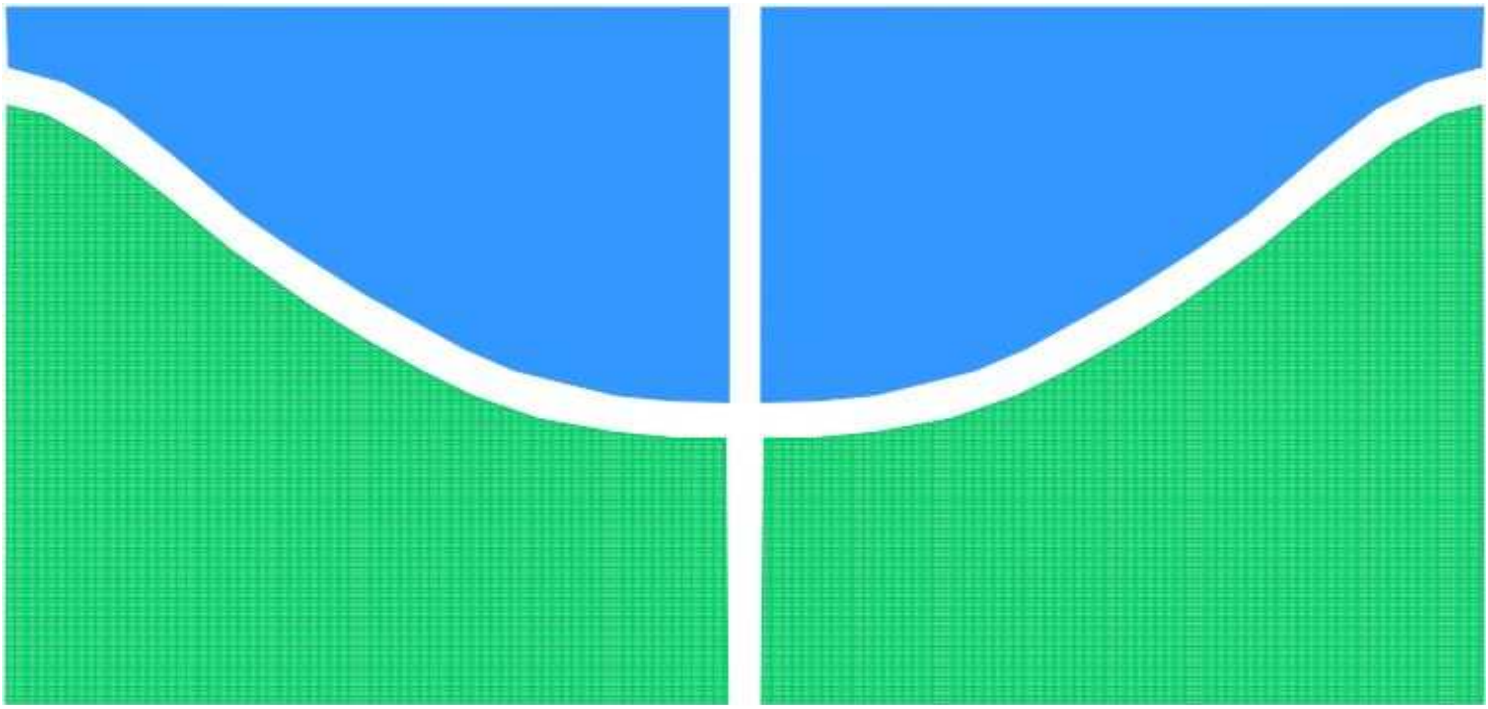

TRABALHO DE GRADUAÇÃO

ANÁLISE DO MODELO DE CONFIANÇA TRAVOS APLICADO A UM *GRID* COMPUTACIONAL COMPOSTO POR GRUPOS DE AGENTES DE SOFTWARE ESPECIALIZADOS UTILIZANDO A PLATAFORMA MULTI-AGENTE JADE

Philipe Laperche dos Santos
Yuri Jorge Sampaio Carvalho

Brasília, dezembro de 2009

UNIVERSIDADE DE BRASILIA

FACULDADE DE TECNOLOGIA

Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

ANÁLISE DO MODELO DE CONFIANÇA TRAVOS APLICADO A UM *GRID* COMPUTACIONAL COMPOSTO POR GRUPOS DE AGENTES DE SOFTWARE ESPECIALIZADOS UTILIZANDO A PLATAFORMA MULTI-AGENTE JADE

Philipe Laperche dos Santos
Yuri Jorge Sampaio Carvalho

Orientador: Prof. Robson de Oliveira Albuquerque
Coorientador: Prof. Rafael Timóteo de Sousa Júnior

BRASÍLIA / DF: 12/2009

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ANÁLISE DO MODELO DE CONFIANÇA TRAVOS APLICADO A UM *GRID* COMPUTACIONAL COMPOSTO PO GRUPOS DE AGENTES DE SOFTWARE ESPECIALIZADOS UTILIZANDO A PLATAFORMA MULTI-AGENTE JADE

PHILIPE LAPERCHE DOS SANTOS
YURI JORGE SAMPAIO CARVALHO

PROJETO FINAL DE GRADUAÇÃO SUBMETIDO AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO.

APROVADA POR:

ROBSON DE OLIVEIRA ALBUQUERQUE, Dr, UnB
(ORIENTADOR)

RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr, UnB
(COORIENTADOR)

FICHA CATALOGRÁFICA

SANTOS , PHILIPE LAPERCHE DOS e CARVALHO, YURI JORGE SAMPAIO

Análise do Modelo de Confiança TRAVOS Aplicado a um *Grid* Computacional Composto por Grupos de Agentes de Softaware Especializados Utilizando a Plataforma Multi-Agente JADE [Distrito Federal] 2009. xiv, 81p., 297 mm (ENE/FT/UnB, Engenheiro de Redes de Comunicação, Engenharia Elétrica, 2009).

Projeto Final de Graduação – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Confiança 2. *Grid* computacional
3. Agentes de *software* 4. TRAVOS

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

SANTOS, PHILIPE LAPERCHE DOS e CARVALHO, YURI JORGE SAMPAIO. (2009). Análise do Modelo de Confiança TRAVOS Aplicado a um *Grid* Computacional Utilizando a Plataforma Multi-Agente JADE. Projeto Final de Graduação. Publicação UnB.LabRedes.PFG/2008, Departamento de Engenharia Elétrica, Universidade de Brasília , Brasília , DF, 95p.

CESSÃO DE DIREITOS

NOME DOS AUTORES: Philipe Laperche dos Santos e Yuri Jorge Sampaio Carvalho
TÍTULO DA DISSERTAÇÃO: Análise do Modelo de Confiança TRAVOS Aplicado a um *Grid* Computacional Utilizando a Plataforma Multi-Agente JADE.
GRAU/ANO: Engenheiro/2008.

Philipe Laperche dos Santos

Yuri Jorge Sampaio Carvalho

Dedicatória(s)

Dedico este trabalho à minha família, da qual sempre tive muito orgulho, aos meus amigos que fizeram a experiência de morar longe de casa uma experiência muito gratificante, e em especial ao meu avô Eugene, que infelizmente se foi e não teve o prazer de me ver graduado.

Philipe Laperche

Dedico este trabalho à minha família pelo apoio incondicional, aos meus amigos que tornaram meus dias mais amenos e ao meu Pai que infelizmente não está presente para compartilhar dessa alegria, mas tenho certeza que interveio de alguma para que tudo acontecesse da melhor forma possível.

Yuri Jorge

Agradecimentos

Primeiramente agradeço a Deus por ter me dado esta oportunidade e força para seguir este caminho. Aos meus pais e irmã por serem o alicerce da minha vida. Aos meus tios, primos e avós, os quais sempre me deram apoio incondicional. Aos professores Rafael Timóteo de Sousa e Robson de Oliveira Albuquerque, pelos seus incentivos, dedicação e paciência diante das dificuldades encontradas. Aos colegas e amigos Yuri Jorge, pelo companheirismo e esforço para concluir esta empreitada, e Ronaldo Sebastião, por dividir seu conhecimento e possibilitar a implementação desse projeto. Aos meus grandes amigos da UnB e de Goiânia, entre alunos e professores, presentes ou distantes, aos quais devo muito pelo companheirismo, cumplicidade e por fazer toda essa caminhada tão prazerosa, em especial a todos aqueles que passaram pela república Kilombola (Loucura, Leitão, Gonssa e Thomaz).

Obrigado a todos vocês.

Philipe Laperche

Agradeço primeiramente a Deus que sempre me motiva diante dos percalços da vida e me mostra sempre o caminho correto a seguir. Agradeço ao professor Robson Albuquerque pela sabedoria e disponibilidade para nos ajudar durante a nossa caminhada em busca dos melhores resultados. Por fim agradeço ao professor Rafael Timóteo que motivou e inspirou esse trabalho.

Yuri Jorge

RESUMO

Este trabalho objetiva analisar a confiança entre nodos de um *Grid* computacional no momento da delegação de tarefas, por meio do desenvolvimento de um ambiente multi-agente JADE. Será apresentado um ambiente baseado em agentes de *software* que possui as funcionalidades básicas de um *Grid* computacional e implementa o modelo de confiança TRAVOS. Utilizaremos esse ambiente para realizar interações entre os agentes, analisando a partir de que momento a experiência pessoal dispensa o uso da reputação, e a influência de opiniões desonestas no cálculo da confiança, além de se calculcar a confiança entre um grupo e outro, e a confiança de um agente em seu próprio grupo, sendo que para se decidir a quantidade de agentes desonestos e outros fatores,nos baseamos na teoria dos generais bizantinos.

ABSTRACT

This work aims to analyze trust issues between computational Grid nodes in the moment of task delegation, throught the development of a JADE multi-agents system. It will be presented a software agent-based environment which implements the basic computational Grid features and the TRAVOS trust model. We'll use this environment to make agents interactions, analyzing when personal experience dismisses the reputation information, and how dishonest opinions influence trust calculation, in addition we calculate the trust between groups, the trust an agent has in its own group, being that to decide the amount of dishonest agents and other factors, we rely on the theory of Byzantine generals.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ACRÔNIMOS

ARPA	Advanced Research Projects Agency
ACL	Agent Communication Language
AID	Agent Identifier
AMS	Agent Management System
CTA	Combined Trust Agent
DF	Directory Facilitator
DTA	Direct Trust Agent
FIPA	Foundation for Intelligent Physical Agents
JADE	Java Agent Development Framework
KQML	Knowledge Query and Manipulation Language
LGPL	Library Gnu Public License
OGF	Open Grid Forum
ODB	Outcome Data Base
RTA	Reputation Trust Agent
SDB	Sociograms Data Base
TRAVOS	Trust and Reputation Model for Agent-based Virtual Organizations
VO	Virtual Organizations

1. INTRODUÇÃO

O aumento na diversidade e complexidade dos sistemas computacionais para aplicações cada vez mais distintas do cotidiano traz, atualmente, a necessidade de um desenvolvimento cada vez maior de sistemas que sejam capazes de lidar com essa realidade.

A tecnologia baseada em sistemas multi-agentes tem tido grande êxito na resolução destes desafios [1]. Isso porque, este pode ser aplicado em diversos contextos e tem a grande vantagem de ser um sistema aberto, flexível e distribuído. Por isso tem sido cada vez mais discutida e desenvolvida no intuito de se chegar a um modelo que seja capaz de suprir todas as necessidades com eficácia e segurança.

A diversidade de aplicações em sistemas multi-agente tem gerado a criação de grupos com características distintas, e que precisam trocar informações entre sí para a realização das tarefas que envolvem o sistema como um todo.

Por ser um sistema aberto, existe a possibilidade de agentes ou até mesmo de grupos inteiros que hajam de maneira maliciosa de acordo com suas necessidades e intenções. Um exemplo é o ataque Sybil [2], aonde não existindo um ponto central para controlar a associação de uma identidade a uma entidade, é sempre possível para uma entidade desconhecida apresentar múltiplas identidades.

Sendo assim, é extremamente importante haver o desenvolvimento de um modelo de confiança que seja capaz de avaliar a honestidade dos agentes e grupos de agentes que venham a interagir dentro do contexto como um todo.

A criação de um modelo de confiança realmente eficaz é um assunto que vem sendo discutido e analisado por diversos pesquisadores. Com isso, há uma grande variedade de modelos já desenvolvidos. Dentre eles, existe um modelo que avalia de muitas maneiras a confiança que se pode ter em um agente denominado TRAVOS [3] (Trust and Reputation Model for Agent-based Virtual Organizations).

1.1. OBJETIVOS

Neste trabalho, será desenvolvida uma implementação utilizando o modelo de confiança TRAVOS [3], no qual os agentes sejam capazes de gerenciar e avaliar a confiança

dentro de um grupo especializado no provimento de um determinado tipo de serviço. Além disso, sejam capazes de trocar informações com outros grupos e assim avaliar a confiança na execução de tarefas externas, necessárias para o funcionamento de um sistema de alta complexidade.

Será estudado como se pode estabelecer um cálculo de confiança intra-grupo baseada na troca de informações entre os membros e controlada por um líder que possa também realizar a troca de informações com outros grupos.

1.2. ESTRUTURA DO TRABALHO

O trabalho em questão divide-se em 5 capítulos. No capítulo 1 é feita uma introdução geral sobre os temas que serão abordados e objetivos do trabalho. No capítulo 2, é discutido um conceito geral sobre confiança computacional em ambientes distribuídos , bem como uma explicação mais detalhada sobre o modelo de confiança TRAVOS, que é o modelo utilizado na implementação de nosso trabalho. No capítulo 3, é dissertado o conceito de agentes de software, bem como da utilização de grupos de agentes (Organizações Virtuais) como solução para complexos ambientes, conceitos bases de Grids computacionais, uma explicação do funcionamento da plataforma JADE e uma introdução à Teoria de Grupos, de modo que assim fazemos uma introdução à todos elementos importantes para aplicação do trabalho. O capítulo 4 é utilizado para a demonstração de resultados de uma implementação empírica do modelo, provando assim a eficácia dos conceitos apresentados anteriormente. Por fim no capítulo 5 é feita uma conclusão do trabalho.

2. CONFIANÇA EM AMBIENTES DISTRIBUÍDOS

Neste capítulo, trataremos de aspectos relacionados à confiança em ambiente distribuídos. Primeiramente, serão abordados aspectos gerais de confiança tratando conceitos de confiança direta e indireta. Em seguida, mostraremos a aplicação da confiança computacional, por meio de um resumo de 3 modelos, o Marsh [4], o REGRET [5] e o TRAVOS [3], sendo esse último apresentado de forma mais detalhada por ter sido utilizado na implementação.

2.1. CONCEITOS GERAIS DE CONFIANÇA

Pela definição da enciclopédia Larousse Cultural [6]:

> *"Confiança s.f.* ***1****. Esperança firme em alguém, em alguma coisa: ter confiança no futuro. –* ***2****. Sentimento de segurança quanto à probidade da conduta de alguém: perder a confiança do chefe. –* ***3****. Segurança íntima: ter confiança em si. –* ***4****. Crédito: homem de confiança. –* ***5****. Pop. Atrevimento. –* ***6****. Amizade, intimidade (...)"*

Os conceitos apresentados acima, utilizados na sociedade, servem de base para chegarmos a uma visão mais específica de confiança, voltada para o ambiente computacional. Sob essa perspectiva, é possível diferenciar dois tipos de confiança. A confiança direta, ou simplesmente confiança, refere-se às experiências vividas, baseando apenas na avaliação pessoal. Já a indireta, conhecida como reputação, leva em conta a opinião de terceiros.

2.1.1. Confiança Computacional

O conceito de confiança computacional tem como base a confiança entre as pessoas. Segundo Marsh [4], confiar em uma pessoa significa crer que ela, tendo a oportunidade, não se comportará de forma a prejudicar a pessoa que está confiando nela. Diferentes pessoas podem ter um grau de confiança diferente em uma mesma pessoa dependendo do passado de interações com ela e da percepção de seu comportamento [4].

Para Gambetta [7], confiança é um nível particular da probabilidade subjetiva com a qual um agente avalia se outro (ou um grupo de agentes) irá realizar uma ação particular,

antes de monitorar essa ação e em um contexto que afete a sua própria ação. Essa confiança pode ser mensurada e assume valores entre 0 e 1.

A confiança é interna aos agentes e é calculada a partir do armazenamento do resultado de interações com outros agentes. Essas interações podem ser bem-sucedidas ou não, sendo bem-sucedidas quando o provedor entregou o serviço conforme o contratado, como definiu Lamsal [8]. Gambetta [7] considera que a confiança sempre possui um risco associado, e que uma forma de minimizá-lo é pré-estabelecer contratos e limites tanto para quem confia, como para em quem se confia. A decisão de se confiar ou não é muito importante principalmente porque segundo Albuquerque [9], a perda causada pela não concretização da confiança depositada em uma pessoa costuma ser maior que a recompensa caso contrário.

Ao se confiar em uma entidade, é importante saber a que aspecto estamos nos referindo, pois uma entidade pode ser mais confiável em certas situações do que em outras. Deve existir a possibilidade de especificar em qual aspecto e nível de confiança as entidades estão interessadas. Por isso, a confiança em uma pessoa deve ser lidada em diferentes contextos, que devem ser usados para decidir se um participante é elegível para certa atividade. A confiança é um valor social que incentiva as entidades a colaborarem entre si. Quanto maior for o nível de confiança acumulado em um participante, maior será sua posição social no sistema [10].

2.1.2. Reputação

Além da confiança (também chamada de confiança direta [4]) obtida por meio de experiências vividas, as pessoas podem decidir confiar ou não com base na reputação (ou confiança indireta [4]), que, segundo Sabater e Sierra [11], é definida como um conjunto de opiniões comuns sobre uma entidade, mais especificamente, sobre o comportamento de uma entidade, tendo como base o seu comportamento passado. O uso de várias fontes de opinião, se combinadas de forma inteligente, pode aumentar a confiabilidade da confiança calculada, porém será necessário adotar um modelo mais complexo [5].

No cálculo da reputação, a informação recebida pode assumir valores contínuos ou booleanos. Isso tem uma grande influência no projeto do modelo a ser adotado. Geralmente, quando o modelo utiliza métodos probabilísticos, ele trata informação booleana, ou seja, não existem valores intermediários ao se avaliar o sucesso de uma interação e se uma entidade

confia ou não em outra. Enquanto que os baseados em mecanismos de agregação usam valores contínuos, que permitem que uma entidade possa confiar parcialmente em outra [5]. Suryanarayana [12] ainda menciona a existência de modelos que utilizam valores discretos de confiança.

Quando os dados para cálculo da confiança não são suficientes ou mesmo inexistentes (no caso de não ter havido uma interação prévia), faz-se necessário o uso da reputação [8]. Neste caso, é importante saber avaliar a probabilidade da opinião fornecida por terceiros ser exata, tendo como parâmetro a concretização de opiniões fornecidas no passado.

Ao se encontrar um valor para a confiança em alguém, precisamos saber quão confiável é esse valor. Dessa forma, o cálculo deve considerar não somente o sucesso ou não de interações passadas e a opinião dos outros, como também [5] a quantidade de interações que geraram o valor da confiança direta, o quanto se confia em quem forneceu as opiniões utilizadas, há quanto tempo a informação usada foi colhida, etc. Essas informações seriam utilizadas para ajustar o valor encontrado.

A confiança [5] pode ser considerada como uma propriedade global ou subjetiva. No primeiro caso, o principal problema é a falta de personalização. Algo considerado ruim para uma pessoa, pode ser aceitável para outra. Essa abordagem é aceitável em cenários simples, porém não é útil quando lidamos com situações mais complexas e subjetivas. Já na abordagem subjetiva, cada agente usa a confiança direta e a indireta obtida para que ele próprio calcule a sua confiança em cada membro da comunidade. Esses modelos são mais indicados para ambientes de pequeno e médio porte, nos quais os agentes interagem freqüentemente.

2.1.3. Confiança e Reputação em uma sociedade

Assim como humanos existem em sociedades, os agentes também têm sua própria sociedade virtual. Dessa forma, podemos expandir nosso estudo de confiança para ambientes multi-agente *Grid*.

Patel [3] lista uma série de características que esses ambientes possuem que demandam o estudo da confiança. Primeiro, são ambientes abertos, influenciados por eventos externos aos limites do sistema; assim, agentes (inclusive os maliciosos) são livres para entrar ou sair a qualquer momento, sendo difícil monitorar suas interações. Segundo, a natureza aberta e

dinâmica desses sistemas implica que um agente pode ter que interagir com outro com o qual ele não possui nenhuma experiência passada. Em terceiro, por serem sistemas distribuídos e heterogêneos, as aplicações para *Grid* devem ser descentralizadas. Por último, a existência de organizações virtuais (VO) influencia o comportamento de um agente, sendo importante considerar não só as experiências pessoais de cada agente, mas também examinar as relações sociais dele com outros agentes.

Nesse contexto, Patel [3] considera os seguintes aspectos a serem analisados para o desenvolvimento de um modelo para confiança:

(a) Armazenar o passado de interação com cada indivíduo.

(b) Caso ainda não tenha ocorrido nenhuma interação com determinado indivíduo, recorrer à opinião de outros.

(c) Ao se incluir opinião de terceiros, adicionar um mecanismo de evitar erros e ajustar a maneira como cada um dá sua opinião.

(d) Analisar a estrutura social para minimizar erros introduzidos por opiniões de outros.

(e) Avaliar a confiabilidade de uma opinião fornecida.

(f) Devido à natureza distribuída do *Grid*, implementar um modelo de confiança escalável, capaz de lidar com vários nodos que entram e saem a todo instante.

Assim, na próxima seção, será dada uma abordagem geral sobre modelos de confiança computacional, mencionando os tipos existentes de acordo com sua arquitetura e citando alguns dos modelos presentes na literatura.

2.2. MODELOS DE CONFIANÇA COMPUTACIONAL

Segundo Patel [3], um modelo de confiança computacional, de acordo com sua arquitetura, pode ser centralizado ou distribuído, refletindo a natureza de armazenamento da confiança calculada. Na abordagem centralizada, a informação de confiança é armazenada em um repositório central; enquanto que, na abordagem distribuída, essa informação é distribuída entre os agentes do sistema.

No modelo centralizado, um repositório central (também chamado de centro de reputação) é responsável por reunir, calcular e armazenar a informação sobre confiança,

tornando-a disponível para todas as entidades, que o atualizam quanto ao sucesso ou não de cada interação que realizam. Josang [13], descreve dois componentes fundamentais presentes nesse tipo de sistema:

(a) Protocolos de comunicação centralizada – Permitem aos participantes do sistema a comunicação com o repositório central para conceder informações sobre suas interações com outros participantes e para obter valores de reputação sobre potenciais nodos a interagir.

(b) *Reputation computation engine* – Utilizado pelo repositório central para calcular o valor de reputação de um nodo baseado nas opiniões recebidas de outros nodos do sistema sobre esse nodo em particular.

A principal vantagem desse tipo de arquitetura é que os protocolos utilizados são simples e os participantes do sistema não precisam ficar procurando por informação de confiança pelos membros da comunidade, já que essa informação está centralizada. Entretanto, existem algumas desvantagens inerentes a essa arquitetura, como o fato de não considerar os aspectos sociais entre quem confia e a quem é depositada a confiança, e a limitação da escalabilidade, um aumento do número de nodos e de interações pode criar um gargalo nas requisições ao repositório central [3].

Como exemplos de modelos de confiança centralizados, temos: SPORAS [14]; HISTOS [14]; e o Sistema de Reputação Beta [15]. Sendo que os dois primeiros foram propostos por Moukas, Zacharia e Maes [14]; e o último, proposto por Josang e Ismail [15].

Diferente do centralizado, o modelo distribuído não possui um repositório central. Nesse modelo, a informação de confiança é armazenada em repositórios distribuídos ou até mesmo em cada indivíduo, que grava apenas seu próprio histórico de interações. A desvantagem dessa abordagem é que cada vez que se deseja interagir com algum agente, a informação necessária estará espalhada na rede [3]. Josang [13], descreve dois componentes fundamentais presentes nesse tipo de sistema:

(a) Protocolos de comunicação distribuída – Permitem aos participantes do sistema buscar informação de confiança em outros membros da comunidade. Para Patel [3] esses protocolos diferem dos requeridos pela abordagem centralizada pelo fato deles proporcionarem ao nodo a capacidade de localizar o nodo correto para se comunicar e pedir informação. No modelo centralizado, não há necessidade de procurar o repositório central, visto que sua localização é fixa e conhecida por todos os nodos do sistema.

(b) Método computacional de reputação – Utilizado por cada agente para a realização do cálculo de reputação a partir dos valores de confiança obtidos dos outros participantes do sistema.

A abordagem distribuída encaixa-se bem em grandes sistemas abertos. Nesses sistemas, pode ser muito caro para um indivíduo reunir todas as informações de confiança sobre todos os participantes da comunidade. Assim, para esses casos, a abordagem distribuída permite um indivíduo obter um subconjunto dessa informação de reputação a partir dos agentes vizinhos, pela busca de opiniões somente a um grupo de indivíduos da comunidade [3].

Como exemplos de modelos de confiança descentralizados ou distribuídos, temos: o modelo de confiança de Marsh [4]; o modelo de confiança cognitivo, construído por Castelfranchi e Falcone [16]; o REGRET, proposto por Sabater e Sierra [5]; o CREDIT, apresentado por Ramchurn [17]; e o TRAVOS, proposto por Patel [3].

Visto que a abordagem do nosso trabalho envolve a aplicação de um modelo de confiança em ambientes distribuídos, será apresentada uma visão geral de alguns dos modelos distribuídos citados. Pelos estudos realizados, decidiu-se optar pela aplicação do modelo do Patel [3], o TRAVOS, devido a ser um modelo dos mais atuais e atender a diversos requisitos importantes satisfeitos, individualmente ou em parte, pelos modelos de confiança já propostos anteriormente. Entre esses requisitos, podemos citar a escalabilidade, a descentralização do modelo, o cálculo de confiança e reputação, o ajuste das opiniões não confiáveis, a manutenção de um histórico de interações, entre outros. Assim sendo, por causa da adoção do TRAVOS, esse modelo será abordado de maneira mais detalhada que os demais.

2.2.1. Modelo de Confiança de Marsh

Um dos primeiros modelos de confiança considerados por uma perspectiva computacional foi proposto por Stephen Paul Marsh em 1994 [4]. O modelo leva em conta algumas variáveis para representação: um grupo de agentes A; uma variedade de situações ($\alpha, \beta, \gamma, ...$); e um conjunto de predicados booleanos $K_x(y)^t$, que indica se um agente *x* conhece agente *y* no tempo *t*.

Marsh [4] categoriza confiança em 3 tipos:

(a) Confiança Básica – É um valor que indica a disposição de um agente em confiar. Esse valor é calculado baseado no histórico de interações do agente e é representado por T_x^t, para um agente $x \in A$ no tempo *t*, e seu valor está contido no intervalo [-1,1). Boas experiências tendem a uma maior disposição para confiar; em contrapartida, experiências ruins levam a uma menor disposição [5].

(b) Confiança Geral – Representa a confiança em agentes. É o valor que indica o quanto um agente *x* confia em um agente *y*, não levando em conta uma situação específica. Esse valor é denotado por $T_x(y)^t$, que representa a confiança geral que o agente *x* tem no agente *y* no tempo *t*. A Confiança Geral possui valores no intervalo [-1, 1), sendo 0 um valor neutro; ou seja, se o valor de confiança for 0, pode significar que *x* nem conhece *y* ou que *x* confiava em *y*, mas o valor de confiança foi reduzido devido a suas ações. Já, se o valor de confiança for -1, pode representar desconfiança completa.

(c) Confiança Situacional – Consiste em um conceito de confiança em agentes em determinadas situações. É o caso no qual situações diferentes requerem considerações distintas sobre confiança, as quais se apresentarão de forma diferente na maioria dos casos, mesmo em relação a um agente específico. Assim, dados os agentes $x, y \in A$, e a situação α, a confiança situacional, denotada por $T_x(y,\alpha)^t$, representa o valor de confiança que *x* tem em *y* na situação α no tempo *t*. Esse valor de confiança também se encontra no intervalo [-1, 1) [3].

O modelo do Marsh [4] representa os conceitos de importância e utilidade pelas variáveis $U_x(\alpha)^t$ e $I_x(\alpha)^t$, respectivamente. $U_x(\alpha)^t$ corresponde à utilidade que *x* obtém em uma situação α no tempo *t* e varia entre -1 e 1; enquanto que $I_x(\alpha)^t$ corresponde à importância de uma situação α para o agente *x* no tempo *t* e varia em 0 e 1. Pode parecer que a importância e a utilidade são a mesma coisa. Entretanto, esse não é o caso, apesar da diferença não ser óbvia. Em particular, a utilidade é geralmente mensurável ou, pelo menos, simples de ser estimada. Por outro lado, a importância é um julgamento subjetivo ou centrado no agente.

Considerando os conceitos de importância e utilidade, é possível descrever a maneira pela qual a confiança situacional é calculada. Sua estimativa é realizada pela multiplicação do

ganho em utilidade na situação α pela importância da situação α e a confiança geral de *x* em *y*.

Nesse modelo, a confiança pode ser utilizada pelo agente *x* para decidir cooperar ou não com o outro agente. Para utilizar confiança nesse contexto, algumas suposições são feitas:

(a) Um agente *x* tem a livre escolha de cooperar ou não;

(b) Existe outro agente *y* para se cooperar com, nessa situação;

(c) O agente *x* conhece o agente *y*;

(d) O agente *x* não deve nada a *y*, o que descarta a consideração de que *x* ajuda *y* por conta de alguma dívida do passado;

(e) O agente *x* conhece a situação. Em outras palavras, *x* percebe as similaridades entre essa situação e outras que ele já experimentou.

Se todas as condições forem satisfeitas, então a cooperação pode acontecer na situação em que a confiança de *x* em *y* excede um valor limite determinado. Um ponto importante a ser destacado é que, além da confiança, o mecanismo de decisão leva em conta a importância da ação, o risco associado e a competência do agente [5]. Para calcular o risco e a competência, os diferentes tipos de confiança descritos anteriormente são utilizados.

Analisando a forma como é calculada a confiança computacional, pode-se deduzir que a abordagem de Marsh possui algumas limitações. Se dois componentes do cálculo forem negativos em um dado momento (como pode acontecer com os valores de importância e de confiança geral entre agentes), resultará em um valor positivo de confiança, fazendo com que a aplicação do valor de confiança seja ambíguo. Além disso, os componentes podem ser frações, resultando em produtos pequenos, tornando o resultado final difícil de ser comparado [3].

A estimativa da confiança situacional somente é completa quando todos os valores de confiança possíveis de *x* em *y* são considerados. Os valores de confiança anteriores são considerados utilizando três abordagens: otimista, pessimista e pragmática. A otimista considera sempre o maior valor de confiança observado nas experiências anteriores. A pessimista escolhe o valor menor, sempre esperando o pior retorno possível de determinada situação. Já a abordagem pragmática é mais realista, consiste em um meio termo entre a pessimista e a otimista.

Devido a restrições de memória, o modelo do Marsh permite a imposição de um limite no número de experiências que são levadas em consideração no cálculo da confiança. Assim, experiências muito antigas não possuem grande influência no valor de confiança obtido. Essa característica é desejável, pois, se o agente for capaz de mudar seu comportamento dinamicamente, então um agrupamento de experiências recentes irá representar melhor o comportamento do agente [3].

Outra característica importante, destacada por Sabater e Sierra [5], é que o modelo também considera é a reciprocidade como modificador dos valores de confiança. A idéia por traz desse conceito é que, se um agente *x* ajudou um agente *y* no passado e *y* não respondeu como deveria, a confiança de *x* em *y* diminui. Assim, quando há cooperação entre agentes, a confiança entre eles aumenta.

O modelo do Marsh leva em conta somente as experiências diretas do agente, calculando a confiança direta. Não existe uma troca de informações sobre os outros agentes do sistema, não permitindo a estimativa de confiança quando o próprio agente ainda não interagiu com o outro. Dessa forma, a reputação não é levada em conta no cálculo de confiança.

2.2.2. REGRET

Outro modelo de confiança distribuído citado é o REGRET [5], proposto por Sabater e Sierra. Segundo Sabater e Sierra [11], consiste em um sistema modular de confiança e reputação direcionado para ambientes complexos, pequenos ou médios, de comércio eletrônico, no qual as relações sociais entre os indivíduos são muito importantes. O sistema considera três tipos diferentes de fontes de informação: experiências diretas, informação de terceiros e estruturas sociais. Assim sendo, apresenta-se como um modelo de reputação que leva em conta a dimensão individual e social dos agentes, além da estrutura hierárquica ontológica que permite considerar diversos tipos de reputação ao mesmo tempo [5].

Essas dimensões que definem a reputação são caracterizadas da seguinte forma:

(a) Dimensão individual – A dimensão individual modela a interação direta entre dois agentes.

(b) Dimensão social – Essa dimensão concede a possibilidade de refletir no modelo a característica de sociedades complexas: a relação de grupo. Quando falta a informação direta

obtida por interações pessoais com a entidade, a reputação do grupo mostra-se como uma expectativa inicial sobre o comportamento de um agente, pois, pertencendo a um grupo determinado, implica, em uma primeira análise, que seus membros compartilham um modo de pensar comum. Então, além de considerar as interações diretas, é necessário considerar também a interação com os membros do grupo, a informação que o nosso grupo tem em relação àquele agente e a informação que o nosso grupo tem em relação àquele grupo.

(c) Dimensão ontológica – Essa dimensão acrescenta a possibilidade de combinar valores de reputação e confiança ligados a aspectos simples de modo a calcular valores associados a atributos mais complexos. Para calcular uma dada reputação, considerando a dimensão ontológica, um agente tem que calcular a reputação de cada um para cada aspecto relacionado.

O diálogo entre agentes no REGRET é representado por uma saída, que consiste no contrato inicial o qual estabelece os termos e condições de transação e no resultado atual da ação realizada. Essas saídas possuem dois tipos de variáveis: as variáveis comuns, as quais refletem os aspectos do acordo do contrato; e as variáveis esperadas, que refletem os aspectos que implicitamente devem acontecer por uma das partes. Essas variáveis esperadas estão relacionadas à subjetividade dos agentes; assim, para um dado diálogo entre dois agentes, haverá duas saídas diferentes.

A avaliação subjetiva realizada por um agente em um determinado aspecto de uma saída é denominada impressão, que é representada pela tupla *(a, b, o,* φ *, t, W)*, onde $a,b \in A$ são os agentes envolvidos na transação, $o \in O$ é a saída, φ é a variável da saída que está sendo julgada, *t* é o horário no qual a impressão foi feita, e $W \in [-1,1]$ é a nota associada ao aspecto específico que está sendo avaliado pelo ponto de vista do agente *a*.

A reputação subjetiva de um agente com respeito a um atributo de saída é obtida por meio de um subconjunto, que coincide com um dado padrão, de um grupo de impressões armazenadas no banco de dados de impressões. O valor atual é calculado pela média ponderada dos fatores de nota de impressões no subconjunto encontrado como resultado da busca. É dada maior relevância às impressões mais recentes no cálculo do valor de reputação. Além disso, a confiabilidade do valor de reputação calculado é obtida considerando a combinação do número de impressões utilizadas nos cálculos e a variância das notas de impressões utilizadas. Entretanto, essa abordagem é falha [3], pois não trata a possível

presença de informações incorretas pela suposição de que existe uma sociedade solidária. Acrescenta-se a isso, o fato do modelo ser altamente susceptível a ruído como resultado da maneira com que as impressões são ponderadas e somadas.

O sistema mantém três bases de conhecimento. A base de dados de saídas (ODB – Outcome Data Base) guarda os contratos prévios e seus resultados; a base de dados de informação (IDB – Information Data Base) é utilizada como um container para as informações recebidas de outros nodos; e a base de dados sociográficos (SDB – Sociograms Data Base) para guardar gráficos que definem a visão social de mundo do agente. Essas bases de dado alimentam os diferentes módulos do sistema.

Segundo Sabater e Sierra [5], uma vantagem dessa abordagem modular é a adaptabilidade que o sistema tem para diferentes graus de conhecimento. O sistema pode operar mesmo se um agente é recém-chegado no sistema e tem uma importante falta de informação. Com o aumento do seu conhecimento sobre os outros membros da comunidade e sobre as relações sociais entre eles, o sistema começa a utilizar outros módulos para melhorar a exatidão dos valores de confiança e reputação. Isso permite a utilização do sistema em vários cenários, do mais simples para o mais complexo. Se a informação estiver disponível, o sistema irá utilizá-la.

2.2.3. TRAVOS

O modelo de confiança explorado na implementação é o TRAVOS (*Trust and Reputation Model for Agent-based Virtual Organisations*) [3]. Consiste em um modelo de confiança e reputação para organizações virtuais baseadas em agentes, onde a confiança é tratada utilizando probabilidade. Nesse modelo, a obtenção do valor de confiança é baseada nas interações passadas e na reputação obtida por outros nodos da rede.

Segundo Patel [3], TRAVOS é um modelo de confiança e reputação que pode ser utilizado para dar suporte a informações de decisão para assegurar boas interações em um ambiente distribuído. A utilização de confiança no processo de decisão pode assegurar boas interações ao habilitar o agente a apresentar motivos pelos quais ele deve interagir ou não com um potencial parceiro.

No TRAVOS, existem três métodos de definir a confiança em outro agente: baseada nas interações diretas com o agente; nas opiniões de outros agentes presentes no sistema; ou em uma combinação desses dois métodos, juntando confiança direta e reputação.

Antes de definir uma forma de calcular a confiança, é importante saber representar e modelar o comportamento de um agente. O TRAVOS considera dois comportamentos, o confiável e o não-confiável, baseado na probabilidade do agente interagir com sucesso ou não. É considerada uma interação bem sucedida aquela que atende todas as expectativas das partes envolvidas, sendo elas definidas explicitamente em contrato ou não.

2.2.3.1. Notação Utilizada

Patel [3] modela o ambiente no qual o TRAVOS é aplicado como um sistema multi-agente formado por *n* agentes, e denota o conjunto de todos os agentes como $\mathrm{A} = \{a_1, a_2, ..., a_n\}$. Vários pares de agentes $\{a_x, a_y\} \subseteq \mathrm{A}$ podem interagir uns com os outros, governados por contratos que especificam as obrigações de cada agente em relação a seu parceiro de interação. Uma interação entre a_1 e a_2 é considerada bem sucedida por a_1 se a_2 cumpre suas obrigações. Pela perspectiva de a_1, a avaliação de uma interação entre a_1 e a_2 é resumida em uma variável binária, O_{a_1,a_2}, onde $O_{a_1,a_2} = 1$ indica uma interação bem sucedida para a_1 com a_2 e $O_{a_1,a_2} = 0$, uma interação mal sucedida. Mais ainda, uma avaliação de uma interação observada no tempo *t* é denotada como $O^t_{a_1,a_2}$, e o conjunto de todas as avaliações observadas de um tempo t_0 a *t*, como $O^{t_0:t}_{a_1,a_2}$.

Em qualquer ponto no tempo *t*, o histórico de interações entre agentes a_1 e a_2 é guardado como valores ordenados, $\Re^t_{a_1,a_2} = \left(m^t_{a_1,a_2}, n^t_{a_1,a_2}\right)$, onde o valor de $m^t_{a_1,a_2}$ é o número de interações bem sucedidas de a_1 com a_2 no tempo *t*, enquanto $n^t_{a_1,a_2}$ é o número de interações mal sucedidas de a_1 com a_2 no tempo *t*.

A tendência de um agente a_2 cumprir ou não com as obrigações para com o agente a_1 é governada por seu comportamento. O comportamento de a_2 em relação à a_1, denotado por B_{a_1,a_2}, é modelado como o valor esperado de O_{a_1,a_2}.

No TRAVOS, cada agente mantém um nível de confiança em cada agente presente no sistema. O nível de confiança de um agente a_1 em um agente a_2, denotado por τ_{a_1,a_2}. Especificamente, o nível de confiança calculado utilizando somente as interações do próprio agente com outro é conhecido como confiança direta, denotado por $\tau^d_{a_1,a_2}$. Por outro lado, o nível de confiança calculado utilizando somente opiniões capturadas por outros é conhecido como reputação, denotado por $\tau^r_{a_1,a_2}$. A confiança calculada da combinação de experiência direta com opiniões de terceiros é conhecida como confiança combinada, denotada por $\tau^c_{a_1,a_2}$.

Outra métrica utilizada nesse modelo de confiança é o grau de certeza. Consiste em uma métrica que representa a acuidade do valor de confiança calculado por um agente, dado o número de observações que são utilizadas para o cálculo dessa confiança. O grau de certeza de a_1 na sua avaliação de a_2 é denotado por γ_{a_1,a_2}.

2.2.3.2. Confiança Direta

No cálculo da confiança direta, é realizada uma abordagem probabilística baseada nas experiências individuais de um agente no papel daquele que confia. Se o agente a_1 tem informações completas sobre o agente a_2, a probabilidade de a_2 cumprir com suas obrigações é expressa por B_{a_1,a_2}, de acordo com a_1. Entretanto, geralmente, não se assume que haja informação completa. Assim, a confiança direta $\tau^d_{a_1,a_2}$ em um tempo *t* é definida como o valor esperado de B_{a_1,a_2}, dado um conjunto de avaliações $O^{1:t}_{a_1,a_2}$ de interações observadas.

$$\tau^d_{a_1,a_2} = \mathrm{E}\left[\mathrm{B}_{a_1,a_2} \mid O^{1:t}_{a_1,a_2}\right] \tag{2.1}$$

O valor esperado de uma variável aleatória contínua depende da função densidade de probabilidade (PDF) utilizada para modelar a probabilidade que a variável terá um determinado valor. Nas análises Bayesianas [18], a família beta de PDFs é comumente

utilizada como uma distribuição prévia para variáveis aleatórias que possuem valores contínuos no intervalo [0,1].

A fórmula geral para distribuições beta é dada pela equação 3.2. Essa fórmula possui dois parâmetros, α e β, os quais definem o formato da função densidade quando plotada.

$$f(b \mid \alpha, \beta) = \frac{b^{\alpha-1}(1-b)^{\beta-1}}{\int U^{\alpha-1}(1-U)^{\beta-1} dU}, \text{ onde } \alpha, \beta > 0 \qquad (2.2)$$

Utilizando-se dessa função, é possível calcular o valor da confiança direta. Primeiro, é preciso encontrar o valor de α e β. Assumindo previamente, antes de interagir, que todos os valores possíveis de B_{a_1,a_2} são iguais, os valores iniciais de α e β são $\alpha = \beta = 1$. Baseando-se em técnicas padronizadas, considerando as observações realizadas nas interações, esses parâmetros podem ser calculados adicionando o número de interações bem sucedidas ao valor inicial de α e o número de interações mal sucedidas a β.

$$\hat{\alpha} = m^{1:t}_{a_1,a_2} + 1 \text{ e } \hat{\beta} = n^{1:t}_{a_1,a_2} + 1 \text{ onde } t \text{ é o tempo de avaliação} \qquad (2.3)$$

Vale ressaltar que a utilização na notação com acento circunflexo ($\hat{\alpha}$ e $\hat{\beta}$) é para indicar que os valores dos parâmetros da distribuição beta são estimados baseados na evidência do que é observado.

Assim, o valor final de $\tau^d_{a_1,a_2}$ é calculado aplicando a equação padrão para o valor esperado de uma distribuição beta.

$$\tau^d_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} \qquad (2.4)$$

Pelas equações 2.2 e 2.4, pode-se deduzir que o valor da confiança direta e a distribuição mudam conforme um agente ganha experiência interagindo com o outro agente, o que promove a modificação do formato da curva de distribuição, visto que o valor dos parâmetros é alterado com o tempo.

Sabendo como calcular a confiança direta, é possível criar um agente de confiança direta (DTA – *Direct Trust Agent*), que avalia a confiabilidade somente pela confiança direta depositada no confiado. Assim, para esse caso, toda a confiança de a_1 em a_2 é igual à confiança direta, $\tau_{a_1,a_2} = \tau^d_{a_1,a_2}$.

O DTA tem a vantagem de saber que a informação utilizada para calcular a confiança é verdadeira, pois foi o próprio agente que participou das interações e as avaliou. A limitação desse agente é que, na ausência de interações com determinado agente, o DTA não possui a capacidade de calcular o valor de confiança.

2.2.3.3. Reputação

A forma mais confiável de prever o comportamento de um agente é pelo histórico de interação direta com esse agente. Todavia, haverá casos em que essa interação ainda não ocorreu e será preciso estimar o nível de confiança no agente com o qual se deseja interagir. Assim, a reputação apresenta-se como uma métrica para avaliação do agente. Essa métrica envolve consultar outros nodos que estiveram em contato com esse agente no passado para colher informações de confiança.

Como definido anteriormente, o histórico de interações de a_3 com a_2 no tempo *t* pode ser representado por $\Re^t_{a_3,a_2} = \left(m^t_{a_3,a_2}, n^t_{a_3,a_2}\right)$. Semelhantemente, a opinião de a_3 com relação à a_2 como $\hat{\Re}^t_{a_3,a_2} = \left(\hat{m}^t_{a_3,a_2}, \hat{n}^t_{a_3,a_2}\right)$. Pode-se notar que, em geral, $\Re^t_{a_3,a_2} \neq \hat{\Re}^t_{a_3,a_2}$, pois a opinião provida não é imparcial, possui a tendência positiva ou negativa de alterar o que seria de fato resultado da interação. Caso o opinante seja honesto, $\Re^t_{a_3,a_2} = \hat{\Re}^t_{a_3,a_2}$.

O agente a_1 deve calcular o valor de reputação, $\tau^r_{a_1,a_2}$, em relação à a_2 pelas opiniões coletadas de outros agentes. As interações bem sucedidas e as mal sucedidas precisam ser enumeradas e somadas resultando nos valores de N_{a_1,a_2} e M_{a_1,a_2}, os quais serão utilizados

para calcular os parâmetros da distribuição beta. Esses parâmetros servirão para o cálculo da reputação (equação 2.7).

$$N_{a_1,a_2} = \sum_{k=0}^{p} \hat{n}_{a_k,a_2} \text{ , } M_{a_1,a_2} = \sum_{k=0}^{p} \hat{m}_{a_k,a_2} \text{ , onde p = número de reportes} \quad (2.5)$$

$$\hat{\alpha} = M_{a_1,a_2} + 1 \text{ e } \hat{\beta} = N_{a_1,a_2} + 1 \quad (2.6)$$

$$\tau^{r}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} \quad (2.7)$$

A figura 2.1 mostra a distribuição beta para opiniões providas por três agentes diferentes e a distribuição resultante da combinação dessas opiniões. Pela análise desse gráfico, é possível perceber que, considerando as opiniões separadas, a distribuição resultante da combinação possui menos variância, o que significa que o agente tem mais certeza no valor de confiança obtido pela distribuição da combinação.

Sabendo como calcular a reputação, é possível criar o agente de reputação (RTA – *Reputation Trust Agent*), que avalia a confiabilidade pelas informações coletadas de terceiros. Assim, para esse caso, toda a confiança de a_1 em a_2 é igual à reputação, $\tau_{a_1,a_2} = \tau^{r}_{a_1,a_2}$. Esse modelo do cálculo de confiança resolve o problema de não ter informações sobre o agente com o qual se deseja interagir, o que inviabilizaria o cálculo de confiança, como acontece com o DTA.

Se for considerado que o comportamento de um agente é sempre o mesmo independente do agente com o qual está interagindo e que aquele que provê a reputação está sempre dizendo a verdade, o valor de confiança resultante será o mesmo se for calculado tanto com base nas informações de terceiros ou na própria experiência direta com o agente. Entretanto, na prática, não é o que acontece. Então, o modelo precisa prever a existência de opiniões inconsistentes ou imprecisas.

Opiniões imprecisas não são somente resultado de ação maliciosa, mas podem ser causadas pela existência de informações incompletas no agente que provê essa opinião. Assim, é importante que o agente que está coletando as informações de confiança possa

avaliar a probabilidade de precisão daquela opinião dada. Dessa forma, a solução é ajustar ou ignorar as opiniões não confiáveis antes da combinação delas no valor total de reputação.

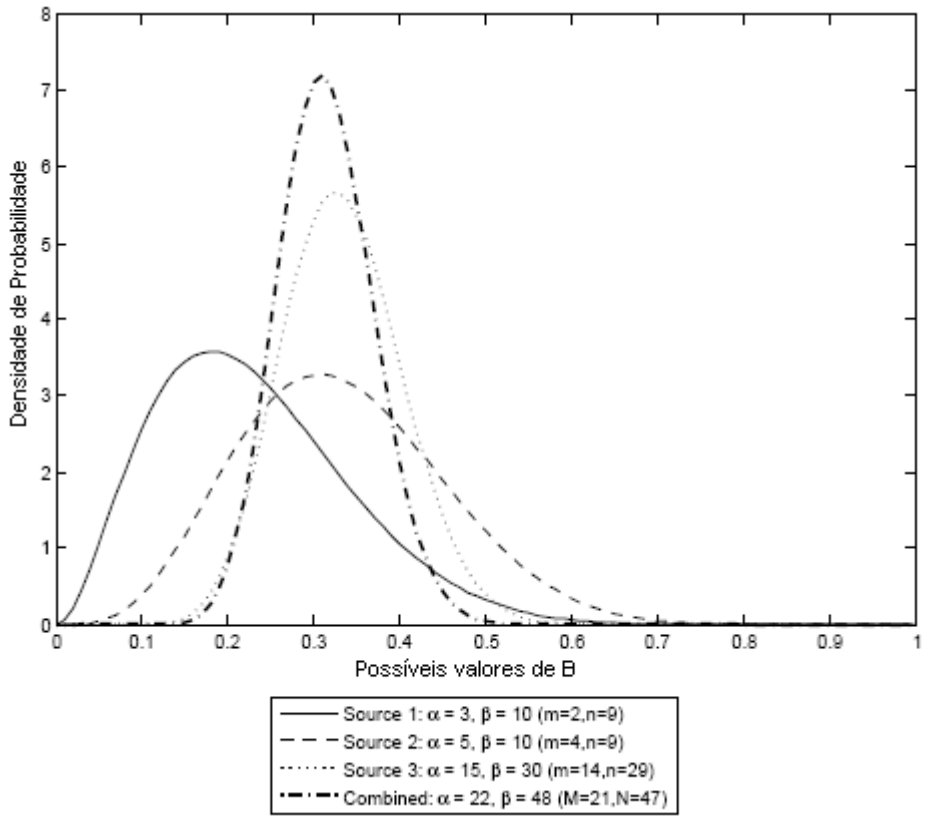

Figura 2.1 - Três opiniões separadas e a reputação calculada com a combinação delas (Adaptado de [3])

A probabilidade de precisão é uma métrica que representa o quanto o agente acredita que a opinião de determinado nodo é precisa, dados os resultados das interações passadas nas quais o mesmo nodo forneceu opiniões semelhantes. Assim, o primeiro passo para barrar opiniões imprecisas é guardar o histórico de opiniões providas e o resultado da interação na qual a opinião foi requerida.

As opiniões gravadas são representadas como $\hat{\Re}^t$. Todos os valores possíveis de $\hat{\Re}^t$ são divididos em faixas predefinidas, de acordo com o valor esperado $E_{\hat{\Re}^t}$ resultante da distribuição beta obtida de $\hat{\Re}^t$ (equação 3.8). Essas faixas possuem um limiar máximo ($bin_{\max}$) e um mínimo ($bin_{\min}$), sendo $\hat{\Re}^t$ pertencente a essa faixa. O histórico de opiniões H é representado por um conjunto de tuplas na forma (a_3, a_2, $bin_{\min}, bin_{\max}$, $\hat{\Re}^t_{a_3,a_2}$, $O^t_{a_3,a_2}$)

$$\mathrm{E}_{\hat{\Re}^t} = \frac{\alpha}{\alpha+\beta}, \text{ onde } \alpha = \hat{m}^t + 1 \text{ e } \beta = \hat{n}^t + 1 \tag{2.8}$$

O segundo passo para barrar opiniões imprecisas é calcular a probabilidade da opinião $\hat{\Re}^t_{a_3,a_2}$ provida por um agente específico ser precisa. Denota-se por $\rho^t_{a_1,a_3}$ a precisão da opinião provida por um opinante a_3 de acordo com a visão de a_1 no tempo *t*.

Para o cálculo dessa probabilidade, é necessário calcular o valor esperado de $\hat{\Re}^t_{a_3,a_2}$ para que seja possível determinar a faixa na qual essa opinião se encaixa, encontrando o limiar dessa faixa. Com isso, obtém-se um subconjunto, ***h***, de H que contém todas as tuplas que coincidem com (a_3, - , $bin^{\hat{\Re}^t_{a_3,a_2}}_{\min}$, $bin^{\hat{\Re}^t_{a_3,a_2}}_{\max}$, - , -), contendo opiniões anteriores de a_3 similares a $\hat{\Re}^t_{a_3,a_2}$. O conjunto ***h*** é utilizado para determinar os parâmetros α e β de uma distribuição beta que representa o comportamento atual de a_2, na perspectiva de a_1, em todas as situações nas quais o opinante a_3 provê uma opinião parecida com $\hat{\Re}^t_{a_3,a_2}$.

α = número de tuplas em ***h*** (onde O_{a_x,a_y} = 1) + 1 (2.9)

β = número de tuplas em ***h*** (onde O_{a_x,a_y} = 0) + 1 (2.10)

Dessa forma, a probabilidade de precisão $\rho^t_{a_3,a_2}$ é definida com a área abaixo da distribuição beta produzida utilizando ***h***, delimitada pelos limites da faixa pertencente a $\hat{\Re}^t_{a_3,a_2}$.

$$\rho_{a_1,a_3} = \frac{\int_{bin^{\hat{\Re}^t_{a_3,a_2}}_{\max}}^{bin^{\hat{\Re}^t_{a_3,a_2}}_{\min}} (\mathrm{B}_{a_1,a_2})^{\alpha-1}\left(1-\mathrm{B}_{a_1,a_2}\right)^{\beta-1} d\mathrm{B}_{a_1,a_2}}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1} dU}, \tag{2.11}$$

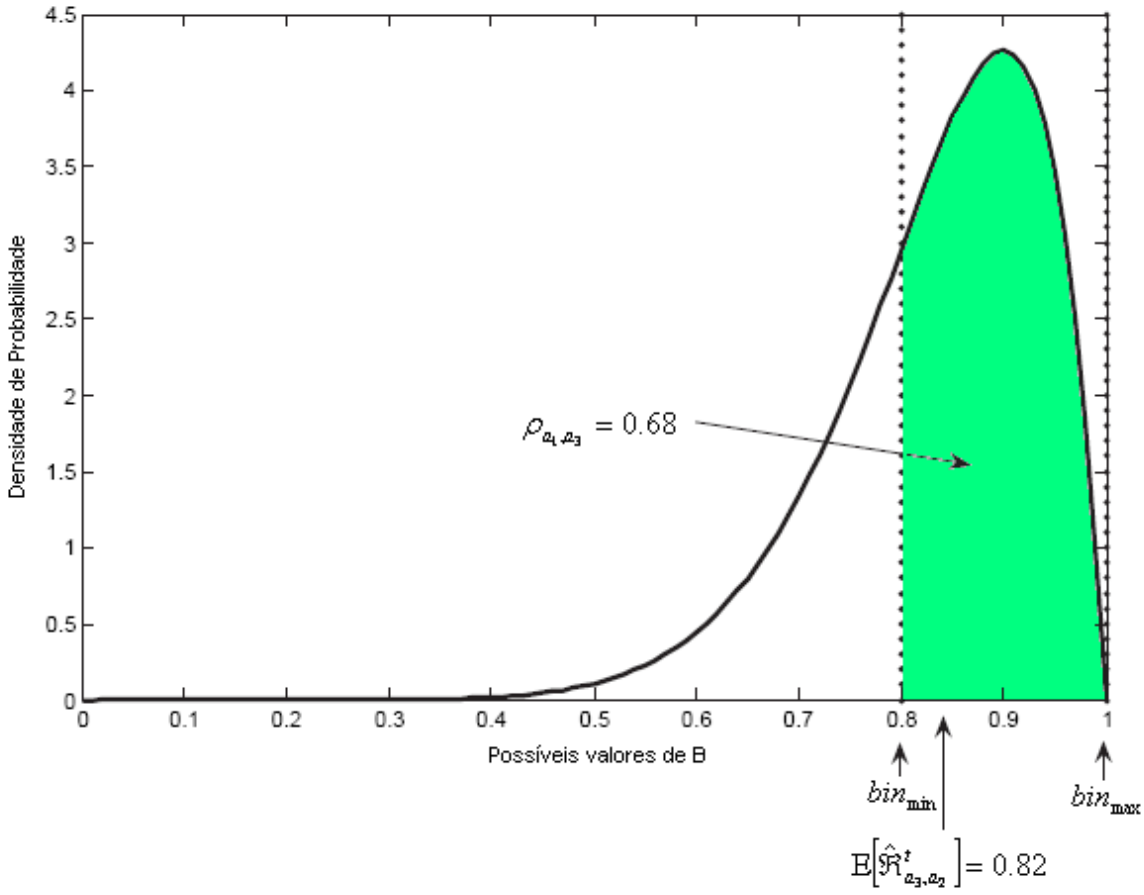

Figura 2.2 - Curva da distribuição beta demonstrando o alto valor de ρ obtido quando o opinante fornece opiniões precisas e honestas (Adaptado de [3])

Se o opinante a_3 tem falado sempre a verdade e provido opiniões precisas, com o passar do tempo o pico da curva da distribuição beta estará na faixa que $\hat{\Re}^t_{a_3,a_2}$ está presente, resultando em um elevado valor para $\rho^t_{a_3,a_2}$, conforme pode ser observado pela figura 2.2.

Por outro lado, se o agente opinante a_3 mente de forma constante e provê opiniões imprecisas, o pico da distribuição beta não estará na faixa na qual se encontra $\hat{\Re}^t_{a_3,a_2}$, resultando em um baixo valor para $\rho^t_{a_3,a_2}$. Esse comportamento pode ser observado pela figura 2.3.

O último estágio para barrar opiniões imprecisas é reduzir seu impacto na reputação de um nodo. Para tal, é preciso avaliar as propriedades da distribuição beta, analisando o efeito

do acréscimo de novas opiniões na combinação final. O desvio da curva contribui para a certeza da distribuição da combinação.

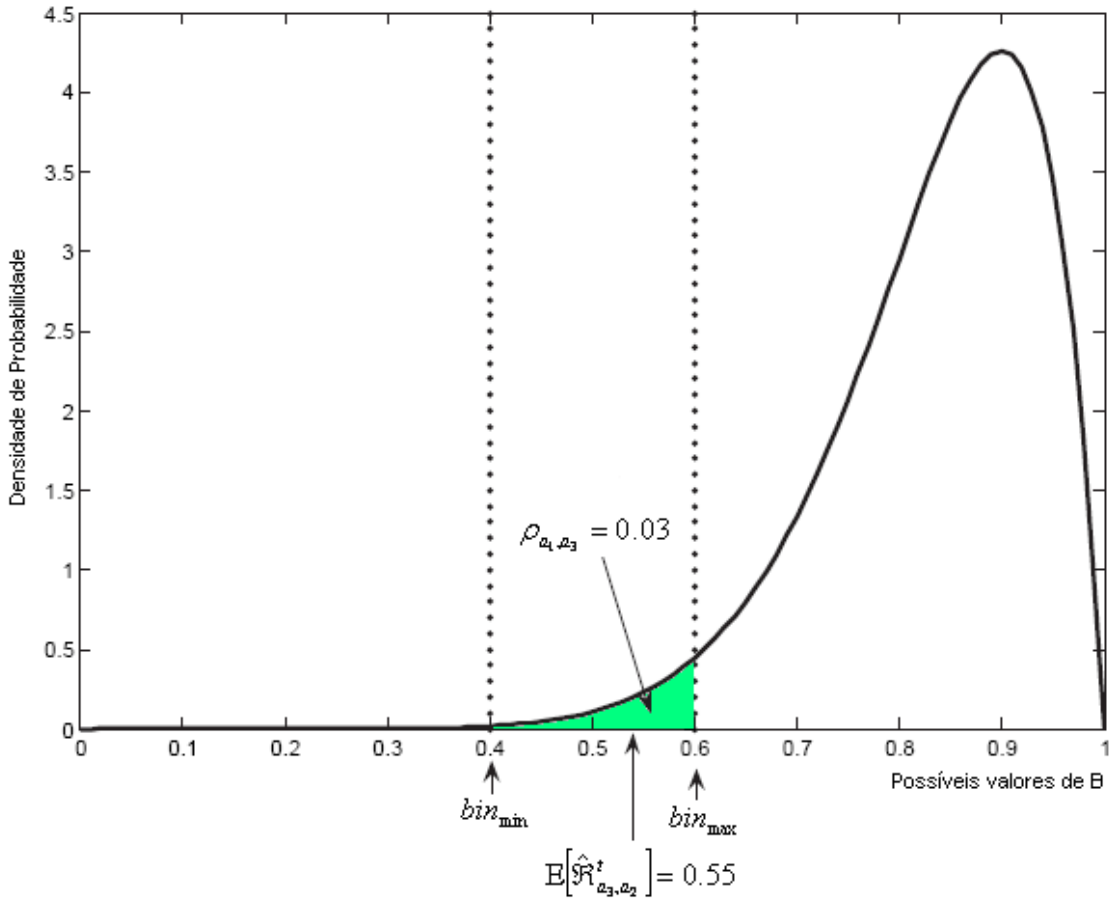

Figura 2.3 - Curva da distribuição beta demonstrando o baixo valor de ρ obtido quando o opinante fornece opiniões imprecisas e desonestas (Adaptado de [3])

O modelo utilizado visa diminuir a distância entre o valor esperado $E_{\hat{\Re}^t}$ e a variância $\sigma^2_{\hat{\Re}^t}$ (da distribuição da opinião) e a distribuição uniforme ($\alpha = 1$ e $\beta = 1$). Denota-se o valor esperado da distribuição uniforme como $E_{uniforme}$ e sua variância como $\sigma^2_{uniforme}$. Considerando um agente a_3 que provê opinião para a_1 sobre a_2, as equações 2.12 e 2.13 mostram essa redução de distância. A barra presente acima da letra (por exemplo, $\overline{\alpha}$) refere-se à distribuição ajustada.

$$\overline{\mathrm{E}}_{\hat{\Re}^t_{a_1,a_3}} = \mathrm{E}_{uniforme} + \rho_{a_1,a_3}.\left(\mathrm{E}_{\hat{\Re}^t_{a_1,a_3}} - \mathrm{E}_{uniforme}\right) \tag{2.12}$$

$$\overline{\sigma}^2{}_{\hat{\Re}^t_{a_1,a_3}} = \sigma^2{}_{uniforme} + \rho_{a_1,a_3}.\left(\sigma^2{}_{\hat{\Re}^t_{a_1,a_3}} - \sigma^2{}_{uniforme}\right) \tag{2.13}$$

Para combinar os valores das opiniões obtidas, além de ajustar todas as distribuições das opiniões, é necessário ajustar os parâmetros da distribuição beta e os valores de $\hat{m}^t$ e $\hat{n}^t$ que formam a opinião.

$$\overline{\alpha} = \frac{\left(\overline{\mathrm{E}}_{\hat{\Re}^t_{a_1,a_3}}\right)^2 - \left(\overline{\mathrm{E}}_{\hat{\Re}^t_{a_1,a_3}}\right)^3}{\left(\overline{\sigma}_{\hat{\Re}^t_{a_1,a_3}}\right)^2} - \left(\overline{\mathrm{E}}_{\hat{\Re}^t_{a_1,a_3}}\right) \tag{2.14}$$

$$\overline{\beta} = \frac{\left(1-\overline{\mathrm{E}}_{\hat{\Re}^t_{a_1,a_3}}\right)^2 - \left(1-\overline{\mathrm{E}}_{\hat{\Re}^t_{a_1,a_3}}\right)^3}{\left(\overline{\sigma}_{\hat{\Re}^t_{a_1,a_3}}\right)^2} - \left(1-\overline{\mathrm{E}}_{\hat{\Re}^t_{a_1,a_3}}\right) \tag{2.15}$$

$$\overline{m}_{a_3,a_2} = \overline{\alpha} - 1 \tag{2.16}$$

$$\overline{n}_{a_3,a_2} = \overline{\beta} - 1 \tag{2.17}$$

2.2.3.4. Combinação de Confiança Direta e Reputação

Utilizando distribuição beta para modelar o comportamento de um agente, é possível combinar as informações obtidas por experiência direta com as opiniões coletadas por outros agentes. Conforme apresentado anteriormente, o histórico de interações diretas entre os agentes a_1 e a_2 é representada por $\Re^t_{a_1,a_2} = \left(m^t_{a_1,a_2}, n^t_{a_1,a_2}\right)$ e as opiniões coletadas de outros nós por $\hat{\Re}^t_{a_3,a_2} = \left(\hat{m}^t_{a_3,a_2}, \hat{n}^t_{a_3,a_2}\right)$. Assim, para combinar a opinião pessoal com a reputação, é preciso enumerar todas as opiniões, como apresentado pela equação 2.5. Obtidas todas as

informações, é possível calcular os parâmetros da distribuição beta e o nível de confiança combinada $\tau^{c}_{a_1,a_2}$.

$$\hat{\alpha} = M_{a_1,a_2} + \overline{m}^{t}_{a_1,a_2} + 1 \text{ e } \hat{\beta} = N_{a_1,a_2} + \overline{n}^{t}_{a_1,a_2} + 1 \tag{2.18}$$

$$\tau^{c}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} \tag{2.19}$$

Sabendo como calcular a combinação de confiança direta e reputação, é possível criar o agente de confiança combinada (CTA – Combined Trust Agent). Assim, para esse agente, o nível de confiança total é igual ao nível de confiança combinada, $\tau_{a_1,a_2} = \tau^{c}_{a_1,a_2}$. A vantagem desse método é que ele utiliza tanto informações pessoais do agente como evidências externas obtidas de outros agentes. A desvantagem é que, mesmo com a combinação, poderá haver alguns agentes que exercerão influência no cálculo de confiança provendo informações imparciais, tanto para o lado negativo quanto para o positivo. Essa limitação pode ser contornada concedendo ao agente a habilidade de determinar a certeza presente em suas observações e somente buscar opiniões de terceiros se esse nível não for suficiente.

2.2.3.5. Grau de certeza

O grau de certeza γ_{a_1,a_2} é uma métrica que mede a probabilidade do valor atual B_{a_1,a_2} estar dentro de um nível de erro aceitável ε sobre τ_{a_1,a_2}. Essa métrica é calculada pela equação 2.20, cujo gráfico está representado na figura 2.4. O nível aceitável de erro ε influencia quão confidente é um agente, dado o mesmo número de observações.

$$\gamma_{a_1,a_3} = \frac{\int_{\tau_{a_1,a_2}+\varepsilon}^{\tau_{a_1,a_2}-\varepsilon} (\mathrm{B}_{a_1,a_2})^{\alpha-1}\left(1-\mathrm{B}_{a_1,a_2}\right)^{\beta-1} d\mathrm{B}_{a_1,a_2}}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1} dU} \tag{2.20}$$

Com a inserção desse conceito na implementação de um agente CTA, ocorre alteração no cálculo de confiança. O agente a_1 calcula τ_{a_1,a_2} baseado na sua experiência direta com a_2. Se o valor de τ_{a_1,a_2} tiver um grau de certeza correspondente γ_{a_1,a_2} abaixo de um valor mínimo predeterminado, denotado por θ_γ, a_1 buscará a opinião de outros agentes sobre a_2 de modo a obter um grau de certeza acima desse valor.

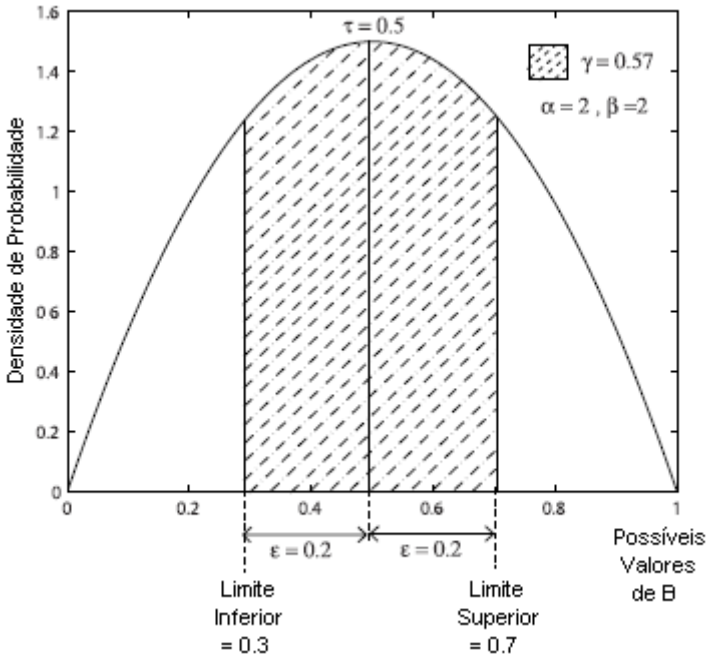

Figura 2.4: Certeza é a área abaixo da distribuição beta cercada pelos limites superior e inferior, calculados pela adição e subtração do erro ε obtido do valor de confiança τ (Adaptado de [3])

3. CONCEITOS SOBRE *GRID* COMPUTACIONAL, AGENTES DE *SOFTWARE, JADE E TEORIA DE GRUPOS.*

Este capítulo está dividido em duas partes distintas. A primeira parte, item 3.1, diz respeito a definição de *Grid* computacional, seus atributos, arquitetura, tipos e componentes. A segunda parte, item 3.2, introduz o conceito de agentes de *software* e aspectos relacionados, como a padronização FIPA [19] e a plataforma JADE [20].

3.1. *GRID* COMPUTACIONAL

Em várias empresas, existem recursos computacionais que são subutilizados, como processamento e espaço em disco. Tanto as estações de trabalho como os servidores possuem capacidade ociosa para ser utilizada. O g*rid* computacional fornece um *framework* para explorar esses recursos subutilizados, aumentando a sua eficiência de utilização. Neste trabalho, quando mencionarmos apenas *grid*, estaremos nos referindo ao *grid* computacional.

Nos últimos anos, a computação em *grid* evoluiu de um nicho tecnológico associado à computação científica e técnica para uma tecnologia inovadora de negócios que tem levado ao crescimento da sua adoção comercial. A implementação de *grid* acelera a performance de aplicações, melhora a produtividade, e otimiza a utilização da infra-estrutura de tecnologia da informação. Pela aceleração do desempenho de aplicação, empresas podem entregar com mais rapidez os resultados do negócio; alcançando maior produtividade, acelera o tempo de entrega da venda, e aumenta a satisfação dos clientes. Segundo Olso, Cowles, Mullen e Helm [21], o *grid* otimiza a utilização de recursos e reduz os custos, enquanto mantém uma infra-estrutura flexível que pode se adaptar a mudanças nas demandas dos negócios, mantendo a segurança e disponibilidade.

3.1.1. Definições Gerais

O conceito de *grid* como um ambiente distribuído, bem como sua implementação tecnológica, tem evoluído desde meados da década de noventa. O *grid* [21] evoluiu de uma infra-estrutura cuidadosamente configurada que suportava um número limitado de aplicações executadas em hardware de alta performance para um ambiente virtual dinâmico o qual provê, entre outras, soluções para problemas de compartilhamento de recursos coordenados.

Um dos primeiros conceitos que surgiu define *grid* computacional como uma infra-estrutura de hardware e software, que provê acesso dependente, consistente, penetrante, e não muito caro, a capacidades computacionais fins [21]. Essa definição particular vem do surgimento do *grid*, cuja finalidade era interconectar facilidades de alta performance entre vários laboratórios e universidades norte-americanas [21].

Com a evolução do *grid*, diversos autores conceituaram e publicaram definições sobre essa tecnologia. De acordo com Thompson [22], no conceito central de *grid*, é salientada a habilidade de negociar arranjos de compartilhamento de recursos entre um conjunto de participantes (consumidores e provedores). Sendo que, esse compartilhamento não é simplesmente de arquivos, mas também de acesso direto a computadores, software, dados e outros recursos. Além disso, o compartilhamento é, necessariamente, controlado e coordenado, com fornecedores e consumidores de recursos que definem de forma clara e exata o que pode ser compartilhado e sob que condições esse compartilhamento ocorre. Um grupo de indivíduos e/ou instituições que seguem essas regras de compartilhamento formam o que é chamado de organização virtual.

Foster [23] propõe uma lista de 3 pontos que definem um sistema *grid*. Primeiro, o *grid* é um sistema que coordena recursos que não são submetidos a um controle centralizado. Um *grid* integra e coordena recursos e usuários presentes em diferentes domínios, diferentes unidades administrativas de uma mesma empresa ou diferentes empresas; e lida com questões que aparecem nesse tipo de ambiente, como membresia, segurança, políticas, regras de acesso, entre outras. Segundo, esse tipo de ambiente distribuído utiliza interfaces e protocolos abertos, padronizados e de âmbito geral. O *grid* é construído com interfaces e protocolos de diversos propósitos que trabalham com questões fundamentais como autenticação, autorização, descoberta de recursos e acesso a recursos. Por último, o *grid* consiste em um sistema o qual possui entrega de qualidade de serviço não trivial. Esse sistema permite a utilização dos seus recursos constituintes de forma coordenada para entregar diversos tipos de qualidades de serviço, relacionadas a tempo de resposta, *throughput*, disponibilidade e segurança, ou alocação de múltiplos tipos de recursos para atender a demandas complexas de tal forma que a utilização do sistema combinado é significantemente maior que a soma das suas partes.

Outro conceito mais simplificado de *grid* é apresentado por Papalilo e Freisleben [24], que o define como um ambiente de tecnologia da informação distribuído e flexível o qual permite a criação de múltiplos serviços com um grau de independência significante dos atributos específicos presentes na infra-estrutura de suporte.

Em suma, o *grid* pode ser considerado um conjunto de máquinas que compartilham um ou mais recursos/serviços, sem um controle centralizado. Os membros do *grid* devem utilizar protocolos e interfaces padronizados para que haja a comunicação entre as máquinas, permitindo a integração. Nesse tipo de sistema, a interoperabilidade é fundamental visto que, para um funcionamento real e eficiente, partes com diferentes plataformas, linguagens e ambientes de programação devem conversar entre si. Para que essa comunicação aconteça, é necessária a definição de como os elementos desse sistema distribuído vão interagir. Assim, existem os protocolos que especificam essa padronização de comunicação.

Aplicações em *grids* são distintas das aplicações tradicionais do tipo cliente-servidor devido ao uso de grandes quantidades de recursos com necessidades dinâmicas. Estes recursos são tipicamente extraídos de múltiplos domínios administrativos interconectados por estruturas complexas de comunicações, e precisam ser acessados com requisitos de desempenhos estritos [22].

Além do compartilhamento de recursos, outros benefícios decorrentes da implementação de *grid* são: a possibilidade de acesso transparente a sites remotos; a possível redução do número de servidores; a otimização da utilização dos recursos da rede; provê tolerância a falhas; permite a realização de um centro de dados virtual.

Um desafio de qualquer tecnologia computacional é conseguir a comunicação entre diversos componentes. Para haver essa interoperabilidade, é necessário a existência de uma padronização. No caso do *grid*, o órgão responsável pelo desenvolvimento e manutenção desse padrão é o OGF – *Open Grid Forum*. Constitui em um fórum para troca de informações e definição de padrões relacionados à computação distribuída e tecnologias de *grid*.

3.1.2. Atributos

Algumas características presentes em ambientes em *grid* conferem a esse sistema atributos importantes para diversas aplicações. Dependendo da aplicação utilizada ou do

serviço a ser provido, esses atributos são explorados conforme a necessidade verificada pelo projetista do sistema. Entre os atributos mencionados por Thompson [22], destacam-se:

(a) Abstração/virtualização. O nível de abstração em *grid* suporta várias categorias de aplicações inovadoras que não podem ser criadas com infra-estrutura tradicional, porque provê métodos únicos de reduzir dependências locais específicas e de compartilhamento de recursos e integração.

(b) Compartilhamento de recursos. Ambientes em *grid* permitem o compartilhamento de recursos em larga escala.

(c) Determinação. Processos no *grid* permitem que aplicações garantam, por processos autônomos, a compatibilidade com níveis de serviços apropriados e recursos requeridos.

(d) Gerência e controle descentralizado. Sua arquitetura suporta a descentralização da gerência e controle de recursos.

(e) Escalabilidade. Ambientes em *grid* podem ser implementados localmente ou distribuídos por grandes regiões geográficas, permitindo o alcance de capacidades especializadas a sites remotos.

(f) Desempenho elevado. *Grids* podem prover serviços com desempenho muito elevado pela agregação de múltiplos recursos.

(g) Segurança. *Grids* podem ser muito seguros, especialmente quando técnicas de segmentação são utilizadas para isolar áreas particionadas do ambiente.

(h) Customização. *Grids* podem ser customizados para atender requisitos, condições e recursos muito especializados.

Uma das características mais conhecidas e mais exploradas na implementação de *grid* é o compartilhamento de recursos, para atender às diversas necessidades da rede, seja de desempenho ou da existência do próprio recurso remotamente. Esse compartilhamento não consiste somente em troca de arquivos, mas, também, acesso direto a máquinas, software, dados e outros recursos. Nesse processo, ocorre um controle do que é compartilhado, definindo o que será compartilhado, quem tem permissão de utilização e em quais condições o compartilhamento ocorre.

Entre esses recursos, um dos mais utilizados é o de processamento. Para esse caso, existe o aproveitamento da capacidade ociosa do processador de algumas máquinas na realização de tarefas designadas por outras máquinas e o particionamento de serviços em subtarefas para a execução paralela em diferentes máquinas.

Outro recurso compartilhado no *grid* é o dispositivo de armazenamento. A otimização da utilização desse recurso provê o aumento da capacidade e da performance de sistemas. Além disso, pode haver a exploração de mecanismos de redundância, tornando o ambiente tolerante a falhas.

Não se pode deixar de mencionar a capacidade de comunicação que deve aumentar com a implementação do *grid*. Considerando que as máquinas presentes nesse agrupamento possuem conexões para internet independentes, não ligadas ao mesmo caminho, haverá um aumento na banda total de acesso à internet, o que pode ser interessante para determinadas aplicações.

Também é possível considerar a utilização de licenças por várias máquinas a partir de uma estação. Visto que existem softwares que possuem alto custo ligado a sua licença, haverá uma economia no que diz respeito a esse tipo de gasto em ambiente de *grid*.

3.1.3. Arquitetura

A arquitetura do grid compreende um agrupamento de camadas lógicas, dispostas hierarquicamente, conforme pode ser observado na figura abaixo [22].

Figura 3.1 - Arquitetura do Grid

Na base dessa estrutura, encontra-se o *grid fabric*. Consiste nos recursos distribuídos que são gerenciados por um administrador de recursos local com aplicação de política local. Esses recursos são interconectados via rede local, metropolitana ou *wide-area networks*. O *grid fabric* engloba redes; computadores, como computadores pessoais e processadores os quais utilizam sistemas operacionais como Unix, Linux ou Windows; *clusters*, utilizando vários sistemas operacionais; sistemas de gerenciamento de recurso; dispositivos de armazenamento; e banco de dados.

A camada de infra-estrutura de segurança provê conectividade e acesso seguro e autorizado aos recursos e serviços do *grid*. Implementa mecanismos de autenticação, autorização, confidencialidade, integridade e disponibilidade de dados.

Acima dessa camada, o core *grid middleware* provê acesso uniforme a recursos na *fabric*. Constitui em um *middleware* projetado para abrigar as complexidades de particionamento, distribuição e balanceamento de carga. Engloba o gerenciamento de tarefas, acesso a dispositivos de armazenamento e auditoria.

A próxima camada, a *middleware* de nível de usuário, consiste de *resource brokers* e *schedulers* responsáveis por agregar recursos. Provê serviços de agendamento e gerenciamento de recursos. Os *brokers* possuem informação sobre os recursos disponíveis no *grid* e sobre o estado de trabalho desses recursos. Os *schedulers* coordenam a execução das tarefas na utilização dos recursos, gerenciando a utilização concorrente.

A camada de ferramentas e ambientes de programação do *grid* inclui os compiladores, bibliotecas e ferramentas de desenvolvimento, os quais são recursos necessários à execução das aplicações.

A camada superior consiste das aplicações do *grid* propriamente ditas. Nesse caso, podem ser consideradas aplicações comerciais, científicas, de engenharia, entre outras.

3.1.4. Tipos

Os *grids* podem ser classificados de diversas formas, como configuração física, topologia e localização. Segundo Ferreira [25], os *grids* são agrupados em três tipos. A primeira categoria consiste no *grid* mais simples, composto por poucas máquinas com mesma arquitetura de hardware e mesmo sistema operacional, conectadas na rede local. Essas

máquinas podem estar localizadas em um departamento de uma organização. Assim, essa implementação é denominada, em muitos casos, "*cluster*" em vez de "*grid*".

O segundo tipo seria uma progressão do primeiro com a inclusão de máquinas heterogêneas com a disponibilidade de mais tipos de recursos e a inclusão de alguns componentes de agendamento e gerência. Máquinas presentes nesse arranjo podem fazer parte de vários departamentos, mas na mesma organização. Esse *grid* é referenciado como "*intragrid*".

Com a expansão do *grid* para vários departamentos, políticas de utilização e de segurança passam a ser requeridas. Além disso, critérios de prioridade e controle de acesso tornam-se necessários. O *grid* pode crescer geograficamente em uma organização que tem facilidades em diferentes cidades, as quais devem possuir conexões dedicadas entre si para comunicação. Essa categoria é conhecida como "*intergrid*".

3.1.5. Componentes

Para melhor entendimento do *grid*, é importante conhecer seus componentes principais, como componentes de gerência, *schedulers* e softwares que auxiliam na comunicação. De acordo com Ferreira [25], será realizada um breve explanação de alguns desses componentes.

Qualquer *grid* deve possuir componentes de gerência para controlar os recursos disponíveis, os membros presentes no *grid*, os dados e os serviços. Esses componentes também servem para monitorar as máquinas, gerando estatísticas e alertando em caso de eventuais problemas. Em situações de grandes ambientes distribuídos, a gerência do *grid* como um todo também é distribuída, em níveis hierárquicos, para aumentar sua escalabilidade.

Cada máquina que fará parte do *grid* precisa tornar-se membro, instalando um software que realizará a gerência da utilização dos seus recursos. Para o acesso aos recursos das outras máquinas, é atribuído um identificador ao qual são concedidas permissões de acesso e utilização aos recursos disponibilizados. Em alguns casos, é necessária a instalação de software que concede à máquina o direito de submeter o serviço ou tarefa ao *grid*.

Outro componente que está presente na maioria dos *Grids* funciona como um agendador de tarefas (*scheduler*). Esse componente verifica qual máquina possui o recurso necessário

para a execução de uma tarefa solicitada por um usuário. Em algumas situações, há o estabelecimento de prioridades de acordo com quem solicita o recurso.

Além dos componentes já mencionados, outro também presente no *grid* é o software para auxiliar na comunicação entre as tarefas que estão sendo executadas. Como um serviço pode ser dividido em partes, é preciso que as subtarefas possam localizar-se caso seja necessário trocar dados entre si.

Não se pode deixar de levar em conta que não é qualquer aplicação que roda em *grid* ou que utiliza de maneira eficiente os benefícios oferecidos por esse tipo de ambiente. Algumas aplicações não aceitam a divisão em subtarefas para execução e processamento em paralelo. Outras requerem um trabalho muito árduo para permitirem sua execução e otimização em *grid*. Além disso, diversos fatores devem ser levados em conta na implementação dessa tecnologia em uma empresa, como segurança e performance.

3.2. AGENTES DE *SOFTWARE*

Na visão de Patel [3], um agente de *software* é uma entidade autônoma capaz de resolver problemas de forma flexível, sendo que, de forma geral, soluções baseadas em agentes de *software* envolvem vários desses agentes. Um sistema multi-agente é um sistema modular no qual uma tarefa pode ser subdividida em partes menores e resolvida por um único agente. Assim, a solução geral será produzida por meio da interação entre esses agentes. Apesar de agentes de *software* serem úteis para uma grande variedade de aplicações, eles são especialmente indicados para aplicações que envolvem processamento aberto, complexo e distribuído, e comunicação entre computadores em rede.

Agentes autônomos são a chave para realizar a visão do *Grid*. Sistemas *Grid* baseados em VOs podem ser modelados por um sistema multi-agente cujos agentes pertencem a diferentes organizações, têm diferentes capacidades de resolver problemas e podem decidir se interagem e cooperam com outros agentes. Neste caso, eles estão associados a uma organização e é provável que agentes pertencentes a organizações diferentes trabalhem juntos para atingir um objetivo. Porém, mesmo trabalhando junto, cada um tem sua própria meta, o que pode levar a um comportamento egoísta, com agentes interagindo de forma a maximizar seu próprio ganho, talvez até em detrimento de outro, manipulando ou escondendo partes da

informação para seu próprio benefício. Em um ambiente *Grid*, é essencial garantir boas interações entre os agentes para que nenhum agente tire vantagem de outro [3]. Ainda neste cenário, podem existir agentes que saem e entram novamente no sistema para evitar penalidades por ações passadas e outros que possuem comportamentos diferentes e terão que decidir quanto a interagir com agentes com características diferentes.

Para estudar confiança entre agentes de *software Grid*, desenvolvemos um ambiente utilizando agentes JADE (*Java Agent DEvelopment Framework*) [20], que oferecem diversas facilidades para comunicação e interação entre agentes. Os agentes JADE simplificam a construção de sistemas multi-agente por meio de um *middleware*. *Middleware* é um termo usado para descrever bibliotecas de alto nível que facilitam o desenvolvimento de aplicações provendo serviços genéricos que são úteis não só para uma única aplicação, mas para uma variedade delas. A idéia é prover facilidades nativas, agregadas às APIs, independentes do sistema operacional utilizado. A plataforma de agentes pode ser distribuída entre diferentes máquinas e a configuração pode ser remota por meio de uma interface gráfica remota.

3.2.1. Definição

Não existe um conceito único para definir agentes de *software*. Os diversos autores possuem abordagens diferentes que são direcionadas conforme o objetivo requerido. De acordo com a aplicação citada ou desenvolvida, o conceito de agente de *software* segue um ponto de vista determinado e específico. Para facilitar a compreensão, a partir de agora, mesmo quando não mencionarmos explicitamente que o agente do qual estamos falando é um agente de *software*, ficará subtendido que são agentes de *software*.

Mesmo não havendo uma definição única para agentes, as definições existentes concordam que um agente é essencialmente um componente especial de software que tem autonomia a qual provê uma interface interoperável em um sistema arbitrário e/ou se comporta como um agente humano, trabalhando para alguns clientes de acordo com um cronograma [25].

Segundo Wooldridge [26], agentes seriam entidades de software autônomas que atuam em determinado ambiente de forma a interagir com este e com outros agentes, produzindo ações e percepções sem requerer intervenções humanas constantes. Para o caso ideal,

considerando uma abordagem aplicada à Inteligência Artificial, o agente teria que ser capaz de funcionar continuamente e adquirir experiências e conhecimentos acerca do ambiente que está interagindo.

Genesereth e Ketchpel [27] definem agentes de *software* como componentes que se comunicam com nodos pela troca de mensagens utilizando uma linguagem de comunicação específica. Os agentes podem ser simples como sub-rotinas; mas, tipicamente, eles são entidades maiores com algum tipo de controle persistente, como o controle de threads e processos.

Para se obter uma noção melhor do que é um agente, Wooldridge [26] enumera algumas de suas propriedades:

(a) Autonomia – agentes funcionam sem intervenção e têm controle sobre suas próprias ações e estado interno. Ou seja, são capazes de agir de maneira independente com seu ambiente através de suas próprias percepções com o objetivo de realizar alguma tarefa, seja ela externa ou gerada por ele próprio [26].

(b) Sociabilidade – agentes interagem com outros agentes e humanos usando uma linguagem de comunicação definida. Com o estabelecimento dessa interação, quando há cooperação entre as partes, aumentam as chances de sucesso no alcance das metas estabelecidas pelos agentes.

(c) Reatividade – agentes são capazes de perceber seu meio e agir em resposta às mudanças nesse meio.

(d) Proatividade – as ações de um agente são guiadas por metas que definem seu comportamento, tornando-os pró-ativos e não somente reativos a estímulos externos. Assim, o agente possui a capacidade de tomar iniciativa por conta própria.

Além dessas propriedades, um agente pode ser móvel [25], com a habilidade de se locomover entre diferentes nodos em uma rede de computadores; verdadeiro e honesto, provendo a certeza de que não irá passar informações incorretas; benevolente, sempre tentando fazer o que foi pedido a ele; racional, sempre agindo de forma a alcançar suas metas; e aprendiz, adaptando-se aos ambientes e aos desejos dos seus usuários. Outra característica importante, mencionada por Wooldridge [26], é a robustez que o agente deve possuir para ser

capaz de tomar decisões baseando-se em informações incompletas e escassas, conseguindo lidar com erros e aprendendo com a experiência.

Todas as propriedades citadas acima são importantes quando presentes em um componente de software caracterizado como agente. Todavia, para ser considerado um agente, esse componente não precisa necessariamente possuir todas essas características ao mesmo tempo.

3.2.1.1. Tipos

Os agentes podem ser classificados de acordo com diversos aspectos como mobilidade, tipo de interação entre agentes e capacidade de raciocínio. Quanto à mobilidade, podem ser categorizados em dois tipos: móveis e estacionários. Quanto ao tipo de interação entre agentes, em competitivos ou colaborativos. Quanto à capacidade de raciocínio, em reativos ou cognitivos.

Segundo Wooldride [26], os tipos citados acima são caracterizados da seguinte forma:

(a) Agentes móveis: consistem em agentes que possuem a mobilidade como característica principal. Esses agentes têm a capacidade de se mover tanto pela rede local como pela web, deslocando-se entre as plataformas. Seu uso tem crescido devido a alguns fatos como a heterogeneidade cada vez maior das redes e seu grande auxílio na tomada de decisões baseadas em grandes quantidades de informação.

(b) Agentes estacionários: ao contrário dos móveis, esses agentes não possuem mobilidade, permanecendo fixos em um mesmo ambiente.

(c) Agentes competitivos: são agentes que agem em interesse próprio. Constituem agentes que “competem” entre si para a realização de seus objetivos ou tarefas.

(d) Agentes colaborativos: consistem em agentes que realizam suas ações visando o bem maior de todos, buscando alcançar um objetivo coletivo. Realizam tarefas específicas coordenando-as entre si de forma que suas atividades se completem.

(e) Agentes reativos: são agentes que agem em resposta a estímulos do ambiente, sem ter memória do que já foi realizado no passado e nem previsão da ação a ser tomada no futuro. Não possuem capacidade de raciocínio sobre suas intenções, sendo incapazes de prever e

antecipar ações. Geralmente, atuam em sociedades, pois sua força vem da capacidade de formar um grupo capaz de se adaptar a um ambiente [26][28].

(f) Agentes cognitivos: também conhecidos como intencionais, racionais ou deliberativos, possuem a habilidade de raciocínio sobre suas intenções e crenças, e podem, por um processo explícito, criar e escolher ações e planos a serem realizados. Esses planos podem ser revisados em caso de detecção de conflitos [28]. Para planejar as ações, cada agente deve ter uma base de conhecimento disponível, que compreende todos os dados e toda a experiência para realizar suas tarefas e interagir com outros agentes e com o próprio ambiente. Devido à sua capacidade de raciocínio baseado nas representações do mundo, esses agentes são capazes de memorizar situações, analisá-las e prever possíveis reações [24].

3.2.1.2. Comunicação e Coordenação

Um componente importante em um sistema multi-agente é a comunicação. É fundamental que o agente saiba e possa se comunicar com outros agentes e com os recursos do sistema para que haja cooperação, colaboração e negociação entre os componentes. Assim, um padrão de interação precisa ser estabelecido, o que é obtido pela existência de uma linguagem comum.

A primeira linguagem de comunicação de agentes mais vastamente utilizada foi KQML (Knowledge Query and Manipulation Language). Essa linguagem, que foi desenvolvida pela ARPA Knowledge Sharing Effort, consiste em linguagem e protocolo de troca de informações e conhecimento. KQML provê uma arquitetura básica de compartilhamento de conhecimento por uma classe especial de agentes chamados facilitadores de comunicação que coordenam as interações de outros agentes [28]. Atualmente, a linguagem de comunicação de agentes mais utilizada e estudada é a FIPA ACL, que incorpora vários aspectos do KQML.

Para que haja interação e comunicação de maneira correta entre os agentes, é necessário haver um mecanismo de coordenação das ações de cada membro do grupo. Coordenação é um processo no qual vários agentes procuram assegurar que a comunidade de agentes individuais comportem-se de maneira coerente. Existem várias razões do motivo pelo qual vários agentes precisam estar coordenados [25]:

(a) As metas dos agentes podem causar conflitos entre as ações dos agentes;

(b) As metas dos agentes podem ser interdependentes;

(c) Os agentes podem ter capacidades e conhecimentos diferentes;

(d) Os alvos dos agentes podem ser mais rapidamente alcançados se agentes diferentes trabalharem em cada um deles;

A coordenação entre agentes pode ser assegurada por diversas abordagens o que inclui estruturação organizacional, contratação, negociação e planejamento multi-agente. Dessas, a negociação é uma das mais importantes técnicas. Consiste em um processo realizado entre um conjunto de agentes que estabelecem um acordo entre si. A negociação pode ser competitiva ou cooperativa, dependendo do comportamento dos agentes envolvidos. A competitiva é utilizada em situações onde os agentes possuem metas independentes; assim, não há o intuito de estar cooperando e contribuindo para um bem maior de todos. Já a negociação cooperativa é utilizada quando os agentes possuem um alvo comum; todos os nodos trabalhando juntos para executar uma tarefa e atingir uma meta única global [25].

3.2.1.3. Ferramentas e Linguagens de Programação

Sistemas multi-agente [25] podem ser implementados por qualquer tipo de linguagem de programação, mas a que mais se encaixa no perfil dos agentes é a linguagem orientada a objeto, pois existem semelhanças no conceito de agente e de objeto. Os agentes possuem várias propriedades também presentes nos objetos, como encapsulamento e herança. Todavia, também existem as diferenças evidentes. De acordo com Weiss [29], os agentes possuem uma noção mais forte de autonomia em relação aos objetos, são capazes de um comportamento flexível e um sistema multi-agente é inerentemente multi-thread, possuindo mecanismos de controle de threads.

Plataformas e frameworks também contribuem na implementação de sistemas multi-agente, pois funcionam como um meio de rodar esse tipo de sistema em diferentes tipos de hardware e sistema operacional, provendo, normalmente, um middleware para suportar a execução e as operações essenciais como comunicação e coordenação. Algumas dessas plataformas e frameworks disponibilizam funcionalidades em acordo com as especificações da FIPA para suportar a interoperação de diferentes sistemas multi-agente. A FIPA [19] é uma organização regulamentadora que define padrões relacionados aos agentes. Mais adiante, na seção 3.2.2, será realizada uma melhor explanação sobre essa organização.

Uma característica importante que deve ser provida pelos sistemas multi-agente é a capacidade de interoperabilidade entre sistemas de software legado. Dessa forma, a disponibilidade de ferramentas de software para a integração entre eles com outras tecnologias pode ser a chave para o sucesso de tipo de sistema [25].

Com isso, sistemas multi-agente estão sendo utilizados em uma grande variedade de aplicações, desde sistemas pequenos para assistência pessoal até sistemas de missão crítica complexos e abertos para aplicações industriais. Dentre os campos de aplicações desses sistemas, existem: gerenciamento de informação; processos de tráfego e transporte; sistemas de telecomunicação; coordenação entre robôs diferentes; e em cuidados com a saúde, como serviços de monitoração e agendamento de pacientes.

3.3. A PLATAFORME JADE

A plataforma JADE é basicamente um ambiente de desenvolvimento de Sistemas Multiagentes. Como trata do ciclo de vida e da comunicação do Agente, além de auxiliar no monitoramento da execução [20], tal framework torna mais simples o desenvolvimento de Sistemas Multiagentes.

Como qualquer agente pode, a qualquer momento, ser objeto de uma próxima comunicação, ou iniciar com qualquer outro agente a comunicação, o modelo de arquitetura de uma aplicação orientada a agentes é intrisicamente peer to peer [30].

O desenvolvimento do software que veio a se tornar a plataforma JADE foi iniciado pela Telecom Italia (anteriormente CSELT) no final de 1998, motivado pela necessidade de validar o início das especificações FIPA. JADE se tornou open source em 2000 e foi distribuído pela Telecom Italia sob a licença LGPL (Library Gnu Public Licence) [31]. A licença do LGPL fornece alguns direitos e também deveres. Dentre os direitos estão o acesso ao código fonte do software, a permissão de fazer e distribuir cópias, a permissão para fazer melhorias e funcionalidades e a incorporação do JADE a programas proprietários. Já entre os deveres estão não fazer modificações privadas e não alterar a licença do JADE e suas modificações. O JADE possui um site no qual o software, a documentação, os exemplos de códigos e outras informações sobre os usos do JADE estão disponíveis [20].

FIPA é um acrônimo para Foundation for Intelligent Physical Agents. Consiste em uma organização internacional sem fins lucrativos destinada ao desenvolvimento de padrões de software voltados à utilização em sistemas baseados em agentes [19]. Foi criada em 1996 e

localiza-se em Genebra, na Suíça. A FIPA cria, divulga e provê especificações a fim de fornecer maior interoperabilidade entre Sistemas de Agentes. O JADE contribuiu para a difusão das especificações da FIPA, pois ele fornece um conjunto de abstrações de software e ferramentas que proporcionam aos programadores a capacidade de aplicar as especificações da FIPA, sem a necessidade de estudá-las.

3.3.1. Características da Plataforma JADE

O JADE representa uma plataforma de software que possui funcionalidades básicas voltadas para o auxílio da realização de aplicações distribuídas de Agentes de software. Uma característica importante do JADE é que ele implementa a abstração das especificações da FIPA por meio da linguagem orientada a objetos, Java, fornecendo uma API simples e amigável. A plataforma JADE fornece o seguinte núcleo de funcionalidades [20]:

a) Plena conformidade com as especificações da FIPA;
b) Um conjunto de ferramentas gráficas para apoiar os programadores na depuração e no monitoramento dos Agentes;
c) Uma API de localização independente, que representa a base da infraestrutura de comunicação entre Agentes. Cada Agente é executado em uma thread diferente, potencialmente em máquinas remotas. Oferece a capacidade de se comunicar de forma transparente com outro Agente;
d) Eficiente transporte de mensagens assíncronas por meio de uma API de localização e transporte, locationtransparent API. A plataforma seleciona os melhores meios de comunicação disponíveis, e, ao atravessar as fronteiras da plataforma, as mensagens são automaticamente transformadas da representação interna de Java do JADE para a sintaxe compatível com a FIPA;
e) Implementações de white pages e yellow pages, que consistem, respectivamente, nos serviços de Gerenciamento de Agentes e Localização e Registro de Agentes. Sistemas federados podem ser implementados para representar domínios e subdomínios, tal como um gráfico de diretórios federados;
f) Um simples, mas eficaz, gestor de ciclo de vida de Agente. Quando os Agentes são criados, recebem automaticamente um identificador global único e um endereço de transporte, que são utilizados para se registrar com os serviços de white pages da sua

plataforma. A plataforma fornece, também, APIs e ferramentas gráficas simples, para gerenciar, tanto local quanto remotamente, os ciclos de vida dos Agentes;

g) Suporte para mobilidade do Agente. O código do agente e, com certas restrições, o estado do agente podem migrar entre processos e máquinas;
h) Suporte para ontologias e linguagens de conteúdo. É possível implementar novas linguagens de conteúdo para satisfazer requisitos de aplicações específicos.

3.3.2. Arquitetura JADE

A plataforma JADE é composta por containeres de agentes que podem ser distribuídos na rede. Existe um *container* especial, chamado *Main Container*, que representa o ponto de *bootstrap* de uma plataforma: é o primeiro *container* a ser lançado, e todos os outros containeres devem aderir a um *container* principal, registrando-se com ele. Por padrão, o *container* principal é chamando '*Main Container*', enquanto os outros são chamados '*Container-1*', '*Container-2*' etc, como mostrado na Figura 3.2.

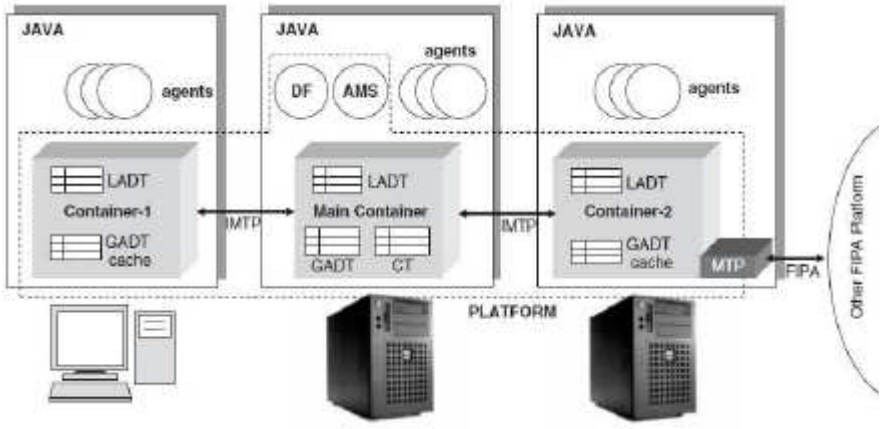

Fig 3.2 – Relação entre os elementos da arquitetura principal [31]

Em resumo, uma plataforma é composta por um ou muitos Agentes, que vivem em containeres e se registram a um *Main Container*. O *Main Container* possui as seguintes responsabilidades [31]:

a) Gerenciamento da tabela do container (*container table* - CT), que é o registro das referências do objeto e endereços de transporte de todos os containeres nodos que compõem a plataforma;

b) Hospedagem do AMS (*Agent Management System*) e do DF (*Directory Facilitator*), os dois agentes especiais que proporcionam os serviços de *white page*, bem como o serviço padrão de *yellow page* da plataforma. Quando um Agente deseja registrar um serviço que pode prover, ele se registra no Agente DF da plataforma. Caso um Agente busque por um serviço, é ao Agente DF que é solicitada a realização dessa busca, e este, por sua vez, caso a busca retorne um resultado positivo, devolve o nome do Agente capaz de prover o serviço procurado.

A identidade do agente está contida em um *Agente Identifier* (AID), composto por um conjunto de *slots* que respeitam a estrutura e a semântica definidas pela FIPA. A maioria dos elementos básicos do AID são os nomes de agentes e seus endereços. O nome de um agente é um identificador global único que JADE constrói por concatenamento de um *nickname* definido pelo usuário (também conhecido como um nome local, uma vez que é suficiente para desambiguação na comunicação intraplataforma) com o nome da plataforma [31].

Quando o *Main Container* é lançado, dois agentes especiais são automaticamente instanciados e inicializados pelo JADE, cujos papéis são definidos pelo padrão *FIPA Agent Management*:

a) O *Agent Management System* (AMS) é o agente que supervisiona toda a plataforma. É o ponto de contato para todos os agentes que têm de interagir, a fim de usufruir do serviço de *white page* da plataforma, bem como de gerir seu ciclo de vida. Cada agente é obrigado a registrar-se com o AMS (automaticamente realizado por agentes JADE no *start-up*), a fim de obter um AID válido.
b) O *Directory Facilitator* (DF), como citado anteriormente, é o agente que executa o serviço de páginas amarelas (*yellow pages*), usado por qualquer agente que pretenda registrar seus serviços ou buscar por outros serviços disponíveis. O JADE DF também aceita inscrições de Agentes que desejarem ser notificados sempre que um registro de serviço ou modificação for realizada. Múltiplos Agentes do tipo DF podem ser iniciados concomitantemente, a fim de distribuir o serviço de páginas amarelas em vários domínios. Esses DF’s podem ser federados, se necessário, por meio da criação de registros cruzados com um outro, permitindo a propagação do pedido do agente em toda a federação [31].

3.4. TEORIA DE GRUPOS

Grupos são usados na Matemática e nas ciências em geral para capturar a simetria interna de uma estrutura na forma de automorfismos de grupo. Uma simetria interna está normalmente associada com alguma propriedade invariante, e o conjunto de transformações que preserva este invariante, juntamente com a operação de composição de transformações, forma um grupo chamado um grupo de simetria.

A teoria de Galois [32], que é a origem histórica do conceito de grupo, procura descrever as simetrias das equações satisfeitas pelas soluções de uma equação polinomial. Os grupos solúveis são assim chamados devido ao papel proeminente que possuem nesta teoria.

Grupos abelianos estão presentes em várias estruturas estudadas em álgebra abstrata, como anéis, corpos, e módulos.

Na topologia algébrica, grupos são usados para descrever os invariantes de espaços topológicos. Eles são chamados de "invariantes" porque não mudam se o espaço é submetido a uma transformação. Exemplos incluem o grupo fundamental, grupo de homologias e o grupo de cohomologias.

O conceito de grupo de Lie [33] (em homenagem ao matemático Sophus Lie) é importante no estudo de equações diferenciais em variedades; ele combina análise e teoria de grupos e é portanto a ferramenta certa para descrever as simetrias das estruturas analíticas. Ánalise neste e outros grupos é chamada de análise harmônica.

Na análise combinatória, a noção de grupo de permutação e o conceito de ação de um grupo são frequentemente utilizados para simplificar a contagem de um conjunto de objetos.

A compreensão da teoria de grupos é fundamental na Física, onde é utilizada para descrever as simetrias que as leis da Física devem obedecer. O interesse da Física na representação de grupos é grande, especialmente em grupos de Lie [33], pois suas representações podem apontar o caminho para "possíveis" teorias físicas. Em Química, grupos são utilizados para classificar estruturas cristalinas e a simetrias das moléculas.

Em matemática, teoria de grupo computacional é o estudo de grupos por meios de computadores. Preocupa-se com a análise e esboço de algoritmos e estruturas de dados para

computar informações sobre grupos. O tópico tem atraído interesse pois, para muitos grupos de interesse (incluindo muitos dos grupos esporádicos), é impraticável fazer cálculos manualmente.

3.4.1. Ataque Bizantino

Neste ataque, geralmente ligado a problemas de tolerância a falhas, um ou mais nós maliciosos trabalham em conluio para gerar problemas como loops de roteamento, pacotes de roteamento falsos, escolha de caminhos não-ótimos, entre outros, utilizando mensagens de controle dos protocolos que estão sendo utilizados. Murthy cita esse problema com o nome de mensagens de roteamento alteradas, restringindo-o apenas aos problemas de roteamento. Além disso, os nós também podem executar um encaminhamento seletivo. Esse tipo de ataque é de difícil detecção, pois para os nós comuns, o funcionamento estará correto, embora, de fato, esteja apresentando anomalias.

O nome ataque bizantino tem uma origem curiosa. A idéia é baseada no problema dos generais bizantinos [34], distribuídos em campo com suas tropas para organizar o ataque à cidade inimiga. A comunicação entre eles é feita apenas por mensagens e isso deve ser suficiente para organizar o ataque. No entanto, um ou mais generais podem ser traidores tentando confundir os demais, o que gera a necessidade de um algoritmo capaz de garantir que os generais leais conseguirão chegar a um acordo. Tem-se como objetivos que todos os generais leais devem decidir pelo mesmo plano de ação e um pequeno número de generais maliciosos não deve levar os generais leais a adotar um plano ruim. Para satisfazer estes dois objetivos, é necessário que todos os generais leais recebam a mesma informação, e se um general é leal, então sua informação deve ser utilizada por todos os generais leais. Como ambas as condições levam ao mesmo ponto sobre como um general envia sua ordem, é possível simplificar o problema a um general e dois tenentes que devem receber a sua ordem. No caso de apenas um general e um tenente, a solução é trivial, pois a comunicação é direta e não existem mais versões sobre o que foi dito. Assim, o problema acontece a partir de três generais. Por simplificação, a mensagem só pode ser de atacar ou bater em retirada. O problema pode ser caracterizado como na Figura 3.3, onde há um general e dois tenentes. No primeiro caso, temos um tenente traidor, e o tenente leal receberá duas mensagens, uma do general mandando atacar e uma do traidor dizendo que as ordens do general são de bater em retirada. No segundo caso, mostrado na Figura 3.4, o general é traidor, e o mesmo grupo de

informações chega ao tenente leal, de forma que, com três entidades, sendo uma traidora, não há solução. Generalizando o problema, se cada entidade representasse m generais/tenentes, ainda assim, não seria possível resolver o problema, pois seria necessário mais algum testemunho para concluir qual é a informação que deve ser usada. Isto leva a regra de que são necessários pelo menos 3m+1 generais, sendo m o número de generais mentirosos, para que os generais leais cheguem a uma solução única e verdadeira. A solução matemática para o problema é descrita em [35].

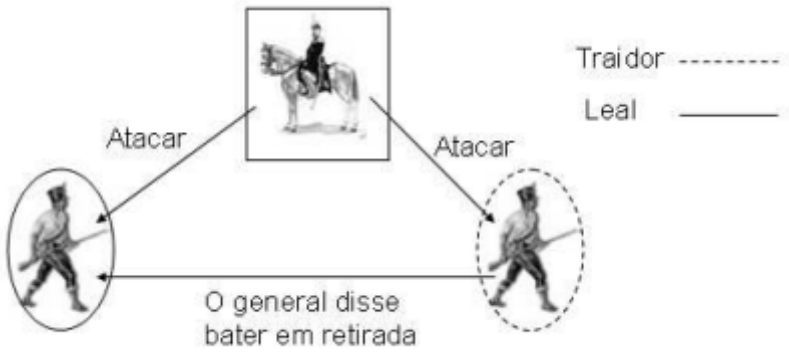

Fig 3.3 – Ataque Bizantino onde um tenente é o traidor [2]

As soluções para o ataque bizantino podem ser a assinatura digital, o uso de múltiplos caminhos e ainda a autorização, mecanismos que serão descritos na Seção 2.4.

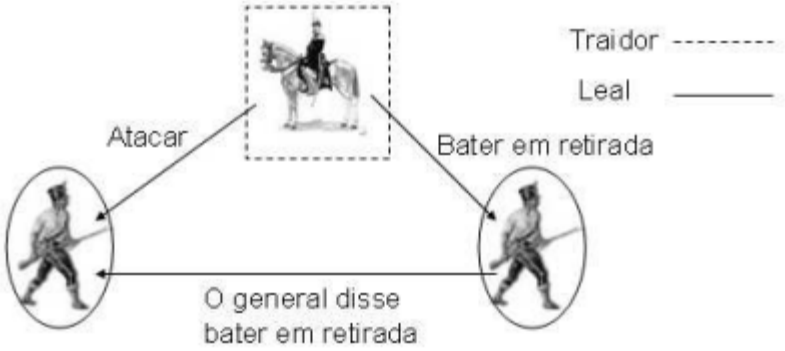

Fig 3.4 – Ataque Bizantino onde o general é traidor [2]

4. IMPLEMENTAÇÃO E RESULTADOS EXPERIMENTAIS

A comprovação prática da tese apresentada neste trabalho consistiu em desenvolver um ambiente de grupos de agentes de software em que cada grupo possuísse uma especialidade de atuação e que pudessem fazer uso da confiança computacional de acordo com as necessidades apresentadas. O ambiente foi desenvolvido de forma que pudesse ser aplicado em um *GRID* computacional. A execução dos testes foi feita com a inicialização de cada agente sendo feita em cima de uma plataforma JADE previamente inicializada. Os testes foram feitos em apenas uma máquina, pois o objetivo era somente analisar a aplicação da confiança computacional em grupos de agentes de software.

O hardware utilizado na avaliação dos testes foi um processador Intel® Core™2 DUO CPU E6750 de 2,66GHz e 4GB de RAM. O sistema operacional utilizado foi o Microsoft Windows 7 x64. Foi também simulado em um sistema com o Windows Vista x86, porém não foi feito o teste em um SO não proprietário da Microsoft, mas foi verificado empiricamente que a aplicação pôde funcionar corretamente em sistemas Linux e MacOS. Isso comprova uma das maiores vantagens de aplicações feitas em Java, a grande interoperabilidade entre sistemas.

4.1. IMPLEMENTAÇÃO

Visando avaliar a eficiência do TRAVOS para identificar grupos completos desonestos ou mesmo agentes isolados, foi desenvolvido um software que simula grupos de agentes dentro de um GRID computacional. Cada grupo possui um líder que coordena todas as ações praticadas, mas todos os agentes são capazes de utilizar do modelo de confiança computacional para tratar suas interações dentro do ambiente. É importante ressaltar que a implementação feita aqui não se adequa ao conceito correto de GRID, pois todos os agentes se registram em um serviço de diretório 1 centralizado. No entanto, essa é uma das características da plataforma na qual o ambiente foi desenvolvido, a JADE, um middleware já

[1]"Um serviço de diretório é um software que armazena e organiza informações sobre os recursos e os usuários de uma rede de computadores, e que permite os administradores de rede gerenciar o acesso de usuários e sistemas a esses recursos."

explicado na seção 3.3. Como o objetivo maior do trabalho não é abordar o conceito de GRID computacional, esse empecilho não foi abordado.

A seguir será dada uma breve explicação das capacidades que cada agente diferenciando escravos e líderes:

Escravos:

- São capazes de se registrar no DF (*Directory Facilitator*) como provedores de determinado serviço ao serem criados;
- Podem encontrar outros agentes capazes de prover serviços, desde que estejam dentro do mesmo grupo;
- Podem resolver desafios gerados por outros agentes, assim como tarefas características do grupo ao qual fazem parte;
- São capazes de requisitar opiniões de outros agentes sobre um determinado membro, desde que todos pertençam ao mesmo grupo; assim como responder pedidos de opiniões advindos de outros agentes do grupo;
- Podem ser eleitos líderes.

Líderes:

- São capazes de fazer o mesmo que os escravos;
- Controlam todas as atividades (tarefas) executadas dentro do grupo;
- Enviam em recebem tarefas enviadas por líderes de outros grupos;
- Podem pedir opiniões a líderes de outros grupos sobre determinado grupo com o qual pretendam interagir.

A distribuição de tarefas por um grupo a outro é mostrada no fluxograma da figura 4.1 quando esse possui um grupo de tarefas a serem processadas.

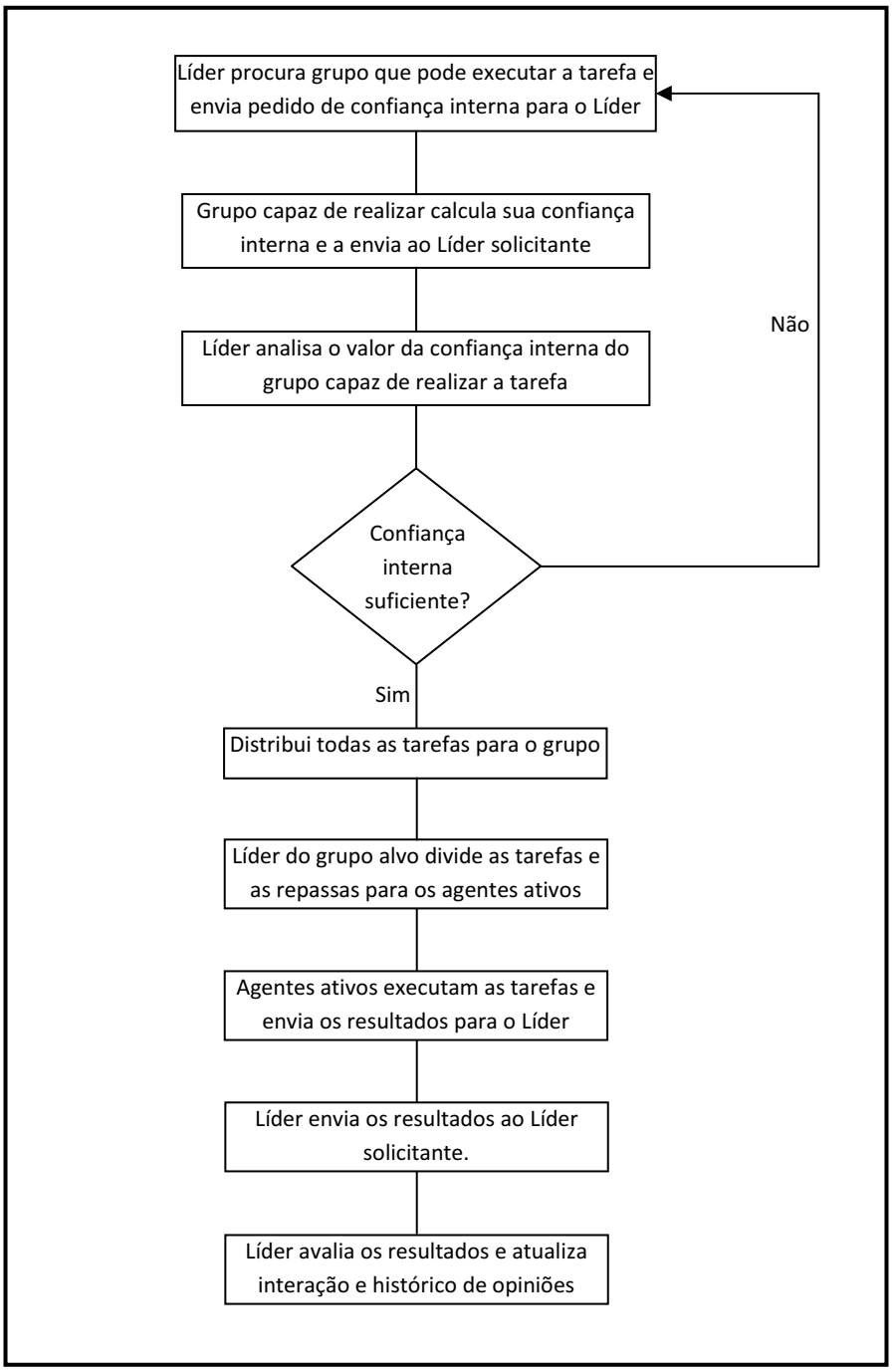

Fig 4.1 - Fluxograma da Distribuição de Tarefas Entre Grupos

Já o fluxograma da figura 4.2 demonstra como é calculada a confiança interna do grupo.

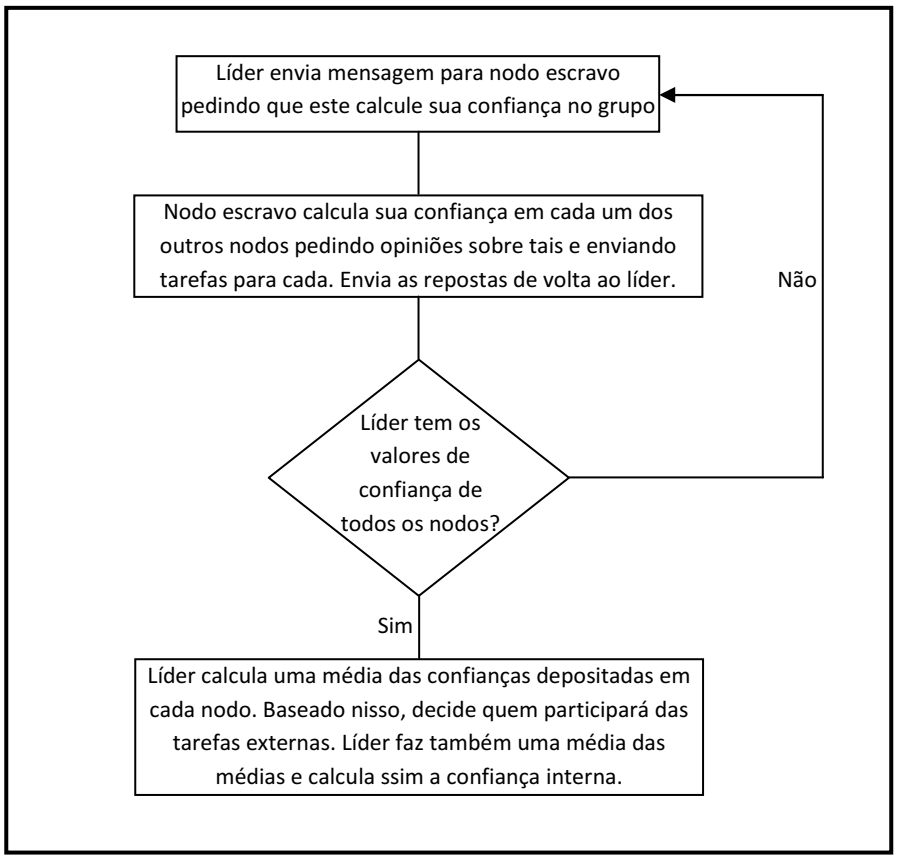

Fig 4.2 - Fluxograma do cálculo da Confiança Interna

Todos os GRIDS em questão tiveram uma escolha da liderança feita de forma aleatória pelo fato de o processo de eleição do líder ser um processo muito complexo e demandar uma estrutura que possibilite sua implementação. Essa questão poderá ser abordada em trabalhos futuros, pois os processos se aproximam do real quando líder é eleito, pois em um sistema aberto e móvel a rotatividade dos agentes é muito grande e consequentemente o líder pode e, às vezes, o líder tem que mudar. No processo de eleição será escolhido como líder aquele que tem o melhor comportamento segundo uma eleição feita com todos os membros do GRID, porém como já dito devido ao dinamismo dos grupo o processo de escolha de líderes deve ser constante, pois um líder, que será interface para várias interações externas, pode não mais estar presente em certo momento ou mesmo alterar seu comportamento e comprometer a segurança de todo o grupo.

É importante ressaltar que os grupos analisados em nosso trabalho, apesar da escolha do líder ser considerada aleatória, os líderes são sempre honestos, ou seja, os líderes são atribuídos de forma mecânica para atuarem como tal, mas antes é verificado se esse é honesto

ou não para que a função a ele seja atribuída. Os parâmetros para atribuição de honestidade são mensurados por meio de três parâmetros anteriormente estabelecidos e a negação em qualquer um desses parâmetros torna o agente desonesto.

Quando fizemos a análise do comportamento do grupo, ao considerarmos um agente desonesto todos os parâmetros de honestidades foram considerados inválidos, a volta vale para os agentes honestos, ou seja, para que um agente seja honesto todos os parâmetros terão que ser, obrigatoriamente, válidos.

Outra questão importante a ser abordada foi a de que caso não haja uma interação anterior com um determinado agente a ele será atribuída uma confiança de valor 0,5, o que é uma medida severamente segura, pois uma confiança de 0,5 não carrega informação nenhuma sobre a confiança em determinado agente haja vista que ele se torna imprevisível com esse valor de confiança igual ao citado. O mesmo foi considerado para grupos que não tinham interagido com um segundo grupo, ao ser questionado sobre outro grupo, na inexistência de interações um valor de confiança 0,5 lhe será atribuído.

Para os grupos analisados foi considerado que para valores de gama maiores que 0,95 apenas a confiança interna será considerada, diferente do comportamento no inicio do processo, pois eram consideradas a confiança interna e a reputação. Um fator que também não pode ser ignorado é o de que quando a confiança de determinado agente atinge 0,4; tarefas passam a não ser mais designadas para tal agente, o que gera um menor comprometimento das estatísticas do grupo e evita maiores problemas na solução das tarefas solicitadas.

4.2. RESULTADOS EXPERIMENTAIS

Para verificarmos a eficácia do método proposto, simulamos com grupos de cinco, dez e quinze agentes, aplicando 20% e 50% de agentes desonestos, e analisando o resultado quanto ao valor do ρ, que se refere à probabilidade de acurácia para o grupo, dos resultados obtidos, pela quantidade de interações entre os agentes, considerando os valores de confiança e o cálculo da confiança em grupos. A escolha de no mínimo 5 agentes por grupo, se deu devida à teoria dos generais bizantinos [2], ou seja, teríamos para o grupo de agentes, uma solução verdadeira e única para 20% dos agentes desonestos no máximo. Para 10 agentes no

máximo para 30% e para 15 agentes no máximo para 26,67%. Caso tivéssemos escolhido 16 agentes teríamos uma solução única e verdadeira para no máximo 31,25% de agentes desonestos. Para evitar quaisquer problemas, entre uma análise e outra o ambiente foi reinicializado para evitar que hajam relações entre uma simulação e outra. Os testes foram feitos de modo a se comprovar a teoria dos generais bizantinos.

Em nosso trabalho os agentes honestos e desonestos são configurados manualmente, no código do programa, de modo que o líder já é ajustado para 100% honesto, e os agentes desonestos para 100% desonestos.

Todos os gráficos são da Acurácia da Confiança Interna pelo número de interações.

4.2.1. Cálculo da Confiança

Esta seção do trabalho dedica-se análise e cálculo da confiança por meio de simulação para grupos de agentes sendo que a porcentagem de agentes honestos dentro grupo pode variar. Foi estabelecido também que o comportamento dos agentes não muda com o tempo, ou seja, se um agente é honesto ele continuará se comportando dessa forma até o fim do experimento e o mesmo ocorre quando agente for desonesto.

Para o nosso caso realizamos simulações para grupos com 100% dos agentes honestos e 80% dos agentes honestos, ambas para grupos com cinco agentes. Primeiramente fez-se uma análise para caso individual e em seguida para o caso onde se analisa apenas o grupo com um todo.

4.2.1.1. Simulação com 100% dos agentes honestos

Os resultados obtidos nessa simulação servirão como parâmetro para as demais, pois este caso trata-se de uma rede operando em modo ideal haja vista que todos os agentes são confiáveis. Como podemos notar na Tabela 4.1 os valores de confiança direta aumentam com o passar do tempo, o que já era esperado visto que estamos trabalhando com um grupo onde todos os agentes são confiáveis.

100% Confiável - Direta	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,75	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 2	0	0,667	0,75	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 3	0	0,667	0,75	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 4	0	0,667	0,75	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 5	0	0,667	0,75	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918

Tabela 4.1 – Tabela da confiança direta de cada agente para um grupo de 5 agentes com 100% dos agentes honestos

Como pode ser visto os valores são iguais para todos os agentes, pois todos os agentes foram inicializados ao mesmo tempo e têm o mesmo comportamento por serem todo considerados confiáveis.

100% Confiável - Comb	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,750	0,830	0,904	0,951	0,975	0,889	0,900	0,909	0,918
Agente 2	0	0,667	0,750	0,830	0,904	0,951	0,975	0,889	0,900	0,909	0,918
Agente 3	0	0,667	0,750	0,830	0,904	0,951	0,975	0,889	0,900	0,909	0,918
Agente 4	0	0,667	0,750	0,830	0,904	0,951	0,975	0,889	0,900	0,909	0,918
Agente 5	0	0,667	0,750	0,830	0,904	0,951	0,975	0,889	0,900	0,909	0,918

Tabela 4.2 – Tabela da confiança combinada de cada agente para um grupo de 5 agentes com 100% dos agentes honestos

Na tabela 4.2 temos os dados para o cálculo da confiança combinada. Nota-se que a partir da sétima interação observamos uma queda nos valores de confiança que se dá pelo fato de que a partir desse momento consideramos apenas a confiança direta, pois o valor de Gama atingiu o limiar e previamente especificado (0.95), onde a confiança direta é única que é considerada. Há discrepância entre valores de confiança direta e combinada devido ao fato de que as opiniões solicitadas impactam positivamente no resultado final.

4.2.1.2. Simulação com 80% dos agentes honestos

Nesta simulação considerou-se um Grupo de 5 agentes sendo um deles é considerado um elemento desonesto. Os resultados da confiança direta e combinada são apresentados nas tabelas 4.3 e 4.4.

Primeiramente analisaremos a confiança direta para cada em separado, como pode ser visto na tabela 4.3 os agentes honestos presentes no grupo já podem ser considerados honestos a partir da terceira interação, onde o valor da confiança atingir o valor condicional (0.8) que serve como parâmetro que possa considerar um agente como sendo honesto.

80% Confiável - Direta	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,750	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 2	0	0,667	0,750	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 3	0	0,667	0,750	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 4	0	0,667	0,750	0,800	0,833	0,857	0,875	0,889	0,900	0,909	0,918
Agente 5	0	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500

Tabela 4.3 – Tabela da confiança direta de cada agente para um grupo de 5 agentes com 80% dos agentes honestos

Porém o mesmo não pode se dizer para o caso do Agente 5, que é o agente desonesto. No caso atribuímos o valor de 0,5 mesmo quando a confiança no agente é menor que tal valor. De qualquer forma o agente não foi considerado honesto em nenhum momento, pois não atingiu o limiar (0.8) que tornaria tal condição válida. É plausível essa atribuição, pois com um valor de confiança 0.5 torna-se imprevisível o comportamento do agente ao qual foi atribuído tal valor.

Nota-se novamente que os valores se repetem para os agentes de 1 a 4, isso ocorre pois os agentes foram inicializados ao mesmo tempo e possuem comportamento semelhante perante os demais agentes.

Para o caso da confiança combinada observamos na tabela 4.4, semelhante à análise anterior, que os valores da confiança combinada sobem mais rapidamente que os valores de confiança que direta devido ao fato da influência das opiniões que pesam positivamente nos resultados.

80% Confiável - Comb	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,75	0,83	0,904	0,951	0,975	0,889	0,9	0,909	0,918
Agente 2	0	0,667	0,75	0,83	0,904	0,951	0,975	0,889	0,9	0,909	0,918
Agente 3	0	0,667	0,75	0,83	0,904	0,951	0,975	0,889	0,9	0,909	0,918
Agente 4	0	0,667	0,75	0,83	0,904	0,951	0,975	0,889	0,9	0,909	0,918
Agente 5	0	0,667	0,574	0,54	0,521	0,5440	0,5437	0,500	0,500	0,500	0,500

Tabela 4.4 – Tabela da confiança combinada de cada agente para um grupo de 5 agentes com 80% dos agentes honestos

Observa-se também que há uma queda nos valores de confiança na interação de número 7 devido ao fato de nesse momento o limiar anteriormente citado atinge seu valor condicional (0.95), o que faz com que a partir desse momento seja considerada apenas a confiança direta.

Nota-se também que para agente desonesto em nenhum momento é considerado honesto haja vista que o valor de sua confiança não ultrapassa o valor estipulado para um certo agente seja considerado desonesto (0.8). É possível percebermos que o valor da confiança para o agente desonesto não se distancia muito de 0.5 devido à influência dos valores de reputação passados serem sempre iguais à 0.5 mesmo quando esses são inferiores a tal valor.

4.2.1.3. Análise dos resultados para o grupo como um todo

Para este caso consideraremos a confiança em grupo baseada em resultados baseados na confiança de cada agente obtida separadamente. É importante frisar que neste caso é importante a figura de um líder que para esta simulação é sempre o Agente 1 de cada grupo. Essa decisão foi tomada aleatoriamente, pois não foi implementado um sistema de escolha de líder baseado em seu comportamento haja vista a complexidade matemática e de implementação.

As simulações foram feitas para grupos com 5, 10 e 15 agentes onde o número de agentes honestos dentro grupo varia para que se tenha noção do impacto dessas mudanças em diferentes ambientes e notar, caso haja, a influência do número de agentes nos cálculos.

4.2.1.3.1. Grupos com 5 Agentes

Para começarmos nossa análise de simulação para grupo escolhemos um grupo com 5 agentes inicialmente, pois consideramos 5 um razoável que já pode nos fornecer algumas conclusões haja vista que foge da teoria Generais Bizantinos, que considera inviável análise para grupos com menos de 4 agentes.

A análise foi feita considerando 80% do grupo confiável inicialmente e em seguida considerando apenas 40% do grupo confiável. Um agente foi considerado confiável quando sua confinça ultrapassou 0,8. Outra consideração importante é a de que quando γ atinge 0,95 a confiança direta se torna suficiente para o cálculo da confiança total.

Sendo assim percebe-se na tabela 4.5, nos resultados para um grupo com 80% dos agentes confiáveis, que a influência do agente desonesto é limitada e não impede, que em número pequeno de interações, os nodos confiáveis se tornem de fato confiáveis.

Nota-se também que o agente desonesto em nenhum momento se torna confiável impactando assim no resultado final da confiança do grupo, ou seja, pelo fato da confiança do agente desonesto não ultrapassar certos limites a confiança do grupo como um todo é afetada.

Grupo 5 Agentes 80%	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,728	0,783	0,828	0,865	0,891	0,91	0,924	0,938	0,948
Agente 2	0	0,667	0,728	0,781	0,824	0,862	0,89	0,91	0,924	0,936	0,949
Agente 3	0	0,667	0,728	0,783	0,828	0,865	0,891	0,91	0,924	0,935	0,947
Agente 4	0	0,667	0,728	0,783	0,825	0,856	0,88	0,899	0,914	0,928	0,942
Agente 5	0	0,667	0,574	0,518	0,517	0,508	0,484	0,444	0,411	0,369	0,331
Grupo	0	0,5	0,59	0,624	0,765	0,79	0,807	0,815	0,82	0,934	0,946

Tabela 4.5 – Tabela da confiança de cada agente para um grupo de 5 agentes com 80% dos agentes honestos

É possível notar também o momento em que apenas a confiança interna é considerada, ou seja, γ atinge 0,95. Na tabela é marcado com a cor vermelha esse momento. O agente marcedo de verde é o agente considerado desonesto na simulação, o procedimento se repetirá para as simulações seguintes.

No caso onde consideramos 60% do grupo honesto a influência dos agentes desonestos foi bastante significativa e fez com que o grupo em nenhum momento se tornasse confiável, o que quer dizer que em nenhum momento, dentro do intervalo observado, a confiança do grupo ultrapassou 0,8, apesar de que os todos os agentes confiáveis foram considerados confiáveis ao final da análise. O valor baixo da confiança para os agentes desonestos que impactou para que a confiança do grupo como um todo não subisse muito, como podemos ver na tabela 4.6.

Grupo 5 Agentes 40%	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,713	0,726	0,745	0,764	0,783	0,802	0,819	0,834	0,849
Agente 2	0	0,667	0,714	0,735	0,755	0,773	0,78	0,796	0,814	0,83	0,842
Agente 3	0	0,667	0,713	0,726	0,745	0,763	0,78	0,796	0,814	0,83	0,845
Agente 4	0	0,667	0,572	0,536	0,535	0,516	0,508	0,474	0,45	0,429	0,407
Agente 5	0	0,667	0,572	0,536	0,534	0,532	0,53	0,529	0,527	0,525	0,522
Grupo	0	0,5	0,525	0,534	0,539	0,67	0,677	0,679	0,685	0,689	0,69

Tabela 4.6 – Tabela da confiança de cada agente para um grupo de 5 agentes com 40% dos agentes honestos

Pode-se notar, pela região marcada em vermelho, onde apenas a confiança direta é considerada. O valor da uma subida mais brusca pelo fato de a confiança final do grupo ser

influenciada pela opinião de um grupo externo que nunca tinha interagido o grupo em questão e conseqüentemente atribuiu 0,5 para reputação do grupo solicitado.

4.2.1.3.2. Grupos com 10 agentes

Para termos uma idéia mais próxima do ideal, simulamos grupos com 10 agentes, pois um número maior de agentes gera conclusões mais reais do alastramento do comportamento de agentes sejam eles honestos ou não. Pode-se perceber pelas tabelas 4.7 e 4.8 que o comportamento quando se aumenta o número de agentes se torna menos regular, e leva um tempo para que as opiniões desonestas sejam neutralizadas.

Percebe-se que apesar da realização da simulação com o mesmo percentual de quando considerou-se apenas 5 agentes no grupo e os mesmos 80% de agentes confiáveis, tabela 4.5, nota-se que em se tratando de dois agentes agora ao invés de 1, tivemos um alastramento maior das conseqüências das opiniões desses, o que levou a um tempo maior para estabelecimento de níveis adequados de confiança para aqueles que mereciam tal resultado.

Grupo 10 Agentes 80%	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,682	0,668	0,735	0,791	0,838	0,873	0,894	0,907	0,912
Agente 2	0	0,667	0,677	0,669	0,747	0,793	0,834	0,87	0,893	0,905	0,912
Agente 3	0	0,667	0,678	0,671	0,746	0,795	0,834	0,87	0,893	0,907	0,911
Agente 4	0	0,667	0,646	0,658	0,741	0,802	0,844	0,875	0,895	0,906	0,91
Agente 5	0	0,667	0,651	0,658	0,733	0,788	0,833	0,868	0,892	0,906	0,912
Agente 6	0	0,667	0,655	0,661	0,74	0,793	0,838	0,872	0,893	0,906	0,911
Agente 7	0	0,667	0,646	0,656	0,731	0,787	0,833	0,868	0,893	0,906	0,911
Agente 8	0	0,667	0,646	0,656	0,731	0,787	0,833	0,868	0,893	0,906	0,911
Agente 9	0	0,667	0,518	0,52	0,536	0,478	0,435	0,401	0,383	0,367	0,352
Agente 10	0	0,667	0,519	0,486	0,506	0,455	0,395	0,354	0,33	0,313	0,304
Grupo	0	0,5	0,624	0,63	0,697	0,729	0,793	0,819	0,894	0,906	0,911

Tabela 4.7 – Tabela da confiança direta de cada agente para um grupo de 10 agentes com 80% dos agentes honestos

Apesar da influência negativa desses agentes no comportamento do grupo como um todo, nota-se na tabela 4.7 que um número maior de agentes levou a uma queda mais acentuada dos níveis de confiança para os agentes desonestos até que se chegue ao momento em esses são excluídos das divisões de tarefas, que acontece quando o nível de confiança cai abaixo de 0,4 está destacado na tabela.

O momento marcado de vermelho é transição ocasionada pelo alcance do valor anteriormente especificado de γ (0,95). A partir desse momento apenas a confiança direta é considerada necessária.

Conclui-se também que apesar da presença dos agentes desonestos, o grupo como um todo consegue atingir níveis de confiança para que possa ser considerado como confiável (0,8) em período de tempo relativamente pequeno.

Grupo 10 Agentes 50%	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,707	0,629	0,585	0,588	0,601	0,611	0,619	0,626	0,641
Agente 2	0	0,667	0,674	0,6	0,573	0,6	0,614	0,632	0,641	0,649	0,66
Agente 3	0	0,667	0,678	0,602	0,569	0,587	0,599	0,61	0,621	0,632	0,65
Agente 4	0	0,667	0,674	0,601	0,571	0,585	0,596	0,608	0,618	0,627	0,641
Agente 5	0	0,667	0,674	0,601	0,57	0,585	0,594	0,601	0,609	0,616	0,629
Agente 6	0	0,667	0,557	0,471	0,45	0,449	0,447	0,621	0,446	0,448	0,449
Agente 7	0	0,667	0,503	0,427	0,383	0,382	0,387	0,393	0,397	0,335	0,312
Agente 8	0	0,667	0,471	0,384	0,335	0,327	0,314	0,327	0,33	0,251	0,221
Agente 9	0	0,667	0,557	0,468	0,451	0,459	0,47	0,472	0,47	0,454	0,439
Agente 10	0	0,667	0,557	0,486	0,47	0,479	0,482	0,484	0,481	0,466	0,455
Grupo	0	0,5	0,5	0,543	0,53	0,542	0,55	0,557	0,563	0,565	0,57

Tabela 4.8 – Tabela da confiança direta de cada agente para um grupo de 5 agentes com 100% dos agentes honestos

Diferentemente do que aconteceu quando o percentual de agentes era de 20%, a tabela 4.8 mostra que quando o percentual de agentes desonestos atinge 50% o grupo não pode ser considerado como confiável em nenhum momento da mesma forma que nenhum agente pode ser considerado como sendo confiável, ou seja, mesmo sabendo que 50% dos agentes do grupo eram honestos, os resultados mostram que devido influência de opiniões desonestas provocou a omissão desse fator. É possível notar também o momento em que os agentes desonestos são neutralizados quando sua respectiva confiança atinge níveis menores que 0,4

4.2.1.3.3. Grupo com 15 agentes

Conclusões similares as expostas na seção anterior foram retiradas quando faz-se análise para grupos contendo 10 agentes. Nota-se claramente a influência dos agentes desonestos com decorrer do tempo, porém nota-se reação nos níveis de confiancça mais acentuada que para o caso onde temos apenas 10 agentes.

Na primeira análise quando temos 80% dos agentes honestos, tabela 4.9, nota-se que a presença dos agentes desonestos prejudica a real revelação do comportamento, mas não impede que o grupo em poucas interações se mostre confiável e os próprios agentes também recebam uma designação condizente com suas intenções.

É possível notar o momento em que a apenas a opinião interna é considerada, que quando é quando γ atinge o valor especificado (0,95).

É destacado também o momento onde os agentes desonestos passam a não participar da distribuição de tarefas e a partir de então não oferecem mais riscos ao grupo.

Grupo 15 Agentes 80%	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,705	0,697	0,765	0,832	0,866	0,93	0,977	0,994	0,995
Agente 2	0	0,667	0,684	0,693	0,765	0,832	0,865	0,927	0,973	0,991	0,993
Agente 3	0	0,667	0,702	0,704	0,77	0,836	0,866	0,93	0,976	0,994	0,995
Agente 4	0	0,595	0,631	0,655	0,731	0,798	0,837	0,905	0,96	0,984	0,988
Agente 5	0	0,667	0,628	0,63	0,705	0,753	0,786	0,844	0,895	0,93	0,953
Agente 6	0	0,667	0,615	0,62	0,694	0,743	0,777	0,839	0,894	0,93	0,954
Agente 7	0	0,642	0,664	0,678	0,759	0,818	0,851	0,917	0,968	0,989	0,991
Agente 8	0	0,667	0,609	0,618	0,688	0,735	0,768	0,802	0,827	0,852	0,879
Agente 9	0	0,667	0,607	0,621	0,704	0,765	0,808	0,845	0,87	0,888	0,906
Agente 10	0	0,667	0,607	0,612	0,688	0,737	0,771	0,805	0,829	0,852	0,88
Agente 11	0	0,619	0,662	0,685	0,762	0,819	0,849	0,917	0,968	0,99	0,993
Agente 12	0	0,642	0,618	0,63	0,708	0,757	0,786	0,835	0,883	0,925	0,958
Agente 13	0	0,667	0,509	0,483	0,459	0,404	0,371	0,354	0,377	0,296	0,286
Agente 14	0	0,667	0,502	0,504	0,489	0,438	0,414	0,403	0,376	0,331	0,307
Agente 15	0	0,571	0,449	0,48	0,463	0,423	0,386	0,358	0,34	0,303	0,27
Grupo	0	0,5	0,613	0,621	0,677	0,713	0,788	0,838	0,918	0,943	0,957

Tabela 4.9 – Tabela da confiança de cada agente para um grupo de 15 agentes com 80% dos agentes honestos

No caso onde foi considerado aproximadamente 50% do grupo honesto, tabela 4.10, nota-se claramente a influência dos agentes desonestos no cálculo da confiança final do grupo e da mesma forma que para o grupo contendo apenas 10 agentes nota-se que o grupo nem os agentes obtiveram nível de confiança suficiente para serem considerados confiáveis, porém os níveis de confiança individual e de grupo atingiram valores mais elevados que quando comparados com o grupo contendo 10 agentes.

Grupo 15 Agentes 50%	0	1	2	3	4	5	6	7	8	9	10
Agente 1	0	0,667	0,717	0,587	0,528	0,549	0,568	0,597	0,645	0,7	0,76
Agente 2	0	0,667	0,699	0,566	0,521	0,543	0,56	0,58	0,607	0,638	0,674
Agente 3	0	0,667	0,715	0,583	0,528	0,549	0,567	0,592	0,628	0,671	0,717
Agente 4	0	0,642	0,667	0,576	0,525	0,546	0,561	0,576	0,598	0,625	0,659
Agente 5	0	0,667	0,674	0,559	0,519	0,541	0,556	0,572	0,594	0,622	0,655
Agente 6	0	0,667	0,676	0,562	0,52	0,543	0,559	0,575	0,598	0,626	0,659
Agente 7	0	0,667	0,669	0,574	0,525	0,544	0,558	0,573	0,594	0,619	0,65
Agente 8	0	0,642	0,693	0,569	0,521	0,547	0,566	0,594	0,633	0,678	0,726
Agente 9	0	0,667	0,539	0,455	0,459	0,491	0,502	0,509	0,507	0,489	0,462
Agente 10	0	0,667	0,495	0,423	0,429	0,461	0,476	0,472	0,429	0,354	0,288
Agente 11	0	0,642	0,513	0,436	0,435	0,458	0,458	0,455	0,433	0,377	0,347
Agente 12	0	0,667	0,509	0,412	0,407	0,44	0,456	0,448	0,443	0,414	0,377
Agente 13	0	0,642	0,477	0,396	0,4	0,429	0,434	0,436	0,417	0,318	0,231
Agente 14	0	0,667	0,555	0,462	0,456	0,485	0,499	0,504	0,508	0,497	0,484
Agente 15	0	0,667	0,525	0,459	0,441	0,45	0,447	0,462	0,443	0,408	0,359
Grupo	0	0,5	0,543	0,516	0,481	0,505	0,518	0,53	0,538	0,583	0,645

Tabela 4.10 – Tabela da confiança de cada agente para um grupo de 15 agentes com 50% dos agentes honestos

Pode-se notar também na tabela acima o momento em que γ atinge o valor especificado (0,95). A desse momento será considerada apenas a opinião direta.

4.2.2. Análise feita para Grupos de 5 Agentes

No caso de 100% dos agentes honestos, é simulado um ambiente ideal, sendo que neste caso todas opiniões são verídicas, e todas sub-tarefas são realizadas com sucesso. A figura 4.3 representa o caso ilustrado, para grupos de 5 agentes simulados.

Neste caso percebe-se que quanto maior o número de interações com o sistema, mais a tabela de interações do agente se aproxima à real do sistema no momento, considerando que ρ depende somente do passado de opiniões. A partir de t=3, todos os agentes já realizam tarefas a contento. Podemos ver que a primeira interação têm como resultado zero, já que o agente que requisitou a tarefa ainda não possuía opinião catalogada do outro agente, fazendo com que ocorra a primeira opinião passível de verificação, a qual será utilizada para o cálculo do ρ na próxima vez que ele solicitar a opinião do agente em questão.

Fig 4.3 – Gráfico de Acurácia da Confiança Interna com 100% de agentes honestos em um grupo com 5 agentes

Neste caso percebe-se que quanto maior o número de interações com o sistema, mais a tabela de interações do agente se aproxima à real do sistema no momento, considerando que ρ depende somente do passado de opiniões. A partir de t=3, todos os agentes já realizam tarefas a contento. Podemos ver que a primeira interação têm como resultado zero, já que o agente que requisitou a tarefa ainda não possuía opinião catalogada do outro agente, fazendo com que ocorra a primeira opinião passível de verificação, a qual será utilizada para o cálculo do ρ na próxima vez que ele solicitar a opinião do agente em questão.

Nestes casos, o coeficiente tende a se estabilizar em um valor muito próximo de um, e o grupo é considerado bastante confiável. Mesmo não tendo um exemplo retratado, podemos ter a conclusão que, devido ao fator depender do passado de opiniões para se calcular o ρ, quando o agente tem poucas interações seus valores de confiabilidade tendem a decrescer com o tempo.

Agora inserimos um agente desonesto no grupo, e calculamos o valor da acurácia da confiança interna para este caso, utilizando o conceito de delegação de tarefas. De tal modo obtivemos a figura 4.4 ilustrada abaixo.

Neste caso percebe-se que o grupo também tende a 1, já que ele também atende ao caso da teoria dos generais bizantinos [2]. Vale ressaltar que neste caso demora-se mais para atingir a proximidade ao valor 1, comparando-se com o grupo 100% honesto. Tal fato é

melhor observado na figura 4.6, aonde compara-se os casos estudados para o grupo de 5 agentes, e a razão para isso obviamente é a existência de um agente desonesto.

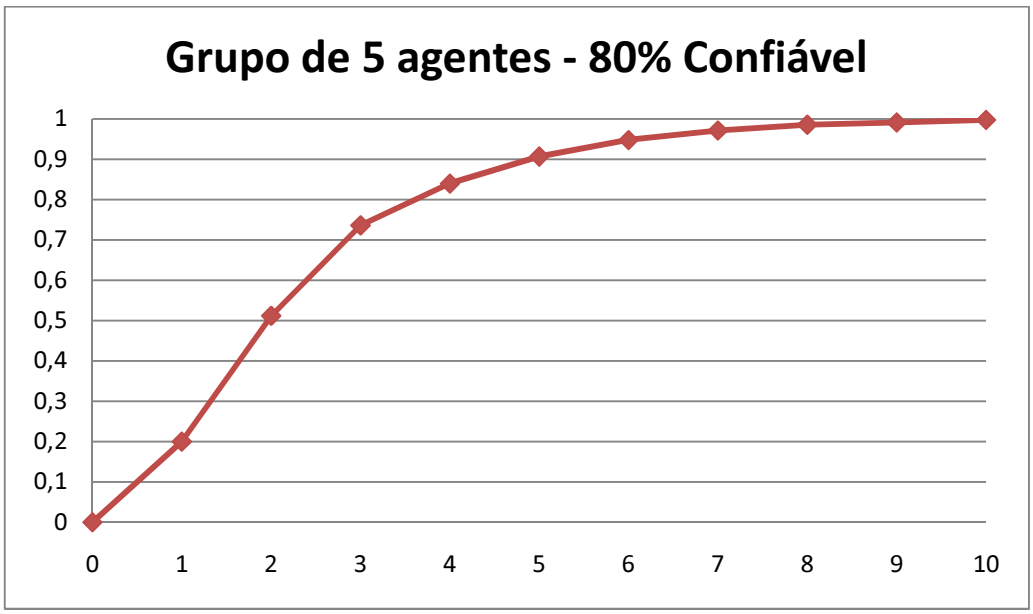

Fig 4.4 – Gráfico de Acurácia da Confiança Interna com 80% de agentes honestos em um grupo com 5 agentes

Neste caso percebe-se que o grupo também tende a 1, já que ele também atende ao caso da teoria dos generais bizantinos [2]. Vale ressaltar que neste caso demora-se mais para atingir a proximidade ao valor 1, comparando-se com o grupo 100% honesto. Tal fato é melhor observado na figura 4.6, aonde compara-se os casos estudados para o grupo de 5 agentes, e a razão para isso obviamente é a existência de um agente desonesto.

Chegamos à conclusão que não era razoável fazer simulações para grupos com mais da metade de agentes desonestos, por tal motivo tentamos sempre chegar o mais próximo do primeiro valor menor que igual a 50% de agentes desonestos, sendo que para grupos de 5 agentes, temos 40% de agentes desonestos, ilustrado na figura 4.5.

Analisando a figura 4.6, verificamos que o início segue um padrão, aonde sem existir um histórico de interações, é atribuído um valor baixo, para início das relações, e incrementá-lo de acordo com o que ocorrer. No nosso caso o valor inicial é na casa de 0,2 e, como é menos de 50% de agentes desonestos, o valor da acurácia ainda cresce porém não até um padrão aceitável.

Apesar de não simulado, é possível perceber que para casos com mais de 50% de agentes desonestos, a tendência é que após o valor atribuído incialmente, a tendência é que o

valor da acurácia nem seja incrementado, e sim decrementado, sendo que quanto maior a porcentagem de agentes desonestos, mais bruscamente a acurácia interna tenderá a zero.

Como a confiança interna reflete como as relações de confiança entre os membros funcionam, a existência de agentes desonestos faz com que a confiança interna tenha a tendência de cair, à medida que mais e mais interações ocorram. Já a confiança referente ao agente desonesto em particular tende a cair até chegar a um valor muito próximo de zero. Caso um grupo deseja que sua confiança não seja afetada pela presença de agentes desonestos, o grupo deverá isolar, remover o agente em questão, para que então a confiança interna volte a subir e atingir patamares desejáveis, isto levando em consideração que a reputação também faz parte do cálculo da confiança do grupo.

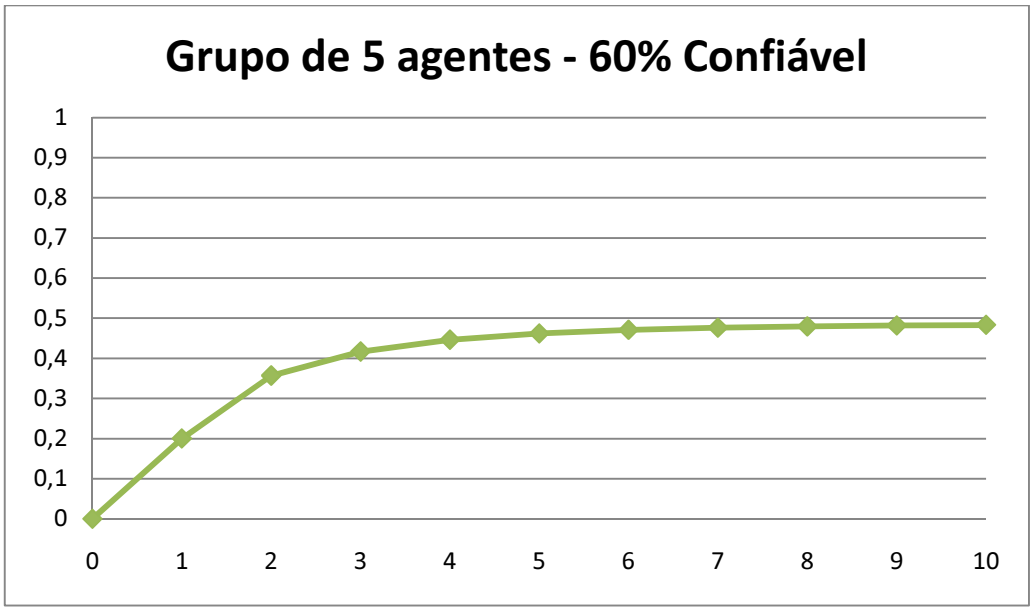

Fig 4.5 – Gráfico de Acurácia da Confiança Interna com 60% de agentes honestos em um grupo com 5 agentes

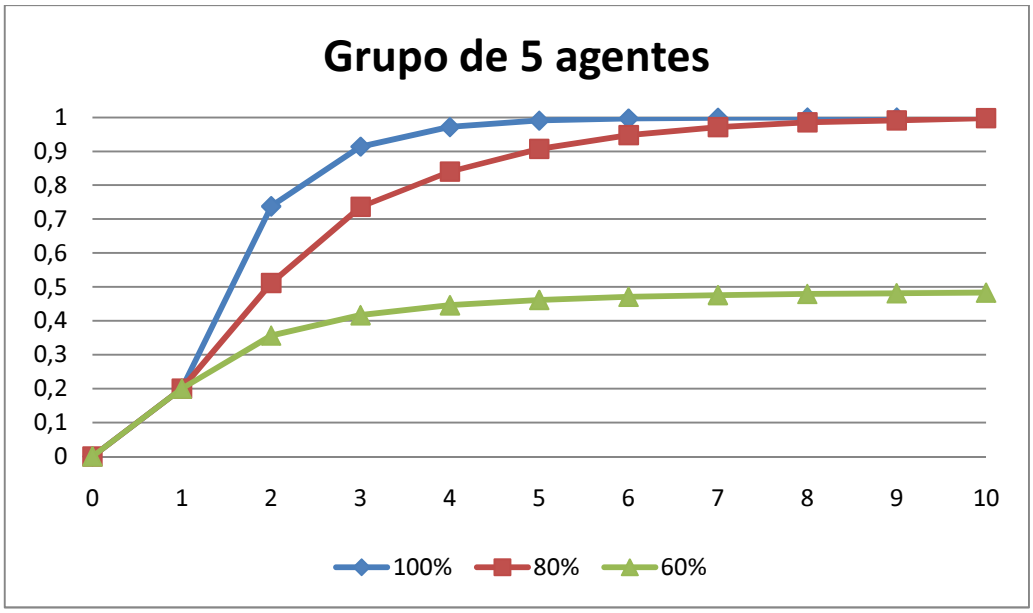

Fig 4.6 – Gráfico de Acurácia da Confiança Interna para diferentes porcentagens de agentes honestos em um grupo com 5 agentes

4.2.3. Análise feita para Grupos de 10 Agentes

Para grupos de 10 agentes, com 100% dos agentes confiáveis, temos o mesmo comportamento para o caso de 5 agentes, tendo como diferença o fato que, tendo mais agentes a tendência a se atingir o valor 1 é mais rápida neste caso. Para 5 agentes atingi-se tal valor próximo da quinta interação, e com 10 agentes na terceira interação tal valor já é consideravelmente próximo a 1.

Fig 4.7 – Gráfico de Acurácia da Confiança Interna com 100% de agentes honestos em um grupo com 5 agentes

Com 80% dos agentes confiáveis, novamente temos o mesmo comportamento da figura 4.7 e com o valor consideravelmente próximo a um, o qual já é atingido na quarta interação, contra a sétima interação para grupos de 5 agentes sob as mesmas condições.

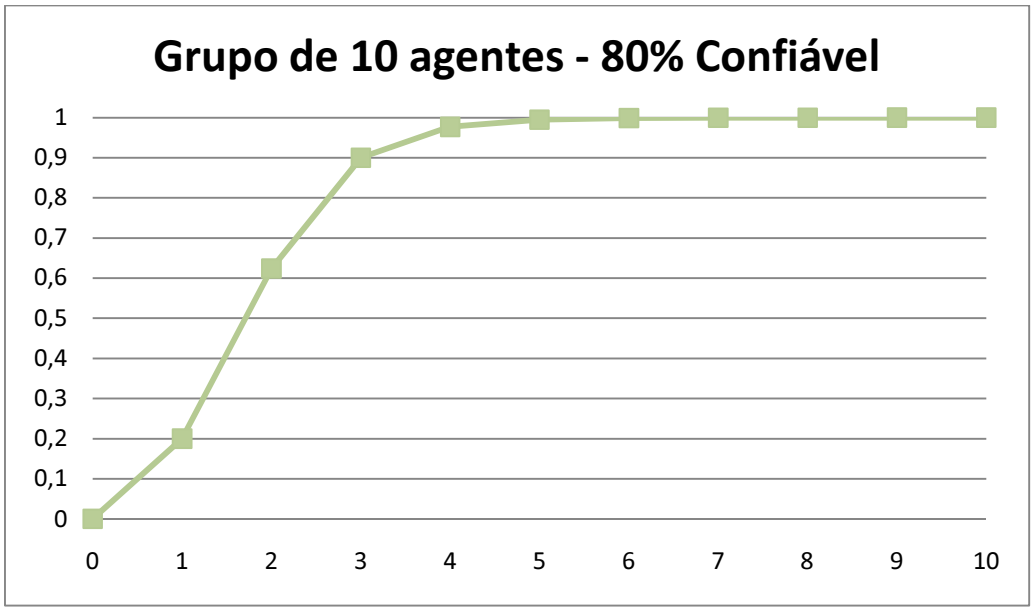

Fig 4.8 – Gráfico de Acurácia da Confiança Interna com 80% de agentes honestos em um grupo com 10 agentes

Para caso em que temos exatamente 50% dos agentes desonestos, de acordo com a figura 4.9, percebemos que após o instante t=3, o valor da acurácia começa a cair, e tende a se aproximar de zero cada vez mais interações são feitas no sistema, como já foi ressaltado anteriormente. A queda mais brusca entre os instantes t=9 e t=10, se deve à mudança no valor de confiança total do grupo, já detalhado na seção 4.1.4.

Já na figura 4.10 é feito um comparativo entre as diferentes porcentagens utilizadas no estudo para grupos de 10 agentes, aonde novamente evidencia-se o fato do grupo com 80% de agentes honestos, mesmo atendendo à teoria dos generais bizantinos [2], demorar mais para atingir a proximidade a 1, do que o 100% honesto, o que é devidamente esperado.

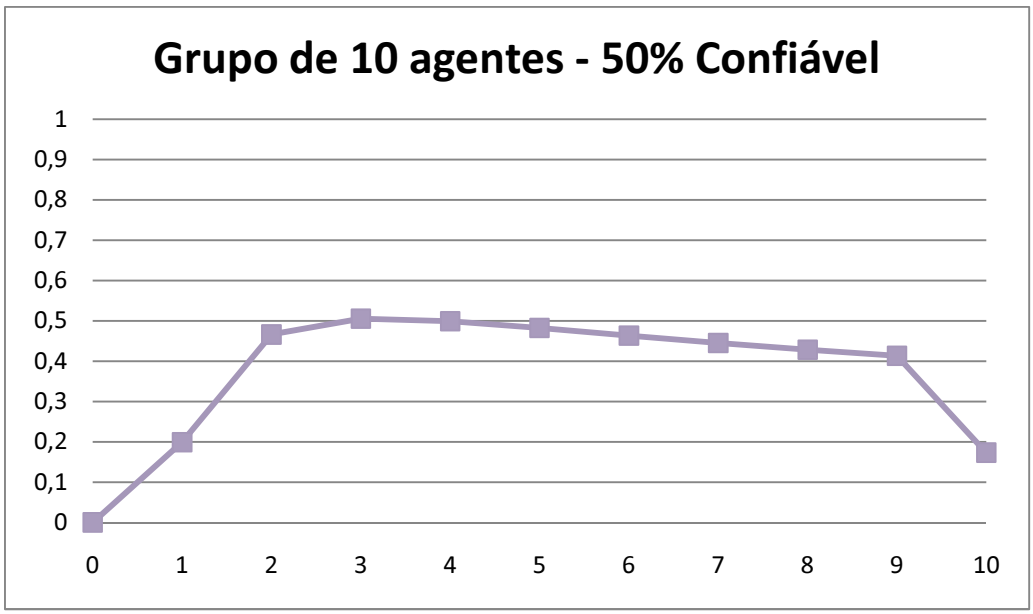

Fig 4.9 – Gráfico de Acurácia da Confiança Interna com 50% de agentes honestos em um grupo com 10 agentes

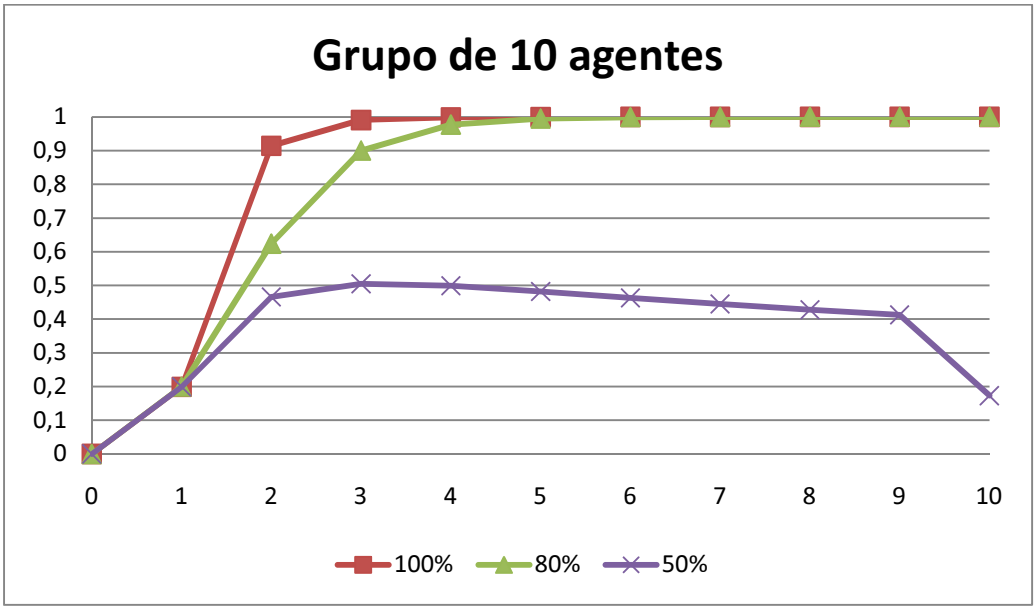

Fig 4.10 – Gráfico de Acurácia da Confiança Interna para diferentes porcentagens de agentes honestos em um grupo com 10 agentes

.

4.2.4. Análise para Grupos de 15 Agentes

Os grupos de 15 agentes se comportam da mesma maneira que os grupos de 10 agentes para os casos de 100% dos agente confiáveis, e 80% dos agentes confiáveis, somente com agora a tendência de se tender para 1 é mais rápida ainda do que para 10 agentes, como observado nas figuras 4.11 e 4.12. Tais fatos serão evidenciados nas figuras 4.15, 4.16 e 4.17, as quais comparam as diferentes porcentagens para diferentes quantidades de agentes nos grupos.

Fig 4.11 – Gráfico de Acurácia da Confiança Interna com 100% de agentes honestos em um grupo com 15 agentes

Como 15 agentes se trata de um número ímpar, é impossível obter 50% exatamente de agentes desonestos, de modo que apesar de continuarmos sinalizando como 50% de agentes, para facilitar futura análise, a real porcentagem utilizada foi de 46,67% de agentes desonestos.

Neste caso tivemos um comportamento curioso, aonde o acurácia cresceu até o momento em que houve mudança no valor de confiança total do grupo (seção 4.2.4), e a partir deste momento ela voltou a crescer. O estudo que pode ser feito é que como ainda temos menos da metade de agentes desonestos o grupo realmente tende a subir porém nunca tenderá a 1, pelo contrário, a partir de um certo ponto ele deve começar a cair e tender a zero, porém após muitas interações. Na figura 4.14 é feito um comparativo novamente para o caso.

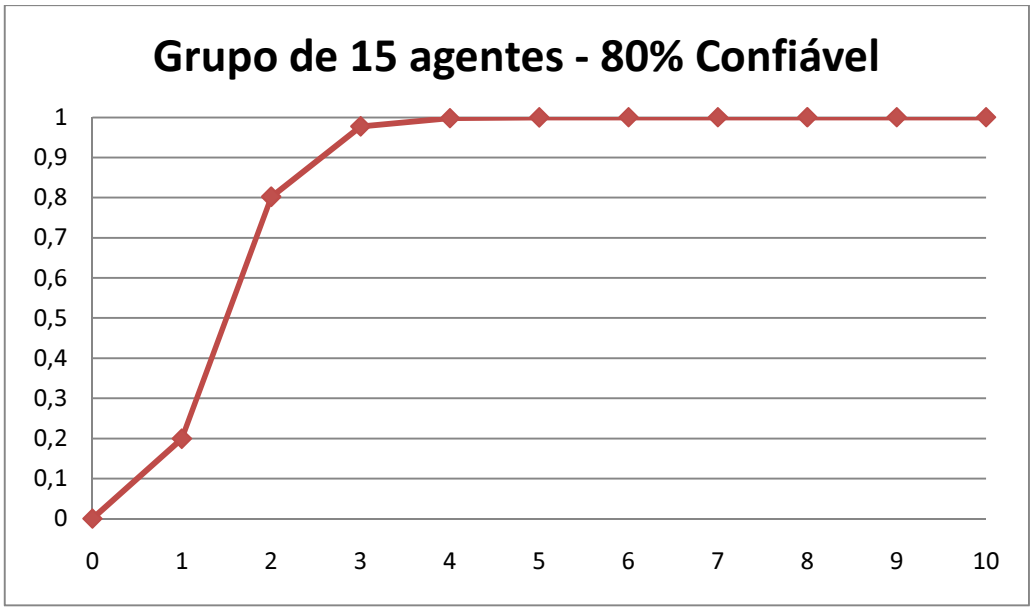

Fig 4.12 – Gráfico de Acurácia da Confiança Interna com 80% de agentes honestos em um grupo com 15 agentes

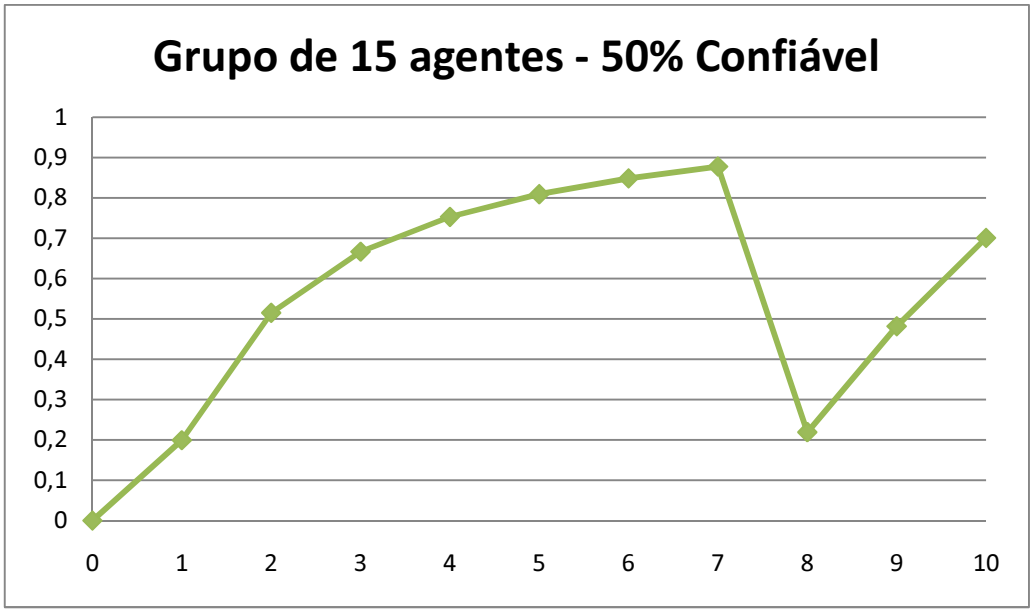

Fig 4.13 – Gráfico de Acurácia da Confiança Interna com 50% de agentes honestos em um grupo com 15 agentes

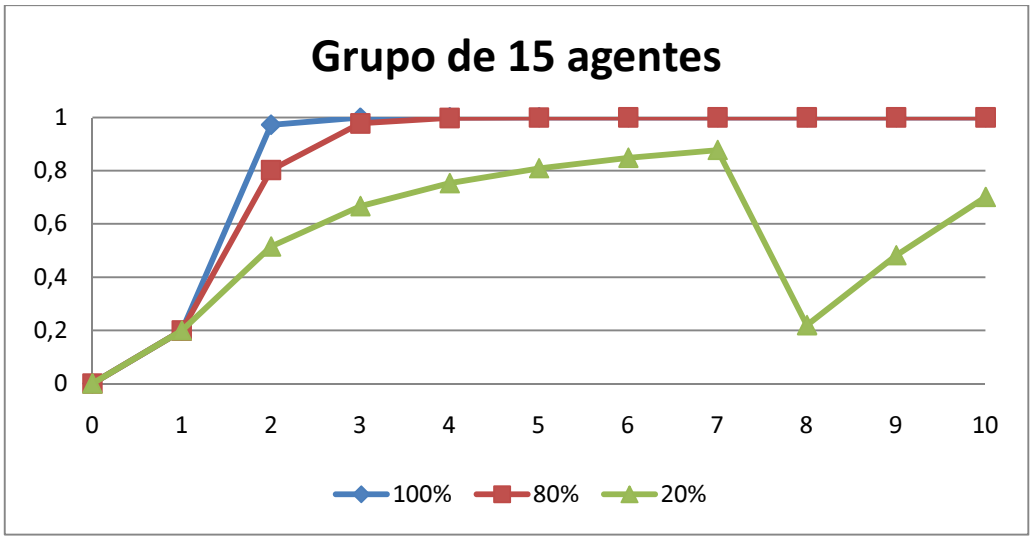

Fig 4.14 – Gráfico de Acurácia da Confiança Interna comparativo para diferentes porcentagens em um grupo com 15 agentes

4.2.5. Comparativo das Análises de Grupos

Com os gráficos 4.15 e 4.16, fica mais evidente os pontos já anteriormente ressaltados, aonde com o aumento do número de agentes mais rapidamente se tende a 1 para o caso de 100% de agentes honestos e para o caso de 80% de agentes honestos. Para a figura 4.17 vale lembrar que no caso de grupos de 5 agentes, foi aplicado na verdade 40% de agentes desonestos, para grupos de 10 agentes foi exatamente 50% de agentes desonestos, e nos grupos de 15 agentes foi aplicado 46,67% de agentes desonestos, o que por si só já é razoável para gerar as discrepâncias observadas e já citadas.

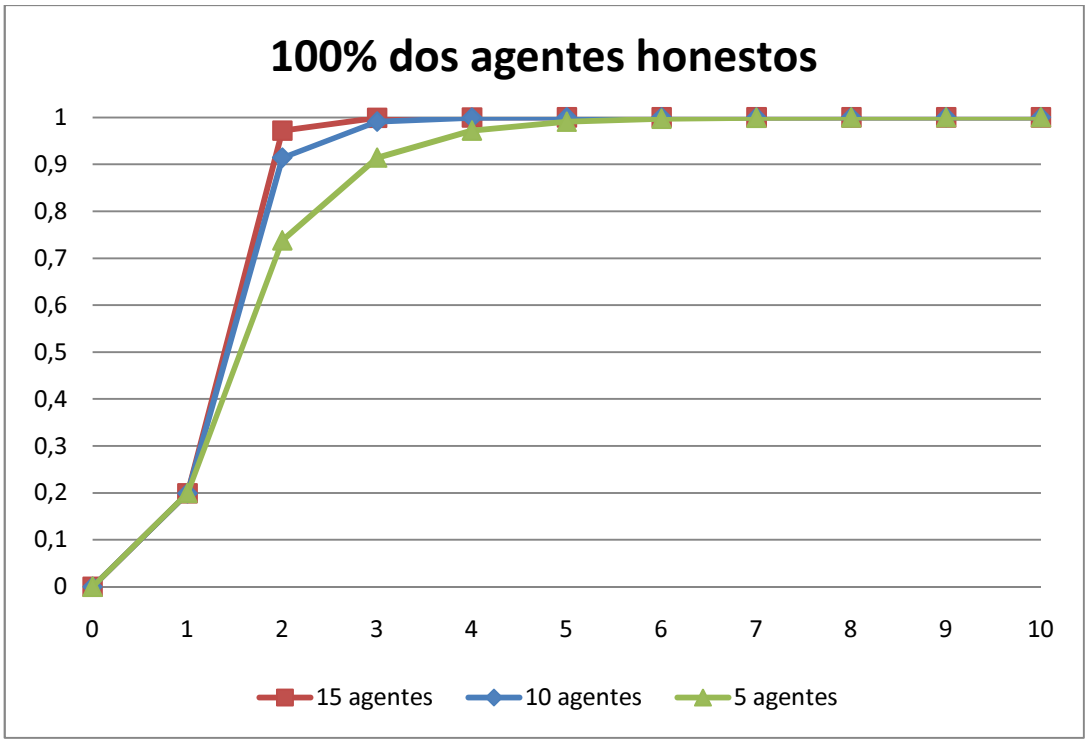

Fig 4.15 – Gráfico de Acurácia da Confiança Interna comparativo para diferentes quantidades de agentes por grupo, sendo 100% honestos

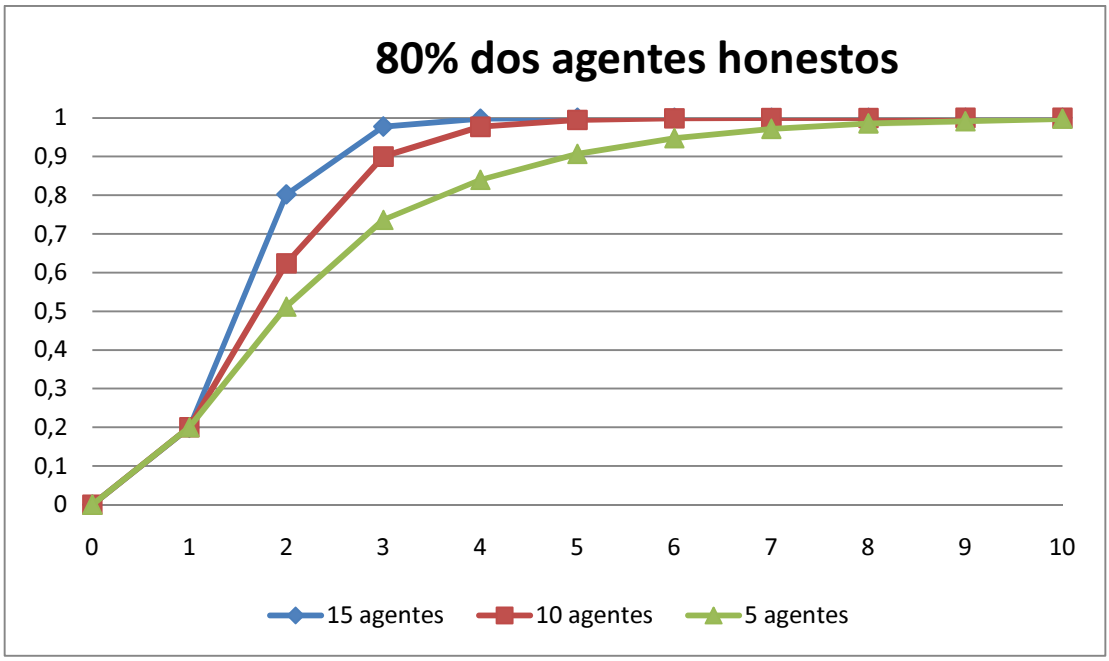

Fig 4.16 – Gráfico de Acurácia da Confiança Interna comparativo para diferentes quantidades de agentes por grupo, sendo 80% honestos

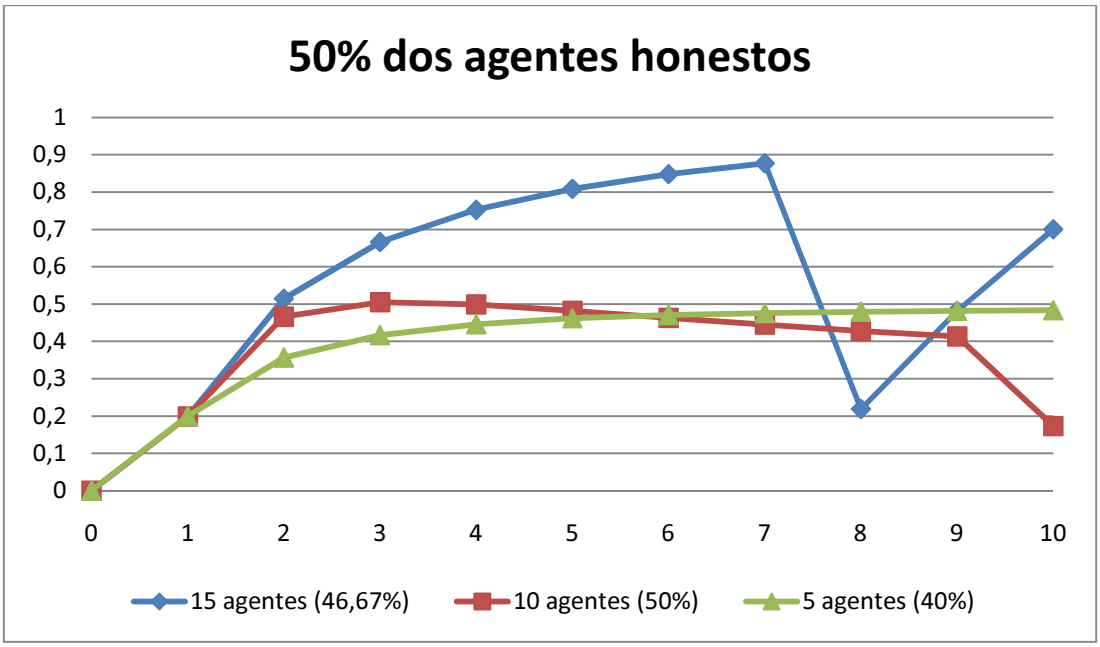

Fig 4.17 – Gráfico de Acurácia da Confiança Interna comparativo para diferentes quantidades de agentes por grupo, sendo 50% honestos

5. CONCLUSÕES

A implementação nos permitiu concluir por meio da análise de vários resultados, sejam eles em forma de gráficos ou tabelas, que se conseguiu representar o comportamento dos agentes por meio de parâmetros como reputação e confiança. Através do sistema implementado pode-se obter informações individuais ou mesmo informações em contexto de vários agentes onde fica claro a influência do comportamento de cada agente nos resultados totais, ou seja, o comportamento em contexto de grupo reflete o comportamento de seus agentes, se há agentes desonestos no grupo, com certeza está situação levará a uma diminuição da confiança total do grupo sendo que se houver uma predominância de agentes desonestos, o grupo poderá também ser considerado desonesto.

Notou-se também a influência de parâmetros auxiliares de decisão como a Certeza que auxilia no processo de atribuição da confiança e se mostra essencial para a obtenção de resultados cada vez mais confiáveis, o que diminui os riscos de se trabalhar em ambientes como o citado. Entra-se assim na importância do modelo na garantia da segurança da informação, atitudes maliciosas servirão como critério de interação e impedirão a participação de agentes que não atendam as especificações, ou seja, agentes que não atingiram os limites mínimos desejáveis para que se possa considerá-lo confiável e assim impossibilita-se interações cujo resultados seriam duvidosos.

É importante reforçarmos a necessidade da imagem do líder honesto para análises de perspectiva de grupo, apesar de que, como citado anteriormente, sua escolha foi feita aleatoriamente haja vista a ausência de um sistema capaz de eleger um líder. Contudo todos os líderes dos grupos simulados tinham comportamento honesto, o que torna ainda mais confiável as informações providas referentes ao comportamento dos demais membros.

Como trabalhos futuros, para aproximar mais o trabalho para a realidade atual, teríamos que, primeiramente, implantar um sistema de decisão de liderança. A idéia inicial é fazer testes, e após esses testes, decidir qual agente foi eleito o mais confiável do grupo. Seria necessário também implantar a idéia de “aging”, obsolescência, aonde, por exemplo, poderíamos considerar somente as últimas 100 interações. Tal atitude faria com que fossem mais facilmente identificados mudanças de atitude, algo que não consideramos em nossa implementação, sendo essa mudança tanto para o bem ou para o mal. No nosso projeto, um

agente desonesto é banido para sempre do grupo, isolado. O ideal seria também implementar um sistema de redenção, para caso esse agente passe a atuar honestamente, que ele possa, talvez com uma triagem, voltar a fazer parte do grupo. Tais idéias juntas já formariam um sistema mais completo e mais próximo da realidade.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Poggi, Agostino; Tomaiulo, Michele; Vitaglione, Giosuè. “A Security Infrastructure for Trust Management in Multi-agent Sistems”. Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems.

[2] Fernandes, Natalia C.; Moreira, Marcelo D. D.; Velloso, Pedro B. “Ataques e Mecanismos de Segurança em Redes Ad Hoc”. Material encontrado em http://www.gta.ufrj.br/ftp/gta/TechReports/FMVC06.pdf . Acessado em 15 de novembro de 2009

[3] Patel, Jigar. “A Trust and Reputation Model for Agente-Based Virtual Organizations”. Thesis of Doctor of Philosophy. Faculty of Engineering and Applied Science. School of Eletronics and Computer Science University of Southampton. January 2007.

[4] Marsh, Stephen Paul. “Formalizing Trust as a Computational Concept”. Department of Computing Science and Mathematics, University of Stirling. Doctorate Thesis. April 1994.

[5] Sabater, Jordi; Sierra, Carles. “Review on Computational Trust and Reputation Models”. Artificial Intelligence Review (2005), Springer 2005.

[6] Larousse. “Grande Enciclopédia Larousse Cultural. Editora Nova Cultura. ISBN 85-13-00755-2. 1998

[7] Gambetta, Diego. ´Can We Trust Trust?´, in Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, eletronic edition, Department of Sociology, University of Oxford, chapter 13, pp. 213-237, 2000.

[8] Lamsal, Pradip.”Understanding Trust and Security”. Department of Computer Science University of Helsiki, Finland, October 2001. Disponível em http://www.cs.helsinki.fi/u/lamsal/asgn/trust/UnderstandingTrustAndSecurity.pdf. Acessado em 17 de novembro de 2009.

[9] Albuquerque, Robson de Oliveira. “Uma proposta de um modelo de confiança computacional para grupos em sistemas distribuídos” Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.[Distrito Federal], 2008.

[10] Gaston, Matthew E.; desJardins, Marie. “Social Networks and Multi-agent Organizational Performance”. Department of Computer Science. University of Maryland Baltimore County.

[11] Sabater, Jordi; Sierra, Carles. “Reputation and social network analysis in multiagent systems.” Proceedings of The First International Joint Conference on Autonomous Agents & Multiagent Systems, ACM, Italy, 2002

[12] Suryanarayana, Girish; Taylor, Richard N. “A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications.” Institute for Software Research, University of California, ISR Technical Report #UCI-ISR-04-6, July 2004.

[13] Audun Jøsang, Ross Hayward, Simon Pope. “Trust Network Analysis with Subjective Logic”. Proceedings of the Australasian Computer Science Conference (ACSC'06), Hobart, January 2006.

[14] Zacharia, G., Maes, P.”Trust Management through Reputation Mechanisms, Applied Artificial Intelligence” 14 (2000) 881-907.

[15] A. Jøsang and R. Ismail.” The Beta Reputation System”. Proceedings of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia, 17-19 June 2002

[16] Castelfranchi, Cristiano; Falcone, Rino. “Social Trust: A Congnitive Approach”. National Research Council – Institute of Psychology.

[17] Ramchurn S. D. “Multi-Agent Negotiation using Trust and Persuasion”. PhD thesis, Electronics and Computer Science, University of Southampton, UK, 2004.
[18] L J Savage, The foundations of statistics (New York-London, 1954).
[19] FIPA. Disponível em http://www.fipa.org/.
[20] JADE. JADE - Java Agent Development Framework. Disponível em http://jade.tilab.com/.
[21] Olso, Doug; Cowles, Robert; Mullen, Shawn; Helm, Mike. “Grid Trust Model for CA Signed Identity Certificates”. Global Grid Forum. Grid Certificate Policy Working Group, July 2000.
[22] Thompson, Mary R.; Olson, Doug; Cowles, Robert; Mullen, Shawn; Helm, Mike. “CA-based Trust Issues for Grid Authentication and Identity Delegation.” Grid Certificate Policy Working Group, June 2003.
[23] Foster, Ian. “What is Grid? A Three Point Checklist”. Argonne National Laboratory & University of Chicago, July 20, 2002.
[24] Papalilo, Elvis; Freisleben, Bernd. “Towards a Flexible Trust Model for Grid Environments”. Lecture Notes in Computer Science. Springer Berlin/Heildelberg Volume 3270, 2004.
[25] Ferreira, Luis. “Introduction to Grid Computing with Globus.” IBM RedBooks, September 2003.
[26] Wooldridge, M; Jennings, N.R. “Intelligent Agents: Theory and Practice” – Knowledge Engineering Review, October 1994.
[27] Genesereth, M. R. and Nilsson, N. (1994). Software agents. Communications of the ACM.
[28] Nwana, Hyacinth S.; “Software Agents: Na Overview”. Intelligent Systems Research AA&T, BT Laboratories, 2000.
[29] G. Weiss (Ed.). “Multiagent systems. A modern approach to distributed artificial intelligence.” The MIT Press. (ISBN 0-262-23203-0). 1999.
[30] Wooldridge, Michael. An Introduction to Multiagent Systems. 2002, John Wiley & Sons, Ltd.
[31] BELLIFEMINE, Fabio Luigi; GIOVANNI, Caire; GREENWOOD, Dominic. Developing Multi-Agent Systems with JADE. 2007, John Wiley & Sons, Ltd.
[32] L Toti Rigatelli, Evariste Galois (1811-1832) (Boston, 1996).
[33] R Hermann (ed.), Lie Groups : History, Frontiers and Applications (Brookline, Mass., 1975).
[34]] Lamport, L., Shostak, R. e Pease, M. (1982). The byzantine generals problem. Em ACM Transactions on Programming Languages and Systems (TOPLAS), volume 4 of 3, páginas 382–401.
[35] Pease, M., Shostak, R. e Lamport, L. (1980). Reaching agreement in the presence of faults. Em Journal of ACM 27, volume 2, páginas 228–234.