

Mestrado em Engenharia Informática
Estágio
Relatório Final

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Requisitos de Qualidade de Serviço

José Miguel Martins Brandão
jbrandao@student.dei.uc.pt

Orientador:

Jorge Sá Silva

Data: 1 de Setembro de 2011



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

Neste relatório é apresentado o trabalho desenvolvido durante o ano lectivo 2010-2011 para a criação de uma plataforma de detecção de proximidade e monitorização utilizando redes de sensores sem fios. Esta proposta consiste numa ferramenta na qual são distribuídos nós sensores por crianças num jardim escola para analisar os seus comportamentos sociais e relacionamentos com outras crianças. Este sistema detecta que outras crianças com nós sensores estão próximas, lê a informação relativa aos sensores do dispositivo e adiciona toda essa informação numa base de dados que pode ser visualizada numa página Web, para posterior análise do tutor, de psicólogos, médicos, educadores de infância ou do administrador do sistema.

Palavras-Chave

Deployment, iTakeCare, Monitorização, Proximidade, Qualidade de Serviço, Redes de Sensores Sem Fios, Sensores, TCP, TinyOS.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Índice

Capítulo 1 Introdução.....	1
1.1 Redes de Sensores Sem Fios e desafios atuais	1
1.2 Objectivos.....	2
1.3 Estrutura do Relatório	3
Capítulo 2 Estado da Arte.....	4
2.1 Aplicações	4
Smart Kindergarten	4
WSN/RFID Architecture For Children’s Monitoring	5
An Advanced Wireless Sensor Network for Health Monitoring	6
2.2 Qualidade de Serviço nas RSSF.....	7
2.2.1. Camada de aplicação	7
TCP Light	7
2.2.2. Camada de transporte	9
Berkeley Low-power IP	9
Distributed TCP Caching	9
Event-to-sink Reliable Transport (ESRT)	10
2.2.3. Camada de rede	10
SPEED	10
Energy-Aware QoS Routing Protocol.....	10
2.2.4. Camada MAC	11
2.3 Deployment.....	12
SpyGlass	12
MoonSense	12
Sensor Network Inspection Framework (SNIF)	12

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Mote-View	13
TinyViz	13
Surge Network Viewer.....	13
An Intelligent Deployment and Clustering Algorithm for a Distributed Mobile Sensor Network.....	13
Collaborative Deployment Optimization and Dynamic Power Management in Wireless Sensor Networks.....	14
Capítulo 3 Metodologia e Etapas.....	15
3.1 Metodologia e Plano de Trabalho.....	15
3.2 Requisitos.....	16
3.2.1 iTakeCare	16
3.2.2 Deployment	17
3.3 Arquitetura.....	19
3.4 Sistema de Back-end	19
Capítulo 4 Trabalho Efectuado e Avaliação de Resultados.....	20
4.1 A Aplicação	20
Rede de Sensores Sem Fios.....	20
Mensagens.....	21
MyTCP	22
Confiança	23
Servidor	24
Servidor Web.....	26
Base de Dados.....	26
Web Site	27
Administrador	27
Cliente	33
4.2 Deployment.....	38
4.3 Avaliação Funcional	40

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

4.4	Avaliação de Desempenho.....	41
Capítulo 5 Conclusão.....		44
5.1	Considerações	44
5.2	Trabalho futuro.....	44
Referências		45

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Lista de Figuras

Fig 1: Arquitetura Smart Kindergarten [1]	5
Fig. 2: Esquema da aplicação baseada em RFID[10].....	5
Fig. 3: Arquitetura de sistema para “Health Monitoring” [13].....	6
Fig. 4: Formato da mensagem.	7
Fig. 5: Exemplo do controlo de erros TCP Light.	8
Fig 6: Distributed TCP Caching [2].....	9
Fig. 7: <i>Deployment</i> de nós estáticos e resultados da optimização:.....	14
Fig. 8: Diagrama de Gantt.....	15
Fig. 9: Arquitetura do Sistema.	19
Fig. 10: Mensagens iTakeCare.	21
Fig. 11: Mensagem recebida com sucesso.	22
Fig. 12: Mensagem perdida.	23
Fig. 13: Diagrama de Classes	25
Fig. 14: Diagrama Entidade-Relação	26
Fig. 15: Vista <i>List Users</i> para o administrador.	28
Fig. 16: Vista <i>Add Users</i> para o administrador.	29
Fig. 17: Vista <i>Del Users</i> para o administrador.	30
Fig. 18: Vista <i>Live Stats</i> para o administrador.....	30
Fig. 19: Vista <i>Neighbors</i> para o administrador.	31
Fig. 20: Vista <i>Trust Level</i> para o administrador.....	32
Fig. 21: Vista <i>Proximity</i> para o administrador.	33
Fig. 22: Vista <i>Live Stats</i> para o cliente.	34

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Fig. 23: Vista <i>Neighbors</i> para o cliente.	35
Fig. 24: Vista <i>Trust Level</i> para o cliente.	36
Fig. 25: Vista <i>Proximity</i> para o cliente.	37
Fig. 26: Arquitetura da ferramenta de detecção de erros	38
Fig. 27: Árvore de decisão para detecção de problemas.	39

Acrónimos

ACK	Acknowledgement
BLIP	Berkeley Low-power IP
ESRT	Event-to-sink Reliable Transport
QoS	Qualidade de Serviço
QUIRE	Quality-of-service specific Information Retrieval
RFID	Radio-Frequency IDentification
RRSI	Received Signal Strength Indication
RSSF	Rede de Sensores Sem Fios
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Capítulo 1

Introdução

1.1 Redes de Sensores Sem Fios e desafios atuais

As Rede de Sensores sem fios (RSSF) são redes agregadas constituídas por sistemas autónomos de nós sensores que colaboram cooperativamente entre si para recolher informação do mundo físico á sua volta, convertendo-a para dados virtuais, para assim poder ser processada e analisada por computadores e posteriormente entregue ao utilizador. Estes dados podem assumir valores de temperatura, luminosidade, som, pressão, medição de gases, proximidades, entre outros tipos de sensores.

Cada nó sensor está equipado com um transmissor rádio, através do qual envia e recebe informações. Esta informação dos nós sensores pode ser enviada para outros nós sensores ou para o servidor de destino dos dados, normalmente chamado de *sink*. Cada nó pode retransmitir mensagens que tenha recebido e que não sejam para ele, para a mensagem poder chegar ao seu destino. Esta é uma das grandes vantagens das RSSFs, podendo os nós funcionar como encaminhadores de mensagens, não necessitando de estarem todos os nós sensores ao alcance direto do *sink*. O *sink* é o orquestrador da rede e destino para grande parte das mensagens. Posteriormente, é no servidor que os dados são tratados através de um *middleware* para serem inseridos numa base de dados, num servidor Web, ou qualquer outro destino definido pelo utilizador.

As RSSFs são uma tecnologia inovadora que ainda precisa de alguns melhoramentos para chegar ao seu potencial máximo. O maior problema neste tipo de redes é o consumo energético, pois os nós sensores funcionam normalmente através de baterias para poderem ser móveis e de fácil instalação. Grande parte do consumo energético é devido ao excessivo número de transmissões e recepções. Para reduzir um pouco o consumo energético, os nós tendem a entrar em modo adormecido quando não têm informação para transmitir. desligando o seu rádio e poupando bateria. A duração da bateria dos sensores deve ser o mais longa possível para evitar a intervenção humana na troca destas.

1.2 Objectivos

Este estágio está focado no desenvolvimento de uma aplicação para as RSSFs, o projeto chama-se iTakeCare e tem dois grandes objectivos. Primariamente, monitorizar a distancia entre nós sensores, para ser possível determinar que nós são mais ou menos solitários, e posteriormente recolher dados sobre as proximidades de um determinado número de nós e disponibilizar essa informação ao utilizador através de uma página Web. Neste serviço é também oferecido ao utilizador informações de sensores embutidos nos dispositivos, tal como a temperatura ou outros indicadores relevantes para a utilização desta aplicação.

A ideia principal deste projeto é monitorizar pessoas. Pode-se monitorizar crianças num jardim escola, e saber o estado das suas relações sociais. Isto para saber se uma criança passa demasiado tempo sozinha, com que outros colegas ela passa mais tempo, ou até detectar um possível caso de *bullying* entre dois elementos. Para enriquecer esta funcionalidade são adicionados os valores de alguns parâmetros relativos ao estado de saúde da criança e são enviados alarmes no caso de ser detectada alguma anomalia, tal como o aumento sucessivo de temperatura, que pode ser um sintoma de febre.

Outra possibilidade é a monitorização de idosos a longa distancia. O idoso pode estar em sua casa, e o seu auxiliar de saúde é alertado caso seja detectada alguma irregularidade.

No âmbito deste estágio foi também efectuado um estudo e proposta de uma ferramenta, para o desenvolvimento e apoio da fase de *deployment* nas RSSF, uma vez que esta fase é muito crítica nas ferramentas desenvolvidas.

Deployment é o ato de colocar os nós no mundo físico no local onde se quer fazer o estudo. É uma das etapas mais importantes das RSSFs, pois caso uma rede esteja mal estruturada pode levar a número excessivo de nós e a um mau desempenho da rede, ou mesmo fazer com que alguns nós fiquem sozinhos. Com este cenário, outros nós não podem enviar as suas mensagens para o seu destino, o que implica uma intervenção humana e um aumento de custos. Esta tarefa acontece em todo o tipo de RSSFs.

Deployment é, por vezes, uma tarefa complexa, visto que existem vários problemas físicos, de comunicação e que podem afectar toda a rede. Estes problemas serão detalhados na secção 3.2.2.

Será apresentada uma proposta de ferramenta para apoio ao *deployment* em RSSF com base de comunicação dividida por tempo TDMA. Este tipo de comunicação é baseada numa

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

divisão temporal onde cada nó tem um espaço reservado para efetuar as suas transmissões, podendo assim quando não está a transmitir entrar em modo adormecido para poupar recursos. Este sistema obriga a uma sincronização de relógios entre os vários dispositivos, para que o nó possa enviar as suas mensagens quando deve, caso contrário elas serão perdidas.

1.3 Estrutura do Relatório

Este relatório está dividido em cinco grandes secções, sendo a primeira a introdução onde foi referida a ideia principal do trabalho realizado. De seguida temos o estado da arte das RSSFs, mais especificamente para a Qualidade de Serviço (QoS) e monitorização a longa distancia. No Capítulo 3 será introduzida a metodologia de trabalho utilizada, assim como os requisitos e arquitetura do sistema elaborado. No 4º Capítulo está a especificação detalhada do trabalho, onde será apresentada a aplicação e os testes realizados. Neste capítulo será também detalhado a proposta de uma ferramenta para apoio ao *deployment* de RSSF. Por fim, será apresentada a conclusão do trabalho realizado e definidas algumas propostas para futuros melhoramentos da aplicação.

Capítulo 2

Estado da Arte

Nesta secção será apresentado o estudo ao estado da arte relacionado com a área do trabalho realizado. Detalha-se primariamente as aplicações, a QoS nas RSSF e, por fim, as ferramentas de apoio ao *deployment* e monitorização do estado da rede.

2.1 Aplicações

No âmbito das aplicações para monitorização do estado de saúde e comportamento de pessoas, existem inúmeras propostas.

Existem aplicações para monitorizar doentes com Alzheimer[18], para monitorizar o estado de saúde, tanto em centros hospitalares como fora deles, como o projeto CodeBlue[19], que serve para monitorizar doenças crónicas, como os diabetes, asma, doenças cardíacas, ou desordens do sono, para efetuar eletrocardiogramas e outras possibilidades de diagnósticos[20][21][22]. Entre estes projeto existem ainda aplicações que monitorizam toda a casa e o estado de saúde dos seus habitantes, e que enviam alarmes caso algo de anormal aconteça [23].

Com este tipo de tecnologias é possível obter alarmes e o estado de saúde de pessoas com problemas instantaneamente, sem ser necessária qualquer deslocação ou a intervenção do paciente. Estas tecnologias dão assim, um grande avanço ao campo preventivo e de apoio a pessoas com problemas de saúde.

De seguida serão detalhados alguns exemplos mais importantes das aplicações e das suas funcionalidades.

Smart Kindergarten

O objectivo deste projeto é monitorizar a localização de crianças através de sensores com a tecnologia Bluetooth. Este sistema pode ser utilizado por tutores e professores, para saber onde as crianças passam mais tempo, se estão sozinhas ou se passam demasiado tempo na sala de aula sem conviver com outras crianças. Os sensores são distribuídos por crianças dentro de um jardim escola, onde estes recolhem informação e enviam-na através de uma rede do tipo piconet, que é um sistema ad-hoc de dispositivos similar às RSSF, para um

An Advanced Wireless Sensor Network for Health Monitoring

O sistema apresentado em [13] integra uma elevada quantidade de sensores, desde sensores corporais tais como oxímetros, pressão arterial, temperatura e acelerómetro, até sensores físicos distribuídos pela casa do utilizador, como de pressão, temperatura, RFID, entre outros. Integra também um sistema de *backbone* que liga os dispositivos com uma base de dados e um servidor que pode enviar alarmes.

Pode enviar alarmes tanto a uma pessoa pré definida, como ao próprio utilizador através de actuadores. Um alarme pode por exemplo ter a informação de que o forno está a atingir uma temperatura elevada e deve ser verificado. Este alarme é útil para os doentes de Alzheimer que vivem sozinhos. Esta informação pode ser acedida através de uma aplicação para PDA ou no computador local com os dados.[Fig. 3]

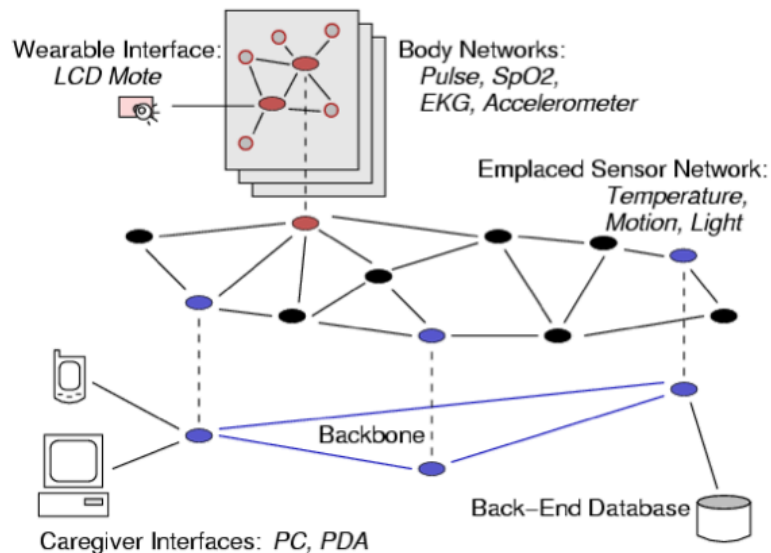


Fig. 3: Arquitetura de sistema para “Health Monitoring” [13].

Ainda existe um grande caminho a percorrer no âmbito das aplicações das RSSFs para a monitorização do estado de saúde, para que estas possam ser verdadeiramente implementadas e utilizadas sem problemas técnicos e com o rigor exigido. Para isso é necessário um maior desenvolvimento e investigação no desenvolvimento de mecanismos de robustez, QoS, segurança e privacidade.

2.2 Qualidade de Serviço nas RSSF

As RSSF como qualquer outra rede, partilham quase todas as camadas OSI, mas com algumas ligeiras diferenças. Referente á QoS foram analisadas varias propostas de melhoramento para as varias camadas, começando na camada de aplicação até á camada MAC.

2.2.1. Camada de aplicação

TCP Light

No decorrer da cadeira de Protocolos de Comunicação II, do Mestrado em Engenharia de Software foi desenvolvido pelo grupo de alunos no qual eu estava incluído, um protocolo de comunicação com as funcionalidades de um protocolo TCP e com uma performance reduzida, para assim se adaptar as RSSF.

Este projeto foi uma prova de conceito desenvolvido em TinyOS, para provar que é possível desenvolver uma versão TCP para RSSF.

Neste projeto foram definidos 2 tipos de mensagens para manusear o TCP.[Fig. 4]

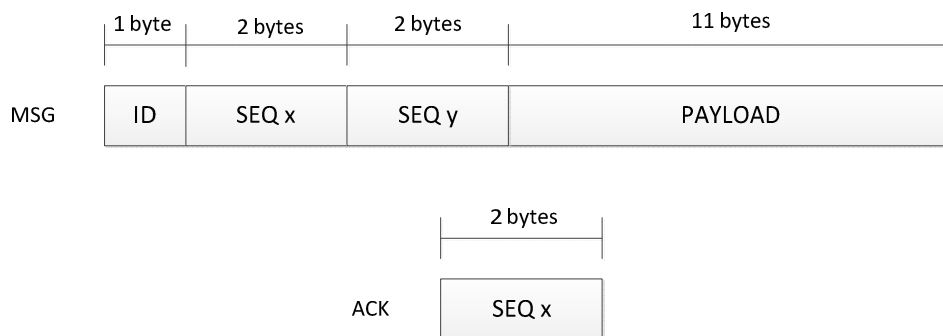


Fig. 4: Formato da mensagem.

A mensagem de transmissão de informação (MSG) está dividida em quatro secções: ID, SEQx, SEQy e PAYLOAD.

O ID é o número identificador para designar o tipo de mensagem. Pode ter os valores 0, 1 ou 2. 0 é o identificador para os tipos de mensagem Syn e Syn + Ack, 1 para a troca de mensagens de informação e 2 para os tipos Fin e Fin + Ack. Sendo que Syn significa o pedido de inicialização do processo TCP e Fin o processo para finalizar a transferência de pacotes.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

SEQx é o número de sequência da mensagem, que identifica unicamente cada mensagem, definido pelo remetente.

O SEQy é utilizado como um ACK, assim caso o remetente precise de enviar uma mensagem e um ACK, pode enviar tudo numa transmissão poupando recursos e tempo.

O PAYLOAD é o campo onde a informação principal da mensagem vai ser enviada.

O ACK é o tipo de mensagem que serve para confirmar a recepção de uma mensagem posterior. É o mais pequeno possível para poupar alguns recursos. Este tipo de mensagem é composto apenas pelo número de sequência da mensagem recebida com sucesso. É unicamente usado caso não seja necessário enviar mensagem nesse instante para o destino do ACK, pois caso seja necessário enviar alguma informação, será enviada uma mensagem normal, que já incluir ACKs.

Este protocolo não guarda mensagens desordenadas para posterior ordenação. Caso se perca uma mensagem, é reenviado o ACK da ultima mensagem recebida corretamente dentro da ordem, e retomada a transmissão a partir desse número de sequência.[Fig. 5]

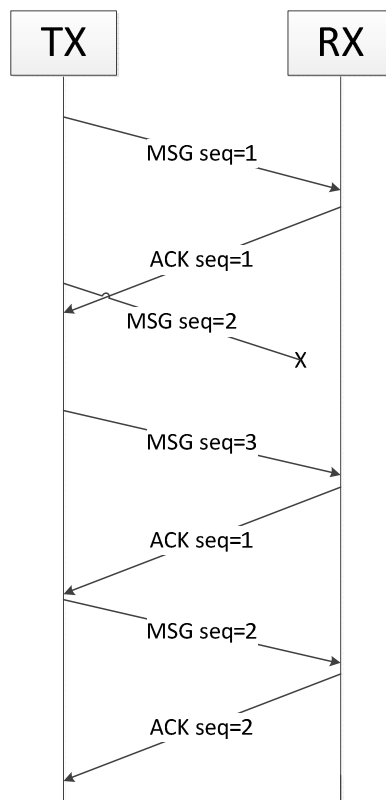


Fig. 5: Exemplo do controlo de erros TCP Light.

2.2.2. Camada de transporte

Berkeley Low-power IP

Berkeley Low-power IP (BLIP) é uma implementação para as RSSF com vários protocolos IP, tal como UDP, ICMP e TCP. A versão de TCP é uma implementação leve deste mecanismo, pois não guarda as mensagens desordenadas para serem depois reintegradas na ordem correta através de uma retransmissão. Este processo pode levar a algum atraso no tempo total de transmissão. Este protocolo permite também a transmissão de informação entre nós sensores, e não apenas com o *sink*. [8]

Distributed TCP Caching

Em [2], é explicado que o protocolo Distributed TCP Caching é desenhado para trabalhar em RSSF *multi-hop*. Este reduz significativamente o número de retransmissões de pacotes, pois cada nó no caminho da rede vai guardando em *buffer* as mensagens que retransmite, até lhe chegar a informação de que a mensagem foi recebida com sucesso (ACK). Quando uma mensagem é perdida, esta não será reenviada pelo nó de origem, mas sim, pelo nó mais próximo que tiver a mensagem em *buffer*, retomando assim a transmissão normal das mensagens. [Fig 6]

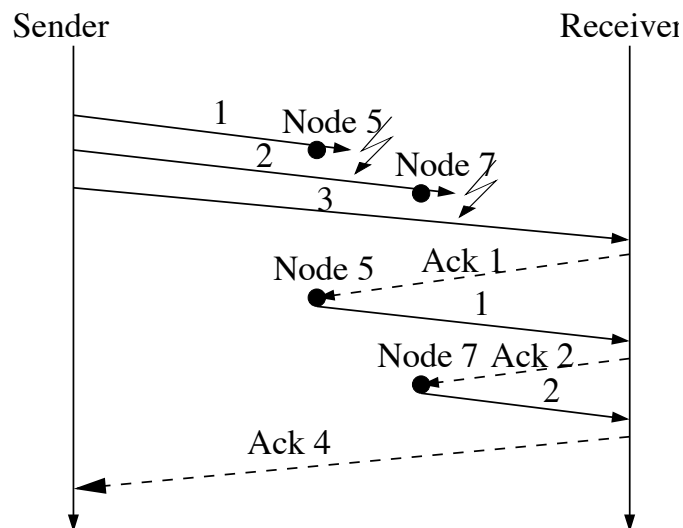


Fig 6: Distributed TCP Caching [2]

Event-to-sink Reliable Transport (ESRT)

Este protocolo de transporte providencia reabilidade ao sistema e gestão de congestionamento de mensagens, podendo ainda preservar o consumo energético da rede. Assim, esta mais orientado para redes com sistema *multi-hop*, pois permite agregar mensagens de diversos nós e enviar apenas uma para o *sink*. Isto é feito por um nó que se encontre numa zona ótima para transmissão de mensagens. O protocolo auto configurasse para cada tipo de rede, adaptando-se a topologias dinâmicas ou de posicionamento aleatório.[14]

2.2.3. Camada de rede

A qualidade de serviço na camada de rede passa por protocolos e medidas que tenham um menor caminho de comunicação, para reduzir o consumo energético e o atraso das mensagens.

SPEED

Este protocolo garante uma melhoria na comunicação *end-to-end*. Usa a localização dos sensores para descobrir a rota mais rápida para o *sink*. Ao saber a localização de todos os elementos da rede, calcula a distancia e o tempo que a mensagem demora a chegar ao destino, podendo evitar o congestionamento de mensagens. SPEED mantém uma tabela dos vizinhos mais próximos, e não guarda as rotas para posterior uso, para assim poupar memória. Por outro lado, preocupa-se com a velocidade de transmissão das mensagens, e não com a sua confirmação de chegada ao destino.[15]

Energy-Aware QoS Routing Protocol

Em [16], é apresentado um protocolo com o objectivo de melhorar a eficiência energética das comunicações. Este descobre o melhor caminho de comunicação, isto é, o que tiver menor custo energético. Adicionalmente implementa um sistema de filas de mensagens, onde existem dois tipos de mensagens. Um dos tipos alberga mensagens em tempo real e o outro mensagens sem ser em tempo real. Seguidamente, pode-se distribui estas mensagens pelas filas, de acordo com a opção definida pelos nós no cabeçalho da mensagem. Com este sistema é possível garantir alguma velocidade na entrega das mensagens, podendo as mais prioritárias passar a frente das menos prioritárias.

2.2.4. Camada MAC

Quality-of-service specific Information Retrieval

O Quality-of-service specific Information Retrieval (QUIRE), é um protocolo MAC que otimiza a performance da rede relativamente á qualidade de serviço. Este protocolo baseando se na distribuição aleatória e densidade distributiva dos nós e nos requisitos de QoS, divide os nós por vários grupos de células. Nestas células existe um único nó que transmite as mensagens para o *sink* de acordo com o esquema de controlo de transmissão, para assim não efetuar demasiadas transições, nem causar o aumento da latência do sistema.[17]

2.3 Deployment

SpyGlass

É uma simples ferramenta para visualizar e armazenar o estado corrente das RSSF, sem verificação de mensagens ou suporte TinyOS. As mensagens são analisadas quando chegam ao servidor do *sink*, e este pode enviar as informações recolhidas por TCP/IP para um *gateway*, onde o utilizador visualizará os dados da RSSF, no entanto apenas suporta TinyOS.[4]

MoonSense

É uma aplicação que tem apenas suporte para TinyOS. Ele analisa as mensagens que chegam para ao *sink*. Tem algumas características avançadas, como mostrar os nós num mapa real de imagens de satélite e pode ser adicionada a localização GPS dos nós para uma melhor compreensão da localização. Carece da detecção de erros. O aplicativo também pode ser usado para controlar RSSF, enviando comandos para a rede. Estes comandos conseguem dizer aos dispositivos de rede para parar de enviar mensagens, aumentar o tempo entre as mensagens, ou até mesmo, redefinir a rede (reiniciar o algoritmo *multi-hop*). A ferramenta é principalmente para monitorar e visualizar a RSSF, leituras de sensores e o estado dos nós, ou ainda para proporcionar um sistema de alarme configurável pelo usuário.[5]

Sensor Network Inspection Framework (SNIF)

SNIF permite a inspeção e depuração de uma RSSF usando nós especiais com duas interfaces de comunicação. A primeira interface serve para capturar todas as mensagens na rede e para devolver mensagens para uma estação de base. Nesta estação existe uma representação do estado da rede. Os nós com este tipo de interface devem ser implementados no meio da rede e estar ao alcance de todos os nós que estão a ser monitorados. A segunda interface de comunicação utilizada é o Bluetooth, porque é mais robusto e porque para obter um estado óptimo da rede é melhor não perder mensagens que podem conter detalhes importantes sobre o estado de toda a rede. Esta ferramenta também pode ser útil para monitorar o estado da rede se usado em uma RSSF totalmente implantada. O principal problema de SNIF é que o Bluetooth funciona em 2.4GHz e a interface 802.15.4 dos nós usados estão em 433-915 MHz. Para alguns caso isto funciona, mas no entanto muitos projetos funcionam com 802.15.4 a 2.4GHz, o que ao usar esta ferramenta

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

iria trazer mais ruído para as comunicações e provocar novos problemas com a sua implementação. [6]

Mote-View

Mote-View é uma ferramenta para visualizar RSSF. Providencia um instrumento para monitorizar a rede facilmente e guardar informação. O utilizador pode fazer pedidos aos nós, e caso um nó deixe de enviar mensagens durante um intervalo de tempo definido, ele é denominado como com problemas, e por isso, fica a cinzento na imagem representativa da rede dada pelo Mote-View. Podem ser definidos alarmes para avisar o utilizador.[7]

TinyViz

TinyViz é uma ferramenta mais genérica para monitorizar RSSF. Mostra as leituras dos sensores, o estado dos LEDs, os *links* de comunicação e pode ser integrada com o TOSSIM, que é um simulador para RSSF em TinyOS. Tanto o TinyViz como o TOSSIM só estão disponíveis para TinyOS, o que pode ser uma limitação do sistema. Por outro lado, o TinyViz possibilita a visualização em tempo real das simulações em TOSSIM, escutando o processo de simulação e permitir ao utilizador controlar a simulação através de uma interface controlos. [7]

Surge Network Viewer

Surge Network Viewer é uma distribuição Java de uma aplicação que integra o pacote do TinyOS. Esta permite monitorizar e analisar a performance da rede, e por isso, pode ser vista a topologia da rede, e armazenado um *log* de estatísticas da rede. Este *log* que pode ser visualizado através de um gráfico. O mais vantajoso desta ferramenta é que descobre automaticamente a configuração da rede. [7]

An Intelligent Deployment and Clustering Algorithm for a Distributed Mobile Sensor Network

Em [11], é proposto um algoritmo em que os sensores dispostos aleatoriamente numa rede, optam por um método de transmissão de mensagens de acordo com as suas posições e proximidade, isto é, ou adoptam uma performance de *clustering* em que apenas um transmite mensagens para fora do *cluster*, ou assumem uma postura *peer-to-peer* e os próprios sensores

enviam as suas mensagens. Este protocolo oferece uma nova técnica para melhorar o *deployment*, dando inteligência aos nós para optarem pela sua distribuição.

Collaborative Deployment Optimization and Dynamic Power Management in Wireless Sensor Networks

No artigo [12], é apresentada uma técnica de *deployment* que relaciona o consumo energético com a área abrangida pelos sensores. O que este algoritmo propõe é que seja encontrado um ponto de equilíbrio entre o consumo energético e a leitura dos sensores, uma vez que quando os nós estão muito juntos, diversos podem ler os mesmos valores. Se tal cenário ocorre, não é necessário que todos os nós estejam ligados ao mesmo tempo. No entanto, a área que cobrem é menor. Para este efeito, usam um valor de rácio que varia entre 0 e 1, onde 0 é maior área e menos eficiência energética e 1 é uma menor área mas uma eficiência melhor.[Fig. 7]

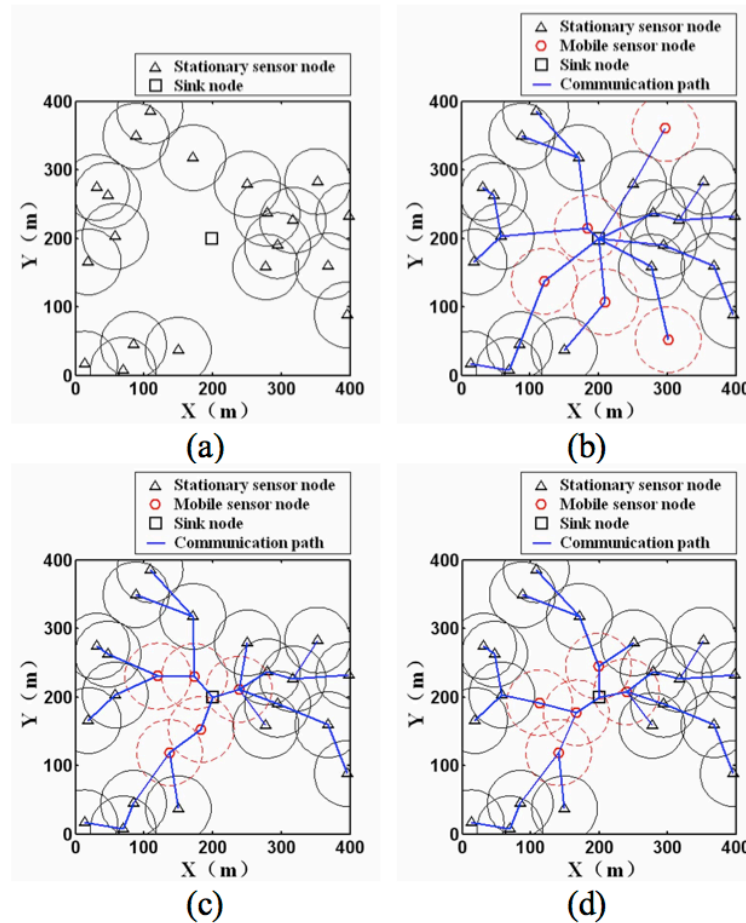


Fig. 7: *Deployment* de nós estáticos e resultados da otimização:

(a) Inicial (b) rácio =0 (c) rácio =0.35 (d) rácio = 1 [12]

Capítulo 3

Metodologia e Etapas

3.1 Metodologia e Plano de Trabalho

Durante este estágio foi abordada uma metodologia variante do SCRUM, que consistiu em pequenas reuniões semanais com o orientador para apresentar trabalho realizado, discutir melhorias e definir próximas tarefas. Foram também realizadas ao longo do ano, varias reuniões com o grupo das redes de sensores sem fios, pertencente ao laboratório de comunicação e telemática, onde eram apresentados os diversos trabalhos e discutidas varias abordagens e sugestões.

	Setembro	Outubro	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maió	Junho	Julho	Agosto
Estudo e Análise de requisitos de Deployment												
Artigo da Ferramenta de Deployment												
Relatório Intermédio												
Estudo e Análise de requisitos iTakeCare												
Protótipo iTakeCare												
Aplicação iTakeCare (RSSF)												
Aplicação iTakeCare (Middleware, servidor Web e Base de Dados)												
Testes e Debug												
Relatório Final												

Fig. 8: Diagrama de Gantt

Na Figura 8 apresenta-se o diagrama de Gantt deste projeto. No início do projeto começou-se por fazer um estudo de ferramentas para apoio ao *deployment* de sensores no terreno, para depois propor uma nova ferramenta de assistência nesta tarefa. Esta ferramenta foi proposta através de um artigo para a Conferência Nacional de Redes de Sensores Sem Fios, o qual está em anexo. De seguida, foi feito o relatório intermédio do estágio, tal como o estudo e análise de requisitos para o projeto iTakeCare e um protótipo funcional da aplicação.

Posteriormente, foi elaborada a aplicação das RSSF e, por fim, o servidor com integração na base de dados. Na fase final deste estágio, foram produzidos testes e corrigidos erros na aplicação. Foi também feito este relatório com o objectivo de apresentar o trabalho efectuado e o seu resultado.

3.2 Requisitos

3.2.1 iTakeCare

Existe a necessidade de estudar o comportamento de crianças, não apenas para a sua monitorização, mas também para saber o motivo de alguns comportamentos erráticos. É do interesse de psicólogos analisar o comportamento de uma criança de forma não muito intrusiva, podendo assim a criança realizar as suas atividades normais e rotineiras sem qualquer preconceito ou pressão. Este sistema permite transmitir diretamente para o servidor os dados relacionados com a proximidade entre crianças. Deseja-se analisar esta proximidade, porque a socialização entre elas é um dos problemas que as afectam certas crianças tendem a esconder-se e não interagir com outras, o que não beneficia a criança para o seu futuro, devendo estes casos ser tratados e analisados por especialistas.

Outra vertente do sistema é o uso dos sensores, por exemplo para monitorizar sinais vitais ou de quedas. Isto permite que o sistema seja usado para monitorizar a longa distancia pessoas com problemas físicos, doentes ou idosas. É possível através do Web Site saber o estado dos sensores e as suas variações ao longo do tempo. Neste caso seria desprezado a funcionalidade da proximidade de outros sensores, mas optimizada a leitura dos sensores.

Um dos requisitos mais importantes deste sistema é a certeza da entrega das mensagens através de uma comunicação estável. Por esta razão, é implementada uma versão em UDP que simula o comportamento da entrega de mensagens em TCP, com a utilização de um sistema de confirmação de recepção das mensagens por parte do servidor ao nó.

Para este trabalho foram utilizados nós sensores do tipo MICAz da Crossbow. Estes sensores são baseados em Atmel Atmega 128 L com um processador de 8bits, 128Kbytes de memoria flash e 4Kbytes de memória RAM. Possuem um chip de rádio CC2420 com transmite a uma frequência de 2,4GHz [3]. Para sensores foram usadas placas MDA100CB Esta placa contem os sensores de temperatura e luminosidade, porém podem ser adicionados mais sensores. A placa tem um encaixe direto no nó MICAZ, mas existem

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

outros tipos de placas como a MTS310CB que permite medir a luminosidade, temperatura, acústica e aceleração em dois eixos. Para conexão a um computador como no caso do *sink*, existem as placas MIB520 que com o conector de 51-pins encaixam diretamente no nó.

Como servidor foi utilizado Ubuntu 10.4 Desktop Edition. Neste servidor foi instalado o TinyOS que foi eleito como o sistema desejado para esta aplicação, um servidor Web do tipo Glassfish 3.0 e uma base de dados MySQL.

3.2.2 Deployment

Na maioria das RSSF, os nós são dispostos aleatoriamente sobre o campo de estudo. Assim, pode-se reduzir significativamente a performance da rede, pois existem imensos problemas que não podem ser previstos sem uma análise do terreno[6]. Esta ferramenta vai ajudar significativamente nesta tarefa.

Os problemas nas RSSF podem ser categorizados em três áreas distintas: problemas físicos, de Conexão e da Rede.

Nos problemas físicos temos problemas associados aos dispositivos, que são:

- **Nó Morto:** Que acontece quando um nó deixa de transmitir, devido á falta de bateria ou porque simplesmente avariou;
- **Reiniciar de um nó:** Ao reiniciar um nó durante uma transmissão de pacotes pode danificar significativamente a comunicação e fazer com que mensagens importantes sejam perdidas;
- **Leituras erradas dos sensores:** Isto pode acontecer devido a defeitos nos próprios sensores o que provoca uma análise incorreta e não retorna valores reais.

Já nos problemas de conexão temos problemas associados com o estado da comunicação e com a entrega das mensagens. Alguns deles são:

- **Descoberta de vizinhos:** Quando um nó não sabe os seus vizinhos, tem de ir á sua descoberta para se poder integrar na RSSF e enviar a sua informação;
- **Mensagem perdida:** Quando uma mensagem não chega ao seu destino devido a qualquer problema de comunicação da rede;
- **Latência:** o excesso de latência na comunicação pode provocar atrasos na transmissão de informação, o que pode ser problemático no caso de informação critica e prioritária;

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

- Caminho: Ocorre quando não é detectado pelo nó, um caminho de comunicação para o destino das mensagens. Esta situação pode ser causada pela morte de algum nó no caminho de comunicação ou mesmo porque no nó esta afastado de outros;
- *Loops*: Dá-se este nome ao problema quando uma mensagem é enviada para o seu destino, mas não chega lá andando a passar de nó em nó, varias vezes.

Os problemas associados com a rede não afectam apenas um nó, mas a rede como um todo. Neste caso temos:

- Curto tempo de vida: Pode acontecer que devido ao número excessivo de mensagens transmitidas e trafego gerado, as baterias dos nós sensores se esgotem mais rapidamente do que o previsto, o que leva a falhas de comunicação e perda de informação para análise;
- Baixo rendimento de dados: Significa que não existe informação recolhida suficiente para análise ou para retirar conclusões importantes do caso de estudo. Isto pode ser causado por perdas de mensagens ou qualquer outra falha na entrada das mesmas.

3.3 Arquitetura

Este sistema é composto por uma rede de sensores sem fios, que transmite diretamente para um *sink* estacionário ligado a um servidor. Por sua vez, este servidor comunica com o servidor Web que interage com a base de dados para guarda a informação. O servidor serve de *middleware* fazendo a ligação entre a RSSF e o servidor Web.[Fig. 9]

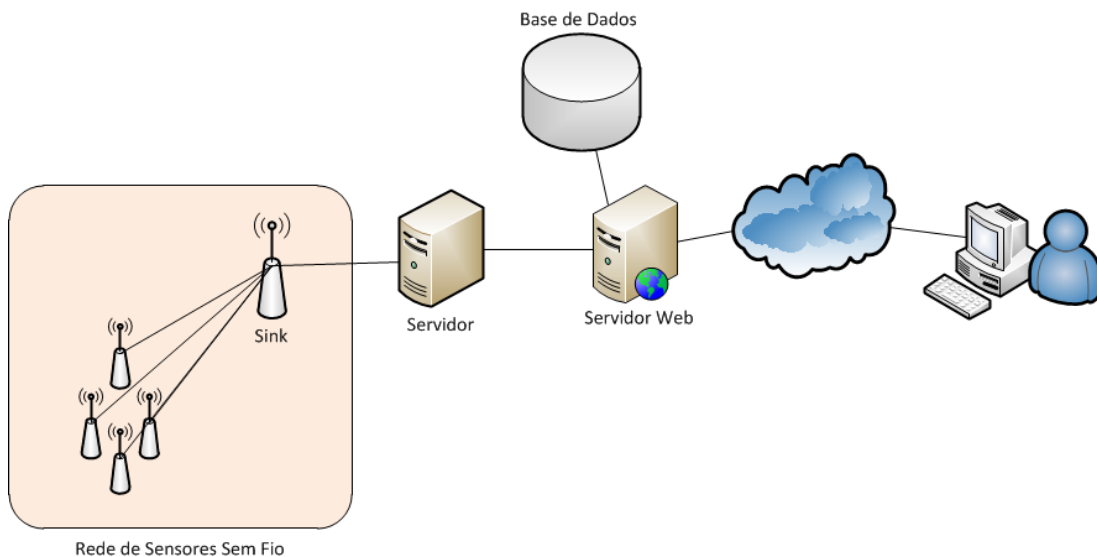


Fig. 9: Arquitetura do Sistema.

3.4 Sistema de Back-end

O sistema de *back-end* é composto pelos servidores de *middleware* e Web, constituídos por uma aplicação Java que lendo os pacotes através do programa serialforwarder do TinyOS ligado. Posteriormente os dados são inseridos na base de dados MySQL para uma posterior análise e consulta através do Web site. Este sistema será detalhado e explicado no ponto 4.1.

Capítulo 4

Trabalho Efectuado e Avaliação de Resultados

4.1 A Aplicação

Rede de Sensores Sem Fios

A aplicação dos sensores está implementada em TinyOS. Esta aplicação é constituída por dois temporizadores diferentes, um para enviar o *broadcast* para outros sensores com a sua informação de 6 em 6 segundos, e outro de 10 em 10 segundos para enviar mensagem diretamente para o *sink*.

No envio de um *broadcast* vai inserida a informação sobre o nó remetente para caso seja detectado como sendo um vizinho, seja adicionado á tabela de vizinhança, para posterior envio de informação.

Para ser detectado se um nó está próximo do outro ou não, foi utilizada a força do sinal dos *broadcasts* recebidos em Received Signal Strength Indication (RSSI). Caso este valor seja maior que -60 é então designado como vizinho. Este valor de força de sinal corresponde a uma distancia máxima de aproximadamente três metros. O valor de distancia máxima foi obtido através de vários testes, movendo os sensores para assim conseguir uma distancia realista sobre a proximidade dos nós.

No envio da mensagem para o servidor, são lidos os valores dos sensores disponíveis, que juntamente com os dados dos vizinhos e da confiança destes é enviada para o *sink*.

Mensagens

Existem três tipos de mensagens. A mensagem para o *sink* (MSG) com 24 bytes, a mensagem de *broadcast* para os outros sensores (*broadcast*) com 2 bytes e a mensagem de confirmação de recepção de mensagens (ACK) com 4 bytes [Fig. 10].

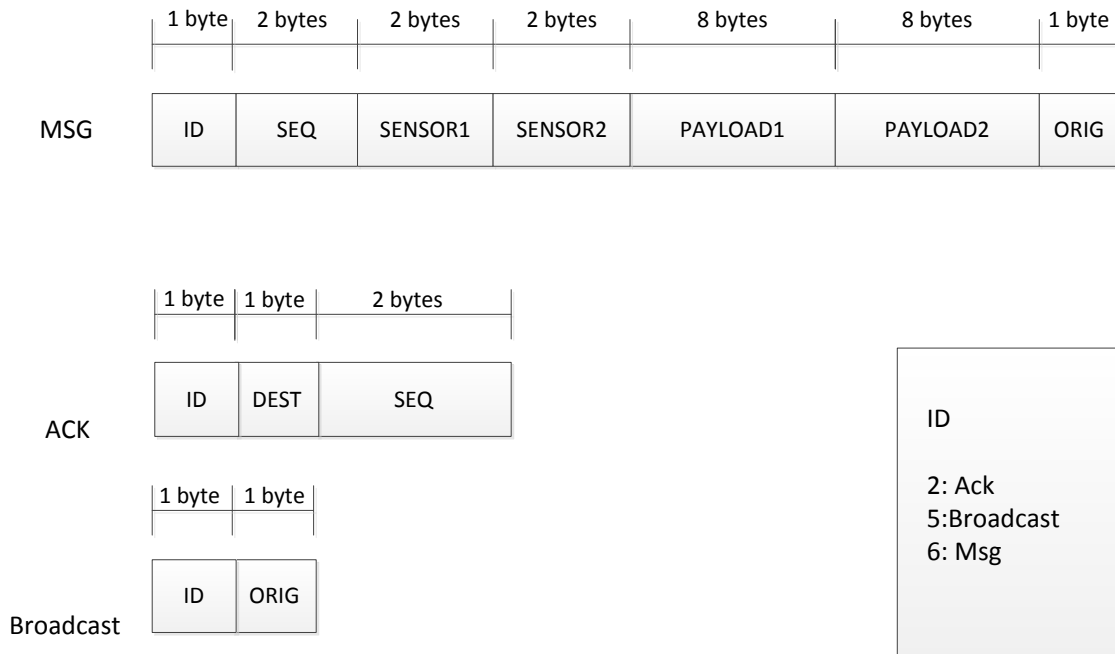


Fig. 10: Mensagens iTakeCare.

Existem três tipos de identificadores para as mensagens, sendo este identificador que define o tipo de mensagem. Tem o valor 2 para os ACKs, 5 para os *broadcasts* e 6 para as mensagens de informação para o *sink*.

A mensagem para o *sink* integra para além do ID, o número de sequência da mensagem atribuído pelo nó e os valores para dois sensores nos campos SENSOR1 e SENSOR2. No campo PAYLOAD1 vai a informação sobre os nós vizinhos. Já no campo PAYLOAD2, vai a informação sobre a confiança dos nós vizinhos. O campo ORIG leva o identificador do nó origem da mensagem.

A mensagem ACK que o *sink* envia para o nó, leva o ID, o identificador do nó para qual este ACK é destinado no campo DEST, e o número de sequência relativo ao ACK.

O *broadcast* é composto pelo ID e pelo identificador do originador da mensagem, inserido no campo ORIG.

MyTCP

A versão de TCP que foi implementada é bastante simples para permitir que as mensagens sejam entregues no servidor. Nesta versão, cada vez que é enviada uma mensagem para o *sink*, ela é guardada no nó sensor até um número máximo de cinco mensagens. Foi optado este limite para poupar memória nos nós, e porque estes têm um tempo limite de 50 segundos para as mensagens serem reencaminhadas para o servidor. Caso este tempo seja excedido o servidor assume que o nó está desligado e quando recommençar a receber mensagens, será a partir da primeira mensagem que retoma o serviço de TCP.

Se uma mensagem for recebida com sucesso no servidor, será enviada uma mensagem de notificação ao nó para este apagar essa mensagem do seu *buffer*, e a informação recebida é inserida na base de dados[Fig. 11]

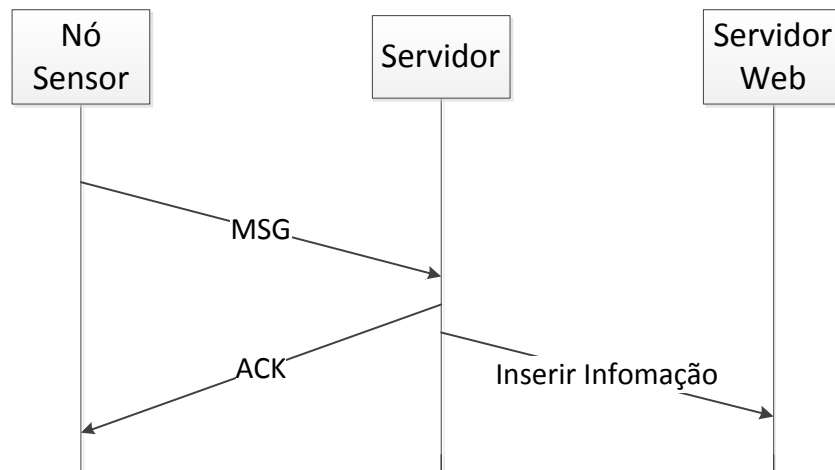


Fig. 11: Mensagem recebida com sucesso.

Caso uma mensagem seja perdida durante o processo de troca de mensagens, é retomada a transmissão dessa mensagem enviando o ACK da ultima mensagem recebida corretamente dentro da ordem esperada [Fig. 12].

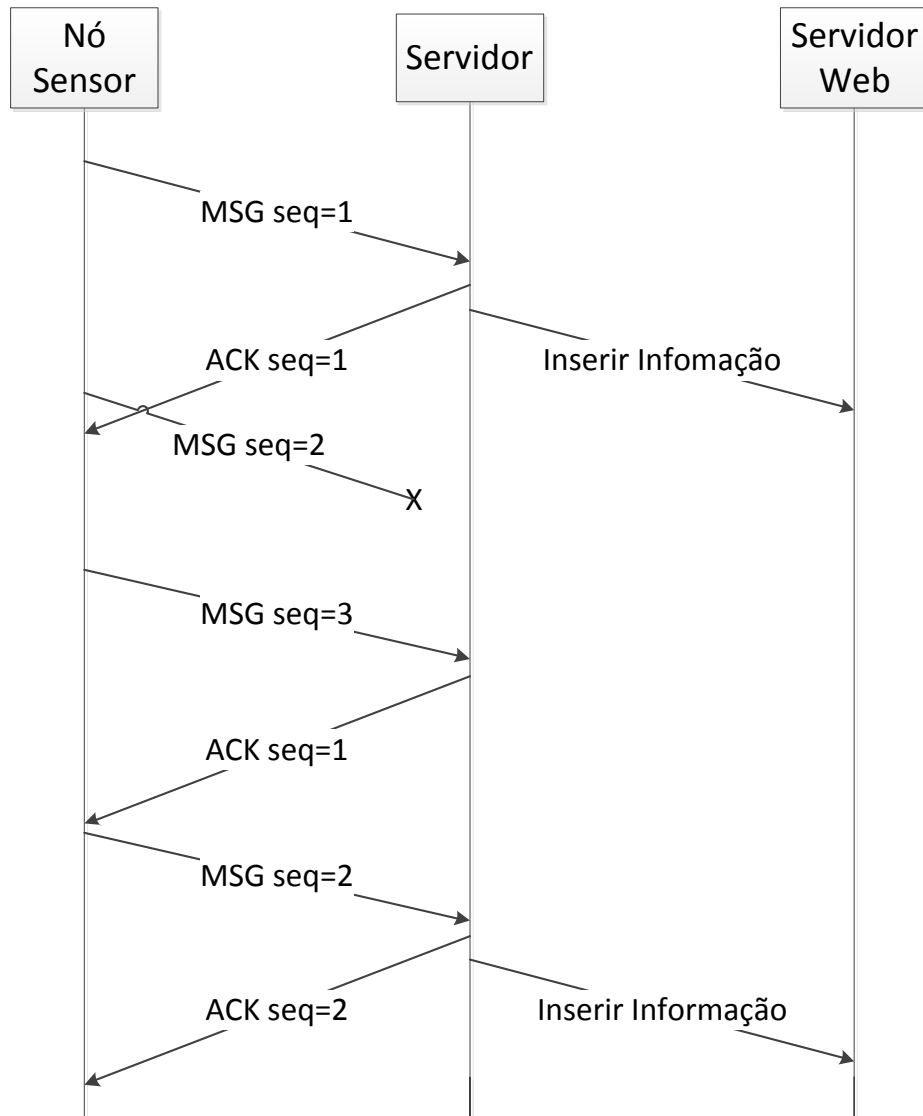


Fig. 12: Mensagem perdida.

Confiança

Neste projeto foi implementado um sistema de segurança para a transmissão de pacotes. Este sistema permite definir níveis de confiança para cada nó, sendo possível assim detectar nós intrusos na rede. Outra funcionalidade é a de ao atribuir um grau de confiança aos vizinhos, este valor é também atribuído de acordo com a probabilidade de esse vizinho entregar a mensagem ao seu destino com sucesso. Assim, permite uma melhor estabilidade e

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

segurança para a entrega das mensagens. Como referencia foi usado o artigo [9], onde é apresentada esta solução que será explorada em trabalho futuro para decrescer os valores de confiança e obter um valor mais apurado, visto que neste momento apenas é incrementado grau de confiança cada vez que uma mensagem é recebida numa escala de 0 a 5, sendo 0 quando nunca foi recebido nenhuma mensagem desse vizinho, e 5 quando foram recebidas 5 ou mais mensagens do mesmo.

Servidor

No servidor processa-se toda a informação recebida da RSSF. Este é constituído por dois programas:

- O serialforwader do TinyOS: que permite ler mensagens da estação base e transmiti-las para outros programas;
- O MessageAnaliser: que processa as mensagens recebidas.

Na Figura 13, está representado o diagrama de classes para o MessageAnaliser, onde se pode ver a interação entre as classes Main e TimeKeeper.

O MessageAnaliser, na classe Main, ao receber uma mensagem detecta o tipo de mensagem. Se a mensagem não for do tipo MSG é descartada. Ao receber a mensagem, caso esta seja a primeira do utilizador, é guardado o número de sequencia e será a partir deste número que vai começar o processo para assegurar a retransmissão de pacotes. Assim sendo, o número de sequencia é verificado para ver se é o esperado. Caso não seja, é enviado o ACK para a ultima mensagem recebida corretamente e descarta-se a mensagem atual. Caso seja o esperado, é então através do objecto do tipo DataMsg criado com a MSG, que é enviada a mensagem do tipo ACK. Seguidamente, é processada toda a informação que vinha nessa mensagem, tal como a conversão das unidades dos sensores e a transmissão das vizinhanças e confiança para o servidor.

Cada vez que é estabelecida comunicação com um nó, é a Thread TimeKeeper que atualiza o seu estado para online. Se um nó se encontrar incomunicável durante 50 segundos é detectado como desligado, e a informação sobre a sua ordem de transmissão de mensagens é descartada, para que quando reestabelecida a comunicação, seja reiniciada a transmissão de mensagens e o bom funcionamento do MyTCP.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

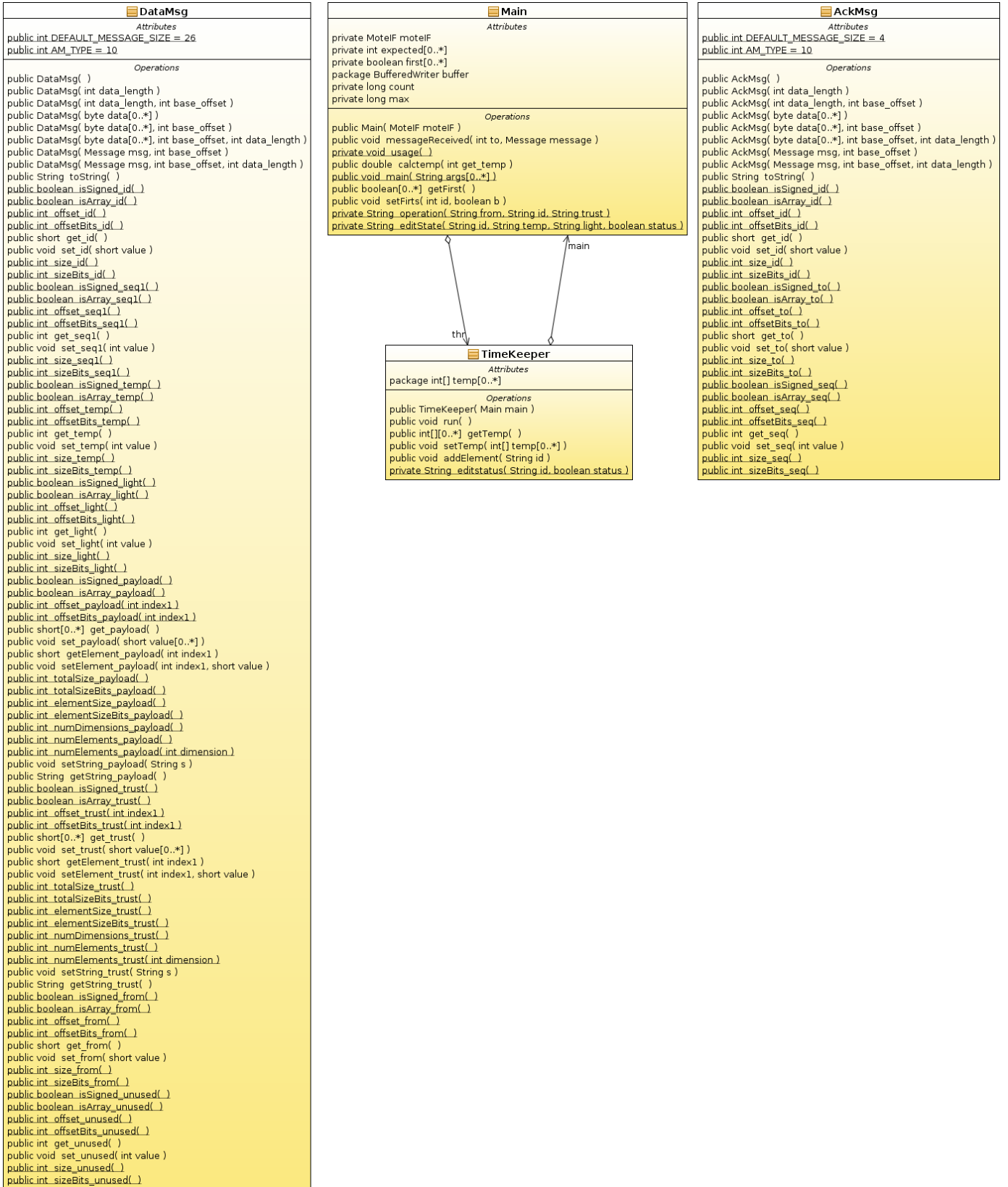


Fig. 13: Diagrama de Classes

Servidor Web

Para o servidor Web foi definido utilizar Java. Este servidor Web é composto por três funções principais: Operation, EditStatus, EditState.

A função Operation permite adicionar a informação sobre um nó que seja vizinho do nó do utilizador em causa. Esta função recebe como entrada o número de utilizador, o identificador para o nó vizinho detectado e o grau de confiança entre o utilizador e o seu vizinho.

A operação EditStatus proporciona a possibilidade de definir o utilizador como ligado ou desligado, caso este esteja ou não a transmitir mensagens para o servidor. Esta função tem como parâmetros o identificador do utilizador e um booleano para definir o seu estado.

EditState é uma função para atualizar os dados dos sensores do utilizador. Esta recebe o identificador do utilizador e os valores para os dois tipos de sensores. Permite também, definir o estado do dispositivo como ligado, necessitando apenas duma conexão á base de dados para inserção dos dados.

Base de Dados

A base de dados é composta por seis tabelas representadas no seguinte diagrama Entidade-Relação [Fig. 14].

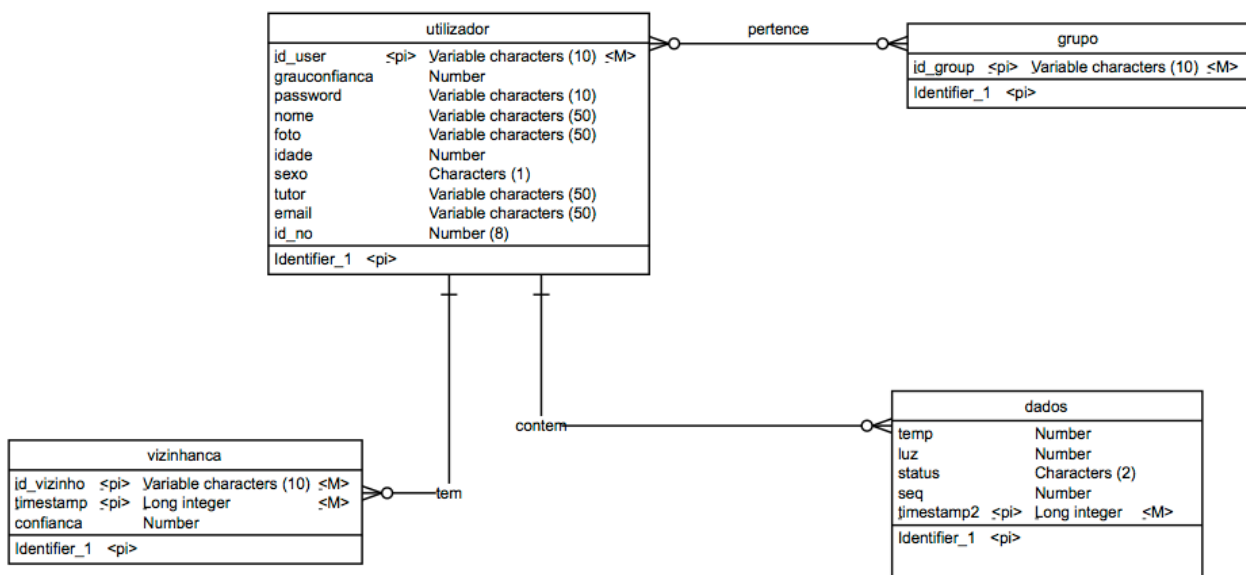


Fig. 14: Diagrama Entidade-Relação

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Neste diagrama, a tabela utilizador é a principal e é nesta que ficam registados os dados do utilizador. Esta tabela interage com as tabelas vizinhança, dados e grupo, onde cada utilizador pertence a um grupo. O grupo terá o valor administrador ou cliente. Cada utilizador contém dados, como os valores dos sensores, o estado da comunicação (se está ligado ou desligado) e o número de sequência das mensagens para análise. Cada utilizador pode ter diversos vizinhos, que são os nós próximos do nó em questão.

Web Site

É neste elemento que toda a informação recolhida pode ser acedida pelos dois tipos de utilizadores, o administrador ou o cliente. Cabe ao administrador criar as contas de cliente. Nestas contas são armazenados os dados do cliente e é-lhe atribuído um identificador numérico correspondente ao nó sensor do utilizador.

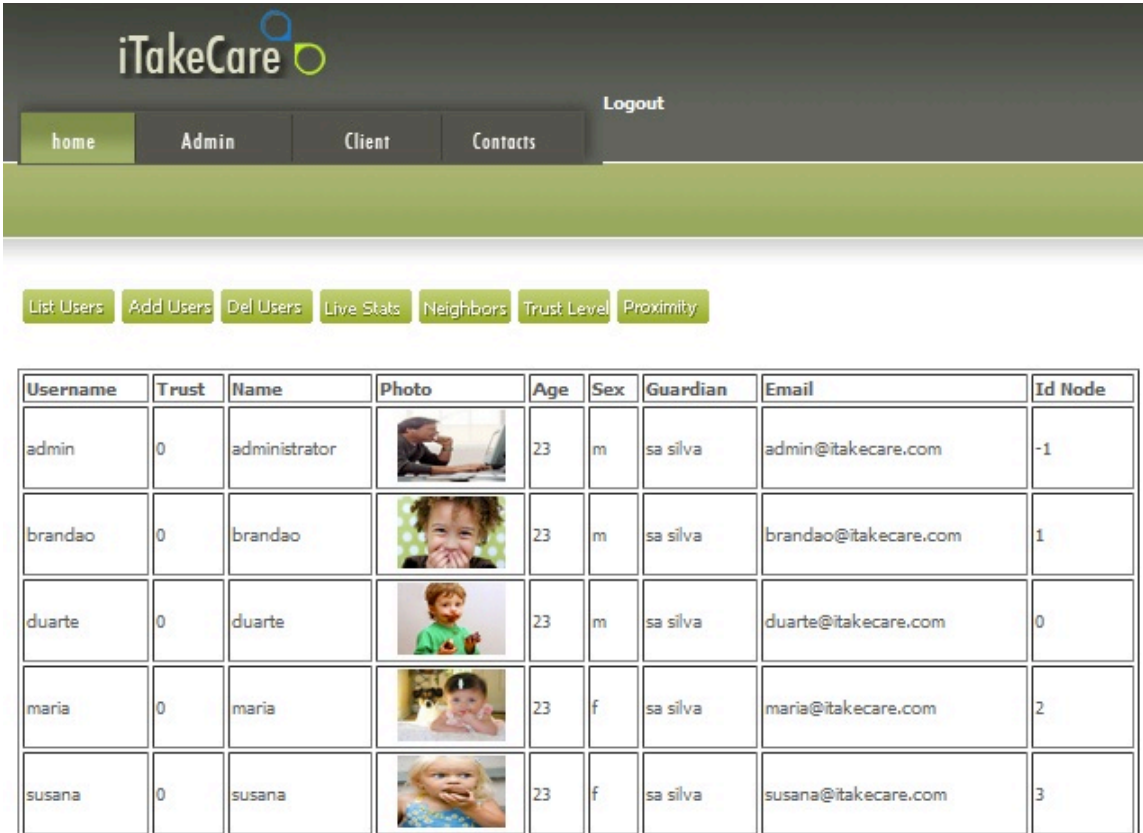
Administrador

Ao administrador é possível visualizar os utilizadores já existentes, adicionar novos clientes ou administradores, apagar tanto utilizadores como apenas a sua informação guardada, ver o estado de todos os sensores em tempo real, analisar graficamente a quantidade de vizinhos em função do tempo para todos os utilizadores, a evolução dos níveis de confiança entre os diferentes dispositivos e as relações de proximidade individualmente.

Seguidamente serão explicadas as diferentes opções de utilização da plataforma para o administrador.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

No separador *List Users* observa-se a informação para todos os utilizadores, tal como os seus nomes de utilizador, nomes, fotos, idades, sexo, tutor, endereço electrónico e número identificador do nó [Fig. 15].







Username	Trust	Name	Photo	Age	Sex	Guardian	Email	Id Node
admin	0	administrator		23	m	sa silva	admin@itakecare.com	-1
brandao	0	brandao		23	m	sa silva	brandao@itakecare.com	1
duarte	0	duarte		23	m	sa silva	duarte@itakecare.com	0
maria	0	maria		23	f	sa silva	maria@itakecare.com	2
susana	0	susana		23	f	sa silva	susana@itakecare.com	3

Fig. 15: Vista *List Users* para o administrador.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Em *Add Users* é permitido adicionar os diferentes utilizadores com os seus dados pessoais, o identificador do nó sensor, fazer *upload* de uma foto, definir o nome de utilizador a palavra-chave e o seu tipo de utilizador. Todos os campos são de preenchimento obrigatório para uma estrutura de dados mais completa[Fig. 16].

The screenshot displays the iTakeCare web application interface. At the top, there is a navigation bar with the logo 'iTakeCare' and a 'Logout' link. Below the navigation bar, there is a horizontal menu with buttons for 'home', 'Admin', 'Client', and 'Contacts'. Under the 'Admin' button, a sub-menu is visible with buttons for 'List Users', 'Add Users', 'Del Users', 'Live Stats', 'Neighbors', 'Trust Level', and 'Proximity'. The 'Add Users' button is highlighted. The main content area shows a form for adding a new user. The form includes the following fields and options:

- First Name:
- Last Name:
- Id Node:
- Email:
- Gender: ☐ M ☐ F
- Age:
- Photo: Nenhum ficheiro seleccionado
- Guardian:
- Username:
- Password:
- User Type: ☐ users ☐ admins

A 'Register' button is located at the bottom right of the form.

Fig. 16: Vista *Add Users* para o administrador.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

O terceiro separador, *Del Users* permite ao administrador eliminar tanto a informação completa dos vários utilizadores, como a informação recolhida pela RSSF e o seu histórico [Fig. 17].

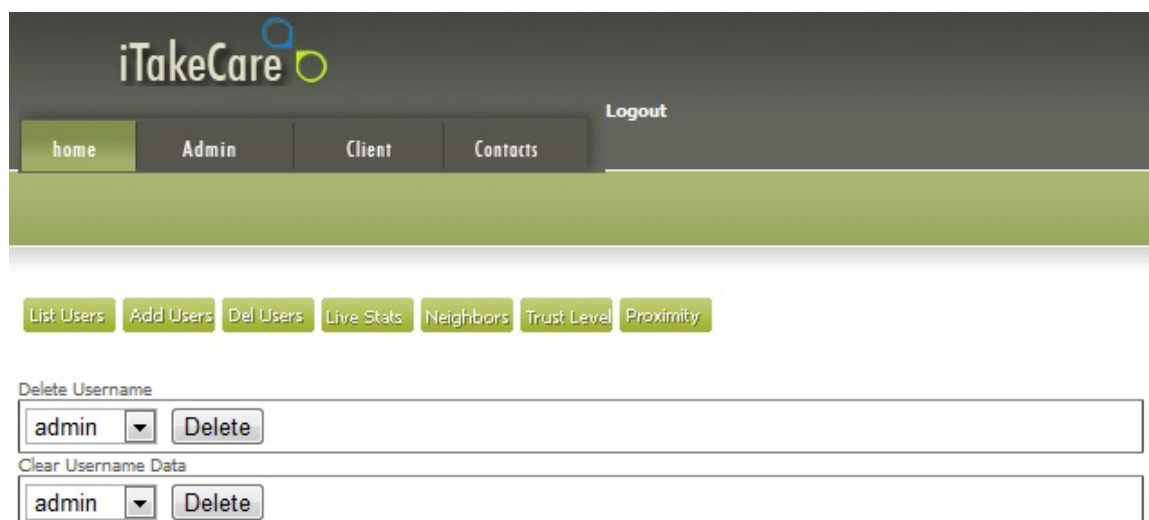


Fig. 17: Vista *Del Users* para o administrador.

Ao administrador através do separador *Live Stats* é possível ver os dados dos sensores e de vizinhança de todos os utilizadores em tempo real. Para mudar de utilizador, basta seleccionar o desejado na caixa de selecção apresentada na figura 18.

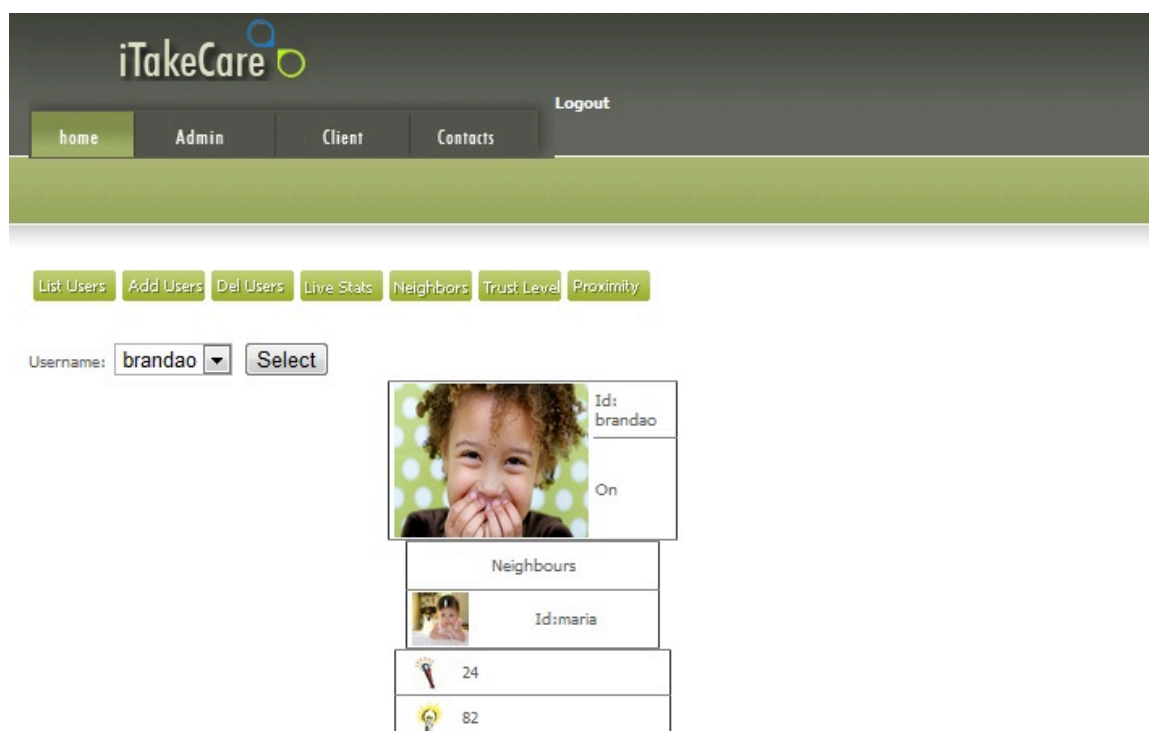


Fig. 18: Vista *Live Stats* para o administrador.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

No separador *Neighbors*, o administrador vê para qualquer utilizador, o número de vizinhos que ele tem no decorrer do tempo. Através da caixa de seleção, o administrador pode mudar o utilizador sobre o qual quer ver a informação [Fig. 19].

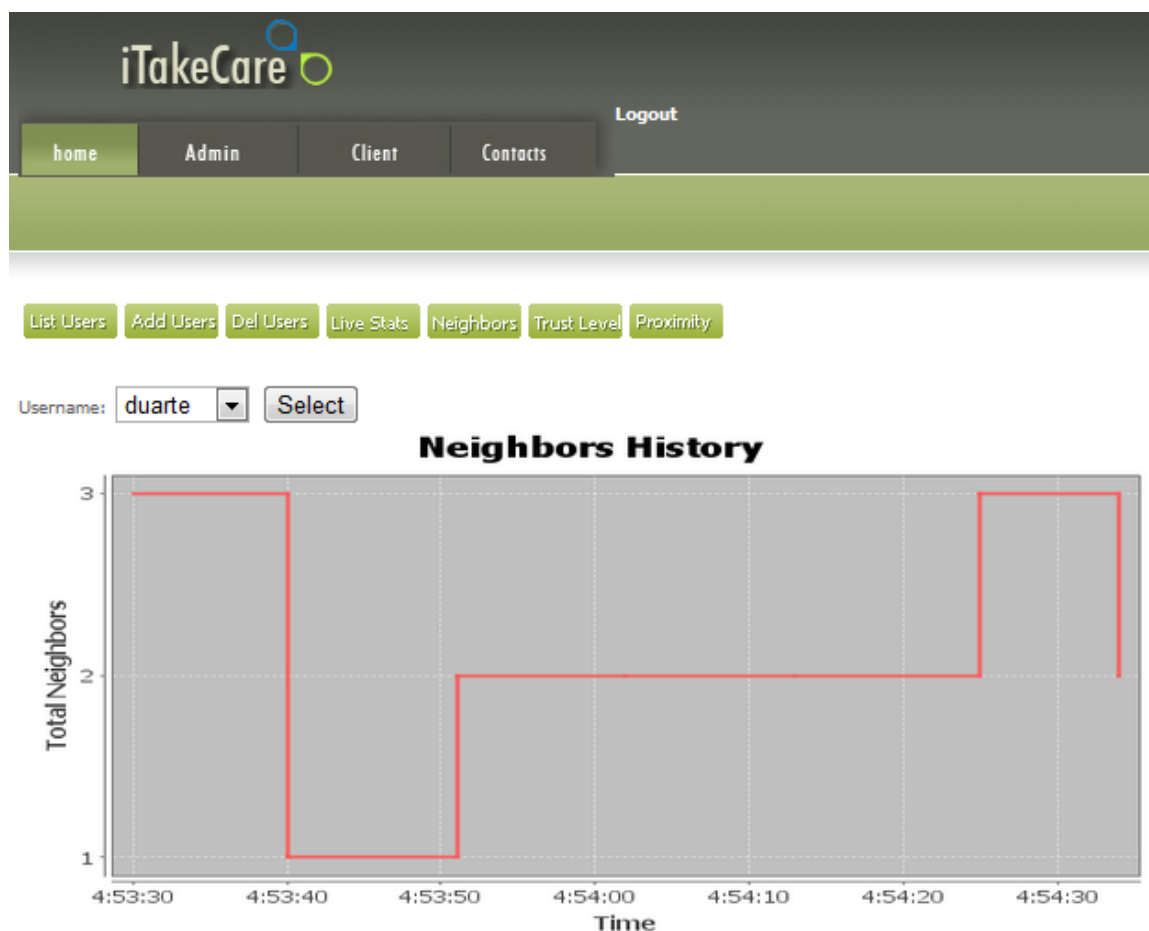


Fig. 19: Vista *Neighbors* para o administrador.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

É possível visualizar o nível de confiança entre os diferentes dispositivos através do separador *Trust Level*. Depois de selecionado, é apresentado um gráfico com as variações na confiança ao longo do tempo. Pode ser visualizado para todos os seus vizinhos ou apenas para um, para melhor análise [Fig. 20].



Fig. 20: Vista *Trust Level* para o administrador.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

O separador *Proximity* permite a visualização da presença de um vizinho ao longo do tempo. Ou seja, pode ser selecionado o utilizador desejado para análise e o seu vizinho, para verificar quando é que estes estiveram próximos. Essa informação é apresentada através de um gráfico, onde o valor 1 indica que eles se encontram [Fig. 21].

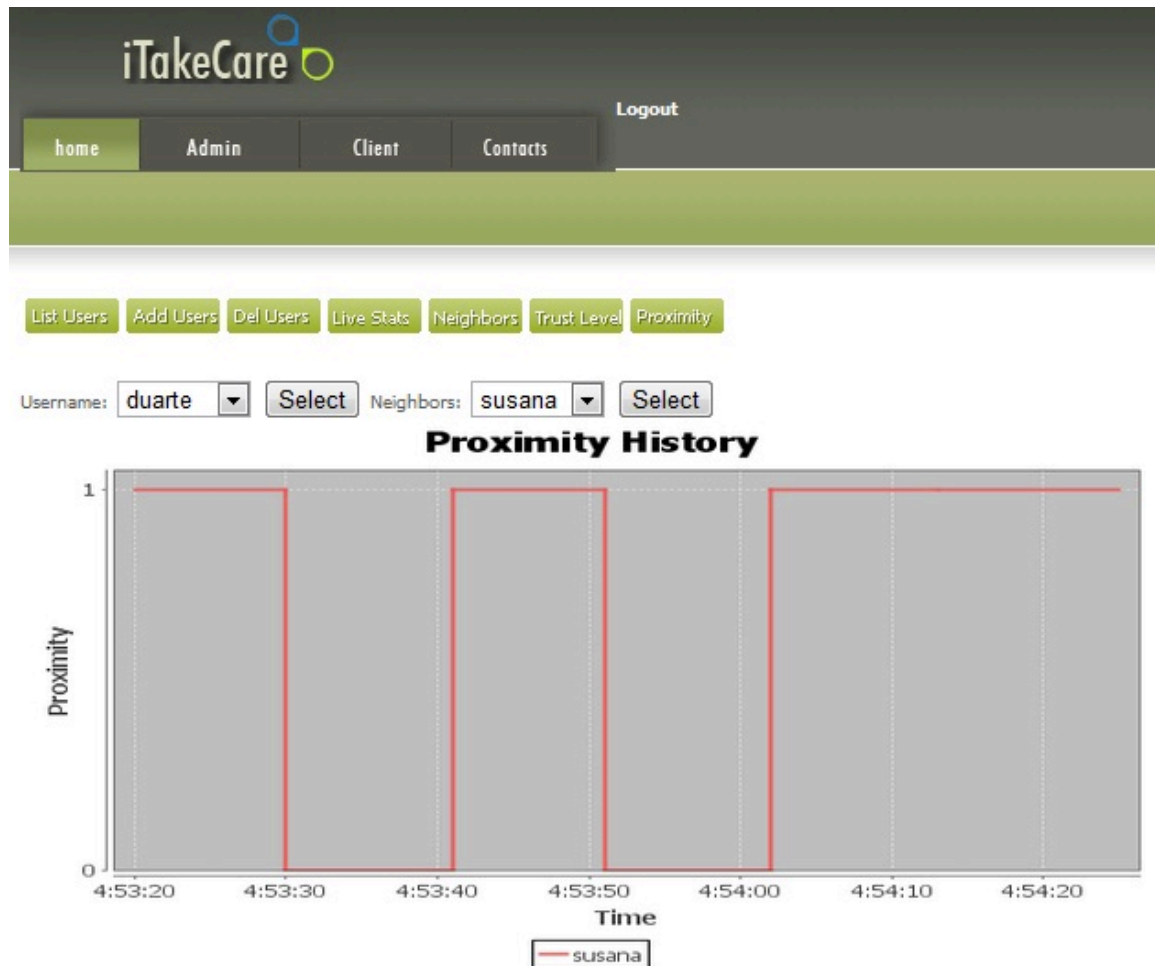


Fig. 21: Vista *Proximity* para o administrador.

Cliente

Cada cliente apenas pode visualizar informação sobre os dados do utilizador que lhe está associado. Assim sendo, não tem tantas funcionalidades como o administrador. Pode visualizar a informação em direto, o número de vizinhos ao longo do tempo, o nível de confiança entre os vários nós vizinhos e a relação de proximidade com vizinhos específicos.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

No separador *Live Stats* visualizam-se os dados em tempo real do nó que lhe está associado, ou seja, os vizinhos próximos nesse instante e a informação sobre os sensores do dispositivo [Fig. 22].

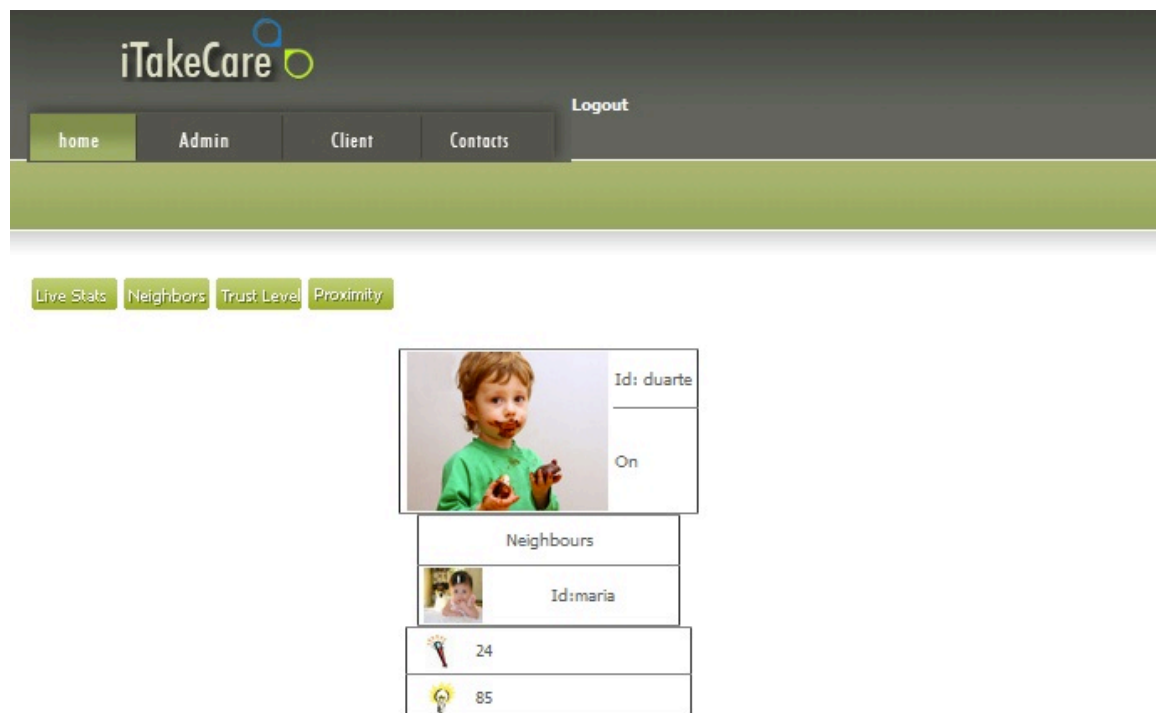


Fig. 22: Vista *Live Stats* para o cliente.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Ao cliente, no separador *Neighbors*, é permitido observar a quantidade de nós vizinhos que o utilizador tem com o decorrer do tempo [Fig. 23].

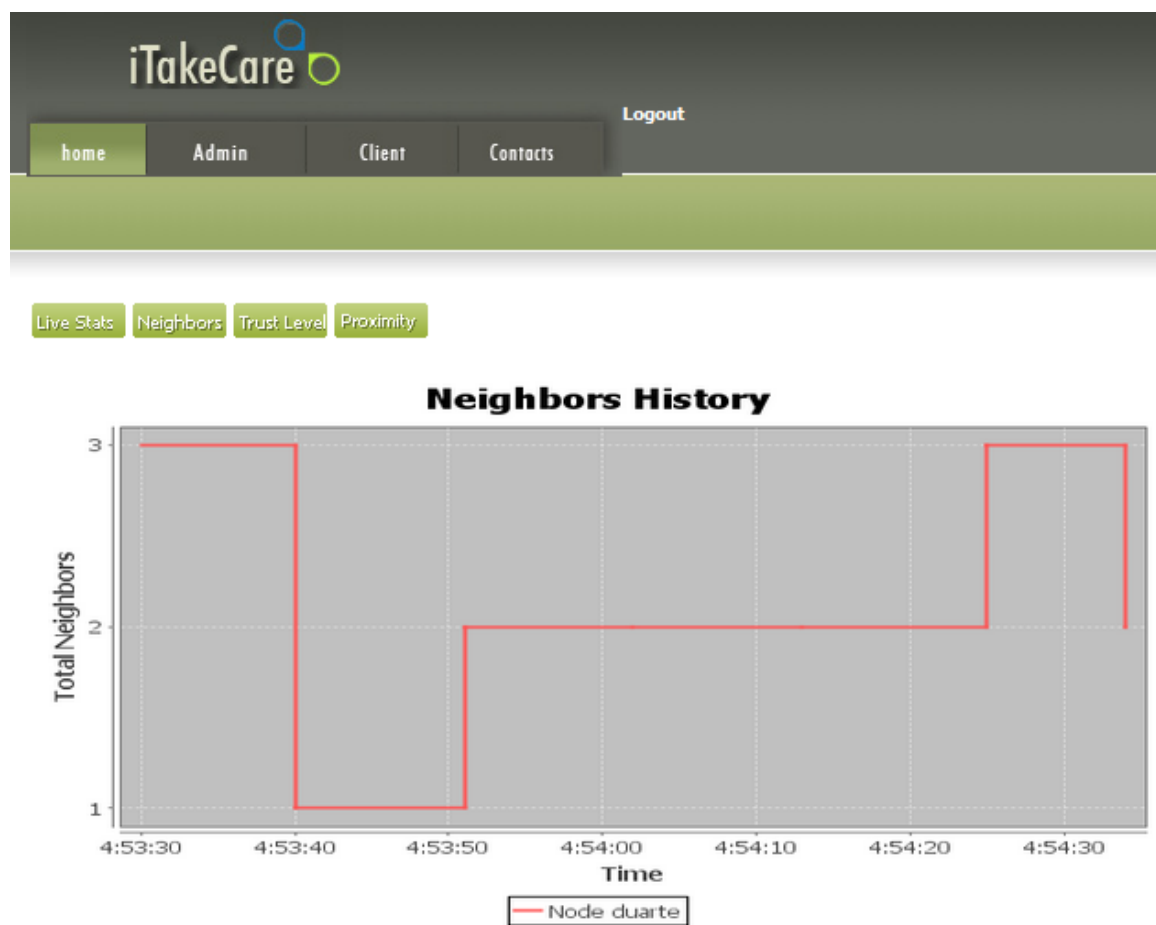


Fig. 23: Vista *Neighbors* para o cliente.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

No terceiro separado, *Trust Level*, são apresentados os níveis de confiança para todos os nós, podendo ser escolhido apenas um. Ao contrário de o administrador, o cliente apenas pode analisar confianças relativas ao nó que lhe foi atribuído[Fig. 24].



Fig. 24: Vista *Trust Level* para o cliente.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Como ultimo separador temos o *Proximity*, onde pode ser verificada a proximidade individual com diferentes vizinhos. Aqui o valor de proximidade varia entre 0 ou 1, sendo 1 quando o nó se encontra próximo e 0 quando ele está afastado do limite [Fig. 25].



Fig. 25: Vista *Proximity* para o cliente.

4.2 Deployment

O propósito desta ferramenta é analisar o estado de comunicação entre o *sink* e os nós independentemente do estado e configuração física da rede. No entanto, está otimizada para funcionar em redes do tipo TDMA. Estas redes são desenhadas com divisão temporal, para cada nó fazer as suas transmissões.

No caso de uma rede TDMA, há alguns problemas que podem ser excluídos, tais como, a congestão, a descoberta de vizinhos, os *loops*, e as leituras erradas dos sensores. Isto porque no caso da congestão, cada nó tem um tempo reservado para a sua transmissão, não congestionando a rede na descoberta de vizinhos, pois todos os nós têm de saber á priori os seus vizinhos para um caminho de comunicação. Como só existe um caminho de comunicação não existem *loops*. As leituras erradas dos sensores derivam de problemas de hardware ou de situações que não podem ser detectadas ou reparadas sem intervenção humana. O processo de eliminação destes problemas é efectuado após a recepção das mensagens no *sink*. Os dados da recepção das mensagens são analisados pela ferramenta, para detectar quaisquer anomalias que possam ser corrigidas de imediato. Como o mundo real é diferente dos cenários de ensaio, podem aparecer problemas que não era suposto. Mas, esta ferramenta permite avaliar e determinar que erros acontecem e que devem ser corrigidos durante a fase de colocação dos nós, para que deixe de ser necessário voltar a intervir nas suas localizações devido a problemas de comunicação. Desta forma, o sistema torna-se mais viável para o trabalho futuro da RSSF.

Este sistema terá um programa que correrá no servidor, onde o *sink* está conectado, e mostrará os dados ao utilizador bem como os problemas existentes na rede associada [Fig. 26].

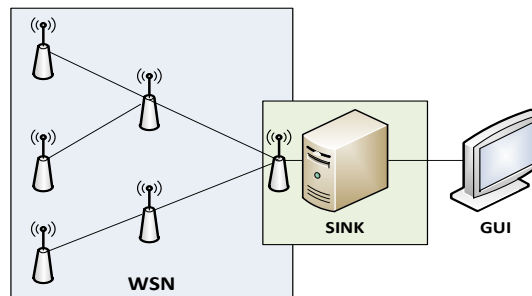


Fig. 26: Arquitetura da ferramenta de detecção de erros

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Para detetar erros e identificar o problema do nó em análise é seguida a árvore de decisão representada na figura 27.

Caso seja recebida uma mensagem durante o tempo em que é suposto o nó comunicar essa mensagem, passamos ao ramo do teste de latência. Ao passar neste teste, é verificada a integridade da mensagem e seu o identificador da. Ao ultrapassar este testes pode-se afirmar que o nó esta a funcionar corretamente. Caso não seja recebida nenhuma mensagem no tempo reservado para o nó, é verificado o caminho de comunicação, e é tentada uma comunicação através de um *ping* ao nó. É então designado um erro associado ao nó.

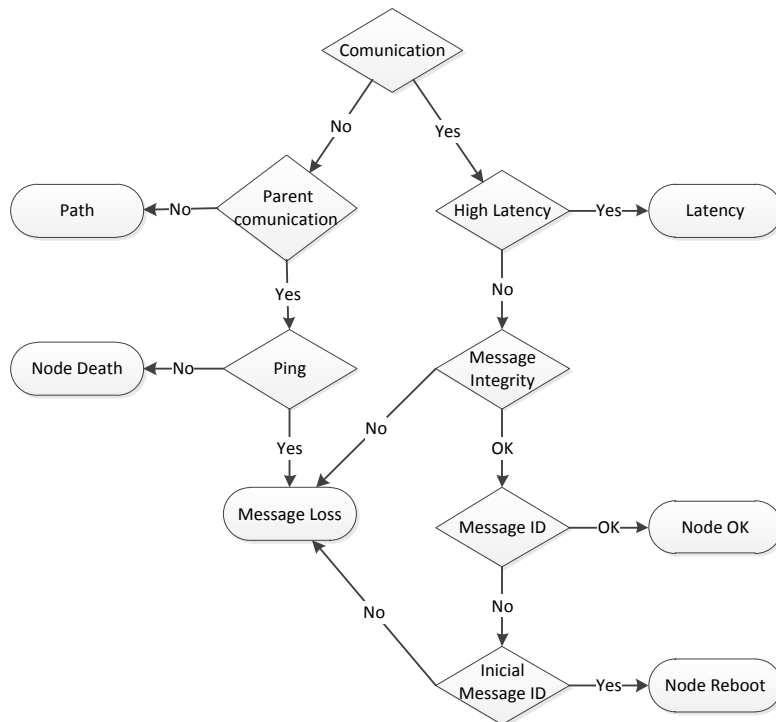


Fig. 27: Árvore de decisão para detecção de problemas.

4.3 Avaliação Funcional

Para a avaliação da aplicação foram efectuados alguns testes, os quais serão enumerados na seguinte tabela.

Comunicação 1Nó-Sink	PASSOU
Comunicação 1Nó-1Nó	PASSOU
Comunicação 1Nó-1Nó-Sink	PASSOU
Leitura do Sensor1	PASSOU
Envio da informação do Sensor1 para o <i>Sink</i>	PASSOU
Leitura do Sensor2	PASSOU
Envio da informação do Sensor2 para o <i>Sink</i>	PASSOU
Contador para decrescer tempo de proximidade de outros nós	FALHOU
Aumento do nível de confiança de nós vizinhos	PASSOU
Criação e envio de mensagem completa para o <i>Sink</i>	PASSOU
Recepção de ACK	PASSOU
Envio de ACK por parte do <i>Sink</i>	PASSOU
Comunicação com o servidor Web	PASSOU
Apresentação de informação no Web Site	PASSOU
Criação de perfis no Web Site	PASSOU
Comunicação entre Web Site e base de dados	PASSOU
Armazenamento de informação recebida no servidor	PASSOU
Funcionamento total da Aplicação com 1 Nó + <i>Sink</i>	PASSOU
Funcionamento total da Aplicação com 2 Nó + <i>Sink</i>	PASSOU
Funcionamento total da Aplicação com 3 Nó + <i>Sink</i>	PASSOU

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

O contador para decrescer o tempo de proximidade de outros nós falhou, pois excedia a memória do nó. O temporizador de segundo a segundo, iria aceder á tabela e baixar os valores de *timestamp*. Ao mesmo tempo, eram recebidas novas mensagens de outros nós, e a mesma tabela era acedida para atualizar os dados referentes a novas proximidades. Assim, para contornar este problema, deixou de se ter um tempo exato ao segundo, passando a um intervalo de 10 segundos, pois cada vez que uma mensagem é enviada para o *sink* a tabela de proximidades é apagada para por novas entradas.

A aplicação foi testada com quatro nós sensores a transmitir entre si e com mais um a fazer de *Sink* ligado ao servidor. Nestes testes foram variadas as posições e proximidades dos nós.

Na tabela seguinte serão demonstrados os testes efectuados em função do número de nós vizinhos que cada nó tem.

0-0-0	PASSOU
0-1-1	PASSOU
1-2-1	PASSOU
2-2-2	PASSOU

4.4 Avaliação de Desempenho

Neste capítulo, serão apresentados os resultados dos testes realizados para análise da performance do protocolo de TCP implementado. Estes testes foram realizados com um, dois e três nós sensores.

Cada teste teve a duração de 10 minutos, o que equivale a cerca de 60 mensagens enviadas para o *sink* por cada nó sensor. No decorrer dos testes a posição dos nós foi variada, ficando sempre todos os nós sensores a uma certa distancia de comunicação do *sink*. Existiu uma exceção para um dos testes com três nós, onde foi movido o nó número 3 para fora de alcance, provocando assim perdas de mensagens, para testar a performance do protocolo com perdas sucessivas de mensagens.

Nos testes realizados com um e dois nós sensores, não foram detectadas quaisquer perdas de pacotes na comunicação entre o *sink* e estes nós.

Proposta e avaliação de um mecanismo de análise de proximidade baseado em Redes de Sensores Sem Fios

Já nos testes com três nós, podemos detectar algumas perdas, sendo posteriormente estas mensagens retransmitidas.

No gráfico 1, perde-se uma mensagem quando os três nós estão em comunicação e sempre dentro do alcance do *sink*. Com este raio de comunicação não se perde uma grande percentagem de mensagens.

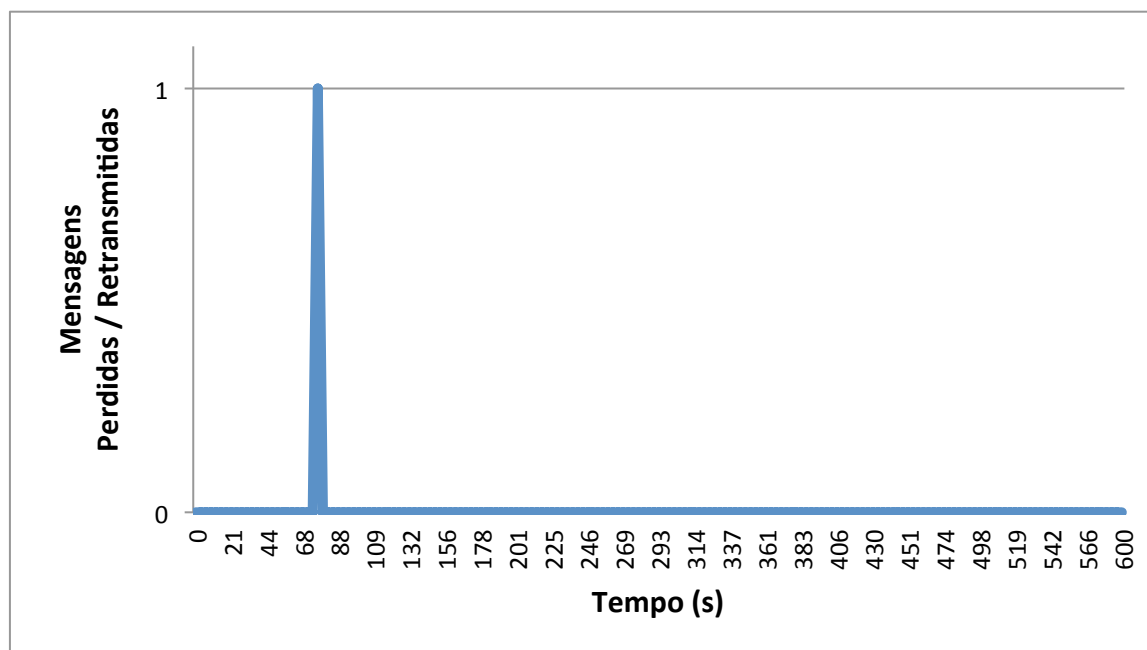


Gráfico 1: Mensagens perdidas/retransmitidas com 3 nós.

No entanto no gráfico 2, foram introduzidas perdas de mensagens no nó com o id 3. Estas perdas estão discriminadas no gráfico 3. Quando existe uma comunicação com mais retransmissões de pacotes, são perdidas mais mensagens. Todavia, todas as mensagens perdidas são retransmitidas.

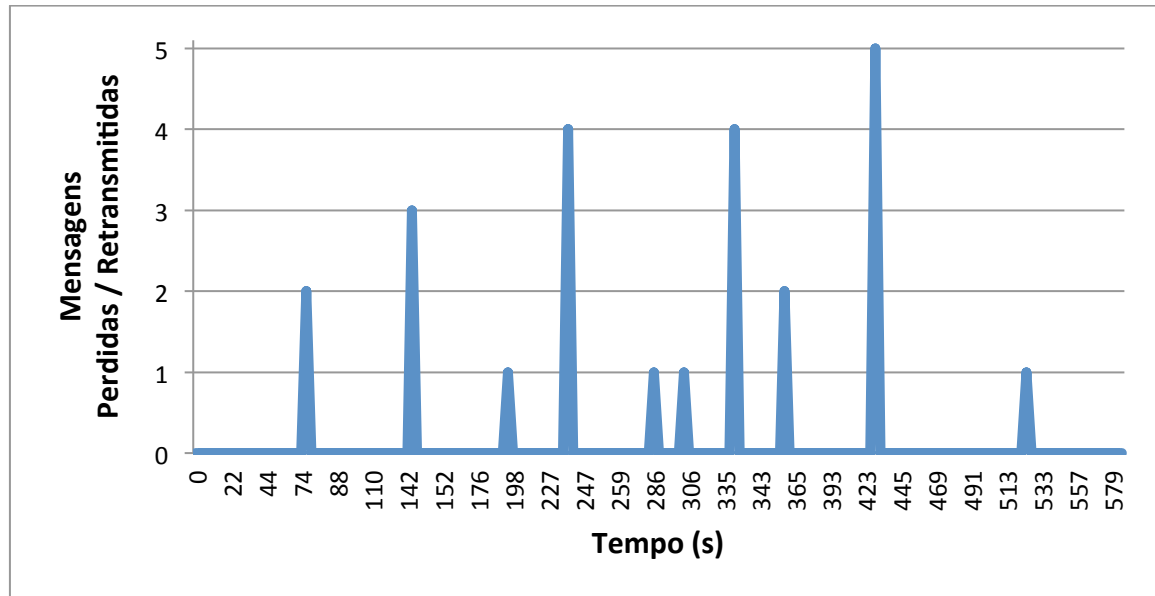


Gráfico 2: Mensagens perdidas/retransmitidas com 3 nós e perdas introduzidas intencionalmente no nó 3.

Aproximadamente ao segundo 425, é introduzida uma perda total de mensagens, ou seja, são perdidas pelo menos 5 mensagens, e o nó é assinalado como offline no servidor. Ao provocar o cenário anterior, o nó apaga todas as mensagens que tem em memória. Quando a comunicação for reestabelecida, o nó começa a transmitir normalmente a partir do número de sequência da primeira mensagem recebida no servidor, voltando assim a guardar as mensagens em *buffer* para efetuar retransmissões, caso seja necessário.

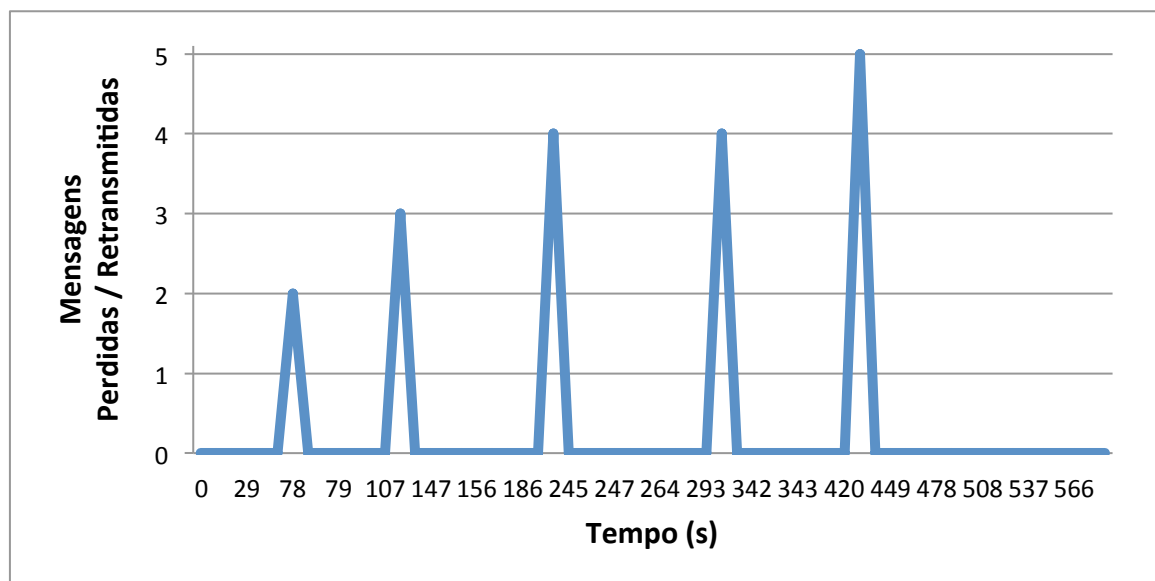


Gráfico 3: Mensagens perdidas/retransmitidas introduzidas intencionalmente no nó 3.

Com estes testes foi possível comprovar que o mecanismo MyTCP funciona corretamente e trás grandes vantagens para um sistema de RSSF, pois é possível garantir a entrega de mensagens aumentando assim a QoS do sistema.

Capítulo 5

Conclusão

5.1 Considerações

Este trabalho oferece uma mais valia para a análise de comportamentos sociais, por exemplo, de crianças. É possível analisar o comportamento e relações sociais entre diversos sujeitos de forma não intrusiva, podendo esta informação ser analisada em direto á distancia. Mas, também poderá guardar a informação para posterior análise. Outra vertente deste projeto é a confiança atribuída a nós vizinhos nas RSSF. Este parâmetro trás mais segurança e confiança ao próprio sistema e á entrega de mensagens nas RSSF.

5.2 Trabalho futuro

Como trabalho futuro ficou definido, finalizar a implementação da variação dos níveis de confiança.

Deverão ser acrescentadas algumas funcionalidades no Web Site, nomeadamente a alteração os dados dos seus perfis no grupo cliente, para que com o decorrer do tempo se mantenham atualizados, e também o envio de alarmes para o endereço electrónico e ou telemóvel do tutor.

Este trabalho poderá ainda permitir a realização de alguns artigos científicos nas áreas das RSSF, de cuidados de saúde e cuidados psicológicos.

Referências

- [1] M. Srivastava, R. Muntz, and M. Potkonjak, “Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments,” *ACM SIGMOBILE 7th Annual International Conference on Mobile Computing and Networking*, 2001, pp. 132-138.
- [2] Dunkels, J. Alonso, and T. Voigt, “Distributed TCP Caching for Wireless Sensor Networks,” *Science*, 2004, pp. 1-17.
- [3] Crossbow MICAz DataSheet
http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf
- [4] C. Buschmann, D. Pfisterer, S. Fischer, and A. Kröller, “SpyGlass : A Wireless Sensor Network Visualizer,” *ACM SIGBED Review - Special issue: Best of sensys 2004 work-in-progress*, vol. 2, 2005.
- [5] J. Pinto, A. Sousa, P. Lebres, G.M. Gonçalves, and J. Sousa, “MonSense – Application for Deployment , Monitoring and Control of Wireless Sensor Networks,” Poster in ACM RealWSN’06, 2006.
- [6] M. Ringwald, K. R, and A. Vitaletti, “SNIF : Sensor Network Inspection Framework,” Technical Report 535, 2006.
- [7] B. Parbat, a K. Dwivedi, and O.P. Vyas, “Data Visualization Tools for WSNs: A Glimpse,” *International Journal of Computer Applications*, vol. 2, May. 2010, pp. 14-20.
- [8] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, “RFC4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” RFC4944, 2007.
- [9] G. Zhan, W. Shi, and J. Deng, “TARF : A Trust-aware Routing Framework For Wireless Sensor Networks,” *Networks*.
- [10] G. Pau and G. Scatà, “WSN / RFID Architecture For Children ’ s Monitoring,” *Architecture*
- [11] N. Heo, “An Intelligent Deployment and Clustering Algorithm for a Distributed Mobile Sensor Network *,” *Electrical Engineering*, 2003, pp. 4576-4581.
- [12] X. Wang, J. Ma, and S. Wang, “Collaborative Deployment Optimization and Dynamic Power Management in Wireless Sensor Networks,” *2006 Fifth International Conference on Grid and Cooperative Computing (GCC’06)*, 2006, pp. 121-128.

- [13] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J.A. Stankovic, “An Advanced Wireless Sensor Network for Health Monitoring,” *Office*, pp. 2-5.
- [14] O.B. Akan and I.F. Akyildiz, “Event-to-sink reliable transport in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 13, Oct. 2005, pp. 1003-1016.
- [15] J. a Stankovic and T. Abdelzaher, “SPEED: a stateless protocol for real-time communication in sensor networks,” *23rd International Conference on Distributed Computing Systems, 2003. Proceedings.*, pp. 46-55.
- [16] K. Akkaya and M. Younis, “An energy-aware QoS routing protocol for wireless sensor networks,” *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, 2003, pp. 710-715.
- [17] Q. Zhao and L. Tong, “Quality-of-service specific information retrieval for densely deployed sensor networks,” *IEEE Military Communications Conference, 2003. MILCOM 2003.*, pp. 591-596.
- [18] Baker, C., Light, J., Tulpan, D., Vecchio, A., Saint, U. N. B., Unb, J., & John, S. (2009). Non-intrusive Patient Monitoring of Alzheimer ’ s Disease Subjects Using Wireless Sensor Networks, 161-165. doi:10.1109/CONGRESS.2009.25
- [19] Malan, D., Fulford-jones, T., Welsh, M., & Moulton, S. (n.d.). CodeBlue : An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. *Robotics*, 3-6.
- [20] Alliance, Z. (2009). ZigBee Wireless Sensor Applications for Health , Wellness and Fitness. *World Health*, (March).
- [21] Ko, J., Lu, C., Srivastava, M. B., Stankovic, J. A., Terzis, A., & Welsh, M. (n.d.). Wireless Sensor Networks for Healthcare. *World Trade*.
- [22] Lo, B. P. L., Thiemjarus, S., & King, R. (n.d.). BODY SENSOR NETWORK – A WIRELESS SENSOR PLATFORM FOR PERVASIVE HEALTHCARE MONITORING. *Architectural Design*, 77-80.
- [23] Lee, R.-guey, Lai, C.-chih, Chiang, S.-shan, Liu, H.-sheng, Chen, C.-chang, & Hsieh, G.-yu. (2006). Design and Implementation of a Mobile-Care System over Wireless Sensor Network for Home Healthcare Applications. *Heart Disease*, 6004-6007.

ANEXOS

Deployment Tool for Wireless Sensor Networks

José Brandão
Laboratório de Comunicações e Telemática
FCTUC
Coimbra, Portugal
jbrandao@student.dei.uc.pt

Abstract - In this paper we report some of the problems usually associated in Wireless Sensor Networks (WSN) and present a framework to help in the deployment, to analyze and detect problems for a more, better and reliable WSN.

1. Introduction

Deployment is one of the most important phases in Wireless Sensor Network (WSN) and it happens in all kind of WSN. We'll present a technic of deployment for a Time Division Multiple Access (TDMA) network. Deployment is a hard task to develop, since there are many physical and connection problems that interfere in the good performance of a WSN, those are detailed in Section 2. In the majority of cases, motes are planted randomly over the field, which can reduce the performance of a WSN, since there are several field problems that can't be predicted without the analyses and tests, this tool will reduce significantly the work and tests to be made in the field, because it will be used in a live deployment scenario in the deployment phase. In this case we present a tool to add more reliability to the GINSENG project, this work is part of a study this project research group, in order to extend the system reliability.

1.1. GINSENG Project

GINSENG is a European project that includes several institutes from all over Europe. These are University College Cork, University of Coimbra, Petrogal SA, SAP AG, Swedish Institute of Computer Science AB, Technische Universitaet Carolo-Wilhelmina zu Braunschweig, Univeristy of Cyprus and Lancaster University. The GINSENG main objective is to enable

performance control and manage WSN in critical environments. The project goal requires innovative approach to wireless sensor network research. Firstly, GINSENG adopts a planned approach for sensor node deployment as a basis to enable performance control. The second basis of GINSENG is software with assured performance, including operating systems and protocols for radio medium access. The third basis of GINSENG is the algorithms that ensure control with respect to network topology and traffic. Recognizing the inherent uncertainties of the real-world, GINSENG will provide performance and debugging of deployed systems.[4] For this project the test case is the Petrogal Refinery in Sines, Portugal. Where it is being deployed a WSN with multiple sensor nodes to monitor the refinery, both pipeline and health status of workers when in contact with toxic substances, for it will be protective suits with sensor nodes integrated. These suits will be monitoring vital signs for safety, because even with the suits to protect, the environment is very toxic and dangerous for a long term presence.

1.2. GinMAC

This is the Communication protocol used in GINSENG, it's based on Time Division Multiple Access (TDMA), and this means that each node has a number (n) of slots reserved for his communication for each transmission cycle (Epoch) [Fig 1].

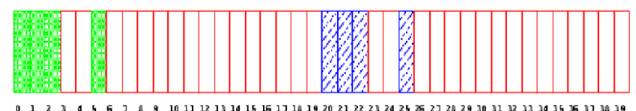


Fig 1: TDMA Epoch example

Being k the number of nodes in the network, the total time of the epoch is $n \times k$ plus some extra slots used for special cases (s).

This Give us the Epoch total time (E_{tt}) by the formula:

$$E_{tt} = (n \times k) + s$$

On the specific case of GINSENG it's used Exclusive TDMA protocol, which limits the max number of nodes to 25 per network ($n=25$), and the associated slots for each node are 2 ($k=2$). This way the max number of communication in each epoch is $25 \times 2 = 50$ communications plus the extra slots.

In this protocol the placement of the nodes must be previously known. It must be in a tree architecture which means that communication works in a hierarchical state that the leaf nodes communicate through their parents to reach the sink using multi-hop.

2. Wireless Sensor Network Problems

There are several types of problems in a WSN, for me they are distributed in three types: Physical Problems, Connection Problems and Network Problems. Each one of them has many problems associated such as:

Physical Problems:

- Node Death
- Node Reboot
- Wrong Sensor Readings

Connection Problems:

- Discovering Neighbors
- Message Loss
- Latency
- Congestion
- Path
- Loops

Network Problems:

- Short Lifetime
- Low Data Yield

2.1. Physical Problems

These types of problems are associated with the hardware, the nodes and the sensors.

Node Death: Is a node that stopped transmitting to the network, the problem can rely on lack of battery or even because it's broken.

Node Reboot: When a node reboot during a transmissions, this reboot can be caused by some issues on the exchanged messages.

Wrong Sensor Readings: due to a malformed sensor or reduced battery the node can have problems on sensor readings that don't give the real results.

2.2. Connection Problems

Problems associated with in transaction of information and connectivity state.

Discovering Neighbors: When the node doesn't know his neighbors it need to discover them, so it can connect to the WSN.

Message loss: When a message don't arrive to the destination, caused by some problem in the network that don't deliver it.

Latency: Can cause delays in the overall processing of data, especially when dealing with priority and critical data.

Congestion: Can cause some message loss due to packet collision and overhead processing in the nodes.

Path: When can't be found a path from one node to other, like the communication to the sink has been broke because some middle node as died

Loops: A message while trying to be sent to the sink can be lost in the networking, making loops trough the nodes and don't reach the sink.

2.3. Network Problems

Problems related to the whole network itself. Problems those are associated with the set of nodes.

Short Life Time: Due to large amount of traffic the nodes can have a lot of messages to pass and analyze, that can reduce significantly the battery of the nodes which reduces the lifetime of the whole network.

Low Data Yield: Means that network doesn't deliver enough information for processing, mainly because of messages lost.

3. Related Work

There are several interesting tools to help deployment in WSN, these tools are specialized in analyze and record the messages that are transmitted over the network. Some tools just analyze the messages in the sink and make the analyses of the state of the network from there.

In this segment we have MonSense[1], an application that has only support for TinyOS and analyses the messages that arrive to the sink. It has some enhanced features like display the nodes in a real satellite image map for better understanding.

SpyGlass[2] analyses the messages as they came to the sink, here it has a gateway that passes the messages to a different visualizer through TCP/IP, it hasn't TinyOS support or a mode to display the network according to the real world placement of the nodes, like the map in the case of MonSense.

Sensor Network Inspection Framework (SNIF)[3] allows inspection and debugging of a WSN using special nodes with two interfaces of communication, this way it has an interface to capture all the messages in the network and using other interface to transport messages back to a base station that has a GUI of the network state. This nodes should be implemented in the middle of the network and be in range of all the nodes that are to be monitored. The second interface of communication that's used is Bluetooth, because is more robust and for an optimal state of the network is better to don't lost messages that can contain important details about the state of the whole network. This tool can also be useful to monitor the state of the network if used in a fully

deployed WSN. The main problem of SNIF is that Bluetooth works under 2.4GHz and the 802.15.4 interface of BTnodes that they use are in 433-915 MHz. In the case of GINSENG communications over 802.15.4 are already in 2.4GHz so adding nodes with Bluetooth will add more noise and possibly provoke extra packet losses.

4. Proposal

Under Ginseng Project my proposal is to develop a framework to aid in deployment of Ginseng WSN for a reliability enhancement. The work of this framework is to analyze the communications between the sink and the nodes so it can help in the placement of the nodes for a best place for communication. Since the real world is different then a test bed, we can face real problems while deploying so this framework will let the user know the state of the network to make some adjustments in order to get a best state of communication. This way the system it will get more reliability for future work, on the purpose of the WSN.

4.1. Ginseng Problems

Since GinMAC is an exclusive TDMA protocol there are some problems that we can exclude from our list, because in this protocol we don't have congestion, discovering neighbors or either loops. Congestion because each node has its slots to communicate and don't use others. Discovering Neighbors don't happened because the topology of the network is known before the deployment. Loops aren't an issue because there's only one path of communication to the sink.

This leaves us with these problems to detect and correct in the deployment phase:

- Node Death
- Node Reboot
- Wrong Sensor Readings
- Message Loss
- Latency
- Path
- Short Life Time
- Low Data Yield

4.2. Architecture

The Architecture is based on the Sink as a gateway for the information to be analyzed by a user through the Graphical User Interface (GUI) as can be seen in Figure 2.

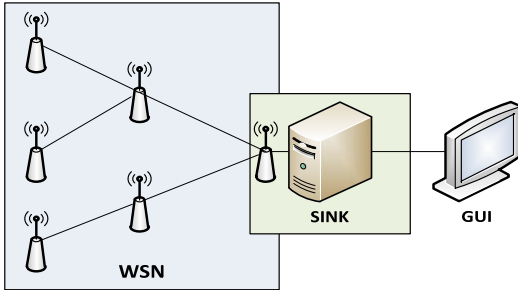


Fig 2: System Architecture

4.3. Message analyze

Since GinMac is a TDMA based protocol, each node have his slots to communicate, by that time the communication is analyzed by the decision tree[Fig 3] to detect if there is any problem between the Sink and the respective node. If the communication is working, the message received pass through the tests of Latency, integrity and ID verification, if it fails in any of this test the associated problem is diagnosed the node. If the node don't communicate in the specific time, its needed to verify if his parent is alive, and if he communicates, it's made a special communication to verify if the node respond or if it's dead.

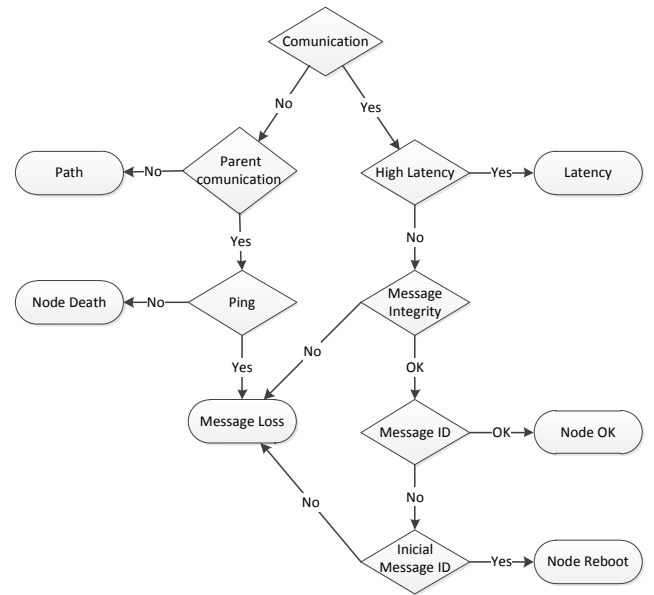


Fig 3: Problem detection decision Tree

5. Conclusion and Future Work

This Framework will be very useful under the deployment task to prevent later problems like message loss, in a critical environment like in the GINSENG project. It will detect the main problems that persist in a WSN TDMA base, these are Path problems, Node Death, Latency and reboot.

As Future work this tool can be used for control, monitoring and maintenance of a WSN. At this level it will be also implemented an error control protocol to prevent message loss to enhance the reliability of the network.

6. References

- [1] "MonSense – Application for Deployment, Monitoring and Control of Wireless Sensor Networks"
- [2] "SpyGlass: A Wireless Sensor Network Visualizer"
- [3] "SNIF: Sensor Network Inspection Framework"
- [4] About GINSENG, <http://www.ict-ginseng.eu/about.php>