UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM SISTEMA DE REPUTAÇÃO PARA INTERAÇÃO BASEADA EM SERVIÇOS

FERNANDO HENRIQUE FERRAREZI MOLINA

ORIENTADOR: PROF. DR. HÉLIO CRESTANA GUARDIA

São Carlos – SP

Fevereiro/2016

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM SISTEMA DE REPUTAÇÃO PARA INTERAÇÃO BASEADA EM SERVIÇOS

FERNANDO HENRIQUE FERRAREZI MOLINA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes de Computadores
Orientador: Prof. Dr. Hélio Crestana Guardia

São Carlos – SP

Fevereiro/2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

Molina, Fernando Henrique Ferrarezi
M722s Um sistema de reputação para interação baseada em serviços / Fernando Henrique Ferrarezi Molina. -- São Carlos : UFSCar, 2016.
125 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2016.

1. D2D. 2. Internet das coisas. 3. Sistemas de reputação. 4. Comunicação baseada em serviços. 5. P2P. I. Título.

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de dissertação de mestrado do candidato Fernando Henrique Ferrarezi Molina, realizada em 03/03/2016.

Prof. Dr. Hélio Crestana Guardia
(UFSCar)

Prof. Dr. Hermes Senger
(UFSCar)

Prof. Dr. Adriano Mauro Cansian
(UNESP)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Prof. Dr. Adriano Mauro Cansian e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Fernando Henrique Ferrarezi Molina.

Prof. Dr. Hélio Crestana Guardia
Presidente da Comissão Examinadora
(UFSCar)

Dedico este trabalho aos meus amigos e familiares.

AGRADECIMENTOS

Agradeço primeiramente aos meus familiares e amigos pelo apoio, especialmente à Flávia Molina. Agradeço também aos meus colegas de trabalho Igor Maldonado Floôr, Carla Rossetto, Maurício Lisboa Perez, Bruno da Silva Melo, Denis Wilson de Souza Oliveira e Danilo da Silva pelas suas contribuições e pelo apoio. Por último, e não menos importante, agradeço ao meu orientador Prof. Dr. Hélio Guardia pelos seus ensinamentos.

“Cada estação da vida é uma edição, que corrige a anterior, e que será corrigida também, até a edição definitiva, que o editor dá de graça aos vermes”

Memórias Póstumas de Brás Cubas - Machado de Assis

RESUMO

A comunicação por meio de serviços tem se tornado cada vez mais relevante, como em cenários da Internet das Coisas. A escolha entre serviços, contudo, pode ser uma tarefa difícil, já que essa forma de comunicação pode envolver recursos e dados sensíveis. Assim, é preciso um sistema de seleção de serviços que seja confiável, seguro e privado, garantindo que as expectativas dos utilizadores dos serviços sejam alcançadas. Sistemas de reputação podem ajudar nessa seleção. Assim, este trabalho apresenta um sistema de reputação distribuído para avaliação de interações baseadas em serviços utilizando conceitos da arquitetura de sistemas das cripto moedas. Para avaliar o desempenho na detecção de serviços de baixa qualidade utilizando o mecanismo proposto, apresenta-se também um estudo do comportamento de diferentes formulações para o cálculo de reputações. Resultados obtidos via simulação mostram que a arquitetura do sistema de reputação proposto é viável e permite uma ampla adoção com uma infraestrutura escalável. A análise do comportamento das formulações também permite concluir que o histórico recente do comportamento dos elementos avaliados é o fator que mais influencia na qualificação dos dispositivos.

Palavras-chave: D2D, Internet das Coisas, Sistemas de Reputação, Comunicação baseada em serviços, P2P

ABSTRACT

Service-based communication is becoming increasingly relevant, as seen in the Internet of Things scenarios. Choosing an adequate service however can be a difficult task, as this form of communication may involve sensible resources and data. A trustworthy, safe, and privacy preserving mechanism is desirable for service selection. Reputation systems can help in this selection. This thesis presents a reputation system for service selection based on the security and privacy mechanisms of crypto-coins. In order to evaluate the performance of the proposed mechanism in the identification of low quality service providers we also investigated the use of different formulations used to determine a reputation based on peer voting. Results from simulation show the proposed system architecture is viable and supports large adoption via a scalable infrastructure. The performance analysis of different formulations for the calculus of reputations also shows that the recent behavior of an entity is the most impacting factor in establishing its reputation.

Keywords: D2D, Internet of Things, Reputation System, Service-based communication, P2P

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1

INTRODUÇÃO

Diferentes tecnologias de transmissão sem fio oferecem capacidade de comunicação para dispositivos móveis. Interfaces WiFi, Bluetooth e 3/4G, por exemplo, estão presentes em grande parte dos *smartphones* e outros dispositivos computacionais frequentemente utilizados pelas pessoas.

Apesar de a conectividade à Internet provida por essas tecnologias possibilitar o acesso de qualquer dispositivo móvel a outros dispositivos conectados à rede, questões de endereçamento e mobilidade podem restringir a interação direta entre dispositivos (CARPENTER; BRIM, 2002). Além disso, alguns tipos de trocas de informação, como o acesso circunstancial a um recurso fisicamente próximo requer tecnologias e protocolos de comunicação que levem em consideração a localização do usuário.

Usando a tecnologia Bluetooth, por exemplo, é possível identificar dispositivos próximos e funcionalidades providas para interação com outros dispositivos. Um mecanismo de descoberta permite a identificação de dispositivos próximos, e perfis presentes nesses dispositivos indicam serviços que podem prover (BLUETOOTH SIG, 2014).

WiFi permite que diferentes protocolos de comunicação transmitam informações entre dispositivos conectados entre si ou ao mesmo ponto de acesso. Atuando no nível da camada de aplicação na arquitetura TCP/IP, protocolos como UPnP, Bonjour e Zeroconf utilizam mecanismos de transmissão em *broadcast* para a descoberta de dispositivos próximos e que possuem funcionalidades básicas pré-definidas. Assim, esses protocolos permitem, entre outros aspectos, identificar dispositivos capazes de realizar impressões, de tratar arquivos de áudio e outros tipos de mídia.

Aplicações específicas, executadas simultaneamente em dois dispositivos podem permitir a troca de informações entre eles. Observa-se, contudo, que este modelo de interação direta

entre dispositivos poderia ser considerado de maneira mais ampla, usando-se uma interface padronizada que permita diferentes trocas de informações. Modelos de definição de serviços, especificando protocolos e formatos de trocas de informações, podem ser usados para esse fim, como os serviços Web, que proveem uma interface que possibilita a interação direta entre dispositivos, definindo a interface e o formato de dados. O modelo REST, por exemplo, utiliza mensagens padronizadas do protocolo HTTP(S) para a solicitação de serviços entre dois computadores, com trocas de informações definidas pelos formatos JSON ou XML (FIELDING, 2000).

Utilizando requisições REST, a comunicação direta entre dispositivos próximos pode ser resumida em três fases: a descoberta de nós próximos, a identificação de serviços providos pelos nós identificados, e as comunicações realizadas a fim de utilizar um serviço selecionado.

Para qualquer serviço, a troca de informações entre dois nós pode ser considerada em função da transferência de algum objeto, ou de uma referência a um objeto, e de alguma ação que deve ser realizada, possivelmente sobre um objeto relacionado. Ações típicas podem ser estabelecidas sobre os diferentes tipos de objeto. Por exemplo, tipos MIME (FREED; BORENSTEIN, 1996) e extensões de nomes de arquivos podem definir conjuntos de ações padronizadas para esses objetos.

Diferentes mecanismos podem ser empregados para, dada a definição de serviços disponíveis e dos recursos envolvidos, identificar um serviço desejado e qual provedor desse serviço deve ser selecionado em local e circunstâncias específicos. Conceitos de ontologia são frequentemente usados na identificação dos serviços (ESPINOZA; MENA, 2007), utilizando associações entre palavras para buscar a descrição mais adequada ao caso. Havendo vários possíveis provedores para um serviço de interesse, a escolha do provedor entre os serviços encontrados pode ser feita com métricas definidas por parâmetros da qualidade dos serviços de comunicação (QoS) (AKINGBESOTE, 2013) (RAMACHER; MONCH, 2014). Considerando parâmetros associados aos serviços em si, sistemas de reputação (YAN; ZHANG; DENG, 2012) (KOUTROULI; TSALGATIDOU, 2013) (SU, 2010) podem ser usados nas suas seleções.

Parâmetros típicos de QoS podem avaliar aspectos técnicos como: taxas de erro, largura de banda, vazão de pacotes, variação da latência, etc. Com base nos valores é possível definir qual provedor oferece a melhor conexão com o consumidor. Contudo, esses parâmetros de QoS não indicam a qualidade do serviço oferecido, tornando a avaliação dos provedores incompleta.

Já sistemas de reputação podem utilizar avaliações de consumidores para classificar os provedores de serviço, definindo métricas e funções que transformam as avaliações realizadas pelos clientes em valores representativos das reputações. Essas funções podem variar

dependendo dos cenários em que o sistema de reputação está inserido e de quais atributos são relevantes na avaliação.

Diferentes sistemas de reputação foram propostos. Alguns utilizam arquitetura de sistema distribuída (NGUYêN; CAMP, 2008) (KAMVAR; SCHLOSSER; GARCIA-MOLINA, 2003) (KOUTROULI; TSALGATIDOU, 2013) (LIU, 2007) (SHEN; LI, 2014) e focam em classificar arquivos e dispositivos em redes ponto a ponto ou em redes de sensores. Eles permitem obter uma reputação parcial com base nos dispositivos *online*. Contudo, as avaliações dos nós *offline* são desconsideradas no momento da consulta. Assim, dependendo dos dispositivos que respondem a requisição de consulta da reputação, o valor parcial calculado pode não refletir um valor satisfatório.

Outras propostas de sistemas de reputação utilizam abordagens centralizadas, que permitem ter todos os dados e reputações em um único local, sendo tipicamente empregadas no comércio eletrônico (ACAMPORA; CASTIGLIONE; VITIELLO, 2014) (SATO, 2014). Essas propostas possuem implementação mais simples e com maior visão dos dados, porém estão mais suscetíveis a algumas formas de ataque, como a negação de serviço.

Um estudo de sistemas de reputação pode ser feito pela análise de três de suas características (HOFFMAN; ZAGE; NITA-ROTARU, 2009): a formulação, o cálculo e a disseminação. Cada uma delas caracteriza-se pelas escolhas arquiteturais e está sujeita a vulnerabilidades.

A formulação de um sistema de reputação é a especificação matemática e abstrata de como as informações disponíveis serão transformadas em uma métrica. Ela deve ser uma equação ou um algoritmo que resulte em valores precisos e representativos.

O cálculo é a implementação concreta da formulação. Apesar de parecer que o cálculo é uma consequência direta da formulação, nem sempre é possível converter a representação ideal em algo concreto de forma simples. Isso pode acontecer devido à incompatibilidade entre a arquitetura escolhida para o sistema de reputação, centralizada ou distribuída, e a formulação desejada.

Por fim, a disseminação está relacionada à forma como os dados são armazenados e transmitidos na rede, sujeita a dificuldades no alcance dos nós envolvidos e na segurança da informação. Ao usar avaliações de usuários, um sistema de reputação pode ser empregado na seleção de um provedor de serviços, facilitando a comparação da qualidade dos serviços disponíveis. Assim, este trabalho apresenta um sistema de reputação distribuído para avaliação de interações baseadas em serviços.

Com base em sistemas de reputação existentes, desenvolveu-se uma formulação

compatível com um cenário de interação direta dispositivo a dispositivo. O sistema de reputação desenvolvido procura criar mecanismos para prover um método único de avaliação de serviços com o qual seja possível computar todos os votos na geração de uma reputação global e manter a privacidade dos envolvidos.

A escolha da arquitetura distribuída foi feita por diversos motivos. Em primeiro lugar, evita gargalos pois não possui uma entidade central para onde todos os dados seriam redirecionados. Essa arquitetura também permite uma maior escalabilidade do sistema, uma vez que novos participantes assumem novas responsabilidades, aumentando o poder de processamento do sistema como um todo. Por fim, o modelo distribuído garante grande transparência nos dados utilizados, pois não existe empresa ou organização responsável e detentora das informações como no caso dos sistemas centralizados.

Tendo em vista a preocupação crescente com privacidade na Internet, o sistema foi planejado para que não seja possível relacionar o identificador de uma reputação com o seu detentor, a menos que seja a vontade do seu detentor divulgá-la. A associação entre marcas e reputações pode ser interessante para muitos provedores e, por isso, também não é restringida.

A validação do sistema proposto foi feita com simulações utilizando o simulador SUMO para representação da mobilidade dos dispositivos e o *framework* OMNET++ para a representação da comunicação entre eles. Procurou-se avaliar a escalabilidade do sistema em alguns casos específicos e também comprar algumas formulações para a validação do modelo, tanto em cenários ideais quanto em cenários de ataques.

De maneira geral, os resultados obtidos mostram a viabilidade do sistema de reputação proposto, tanto em sua escalabilidade como na compatibilidade com as formulações propostas na literatura. Outro aspecto observado foi a influência de certas características das formulações, como a relevância dada aos votos recente, na eficiência de detecções das interações de baixa qualidade.

O restante desta dissertação se organiza da seguinte forma: o capítulo 2 apresenta os conceitos preliminares dos sistemas de reputação, segurança e protocolos utilizados; o sistema de reputação proposto é detalhado no capítulo 3; os métodos de avaliação e os resultados são apresentados no capítulo 4 e no capítulo 5 a conclusão.

Capítulo 2

CRIPTOGRAFIA E REPUTAÇÃO

2.1 Avaliação e Seleção por Sistemas de Reputação

A seleção de um serviço, seja ele disponibilizado na internet, em uma rede local ou por interação direta, se torna cada vez mais importante conforme esse método de interação se torna popular. A escolha de um provedor de serviços é usualmente feita com as métricas estabelecidas de QoS (AKINGBESOTE, 2013) (RAMACHER; MONCH, 2014), usando avaliações técnicas da qualidade da rede para a classificação dos serviços.

Contudo, essa abordagem não consegue avaliar a qualidade do serviço prestado e se as respostas, ou as funcionalidades oferecidas, estão de acordo com a expectativa do utilizador. Por esse motivo, a escolha baseada apenas em parâmetros de QoS é ineficaz em cenários em que são disponibilizados dados sensíveis aos serviços.

Para estes casos, é útil o uso de sistemas externos para a classificação dos provedores. Sistemas de reputação desempenham essa função, pontuando seus objetos de avaliação por meio do histórico de interações. No caso de serviços, votos podem ser enviados por seus consumidores e, com a utilização de uma formulação, gerar um valor de reputação. Esse valor deve representar a expectativa de qualidade de serviço para novas interações.

Sistemas de reputação são amplamente usados em aplicações atuais, sendo mais visíveis em segmentos como o comércio eletrônico (SATO, 2014) (ACAMPORA; CASTIGLIONE; VITIELLO, 2014). De modo geral, compradores avaliam e recomendam os vendedores e estas avaliações geram pontos de classificação para as lojas virtuais. Conforme a reputação de um vendedor aumenta, este é melhor classificado dentre as categorias do sistema e maiores são as chances de obterem novos pedidos. Esses mecanismos diminuem o risco de compras fraudulentas e incentivam os comerciantes a buscarem boas reputações.

Sistemas de reputação também têm sido empregados em sistema ponto a ponto, comumente utilizados para transferência de arquivos (NGUYêN; CAMP, 2008) (KAMVAR; SCHLOSSER; GARCIA-MOLINA, 2003) (KOUTROULI; TSALGATIDOU, 2013) (LIU, 2007). A implementação desses sistemas se torna um pouco mais complexa pelo seu caráter distribuído, pois é preciso levar em consideração como serão armazenadas as reputações e como serão avaliadas as recomendações de outros participantes da rede.

Um dos aspectos mais relevantes em um sistema de reputação distribuído é a convergência de dados. Os algoritmos se tornam muito mais complexos nesses sistemas pois é preciso atualizar as informações em um número maior de dispositivos e garantir que as divergências entres eles não sejam relevantes. Além disso, o controle das identidades e da autenticidade dos dados pode aumentar ainda mais a complexidade do sistema.

Diferentes propostas foram feitas para comparar e avaliar sistemas de reputação (HOFFMAN; ZAGE; NITA-ROTARU, 2009) (CHAOKAI; MENG, 2010) (CELESTINI; De Nicola; TIEZZI, 2013) (CHOO; JIANG; YU, 2014). Segundo Hoffman et al., pode-se analisar esses sistemas por três aspectos: a formulação, o cálculo e a disseminação de votos. O primeiro determina os fundamentos matemáticos ideais do sistema; o cálculo é a forma concreta de como os dados obtidos são transformados na reputação do objeto; e a disseminação está relacionada à forma como esses valores são transmitidos e armazenados.

A transformação de um conjunto de votos em reputação pode ser feita de diversas maneiras. Elas dependem de como o votante poderá representar sua satisfação com o objeto avaliado, das variáveis analisadas e de como a reputação será representada.

Os votos podem ser representados por classes (ABDUL-RAHMAN; HAILES, 2000) (ACAMPORA; CASTIGLIONE; VITIELLO, 2014), mas essa abordagem é pouco utilizada, em favor de representações numéricas das avaliações. Essa escolha irá influenciar em como a satisfação do votante será transformada em voto. Ao se deparar com classificações abstratas como: bom, médio e ruim, o votante precisa encaixar seu sentimento de satisfação nas classes apresentadas. Já em escalas numéricas ele terá que transformá-la em algo quantitativo. Assim, é comum haver uma transformação da representação do voto em um valor numérico intermediário que será utilizado para o cálculo da reputação; isso é feito para que o valor do voto possa ser de maior compreensão para o votante. Essa transformação pode ocorrer mesmo em escalas numéricas, pois é mais fácil para o votante representar sua satisfação em escalas habituais do seu cotidiano, como no intervalo de 0 a 10, do que nas escalas usadas nos sistemas de reputação, usualmente [0,1] e [-1,1].

Os sistemas de reputação também podem avaliar outros aspectos além da satisfação geral do

votante em relação a um serviço, ou podem ainda questioná-lo sobre eventos específicos. Essas abordagens tornam a reputação mais específica ao cenário abordado, porém mais complexa, pois é preciso ponderar cada um dos aspectos analisados.

Por exemplo, o sistema de reputação TruBeRepec (YAN; ZHANG; DENG, 2012) coleta dados de frequência e de tempo de uso de aplicativos, além das opiniões dos usuários. O cálculo da reputação, contudo, é complexo, pois precisa levar em consideração qual o peso desses valores para representar de forma mais real possível a reputação esperada para cada aplicativo. Neste modelo, os votos dos usuários têm pouca relevância na reputação dos aplicativos por não haver controle dos votos, possibilitando fraudes. Porém, mecanismos de controle de votos poderiam ajudar contra as fraudes, tornando o voto mais relevante e a representação da reputação mais adequada.

O modo como é feita a seleção de um objeto tendo como base suas reputações também é uma questão bastante discutida. No caso da seleção feita por pessoas, é preciso que a reputação seja apresentada de forma compreensível e de fácil comparação. Para isso, podem ser necessárias conversões ou classificações usando processos semelhantes aos discutidos na representação dos votos. Já a seleção automática deve ser de tal forma que não exclua novos candidatos honestos, porém que ainda não possuem boa reputação. Isso causaria a impossibilidade de obterem uma reputação equivalente ao serviço prestado. Assim, escolher sempre o serviço com maior reputação pode prejudicar novos provedores de serviço de qualidade efetivamente igual ou superior.

Diferentes algoritmos para o cálculo da reputação têm sido propostos. Alguns sistemas utilizam funções simples, como a média dos valores dos votos para calcular uma reputação. O problema deste tipo de formulação, contudo, é que votos falsos, como em ataques de autopromoção, podem afetar a média de forma significativa. Para evitar estes tipos de ataques, votos nos quais o módulo da diferença do seu valor e da média sejam maiores que um determinado limiar são retirados do cálculo da reputação. Caso isto ocorra, a média é refeita e o valor resultante é usado como a reputação do objeto (KAMVAR; SCHLOSSER; GARCIA-MOLINA, 2003)].

Outra abordagem é o uso de técnicas de inferência para classificar os objetos de avaliação. Métodos da lógica *Fuzzy*, por exemplo, podem ser empregadas no cálculo da reputação. Ao estabelecer variáveis e suas funções de representação é possível utilizar métodos como o de Mandani para extrair uma reputação (ACAMPORA; CASTIGLIONE; VITIELLO, 2014).

Abordagens Bayesianas também se destacam nos sistemas de reputação. Usualmente, duas abordagens são utilizadas. Em uma delas utiliza-se uma distribuição estatística para

representar a probabilidade do comportamento esperado (BOUDEC; BUCHEGGER, 2004) (DENKO; SUN; WOUNGANG, 2011), enquanto na segunda, utiliza-se um conjunto de informações coletadas em interações anteriores, como uma tabela de dados, para criar uma rede Bayesiana e tentar inferir qual a reputação do objeto avaliado (NGUYêN; CAMP, 2008).

No caso das reputações por distribuição estatística, é necessário escolher uma que represente de forma satisfatória o cenário no qual o sistema está inserido. Assim, a esperança dessa distribuição pode ser interpretada como a probabilidade do objeto analisado ter um determinado comportamento ou atributo. A distribuição Beta, por exemplo, muito utilizada para representações de eventos binários, é definida pela equação 2.1:

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{2.1}$$

Sendo p a probabilidade do evento ocorrer e

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \tag{2.2}$$

A esperança desta função é dada pela equação 2.3.

$$E(p) = \frac{\alpha}{\alpha+\beta} \tag{2.3}$$

Assim, se considerarmos α como o número de votos positivos com acréscimo de uma unidade e β como o número de votos negativos também com acréscimo de um, podemos calcular a esperança e interpretá-la como a probabilidade de um objeto ter um comportamento positivo.

Outro método que pode ser usado em avaliações é a rede Bayesiana. Neste modelo são utilizadas variáveis aleatórias, que são representações de características analisadas, para inferir a qualidade ou a categoria de um determinado objeto avaliado. Primeiro, cria-se uma árvore em que a raiz é o ponto inicial do teste e cada aresta é representada por variáveis aleatórias que serão testadas neste vértice, a aresta com maior probabilidade será usada e indicará o próximo vértice utilizado. A classificação da um objeto é feita percorrendo-se a árvore até encontrar um nó folha, que contém a classe do objeto avaliado.

Em (NGUYêN; CAMP, 2008), utiliza-se um conceito de contexto para definir as variáveis aleatórias da rede Bayesiana. Cada voto está associado com um contexto que deve conter ao menos os identificadores do consumidor, do provedor e do serviço e a data da interação.

Quaisquer outros valores podem ser associados ao voto de forma que a rede Bayesiana se torne a mais relevante possível. Assim, cada participante do sistema possui um conjunto de votos associados com seus respectivos contextos, que possuem as variáveis aleatórias que o participante julga pertinentes. Por fim, quando é necessário utilizar um serviço, as regras extraídas da rede são aplicadas ao contexto atual para tentar prever o comportamento dos serviços disponíveis.

Outro aspecto relevante de um sistema de reputação é o cálculo. Ele é a implementação concreta dos métodos idealizados na formulação. Mesmo parecendo ser um resultado direto, algumas questões são intrínsecas à sua definição. Mudanças podem ser necessárias para que seja possível implementar a formulação na arquitetura desejada, centralizada ou distribuída, ou ainda para prevenir determinados ataques contra o sistema, relacionados a estes modelos arquiteturais.

Em arquiteturas centralizas, a implementação da formulação é mais direta, pois todos os dados estão concentrados em um único local. O controle da criação de identificadores, nesse caso, também se torna mais fácil, pois pode-se atribuir esta função à mesma entidade responsável pelas reputações. Em um sistema centralizado também são necessários alguns cuidados para que ele não se torne um ponto frágil, por exemplo, alvo de ataques de negação de serviço (*DoS*).

Na arquitetura distribuída, a conversão da formulação no cálculo pode se tornar mais complexa pois é preciso definir como serão usados os dados que cada um dos participantes possui (KAMVAR; SCHLOSSER; GARCIA-MOLINA, 2003) (SINGH, 2003) (NGUYêN; CAMP, 2008) ou ainda estruturar o sistema utilizando *Chords* e *DHTs* (LIU, 2007) (KOUTROULI; TSALGATIDOU, 2013) (CHEN, 2014).

Outra questão importante para o cálculo é o significado da reputação. Esta pode representar um valor global, ou seja, o valor da reputação de um determinado objeto é igual para todos os participantes do sistema, ou local, em que cada participante calcula sua própria visão da reputação dos objetos avaliados. Normalmente, o primeiro método é usado em sistemas de reputação centralizados e o segundo nos distribuídos.

Reputações locais são usadas quando não há uma entidade responsável por calcular as reputações; ou quando a escolha de uma entidade com esta função pode comprometer a segurança e a confiabilidade dos dados; ou ainda quando se dá preferência a uma personalização da reputação. Esta última opção tenta inferir uma reputação baseando-se em votos de participantes que possuem escolhas parecidas.

Nas MANETs (SHEN; LI, 2014) e outras redes *Ad-Hoc*, por exemplo, as reputações normalmente são locais, cada participante calcula a reputação dos demais integrantes da rede, podendo para isso consultar outros participantes. Isso é necessário pois este modelo de comunicação se dá entre poucos participantes e seus integrantes se alteram constantemente. Assim, responsabilizar um único dispositivo para calcular ou armazenar a reputação é um risco à segurança do sistema.

Apesar de todos os esforços ainda não foram propostos sistemas de reputação que unam as características de privacidade e transparência dos sistemas distribuídos com a confiabilidade de um valor global de reputação. A dificuldade de criar um sistema distribuído que calcule uma reputação global de forma eficiente e segura é a principal limitação das propostas existentes. Observa-se, contudo, que as cripto moedas tratam um problema parecido de forma satisfatória.

2.2 Cripto Moedas

As cripto moedas são sistemas monetários virtuais e distribuídos que utilizam a criptografia como base da autenticidade e da segurança de seus protocolos (PECK, 2012). A primeira cripto moeda criada foi o Bitcoin (NAKAMOTO, 2008) e também a moeda virtual mais utilizada.

A principal motivação para sua criação foi a falta de um sistema em que se pudesse transferir valores monetários sem o intermédio de outras instituições e que isso fosse feito de forma anônima, assim como compras e vendas em espécie. Para que isso fosse possível, as identificações dos usuários da rede não poderiam estar associadas aos dados pessoais dos seus representantes. Para tanto, os algoritmos ECDSA (BROWN, 2010) geram um par de chaves pública-privada para cada identidade utilizando números aleatórios. Como a execução desses algoritmos não necessita de uma entidade centralizada para verificação, permitindo que cada participante possua quantas identidades forem necessárias para preservar seus dados pessoais, eles foram adotados como forma padrão de criação de identidades das cripto moedas.

As cripto moedas são sistemas totalmente distribuídos, de forma que não existe entidade única responsável por nenhuma tarefa e todos os dados são acessíveis por todos os participantes. Por isso, algumas questões devem ser levantadas: como restringir a transferência de valores somente pelos seus respectivos donos? Como impedir que o mesmo valor seja transferido para duas contas diferentes? Como contabilizar e atualizar o balanço de cada conta para todos os participantes? No Bitcoin, estas questões são solucionadas pelos mecanismos de transferências e consolidação de valores.

Uma transação de valores é feita quando um determinado identificador encaminha ao

sistema uma mensagem contendo o valor a ser transferido a um destinatário. Parte desta mensagem é cifrada utilizando a chave privada do remetente, garantindo que somente ele possa transferir seus valores.

Alguns dispositivos no sistema, chamados mineradores, são responsáveis por coletar essas transações e agrupá-las em blocos. Os blocos são as unidades de histórico e existe um desafio computacional para criá-los. Esse desafio é encontrar um número que adicionado aos dados do blocos e aplicada uma função *hash*, obtenha-se como resultado um número com uma determinada quantidade de zeros em seu início. Assim, uma transação só se torna válida ao ser inserida em um bloco e este ser distribuído aos demais participantes. Esse modelo de comunicação dificulta a alteração dos dados do histórico e, com um histórico bem definido, torna-se inviável gastar duas vezes o mesmo recurso financeiro ou negar transações efetuadas (SINGH, 2013).

O agrupamento de votos em um bloco é feito por meio de uma estrutura de dados chamada árvore Merkle (MERKLE, 1982). Essa estrutura é muito utilizada na autenticação de redes *Ad-Hoc* (SANTHANAM; XIE; AGRAWAL, 2008) (LI, 2014) e na validação de integridade de memória (SZEFER; BIEDERMANN, 2014), devido à dificuldade de alterar os dados da estrutura indevidamente e à eficácia na confirmação de seus componentes.

Em uma definição formal, uma árvore Merkle é uma árvore em que cada nó folha é uma *hash* de um bloco de dados e os nós pais são *hashes* dos seus nós filhos, sempre agrupando dois nós filhos para cada nó pai, como pode ser visto na figura 2.1 (O'REILLY MEDIA, 2013). No caso em que o número de nós de uma determinada camada é impar, um dos nós pai terá apenas um nó filho e utilizará duas vezes este valor na geração de sua *hash*.

Assim, somente com a *hash* da raiz da árvore é possível obter uma árvore Merkle, verificar se essa árvore é verdadeira e se um conjunto específico de dados foi usado em sua criação. Por essas características, ela é utilizada no bloco para referenciar as transações representadas nele, garantindo que uma vez criado o bloco com o histórico de suas transações, essas não poderão ser alteradas.

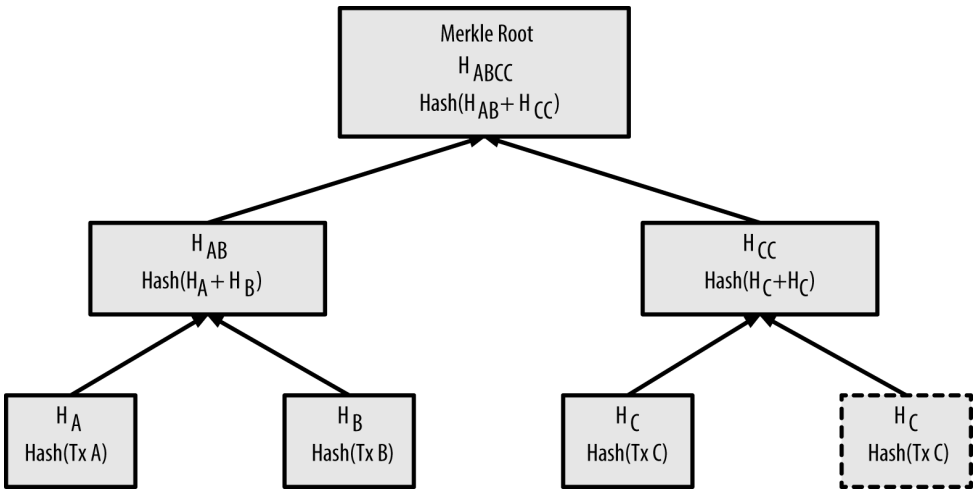

Figura 2.1: Exemplo de árvore Merkle (O'REILLY MEDIA, 2013)

.

Quando um dispositivo cria um bloco este precisa ser repassado aos demais participantes da rede. Isso é feito propagando o bloco aos dispositivos de que se tem conhecimento. Assim, ao criar um bloco, um dispositivo anuncia este aos pares conectados. Os dispositivos, que recebem a notificação, requisitam o novo bloco e anunciam aos dispositivos com os quais possuí conexão. Dessa forma, o bloco será propagado aos dispositivos na rede até que todos eles tomem conhecimento dos dados. Estudos apontam que utilizando este método para nós conectados à Internet, noventa e cinco porcento deles estão atualizados após quarenta segundos (DECKER; WATTENHOFER, 2013).

Nesse cenário, é comum ocorrem bifurcações em alguns momentos. Elas acontecem quando dois participantes criam um bloco ao mesmo tempo e concorrem para saber qual deles irá se tornar o bloco definitivo do histórico de transações recentes. Essa definição ocorre quando um terceiro bloco é criado e disseminado. O bloco que este terceiro participante escolheu como sendo seu antecessor será o bloco que permanecerá, pois os demais participantes irão aceitar o bloco mais recente como o bloco mais confiável e descartarão os blocos conflitantes.

Para o funcionamento da rede de uma cripto moeda ainda são necessários dispositivos exercendo outras funcionalidades. A primeira é a de servidor de registro, que atua como um super nó na rede, informando os endereços IP de outros dispositivos *online* para quem deseja se conectar à rede. Outra função importante é a de verificadores completos, que são dispositivos responsáveis por armazenar o histórico e verificar a validade de blocos e de transações.

A complexidade necessária para o funcionamento de uma cripto moeda é justificável pela

sua segurança e privacidade. As transações possuem certa anonimidade, tanto para a pessoa que envia as cripto moedas quanto para quem as recebe, as transações são processadas e armazenadas por dispositivos aleatórios, além de garantir baixo risco de fraudes mesmo não tendo vínculo com entidades regulamentadoras. Essas características são desejáveis para o sistema idealizado.

Assim, se propõe uso dos mecanismo de armazenamento e contabilização dos saldos das contas das cripto moedas em um sistema de reputação, tornando possível ter uma contabilização global da reputação aliado a sistemas de segurança e privacidade satisfatórios.

Para a compreensão do sistema proposto ainda são necessários os conceitos de Criptografia de Curvas Elípticas e de Autoridades Certificadoras Distribuídas que serão abordados nas próximas seções.

2.3 Criptografia de Curvas Elípticas

A segurança das cripto moedas está diretamente ligada aos certificados emitidos nas mensagens de transação de montantes. Esses certificados são gerados pelos algoritmos ECDSA que utilizam o formato de equações apresentado em 2.4 para gerar pares de chaves pública-privada de forma pseudo aleatória, sendo k o domínio no qual a equação está definida.

$$y^2 = x^3 + ax + b, \qquad a, b \in k \tag{2.4}$$

Para entender o funcionamento deste esquema de criptografia primeiro é preciso definir alguns outros conceitos importantes. Um domínio finito, denominado $GF(k)$, é um domínio de ordem finita, ou seja, o número de elementos desse domínio é determinado pelo valor de k, que deve ser sempre um número primo ou uma potência de um número primo. Os elementos desse conjunto são $0, 1, 2, ..., k-1$ e $a = b$ equivale à $a \equiv b(modk)$.

A definição da *característica* de um domínio p é dada pela quantidade de vezes em que a identidade de multiplicação é somada para obter-se o resultado 0. Por exemplo se o domínio de uma função é o conjunto de número reais, sua identidade de multiplicação é o valor 1, então a característica dele somente pode ser zero, pois não importa quantas vezes a identidade de multiplicação seja somada, o resultado sempre será um número diferente de zero.

Por outro lado, caso o domínio da função seja finito, $GF(k)$, ao somar identidade de multiplicação k vezes o resultado será zero. O valor da característica de um domínio sempre será um número primo.

Com esses conceitos podemos então definir uma curva elíptica, E, com característica de domínio diferente de dois e diferente de três como sendo o conjunto de pontos que satisfaz a equação 2.5 dentro do domínio k mais o "ponto no infinito", aqui representado por O . A restrição da característica de domínio é feita pois a função que as representa não pode ser reduzida à equação 2.4. No caso de domínios com característica igual a dois, a equação pode ser reduzida até 2.6 e igual a três até 2.7.

$$E = \left\{(x,y) : y^2 = x^3 + ax + b\right\} \cup \{O\} \tag{2.5}$$

$$y^2 + ay = x^3 + bx^2 + cxy + dx + e, \qquad a,b,c,d,e \in k \tag{2.6}$$

$$y^2 = x^3 + ax^2 + bx + c, \qquad a,b,c \in k \tag{2.7}$$

Também é necessário que a equação 2.5 não possua múltiplas raízes, ou seja, seu descriminante deve ser diferente de zero, como mostra a equação 2.8.

$$\Delta = 4a^3 + 27b^2 \neq 0 \tag{2.8}$$

Definida a equação da curva elíptica podemos então definir operações sobre seu domínio. Assim, a negativa de um ponto $P \in E_k$ é o outro ponto pertencente ao domínio que possui a mesma coordenada no eixo X.

A soma de dois pontos P_1 e P_2 não opostos é definida como a negativa do terceiro ponto de intersecção da curva elíptica com a reta formada por estes pontos ou pela tangente, caso $P_1 = P_2$. No caso de pontos opostos, sua soma é igual ao "ponto no infinito", isso se dá porque a reta formada por estes pontos é paralela ao eixo Y e não cruza com a curva elíptica uma terceira vez. Um exemplo pode ser visto na figura 2.2 (CERTICOM, 2015).

Neste exemplo, a reta formada pelos pontos P e Q passa também pelo ponto $-R$. Negando o terceiro ponto encontrado, obtêm-se R, o resultado da soma de P e Q.

Algebricamente podemos definir a soma de dois pontos não opostos $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ em $P_3 = (x_3, y_3)$ pelas equações 2.9 e 2.10.

$$x_3 = \lambda^2 - x_1 - x_2 \tag{2.9}$$

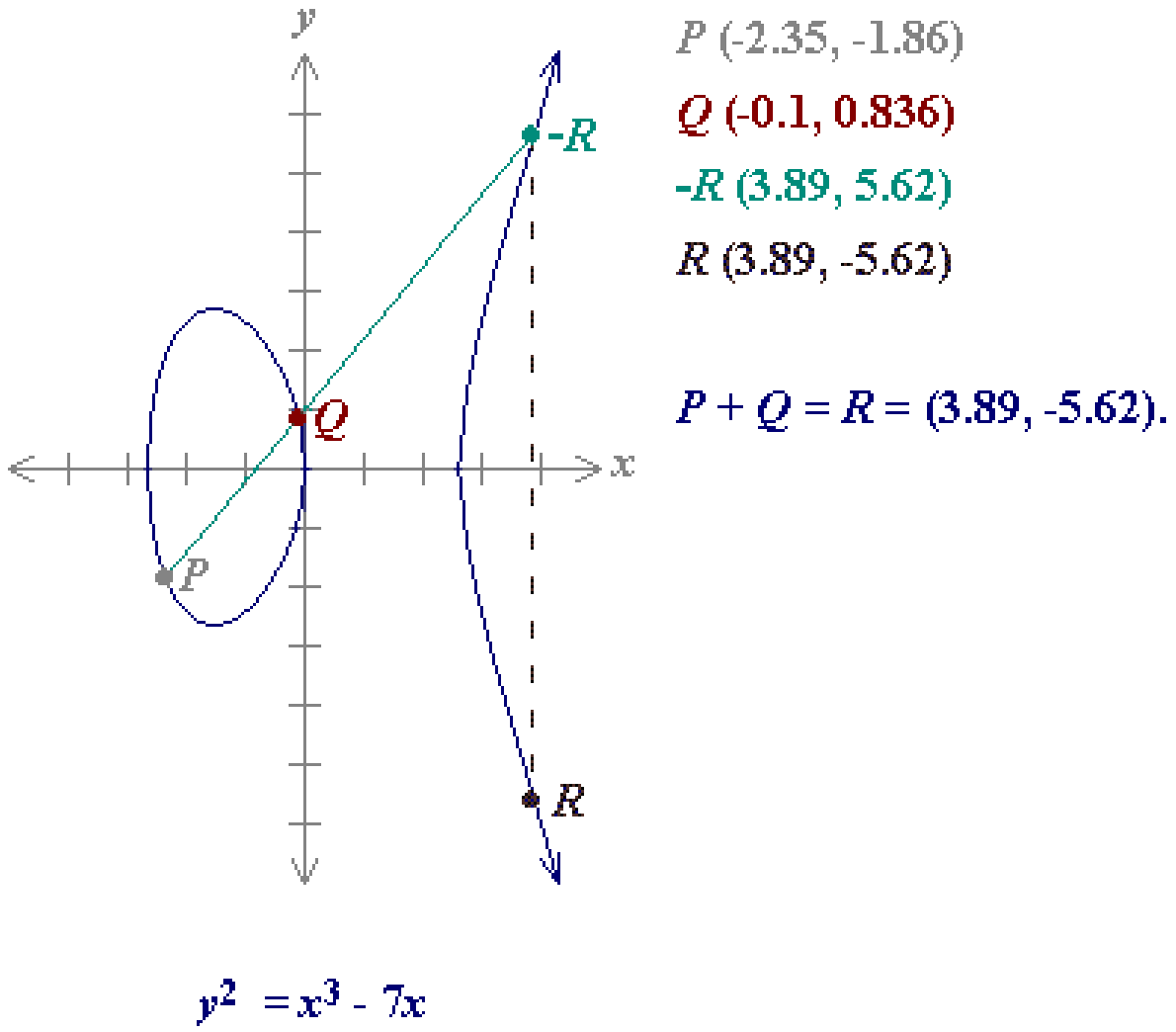

Figura 2.2: Exemplo de soma dos pontos P e Q no domínio de uma curva elíptica. (CERTICOM, 2015)

$$y_3 = \lambda(x_3 - x_1) + y_1 \tag{2.10}$$

sendo

$$\lambda = \begin{cases} \frac{3x_1^2 + a}{2y_1} & \text{se } x_1 = x_2 \\ \frac{y_1 - y_2}{x_1 - x_2} & \text{se } x_1 \neq x_2 \end{cases} \tag{2.11}$$

Os modelos de criptografias que utilizam curvas elípticas são definidos em domínios finitos, normalmente potências de 2, sendo representandos como $\mathbb{F}_{2^m}$, e se utilizam da dificuldade necessária para resolver o problema logarítmico discreto de curvas elípticas (ECDLP, sigla em inglês). Este problema é definido por dado os pontos P e Q, sendo Q

múltiplo de P, encontrar n, tal que $nP = Q$. A melhor solução encontrada até hoje para este problema tem solução $O(\sqrt{k})$, sendo $k = 2^m$. Portanto, são necessárias em média $2^{m/2}$ iterações para resolvê-lo.

Para que seja possível utilizar um esquema de criptografia de curvas elípticas é necessário definir uma série de parâmetros que devem ser conhecidos por todos os participantes que o utilizarão. Eles são definidos pela tupla $T = \{p, a, b, G, q, h, H(x)\}$ sendo: p um número primo que define o tamanho do domínio finito $\mathbb{F}_p$ no qual está definida a curva elíptica; a e b os parâmetros da curva elíptica; G o ponto base; q o tamanho do domínio finito representado computacionalmente, $\mathbb{F}_q = \mathbb{F}_{2^m}$; h o cofator da função definido por $h = \frac{|E(\mathbb{F}_p)|}{n}$ sendo n a ordem do ponto G e primo; e $H(x)$ uma função *hash*. Por questões de segurança algumas restrições são feitas aos valores de n e de h. O valor de n deve ser maior que 2^{160} e maior que $4\sqrt{q}$ e o valor de h menor que 4, preferencialmente 1. Definidos e divulgados os parâmetros o sistema está apto a entrar em funcionamento.

Este modelo de criptográfica é utilizado pelas cripto moedas para definir os identificadores dos usuários e para gerar as assinaturas dos dados transmitidos. Este proposta também utilizou esta este modelo com os mesmos fins.

2.4 Autoridades Certificadoras Distribuídas

Autoridades certificadoras Distribuídas são utilizadas quando deseja-se que um grupo de dispositivos, não necessariamente todos confiáveis, exerça a função de criação de certificados. A necessidade de múltiplas assinaturas para comprovar um certificados aumenta o grau de segurança do sistema de autenticação em ambientes distribuídos. Outra vantagem é a redundância na verificação da informação certificada, pois é feita por um número mínimo de dispositivos.

As abordagens distribuídas para autoridades certificadores são inspiradas na proposta de compartilhamento de chaves feita por Shamir (BLöMER, 2011). Nele, um "*dealer*", dispositivo que inicializa o sistema, gera um segredo d e o compartilha com outros dispositivos utilizando um polinômio $f(x)$ de grau $t-1$, equação 2.12. Neste polinômio, $a_0 = d$ e $a_1, a_2, \ldots, a_{t-1}$ são números aleatórios.

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \tag{2.12}$$

Assim, para cada dispositivo i com identificador ID_i, calcula-se $d_i = f(ID_i)$ e envia-se esse

valor ao dispositivo i por uma transmissão segura. Desta forma é possível recalcular o valor d pelo método de interpolação de Lagrange tendo t chaves d_i, equação 2.13. Com qualquer número inferior a t de chaves não é possível deduzir o valor de d.

$$f(x) = \sum_{i=1}^{t} d_i \prod_{j=1, j\neq i}^{t} \frac{x - ID_j}{ID_i - ID_j} \tag{2.13}$$

Um problema desse mecanismo é a necessidade de utilizar um "*dealer*"para inicializar o sistema. Ao conhecer o segredo ele torna-se um único ponto de falha e basta comprometê-lo para comprometer o sistema. Por isso propostas recentes tentam evitar sua utilização.

Zhengfeng et al. aplicaram este método ao mecanismo de certificação evitando o uso de um "*dealer*"(ZHENGFENG; JIANGHONG; DONGHUI, 2008). Eles assumem que n dispositivos fazem parte de uma autoridade certificadora (CA) e suas chaves privadas são d_i e suas chaves públicas $y_i = g^{d_i} \mod p$, em que p e q são dois número primos sendo $p = mq + 1$ e $g = h^{(p-1)/q}$ para h e m aleatórios.

Durante a inicialização do sistema cada dispositivo i pertencente à autoridade certificadora cria um polinômio $f_i(x)$ como na equação 2.12 e gera um valor $d_{i,j} = f_i(ID_j)$ para cada um dos demais integrantes. Esse valores devem ser encaminhados aos respectivos dispositivos com identificação ID_j que pertencem à CA para que eles possam calcular seu segredo $F(ID_j)$ pela equação 2.14.

$$F(ID_j) = \sum_{i=1}^{n} f_i(ID_j) \mod p \tag{2.14}$$

Cada um dos integrantes da CA deve divulgar então sua chave pública $y_j = g^{F(ID_j)} \mod p$ para que todos possam calcular a chave pública da CA pela equação 2.15, lembrando que $t-1$ é o grau do polinômio gerado pelos dispositivos.

$$y = g^{\sum_{i=1}^{t} F(ID_i) \prod_{j=1, j\neq i}^{t} \frac{-ID_j}{ID_i - ID_j} \mod p} \mod p \tag{2.15}$$

A criação do certificado se dá pelos seguintes passos: (1) O usuário A cria um *hash* da informação certificada $C = H(ID_a, y_a, \text{outras informações})$; (2) encaminha este valor para t participantes da CA; (3) cada membro da CA escolhe um número aleatório k_i, calcula $r_i = g^{k_i}$ e encaminha r_i aos demais membros; (4) cada membro calcula uma assinatura parcial (C, r, s_i) pelas equações 2.16 e 2.17 e a encaminha ao usuário; (5) o usuário A pode então calcular sua

assinatura pela equação 2.18 obtendo (C, r, s). A verificação pode ser feita pela equação 2.19.

$$r = \prod_{i=1}^{t} r_i \tag{2.16}$$

$$s_i = k_i C + rF(ID_i) \prod_{j=1, j\neq i}^{t} \frac{-ID_j}{ID_i - ID_j} \mod p \tag{2.17}$$

$$s = \sum_{i=1}^{t} s_i \mod p \tag{2.18}$$

$$g^s = r^C y^r \tag{2.19}$$

Uma proposta parecida foi aplicada a sistemas de criptografia de curvas elípticas por (FOURNARIS, 2011). Dessa maneira, pode-se criar um sistema de autoridades certificadoras distribuídas para autenticar os identificadores de cada participante do sistema de reputação. Esta proposta será melhor detalhada no próximo capítulo.

Capítulo 3

BitTrust- Sistema de Reputação para Serviços

3.1 Sistema de Reputação

A escolha de um serviço a ser utilizado é um fator crítico na interação entre dispositivos, tornando-se ainda mais importante em cenários em que o consumidor não possui conhecimento prévio a respeito dos provedores. Isso porque algumas interações exigem troca de informações privadas. Além disso, sistemas também podem ser prejudicados por dados imprecisos ou incorretos provenientes de serviços não confiáveis. Sistemas de reputação podem auxiliar nesta escolha.

Sistemas de reputação qualificam objetos de avaliação utilizando votos de participantes que interagem com estes objetos. No caso de um sistema de reputação para serviços, tanto os consumidores quanto os provedores de serviço devem ser objetos de avaliação. Para tanto, votos devem ser emitidos após cada interação e computados para alterar, de forma positiva ou negativa, as reputações de provedores e utilizadores de serviços.

Independentemente da forma como são implementados, os protocolos de comunicação entre consumidores e provedores seguem algumas etapas básicas. Primeiro, é necessário encontrar os serviços disponíveis que atendem às exigências do consumidor. Esta etapa pode ser realizada de forma ativa, em que o consumidor consulta outros dispositivos em busca dos serviços, ou de forma passiva, em que o consumidor apenas recebe informações dos serviços locais disponíveis sem a necessidade de requisitá-las.

A segunda etapa é a escolha do serviço a ser utilizado dentre os disponíveis. Dados o modelo e os protocolos existentes para descoberta de serviços, a escolha pode ser feita em

função do serviço com melhor descrição ou da melhor conectividade com o provedor, mas essas duas abordagens não consideram a qualidade do serviço prestado, que corresponde à sua reputação.

As duas últimas etapas são a requisição e a resposta do serviço. Na requisição, uma mensagem é encaminhada ao provedor do serviço com os dados necessários para que este consiga realizar a tarefa ofertada. Por último, depois de realizar a tarefa, o provedor encaminha a resposta ao consumidor.

Como mencionado anteriormente, as informações trocadas nesta etapa podem ser de extrema importância. Fornecer e receber dados de participantes não confiáveis pode ser um risco. Um sistema de reputação pode, então, auxiliar na detecção desses participantes, classificando-os a partir de votos de interações anteriores.

Dada a inexistência de um mecanismo de avaliação que pondere os diversos parâmetros relevantes nessa classificação utilizando-se de plataformas distribuídas e considerando aspectos de confiabilidade e anonimização, este trabalho apresenta um sistema de reputação distribuído para avaliação de interações baseadas em serviços. Para apoiar a seleção e o uso de serviços, essa proposta trata da identificação de serviços, de mecanismos e de valores de reputação para serviços e *softwares* acessíveis diretamente entre dispositivos. Aspectos relevantes na criação deste sistema incluem a (1) identificação de cada provedor e de cada consumidor de serviços; (2) protocolos para descoberta de serviço e de sua reputação; (3) mecanismos para comprovar interações entre clientes e provedores de serviço e (4) mecanismos para transmitir e computar votos que atestem a qualidade de um serviço disponível a partir da sua reputação.

3.2 Arquitetura do Sistema

No modelo de reputação distribuído proposto neste trabalho, as funções do sistema são executadas pelos próprios participantes da rede. Assim, cada participante deve desempenhar um dos papeis que definem funções essenciais para o funcionamento correto do sistema, que são: **minerador**, **servidor de registro**, **verificador completo** e **verificador simples**. O primeiro, **minerador**, é responsável por verificar a validade dos votos transmitidos na rede e por agrupá-los em unidades de histórico, chamadas de blocos. O **servidor de registro** armazena e distribui os endereços dos participantes *online*, funcionando como um super nó de uma rede ponto a ponto, dentro da sub rede do sistema de reputação. O **verificador completo** é responsável por armazenar e verificar os blocos e votos transmitidos na rede. Por último, **verificador simples** é o papel desempenhado por um dispositivo móvel que se utiliza da rede local para interações

por meio de serviços; ele é o gerador de voto e o consumidor das reputações.

De modo geral, o fluxo das interações entre os participantes começa na busca por um serviço. Ao requisitá-lo aos dispositivos próximos um consumidor encontra os serviços à disposição que cumprem as funções desejadas. Junto com os serviços oferecidos, cada provedor pode enviar seu identificador; com ele, o consumidor pode consultar a reputação do mesmo. Para que isto ocorra, o consumidor deve estar conectado à rede, tendo acesso a outros participantes que armazenam essas informações. Este dispositivos consultados podem ser mineradores ou verificadores completos.

Também é possível que o próprio provedor de um serviço forneça dados de sua reputação em formato verificável pelo cliente. Uma possível solução seria utilizar autoridades certificadoras na autenticação dos dados fornecidos, que podem garantir por meio de assinaturas confiáveis a confiabilidade dos dados transmitidos e também o tempo de vida da informação.

Dispositivos que exercem a função de verificador completo são responsáveis por armazenar os votos, os blocos e as reputações do sistema, podendo ser usados para consultar a reputação de qualquer dispositivo conhecido. Utilizando endereços IPs previamente conhecidos, um consumidor então consulta algum servidor de registro em busca de dispositivos que exercem essa função. Ao receber uma lista de IPs de dispositivos conectados à rede, o consumidor deve estabelecer comunicação com eles e então poderá consultar as informações que desejar. Tendo os valores das reputações, o consumidor decide qual provedor utilizar.

Feita a escolha, inicia-se a comunicação com o provedor enviando uma mensagem de requisição do serviço desejado. Este deve encaminhar os dados de resposta do serviço utilizado. Durante a comunicação, um registro, chamado **prova de interação**, deve ser gerado. Esse registro, acrescido de assinaturas associadas a pares de chave pública-privada, garante que uma interação realmente ocorreu

Terminada esta etapa, ambos, provedor e consumidor, podem votar na qualidade da interação. O voto inclui a prova de interação, a indicação da data e da hora do ocorrido além das informações dos dois participantes. Estes dados devem ser encaminhados aos verificadores completos e mineradores conhecidos. Caso não seja possível estabelecer conexão com estes dispositivos, pode-se encaminhar o voto para um outro verificador simples para que este o encaminhe aos responsáveis. O dispositivo intermediário deve verificar a validade dos dados antes de repassá-los. Vale observar que, devido ao uso de mecanismos de cálculo de integridade e assinaturas, não é possível a um nó intermediário adulterar ou forjar conteúdos

associados aos votos.

Quando um minerador ou verificador completo recebe um voto, ele deve encaminhá-lo aos demais participantes com mesmas funções sobre os quais possui conhecimento. O verificador completo apenas armazena o voto para futuras consultas enquanto o minerador agrupa votos que ainda não foram utilizados para criar novos blocos. Os blocos são unidades de histórico do sistema e representam a atualização das reputações, ou seja, quando um bloco é criado as reputações são recalculadas.

Para que um bloco seja criado, os mineradores devem empregar um esforço computacional que garante a estabilidade do sistema (NAKAMOTO, 2008). Devido ao seu modelo de corrente e a dificuldade para encontrar a prova de trabalho, o esforço computacional necessário para alterar um determinado bloco é proporcional à quantidade de blocos subsequentes (NAKAMOTO, 2008). Esta relação será melhor detalhada na seção 3.4.

Quando um minerador consegue criar um bloco, ele deve transmiti-lo ao maior número de dispositivos conhecidos para que eles verifiquem a validade dos dados e atualizem os valores das reputações. Nesta etapa, o ciclo se fecha e as próximas consultas às reputações devem obter os valores atualizados.

A figura 3.1 ilustra os atores e suas interações do modelo proposto para cálculo, disseminação e uso de informações de reputação.

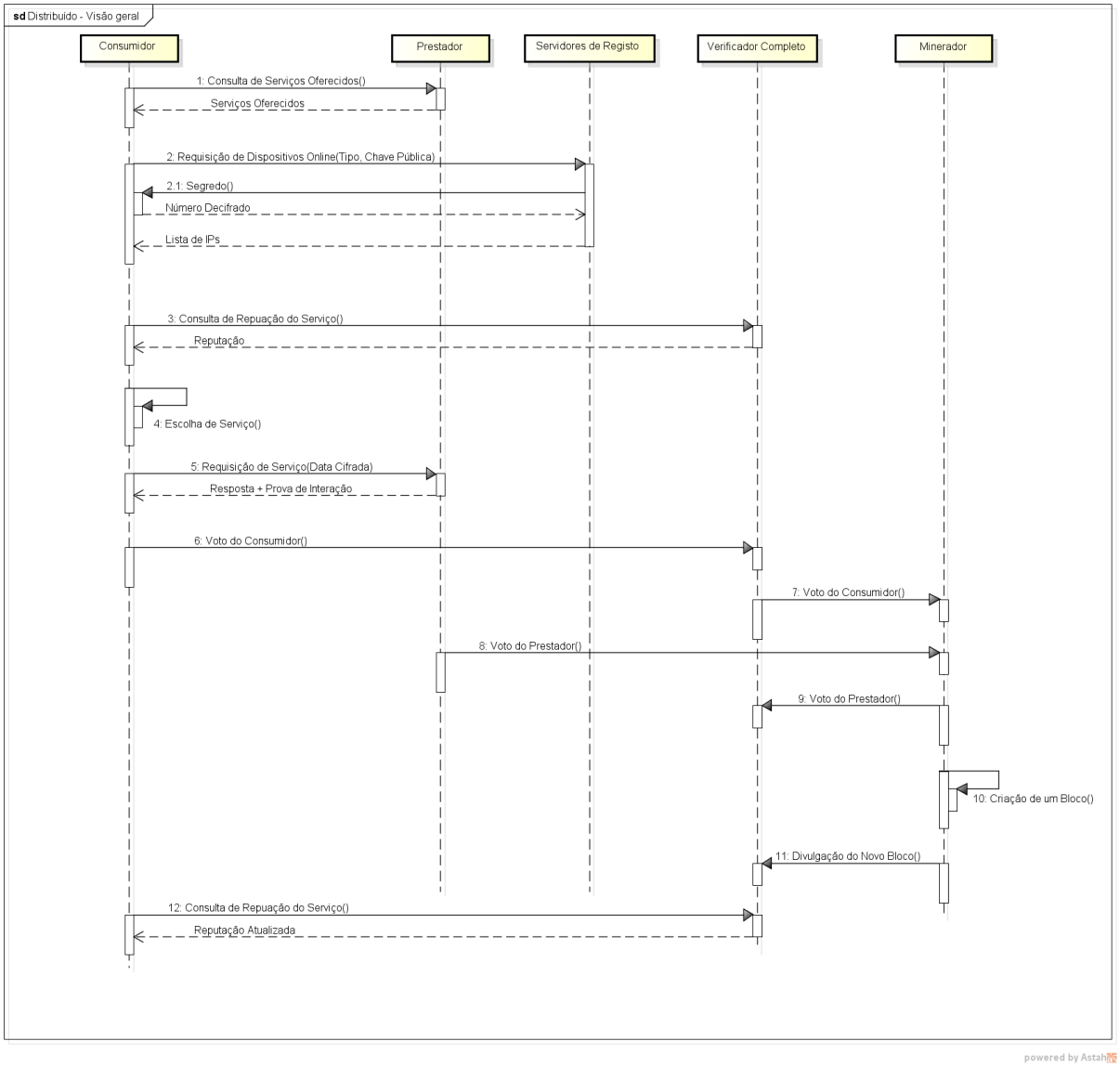

Figura 3.1: Modelo Distribuído - Fluxo Geral

3.3 Modelo de Dados

O modelo de reputação distribuído utiliza algumas mensagens para garantir a convergência dos dados entre os participantes. Neste sistema, o voto é uma estrutura simples usada para transmitir as avaliações dos participantes sobre uma interação. Ele contém dois identificadores, sendo um do dispositivo votante e o outro do respectivo votado, o tipo do voto, uma prova de interação, um resumo da mensagem e a data da interação. A imagem 3.2 ilustra os dados contidos em um voto.

A prova de interação é o resultado da cifra da data da interação e dos identificadores dos participantes. Estes dados são trocados durante a comunicação para realização do serviço e

cifrados com a chave privada do consumidor e depois com a chave privada do provedor. Ao final da comunicação ambos devem possuir a prova de interação que será encaminhada junto com o voto.

Figura 3.2: Representação dos dados presentes em um voto

Duas informações do voto são utilizadas para verificar a sua validade. A prova de interação garante que houve troca de informações entre o votante e o votado, pois somente pode ser criada com as chaves privadas que identificam os participantes. O resumo, segundo dado utilizado na validação, garante a autenticidade das informações criando um *hash* dos dados transmitidos com a chave privada do remetente. Dessa forma, votos falsos não podem ser criados e nem os dados de votos, modificados.

Um voto deve ser encaminhado aos demais participantes da rede até que seja transmitido aos dispositivos responsáveis pelo gerenciamento do histórico para que ele seja contabilizado e as reputações sejam atualizadas. Isto é feito criando-se blocos, que são unidades de histórico que agrupam votos e reputações ao longo do tempo. Cada bloco pode ser separado em cabeçalho e corpo. O primeiro contém as informações mínimas de um bloco enquanto o corpo armazena as informações completas sobre os votos e as reputações.

No cabeçalho estão presentes: a raiz de uma árvore Merkle (MERKLE, 1982) dos votos incluídos no bloco, a *hash* do bloco anterior, um *nonce*, a data de criação deste bloco e a prova de trabalho. Esse modelo, que utiliza a *hash* do bloco anterior no novo bloco, possibilita uma representação ordenada dos blocos criados, chamada corrente de blocos.

O *nonce* é um número aleatório adicionado às informações do bloco. Esse número é usado como validador do bloco pois este só é aceito caso o valor *hash* dos dados do cabeçalho, incluindo este número, contenha uma quantidade específica de zeros em seu início. O *hash* resultante é chamado de prova de trabalho. Essa técnica faz com que o custo computacional da criação de um bloco aumente, impedindo que os mesmos sejam alterados. Conforme a quantidade de poder computacional empregada na solução desse problema aumenta, mais

rápido encontra-se o *nonce* e por isso a quantidade de zeros exigida deve ser definida e atualizada pelo sistema.

O cálculo deste valor é feito com a média de tempo do intervalo entre criação dos últimos blocos. Com este dado é possível inferir o poder médio de processamento empregado na criação de blocos e, portanto, determinar o número de zeros necessários para manter o sistema seguro. O tempo médio ideal definido para a criação de um bloco precisa ser cuidadosamente escolhido para que não seja fácil demais criar um bloco, a ponto de permitir alterações indevidas, e para que não leve tempo desnecessário, atrasando a atualização das reputações.

De tempos em tempos é necessário recalcular a quantidade de zeros, pois o poder de processamento dedicado a esta função pode ter se alterado. Mineradores e verificadores completos, por terem controle de quais blocos são aceitos pelo sistema, devem garantir que este valor seja recalculado. Isto pode ser feito determinando uma quantidade de blocos gerados antes que o cálculo seja refeito.

Outra estrutura importante do bloco é a árvore Merkle, que contém os *hashes* dos votos incluídos no bloco. Esta estrutura é usada para relacionar um determinado voto ao seu bloco. Assim, tendo um voto e a árvore Merkle, é possível saber se o voto pertence ao bloco que possui esta árvore. Isso é feito para garantir que os votos do histórico não sejam modificados. A raiz da árvore é um dos dados utilizados na criação da prova de trabalho, que será detalhada na seção 3.6.

Para a construção de uma árvore Merkle são necessários ao menos dois elementos, neste caso, dois votos. Porém, a estrutura não limita a quantidade máxima de elementos, permitindo que novos votos sejam adicionados conforme são recebidos. Para isso, basta inseri-los na base da árvore e criar os *hashes* dos nós pais até a raiz, como descrito na seção 2.2. Porém, adicionar votos a uma árvore altera o valor de sua raiz, o que força o dispositivo a reiniciar a busca por um *nonce*.

Cada novo bloco criado deve ser transmitido aos mineradores e verificadores completos. Quando este novo bloco chega a um dispositivo que exerce uma dessas funções, este verifica os dados contidos e, caso estejam corretos, o bloco é adicionado ao histórico. Logo em seguida ele descarta os votos que foram usados no novo bloco, cria uma nova árvore Merkle com os votos restantes e reinicia o trabalho de busca pelo *nonce*.

Como cada bloco possui um *hash* do bloco anterior podemos dizer que eles estão ligados e por isso o histórico é referenciado como corrente de blocos. A imagem 3.3 ilustra o conteúdo de um bloco e suas referências aos outros blocos.

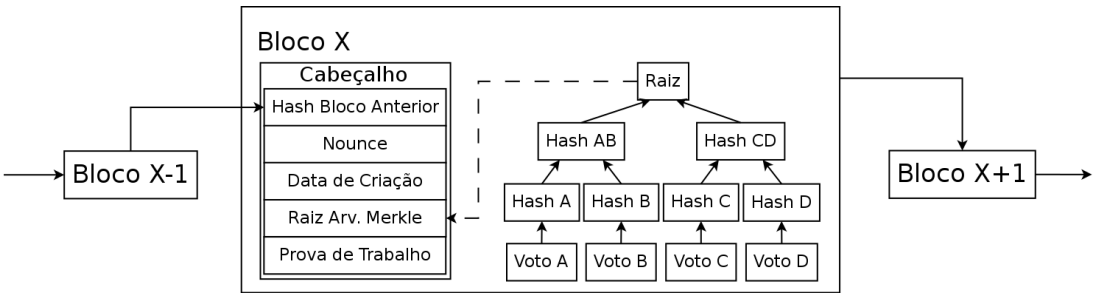

Figura 3.3: Representação dos dados presentes em um bloco

3.4 Funções do Sistema

As estruturas listadas na seção anterior são geradas e armazenadas por dispositivos que executam funções pré-definidas, essenciais ao sistema de reputação. A função mais trivial é chamada de verificação simples. Os dispositivos que executam essa função apenas geram votos quando oferecem ou consomem serviços. Estes também podem verificar e encaminhar votos quando preciso. Desta forma, não é necessário muito poder de processamento ou armazenamento, podendo ser executa por dispositivos móveis, por exemplo.

Esses dispositivos apenas interagem com outros, provendo e/ou consumindo serviços e encaminhando seus votos aos responsáveis. Contudo, votos podem ser transmitidos aos verificadores simples, mas isso só deve ocorrer quando o transmissor não possui comunicação com mineradores ou verificadores completos. Nesses casos, os votos são repassados a outros verificadores simples até que um deles, possuindo conexão com a rede do sistema de reputação, encaminhe os votos a um verificador completo ou a um minerador. Estes são responsáveis por encaminhar o voto recebido aos demais dispositivos que desempenham essas funções.

Um verificador completo tem a responsabilidade de verificar, armazenar e referenciar os votos e os blocos do sistema de reputação. Esse papel deve ser desempenhado por dispositivos com poder de armazenamento considerável, pois precisam manter todo o histórico de blocos e votos do sistema. As funções desempenhadas por eles têm grande importância, pois garantem a integridade e confiabilidade das reputações além de disponibilizarem esses dados aos demais usuários.

O papel do minerador é criar blocos. Para tanto, um minerador recebe novos votos e os armazena para criar uma árvore Merkle, sendo que o valor da raiz da árvore é utilizado no cabeçalho do bloco. Porém, a maior parte dos seus recursos é dedicada a encontrar um valor *nonce* tal que o *hash* do cabeçalho do bloco, como descrito anteriormente, tenha uma

quantidade mínima de zeros em seu início. Ao criá-lo, um minerador, deve repassá-lo aos demais dispositivos para que eles validem as informações e consolidem um de seus blocos como o mais novo bloco da corrente. A atualização é feita transmitindo o bloco aos demais participantes online da lista recebida do servidor de registro. Estes dispositivos, ao receberem o bloco, validam as informações da prova de trabalho e do resumo, e, caso estejam corretos, também anunciam o novo bloco.

A dificuldade de se criar um bloco é necessária para que seus valores não possam ser alterados. A corrente de blocos, como é chamada a estrutura sequencial de blocos, garante que para alterar um deles seja necessário alterar também todos os blocos seguintes pois uma das verificações é o valor do *hash* do bloco anterior. Pela dificuldade de encontrar o *nonce* para cada um deles, isso torna-se computacionalmente inviável.

O papel do servidor de registros, neste sistema, é traduzir identificadores em endereços IPs, ou seja, localizadores dos nós na rede ativos para a função desejada. O servidor deve repassar aos usuários, que desejam se conectar à rede, apenas os IPs de dispositivos que desempenham funções de mineradores ou verificadores completos. Essa restrição tem a finalidade de diminuir o fluxo de dados entre verificadores simples que não desejam e nem precisam armazenar votos de outros dispositivos. Todo voto ou bloco deve ser validado antes de ser retransmitido. O servidores de registro devem ser mantidos pelos participantes do sistema.

3.5 ECDSA e Reputação Inicial

Considerando o caráter distribuído da arquitetura do sistema, a identidade de um dispositivo é definida por um número e um par de chaves pública-privada. Estes identificadores podem ser criados por qualquer dispositivo. A primeira etapa é criar um par de chaves e depois fazer um *hash* da chave pública. A *hash* será usada como identificador único da reputação e o par como autenticador das mensagens trocadas. O fato do identificador ser consequência direta da chave pública garante a autenticação da identidade pela chave privada. Como cada mensagem possui um resumo, ela só é válida se criada pela chave privada correspondente à identidade.

Neste momento ainda não existe nenhum registro de voto ou de reputação do participante, então qualquer consulta por uma reputação de um identificador sem votos retorna o valor inicial definido pela formulação.

No modelo distribuído aqui apresentado, a criação das chaves é feita por algoritmos ECDSA (*Elliptic Curve Digital Signature Algorithm*) (BROWN, 2010). Esse algoritmo utiliza a equação 3.1 para criar certificados e garantir a autenticidade das mensagens enviadas no

sistema. Os parâmetros a e b irão definir algumas propriedades da curva elíptica como eficiência computacional e previsibilidade dos pontos.

$$y^2 = x^3 + ax + b \tag{3.1}$$

Os parâmetros necessários para o funcionamento do algoritmo são definidos por uma sêxtupla $T = (p, a, b, G, n, h)$. O primeiro parâmetro, p, representa o intervalo de valores válidos da função e deve ser um número primo maior que três; sendo assim, a função utilizada no algoritmo é definida por $E(\mathbb{F}_p)$. A equação 3.2 exemplifica a função da curva elíptica com seu domínio limitado a p, para isso utiliza-se o operador *mod* que representa o resto de uma divisão por p (NACY; OH; LEONE, 2013).

$$E : y^2 \equiv x^3 + ax + b \mod p \tag{3.2}$$

Os parâmetros a e b definem as características da função, sendo $a, b \in \mathbb{F}_p$; G um ponto da função $E(\mathbb{F}_p)$ que será usado como ponto base; n, a ordem do ponto base G; e h, o cofator definido pela divisão da ordem da equação $E(\mathbb{F}_p)$ pela ordem n, demonstrada na equação 3.3.

$$h = \frac{\sharp E(\mathbb{F}_p)}{n} \tag{3.3}$$

3.5.1 Geração do par de chaves pública-privada

A criação do par de chaves deve ser feita seguindo o seguinte procedimento:

1. Seleciona-se um número inteiro e aleatório, d, no intervalo $[1, n-1]$.
2. Encontra-se $Q = dG$.

Assim, d será chave privada e Q a chave pública. A função definida para a obtenção do valor aleatório d deve ser tal que exista pouco risco de conflitos, ou seja, é necessário que a probabilidade desse algoritmo fornecer número aleatórios iguais seja baixa.

3.5.2 Geração da Assinatura

Tendo os parâmetros do algoritmo e o par de chaves (d, Q) definidos, a criação da assinatura de uma mensagem m se dá pelo procedimento a seguir:

1. Seleciona-se um número inteiro e aleatório k no intervalo $[1, n-1]$.

2. Encontra-se: $kQ = (x_1, y_1)$.
3. Encontra-se: $r = (x_1 \mod n)$. Se $r = 0$ então volta-se ao passo 1.
4. Encontra-se: $Hash$(m) e transforma-o em um número inteiro $H(m)$.
5. Encontra-se: $s = (k^{-1}(H(m) + dr) \mod n)$. Se $s = 0$, então volta-se ao passo 1.

3.5.3 Verificação da Assinatura

A verificação da assinatura se dá pelos algoritmo apresentado nesta subseção.

1. Verifica-se que r e s são inteiros no intervalo $[1, n-1]$.
2. Encontra-se: Hash(m) e transforma-o em um número inteiro $H(m)$.
3. Encontra-se: $w = (s^{-1} \mod n)$.
4. Encontra-se: $u_1 = (H(m)w \mod n)$ e $u_2 = (rw \mod n)$.
5. Encontra-se: $X = (x_1, y_1) = (u_1 G + u_2 Q)$. Se $X = 0$, rejeita-se a assinatura.
6. Encontra-se: $v = (x_1 \mod n)$.
7. Se $v = R$, então aceita-se a assinatura.

Muitas implementações desse algoritmos estão disponíveis na forma de código livre. Além da tradicional implementação da OpenSSL (FOUNDATION, 2015), a PolarSSL (LIMITED, 2015) também está disponível e pode ser usada como forma alternativa. Essas implementações são importantes para fortalecer o uso desse modelo criptográfico.

3.5.4 Segurança

Como descrito na seção 2.3, o melhor algoritmo para resolução do ECDLP possui complexidade de ordem $O(\sqrt{k})$. O número esperado de interações necessárias para quebrar a assinatura se dá pela fórmula 3.4, sendo t o número de bits de segurança. Portanto, para quebrar uma criptografia de 2^{2t} bits seriam necessárias aproximadamente 2^t operações. Na implementação deste projeto sugere-se o uso de criptografias de 2^{2048} bits ou superior. Dessa forma, a o número de interações esperadas seria, no mínimo, da ordem de 2^{1024}.

$$\log_2 p = 2t \tag{3.4}$$

Como, usando essa quantidade de bits, a probabilidade de conflitos é baixa, qualquer par de chaves gerado usando as especificações do algoritmo ECDSA muito provavelmente pode ser considerado válido. No caso de conflitos a mesma reputação será utilizada por ambos os participantes, assim pode-se tirar vantagem nos casos em que a reputação de um outro

participante é bem qualificada. Contudo, o custo necessário para buscar por força bruta um par de chaves válido já utilizado é superior ao necessário para adquirir uma boa reputação. A maioria das formulações utilizadas permite obter uma boa reputação com poucas interações.

Outro aspecto de segurança importante é o controle de criação dessas identidades. Isso pode ser feito por um grupo de verificadores completos que unem-se para criar uma autoridade certificadora distribuída. Assim, elas assinariam a chave pública do participante e, ao fazê-lo, poderiam verificar se é uma identidade já existente ou reconhecer uma tentativa de criação de múltiplas identidades por meio do endereço de requisição.

3.6 Protocolos de Comunicação

Pensando nas boas práticas de implementação de serviços, este trabalho propõe um protocolo de comunicação utilizando o modelo de RESTful e a inserção de algumas informações do provedor, referentes ao sistema de reputação, nos protocolos existentes para descoberta de serviços.

A inserção dessas informações nos protocolos existentes tem como intuito promover o uso do sistema sem a necessidade de trocas adicionais de mensagens. Os principais protocolos de descoberta de serviço utilizam XML na divulgação de serviços e provedores. Esse modelo de mensagem permite a inserção de informações adicionais sem prejudicar sistemas legados. Assim, possibilita que um provedor de serviço divulgue seu identificador e sua reputação junto com as descrições dos serviços que oferece utilizando o mesmo arquivo. Com essas informações armazenadas localmente e transmitidas junto com as mensagens dos protocolos de descoberta, não é necessário que o consumidor de um serviço precise fazer uma nova requisição para o provedor solicitando seu identificador no sistema de reputação e também não é preciso ter acesso à Internet para solicitar a reputação deste identificador aos verificadores completos.

Para que as informações transmitidas diretamente pelo provedor de serviço sejam confiáveis é preciso que elas possuam algum método de controle para o tempo de validade e para a integridade dos dados transmitidos. Isso pode ser implementado por meio de certificados autenticadores que garantam a procedência das informações. Verificadores completos podem criar autoridades certificadoras distribuídas como descrito na seção 2.4. Esses verificadores podem transmitir as reputações autenticadas para facilitar a verificação das informações.

Usando esse mecanismo um provedor de serviços que requisitasse sua própria reputação a

um verificador completo confiável receberia como resposta um arquivo contendo sua reputação, a data de emissão dessa informação e uma assinatura do verificador que a emitiu. O cliente, possuindo uma lista de certificados confiáveis, poderia validar as informações recebidas do provedor. Esse mecanismo facilita a comunicação pois o cliente não precisa possuir acesso à Internet para requisitar a reputação pela rede do sistema de reputação.

Durante a interação para o consumo do serviço sugere-se o envio de dados adicionais, pois é preciso criar um arquivo que comprove a interação entre os dispositivos. Para isso, o consumidor deve enviar a cifra do instante atual (*timestamp*) e os identificadores de ambos os envolvidos. O provedor, ao receber os dados, deve cifrá-lo novamente com sua chave privada e encaminhá-lo junto com a resposta do serviço. Esse dado é chamado de prova de interação e os dados contidos garantem que houve comunicação entre os dispositivos.

Alguns ataques poderiam ocorrer sobre a chave interação. Primeiro, ela poderia ser usada em mais de um voto para representar interações diferentes ou ainda a última cifra poderia ser refeita utilizando outra chave para gerar uma prova de interação falsa. Contra isso, a data e hora garantem que esta prova de interação não será usada novamente para representar outra comunicação entre os mesmos dispositivos e os identificadores garantem que ela não sera usada por outros dispositivos para gerar provas de interação falsas.

Caso os serviços utilizem XML ou JSON na comunicação, tipos de dados comuns em serviços web, a inserção de dados é trivial e não compromete a comunicação com outros dispositivos que não possuem a implementação para tratar destes dados. Caso o sistema não utilize um modelo de comunicação flexível, é recomendável que outro serviço seja implementado para que essas informações sejam acessíveis. Para isso, é preciso referenciar a requisição realizada. É importante que esses dados estejam disponíveis para que ambos possam encaminhar seus votos.

Terminada a interação, ambos os dispositivos precisam divulgar seus votos para os sistema de reputação. Essa e as demais comunicações entre dispositivos que envolvem somente dados do sistema precisam ser flexíveis e de fácil acesso. Pensando nisso, um modelo de serviços que atende a esses requisitos de forma adequada seria o RESTful (FIELDING; TAYLOR, 2000) com transmissão de dados em formato JSON. Assim, os demais exemplos de comunicação apresentados nesta seção seguirão este modelo.

Existem cinco serviços essenciais para que o sistema de reputação funcione corretamente: o de **obtenção de referência à rede**, a **divulgação de novos blocos**, a **divulgação de novos votos**, a **requisição de reputação** e a **requisição de blocos**.

A **obtenção de referência à rede** do sistema de reputação é feita requisitando a um servidor de registro o endereço de dispositivos online para que seja possível enviar e receber informações em uma conexão ponto a ponto. Nessa requisição deve ser informado o tipo de função exercida, o seu identificador e a sua chave pública. A informação da função exercida pelo dispositivo é importante para que seja possível separar mineradores e verificadores completos, que desejam conectar-se com outros dispositivos e receber atualizações, dos verificadores simples, que apenas desejam enviar dados das suas ações. Como resposta à solicitação, o dispositivo recebe uma cifra de um número aleatório feita com sua chave pública, esta deve ser decifrada e encaminhada de volta para confirmar a identidade do solicitante. O fluxo das mensagens pode ser observado na figura 3.4 e os modelos nos códigos 3.1, 3.2, 3.3 e 3.4 (os exemplos de chaves e cifras estão utilizando criptografia de 128 bits apenas pera facilitar a visualização).

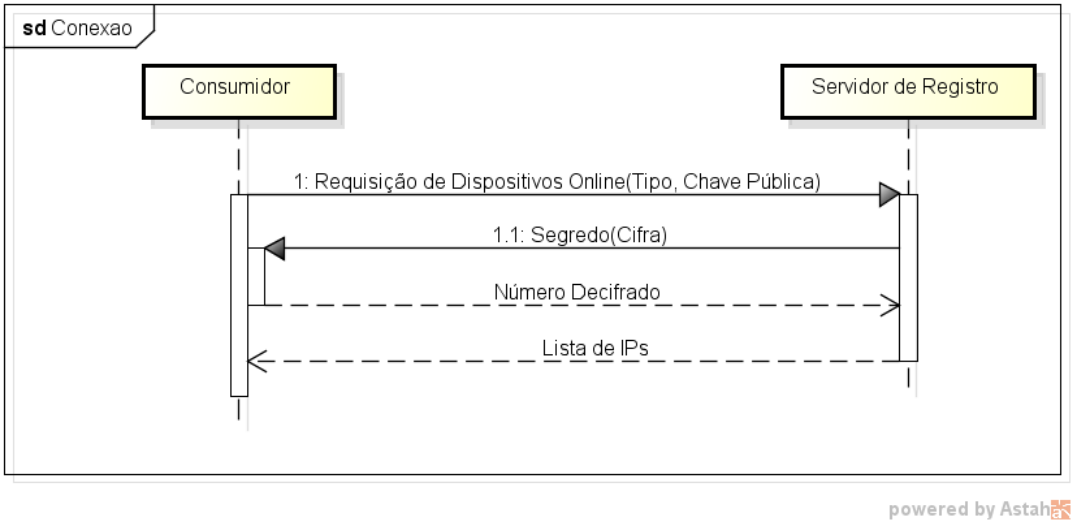

Figura 3.4: Representação dos dados presentes em um bloco

Código 3.1: Modelo da mensagem da requisição de dispositivos *online*.

```
{
  "type":          "Simple Inspector",
  "id":            "28c5c77b45563634fd8633b5a114ec5b",
  "public-key":    "bf7cdd2a48a4416c061798676961f65d"
}
```

Código 3.2: Modelo da mensagem de resposta da requisição contendo uma cifra.

```
{
  "cypher": "4d9133d82be6de5a7cd977bb7ee706ea"
```

```
}
```

Código 3.3: Modelo da mensagem de confirmação de identidade - número decifrado.

```
{
  "number": "4589642258"
}
```

Código 3.4: Modelo da mensagem de resposta contendo uma lista de dispositivos *online*.

```
[
  {
    "id":   "28c5c77b45563634fd8633b5a114ec5b",
    "type": "Complete Inspector",
    "IP":   "0.0.0.1"
  },
  {
    "id":   "bf7cdd2a48a4416c061798676961f65d",
    "type": "Miner"
    "IP":   "0.0.0.2"
  }
]
```

Vale ressaltar que este protocolo de cadastro tenta prevenir o uso de chaves públicas de terceiros em servidores de registro. Nesta etapa poderiam ser adicionadas mensagens para a verificação de chave pública já existente utilizando-se da autenticação de um autoridade certificadora, porém esta abordagem exigiria um estudo adicional sobre a confiabilidade de autoridades certificadoras distribuídas.

O serviço de **divulgação de novos votos** é utilizado tanto por um consumidor, ou provedor de serviços, que deseja encaminhar seu voto, quanto por mineradores e verificadores completos que recebem um voto e precisam divulgá-lo aos demais dispositivos de mesma função. Essa divulgação deve ser feita a todos os dispositivos que exercem a função de minerador ou de verificador completo. Ela deve conter todas as informações do voto: tipo do voto, identificador do votante, identificador do votado, valor do voto, prova de interação e um resumo da mensagem. O modelo dessa mensagem está exemplificado no código 3.5.

Código 3.5: Modelo da mensagem da divulgação de um voto.

```
{
```

```
  "type":              "Consumer Vote",
  "voter PubKey":      "28c5c77b45563634fd8633b5a114ec5b",
  "voted PubKey":      "bf7cdd2a48a4416c061798676961f65d",
  "value":             0.75,
  "proof of inter":    "4d9133d82be6de5a7cd977bb7ee706ea",
  "digest":            "4419dafda138a725a0832258a8b5c42e"
}
```

O tipo do voto indica se é um voto proveniente de um provedor de serviço ou de um consumidor. Isso facilita a verificação da prova de interação, pois não é necessário tentar decifrá-la com as duas chaves. Sabendo o tipo do voto e tendo as chaves públicas, é possível inferir qual das duas chaves utilizar para validar corretamente a prova de interação e o resumo.

A divulgação de um novo bloco acontece quando um minerador encontra um *nonce* válido, como descrito na seção 3.4, e cria um novo bloco. Esse bloco precisa ser divulgado para que os demais mineradores e verificadores completos atualizem seus históricos e a tabela de reputações. A figura 3.5 exibe a sequência de mensagens necessárias para que a atualização do bloco seja feita.

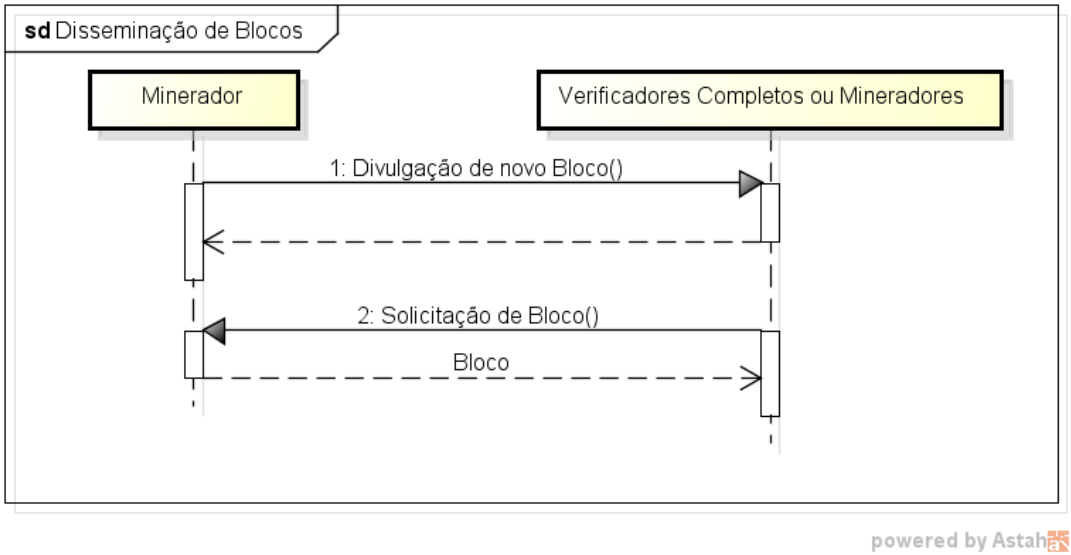

Figura 3.5: Representação dos dados presentes em um bloco

Ao criar o bloco, o minerador deve utilizar o serviço de **divulgação de novos blocos** para todos os dispositivos que exercem a função de minerador ou de verificadores completos. A solicitação encaminha o cabeçalho e o número do bloco na corrente. Os dispositivos que recebem essa requisição verificam se o número do bloco é maior do que o último bloco de suas

correntes e, em caso afirmativo, se o cabeçalho do bloco é válido. O modelo das mensagens de divulgação de blocos pode ser observado no código 3.6.

Código 3.6: Modelo da mensagem da divulgação de um bloco.

```
{
  "number":         323,
  "miner PubKey": "28c5c77b45563634fd8633b5a114ec5b",
  "header": {
    "previus hash":   "17050db914dd42240d08d38c528e1353",
    "proof of work": "000000000289270498942f16a12793b",
    "date":             "2015-06-06 00:00",
    "merkle root":    "723e05b69db0583495a195c8db936571"
  },
  "digest":         "4419dafda138a725a0832258a8b5c42e"
}
```

Se um dispositivo possui uma corrente menor que a corrente divulgada pelo serviço descrito anteriormente então ele deve atualizar sua corrente requisitando os blocos faltantes. Para isso, utiliza o serviço de **requisição de blocos**, encaminhando os números dos blocos que deseja receber. Como resposta, recebe os blocos solicitados e deve verificar a validade de cada um. Quando um dispositivo recebe blocos novos deve usar o serviço de divulgação de blocos para anunciar a nova corrente aos dispositivos conhecidos. Isso faz com que os votos criados se propaguem. O modelo da mensagem de requisição de blocos e o modelo de reposta dessa requisição podem ser observados nos códigos 3.7 e 3.8 respectivamente.

Código 3.7: Modelo da mensagem de requisição de blocos.

```
{
  "blocks": [ 323, 322 ],
  "digest": "4419dafda138a725a0832258a8b5c42e"
}
```

Código 3.8: Modelo da mensagem de resposta da requisição de blocos.

```
{
  "blocks": [
     {
       "number": 323,
       "header": { ... },
```

```
        "body": { ... }
      },
      {
        "number": 322,
        "header": { ... },
        "body": { ... }
      }
  ],
  "digest": "4419dafda138a725a0832258a8b5c42e"
}
```

Ao receber um novo bloco é preciso conferir se os dados estão corretos, utilizando a chave pública enviada, verificar se os votos contidos também são válidos e recalcular e salvar as reputações alteradas. Finalizadas estas etapas, o bloco pode ser adicionado a corrente de blocos e as reputações podem ser atualizadas.

O último serviço essencial é o de **requisição de reputação**. Mesmo que a reputação possa ser encaminhada pelo próprio provedor de serviço, é desejável poder conferi-la em outros verificadores completos. Para requisitar esse serviço deve-se encaminhar uma lista com os identificadores das reputações desejadas, obtendo como resposta uma lista de identificadores com suas respectivas reputações. Os modelos de requisição e de resposta podem ser vistos nos códigos 3.9 e 3.10 respectivamente.

Código 3.9: Modelo da mensagem de requisição de reputações.

```
{
  "ids": [
    "28c5c77b45563634fd8633b5a114ec5b",
    "bf7cdd2a48a4416c061798676961f65d"
  ],
  "digest": "4419dafda138a725a0832258a8b5c42e"
}
```

Código 3.10: Modelo da mensagem de resposta da requisição de reputações.

```
{
  "reputations": [
    {
      "id": "28c5c77b45563634fd8633b5a114ec5b",
```

```
      "reputation": 8.4,
    },
    {
      "id": "bf7cdd2a48a4416c061798676961f65d",
      "reputation": 3.1,
    }
  ],
  "digest": "4419dafda138a725a0832258a8b5c42e"
}
```


3.7 Validação dos Dados

A validação das mensagens transmitidas na rede é necessária para manter a integridade das informações e reputações. A seguir, serão descritos os métodos adotados para validação das mensagens em geral e dos votos e blocos criados, as duas principais estruturas de dados do sistema de reputação proposto.

3.7.1 Validação das Mensagens

Todas as mensagens transmitidas no sistema possuem um resumo. O resumo é uma assinatura digital feita com a chave privada do seu emissor, que pode ser verificado como descrito na seção 3.5. Com o resumo, pode-se assegurar o emissor da mensagem e se houve alguma alteração em seu conteúdo.

3.7.2 Validação de Votos

As validações dos votos possuem duas partes, já que, além da verificação do resumo, é necessário validar a prova de interação. A verificação do resumo se dá pela assinatura digital, como descrito anteriormente. A verificação da prova de interação é feita decifrando o valor deste campo primeiro com a chave pública do provedor e depois com a chave pública do consumidor do serviço. O valor resultante deve ser uma data válida e igual à enviada no corpo do voto concatenada aos identificadores do votante e do votado. Dessa forma, podemos garantir que houve uma interação entre o votante e o votado, impedindo que votos falsos, não provenientes de uma interação real, sejam contabilizados. Esta verificação deve ser feita sempre que um dispositivo recebe um novo voto e, caso não seja válido, o voto deve ser descartado.

Por exemplo, se um dispositivo A solicita um serviço de um dispositivo B, A deve encaminhar, junto com a solicitação, a data da interação e os identificadores cifrados com sua chave privada. Ao enviar a resposta da solicitação B deve cifrar novamente o valor recebido com a sua chave. Assim, ambos possuem a prova de interação que deve ser encaminhada no voto. Ao encaminhar o voto a data e prova de interação devem estar presentes assim como as chaves públicas dos participantes. Portanto, quando for necessário verificar a veracidade de um voto basta decifrar a prova de interação primeiramente com a chave pública de B e depois com a chave pública de A e verificar se a data obtida é igual a encaminhada no voto e os identificadores estão corretos. Como o resumo garante a veracidade do emissor da mensagem e a prova de interação garante que houve comunicação entre os dispositivos, pode-se assumir que é um voto válido.

3.7.3 Validação de Blocos

A validação de blocos exige três etapas: validação da prova de trabalho, validação da árvore de Merkle e validação dos votos. Cada uma das etapas irá verificar uma parte dos dados contidos: a primeira comprova que o criador do bloco realmente disponibilizou seus recursos para encontrar o *nonce*, a segunda que a árvore de consulta dos votos está correta e a terceira que os votos utilizados no bloco são válidos.

A validação da prova de trabalho é feita em quatro etapas. A primeira consiste em verificar se ela possui a quantidade mínima de zeros em seu início conforme estabelecido pelo algoritmo do sistema. A segunda etapa é refazer o *hash* dos dados do cabeçalho do bloco com a chave pública do seu emissor validando sua autenticidade. Em seguida, deve-se verificar se a árvore Merkle está correta refazendo os *hashes* de cada um de seus nós. Por último, deve-se validar cada um dos votos do bloco como descrito na seção anterior, 3.7.2.

3.8 Formulação

Este trabalho apresenta uma proposta de um sistema de reputação, definindo desde o modelo de comunicação entre os participantes até a forma como devem ser armazenados os votos e as reputações, tornando flexível o modo como os votos são computados em reputações. A possibilidade de consulta de todo o histórico mantendo a privacidade dos participantes faz com que quase qualquer formulação seja compatível com o sistema de reputação proposto.

A formulação de um sistema de reputação é composta pelas funções e pelos algoritmos

utilizados para calcular as reputações. Estes componentes irão definir: a reputação inicial, os fatores que influenciam no seu crescimento e no seu declínio, a intensidade do impacto desses fatores e o tempo necessários para adquirir uma boa reputação. As características definidas na formulação devem ser escolhidas com cautela pois irão definir o comportamento dos usuários em relação ao sistema de reputação, tanto para os usuários honestos quanto para os que tentem fraudar o sistema. Os cenários de ataque à formulação serão tratados na seção 3.9.2.

Com a finalidade de comparar as formulações com maior destaque na literatura e eleger a mais adequada ao cenário proposto, implementamos cinco tipos de formulações. Duas delas utilizam todo o histórico para o cálculo da média sem distinção os votos. As outras duas são modificações dessas, dando maior relevância aos votos mais recentes. Também foi avaliada uma proposta de formulação desenvolvida particularmente para este cenário. Ela utiliza alguns elementos das formulações apresentadas anteriormente.

O comportamento de cada umas dessas formulações foi testado e avaliado dentro do cenário proposto neste trabalho e os resultados são apresentados no capítulo 4.

3.8.1 Média

A primeira formulação analisada é a média dos valores dos votos para um determinado objeto avaliado. Nessa formulação o valor dos votos deve variar dentro do intervalo $[0,1]$ e o valor da reputação é alterado conforme os votos são computados. O valor inicial de sua reputação é zero e isto pode causar fenômenos como o "*cold start*". A indicação de que uma identidade é nova pode ajudar nesse quesito. Contudo, tendo o primeiro voto, a média irá refletir esse valor, possibilitando avaliar, de forma mais real, o comportamento do serviço. Ao longo do tempo, conforme o número de votos cresce, a média se torna mais estável, sendo necessário um grande volume de avaliações discrepantes da média para alterá-la de forma significativa. A equação 3.5 demonstra essa formulação.

$$R(V) = \sum_{i=1}^{n} \frac{v_i}{n} \tag{3.5}$$

Sendo: R a reputação do objeto avaliado; V o conjunto de votos desse objeto; v_i o valor do voto de índice i; e n o número de votos do conjunto.

3.8.2 Média Móvel Exponencial

A segunda formulação é a média móvel exponencial. Neste modelo, os votos são computados dando maior relevância aos votos mais recentes. Para este cálculo utiliza-se o valor da reputação antiga e a média dos valores dos votos inseridos por um novo bloco. A quantidade de votos recentes utilizado no cálculo pode variar.

Existe também um fator que representa o peso dado a reputação antiga e a nova média dos votos, este determinar o quão mais relevante será a influência do histórico e dos votos recentes. A equação desta formulação está descrita em 3.6.

A partir da reputação $(0,5)$ os primeiros votos são computados utilizando o fator determinado, o que gera uma curva mais acentuada. Assim, as reputações demoram um número maior de interações para atingir o valor médio esperado. Por outro lado, a quantidade de votos não tem relevância na estabilidade de um reputação, refletindo de forma mais precisa o comportamento atual do avaliado.

$$R_t(V) = \alpha * R_{t-1} + (1-\alpha) * \frac{\sum_{i=1}^{n} v_i}{n} \tag{3.6}$$

Sendo R_t a reputação na iteração t; R_{t-1} a reputação da iteração anterior, $t-1$; V o conjunto de votos recentes desse objeto; α o fator que indica o peso da reputação atual na nova reputação, deve ser um valor do intervalo $(0,1)$; v_i o valor do voto de índice i; e n o número de votos do conjunto. Como descrito anteriormente o valor inicial da reputação é zero, ou seja, $R_0 = 0$.

3.8.3 Esperança da Distribuição Beta

Outro modelo analisado neste trabalho foi a esperança da distribuição Beta. Nesse modelo, os votos possuem apenas duas possibilidades, ou positivos, ou negativos. Esta distribuição modela de forma satisfatória eventos binários, como por exemplo a probabilidade de um jogo de cara ou coroa (DENKO; SUN; WOUNGANG, 2011). A reputação inicial também é $0,5$, a metade da reputação máxima possível neste modelo. Contudo, possui características parecidas com a média com relação à estabilização do valor de reputação ao longo do tempo. Conforme a quantidade de votos aumenta a reputação torna-se estável, sendo necessário muitos votos para alterá-la. Essa formulação é representada pela equação 3.7.

$$R(V) = \frac{P+1}{N+P+2} \tag{3.7}$$

Sendo: R a reputação do objeto avaliado; V o conjunto de votos desse objeto; P o número de votos positivos do conjunto V; e N o número de votos negativos do conjunto V;

3.8.4 Esperança Móvel da Distribuição Beta

Esta formulação é uma modificação da esperança da distribuição Beta nos mesmos moldes da média móvel exponencial, uma tentativa de dar maior relevância aos votos recentes. Para isto, utiliza-se a reputação anterior junto com o cálculo da esperança para um determinado número de novos votos. A equação 3.8 descreve seu comportamento.

$$R_t(V) = \alpha * R_{t-1} + (1-\alpha) * \frac{P+1}{N+P+2} \tag{3.8}$$

Sendo: R_t a reputação na iteração t; R_{t-1} a reputação da iteração anterior, $t-1$; V o conjunto de votos recentes desse objeto; α o fator que indica o peso da reputação atual na nova reputação, deve ser um valor do intervalo $(0,1)$; P o número de votos positivos do conjunto V; e N o número de votos negativos do conjunto V.

3.8.5 Proposta de Formulação

Com base em algumas características do modelo de sistema de reputação apresentado, propomos uma formulação para as diferentes necessidades dos provedores e dos consumidores.

Pensando nisso propomos a separação da reputação em duas, uma como provedor de serviços (R_p) e outra como consumidor (R_c). Os votos de cada uma dessas interações são computados separadamente para gerar cada um dos valores. O cálculo da reputação como consumidor é feito utilizando a fórmula de esperança da distribuição beta modificada, equação 3.9, pois, para o provedor, que analisa esta reputação, a relevância de uma interação ruim é menor.

Por outro lado, o cálculo da reputação de um provedor usa o conceito de média móvel exponencial aplicado tanto nos votos recebidos quanto no calculo da reputação. Primeiramente os votos recentes são separados em dois grupos, os com valores maiores que $0,5$ (V_P) e os com valores menores (V_N). Esses grupos são multiplicados por um fator θ que tem como objetivo elevar a importância dos votos negativos conforme a reputação cresce e dos positivos conforme ela decai. Para isso, utiliza-se a própria reputação como valor de θ. A equação 3.10 demonstra os cálculos descritos.

O valor inicial da reputação de um consumidor é $0,5$ e a de um provedor de serviço é zero.

$$R_{c_t}(V) = \alpha * R_{c_{t-1}} + (1-\alpha) * \frac{P+1}{N+P+2} \tag{3.9}$$

$$R_{p_t}(V) = \alpha * R_{p_{t-1}} + (1-\alpha) * \frac{\theta \sum_{i=1}^{|V_N|} v_{N_i} + (1-\theta) \sum_{j=1}^{|V_P|} v_{P_j}}{\theta |V_N| + (1-\theta)|V_P|} \tag{3.10}$$

A fórmula da esperança da distribuição beta modificada foi escolhida para o cálculo da reputação como consumidor pois ela exige que ele tenha algumas interações antes de obter uma reputação considerada boa e tende a se permanecer estável ao longo do tempo. No caso da abordagem utilizada no cálculo da reputação dos provedores, a separação dos votos em positivos e negativos incentiva os provedores a manterem a qualidade de seus serviços, desencorajando abusos de curto período.

3.8.6 Valores de α

Um parâmetro importante para as formulações analisadas que diferenciam novos votos do restante do histórico é o valor de α. Ele irá determinar qual o peso desses votos na reputação total e também o crescimento ou o declínio da curva de reputação dos objetos. O valor desse parâmetro pode ser fixo ou modificado conforme o tempo e/ou o valor da reputação analisada. Neste segundo caso, cada objeto avaliado deve possuir seu próprio valor de α que deve ser atualizado a cada novo cálculo.

Foram analisados alguns valores fixos para o fator α: 0,05; 0,1; 0,15; 0,2; 0,25 e 0,3. Duas funções, que levam em consideração o tempo e o valor da reputação, também foram analisadas. Em ambas as funções deve haver um limite máximo e mínimo para que nenhum dos valores envolvidos seja desconsiderado no cálculo da nova reputação. Nas fórmulas analisadas esses valores foram estabelecidos em 0,05 para o valor mínimo e 0,3 para o valor máximo.

A primeira função para o cálculo de α é demonstrada na equação 3.11, ela leva em consideração a média dos valores dos novos votos e o valor da última reputação. Caso esses valores sejam próximos, o módulo da diferença menor que um limiar estabelecido, o peso da média diminui e caso eles sejam discrepantes o peso da média aumenta.

$$\alpha_t = \begin{cases} mult(\alpha_{t-1}), & \text{se } |R_t - R_{t-1}| \geq LIMIAR \\ div(\alpha_{t-1}), & \text{caso contrário} \end{cases} \tag{3.11}$$

$$mult(x) = \begin{cases} 0,3, & \text{se } x*2 > 0,3 \\ x*2, & \text{caso contrário} \end{cases} \tag{3.12}$$

$$div(x) = \begin{cases} 0,05, & \text{se } x/3 < 0,05 \\ x/3, & \text{caso contrário} \end{cases} \tag{3.13}$$

Sendo α_t o próximo valor de α utilizado; α_{t-1} o valor de α utilizado no último cálculo da reputação; R_t a última reputação calculada e M_t a média dos valores do voto, ou a esperança, usada no último cálculo da reputação.

A segunda função é uma variação da fórmula anterior, porém o valor de α cresce linearmente.

$$\alpha_t = \begin{cases} add(\alpha_{t-1}), & \text{se } |R_t - \mathrm{M}_t| \geq LIMIAR \\ sub(\alpha_{t-1}), & \text{caso contrário} \end{cases} \tag{3.14}$$

$$add(x) = \begin{cases} 0,3, & \text{se } x+0,005 > 0,3 \\ x+0,005, & \text{caso contrário} \end{cases} \tag{3.15}$$

$$div(x) = \begin{cases} 0,05, & \text{se } x-0,005 < 0,05 \\ x-0,005, & \text{caso contrário} \end{cases} \tag{3.16}$$

Os intervalos testados nos valores de α em cada uma das abordagens foi o mesmo, $[0.05, 0.3]$, para que se pudesse compará-las de forma justa.

3.9 Segurança

3.9.1 Arquitetura

Na arquitetura de um sistema de reputação três pontos podem ser alvos de ataques: a disponibilidade do sistema, os votos e as identidades. Cada uma delas possui vulnerabilidades específicas.

Uma forma de ataque à disponibilidade do sistema seria impedir que um participante, ou um grupo deles, tenha acesso à rede do sistema de reputação. Apesar do participante ter seus

serviços indisponíveis por determinado momento, a entrega dos votos não é crítica e pode ser feita em outro momento. O uso de certificados também possibilita a validação de uma reputação entregue pelo próprio serviço em uma comunicação direta. Essa ferramenta permite ao consumidor concluir o uso do serviço local mesmo sem acesso à dispositivos externos, tornando ineficaz essa forma de ataque.

Outra forma de ataque com foco na indisponibilidade do sistema, seria prejudicar ou inibir o funcionamento dos dispositivos que exercem a função de servidor de registro. Para isso, poderia-se usar métodos de DoS, como inundar o dispositivo com requisições falsas ou por envenenamento de cache. A arquitetura distribuída do sistema possui uma melhor resiliência contra ataques de DoS. Assim, com alguns dispositivos exercendo a função de servidor de registro, o custo de sobrecarregá-las torna-se muito alto e não impede que os participantes já conectados continuem funcionando normalmente. Contra os ataques de envenenamento de cache o sistema possui a autenticação provida pelo par de chaves pública-privada. Com elas, um servidor de registro pode validar o participante que requisita conectividade.

A emissão de votos falsos é outro ataque comum aos sistemas de reputação. Para limitar este tipo de ataque o sistema exige um arquivo criptografado por ambos os participantes da operação, votante e votado. Este arquivo é chamado de prova de interação. A ordem em que a cifras são feitas também define quem é o consumidor e quem é o provedor do serviço utilizado, podendo só existir um voto de cada um para esta interação. O dado criptografado é a data e hora da comunicação e os identificadores dos participantes. Não são aceitas provas de interação iguais em votos distintos e a data inserida na prova pode ser conferida com o valor enviado junto com voto, figura 3.2. Assim, o custo de emitir um voto falso é igual ao de roubar uma identidade, ou seja, adquirir um par de chaves pública-privada iguais ao do participante que se deseja atacar.

Ataque de roubos de identidade são feitos ou por raízes comprometidas, que permitem o atacante prever a chave próxima chave aleatória gerada, ou por roubo das chaves, no caso em que a segurança do dispositivo é comprometido permitindo que terceiros tenham acesso à chave privada, ou por força bruta, tentando criar as combinações possíveis para a chave privada e verificando se a chave pública gerada é igual. No primeiro, a escolha da raiz deve obedecer alguns critérios para que este tipo de ataque não ocorra, estes requisitos são determinados pelas organizações padronizadoras de segurança digital como a Certicom (CERTICOM, 2015). No segundo caso, cabe ao participante manter seu sistema seguro contra outras ameças que permitam o ter acesso ao seu sistema e demais dados. Já no caso da força bruta o tempo médio necessário para encontrar a chave desejada irá depender do tamanho de

chave escolhido, cabendo ao implementador do sistema balancear o tamanho de chave entre o poder computacional dos sistemas atuais para que possa ser usado em dispositivos móveis como *smartphones* mas também não seja fácil encontrar chaves por meio da força bruta.

Ainda se tratando de identidade, há outro tipo de ataque em que identidades falsas são geradas para serem usadas para autopromoção e/ou para difamação. Neste trabalhos, definimos identidade falsa como qualquer par de chaves publica-privada gerada com a intenção diferente da de consumo ou prestação de serviço. Este tipo de ataque utiliza um conjunto de identidades que interagem entre si, criando votos "válidos" para aumentar a reputação do provedor de serviço real. Para restringir a ação desse tipo de ataque é necessário criar uma formulação que exija algum esforço para se obter uma reputação considerada positiva e limitar os votos aceitos por um limiar, tornando assim o custo necessário para executar este ataque muito superior ao necessário para obter uma boa reputação por meios válidos.

3.9.2 Formulação

Algumas questões de segurança de um sistema de reputação são diretamente ligadas à formulação do sistema. Esta seção discute as principais vulnerabilidades e suas causas.

Um ataque comum ao sistemas de reputação é o "fresh start". Nele, participantes com reputações ruins as abandonam e adquirem uma nova identidade. Sistemas em que o esforço necessário para se obter uma reputação boa por uma nova identidade é menor do que o esforço de necessário para recuperar uma reputação ruim favorecem a viabilidade deste ataque. Formulações que utilizam a média simples dos valores dos votos como reputação estão particularmente vulneráveis a este tipo de ataque. Nelas é possível adquirir reputação máxima com apenas um voto e conforme a quantidade de votos aumenta mais difícil é alterar o valor da média. Portanto, se a média é baixa, torna-se vantajoso abondar a reputação e começar novamente com uma nova identidade.

Por outro lado, formulações que apresentam o cenário inverso, em que adquirir uma boa reputação é muito difícil para novas identidades, tornam-se pouco atrativas para novos participantes. O esforço para adquirir uma reputação competitiva com os participantes já estabelecidos no sistema é muito alto, sendo preferível aliar-se aos participantes que já possuem boa reputação. Essa vulnerabilidade é chamada de "cold start".

Outra questão relacionada à formulação são os ataques de curto período. Este tipo de ataque possui este nome pois o participante deliberadamente utiliza-se da sua boa reputação e da pouca influência que poucos votos negativos exercem sobre ela, para oferecer serviços ruins por um

curto período de tempo ou para poucos participantes. Ele é caracterizado pela sua periodicidade. Uma possível solução seria punir de forma mais rigorosa participantes com boa reputação que recebem votos negativos, porém isso deve ser feito de forma cautelosa para não punir de forma muito agressiva participantes que passam por dificuldades momentâneas. É preciso criar formas de identificar esse tipo de comportamento.

Por último, há os ataques de promoção e difamação feitos por coligações. Estes são os ataques mais elaborados e difíceis de serem detectados. Nele, um grupo de participantes une-se para emitir votos para uma determinada identidade com a finalidade de promovê-la ou difamá-la. As coligações podem ser caracterizadas por participantes que relacionam-se somente entre eles e periodicamente emitem votos falsos destinados ao alvo do ataque. Por serem organizados podem burlar facilmente os mecanismos de detecção criados para encontrar coligações mudando periodicamente os comportamentos adotados.

Todas essas questões devem ser levadas em consideração durante a criação de uma formulação e serão usadas para avaliar as formulações propostas. Boas práticas envolvem equilibrar o esforço necessário para que todos os participantes possam adquirir uma boa reputação sem a necessidade de usar métodos inadequados e prevenir ataques da melhor forma possível.

Capítulo 4

RESULTADOS

Os métodos de análise utilizados neste trabalho foram divididos em: (1) análise das formulações e de suas curvas de reputação e (2) análise do modelo arquitetural e da comunicação entre os dispositivos. O primeiro, focado nas formulações, avalia a forma como a reputação de um dispositivo é formada em alguns dos cenários comuns a um sistema baseado em serviços. O segundo avalia a escalabilidade do modelo de comunicação e da arquitetura no cálculo e na disseminação das reputações. Para avaliar ambos os aspectos, foram utilizadas simulações que representam o deslocamento e a comunicação entre dispositivos. Os encontros entre dispositivos na simulação podem gerar uma interação que resultaria em votos e, consequentemente, na alteração de suas reputações. Essa forma de análise permite simular curvas de reputação mais próximas da realidade, pois a interação torna-se ocasional, e também analisar a convergência das informações conforme o número dispositivos de infraestrutura aumenta.

4.1 Formas de Avaliação

A avaliação do sistema de reputação exige uma grande quantidade de votos e dispositivos para que seja possível analisar as curvas geradas e, também, a forma como o sistema dissemina as informações (votos e reputações). Para que testes reais fossem realizados, seria necessário implementar todo o sistema, garantindo a segurança das informações, e utilizar um grande número de usuários e serviços dispostos a integrar o protocolo proposto ao seu procedimento padrão. Além disso, caso comprovado um sistema ineficiente, todo este esforço seria inapropriado para a fase inicial do trabalho.

Desta maneira, optou-se por simular um ambiente de comunicação e de mobilidade dos nós em que são representados os dispositivos, com seus papéis definidos, e a infraestrutura

necessária. Pode-se, assim, variar livremente o número de dispositivos simulados e os ambientes dos testes. Isto permite analisar o comportamento do sistema de reputação e das formulações em um maior número de situações e com um maior controle dos agentes.

Os simuladores, por sua vez, precisavam apresentar dois aspectos básicos do sistema de reputação proposto: a mobilidade dos dispositivos e a comunicação entre eles. O primeiro é necessário para representar o deslocamento dos dispositivos móveis, o que possibilita a interação com múltiplos serviços e o segundo, para representar a transmissão dos dados de forma realística.

Entre os simuladores mais utilizados nos testes de protocolos de redes estão: ns-2/3, OMNet++, OPNET, GloMoSim e J-Sim. Estudos (Koeksal Miran Murat, 2008) (ATTA; BILAL; OTHMAN, 2012) apontam que ns-2/3 e OMNet++ apresentam melhores representações do meio físico e, portanto, geram dados mais realísticos, além de resultados parecidos quando comparados. Enquanto ns-2/3 apresenta melhor desempenho durante as simulações, OMNet++ possui melhor interface de desenvolvimento. Devido à quantidade de classes e módulos necessários para a implementação deste trabalho, escolheu-se utilizar o simulador OMNet++ para simular a comunicação entre os dispositivos.

Para uma melhor representação da mobilidade nos testes desenvolvidos, optou-se por utilizar um segundo simulador específico para esta questão. O simulador SUMO (HILBRICH, 2015) apresenta todas as características exigidas para os testes previstos, possibilitando a importação de mapas e flexibilidade na criação dos destinos dos objetos simulados dentro deste mapa. Sua integração com o simulador OMNet++ também é possível com o *framework* Veins (SOMMER; GERMAN; DRESSLER, 2011).

4.2 Simulação de Mobilidade e Comunicação

O *framework* de simulação utilizado nas análises deste trabalho, Veins (SOMMER; GERMAN; DRESSLER, 2011), é formado pela integração de dois outros simuladores: SUMO (HILBRICH, 2015), responsável por simular o deslocamento dos dispositivos, e OMNet++ (LTD., 2015), responsável pela simulação da comunicação entre eles.

OMNet++ é um simulador discreto; seu funcionamento se dá por uma lista de eventos ordenados pelo tempo de ocorrência. A lista é percorrida simulando os eventos que desencadeiam ações e a passagem do tempo. Entre as possíveis ações estão o envio de mensagens, o processamento de dados e a criação de outros eventos futuros.

Assim, para simular o comportamento de dispositivos e suas interfaces de rede, são usados módulos para gerar e processar eventos que determinam quando e como as transmissões irão ocorrer. Se um evento de transmissão é disparado no simulador, pacotes são gerados de acordo com a pilha de protocolos selecionados (UDP, IP, Wi-Fi, etc.). Um evento correspondente ao recebimento do pacote é então criado e inserido na fila, levando em consideração os tempos médios de transmissão para o cenário em questão.

As funcionalidades do simulador devem ser implementadas na forma de módulos e cada módulo é composto de duas partes. A primeira descreve as propriedades de um módulo, a quais outros módulos este está ligado, quais submódulos ele possui e como este submódulos estão interligados. Dessa forma, se a aplicação de um dispositivo deseja transmitir dados em pacotes UDP por Wi-Fi, a implementação do módulo referente ao dispositivo deve conter o submódulo da interface de comunicação sem fio, o submódulo IP, o submódulo UDP e o submódulo da aplicação. Estes submódulos devem estar conectados na ordem de execução definida pela pilha de protocolos da Internet. Os pacotes transmitidos seguem pelos submódulo que executam as funções dos protocolos até chegar à interface de transmissão. Então, é criado um evento para a transmissão do pacote na rede. Quando o evento de transmissão é disparado, o simulador verifica quais módulos estão aptos a receber a mensagem e cria um evento de recebimento para cada um deles. Ao ser disparado o evento de recebimento, os dados seguem o caminho inverso até o submódulo de aplicação, que os processa e cria os eventos decorrentes do recebimento da mensagem.

As ações tomadas por um módulo são descritas na segunda parte de sua implementação. Nesta seção são implementados, em classes C++, os recebimentos das mensagens e os processamentos aplicados sobre elas. Também nessa seção são criados os eventos que serão inseridos na lista utilizando a API disponibilizada pelo simulador.

Ambas as partes estão ligadas pelo nome do módulo e da classe criada, que devem ser iguais, sendo possível definir valores de variáveis na primeira e recuperá-las na segunda. Como somente as classes são compiladas, é possível alterar os parâmetros para cada execução na definição dos módulos, sem a necessidade de recompilar as classes.

Em um ambiente com vários dispositivos, cabe ao simulador de comunicação levar em consideração as localizações de cada um deles em um cenário alvo considerado na simulação e as capacidades de sinalização das interfaces simuladas para determinar quais deles receberão os pacotes a eles destinados. É nesse aspecto que o modelo de mobilidade provido pelo simulador SUMO é relevante nos testes.

SUMO foi desenvolvido para simular o deslocamento de pessoas e veículos sobre um

determinado cenário, levando em consideração detalhes como semáforos, calçadas, quantidade de faixas de uma rua, velocidade máximas das vias e a própria interação com outras pessoas e veículos simulados. Ele é capaz, por exemplo, de simular a ultrapassagem de um veículo mais lento ou a parada em uma faixa de pedestres para que as pessoas simuladas atravessem a rua.

Os cenários utilizados por esse simulador são descritos em arquivos XML, podendo representar ruas, calçadas e rodovias. Neste também são especificadas as características desses componentes (comprimento, formato, localização, velocidade máxima, etc.) e as interligações entre eles.

As rotas que os veículos e pessoas simuladas percorrem também são detalhados por documentos XML. Neles, estão presentes: o tempo de inserção, o tipo de objeto (pessoa ou veículo) e a sequência das vias que serão utilizadas. É necessário que os caminhos descritos sejam possíveis e que todas as conexões entre ruas e calçadas estejam declaradas no arquivo do cenário.

A execução dos simuladores que compõem o *framework* Veins é alternada. O simulador SUMO executa o primeiro intervalo de tempo, posicionando os dispositivos iniciais e locomovendo-os pelo cenário. Terminado seu intervalo de execução, repassa o posicionamento de cada dispositivo ao simulador OMNet++. Este recebe os dados, atualiza as posições no seu contexto e simula a transmissão das mensagens. Ao finalizar, requisita ao primeiro simulador as posições no próximo intervalo de tempo.

4.2.1 Aprimoramentos nos Simuladores

Algumas modificações foram necessárias nos simuladores para que os dados sobre as pessoas simuladas no SUMO pudessem ser interpretadas pelo OMNet++. Até a versão 0.21.0 do SUMO, a implementação da simulação de pessoas estava completa, porém não havia interface disponível para obtenção dos dados que seriam necessários no simulador OMNet++. Usualmente, os dados do SUMO são obtidos por outras aplicações pelo módulo TraCI (*Traffic Control Interface*). Este módulo disponibiliza os dados da simulação por meio de comandos solicitados a um *socket*. Como dito, até a versão utilizada neste trabalho, os comandos referentes à obtenção dos dados das pessoas simuladas ainda não haviam sido implementados, sendo necessário modificar o módulo TraCI para que estes dados estivessem disponíveis ao simulador de comunicação.

Modificações nos módulos do OMNet++ que implementam a comunicação com o TraCI também foram necessárias para requisitar e tratar os dados recebidos pelo SUMO. Assim, os

dados de inserção, movimentação e remoção de pessoas de simulações no SUMO poderiam ser aplicados também às simulações do OMNet++.

Personalizações foram implementadas para que as pessoas representadas no simulador OMNet++ pudessem receber identificadores (prefixos) e módulos diferentes dos usados pelos veículos, possibilitando distingui-los durante a simulação e diferenciar seus comportamentos.

Módulos	Simulador
TraCIMobility	OMNet++
TraCIScenarioManager	OMNet++
TraCIScenarioManagerLaunchd	OMNet++
TraCIServer	SUMO
TraCIServerAPI_Person	SUMO

Tabela 4.1: Lista de módulos alterados em cada simulador

4.2.2 Implementações e aspectos das simulações

No simulador OMNet++, foram implementados módulos referentes aos papéis desempenhados pelos dispositivos no sistema de reputação proposto. Cada módulo implementado corresponde a uma das quatro funções descritas na seção 3.4.

O módulo de servidor de registro atua de forma passiva durante a simulação. Os demais dispositivos possuem uma lista de servidores e escolhem de forma aleatória com qual servidor de registros irão estabelecer comunicação. Ao receber uma requisição, um servidor de registro armazena o endereço IP, o nome e o tipo (verificador completo, verificador simples ou minerador) do dispositivo requisitante e responde com uma lista de mineradores e verificadores completos que se registraram anteriormente pelo mesmo processo. Um dispositivo, ao receber uma lista de endereços, envia uma solicitação de conexão para confirmar a viabilidade da comunicação. Se o número de verificadores contactados é inferior a um limiar estabelecido na simulação, este dispositivo escolhe outro servidor de registro e solicita uma nova relação de mineradores e de verificadores completos. O processo de descoberta e contato prossegue até que um número adequado de verificadores completos e mineradores tenham sido contactados.

O módulo de verificador simples representa os provedores e os consumidores na simulação. São instâncias deste módulo que se deslocam pelo cenário simulado e interagem entre si para gerar os votos que serão usados nos cálculos das reputações. Uma instância de verificador simples pode desempenhar tanto o papel de consumidor quanto o de provedor de serviços e a execução de ambos os papéis é definida por parâmetros no inicio da simulação.

As interações entre os dispositivos simulados iniciam-se por meio de mensagens enviadas

em *broadcast*. Tais mensagens são enviadas por provedores de serviço para todos os dispositivos ao alcance de sua rede *Ad-hoc*. Estas mensagens de oferta são enviadas em intervalos fixos e somente por provedores de serviços. Um consumidor, ao receber uma destas mensagens, requisita o serviço disponibilizado por um provedor, informando um valor no intervalo $[0,1]$ que representa a qualidade da interação que ele realizaria neste caso. O provedor, ao receber a requisição, também responde com o valor da qualidade estabelecida para esta prestação de serviço.

Os valores da qualidade do serviço provido e da interação realizada por um cliente são definidos por uma lista de parâmetros. Nesta lista, cada entrada contém a probabilidade de ocorrência e o intervalo para sorteio da qualidade como cliente ou servidor. A cada interação com outro dispositivo, um número aleatório é sorteado para definir a faixa de qualidade de serviço que será usada. Um novo sorteio é então realizado para obter o valor efetivo dentro da faixa selecionada. Este último valor é encaminhado junto com as mensagens de requisição e resposta de serviços e correspondem à qualidade da interação oferecida. Posteriormente, será utilizado como valor do voto.

Por exemplo, um nó poderia ter 60% de probabilidade de prover serviços no intervalo de $[0.8,1]$, 30% de probabilidade de prover serviços no intervalo de $[0.6,0.8]$ e 10% de probabilidade de prover serviços no intervalo $[0.4,0.6]$. Assim, durante uma interação, se o número sorteado estivesse no intervalo $[0,0.6]$ um valor entre $[0.8,1]$ representariam a sua qualidade nesta interação. Caso o valor sorteado estivesse entre $]0.6,0.9]$ a qualidade seria entre $[0.6,0.8]$ e se fosse maior que $0,9$, entre $[0.4,0.6]$.

Uma vez que o escopo desse trabalho limita-se a estudar a evolução da reputação gerada pelas interações e não o desempenho da oferta e do consumo de serviços, os demais dados transmitidos no protocolo descrito na seção 3.6, como prova de interação e descrição de serviços não foram representados durante a simulação. Contudo, em uma implementação real, seriam de extrema importância para a segurança e a confiabilidade das reputações.

Terminada a interação, ambos os dispositivos simulados geram seus votos utilizando os valores de qualidade de interação recebidos e os encaminham para os mineradores e verificadores completos conhecidos. Caso não possuam em sua lista de contatos nenhum dispositivo destas categorias, há duas possibilidades: enviar uma requisição aos servidores de registro em busca de novas conexões ou transmitir o voto por *broadcast* aos demais dispositivos próximos, para que estes, ao terem acesso à rede, encaminhem o voto. Quando o dispositivo não possui conexão com a *Internet* ele opta por encaminhar os votos para os demais verificadores simples; caso contrário, insiste na busca por novas conexões com

verificadores completos e mineradores. Um estudo sobre a convergência de dados neste cenário foi realizado por (DECKER; WATTENHOFER, 2013).

A conexão com a *Internet* se dá por pontos de acesso distribuídos aleatoriamente. Nas simulações foram utilizados 40 pontos de acesso com alcance de 100 metros de distância, o que corresponde aproximadamente a 1.25 km^2 de área com acesso à rede da *Internet*. Nestes pontos de acesso os dispositivos poderiam estabelecer comunicação com servidores de registros para adquirir os endereços dos mineradores e verificadores completos e transmitir os votos armazenados.

Com a finalidade de analisar o comportamento do cálculo das reputações sob diversos ambientes, foram implementadas duas funcionalidades adicionais nos módulos que representam os dispositivos. A primeira é a possibilidade de inserir uma lista diferente de possíveis valores para a qualidade de interação, com a finalidade de validar as formulações em ambientes de ataques de curto período de tempo. Assim, o dispositivo simulado que executa este tipo de ataque alterna a lista usada para calcular a qualidade de interação usando os tempos especificados nos parâmetros do módulo. Para que este ataque tenha efeito, uma das listas deve ter altas probabilidades de uma boa qualidade de interação, valores superiores de 0.5, e a outra altas probabilidades de uma baixa qualidade, valores inferiores à 0.5, possibilitando verificar a detecção desses dispositivos em relação ao tempo de prestação de serviços de baixa qualidade.

Os dois outros módulos implementados são responsáveis pela infraestrutura de armazenamento e de manutenção dos sistema de reputação. Os verificadores completos são pontos de entrada para os votos, além de armazenar o histórico completo de interações, de blocos e de reputações gerados no sistema. Ao iniciar uma instância desse módulo, incia-se o processo de requisição aos servidores de registro até que o número de contatos mínimos seja atingido. Posteriormente, encaminha todos os blocos e votos recebidos aos demais dispositivos conectados.

O minerador implementa todas as funcionalidade de um verificador completo, além de, em intervalos específicos, tentar gerar um bloco com base em uma probabilidade definida. Caso isso ocorra, seleciona um conjunto de votos não computados e gera um bloco. As reputações são recalculadas e o bloco é encaminhado aos dispositivos conhecidos, que também o encaminham pelas suas conexões.

O tempo necessário para a disseminação completa de votos e de blocos foi utilizado como métrica de análise para compreender quais variáveis influenciam no tempo de convergência dos dados. Assim, mediu-se o tempo necessário para que todos os dispositivos de infraestrutura

fossem atualizados quando um bloco é criado ou um novo voto recebido.

Existe a probabilidade de mais de um minerador simulado gerar um bloco; neste caso, eles irão competir para definir qual bloco permanecerá na corrente principal, corrente mais longa, e qual será descartado. O minerador que gerar o próximo bloco irá definir esta concorrência pois utilizará um dos blocos gerados anteriormente como base. Ao disseminar este novo bloco pela rede, os dispositivos irão adotar a corrente mais longa, descartando os blocos divergentes das correntes mais curtas. Os votos dos blocos descartados voltam para o conjunto de votos ainda não computados, tendo o cuidado de remover os já utilizados na corrente principal.

4.3 Métodos de Análise

4.3.1 Análise das Formulações e dos Cálculos

A implementação das funcionalidades do sistemas de reputação no simulador possibilitou observar tanto o comportamento das formulações quanto sua escalabilidade. Para a análise das formulações foram coletados os votos e os blocos criados durante a simulação. Os provedores de serviços foram divididos igualmente em cinco grupos de qualidade. Nesses grupos, os valores dos votos têm média de $0,8$ e $0,65$ para os dois grupos de maior qualidade, $0,5$ para o grupo mediano e $0,35$ e $0,2$ para os grupos de qualidade inferior, como mostrado tabela 4.2. Isso possibilitou observar a curva da reputação para cada um desses comportamentos e comparar as formulações quanto à rapidez de crescimento e de declínio, comportamentos oscilatórios e quedas na reputação.

Nome	Média	Lista de Parâmetros
Grupo 1	0.80	$[0.6, 1]$ probabilidade 1
Grupo 2	0.65	$[0.6, 1]$ probabilidade 0.5 $[0.4, 0.6]$ probabilidade 0.5
Grupo 3	0.50	$[0.6, 1]$ probabilidade 0.25 $[0.4, 0.6]$ probabilidade 0.5 $[0, 0.2]$ probabilidade 0.25
Grupo 4	0.35	$[0.4, 0.6]$ probabilidade 0.5 $[0, 0.2]$ probabilidade 0.5
Grupo 5	0.20	$[0, 0.4]$ probabilidade 1

Tabela 4.2: Lista probabilidades por grupo

Os valores de α também foram analisados nas formulações. Para a abordagem fixa foram testados os valores: 0,05, 0,10, 0,15, 0,20, 0,25 e 0,30. Para as abordagens de α variáveis, linear e escalar, foram modificados os valores do limiar. Estes valores indicam qual a variação

na reputação necessária para que o valor de α aumente ou diminua conforme as equações apresentadas na seção 3.8.6. Os valores testados para o limiar foram os mesmos utilizados na abordagem fixa.

Ataques de curto período foram simulados com a finalidade de verificar a capacidade das formulações na detecção desse tipo de comportamento. Para isso, além dos grupos apresentados, foram introduzidos outros cinco grupos. Os dispositivos pertencentes a esses grupos se comportavam como os grupos descritos anteriormente durante um determinado período, porém por algum tempo ofereciam serviços de baixa qualidade, com média de $0,1$. Dessa forma, para cada grupo anterior foi criado um grupo em que os serviços oferecidos não correspondiam ao habitual durante o intervalo especificado, caracterizando ataques de curto período. Os valores de tempo testados em que serviços de baixa qualidade eram oferecidos foram: 5, 10, 20, 35, 50, 55, 80, 100 e 110 segundos. O tempo em que o provedor oferecia serviços normalmente permaneceu constante em 50 segundos.

Os intervalos de tempo definidos para os ataques de curto período foram escolhidos devido ao tempo total de simulação realizado, para que se pudesse observar o comportamento de oscilação da curva em diversos períodos. Tomando como base o maior tempo de simulação, 9 mil segundos, e o tempo máximo de duração do ciclo de ataque, 160 segundos, ocorreriam aproximadamente 56 oscilações de comportamento no melhor caso. Contudo, os dispositivos são inseridos ao longo da simulação o que pode acarretar em quantidades variadas de ciclos para cada um deles.

Com estes dois testes foi possível observar o comportamento de cada formulação para os grupos de qualidade de serviços oferecidos e também a influência de ataques de curto período. Assim, com a finalidade de comprovar a eficiência das formulações analisadas na detecção de provedores que oferecem serviços de baixa qualidade, verificamos quais interações seriam evitadas no caso dos consumidores possuírem uma lista com as reputações dos provedores conhecidos pelo sistema. A lista é atualizada sempre que o consumidor conecta-se à rede do sistema de reputação ou envia votos aos mineradores e aos verificadores completos.

Definiu-se então que interações com provedores de serviço com reputação inferior a $0,6$ seriam evitadas. Portanto, monitorou-se quantas interações com dispositivos dos grupos 4 e 5 foram realizadas e quantas poderiam ser evitadas. No caso dos ataques de curto período, os grupos que oscilavam seu comportamento também foram monitorados com a finalidade de averiguar como as reputações refletiam este comportamento e em quais circunstâncias as interações poderiam ser evitadas.

Nesse cenário existe a possibilidade de descartar interações com provedores de reputação

inferior a $0,6$ que ainda não atingiram uma reputação correspondente à qualidade do seu serviço por falta de interações positivas, fenômeno que é chamado de "*cold start*". O número de interações que foram negligenciadas para estes dispositivos dos grupos de alta qualidade também foi monitorado. Isto possibilitou analisar a quantidade de interações necessárias em cada uma das formulações para que este fenômeno não seja um problema real.

Para esses dois casos de testes utilizou-se o mapa do centro da cidade de São Carlos, cobrindo uma área de $5km^2$. Dois cenários foram feitos usando este mapa, um com 150 pessoas e 392 veículos e outro com 300 e 871, respectivamente. Para cada um dos dispositivos foi gerada uma rota escolhendo duas vias aleatórias e a menor sequência de vias possível que completasse o caminho de um ponto ao outro. Tanto os carros quanto as pessoas simuladas possuem dispositivos capazes de oferecer e consumir serviços e, durante o trajeto, o encontro desses dispositivos possibilitava a comunicação entre eles.

4.3.2 Análise da Escalabilidade

Os testes de escalabilidade foram feitos comparando o número de servidores de nome e de mineradores, a quantidade mínima de conexões estabelecidas entre eles e de contatos enviados a cada requisição aos servidores de nomes. Foi possível observar o grau médio de conexões entre os mineradores e também o tamanho do maior caminho necessário para a atualização de um bloco. Os valores testados para cada uma das variáveis estão na tabela 4.3.

Servidores de nome	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Mineradores	10, 20, ..., 90, 100, 200, ..., 1000
Grau mínimo de conexões	1 a 10
Quantidade de Contatos por requisição	1, 3, 5, 7, 9

Tabela 4.3: Lista de valores usados no teste de escalabilidade do sistema

4.4 Simulações

4.4.1 Estudo das Reputações

Os resultados preliminares das simulações serviram para averiguar o comportamento dos simuladores e da implementação para o problema proposto. Nos gráficos a seguir, podemos observar a curvas geradas em cada uma das formulações para os grupos de qualidade.

Os gráficos apresentam dois dispositivos com os mesmos parâmetros iniciais e desempenham a função de provedores de serviços. A variação das reputações, dentro de um

mesmo grupo, são causadas pela diferença no número de interações, pelo momento em que elas ocorreram e pelo tamanho do intervalo dos valores de qualidade.

Os gráficos 4.1, 4.2, 4.3, 4.4 e 4.5 exibem o comportamento de dois dispositivos pertencentes ao grupo 1 de reputações, com média $0,8$, ao longo do tempo de simulação para as formulações descritas na seção 3.8.

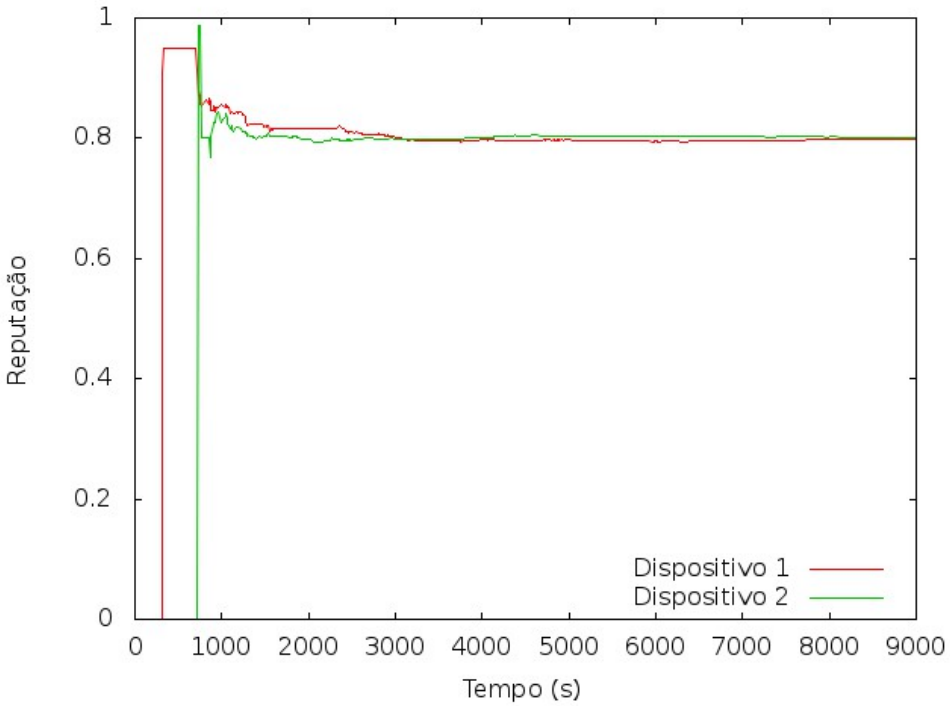

Figura 4.1: Grupo 1: Média

Como exibido na figura 4.1, em que o cálculo da reputação é feito pela média, percebe-se que o crescimento rápido das reputações logo nas primeiras interações, além de oscilações significantes durante as primeira interações. Porém, ao longo do tempo a curva se torna cada vez mais constante.

A figura 4.2 mostra os mesmos dispositivos quando utilizada a média móvel no cálculo de suas reputações. Percebemos que ela atinge o mesmo patamar da média simples, porém, com uma curva mais suave e com variações constantes ao longo do tempo. Percebe-se também uma oscilação na reputação do dispositivo 1 e a dificuldade em aproximar novamente sua reputação da qualidade média de seus serviços.

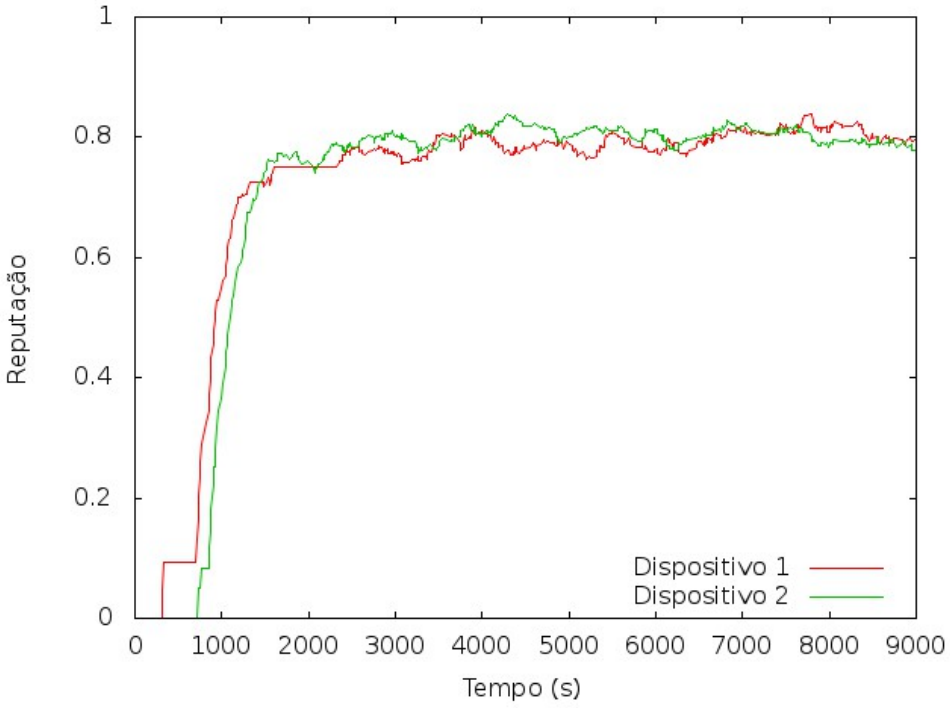

Figura 4.2: Grupo 1: Média móvel exponencial

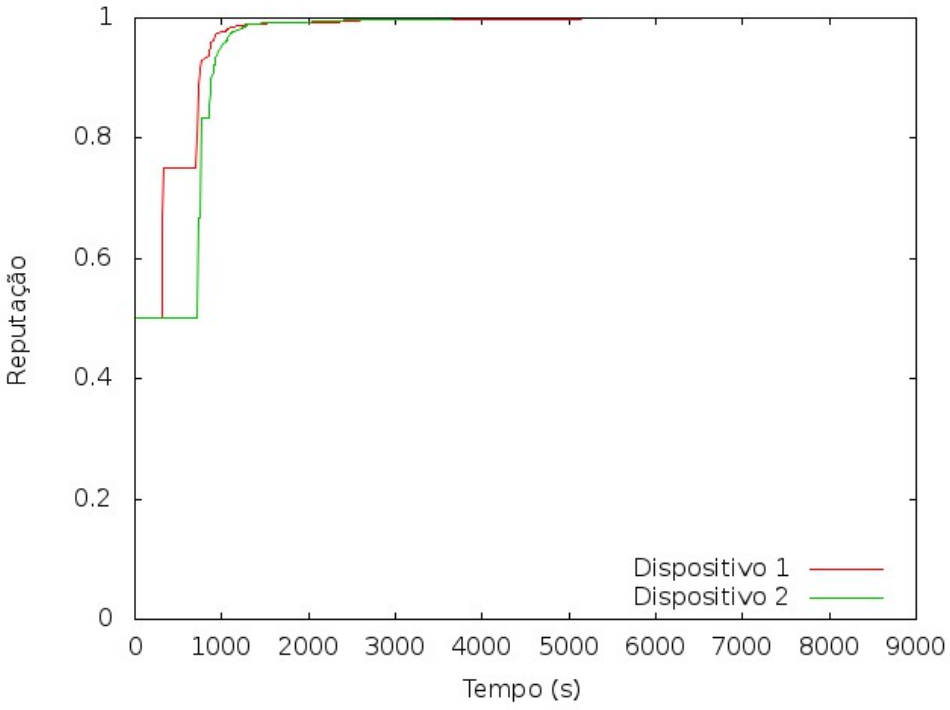

Figura 4.3: Grupo 1: Esperança da distribuição beta

As reputações pelo cálculo com a esperança da distribuição beta são apresentadas na figura 4.3. Sua curva atinge a reputação máxima com alguns votos e permanece neste patamar. Como sua formulação transforma o valor do voto em duas categorias, positivo ou negativo, quando

uma reputação recebe apenas votos positivos atinge a reputação máxima.

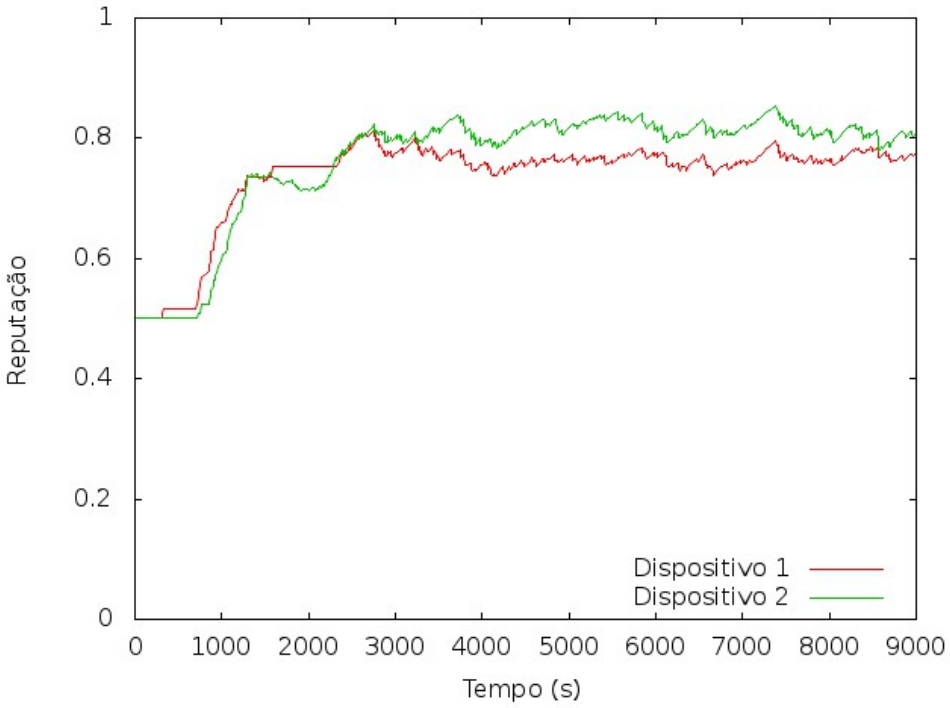

Figura 4.4: Grupo 1: Esperança móvel da distribuição beta

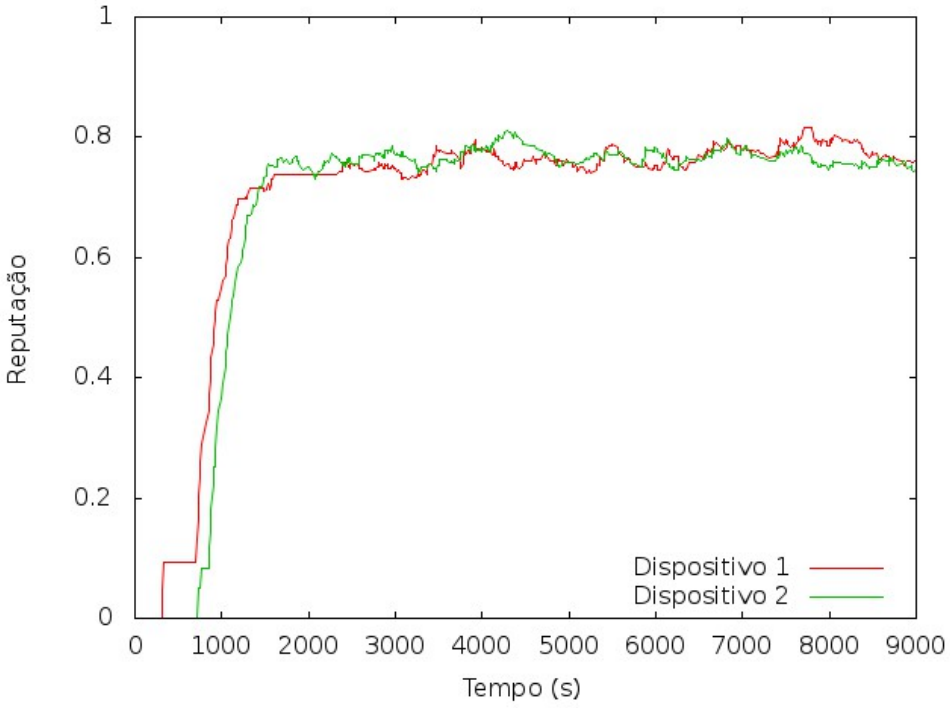

Figura 4.5: Grupo 1: Proposta de formulação apresentada neste trabalho

A formulação da esperança móvel comporta-se de forma mais parecida com média móvel

do que com a esperança comum, figura 4.4. O crescimento da curva é ainda mais lento que a da média móvel e as oscilações são um pouco mais intensas.

Por fim, a formulação proposta apresenta comportamento parecido com a das formulações móveis nas reputações de provedores de serviço, figura 4.5. Mais detalhes podem ser encontrados na seção 3.8.5.

Nos gráficos 4.6, 4.7, 4.8, 4.9 e 4.10 estão representadas reputações do grupo com média de qualidade de serviço prestado $0,65$.

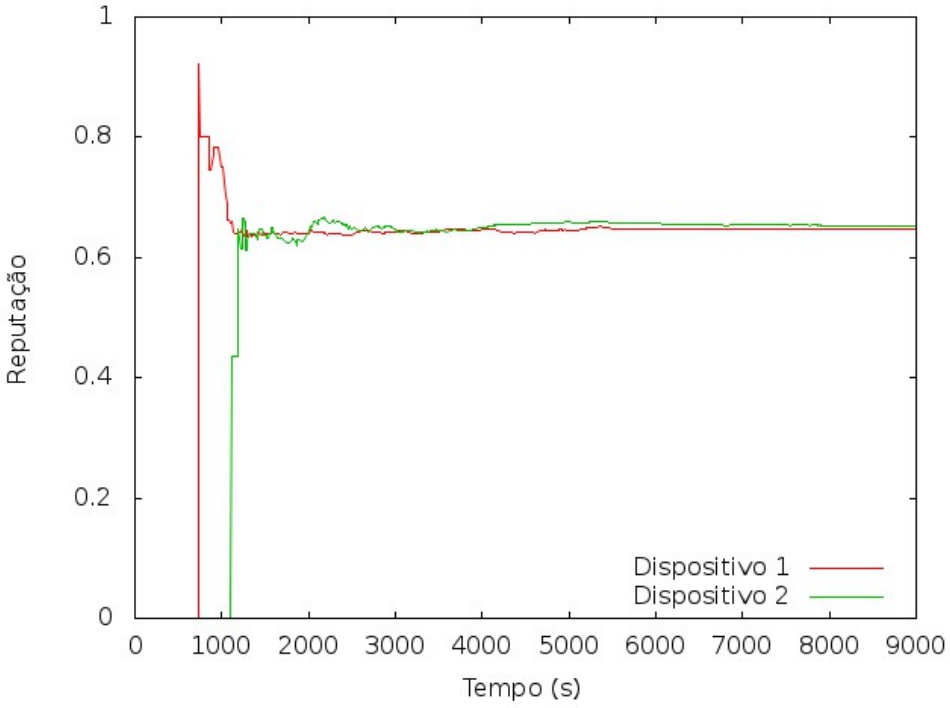

Figura 4.6: Grupo 2: Média

As mesmas características observadas no primeiro gráfico desta formulação também podem ser reconhecidas na figura 4.6, assim como acontece nas demais formulações.

A partir dos 6 mil segundos, figura 4.7, observa-se uma linha contínua na reputação do dispositivo 1. Este comportamento na curva indica que não houve votos durante este período. Este comportamento se repete em todas as formulações, pois sem voto não há atualizações das reputações.

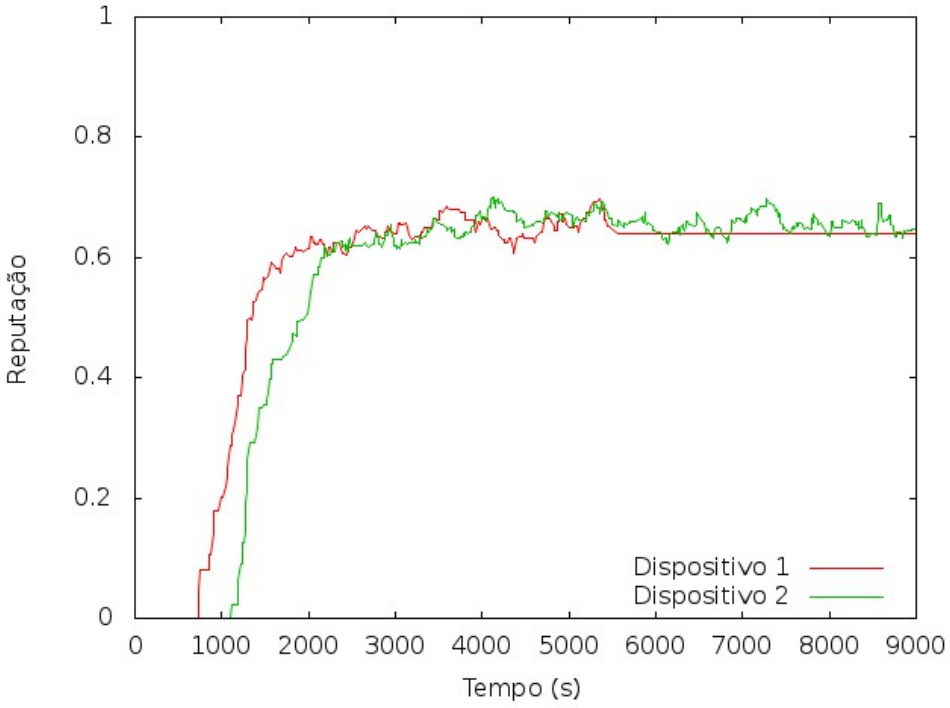

Figura 4.7: Grupo 2: Média móvel

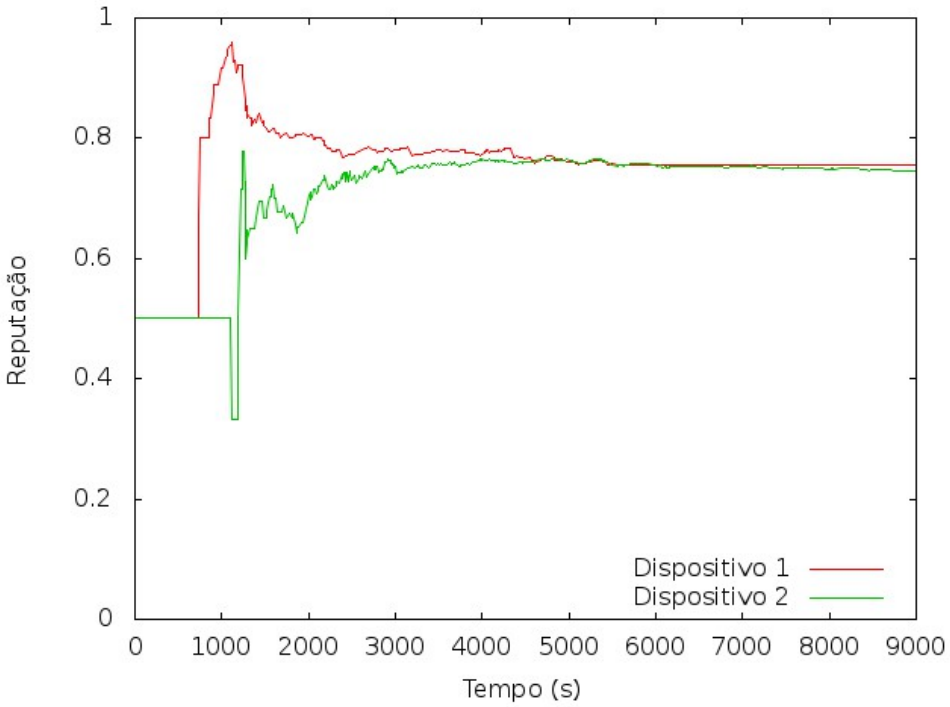

Figura 4.8: Grupo 2: Esperança da distribuição beta

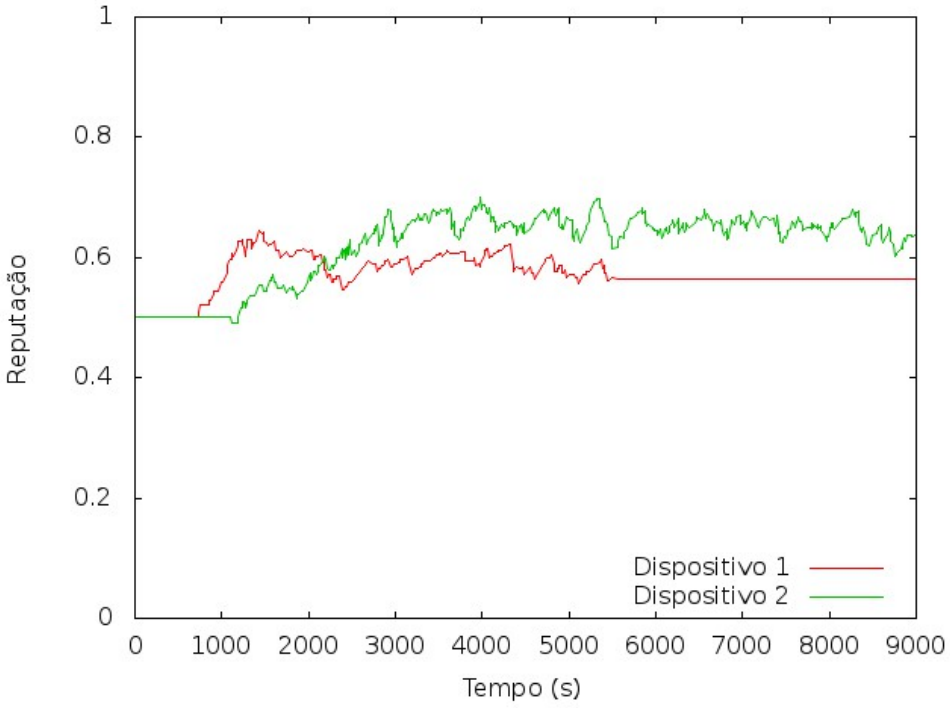

Figura 4.9: Grupo 2: Esperança móvel da distribuição beta

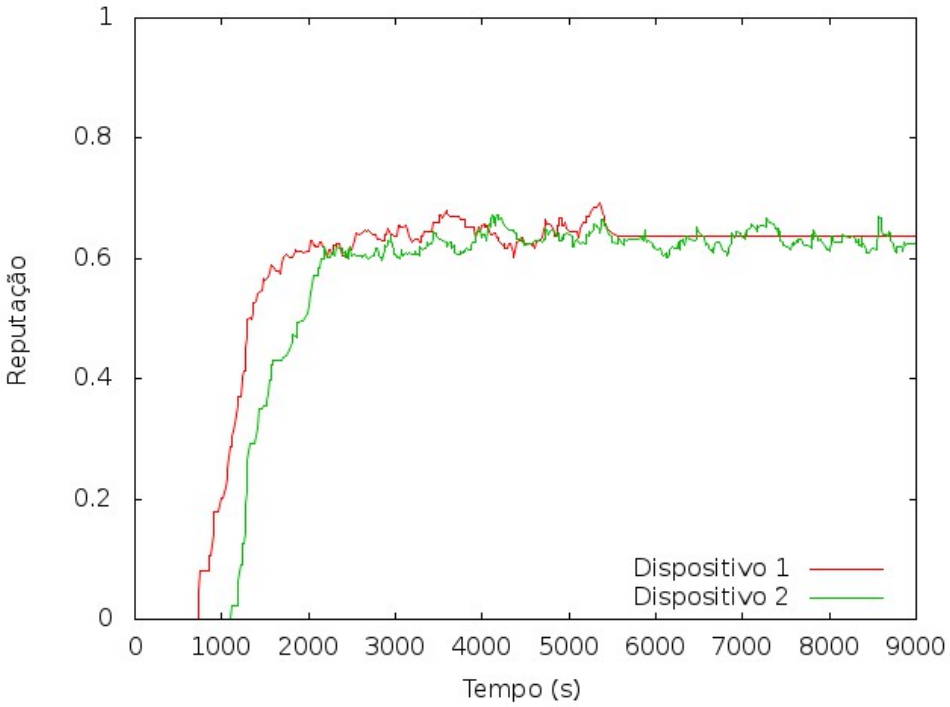

Figura 4.10: Grupo 2: Proposta de formulação apresentada neste trabalho

Comparando as figuras 4.9 e 4.10 observamos que as oscilações na formulação proposta são menores do que na esperança móvel mesmo quando utiliza-se os mesmos valores para α.

Na análise das figuras 4.6 a 4.10 observa-se que, para um dispositivo com comportamento estável, a formulação da média e da esperança da distribuição móvel reflete de forma melhor essa característica. Isto pode ser verificado com as curvas sem oscilações e em torno dos valores médios esperados. Por outro lado, as outras formulações refletem mais precisamente o comportamento mais recente de um dispositivo.

Podemos observar que em todos os casos as reputações se comportaram dentro dos valores esperados considerando uma pequena variação dada pela aleatoriedade dos valores gerados. O comportamento dos demais grupos podem ser conferidos no apêndice A, seção A.1.

Os gráficos que exibem reputações da formulação de média simples chegam mais rápido ao patamar do grupo de qualidade em que o dispositivo se encontra. Isso porque logo nos primeiros votos, ela já assume os valores médios. A formulação de esperança demora um pouco mais para chegar ao patamar estabelecido e pode ultrapassá-lo, chagando aos extremos da reputação, caso o dispositivo não possua votos variados. Por fim, os gráficos da esperança móvel da distribuição beta, da média móvel e da formulação proposta apresentam oscilações mesmo após chegar ao patamar referente ao grupo de qualidade, pois votos recentes possuem maior peso no cálculo dessas reputações.

Essas diferenças ficam mais claras nas figuras 4.11, 4.12, 4.13, 4.14 e 4.15, em que são exibidas as curvas de reputação de um mesmo conjunto de votos calculados em cada uma das formulações.

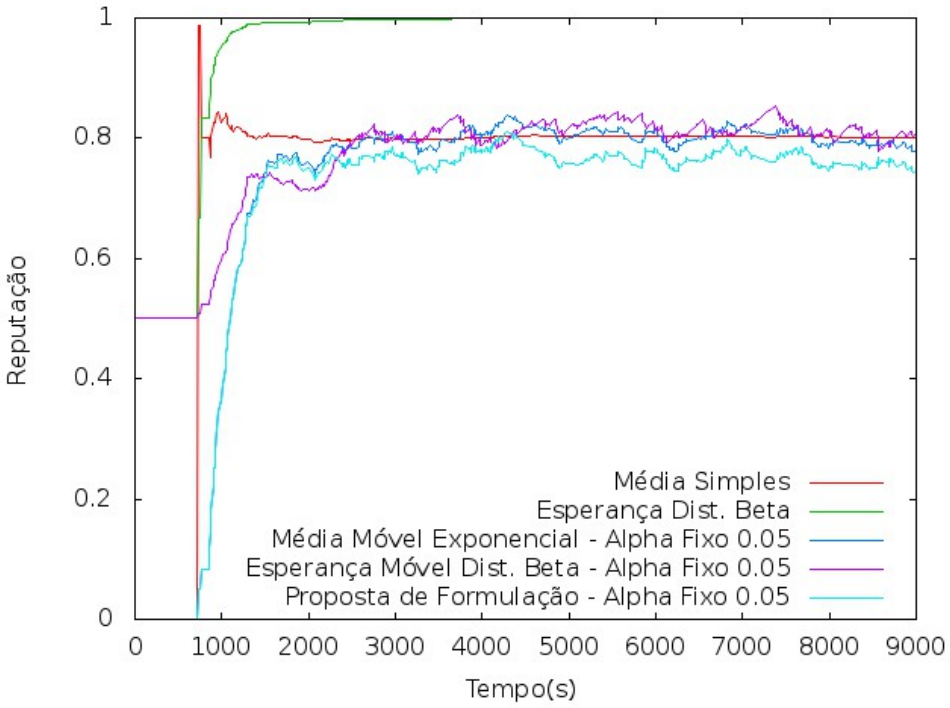

Figura 4.11: Exemplo das curvas de reputação para o grupo 1 em cada formulação analisada

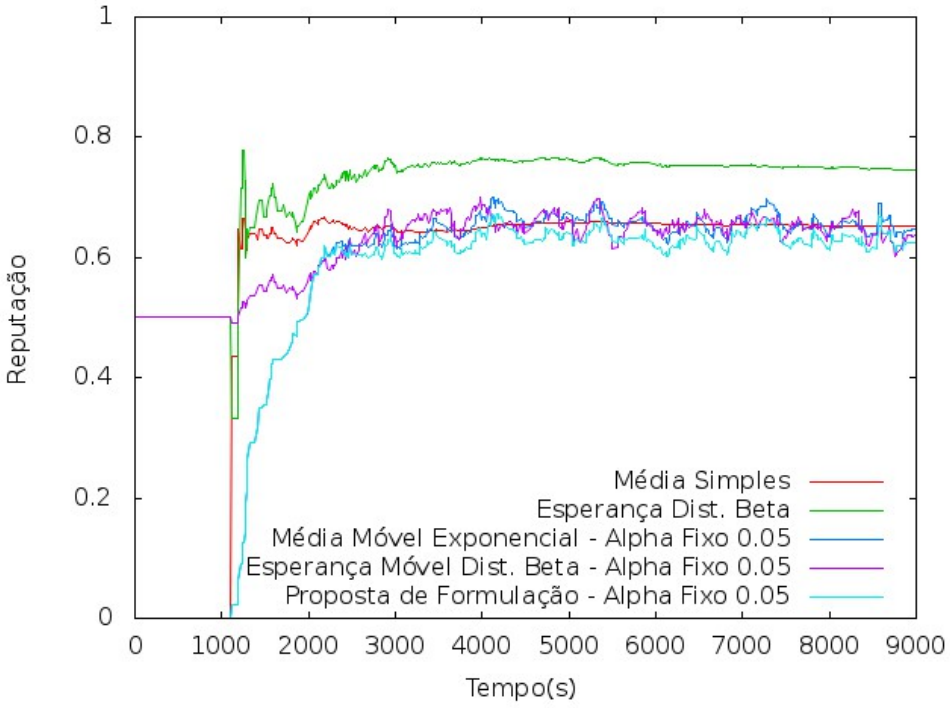

Figura 4.12: Exemplo das curvas de reputação para o grupo 2 em cada formulação analisada

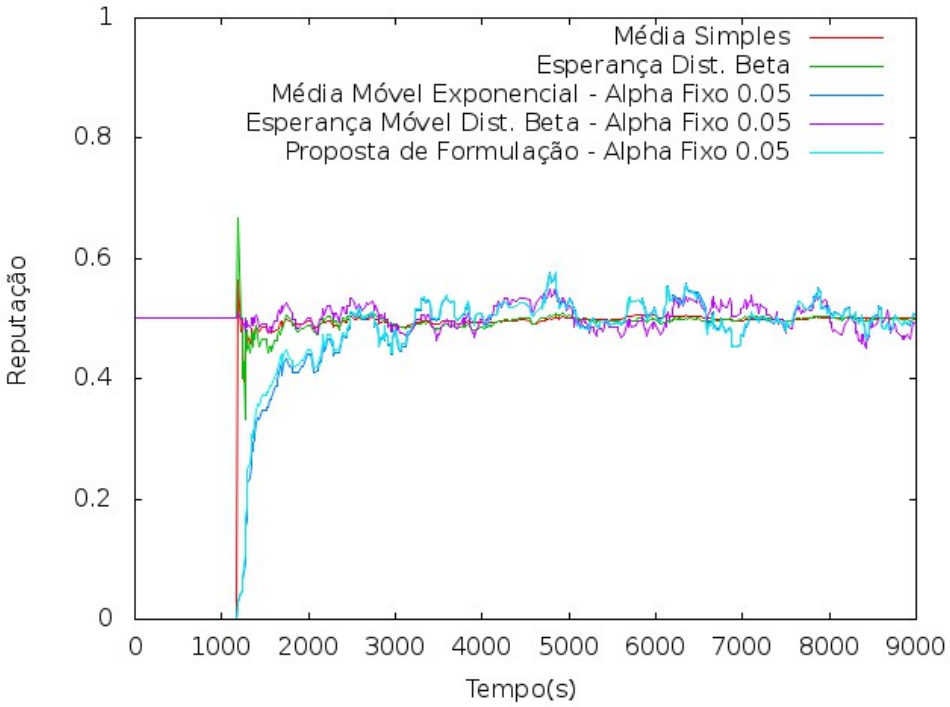

Figura 4.13: Exemplo das curvas de reputação para o grupo 3 em cada formulação analisada

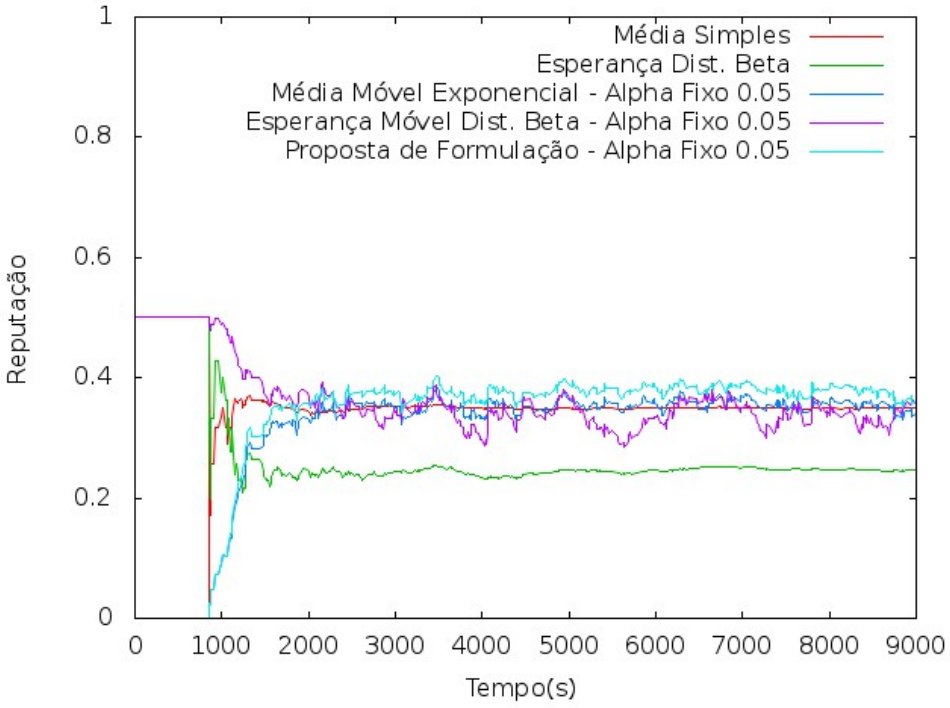

Figura 4.14: Exemplo das curvas de reputação para o grupo 4 em cada formulação analisada

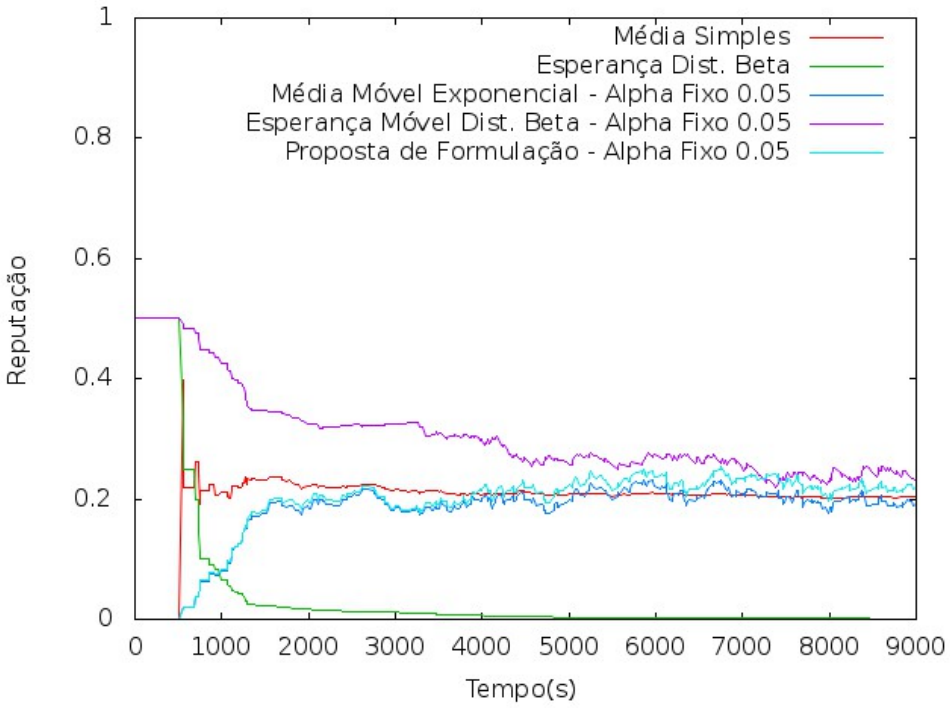

Figura 4.15: Exemplo das curvas de reputação para o grupo 5 em cada formulação analisada

Para as diferentes faixas de qualidade as formulações se comportaram de forma esperada, sendo a principal diferença a pouca representação do comportamento recente na média e na esperança da distribuição beta. Isto implica uma dificuldade na alteração do valor da reputação

depois de decorrido algum tempo.

Comprovado o comportamento do simulador e das formulações, é possível, então, avaliar cada uma de forma mais objetiva. Uma estratégia é verificar a porcentagem das interações com provedores de baixa qualidade que foram recusadas devido ao sistema de reputação.

Considerando o tipo de dispositivo e a predominância do serviço que oferece, pode-se classificá-lo com comportamento médio superior ou inferior ao limiar (0.6). Assim, uma formulação terá um bom desempenho quando um dispositivo com reputação inaceitável é classificado como tal.

Outra métrica é a quantidade de interações recusadas que seriam feitas com provedores de qualidade boa. Neste caso, analisa-se a possibilidade de *cold start* verificando a porcentagem de detecções erradas.

O gráfico 4.16 demonstra a porcentagem de interações de baixa qualidade que foram detectadas utilizando cada uma das formulações. É visível o melhor desempenho das abordagens que dão maior relevância ao votos mais recentes, sendo 7% mais eficientes nas simulações com duração de 3600 segundos e chegando a 10% nas simulações de 9000 segundos.

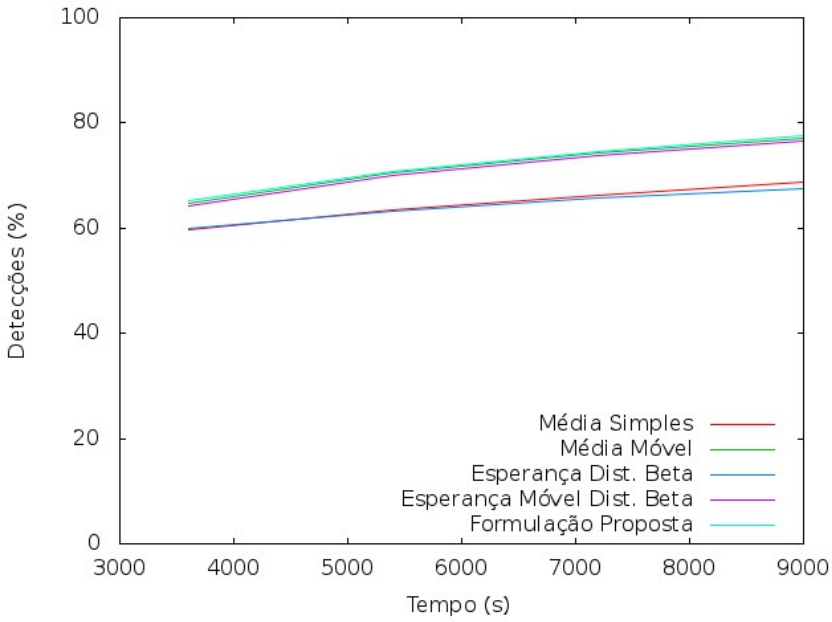

Figura 4.16: Porcentagem de detecções de interações ruins por tempo de simulação

Por outro lado, todas as formulações obtiveram números muito próximos de detecções válidas, ou seja, o sistema de reputação, ao sugerir que o dispositivo recuse uma interação, esta realmente iria ocorrer com um provedor de serviço de baixa qualidade. Isso fica claro pelo

gráfico da figura 4.17 em que as porcentagens de detecções válidas pelo tempo de simulação são próximas em todas as curvas.

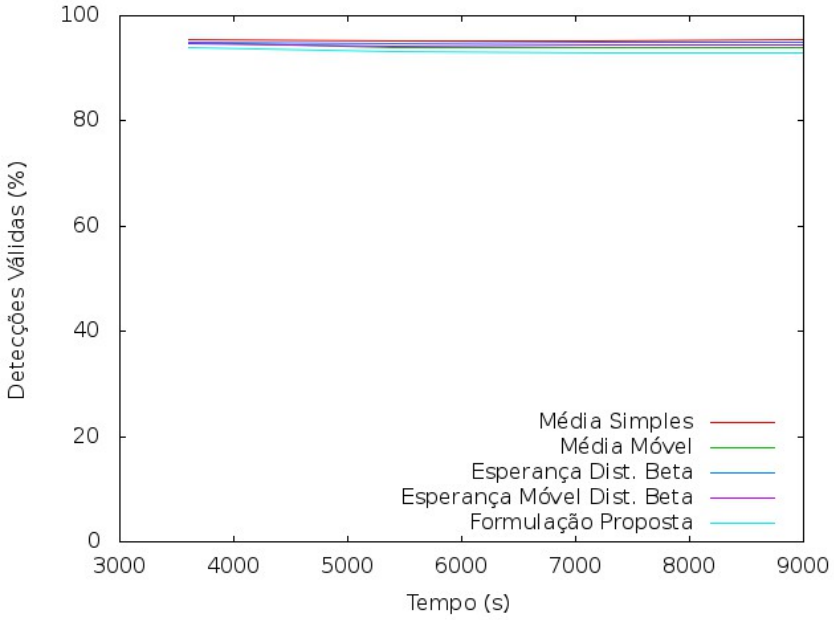

Figura 4.17: Exemplo das curvas de reputação para o grupo 5 em cada formulação analisada

Os gráficos a seguir exemplificam melhor este aspecto, mostrando a porcentagem de detecções válidas quando determina-se um número de atualizações iniciais em que as reputações são consideradas "novas". Enquanto as reputações não atingem este número, mesmo que elas sejam menores que o limiar estabelecido, o sistema de reputação incentiva a comunicação com estes provedores de serviço. Como as atualizações ocorrem somente quando um bloco é formado, utilizou-se o número de blocos que atualizam uma determinada reputação como unidade de medida para esta análise.

As figuras 4.18, 4.19, 4.20, 4.21 e 4.22 mostram esta porcentagem ao longo do tempo para cada uma das formulações: média simples, média móvel exponencial, esperança da distribuição beta, esperança móvel da distribuição beta e formulação proposta neste trabalho, respectivamente.

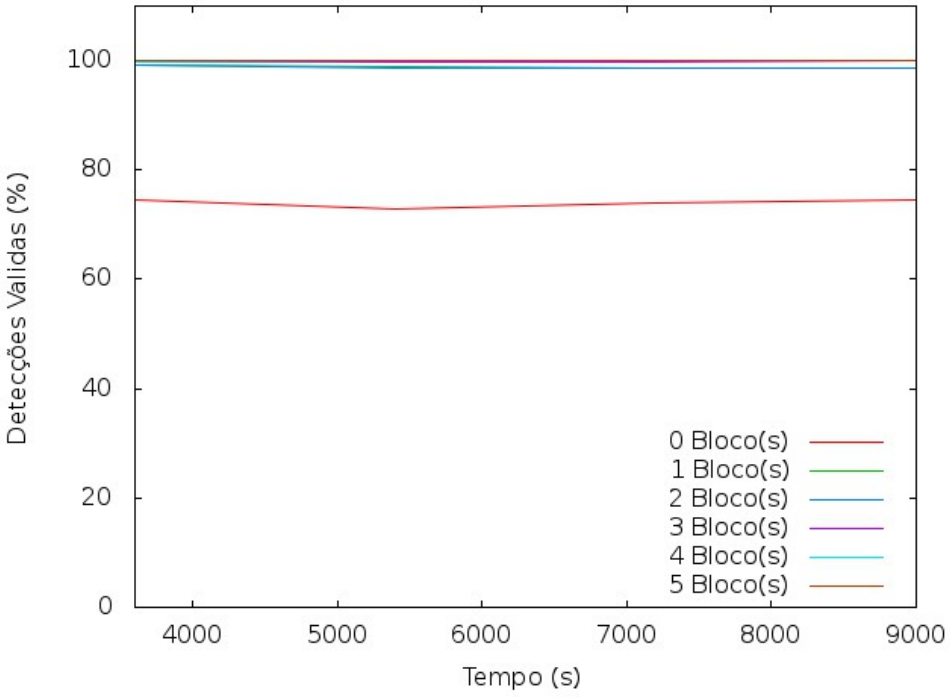

Figura 4.18: Média: Detecções válidas x Tempo

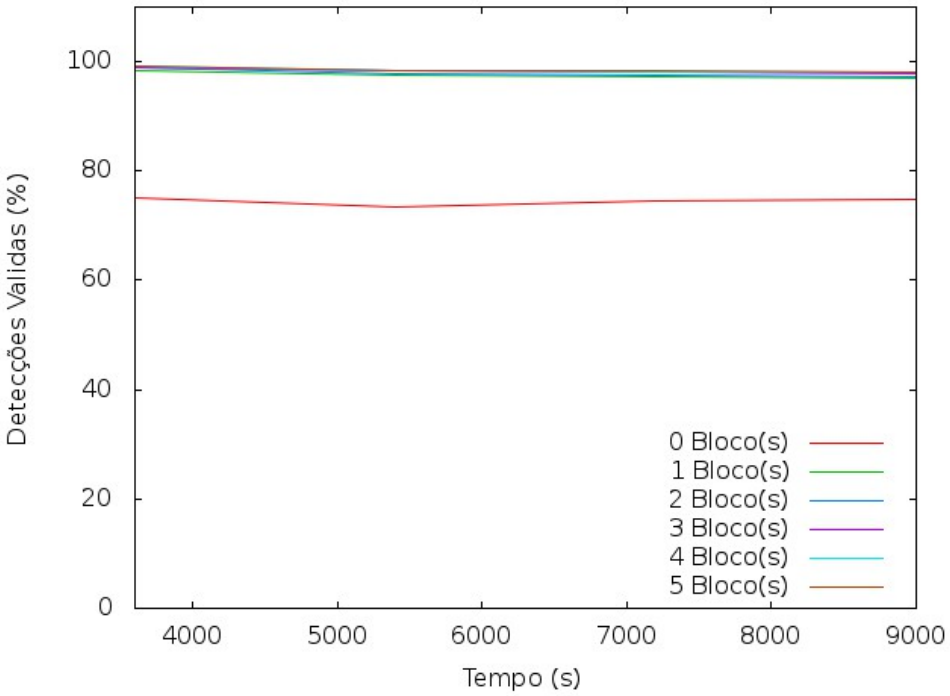

Figura 4.19: Média Móvel Exponencial: Detecções válidas x Tempo

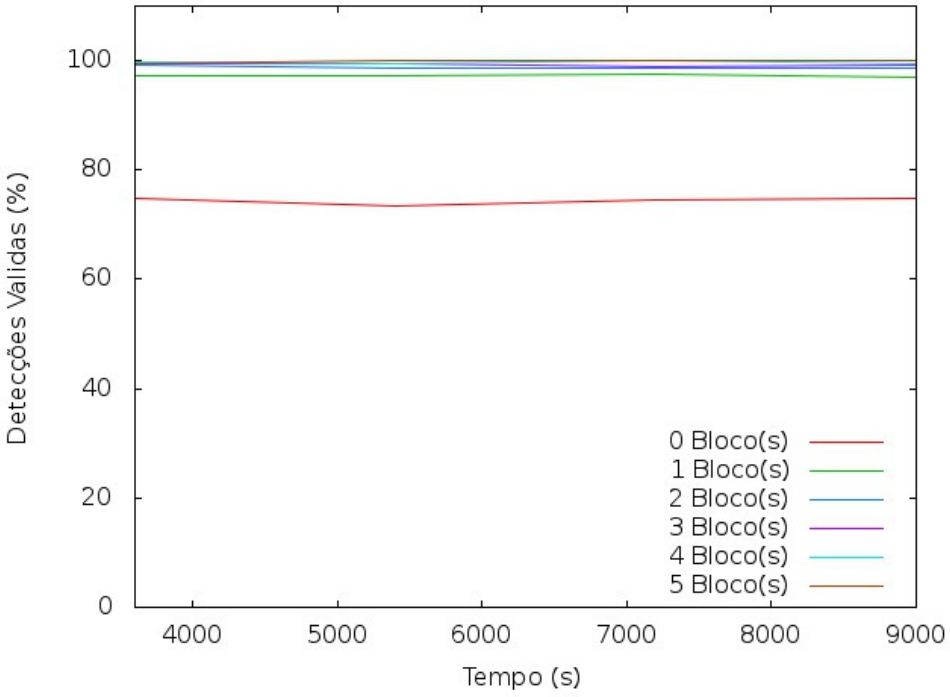

Figura 4.20: Esperança Dist. Beta: Detecções válidas x Tempo

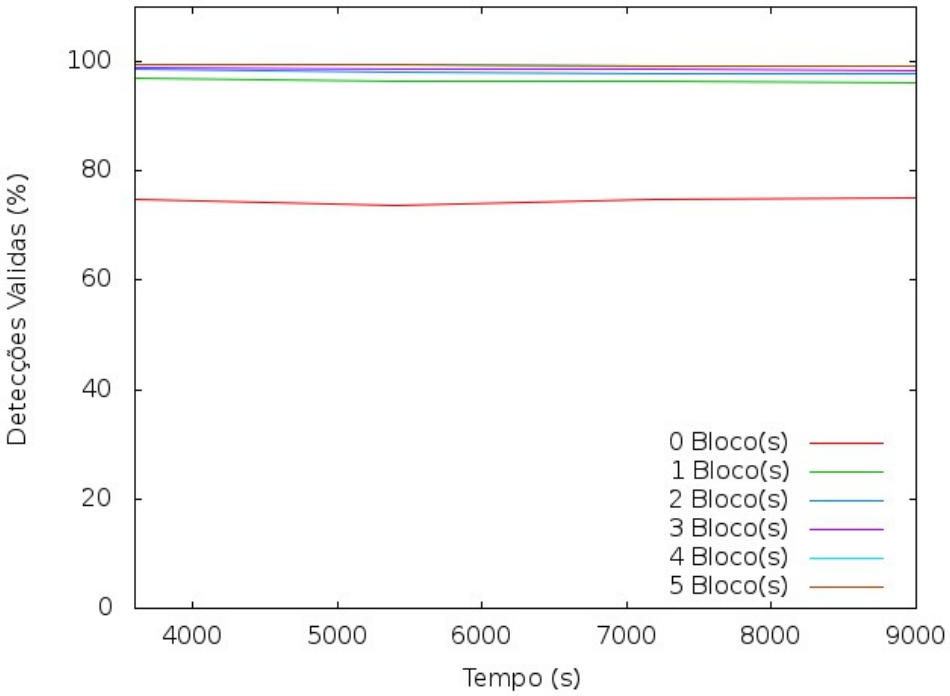

Figura 4.21: Esperança Móvel Dist. Beta: Detecções válidas x Tempo

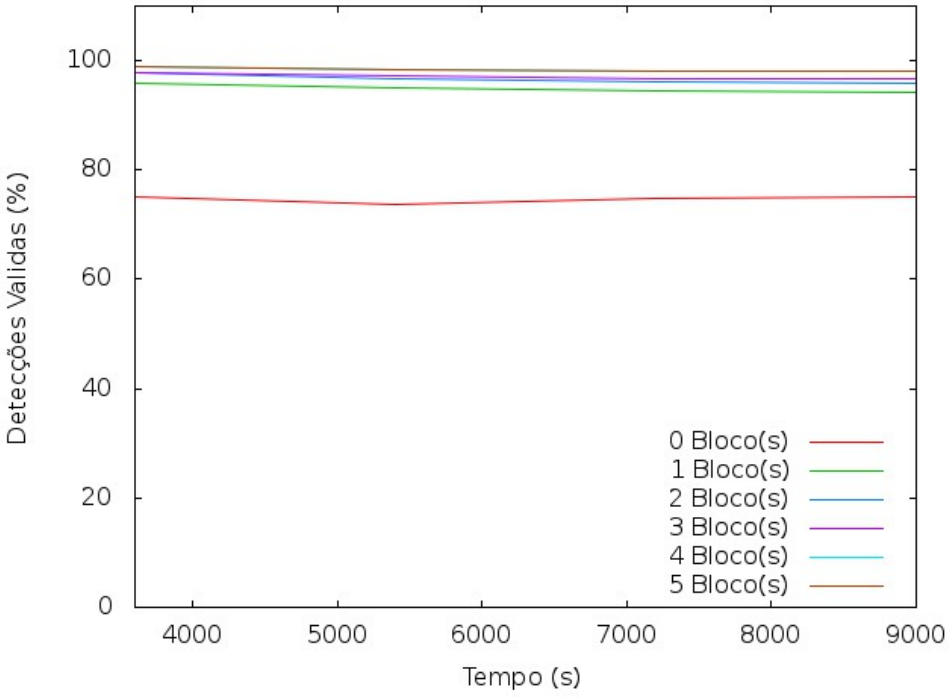

Figura 4.22: Formulação Proposta: Detecções válidas x Tempo

As figuras de 4.18 a 4.22 mostram que quanto mais blocos são utilizados antes de aplicar-se o limiar e classificar o serviço, maiores são as chances do sistema de reputação efetuar uma boa decisão. Quando não é utilizado nenhum intervalo de tempo, a acurácia dos sistemas de reputação é menor. Contudo, a porcentagem de detecções válidas cresce pouco quando utilizados mais que dois intervalos de tempo. Assim, pelos gráficos apresentados, conclui-se que com apenas um intervalo de tempo as detecções tornam-se suficientemente eficazes, evitando o "*cold start*".

Outro ponto relvante é a forma como valor de α interfere nas reputações móveis. Como nenhuma das três formulações que utilizavam esta abordagem apresentaram grande destaque sobre as demais, cabe analisar de forma aprofundada as diferenças entre os métodos utilizados no cálculo de α.

O valor de α é o peso atribuído aos valores recentes, ou seja, valore maiores tornam a reputação mais suscetível às oscilações. Nas figuras a seguir são apresentadas as curvas de reputação separadas por valores de α em cada uma das abordagens (fixo, linear e escalar) e por formulações.

As figuras 4.23, 4.24, 4.25 exemplificam a influência dos valores de α na formulação da esperança móvel da distribuição beta.

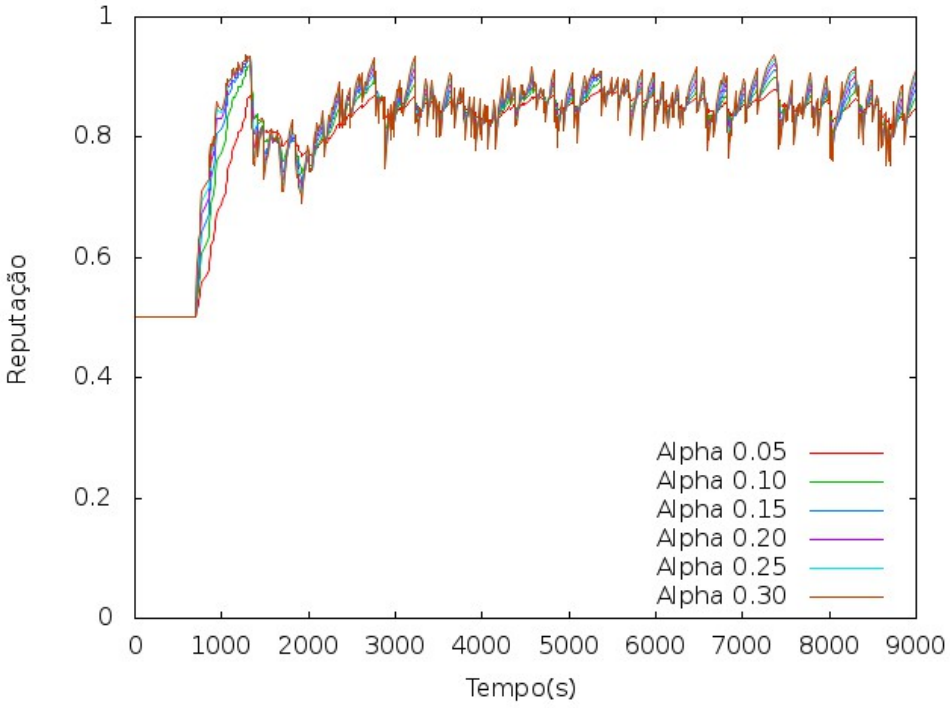

Figura 4.23: Exemplo das curvas de reputação para o grupo 1 utilizando formulação da esperança móvel da distribuição beta com α fixo

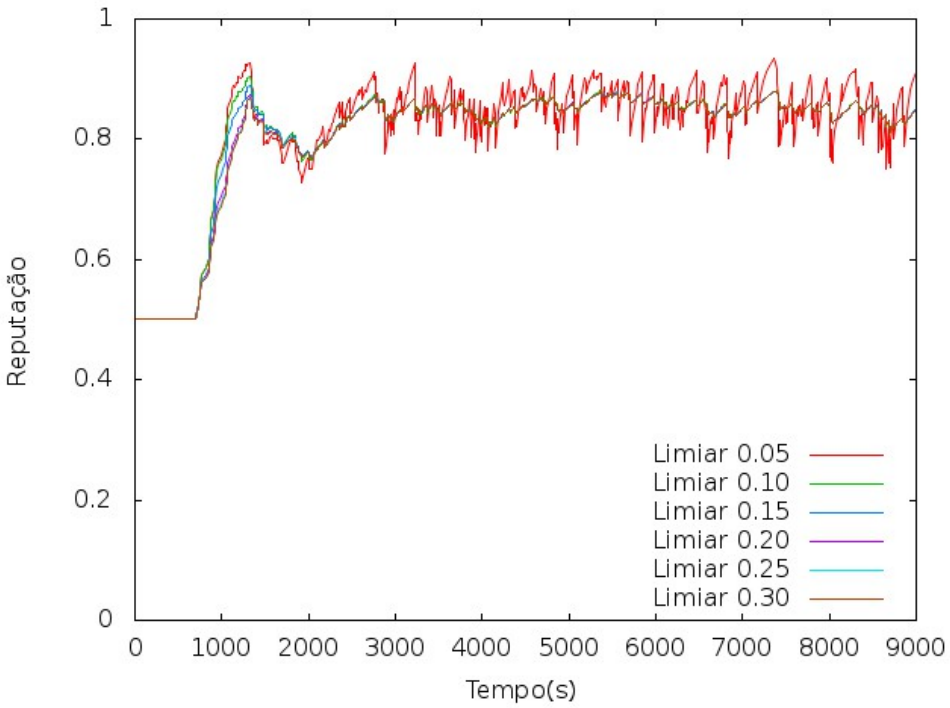

Figura 4.24: Exemplo das curvas de reputação para o grupo 1 utilizando formulação da esperança móvel da distribuição beta com α linear

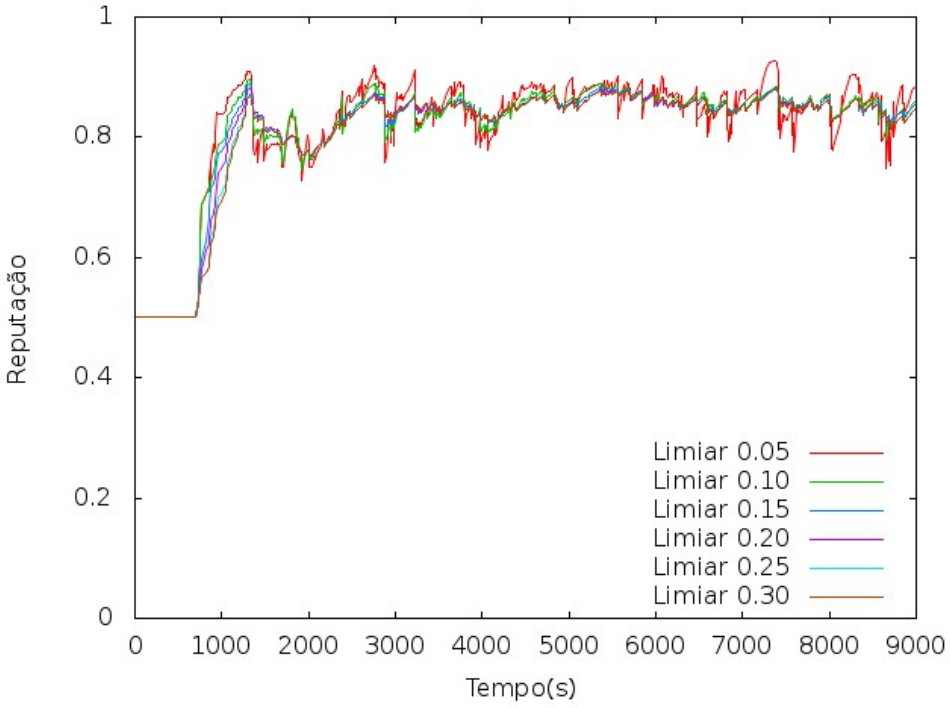

Figura 4.25: Exemplo das curvas de reputação para o grupo 1 utilizando formulação da esperança móvel da distribuição beta com α escalar

É possível perceber que valores de α fixo causam uma maior oscilação da reputação. Os outros dois métodos, linear e escalar, tornam a curva mais suave durante a maioria do tempo. Porém, em alguns momentos, ressaltam o comportamento atual, principalmente quando o valor do limiar é baixo. Isto pode ser observado nas figura 4.24 e 4.25 entre os tempos 7 e 8 mil segundos, quando ocorre um pico na curva de reputação com limiar $0,05$.

As figuras 4.26, 4.27 e 4.28 exibem a reputação do mesmo dispositivo, porém utilizando a média móvel exponencial. Nesta formulação, os valores de α tendem a ser menos impactantes e no caso das abordagens linear e escalar, a convergir ao longo do tempo.

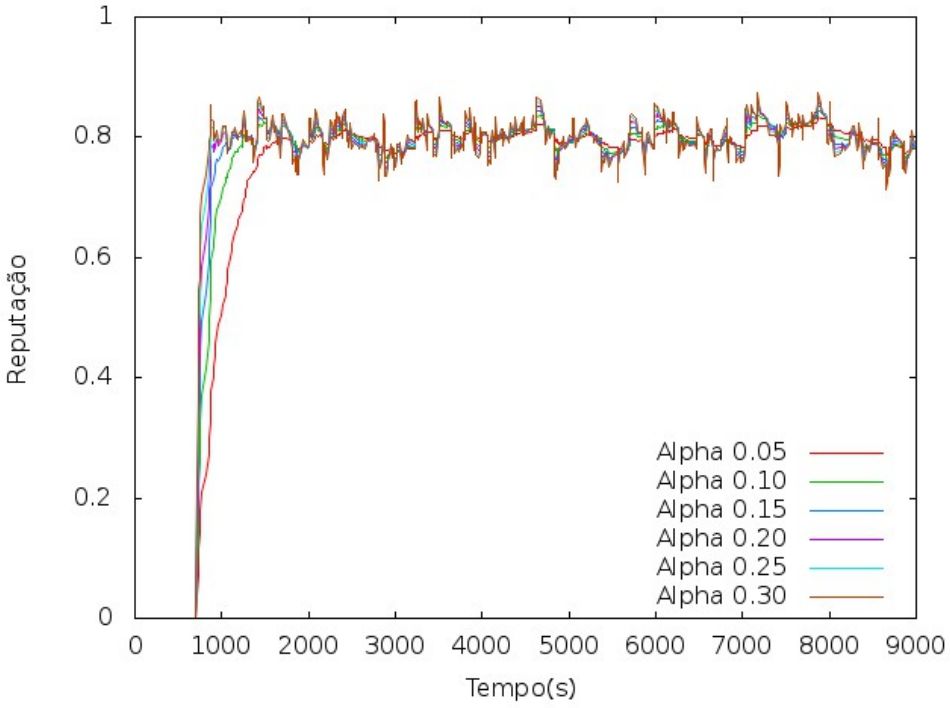

Figura 4.26: Exemplo das curvas de reputação para o grupo 1 utilizando formulação da média móvel exponencial com α fixo

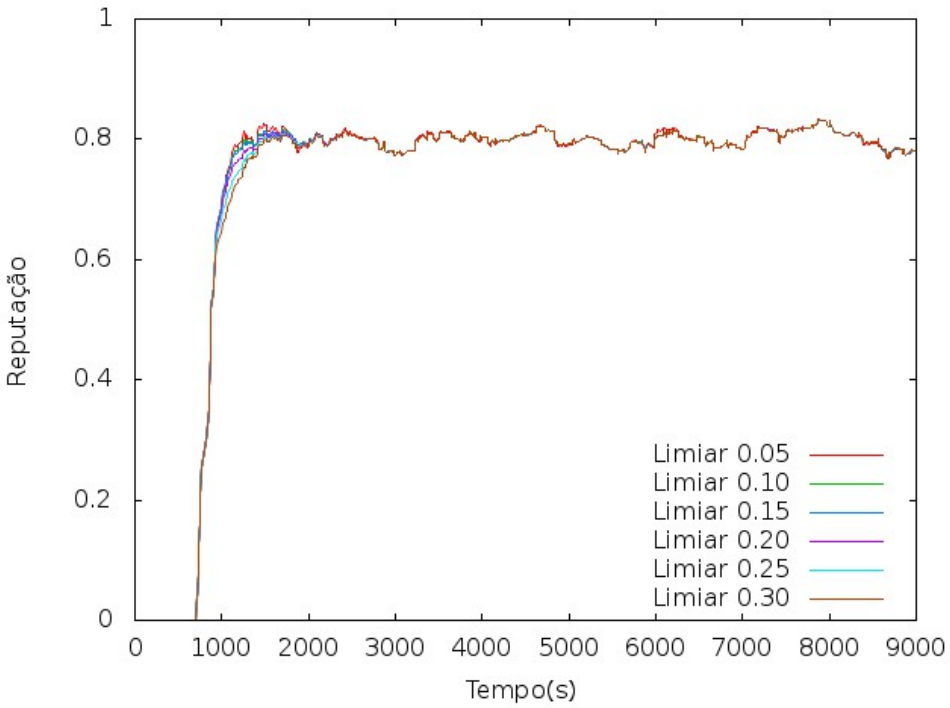

Figura 4.27: Exemplo das curvas de reputação para o grupo 1 utilizando formulação da média móvel exponencial com α linear

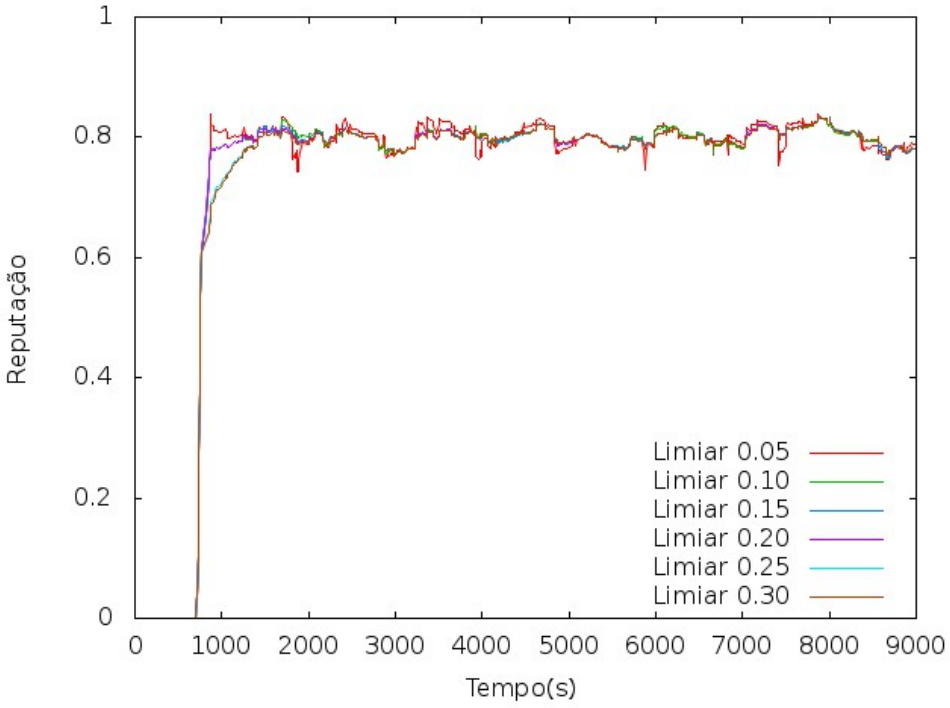

Figura 4.28: Exemplo das curvas de reputação para o grupo 1 utilizando formulação da média móvel exponencial com α escalar

A esperança móvel da distribuição beta apresenta oscilações menores que as encontradas na média móvel quando comparado os mesmos métodos de cálculo para α. Isso ocorre porque a formulação da esperança beta naturalmente suaviza o crescimento da reputação, assim, a cada interação, a diferença entre a reputação atual e o resultado do cálculo sobre os novos votos é pequena.

Por fim, as figuras 4.29, 4.30 e 4.31 mostram o comportamento das reputações utilizando a formulação proposta na seção 3.8.

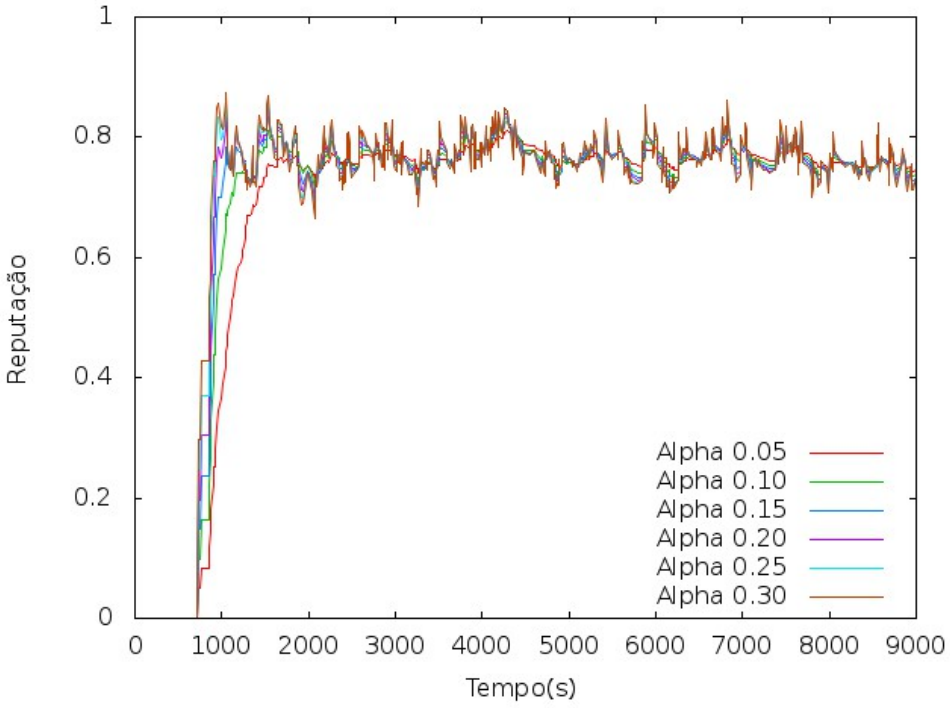

Figura 4.29: Exemplo das curvas de reputação para o grupo 1 utilizando formulação proposta com α fixo

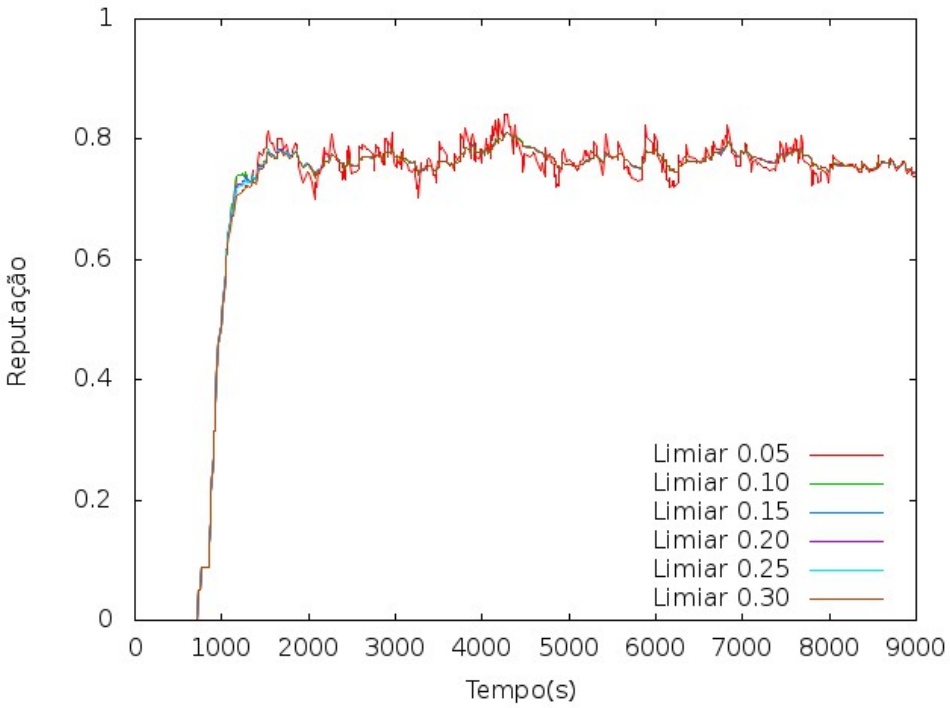

Figura 4.30: Exemplo das curvas de reputação para o grupo 1 utilizando formulação proposta com α linear

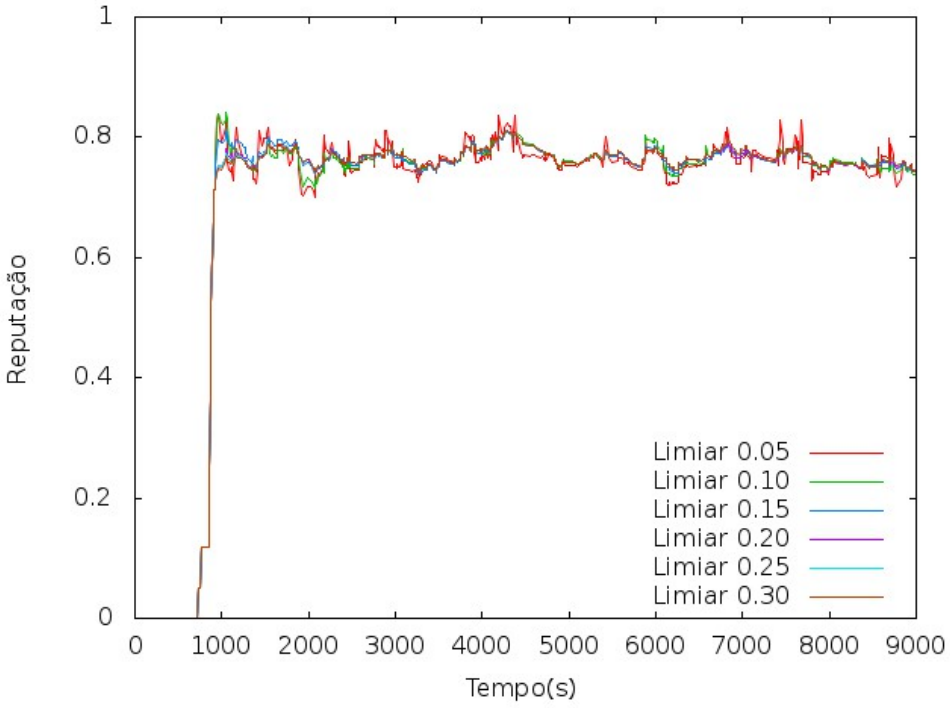

Figura 4.31: Exemplo das curvas de reputação para o grupo 1 utilizando formulação proposta com α escalar

A formulação proposta apresenta oscilações ainda maiores que as outras duas formulações. Isso se deve ao fato do peso de cada voto sofrer influência do valor atual da reputação, ou seja, votos negativos têm maior relevância quando a reputação é alta.

Estes gráficos mostraram o comportamento de cada uma das formulações em relação ao método utilizado para α, exemplificando as oscilações causadas em cada combinação. Os gráficos dos demais grupos podem ser conferidos no apêndice A, seção A.2.

As detecções também foram analisadas para cada um desses valores, sendo exibidos nas figuras 4.32 a 4.34. A figura 4.32 mostra a porcentagem de detecção de interações ruins ao longo do tempo para cada abordagem de α quando utilizado a formulação de média móvel exponencial. As figura 4.33 e 4.34 demonstram os mesmos aspectos porém para a utilização da esperança móvel da distribuição beta e para a formulação proposta, respectivamente.

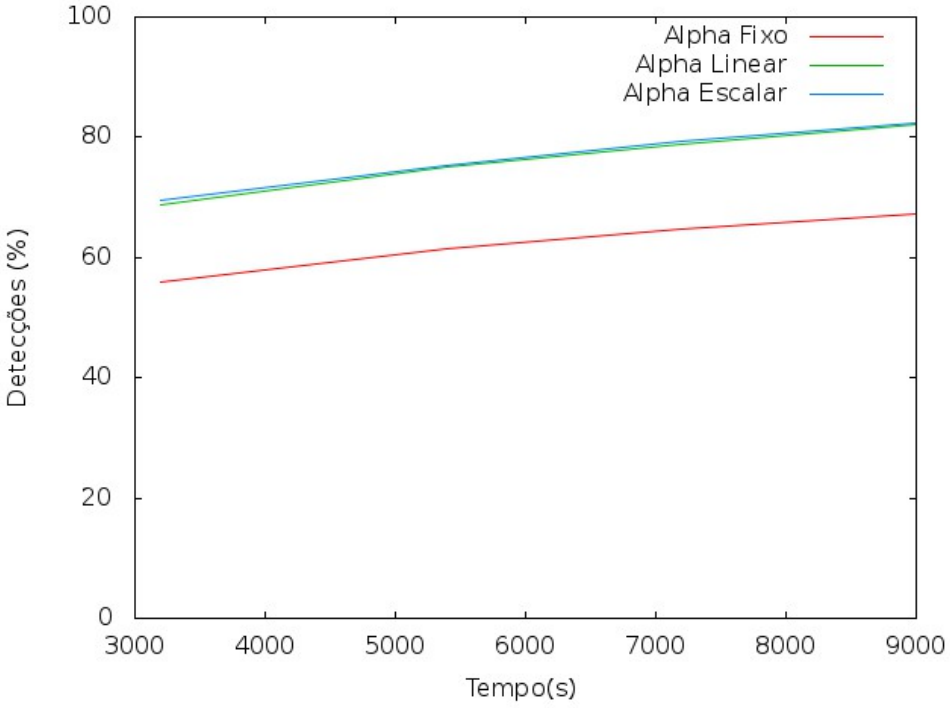

Figura 4.32: Média Móvel Exponencial - Detecções x Tempo

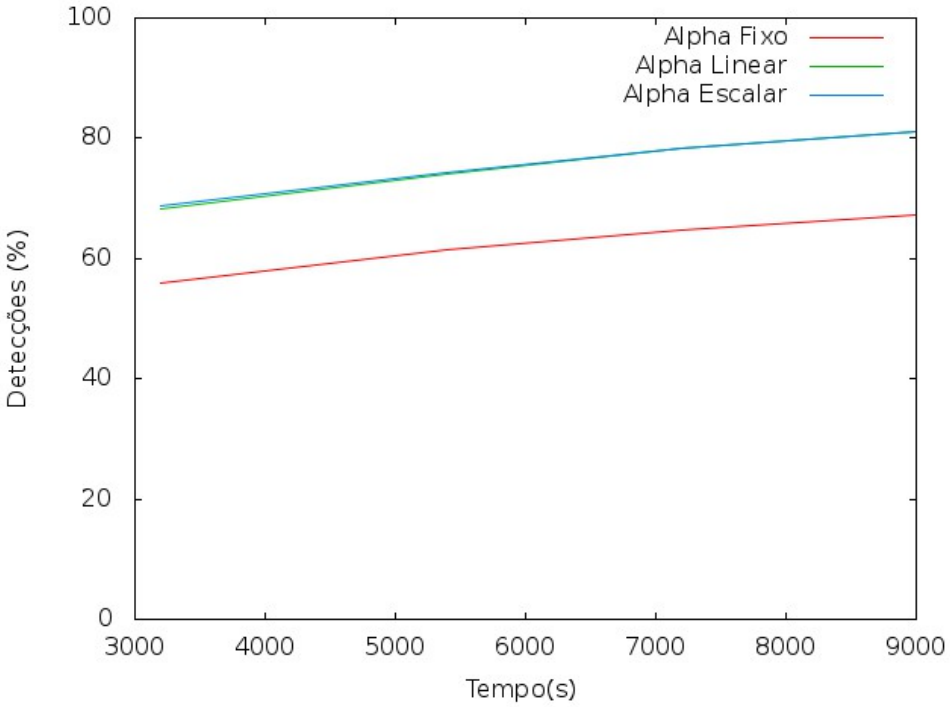

Figura 4.33: Esp. Móvel Dist. Beta - Detecções x Tempo

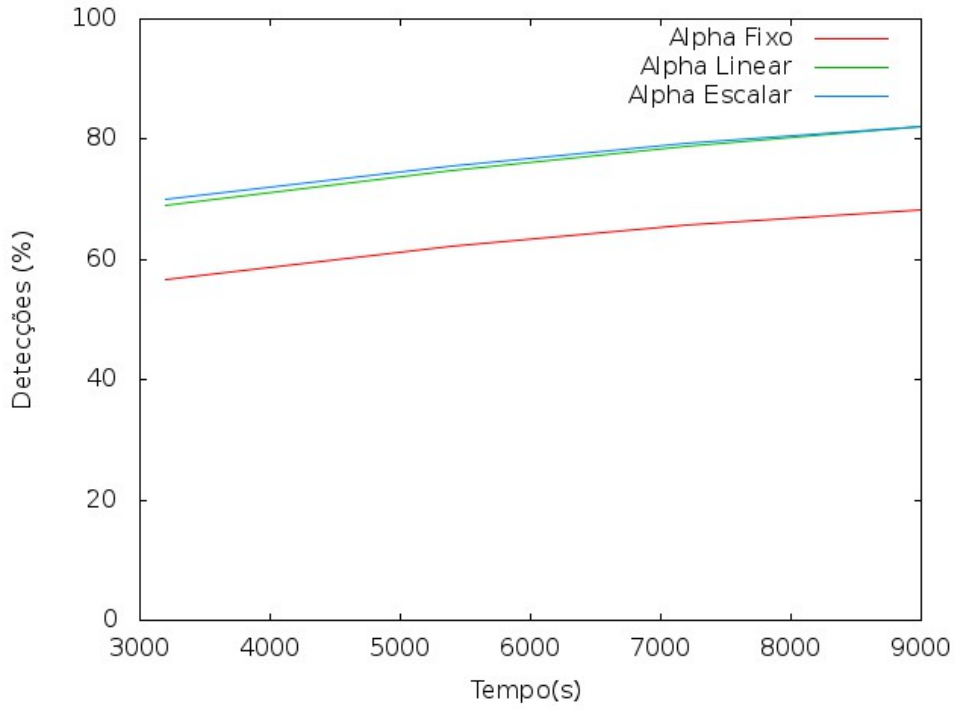

Figura 4.34: Formulação Proposta - Detecções x Tempo

Pelos gráficos apresentados percebe-se que as abordagens α linear e escalar apresentam detecção superior, enquanto valores fixos possuem taxas de detecção inferiores a 70%. Conclui-se então que valores flutuantes de α são mais relevantes na detecção de provedores de baixa qualidade. Como desejado a insistência no erro ocasiona uma queda mais acentuada na reputação, permitindo detectar este dispositivo como um provedor de baixa qualidade sem necessariamente punir pequenos deslizes.

4.4.2 Ataques de Curto período

As formulações também foram avaliadas do ponto de vista de ataques de curto período de tempo. Como descrito, durante algum tempo alguns provedores ofereciam serviços de baixa qualidade. Exemplos do comportamento das reputações de cada grupo podem ser observados nas figuras 4.35, 4.36, 4.37 e 4.38, que exemplificam a reputação de um dispositivo ao longo do tempo com variados tempos de ataques de curto período: $1/11$ (5 segundos de ataque), $1/6$ (10 segundos), $1/2$ (50 segundos) e $2/3$ (100 segundos), respectivamente. Pelas curvas, é possível perceber que ataques de curta duração quase não são percebidos pelas formulações e ataques mais longos, superiores a metade do tempo, são mais perceptíveis e influenciam diretamente na reputação.

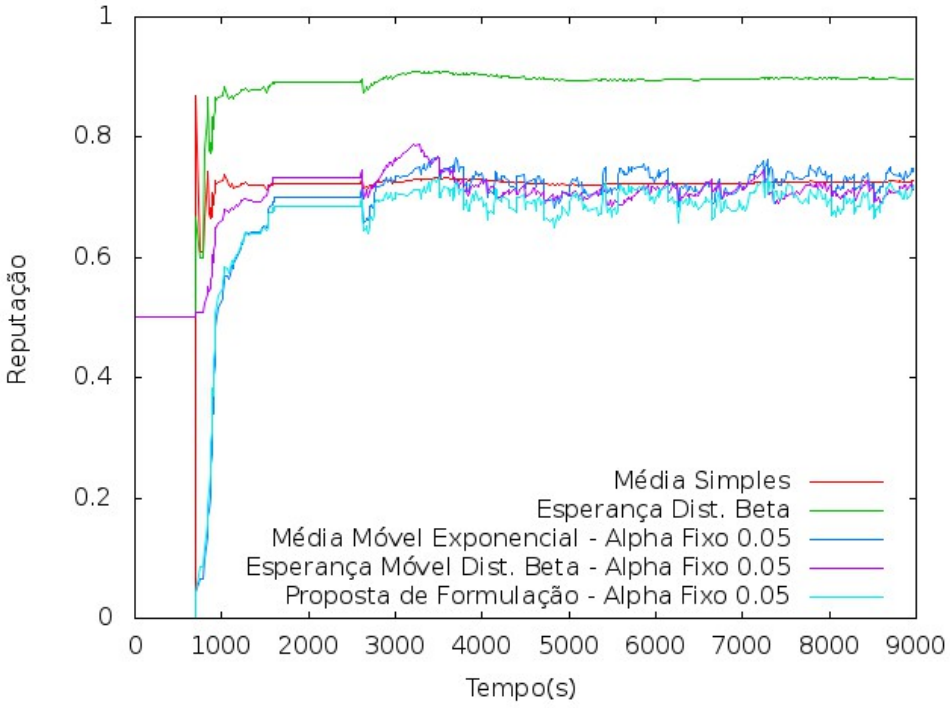

Figura 4.35: Ataques de $1/11$ - Reputação x Tempo de Simulação

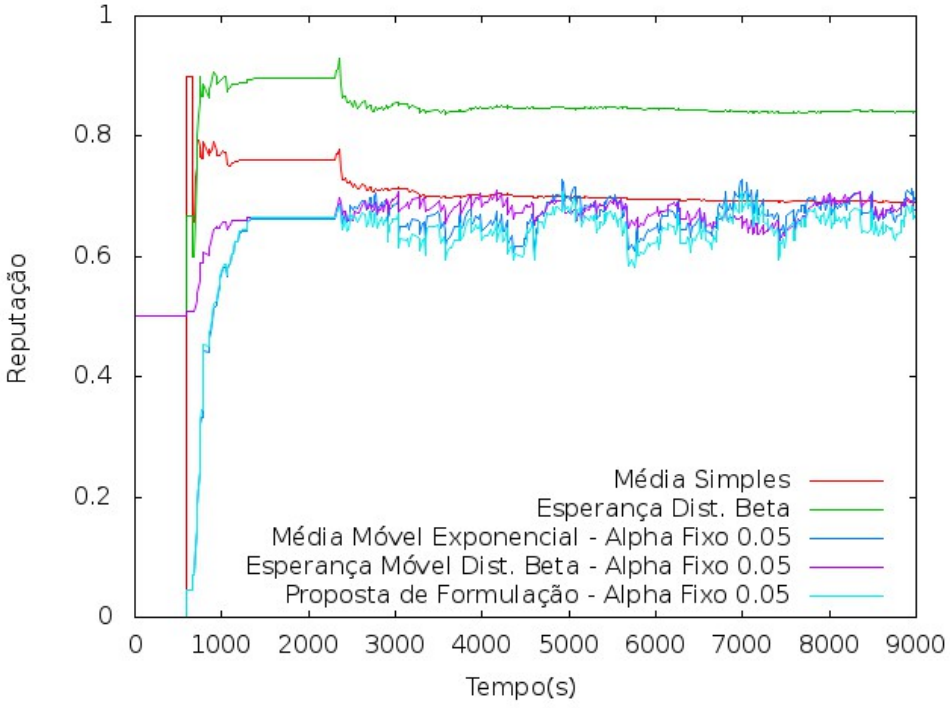

Figura 4.36: Ataques de $1/6$ - Reputação x Tempo de Simulação

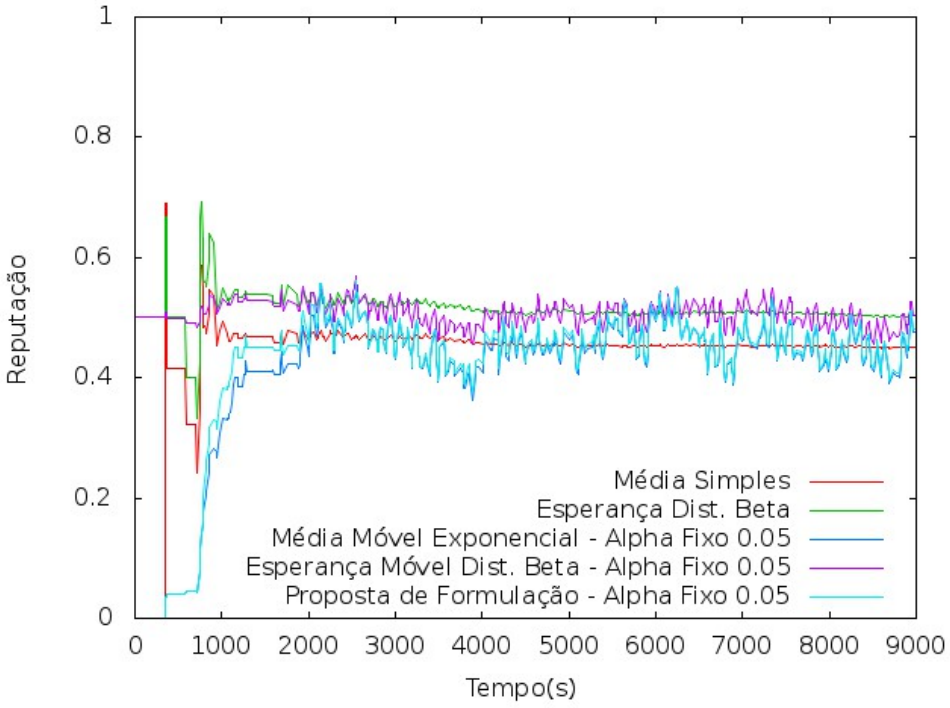

Figura 4.37: Ataques de $1/2$ - Reputação x Tempo de Simulação

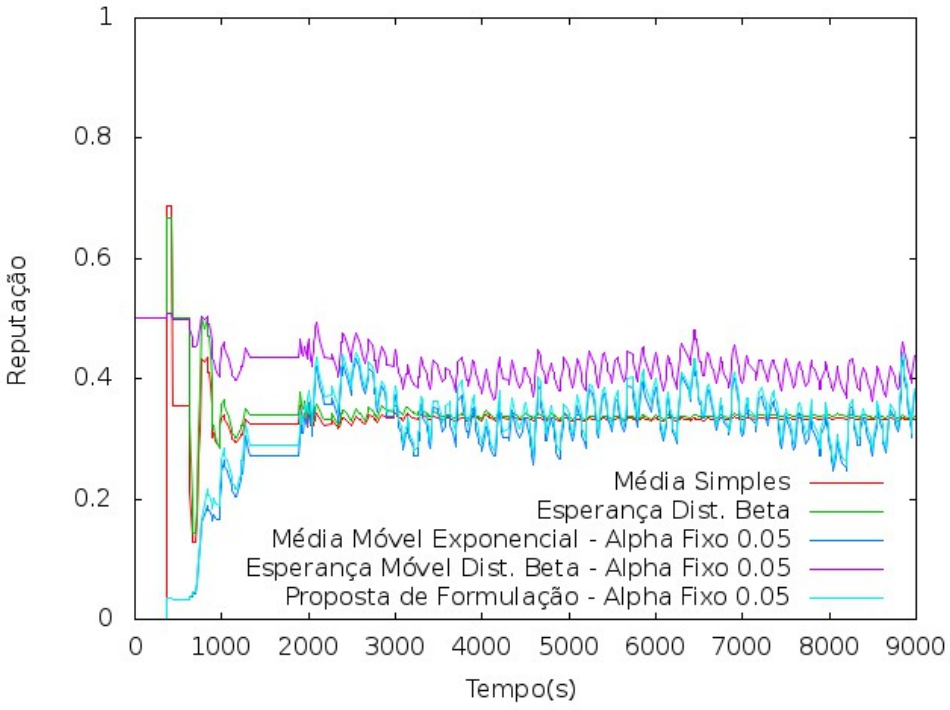

Figura 4.38: Ataques de $2/3$ - Reputação x Tempo de Simulação

Nas figuras 4.35 e 4.36 as reputações oscilaram um pouco mais porém continuaram em torno da média padrão do dispositivo, que pertencia ao grupo 1 de qualidade. A figura 4.36 também mostra como a formulação de média simples pode oscilar muito no começo mas se

mantém estável quando uma reputação já contém quantidade maior de votos. A queda inicial nesta reputação pela média se deve a poucos votos com valores muito baixos, o que leva a reputação a valores também baixos. Porém, quando volta a oferecer serviços de boa qualidade sua reputação ainda consegue voltar ao patamar do seu grupo e ao longo do tempo as oscilações se tornam menores.

Quando o tempo de ataque se torna maior o dispositivo é classificado como pertente aos grupos reputação de menor qualidade. Por exemplo, nas figuras 4.37 e 4.38, um dispositivo do grupo 1 de qualidade efetuando um ataque de 50 e 100 segundos é classificado como sendo do grupo 3.

Em relação às detecções, as formulações obtiveram desempenhos muito parecidos entre si, porém com uma pequena diferença entre as formulações que dão relevância aos votos recentes. A figura 4.39 mostra o desempenho das formulações em relação ao tempo usado para um ataque, por exemplo, 5 segundos oferecendo serviços de baixa qualidade para 50 segundos de comportamento normal. O tempo de comportamento normal não se altera, permanecendo constante em 50 segundos, enquanto o tempo de oferta de serviços de baixa qualidade cresce conforme os valores descritos na seção 4.3.1.

Percebe-se uma relação entre o tempo de ataque e o desempenho na detecção das formulações. Nenhuma das formulações apresentou destaque neste tipo de ataque tendo uma pequena queda na porcentagem de detecções para tempos de ataque pequenos e mantendo os valores de detecções dos testes anteriores para tempos maiores, quando as reputações realmente são classificadas como de grupos de menor qualidade.

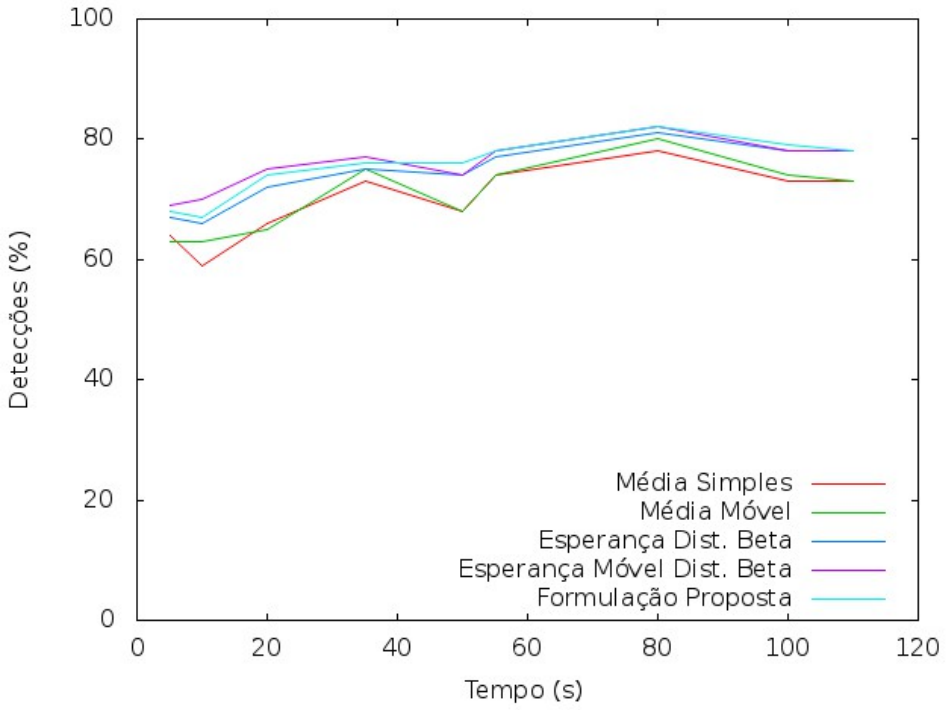

Figura 4.39: Detecção x Tempo de Ataque

Porém, quando analisa-se a porcentagem de detecções em relação ao método utilizado para o cálculo do α, as porcentagens de detecções tornam-se melhores, permanecendo a diferença entre as abordagens linear e escalar em relação a abordagem de α fixo. Esse comportamento pode ser observado nas figuras 4.40, 4.41 e 4.42.

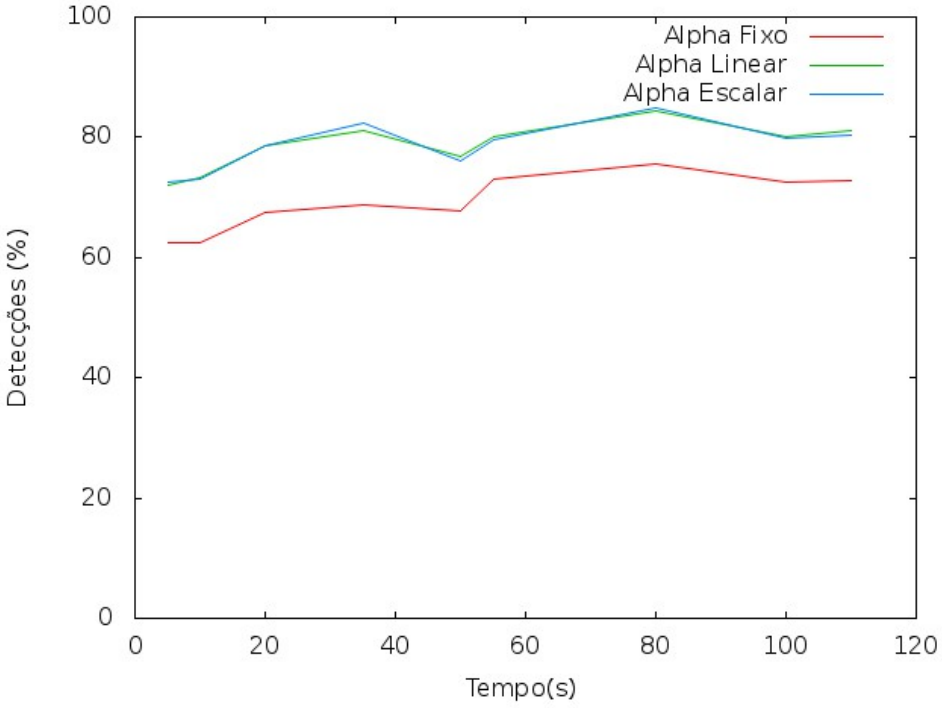

Figura 4.40: Média Móvel Exponencial - Detecção x Tempo de Ataque

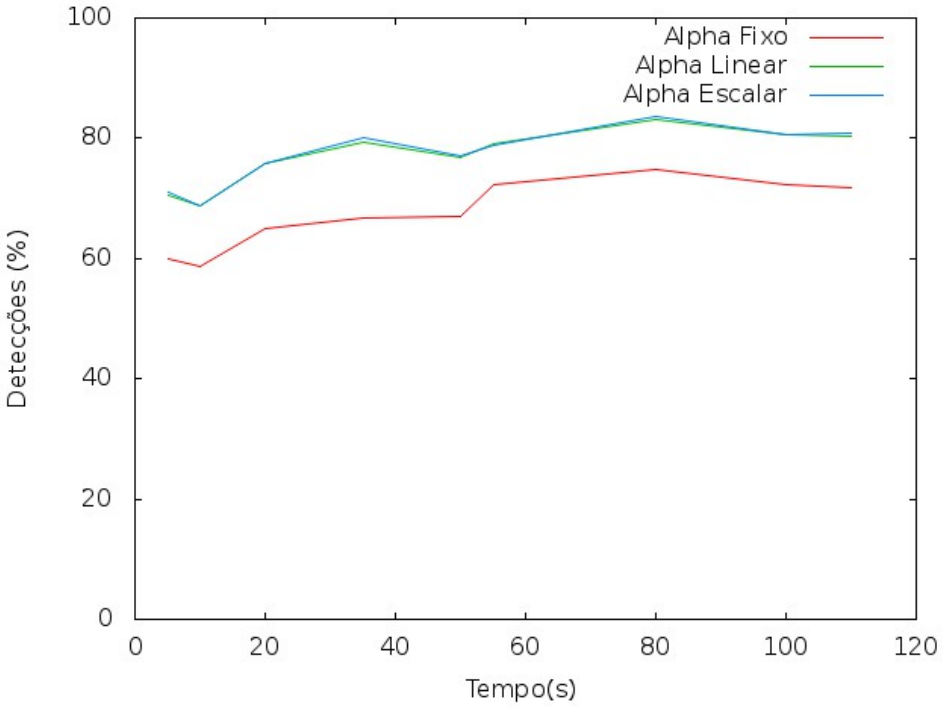

Figura 4.41: Esperança Móvel Dist. Beta - Detecção x Tempo de Ataque

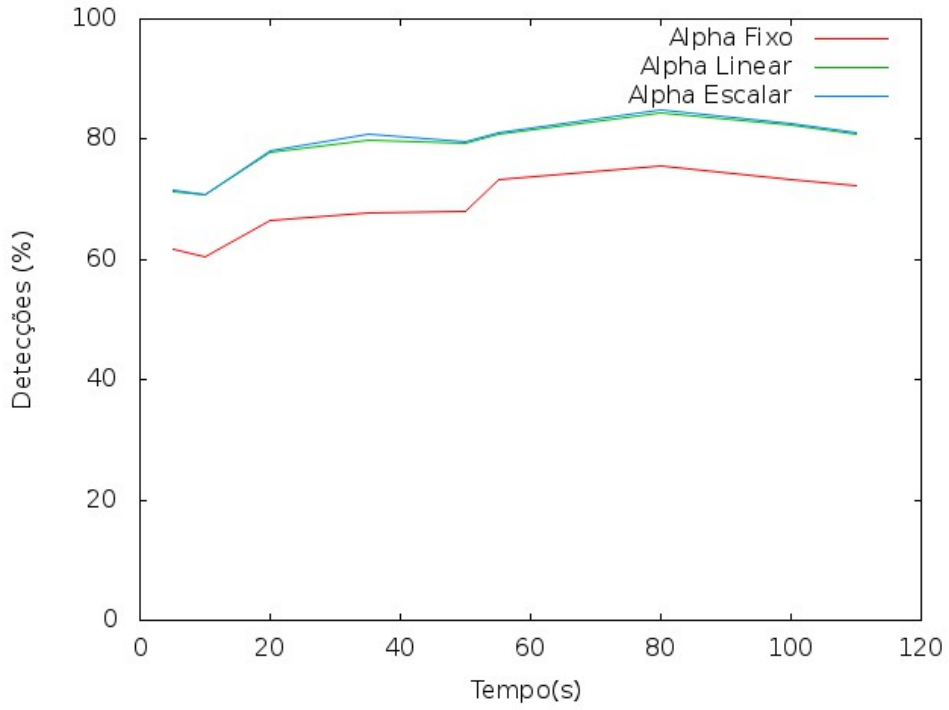

Figura 4.42: Formulação Proposta - Detecção x Tempo de Ataque

O desempenho das formulações em situações padrões e sobre ataques de curto período indicam a importância do peso dado ao comportamento mais recente do provedor de serviço. A adaptação deste valor ao comportamento torna-se muito mais relevante que as próprias formulações analisadas neste trabalho.

4.4.3 Escalabilidade

Os resultados das simulações feitas para analisar a escalabilidade do sistema de reputação proposto mostraram a influência das variáveis analisadas no tempo de atualização dos dados e na construção das redes ponto a ponto. Foram analisados o número médio de contatos estabelecidas por dispositivo, o tempo necessário para a atualização de mineradores e verificadores completos na criação de blocos e no recebimento de um novo voto, o número de redes distintas criadas no sistema e o tamanho médio do maior caminho entre os dispositivos.

O número de servidores de registros não afeta de modo geral o comportamento do sistema de reputação (tabela 4.4). O número médio de contatos varia pouco e o tempo de atualização de votos e blocos tende a aumentar. Isso ocorre quando existe um número maior de servidores de registro do que de mineradores e verificadores completos, desta forma o número médio de contatos tende a diminuir por falta de balanceamento na rede.

Núm. de Servidores de Registro	Número Médio de Contatos	Tempo de Atualização (s)		Núm. de Redes	Maior Caminho
		Votos	Blocos		
10	11.840	0.492	0.677	1	4
20	11.493	0.571	0.716	1	4
30	11.100	0.548	0.666	1	4
40	10.876	0.610	0.811	1	4
50	10.811	0.679	0.905	1	4
60	10.600	0.625	0.915	1	4
70	10.417	0.675	0.927	1	4
80	10.044	0.702	0.878	1	4
90	10.000	0.675	0.944	1	4
100	9.953	0.714	0.982	1	4

Tabela 4.4: Resultados da análise da influência do número de servidores de registros

O número de mineradores também interfere no tempo de atualização e na complexidade da rede formada no sistema de reputação. Contudo, os valores destas variáveis não crescem na mesma proporção que o número de dispositivos, sendo necessário um maior número de dispositivos para causar uma alteração significativa na rede. Isto é exemplificado pelas figuras 4.43 e 4.44.

A figura 4.43 mostra o crescimento do maior caminho médio entre os dispositivos conforme o número de mineradores aumenta. É possível perceber que a curva tende a crescer vagarosamente em relação ao número de dispositivos simulados.

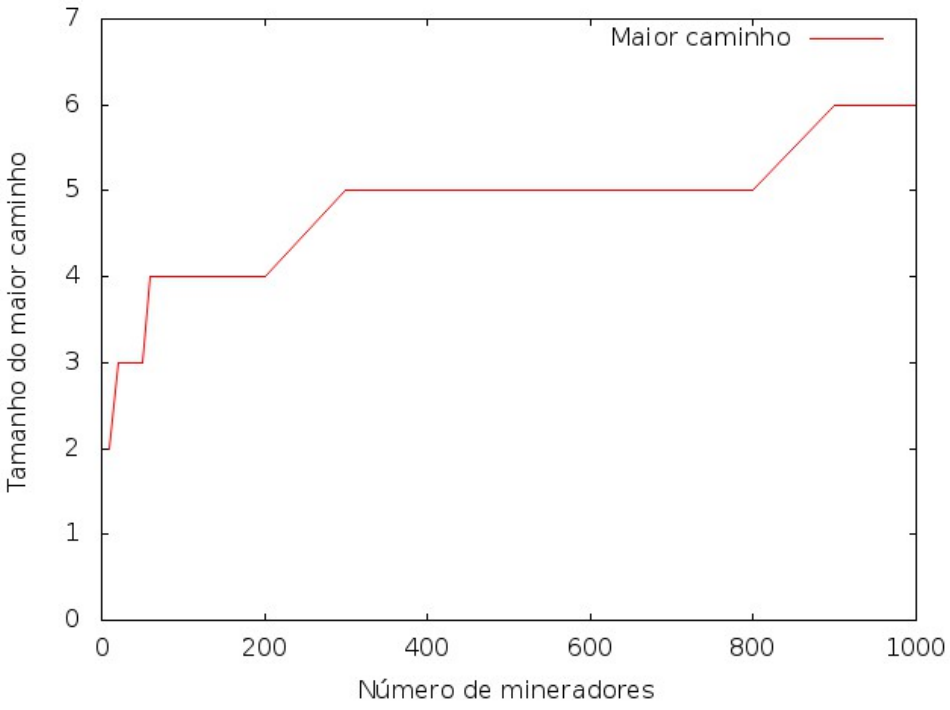

Figura 4.43: Maior caminho médio x Número de dispositivos simulados

O tempo de atualização dos votos em relação ao número de mineradores, figura 4.44, também apresenta uma curva crescente porém lenta. Os picos nos tempos de atualização no início do gráfico são causados pela desproporção entre o número de mineradores e de servidores de registro, o que ocasiona um menor grau de conectividade e portanto um maior tempo de atualização. Este comportamento também pode ser observado na tabela 4.5.

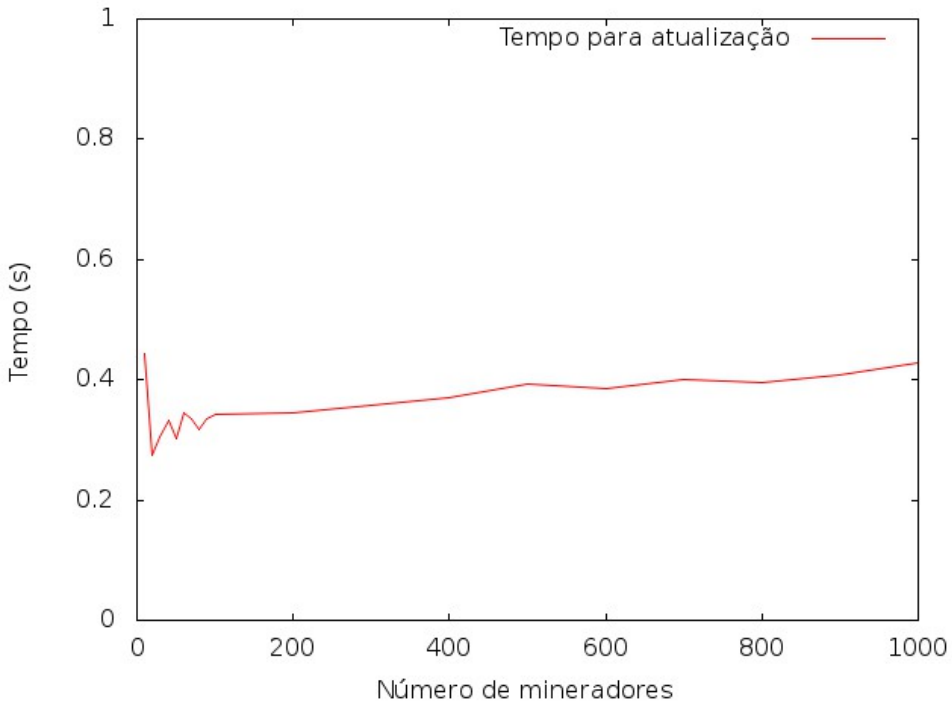

Figura 4.44: Tempo para atualização dos votos x Número de dispositivos simulados

O tempo necessário para atualização dos blocos segue uma curva um pouco diferente. Há um grande crescimento no tempo de atualização quando número de mineradores é relativamente próximo ao número de servidores de registro. Conforme o número de mineradores e de verificadores completos se torna maior, o tempo torna-se mais estável, figura 4.45. Isso ocorre pois o número mínimo de dispositivos necessários para obter um grau de conectividade satisfatório é pequeno.

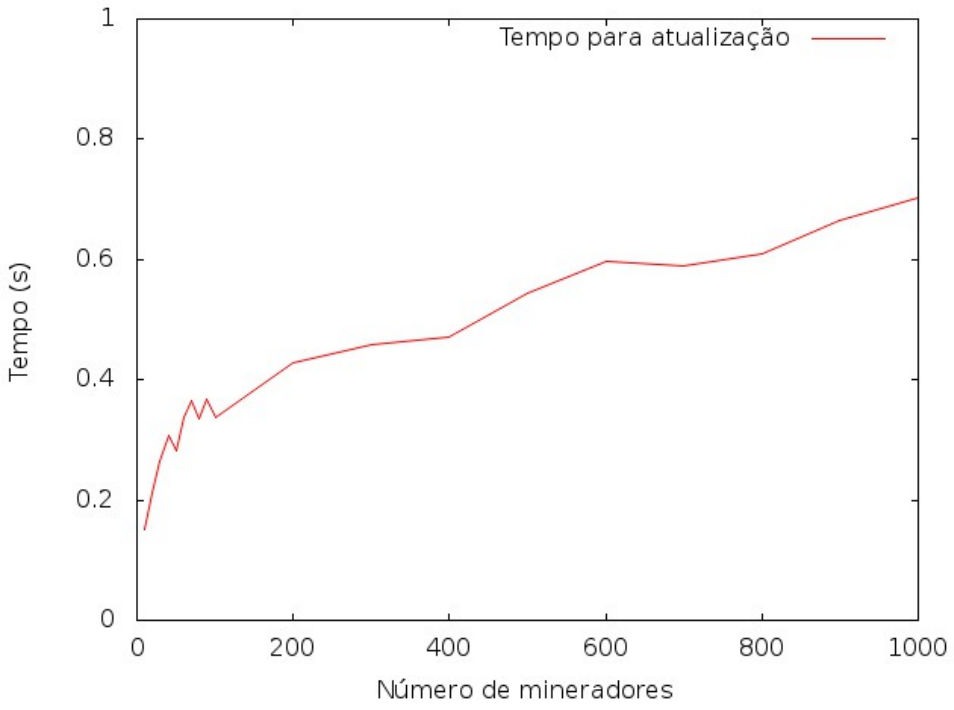

Figura 4.45: Tempo para atualização dos blocos x Número de dispositivos simulados

A tabela 4.5 exibe a relação entre o número de mineradores e de servidores de registro com o número médio de contatos de cada minerador. Observa-se uma relação entre estes valores, conforme o número de servidores de registro aumenta mais dispositivos são necessário para manter um bom número de contatos. Isto deve-se aos servidores de registro serem responsáveis por balancear a rede e um número muito grande destes servidores faz com quem cada um possua conhecimento de pouco dispositivos, atrapalhando este balanceamento.

O número mínimo de contatos entre os mineradores e os verificadores exerce uma influência maior sobre o desempenho do sistema. A tabela 4.6 exibe o resultados das simulações.

É perceptível como o aumento do número de contatos diminui o tempo médio de atualização dos blocos e dos votos além do maior caminho médio necessário para atualizar o sistema. Contudo, como é possível observar no gráfico 4.46, para graus mínimos maiores que seis, o sistema não tem um ganho tão significativo para nenhuma dos atributos avaliados, ou seja, possuir contato com aproximadamente doze dispositivos é suficiente para manter um bom desempenho na convergência dos dados.

Núm. Dispositivos	10	20	30	40	50	60	70	80	90
10	6.62	6.75	6.44	6.33	6.21	6.336	6.204	6.224	6.180
20	8.932	8.170	7.986	7.814	7.680	7.524	7.404	7.228	7.328
30	10.219	9.112	8.684	8.448	8.311	8.169	7.856	7.820	7.833
40	11.106	10.062	9.490	8.881	8.475	8.558	8.399	8.143	8.066
50	11.74	10.365	10.041	9.414	9.273	8.926	8.746	8.602	8.568
60	12.051	10.979	10.318	9.591	9.477	9.286	8.994	8.780	8.762
70	12.309	11.374	10.726	10.297	9.767	9.730	9.565	8.967	8.998
80	12.667	11.661	10.931	10.627	10.307	9.923	9.563	9.371	9.213
90	12.785	12.038	11.280	10.679	10.389	9.843	9.627	9.650	9.395
100	13.032	12.115	11.567	10.966	10.711	10.170	10.159	9.819	9.650
200	13.673	13.213	13.043	12.312	12.057	11.771	11.463	11.240	11.195
300	13.934	13.626	13.373	13.105	12.802	12.664	12.694	12.408	12.104
400	14.046	14.083	13.631	13.406	13.215	13.198	12.747	12.547	12.508
500	14.551	13.976	13.780	13.891	13.610	13.414	13.128	13.143	13.033
600	14.188	14.035	14.116	13.769	13.860	13.713	13.370	13.147	13.079
700	14.238	14.108	14.197	14.103	13.745	13.916	13.756	13.430	13.280
800	14.262	14.133	14.207	14.078	13.811	13.724	14.022	13.755	13.404
900	14.474	14.326	14.343	14.129	13.889	14.043	13.952	13.893	13.518
1000	14.550	14.201	14.363	14.037	13.956	13.847	14.186	14.077	13.799

Tabela 4.5: Resultados da análise da influência do número de servidores de registros em relacão ao número de dispositivos

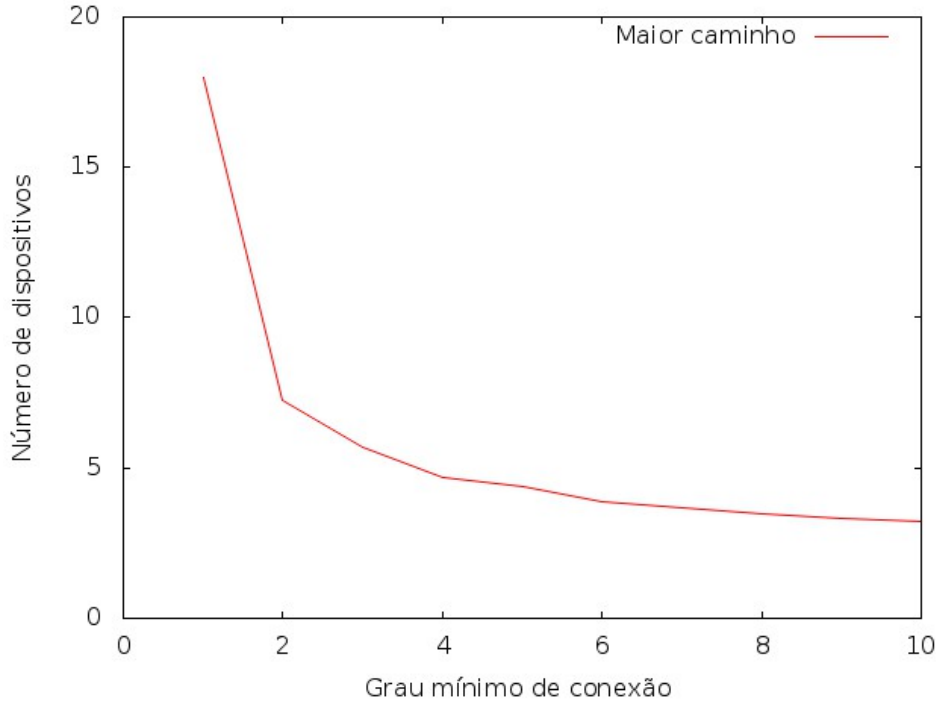

Figura 4.46: Maior caminho médio x Número mínimo de contatos

Núm. Mínimo de Contatos	Número Médio de Contatos	Tempo de Atualização (s)		Núm. de Redes	Maior Caminho
		Votos	Blocos		
1	5.264	4.273	5.845	2.229	17.969
2	6.615	1.039	1.762	1	7.240
3	7.883	0.700	1.138	1	5.692
4	9.440	0.489	0.751	1	4.668
5	10.454	0.415	0.650	1	4.360
6	11.919	0.348	0.491	1	3.888
7	13.318	0.312	0.413	1	3.692
8	14.698	0.278	0.407	1	3.460
9	15.588	0.262	0.406	1	3.319
10	17.047	0.242	0.365	1	3.207

Tabela 4.6: Resultados da análise do número mínimo de contatos por dispositivo

O número de contatos também é responsável por formações de redes distintas de mineradores e verificadores completos. Isso faz com que as redes não troquem informações sobre votos e blocos recebidos, criando reputações diferentes para cada usuário em cada uma das redes. Este fenômeno ocorreu apenas nas simulações com o grau mínimo um, sendo solucionado com graus mínimos maiores ou iguais a dois.

Na tabela 4.7 podemos observar que a variação do número de registros encaminhados na resposta de uma requisição de conexão aos servidores de registros. A variação deste número não altera de forma significativa os padrões de conexão entre os mineradores e verificadores completos. O caminho máximo permanece constante em todas as simulações e não correm casos de criação de redes distintos. Esses resultados são consequência do modo como os servidores de registros e os próprios registros enviados são escolhidos. O método aleatório dessas escolhas tende a balancear os contatos na rede.

Núm. de Registros	Número Médio de Contatos	Tempo de Atualização (s)		Núm. de Redes	Maior Caminho
		Votos	Blocos		
1	7.138	0.800	1.021	1	4
3	10.342	0.646	0.890	1	4
5	10.693	0.616	0.874	1	4
7	12.803	0.593	0.833	1	4
9	13.831	0.588	0.829	1	4

Tabela 4.7: Resultados da análise do número de registros por requisição

A queda no tempo de atualização de blocos e votos ocorre pois os dispositivos estão programados para estabelecer contato com todos os servidores da lista, o que causa um maior grau de conectividade entre os dispositivos e consequentemente uma maior facilidade na convergência dos dados.

A análise da escalabilidade do sistema de reputação se mostrou favorável. O aumento do número de dispositivos conectados parece trazer pouca influência para a convergência dos dados e melhora a interligação entre os nós, prevenindo o surgimento de redes distintas.

Capítulo 5

CONCLUSÕES

Este trabalho apresentou um sistema de reputação distribuído para avaliação de interações baseadas em serviços. Usando uma abordagem distribuída, baseada no modelo das cripto moedas, o sistema reputação proposto permite que votos associados à qualidade da prestação e consumo de serviços sejam transformados em uma medida objetiva. A arquitetura do sistema prevê papéis de servidores de registro, mineradores e verificadores completos, que formam a infraestrutura da plataforma, além dos clientes e consumidores de serviço.

O funcionamento do sistema introduz duas etapas a mais no protocolo de comunicação via serviços: uma consulta da reputação e o encaminhamento de um voto.

Cinco formulações diferentes foram analisadas para o cálculo das reputações, procurando identificar formas de refletir a qualidade das interações de um dispositivo, e evitando influências indesejadas de ataques.

O sistema foi avaliado por meio de simulações da comunicação de dispositivos portados por pessoas e veículos em um cenário urbano.

Os resultados obtidos mostraram que a proposta é funcional e capaz de limitar os efeitos de ataques e comportamentos indesejados dos dispositivos.

Foi observado que formulações como a média móvel exponencial, a esperança móvel da distribuição beta e formulação proposta obtiveram melhores níveis de detecção do que as demais. Isso se deve ao fato de darem mais relevância ao comportamento recente dos dispositivos analisados. Em uma análise mais profundada, reparou-se que o peso dado aos votos recentes é ainda mais significativo para esta detecção. As abordagens linear e escalar para alfa se mostraram mais eficazes que a abordagem de valores fixos, tendo um desempenho superior de até 10%.

Do ponto de vista arquitetural, o sistema de reputação se mostrou escalável. Os tempos de atualização foram aceitáveis e com crescimento lento. Desta maneira, o sistema mostrou-se possível de ser implementando em uma situação real, garantindo escalabilidade quando acessado por muitos dispositivos. Isto, porém, também depende do fator de adoção que o sistema terá pelos usuários, pois estes devem contribuir com processamento para o seu bom funcionamento.

Um outro aspecto favorável da arquitetura distribuída escolhida está relacionado à ausência de controle centralizado e ao favorecimento da operação confiável porém anonimizada. Isto é provido pelos métodos de identificação associadas a chaves públicas e privadas da criptografia assimétrica.

A formulação proposta neste trabalho não apresentou ganhos quando comparada as formulações que utilizavam métodos de diferenciação entre votos recentes e votos antigos. Desta maneira, propõe-se que novos estudos sejam feitos para aprimorar a eficácia da formulação utilizada no sistema de reputação.

Assim, conclui-se que o sistema de reputação proposto, aliado a uma boa formulação, pode trazer benefícios na detecção de serviços de baixa qualidade, garantindo a privacidade dos seus utilizadores.

5.1 Trabalhos Futuros

O sistema apresentado torna-se relevante diante do crescente uso de serviços para comunicação entre dispositivos. Porém, para que se torne algo implementável, ainda são necessárias análises em outros pontos da arquitetura e encontrar soluções para questões de segurança da formulação utilizada. Abaixo estão listadas algumas dessas análises:

- Métodos de contabilização e redução do consumo de energia necessário para o uso do protocolo.
- Métodos de detecção para o comportamento de ataques de curto período.
- Métodos de detecção de coligações.
- Métodos para evitar "*fresh start*".
- Custo ideal para encontrar a prova de trabalho e métodos de balanceamento de custo automatizados.

Devido ao comportamento ineficiente das formulações estudas na detecção de ataques de curto período, é necessário investigar outros métodos para evitar este tipo de ataque. As

coligações, que também podem atuar tanto em ataques de autopromoção quanto de difamação, também devem ser analisadas pois a facilidade de criação de identidades incentiva este tipo de comportamento.

Esta característica da criação de identidades também incentiva o "*fresh start*", em que um dispositivo abandona sua reputação ruim por uma nova. Este fatores são críticos na escolha de uma formulação e precisam ser analisados antes de da decisão.

No ponto de vista da arquitetura, estudos para descobrir o tempo ideal de criação dos blocos e a quantidade de mineradores e verificadores completos necessários para atender a demanda de votos são pontos fundamentais para a implantação deste sistema em ambientes reais.

REFERÊNCIAS

ABDUL-RAHMAN, A.; HAILES, S. Supporting trust in virtual communities. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, v. 00, n. c, p. 1–9, 2000. ISSN 15301605.

ACAMPORA, G.; CASTIGLIONE, A.; VITIELLO, A. A fuzzy logic based reputation system for E-markets. In: *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.: s.n.], 2014. p. 865–872. ISBN 978-1-4799-2072-3.

AKINGBESOTE, a. O. et al. A quality of service aware multi-level strategy for selection of optimal web service. *IEEE International Conference on Adaptive Science and Technology, ICAST*, 2013. ISSN 23269448.

ATTA; BILAL, S. M.; OTHMAN, M. A Performance Comparison of Network Simulators for Wireless Networks. *2012 IEEE International Conference on Control System, Computing and Engineering*, p. 34–38, 2012.

BLöMER, J. How to share a secret. *Algorithms Unplugged*, p. 159–168, 2011. ISSN 0001-0782.

BLUETOOTH SIG, I. *Core Specification v4.2*. 2014. Disponível em: <https://www.bluetooth.org/en-us/specification/adopted-specifications>. Acesso em: 2015-09-3.

BOUDEC, J. L.; BUCHEGGER, S. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. *P2PEcon 2004*, n. 5005, p. 1–6, 2004.

BROWN, D. R. L. Sec 2: Recommended elliptic curve domain parameters. In: *Standards for Efficient Cryptography*. [S.l.]: Certicom Corp., 2010.

CARPENTER, B.; BRIM, S. *Middleboxes: Taxonomy and Issues*. IETF, fev. 2002. RFC 3234 (Informational). (Request for Comments, 3234). Disponível em: <http://www.ietf.org/rfc/rfc3234.txt>.

CELESTINI, A.; De Nicola, R.; TIEZZI, F. Specifying and Analysing Reputation Systems with a Coordination Language. *28th Annual ACM Symposium on Applied Computing (SAC'13)*, p. 1363–1368, 2013.

CERTICOM. *Certicom*. 2015. Disponível em: <https://www.certicom.com/>. Acesso em: 2016-03-10.

CHAOKAI, H. C. H.; MENG, W. M. W. Comparison and analysis of different reputation systems for peer-to-peer networks. *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, v. 3, p. 20–23, 2010. ISSN 2154-7491.

CHEN, K. et al. A Social Network based Reputation System for Cooperative P2P File Sharing. *IEEE Transactions on Parallel and Distributed Systems*, v. 9219, n. 3, p. 1–1, 2014. ISSN 1045-9219.

CHOO, E.; JIANG, J.; YU, T. COMPARS. In: *Proceedings of the 4th ACM conference on Data and application security and privacy - CODASPY '14*. New York, New York, USA: ACM Press, 2014. p. 87–98. ISBN 9781450322782.

DECKER, C.; WATTENHOFER, R. Information propagation in the Bitcoin network. In: *IEEE P2P 2013 Proceedings*. [S.l.]: IEEE, 2013. p. 1–10. ISBN 978-1-4799-0515-7.

DENKO, M. K.; SUN, T.; WOUNGANG, I. Trust management in ubiquitous computing: A Bayesian approach. *Computer Communications*, Elsevier B.V., v. 34, n. 3, p. 398–406, mar. 2011. ISSN 01403664.

ESPINOZA, M.; MENA, E. Discovering Web Services Using Semantic Keywords. *2007 5th IEEE International Conference on Industrial Informatics*, p. 725–730, 2007.

FIELDING, R.; TAYLOR, R. Principled design of the modern Web architecture. *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*, p. 407–416, 2000. ISSN 0270-5257.

FIELDING, R. T. *CHAPTER 5: Representational State Transfer (REST)*. 2000. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>. Acesso em: 2014-06-04.

FOUNDATION, O. S. *OpenSSL*. 2015. Disponível em: <https://www.openssl.org>. Acesso em: 2015-04-28.

FOURNARIS, A. P. Distributed Threshold Cryptography Certification With No Trusted Dealer. *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*, p. 400–404, 2011.

FREED, N.; BORENSTEIN, N. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. IETF, nov. 1996. RFC 2046 (Draft Standard). (Request for Comments, 2046). Updated by RFCs 2646, 3798, 5147, 6657. Disponível em: <http://www.ietf.org/rfc/rfc2046.txt>.

HILBRICH, R. *Simulation of Urban MObility*. 2015. Disponível em: <www.dlr.de/ts/sumo>. Acesso em: 2015-07-14.

HOFFMAN, K.; ZAGE, D.; NITA-ROTARU, C. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, v. 42, n. 1, p. 1–31, dez. 2009. ISSN 03600300.

KAMVAR, S. D.; SCHLOSSER, M. T.; GARCIA-MOLINA, H. The Eigentrust algorithm for reputation management in P2P networks. In: *Proceedings of the twelfth international conference on World Wide Web - WWW '03*. New York, New York, USA: ACM Press, 2003. p. 640. ISBN 1581136803.

Koeksal Miran Murat. A Survey of Network Simulators Supporting Wireless Networks. p. 11, 2008.

KOUTROULI, E.; TSALGATIDOU, A. Credible recommendation exchange mechanism for P2P reputation systems. *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, ACM Press, New York, New York, USA, p. 1943, 2013.

LI, H. et al. An efficient Merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, v. 8, n. 2, p. 655–663, 2014. ISSN 19379234.

LIMITED, A. *PolarSSL*. 2015. Disponível em: <https://tls.mbed.org>. Acesso em: 2015-04-28.

LIU, Y.-m. et al. A Distributed Trust-based Reputation Model in P2P System. *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, Ieee, n. 60273041, p. 294–299, jul. 2007.

LTD., O. *OMNet++*. 2015. Disponível em: <https://omnetpp.org>. Acesso em: 2015-01-22.

MERKLE, R. *Method of providing digital signatures*. Google Patents, jan. 5 1982. US Patent 4,309,569. Disponível em: <http://www.google.com/patents/US4309569>.

NACY, S.; OH, T.; LEONE, J. Implementation of SHA-1 and ECDSA for vehicular ad-hoc network using NS-3. *Proceedings of the 2nd annual conference on Research in information technology - RIIT '13*, p. 83, 2013.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, p. 1–9, 2008.

NGUYêN, C. T.; CAMP, O. Using Context Information to Improve Computation of Trust in Ad Hoc Networks. *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Ieee, p. 619–624, out. 2008.

O'REILLY MEDIA, I. *The Chainblock*. 2013. Disponível em: <http://chimera.labs.oreilly.com/books/1234000001802/ch07.html>. Acesso em: 2015-03-10.

PECK, M. E. The cryptoanarchists' answer to cash. *IEEE Spectrum*, v. 49, n. 6, p. 50–56, jun. 2012. ISSN 0018-9235.

RAMACHER, R.; MONCH, L. Service Selection with Runtime Aspects: A Hierarchical Approach. *IEEE Transactions on Services Computing*, PP, n. 99, p. 1–1, 2014. ISSN 1939-1374.

SANTHANAM, L.; XIE, B.; AGRAWAL, D. P. Secure and efficient authentication in wireless mesh networks using Merkle trees. *Proceedings - Conference on Local Computer Networks, LCN*, p. 966–972, 2008. ISSN 0742-1303.

SATO, F. A Reputation System Resisting to Undercover Marketing. *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, p. 427–432, 2014.

SHEN, H.; LI, Z. A Hierarchical Account-Aided Reputation Management System for MANETs. *IEEE/ACM Transactions on Networking*, p. 1–1, 2014. ISSN 1063-6692.

SINGH, A. TrustMe: anonymous management of trust relationships in decentralized P2P systems. *Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003)*, IEEE Comput. Soc, p. 142–149, 2003.

SINGH, P. et al. Performance Comparison of Executing Fast Transactions in Bitcoin Network Using Verifiable Code Execution. *2013 2nd International Conference on Advanced Computing, Networking and Security*, Ieee, p. 193–198, dez. 2013.

SOMMER, C.; GERMAN, R.; DRESSLER, F. Bidirectionally coupled network and road simulation for improved IVC analysis. *IEEE Transactions on Mobile Computing*, v. 10, n. 1, p. 3–15, 2011. ISSN 15361233.

SU, X. et al. PBTrust: A Priority-Based Trust model for service selection in general service-oriented environments. *Proceedings - IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2010*, p. 841–848, 2010.

SZEFER, J.; BIEDERMANN, S. Towards fast hardware memory integrity checking with skewed Merkle trees. In: *Proceedings of the Third Workshop on Hardware and Architectural Support for Security and Privacy - HASP '14*. [S.l.: s.n.], 2014. p. 1–8. ISBN 9781450327770.

YAN, Z.; ZHANG, P.; DENG, R. TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applications. *Personal and Ubiquitous Computing*, p. 485–506, 2012.

ZHENGFENG, H.; JIANGHONG, H.; DONGHUI, H. A new authentication scheme based on verifiable secret sharing. *Proceedings - International Conference on Computer Science and Software Engineering, CSSE 2008*, v. 3, p. 1028–1030, 2008.

Apêndice A

COMPORTAMENTO DAS CURVAS DE REPUTAÇÃO

Este apêndice apresenta os gráficos referentes aos grupos de qualidade em cada uma das formulações analisadas.

A.1 Grupos de qualidade e Formulações

Nos gráficos A.1, A.2, A.3, A.4 e A.5 estão representadas reputações do grupo com média 0,5.

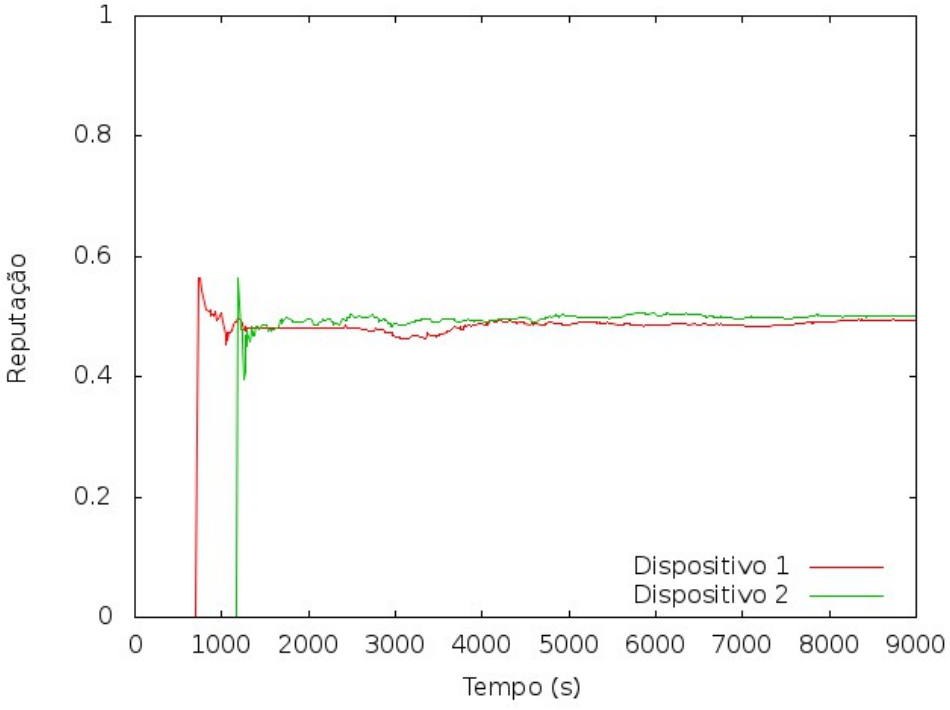

Figura A.1: Grupo 3: Média

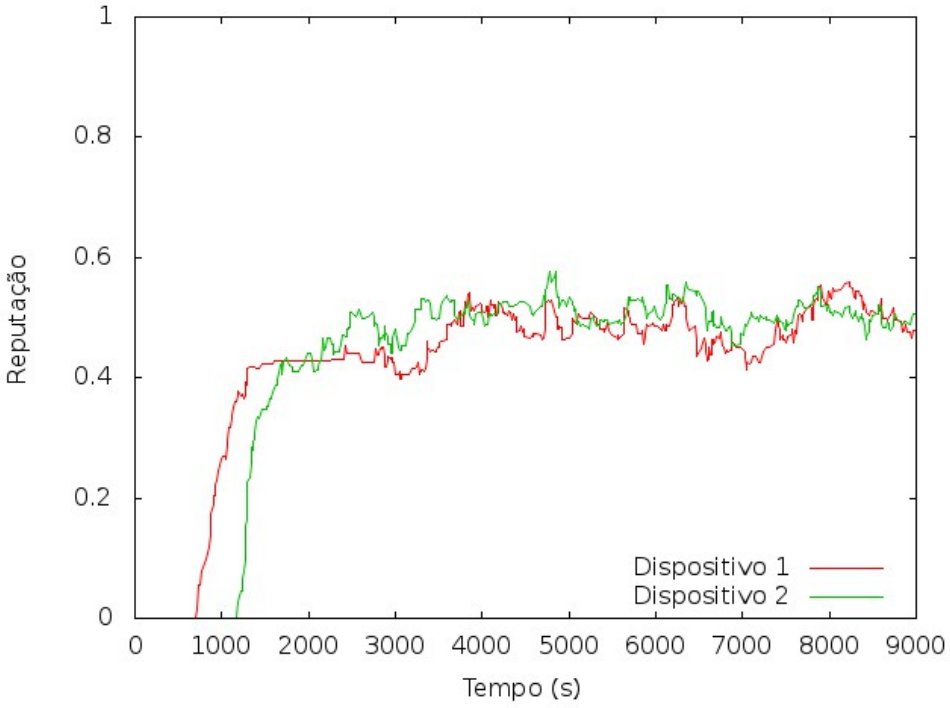

Figura A.2: Grupo 3: Média móvel

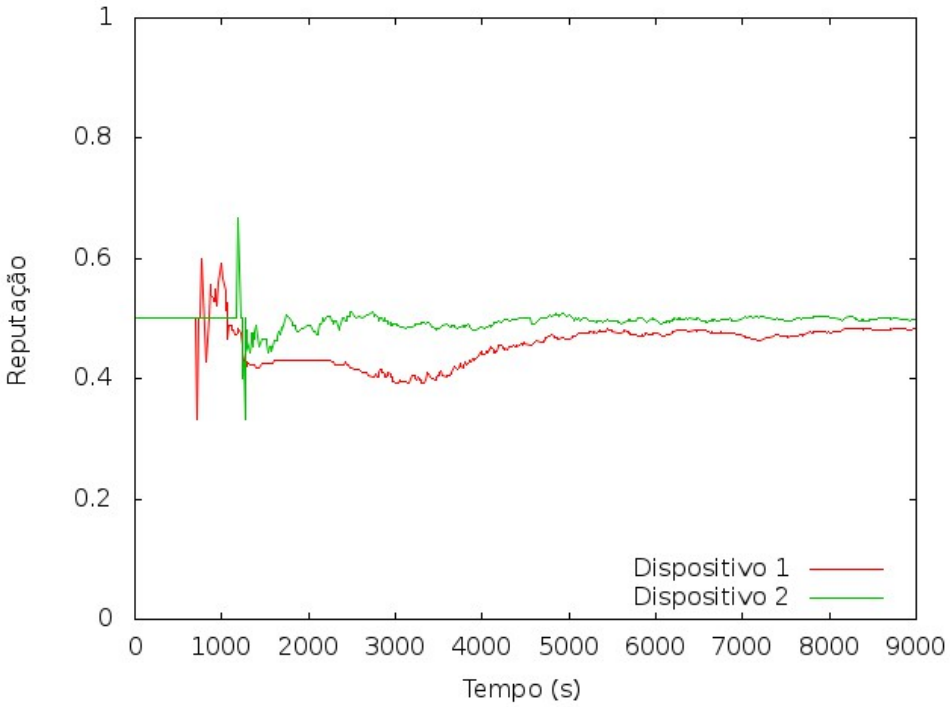

Figura A.3: Grupo 3: Esperança da distribuição beta

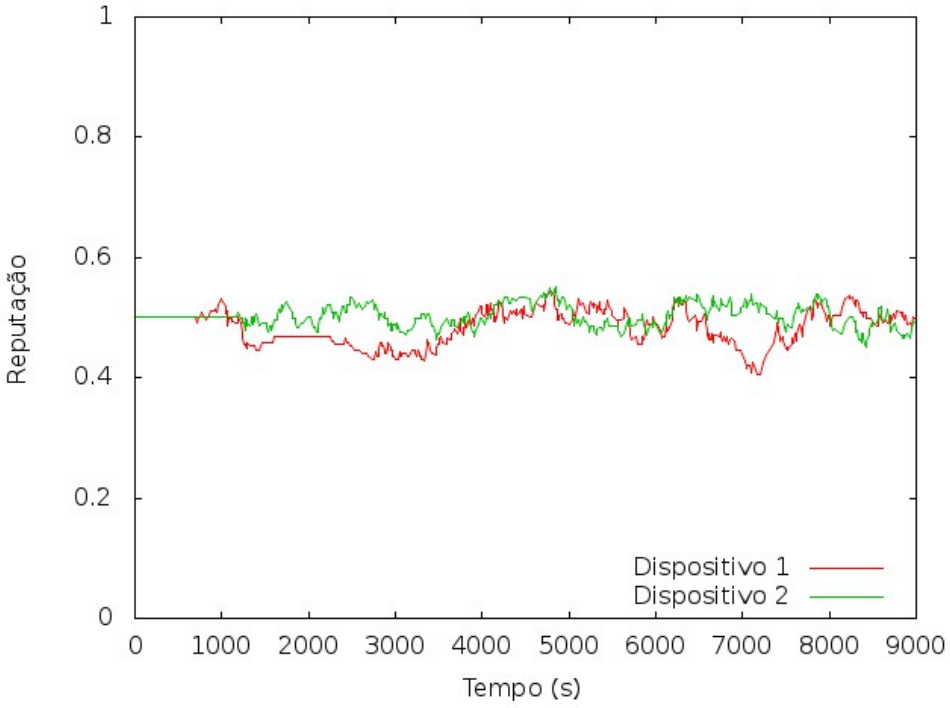

Figura A.4: Grupo 3: Esperança móvel da distribuição beta

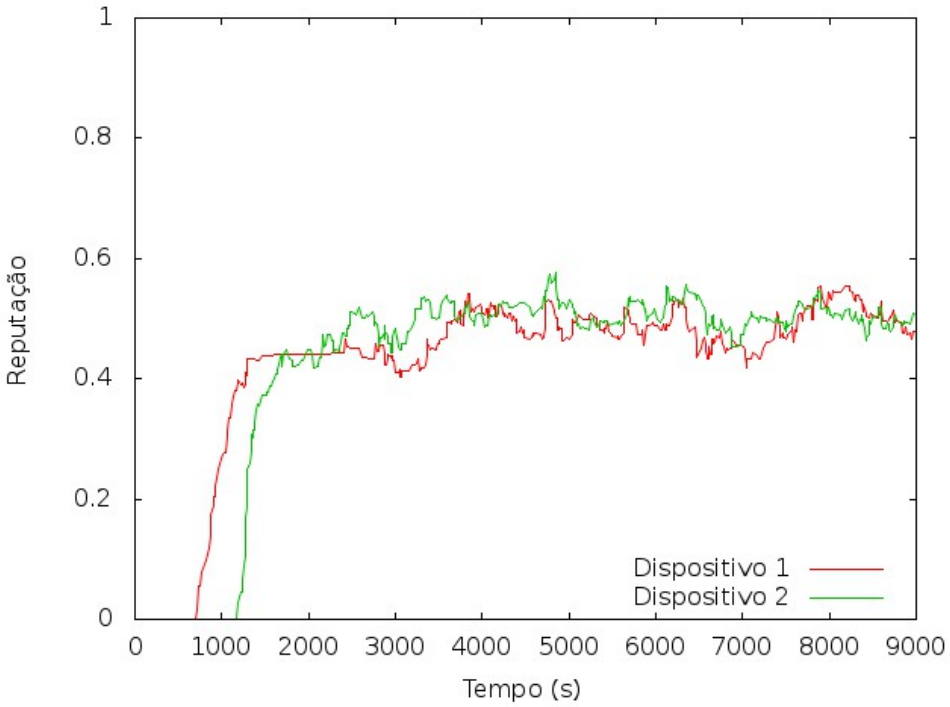

Figura A.5: Grupo 3: Proposta de formulação apresentada neste trabalho

Nos gráficos A.6, A.7, A.8, A.9 e A.10 estão representadas reputações do grupo com média $0,35$.

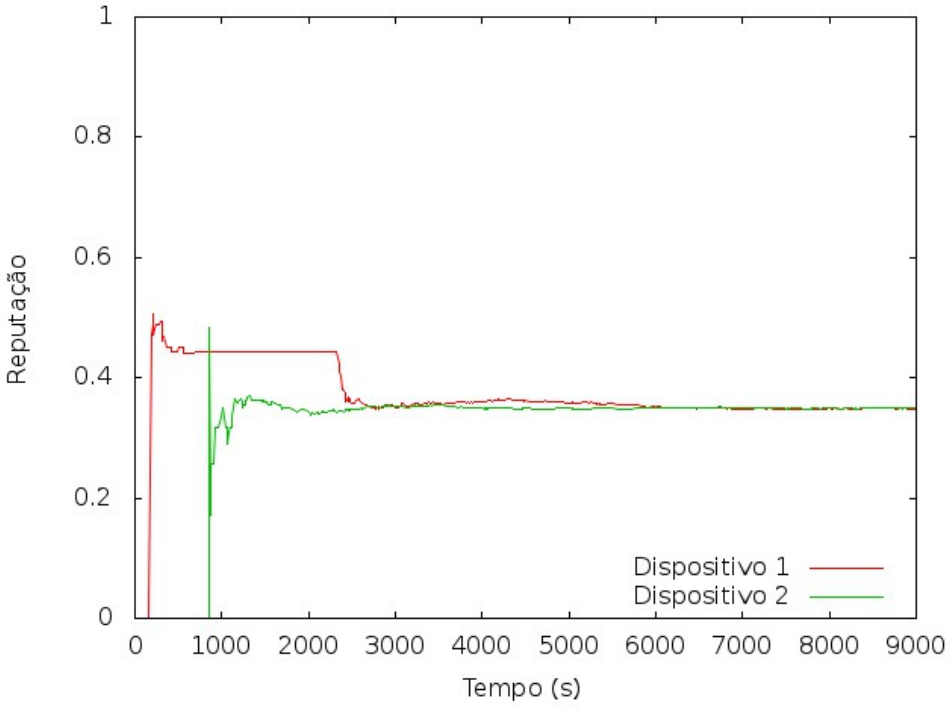

Figura A.6: Grupo 4: Média

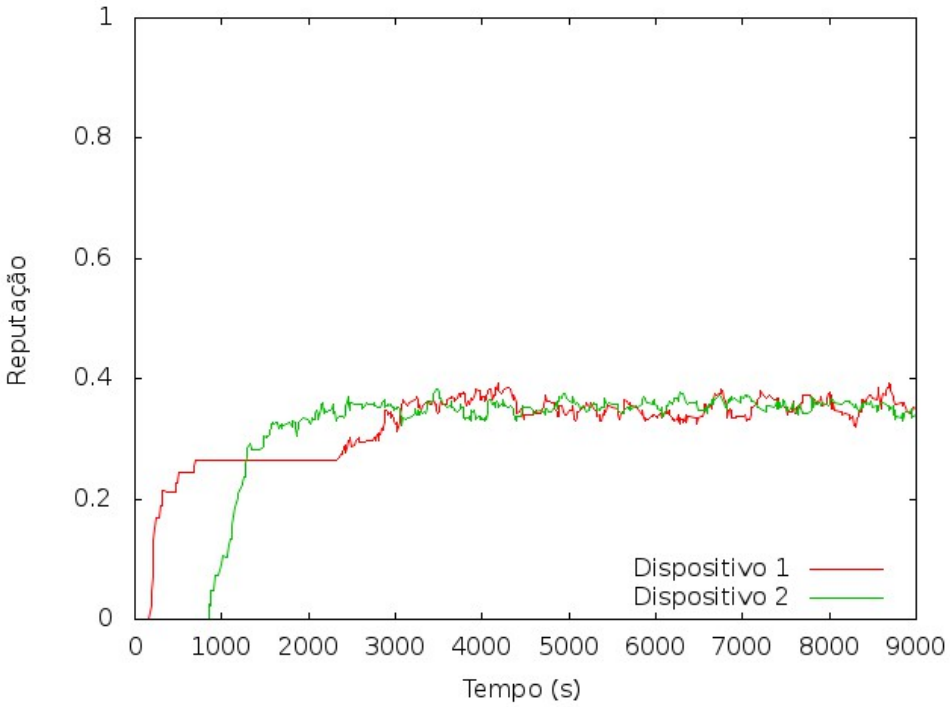

Figura A.7: Grupo 4: Média móvel

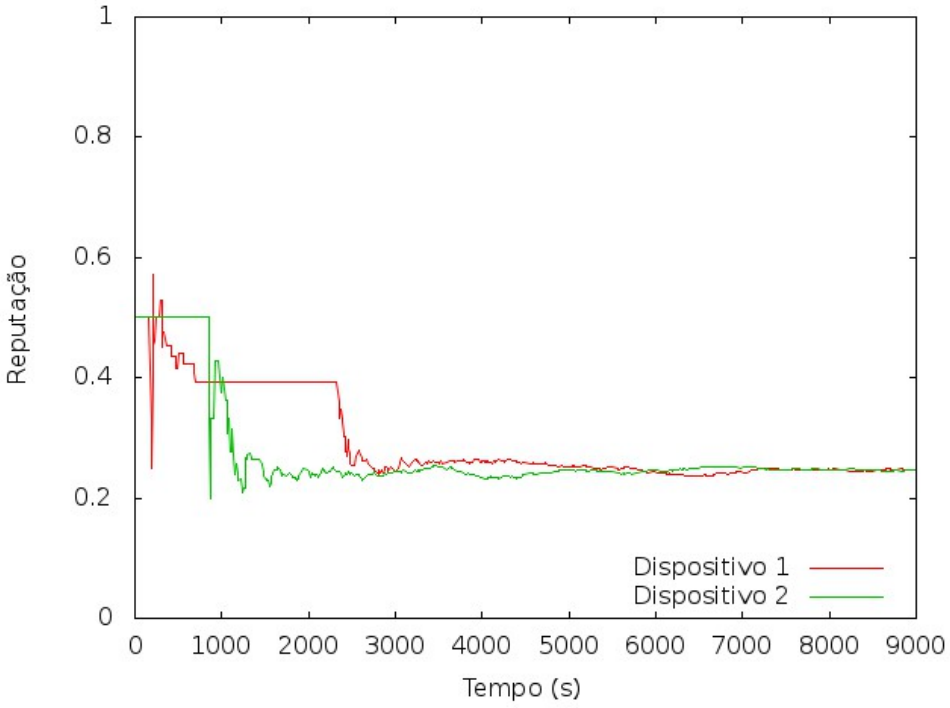

Figura A.8: Grupo 4: Esperança da distribuição beta

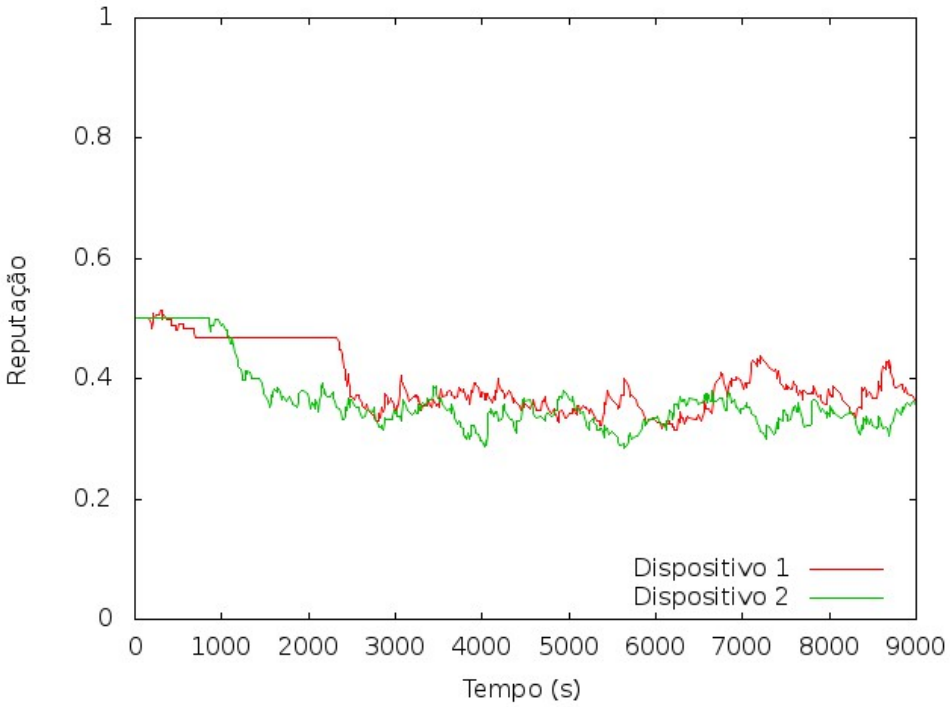

Figura A.9: Grupo 4: Esperança móvel da distribuição beta

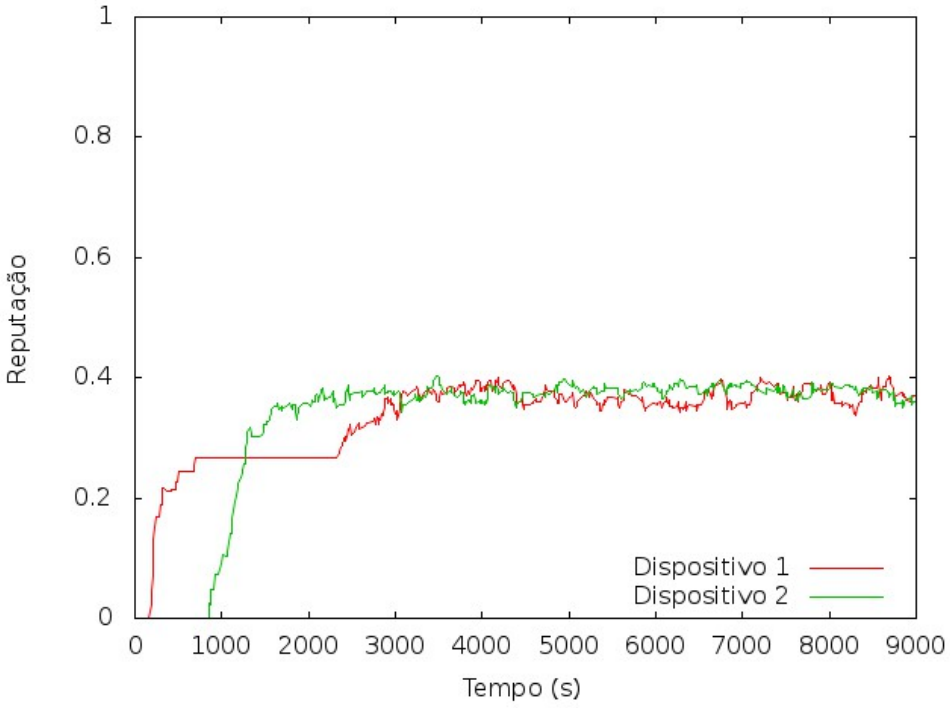

Figura A.10: Grupo 4: Proposta de formulação apresentada neste trabalho

E finalmente os gráficos A.11, A.12, A.13, A.14 e A.15 estão representadas reputações do grupo com média $0,2$.

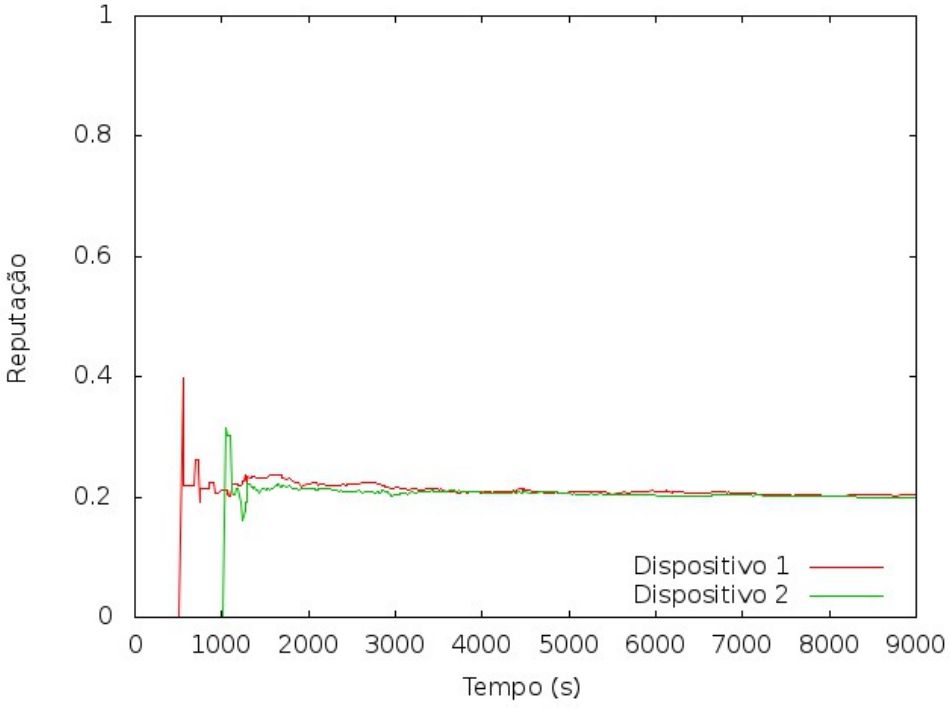

Figura A.11: Grupo 5: Média

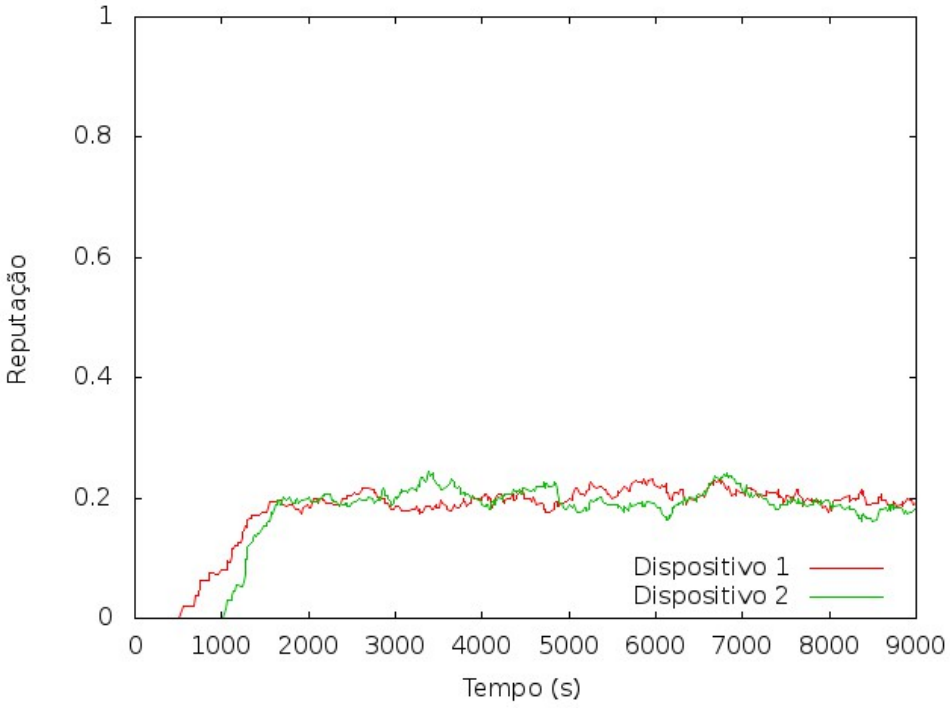

Figura A.12: Grupo 5: Média móvel

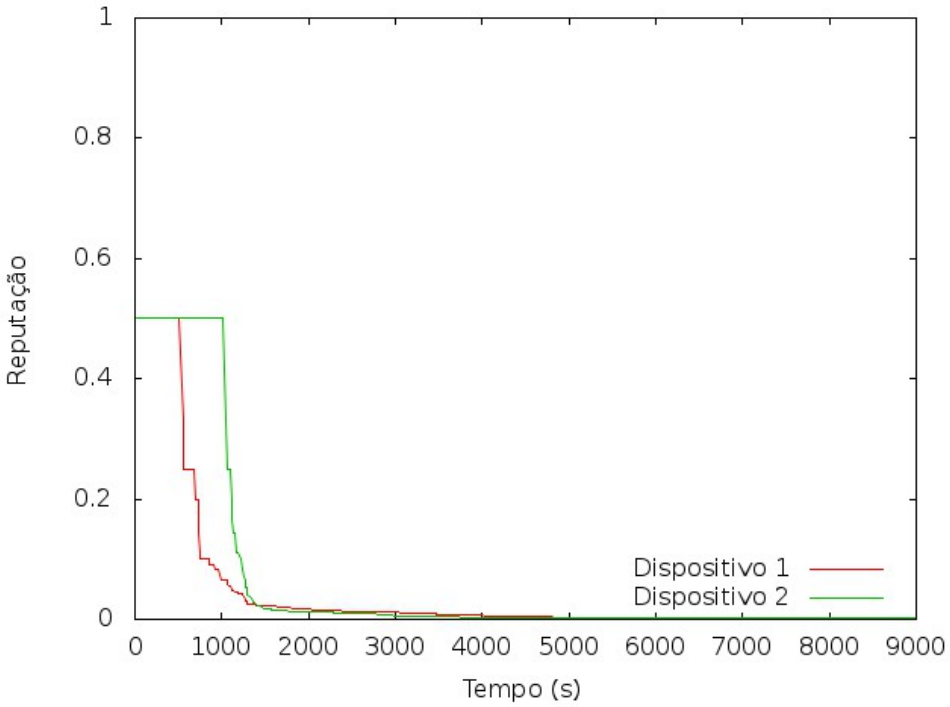

Figura A.13: Grupo 5: Esperança da distribuição beta

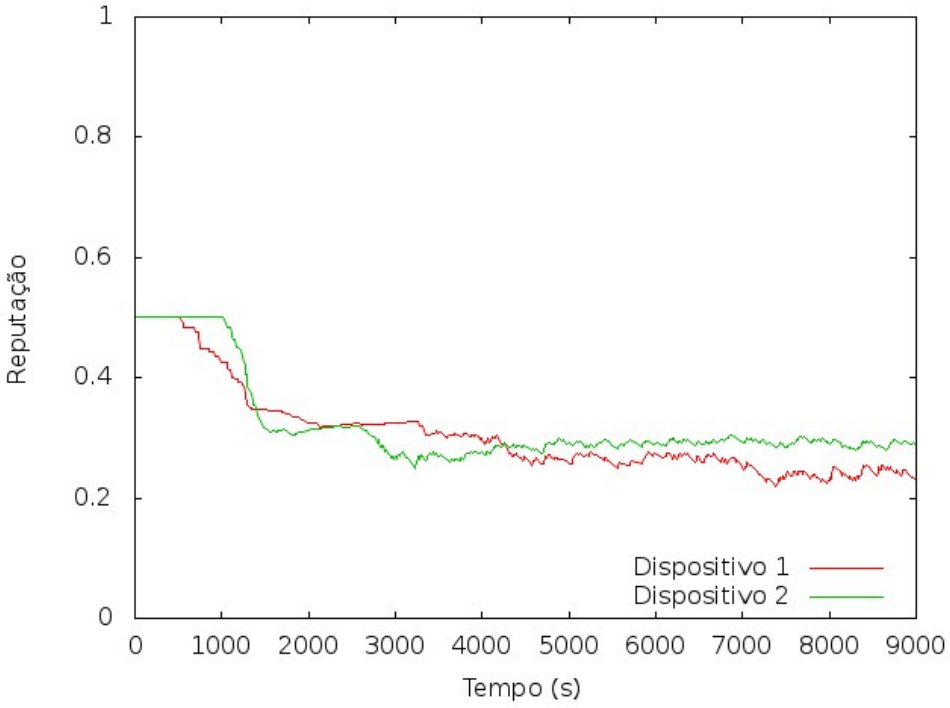

Figura A.14: Grupo 5: Esperança móvel da distribuição beta

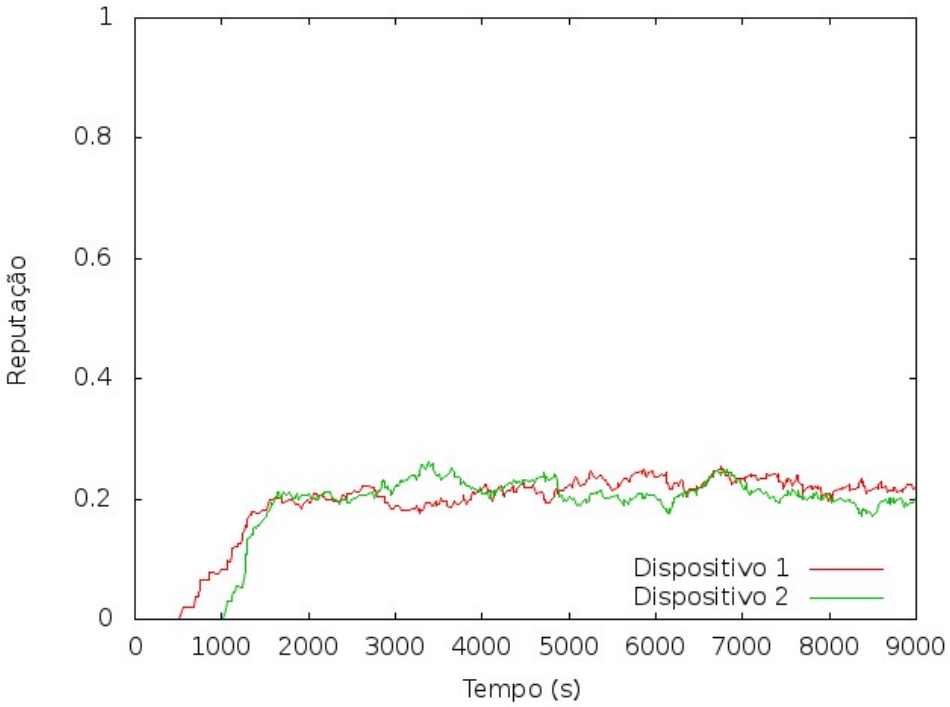

Figura A.15: Grupo 5: Proposta de formulação apresentada neste trabalho

A.2 Grupos de qualidade e Valores de α

As figuras A.16, A.17, A.18 exemplificam a influência dos valores de α na formulação da esperança móvel da distribuição beta no grupo 2 de qualidade.

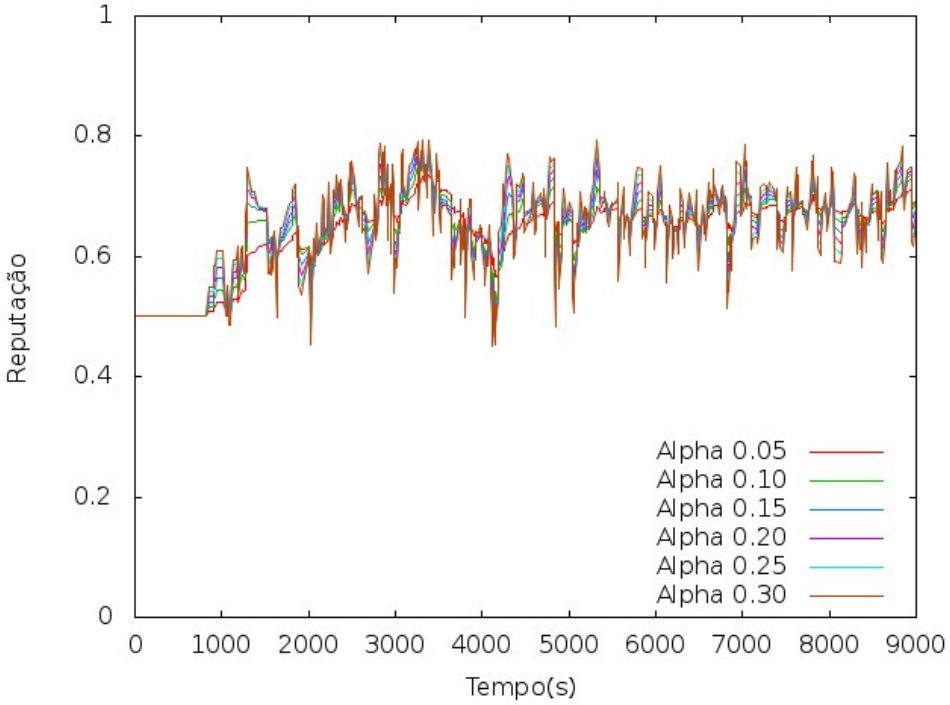

Figura A.16: Exemplo das curvas de reputação para o grupo 2 utilizando formulação da esperança móvel da distribuição beta com α fixo

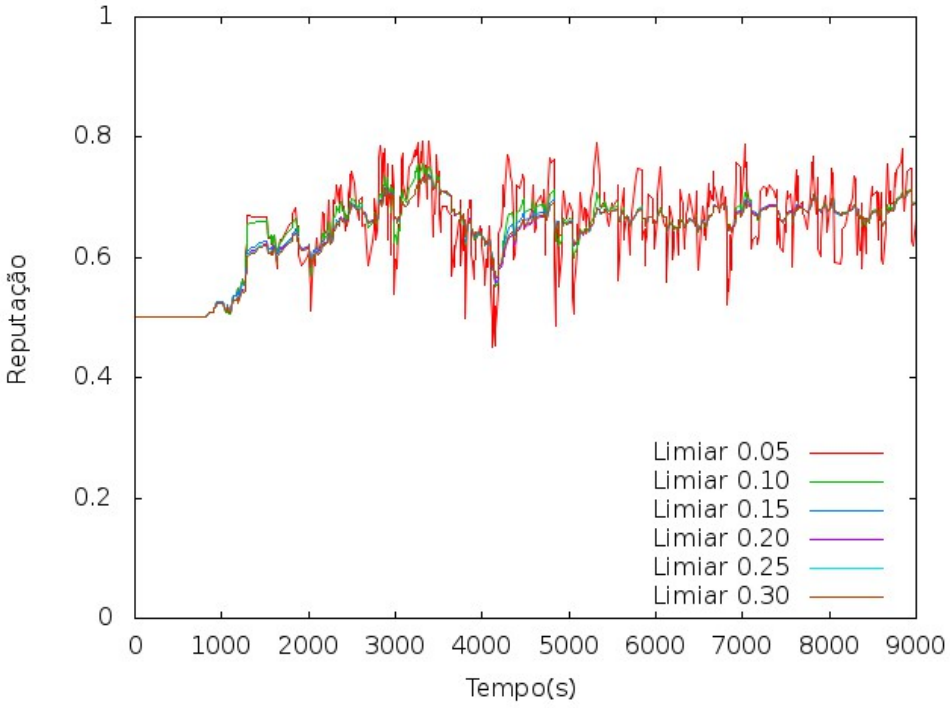

Figura A.17: Exemplo das curvas de reputação para o grupo 2 utilizando formulação da esperança móvel da distribuição beta com α linear

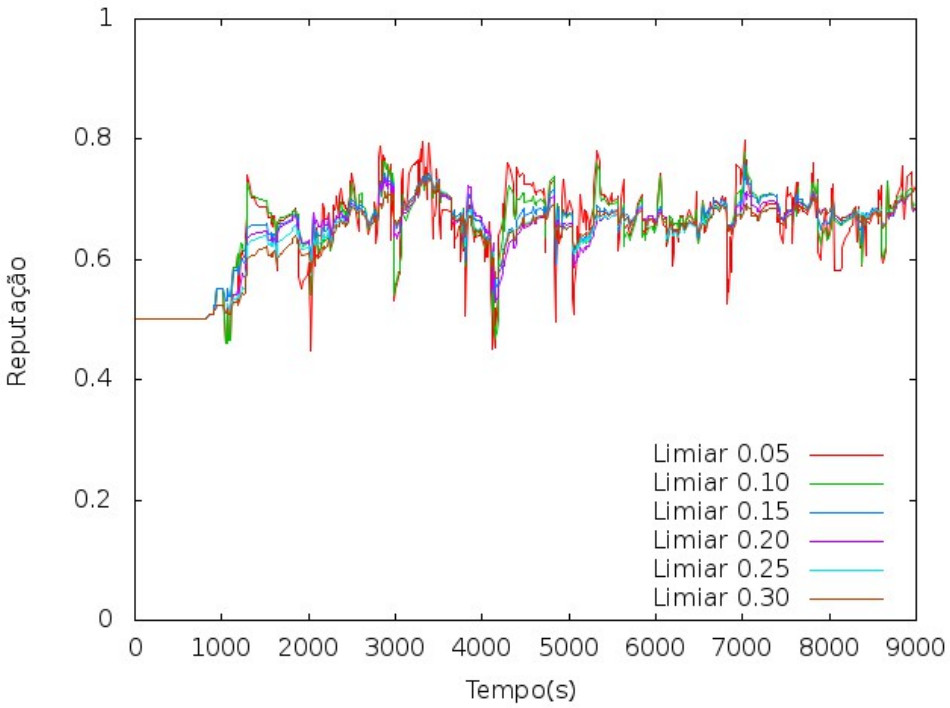

Figura A.18: Exemplo das curvas de reputação para o grupo 2 utilizando formulação da esperança móvel da distribuição beta com α escalar

As figuras A.19, A.20 e A.21 exibem a reputação do mesmo dispositivo, porém utilizando

a média móvel exponencial.

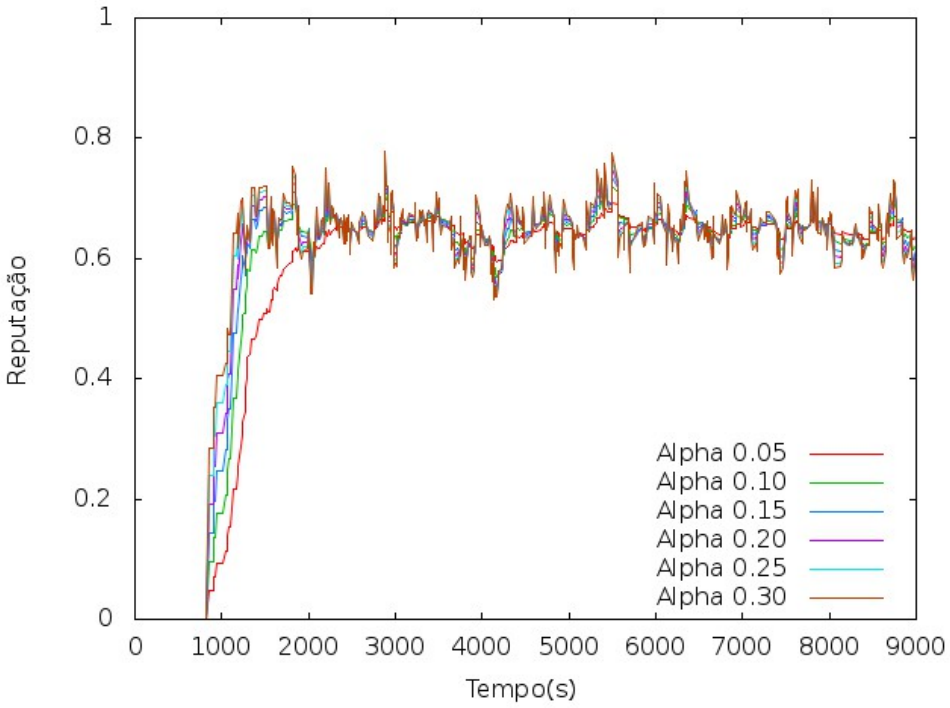

Figura A.19: Exemplo das curvas de reputação para o grupo 2 utilizando formulação da média móvel exponencial com α fixo

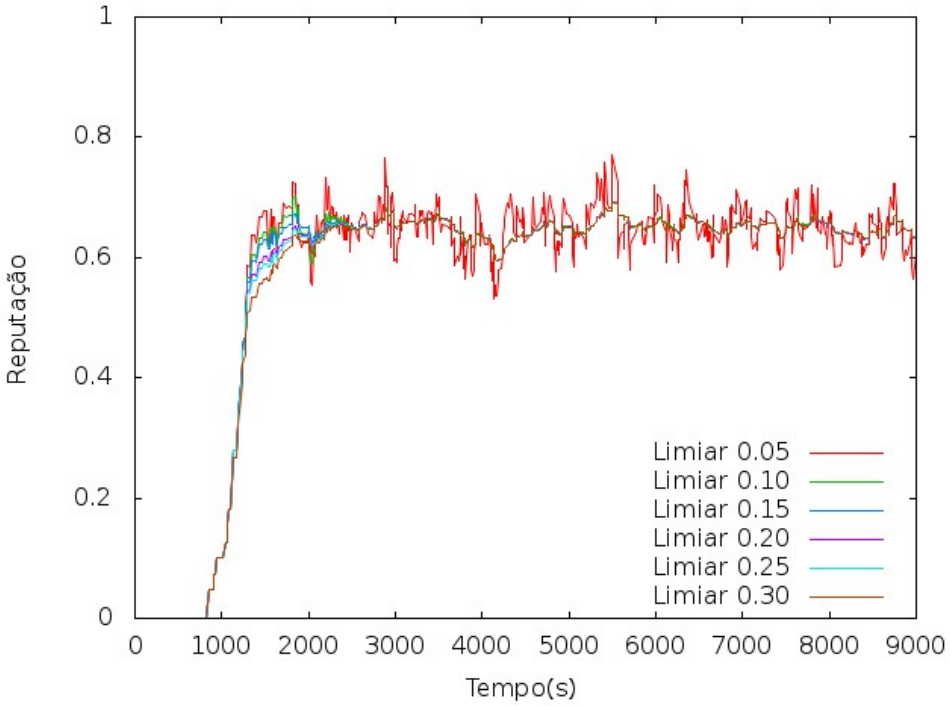

Figura A.20: Exemplo das curvas de reputação para o grupo 2 utilizando formulação da média móvel exponencial com α linear

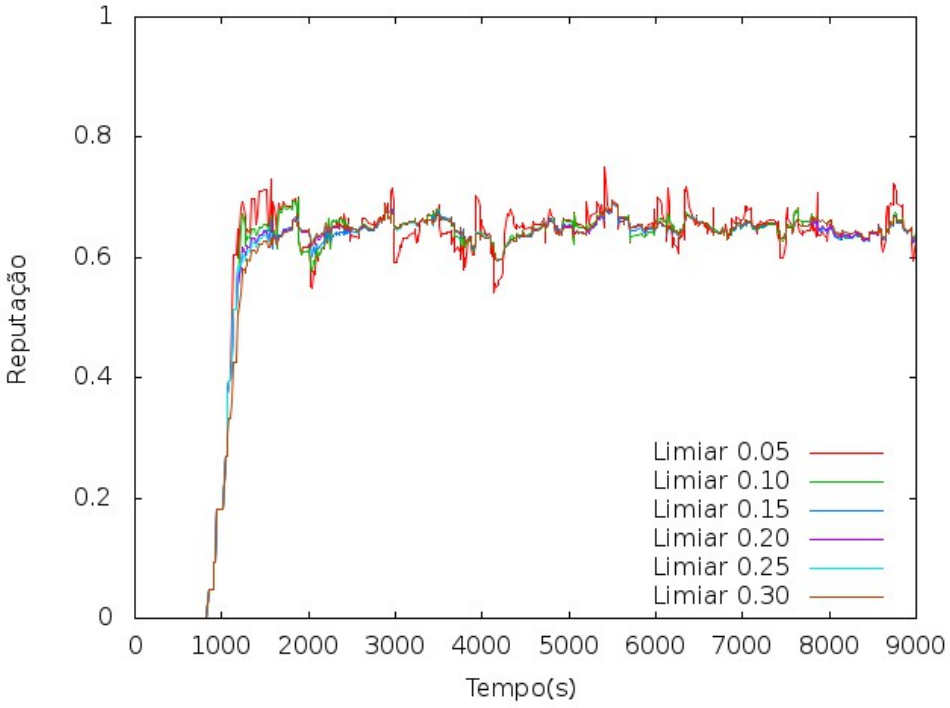

Figura A.21: Exemplo das curvas de reputação para o grupo 2 utilizando formulação da média móvel exponencial com α escalar

As figuras A.22, A.23 e A.24 mostram o comportamento das reputações utilizando a formulação proposta.

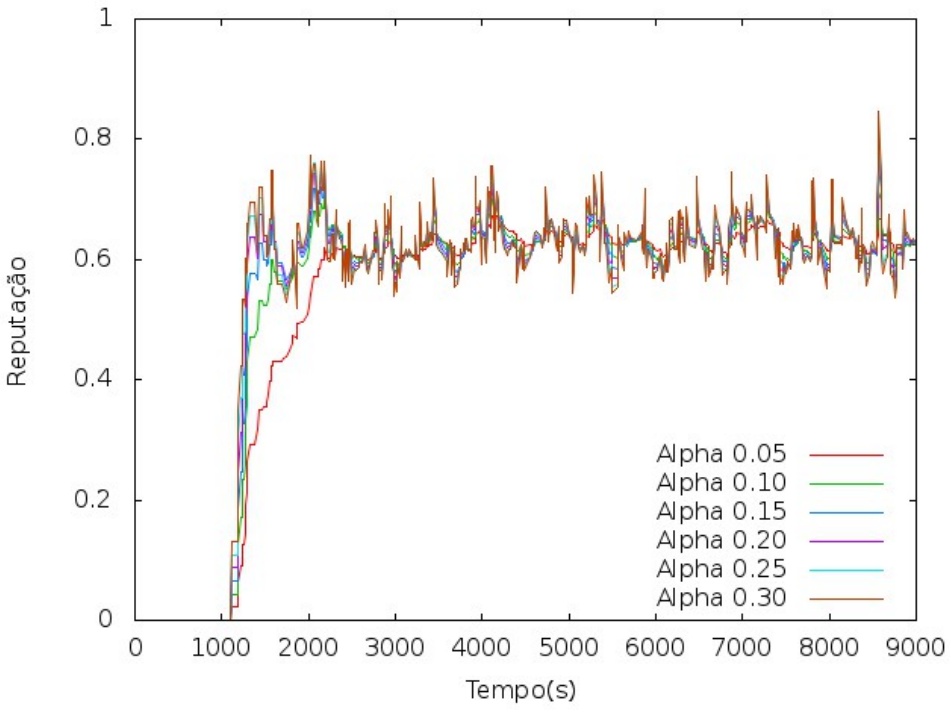

Figura A.22: Exemplo das curvas de reputação para o grupo 2 utilizando formulação proposta com α fixo

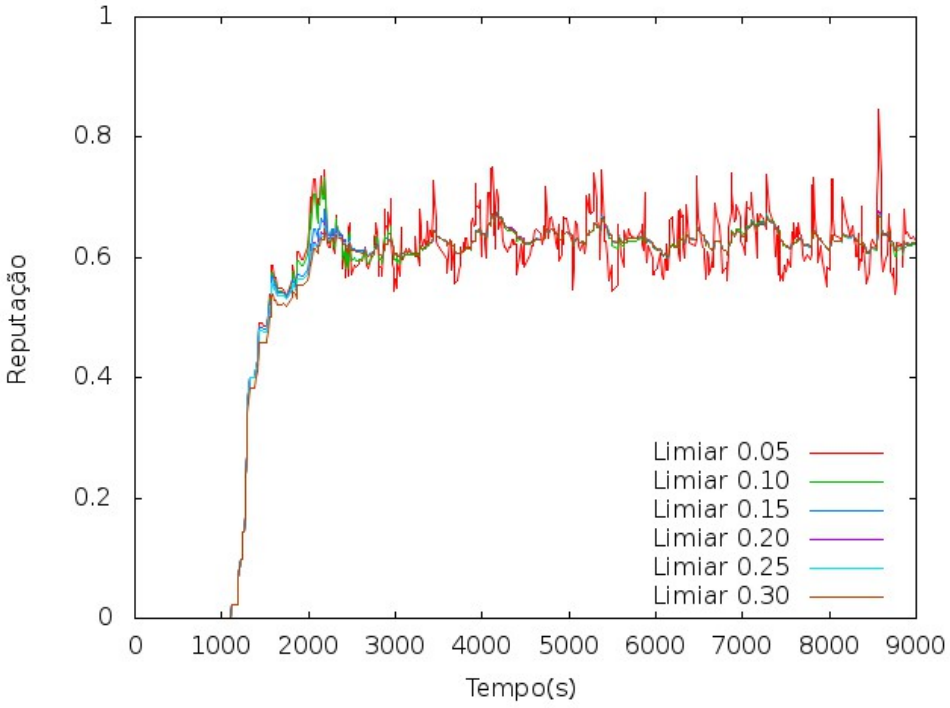

Figura A.23: Exemplo das curvas de reputação para o grupo 2 utilizando formulação proposta com α linear

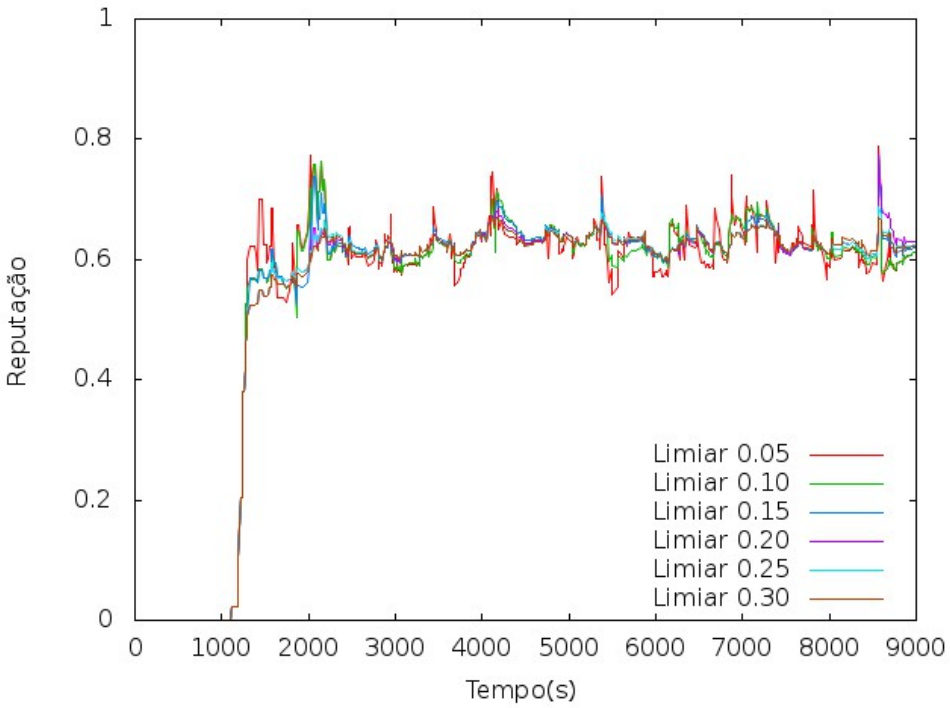

Figura A.24: Exemplo das curvas de reputação para o grupo 2 utilizando formulação proposta com α escalar

As figuras A.25, A.26, A.27 exemplificam a influência dos valores de α na formulação da

esperança móvel da distribuição beta no grupo 3 de qualidade.

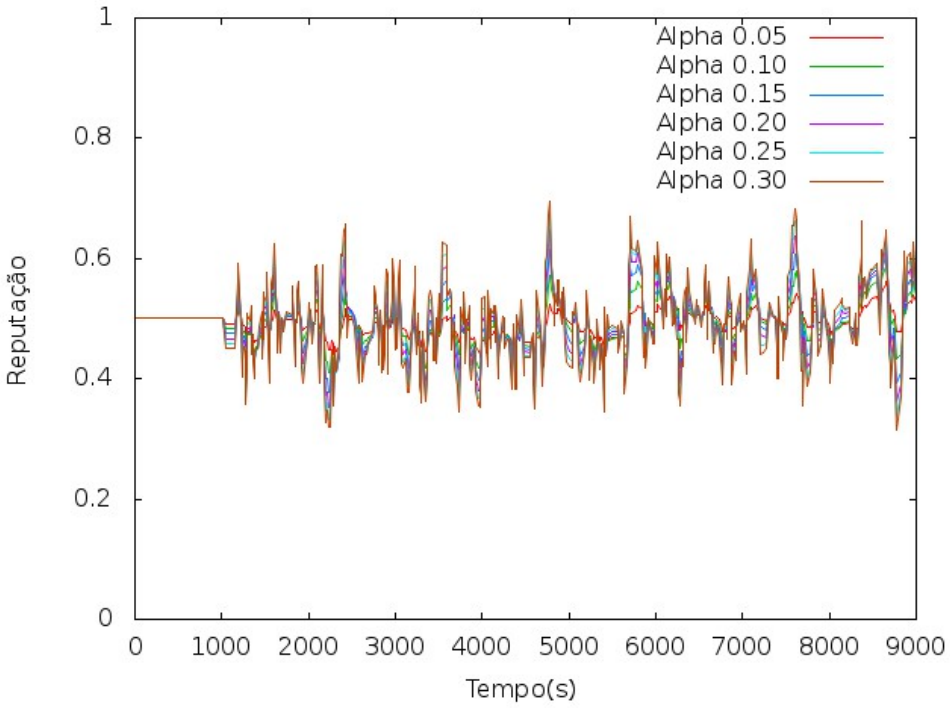

Figura A.25: Exemplo das curvas de reputação para o grupo 3 utilizando formulação da esperança móvel da distribuição beta com α fixo

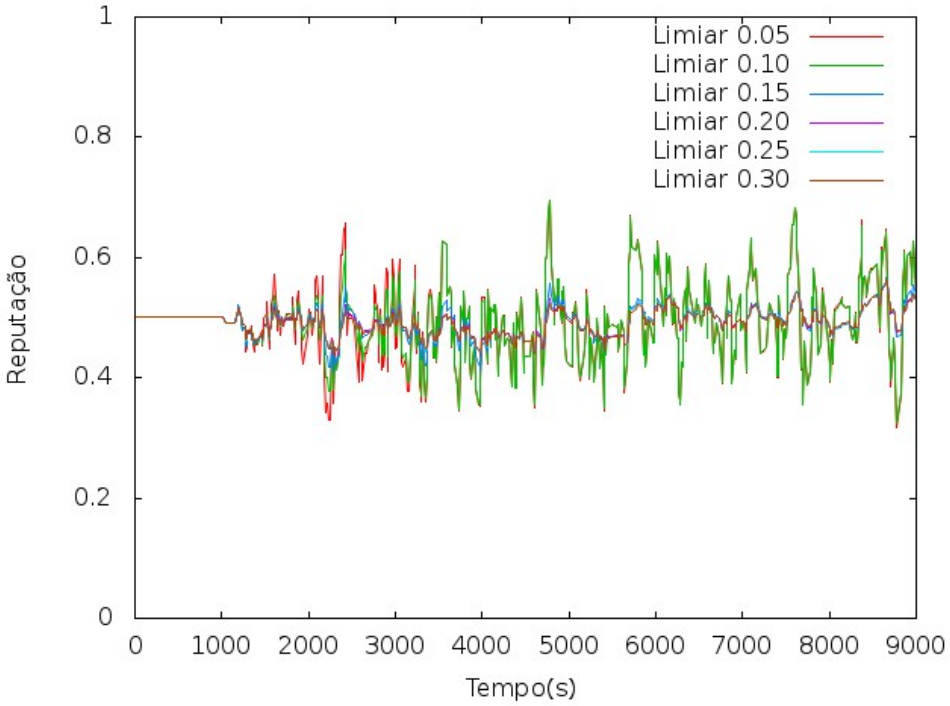

Figura A.26: Exemplo das curvas de reputação para o grupo 3 utilizando formulação da esperança móvel da distribuição beta com α linear

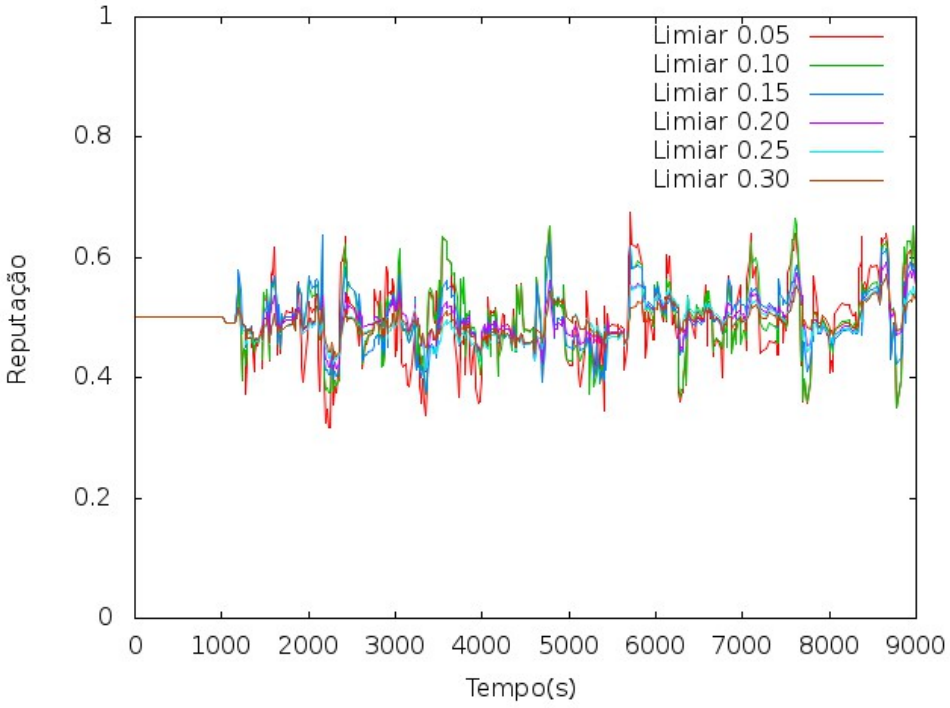

Figura A.27: Exemplo das curvas de reputação para o grupo 3 utilizando formulação da esperança móvel da distribuição beta com α escalar

As figuras A.28, A.29 e A.30 exibem a reputação do mesmo dispositivo, porém utilizando a média móvel exponencial.

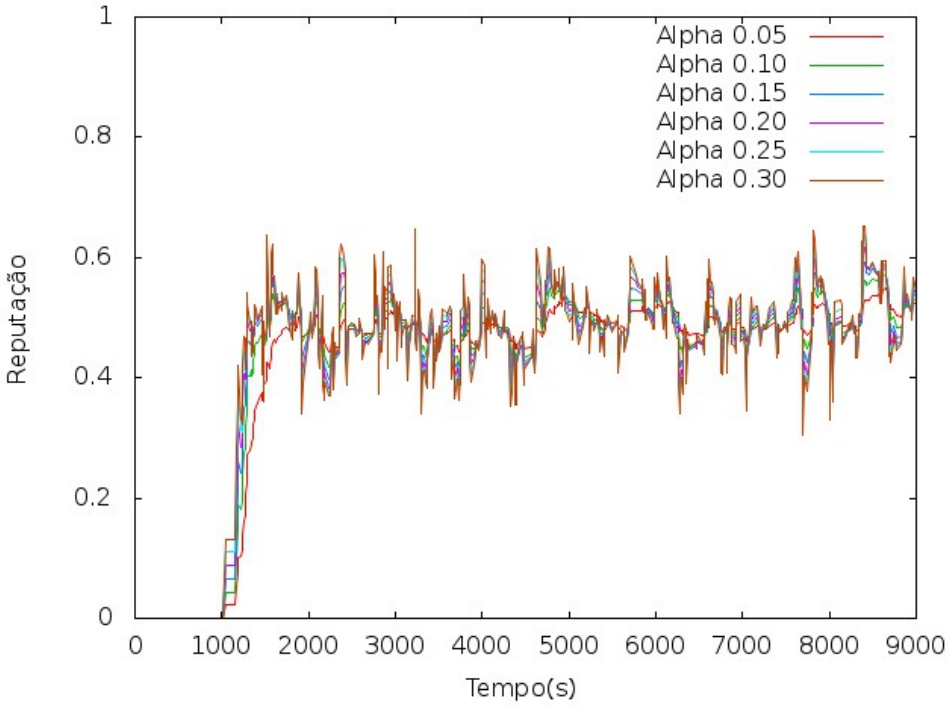

Figura A.28: Exemplo das curvas de reputação para o grupo 3 utilizando formulação da média móvel exponencial com α fixo

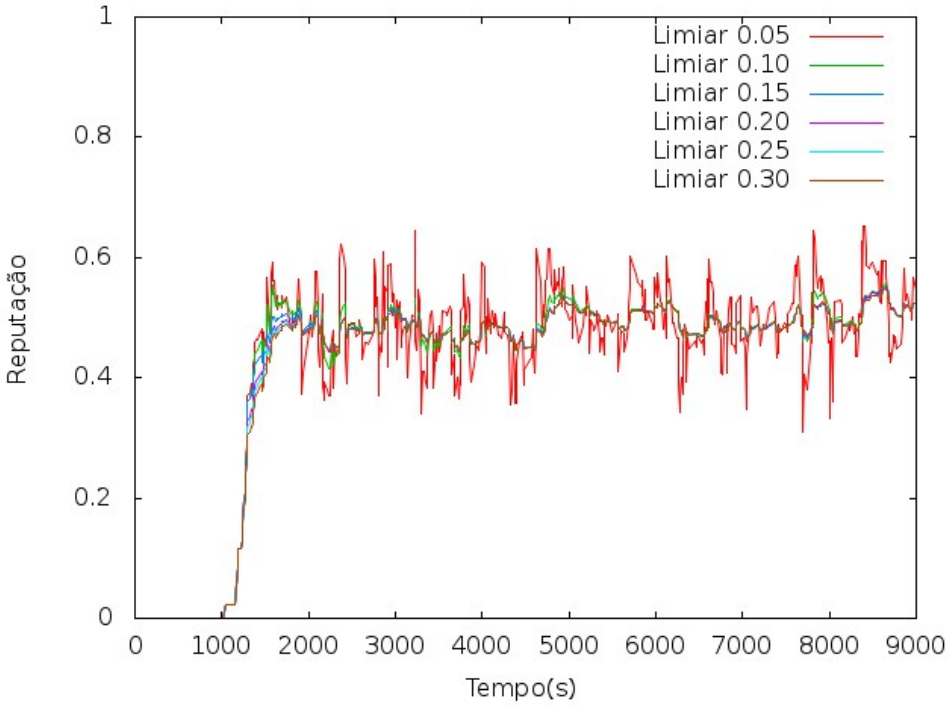

Figura A.29: Exemplo das curvas de reputação para o grupo 3 utilizando formulação da média móvel exponencial com α linear

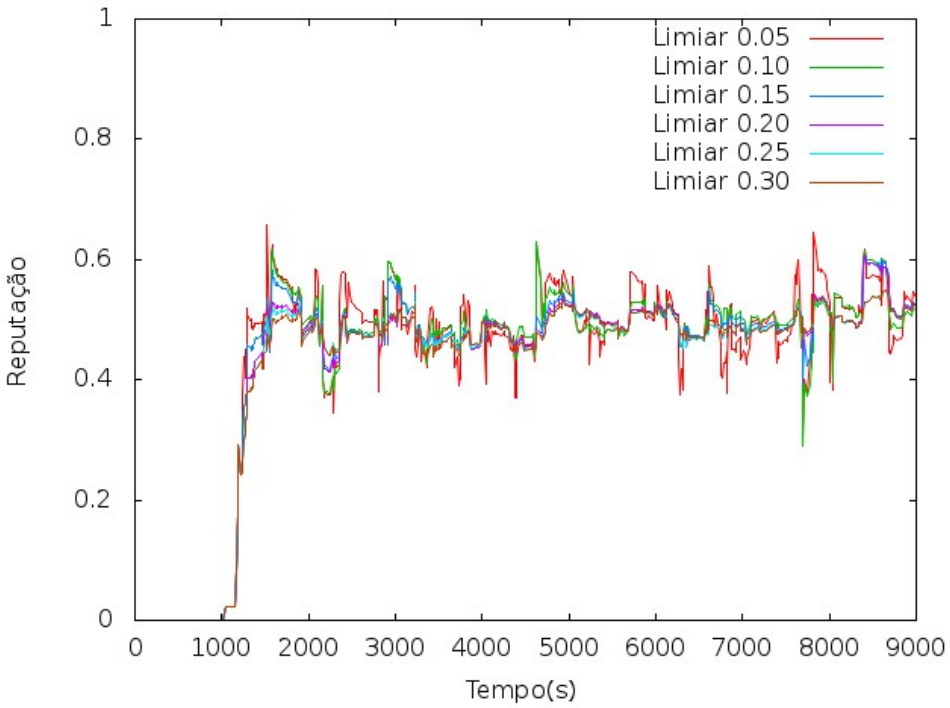

Figura A.30: Exemplo das curvas de reputação para o grupo 3 utilizando formulação da média móvel exponencial com α escalar

As figuras A.31, A.32 e A.33 mostram o comportamento das reputações utilizando a

formulação proposta.

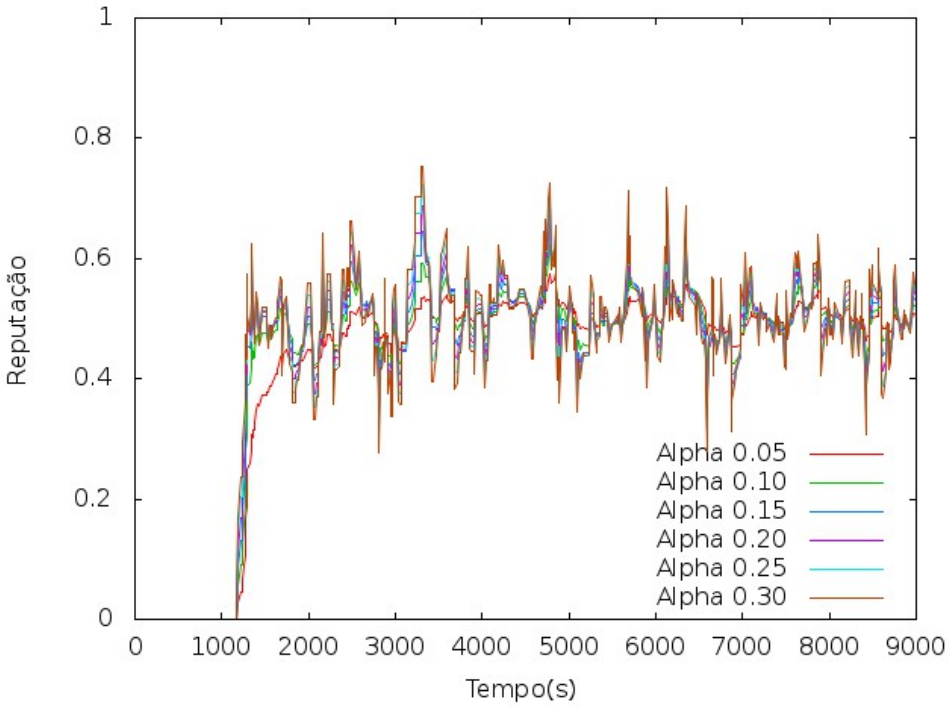

Figura A.31: Exemplo das curvas de reputação para o grupo 3 utilizando formulação proposta com α fixo

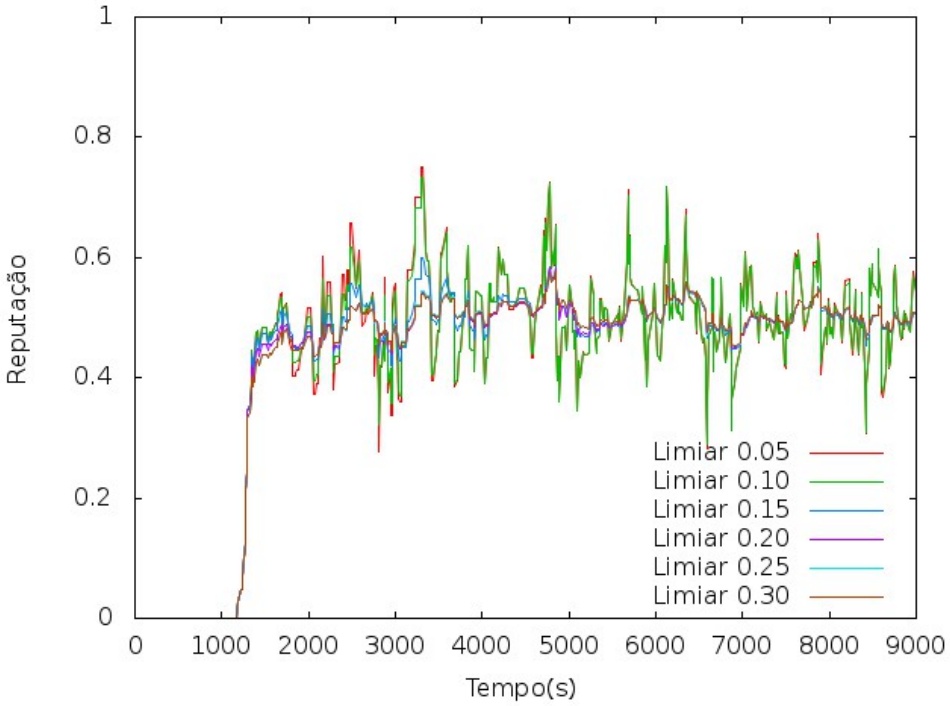

Figura A.32: Exemplo das curvas de reputação para o grupo 3 utilizando formulação proposta com α linear

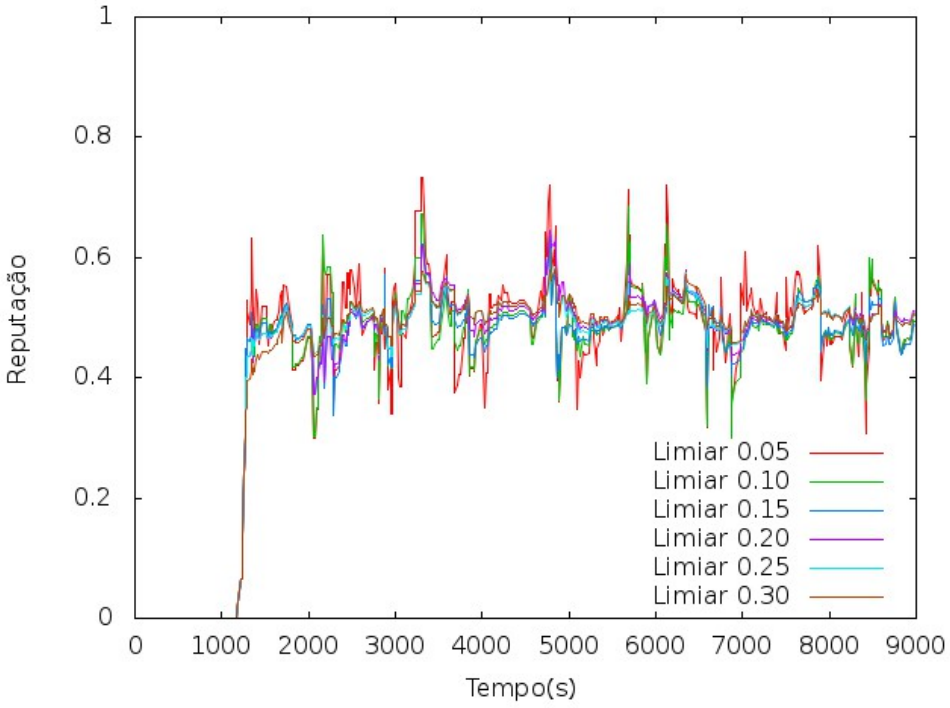

Figura A.33: Exemplo das curvas de reputação para o grupo 3 utilizando formulação proposta com α escalar

As figuras A.34, A.35, A.36 exemplificam a influência dos valores de α na formulação da esperança móvel da distribuição beta no grupo 4 de qualidade.

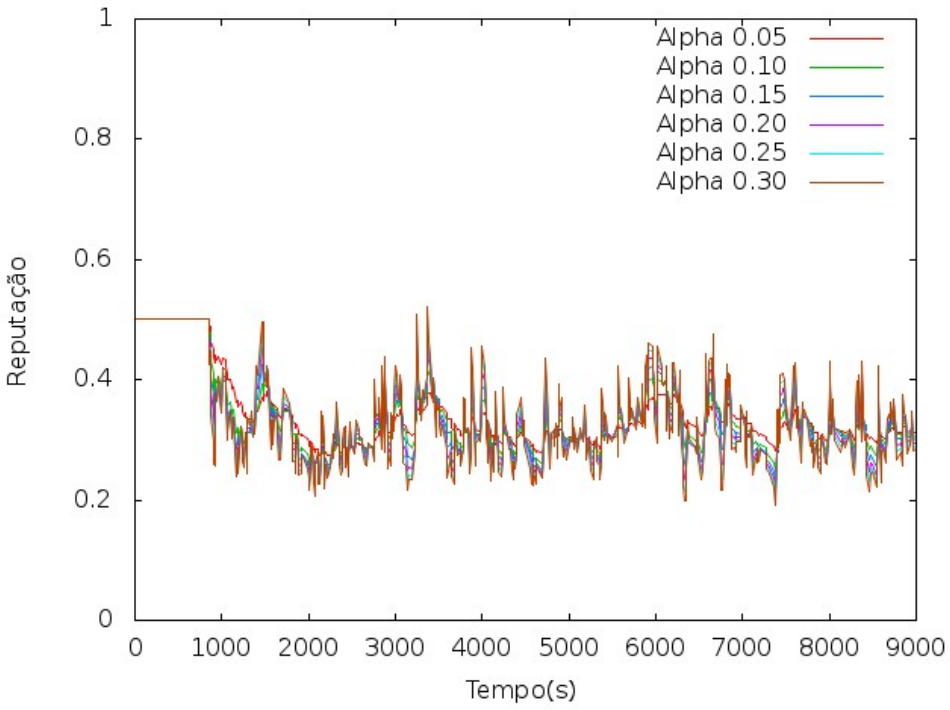

Figura A.34: Exemplo das curvas de reputação para o grupo 4 utilizando formulação da esperança móvel da distribuição beta com α fixo

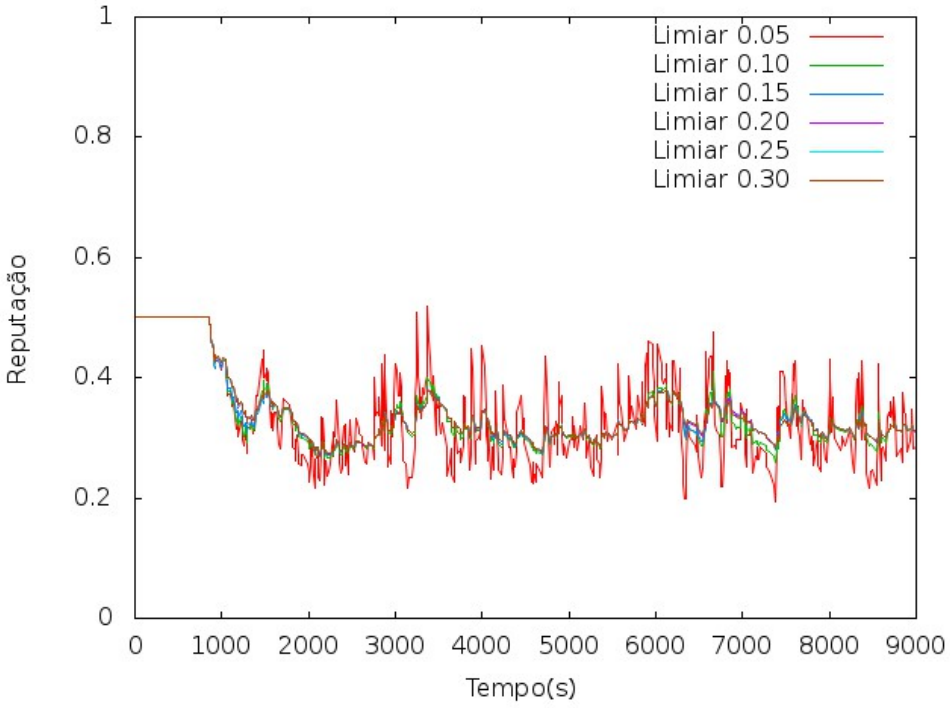

Figura A.35: Exemplo das curvas de reputação para o grupo 4 utilizando formulação da esperança móvel da distribuição beta com α linear

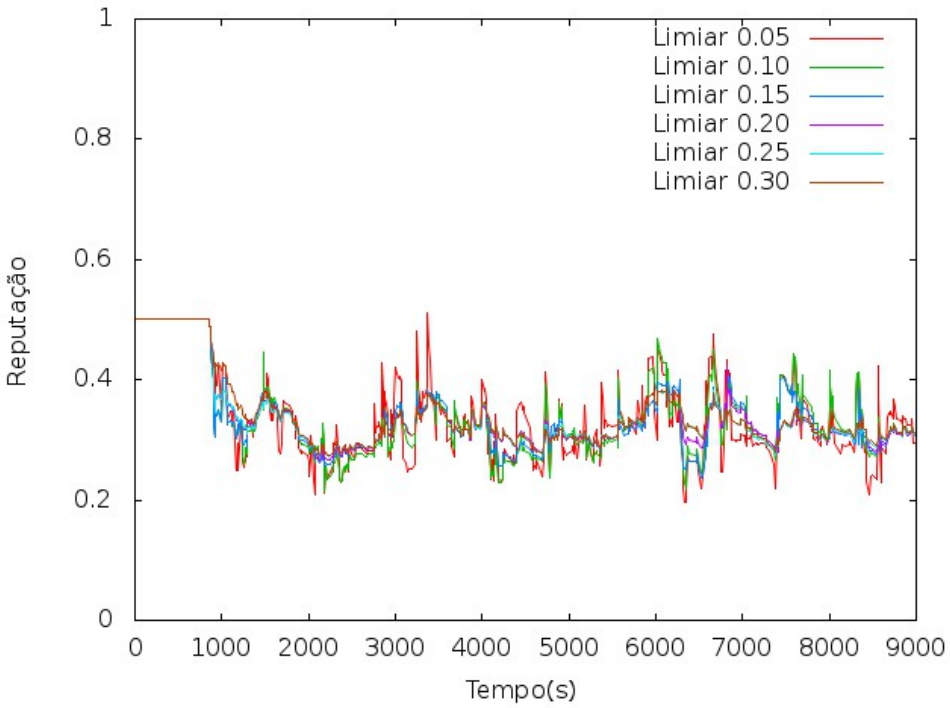

Figura A.36: Exemplo das curvas de reputação para o grupo 4 utilizando formulação da esperança móvel da distribuição beta com α escalar

As figuras A.37, A.38 e A.39 exibem a reputação do mesmo dispositivo, porém utilizando

a média móvel exponencial.

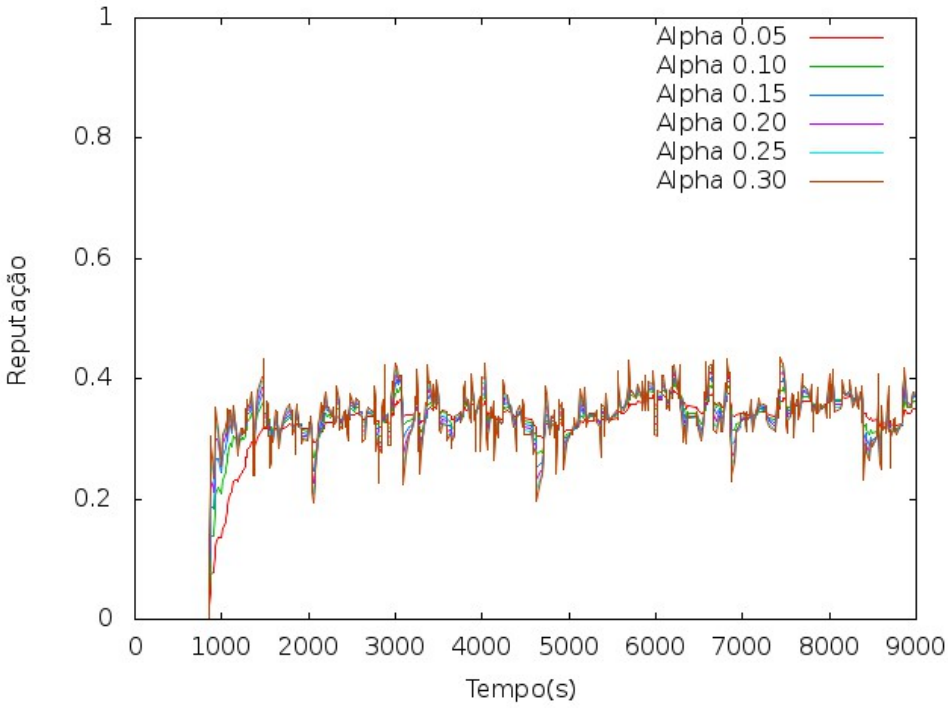

Figura A.37: Exemplo das curvas de reputação para o grupo 4 utilizando formulação da média móvel exponencial com α fixo

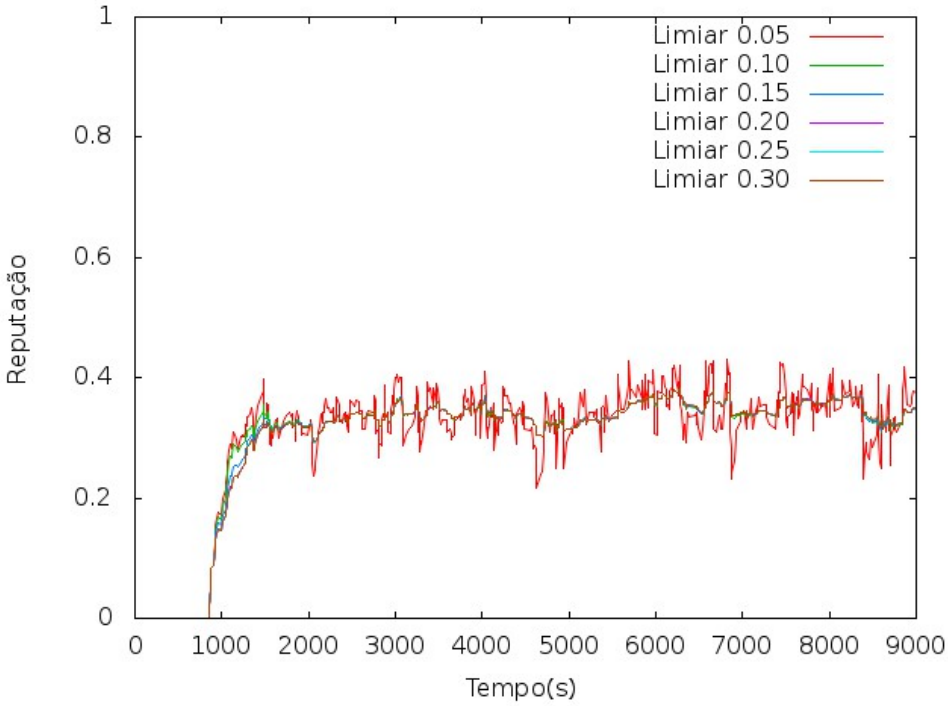

Figura A.38: Exemplo das curvas de reputação para o grupo 4 utilizando formulação da média móvel exponencial com α linear

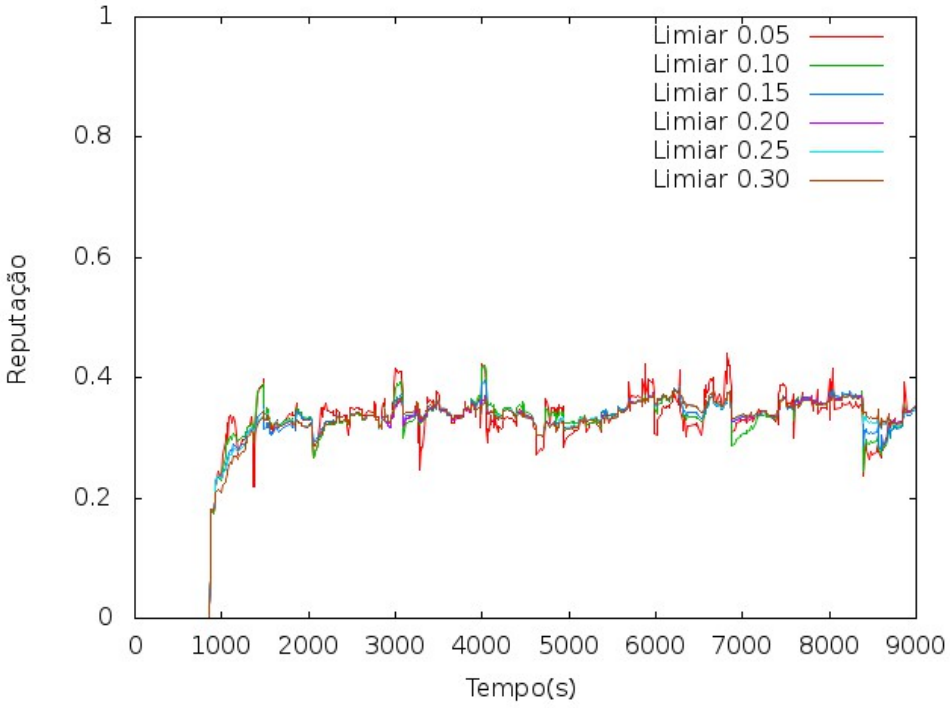

Figura A.39: Exemplo das curvas de reputação para o grupo 4 utilizando formulação da média móvel exponencial com α escalar

As figuras A.40, A.41 e A.42 mostram o comportamento das reputações utilizando a formulação proposta.

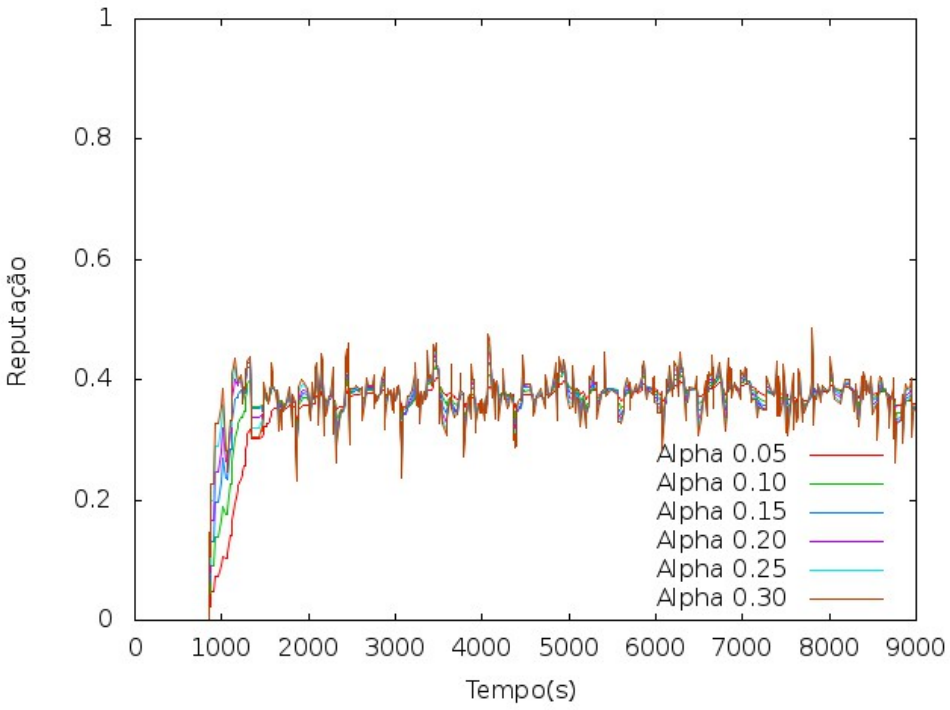

Figura A.40: Exemplo das curvas de reputação para o grupo 4 utilizando formulação proposta com α fixo

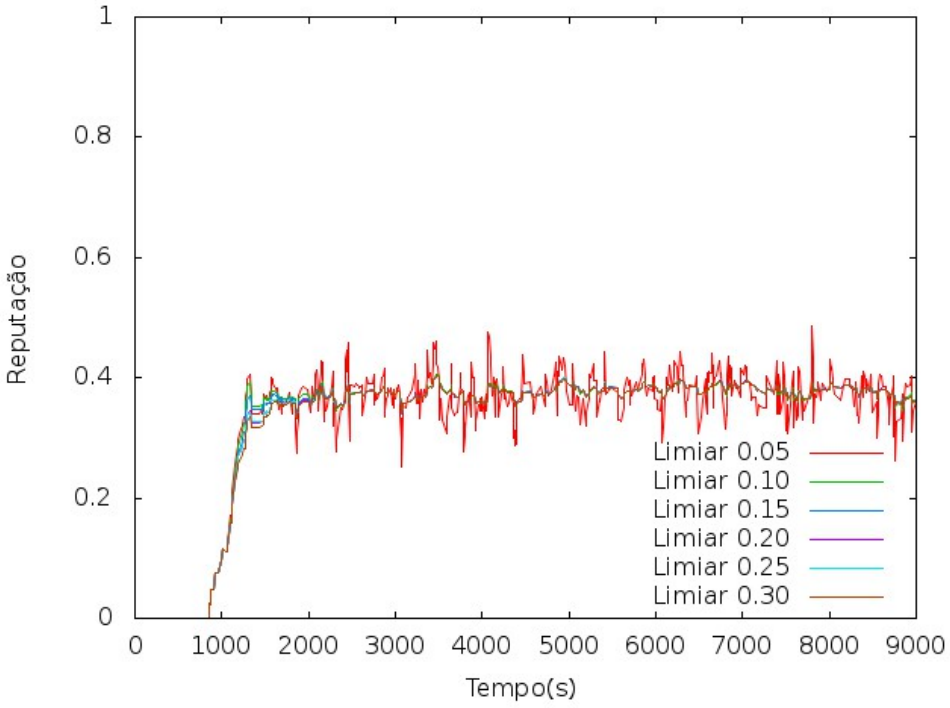

Figura A.41: Exemplo das curvas de reputação para o grupo 4 utilizando formulação proposta com α linear

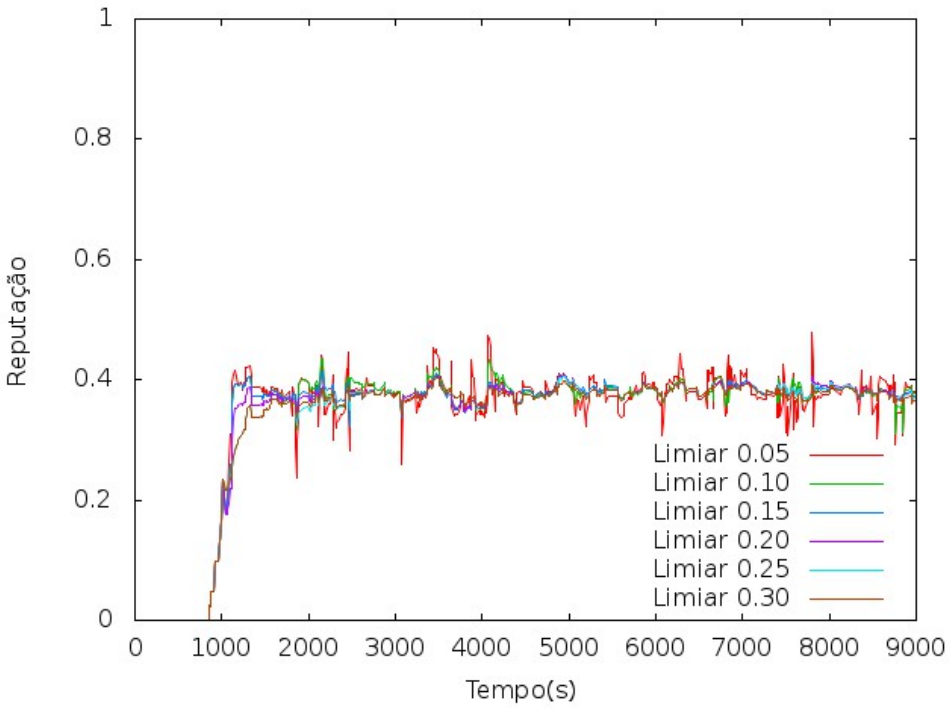

Figura A.42: Exemplo das curvas de reputação para o grupo 4 utilizando formulação proposta com α escalar

As figuras A.43, A.44, A.45 exemplificam a influência dos valores de α na formulação da

esperança móvel da distribuição beta no grupo 5 de qualidade.

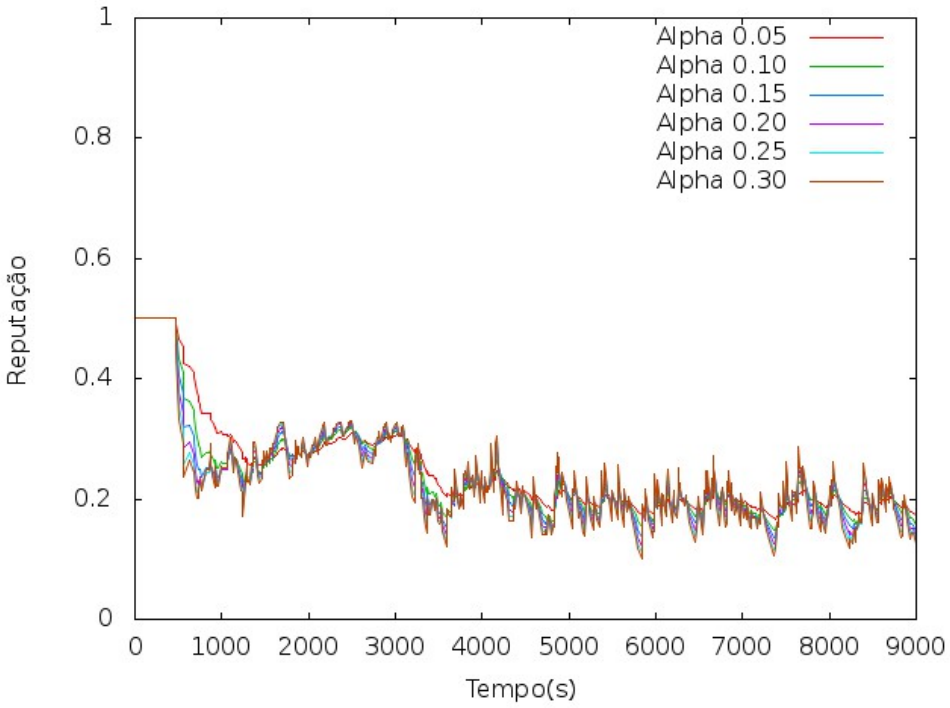

Figura A.43: Exemplo das curvas de reputação para o grupo 5 utilizando formulação da esperança móvel da distribuição beta com α fixo

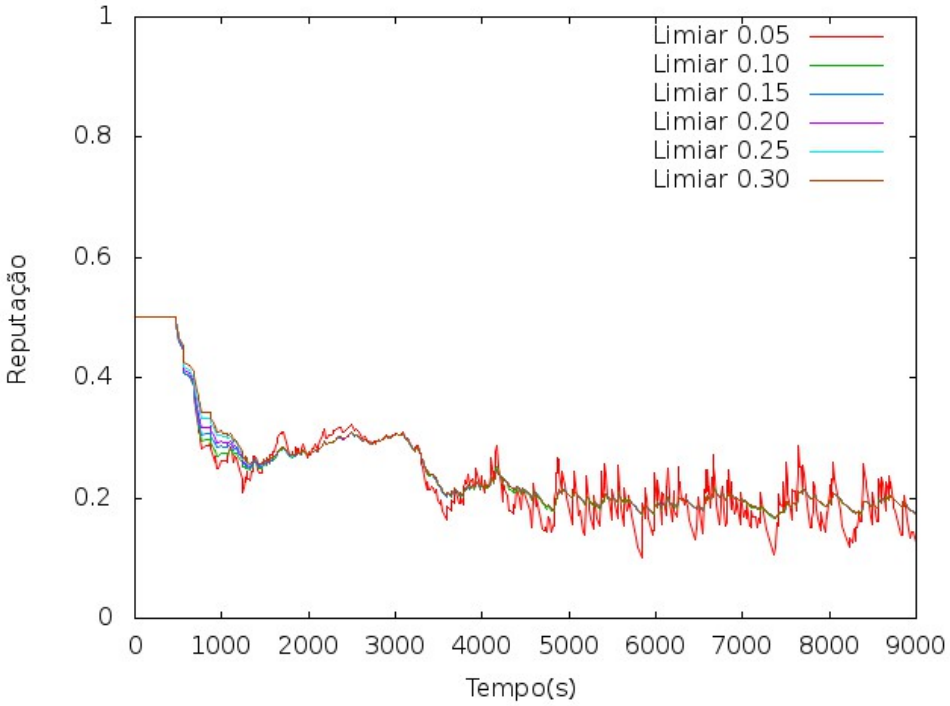

Figura A.44: Exemplo das curvas de reputação para o grupo 5 utilizando formulação da esperança móvel da distribuição beta com α linear

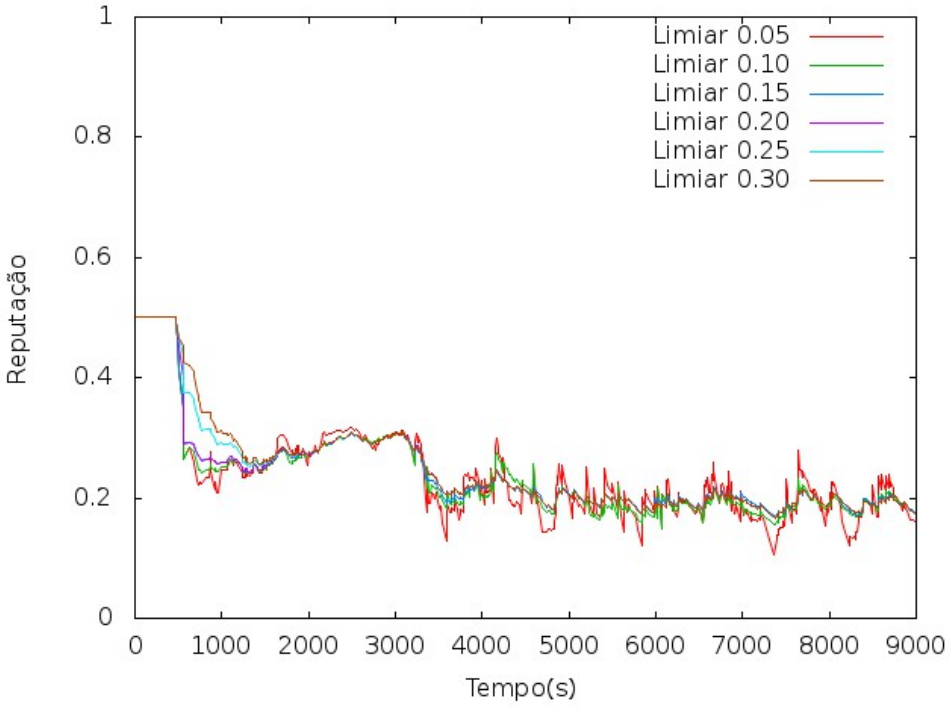

Figura A.45: Exemplo das curvas de reputação para o grupo 5 utilizando formulação da esperança móvel da distribuição beta com α escalar

As figuras A.46, A.47 e A.48 exibem a reputação do mesmo dispositivo, porém utilizando a média móvel exponencial.

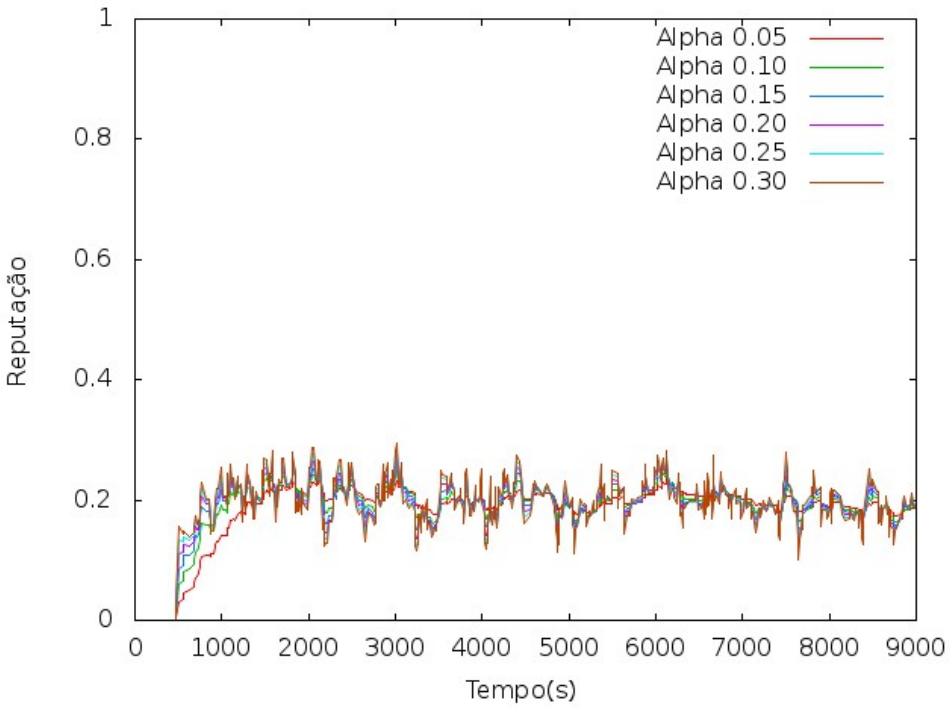

Figura A.46: Exemplo das curvas de reputação para o grupo 5 utilizando formulação da média móvel exponencial com α fixo

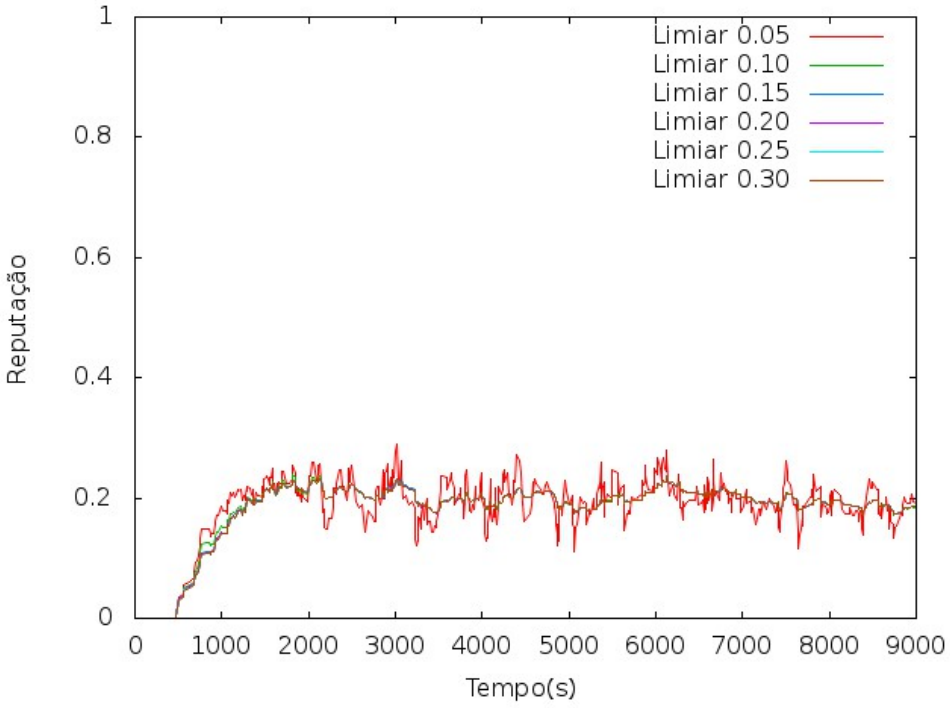

Figura A.47: Exemplo das curvas de reputação para o grupo 5 utilizando formulação da média móvel exponencial com α linear

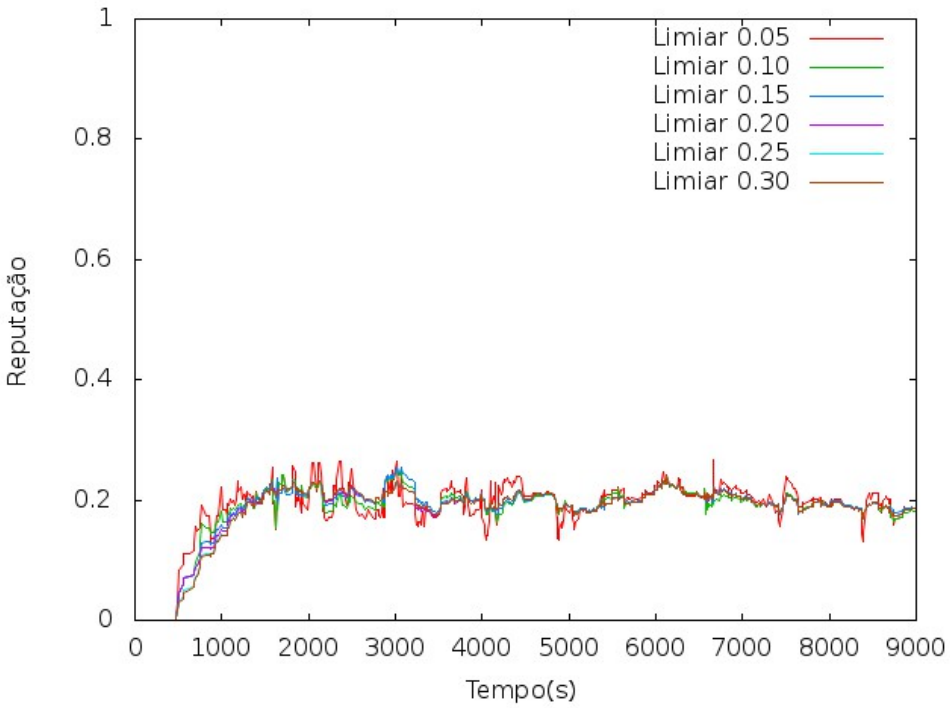

Figura A.48: Exemplo das curvas de reputação para o grupo 5 utilizando formulação da média móvel exponencial com α escalar

As figuras A.49, A.50 e A.51 mostram o comportamento das reputações utilizando a

formulação proposta.

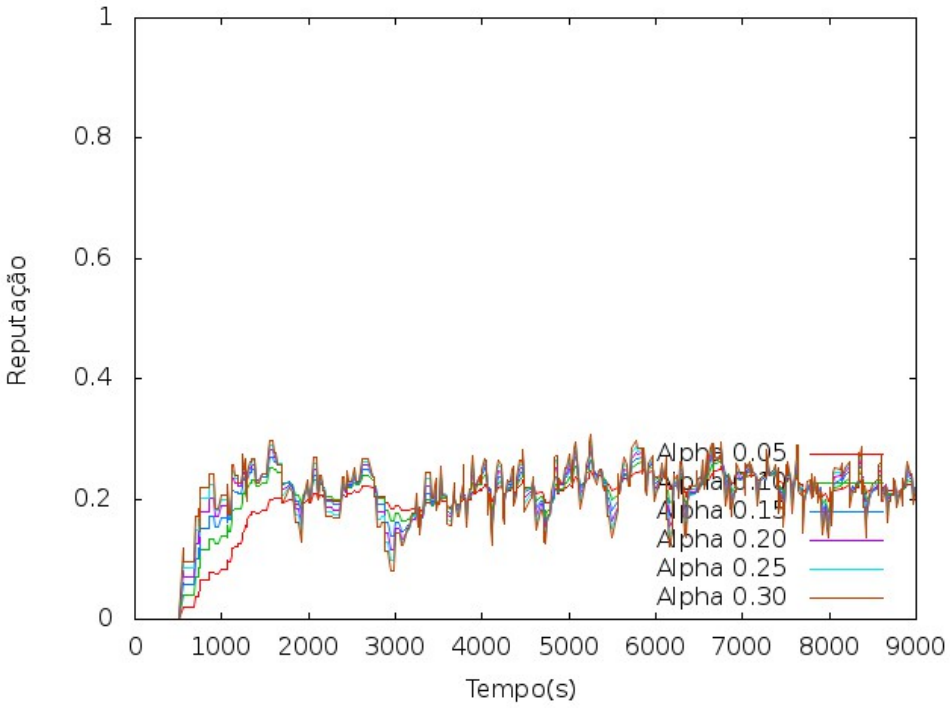

Figura A.49: Exemplo das curvas de reputação para o grupo 5 utilizando formulação proposta com α fixo

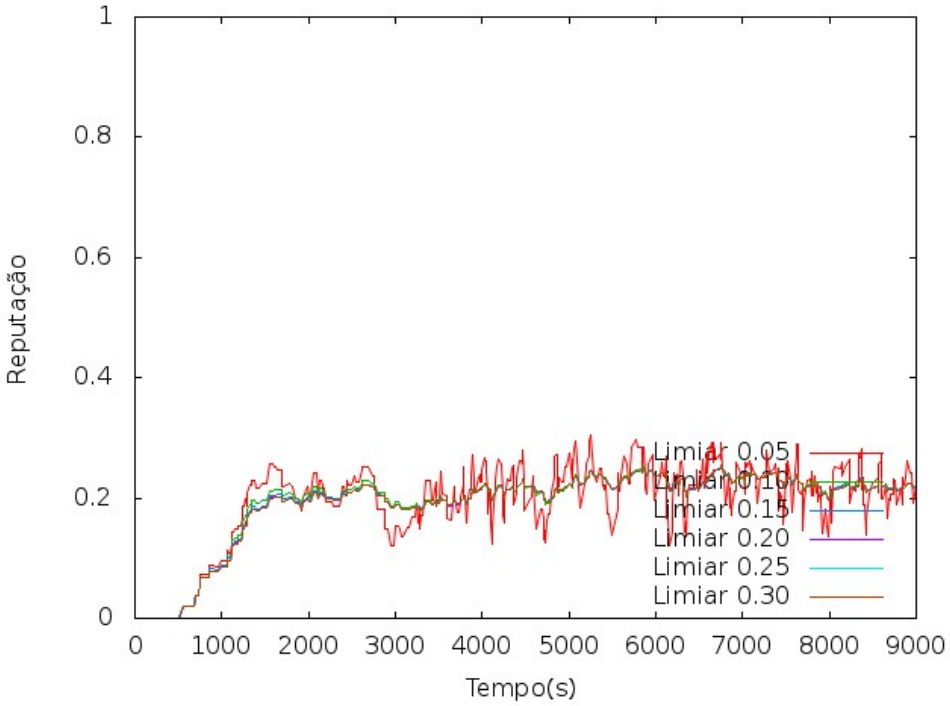

Figura A.50: Exemplo das curvas de reputação para o grupo 5 utilizando formulação proposta com α linear

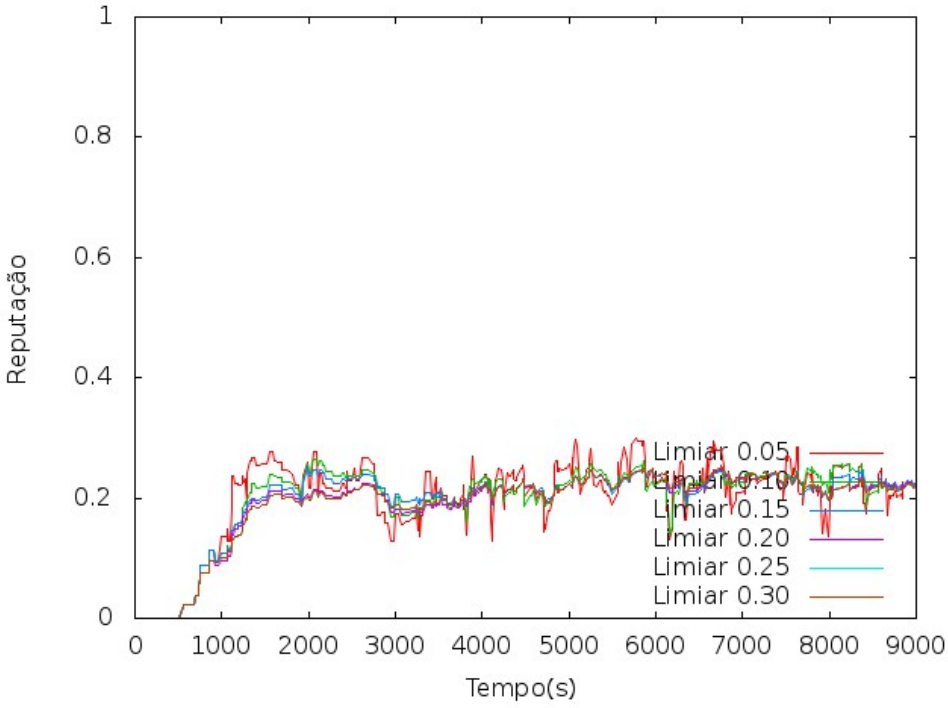

Figura A.51: Exemplo das curvas de reputação para o grupo 5 utilizando formulação proposta com α escalar

GLOSSÁRIO

Ad-Hoc – *Tipo de rede não possuí nós ou terminais especiais*

Anonimidade – *Qualidade do que é anônimo*

DoS – *Denial of Service*

Grau de Confiança – *Valor da reputação de um indivíduo*

IoT – *Internet of Things*

P2P – *Peer-to-Peer*

QoS – *Quality of Service*

TBRM – *Distributed Trust-based Reputation Model*