram configurados $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$ e $\Delta = 0.05 \cdot \mathcal{C}$ para todos os nós cooperativos. Por fim, um total de 5000 passos foram executados em cada simulação, o que se mostrou suficiente para levar o sistema a um estado estável.

Resumindo, o projeto de experimentos inclui as seguintes constantes: número total de nós ($\mathcal{N} = 200$), capacidade total de recursos dos nós cooperativos ($\mathcal{C} = 1$), limites inferior e superior e constante de variação para o algoritmo de controle de capacidades da FD-NoF ($\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$ e $\Delta = 0.05 \cdot \mathcal{C}$), e o número de passos da simulação (5000 passos). Além disso, o projeto considera três variáveis: quantidade de recursos demandada ($\mathcal{D}_{fed}$), probabilidade de estar em estado consumidor ($\mathcal{P}$) e a proporção de nós caronas no sistema ($\mathcal{F}$). Para facilitar a análise do desempenho dos nós cooperativos e nós caronas com relação às variáveis independentes, os experimentos foram divididos em dois grupos de projeto diferentes, um envolvendo nós homogêneos, e o outro envolvendo nós heterogêneos.

5.2.1 Federação Homogênea

A princípio, deseja-se entender como os nós cooperativos desempenham em relação aos caronas. Portanto, para eliminar o impacto que diferentes tipos de nós cooperativos teriam nos resultados, foi assumido que todos eles têm a mesma probabilidade de estar em estado consumidor ($\mathcal{P}$), além de possuírem as mesmas necessidades de demanda ($\mathcal{D}_{fed}$). Todos os caronas também são idênticos, com $\mathcal{D}_{fed} = \infty$ e $\mathcal{P} = 1$. Este cenário de nós homogêneos foi simulado com $\mathcal{F} = 0.25$ e $\mathcal{F} = 0.75$.

Para cada valor de $\mathbb{E}[\kappa]$ no conjunto $\{0.25, 0.5, 0.75, 1, 2, 4\}$ foram simulados 30 cenários diferentes. Para investigar a influência de $\mathcal{D}_{fed}$ e $\mathcal{P}$ utilizamos um projeto de varredura de parâmetros que que definiu cenários com valores extremos e intermediários para as duas variáveis. Os primeiros 15 cenários foram gerados através da escolha de 15 valores para $\mathcal{D}_{fed}$ igualmente distribuídos no intervalo $[1..3]$; para cada valor de $\mathcal{D}_{fed}$ foram calculados os valores de $\mathcal{P}$ correspondentes aos seis valores de $\mathbb{E}[\kappa] \in \{0.25, 0.5, 0.75, 1, 2, 4\}$ usando a Equação 5.1, *i.e.*, $\mathcal{P} = \mathbb{E}[\kappa]/(\mathbb{E}[\kappa] + \mathcal{D}_{fed})$. Os outros 15 cenários foram gerados de modo similar mas escolhendo primeiramente 15 valores para $\mathcal{P}$ igualmente distribuídos no

intervalo $[0.2..0.8]$, e depois configurando $\mathcal{D}_{fed}$ de acordo com $\mathbb{E}[\kappa] \cdot (1 - \mathcal{P})/\mathcal{P}$, segundo a Equação 5.1, para cada valor de $\mathbb{E}[\kappa]$ em $\{0.25, 0.5, 0.75, 1, 2, 4\}$.

5.2.2 Federação Heterogênea

É natural esperar que os participantes de uma federação tenham comportamentos diferentes no que concerne a relação entre a quantidade de recursos demandados e ofertados à federação. Para simular este cenário, fixou-se $\mathcal{D}_{fed}$, a demanda dos nós cooperativos, e variou-se $\mathcal{P}$, a probabilidade de estar em estado consumidor. Para tal, foi utilizada uma função de distribuição normal truncada com valor médio 0.5, desvio padrão 0.1, limitada inferiormente por 0 e superiormente por 1, de onde foram obtidos os valores de $\mathcal{P}$ para cada nó cooperativo.

Esses experimentos também contemplaram cenários com diferentes níveis de contenção de recursos. Em cada caso um valor diferente de $\mathcal{D}_{fed}$ foi escolhido para todos os nós cooperativos de tal modo que os níveis esperados de contenção de recursos $\mathbb{E}[\kappa]$ assumissem os valores 0.5, 1, 2 e 4. Uma vez que a média da distribuição de onde os valores de $\mathcal{P}$ foram obtidos é 0.5, partindo da Equação 5.1 tem-se que $\mathcal{D}_{fed} = \mathbb{E}[\kappa]$. Portanto, os valores escolhidos para $\mathcal{D}_{fed}$, foram 0.5, 1, 2 e 4. De modo similar ao caso dos nós homogêneos, a simulação para cada nível de contenção foi replicada 30 vezes, cada uma com diferentes valores de $\mathcal{P}$ para os nós cooperativos.

Para melhor entendimento do impacto real dos caronas no desempenho dos nós cooperativos em ambas SD-NoF e FD-NoF, foram simulados dois níveis de caronas: $\mathcal{F} = 0$ e $\mathcal{F} = 0.25$. Para $\mathcal{F} = 0$, ao invés de $\mathcal{N} = 200$ a federação foi configurada com $\mathcal{N} = 150$, com o objetivo de manter o mesmo número de nós com o mesmo comportamento em ambos os casos. Adicionalmente, os cenários configurados com $\mathcal{F} = 0$ podem ser interpretados de forma diferente. Ao invés de ser composto por caronas "puros" que nunca doam, é mais provável que um sistema P2P real seja composto por nós que são apenas parcialmente caronas, no sentido de que demandam muito mais recursos do que oferecem. Portanto, essas simulações podem ser analisadas considerando que quando $\mathcal{P} > 0.5$ o nó é tão mais carona quanto maior for $\mathcal{P}$, ou em outras palavras, um nó será mais carona quanto maior for seu consumo em relação a sua oferta. Quando $\mathcal{P} < 0.5$, então o nível de altruísmo do nó vai depender de como o mecanismo de controle é configurado. Em suma, quanto menor é $\mathcal{P}$ e menor é $\mathcal{D}_{fed}$, maior a importância desse mecanismo para que o nó obtenha níveis adequados de paridade.

5.3 Resultados e Análise

Os resultados e análises dos desempenhos dos nós em cenários formados apenas por nós homogêneos são apresentados a seguir. Posteriormente, é realizada uma análise de sensibilidade dos parâmetros do controlador para compreender suas influências no desempenho dos nós. Por fim, são apresentados os cenários formados por nós heterogêneos.

5.3.1 Federação Homogênea

Para cenários nos quais os nós são homogêneos, uma combinação crítica de baixos valores de $\mathbb{E}[\kappa]$ com altos valores de $\mathcal{F}$ pode tornar a federação inviável para os nós cooperativos. Nos cenários em que $\mathbb{E}[\kappa] = 0.25$, até mesmo uma proporção moderada de caronas ($\mathcal{F} = 0.25$) pode fazer com que a federação entre em colapso. Quando $\mathbb{E}[\kappa] = 0.5$, um terço dos cenários compostos por uma alta proporção de caronas ($\mathcal{F} = 0.75$) entraram em colapso, mais especificamente os cenários nos quais os nós cooperativos foram configurados com $\mathcal{P} \leq 0.14$ ou $\mathcal{D}_{fed} \leq 0.59$. A estratégia utilizada pelos nós para obterem níveis adequados de paridade é a limitação da quantidade de recursos ofertada a outros nós, que consequentemente pode levar a uma debandada em massa da federação. Uma vez que nos cenários apresentados nesta seção todos os nós são homogêneos, todos eles reduziriam em poucos passos de tempo a quantidade de recursos ofertada à federação para $\alpha(A, P, t) = 0$, o que levaria à uma deserção em massa.

Vale salientar que nestes cenários o controlador está configurado com $\Delta = 0.05 \cdot \mathcal{C}$, $\mathcal{L}_{min} = 0.75$ e $\mathcal{L}_{max} = 0.95$. A configuração do controlador também tem influência sobre os colapsos acontecidos nos casos mencionados anteriormente. A Seção 5.3.2 apresenta uma análise de sensibilidade acerca das influências de Δ, $\mathcal{L}_{min}$ e $\mathcal{L}_{max}$.

Na Figura 5.1 é possível observar os níveis de paridade e satisfação dos nós cooperativos através de suas correspondentes médias e intervalos de confiança (nível de confiança de 95%) no último passo das simulações com $\mathbb{E}[\kappa]$ assumindo os valores do conjunto $\{0.5, 0.75, 1, 2, 4\}$. Adicionalmente, também é possível comparar seus desempenhos na SD-NoF e FD-NoF, em cenários configurados com $\mathcal{F} = 0.25$ e $\mathcal{F} = 0.75$. Todos os cenários nos quais a federação colapsa (todos nos quais $\mathbb{E}[\kappa] = 0.25$ e alguns nos quais $\mathbb{E}[\kappa] = 0.5$ e $\mathcal{F} = 0.75$) foram excluídos da Figura 5.1 para fins de comparação entre os dois cenários

com diferentes níveis de carona.

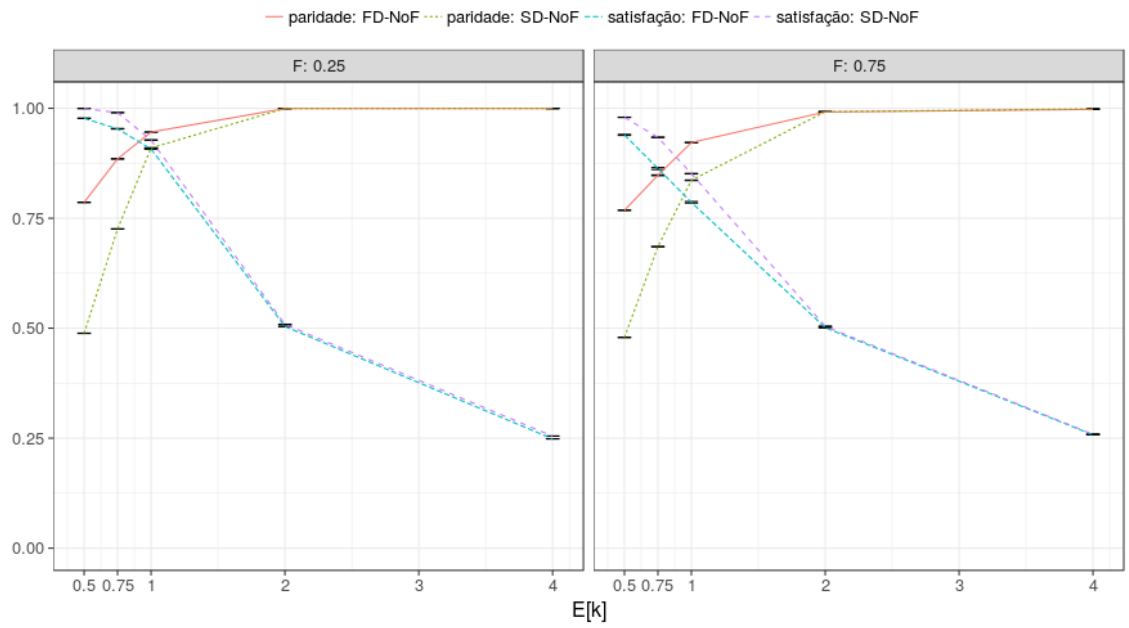


Figura 5.1: Média e intervalo de confiança para paridade e satisfação no passo de tempo 5000. Neste cenário, todos os nós cooperativos possuem características homogêneas e as simulações foram executadas na SD-NoF e FD-NoF.

Em relação ao percentual de nós caronas ($\mathcal{F}$), a principal conclusão que pode ser tirada da Figura 5.1 é que, fora o fato da federação ser mais propensa a entrar em colapso quando $\mathcal{F}$ é alto, não há muita diferença entre o impacto que os nós caronas causarão aos níveis de paridade e satisfação dos nós cooperativos – os recursos excedentes sempre serão consumidos por caronas, já que eles possuem $\mathcal{D}_{fed} = \infty$. Considerando que não faz sentido avaliar cenários em que a quantidade de caronas inviabilizam a federação, cenários em que os nós cooperativos desertam graças ao controlador de capacidades, o restante das simulações foi executado com $\mathcal{F} = 0.25$.

A Figura 5.1 também mostra que a SD-NoF é eficiente em garantir altos níveis de paridade para nós cooperativos em cenários de alta contenção de recursos tais como $\mathbb{E}[\kappa] = 2$ ou $\mathbb{E}[\kappa] = 4$. No entanto, também é possível observar que a SD-NoF não é suficiente para garantir paridade em cenários com baixa contenção de recursos, apesar de assegurar altos níveis de satisfação. São nestas situações que o controlador de capacidade entra em cena. Ele aumenta a paridade de nós cooperativos em cenários de contenção baixa a moderada

($0.5 \leq \mathbb{E}[\kappa] \leq 1$) enquanto os níveis de satisfação são apenas levemente reduzidos. Para $\mathbb{E}[\kappa] \in \{0.5, 0.75, 1\}$ a FD-NoF apresentou um aumento de respectivamente 61.2%, 17.9% e 4.4% de paridade, com uma redução de apenas 2%, 4% e 2.2% de satisfação; para $\mathbb{E}[\kappa] > 1$, os níveis médios de paridade e satisfação permanecem iguais. Mais detalhes sobre a dispersão dos níveis de paridade e satisfação dos nós cooperativos no último momento da simulação, para diferentes níveis de contenção de recursos, podem ser visualizados nos diagramas de caixa apresentados na Figura 5.2.

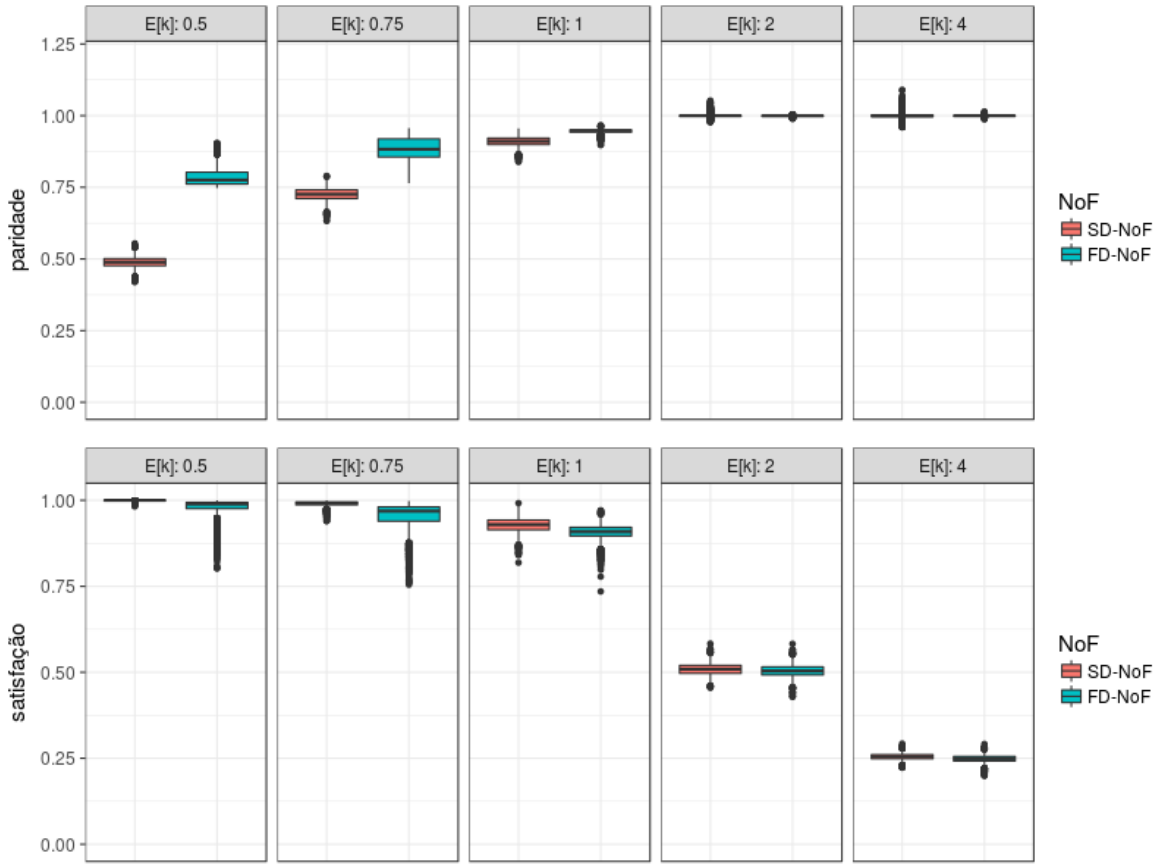


Figura 5.2: Diagramas de caixa com os níveis de paridade e satisfação dos nós cooperativos, no último passo das simulações, executadas na SD-NoF e FD-NoF para $\mathcal{F} = 0.25$.

Esta melhoria se deu basicamente pela regulação do nível de contenção de recursos, possibilitada pelo controlador de capacidade. Esse fato pode ser conferido na Figura 5.3, que mostra a evolução da contenção de recursos $\kappa(t)$ nos cenários configurados com $\mathcal{F} = 0.25$, para $0 < t \leq 5000$. Dado que no modelo de simulação utilizado os caronas consomem todos os recursos que sobram na federação, a quantidade de recursos doados será sempre

igual à quantidade de recursos ofertados. Deste modo, a contenção de recursos é calculada em cada passo de tempo t através da razão entre a quantidade total de recursos requisitados por nós cooperativos e a quantidade total de recursos doados por nós cooperativos, ambas no passo de tempo t. É preciso adotar essa abordagem pois em cada passo de tempo cada nó cooperativo oferece diferentes quantidades de recursos a diferentes nós, e portanto não haveria outro modo de descobrir a quantidade de recursos ofertada por um nó cooperativo em um determinado passo de tempo. Dada a ampla variação nos valores de $\kappa(t)$ medidos em cada passo, plotou-se médias móveis para $\kappa(t)$ considerando os últimos 500 passos. Cada cenário simulado é representado por uma curva no gráfico – quanto mais as curvas se aproximarem de $\kappa(t) = 1$, melhores serão os níveis de paridade dos nós cooperativos.

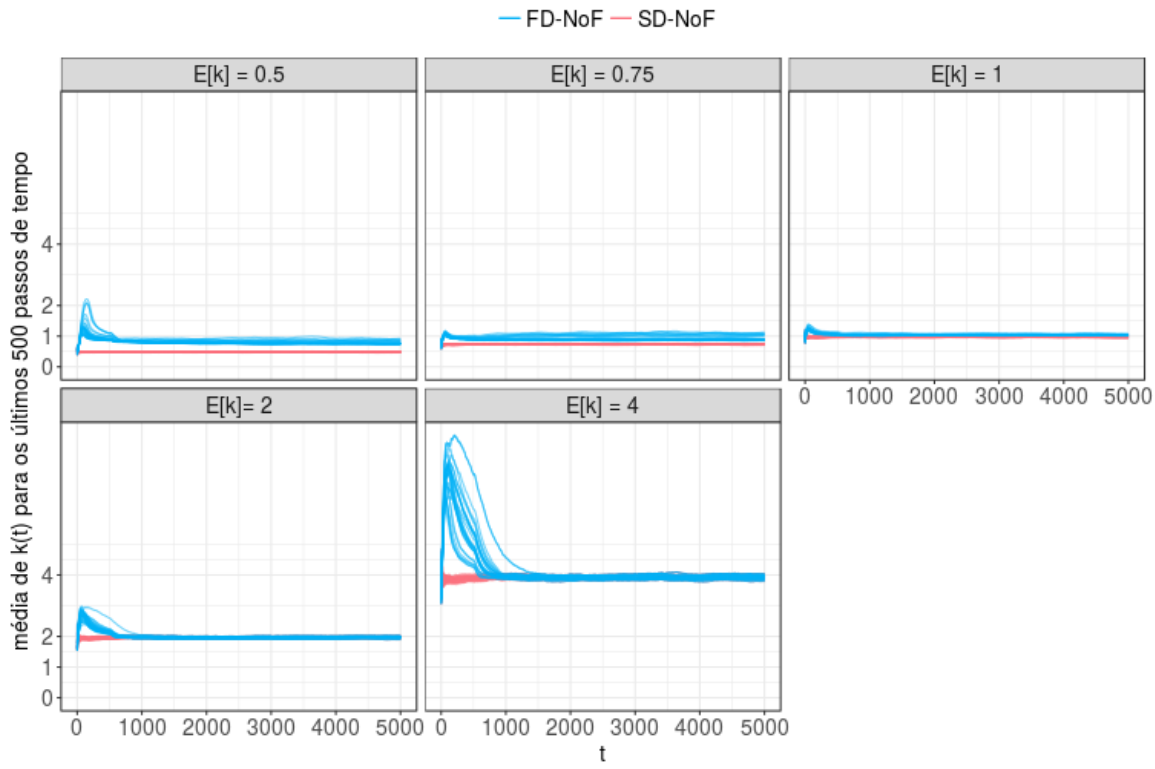


Figura 5.3: Média móvel de $\kappa(t)$ sobre os últimos 500 passos para cada simulação.

Devido ao modelo de simulação simplificado, quando $\mathbb{E}[\kappa]$ é configurado como 2 ou 4 os nós cooperativos não conseguem reduzir a contenção de recursos para um valor próximo ou ligeiramente superior a 1 — o cenário no qual existiriam recursos suficientes para todos os nós cooperativos. Isto ocorre porque os níveis esperados de contenção de recursos para cada cenário dependem dos valores de $\mathcal{P}$ e $\mathcal{D}_{fed}$, que não mudam ao longo da simulação, e também

do valor de $\mathcal{C}$, que é limitado superiormente pelo valor inicial $\mathcal{C} = 1$. Portanto, conclui-se que o controlador de capacidade tem bom desempenho já que a longo prazo estabiliza a contenção média de recursos nos valores 1.03, 1.96 e 3.93, para os cenários de $\mathbb{E}[\kappa]$ iguais a 1, 2 e 4, respectivamente. Nos cenários em que $\mathbb{E}[\kappa] = 0.75$, o controlador estabiliza a contenção de recursos em 0.93, valor suficientemente alto para garantir que todos os nós cooperativos obtenham os níveis desejados de paridade (≥ 0.75). Quando a contenção de recursos esperada é 0.5, o controlador de capacidade aumenta $\kappa(t = 5000)$ para 0.78, que não é o melhor resultado uma vez que ainda existirão recursos excedentes para serem consumidos pelos caronas. No entanto, $\kappa(5000) = 0.78$ é suficiente para assegurar a 98.75% dos nós cooperativos pelo menos 75% de paridade, como desejado quando $\mathcal{L}_{min}$ foi configurado como 0.75. Adicionalmente, os demais 1.25% dos nós cooperativos alcançaram valores de paridade de $\approx 74\%$, e acredita-se que o controlador de capacidade os permitiria aumentar seus níveis de paridade com mais alguns passos para um patamar superior a $\mathcal{L}_{min}$ — talvez uma margem de segurança de 0.02 ou 0.05 acima de $\mathcal{L}_{min}$ fosse suficiente para sempre garantir níveis de paridade acima desse limite.

5.3.2 Análise de Sensibilidade do Controlador

Nas simulações apresentadas na seção anterior, os valores $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$ e $\Delta = 0.05 \cdot \mathcal{C}$ foram escolhidos para obter uma primeira impressão do desempenho que o controlador proporciona aos nós. $\mathcal{L}_{min}$ e $\mathcal{L}_{max}$ foram configurados com 0.75 e 0.95 porque, intuitivamente, o intervalo $[0.75..0.95]$ compreende bons níveis de paridade. Contudo, resta saber até quais níveis de $\mathcal{L}_{min}$ e $\mathcal{L}_{max}$ o controlador pode ser configurado de modo que consiga assegurar os níveis de paridade desejados. Também é importante estudar a influência de Δ para compreender as consequências de utilizar valores muito altos ou muito baixos.

Primeiramente, serão abordados os cenários com nós homogêneos nos quais a federação entrou em colapso: i) todos os cenários com $\mathbb{E}[\kappa] = 0.25$; e ii) cenários com $\mathbb{E}[\kappa] = 0.5$ e $\mathcal{F} = 0.75$ em que os nós cooperativos foram configurados com $\mathcal{P} \leq 0.14$ ou $\mathcal{D}_{fed} \leq 0.59$. Estes cenários são considerados extremos pois possuem o nível esperado da contenção por recursos muito baixo ($\mathbb{E}[\kappa] = 0.25$) ou possuem o nível esperado da contenção por recursos baixo aliado a uma porcentagem muito alta de caronas ($\mathbb{E}[\kappa] = 0.5$ e $\mathcal{F} = 0.75$), o que facilita o consumo de recursos de nós cooperativos por nós não recíprocos. Com $\Delta = 0.05 \cdot \mathcal{C}$ todos

os nós cooperativos reduzem em poucos passos a oferta de recursos para 0, o que faz com que a federação entre em colapso.

Para evitar o colapso da federação seria preciso alterar $\mathcal{L}_{min}$ e/ou Δ. Neste sentido, os mesmos cenários foram simulados com valores ainda menores de Δ, $\Delta \in \{0.005 \cdot \mathcal{C}, 0.001 \cdot \mathcal{C}\}$, e mantendo fixo $\mathcal{L}_{min} = 0.75$. Como resultado foi observado que apenas um valor muito baixo, $\Delta = 0.001 \cdot \mathcal{C}$, foi suficiente para manter a federação em funcionamento. Contudo, quando $\mathbb{E}[\kappa] = 0.25$, os níveis de paridade dos nós cooperativos no passo de tempo 5000 variou entre o intervalo $[0.27, 0.52]$, enquanto que a satisfação variou entre o intervalo $[0.14, 0.26]$, para os dois níveis de carona ($\mathcal{F} \in \{0.25, 0.75\}$). Para cenários configurados com $\mathbb{E}[\kappa] = 0.5$ e $\mathcal{F} = 0.75$, os níveis de paridade variaram no intervalo $[0.55, 0.75]$ e os níveis de satisfação variaram no intervalo $[0.14, 0.23]$. Em suma, ambos os cenários apresentaram resultados insatisfatórios quando $\Delta = 0.001 \cdot \mathcal{C}$, apesar da federação se manter em funcionamento.

Para evitar que a federação entrasse em colapso foi preciso que os nós cooperativos reagissem mais vagarosamente, o que permitiu maior proveito por parte dos caronas e afetou consideravelmente a paridade dos nós cooperativos. Porém, em um cenário real, não é aconselhável que um nó de uma federação configure $\Delta = 0.001 \cdot \mathcal{C}$, pois deste modo, caso haja uma quantidade exagerada de caronas na federação, sua paridade seria consideravelmente afetada até que este nó perceba que a federação não lhe é recíproca (pelo menos na mesma proporção) e decida sair dela.

Para análise de sensibilidade mais profunda acerca dos impactos dos valores dos parâmetros do controlador, daqui em diante são focados os casos menos extremos ($\mathcal{F} = 0.25$) e mais interessantes ($\mathbb{E}[\kappa] = 0.5$[b]). Para este experimento, os parâmetros assumiram os seguintes valores: $\Delta \in \{0.01 \cdot \mathcal{C}, 0.05 \cdot \mathcal{C}, 0.1 \cdot \mathcal{C}\}$, $\mathcal{L}_{min} \in \{0.75, 0.85, 0.95\}$ e $\mathcal{L}_{max} \in \{0.95, 1\}$; o que resultou em 15 cenários, dado que não faria sentido simular $\mathcal{L}_{min} = \mathcal{L}_{max} = 0.95$. Os gráficos de caixa da Figura 5.4 apresentam os níveis de paridade e satisfação dos nós cooperativos no último passo da simulação.

A primeira conclusão que pode ser tirada a partir da Figura 5.4 é que há pouquíssima diferença entre os resultados dos cenários nos quais $\mathcal{L}_{max} = 0.95$ e $\mathcal{L}_{max} = 1$. Apesar

[b] Através dos resultados da Figura 5.1 e 5.2 foi observado que o desempenho da FD-NoF é similar ao desempenho da SD-NoF em cenários de contenção por recursos, *i.e.*, quando $\mathbb{E}[\kappa] \geq 1$.

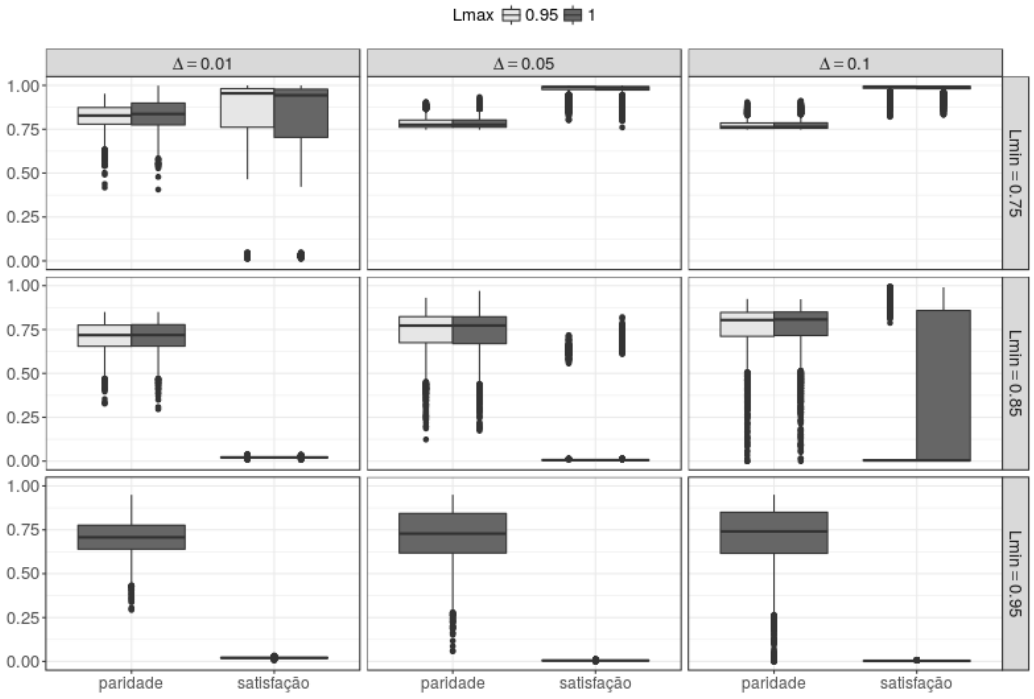


Figura 5.4: Níveis de paridade e satisfação dos nós cooperativos no último passo da simulação, em uma federação com $\mathcal{F} = 0.25$ e $\mathbb{E}[\kappa] = 0.5$.

dos gráficos de caixa levarem a crer que os resultados são consideravelmente diferentes para $\mathcal{L}_{min} = 0.85$ e $\mathcal{L}_{max} \in \{0.95, 1\}$, a análise detalhada dos dados não confirmou a interpretação visual do gráfico: quando $\mathcal{L}_{max} = 0.95$ e $\mathcal{L}_{max} = 1$, a quantidade de cenários que entraram em colapso foi, respectivamente, 23 e 22; considerando os demais cenários (com $\mathcal{L}_{min} = 0.85$ e $\Delta = 0.1 \cdot \mathcal{C}$) nos quais a federação se manteve funcionando, quando $\mathcal{L}_{max} = 0.95$ (7 cenários) e $\mathcal{L}_{max} = 1$ (8 cenários), o nível de satisfação dos nós cooperativos foi abrangido, respectivamente, pelos intervalos $[0.79, 0.99]$ e $[0.77, 0.99]$. Também é possível perceber que à medida em que $\mathcal{L}_{min}$ aumenta, mais árdua é a tarefa do controlador de assegurar os níveis mínimos desejados de paridade, o que leva os nós cooperativos a saírem da federação; $\mathcal{L}_{min} = 0.85$ é suficiente para inviabilizar a federação na maioria dos cenários e $\mathcal{L}_{min} = 0.95$ faz com que a federação colapse em todos os casos. Contudo, os nós cooperativos conseguem obter bons níveis de paridade considerando o nível de contenção demasiadamente baixo aliado à ação coletiva dos caronas. A partir dessa análise, foi decidido que o restante das simulações seriam executadas com $\mathcal{L}_{min} = 0.75$ e $\mathcal{L}_{max} = 0.95$, por se tratarem de valores conservadores, constituindo um intervalo que abrange bons níveis de

paridade.

Com relação à influência do Δ, os níveis de paridade e satisfação tendem a ser inferiores para valores menores de Δ, uma vez que a reação dos nós cooperativos contra os caronas será mais lenta; $\Delta = 0.01 \cdot \mathcal{C}$ faz com que a federação se extinga em alguns cenários mesmo para níveis razoáveis de $\mathcal{L}_{min}$, *e.g.*, 0.75.

Para entender a evolução inicial (*bootstrapping*) da federação com diferentes valores para Δ, foi observado o nível de compartilhamento de recursos nos 1000 passos iniciais de cada simulação. O *nível de compartilhamento de recursos* é definido como a razão entre a quantidade total de recursos doados e a quantidade total de recursos ofertados à federação no tempo t. Considerando que nos cenários simulados $\mathbb{E}[\kappa] = 0.5$ e que os nós caronas tem demanda infinita, o melhor nível de compartilhamento de recursos também seria de aproximadamente 0.5, o que satisfaria a maioria dos nós cooperativos. A Figura 5.5 apresenta a média móvel (com 10 amostras) do nível de compartilhamento de recursos da federação, para diferentes valores de Δ, e cada linha representa um cenário.

É possível observar na Figura 5.5 que em todos os cenários que não entram em colapso o nível de compartilhamento inicia em 1, sofre uma queda e em seguida se recupera. Esta queda representa o processo de identificação dos caronas pelos nós cooperativos, e portanto, quanto mais abrupta ela for, mais rápida será a segregação dos caronas. Deste modo, pode-se notar que nos turnos iniciais da simulação, o nível de compartilhamento dos cenários configurados com $\Delta = 0.01 \cdot \mathcal{C}$ reduz e se recupera lentamente em comparação com $\Delta = 0.05 \cdot \mathcal{C}$ e $\Delta = 0.1 \cdot \mathcal{C}$. Porém, valores muito altos para Δ podem inibir algumas relações proveitosas que poderiam existir se um dado nó provedor oferecesse mais tempo para que o nó consumidor pudesse retribuir os recursos recebidos. Como $\Delta = 0.05 \cdot \mathcal{C}$ é um valor intermediário que retalia os caronas rapidamente e provê tempo aos nós cooperativos retribuírem os recursos recebidos, decidiu-se que as demais simulações seriam executadas com $\Delta = 0.05 \cdot \mathcal{C}$.

5.3.3 Federação Heterogênea

As Figuras 5.6 e 5.7 mostram os níveis de paridade e satisfação dos nós cooperativos no último passo da simulação para $\mathcal{F} = 0$ e $\mathcal{F} = 0.25$. Os resultados são apresentados em diferentes cenários de $\mathbb{E}[\kappa]$, nos quais cada nó é representado por uma geometria. Os nós estão organizados em grupos de acordo com suas probabilidades de consumo ($\mathcal{P}$), onde cada

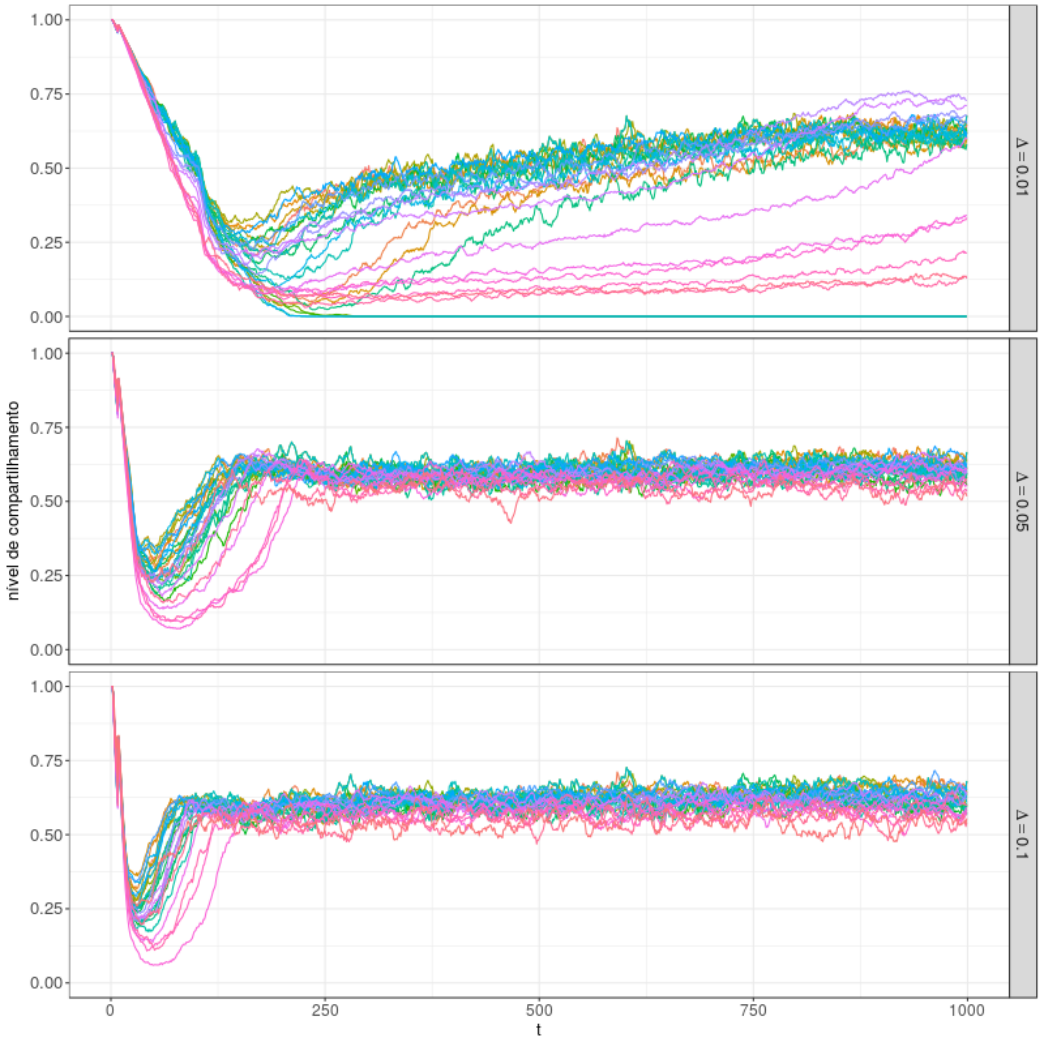
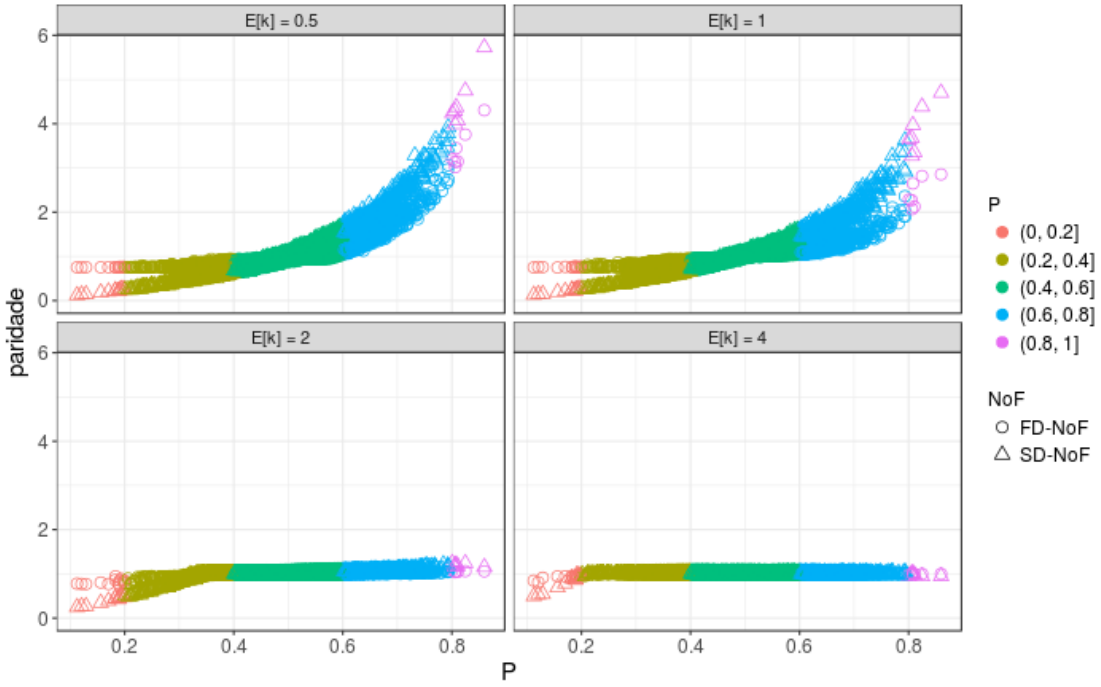


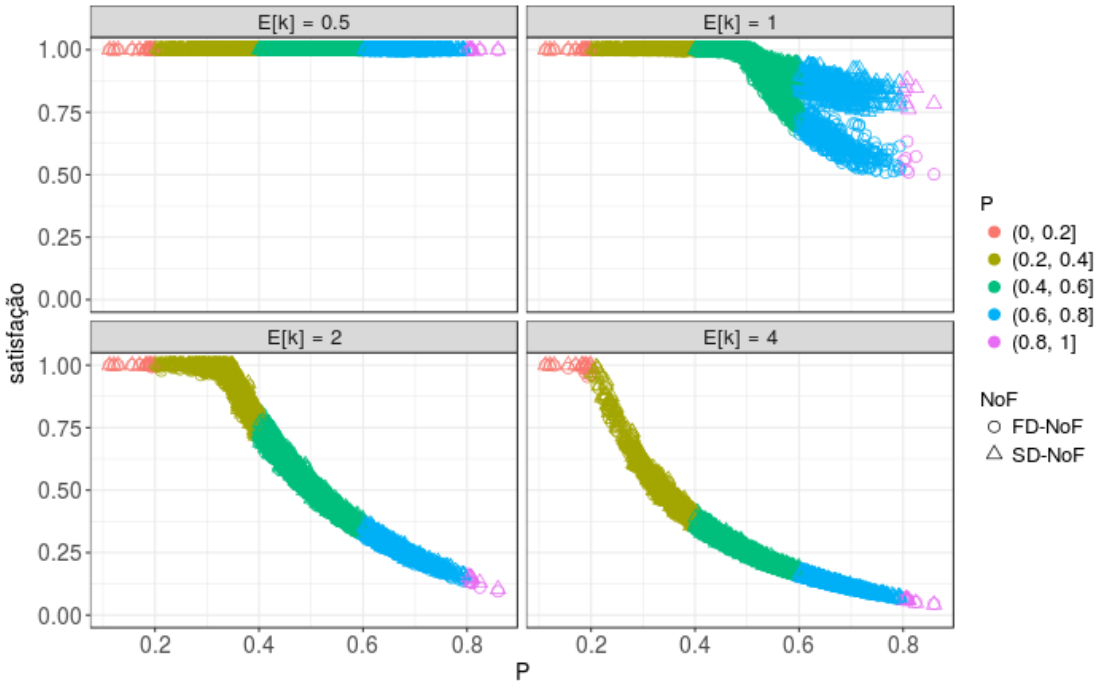
Figura 5.5: Média móvel do nível de compartilhamento de recursos utilizando as últimas 10 amostras, nos primeiros 1000 turnos da simulação. Controlador: $\Delta \in \{0.01 \cdot \mathcal{C}, 0.05 \cdot \mathcal{C}, 0.1 \cdot \mathcal{C}\}$, $\mathcal{L}_{min} = 0.75$ e $\mathcal{L}_{max} = 0.95$.

grupo é plotado com uma cor diferente. Foram executadas simulações para SD-NoF e FD-NoF, esta última com $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$ e $\Delta = 0.05 \cdot \mathcal{C}$. Lembre-se que nessas simulações os nós são diferentes, cada um tendo uma probabilidade diferente de estar em estado consumidor ($\mathcal{P}$), com valores retirados de uma distribuição Normal com média 0.5 e desvio padrão 0.1.

A maneira mais simples de analisar a Figura 5.6 é avaliando o desempenho de cada nó de acordo com sua *taxa de consumo*, definida como $\mathcal{P} \cdot \mathcal{D}_{fed}$. Para os cenários apresentados na Figura 5.6, a taxa de consumo depende apenas do valor de $\mathcal{P}$, uma vez que $\mathcal{D}_{fed}$ é uma constante configurada de acordo com o nível de $\mathbb{E}[\kappa]$ em cada cenário específico. No que

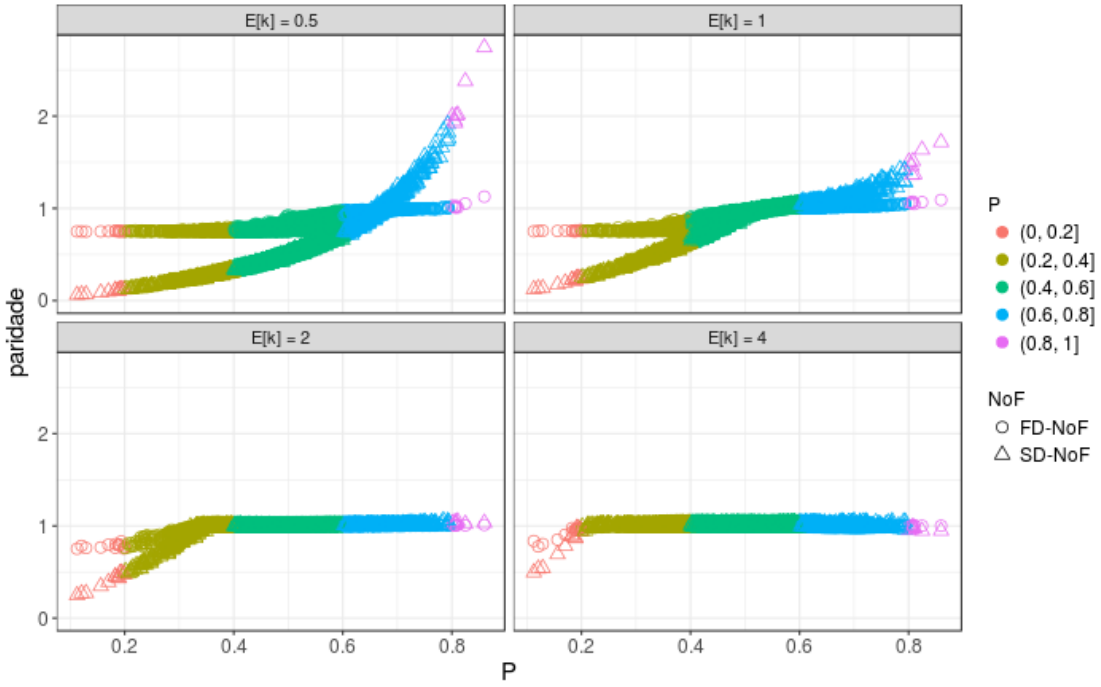


(a) Paridade.

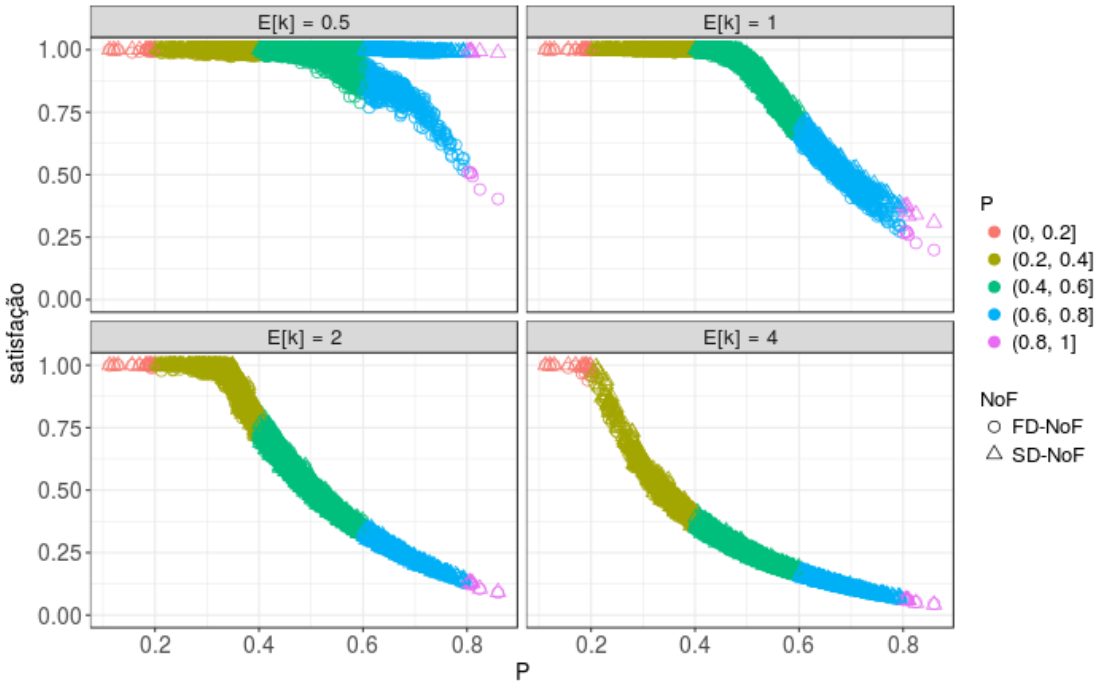


(b) Satisfação.

Figura 5.6: Paridade e satisfação dos nós cooperativos, na SD-NoF e FD-NoF, ambas com $\mathcal{F} = 0$. Os valores de $\mathcal{P}$ para cada nó foram atribuídos de acordo com uma distribuição normal com média igual a 0.5 e desvio padrão igual a 0.1.



(a) Paridade.



(b) Satisfação.

Figura 5.7: Paridade e satisfação dos nós cooperativos, na SD-NoF e FD-NoF, ambas com $\mathcal{F} = 0.25$. Os valores de $\mathcal{P}$ para cada nó foram atribuídos de acordo com uma distribuição normal com média igual a 0.5 e desvio padrão igual a 0.1.

concerne a SD-NoF, pode-se observar que existe uma forte correlação entre os níveis de paridade obtidos pelos nós e suas taxas de consumo, nestes casos, influenciadas pelo valor de $\mathcal{P}$. Em cenários de contenção baixa e moderada, os nós cooperativos obtêm altos níveis de paridade, porém, nos cenários nos quais o nível de contenção por recursos é alto ou muito alto ($\mathbb{E}[\kappa] \geq 2$), apenas o valor de $\mathcal{P}$ não é suficiente para garantir níveis de paridade acima da média. Ainda sobre cenários onde $\mathbb{E}[\kappa] \geq 2$, também é possível notar que a alta contenção por si só não é suficiente para garantir bons níveis de paridade, pelo menos para alguns nós com valores muito baixos de $\mathcal{P}$. Por fim, os gráficos também confirmam que a satisfação é inversamente proporcional a $\mathcal{P}$ e $\kappa(t)$, provendo aos nós com altos valores de $\mathcal{P}$ baixos níveis de satisfação.

Em relação à FD-NoF, os mesmos nós alcançam seus principais objetivos de ter níveis de paridade maiores que $\mathcal{L}_{min} = 0.75$, independente do grau de contenção por recursos. Essencialmente, isso torna a federação uma opção interessante até para os nós cujas taxas de consumo sejam baixas demais. Ainda sobre a Figura 5.6a, também pode-se observar que devido às limitações do modelo de simulação, os nós com altos valores de $\mathcal{P}$ podem não ser capazes de reduzir seus níveis de paridade para um valor menor do que $\mathcal{L}_{max} = 0.95$. Note que, uma vez que a taxa de consumo é fixa, para reduzir sua paridade um nó cooperativo deveria aumentar sua oferta. Contudo, os nós cooperativos não conseguem aumentar essa oferta para valores acima de $\mathcal{C} = 1$, dado que esta é a quantidade máxima de recursos que cada um possui. Porém, em cenários reais nos quais os nós podem controlar suas taxas de consumo através do ajuste dos valores de $\mathcal{P}$ ou $\mathcal{D}_{fed}$, o controlador de capacidades deve ser efetivo em assegurar a esses nós um nível de paridade entre $[\mathcal{L}_{min}, \mathcal{L}_{max}]$.

As Figuras 5.6 e 5.7 podem ser comparadas no intuito de analisar o impacto da presença dos caronas ávidos ($\mathcal{F} = 0.25$), *i.e.*, nós com $\mathcal{P} = 1$ e $\mathcal{D}_{fed} = \infty$. Nos cenários com alta contenção ($\mathbb{E}[\kappa] \geq 2$), os caronas não causam impacto algum na paridade e satisfação dos nós cooperativos. Este fato leva a duas conclusões: i) a SD-NoF é eficiente contra caronas em cenários de contenção; e ii) o controlador da FD-NoF é inteligente o suficiente para não aumentar ainda mais os níveis de contenção, o que reduziria os níveis de satisfação dos nós cooperativos.

Analisando a Figura 5.7 isoladamente, especificamente os cenários de baixa e moderada contenção por recursos, $\mathbb{E}[\kappa] \leq 1$, é possível observar que os mesmos nós cooperativos

com alto $\mathcal{P}$ têm níveis de paridade e satisfação inferiores quando usam a FD-NoF ao invés da SD-NoF. Esse resultado não deve ser atribuído ao controlador dos nós com alto $\mathcal{P}$, mas principalmente ao fato de que, na FD-NoF, os nós com baixa taxa de consumo passam a limitar a capacidade ofertada para obter melhores níveis de paridade, e consequentemente menos recursos são disponibilizados à federação, afetando diretamente os nós cooperativos com alta taxa de consumo. Acredita-se que na FD-NoF, os nós com baixa taxa de consumo são protegidos das ações dos caronas graças ao controlador de capacidade. Isto reforça que a FD-NoF atinge seu objetivo de garantir níveis de paridade maiores ou iguais a $\mathcal{L}_{min}$, mesmo na presença de caronas. No entanto, como já mostrado nos resultados dos nós homogêneos, uma alta proporção de caronas levaria a um colapso total da federação.

5.4 Considerações

Neste capítulo foi avaliado o mecanismo de laço de controle retroalimentado, o controlador de recursos, cujo objetivo é garantir níveis adequados de paridade e satisfação aos nós cooperativos de uma federação. A partir das simulações executadas é possível responder os 3 questionamentos levantados no final do Capítulo 2:

1. **Como controlar a contenção de recursos em um sistema P2P completamente descentralizado?**

2. **O controle da contenção de recursos é suficiente para assegurar aos nós cooperativos altos níveis de paridade?**

3. **Quão intensamente são afetadas as satisfações dos nós cooperativos e do sistema como um todo diante da redução da oferta de recursos ao sistema?**

Resposta para questão 1: quando cada nó controla de maneira individual a quantidade de recursos ofertada à federação, tem-se como resultado um controle indireto da contenção de recursos. Para $\mathbb{E}[\kappa] = 0.5$ obteve-se $\kappa(5000) = 0.78$[c], para $\mathbb{E}[\kappa] = 0.75 \rightarrow \kappa(5000) = 0.93$, para $\mathbb{E}[\kappa] = 1 \rightarrow \kappa(5000) = 1.03$[d], para $\mathbb{E}[\kappa] = 2 \rightarrow \kappa(5000) = 1.96$[e], e para

[c]Como $\mathcal{L}_{min} = 0.75$, é normal que $\kappa(5000)$ não tenha convergido para 1.

[d]Ver nota de rodapé c.

[e]O modelo simplificado de simulação impede que $\kappa(5000)$ convirja para 1.

$\mathbb{E}[\kappa] = 4 \rightarrow \kappa(5000) = 3.93$[f].

Resposta para questão 2: nos cenários de baixa contenção por recursos ($\mathbb{E}[\kappa] = 0.5$), 98.75% dos nós cooperativos conseguiram alcançar os níveis desejados de paridade, enquanto os demais 1.25% conseguiram aumentar seu níveis de paridade para valores ligeiramente inferiores ao desejado.

Resposta para questão 3: de modo geral, o aumento de paridade vem a um custo desprezível de satisfação. Esse custo pode variar dependendo da configuração escolhida para $\mathcal{L}_{min}$ e $\mathcal{L}_{max}$. Nos cenários em que $\mathbb{E}[\kappa] = 0.5$ e o controlador foi configurado com $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$ e $\Delta = 0.05$, a paridade dos nós cooperativos teve um aumento de 61.2%, ao custo de um sacrifício de apenas 2% de satisfação, em média. Em cenários de alta contenção, os resultados da FD-NoF foram similares aos resultados da SD-NoF, o que é bom, uma vez que a SD-NoF por si só já garante bons níveis de paridade e satisfação aos nós cooperativos.

Por fim, a FD-NoF também permitiu que nós cooperativos com baixa taxa de consumo alcançassem níveis de paridade acima do mínimo desejado. Tal fato aumenta a escalabilidade, uma vez que fornece bons níveis de paridade e satisfação até para os nós com baixa demanda ou frequência de consumo.

[f] Ver nota de rodapé e.

Capítulo 6

Validação

Resultados de simulações da FD-NoF em cenários com baixa contenção de recursos sugerem que os participantes que executam o laço de controle retroalimentado conseguem elevar seus níveis de paridade ao mesmo tempo que mantêm bons níveis de satisfação. Para validar o simulador, e consequentemente elevar a confiabilidade dos resultados obtidos, a FD-NoF foi implementada em um *middleware* para federação de provedores de computação na nuvem.

Neste capítulo são discutidos os desafios envolvidos com esta implementação e é apresentada uma avaliação de desempenho da execução da FD-NoF no *middleware*. Isto permitiu identificar as simplificações que distanciam o modelo de simulação do funcionamento da FD-NoF em um modelo mais realista.

6.1 Projeto e Implementação da FD-NoF

O *middleware* no qual a FD-NoF foi implementada é o *Fogbow* [14; 18], visto que o mesmo se baseia em uma arquitetura descentralizada, alinhando-se à abordagem da FD-NoF, e possui código aberto. Sua arquitetura e funcionamento são descritos a seguir.[a]

6.1.1 O *middleware* Fogbow

O Fogbow é basicamente uma camada de *software* implantada sobre os provedores de nuvem IaaS (do inglês *Infrastructure as a Service*) que permite a comunicação e interoperabilidade

[a]O código-fonte da FD-NoF no Fogbow está disponível publicamente no sítio `https://github.com/eduardolfalcao/fogbow-manager`.

entre os diferentes membros da federação. Suas principais funções e requisitos são:

1. gerência de membros: como os membros da federação irão se descobrir;
2. equiparação de recursos: trata de como descobrir os recursos equivalentes (*e.g.*, imagens, volumes e tipos de instância) em provedores diferentes;
3. autenticação e autorização: garantir interoperabilidade em diferentes níveis de autenticação e autorização (nível de nuvem local e de federação) e diferentes tecnologias de identificação;
4. intrusividade: permitir a inclusão de um provedor de nuvem IaaS sem violar políticas de segurança, usuários, topologias de rede, etc.;
5. acessibilidade externa: prover acessibilidade ao mundo externo para os recursos que possuírem conectividade limitada;
6. mercado: decidir qual membro deve ser priorizado na solicitação ou provisão de recursos.

Uma federação Fogbow é formada por pelo menos dois componentes: o agente fogbow (FM, do inglês *Fogbow Manager*) e o componente de descoberta (FR, do inglês *Fogbow Rendezvous*). O FM, implantado sobre os provedores de nuvem IaaS, funciona como um *proxy* que recebe requisições realizadas localmente e as redireciona para a nuvem subjacente local, ou encaminha para os demais membros da federação, quando a nuvem local estiver sobrecarregada. O FR implementa o serviço de descoberta entre os FMs – de tempos em tempos os FMs confirmam sua presença na federação e recebem a lista dos demais participantes. Uma federação pode possuir um ou vários FRs, como apresenta a Figura 6.1.

Das funções mencionadas[b], a gerência de membros é implementada no FR, a acessibilidade externa é assegurada pelo FM, e a equiparação de recursos, autenticação e autorização, e o mercado são responsabilidade de *plugins* comportamentais que são acionados pelo FM. A seguir são explicados os *plugins* comportamentais envolvidos na operacionalização de um mercado que usa a FD-NoF como mecanismo de escambo. Detalhes das demais funcionalidades podem ser encontrados em [14; 18].

[b]Intrusividade é um requisito não funcional.

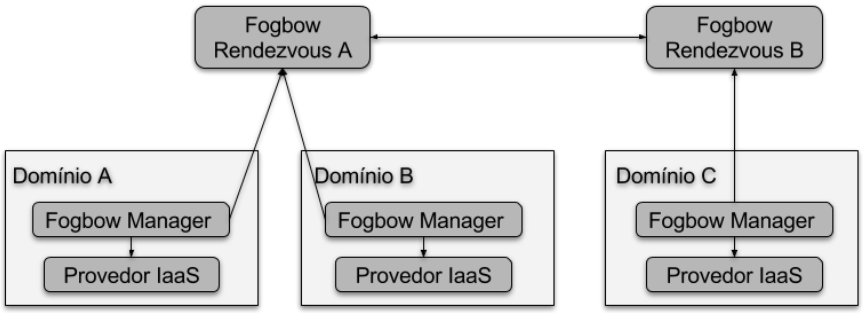


Figura 6.1: Exemplo de uma federação Fogbow.

6.1.2 A FD-NoF no Fogbow

A arquitetura do Fogbow baseia-se em *plugins* comportamentais que oferecem uma maneira simples de customização das principais operações realizadas pelo Fogbow, sem necessidade de alterar o código do componente principal, o FM. Na prática, esses *plugins* são interfaces (Java) com funções bem definidas que podem ser implementados pelos administradores de cada nuvem como lhes convirem.

Para explicar como a FD-NoF foi implementada no Fogbow é preciso primeiro descrever o fluxo que uma requisição de alocação de recursos percorre, desde o momento de sua submissão, até o momento em que ela é atendida. Nessa descrição são explicados os *plugins* envolvidos e como eles foram customizados para implementar a FD-NoF.

A Figura 6.2 ilustra os dois principais processos relacionados ao escambo de recursos: o controle de admissão de requisições e a gerência de *orders*. Sempre que uma requisição é admitida o FM cria uma *order* e a adiciona a um *pool de orders*. Os principais estados de uma *order* são *OPEN* – não processada, *PENDING* – enviada para outro membro da federação, *FULFILLED* – recurso criado e disponível, e *CLOSED* – encerrada. O processo de controle de admissão (Fig. 6.2) pode ser executado quando um usuário requisita algum recurso a sua nuvem, ocasião em que uma *order* local no estado *OPEN* é adicionada ao *pool de orders* do FM, ou quando uma nuvem participante requisita recursos a outro membro da federação, o que aciona o *plugin* de Controle de Capacidades.

O ***plugin* de Controle de Capacidades** atualiza periodicamente as quotas dos demais participantes e é acionado para decidir se existe quota disponível para o membro solicitante. Basicamente, o agente admitirá uma *order* remota se a quantidade de *orders* servidas, *i.e.*, *orders* no estado *FULFILLED* cujo provedor é o próprio agente, é maior ou igual à quota

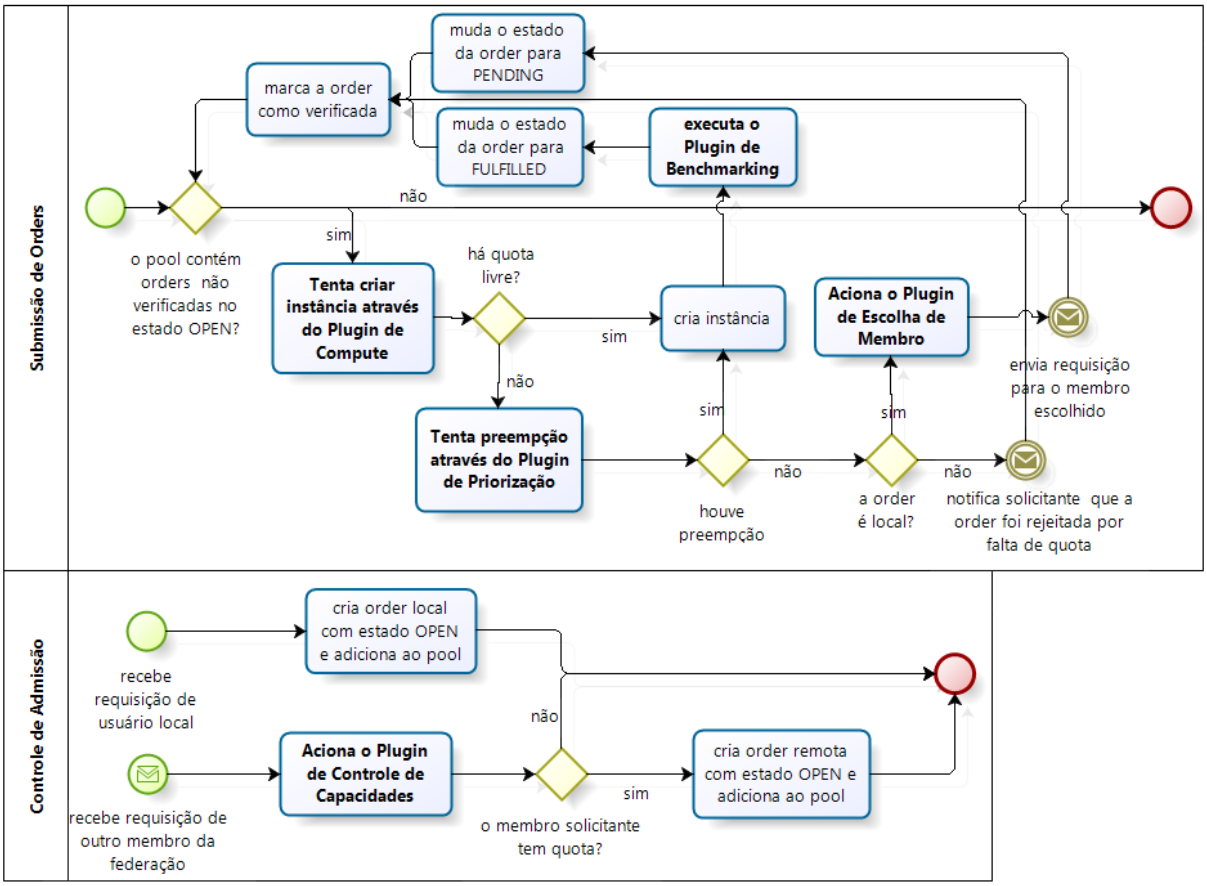


Figura 6.2: Processo de criação e alocação de *orders*.

calculada pelo *plugin*.

Como explicado na Seção 4.1, o Controlador de Capacidades de um agente utiliza os *níveis de paridade com a federação* para definir as quotas de nós dos quais apenas consumiu recursos ou com os quais nunca interagiu, e os *níveis de paridade bilateral* com cada participante para decidir as quotas dos participantes para os quais já realizou pelo menos uma doação.

O processo de submissão de *orders* (Fig. 6.2) também é executado periodicamente, independentemente da admissão de novas *orders*. Enquanto existir *orders* não verificadas no estado *OPEN*, o agente escolhe uma *order* segundo uma determinada sequência — *e.g.*, baseado no tempo de chegada — e tenta alocar recursos localmente através do *plugin* de *Compute* – responsável por toda comunicação com a nuvem. Caso haja quota disponível, a nuvem subjacente cria o recurso requisitado e envia sua localização ao *plugin* de *Benchmarking*, responsável por medir o desempenho do recurso, para fins de contabilidade. Após a

execução do *benchmarking*, o estado da *order* é configurado como *FULFILLED*, e a *order* é marcada como verificada, para que o agente não tente alocar uma mesma *order* mais de uma vez. Se a nuvem não possuir quota livre, o ***plugin* de Priorização** é acionado para verificar se há alguma *order* com menor prioridade a ser preemptada. Se alguma preempção acontecer, o recurso é criado, o *benchmarking* é executado, e o estado da *order* é atualizado para *FULFILLED*. Caso contrário, se a *order* não for local, o nó solicitante é notificado de que a *order* foi rejeitada por falta de quota, permitindo-o explorar o restante da federação. A *order* é marcada como verificada, de modo que seja considerada para alocação a posteriori. Caso não haja preempção, mas a *order* seja local, o FM aciona o ***plugin* de Escolha de Membro** e envia a requisição para o nó escolhido, o que aciona o controle de admissão em um outro membro da federação, e por fim, altera o estado da *order* para *PENDING*.

O *plugin* de Priorização pode ter implementação trivial, como nenhuma priorização (abordagem *First Come First Served*), ou pode se basear em um sistema de saldos como o da Rede de Favores, em que membros com maiores saldos recebem maior prioridade em suas solicitações. A estratégia do *plugin* de Escolha de Membro também pode ser simples como o *Round-Robin* ou basear-se no sistema de saldos da Rede de Favores, onde nós com maior débito teriam maior probabilidade de serem escolhidos, visto que esses nós estariam mais propensos a colaborar com participantes que lhe são recíprocos. Analogamente, o *plugin* de Controle de Capacidades pode ser implementado de forma trivial, segundo a estratégia *Satisfaction-Driven*, que não limita as quotas dos demais participantes, ou calcular as quotas de acordo com níveis de paridade (abordagem *Fairness-Driven*). Todas as configurações de *plugins*, tempos de processo de submissão de *orders* e de atualização das quotas, bem como os parâmetros do controlador ($\mathcal{L}_{min}, \mathcal{L}_{max}, \Delta$) são escolhidos de forma autônoma por cada participante da federação.

6.2 Avaliação de Desempenho

Com o objetivo de validar o simulador, alguns cenários foram experimentados com o mecanismo da FD-NoF em execução no Fogbow, e os resultados foram confrontados com o desempenho dos mesmos cenários no simulador. A seguir são detalhadas a infraestrutura e simplificações para os experimentos, o método de coleta dos resultados, as métricas de

comparação dos experimentos com as simulações equivalentes, e os cenários que foram executados.

6.2.1 Infraestrutura e Simplificações

Para viabilizar cenários com intensa carga de trabalho, alguns detalhes de infraestrutura foram simplificados, mas sem afetar significativamente a complexidade do modelo de mercado. Nesses experimentos toda a infraestrutura da federação é emulada. A nuvem subjacente, por exemplo, é emulada através de uma implementação básica para o *plugin* de *Compute*, o *CloudEmulatorComputePlugin*, uma vez que a complexidade e tempo envolvidos na criação e término de VMs não é trivial. Além disto, para facilitar a análise dos resultados, todas as VMs solicitadas possuem os mesmos requisitos computacionais. Por esta razão, o *plugin* de *Benchmarking* (*VanillaBenchmarkingPlugin*) retorna o mesmo valor referente ao poder de processamento das VMs. Um resumo com a descrição dos demais *plugins* além das implementações utilizadas é apresentado na Tabela 6.1.

Para facilitar a implantação, embora troquem mensagens normalmente via rede, todos os nós são implantados em uma mesma máquina (20GB RAM, 16 vcpus, 500GB HD, Ubuntu 16.04 Xenial), o que nos levou a usar banco de dados na memória principal, pois o processamento de toda a carga de trabalho da federação em apenas uma máquina causaria gargalos nas operações de entrada e saída. Com relação à descoberta, todos os FMs se encontram através de apenas um FR.

Nesses experimentos, cada *order* criada carrega consigo o tempo de duração total do recurso a ser alocado, de modo que seja possível regular a demanda e consequentemente os níveis de contenção desejados. Além disto, também foi necessário criar um campo adicional para monitorar o tempo de execução de cada *order*. Isto permite, por exemplo, reescalonar *orders* preemptadas cuja duração de execução ainda não tenha atingido a duração total especificada na criação da *order*.

O método de coleta dos resultados bem como as métricas que servirão de entrada para o simulador são explicados a seguir.

[c]Utiliza por composição o *GlobalFairnessDrivenController* e o *PairwiseFairnessDrivenController*.

Plugin	**Implementação usada**	**Descrição**
Priorização	*NoFPrioritizationPlugin*	Prioriza as *orders* segundo a política da NoF.
Controle de Capacidades	*SatisfactionDrivenCapacityControllerPlugin* ou *TwoFoldCapacityController*[c]	Calcula as quotas dos demais membros de acordo com a abordagem SD ou FD.
Compute	*CloudEmulatorComputePlugin*	Emula um provedor de nuvem IaaS.
Benchmarking	*VanillaBenchmarkingPlugin*	Define a potência das VMs baseado em suas especificações de *hardware*.
Contabilidade	*FCUAccountingPlugin*	Contabiliza o uso dos recursos.
Escolha de Membro	*RandomizedNoFMemberPickerPlugin*	Prioriza a solicitação de recursos aos membros com maior débito na NoF.
Identificação	*NoCloudIdentityPlugin*	Libera a autenticação de usuários na nuvem subjacente local.
Autorização	*AllowAllAuthorizationPlugin*	Permite todos os usuários da nuvem local criar requisições na federação.
Mapeamento	*SingleMapperPlugin*	Mapeia os participantes da federação para apenas um usuário na nuvem local.

Tabela 6.1: *Plugins* utilizados nos experimentos com o *middleware* Fogbow.

6.2.2 Método de Coleta dos Resultados e Métricas de Comparação dos Experimentos com Simulações Equivalentes

As métricas avaliadas no experimento são os níveis de paridade e satisfação de cada nó. Para isto, foi preciso monitorar em cada nó P_i a quantidade de *orders* no estado *OPEN*, *PENDING* e *FULFILLED*. Uma vez que os nós podem ter *orders* com origem local ou remota, as seguintes notações são usadas para permitir esta identificação:

- $O_{r=i}$: número de *orders* no estado *OPEN* em que o requerente é P_i;
- $P_{r=i}$: número de *orders* no estado *PENDING* em que o requerente é P_i;
- $F_{r=i \wedge p \neq i}$: número de *orders* no estado *FULFILLED* em que o requerente é P_i e o provedor não é P_i;

- $F_{r=i \wedge p=i}$: número de *orders* no estado *FULFILLED* em que o requerente é P_i e o provedor também é P_i; e

- $F_{r \neq i \wedge p=i}$: número de *orders* no estado *FULFILLED* em que o requerente não é P_i e o provedor é P_i.

Foi necessário registrar nos *logs* variáveis derivadas das variáveis acima. Isso acontece toda vez que uma dessas variáveis derivadas, explicadas abaixo, muda de valor. Quando isso ocorre, uma nova entrada x no *log* é registrada, com $T[x]$ igual ao tempo atual e os valores das variáveis derivadas calculados da seguinte forma:

- demanda total de P_i por unidade de tempo, entre x e $x+1$: $D_i^T[x] = O_{r=i} + P_{r=i} + F_{r=i \wedge p=i} + F_{r=i \wedge p \neq i}$;

- demanda de P_i à federação por unidade de tempo entre x e $x+1$: $D_i^F[x] = max(0, D_i^T[x] - \mathcal{C}_i)$, onde $\mathcal{C}_i$ é a capacidade máxima de P_i;

- total de recursos da federação recebidos por P_i por unidade de tempo entre x e $x+1$: $R_i^F[x] = F_{r=i \wedge p \neq i}$;

- total de recursos ofertados por P_i à federação por unidade de tempo entre x e $x+1$: $O_i^F[x] = \mathcal{C}_i - F_{r=i \wedge p=i}$; e

- total de recursos de fato providos por P_i à federação por unidade de tempo entre x e $x+1$: $S_i^F[x] = F_{r \neq i \wedge p=i}$;

- total de recursos locais recebidos por P_i por unidade de tempo entre x e $x+1$: $R_i^L[x] = F_{r=i \wedge p=i}$; e

- demanda não atendida (*unattended*) pela federação por unidade de tempo, entre x e $x+1$: $U_i^F[x] = D_i^T[x] - R_i^L[x] - R_i^F[x]$.

Portanto, a paridade e satisfação até o tempo imediatamente anterior a $T[x]$ são calculadas da seguinte forma:

$$\phi_i[x] = \begin{cases} \frac{\sum_{j=1}^{x-1}(T[j+1]-T[j]) \cdot R_i^F[j]}{\sum_{j=1}^{x-1}(T[j+1]-T[j]) \cdot S_i^F[j]}, & \sum_{j=1}^{x-1} S_i^F[j] > 0 \\ \text{indefinida}, & \text{caso contrário;} \end{cases} \tag{6.1}$$

$$\psi_i[x] = \begin{cases} 1 - \frac{\sum_{j=1}^{x-1}(T[j+1]-T[j])\cdot U_i^F[j]}{\sum_{j=1}^{x-1}(T[j+1]-T[j])\cdot (U_i^F[j]+R_i^F[j])}, & \sum_{j=1}^{x-1} U_i^F[j] + R_i^F[j] > 0 \\ \text{indefinida}, & \text{caso contrário.} \end{cases} \tag{6.2}$$

A equação 6.2 define o cálculo da satisfação utilizando uma abordagem inversa: a diferença entre 1 e a porcentagem de *orders* não atendidas. Se fosse calculada do modo mais simples, razão entre recursos recebidos ($\sum_{j=1}^{x-1}(T[j+1]-T[j]) \cdot R_i^F[x]$) por P_i e demandados ($\sum_{j=1}^{x-1}(T[j+1]-T[j]) \cdot D_i^F[x]$) por P_i, em alguns momentos a satisfação poderia assumir valores maiores que 1. Isto acontece, por exemplo, quando um nó P_i com $F_{r=i \wedge p \neq i} > 0$ tem uma *order* local no estado *FULFILLED* terminada, levando P_i a consumir mais recursos da federação do que necessitava ($R_i^F[x] > D_i^F[x]$). Portanto, calcular a satisfação a partir da equação 6.2 tem efeito equivalente ao cálculo realizado nas simulações — em ambos, a satisfação é computada pela porcentagem de requisições atendidas pela federação em relação à quantidade de recursos da federação necessários para atender a demanda total de um nó.

Para efeito de medir qual foi a probabilidade média de consumo ($\bar{\mathcal{P}}_i$) e a demanda média solicitada à federação ($\bar{\mathcal{D}}_i$) durante a execução do experimento, *i.e.*, as métricas de entrada do simulador, faz-se o seguinte cálculo:

$$\bar{\mathcal{P}}_i = \frac{1}{T_i[n]} \cdot \sum_{x=1}^{n-1} \delta^{\mathcal{D}}(x), \tag{6.3}$$

e

$$\bar{\mathcal{D}}_i = \frac{\sum_{x=1}^{n-1}(T_i[x+1]-T_i[x]) \cdot D_i^F[x]}{\sum_{x=1}^{n-1} \delta^{\mathcal{D}}(x)}, \tag{6.4}$$

onde

$$\delta^{\mathcal{D}}(x) = \begin{cases} T_i[x+1]-T_i[x], & \text{se } D_i^F[x] > 0 \\ 0, & \text{caso contrário.} \end{cases}$$

6.2.3 Cenários

O *nível de contenção* (κ, razão entre quantidade de recursos requisitados e ofertados por nós cooperativos), a estratégia utilizada pelo *plugin de Controle de Capacidades* (*Satisfaction-Driven* ou *Fairness-Driven*) e a presença ou ausência de *caronas* são os principais fatores a serem variados para gerar cenários de interesse.

O primeiro passo é entender quais fatores precisam ser variados, e como devem ser variados, para produzir os valores desejados de κ. Uma forma de controlar a contenção do sistema é gerando uma carga de trabalho na qual uma fração dos nós cooperativos, denotada por μ, estaria apenas doando recursos, e o restante $(1 - \mu)$ exclusivamente demandando recursos, em determinados intervalos de tempo do experimento. Chamemos esse intervalo de tempo que um nó deve permanecer em um dado estado (provedor ou consumidor) de λ. Portanto, considerando que os nós têm as mesmas capacidades e demandas, para produzir um determinado valor para κ basta gerar uma carga de trabalho que contenha a demanda total de recursos ($\mathcal{D}$) de cada nó em função de sua capacidade total de recursos ($\mathcal{C}$). Para gerar a contenção κ desejada basta configurar os nós consumidores com uma demanda total de recursos $\mathcal{D} = \mathcal{C} + (\kappa \cdot \mathcal{C})$, onde $\mathcal{C}$ recursos são satisfeitos localmente, e $\kappa \cdot \mathcal{C}$ representa a quantidade de recursos requisitados à federação. Note que a dinâmica do sistema pode levar um nó que em um dado intervalo de tempo (λ) deveria apenas ofertar recursos a também demandar recursos remanescentes do ciclo anterior por ter demorado a encontrar (ou não haver encontrado) nós provedores durante o intervalo de tempo λ em que se encontrava no estado consumidor. Dito isto, quanto menor for λ, maior será a probabilidade de acúmulo das demandas em ciclos posteriores, elevando a contenção do sistema. Por isso, λ é fixo em um valor intermediário de 600s, e por questões de simplificação, o valor da fração é fixo em $\mu = 0.5$.

Os dois primeiros cenários executados são configurados com contenção de recursos baixa ($\kappa < 1$, *e.g.* $\mathbb{E}[\kappa] = 0.5$) e moderada ($\mathbb{E}[\kappa] = 1$). Assim como no modelo de simulação simplificado, tais valores de contenção são apenas um limite mínimo, pois além da redução da oferta realizada pelo controlador de capacidades, outros fatores que só aparecem no modelo real, como preempções e o tempo que um nó consumidor leva para encontrar um nó provedor disposto a doar recursos, podem elevar a contenção ao longo do experimento. Nestes dois cenários, todos os participantes serão configurados com capacidade total de recursos $\mathcal{C} = 20$. Sempre que estiver em estado provedor, o nó doará todos os seus recursos, e portanto $\mathcal{D} = 0$. Quando um nó estiver em estado consumidor, para gerar um nível de contenção $\mathbb{E}[\kappa] = 0.5$, a demanda precisa ser computada por $\mathcal{D} = \mathcal{C} + \frac{\mathcal{C}}{2}$, e para produzir $\mathbb{E}[\kappa] = 1$ a demanda deve ser calculada através da equação $\mathcal{D} = 2 \cdot \mathcal{C}$. Portanto, os participantes em estado consumidor serão configurados com $\mathcal{D} = 30$ para $\mathbb{E}[\kappa] = 0.5$, e $\mathcal{D} = 40$ para $\mathbb{E}[\kappa] = 1$.

Para todos os cenários, a comunidade é composta por $\mathcal{N} \cdot (1 + \mathcal{F})$ nós, onde $\mathcal{N}$ nós são colaboradores e $\mathcal{F} \cdot \mathcal{N}$ são caronas. Primeiramente é avaliado o desempenho da federação sob contenção de recursos baixa e moderada, com 40 participantes cooperativos ($\mathcal{N} = 40$ e $\mathcal{F} = 0$), contrapondo os resultados da SD-NoF e FD-NoF ($\Delta = 0.05 \cdot \mathcal{C}$, $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$, com o controlador executando a cada 30 segundos). Em seguida, o mesmo cenário de baixa contenção é reavaliado na presença de caronas, visto que é neste tipo de situação que eles conseguem se aproveitar da federação. Para isto, são adicionados 10 caronas à federação ($\mathcal{N} = 40$ e $\mathcal{F} = 0.25$) configurados com $\mathcal{C} = 0$ e $\mathcal{D} = 40$, *i.e.*, os caronas tentam coletivamente consumir todos os recursos ofertados à federação.

6.2.4 Resultados e Discussão - Fogbow

O primeiro experimento é composto por 40 participantes, todos cooperativos, em cenários com $\mathbb{E}[\kappa] \in \{0.5, 1\}$, e tem duração de 24 horas. Cada cenário foi executado com 6 replicações. A Figura 6.3 apresenta os valores mínimo, primeiro quartil, mediana, média, terceiro quartil, 99.48^{o} percentil[d] e máximo para os níveis de paridade, e os valores mínimo, 0.42^{o} percentil[e], primeiro quartil, mediana, média, terceiro quartil e máximo de satisfação dos nós, nas 6 replicações, ao longo do experimento.

A partir da Figura 6.3 é possível observar que a FD-NoF teve desempenho semelhante à SD-NoF, o que é bom, visto que na ausência de caronas faz sentido ofertar todos os recursos ociosos. Em suma, ao monitorar a paridade do nó com os demais membros, o controlador percebe que eles são recíprocos e por isso mantém a oferta em um alto patamar. No entanto, pode-se notar que um nó teve desempenho atípico (vide níveis de satisfação em cenários com FD-NoF e $\mathbb{E}[\kappa] = 0.5$). Isto aconteceu pois, a longo prazo, este nó retribuiu aproximadamente apenas metade dos recursos recebidos da federação, o que lhe rendeu ao fim do experimento uma paridade de 1.56. Como neste cenário $\mathbb{E}[\kappa] = 0.5$, mais recursos são ofertados do que demandados e, com isto, um provedor pode não conseguir doar todos os seus recursos ociosos na mesma proporção que receber, aparentando ser um carona. Apesar de improvável, isto aconteceu com 1 dos 240 nós (soma dos participantes dos 6 experimentos), ou seja, apenas $\approx 0.42\%$ dos participantes.

[d]Este percentil foi plotado para evidenciar um valor discrepante (*outlier*, em inglês).

[e]Ver nota de rodapé d.

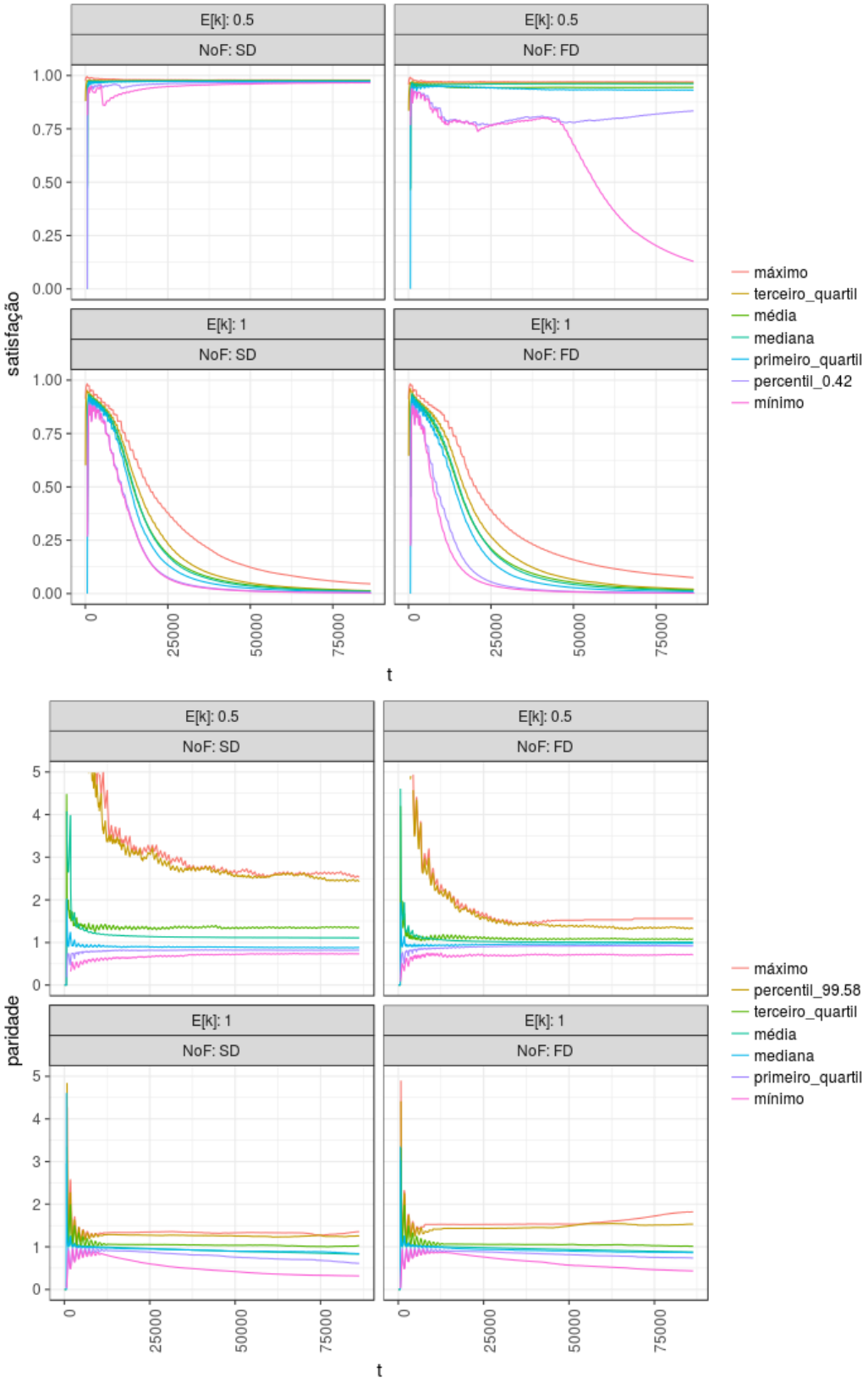


Figura 6.3: Níveis de paridade e satisfação dos participantes na SD-NoF e FD-NoF, nas 6 replicações do cenário sem caronas e com $\mathbb{E}[\kappa] \in \{0.5, 1\}$.

Contudo, ao contrário do cenário com SD-NoF e $\mathbb{E}[\kappa] = 0.5$, no qual a contenção real se manteve em $\kappa(t) \approx 0.5, \forall t \in [0, 86400]$, os níveis de contenção real para o cenário com SD-NoF e $\mathbb{E}[\kappa] = 1$ aumentaram consideravelmente — $\kappa(t = 28800) = 26.9$, $\kappa(t = 57600) = 65.1$ e $\kappa(t = 86400) = 88.9$ — o que atenuou os níveis de satisfação no último momento do experimento para um valor máximo de 4%. Isto aconteceu graças ao tempo elevado que os consumidores levaram para encontrar provedores com recursos disponíveis e à ocorrência de preempções, cuja frequência é maior em cenários de contenção elevada. Isto fez com que a demanda dos nós aumentasse ao longo do experimento – situação não percebida no modelo de simulação simplificado.

Com relação aos níveis de paridade, observa-se que a FD-NoF consegue, em cenários de baixa contenção por recursos, diminuir a dispersão entre os níveis de paridade dos participantes, tornando a federação mais justa. Em cenários com $\mathbb{E}[\kappa] = 0.5$, no último momento do experimento os níveis de paridade dos nós na SD-NoF e FD-NoF permaneceram, respectivamente, dentro dos intervalos $[0.73..2.53]$ e $[0.7..1.56]$. Quando $\mathbb{E}[\kappa] = 1$, 99.58% dos nós na SD-NoF obtiveram níveis de paridade entre 0.31 e 1.35, ao passo que na FD-NoF, os mesmos nós obtiveram níveis de paridade no intervalo $[0.43..1.53]$, que são patamares semelhantes, uma vez que para $\mathbb{E}[\kappa] = 1$ a SD-NoF por si só já é eficiente.

O segundo cenário é composto por 50 participantes, onde 10 são caronas. Os participantes cooperativos são configurados com as mesmas demandas e capacidades do cenário anterior, gerando uma contenção aproximada de $\mathbb{E}[\kappa] = 0.5$. Os caronas, por outro lado, são configurados com $\mathcal{C} = 0$ e $\mathcal{D} = 40$, *i.e.*, nunca oferecem mas sempre demandam todos os recursos ofertados à federação. O experimento tem duração de 24 horas com $\lambda = 600$ segundos e 6 replicações são executadas para cada cenário. A Figura 6.4 apresenta os níveis de satisfação de cada nó, no último momento do experimento, plotados em diagramas de caixa ordenados por replicação (*i.e.*, as caixas vermelha e azul mais à esquerda exibem os resultados de uma mesma replicação), em cenários nos quais todos os nós interagem usando apenas a SD-NoF ou FD-NoF ($\Delta = 0.05 \cdot \mathcal{C}$, $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$). A paridade dos nós cooperativos[f] é apresentada de modo resumido na Figura 6.5 através dos seguintes intervalos: mínimo, primeiro quartil, mediana, média, terceiro quartil, 90^{o} percentil e máximo.

A Figura 6.4 torna evidente que os nós equipados com a SD-NoF não foram capazes de

[f]Por definição, os caronas tem paridade indefinida, visto que nunca doam recursos.

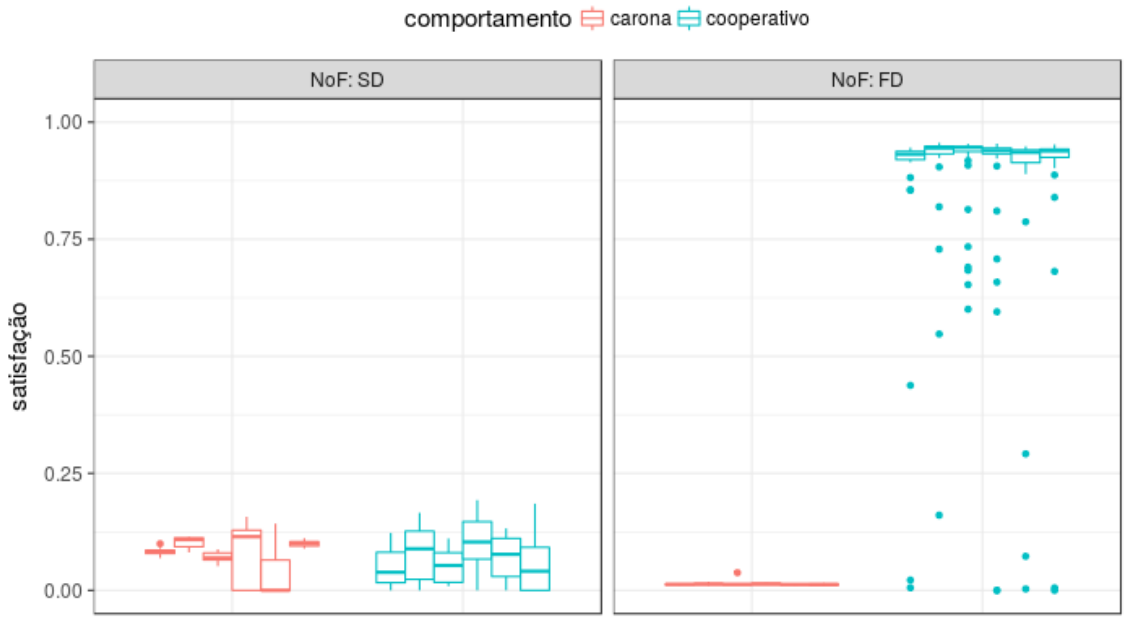


Figura 6.4: Níveis de satisfação dos nós cooperativos e caronas na SD-NoF e FD-NoF, nas 6 replicações do cenário com $f = 0.25$ e $\mathbb{E}[\kappa] = 0.5$.

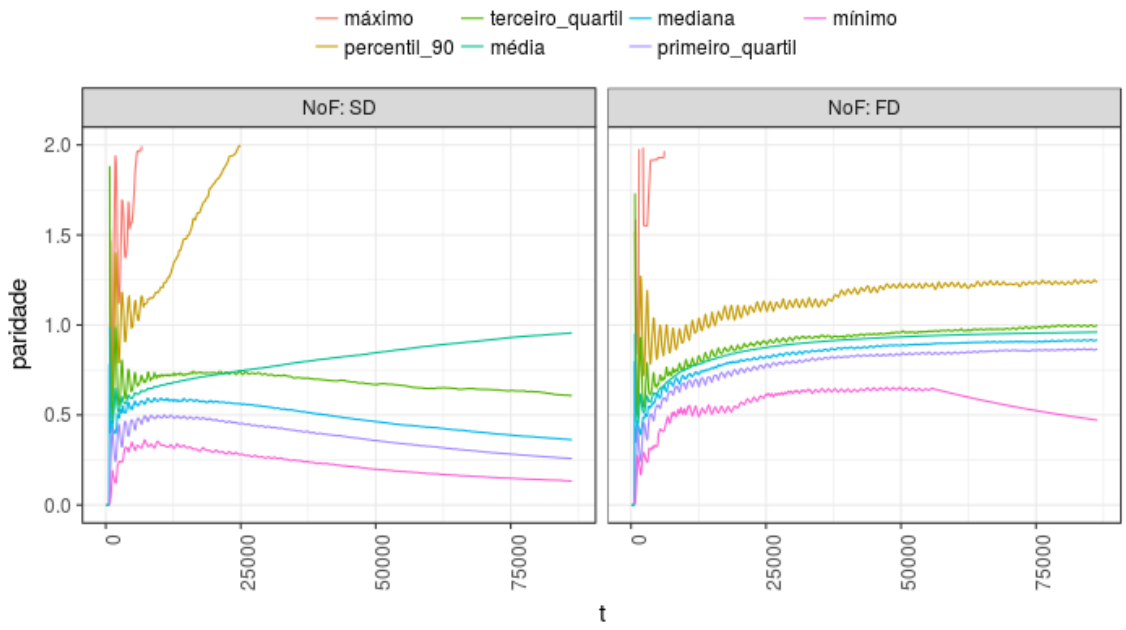


Figura 6.5: Níveis de satisfação dos nós cooperativos e caronas, no último momento dos experimentos, nas 6 replicações do cenário com $f = 0.25$, $\mathbb{E}[\kappa] = 0.5$ e NoF $\in \{SD, FD\}$.

identificar e evitar os caronas em cenários com baixa contenção de recursos. Nas 6 replicações desses cenários, o nível de satisfação dos nós cooperativos e caronas no último mo-

mento do experimento variou em patamares semelhantes, respectivamente entre $[0.009..0.19]$ e $[0.05..0.15]$. A FD-NoF, por outro lado, assegurou altos níveis ($[0.9..0.95]$) de satisfação a 85% dos nós cooperativos enquanto todos os caronas só tiveram de 1% à 3% de suas solicitações atendidas. No entanto, ainda na FD-NoF, alguns nós apresentaram desempenho insatisfatório (*e.g.*, $\approx 4.5\%$ dos nós cooperativos obtiveram níveis de satisfação inferiores a 50%) por terem alcançado altos níveis de paridade (superiores a 1.3) e, por isto, os demais nós o perceberam como carona e cessaram as doações para eles. Isto fez com que as demandas deles se acumulassem ao longo do experimento, tornando-se maior que sua capacidade, e, por isso, seus níveis de satisfação foram atenuados significativamente (vide Fig. 6.4). Adicionalmente, a FD-NoF também conseguiu diminuir a diferença de paridade em relação à SD-NoF, em que ao se considerar 90% dos nós, permaneceram respectivamente entre $[0.47..1.25]$ e $[0.13..4.02]$.

Dois fatores que têm impacto direto nos resultados, e consequentemente nestes experimentos, são a configuração do Controlador de Capacidades e o valor escolhido para λ. É possível que menores valores para $\mathcal{L}_{min}$ e Δ e um período mais longo para execução do Controlador de Capacidades (ao invés de 30 segundos) oferecesse mais oportunidades dos nós cooperativos que foram segregados juntamente com os caronas de retribuírem os recursos recebidos e se mostrarem recíprocos. No entanto, esta configuração também poderia favorecer os caronas, uma vez que a FD-NoF levaria mais tempo para identificá-los. O tempo (λ) que um participante permanece no estado provedor/consumidor também influenciou este resultado. Quanto menor for λ mais difícil se torna a retribuição dos recursos pelos nós cooperativos em tempo hábil[g], pois os consumidores não encontram potenciais provedores de modo simples e direto como acontece em um sistema de mercado centralizado; de maneira oposta, mecanismos de escambo, por se tratarem de uma abordagem descentralizada, contam com uma morosidade natural no processo de descoberta de participantes dispostos a compartilhar recursos.

A seguir são discutidos os resultados de cenários equivalentes executados com auxílio do modelo de simulação simplificado.

[g]A tempo de não serem considerados caronas.

6.2.5 Resultados e Discussão - Simulador

Com o objetivo de validar o simulador, foram monitorados o tempo total em que cada nó requisitou recursos à federação, *i.e.*, os momentos em que cada nó não conseguiu atender sua demanda com seus próprios recursos, e a demanda total dos nós ao longo do experimento, o que permitiu calcular respectivamente $\bar{\mathcal{P}}_i$ e $\bar{\mathcal{D}}_i$ — os valores de entrada do simulador. No simulador, $\bar{\mathcal{P}}_i$ é mapeado para $\mathcal{P}$, que representa a probabilidade de um nó estar em estado consumidor em um passo de tempo (vide Seção 5.1), e $\bar{\mathcal{D}}_i$ é mapeado para $\mathcal{D}_{fed}$, que representa a quantidade de recursos solicitados à federação por um nó em um passo de tempo. A Tabela 6.2 apresenta os valores de $\bar{\mathcal{P}}_i$ e $\bar{\mathcal{D}}_i$, computados a partir das Equações 6.3 e 6.4, para todas as replicações dos cenários sem caronas em que a contenção esperada do experimento no Fogbow ($\mathbb{E}_F[\kappa]$) assumiu os valores $\mathbb{E}_F[\kappa] \in \{0.5, 1\}$, e para todas as replicações dos cenários com caronas e $\mathbb{E}_F[\kappa] = 0.5$.

A primeira análise que pode ser feita a partir dos valores apresentados na Tabela 6.2 é sobre a contenção de recursos esperada no simulador, denotado por $\mathbb{E}_S[\kappa]$. Os cenários que mais chamam atenção são aqueles com $\mathbb{E}_F[\kappa] = 1$. Utilizando a Equação 5.1, quando $\mathbb{E}_F[\kappa] = 1$ e a federação não contém caronas, a média dos valores esperados de contenção no simulador para as 6 replicações são respectivamente $\overline{\mathbb{E}_S[\kappa]} = 77.61$ e $\overline{\mathbb{E}_S[\kappa]} = 55.6$ para a SD-NoF e FD-NoF [h] . Isto acontece pois os nós solicitantes não conseguem encontrar nós provedores dispostos a doar recursos de modo simples e direto, o que leva as *orders* (demanda) a se acumularem a cada λ segundos, elevando a contenção de recursos da federação.

Esta é a simplificação que mais distancia o simulador do mundo real. O processo de *bootstrapping* do simulador, por exemplo, pode levar um nó a conhecer todos os membros da comunidade em apenas um passo de tempo. Em um contexto mais realista esse processo poderia demorar alguns minutos ou horas. No entanto, quando $\mathbb{E}_F[\kappa]$ é baixa, as *orders* não alocadas em um ciclo λ pode(m) ser atendida(s) no(s) ciclo(s) subsequente(s), e isto não eleva o nível de contenção em longo prazo.

Quando $\mathbb{E}_F[\kappa] = 0.5$ e a federação não contém caronas, a média dos valores esperados de contenção no simulador para as 6 replicações são respectivamente $\overline{\mathbb{E}_S[\kappa]} = 0.503$ e $\overline{\mathbb{E}_S[\kappa]} = 0.509$ para a SD-NoF e FD-NoF, pois as *orders* que porventura não tenham sido

[h] Note que na FD-NoF esse valor é apenas um limite mínimo, visto que a quantidade de recursos ofertada pode ser reduzida pelo Controlador de Capacidades.

$\mathbb{E}_F[\kappa]$	rep	sem caronas				com caronas			
		SD-NoF		FD-NoF		SD-NoF		FD-NoF	
		$\bar{\mathcal{P}}_i$	$\bar{\mathcal{D}}_i$	$\bar{\mathcal{P}}_i$	$\bar{\mathcal{D}}_i$	$\bar{\mathcal{P}}_i$	$\bar{\mathcal{D}}_i$	$\bar{\mathcal{P}}_i$	$\bar{\mathcal{D}}_i$
0.5	1	0.502	9.997	0.502	9.997	0.765	28.13	0.522	15.917
	2	0.502	9.998	0.502	9.997	0.691	31.264	0.504	10.034
	3	0.502	9.999	0.502	9.997	0.796	46.008	0.503	10.007
	4	0.502	9.999	0.507	10.485	0.66	16.732	0.51	11.149
	5	0.502	9.999	0.502	9.997	0.749	42.563	0.523	20.442
	6	0.502	9.999	0.502	9.997	0.692	22.703	0.515	14.769
1	1	0.925	111.552	0.919	105.61	-	-	-	-
	2	0.927	129.249	0.918	110.084	-	-	-	-
	3	0.922	133.231	0.911	110.711	-	-	-	-
	4	0.921	113.548	0.902	95.034	-	-	-	-
	5	0.93	157.406	0.916	95.393	-	-	-	-
	6	0.92	113.849	0.917	108.058	-	-	-	-

Tabela 6.2: Valores de $\bar{\mathcal{P}}_i$ e $\bar{\mathcal{D}}_i$ calculados a partir dos experimentos com o Fogbow.

completamente atendidas em um ciclo conseguem terminar no ciclo subsequente. Na presença de caronas, a média dos valores esperados de contenção para as 6 replicações são respectivamente $\overline{\mathbb{E}_S[\kappa]} = 4.59$ e $\overline{\mathbb{E}_S[\kappa]} = 0.72$; o que significa que os caronas conseguem consumir recursos na SD-NoF, elevando a contenção do sistema, mas possui pouco impacto em federações usando a FD-NoF, mesmo em cenários de baixa contenção.

Para a execução dos cenários equivalentes no simulador foram utilizadas como entradas as métricas apresentadas na Tabela 6.2 e um total de 2880 turnos, resultado da divisão entre o tempo total de execução dos experimentos no Fogbow (864000 segundos) e o intervalo de execução do Controlador de Capacidades (30 segundos). Os caronas, quando presentes, consomem com $\mathcal{P} = 1$ e $\mathcal{D} = \mathcal{D}_{fed} = 40$, com $\mathcal{C} = 0$. A Figura 6.6 exibe os níveis de satisfação e paridade dos nós cooperativos, no último momento do experimento, plotados em

diagramas de caixa ordenados por replicação (*i.e.*, as caixas vermelha e azul mais à esquerda exibem os resultados de uma mesma replicação), em cenários nos quais os nós interagem segundo a SD-NoF ou FD-NoF em que $\mathbb{E}[\kappa] = 0.5$ (onde a FD-NoF poderia apresentar algum ganho), com ou sem caronas. Os cenários em que o nível de contenção inicial é 1 não são apresentados pois o nível real de contenção aumenta de tal maneira que os níveis de satisfação sempre convergem para zero.

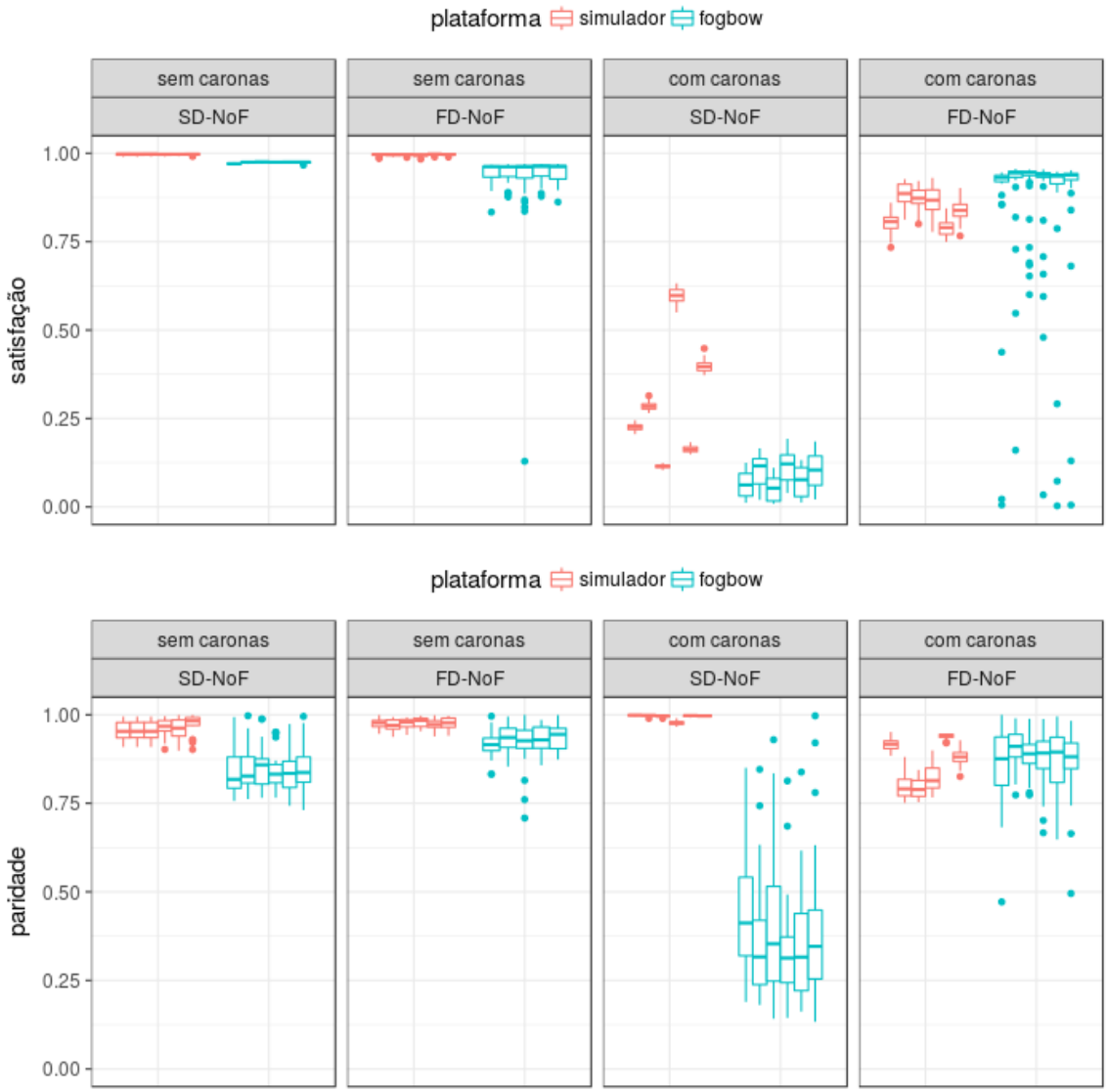


Figura 6.6: Níveis de paridade e satisfação dos participantes na SD-NoF e FD-NoF, nas 6 replicações do cenário sem caronas e com $\mathbb{E}[\kappa] \in \{0.5, 1\}$, executados no simulador e no Fogbow.

Os resultados apresentados na Figura 6.6 sugerem que, em cenários de baixa contenção,

o simulador pode servir como uma ferramenta de previsão dos resultados de um cenário mais realista, exceto para os cenários em que os nós cooperativos usam a SD-NoF e há caronas na comunidade. Isto acontece especificamente na SD-NoF pois o baixo nível de contenção (0.5) permite que os caronas consumam uma quantidade considerável de recursos dos nós cooperativos. A longo prazo essa interferência acarreta no acúmulo das demandas dos nós cooperativos, elevando o nível de contenção. Nos cenários sem caronas, por exemplo, os níveis médios de paridade e satisfação dos nós cooperativos são bastante semelhantes, embora o Fogbow apresente níveis ligeiramente menores de satisfação, o que é natural graças ao tempo que os consumidores levam para encontrar nós provedores dispostos a colaborar. No cenário com FD-NoF e caronas as plataformas do simulador e Fogbow apresentam uma dispersão um pouco maior, embora apontem a mesma tendência: bons níveis de satisfação e paridade para os nós cooperativos.

A principal simplificação que impactou na diferença dos resultados gerados pelo simulador e Fogbow é o reescalonamento de *orders*. No simulador, as requisições não atendidas em um turno são desconsideradas no turno subsequente, enquanto no Fogbow elas são consideradas para alocação a posteriori, até que sejam atendidas. Além de elevar a contenção de recursos, isto tende a afetar com mais veemência os níveis de satisfação e paridade no Fogbow. Para esclarecer este fato, imagine que um nó tenha 2 requisições para a federação mas apenas uma seja atendida. Neste caso, no simulador ele teria uma satisfação de 0.5, enquanto que em um experimento no Fogbow seu nível de satisfação seria tão menor do que 0.5 quanto maior fosse o intervalo de tempo em que a *order* não tivesse sido atendida.

Em nível de simulação, o primeiro objetivo era entender como se comportavam os níveis de satisfação e paridade dos nós cooperativos na presença de caronas e em diferentes níveis de contenção. Contudo, seria complexo entender o impacto de cada fator diante da interação dessas variáveis. Por isso, foi de suma importância simplificar o modelo de simulação para obter um entendimento inicial da dinâmica da comunidade, e uma destas simplificações foi a delimitação da vigência de uma requisição, para que fossem gerados cenários com uma contenção de recursos estável ao longo do tempo. Provavelmente os resultados provenientes do Fogbow e simulador seriam mais semelhantes, independentemente do nível de contenção, se no Fogbow as *orders* fossem descartadas a λ segundos do momento de suas criações, onde λ equivaleria um turno do simulador. No entanto, não é objetivo dessa avaliação compatibilizar

os resultados nas diferentes plataformas, mas sim entender quais simplificações distanciam o simulador do mundo real.

Da forma como foi concebido, é possível afirmar que o simulador provê boas estimativas para os cenários que iniciam e se mantêm com baixa contenção de recursos, *i.e.*, os cenários de real interesse desta tese, com exceção dos cenários que contêm caronas e são regidos pela SD-NoF, pois a interferência causada pelo consumo de recursos pelos caronas resulta em um aumento progressivo da contenção de recursos ao longo do tempo. Além disso, nos cenários com caronas, regulados pela FD-NoF e nos quais $\mathbb{E}[\kappa] = 0.5$, os níveis de satisfação dos nós cooperativos resultantes da simulação são sempre piores ou iguais aos níveis resultantes do experimento no Fogbow, indicando que, para estes cenários, os resultados do simulador podem ser interpretados como uma espécie de limite inferior do que pode ser atingido em um sistema real.

Outra situação não abordada no simulador é que, eventualmente, alguns participantes consumiram recursos da federação mesmo quando possuíam recursos disponíveis localmente. Isto aconteceu pois nos experimentos os participantes não migram a utilização de recursos remotos para locais, e eventualmente, recursos locais são liberados e permanecem ociosos ao mesmo tempo que o mesmo participante continua consumindo recursos da federação. A última simplificação percebida trata da fragmentabilidade do recurso, considerada no simulador, mas que geralmente não existe em um contexto realista quando os recursos compartilhados são VMs. Contudo, o único impacto dessa simplificação é que no mundo real os níveis de paridade e satisfação dos nós tenderiam a ser mais dispersos.

6.3 Considerações

Este Capítulo descreveu detalhes de implementação da FD-NoF no *middleware* Fogbow e apresentou seu desempenho, comparando com o desempenho da SD-NoF, através de experimentos conduzidos em um ambiente controlado. Adicionalmente, algumas métricas foram monitoradas para viabilizar a execução de simulações equivalentes aos experimentos, com o objetivo de confrontar os resultados e validar as discussões apresentadas no Capítulo 5 com o auxílio do simulador.

Em suma, os experimentos no Fogbow ratificam as principais conclusões apresentadas

no Capítulo 5: a FD-NoF obtêm resultados semelhantes aos da SD-NoF em cenários de alta contenção (bons níveis de paridade), e em cenários de baixa contenção a FD-NoF consegue identificar e isolar os caronas, assegurando reciprocidade entre os participantes. Os níveis de paridade e satisfação dos nós cooperativos em cenários equivalentes no Fogbow e simulador corroboram com estas conclusões.

Os experimentos também permitiram quantificar os impactos de algumas simplificações do simulador. As principais simplificações são: i) a simplicidade com que o simulador combina os consumidores e provedores, sem considerar o tempo natural de descoberta como um fator importante, e ii) o fato de que *orders* não atendidas em um turno são desconsideradas em turnos subsequentes. Estas duas simplificações tendem a elevar a contenção do sistema em federações SD-NoF que contenham caronas, ou quando o nível inicial de contenção já for moderado, independentemente de haver caronas (como $\mathbb{E}[\kappa] = 1$, apresentado na Fig. 6.3). Outras simplificações com impacto desprezível são a possibilidade de um nó consumir recursos da federação mesmo possuindo recursos locais ociosos e a fragmentabilidade dos recursos.

No próximo Capítulo é apresentado um estudo paralelo sobre um tipo de reciprocidade pouco abordado e utilizado, a **reciprocidade transitiva** – uma estratégia que permite mecanismos de reciprocidade direta lidarem com a assimetria de interesses ou disponibilidades entre indivíduos, ao mesmo tempo em que mantém os benefícios da reciprocidade direta.

Capítulo 7

Reciprocidade Transitiva

Em sistemas P2P, existem algumas situações nas quais a reciprocidade direta apresenta maus resultados, e portanto alguma forma de reciprocidade indireta deve ser considerada. De modo geral, a reciprocidade indireta é mais útil para sistemas P2P de larga escala, onde as chances dos nós se encontrarem repetidamente são pequenas, e por isso, os mecanismos de reputação poderiam prover melhores resultados. Contudo, também existem circunstâncias (*e.g.*, envolvendo cadeias de crédito [35]) nas quais alguma forma de reciprocidade indireta pode ser útil também para os casos nos quais a maioria dos nós interagem entre si, o que a priori eliminaria a necessidade de informações de terceiros. No intuito de solucionar esse tipo de problema, uma forma mais limitada de reciprocidade indireta que utiliza informações de terceiros de forma mais restrita seria suficiente.

Este capítulo apresenta um tipo limitado de reciprocidade indireta, a *reciprocidade transitiva*, que pode ser utilizada conjuntamente com mecanismos de reciprocidade direta, mas sem pôr em risco a integridade das informações sobre os comportamentos dos demais nós. Neste trabalho, a reciprocidade transitiva foi instanciada e avaliada conjuntamente com a Rede de Favores, e portanto, nesse contexto é chamada de *transitividade de favores* ou, pelo fato da Rede de Favores ser baseada em um sistema de créditos, *transitividade de créditos*.

A próxima seção apresenta o problema da assimetria de interesses/disponibilidades, seguida por uma definição da reciprocidade transitiva, e por fim, são apresentados os resultados de simulações que mostram sua importância em determinados cenários.

7.1 Contextualização do Problema

Suponha que os nós de uma federação sejam divididos em três regiões ou grupos de nós similares, R_1, R_2 e R_3. O motivo poderia ser assimetria de disponibilidades (*e.g.*, diferentes fusos horários) ou de interesses (*e.g.*, oferta e consumo de diferentes serviços). Assuma que em um dado passo de tempo t um nó pode estar em um dos seguintes estados: *consumidor*, *ocioso*, ou *provedor*; graças ao nível de demanda que está experimentando em um dado momento t. Um nó A *sobrecarregado* de demanda estará em estado consumidor, e portanto requisitará recursos de outros nós; quando estiver ocioso, um nó A estará apenas *carregado*, e por isso não irá requisitar nem prover recursos; por fim, quando estiver em um estado provedor, um nó A estará *descarregado*, e nesses casos irá ofertar os recursos excedentes à federação. Suponha agora que em um dado passo de tempo t, os nós do grupo R_1 serão consumidores, os nós de R_2 estarão ociosos, e os nós do grupo R_3 serão provedores. No próximo passo de tempo $t + 1$, a ordem desses papéis é transposta de maneira cíclica, e os nós do grupo R_1 serão provedores, os nós em R_2 serão consumidores, e os nós do grupo R_3 estarão ociosos. No tempo $t + 2$, acontece uma nova transposição cíclica desses papéis, e os nós de R_1 ficarão ociosos, os nós em R_2 serão provedores, e os nós em R_3 serão consumidores. Suponha também que esse padrão recorre indefinidamente a cada três períodos de tempo.

Nesse caso, tem-se uma situação na qual, se $A \in R_1$, $B \in R_2$, $C \in R_3$, C poderá prover recursos para A no tempo t_n, A poderá prover recursos para B no tempo t_{n+1}, e B poderá prover recursos para C no tempo t_{n+2}, para todo $n \in \mathbb{N}$. Contudo, A, B e C nunca serão capazes de retribuir esses favores a C, A e B, respectivamente. Isto significa que os débitos de cada nó irão aumentar indefinidamente com o passar do tempo, ao mesmo tempo em que seus níveis de paridade permanecerão iguais a 0. Contudo, com o passar do tempo, os níveis de satisfação e de paridade relacionados à federação, para cada nó, estarão próximos de 1, que significa níveis quase perfeitos de satisfação e paridade; apesar disso, o controlador da FD-NoF levará a federação a entrar em colapso.

Note que, nesse caso, a SD-NoF obteria êxito, uma vez que os nós sempre ofertarão toda sua capacidade ociosa, desconsiderando seus níveis de paridade. Não obstante, problemas surgirão com a presença de caronas, e nesses casos a federação também irá colapsar devido

à proliferação de caronas. Esta situação é explicada a seguir.

Após algumas interações, A suspeita que B é um carona, B suspeita que C é um carona, e C suspeita que A é um carona, apesar de cada um deles estarem provendo recursos a outros nós. Portanto, se um nó carona D entrar em cena, D será capaz de consumir pelo menos a mesma quantidade de recursos que B recebe de A (uma vez que $\gamma(A, B, t) = \gamma(A, D, t) = 0, \forall t \in \mathbb{N}$), a mesma quantidade de recursos que C recebe de B, e a mesma quantidade de recursos que A recebe de B. A quantidade total de recursos consumidos pelos nós cooperativos será reduzida pela metade, e D irá obter a mesma quantidade que A, B e C juntos. Mais caronas aparecerão e os nós cooperativos obterão apenas uma fração dos recursos disponíveis, inversamente proporcional ao número de caronas. Os níveis de satisfação e paridade em relação à federação, percebidos pelos nós cooperativos, irão se aproximar de zero, o que levará a uma deserção de todos os nós cooperativos e consequentemente a um colapso da federação.

Em um cenário mais realista no qual apenas a maioria dos nós em cada grupo acompanharem o estado do grupo em cada período de tempo t, com uma proporção de nós desviando da norma, a reciprocidade indireta ainda será útil. Esses são, de fato, os casos mais interessantes a serem investigados. De modo geral, pode-se dizer que a reciprocidade indireta é útil sempre que a probabilidade de algum nó A ser provedor para um nó B seja diferente da probabilidade de que B seja provedor para A, uma vez que nesse caso um longo desequilíbrio entre a quantidade de recursos providos entre A e B pode facilmente aparecer e piorar com o tempo.

A presente análise mostra que, em alguns cenários, não é possível que um nó tome boas decisões de doação baseado somente em informações locais, *i.e.*, com uma visão mais limitada do sistema. Se os três nós tivessem uma visão do sistema como um todo ou pelo menos informações adicionais, eles simplesmente continuariam cooperando harmonicamente, mesmo na presença de caronas.

7.1.1 Cadeias de Crédito e Risco Sistêmico

O cenário descrito na seção anterior tem muitas semelhanças com o procedimento de pagamentos interbancários (*interbank payments*, em inglês), que já foi bem explorado na área da teoria econômica financeira. Kiyotaki e Moore (1997), por exemplo, proveem um modelo

formal de redes de crédito além de um estudo teórico sobre o *risco sistêmico* associado à rede na qual as empresas emprestam diferentes tipos de recurso umas às outras. Na situação descrita por Kiyotaki e Moore (1997), os indivíduos envolvidos em uma cadeia de créditos obteriam melhores resultados ao simplesmente perdoarem os débitos uns dos outros. Apesar de existirem diferenças óbvias entre redes de créditos e redes de favores, também existem muitas similaridades, e a noção de risco sistêmico no contexto financeiro, *i.e.*, o risco ou probabilidades de quebra de todo um sistema [37], também pode ser útil no contexto de sistemas e federações P2P.

Uma cadeia de crédito na qual cada agente é ligado a apenas um vizinho ao longo da cadeia, é similar a uma cadeia de favores descrita acima entre os indivíduos A, B e C. Em uma cadeia de crédito, sabe-se que a probabilidade de colapso em caso dela ser atingida por um choque[a] é alta [4; 38]. Contudo, se o número de parceiros aumenta e a rede evolui para completude, o risco de colapso cai assintoticamente para zero [15]. Allen e Gale (1998) também dividiram o sistema bancário topologicamente em regiões para ilustrar esse fato, assim como descrito na Seção 7.1, assumindo que os bancos de cada região são idênticos e adotam o mesmo comportamento. Um estudo mais recente sobre a interação entre o risco sistêmico e as mudanças topológicas é provido por Squartini *et al.* (2013).

O risco sistêmico da NoF pode ser estabelecido em termos da quantidade de débitos no sistema, uma vez que é esperado que um nó A estará mais disposto a prover recursos a um nó B se a quantidade de débitos de B para A for baixa. Em outras palavras, quanto mais créditos B possuir na visão de A, mais A estará mais propenso a prover recursos para B. Portanto, se no tempo t a conjuntura for a seguinte, $\gamma(B, A, t) = \gamma(C, B, t) = \gamma(A, C, t) = R_t$, significando que até o passo de tempo t, B deve para A a mesma quantidade de recursos que C deve para B, e também a mesma quantidade que A deve para C, mais precisamente R_t unidades de recurso, então é axiomático que A estaria mais inclinado a prover mais recursos para B se o débito de B na visão de A fosse igual a zero, *i.e.*, se $\gamma(B, A, t) = 0$, e o mesmo se aplica para B e C, e C e A. Nesse caso, o risco sistêmico seria menor e todos os nós estariam em uma situação melhor se pudessem quitar seus débitos uns com os outros.

[a]O choque financeiro referido acontece quando uma empresa não consegue efetuar seus pagamentos, e por isso, as empresas receptoras desses pagamentos também não conseguem realizar seus pagamentos, o que leva a um endividamento em cascata.

Até mesmo se a cadeia de créditos não formar um ciclo, seria melhor se de algum modo os nós pudessem quitar seus débitos. Esta é, de fato, uma prática padrão em sistemas de pagamento interbancário, chamada compensação multilateral (*multilateral netting*, em inglês), permitindo maior fluxo de caixa entre mais de duas contrapartidas.

7.2 Reciprocidade Transitiva

A reciprocidade transitiva é um mecanismo pouco abordado na área da teoria da cooperação e os trabalhos que a utilizam apresentam definições e protocolos razoavelmente divergentes. Esta seção tem como objetivo unificar a definição de reciprocidade transitiva, dividindo-na em duas categorias: *reciprocidade transitiva por permuta imediata* e *reciprocidade transitiva por permuta tardia*. Neste sentido, na literatura foram encontrados cinco trabalhos que utilizam a transitividade: os trabalhos de Zhang e Antonopoulos (2009) e Eidenbenz *et al.* (2012) na modalidade de permuta imediata e os trabalhos de Bocek *et al.* (2009), Landa *et al.* (2009) e Nandi *et al.* (2005) na modalidade de permuta tardia. Nestes trabalhos, todos os protocolos para a realização da transitividade são diferentes, o que não é um problema, mas todos eles têm como objetivo final a concretização de uma provisão de recursos via transitividade/transferência de créditos.

Neste capítulo, a transitividade será explicada reutilizando a noção da Rede de Favores, um mecanismo de reciprocidade direta por permuta tardia, utilizado no contexto de sistemas P2P para compartilhamento de recursos computacionais, como as federações de provedores de computação na nuvem.

A reciprocidade transitiva pode ser descrita como uma expansão da reciprocidade indireta por *pay-it-forward*. No *pay-it-forward* puro, quando um nó A presta um favor para outro nó cooperativo B, B se sente bem e passa esse favor adiante para outro nó escolhido aleatoriamente. Quando realizado por permuta tardia, esse tipo de reciprocidade requer a prestação de pelo menos dois favores em momentos distintos, o favor inicial de A para B, *i.e.*, $\upsilon(A, B, t), t \in \mathbb{N}$, e o favor de B para um nó qualquer C, *i.e.*, $\upsilon(B, C, t + n), n > 0, n \in \mathbb{N} \wedge t \in \mathbb{N}$, sendo este último a retribuição indireta do primeiro favor, de forma altruísta para algum nó desconhecido.

A reciprocidade transitiva por permuta tardia, por outro lado, acontecerá com a prestação

de pelo menos três favores em diferentes instantes de tempo. Os dois favores iniciais acontecem via reciprocidade direta, através de interações convencionais. A princípio, suponha que A prestará um favor para B ($\upsilon(A, B, t), t \in \mathbb{N}$) e, em um momento posterior, B prestará um favor para C ($\upsilon(B, C, t + n'), n' > 0, n' \in \mathbb{N} \wedge t \in \mathbb{N}$); duas interações completamente independentes, sem relação uma com a outra. Na reciprocidade transitiva, a terceira e última interação acontecerá via uma versão mais restrita do *pay-it-forward*. Para consolidar a reciprocidade transitiva, o nó B informará que C deverá pagar o favor que recebeu de B para o nó A ($\upsilon(C, A, t + n''), n'' > n', n'' \in \mathbb{N} \wedge t \in \mathbb{N}$). C deve prestar um favor para A, *em nome de* B, via *pay-it-forward*, porque A já ajudou B e B ajudou C anteriormente. Por fim, os débitos entre cada par de nós seriam quitados ao longo da cadeia formada por A, B e C.

Deste modo, a Figura 2.1 na Seção 2.1.1 (pg 14), que ilustra os tipos de reciprocidade, pode ser atualizada para a Figura 7.1, que se limita a exibir os tipos de reciprocidade indireta, incluindo a reciprocidade transitiva. Ao invés de B ajudar C porque A ajudou B anteriormente (*pay-it-forward*), ou C ajudar A porque A ajudou B (reputação), C ajudará A porque A já ajudou B e B já ajudou C anteriormente.

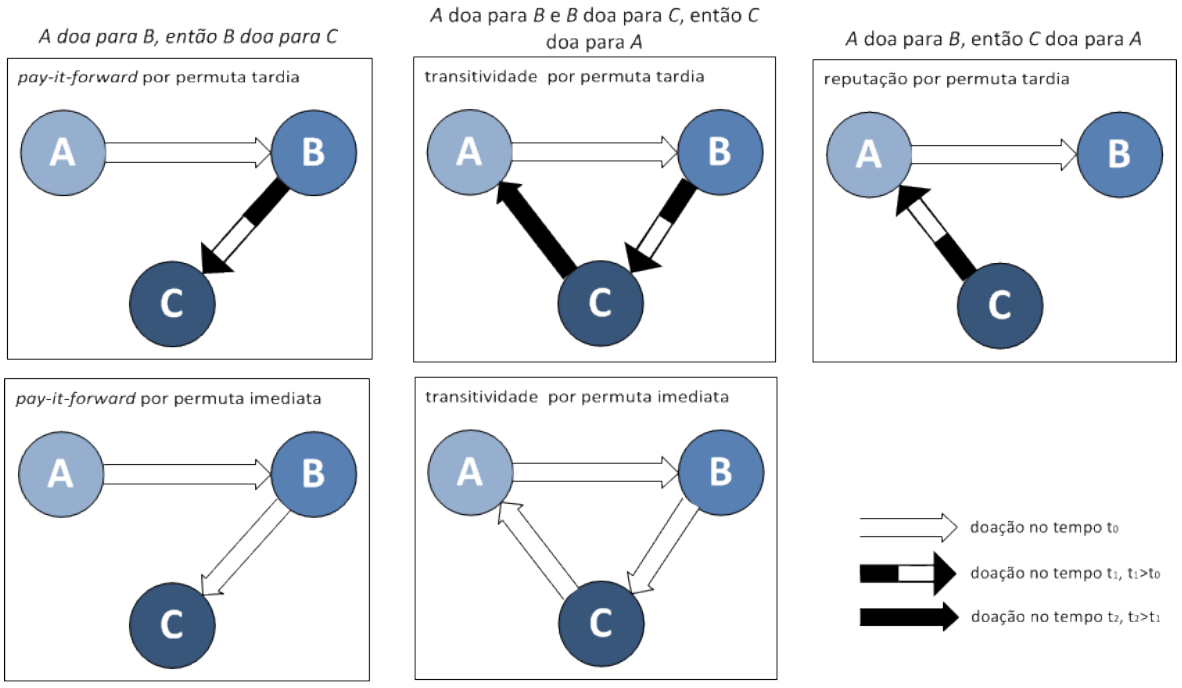


Figura 7.1: Mecanismos de incentivo à cooperação baseados em reciprocidade indireta.

Nenhum agente externo é necessário para realizar essa compensação multilateral, como requer os sistemas de pagamentos interbancários, uma vez que isso pode ser conseguido de

forma pareada, como explicado a seguir.

Assume-se apenas que cada nó tem um par de chaves pública e privada. Em um dado passo de tempo t, assuma que A está em estado consumidor. A envia primeiramente requisições para os nós com quem ele sabe que possui crédito, e se nenhum destes puder atender sua requisição, ele passa a enviar requisições a nós desconhecidos, escolhidos de forma aleatória, requisitando uma certa quantidade R de recursos, junto com um identificador de mensagem *mess_id*. A requisição chega em B (dentre outros nós). B (e talvez também os outros nós) verifica $\gamma(B, A, t)$, e percebe que está em débito com A. Para simplificar, assuma que a quantidade deste débito é R ou maior. O estado de B em t é tal que B não pode prover para A recurso algum, pelo menos nesse momento. B decide então enviar outra requisição à federação, pedindo a outros nós para prover R recursos para A por conta de B, com o intuito de cancelar eventual débito com B, encaminhando o identificador de mensagem *mess_id* junto com um novo identificador único ($nonce$). Um nó C (e talvez também outros nós) em estado provedor no momento t, recebe a requisição e percebe que está em débito com B. Assuma também que $\gamma(C, B, t) \geq R$. C então decide prover os recursos requisitados por A, enviando uma mensagem para A, junto com o mesmo identificador *mess_id*, o identificador de B, e o $nonce$. Se A aceitar a oferta, C provê os recursos requisitados diretamente para A. Uma vez terminado, assume-se que A assinará uma mensagem com sua chave privada, contendo o *mess_id*, o $nonce$, e o identificador de C, e a enviará para C (e talvez também para B, mas não é obrigatório). Ao assinar a mensagem, A irá reduzir o débito de B por R, considerando como tendo sido pago, e não faz nada além disso. A não irá registrar nenhum crédito para C, bem como C não irá registrar nenhum débito para A. Neste momento, C irá enviar esta mensagem assinada por A para B, com o objetivo de ter o favor prestado para A, reconhecido por B. B terá certeza que o favor foi feito por C para A (dado que A não teria nenhum benefício em mentir sobre este fato), e reduz o débito de C por R, o considerando pago. C também irá reduzir seu débito com B pela mesma quantidade. Neste ponto, os débitos entre A, B e C terão sido reduzidos por uma quantidade equivalente a R. O sistema, de modo geral, estará em um estado de menor risco, no sentido de que todos os nós estarão satisfeitos e inclinados a continuar cooperando entre si. Este esquema pode ser facilmente generalizado para cadeias de qualquer tamanho.

Por fim, o protocolo também deve resolver o problema da contabilidade autônoma, onde

cada nó contabiliza o crédito de forma diferente. O nó C, por exemplo, precisa decidir quantas unidades de seu recurso, R_C, equivaleria a R_B unidades de recurso na visão de B. Uma abordagem simples e comprovadamente efetiva é a utilização de *(micro)benchmarks* para realização de uma contabilidade relativa [79]. Se C executa seu microbenchmark em uma dada máquina virtual (VM, do inglês *Virtual Machine*) de sua posse, e este demora a metade do tempo que o mesmo microbenchmark leva para executar em uma VM de posse de B, então quando B pede para C doar R_B unidades de recurso para A, C deveria doar $\frac{R_C}{2}$ unidades de recurso. Neste caso, sempre que C requisitar e utilizar uma VM de B, C executaria seu microbenchmark, e ao longo do tempo obteria valores mais estáveis para a contabilidade relativa.

7.2.1 Possíveis Ataques

O protocolo descrito anteriormente possui algumas vulnerabilidades que giram em torno de A ou C trapacear, descritas a seguir.

Se A receber o favor mas se recusar a assinar essa mensagem, uma abordagem que C poderia seguir é de encarar este favor como sendo provido para A via reciprocidade direta, e não via transitividade (em nome de B). Ou então, C também poderia encarar esse favor como sendo provido para B via reciprocidade direta, independente do reconhecimento de B. Nestes casos, C simplesmente atualizaria suas anotações locais com um débito para A ou B. Contudo, B não iria diminuir $\gamma(B, A, t)$, e A poderia utilizar esse crédito para consumir recursos de outros nós via transitividade. Dessa maneira, esse crédito jamais seria gasto via transitividade, uma vez que A se negaria a assinar a mensagem reconhecendo o favor transitivo. Utilizando essa estratégia, A poderia gastar todos os créditos que o nó B teria com outros nós cooperativos. Por esta razão, essa abordagem não é adequada.

Se A se recusar a assinar essa mensagem, e C reivindicar para B que proveu o serviço para A, B não tem como descobrir quem está trapaceando, se é A por não ter assinado, ou C por não ter provido o serviço. Portanto, nesse caso, B deveria colocar tanto C como A em uma lista de nós não confiáveis mantida por B para evitar interações com nós suspeitos via transitividade. Adicionalmente, B manteria o débito de C (ao invés de reduzi-lo), e diminuiria o crédito de A por R. Isso desencoraja as trapaças por parte de C, pois C não conseguirá obter vantagem sobre B, já que seu débito com B só diminuirá mediante apresentação da

assinatura de A. C até poderia praticar essa calúnia apenas para difamar e prejudicar A, mas não obteria nenhuma vantagem com isto, muito pelo contrário, seria prejudicado, uma vez que não poderia interagir com B via transitividade, com o objetivo de receber serviços de B mais facilmente. A também não terá incentivo para trapacear, pois se A receber o recurso e não assinar, seu crédito $\gamma(B, A, t)$ será reduzido, como se A tivesse recebido o recurso normalmente, e além disso, B não voltará a interagir com A de maneira transitiva. Em suma, B não voltaria a interagir via transitividade com A nem com C, e A não teria mais crédito com B para utilizar de maneira irrestrita. Isso desencoraja as trapaças pois para cada R recursos que A ou C tentar consumir trapaceando, A ou C pagará a mesma quantidade via reciprocidade direta.

É importante ressaltar que o objetivo da reciprocidade transitiva é que ela evite situações de impasse, mas ela só obterá êxito se as partes envolvidas na cadeia de créditos agirem de modo cooperativo. Se pelo menos um membro trapacear, é normal que a confiança entre esses nós seja abalada, uma vez que não é possível descobrir de maneira exata quem é o trapaceiro, e portanto os nós deixarão de interagir entre si via reciprocidade transitiva.

7.2.2 Teoria dos Jogos na FD-NoF com Reciprocidade Transitiva (FD-NoF Transitiva)

A FD-NoF com reciprocidade transitiva (daqui por diante, FD-NoF Transitiva) é complexa de se descrever em termos da Teoria do Jogos. A maioria das bases teóricas utilizadas para reciprocidade indireta assumem que os indivíduos se encontram esporadicamente, e nestas ocasiões interagem de forma similar ao jogo do "dilema do prisioneiro" [75], onde em cada ocasião cada jogador coopera ou deserta.

Na NoF, a análise é um pouco mais complexa. Indivíduos não se encontram diretamente antes de requisitar recursos. Eles podem estar tanto em um estado consumidor, provedor, ou ocioso. Quando estiver em estado consumidor, um indivíduo A envia uma requisição para um nó específico, mas sem julgar seu grau de não-cooperatividade baseado nessa resposta. O fato de que um nó B não coopera com A nesse momento não indica que B é necessariamente um desertor/carona, mas talvez apenas que B também está em um estado consumidor ou ocioso (cf. deserção involuntária [36]), ou em um estado provedor mas com recursos escassos

porque outros nós que também estão requisitando recursos no mesmo momento podem ter maior prioridade na perspectiva de B, graças a contribuições passadas. Essa característica pode ser encarada como uma forma de *interação opcional* [71], mas certamente não seria suficiente para analisar a transitividade.

Uma vez que se acredita que o sistema resultante é matematicamente intratável, ou pelo menos muito complexo, este trabalho recorre a simulações para responder algumas questões básicas. Os resultados dessas simulações são apresentados a seguir, nas próximas seções. Contudo, vale salientar que já foi mostrado, sob circunstâncias semelhantes, que a reciprocidade direta é suficiente para garantir um estado de equilíbrio [19]. Dado que a reciprocidade indireta melhora o sistema, no sentido de que permite mais interações, espera-se que, nesse caso, um equilíbrio de Nash também seja alcançado em cenários semelhantes.

7.3 Avaliando a FD-NoF Transitiva

Para avaliar o desempenho da FD-NoF Transitiva, foi construído um simulador[b] para um modelo simplificado de uma federação P2P para compartilhamento de recursos, composta por provedores de computação na nuvem. O mesmo simulador pode ser utilizado para analisar os participantes tanto na FD-NoF como na FD-NoF Transitiva. Deste modo, é possível compreender as principais diferenças de cada abordagem.

7.3.1 Modelo de Simulação

O modelo de simulação é bem semelhante ao modelo utilizado no Capítulo 5. A federação consiste de uma comunidade com $\mathcal{N}$ nós, com $(1-\mathcal{F})\cdot\mathcal{N}$ nós cooperativos e $\mathcal{F}\cdot\mathcal{N}$ caronas, $0 \leq \mathcal{F} < 1$. A simulação acontece em passos discretos de tempo, que correspondem aos passos de tempo das funções degrau definidas nas Seções 2.2, 4.1 e 4.2 (pg 18, 46 e 48).

Supõe-se que cada nó tem uma capacidade total de recursos igual a $\mathcal{C}$. Em cada passo, cada participante pode estar tanto em *estado consumidor*, *ocioso* ou *provedor*. Um nó em estado consumidor em um dado passo de tempo t terá demanda total por recursos $\mathcal{D} = \mathcal{D}_{fed}+$

[b]O código-fonte do simulador da FD-NoF Transitiva, bem como os dados de entrada e saída reportados na tese, estão disponíveis publicamente no sítio `https://github.com/eduardolfalcao/nof-simulator`.

$\mathcal{C}$, das quais $\mathcal{C}$ unidades são satisfeitas por recursos locais e $\mathcal{D}_{fed}$ unidades são requisitadas à federação. Quando estiver ocioso, a quantidade local de recursos de um nó será suficiente para atender sua demanda total mas nada sobrará para a federação, *i.e.*, $\mathcal{D} = \mathcal{C}$ e portanto $\mathcal{D}_{fed} = 0$. Um nó estará em estado provedor quando sua demanda total for inferior à sua capacidade ($\mathcal{D} < \mathcal{C}$), o que significa que ele poderá ofertar até $\mathcal{C}$ unidades de recursos à federação. Para interações via reciprocidade direta, um nó $A \in \mathbb{F}_c$ em estado provedor no passo de tempo t pode prover até $\alpha(A, B, t)$ unidades de recursos a qualquer nó $B \in \mathbb{F}$ que esteja em estado consumidor no mesmo momento, como apresentado na Seção 4.1 (pg 46).

Lembrando que $\mathbb{F}_c$ denota o conjunto de nós cooperativos em $\mathbb{F}$, nas simulações da reciprocidade transitiva (FD-NoF Transitiva), um nó $C \in \mathbb{F}_c$ em estado provedor no passo de tempo t poderá prover para um nó $A \in \mathbb{F}_c$ em estado consumidor, em nome de um nó $B \in \mathbb{F}_c$, uma quantidade de recursos que não seja maior do que o mínimo entre $\alpha(C, B, t)$ (a quantidade que C proveria diretamente para B) e $\alpha(B, A, t)$ (a quantidade que B proveria para A, uma vez que B não iria solicitar que C provesse uma quantidade de recursos para A que fosse maior do que a quantidade que o próprio B ofertaria para A). Em suma, isto significa que os nós envolvidos na reciprocidade transitiva nunca doam mais do que o limite imposto pelo *controlador de recursos*.

Para avaliar os cenários em que a FD-NoF Transitiva poderia ser mais útil à federação, os nós cooperativos são divididos em três grupos, G_1, G_2 e G_3, com tamanhos iguais, cada um com $\frac{(1-\mathcal{F})\cdot\mathcal{N}}{3}$ nós. Assuma $\mathcal{E} = \{consumidor, ocioso, provedor\}$ como sendo o conjunto de estados que cada nó pode assumir, e $\mathcal{G} = \{G_1, G_2, G_3\}$ o conjunto de grupos.

Considere $\bar{e} : \mathbb{N} \rightarrow \mathcal{E} \times \mathcal{E} \times \mathcal{E}$ como sendo uma função vetorial tal que $\bar{e}(0) = (consumidor, ocioso, provedor)$, e $\bar{e}(t+1)$ é uma transposição cíclica para a direita de $\bar{e}(t)$ para todo $t \in \mathbb{N}$.

Considere também $e : \mathcal{G} \times \mathbb{N} \rightarrow \mathcal{E}$ como sendo uma função um-para-um partindo de $\mathcal{G}$ para $\mathcal{E}$, para cada $t \in \mathbb{N}$ tal que $e(G_i, t) = \bar{e}(t)[i]$ (o i-ésimo componente de $\bar{e}(t)$), para $i = 1, 2, 3$. Se $e(G, t) = estado$, diz-se que cada grupo G assumirá o dado $estado$ no passo de tempo t. Nas simulações realizadas, apresentadas abaixo, para cada t e grupo G, é considerado que G está no estado do grupo $e(G, t)$. Isto significa que um membro de G_1, G_2 e G_3, pode consumir de um membro de G_3, G_1 e G_2, respectivamente, mas nunca o oposto.

Adicionalmente, essas simulações também contemplam os cenários nos quais os nós

pertencentes a cada um desses grupos têm apenas uma probabilidade de acompanharem o estado do grupo. Desse modo, em cada passo de tempo t, assume-se que cada nó de um dado grupo tem uma probabilidade η de estar no estado do grupo, e probabilidade $\frac{1-\eta}{2}$ de desviar do estado do grupo e assumir um dos dois estados restantes. Portanto, nos cenários nos quais todos os nós seguem o estado do grupo de maneira estrita, a norma é configurada como $\eta = 1$. Se os nós de um grupo são configurados com $\eta = 0.8$, eles assumirão o estado do grupo com uma probabilidade de 80%, e estarão em cada um dos outros dois estados com probabilidade de 10%. Por fim, para simular uma federação na qual os nós têm a mesma probabilidade de estar em qualquer estado, todos os três grupos são configurados com $\eta = \frac{1}{3}$.

7.3.2 Cenários

Os resultados das simulações apresentados no Capítulo 5 mostram que tanto a SD-NoF como a FD-NoF proveem bons níveis de paridade para os nós cooperativos em cenários de alta contenção por recursos. Desse modo, o objetivo desta investigação são os cenários mais favoráveis aos caronas, *i.e.*, cenários nos quais há abundância de recursos na federação à disposição dos caronas, ou em outras palavras, cenários com baixa contenção de recursos.

Considerando que a federação é composta por três grupos de nós, e que o estado de cada grupo é configurado com a mesma probabilidade η de seguir o estado normativo, é possível observar que o valor esperado da quantidade total de nós no estado consumidor será igual ao valor esperado da quantidade total de nós no estado provedor, para qualquer passo de tempo t. Então, lembrando que a contenção de recursos é dada pela razão entre recursos requisitados e ofertados, o valor esperado do nível de contenção para cada cenário é $\mathbb{E}[\kappa] = \frac{\mathcal{D}}{\mathcal{C}}$.

Note que na FD-NoF os nós cooperativos podem reduzir a quantidade de recursos ofertada a outros nós, graças ao controlador de recursos, para valores inferiores a $\mathcal{C}$. Portanto, $\mathbb{E}[\kappa]$ define simplesmente um limite inferior para o valor esperado da contenção geral do sistema durante a simulação.

Para simular um cenário de baixa contenção por recursos, em que haja o dobro de recursos ofertados em relação à quantidade de recursos requisitada, i.e. $\mathbb{E}[\kappa] = 0.5$, configurou-se $\mathcal{C} = 1$ e $\mathcal{D}_{fed} = 0.5$. Adicionalmente, a comunidade foi configurada com $\mathcal{N} = 300$ para todos os cenários.

Nessas simulações, os caronas sempre estão em estado consumidor, tentando sempre consumir a maior quantidade de recursos que puderem, *i.e.*, $\mathcal{D}_{fed} = \infty$. Foram simulados cenários sem caronas, $\mathcal{F} = 0$, e cenários com um nível moderado de caronas, $\mathcal{F} = 0.25$.

Os mesmos cenários foram simulados tanto com a FD-NoF como com a FD-NoF Transitiva, para que seja possível compará-las. Para obter maior confiabilidade nos resultados, cada simulação foi replicada 30 vezes. As simulações executaram um total de 5000 passos de tempo, que se mostrou suficiente para levar o sistema a uma certa estabilidade.

Em suma, o projeto de simulação inclui os seguintes parâmetros constantes: a quantidade de nós da federação ($\mathcal{N} = 300$), a capacidade máxima de recursos dos nós ($\mathcal{C} = 1$), a quantidade de recursos requisitada por cada nó cooperativo em cada passo de tempo ($\mathcal{D}_{fed} = 0.5$ quando o nós estão em estado *consumidor*, e $\mathcal{D}_{fed} = 0$, caso contrário), a demanda dos caronas em cada passo de tempo ($\mathcal{D}_{fed} = \infty$) e o limite superior do controlador de recursos ($\mathcal{L}_{max} = 1$). Além disso, existem quatro variáveis: a proporção de caronas ($\mathcal{F}$), os demais parâmetros do controlador de capacidade, e a probabilidade (η) de um grupo de nós acompanharem o estado do grupo no passo de tempo t. Os três cenários compostos pela combinação dessas variáveis são descritos a seguir.

Cenário Básico

Aqui deseja-se entender o cenário básico descrito na Seção 7.2 (pg 98). Desse modo, os nós cooperativos foram divididos em três grupos com $\eta = 1$, e foram experimentados tanto com a FD-NoF como com a FD-NoF Transitiva, em uma federação sem caronas ($\mathcal{F} = 0$). No que concerne os parâmetros do controlador de capacidade, foram configurados $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 1$ e $\Delta = 0.05 \cdot \mathcal{C}$, uma configuração básica que é ajustada nos cenários seguintes para compreender melhor os resultados alcançados pelo controlador sob diferentes configurações. Como explicado acima, espera-se que a federação que utiliza a FD-NoF irá colapsar, enquanto a federação que utiliza a FD-NoF Transitiva irá prosperar justamente por causa da reciprocidade transitiva.

Impacto dos Caronas

Para este cenário a federação é configurada com 25% de caronas. Em cada passo de tempo da simulação, os caronas trocam suas identidades, aparecendo como nós recém-chegados. Desta maneira, um carona $P \in \mathbb{F}$ evita que o limite $\alpha(A, P, t)$ que um nó $A \in \mathbb{F}_c$ usa para restringir a oferta a P seja calculada de acordo com sua paridade individual $\phi(A, P, t)$, que

irá diminuir rapidamente para 0. Como resultado, A irá usar sempre sua paridade com a federação $\phi(A, t)$, que na maioria das vezes será maior do que sua paridade bilateral, para definir $\alpha(A, P, t)$.

Para melhor entendimento do impacto dos caronas em cenários com diferentes configurações para o controlador, Δ foi configurado com os valores $0.01 \cdot \mathcal{C}$, $0.05 \cdot \mathcal{C}$ e $0.1 \cdot \mathcal{C}$. Adicionalmente, $\mathcal{L}_{min}$ foi variado assumindo os valores do conjunto $\{0.75, 0.85\}$. Com isso é possível avaliar a influência de Δ na discriminação dos caronas, além de avaliar até que ponto o controlador é capaz de assegurar aos nós cooperativos os níveis desejados de paridade.

Impacto da Assimetria de Interesses

Por fim, foram avaliados e comparados os desempenhos dos nós cooperativos executando a FD-NoF e a FD-NoF Transitiva, quando os diferentes cenários são experimentados com diferentes valores de η, $\eta \in \{0.9, 0.75, 0.5, 1/3\}$. Para cada cenário, os três grupos foram configurados com os mesmos valores de η.

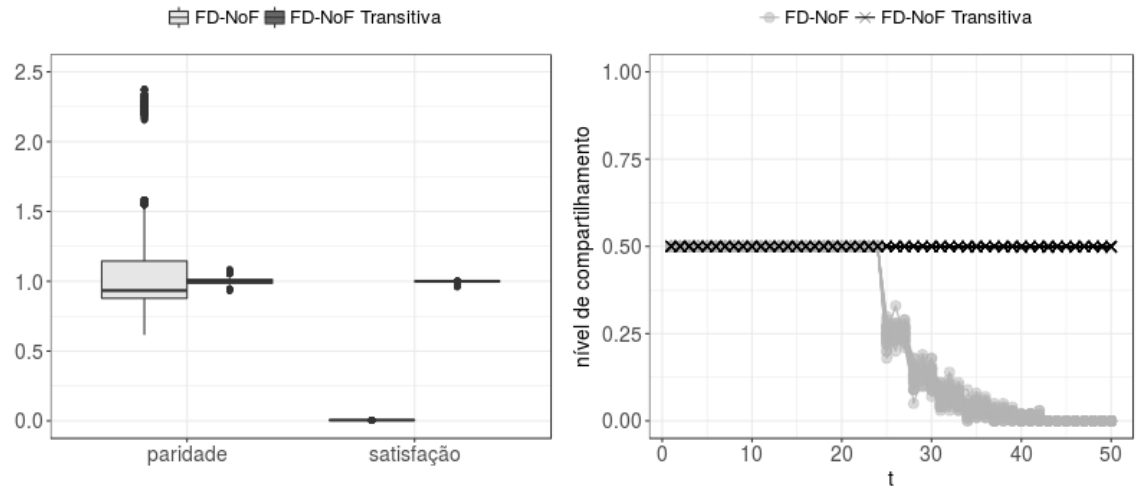
7.4 Resultados e Análise

Os resultados e análises de cada cenário são apresentados a seguir.

7.4.1 Cenário Básico

Na Figura 7.2a, é possível avaliar o desempenho dos nós cooperativos, tanto na FD-NoF como na FD-NoF Transitiva, observando os diagramas de caixa referentes a seus níveis individuais de paridade e satisfação para todos os nós, no último passo da simulação. Adicionalmente, a Figura 7.2b apresenta a evolução dos níveis de compartilhamento de recursos (cada linha é uma replicação) no início da simulação, para que se possa identificar em que passo de tempo a federação colapsa sob a FD-NoF. O *nível de compartilhamento de recursos* é definido como a razão entre quantidade total de recursos doados e quantidade total de recursos ofertados à federação no tempo t. Cada caso foi replicado 30 vezes, gerando um total de 60 simulações.

A primeira conclusão que pode ser tirada a partir da Figura 7.2b é que, quando os nós



(a) Paridade e satisfação dos nós cooperativos no último passo (5000) da simulação.

(b) Nível de compartilhamento da federação ao longo dos 50 passos de tempo iniciais.

Figura 7.2: Federação: $\mathcal{N} = 300$ e $\mathcal{F} = 0$. Controlador de capacidade: $\Delta = 0.05$, $\mathcal{L}_{min} = 0.75$ e $\mathcal{L}_{max} = 1$.

usam a FD-NoF, a federação colapsa entre os passos 37 e 43. Devido ao modo com que os consumidores e provedores alternam seus estados em cada passo de tempo, um nó $B \in \mathbb{F}_c$ em um dado grupo G_2 sempre irá aparentar ser um carona para qualquer nó $A \in \mathbb{F}_c$ em outro grupo G_1 (de acordo com as definições na Seção 7.3.1, pg 103), e portanto todo nó $A \in \mathbb{F}_c$ irá diminuir passo a passo $\alpha(A, B, t)$ para 0, levando a federação a um colapso. Esta também é a razão pela qual a satisfação dos nós cooperativos converge para 0. Não obstante, até mesmo neste cenário nos quais os nós cooperativos aparentam ser caronas, observou-se que a FD-NoF apresentou bom desempenho, assegurando a 85% dos nós cooperativos níveis de paridade maior do que $\mathcal{L}_{min} = 0.75$. Em síntese, a FD-NoF é bastante efetiva em prevenir que nós cooperativos participem de interações injustas.

Por outro lado, a presença da reciprocidade transitiva se mostrou crucial para que a federação continuasse funcionando como esperado: os níveis de compartilhamento estabilizaram em 0.5, que é o maior e melhor nível possível, dado que $\mathbb{E}[\kappa] = 0.5$ e que não havia nenhum carona na federação.

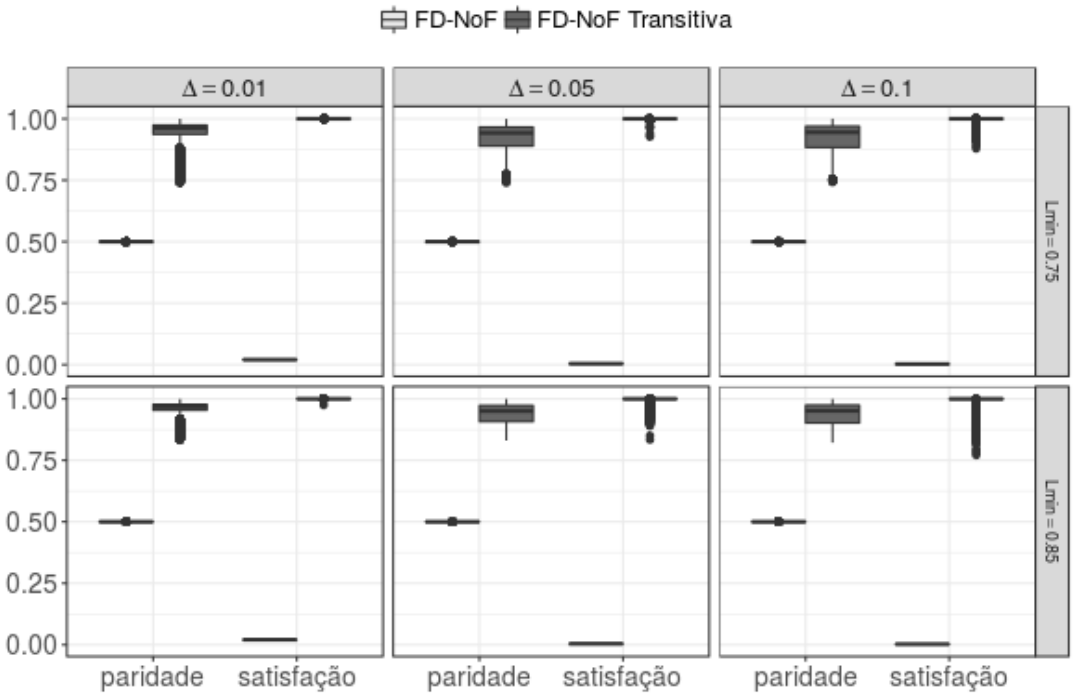
7.4.2 Impacto dos Caronas

Neste cenário a federação é novamente composta por 300 nós, mas destes, 25% são caronas e trocam suas identidades a cada passo de tempo. O controlador de capacidade foi configurado com Δ assumindo os valores do conjunto $\{0.01 \cdot \mathcal{C}, 0.05 \cdot \mathcal{C}, 0.1 \cdot \mathcal{C}\}$. Os diagramas de caixa da Figura 7.3a mostram as distribuições dos níveis de paridade e satisfação dos nós cooperativos no último passo das 30 replicações, em ambas FD-NoF e FD-NoF Transitiva. Cada curva da Figura 7.3b representa o nível de compartilhamento de recursos na federação em uma simulação. Com a ajuda desses resultados, é possível avaliar como os diferentes valores de Δ impactam no desempenho dos nós cooperativos com relação à discriminação dos caronas.

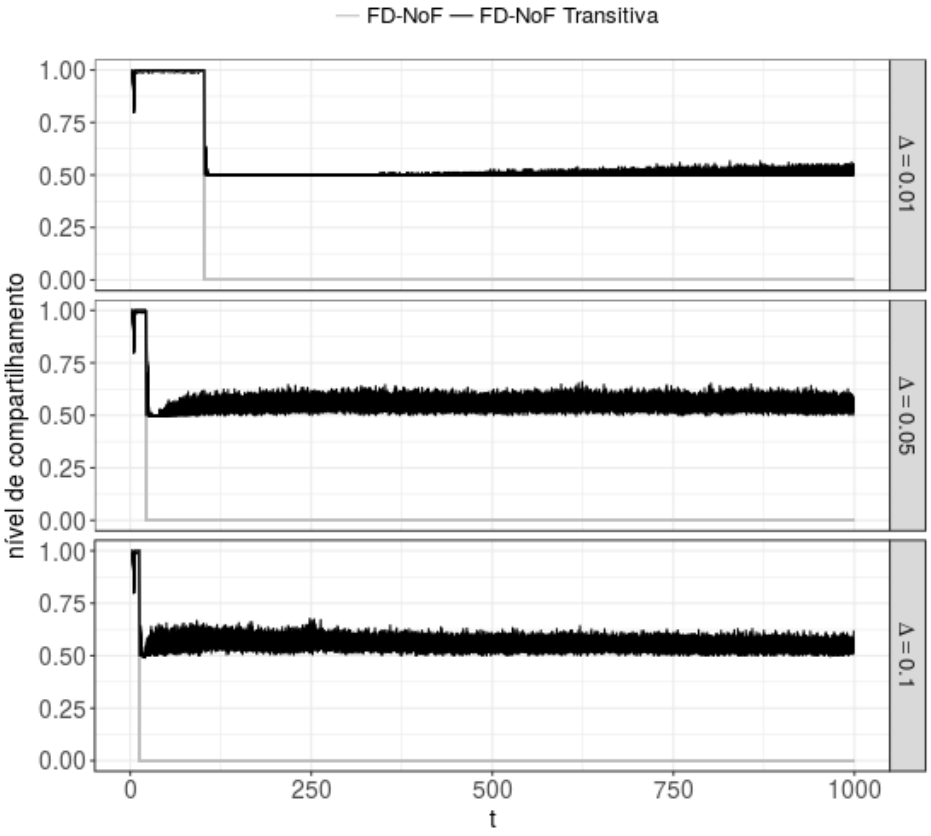
Como esperado, é possível observar nas Figuras 7.3a e 7.3b que a FD-NoF colapsa – o nível de compartilhamento da federação converge para 0, e como resultado os níveis de satisfação se aproximam de zero – independentemente dos valores de Δ e $\mathcal{L}_{min}$. Por outro lado, mais de 98% dos nós cooperativos alcançaram seus objetivos de obterem níveis de paridade $\geq \mathcal{L}_{min}$ quando usam a reciprocidade transitiva. Apenas 678 nós cooperativos, de um total de 40500, não conseguiram alcançar os valores desejados de paridade, e ainda assim muitos deles alcançaram valores muito próximos do seu objetivo, como pode-se observar na Figura 7.3a.

Apesar dos cenários com os três valores diferentes para Δ obterem resultados positivos, existe um compromisso que um nó deve avaliar para escolher o valor mais adequado a seus objetivos. Valores muito pequenos, como $\Delta = 0.01 \cdot \mathcal{C}$, levam a uma reação mais lenta dos nós cooperativos contra os caronas. Isto pode ser observado na Figura 7.3b, onde para $\Delta = 0.01 \cdot \mathcal{C}$ os caronas são isolados[c] apenas no passo 103, enquanto que para $\Delta = 0.05 \cdot \mathcal{C}$ e $\Delta = 0.1 \cdot \mathcal{C}$, os caronas são isolados nos passos 22 e 13, respectivamente. Por outro lado, valores maiores para Δ aumentam a oscilação do nível de compartilhamento da federação, que irá flutuar em um nível um pouco maior do que 0.5, aumentando portanto a quantidade de recursos provida a caronas. Portanto, o compromisso está relacionado ao fato de que quanto maior for Δ, mais rápido os caronas são discriminados, mas a longo prazo, as chances de um carona consumir algum recurso também são um pouco maiores. Por outro lado, pode-se observar que quanto menor o valor de Δ mais demorada é a discriminação dos caronas;

[c] Dado que os caronas têm demanda infinita e que $E[k] = 0.5$, pode-se assumir que eles são isolados quando o nível de compartilhamento da federação se aproxima de 0.5.



(a) Paridade e satisfação dos nós cooperativos no último passo das simulações.



(b) Nível de compartilhamento da federação para $\mathcal{L}_{min} = 0.75$.

Figura 7.3: Federação: $\mathcal{N} = 300$ e $\mathcal{F} = 0.25$. Controlador de capacidade: $\Delta \in \{0.01 \cdot \mathcal{C}, 0.05 \cdot \mathcal{C}, 0.1 \cdot \mathcal{C}\}$ e $\mathcal{L}_{max} = 1$.

contudo, a longo prazo, a probabilidade de um carona consumir recursos é um pouco menor.

Por isso, uma generalização razoável seria a seguinte: usar baixos valores para Δ para participações de longo prazo em uma federação, e altos valores para Δ para breves participações em uma federação.

Com respeito à configuração de $\mathcal{L}_{min}$, pode-se notar que a FD-NoF Transitiva é muito efetiva para os dois valores experimentados: $\mathcal{L}_{min} = 0.75$ e $\mathcal{L}_{min} = 0.85$. Quando $\mathcal{L}_{min} = 0.75$, mais de 99% dos nós cooperativos alcançam níveis de paridade maiores do que 0.75, e quando $\mathcal{L}_{min} = 0.85$, mais de 98% dos nós cooperativos alcançam os valores desejados de paridade. Note que em ambos os cenários a satisfação da maioria dos nós fica próxima de 1, significando que quase 100% de suas requisições são atendidas.

Para fins de verificação, algumas simulações também foram executadas com $\mathcal{L}_{min} = 0.95$. Nesses casos, apenas 41% dos nós cooperativos alcançam os níveis desejados de paridade. Mas os valores mínimo (0.9), médio (0.96) e máximo (≈ 1) mostram que seus níveis de paridade não ficaram muito distantes do desejado, 0.95. Finalmente, seus níveis de satisfação foram substancialmente afetados, diminuindo de uma média de ≈ 1 para $\mathcal{L}_{min} \in \{0.75, 0.85\}$ para uma média de 0.93 para $\mathcal{L}_{min} = 0.95$.

A seleção do valor para $\mathcal{L}_{min}$ também traz consigo outra relação de custo e benefício. Quanto maior for $\mathcal{L}_{min}$, mais difícil será para os nós cooperativos alcançarem os níveis desejados de paridade, e menor serão os níveis de satisfação. Portanto, para as simulações do próximo cenário, $\mathcal{L}_{min}$ foi fixado em 0.85.

7.4.3 Impacto da Assimetria de Interesses

Nos cenários anteriores, η foi configurado como 1, o que significa que o estado de cada nó cooperativo é igual ao estado do grupo em qualquer momento. Outra investigação interessante é encontrar o valor de η no qual a FD-NoF não irá colapsar, e comparar o desempenho dos sistemas com e sem reciprocidade transitiva. Portanto, para este cenário, o estado de cada nó cooperativo é configurado com η variando sobre o conjunto $\{0.9, 0.75, 0.5, 1/3\}$. Os diagramas de caixa na Figura 7.4 mostram o desempenho dos nós cooperativos no último passo das simulações.

A primeira conclusão que pode ser tirada da Figura 7.4 é que com um pequeno desvio de 10% do padrão, *e.g.*, $\eta = 0.9$, a FD-NoF é suficiente para manter a federação funcionando.

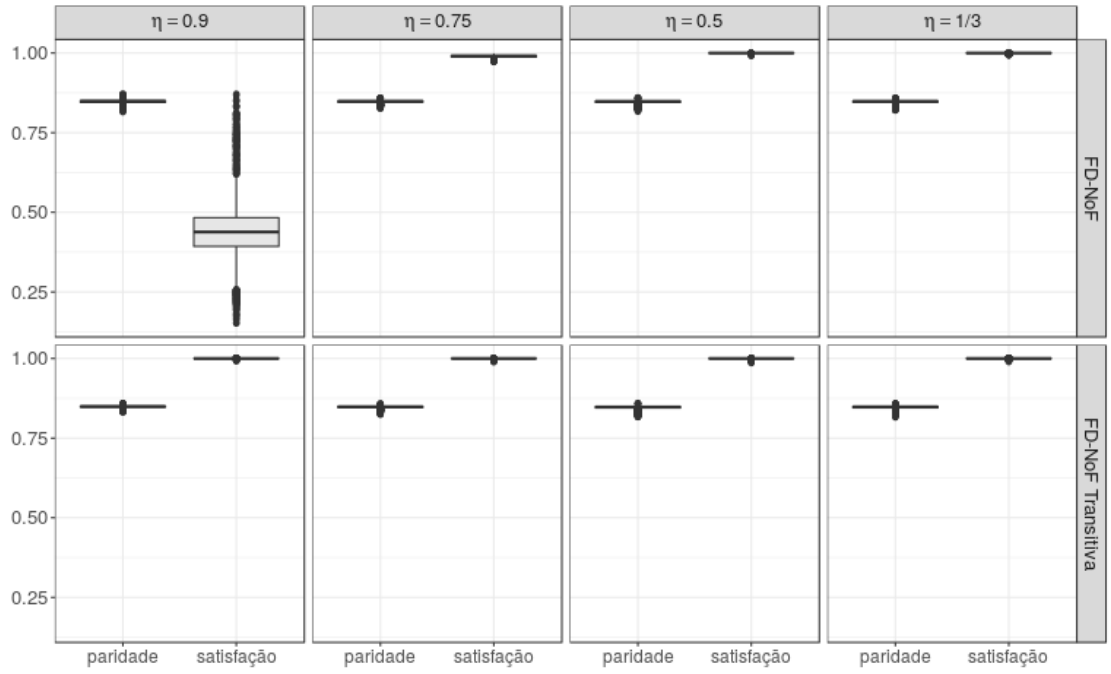


Figura 7.4: Paridade e satisfação dos nós cooperativos no último passo da simulação. Federação: $\mathcal{N} = 300$, $\mathcal{F} = 0.25$, e $\eta \in \{0.9, 0.75, 0.5, \frac{1}{3}\}$. Controlador de capacidade: $\Delta = 0.05 \cdot \mathcal{C}$, $\mathcal{L}_{min} = 0.85$ e $\mathcal{L}_{max} = 1$.

Tanto na FD-NoF como na FD-NoF Transitiva os nós cooperativos obtêm os níveis desejados de paridade, porém, seus níveis de satisfação na FD-NoF são em média 0.44, um valor baixo, considerando que $\mathbb{E}[\kappa] = 0.5$. Para este caso, a FD-NoF Transitiva provê aos nós cooperativos níveis de satisfação ≈ 1.

Quando $\eta = 0.75$, pode-se observar que os níveis de paridade dos nós cooperativos na FD-NoF e FD-NoF Transitiva são praticamente iguais. Contudo, os níveis de satisfação dos nós cooperativos na FD-NoF – em média 0.99 – é ligeiramente inferior do que na nova abordagem – em média ≈ 1.

Os resultados para $\eta \in \{0.5, \frac{1}{3}\}$ são extremamente similares às simulações com $\eta = 0.75$. Para estes cenários, o desempenho da FD-NoF é basicamente igual ao da FD-NoF Transitiva: níveis médios de paridade iguais a 0.85 e níveis médios de satisfação ≈ 1. Como esperado, quanto menor for o valor de η, menor será a melhoria de desempenho proporcionada pela reciprocidade transitiva. Apesar de existirem poucos cenários nos quais a FD-NoF Transitiva possui um desempenho substancialmente melhor do que a FD-NoF – cenários em que os nós cooperativos são configurados com altos valores de η – a principal conclusão é que recipro-

cidade direta e reciprocidade indireta por transitividade juntas geram melhores resultados do que apenas reciprocidade direta.

7.5 Considerações

A reciprocidade transitiva foi projetada para ser utilizada em conjunto com mecanismos de reciprocidade direta, uma vez que estes proveem aos nós uma visão bastante limitada do sistema. A transitividade visa aumentar o nível de colaboração entre nós cooperativos, principalmente em cenários com assimetria de disponibilidades ou interesses, mas sem violar a principal vantagem da reciprocidade direta, a integridade das informações acerca dos comportamentos dos demais nós. Para isto, três ou mais nós irão interagir via reciprocidade transitiva apenas se houver uma cadeia de créditos entre eles. A informação mais sensível para a formação de uma interação transitiva está no próprio nó provedor de recursos, uma vez que ele só doará recursos via transitividade se estiver em débito com algum nó que esteja em débito com o nó solicitante.

A reciprocidade transitiva já foi abordada em outros trabalhos nos contextos da permuta imediata [32; 90] e da permuta tardia [16; 62; 70]. As principais diferenças entre eles e o presente trabalho são os protocolos utilizados e contextos experimentados, que podem requerer tratamento especial. Destes, apenas o trabalho de Landa *et al.* (2009) trata de um contexto genérico, aplicável a qualquer tipo de recurso, enquanto os demais trabalhos foram pensados para sistemas P2P para compartilhamentos de arquivos. Os protocolos mais interessantes dos trabalhos correlatos são aqueles utilizados no *Scrivener* [70] e no *CompactPSH* [16]:

- Scriviner: uma cadeia de créditos é transformada em um único crédito que pode ser utilizado via reciprocidade direta entre o nó consumidor e provedor;
- CompactPSH: um nó consumidor A requisita um certificado a um nó B que deve para A e que tem crédito com C, e de posse desse certificado, A consome recursos de C.

Acredita-se que a mudança do protocolo não acarrete em mudanças nos resultados das simulações realizadas, uma vez que os protocolos apenas viabilizam a mesma interação via reciprocidade transitiva através de abordagens diferentes. Por outro lado, o protocolo apresentado neste trabalho é mais eficiente em termos de quantidades de mensagens trocadas –

para permitir uma interação via transitividade entre três nós, o protocolo da FD-NoF Transitiva precisa de apenas 5 mensagens, enquanto o Scrivener e o CompactPSH utilizam respectivamente 6 e 10 mensagens (cf. [70; 16]).

Outro ponto interessante que diferencia este trabalho dos demais é que, nos cenários avaliados com $\eta = 1$, o mecanismo de reciprocidade direta (FD-NoF) por si só faz a federação entrar em colapso, uma vez que o controlador de capacidade diminui a oferta de recursos para zero; para esses tipos de mecanismo, a transitividade é de suma importância. Adicionalmente, até onde se sabe, este é o único trabalho que analisa a transitividade em sistemas P2P para compartilhamento de recursos computacionais, e por isto, detalhes adicionais sobre a contabilidade de recursos precisam ser considerados no protocolo.

Neste trabalho, os cenários são avaliados sob uma perspectiva diferente, considerando um baixo nível de contenção por recursos, que favorece os caronas, e o grau de assimetria de interesses/disponibilidades (η). A partir dessa avaliação é possível concluir que apenas altos níveis de assimetria, $\eta = 0.9$, tornam a reciprocidade transitiva imprescindível à federação. Sob um menor nível de assimetria, $\eta = 0.75$, a reciprocidade transitiva ainda proporciona aos nós cooperativos, em média, melhores níveis de satisfação, ainda que a diferença seja pequena. Deste modo, é possível concluir que a reciprocidade transitiva aliada à reciprocidade direta sempre irão garantir resultados melhores ou iguais do que apenas reciprocidade direta, e esta melhoria tende a ser mais significativa para federações com poucos participantes, como por exemplo, federações de provedores privados de computação na nuvem.

Capítulo 8

Considerações Finais

Este capítulo encerra o trabalho discutindo suas contribuições, apresentando as principais conclusões e apontando linhas de investigação a serem conduzidas.

8.1 Contribuições e Conclusões

Este trabalho abordou a problemática da ineficiência dos mecanismos de reciprocidade em assegurar aos nós cooperativos níveis adequados de paridade quando a demanda agregada dos nós cooperativos é menor do que a oferta de recursos à comunidade. Este problema é particularmente relevante para comunidades de compartilhamento de recursos nas quais o custo do recurso compartilhado não é desprezível, como por exemplo, federações de provedores de computação na nuvem. Nestes casos, a paridade entre a quantidade de recursos recebidos e doados deve ser assegurada para que os provedores IaaS de nuvem se sintam motivados a entrar e permanecer na federação.

Através de uma revisão de literatura, apresentada no Capítulo 3, foi constatado que este problema decorre da estratégia utilizada para promover cooperação nas comunidades. Basicamente, os mecanismos de reciprocidade definem estratégias de priorização dos nós solicitantes baseadas em sistemas de crédito ou reputação, estratégias estas que garantem paridade apenas quando a contenção de recursos é moderada ou alta. É possível que esta situação tenha chamado pouca atenção dos pesquisadores pois os mecanismos de reciprocidade eram majoritariamente aplicados em sistemas P2P de compartilhamento de arquivos ou grades computacionais P2P, onde o custo dos recursos compartilhados (respectivamente rede e ci-

clos de processamento ociosos de computadores de propósito geral) é considerado baixo. O custo dos recursos também é a razão pela qual a maioria dos trabalhos correlatos sobre federações de nuvens utilizam abordagens monetárias, pois nesses casos os caronas não conseguem consumir recursos sem oferecer uma compensação; apenas [39] e [78] utilizam um mecanismo de escambo baseado em reciprocidade, embora funcionem com topologia centralizada.

O Capítulo 2 explica o funcionamento dos mecanismos de reciprocidade, categorizando-os quanto à forma (direta ou indireta) e ao momento (permuta tardia ou imediata) da retribuição. Adicionalmente, são discutidas as principais características e vantagens/desvantagens de cada modalidade. Essa discussão pode auxiliar pesquisadores a discernirem com mais clareza qual tipo de reciprocidade é mais adequado para um determinado contexto.

Para solucionar o problema dos baixos níveis de paridade em cenários de baixa contenção de recursos, este trabalho propõe que cada nó cooperativo seja equipado com um mecanismo que monitora continuamente seus níveis de paridade para decidir a quantidade de recursos que deve ser ofertada a seus pares. Na prática, este mecanismo tende a baixar a oferta de recursos para nós caronas e manter quotas generosas para participantes recíprocos. Esta ação coletiva dos nós cooperativos tende a elevar a contenção de recursos quando esta for considerada demasiadamente baixa, e é suficientemente inteligente para não aumentar significativamente os níveis de contenção em cenários nos quais a contenção já seja elevada.

O Capítulo 4 apresenta uma implementação para este mecanismo, que por reutilizar a noção da Rede de Favores é chamado de Rede de Favores Dirigida a Paridade (FD-NoF). A FD-NoF requer que cada nó cooperativo defina dois valores, $\mathcal{L}_{min}$ e $\mathcal{L}_{max}$, especificando o intervalo dos patamares desejados de paridade, além de um valor para Δ, que define o montante pelo qual a quota para os demais nós deve ser aumentada ou diminuída em cada verificação do laço de controle retroalimentado.

O desempenho da FD-NoF é apresentado no Capítulo 5 com o auxílio de um modelo de simulação simplificado. Diferentes cenários com níveis de contenção baixo, moderado e alto, e diferentes porcentagens de caronas, além de diferentes configurações do mecanismo FD-NoF foram simulados, de onde foram tiradas as seguintes conclusões. Quando o nível de contenção é muito baixo ($\mathbb{E}[\kappa] = 0.25$) ou a porcentagem de caronas é muito alta ($\mathcal{F} = 0.75$), a federação se torna inviável, pois o impacto inicial dos caronas causa uma deserção

em massa dos nós cooperativos. Porém, para os casos com $\mathcal{F} = 0.25$ e $\mathbb{E}[\kappa] = 0.5$, por exemplo, a FD-NoF conseguiu elevar os níveis de paridade em aproximadamente 61% com uma redução de apenas 2% dos níveis de satisfação. Para os cenários com $\mathcal{F} = 0.25$ e $\mathbb{E}[\kappa] \geq 1$, a FD-NoF alcançou resultados semelhantes aos da SD-NoF, o que é bom, uma vez que a estratégia da SD-NoF é a melhor possível quando a quantidade de recursos ofertados é menor ou igual à quantidade de recursos demandados pelos nós cooperativos. Além disto, a FD-NoF assegura paridade aos nós com baixa demanda e/ou baixa frequência de consumo, o que também torna a federação atrativa para provedores de nuvem com essa característica.

Com relação aos parâmetros do controlador, uma análise de sensibilidade investigou o desempenho da federação sob diferentes valores para $\mathcal{L}_{min}$, $\mathcal{L}_{max}$ e Δ. A partir desta análise observou-se que valores muito altos para $\mathcal{L}_{min}$ ($\mathcal{L}_{min} \geq 0.85$) dificultam a tarefa do controlador, que na maioria dos casos diminui a oferta para zero, inviabilizando a federação. No entanto, não houve diferença significativa ao configurar $\mathcal{L}_{max}$ com os valores 0.95 e 1. Adicionalmente, é possível afirmar que a velocidade de reação dos nós cooperativos contra os caronas é diretamente proporcional ao valor de Δ, embora menores valores favoreçam uma estabilidade na interação com nós cooperativos a longo prazo. Por estes motivos, este trabalho recomenda os seguintes valores para uma configuração conservadora: $\mathcal{L}_{min} = 0.75$, $\mathcal{L}_{max} = 0.95$ e $\Delta = 0.05 \cdot \mathcal{C}$.

Com o objetivo de elevar a confiabilidade dos resultados obtidos com auxílio do simulador, a FD-NoF foi implementada no *middleware* Fogbow e experimentos controlados foram conduzidos para realizar uma avaliação de desempenho. Neste sentido, o Capítulo 6 descreve a arquitetura do Fogbow, explicando com mais detalhes os componentes relacionados à operacionalização do mecanismo de escambo, e como a FD-NoF foi inserida nesse contexto.

A comparação entre os resultados do simulador e experimentos com o Fogbow sugerem que o simulador pode servir como uma ferramenta de previsão dos resultados para cenários em que a contenção não aumenta significativamente ao longo do tempo, ou seja, os cenários de baixa contenção sem caronas e os cenários de baixa contenção com caronas regidos pela FD-NoF. No entanto, foi possível identificar algumas simplificações do simulador que podem distanciar alguns de seus resultados da realidade. As duas principais simplificações do simulador são a maneira simples e imediata com que os consumidores encontram provedores

e o não reescalonamento de requisições não atendidas em um determinado momento. Sem essas duas simplificações, a contenção de recursos em alguns experimentos executados no Fogbow tende, em longo prazo, a aumentar, o que não acontece no simulador. Isto aconteceu especificamente nos cenários de baixa contenção de recursos, com caronas e regulado pela SD-NoF, razão pela qual há certa disparidade nos resultados desse cenário específico.

Por fim, este trabalho apresenta um estudo complementar sobre um tipo de mecanismo de reciprocidade pouco utilizado na prática, a reciprocidade transitiva. O objetivo da reciprocidade transitiva é permitir cooperação entre participantes com assimetria de tempos ou interesses, se apresentando, para isto, como um tipo de reciprocidade indireta mais restrito, que não permite a formação de conluio entre participantes mal-intencionados e, portanto, mantém os benefícios da reciprocidade direta. Resultados de simulações mostram que em cenários extremos, *i.e.*, cenários em que os participantes possuem alto grau de assimetria, a transitividade se torna imprescindível, sendo um fator decisivo na cooperação entre esses participantes e, consequentemente, evitando o colapso da federação. No entanto, em cenários com baixo nível de assimetria, cenários que acreditamos ser mais realistas, o ganho proporcionado pelo acréscimo da transitividade à reciprocidade direta é quase imperceptível. O fato é que a reciprocidade transitiva aliada à reciprocidade direta tende a garantir resultados melhores ou iguais aos resultados de mecanismos baseados exclusivamente em reciprocidade direta, e esta melhoria tende a ser mais significativa para federações compostas por poucos participantes.

8.2 Trabalhos Futuros

Os resultados obtidos neste trabalho de tese expõem algumas ramificações que podem ser exploradas em trabalhos futuros. Essas linhas de investigação são descritas a seguir.

Comparação com outros trabalhos correlatos através simulações

A partir de uma revisão de literatura foram encontrados dois trabalhos [39; 78] que implementam um mecanismo de escambo baseado em reciprocidade para federações de nuvens. Como esses trabalhos são implementados de forma centralizada, é possível que seus desempenhos na combinação entre provedores e consumidores sejam ótimo, ou pelo menos muito

próximo do ótimo. Neste sentido, seria interessante realizar uma comparação entre os desempenhos da FD-NoF Transitiva e dos mecanismos propostos por Goher *et. al.* (2018) e Sadok *et. al.* (2017), a partir da execução dos mesmos cenários com baixos e moderados níveis de contenção.

Remoção de algumas simplificações do simulador

A validação do simulador apontou algumas simplificações que, em alguns cenários, distanciam o simulador da realidade. Ainda mantendo o modelo de simulação simples, onde o tempo é discretizado em turnos, seria interessante reescalonar requisições não atendidas para o turno subsequente e não permitir que os recursos pudessem ser segmentados em frações menor do que a unidade. Uma mudança mais significativa seria não considerar que consumidores e provedores são combinados de maneira simples e imediata — para remover esta simplificação, no entanto, o tempo deveria ser modelado como uma dimensão contínua.

Infraestrutura de experimentos mais realista

Em um primeiro momento, a simplificação de alguns detalhes da infraestrutura dos experimentos é importante para que se possa controlar determinados aspectos do experimento e facilitar a interpretação dos resultados. Em termos de infraestrutura, seria interessante experimentar a FD-NoF com provedores IaaS reais e utilizar uma topologia de rede de longa distância (WAN, do inglês *Wide Area Network*), que torna os atrasos provenientes da comunicação dos participantes mais realistas do que aqueles praticados em uma rede de área local (LAN, do inglês *Local Area Network*). Outro aspecto que poderia tornar os experimentos mais realistas é a carga de trabalho. Nesse sentido, seria interessante experimentar a FD-NoF com cargas de trabalho provenientes de federações reais, ou ao menos sintetizá-las a partir de rastros de sistemas reais.

Investigar impactos da reciprocidade transitiva em cenários mais realistas

O Capítulo 7 apresentou uma investigação que justifica a importância da reciprocidade transitiva em comunidades nas quais os participantes têm entre si um certo grau de assimetria de interesses ou tempos. Contudo, os cenários simulados são simples e não-representativos, *i.e.*, as cargas de trabalho foram sintetizadas especificamente para avaliar os impactos da

reciprocidade em comunidades com diferentes níveis de assimetria. Portanto, uma investigação mais interessante seria analisar os impactos da reciprocidade transitiva com cargas de trabalho provenientes de federações reais, ou pelo menos com cargas de trabalho sintetizadas a partir de rastros de sistemas reais. Ainda nessa linha de investigação, outra questão a ser analisada é o impacto da transitividade considerando cadeias de crédito de diferentes tamanhos, ao invés de considerar somente cadeias que envolvam três nós. Por fim, uma prova de conceito da reciprocidade transitiva poderia ser implementada no *middleware* Fogbow, com o objetivo de validar o protocolo de comunicação que viabiliza a doação por transitividade e realiza a compensação de créditos, bem como entender os principais desafios envolvidos em uma implementação prática desse mecanismo.

Criptomoedas e *Blockchain*

No Capítulo 3 foi mostrado que a maioria dos trabalhos correlatos utilizam abordagens monetárias para assegurar a paridade entre recursos doados e recebidos em federações de nuvens. Neste sentido seria interessante investigar a aplicabilidade de criptomoedas em federações de nuvens, trazendo um entendimento de suas principais vantagens e desvantagens nesse contexto. Além disto, também seria interessante investigar a aplicabilidade de contratos inteligentes (*smart contracts*, em inglês) na implementação de um mercado para federações de nuvens.

Na mesma linha de pesquisa, um ponto que também merece investigação é a utilização da tecnologia *blockchain* para implementar mecanismos de reciprocidade. A existência de um livro-razão em mecanismos de reciprocidade indireta, principalmente em sistemas de reputação, poderia simplificar de modo significativo a verificação das contribuições (ou reputação) e auditoria de eventuais conluios.

Bibliografia

[1] Bittorrent forum for developers. http://www.bittorrent.org/. Acessado em 19/10/2016.

[2] Distributed.net. http://www.distributed.net/. Acessado em 06/11/2016.

[3] Great internet mersenne prime search - gimps. http://www.mersenne.org/. Acessado em 06/11/2016.

[4] Franklin Allen and Douglas Gale. Financial Contagion. Technical report, 1998.

[5] David P. Anderson. Boinc: A system for public-resource computing and storage. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, GRID '04, pages 4–10, Washington, DC, USA, 2004. IEEE Computer Society.

[6] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: An experiment in public-resource computing. *Commun. ACM*, 45(11):56–61, November 2002.

[7] N. Andrade, F. Brasileiro, W. Cirne, and M. Mowbray. Discouraging free riding in a peer-to-peer cpu-sharing grid. In *High performance Distributed Computing, 2004. Proceedings. 13th IEEE International Symposium on*, pages 129–137, June 2004.

[8] Nazareno Andrade, Francisco Brasileiro, Walfredo Cirne, and Miranda Mowbray. Automatic grid assembly by promoting collaboration in peer-to-peer grids. *Journal of Parallel and Distributed Computing*, 67(8):957 – 966, 2007.

[9] Nazareno Andrade, Walfredo Cirne, Francisco Brasileiro, and Paulo Roisenberg. *OurGrid: An Approach to Easily Assemble Grids with Equitable Resource Sharing*, pages 61–86. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[10] M. R. Miranda Assis and L. F. Bittencourt. An analysis of the voluntary aspect in cloud federations. In *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, pages 500–505, Dec 2015.

[11] M.R.M. Assis and L.F. Bittencourt. A survey on cloud federation architectures: Identifying functional and non-functional properties. *Journal of Network and Computer Applications*, 72:51 – 71, 2016.

[12] Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390 – 1396, 1981.

[13] Wayne E. Baker and Nathaniel Bulkley. Paying it forward vs. rewarding reputation: Mechanisms of generalized reciprocity. *Organization Science*, 25(5):1493–1510, 2014.

[14] Abmar Barros, Francisco Brasileiro, Giovanni Farias, Francisco Germano, Marcos Nóbrega, Ana Ribeiro, Igor Silva, and Letícia Teixeira. Using fogbow to federate private clouds. In *Anais do XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - Salão de Ferramentas*, Maio 2015.

[15] Stefano Battiston, Domenico Delli Gatti, Mauro Gallegati, Bruce C. Greenwald, and Joseph E. Stiglitz. Liaisons dangereuses: Increasing connectivity, risk sharing, and systemic risk. Working Paper 15611, National Bureau of Economic Research, January 2009.

[16] T. Bocek, F. V. Hecht, D. Hausheer, B. Stiller, and Y. El-khatib. Compactpsh: An efficient transitive tft incentive scheme for peer-to-peer networks. In *2009 IEEE 34th Conference on Local Computer Networks*, pages 483–490, Oct 2009.

[17] S.A. Boorman and P.R. Levitt. *The genetics of altruism*. Academic Press, 1980.

[18] F. Brasileiro, G. Silva, F. Araújo, M. Nóbrega, I. Silva, and G. Rocha. Fogbow: A middleware for the federation of iaas clouds. In *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 531–534, May 2016.

[19] Chiranjeeb Buragohain, Dvyakant Agrawal, and Subhash Suri. A game theoretic framework for incentives in p2p systems. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, pages 48–56, Linköping, Sweden, 09/2003 2003. IEEE Computer Society, IEEE Computer Society.

[20] M. Capotă, N. Andrade, T. Vinkó, F. Santos, J. Pouwelse, and D. Epema. Inter-swarm resource allocation in bittorrent communities. In *Peer-to-Peer Computing (P2P), 2011 IEEE International Conference on*, pages 300–309, Aug 2011.

[21] A. Celesti, F. Tusa, M. Villari, and A. Puliafito. How to enhance cloud architectures to enable cross-federation. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 337–345, July 2010.

[22] Chun-Ling Cheng, Xiao-Long Xu, and Bing-Zhen Gao. Metrust: A mutual evaluation-based trust model for p2p networks. *International Journal of Automation and Computing*, 9(1):63–71, 2012.

[23] Gianluca Ciccarelli and Renato Lo Cigno. Collusion in peer-to-peer systems. *Computer Networks*, 55(15):3517 – 3532, 2011.

[24] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[25] W. L. da Costa Cordeiro, F. R. Santos, G. H. Mauch, M. P. Barcelos, and L. P. Gaspary. Securing p2p systems from sybil attacks through adaptive identity management. In *2011 7th International Conference on Network and Service Management*, pages 1–6, Oct 2011.

[26] Weverton Luis da Costa Cordeiro, Flávio Roberto Santos, Marinho Pilla Barcellos, Luciano Paschoal Gaspary, Hanna Kavalionak, Alessio Guerrieri, and Alberto Montresor. Making puzzles green and useful for adaptive identity management in large-scale distributed systems. *Computer Networks*, 95:97 – 114, 2016.

[27] Eduardo de Lucena Falcão, Francisco Brasileiro, Andrey Brito, and José Luis Vivas. *Incentivising Resource Sharing in Federated Clouds*, pages 45–50. Springer International Publishing, Cham, 2015.

[28] Eduardo de Lucena Falcão, Francisco Brasileiro, Andrey Brito, and José Luis Vivas. Enhancing p2p cooperation through transitive indirect reciprocity. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, June 2016.

[29] Eduardo de Lucena Falcão, Francisco Brasileiro, Andrey Brito, and José Luis Vivas. Controlando a contenção de recursos para promover justiça em uma federação peer-to-peer de nuvens privadas. In *Anais do WCGA 2015, XIII Workshop em Clouds e Aplicações*, pages 84–97, 2015.

[30] Eduardo de Lucena Falcão, Francisco Brasileiro, Andrey Brito, and José Luis Vivas. Enhancing fairness in p2p cloud federations. *Computers & Electrical Engineering*, pages –, 2016.

[31] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Dover Publications, 2013.

[32] R. Eidenbenz, T. Locher, S. Schmid, and R. Wattenhofer. Boosting market liquidity of peer-to-peer systems through cyclic trading. In *2012 IEEE 12th International Conference on Peer-to-Peer Computing (P2P)*, pages 155–166, Sept 2012.

[33] Bin Fan, John C. S. Lui, and Dah-Ming Chiu. The design trade-offs of bittorrent-like file sharing protocols. *IEEE/ACM Trans. Netw.*, 17(2):365–376, April 2009.

[34] Michal Feldman and John Chuang. Overcoming free-riding behavior in peer-to-peer systems. *SIGecom Exch.*, 5(4):41–50, July 2005.

[35] Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. Robust incentive techniques for peer-to-peer networks. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC '04, pages 102–111, New York, NY, USA, 2004. ACM.

[36] M. Fishman. Indirect reciprocity among imperfect individuals. *Journal of Theoretical Biology*, 225(3):285–292, December 2003.

[37] Xavier Freixas, Bruno M Parigi, and Jean Rochet. Systemic risk, interbank relations, and liquidity provision by the central bank. *Journal of Money, Credit and Banking*, 32(3):611–38, 2000.

[38] DM Gale and F Allen. *Systemic Risk and Regulation*. Chicago University Press, 2006.

[39] Syeda ZarAfshan Goher, Peter Bloodsworth, Raihan Ur Rasool, and Richard McClatchey. Cloud provider capacity augmentation through automated resource bartering. *Future Generation Computer Systems*, 81:203 – 218, 2018.

[40] I. Goiri, J. Guitart, and J. Torres. Characterizing cloud federation for enhancing providers' profit. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 123–130, July 2010.

[41] Eduardo R. Gomes, Quoc Bao Vo, and Ryszard Kowalczyk. Pure exchange markets for resource sharing in federated clouds. *Concurrency and Computation: Practice and Experience*, 24(9):977–991, 2012.

[42] Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. The cost of a cloud: Research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, December 2008.

[43] Nikolay Grozev and Rajkumar Buyya. Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, 44(3):369–390, 2014.

[44] Minaxi Gupta, Paul Judge, and Mostafa Ammar. A reputation system for peer-to-peer networks. In *Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '03, pages 144–152, New York, NY, USA, 2003. ACM.

[45] Rohit Gupta, Varun Sekhri, and Arun K. Somani. Compup2p: An architecture for internet computing using peer-to-peer networks. *IEEE Trans. Parallel Distrib. Syst.*, 17(11):1306–1320, November 2006.

[46] Ruchir Gupta and Yatindra Nath Singh. Trust estimation in peer-to-peer network using BLUE. *CoRR*, abs/1304.1649, 2013.

[47] Fatima Lamia Haddi and Mahfoud Benchaïba. A survey of incentive mechanisms in static and mobile {P2P} systems. *Journal of Network and Computer Applications*, 58:108 – 118, 2015.

[48] MyungJoo Ham and Gul Agha. Ara: a robust audit to prevent free-riding in p2p networks. In *Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05)*, pages 125–132, Aug 2005.

[49] Mohammed Hawa, Loqman As-Sayid-Ahmad, and Loay D. Khalaf. On enhancing reputation management using peer-to-peer interaction history. *Peer-to-Peer Networking and Applications*, 6(1):101–113, 2013.

[50] Joseph L. Hellerstein, Yixin Diao, Sujay Parekh, and Dawn M. Tilbury. *Feedback Control of Computing Systems*. John Wiley & Sons, 2004.

[51] John M. Hinson and J. E. R. Staddon. Matching, maximizing, and hill-climbing. *Journal of the Experimental Analysis of Behavior*, 40(3):321–331, 1983.

[52] Johnson Iyilade, Adesola Aderounmu, and Matthew Adigun. Incentives for resource sharing and cooperation in grid computing system. In *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*, pages 191–198. IEEE, 2007.

[53] Adrian V. Jaeggi, Evelien De Groot, Jeroen M.G. Stevens, and Carel P. Van Schaik. Mechanisms of reciprocity in primates: testing for short-term contingency of grooming and food sharing in bonobos and chimpanzees. *Evolution and Human Behavior*, 34(2):69 – 77, 2013.

[54] C. Joe-Wong, Y. Im, K. Shin, and S. Ha. A performance analysis of incentive mechanisms for cooperative computing. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 108–117, June 2016.

[55] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[56] Gabor Kecskemeti, Attila Kertesz, Attila Marosi, and Peter Kacsuk. Interoperable resource management for establishing federated clouds. *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*, 2:18–35, 2012.

[57] Nobuhiro Kiyotaki and John Moore. Credit Chains. ESE Discussion Papers 118, Edinburgh School of Economics, University of Edinburgh, January 1997.

[58] Eleni Koutrouli and Aphrodite Tsalgatidou. Credible recommendation exchange mechanism for p2p reputation systems. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 1943–1948, New York, NY, USA, 2013. ACM.

[59] Chetan Kumar, Kemal Altinkemer, and Prabuddha De. A mechanism for pricing and resource allocation in peer-to-peer networks. *Electronic Commerce Research and Applications*, 10(1):26 – 37, 2011. Special Section: Service Innovation in E-Commerce.

[60] Subodha Kumar, Kaushik Dutta, and Vijay Mookerjee. Maximizing business value by optimal assignment of jobs to resources in grid computing. *European Journal of Operational Research*, 194(3):856 – 872, 2009.

[61] Kevin Lai, Lars Rasmusson, Eytan Adar, Li Zhang, and Bernardo A. Huberman. Tycoon: An implementation of a distributed, market-based resource allocation system. *Multiagent Grid Syst.*, 1(3):169–182, August 2005.

[62] R. Landa, D. Griffin, R. G. Clegg, E. Mykoniati, and M. Rio. A sybilproof indirect reciprocity mechanism for peer-to-peer networks. In *INFOCOM 2009, IEEE*, pages 343–351, April 2009.

[63] Dave Levin, Katrina LaCurts, Neil Spring, and Bobby Bhattacharjee. Bittorrent is an auction: analyzing and improving bittorrent's incentives. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 243–254. ACM, 2008.

[64] Xiaoyong Li, Feng Zhou, and Xudong Yang. A multi-dimensional trust evaluation model for large-scale {P2P} computing. *Journal of Parallel and Distributed Computing*, 71(6):837 – 847, 2011. Special Issue on Cloud Computing.

[65] Thomas Locher, Patrick Moor, Stefan Schmid, and Roger Wattenhofer. Free riding in bittorrent is cheap. In *Proc. Workshop on Hot Topics in Networks (HotNets)*, pages 85–90. Citeseer, 2006.

[66] P Marshall, K Keahey, and T. Freeman. Elastic site: Using clouds to elastically extend site resources. In *Conference on Cluster, Cloud, and Grid Computing (CCGRID)*, pages 43–52, May 2010.

[67] Daniel Sadoc Menasché, Laurent Massoulié, and Don Towsley. Reciprocity and barter in peer-to-peer systems. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 1505–1513, Piscataway, NJ, USA, 2010. IEEE Press.

[68] P. Merz, F. Kolter, and M. Priebe. Free-riding prevention in super-peer desktop grids. In *Computing in the Global Information Technology, 2008. ICCGI '08. The Third International Multi-Conference on*, pages 297–302, July 2008.

[69] M. Mihailescu and Y. M. Teo. Dynamic resource pricing on federated clouds. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, pages 513–517, May 2010.

[70] Animesh Nandi, Tsuen-Wan Ngan, Atul Singh, Peter Druschel, and Dan S. Wallach. Scrivener: Providing incentives in cooperative content distribution systems. In Gustavo Alonso, editor, *Middleware*, volume 3790 of *Lecture Notes in Computer Science*, pages 270–291. Springer, 2005.

[71] Jason Olejarz, Whan Ghang, and Martin A. Nowak. Indirect reciprocity with optional interactions and private information. *Games*, 6(4):438, 2015.

[72] Panos Parpas and Berç Rustem. A pricing mechanism for resource management in grid computing. *Computational Economics*, 31(4):381–395, 2008.

[73] Michael Piatek, Tomas Isdal, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. Do incentives build robustness in bittorrent. In *Proc. of NSDI*, volume 7, 2007.

[74] Michael Piatek, Tomas Isdal, Tom Anderson, Arvind Krishnamurthy, and Arun Venkataramani. Building bit-tyrant, a (more) strategic bittorrent client. *login*, 32(4):8–13, 2007.

[75] Anatol Rapoport and Albert M Chammah. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press, 1965.

[76] Josep Rius, Soraya Estrada, Fernando Cores, and Francesc Solsona. Incentive mechanism for scheduling jobs in a peer-to-peer computing system. *Simulation Modelling Practice and Theory*, 25:36 – 55, 2012.

[77] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Muñoz, and G. Tofetti. Reservoir - when one cloud is not enough. *Computer*, 44(3):44–51, March 2011.

[78] Hugo Sadok, Miguel Elias M. Campista, and Luís Henrique M. K. Costa. Um mecanismo para compartilhamento de recursos em nuvens colaborativas baseado na credibilidade dos usuários. In *Anais do XXXV SBRC 2017*, 2017.

[79] Robson Santos, Alisson Andrade, Walfredo Cirne, Francisco Brasileiro, and Nazareno Andrade. Relative autonomous accounting for peer-to-peer grids. *Concurrency and Computation: Practice and Experience*, 19(14):1937–1954, 2007.

[80] Alex Sherman, Jason Nieh, and Clifford Stein. Fairtorrent: bringing fairness to peer-to-peer systems. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 133–144. ACM, 2009.

[81] K. Shin, C. Joe-Wong, S. Ha, Y. Yi, I. Rhee, and D. Reeves. T-chain: A general incentive scheme for cooperative computing. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, pages 163–174, June 2015.

[82] Kyuyong Shin, D. S. Reeves, and Injong Rhee. Treat-before-trick : Free-riding prevention for bittorrent-like peer-to-peer networks. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–12, May 2009.

[83] Michael Sirivianos, Jong Han Park, Rex Chen, and Xiaowei Yang. Free-riding in bittorrent networks with the large view exploit. In *IPTPS*, 2007.

[84] Michael Sirivianos, Xiaowei Yang, and Stanislaw Jarecki. Robust and efficient incentives for cooperative content distribution. *IEEE/ACM Trans. Netw.*, 17(6):1766–1779, December 2009.

[85] Tiziano Squartini, Iman van Lelyveld, and Diego Garlaschelli. Early-warning signals of topological collapse in interbank networks. Papers 1302.2063, arXiv.org, February 2013.

[86] Nobuyuki Takahashi. The emergence of generalized exchange. *American Journal of Sociology*, 105(4):1105–1134, 2000.

[87] Karthik Tamilmani, Vinay Pai, and Alexander Mohr. Swift: A system with incentives for trading. In *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[88] Jian Wang, Ruimin Shen, Carsten Ullrich, Heng Luo, and Changyong Niu. Resisting free-riding behavior in bittorrent. *Future Generation Computer Systems*, 26(8):1285 – 1299, 2010.

[89] Juheng Zhang, Subhajyoti Bandyopadhyay, and Selwyn Piramuthu. Real option valuation on grid computing. *Decision Support Systems*, 46(1):333 – 343, 2008.

[90] Kan Zhang and Nick Antonopoulos. A novel bartering exchange ring based incentive mechanism for peer-to-peer systems. *Future Generation Computer Systems*, 29(1):361 – 369, 2013. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.

[91] H. Zhao and X. Li. Vectortrust: Trust vector aggregation scheme for trust management in peer-to-peer networks. In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pages 1–6, Aug 2009.

[92] Huanyu Zhao and Xiaolin Li. H-trust: A group trust management system for peer-to-peer desktop grid. *Journal of Computer Science and Technology*, 24(5):833–843, 2009.

[93] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems*, 18(4):460–473, April 2007.