UNIVERSIDADE FEDERAL FLUMINENSE - UFF
INSTITUTO DE COMPUTAÇÃO

MARCOS SIMÃO GUIMARÃES

TOLERÂNCIA A FALHAS E ECONOMIA DE RECURSOS EM UM SISTEMA DE MITIGAÇÃO DE SINISTROS

Dissertação entregue ao Instituto de Computação da Universidade Federal Fluminense - UFF, como requisito parcial para a obtenção do título de Mestre.

Área de Concentração: Processamento Paralelo e Distribuído

Orientador: ORLANDO GOMES LOQUES FILHO.

Niterói – RJ

Março / 2006

TOLERÂNCIA A FALHAS E ECONOMIA DE RECURSOS EM UM SISTEMA DE MITIGAÇÃO DE SINISTROS

MARCOS SIMÃO GUIMARÃES

Dissertação entregue ao Instituto de Computação da Universidade Federal Fluminense - UFF, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovada em____ de _____ de 2006.

COMISSÃO EXAMINADORA

_____
Professor Orlando Gomes Loques Filho
Orientador
UFF

_____
Professor Aloysio de Castro Pinto Pedroza
COPPE / UFRJ

_____
Professor Julius Leite
UFF

Aos meus professores, minha família e amigos.

AGRADECIMENTOS

Agradeço a Deus, que me guiou em todo o caminho.

Ao meu orientador, Orlando Loques, pela oportunidade de trabalhar com ele e por seu empenho em me guiar até meu objetivo.

A meus pais, Octavio e Maria Rita, meus irmãos Paula e Marcelo que me apoiaram de forma irrestrita. Agradeço, principalmente, a minha mulher, Lorena Dadalto pela sua compreensão, paciência e tolerância.

Aos meus colegas de mestrado, Paulo Motta, Glauco Freitas, Alexsandro Mattos Corradi, cujas conversas inteligentes, acaloradas discussões e apoio mútuo fizeram mais rica essa oportunidade e, em especial, ao colega Luciano Bertini, cujo trabalho é utilizado nessa dissertação.

Aos meus amigos que entenderam a minha ausência.

Aos anônimos motoristas de ônibus, que me levaram com segurança por quase duas voltas à Terra que tive que percorrer para cursar o mestrado.

E a todos aqueles que, mesmo não citados aqui, contribuíram de alguma forma para esse trabalho pudesse ser feito.

Se você acredita que consegue, ou
Se você acredita que não consegue,
De qualquer modo você está certo.
Henry Ford

Resumo da Tese apresentada à UFF como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação (M.Sc.)

Marcos Simão Guimarães

Março 2006

Orientador: Orlando Gomes Loques Filho
Programa de Pós-Graduação em Computação

Este trabalho discute o emprego de dois modelos de segurança que podem ser utilizados para evitar perdas de dados derivadas de falhas. Como base para estudos é utilizado um sistema projetado para gerenciar equipes em trabalho de mitigação de sinistros. Os modelos estudados são: i) Cópia Reserva, que mantém uma cópia atualizada dos dados sensitivos em um ou mais hospedeiros, e ii) Replicação sob Demanda, onde cada dispositivo monitora seu estado, replicando seus dados sensitivos ao detectar a iminência de uma falha. É analisada a eficiência demonstrada por cada uma das técnicas em relação à falhas e ao número de mensagens necessárias. De maneira complementar aos modelos de segurança, será estudada a aplicação do Cache Cooperativo, por meio do qual os dispositivos cooperam entre si para guardar e compartilhar cópias de dados distantes. O objetivo desta técnica é economizar a energia despendida na busca e roteamento do dado até o dispositivo leitor. O objetivo desta dissertação é demonstrar que o uso do Cache Cooperativo, em conjunto com a Replicação sob Demanda, pode permitir uma valiosa economia de recursos. Esta economia permite um ganho em tempo de vida para os dispositivos e proporcionando tolerância a falhas com baixo custo computacional.

FAULT TOLERANCE AND RESOURSES ECONOMY IN A SYSTEM OF MITIGATION OF SINISTERS

ABSTRACT

In this work we discuss the use of two fault-tolerance models that can help to avoid data loss due to failures in an ad hoc mobile network environment. As the basis for this study a system for team support in catastrophic contexts is used. The studied models are: i) Backup Copy, that keeps a up-to-date copy of the sensitive data in one or more hosts and ii) On-Demand Replication, where each device monitors its state, replicating its sensitive data when detecting an imminent failure. We analyze the efficiency for each technique related to failures and the number of messages that they need to work. In a complementary way to the fault-tolerance models, we study the use of Cooperative Cache, allowing devices to cooperate among each other to keep and share copies of remote data. The objective of this technique is to save energy used in locating and routing the data to the reader device. The objective of this dissertation is to demonstrate that the use of Cooperative Cache together with On-Demand Replication can help to save resources. This strategy allows to extend the energy lifetime for the devices and to achieve fault-tolerance at a relatively low-cost.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

SAE	-	Sistema de Apoio a Emergência
GBF	-	Gerenciador da Base Fixa
GBM	-	Gerenciador da Base Móvel
GDMP	-	Gerenciador do Dispositivo Móvel Portátil
SIC	-	Sistema de Informação de Contexto
MDF	-	Módulo Detector de Falhas
SCF	-	Sistema de Contenção de Falhas
DIMUT	-	Dado Imutável
DINDIV	-	Dado Individual
DCOMPART	-	Dado Compartilhado

1 INTRODUÇÃO

Desde os seus primórdios, a humanidade sempre enfrentou grandes desafios: guerras, epidemias e, também, as catástrofes naturais. Na maior parte de sua História, depois de um desses desastres, tudo o que se podia fazer era buscar pelas vítimas.

Com o desenvolvimento da ciência, porém, foi possível sair desse estado de passividade. Notadamente em passado recente, nas décadas finais do século XX, a evolução tecnológica melhorou muito tanto a especialização das equipes de socorro como a capacidade de previsões.

Em contrapartida, o aumento da densidade populacional nos grandes centros urbanos, o avanço do homem a regiões mais propícias a desastres e o atual cenário geopolítico que incita o terrorismo, os sinistros passaram a ter maior potencial de vítimas e de perdas materiais.

Diante disso, a melhor forma para diminuir os efeitos desses eventos é estar preparado. Só assim é possível evitar ou minimizar o número de vítimas e de perdas materiais. Além disso, quanto mais rápido se iniciam as ações de mitigação, maiores são as chances de se encontrar vítimas com vida.

Para tornar mais eficiente o primeiro combate ao sinistro, muito tem sido discutido em todo o mundo. Alguns tratados, inclusive, já foram estabelecidos, como, por exemplo, o de Tampere, de 1998, na Finlândia, validado durante a conferência ICET (*Intergovernamental Conference on Emergency Telecomunication*) e que estabelece mecanismos de provimento de infra-estrutura de telecomunicações e tecnologias para operações de recuperação em situações de desastres.

Em um cenário de catástrofe, a estrutura local de comunicações normalmente se encontra prejudicada ou inoperante devido ao sinistro, o que dificulta o acesso a informações. Nessas situações, a radiocomunicação quase sempre é a única opção disponível.

A evolução da tecnologia, no entanto, começa a oferecer novas opções. Dispositivos portáteis cada vez menores e mais potentes e conectados por redes sem fios se mostram como boas alternativas para interconectar os agentes responsáveis pela mitigação.

Essa tecnologia pode ser utilizada em um sistema projetado para gerenciar equipes que trabalham na mitigação de sinistros.

De posse desses equipamentos, as equipes têm a oportunidade de processar dados no local do evento e, assim, por exemplo, acessar mapas, solicitar recursos e organizar o fluxo de informações gerenciais. Com isso, viabilizam-se melhorias no controle de recursos, na sincronização das ações e na coordenação geral das equipes capazes de contribuir para a diminuição do tempo de mitigação do sinistro.

Com base nos cenários expostos no Quadro 1 e no Quadro 2, é possível entender o contexto de um sinistro com e sem um sistema de apoio.

Cenário 1 – Sem Sistema de Apoio
Considere-se um grande armazém de tintas. Durante o trabalho normal um curto-circuito inicia um incêndio de grandes proporções, que fere alguns trabalhadores e ameaça atingir os prédios vizinhos. Uma equipe de bombeiros é acionada. Somente quando uma pequena multidão começa a se aglomerar alguns policiais da redondeza fecham o trânsito. Quando os bombeiros chegam, mesmo tentando agir em conjunto, têm que tomar uma série de decisões individuais, pois o chefe do grupo tem somente um rádio comunicador para entrar em contato com sua equipe e nem o chefe nem seus homens tem uma visão global do problema. Ao localizarem as vítimas, verificam que algumas estão intoxicadas por um produto químico que necessita de atendimento não convencional e têm que aguardar a remoção para um hospital para que sejam feitos os primeiros socorros corretamente. Quando as ambulâncias chegam, as vítimas são removidas sem que se tenha conhecimento prévio de que o hospital de destino oferece atendimento nas especialidades necessárias e algumas vítimas têm que ser movidas novamente.

Quadro 1: Cenários de Sinistros Sem Sistemas de Apoio

Cenário 2 – Com Sistema de Apoio
Utilizando as mesmas condições iniciais de sinistro, pode-se imaginar uma equipe munida de um sistema distribuído de dispositivos móveis que permitam a disseminação e o controle de informações. Desta forma, no momento da solicitação de atendimento, dados como o mapa da instalação e informações relativas aos tipos de lesões que podem ser encontradas nas vítimas já são repassadas aos agentes. Serviços de apoio recebem automaticamente ordem de prontidão e hospitais são sondados em busca de leitos vagos e especialidades. Ao chegarem ao local, as equipes passam a combatem o incêndio e fazer solicitações de serviços de uma forma integrada. Durante o deslocamento, sensores sem fio são implantados no local criando um ambiente ativo. Este permite que o sistema possa agir de acordo com o contexto de cada área, bem como disponibilizar o conhecimento global de todo o sinistro. Um exemplo de utilização acontece quando os sensores detectam um aumento anormal de temperatura indicando uma explosão iminente. O sistema poderá sinalizar o evento permitindo o envio de uma equipe de apoio ou solicitar o recuo das equipes próximas. Durante o combate ao incêndio, encontra-se uma vítima. O sistema solicita imediatamente o transporte. Durante os primeiros procedimentos, uma intoxicação é identificada e o tratamento não é conhecido pelo bombeiro, uma solicitação é enviada em busca do procedimento correto. Enquanto isso, a vítima entra na ambulância. Quando encontrado, o procedimento é enviado de volta não para o bombeiro que o solicitou, mas para a equipe que está na ambulância com a vítima. E ainda, o sistema identifica o hospital mais próximo que possui leitos vagos e a especialidade requerida. Ao mesmo tempo, outra vítima é encontrada com sintomas semelhantes. O bombeiro, quando solicita o procedimento, já o obtém diretamente dos equipamentos locais. Por um defeito qualquer, o equipamento do chefe dos bombeiros sai do ar, imediatamente, o sistema se reorganiza para oferecer os serviços que foram perdidos, bem como avisar ao bombeiro hierarquicamente superior a sua condição de novo chefe.

Quadro 2: Cenários de Sinistros Com Sistemas de Apoio

O uso de pequenos dispositivos traz consigo uma série de desafios que precisam ser equacionados e resolvidos para que a utilização da tecnologia seja viável e prática. Dentre esses desafios, a limitação de recursos e a conectividade variável assumem especial importância, por diminuírem a confiança necessária a uma aplicação sensitiva como a de mitigação de sinistros.

Desta forma, técnicas que economizem espaço em memória, uso da rede, bateria etc., são necessárias para contornar a escassez de recursos. Em consonância, técnicas que evitem a perda de dados e aumentem sua disponibilidade previnem as flutuações de acesso causadas pela conectividade variável.

Mesmo com o considerável incremento da produção científica mundial proporcionado pelos avanços tecnológicos recentes, porém, é possível se observar ainda uma relativa escassez de literatura científica voltada à implementação de sistemas que auxiliem o atendimento emergencial e o resgate de vítimas de desastres naturais e não naturais com o uso de pequenos dispositivos.

Para definir melhor esse contexto, apresenta-se, no capítulo 2 deste trabalho, o Sistema de Apoio à Emergência – SAE [1], cujo protótipo foi criado para evidenciar, na prática, os diversos desafios de um sistema distribuído sobre uma rede sem fio. Seu objetivo é suportar a gerência de equipes que trabalharão na mitigação de sinistros.

No capítulo 3, são apresentadas as técnicas estudadas neste trabalho. Primeiro, o Cache Cooperativo [4], que consiste em uma cooperação mútua entre os dispositivos móveis portáteis membros de um grupo para, juntos, minimizarem a necessidade da busca e roteamento de um dado que já tenha sido trazido por algum membro do grupo, tornando, assim, o sistema mais eficiente principalmente no que diz respeito ao seu tempo de vida que é aumentado pela economia de energia gerada pela técnica.

Em seguida, trata-se das técnicas de tolerância a falhas que tentam evitar que haja indisponibilidade de um dado quando seu dispositivo hospedeiro falha, seja por falta de bateria, seja por desconexão. Será apresentado um método que utiliza cópias reservas dos dados e, em seguida, é apresentado e analisado o método de Replicação sob Demanda [5], cujo objetivo é replicar os dados de um dispositivo que está na iminência de uma falha, evitando, assim, a indisponibilidade dos dados. Pretende-se demonstrar o modo como o método escolhe o provável melhor hospedeiro e porque essa escolha influi em sua eficiência.

Para comparar os modelos de tolerância a falhas estudados no capítulo 3 foi criado um simulador que processa cenários de movimentação criados no ns-2 [22] e preparados para refletir o comportamento de equipes que trabalham em grupo. O capítulo 4 deste trabalho apresenta os algoritmos desse simulador, além disso, mostra o protótipo do SAE.

No capítulo 5, serão expostos os resultados das simulações fornecidos pelo simulador. Esses resultados demonstram os ganhos gerados por cada uma das técnicas e permitem definir quais técnicas terão comportamento mais eficaz no contexto de gerência de sinistros.

O capítulo 6 apresenta outras pesquisas relevantes envolvendo sistemas de gerência de equipes e, também, novas técnicas da tecnologia de informação para emergências e operações de busca e salvamento, além da relação dessas pesquisas com o presente trabalho.

Nas conclusões deste trabalho, no capítulo 7, serão colocadas em evidência as técnicas capazes de minimizar a indisponibilidade de dados com o menor consumo de recursos, oferecendo, assim, maior segurança em relação ao tempo de vida do sistema.

2 SISTEMA DE APOIO À EMERGÊNCIA – SAE

O SAE [1] é um protótipo de um sistema a ser utilizado para gerência de equipes que irão fazer o combate e mitigação de sinistros. Ele nasceu durante um estudo dirigido, cujo objetivo foi o de verificar onde e como os novos dispositivos portáteis com tecnologia sem fio podem ser úteis junto aos agentes que trabalham na mitigação de sinistros e, também, evidenciar os principais desafios da computação distribuída presente neste contexto.

A idéia inicial foi criar um sistema que, na medida do possível, automatizasse os passos a serem dados quando um sinistro fosse verificado, mantendo, dessa forma, a atenção dos responsáveis pela coordenação nos passos não automáticos e nas exceções particulares de cada evento.

Para criarmos um sistema escalável, propusemos uma arquitetura com três níveis de hierarquia, conforme demonstrado na Figura 1.

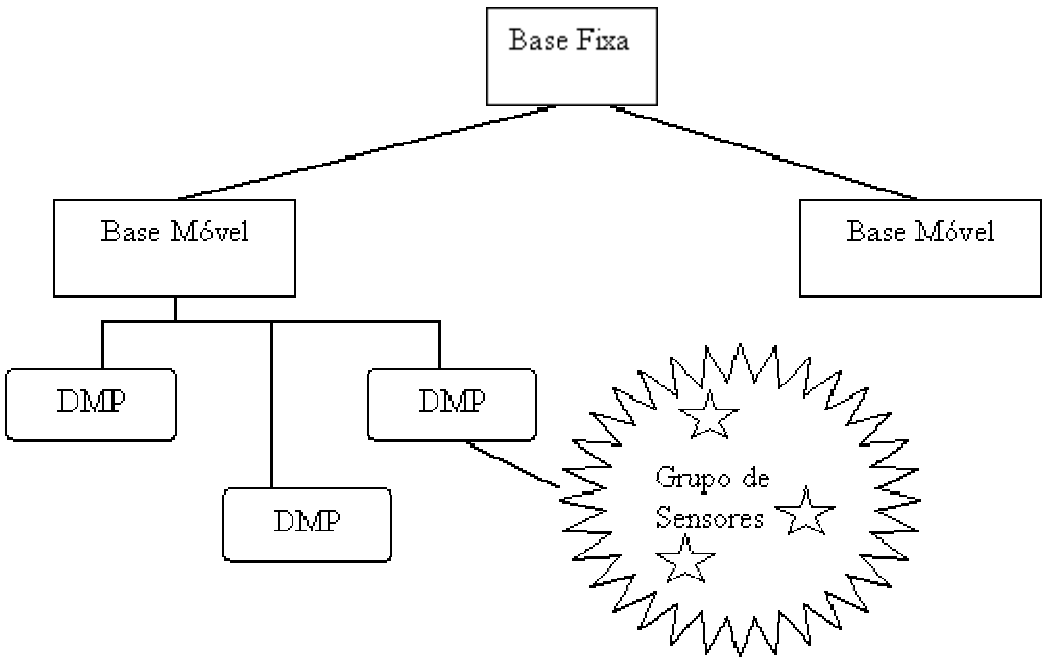

Figura 1: Arquitetura do Sistema SAE

No nível mais alto, temos o sistema que reunirá todas as informações, oferecendo-as da forma mais otimizada possível para os coordenadores. Esse sistema – Gerenciador da Base Fixa (GBF) – foi projetado para ser executado em equipamentos robustos, com alta disponibilidade de recursos.

No nível intermediário, encontramos o sistema Gerenciador da Base Móvel (GBM), cujo objetivo é oferecer ao comandante e aos membros de uma equipe todas as informações desta, bem como disponibilizar os dados oriundos da Base Fixa, podendo, inclusive, ser utilizada como *proxy*.

Vemos a arquitetura das bases Fixa e Móvel na Figura 2.

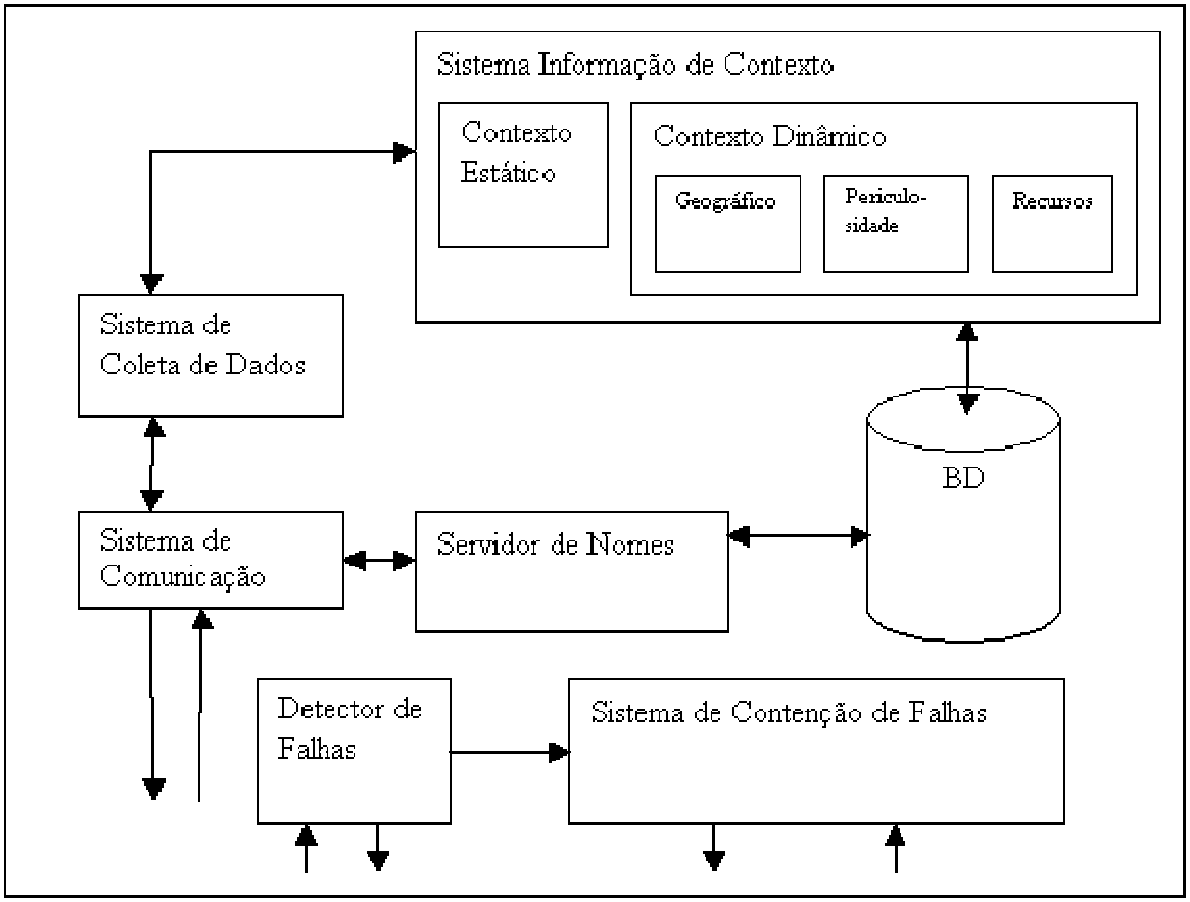

Figura 2: Arquitetura das Bases Fixa e Móvel

No nível inferior, temos o sistema Gerenciador do Dispositivo Móvel Portátil (GDMP). Feito para executar em dispositivos com pequena capacidade de processamento e recursos limitados, é o sistema que será levado pelo agente para frente de combate ao sinistro, onde poderá ser utilizado para colher informações de sensores de ambiente, fornecer mais opções de comunicação e executar outros processamentos que venham a ser necessários no local do sinistro.

A arquitetura do GDPM é mostrada na Figura 3.

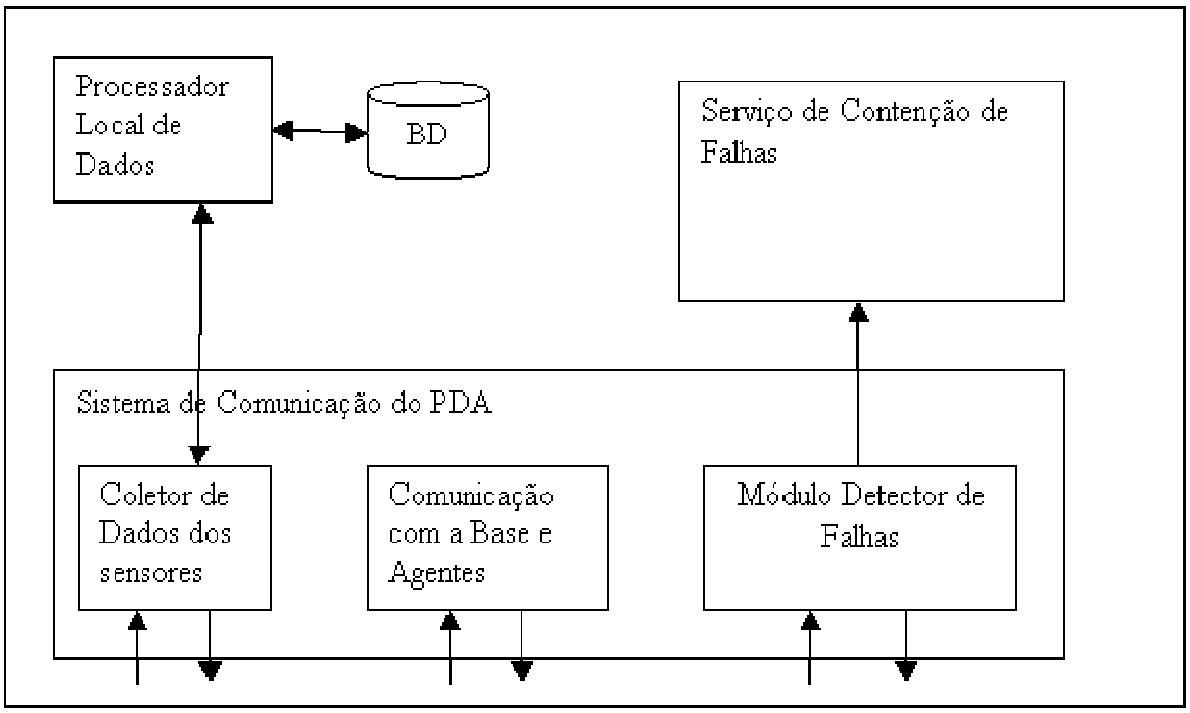

Figura 3: Arquitetura do GDMP

2.1 EQUIPES E GRUPOS

Sejam formadas por policiais, bombeiros ou médicos, as equipes possuem um conjunto de Dispositivos Móveis Portáteis e uma Base Móvel responsável pela sua gerência, conforme mostrado na Figura 4. A principal característica de uma equipe é o fato de que todos os seus dispositivos trabalham para atingir a um objetivo comum.

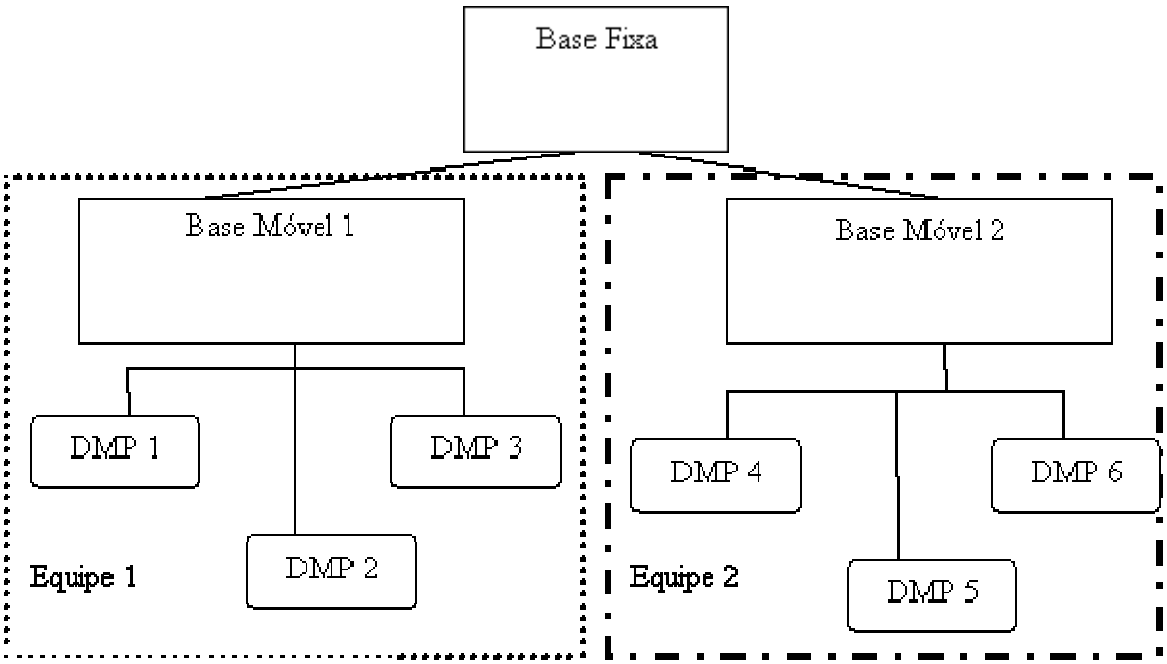

Figura 4: Divisão em Equipes.

Como a Base Móvel é projetada para executar em um computador portátil, em geral não é possível locomovê-la com o restante da equipe. Assim, chamamos de grupo todos os Dispositivos Móveis pertencentes a uma equipe, com exceção da Base Móvel. No capítulo 5, todas as simulações feitas serão sobre esse conceito de grupo.

A proximidade física e o fato de terem objetivos comuns nos levam – pelo princípio da vizinhança [10] – a considerar que os dispositivos de um grupo tendem a necessitar de um mesmo conjunto de dados. No capítulo 3, iremos mostrar como esse princípio nos permite gerar uma grande economia de recursos quando o utilizamos em conjunto com o Cache Cooperativo [4]. Para isso, são definidos como dados do grupo o conjunto de dados pertencentes aos dispositivos de um grupo e, como dados de fora do grupo, todos os dados do sistema, com exceção dos dados do grupo, conforme mostrado no exemplo do Quadro 3.

Dispositivos	Dados			
A	D1	D2	D3	D4
B	D5	D6	D7	D8
C	D9	D10	D11	D12
D	D13	D14	D15	D16
E	D17	D18	D19	D20
F	D21	D22	D23	D24
G	D25	D26	D27	D28
H	D29	D30	D31	D32
I	D33	D34	D35	D36

Grupos	Dispositivos		
1	A	B	C
2	D	E	F
3	G	H	I

Dados do grupo 1	D1 a D12
Dados de fora do grupo 1	D13 a D36

Quadro 3: Exemplo de conjunto de dados do grupo e de fora do grupo.

Esses dados, como se verá adiante, também deverão ser protegidos contra falhas dos dispositivos. As técnicas de proteção ou tolerância a falhas serão expostas no capítulo 3.

2.2 SUB-SISTEMAS PREVISTOS NO SAE

A seguir, apresentam-se os principais subsistemas descritos em [1] como importantes a um sistema de gerenciamento de sinistros.

2.2.1 O Sistema de Comunicação

O sistema de comunicação foi planejado para ser compacto sem prejudicar sua versatilidade. Foi concebido e implementado para facilitar todo o processo de comunicação entre os dispositivos. Cada dispositivo possui:

- Componente Cliente – É responsável por realizar o registro no servidor de nomes e as interações necessárias a uma comunicação;

- Servidor para Comunicação Ponto a Ponto – É responsável pelo recebimento e tratamento das mensagens;

- Servidor de Nomes: É um serviço de infra-estrutura de controle e apoio à comunicação entre as bases e os DMP. Assim, o servidor de nomes cumpre o papel de identificar as diversas entidades do sistema e responde quais estão atualmente conectadas, permitindo a criação de grupos e facilitando a comunicação interna de cada grupo (*multicast*). Como se trata de um serviço importante ao funcionamento do SAE, seus dados também deverão ser protegidos por alguma técnica de tolerância a falhas.

A arquitetura do Sistema de comunicação é mostrada na Figura 5.

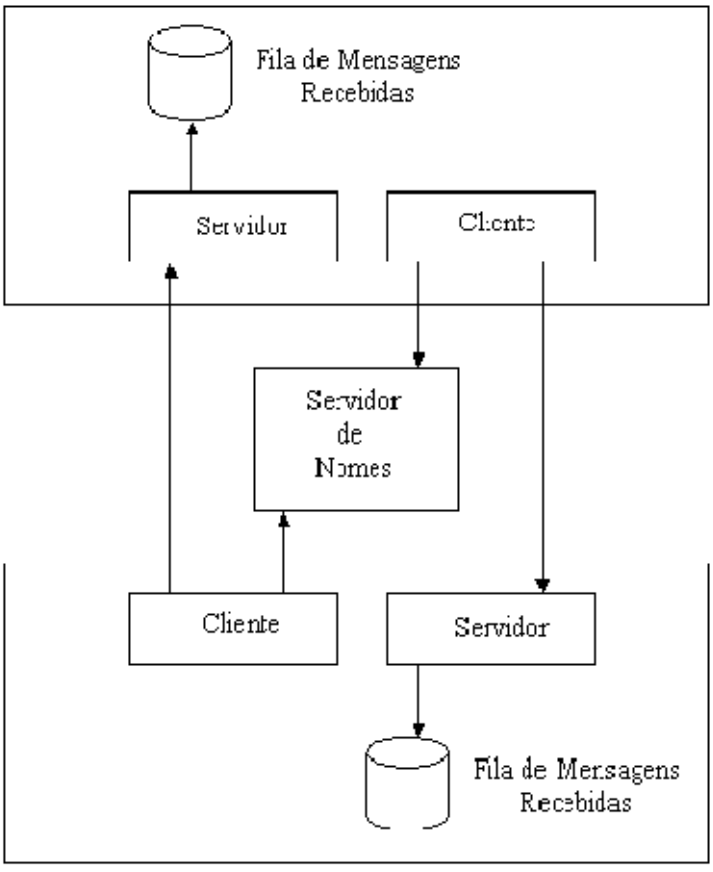

Figura 5: Arquitetura do Servidor de Nomes

2.2.2 Sistema de Informação de Contexto (SIC)

O Módulo de SIC é responsável por receber os dados captados dos sensores e transformá-los em informações de contexto do sinistro. Futuramente, esse módulo poderá fazer inferências e previsões a respeito dos sinistros. No momento, ele é responsável por avisar as equipes que determinadas condições de contorno previamente estabelecidas foram atingidas. É formado por:

- Módulo de Contexto Estático: tem como função tratar todas as informações de contexto que foram previamente armazenadas a respeito de um determinado tipo de sinistro. Por exemplo: guarda todas as ações automáticas e configurações que o sistema deve executar na ocorrência de um incêndio.

- Módulo de Contexto Dinâmico: gerencia as informações que são captadas pelos sensores. É composto por:

  - Sub-Módulo Geográfico: possui a informação de posição das equipes e pode ter também as informações do local do sinistro; consegue medir a distância entre agentes e possíveis pontos críticos, bem como monitorar a localização de cada agente durante a ação.

  - Sub-Módulo Periculosidade: Guarda as informações monitoradas pelos sensores, tais como temperatura, radiação etc. Seu objetivo principal é observar os perigos previsíveis sinalizando quando determinados eventos acontecem. Um exemplo, quando a temperatura chega a um limite ao qual uma equipe não preparada, esta é solicitada a evacuar o local.

- Sub-Módulo de Recursos: monitora as informações de recursos dos dispositivos pertencentes ao sistema, tais como bateria, amplitude do sinal de rádio etc. Esse módulo possibilitará uma análise atualizada dos recursos disponíveis enquanto são realizados os trabalhos de mitigação. Assim, evita-se uma desconexão por falta de bateria ou permite realocação de serviços em dispositivos com mais recursos. Como veremos no capítulo 3, este sub-módulo será utilizado no método de Replicação sob Demanda para minimizar a ocorrência de dados indisponíveis.

2.2.3 Sistema de Coleta de Dados

O Sistema de Coleta de Dados em uma rede de sensores é sozinho motivo para uma dissertação [23]. Esse componente está citado na arquitetura devido à crescente possibilidade de criação de um ambiente ativo através de uma rede de sensores. Aqui não trataremos da forma de coleta de dados nem da rede de sensores [24, 25, 26]. No momento, assumiremos que esse serviço é prestado por um Servidor de Dados que conhece as grandezas (temperatura, pressão, altura etc.) relevantes de cada ponto e os informa quando necessário no decorrer do trabalho.

2.2.4 Processador Local de Dados

Seu objetivo é tratar os dados recebidos de forma a só transmitir o que for relevante. É o responsável por programar os sensores e solicitar destes as informações. Guarda localmente as consultas [23] e as principais informações fornecendo-as ou descartando-as, quando necessário.

2.2.5 Módulo Detector de Falhas (MDF)

O Modulo Detector de Falhas trabalha para informar o sistema de eventuais falhas em quaisquer dispositivos ou serviços. No capítulo 4 trataremos da tolerância a falhas e veremos como esse módulo será utilizado para, junto com o Sistema de Contenção de Falhas, minimizar a perda de dados.

2.2.6 Sistema de Contenção de Falhas (SCF)

No Sistema de Contenção de Falhas é feito todo o processamento necessário das informações colhidas pelo MDF visando minimizar a perda de dados. Para que nenhum dado se perca, analisaremos no capítulo 3 dois modelos que utilizam ou, cópias reservas em hospedeiros distintos ou, replicação dos dados no momento que for detectada uma provável falha. Trataremos o SCF no capítulo 3.

2.3 CONHECIMENTO PRÉVIO DOS DADOS DO SISTEMA

Nos artigos [2,3] vemos que o trabalho para contenção de um sinistro pode ser dividido em diversas ações que deverão ser executadas por equipes determinadas e em uma ordem correta. Podemos dizer que cada uma dessas ações demanda um conjunto bem definido de dados, por exemplo, a contenção do incêndio de uma usina nuclear demanda o mapa da usina, plantas de tubulação de água etc., enquanto que a equipe de evacuação trabalha com o mapa viário das cidades afetadas além da densidade e distribuição populacional naquela área.

Outro conhecimento que podemos obter com a análise dessas ações é a determinação de como cada um dos dados pode ser utilizado. Desta forma, podemos definir previamente a política de leitura e escrita desses dados o que ajudará a definir a melhor estratégia de suas replicações e atualizações.

Com esse conhecimento prévio podemos caracterizar os dados em relação à sua política de escrita como sendo DIMUT (dado imutável), DINDIV (dado individual) ou DCOMPART (dado compartilhado):

- Um dado imutável (DIMUT): os dados imutáveis são aqueles que, uma vez criados, não serão modificados pelo sistema em todo o período do sinistro. Como o dado não possui nenhum dispositivo que o atualize, sua replicação e leitura pelos diversos dispositivos do sistema não demandam qualquer processamento com respeito a atualizações. Podemos citar mapas ou fotografias como exemplos de dados imutáveis.

- Um dado que somente um dispositivo modifica (DINDIV): este dado é caracterizado como tendo somente um escritor, podendo ser utilizado para leitura nos outros dispositivos. São dados DINDIV quaisquer medições de sensores, boletins meteorológicos ou documentos que são protegidos de escrita por quaisquer dispositivos que não sejam seu dono.

- Um dado que pode ser modificado por vários dispositivos (DCOMPART): quando mais de um dispositivo pode ter acesso de escrita e leitura em um dado, este será caracterizado como DCOMPART; um exemplo é o caso em que qualquer dispositivo móvel pode reservar uma vaga em um hospital, diminuindo assim a quantidade de vagas disponíveis.

Além dessas informações, podemos ter, em muitos casos, o conhecimento de parâmetros, tais como: o período de atualização de cada dado, o quanto este é sensível a leituras desatualizadas, dentre outros. Essas informações nos permitirão otimizar a replicação de cada dado procurando aumentar a eficiência do sistema.

2.4 DESAFIOS DO SAE

Dentre os diversos desafios encontrados no desenvolvimento do protótipo, dois chamam a atenção por estarem diretamente ligados à Computação Distribuída e ao Cenário de Sinistro que são:

- Economia de Recursos: ao utilizarmos dispositivos portáteis somos obrigados a conviver com poucos recursos, atentando-nos para uma máxima economia. Essa preocupação irá permitir que o sistema trabalhe por mais tempo (com economia de energia) e de forma mais eficiente (com redução do uso da rede). Veremos no item 3.1 que o trabalho em conjunto dos dispositivos de um grupo os permite economizar recursos, evitando longas buscas e roteamento de dados que já tenham sido utilizados por um dispositivo do grupo.

- Tolerância a Falhas: uma das características mais indesejáveis de um sistema que utiliza dispositivos móveis portáteis é o seu alto índice de falhas, seja por falta de bateria, por desconexão ou por quebra do equipamento. Como no nosso caso o dispositivo deverá ser projetado para trabalhar em situações adversas, abstraímos a quebra dos equipamentos, fixando-nos então nas falhas por falta de bateria e por desconexão. A princípio, não é possível evitarmos que a energia acabe ou que um dispositivo se desconecte. Por outro lado, podemos minimizar os efeitos dessa falha implementando um modelo que evite a indisponibilidade dos dados do dispositivo que falhou. Esse modelo, por sua vez, não pode exigir grande quantidade de recursos, pois, assim irá diminuir o tempo de vida do sistema. Veremos dois modelos de tolerância a falhas no capítulo 3 e estudaremos suas eficiência e seus impactos no tempo de vida do sistema.

2.5 COMENTÁRIOS

O desenvolvimento do SAE nos permitiu imergir no contexto de uma aplicação voltada à gerência de equipes que trabalham na mitigação de sinistros. Notamos a importância de determinadas informações e verificamos a necessidade destas serem protegidas. Além disso, diante das constantes necessidades de recarga, ficou evidente a importância da minimização do consumo de energia. Assim, a necessidade de proteção e de economia nos levou ao estudo das técnicas descritas a seguir.

3 CACHE COOPERATIVO E TOLERÂNCIA A FALHAS

Conforme visto no capítulo anterior, necessitamos de técnicas que diminuam o risco de perda de dados, com o menor dispêndio de recursos possível.

O trabalho em grupo para atingir um objetivo comum nos permite inferir que o conjunto de dados necessários a cada dispositivo de um grupo seja bastante semelhante. O Cache Cooperativo, tratado a seguir, é uma técnica que utiliza essa necessidade comum para economizar energia.

Em seguida, veremos que quando um dispositivo falha, seja por falta de energia, seja por desconexão, seus dados ficam indisponíveis a outros dispositivos. Essa situação é extremamente indesejável. Assim, técnicas de tolerância a falhas devem ser utilizadas para minimizar essa indisponibilidade. Compararemos aqui as técnicas de Cópia Reserva e de Replicação Sob Demanda.

A primeira exige que os dispositivos mantenham uma ou mais cópias atualizadas de seus dados em outros hospedeiros à custa de espaço em memória e de mensagens de atualização sempre que o dado é alterado.

A segunda obriga aos dispositivos a uma vigilância constante às suas condições de conectividade e de energia. Quando se verifica, como veremos a seguir, a iminência de uma falha, o dispositivo cria em um hospedeiro uma réplica de seus dados protegendo-os.

3.1 O CACHE COOPERATIVO

Duas características intrínsecas ao SAE são: *(a)* o fato de que cada uma de suas equipes tem seus objetivos bem definidos; e *(b)* seus componentes tendem a estar próximos durante a atividade de mitigação. Essas características, conforme exposto em [10], faz com que os dispositivos pertencentes a uma equipe tendam a necessitar de um conjunto semelhante de dados. A partir disso, uma forma de economia de recursos que pode ser adotada é o Cache Cooperativo [4].

Podemos verificar a ocorrência das necessidades comuns previstas em [10] voltando ao exemplo exposto no Quadro 2. Vemos que os membros da equipe que necessitam resgatar as vítimas têm a necessidade da planta do armazém. Assim, se um agente solicita esta planta à central e a baixa para seu dispositivo, não há porque os outros bombeiros irem tão longe em busca da mesma informação, podendo obtê-la do dispositivo mais próximo que a possuir. Da mesma forma, policiais que necessitam desviar o trânsito necessitam não da planta do armazém, mas do mapa viário do local e, quando o dispositivo de um policial receber esse mapa, poderá compartilhá-lo com todos os outros policiais.

A técnica do Cache Cooperativo orienta que quando um dispositivo necessita de um dado externo a seu grupo – definido em 2.1 –, este inicia uma busca pelo dado e, ao encontrá-lo, guarda localmente uma cópia cache com um tempo de vida pré-determinado inerente ao dado. Durante esse tempo de vida, se qualquer dispositivo deste grupo necessitar do mesmo dado, o acessa dessa cópia cache local, evitando um longo ciclo de busca e roteamento necessários para trazer o dado de um dispositivo de outro grupo.

Passo a passo, vemos na Figura 6 a aplicação desta técnica: (1) O dispositivo solicita um dado aos outros de seu grupo; (2) Quando não o encontra, inicia uma busca; (3) Ao localizá-lo, o dispositivo armazena-o localmente como uma cópia, registrando o momento da leitura (para verificação da validade da cópia); (4) Quando outro dispositivo do grupo necessita do mesmo dado dentro do período de validade da cópia, este pode obtê-lo diretamente do dispositivo que guardou a cópia cache, evitando todo um ciclo de busca e roteamento.

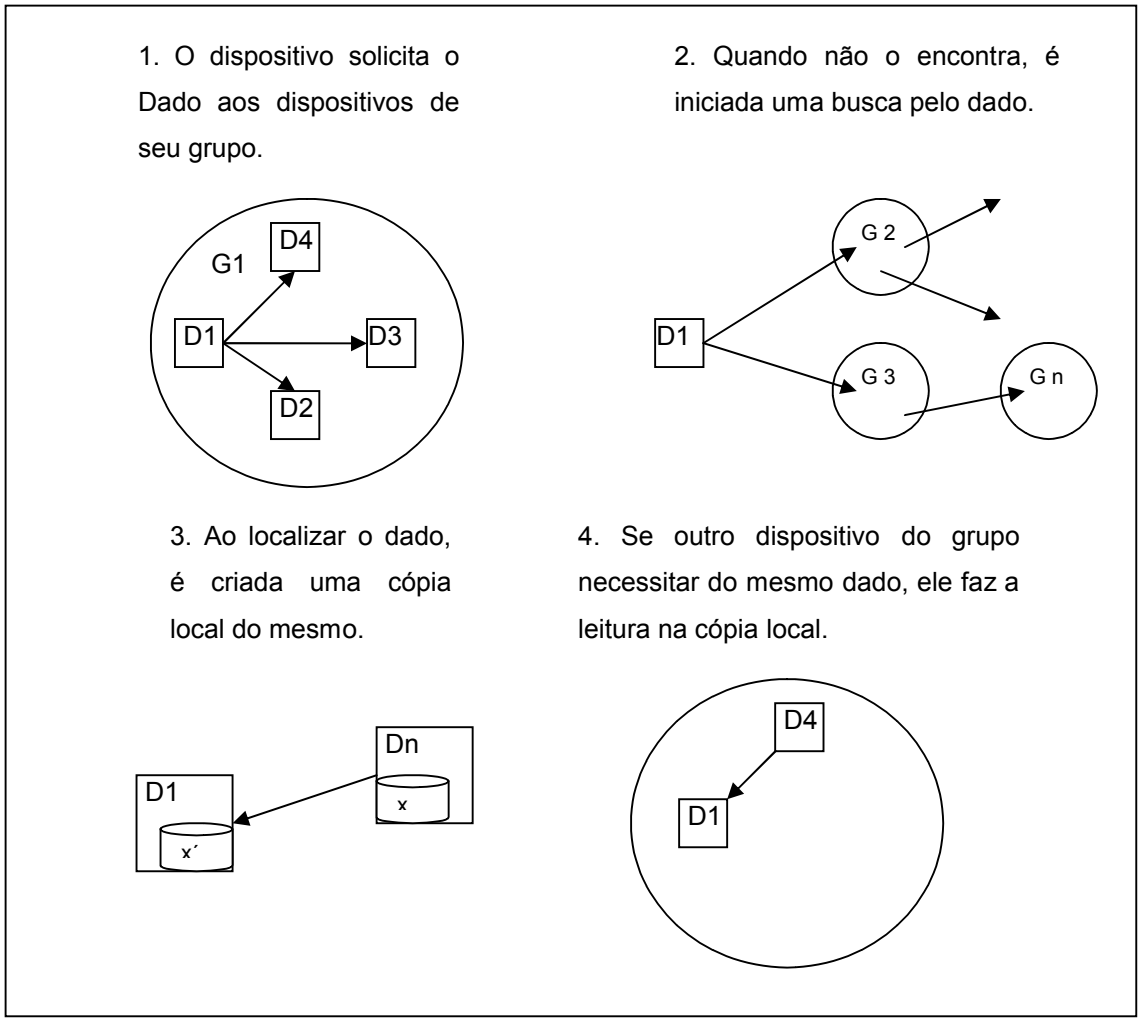

Figura 6: Funcionamento do Cache Cooperativo.

3.1.1 Tempo de Vida da Cópia Cache

Todos os tipos de dados – DIMUT, DINDIV e DCOMPART – podem ser guardados no cache cooperativo. O principal fator que deve ser observado é o tempo de vida utilizado na cópia cache. Esse tempo de vida deve ser escolhido de acordo com:
(a) O tipo de dado:

- Dados do tipo DIMUT: podem ter tempo de vida infinito, posto que, por definição, não serão atualizados. Assim, uma leitura em uma cópia cache de um dado DIMUT nunca retornará um valor diferente (leitura desatualizada) de uma leitura feita no dado original.

- Dados do tipo DINDIV e DCOMPART: podem ser alterados e devem ter um tempo de vida fixado em função de suas características.

(b) O período de atualização:

Dados do tipo DINDIV e DCOMPART podem ter períodos de atualização definidos. Assim, em alguns casos, é possível determinar o tempo de vida em função desse período.

(c) A suscetibilidade à leitura desatualizada

Caso a informação contida em um dado cause problemas no atendimento ao sinistro se for lida com erro, o total de vagas em um hospital, por exemplo, o tempo de vida deverá ser reduzido.

É importante ressaltar que a cópia cache utilizada no Cache Cooperativo não é uma réplica e, por conseguinte, não demanda atualizações. Isto aumenta ainda mais a importância do tempo de vida, posto que o acesso a cópias desatualizadas pode prejudicar as atividades do dispositivo leitor.

3.1.2 Implementação e Objetivos Esperados

O cache cooperativo foi implementado no simulador, onde utilizamos um tempo de vida de 60 segundos. Como a técnica do cache cooperativo se destina a evitar buscas de dados que estejam em dispositivos distantes do dispositivo leitor, ou seja, dados que necessitam de um longo roteamento para serem acessados, as simulações deverão demonstrar uma economia de energia proporcionada pela leitura em uma cópia próxima.

Da mesma forma, iremos mensurar a quantidade de erros introduzidos no sistema por leituras feitas em cópias desatualizadas, nas quais o dado original já tenha sido modificado. Veremos os resultados desta simulação no item 5.1.

3.2 TOLERÂNCIA A FALHAS

Duas falhas comuns das redes ad hoc formadas por dispositivos móveis portáteis são a inconstância da conectividade e a ocorrência de falta de bateria. Para que tenhamos um sistema confiável é necessário, no mínimo, que sejam usados métodos que proporcionem uma tolerância a estas falhas.

Neste trabalho estudamos duas propostas que visam minimizar a perda de dados, o uso de Cópia Reserva e o método de Replicação sob Demanda [5]. Para melhor entendermos a replicação de dados, veremos, no item 3.2.1, seus principais conceitos.

3.2.1 Técnicas de Replicação de Dados

Replicação de dados consiste em manter várias cópias de um determinado dado com objetivo de aumentar o desempenho do sistema, diminuindo a latência existente em acessos remotos [17], bem como permitir que esse dado esteja disponível mesmo em uma eventual desconexão. Em paralelo, havendo mais de uma cópia de um dado, a probabilidade deste ser perdido por uma falha é bastante diminuída.

Existem dois estilos de replicação: a pessimista e a otimista. A Pessimista mantém uma consistência pessimista, ou seja, bloqueia o acesso a uma réplica até que ela esteja atualizada. Um algoritmo elege uma cópia como primária e esta fica responsável pelos acessos ao seu conteúdo. Após uma atualização, as mudanças na cópia primária são propagadas de forma síncrona para as cópias secundárias. Caso haja uma falha com a cópia primária, uma cópia secundária é eleita como uma nova cópia primária.

Esse modelo funciona bem em redes cuja latência é baixa e as falhas incomuns. A arquitetura SAE utiliza uma rede sem fio com grande probabilidade de desconexões, falhas nos equipamentos ou falta de bateria. Além disso, a latência pode ser bastante significativa principalmente quando houver comunicação entre dispositivos de diferentes grupos,.

Por sua vez, o estilo Otimista, permite que acessos de leitura ou escrita possam ser executados sem uma sincronização a priori. Este estilo se baseia no fato de que conflitos de acesso a dados são pouco freqüentes e que podem, quando acontecerem, serem, de alguma forma, dirimidos.

Podemos citar como vantagens potenciais do estilo otimista.

- Aumento da disponibilidade dos dados: por não haver necessidade da espera de um bloqueio de outras cópias, um dado é acessado muito mais rapidamente.

- Flexibilidade com respeito à rede: usando técnicas de disseminação, as operações realizadas seguramente chegarão às outras cópias mesmo com um grafo de comunicação variável.

- Escalabilidade: por requerer menos sincronizações, podemos ter maior número de réplicas.

A grande questão do estilo otimista é a troca da consistência pela disponibilidade. O que gera o desafio de tratar réplicas divergentes e conflitos entre operações concorrentes. Especificamente no SAE temos casos onde a confiabilidade é importante e outros, onde a disponibilidade é necessária. Vemos isso quando tratamos de um mapa que é raramente alterado, mas cuja disponibilidade é necessária.

Em oposição, o número de vagas disponíveis em um hospital que pode ser alterado por qualquer equipe que solicite uma vaga, é um tipo de informação que poderá causar um problema grave quando errada.

Para melhor nos referirmos aos dados, chamaremos somente de dado o dado que pode ser lido e escrito por seu(s) proprietário(s), e de cópia reserva, um dado potencialmente idêntico a sua cópia-mestre, que só pode ser lido durante seu tempo de validade e não pode ser escrito. A cópia reserva só é idêntica dado no momento da sua atualização. Como não há nenhum impedimento que o dado seja escrito, quando isso acontece, as leituras em suas cópias reservas são consideradas desatualizadas.

Como mostrado anteriormente, o SAE possui dados do tipo DIMUT, DINDIV e DCOMPART. Os dados do tipo DIMUT e DINDIV, a princípio, só podem ser escritos por um único dispositivo e, por isso, não têm problemas de consistência. Quando inserirmos a tolerância a falhas, serão criadas, para esses tipos de dados, cópias reservas. Como estas não podem receber operações de escrita, mantêm a característica inicial de não ter problemas de consistência. As leituras feitas nessas cópias reservas podem ser exatas, ou seja, idênticas às realizadas no dado, ou desatualizadas, quando feitas antes da cópia reserva receber uma atualização.

Dados DCOMPART, por definição, existem replicados em mais de um dispositivo. Isto os torna menos suscetíveis a falha de um único dispositivo. A maior preocupação com relação a esses dados passa a ser relativa à sua consistência. Em [17] vemos um trabalho relacionado às técnicas de descoberta e resolução de conflitos em dados que possuem mais de um hospedeiro.

3.3 SISTEMA DE SUPORTE PARA MINIMIZAR PERDAS DE DADOS

Sem nenhuma segurança, quando um dispositivo falha seus dados ficam indisponíveis aos outros dispositivos. Para minimizarmos as perdas de dados serão estudados dois métodos de Tolerância a Falhas: o uso Cópia Reserva e a utilização da Replicação Sob Demanda.

3.3.1 Modelo Cópia Reserva

O modelo Cópia Reserva prevê que todo dado sensitivo ou necessário deve, por segurança, manter uma cópia reserva em outro dispositivo.

Após a criação da cópia reserva, esta deve ser mantida atualizada pelo dono do dado. Isso acontece porque somente o dispositivo que possui o dado tem o controle das suas atualizações e, assim, tem a responsabilidade de enviá-la às cópias reservas.

A necessidade da cópia reserva se justifica mesmo quando o dado passa a ser utilizado por mais dispositivos que o guardam localmente em cópias cache. Essas cópias cache não podem ser confundidas com as cópias reservas, por não possuírem atualização pró-ativa do dono do dado.

A técnica de cópia reserva propõe que os dados tipo DIMUT e DINDIV ao serem criados, devem ser replicados em outro dispositivo que pertença ao grupo de seu proprietário. Desta forma, qualquer eventualidade que cause a falha de seu dispositivo proprietário não causará a perda do dado.

A cópia reserva, como visto anteriormente, deve ser mantida atualizada por seu proprietário de forma pró-ativa, ou seja, periodicamente o proprietário deverá enviar ao dispositivo hospedeiro o conjunto de modificações que foram realizadas desde a ultima atualização.

Como o dispositivo hospedeiro não tem permissão de escrita no dado, não há necessidade de nenhum processo de verificação de conflitos. Porém, o fato de utilizarmos uma cópia reserva que é atualizada periodicamente e não a cada modificação, é caracterizada uma replicação otimista, onde se espera que a cópia reserva esteja atualizada no momento de uma falha.

Se considerarmos que existe a probabilidade Pd do dono falhar, e a probabilidade Ph do hospedeiro falhar, a probabilidade de ambos falharem ao mesmo tempo é de Pd * Ph minimizando a chance de perda.

3.3.2 Modelo de Replicação Sob Demanda

Dois problemas são evidentes no método anterior: a necessidade do dobro de espaço para os dados e o uso constante da rede – com conseqüente desperdício de recursos – para manter as cópias reservas atualizadas.

Uma outra opção que tem o potencial de evitar, na maioria dos casos, a perda de dados sem o grande dispêndio de recursos da técnica anterior é programar os dispositivos para monitorarem suas condições de quantidade de energia e conectividade. Assim, antes de uma falha por falta de bateria ou por desconexão, o dispositivo poderá replicar seus dados em outro que esteja em melhor situação. Desta forma, antes que as falhas mais comuns – falta de bateria e desconexão – aconteçam os dados são replicados em outro dispositivo, permanecendo, assim, acessíveis.

Se observarmos essa solução em funcionamento, poderemos detectar a seguinte situação. Sejam os dispositivos A, B e C com respectivos dados d1, d2 e d3. Em um dado momento, o dispositivo A detecta que vai falhar e inicia um processo de replicação copiando seus dados no dispositivo B, que passou a possuir d1 e d2. Logo após isso, o dispositivo B também detecta uma falha iminente e replica seus dados em C.

Note que, após essas replicações d1 foi replicado duas vezes e d2 uma vez. Se o dispositivo A tivesse replicado seus dados diretamente em C, teríamos d1 e d2 replicados somente uma vez com evidente economia de recursos. Ou seja, quanto melhor for a escolha do dispositivo para replicação, mais eficiente será o sistema; em outras palavras quanto mais tempo o dispositivo escolhido levar para falhar melhor será essa técnica.

A questão, por conseguinte, passa a ser como descobrir qual o melhor dispositivo para replicar os dados. No trabalho [5], é proposto um método (chamaremos de método de Replicação sob Demanda) de utilização de técnicas de probabilidade a posteriori – Bayes – e lógica Fuzzy para determinar qual o melhor hospedeiro que o dispositivo com problemas deverá escolher.

3.3.2.1 Teoria de Decisão Fuzzy-Bayesiana

O teorema de Bayes é usado na inferência estatística para atualizar estimativas da probabilidade de que diferentes hipóteses sejam verdadeiras, baseado no conhecimento de como essas observações se relacionam com as hipóteses. O teorema, então, diz que a probabilidade direta de uma hipótese *H* condicionada a um corpo de dados *E*, *P(H|E)* está relacionada com o inverso da probabilidade dos dados *E* condicionados à hipótese *H*, *P(E|H).*

Matematicamente temos: $\mathrm{P(H \mid E)} = \frac{P(E \mid H)}{P(E)} P(H)$ (1-a)

Para aplicarmos corretamente o teorema, devemos identificar os estados incertos do sistema, as ações a serem tomadas, as funções utilidades e as variáveis envolvidas que precisarão ser observadas.

É possível fazer essa modelagem com estados discretos ou contínuos, No modelo discreto, atribuímos uma probabilidade s_i para ocorrência de cada estado. No modelo contínuo, temos a função densidade de probabilidade para o estado do sistema. Essa probabilidade, chamada *a priori,* deve ser baseada somente no conhecimento prévio do comportamento do sistema.

Para que possamos aplicar o teorema de Bayes é preciso que antes tenhamos feito uma coleta de dados que nos permitirá aproximar os valores de probabilidades da real tendência do sistema. Além disso, devemos gerar uma tabela com as recompensas esperadas para cada ação tomada. A função que gera esta tabela é a *função de utilidade.* Esta tabela informa o benefício de cada ação a ser tomada em cada um dos estados. Ou seja, temos uma tabela *n x m* onde *n* é o número de estados e *m* o número de ações.

Com as probabilidades *a priori* e os dados coletados, fazemos uma observação *x* no sistema, e, aplicando o teorema de Bayes podemos obter novos valores de probabilidades para os estados do sistema, chamadas probabilidades *a posteriori.*

A partir das probabilidades *a posteriori* e da *função de utilidade* temos a *utilidade esperada* que é dada pelo somatório dos produtos de cada probabilidade *a posteriori* pela recompensa associada. A ação que tiver a maior *utilidade esperada* é a melhor ação a ser tomada.

3.3.2.2 Definição dos Estados de Incertezas

Em nosso contexto, o estado é caracterizado por duas variáveis importantes para o tempo de vida conectado de um dispositivo móvel: energia e conectividade. A energia é uma variável aleatória que pode ser representada por uma função estritamente decrescente com taxa de declínio *l* que não pode ser determinada com certeza em um dado tempo *t*. Podemos definir conectividade como o quanto um dispositivo está conectado em uma rede ad hoc. Um dispositivo pode estar mais ou menos conectado de acordo com sua posição com relação aos outros, e até estar completamente desconectado, quando está fora de alcance de qualquer dispositivo.

Podemos definir r estados discretos para cada uma dessas variáveis, sendo que os estados do sistema serão compostos pelo produto dos estados da conectividade pelos da energia. Para minimizarmos o número de estados, os classificamos em três conjuntos Fuzzy. Como a conectividade e a energia são variáveis independentes, sua probabilidade conjunta é dada pelo produto de suas probabilidades.

Para energia definimos os estados *fraco, aceitável e forte,* que representam não a quantidade de bateria, mas uma estimativa de seu tempo de vida. Similarmente, temos esses estados para a conectividade, representando a intensidade do sinal que chega ao dispositivo, que é inversamente proporcional à somatória do quadrado das distâncias do dispositivo até seus vizinhos. Isto nos gera um conjunto de estados para o sistema mostrado na Tabela 1.

Tabela 1: Estados do Sistema

Estado	Energia (e) e Conectividade (c)
s1	e=fraco, c=fraco
s2	e=fraco, c= aceitável
s3	e=fraco, c=ótimo
s4	e= aceitável, c=fraco
s5	e= aceitável, c= aceitável
s6	e= aceitável, c= ótimo
s7	e= ótimo, c=fraco
s8	e= ótimo, c= aceitável
s9	e= ótimo, c= ótimo

3.3.2.3 Ações e Função de Utilidade

Como exposto anteriormente, a função de utilidade determina o valor da recompensa para cada ação tomada em cada estado possível. Ao ser executado, o método deverá determinar qual o dispositivo móvel vizinho ao dispositivo com problemas deverá ser escolhido como hospedeiro. Desta forma, temos o conjunto A de ações dados por $A=\{a_1, a_2, ..., a_i,..., a_n\}$, onde $i \neq j$, sendo i o dispositivo a ser escolhido e j o dispositivo com problemas.

O objetivo de escolher o dispositivo que ofereça maior tempo de disponibilidade dos dados nos indica que devemos atribuir uma recompensa crescente quando vamos do estado s1 para o s9, criando então a matriz que relaciona os estados com as ações possíveis. Como a ação de replicar um dado é única, seu valor é o mesmo, o que muda são as funções de probabilidade que serão selecionadas para cada dispositivo ao qual a ação se refere.

Para a aplicação da teoria fuzzy-bayesiana de decisão, deve-se poder calcular a probabilidade de ocorrência, em um dispositivo móvel, de qualquer valor possível de energia que possa ser observado, posto que um estado ocorreu.

O trabalho [5] obteve a densidade de probabilidade de energia e conectividade através da análise de um conjunto grande de experimentos para os estados possíveis do sistema.

Segundo a teoria da probabilidade, o teorema do limite central estabelece que a média de uma seqüência de variáveis aleatórias independentes, qualquer que seja sua distribuição de probabilidade, tende a se tornar uma distribuição normal. Assim, quaisquer simulações em áreas semelhantes deverão tender a essa mesma distribuição de probabilidade.

Tabela 2: Densidades de probabilidade para energia.

	Intervalos de Valores de Energia							
	0	1	2	3	4	5	6	7
e1	0,913949	0,086001	0,000050	0,000000	0,000000	0,000000	0,000000	0,000000
e2	0,020870	0,841518	0,137411	0,000200	0,000000	0,000000	0,000000	0,000000
e3	0,000000	0,032560	0,780648	0,185892	0,000900	0,000000	0,000000	0,000000
e4	0,000000	0,000000	0,042190	0,728657	0,226722	0,002430	0,000000	0,000000
e5	0,000000	0,000000	0,000000	0,049400	0,680177	0,265793	0,004630	0,000000
e6	0,000000	0,000000	0,000000	0,000000	0,055610	0,639470	0,296480	0,008400
e7	0,000000	0,000000	0,000000	0,000000	0,000090	0,059351	0,603056	0,936489
e8	0,000000	0,000000	0,000000	0,000000	0,000000	0,000130	0,063281	0,936489

Tabela 3: Densidades de probabilidade para conectividade

	Intervalos de Valores de Conectividade							
	0	1	2	3	4	5	6	7
c1	0,564167	0,207523	0,081180	0,039045	0,024648	0,014888	0,010408	0,058142
c2	0,416515	0,272802	0,112785	0,057652	0,033663	0,020820	0,012914	0,072849
c3	0,274731	0,276250	0,156001	0,084930	0,048105	0,032473	0,021792	0,105718
c4	0,191063	0,262295	0,174049	0,102700	0,063073	0,042839	0,029585	0,134395
c5	0,139472	0,234856	0,187008	0,118938	0,074047	0,048566	0,033711	0,163403
c6	0,100367	0,193871	0,182225	0,127961	0,089039	0,061280	0,042884	0,202373
c7	0,075086	0,154516	0,166644	0,131792	0,100211	0,072410	0,051556	0,247785
c8	0,055307	0,128123	0,150686	0,135078	0,105669	0,079178	0,059544	0,286414

3.3.2.4 Equações Fuzzy-Bayesianas

As equações utilizadas pelo método de decisão Fuzzy-Bayesiano modificam as equações de decisão bayesiana convencionais introduzindo as funções de pertinência Fuzzy. A equação 1-a gera a probabilidade a posteriori $\Pr[s_i|x_k]$, representando a probabilidade de ocorrer o estado s_i dado que foi observada a informação x_k. A informação x_k é coletada do sistema e pode assumir um dos valores $X=\{x_1, x_2, \ldots, x_r\}$ onde r é a quantidade de níveis discretos que estão sendo considerados. Assim, a equação 1-a pode ser reescrita como:

$$\Pr[s_i \mid x_k] = \frac{\Pr[x_k \mid s_i]}{P[x_k]} P[s_i] \qquad \text{(1-b)}$$

Onde $\Pr[x_k]$ é a probabilidade marginal do dado estado x_k determinada pelo teorema da probabilidade total:

$$\Pr[x_k] = \sum_{i=1}^{n} \Pr[x_k \mid s_i] \Pr[s_i] \qquad \text{(2)}$$

Tendo calculado todas as probabilidades posteriores para cada estado, sua utilidade esperada é:

$$E(u_j \mid x_k] = \sum_{i=1}^{n} u_{ji} \Pr[s_i \mid x_k] \quad (3)$$

Onde u_{ji} é a recompensa atribuída para a j-ésima alternativa no i-ésimo estado. A melhor ação a ser tomada é a de maior utilidade:

$$E(u^* \mid x_k) = \max_j E(u_j) \quad (4)$$

Como as informações e os estados são inerentemente Fuzzy, devemos modificar as equações 1-b, 2, 3 e 4 para incluirmos a lógica Fuzzy. A probabilidade de um evento Fuzzy é definida por:

$$\Pr[\underset{\sim}{M}] = \sum_{K=1}^{r} \mu_{\underset{\sim}{M}}(x_k) \Pr(x_k) \quad (5)$$

Os eventos Fuzzy – fraco, aceitável e forte – que irão descrever as informações $\underset{\sim}{M}_1$, $\underset{\sim}{M}_2$, $\underset{\sim}{M}_3$ devem ser ortogonais, ou seja, a soma do valor da função de pertinência dos três conjuntos para qualquer ponto x_k deve ser 1. Então:

$$\sum_{i=1}^{3} \mu_{\underset{\sim}{Mi}}(x_k) = 1 \quad (6)$$

A Tabela 4 com oito níveis representa os conjuntos Fuzzy utilizados.

Tabela 4: Definição dos Conjuntos Fuzzy

	v1	v2	v3	v4	v5	v6	v7	v8
$\underset{\sim}{M}_1$	1,0	1,0	0,5	0,0	0,0	0,0	0,0	0,0
$\underset{\sim}{M}_2$	0,0	0,0	0,5	1,0	1,0	0,5	0,0	0,0
$\underset{\sim}{M}_3$	0,0	0,0	0,0	0,0	0,0	0,5	1,0	1,0

Utilizando a equação 5 e o teorema de Bayes obtemos a probabilidade posterior dada a informação Fuzzy:

$$\Pr[s_i \mid \underset{\sim}{M}] = \frac{\Pr[\underset{\sim}{M} \mid s_i]\Pr[s_i]}{\Pr[\underset{\sim}{M}]} \tag{7}$$

Onde,

$$\Pr[\underset{\sim}{M} \mid s_i] = \sum_{k=1}^{r} \mu_{\underset{\sim}{M}}(x_k)\Pr[x_k \mid s_i] \tag{8}$$

Substituindo 5 e 8 em 7 temos:

$$\Pr[s_i \mid \underset{\sim}{M}] = \frac{\sum_{k=1}^{r} \mu_{\underset{\sim}{M}}(x_k)\Pr[x_k \mid s_i]\Pr[s_i]}{\sum_{k=1}^{r} \mu_{\underset{\sim}{M}}(x_k)\Pr(x_k)} \tag{9}$$

Com as informações Fuzzy, obtemos as equações 10 e 11, equivalentes às equações 3 e 4, para um evento Fuzzy $M_{i.}$

$$E(u_j \mid \underset{\sim}{M}_l) = \sum_{i=1}^{n} u_{j_i} \Pr[s_i \mid \underset{\sim}{M}_l] \tag{10}$$

$$E(u^* \mid \underset{\sim}{M}_l) = \max_j \Pr(u_j \mid \underset{\sim}{M}_l) \tag{11}$$

Sejam E_1, E_2 e E_3 os conjuntos Fuzzy para a energia, A função $\mu_{Ei}(e_k)$ é a pertinência do valor e_k de energia ao conjunto Fuzzy E_i Assim, a probabilidade do estado de energia E_s é:

$$\Pr[\underset{\sim}{E}_s] = \sum_{k=1}^{m} \mu_{\underset{\sim}{Es}}(e_k) \Pr[e_k] \tag{12}$$

Esse valor substitui o valor de $\Pr[\underset{\sim}{E}_s]$ na equação 9 gerando:

$$\Pr[\underset{\sim}{E}_s \mid \underset{\sim}{M}_l] = \frac{\sum_{i=1}^{m}\sum_{k=1}^{r} \mu_{\underset{\sim}{Es}}(e_i)\mu_{\underset{\sim}{M_l}}(x_k) \Pr[x_k \mid e_i] \Pr[e_i]}{\sum_{k=1}^{r} \mu_{\underset{\sim}{M_l}}(x_k) \Pr[x_k]} \tag{13}$$

Analogamente, a equação 13 deve ser expressa para conectividade com seus conjuntos Fuzzy $\underset{\sim}{C}_1$, $\underset{\sim}{C}_2$ e $\underset{\sim}{C}_3$.

Por fim, devemos combinar os valores obtidos para gerar as probabilidades dos nove estados possíveis e com eles sermos capazes de calcular a utilidade esperada.

3.3.2.5 O Custo Computacional do Modelo de Replicação sob Demanda

As fórmulas e tabelas apresentadas no item 4.2.2.4 podem, à primeira vista, indicar um alto custo computacional, crescente com o aumento do número de nós – O(n). Porém, só são necessários cálculos quando um dispositivo está na iminência de desconexão ou falta de bateria, tornando esporádica sua necessidade. Além disso, a impressão de que o custo computacional é O(n) se mostra infundada, pois só há necessidade de cálculo dos estados de dispositivos vizinhos ao que está com problemas.

Como a quantidade de vizinhos não depende do número de dispositivos, e sim das suas distâncias relativas ao dispositivo com problemas no instante do cálculo, temos que o teste é feito somente para um subconjunto de dispositivos, fazendo com que a complexidade tenda a ser O(1). Tornando-o compatível com os pré-requisitos de baixa necessidade de processamento e consumo de recursos. Ou seja, tornando-o ideal para uso em uma rede ad hoc de dispositivos móveis.

3.3.2.6 Efeito Circular

Uma descoberta interessante feita no decorrer desse trabalho é o efeito circular que pode ocorrer na seguinte situação: sejam dois dispositivos A e B, onde A necessita replicar seus dados e B é escolhido. Em um momento após isso (pequeno o suficiente para que os estados não variem) B sente a necessidade de replicar e escolhe A, que volta a ter a necessidade de replicar e escolhe B, e assim continuamente.

Para evitar o efeito circular, quando um dispositivo A escolher um dispositivo B para replicar seus dados, A deve comparar seu próprio estado a posteriori com o estado a posteriori de B, e só enviar seus dados para B se este estiver em melhor condição. Quando B fizer o mesmo verá que A estará em pior condição e romperá o círculo. Em caso dos dois terem os mesmos estados a posteriori, deverão manter seus dados sem serem replicados.

3.3.3 Implementação e Objetivos Esperados

Implementamos em um simulador as técnicas de tolerância a falhas que utilizam Cópia Reserva e o Método de Replicação Sob Demanda. Além destas, implementamos o Cache Cooperativo como método de economia de energia. Nossos objetivos foram:

- Mensurar a quantidade de dados que ficaram indisponíveis na simulação com o uso de cada uma das técnicas;

- Avaliar o consumo de recursos do sistema principalmente com relação ao consumo de energia;

- Verificar o uso do Cache Cooperativo com relação ao consumo de energia e a leituras em cópias desatualizadas.

4 O PROTÓTIPO E O SIMULADOR

Com o objetivo de verificarmos os conceitos discutidos no capítulo 3 em uma aplicação, construímos um protótipo do SAE onde implementamos algumas de suas funcionalidades básicas. Especificamente, foram incluídos todo o sistema de comunicação, funcionalidades de automatização de ações e propostas de interfaces ágeis, como veremos no item 4.1.

O simulador, descrito no item 4.2, foi elaborado com o objetivo de colher dados que nos permitirão mensurar as técnicas propostas. O simulador interpretará cenários que definirão a posição e as ações – escrita e leitura – de cada dispositivo em cada instante e terá o comportamento definido em cada uma das técnicas propostas, evidenciando suas características.

4.1 O PROTÓTIPO DO SAE

O protótipo foi dividido em subsistemas independentes que se comunicam através de mensagens. Como vimos no capítulo 2, é formado pelo sistema da base fixa, o sistema da base móvel e o sistema do dispositivo móvel portátil, explicados conforme se segue.

O sistema da Base Fixa, foi projetado para oferecer uma visão geral do sinistro e gerenciar o trabalho conjunto das equipes. Nele, temos um conjunto pré-definido de sinistros, que, se acionados, iniciam um conjunto de operações automatizadas relativas ao sinistro escolhido. A partir daí, o sistema suporta a comunicação e o tráfego de dados com as equipes, sendo capaz de gerenciar toda a operação, podendo iniciar, controlar ou solicitar o término de qualquer ação executada pelas equipes.

Na Figura 7 vemos a tela principal do sistema da Base Fixa. Observamos um mapa que irá indicar o local do sinistro quando detectado por um sensor. Quando o operador identificar o tipo de sinistro o sistema irá executar ações automáticas para contenção do mesmo. O painel ao lado do mapa mostrará seqüência de ações já realizadas. Na parte inferior da tela vemos um sistema de comunicação (Chat) com as equipes e um conjunto de ações que podem ser disparadas pelo operador.

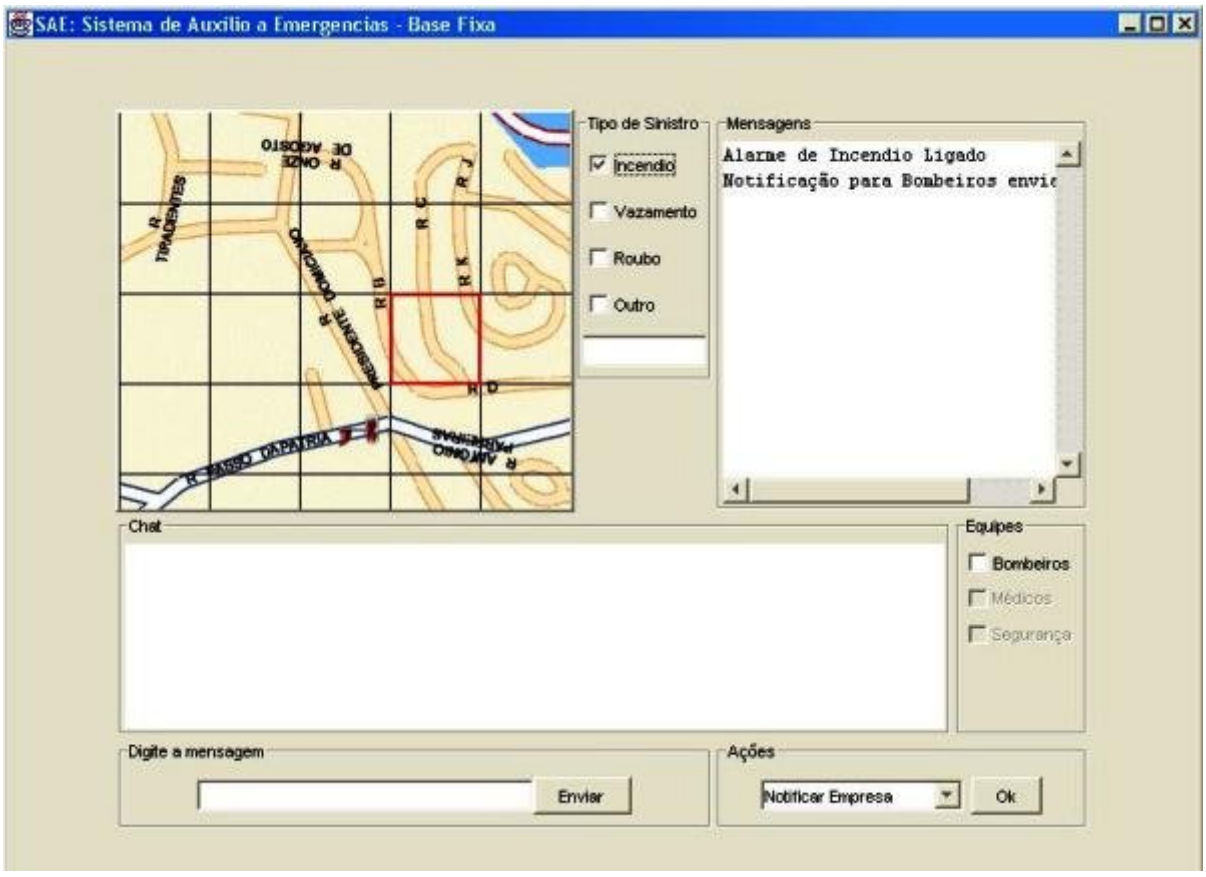

Figura 7: Tela principal da Base Fixa do SAE

Na Figura 8 vemos a Tela principal do sistema gerenciador da Base Móvel. o sistema gerenciador da Base Móvel oferece ao seu usuário uma visão local do sinistro, bem como agiliza a comunicação com os agentes de sua equipe.

À esquerda do mapa temos um local onde ficam listadas as ações que devem ser realizadas pela equipe. E, abaixo, temos um sistema de comunicação que permite o *chat* entre os membros da equipe e um conjunto de ações pré-definidas que podem ser enviadas a qualquer de seus agentes.

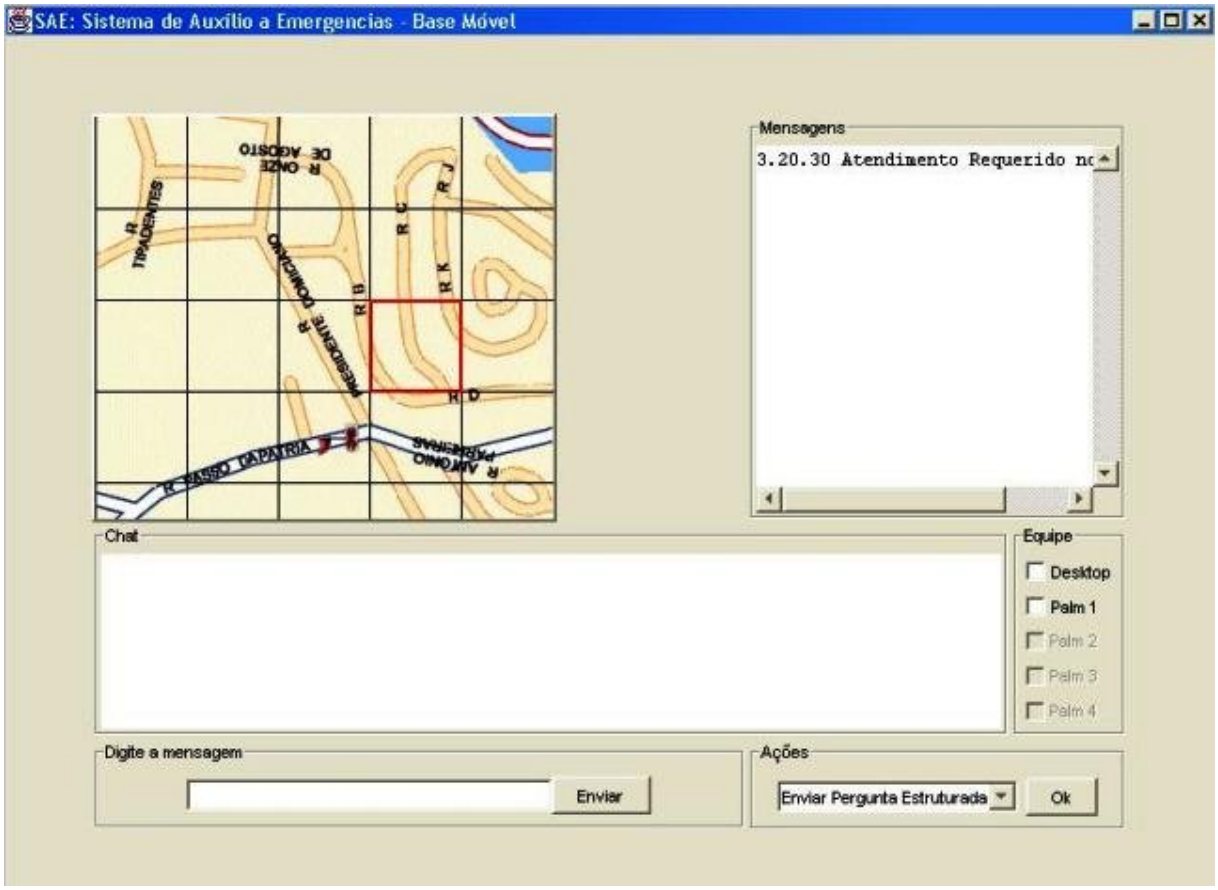

Figura 8: Tela Principal da Base Móvel do SAE

O Sistema desenvolvido para o dispositivo móvel implementa o método de Replicação sob Demanda, para isso verifica periodicamente as condições locais de energia e conectividade. Essas informações são anexadas às mensagens de forma que cada dispositivo pode manter um registro das condições mais atuais dos outros dispositivos com os quais se comunica. A implementação deste modelo, mostrado na Figura 9 necessitou da criação de:

- Uma classe para banco de dados: apta a incluir, excluir e alterar dados sempre que necessário;

- Uma classe para representar o dado;

- A classe de cálculo da probabilidade que, ao ser informada da leitura atual dos valores de energia e conectividade (baseada na intensidade de sinal) de um dispositivo, calcula qual será o seu próximo estado mais provável;

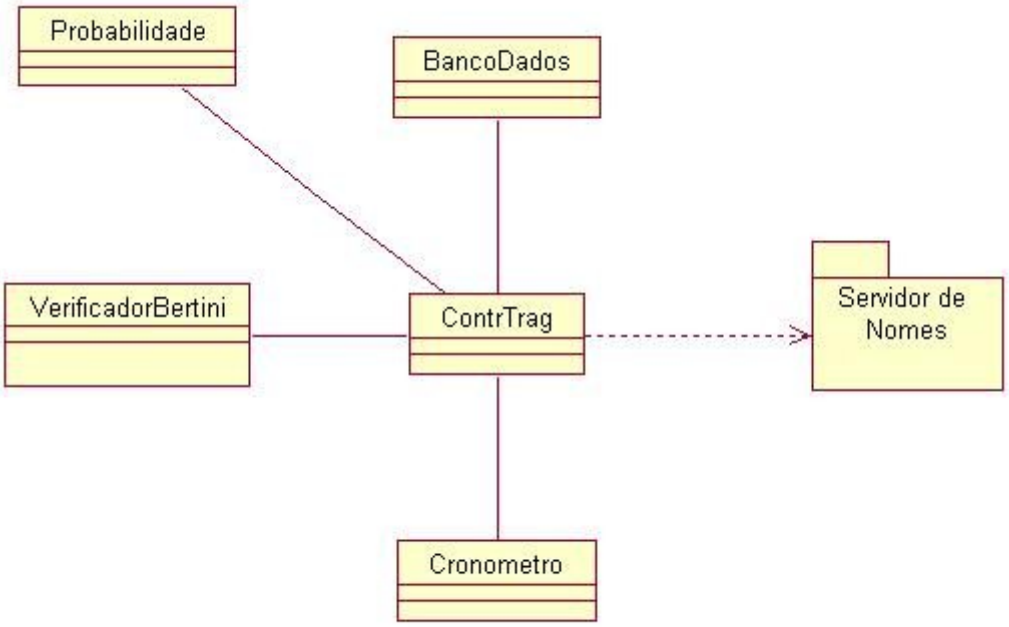

Figura 9: Principais Classes do Sistema do Dispositivo Móvel Portátil

O funcionamento do sistema do dispositivo móvel portátil depende de quatro *threads*:

- A *thread* Principal: Cuida da interface com o usuário. Está na classe ContrTrag;

- A *thread* de Comunicação: responsável por monitorar a chegada de mensagens. O seu trabalho é receber uma mensagem e colocá-la em uma fila para ser tratada pela classe ContrTrag. Esta *thread* é uma classe interna de ContrTrag;

- A *thread* de Monitoramento: periodicamente verifica o estado do dispositivo com relação à sua conectividade e quantidade de bateria. Quando verifica que existe um problema, esta avisa a *thread* Principal, que solicita de seus vizinhos informações sobre seus estados e aciona a *thread* Cronômetro, completando ou atualizando, assim, seu registro de estados de vizinhos. Esta *thread* está na classe VerificadorBertini;

- A *thread* Cronômetro: ao ser acionada, limita o tempo de espera de respostas de vizinhos, quando expira esse tempo, os dados colhidos são submetidos à análise da classe Probabilidade que escolhe, de acordo com o método de Replicação sob Demanda, qual o melhor dispositivo para receber a replicação dos dados. Esta *thread* reside na classe Cronômetro.

O timeline dessa operação é exemplificado na Figura 10.

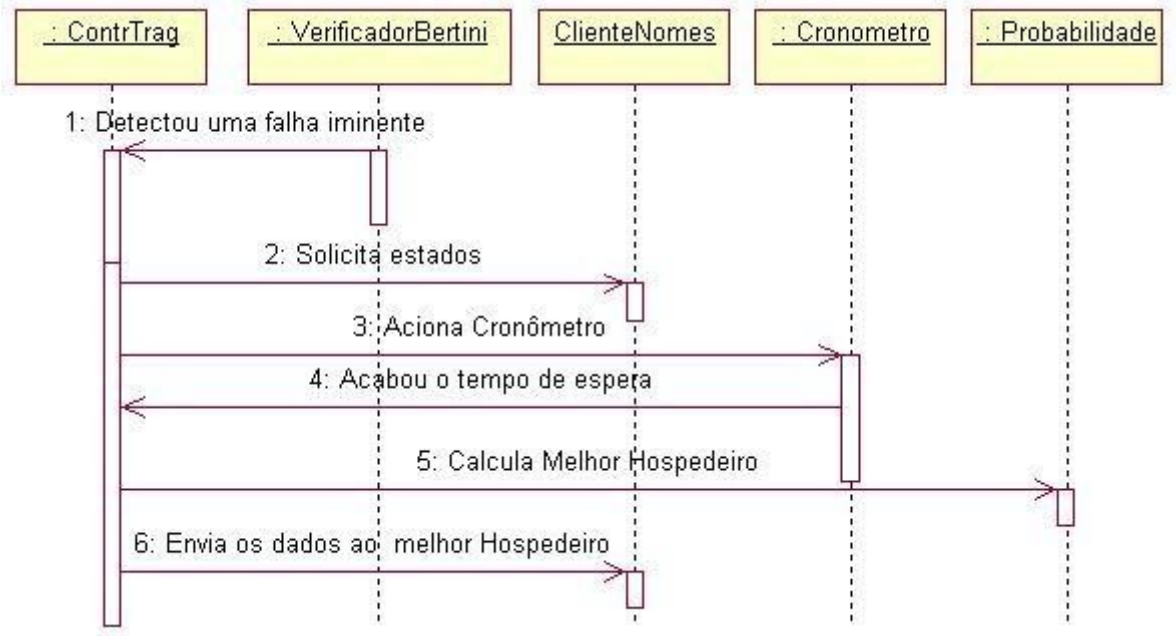

Figura 10: Timeline da descoberta da necessidade de replicação.

Apesar de termos quatro *threads*, somente duas funcionam constantemente, a *thread* Principal e a *thread* de Comunicação. Juntas detêm a maior parte do processamento, sendo que a *thread* Principal por ser a que cuida da interface com o usuário só demanda maior processamento em momentos de interatividade.
A *thread* de Comunicação monitora a rede e, na chegada de mensagens, as põe em uma fila e notifica a *thread* Principal para o devido tratamento.

A *thread* de Verificação funciona a cada período (determinado pelo usuário), verifica duas variáveis e volta a dormir, desta forma seu consumo de recursos é praticamente nulo. E, da mesma forma, a *thread* Cronômetro, funciona somente quando acionada em caso de baixa conectividade ou bateria fraca e, sem quase nenhum consumo de recurso avisa a *thread* Principal que é o momento de definir qual o melhor hospedeiro.

Vemos, desta forma, que o sistema, apesar de complexo, demanda poucos recursos de processamento, qualificando-o para o funcionamento em pequenos dispositivos.

O formato de dados utilizado é bastante simples (Dado em Figura 11), sendo apenas utilizado no protótipo para a verificação prática do modelo de Replicação sob Demanda. Para um projeto real, temos a necessidade de uma estrutura muito mais rica, conforme vemos em Dado1 na Figura 11. Nesta estrutura residirá toda a informação auxiliar necessária às técnicas de replicação utilizadas.

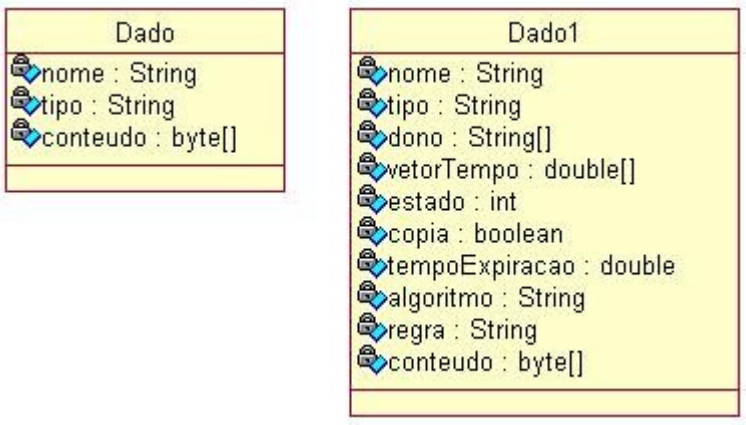

Figura 11: Dados Utilizados

A implementação dos sistemas das Bases Fixa e Móvel foi feita em Java J2SE 1.4.1 o que permite a execução em qualquer sistema operacional que execute a máquina virtual Java. O sistema dos dispositivos móveis portáteis foi, inicialmente, implementado em J2ME utilizando a configuração CLDC por ser a configuração suportada por dispositivos que executam Palm OS.

Essa escolha se mostrou inviável no momento em que a configuração CLDC não oferece acesso a algumas bibliotecas do sistema operacional, impedindo-nos de acessar o estado da bateria e da conectividade. Este problema somente foi resolvido com a troca de configuração para CDC, que também não oferece acesso direto às bibliotecas do sistema operacional, mas nos oferece o Java Native Interface – JNI – que permite acessar bibliotecas escritas em outras linguagens. Assim, foi possível acessar, através de uma função escrita em C, os valores de bateria e conectividade (através da intensidade do sinal) dos dispositivos. Isso obrigou-nos a migrar para dispositivos que executam Windows CE, tais como Pocket PC's.

4.2 O SIMULADOR

O principal objetivo do simulador será, analisando diversos cenários, verificar se a técnica proposta no item 3.1 realmente é eficiente, permitindo o uso do cache cooperativo em trabalhos em grupo realizados por dispositivos móveis como o contexto de ambientes de sinistro. Além disso, o simulador deverá mostrar qual a melhor alternativa para segurança de dados que devemos utilizar, pesando para isso o custo de manutenção dessa segurança em termos de espaço e número de mensagens necessárias, bem como mensurando sua eficiência com relação às falhas na qual serão observados quantos dados ficaram indisponíveis com o uso de cada proposta de segurança que descrevemos nos itens 3.3.1 e 3.3.2.

A simulação se dá através da leitura de um cenário pré-criado que contém informações de posição de cada nó bem como solicitações de escrita e leitura como vemos no Quadro 4. Nele, cada linha é composta pelo tipo de operação (M-movimento, R-leitura e W-escrita) seguidos pelo tempo da simulado e uma seqüência de informações que dependem da operação. Descreveremos detalhadamente a montagem do cenário no item 4.2.1.

```
M 3.00000 0 1057.54 639.29 1 908.28 556.86 2 1069.79 580.35 3 968.04
628.27 4 965.71 533.76 5 1200.44 548.52 6 1241.01 292.16 7 1306.97
553.94 8 1110.89 322.48 9 1023.31 498.57 10 891.73 1077.30 11 944.04
1056.24 12 942.60 1012.70 13 1003.51 1077.68 14 913.29 1154.26 15
1039.45 1224.64 16 930.94 1239.16 17 1036.36 1218.60 18 1081.74
1102.17 19 967.08 1167.09
R 3.0 3 37
W 3.0 2 52
R 3.0 19 33
R 3.0 16 13
W 3.0 1 49
```

Quadro 4: Parte do arquivo de Cenário com uma Movimentação 3 Leituras e 2 escritas

4.2.1 Criação do Cenário

Para que o simulador consiga oferecer dados confiáveis, é necessário que criemos cenários condizentes com a realidade que encontraremos. Dessa forma, definimos como área do sinistro um quadrado de 1500 metros de lado. Neste quadrado, oito equipes com cinco dispositivos móveis cada, trabalharão a uma velocidade mínima e máxima a serem configuradas.

Como essas equipes trabalham como grupos, programamos o ns-2 [22] para que gere movimentos aleatórios para cada dispositivo, porém cada um desses dispositivos respeitará um círculo imaginário que pertence a sua equipe, nunca estando a uma distância maior do que o raio do círculo do ponto central do mesmo.

Cada grupo (representado por um círculo) poderá percorrer toda a área do sinistro conforme mostramos na Figura 12. Esta programação é auxiliada por uma ferramenta chamada Bonnmotion [21] cujo objetivo é criar cenários de movimentação com restrições, como as que utilizamos para fazer com que os dispositivos de cada grupo se locomovam de forma conjunta.

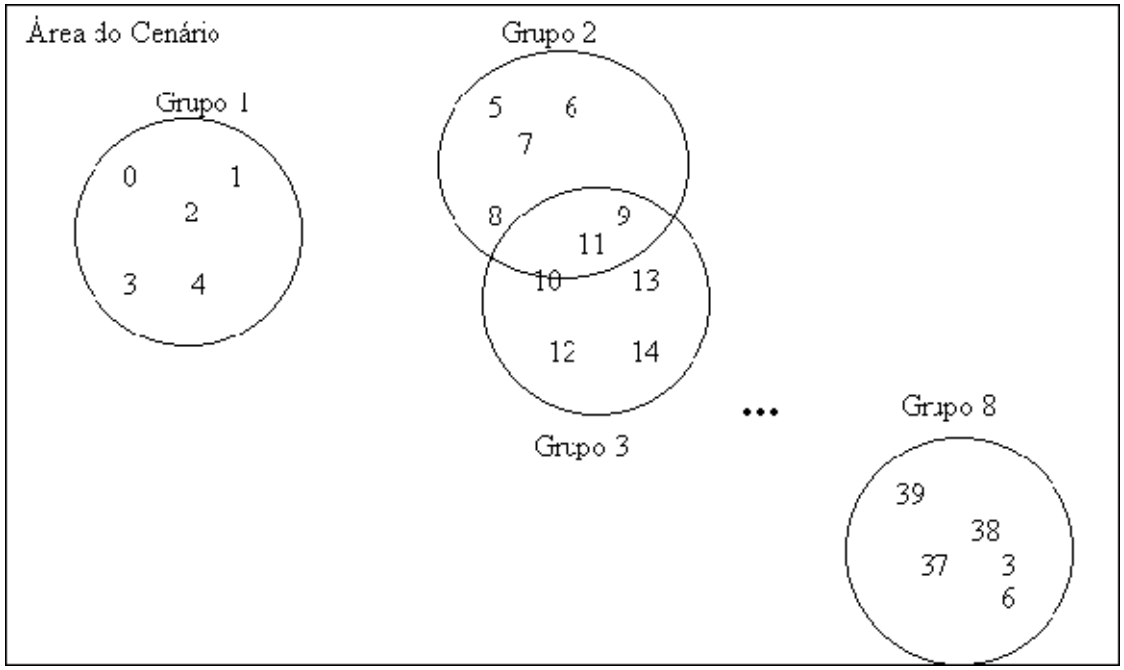

Figura 12: Grupos na área do Sinistro

A Figura 12 mostra os grupos e seus círculos imaginários. Cada dispositivo pertencente a um grupo pode se locomover livremente no interior do círculo imaginário que define o grupo. Este círculo, por sua vez, pode se movimentar por todo o perímetro do sinistro. Assim, cada dispositivo está fisicamente próximo dos outros de seu grupo. Nada impede que os grupos tenham áreas em comum, mas seus movimentos são completamente independentes.

Ao ser executado, o ns-2 gera um arquivo que informa a posição de cada dispositivo em intervalos de tempo de 1 segundo. Depois da geração pelo ns-2 do registro das posições, esse arquivo é processado por um sistema auxiliar que insere comandos de leitura e escrita a serem executados. Esses comandos são criados da seguinte forma:

(a) Para cada dispositivo é feito um sorteio se este irá ou não executar um comando naquele instante.

(b) Se sim, é sorteado um comando – escrita ou leitura.

(c) Se for Escrita é sorteado dentre os dados que o dispositivo pode escrever qual receberá a escrita

(d) Se for leitura é sorteado um dado a ser lido. Sendo que, os dados internos ao grupo têm maior probabilidade de serem escolhidos para mantermos a premissa que os dispositivos necessitam mais de dados locais.

Parte do arquivo resultante desse processamento ao qual chamamos de cenário está representado no quadro 4 (visto no item 4.2).

4.2.2 Simulação do Cache Cooperativo

O mecanismo do cache cooperativo se inicia no momento em que um dispositivo de um grupo tenta ler um dado que não pertence a nenhum dispositivo de seu grupo.

Neste momento, o dispositivo solicita o dado necessário através de uma mensagem em *multicast* para o seu grupo - os dispositivos pertencentes à sua equipe com exceção da base móvel. Como o dado não pertence a nenhum dispositivo de seu grupo, é iniciada uma busca pelo mesmo (as técnicas de busca de um dado em uma rede ad hoc não fazem parte do escopo deste trabalho).

Se o dado for acessível, o dispositivo que iniciou a busca criará localmente uma cópia do mesmo. Esta cópia terá um tempo de vida pré-determinado pelo tipo de dado, na simulação utilizamos 60 segundos como tempo de vida de um DINDIV e de 31.000 segundos – valor maior que a duração da simulação – o tempo de vida dos dados DIMUT. Enquanto este tempo de vida não expirar, qualquer dispositivo de seu grupo poderá acessar o dado diretamente desta cópia, evitando todo o ciclo de busca.

O simulador implementa este modelo utilizando o Algoritmo 1, conforme se segue:

1. Um comando de leitura é lido no cenário;
2. Verifica-se que o dado solicitado não pertence a nenhum dos dispositivos do grupo do leitor;
3. É feita uma busca nos dispositivos do grupo do leitor por uma cópia válida cujo tempo de vida ainda não expirou;
4. Se encontrado, é feita a leitura diretamente nesta cópia;
5. Senão, é feita a leitura no dado localizado no dispositivo remoto e criada uma cópia no dispositivo leitor.

Algoritmo 1: Algoritmo de leitura de um dado utilizando o cache cooperativo

Cada leitura feita através do passo 4 do Algoritmo 1 é contabilizada como uma leitura em cópia. No caso de o dado original já tiver sido alterado no momento dessa leitura, também é contabilizada uma leitura em cópia desatualizada.

É importante ressaltarmos que: o cache cooperativo não influencia nos acessos de escrita.

O consumo de energia do cache cooperativo será descrito no item 4.2.5, onde trataremos do consumo de energia de todas as técnicas simuladas:

4.2.3 Simulação de Tolerância a Falhas Utilizando o Método de Cópia Reserva

A simulação do método de Cópia Reserva requer que, em seu início, o número determinado de cópias reservas seja criado. A partir daí, temos o Algoritmo 2 para leitura e o Algoritmo 3 para escrita.

1. Um comando de leitura é lido no cenário;

2. Se o dado for do leitor é feita a leitura imediatamente;

3. Se o dado pertencer ao grupo do leitor e;

    a. Houver rota entre o leitor e o dono do dado, é feita uma leitura interna no grupo.

    b. Não houver rota entre o leitor e o dono do dado, busca-se uma rota entre o leitor e o hospedeiro de uma cópia reserva.
        i. Se houver, é feita a leitura em cópia reserva no grupo;
        ii. Senão há uma falha na leitura.

4. Se o dado não pertencer ao grupo do leitor;

    a. Verifica-se a existência de uma cópia cache válida no grupo, se houver é feita uma leitura em cópia cache,

    b. Busca-se uma rota entre o leitor e o dono do dado, se houver é feita uma leitura em dado fora do grupo e criada uma cópia cache,

    c. Busca-se uma rota entre o leitor e o hospedeiro de uma cópia reserva, se houver é feita uma leitura em cópia reserva fora do grupo,

    d. Senão há uma falha na leitura

Algoritmo 2: Simulação de uma leitura com Cópia Reserva

É importante ressaltar que, no subitem b.i do item 3 e no subitem c do item 4 do Algoritmo 2, no caso de, no momento da leitura na cópia reserva, o dado já tiver sido modificado, é contabilizada como uma leitura em cópia reserva desatualizada.

1. Um comando de Escrita é lido do Cenário;

2. O dado é alterado com a marcação de novo timestamp;

3. Registra-se que uma alteração foi feita para futuro envio às cópias reservas;

Periodicamente:

1. Verifica-se a existência de uma rota entre o escritor e o hospedeiro da cópia reserva;

   a. Se houver, a cópia reserva é atualizada recebendo o novo timestamp;

   b. Senão, houve uma falha de atualização da cópia resrva;

   O item 1 é repetido para cada cópia reserva existente do dado escrito.

Algoritmo 3: Simulação de uma escrita com Cópia Reserva

4.2.4 Simulação de Tolerância a Falhas Utilizando o Método de Replicação sob Demanda

A simulação utilizando o método de Replicação sob Demanda tem como principal diferencial em relação ao método Cópia Reserva o fato de não existir de prévio uma cópia reserva. Há, no entanto, verificações periódicas do estado de cada um dos dispositivos com o objetivo de localizar possíveis falhas por desconexão ou por falta de bateria. No momento em que um dispositivo detecta a possibilidade de ocorrência de uma falha, este verifica qual dispositivo em sua vizinhança tem a menor probabilidade de falha segundo o método exposto no item 3.3.2 e replica seus dados nele.

Assim, o dispositivo com problemas replica seus dados no dispositivo de menor probabilidade de falha, mantendo localmente uma cópia dos mesmos. No momento em que o motivo da falha é resolvido – recarregada a bateria ou melhora na conectividade – o dispositivo que replicou os dados recupera-os do dispositivo hospedeiro, voltando a sua normalidade.

Os algoritmos de simulação de leitura e escrita do modelo de replicação Sob Demanda são descritos nos algoritmos 4 e 5 respectivamente.

1. Um comando de leitura é lido no cenário;
2. Se o dado for do leitor é feita a leitura imediatamente;
3. Se o dado pertencer ao grupo do leitor e,
    a. houver rota entre o leitor e o dono do dado, é feita uma leitura interna no grupo;
    b. Senão há uma falha na leitura;
4. Se o dado não pertencer ao grupo do leitor,
    a. Verifica-se a existência de uma cópia cache válida no grupo, se houver é feita uma leitura em cópia cache;
    b. Busca-se uma rota entre o leitor e o dono do dado, se houver é feita uma leitura em dado fora do grupo e criada uma cópia cache;
    c. Senão há uma falha na leitura.

Algoritmo 4: Simulação de leitura no método de Replicação sob Demanda

1. Um comando de Escrita é lido do Cenário;
2. O dado é alterado com a marcação de novo timestamp.

Algoritmo 5: Simulação de escrita no Método de Replicação sob Demanda

É importante ressaltar que um comando de escrita não influencia na Replicação Sob Demanda. O dado só poderá ser escrito pelo dispositivo que é seu atual dono. Quando o dado é replicado, o direito de escrita também é transferido para o hospedeiro.

O algoritmo que simula a criação de réplicas Sob Demanda é descrito no algoritmo 6.

Periodicamente para replicação:

1. Cada dispositivo verifica sua condição de Conectividade C e Bateria B;

2. Se C < Cmin ou B < Bmin

    a. Verifica-se a Conectividade e Bateria de todos os vizinhos que possuam rota até o dispositivo em dificuldade;

    b. Calcula a probabilidade de falha de cada Vizinho identificado em a;

    c. O vizinho que possuir menor probabilidade de falha recebe os dados do dispositivo em dificuldade sendo que este fica somente com cópias dos dados.

Periodicamente para recuperação dos dados:

1. Cada dispositivo que tenha replicado seus dados verifica sua condição de Conectividade C e Bateria B dos dispositivos que tiveram seus dados replicados.

2. Se C> Cbase e B> Bbase

    a. Verifica se há rota até o dispositivo que está com os dados

    b. Se sim, recupera os dados, mantendo no hospedeiro somente uma cópia dos mesmos;

    c. Senão faz a verificação no próximo período.

Algoritmo 6: Simulação da replicação no método de Replicação sob Demanda

O algoritmo original descrito em [5] não prevê o processo de recuperação de dados. Por entendemos que um dado deve estar sempre com o seu dono original, a sua replicação só deve ocorrer quando há a necessidade de proteção. Finda essa necessidade, o dado deverá voltar ao seu dono original.

Ressaltamos que:

(a) O simulador obtém diretamente de suas estruturas de dados os valores de conectividade e energia dos dispositivos vizinhos ao dispositivo com dificuldades. Em um sistema real o estado de um dispositivo pode ser anexado às mensagens que este envia, assim cada dispositivo pode manter uma estrutura de dados local que o permite calcular a qualquer momento as probabilidades de cada dispositivo custo mínimo de processamento.

(b) O simulador calcula a conectividade através das distâncias relativas entre os dispositivos. Em um sistema real cada dispositivo pode medir sua conectividade através da potência de sinal recebida.

4.2.5 O Consumo de Energia

A medição do consumo de energia é essencial para determinar se houve aumento do tempo de vida do sistema com o uso das técnicas propostas.

Em [27] são fornecidas estimativas sobre a energia gasta por um dispositivo móvel portátil para enviar e receber mensagens, bem como para se manter ligado (consumo base). A partir desses dados, arbitramos que o consumo de energia de uma operação interna tal como a leitura de memória ou a execução de um programa é de 20% a mais do que o consumo base.

Além disso, para inserirmos no simulador o consumo de energia para o roteamento sem que fiquemos dependentes de qualquer protocolo de roteamento, fizemos um consumo médio de busca e roteamento. Assim, será gasto uma quantidade de energia (50% da energia de transmissão de uma mensagem) por cada dispositivo toda vez que houver necessidade de roteamento, sendo que, quando o roteamento for interno a um grupo, o gasto será em todos os dispositivos do grupo, se for fora do grupo, o gasto será em todos os dispositivos do sistema. A seguir temos a tabela de consumo apresentada em [27].

Tabela 5: Tabela de Consumo

Operação	Consumo (mW)
Transmissão	1400
Recepção	1000
Consumo Base	830
Consumo em Execução	Consumo Base + 166
Roteamento	700

Para melhor esclarecimento, dividiremos o gasto de energia do sistema implementado no simulador nos consumos necessários a cada atividade: funcionamento, leitura, escrita, atualização, verificação da técnica de Replicação sob Demanda e Replicação sob Demanda. Para determinadas ações também dividiremos os consumos em: Necessidade ou não de roteamento (dentro e fora do grupo do dispositivo que executa a atividade). A partir do consumo verificado na tabela 5 e na divisão exposta acima, temos os seguintes consumos:

- Gasto de Funcionamento:
  - A cada segundo o dispositivo consome 830 mW.
- Leitura:
  - No próprio dispositivo leitor: 166 mW (por leitura)
  - Em Dispositivo do Grupo:
    - Sem roteamento:
      - Dispositivo leitor: 1000 mW
      - Dispositivo hospedeiro: 1400 mW
    - Com roteamento
      - Dispositivo leitor: 1000 mW
      - Dispositivo hospedeiro: 1400 mW
      - Todos os dispositivos do Grupo: 700 mW
  - Em Dispositivo Fora do Grupo
    - Sem Roteamento:
      - Dispositivo leitor: 1000 mW
      - Dispositivo hospedeiro: 1400 mW
    - Com Roteamento:
      - Dispositivo leitor: 1000 mW
      - Dispositivo hospedeiro: 1400 mW
      - Todos os Dispositivos: 700 mW
- Escrita:
  - Só é feita em dados no próprio dispositivo: 166 mW
- Atualização (somente de cópias reservas e estas estão, por definição, no grupo)

- Sem roteamento:
  - Dispositivo leitor: 1000 mW
  - Dispositivo hospedeiro: 1400 mW
- Com roteamento
  - Dispositivo leitor: 1000 mW
  - Dispositivo hospedeiro: 1400 mW
  - Todos os dispositivos do Grupo: 700 mW

o Verificação da técnica de Replicação sob Demanda:
- A cada verificação: 166 mW

o Replicação sob Demanda:
- A cada Replicação realizada
  - Dispositivo na iminência de Falha: 1400 mW
  - Dispositivo escolhido para ser o hospedeiro: 1000 mW

É importante notarmos que o consumo de energia de recepção e de transmissão não varia com o local onde se encontra o dado. A variação encontra-se na necessidade ou não de roteamento e se este roteamento é feito entre dispositivos do mesmo grupo ou dispositivos pertencentes a outros grupos. No mesmo grupo, o consumo de energia para o roteamento ocorre somente nos dispositivos do grupo, enquanto que no roteamento que envolve outros grupos, o consumo de energia ocorre em todos os dispositivos.

Com o exposto acima e analisando o funcionamento do cache cooperativo, torna-se claro o motivo pelo qual ele gera economia, pois, no momento em que fazemos uma leitura interna a um grupo no lugar de uma leitura externa a um grupo, consumimos energia somente dos dispositivos do grupo e não dos dispositivos de todo o sistema.

5 TESTES E RESULTADOS

O objetivo dos testes a seguir será de analisar o desempenho das técnicas propostas. Os pontos principais a serem verificados são: *(a)* aumento do tempo de vida do sistema; e *(b)* a disponibilidade dos dados.

Para melhor medirmos a eficiência do modelo, não utilizaremos dados com mais de um dono – DCOMPART – evitando assim que mensagens rotineiras de sincronização confundam os resultados obtidos. Desta forma, somente dados DIMUT e DINDIV estarão presentes nos testes.

As simulações serão feitas em uma área de 1500 m X 1500 m e em um período de 30.000s, além disso, temos como principais configurações:

(a) Um total de 40 nós divididos em 8 grupos de 5 nós. Cada nó se movimenta em direção velocidade aleatória variando entre 3 e 9 m/s;

(b) O círculo de atuação de cada equipe possui 200 m de raio;

(c) Um total de 80 dados divididos em 40 dados DIMUT, 40 DINDIV;

(d) Tempo de Expiração de uma cópia = 60 s;

(e) Total de Energia inicial de 40.000 J;

(f) Tempo de recarga = 60 s.

Para que tenhamos dados independentes de um único cenário, as simulações foram realizadas utilizando de cinco a quinze cenários com movimentações aleatórias (mas com as mesmas configurações acima. Na tabela 6 vemos os números obtidos para a avaliação do cache cooperativo.

Tabela 6: Números das simulações

	Cenários					
	1	2	3	4	5	6
Total de Leituras em Dados Externos	157631	157410	158569	158728	159683	158314
Leituras nos Hospedeiros dos Dados	87567	87838	88555	88019	88481	87866
Leituras em Cópias Cache	47548	47381	47643	48033	48414	47811
Leituras em Cópias Cache Desatualizadas	22516	22191	22371	22676	22788	22637
	Cenários					
	7	8	9	10	11	12
Total de Leituras em Dados Externos	159909	158246	156419	156493	160261	157949
Leituras nos Hospedeiros dos Dados	88769	87852	87230	86701	88554	88386
Leituras em Cópias Cache	48290	47857	47047	47444	48625	47288
Leituras em Cópias Cache Desatualizadas	22850	22537	22142	22348	23082	22275
	Cenários			Resultados		
	13	14	15	Média	Desv. Padrão	%
Total de Leituras em Dados Externos	158376	157495	158305	158253	1076,8	0,68
Leituras nos Hospedeiros dos Dados	88214	87531	88439	88000	552,53	0,62
Leituras em Cópias Cache	47611	47571	47603	47744	420,76	0,88
Leituras em Cópias Cache Desatualizadas	22551	22393	22263	22508	255,45	1,13

Podemos observar que o desvio padrão obtido dos quinze cenários é muito pequeno, indo de 0,68% a 1,13% da média aritmética. Esse valor de desvio padrão não chega a 6% da média aritmética na maioria das experiências. Assim, utilizaremos essa média como o resultado de cada experiência.

5.1 AVALIAÇÃO DO CACHE COOPERATIVO

Inicialmente, iremos avaliar o Cache Cooperativo. Mostraremos que este consegue evitar a necessidade de busca e roteamento de dados, o que diminui o consumo de energia. E ainda, mostraremos que a quantidade de leituras desatualizadas é, como esperado, relacionada com o tempo de vida da cópia cache. Neste experimento não utilizaremos nenhum método de tolerância a falhas.

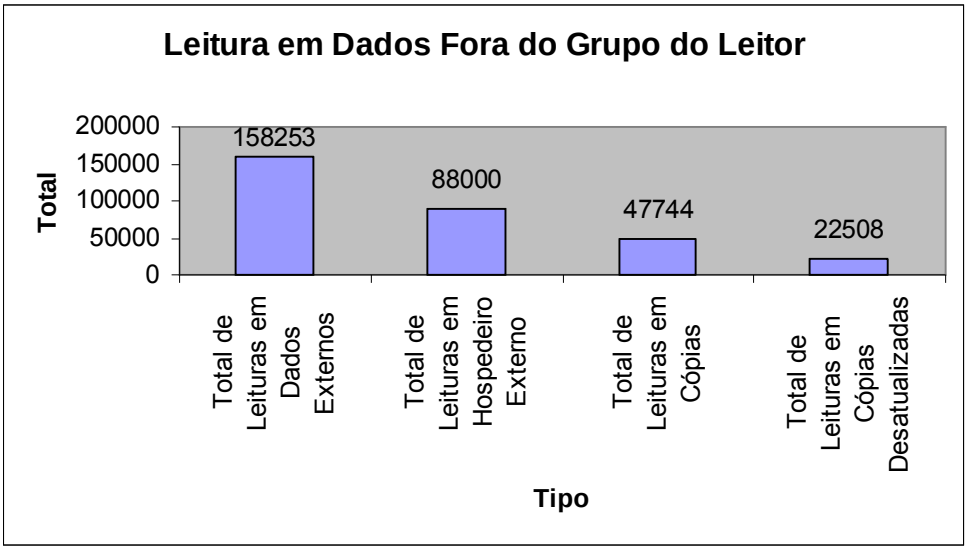

Figura 13: Gráfico Uso do Cache Cooperativo

Vemos na Figura 13 que foram realizados cerca de 160.000 leituras em dados residentes fora do grupo do leitor. Destas, mais de 70.000 (47.744 + 22.508) puderam ser realizadas em cópias cache locais, o que minimizou sensivelmente a necessidade de roteamento. Verificamos, porém, a existência de leituras realizadas em cópias desatualizadas – 22.508 – ou seja, nas quais o dado original já foi modificado. A Figura 14 nos mostra a proporção dessas leituras.

É importante ressaltarmos que as simulações realizadas objetivaram representar um sistema muito utilizado, onde os dispositivos participam intensivamente da mitigação do sinistro colhendo dados de sensores, trocando informações gerenciais, ou sendo utilizado para comunicação. Por isso obtivemos um grande número de leituras e escritas nas simulações.

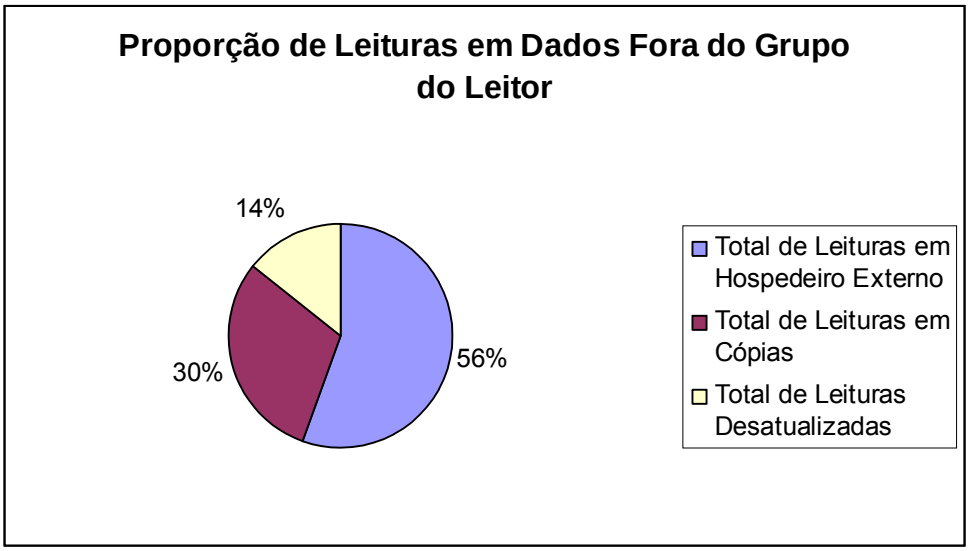

Figura 14: Gráfico da Porcentagem de Dados Lidos na Cópia Cache e da Porcentagem de Leituras Desatualizadas

Como vimos nas Figuras 13 e 14, cerca de 44% das leituras em dados localizados fora do grupo do leitor puderam ser feitas no cache cooperativo. Porém, verificamos a ocorrência de 14% de leituras em cópias desatualizadas. A Figura 15 mostra como essa porcentagem aumenta quando aumentamos o tempo de vida da cópia.

Baseando-nos na Figura 15, vemos que o tempo de vida de uma cópia cache não deve ser longo, sob o risco de esta estar desatualizada quando for lida. Contudo, alguns dados podem ter grande tolerância à leitura desatualizada, permitindo-nos a troca da confiabilidade pela eficiência.

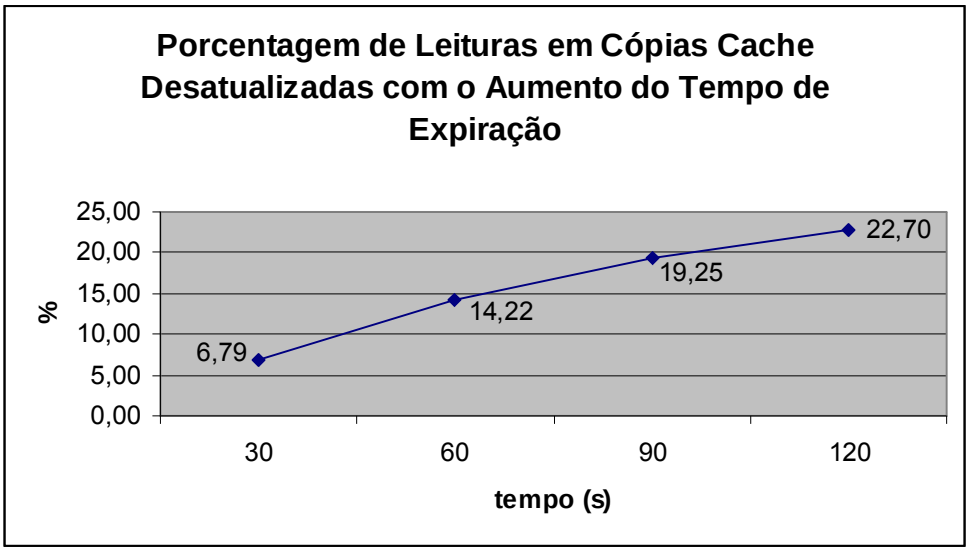

Figura 15: Gráfico da Evolução da Porcentagem de Leituras Desatualizadas com a Variação do Tempo de Vida da Cópia.

5.2 AVALIAÇÃO DO MODELO DE TOLERÂNCIA A FALHAS

Para identificarmos qual o melhor modelo de tolerância a falhas devemos considerar as restrições de recursos características de um sistema baseado em pequenos dispositivos e da porcentagem de erros que puderam ser evitados. Assim, devemos atentar para os seguintes definições e testes:

5.2.1 Dados Indisponíveis

Um dado é considerado indisponível quando está em um ou mais dispositivos inalcançáveis, seja por estarem sem bateria, seja por estarem fora de alcance de qualquer outro dispositivo. Esta situação é extremamente indesejável em um ambiente adverso como o de mitigação de um sinistro. Para tentar evitar a ocorrência dessas situações foram sugeridas duas técnicas:

(a) Manter todos os dados replicados (método de cópia reserva), onde esperamos que não haja falha do dono e do hospedeiro ao mesmo tempo. Simularemos essa técnica utilizando uma e duas cópias reservas; e

(b) Criar uma réplica dos dados (método de Replicação Sob Demanda) quando o dispositivo se considerar na iminência de uma falha. Nesse caso, o dispositivo repassa o dado a outro que esteja em melhores condições de energia e/ou conectividade.

A seguir, temos as comparações entre essas técnicas e, como balizador, temos a medição da não utilização de nenhuma técnica de segurança.

A verificação se um dado está ou não indisponível acontece quando um dispositivo se desconecta ou fica sem bateria. Neste instante, o simulador verifica a existência de cópias reservas de todos os dados desse dispositivo. Se uma cópia reserva for achada em um dispositivo conectado e com bateria o dado continua disponível, senão, o dado é contabilizado como indisponível.

Na Figura 16 temos a quantidade de dados indisponíveis em função do método utilizado.

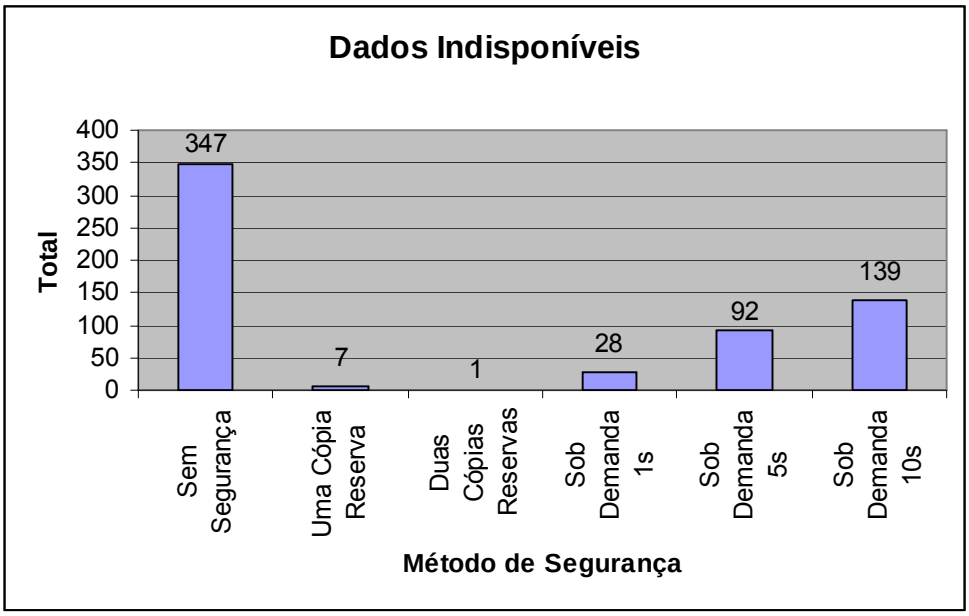

Figura 16: Representação Gráfica dos Dados Indisponíveis.

Quando nos referimos ao método de Replicação Sob Demanda utilizamos o período no qual o dispositivo verifica suas condições de energia e conectividade. Assim, Sob Demanda 1s significa a utilização do método onde o dispositivo verifica seu estado a cada 1 segundo.

Todos os métodos de tolerância a falhas conseguiram reduzir o número de dados indisponíveis. No método Cópia Reserva, como esperado, vemos que quanto mais cópias existem mais eficiente é o método. Com relação à Replicação Sob Demanda, observamos que também é muito boa, reduzindo em cerca de 92,0% o total de dados indisponíveis. Vemos, entretanto, que quanto maior o período de verificação do método menor sua eficiência. Porém, para analisarmos a eficácia do método devemos mensurar o consumo de recursos de cada método que será tratado a seguir.

5.2.2 Quantidade de Mensagens Necessárias à Técnica

Em se tratando de pequenos dispositivos, todos os recursos são escassos e devem ser utilizados com o máximo de eficiência, especialmente quando estão sendo utilizados em um ambiente hostil cuja reposição pode ser difícil ou até mesmo impossível. Desta forma, a quantidade de mensagens necessárias para que o modelo proposto funcione corretamente é de suma importância na medida em que seu envio consome energia. Na Figura 17 mensuramos o total de mensagens necessárias para o funcionamento de cada modelo de tolerância a falhas.

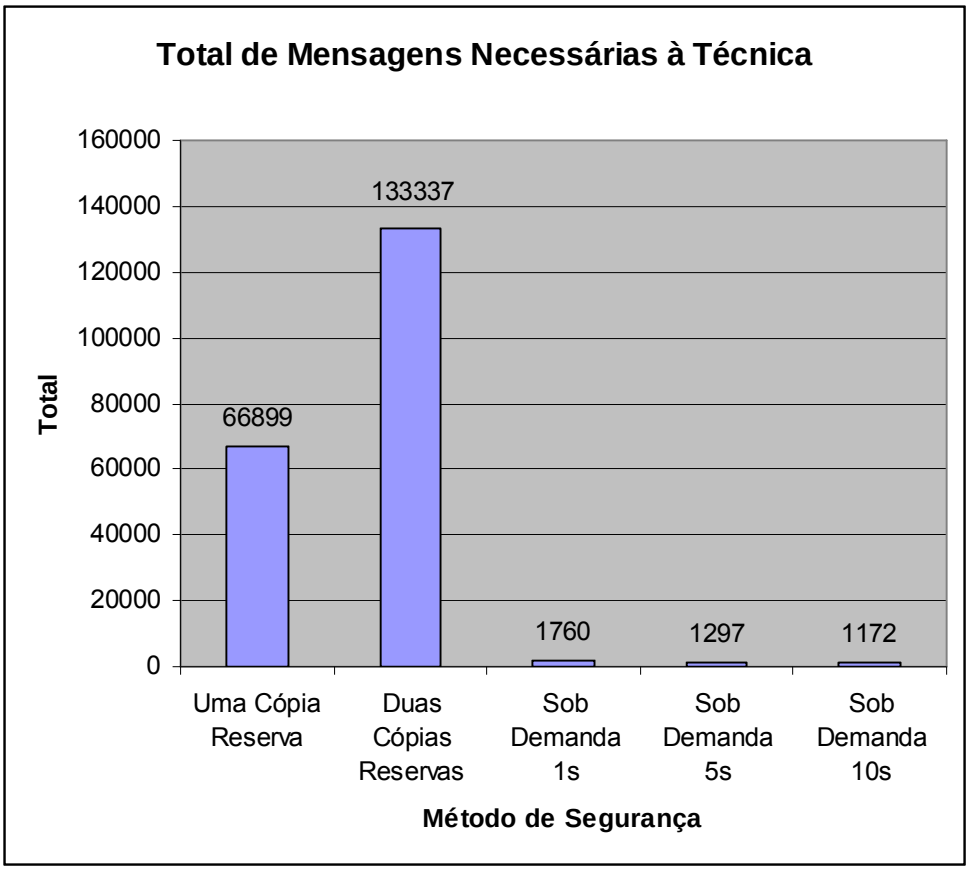

Figura 17: Gráfico da Quantidade de Mensagens para o Funcionamento do Modelo de Tolerância a Falhas.

É evidente a necessidade maior de mensagens no método que utiliza Cópia Reserva. Isso se dá porque a cada atualização do dado, a cópia reserva precisa ser atualizada. A figura 17 mostra que com o uso de uma cópia reserva foram necessárias quase 67.000 mensagens de atualização, número que praticamente dobrou quando foram utilizadas duas cópias reservas. Isso se dá devido à necessidade de atualizar as duas cópias. Não foi obtido exatamente o dobro porque nem sempre é possível enviar a mensagem de atualização pois pode não haver rota entre o dono do dado e algum hospedeiro ou um deles estar sem energia.

Já no método de Replicação sob Demanda, só existem mensagens quando o dado é replicado e quando é recuperado, otimizando, assim, a utilização dos recursos gastos com segurança.

Podemos observar um pequeno decréscimo de mensagens quando aumentamos o período de atualização do método de Replicação Sob Demanda. Isso ocorre porque quanto maior o tempo entre duas verificações de estado maior a possibilidade de haver uma grande mudança no estado do dispositivo. Por exemplo: o dispositivo em uma primeira verificação observa seu estado de conectividade e ele está bom. Por causa do período grande de verificação o dispositivo ao verificar novamente já não tem mais vizinho. O que o impossibilita de replicar seus dados. Isso, por um lado, leva a uma diminuição das mensagens, mas por outro, afeta a capacidade do método em reduzir o número de falhas como vemos na figura 16.

5.2.3 Gasto de Energia e Tempo de Vida

Para estudarmos o gasto de energia, medimos o total de vezes que os dispositivos ficaram sem energia no tempo de simulação. Conforme descrito na configuração, ao ficar sem energia, o dispositivo leva 60 segundos para ser recarregado. Este tempo de recarga representa a estimativa do tempo necessário para que um agente troque a bateria do seu dispositivo.

Essa hipótese de recarga é justificada pelo fato de que, no momento em que o dispositivo assume um papel importante no contexto da gerência do sinistro, é razoável considerar que os agentes levem consigo baterias reservas para mantê-lo funcionando o maior tempo possível. Como o número de baterias reservas em uma situação real não é infinito, toda técnica que economize energia é importante para o sistema. A seguir, a comparação do consumo de cada técnica.

A Figura 18 mostra o total de falhas por falta de energia. As simulações foram realizadas com e sem o uso do cache cooperativo. Para balizarmos os resultados foram realizadas, também, simulações sem nenhum método de segurança. Essas simulações nos permitiram comparar o ganho obtido com a técnica do Cache Cooperativo em relação ao sistema sem qualquer método de economia e, também, nos permitiu verificar o custo, em energia, que cada modelo de tolerância a falhas insere no sistema.

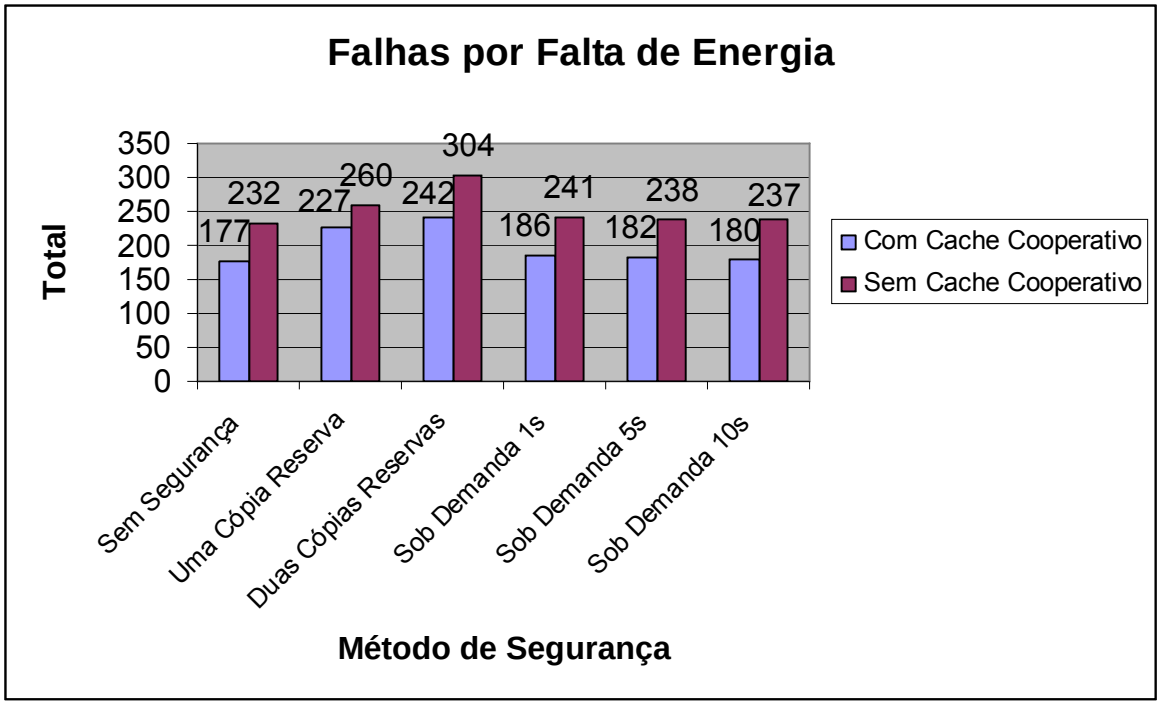

Figura 18: Gráfico das Falhas por Falta de Energia

Quando comparamos o modelo sem segurança e sem cópia cache com o modelo sem segurança com cópia cache (Figura 18), verificamos uma diminuição de 23,7% nas falhas por falta de energia. Quando introduzimos o modelo de tolerância a falhas que utiliza uma cópia reserva, verificamos que o consumo de energia causado pelo envio das mensagens necessárias à manutenção da consistência do modelo aumentaram as falhas em 12,1% sem o uso do cache cooperativo e 28,2% com uso do cache cooperativo . Esse consumo, como o esperado, amplia-se quando são utilizadas duas cópias reserva, fazendo o número de falhas aumentar em 31,0% e 36,7% respectivamente.

Em seguida, verificamos que o método Replicação sob Demanda implica em um ligeiro aumento – entre 1,7% e 5,0% com cache cooperativo e entre 2,1% e 3,9% sem cache cooperativo – de falhas por falta de energia, evidenciando o pequeno consumo de energia necessário para o método. Além disso, observamos que o período de verificação do método de Replicação sob Demanda praticamente não influi no total de faltas por consumir muito pouca energia em sua verificação periódica, esse consumo só acontece quando há necessidade de replicação.

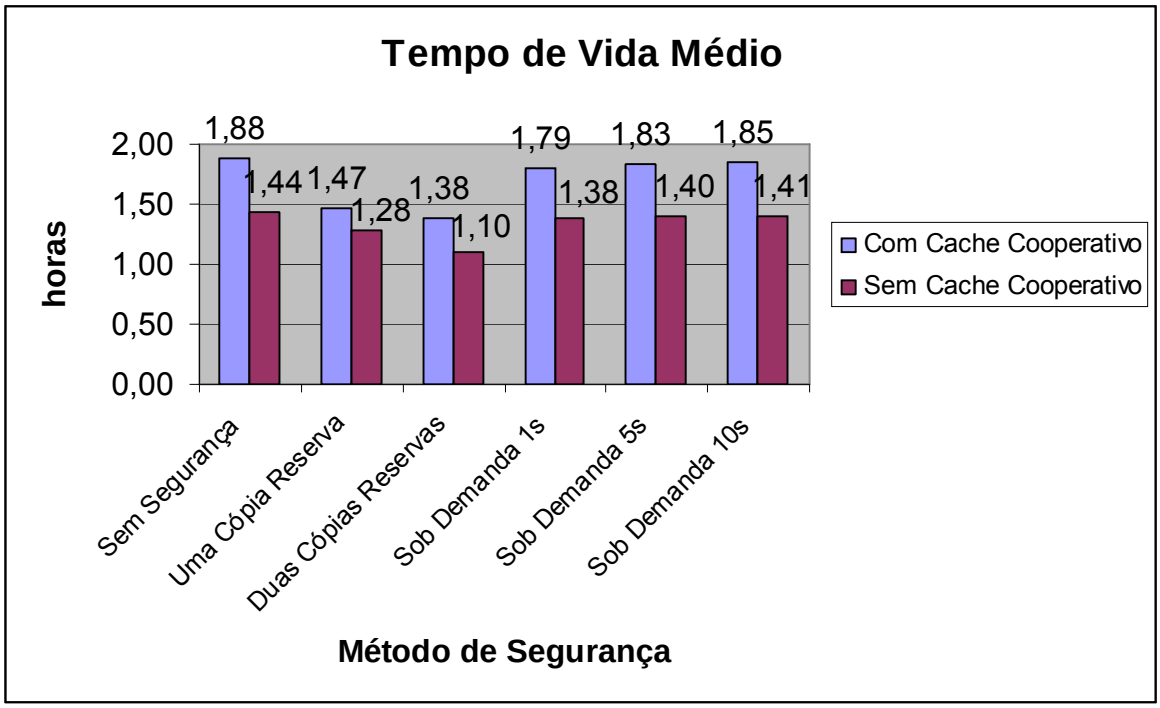

Figura 19: Gráfico do Tempo de Vida Médio dos Dispositivos

A Figura 19 exibe o tempo de vida médio dos dispositivos. Sem qualquer método de segurança ou economia vemos que um dispositivo funciona, em média, por 1,44 horas. No momento em que introduzimos o Cache Cooperativo, vemos esse tempo subir para 1,88 horas, um ganho de 30,6%. Este ganho é obtido porque, ao encontrar o dado que será lido em um dispositivo de seu grupo (sob forma de uma cópia cache) o dispositivo leitor não necessita trazê-lo de um dispositivo distante. Minimizando, assim, a necessidade de roteamento.

Quando utilizamos o modelo de tolerância a falhas que utiliza cópia reserva observamos uma grande queda no tempo de vida médio devido às mensagens necessárias a manutenção da consistência das cópias e, na Replicação sob Demanda, vemos que o tempo de vida médio é pouco afetado.

Por esse experimento, podemos concluir que o uso do método de Replicação sob Demanda em conjunto com o Cache Cooperativo nos oferece tolerância a falhas com baixo consumo de energia, sendo, então, o mais indicado para a maioria dos dados em uso no SAE [4].

É importante observar que dados que não toleram leituras desatualizadas, no entanto, não deverão usar o método de cache cooperativo. Da mesma forma, dados que exigem uma segurança extrema deverão utilizar o método de cópia reserva com, no mínimo, duas cópias reserva em dois hospedeiros distintos.

5.2.4 Outros Testes

Além de quantidade de dados indisponíveis e de quantidade de mensagens necessárias a cada técnica, é importante que analisemos o comportamento dos modelos em relação à densidade de dispositivos utilizados e à velocidade a qual esses dispositivos são submetidos.

5.2.4.1 Densidade de Dispositivos

Neste teste vamos observar o comportamento de cada modelo no momento em que variamos a quantidade de dispositivos em relação a uma mesma área, ou seja, variaremos o número de dispositivos por metro quadrado (dispositivos/m2) que irão atuar em um sinistro. Essa quantidade é relevante pelo fato de influir na conectividade de cada dispositivo, sendo que, quanto mais dispositivos estiverem em uma região, maiores serão suas conectividades.

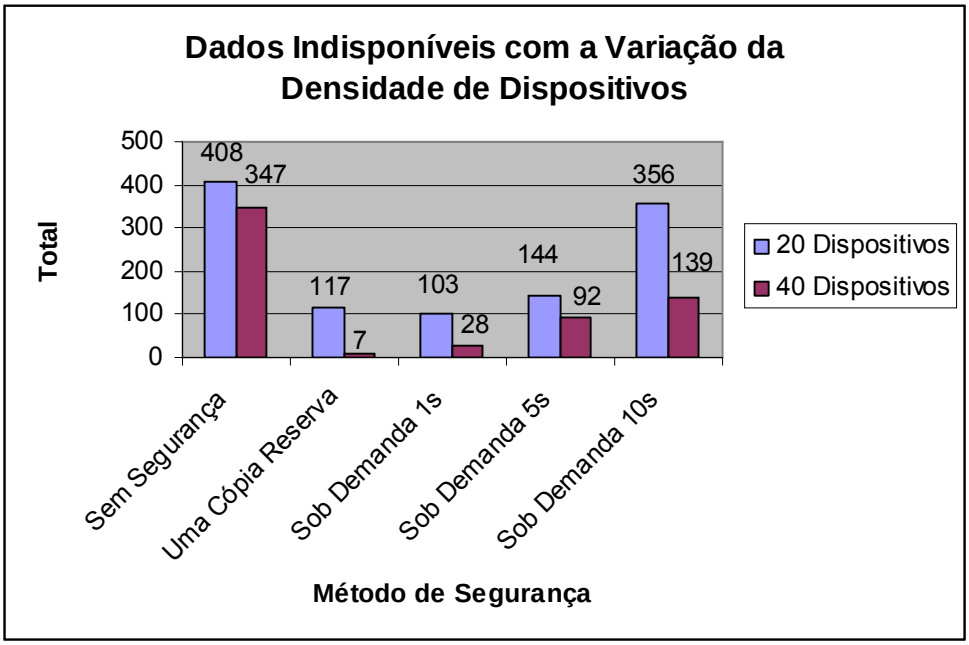

Figura 20: Gráfico dos Dados Indisponíveis Quando da Variação da Quantidade de Dispositivos

Verifica-se na Figura 20 que há uma melhora considerável em todas as técnicas quando aumentamos a densidade de dispositivos. Isso porque, quando elevamos esta densidade (mais dispositivos trabalhando em uma mesma área), a conectividade do grupo aumenta, assim temos menos perdas por desconexão.

5.2.4.2 Velocidade dos Dispositivos

Verificaremos neste teste como cada modelo se comporta quando os dispositivos que compõem os grupos são submetidos a variações de velocidade.

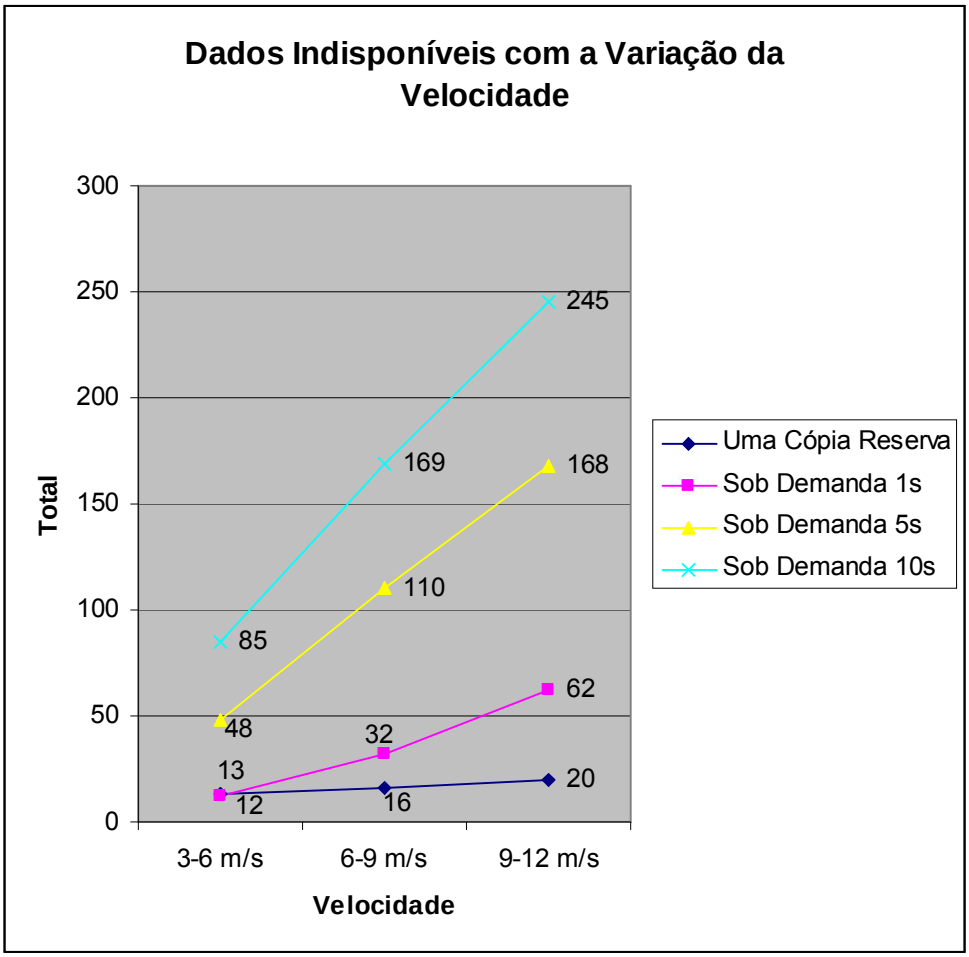

Figura 21: Gráfico dos Dados Indisponíveis Quando da Variação da Velocidade dos Dispositivos.

Vemos na Figura 21 que, quanto maior a velocidade mais dados ficam indisponíveis. É bom notar que este crescimento é menos acentuado quando diminuímos o período de verificação do método de Replicação sob Demanda. Isto acontece porque os dispositivos conseguem realizar a replicação a tempo.

Como o método de Replicação sob Demanda utiliza poucos recursos para ser implementado e o aumento de velocidade de deslocamento só começa a ser um problema quando as velocidades passam a ser muito altas, o que geralmente não tem como acontecer em um ambiente de sinistro onde as equipes, enquanto trabalham, estão, geralmente, a pé. Assim, em nossa opinião, o método continua sendo mais eficaz para um sistema que necessita economizar recursos.

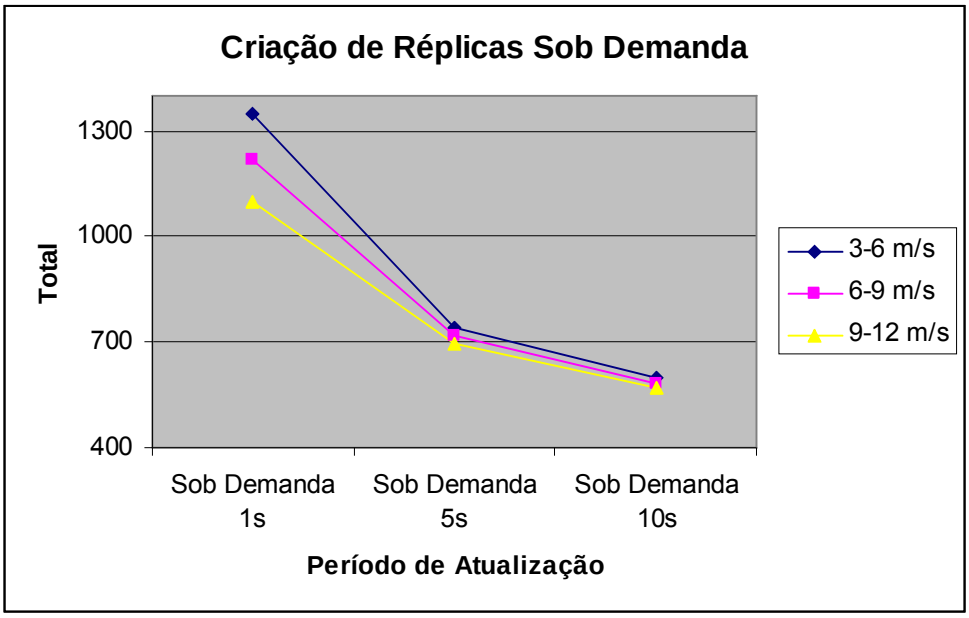

Figura 22: Gráfico da Criação de Réplicas Sob Demanda Quando da Variação da Velocidade.

A Figura 22 mostra que o aumento do período de verificação leva a uma menor quantidade de replicações com conseqüente piora na eficiência do método.

Analisadas em conjunto, a Figura 21 e a 22 mostram que o método de Replicação Sob Demanda é muito sensível à velocidade de deslocamento dos nós. Por outro lado, vemos que quando diminuímos o período em que cada dispositivo verifica seu estado há uma melhora da eficiência do método. Assim, para mantermos a eficiência da técnica devemos diminuir o período de verificação do estado sempre que for necessário aumentar a velocidade de deslocamento dos dispositivos.

5.2.4.3 Uso do SAE Sem Recarga de Bateria

Em todos os experimentos realizados anteriormente, o dispositivo é recarregado após 60 segundos sem energia, simulando, desta forma, uma troca de baterias. A figura 23 mostra o comportamento do sistema sem recarga, ou seja, o sistema funciona até que o último dispositivo fique sem energia.

Medimos o tempo desde o início da simulação até que o primeiro dispositivo fique sem energia, até que metade dos dispositivos fique sem energia e até que todos os dispositivos fiquem sem energia.

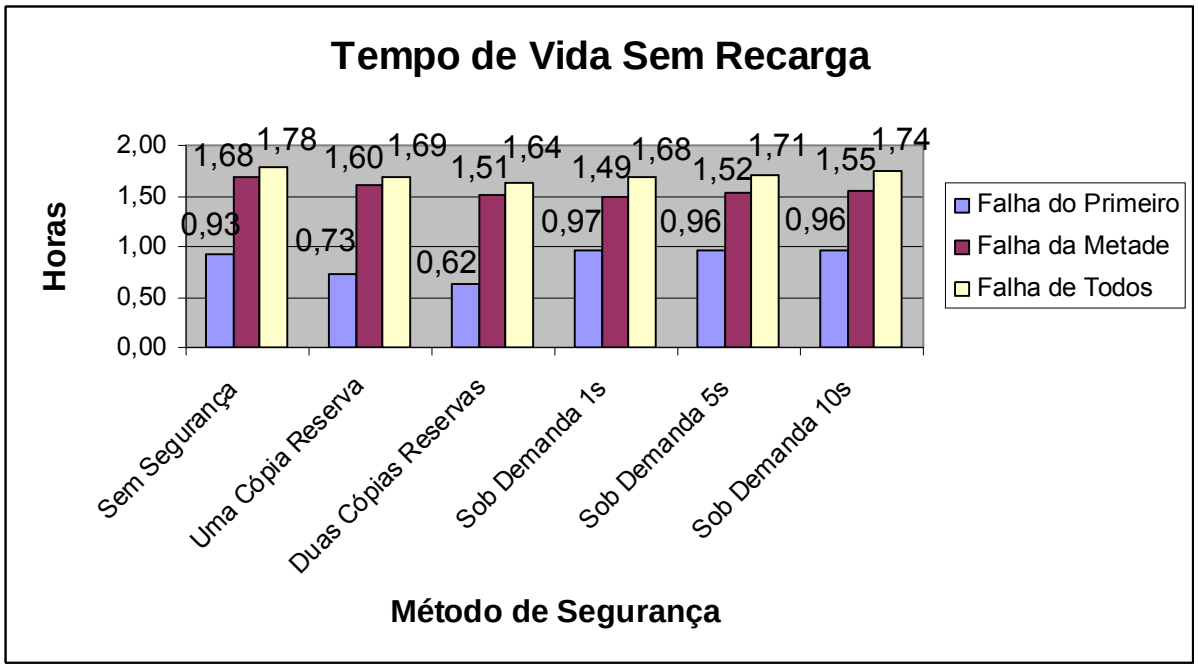

Figura 23: Gráfico de Tempo de Vida dos sistemas sem recarga dos dispositivos

Podemos observar que o tempo de vida do sistema com todos os dispositivos funcionando, ou seja, até a falha do primeiro dispositivo, é cerca de 32,9% maior utilizando o método de Replicação Sob Demanda (verificação a cada 1 segundo) do que quando utilizamos uma cópia reserva e de 56,5% comparado com o uso de duas cópias reservas.

Verificamos também que na medida em que os dispositivos começam a falhar os métodos que utilizam uma ou duas cópias reservas ganham uma sobrevida. Isso é observado porque quando um dispositivo falha tanto as cópias reservas nele hospedadas param de receber atualização quanto ele próprio para de mandar atualizações para as cópias reservas de seus dados. Com essa diminuição de mensagens de atualização há um menor consumo de energia levando a uma maior sobrevida do sistema. Opostamente, observamos que a diminuição da energia leva a um maior número de replicações sob Demanda. Esses fatos acabam por aproximar o tempo de vida total dos sistemas que utilizam as duas técnicas.

Porém, consideramos que o método de Replicação Sob Demanda ainda assim é muito superior porque enquanto o sistema com o método de cópia reserva vai perdendo seus dados com as falhas dos dispositivos, o sistema que utiliza o método de Replicação Sob Demanda procura manter os dados acessíveis até que todos os dispositivos falhem.

6 TRABALHOS RELACIONADOS

Os trabalhos a seguir foram utilizados como base para o desenvolvimento do SAE e permitiram um melhor entendimento do contexto utilizado nesta dissertação.

6.1 PROJETO RESCUE: DESAFIOS NA RESPOSTA AO INESPERADO

O Projeto RESCUE [19] tem como objetivo transformar radicalmente a habilidade dos agentes de colher, armazenar, analisar, interpretar, compartilhar e disseminar dados. Utiliza para isso, tecnologia de redes sem fios, sistemas distribuídos, bancos de dados relacionais, processamento de imagem e vídeo. Tem como premissa que potencializar a velocidade e a acuidade com que as informações sobre a crise viajam através das redes dos agentes responsáveis pela mitigação irá revolucionar o atendimento aos sinistros salvado vidas e patrimônios.

6.1.1 O Processo de Mitigação do Sinistro

Atividades de resposta a sinistros incluem medidas de proteção à vida e à propriedade imediatamente antes – quando possível – durante e imediatamente após o desastre. Essas atividades vão de poucas horas até dias ou mesmo meses, podendo envolver uma operação em larga escala de diversas organizações. Essas entidades devem trabalhar juntas em uma crise como uma única organização virtual com o objetivo de salvar vidas, preservar a infra-estrutura e re-estabelecer a normalidade.

O ciclo de resposta a uma crise envolve as seguintes fases:

- Levantamento dos Danos: são identificadas as perdas diretamente relacionadas com o desastre, sua magnitude e possíveis danos secundários que poderão advir de uma não mitigação imediata de um dano. Além disso, são identificados os problemas de maior urgência e estimados os tempos necessários para restaurar a normalidade;

- Levantamento de Necessidades: nesta fase os incidentes que requerem algum nível de resposta são identificados. Por exemplo, um prédio desmoronado, onde vítimas podem estar presas, é incluído nas operações necessárias de busca e salvamento;

- Priorização das Medidas de Salvamento: aqui, os incidentes identificados na fase, anterior são priorizados e os recursos são distribuídos. Neste momento, os tomadores de decisões devem possuir um ótimo levantamento de necessidades que os permita estabelecer as prioridades de alocação de recursos;

- Resposta Organizacional: neste momento, os recursos estão distribuídos e as decisões organizacionais são disseminadas entre os agentes responsáveis pela mitigação segundo o planejamento.

6.1.2 O Projeto RESCUE

O projeto RESCUE trabalha o fluxo de informações conectando as equipes envolvidas no sinistro. A informação, inicialmente dispersa na hierarquia de coleta é armazenada e analisada sendo, em seguida, distribuída aos tomadores de decisões nas formas mais apropriadas para suas tarefas.

Os principais trabalhos de tecnologia de informação no RESCUE são:

- Coleta de Informação: Coletar de diferentes fontes os dados relevantes de forma eficiente e transmiti-los através de uma rede insegura para bancos de dados distribuídos;

- Análise da Informação: extração de informação rica em contexto dos dados coletados tornando-os úteis na realização das tarefas necessárias;

- Compartilhamento da Informação: A disseminação correta da informação através das organizações, tornando o trabalho colaborativo mais eficiente. Neste caso, os grandes desafios encontrados são a da grande heterogeneidade das organizações e dados, bem como a definição de quais são as informações necessárias a cada organização.

6.1.3 Relação com o Presente Trabalho

Verificamos no Projeto Rescue que há toda uma infra-estrutura para a utilização de dispositivos portáteis, principalmente no que diz respeito à coleta, compartilhamento e disseminação de dados, podendo este se beneficiar com o uso das técnicas aqui propostas.

6.2 ISARS (*INTERACTIVE SEARCH AND RESCUE SYSTEM*): INTEGRAÇÃO DE DADOS GEO-ESPACIAIS E INFORMAÇÃO PARA A BUSCA E SALVAMENTO COSTEIRA E MARINHA [20]

O acesso a dados geográficos é vital para operações de Busca e Salvamento. Esses dados idealmente deveriam:

- Ser disponíveis em formato digital;
- Utilizar a Internet e sistemas GIS (*Global Information Systems*);
- Ter fontes de informações em várias escalas, principalmente locais;
- Fazer uso de tecnologia móvel.

A busca e salvamento marítimos e costeiros requerem acesso a informações de diferentes fontes para poder responder devidamente a um incidente. Além disso, seu ambiente de trabalho deve ser entendido dentro de um contexto apropriado, focando-se no fato que não é fechado nem estático, onde elementos de influência como as condições meteorológicas ou o tráfego, por exemplo, estão em constante mudança. Assim, o sistema de busca e salvamento deve capturar todos esses fatores estruturando um banco de dados para uso durante a emergência.

Para diminuir o “inesperado” ou o “imprevisto”, devem ser utilizados padrões onde os cenários são antecipados.

Os desafios atuais para prever além das conseqüências imediatas dos sinistros são:

- Pequeno conhecimento de onde encontrar a informação necessária à resposta ao sinistro;
- Acesso lento às informações necessárias;
- Baixa integração dos sistemas de informação.

Para a busca e salvamento, o formato mais lógico é a organização geo-espacial dos dados.

O objetivo do iSARS é facilitar o acesso *on-line* a dados geo-espaciais.

O projeto deverá:

- Entender e catalogar as necessidades dos usuários;
- Identificar as características dos dados e informações requeridas pelo sistema;
- Testar o hardware e o software utilizados em operações de busca e salvamento;

- Desenhar interfaces rápidas e intuitivas.

Componentes do iSARS:

- iSARS Client: é a face do sistema. Caberá a ele apresentar as funcionalidades aos clientes em várias plataformas;

- iSARS Catalogue: oferecerá uma busca otimizada pelos dados relevantes;

- iSARS Storage: coletará e armazenará os dados em bancos de dados relacionais;

- iSARS Data Upload, Access and Manipulation: é o cliente do banco de dados que permite o envio de dados bem como sua manipulação.

O estágio final desse trabalho permitirá uma combinação de base de trabalho, banco de dados e interface com o usuário para *desktop* e ambientes móveis. Envolverá grande numero de estudos de casos abrangendo cenários de mitigação de crises.

6.2.1 Relação com o Presente Trabalho

O I-SARS prevê a utilização de ambientes móveis para a mitigação de sinistros. Seu propósito é oferecer um conjunto de dados e funcionalidades para facilitação dos trabalhos de busca e salvamento. Não há uma referência à segurança de dados ou tolerância a falhas no ambiente móvel. Este poderá utilizar a proposta tratada neste trabalho.

6.3 UM *FRAMEWORK* PARA O DOMÍNIO DE BUSCA E SALVAMENTO [2]

O objetivo do projeto SARPA era, inicialmente, automatizar as tarefas de busca e salvamento. A evolução do projeto o encaminhou para, em vez de automatizar tarefas, assistir ao usuário na execução de múltiplas pequenas atividades.

Foi feita uma profunda análise no processo de trabalho para desenvolver um ambiente que reconstrói o utilizado por um coordenador, preservando assim, a mesma flexibilidade que ele tem quando coordena sem auxílio do sistema.

A equipe de busca e salvamento é dividida em RCC (*Rescue Coordination Central* ou Central de Coordenação do Resgate) e a Equipe de Campo. O RCC efetua o planejamento, determina o procedimento e é o responsável por controlar a operação. As informações imprecisas e o dinamismo dos dados tornam o sistema de controle bem complexo. Este é formado por dois módulos principais:

- Controle Operacional: decide o que fazer a seguir e avalia o que foi feito;

- Módulo de Tarefas Primitivas: conjunto de pequenas tarefas frutos da decomposição de tarefas complexas.

6.3.1 Controle de Operações

Continuamente decide o que será feito a seguir e faz uma auditoria de controle na evolução da missão. A missão é composta por sessões e estas compostas por um conjunto de tarefas primitivas. A cada missão realizada, toda informação colhida é passada ao Controle de Operações para análise e registro.

6.3.2 Tarefas Primitivas

Foram identificadas mais de cinqüenta tarefas primitivas no domínio da busca e salvamento. A maioria é administrativa que precisa ser completada em um determinado período de tempo seguindo procedimentos estritos. Requisição para o uso de um avião de outra organização é um exemplo.

6.3.3 O Ambiente e a Arquitetura

Uma parte importante da análise foi construir uma classificação das missões e o desenvolvimento de uma estrutura básica de cada missão contendo um subconjunto de tarefas a serem executadas e em que ordem isso deve acontecer. O coordenador, de posse disso, pode automatizar uma tarefa, executá-la manualmente ou até modificar seus resultados. Na Figura 24 temos a arquitetura do SARPA.

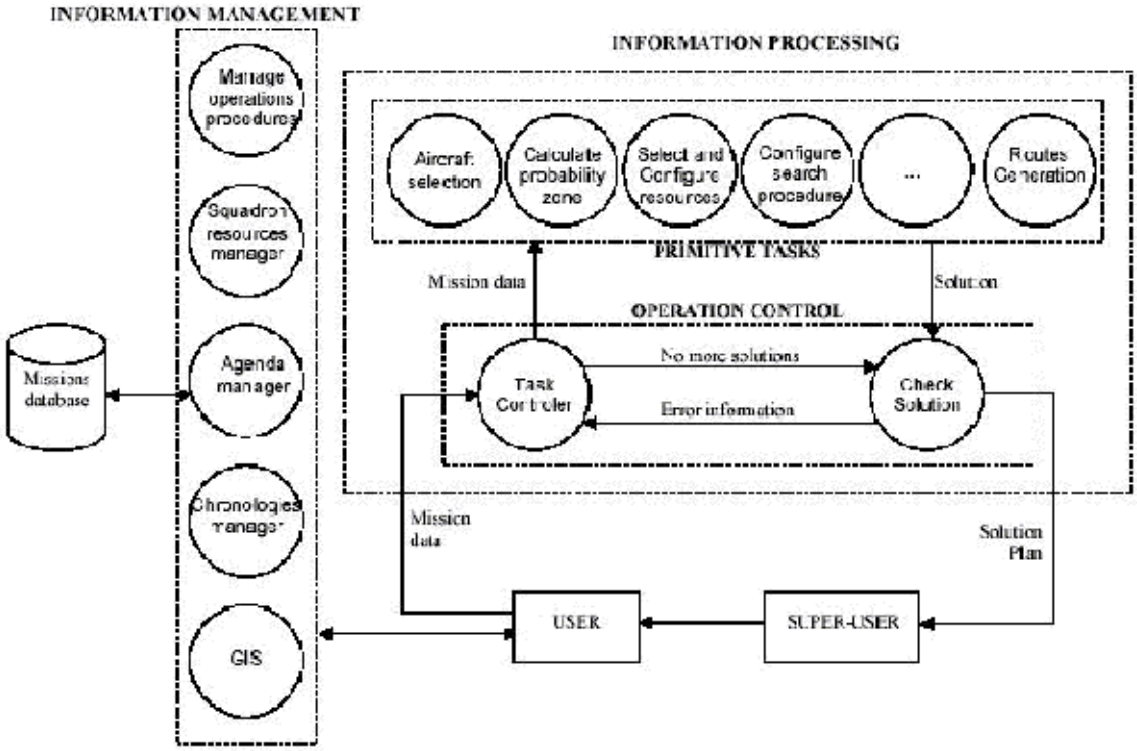

Figura 24: Arquitetura do SARPA

6.3.4 Processamento de Informações

O processamento de informações é responsável tanto pelo planejamento quanto pelas decisões. Feito pelo controle de operações, é responsável por construir e manter a estrutura da missão corrente. A cada nova missão o ambiente é inicializado com um conjunto de atividades, sendo que o usuário pode modificar o ambiente, incluindo novas atividades ou excluindo algumas existentes. Na Figura 25 temos a interface gráfica do SARPA.

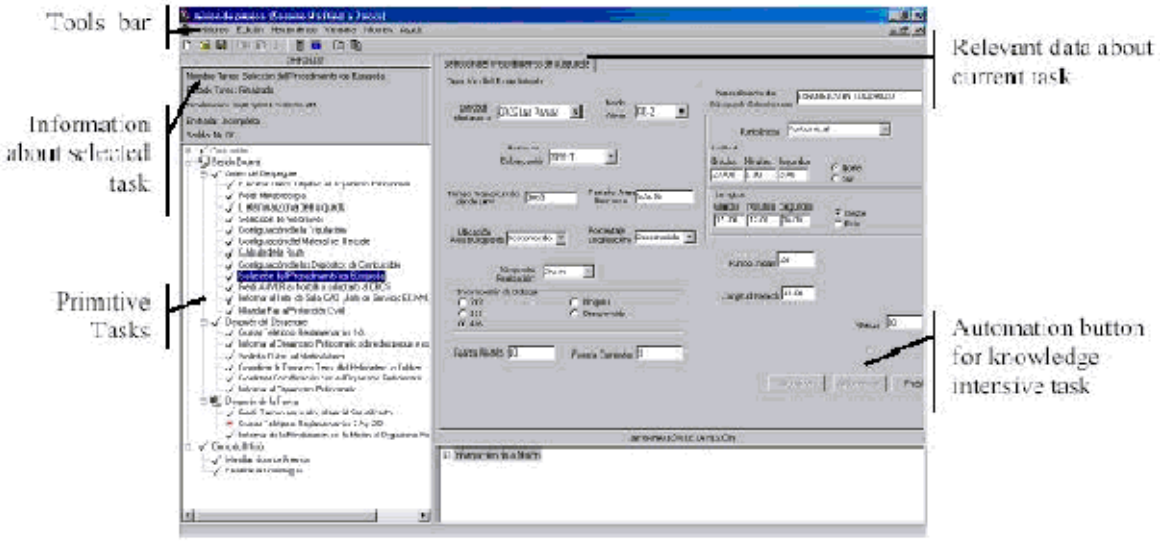

Figura 25: Interface gráfica do SARPA

6.3.5 Gerenciamento de Informações

Nesse módulo temos:

- Gerenciador de procedimentos de operação: é onde são definidos os tipos de missão e seu conjunto de tarefas primitivas;

- Gerenciador de recursos: permite a criação de novos recursos ou a modificação de suas propriedades;

- Agenda de contatos e entidades: compila toda a informação necessária sobre organização e funcionários, acelerando a localização dos contatos mais adequados;

- Gerenciador de relatórios cronológicos: gera todos os documentos associados à missão na ordem que devem ser analisados;

- Sistema de informação geográfica: fornece referências visuais aos tomadores de decisão; integra informações cartográficas em diferentes escalas e inclui pontos relevantes, como hospitais, aeroportos, entre outros.

6.3.6 Gerenciador de Conhecimento

Esta ferramenta captura o conhecimento da organização necessária para executar e controlar operações de busca e salvamento. Provê ao usuário uma lista de ações e atividades que devem ser executadas durante a operação. É importante salientar que o SARPA não constrói um plano global por si só ou substitui o ser humano. Simplesmente provê um conjunto de ferramentas capazes de facilitar a realização das tarefas necessárias.

6.3.7 Relação com o Presente Trabalho

Este é o trabalho mais similar à infraestrutura do SAE. Ele divide o trabalho de busca e salvamento em microtarefas e cada tarefa necessária à busca e salvamento passa a ser um conjunto dessas microtarefas. O SAE amplia esta idéia oferecendo aos agentes de campo dispositivos móveis para facilitar toda a operação. Todo o trabalho discutido nesta dissertação poderá ser empregado em uma possível integração do SARPA com dispositivos móveis.

6.4 I-RESCUE: UM SISTEMA BASEADO EM COALIZÃO PARA SUPORTAR OPERAÇÕES DE REPARAÇÃO DE DESASTRES

Um projeto I-RESCUE [18] tem por objetivo integrar agentes (humanos ou software), para a realização de um trabalho cooperativo. Um sistema de suporte a coalizão é composto de uma estrutura hierárquica de comando e controle, onde os agentes tomam decisões diferentes (e complementares) em cada nível da hierarquia.

Pode ser feito de um modo misto em que humanos tomam decisões com base em experiências e, o *software* gera e compara um grande número de opções mostrando os pontos negativos e positivos de cada uma.

O I-RESCUE identifica três níveis de decisão:

- Estratégica: o que fazer;
- Operacional: quem vai fazer;
- Tática: como vai fazer.

É utilizada a ontologia INCA:

- ISSUES: representa os requerimentos potenciais;
- NODES: representa atividades no processo de planejamento;
- CONSTRAINS: representa restrições temporais, seqüências, etc;
- ANNOTATION: insere informações complementares.

6.4.1 Componentes do Sistema de Coalizão

O processo de suporte a coalizão pode ser ancorado em um grafo, como na Figura 26, o qual permite modelar suas especificações.

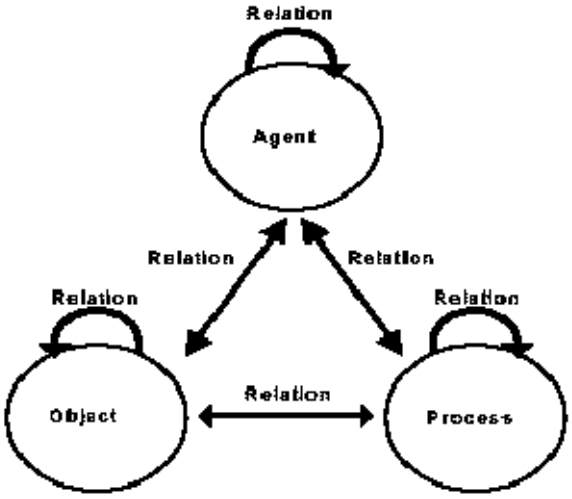

Figura 26: Modelo abstrato de design do sistema de suporte a coalizão.

São componentes do sistema de coalizão:

- Agentes: podem ser descritos por sua capacidade de resolver problemas e suas limitações;
- Objetos: são componentes que formam o domínio (um prédio, por exemplo);
- Processos: especificam um conjunto de atividades que deve ser executado pelos agentes.

Além dos componentes, o sistema identifica as possíveis relações entre estes. São elas:

- Agente-agente: a relação descreve interações entre agentes dentro da organização;
- Objeto-objeto: descreve ligações entre os objetos, por exemplo, as conexões entre ruas;

- Processos-processos: especifica restrições de ligação entre os diferentes processos, mostrando se podem ser executados em paralelo ou seqüencialmente;

- Agente-objeto: captura o objetivo do agente com relação ao objeto;

- Agente-processo: relaciona as responsabilidades de um agente em um processo;

- Processo-objeto: mostra o status da mudança que um processo causa em um objeto; é usado para o acompanhamento da evolução do plano.

O projeto I-RESCUE tem como objetivo dar assistências aos agentes, organizando-os em uma estrutura hierárquica de comando e controle. Oferece a visão dos recursos e necessidades de um sinistro, permite o acompanhamento do estado de cada objeto e, por conseguinte, a evolução da mitigação.

6.5 INTEGRAÇÃO AUTOMATIZADA DE SERVIÇOS PARA GERENCIAMENTO DE CRISES

Agências de emergência criam vários planos de ação para diferentes cenários de desastre. O projeto S-CITI pretende prover uma estrutura robusta de suporte de coleta de dados e tomada de decisão antes, durante ou depois de emergências ou desastres. Aqui discutimos a representação dos planos de ação e coordenação como fluxogramas, integrando agentes e objetivos.

6.5.1 Modelo de Fluxograma

Os planos de ações podem ser representados como um conjunto de tarefas e relações, permitindo a clara especificação do serviço. Esse modelo inclui:

- Pré e pós-condições: especifica o que precisa ser verdadeiro antes da tarefa executada e o que será verdadeiro após sua execução;

- Ligações causais: mostram a seqüência de tarefas;

- Parâmetros de entrada e saída: carregam informação para o fluxograma, como exemplo, o endereço é um parâmetro de entrada em uma tarefa “vá-para-lugar”;

- Restrições temporais: identificam o período de início de duração de cada execução.

- Restrições de recursos: especificam o equipamento, o material e os agentes necessários para a execução de uma tarefa;

- Importância: identifica tarefas vitais que necessitam serem executadas em contraste com as secundárias, que só serão executadas se houver oportunidade.

O modelo de fluxograma nos permite inferir as seguintes propriedades:

- Corretude: é correto se não houver conflitos de restrições ou de tempo, bem como todas as pré-condições e pós-condições forem atendidos na ordem;

- Completude: é completo quando especifica todas as tarefas necessárias para atingir seus objetivos;

- Compatibilidade: é compatível com outro se respectivamente nenhuma das tarefas estiverem em conflito.

Centros de comando deverão gerenciar seus fluxogramas, monitorando o progresso da execução de mitigação do sinistro. Esta monitoração poderá ser feita através de notificações enviadas pelos agentes, que serão replicadas com revisões, se necessário.

7 RESULTADOS, CONCLUSÕES E TRABALHOS FUTUROS

Como vimos, alterações climáticas, o crescimento populacional em direção à áreas mais propensas à desastres, mudanças geo-políticas que incitam cada vez mais guerras e terrorismo, dentre outros fatores aumentaram a probabilidade de ocorrência e o potencial destrutivo dos sinistros.

Por outro lado, a tecnologia de previsão e mitigação de sinistros também evoluiu. No capítulo 6 mostramos sistemas cada vez mais sofisticados que visam organizar e agilizar essa mitigação tornando-a mais eficiente, o que diminui o número de vítimas fatais e as perdas materiais.

No capítulo 2 mostramos um exemplo de como é possível utilizar pequenos dispositivos móveis para tornar ainda mais eficiente a gerência de equipes de emergência, oferecendo processamento de dados e melhor fluxo de informações no local atingido. Afirmamos, porém, que para haver essa utilização diversos desafios devem ser equacionados e resolvidos, dentre eles a escassez de recursos e a propensão a falhas.

Em seguida propusemos a utilização do cache cooperativo para economia de energia e comparamos as técnicas de tolerância a falhas – Cópia Reserva e Replicação Sob Demanda – que aumentam a confiabilidade do sistema. A seguir teremos as conclusões das experiências desenvolvidas no capítulo 5.

7.1 CACHE COOPERATIVO

O cache cooperativo se mostrou muito eficiente em sua proposta de minimizar o tráfego na rede por oferecer uma alternativa de acesso próxima ao dispositivo leitor. A economia é expressiva o suficiente para que seja tolerado o pequeno índice de incorreções introduzido no sistema. Como vimos, a queda da quantidade de leituras desatualizadas quando diminuímos o tempo de vida da cópia também é expressiva. Isso nos permite utilizar a técnica em uma grande variedade de dados, bastando, para isso, ajustes finos em seus tempos de vida.

7.2 TÉCNICAS DE TOLERÂNCIA A FALHAS

Na Figura 16, verificamos que o uso de uma cópia reserva reduziu drasticamente a indisponibilidade dos dados e quando utilizamos duas cópias reserva as falhas praticamente acabaram. Isso se dá porque um dado fica indisponível por falta de energia ou desconexão do seu hospedeiro e, ao mesmo tempo, não há uma cópia reserva em outro hospedeiro ativo e conectado. No momento em que utilizamos o método de Replicação sob Demanda, verificamos que as falhas também diminuíram consideravelmente. O método só não é tão eficiente quanto o de Cópia Reserva pelo fato de que nem sempre existe um dispositivo ao alcance para fazer a replicação.

Quando aumentamos o número de dispositivos presentes na simulação, verificamos uma melhora em todas as técnicas. Isso se dá porque, na técnica que utiliza Cópias Reservas, há uma diminuição na probabilidade de desconexão de cada dispositivo o que implica em uma diminuição ainda maior da probabilidade de que o dono do dado e o hospedeiro da cópia reserva se desconectem ao mesmo tempo. Com relação à Replicação Sob Demanda, o número maior de dispositivos em uma mesma área aumenta as chances de existir um dispositivo ao alcance para hospedar os dados quando houver a necessidade de replicação.

Ao analisarmos os modelos sob o ponto de vista do total de mensagens necessárias para seu funcionamento, vimos que o uso de cópias reservas demanda grande quantidade de mensagens para a manutenção da consistência entre as réplicas e o dado, como mostrado na Figura 17. Já no método de Replicação sob Demanda [5], somente quando um dispositivo corre o risco de desconexão ou falta de energia, há necessidade de envio de mensagens para a replicação do dado, minimizando o gasto de energia e uso da rede.

Para compararmos os dois modelos, devemos fazê-lo à luz dos paradigmas de sistemas móveis com replicação otimista [17], ou seja, devemos considerar que os sistemas devem ser preparados para suportar erros, informações desatualizadas, além de conviver com escassez de recursos de bateria, memória e rede. Com base

nesses princípios, apesar de não ser tão eficiente para evitar falhas quanto o modelo de utilização de Cópia Reserva, o método de Replicação sob Demanda permite uma enorme economia de recursos (1,3% das mensagens usadas pelo outro modelo comparando o uso de duas cópias reservas com a Replicação sob demanda com uma verificação a cada 1 segundo). Sua eficiência foi de 92,0% quando comparada ao cenário sem segurança, ou seja, evitou 92,0% das falhas (com verificação a cada um segundo). Esse ponto – evitou 92,0% das falhas com uso de 1,3% das mensagens – é bastante relevante e foi fundamental para nos levar a concluir que é o melhor modelo a ser utilizado na maioria dos casos.

7.3 CONCLUSÕES

Com base nos resultados apresentados, podemos afirmar que esse trabalho obteve êxito na sua proposição inicial de comparar as técnicas de tolerância a falhas e analisar o Cache Cooperativo como modelo para economia de energia. Além disso, foi possível demonstrar que a combinação do Cache Cooperativo com o método de Replicação sob Demanda de tolerância a falhas permitiu um significativo ganho de tempo de vida do sistema, bem como uma maior disponibilidade dos dados.

Foi demonstrado que o consumo do método de Replicação sob Demanda é muito pequeno, tendo um impacto mínimo no tempo de vida do sistema e com baixa exigência de processamento, podendo, assim, ser empregado em dispositivos móveis portáteis com pequena capacidade de processamento.

Vimos também que, ao tratarmos de um sistema que não possui recarga de energia, o tempo de vida do sistema até que haja a primeira falha é muito maior utilizando a técnica de Replicação Sob Demanda do que quando é utilizada a técnica de Cópia Reserva.

Esses resultados deixam claro que a combinação das técnicas de Cache Cooperativo e Replicação Sob Demanda se apresenta como uma excelente alternativa a ser empregada em sistemas de gerência de equipes de mitigação de sinistros que utilizem dispositivos portáteis conectados por uma rede ad hoc.

A eficiência das técnicas aqui estudadas nos indica que podem ser adaptadas para sistemas com outras finalidades com grande facilidade, inclusive fazendo a verificação de outras variáveis que forem relevantes a cada contexto.

7.4 TRABALHOS FUTUROS

Vários são os desafios encontrados em um sistema de gerenciamento de equipes em mitigações de sinistros. Para a área de computação distribuída, o que consideramos mais relevante é o trabalho de economia de recursos que pode ser explorado com diversas técnicas, que vão desde um roteamento mais eficiente até o uso de novas técnicas para tolerância a falhas. Especificamente, vemos que o contexto de sinistros um tema bastante amplo com diversos desafios à tecnologia da informação.

Desse modo, as características das equipes e o contexto da situação podem ser utilizados para otimizar técnicas de localização de arquivos, como, por exemplo, a busca de um dado médico ser direcionada para a equipe médica, ou um dado a respeito de uma construção estar guardado em dispositivos da equipe de engenharia, e assim por diante. Esse uso para o contexto evitaria buscas por inundação, posto que o conhecimento prévio do dado sinaliza o seu possível local de hospedagem.

Em outro enfoque, dentre as informações trafegadas em um sinistro pode haver informações confidenciais. Assim, um trabalho de segurança que evite que informações sensitivas fluam livremente em uma rede ad hoc pode ser bastante útil para um sistema como o SAE principalmente nos casos de utilização em meio a sinistros causados por guerra ou terrorismo.

O protótipo do SAE, mostrado no item 4.1, pode também ser aprimorado, com a inclusão do Cache Cooperativo e novas técnicas que venham a ser estudadas.

Diversas áreas da informática podem ser pesquisadas para esse sistema, como inteligência artificial para estudo das ações automáticas e interfaces ágeis para comunicação, entre outras.

8 REFERÊNCIAS

[1] GUIMARÃES, M. ***Computação Distribuída em um Sistema de Apoio a Emergência***. Estudo Dirigido, Setembro de 2004.

[2] RODRÍGUEZ-RODRIGUEZ, A., ALEMÁN-FLORES, M., ***A Framework for the Search and Rescue Domain***, In: INAP**,** 2001, p. 305-316.

[3] BERFIELD, A., CHRYSANTHIS, P. K., LABRINIDIS A., ***Automated Service Integration for Crisis Management.*** In: FIRST WORKSHOP ON DATABASES IN VIRTUAL ORGANIZATIONS (DIVO 2004), 2004, Paris - França.

[4] CAO, G., YIN, L., DAS, C. R., ***Cooperative Cache-Based Data Access in Ad Hoc Network***. IEEE Computer, p. 32-39, fevereiro**,** 2004.

[5] BERTINI L., LOQUES, O., LEITE J.C.B. ***Replicação de dados em Redes Ad Hoc para Sistemas de Apoio em Situações de Desastres***. In: 23º SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES - SBRC 2005. Fortaleza – Ceará – Brasil.

[6] HOLLIDAY, J., STEINKE, R. C., AGRAWAL, D., ABBADI, A. E.. ***Epidemic algorithms for replicated databases.*** In: IEEE Transactions on Knowledge and Data Engineering**.** Volume 15, Issue 5. IEEE Educational Activities Department, Setembro – 2003. p. 1218-1238.

[7] LAMPORT, L. ***Time, clocks, and the ordering of events in a distributed system***. Communications of the ACM**,** Julho -1978. p. 558–565.

[8] THOMAS, R. H.. ***A majority consensus approach to concurrency control for multiple copy databases*****.** ACM Transactions on Database Systems (TODS) , Junho – 1979. p. 180–209.

[9] YU, H. AND VAHDAT, A. ***Design and evaluation of a continuous consistency model for replicated services.*** PROCEEDINGS OF USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI), San Diego, CA, USA, Outubro - 2000, pp. 305–318.

[10] WHITEHOUSE, K., SHARP C., BREWER E., CULLER D.. ***Hood: a neighborhood abstraction for sensor networks.*** In: In Proceedings of ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '04). ACM Press, Junho 2004.

[11] GUY, R., ***Ficus: A Very Large Scale Reliable Distributed File System***. Ph.D. Thesis, Junho 1991, UCLA Computer Science Department, technical report CSD-910018.

[12] GUY, R., REICHER, P., RATNER, D., GUNTER, M., MA, W., POPEK G.. ***Rumor: Mobile Data Access Through Optimistic Peer-to-peer Replication***. In: Proceedings: ER'98 Workshop on Mobile Data Access, 1998.

[13] PETERSEN, K., SPREITZER, M. J., TERRY, D. B., THEIMER, M. M.. ***Bayou: Replicated Database Services for World-wide Applications***. In: Proceedings of the Seventh ACM SIGOPS European Workshop, Setembro 1996.

[14] PAGE Jr., T.W., GUY, R.G., POPEK, G.J., ***Consistency algorithms for optimistic Replication***. In: 1st IEEE Int. Conference on Network Protocols, Outubro 1993.

[15] DEMERS, A. et al,. ***The bayou architecture: Support for data sharing among mobile users***. In: Proceedings IEEE Workshop on Mobile Computing Systems & Applications, Santa Cruz, California, pages 2–7 1994.

[16] DEY, A.K. ***Providing Architectural Support for Building Context-Aware Applications***. Ph.D. thesis, Novembro, 2000. Georgia Institute of Technology.

[17] SAITO, Y., SHAPIRO, M. ***Optimistic Replication***. In: ACM Computing Surveys (CSUR). Volume 37, Issue 1. Hewlett-Packard Laboratories, março - 2005.

[18] SIEBRA, C. and TATE, A. (2003) ***I-Rescue: A Coalition Based System to Support Disaster Relief Operations***. In: The Third International Association of Science and Technology for Development (IASTED), International Conference on Artificial Intelligence and Applications (AIA-2003), Benalmadena, Spain, Setembro 2003.

[19] MEHROTRA, S., BUTTS, C., KALASHNIKOV, D., VENKATASUBRAMANIAN, N., RAO, R., CHOCKALINGAM, G., EGUCHI, R., ADAMS, B., HUYCK, C., ***Project RESCUE: Challenges in Responding to the Unexpected.*** In: IS&T/SPIE 16TH ANNUAL SYMPOSIUM ON ELECTRONIC IMAGING, DISPLAYS, & MEDICAL IMAGING, Volume 5304, 2004.

[20] GREEN, D. R., KING, S. D., MCDONALD, J., and WOODS, L. ***iSARS - Interactive Search and Rescue System: Integrating Geospatial Data for Coastal and Marine Search and Rescue***. In: COAST GIS 2003 5TH INTERNATIONAL SYMPOSIUN, COMPUTER MAPPING AND GIS FOR COASTAL ZONE MANAGEMENT, Genoa, Italy, Outubro-2003.

[21] ***Bonn Motion***. Disponível em: <http://web.informatik.unibonn.de/IV/Mitarbeiter/dewaal/BonnMotion/>. Acesso em: 15 de outubro de 2005.

[22] **The Network Simulator**. Disponível em: <http://www.isi.edu/nsnam/ns/>. Acesso em: 15 de outubro de 2005.

[23] GEHRKE J., MADDEN S. ***Query Processing in Sensor Networks***. In: IEEE Pervasive Computig, Jan-Mar, 2004, pg 46-55.

[24] BURRELL J., BROOKE T., BECKWITH R. ***Vineyard Computing: Sensor Networks in Agricultural Production***. In: IEEE Pervasive Computig, Jan-Mar, 2004, pg 38-45.

[25] GIBBONS P., KARP B., KE Y., NATH S., SESHAN S., ***Irisnet: An Architecture for a Worldwide Sensor Web***. In: IEEE Pervasive Computig, Out.-Dez., 2003, pg 22-33.

[26] HUANG C., TSENG Y., ***The Coverage Problem in a Wireless Sensor Network***. In: PROCEEDINGS OF THE 2ND ACM INTERNATIONAL CONFERENCE ON WIRELESS SENSOR NETWORKS AND APPLICATIONS**.** San Diego, CA, USA -2003.

[27] CHEN B., JAMIESON K., BALAKRISHNAN H., MORRIS R., ***Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance***. In: Ad Hoc Wireless Networks, ACM Wireless Networks Journal, Setembro 2002.