

Avaliação de Estratégias e Construção de Software para Medição do Nível de Energia em Sensores de Redes de Sensores sem Fio (RSSF)

Fábio C. ARAÚJO (1); Flávius ANDERSON (2); Katyusco de F. SANTOS (3)

(1) CEFET-PE, Av. Prof Luiz Freire 500 Cidade Universitária, CEP: 50740-540, e-mail: engfabioaraujo@yahoo.com.br

(2) CEFET-PE, e-mail: flavius.anderson@gmail.com

(3) CEFET-PE, e-mail: katyusco@gmail.com

RESUMO

Redes de Sensores sem Fio (RSSFs) são compostas por vários nodos (dispositivos de *hardware* com capacidades de sensoriamento, armazenamento, processamento, transmissão de dados, e bateria). As aplicações das RSSFs são implementadas para durarem muito tempo, contudo os recursos disponíveis nos sensores são limitados, principalmente a quantidade de energia nas baterias. A limitação de energia torna-se um dos desafios mais importante para estas aplicações, problema agravado pela falta de um modelo de medição adequado para o monitoramento dos níveis de energia nos sensores individualmente. Assim, todas as etapas de projeto e funcionamento de RSSFs devem considerar o consumo de energia e procurar aperfeiçoar-lo. Neste trabalho foi realizada uma pesquisa bibliográfica sobre os conceitos, a arquitetura, protocolos de comunicação entre os nós sensores, funcionamento e características das aplicações, limitações e desafios das RSSFs. Foram avaliadas estratégias para gerenciamento e medição do nível de energia em baterias de sensores das RSSFs: o *Residual eScan* (método para obter a distribuição geográfica dos recursos da rede como a energia), o LEACH (protocolo de comunicação eficiente em energia para RSSF); Em seguida, foi construído um componente de *software* capaz de obter o nível de energia das baterias dos nodos. Foram utilizados o sistema operacional para sensores, TinyOS, a linguagem de programação NesC, e o *middleware* para RSSF, Myres.

Palavras-chave: RSSF, sensores, energia, NesC, TinyOS, Myres

1. INTRODUÇÃO

RSSFs são compostas por dezenas de dispositivos de baixo custo e de tamanho reduzido, capazes de realizar sensoriamento, processamento e transmissão de informação através de enlaces (comunicação) sem fio, chamados sensores [1]. Os sensores atuam de forma colaborativa como um nodo da rede, extraindo dados e os transmitindo para um ou mais pontos de saída da rede, chamados sorvedouros, para serem analisados e adicionalmente processados. Eles podem ser instalados um a um em locais pré-estabelecidos, mas na maioria dos casos são lançados de forma *ad hoc* (topologia dinâmica e aleatória) na área de monitoração “alvo”, sem obedecer a um plano de instalação pré-definido.

Nesse trabalho iremos apresentar algumas técnicas e protocolos que auxiliam na busca constante de métodos que procuram otimizar o gerenciamento de energia em baterias de RSSFs, como o eScan e o LEACH. Também iremos apresentar os passos necessários à construção de um software que realiza a medição do nível de bateria em nodos sensores de RSSFs utilizando o sistema operacional para RSSFs, TinyOS; middlewares (softwares que fornecem às aplicações funcionalidades adicionais não existentes no sistema operacional) de RSSFs e a linguagem de programação NesC.

2. AS REDES DE SENSORES SEM FIO

Na última década houve um grande avanço tecnológico nas áreas de sensores, circuitos integrados e comunicação sem fio, que levou a criação de redes de sensores sem fio (RSSF).

As redes de sensores sem fio estão sendo utilizadas nas mais diferentes áreas [2], tais como:

- Militar para detectar movimentos inimigos, explosões e a presença de material perigoso;
- Ambiental, monitorando variáveis ambientais em locais extremos como florestas, desertos, oceanos e vulcões;
- Emergência médica, para monitorar o funcionamento de órgãos como o coração, detectar a presença de substâncias que indicam o surgimento de um problema biológico, seja no corpo humano ou animal;

As redes de sensores sem fio diferem das redes de computadores tradicionais em vários aspectos, normalmente essas redes possuem um elevado número de nodos (também conhecidos como sensores) distribuídos que têm restrições de energia, possuem mecanismos para auto-configuração e adaptação por causa de problemas de comunicação e quebra dos nodos. Esses nodos são equipados com bateria, memória, processador, transceptor e sensor, como mostra a figura abaixo:

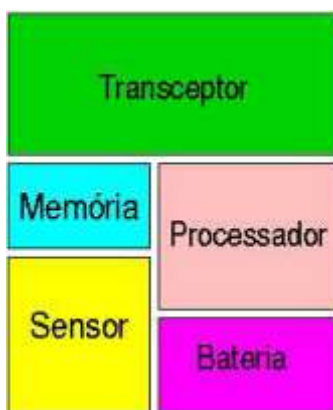


Figura 1 – Imagem com a divisão arquitetônica de um nodo sensor

Nessas redes, cada nodo é equipado com uma variedade de sensores, tais como acústico, sísmico, infravermelho, vídeo-câmera, calor, temperatura e pressão. Esses nodos podem ser organizados em grupos (*clusters*) onde pelo menos um dos sensores deve ser capaz de detectar um evento na região [1], processá-lo e tomar uma decisão se deve fazer ou não uma difusão (*broadcast*) do resultado para outros nodos. A visão é que RSSFs se tornem disponíveis em todos os lugares executando as tarefas mais diferentes possíveis. Essas redes podem ser consideradas um tipo especial de Rede Móvel *ad hoc*, onde os seus elementos computacionais trocam dados diretamente entre si através de enlaces de comunicação sem fio. As redes móveis diferem das redes de sensores sem fio apenas na forma de tratar os dados, enquanto a rede móvel têm como função básica prover suporte à comunicação entre seus elementos que podem estar executando

tarefas diferentes individualmente, as RSSFs se preocupam em executar uma tarefa colaborativa provendo dados à nodos especiais, chamados “sorvedouros”, para que esses possam processar a informação final.

As tabelas 1 e 2 apresentam características de várias tecnologias sem fio e suas aplicações. Cada uma das aplicações potenciais de diferentes serviços sem fio estão abordadas a seguir:

Tabela 1. Tecnologias sem fio e características associadas

Tecnologia	Serviços/Características	Área de Cobertura	Limitações	Exemplo de Sistemas
Celular	Voz e dados através de telefones portáteis	Contínua	Largura de banda muito baixa	Telefones celulares, <i>PAD's</i> e <i>Palm Pilots</i>
<i>WLAN</i>	<i>LAN</i> tradicional com interface sem fio	Somente ambientes locais	Alcance limitado	<i>NCR's Wave LAN</i> e <i>ALTAIR</i>
<i>GPS</i>	Determina posição tridimensional e velocidade	Qualquer lugar do mundo	Custo elevado	GNSS, NAVSTAR e GLONASS
<i>PCS</i> por satélite	Principalmente para mensagens	Quase todo o mundo	Custo elevado	Iridium e Teledesic
Redes <i>ad hoc</i>	Grupo de pessoas que compartilham dados	Similar a <i>LAN</i>	Alcance limitado	Bluetooth
Redes de sensores	Sensores pequenos sem fio	Pequena	Alcance limitado	Aplicações de defesa civil e militar

Tabela 2. Aplicações potenciais de diferentes serviços

Tecnologias sem fio	Celular	<i>WLAN</i>	<i>GPS</i>	<i>PCS</i>	Redes <i>ad hoc</i> e de sensores
Área de Aplicação	serviços no campo, segurança pública, controle de estoque, transbordoras e atividades de linhas aéreas	Lojas varejistas, serviços de saúde, tele-diagnósticos, estudantes, restaurantes, escritórios, indústria manufatureira e estoque	pesquisa, agência de aluguel de carros e esportes	<i>GPS</i> , Multimídia e Telemetria	Sensores de ambiente, Máquinas de prognóstico, detecção de pontes quebradas, condições das estradas e sensores biológicos

2.1. Características das RSSFs

As Redes de Sensores sem Fio possuem características particulares para as suas aplicações, algumas delas estão descritas abaixo:

Mobilidade dos sensores. Indica se os sensores podem se mover ou não em relação ao sistema em que estão coletando dados.

Agregação dos dados. Indica a capacidade de uma RSSF de agregar ou sumarizar dados coletados pelos sensores.

Limitação da energia disponível. Em muitas aplicações, os sensores serão colocados em áreas remotas, o que não permitirá facilmente o acesso a esses elementos para manutenção. Neste cenário, o tempo de vida e um sensor dependem da quantidade de energia disponível. Aplicações, protocolos, e algoritmos para RSSFs não podem ser escolhidos considerando apenas sua “elegância” e capacidade, mas definitivamente a quantidade e energia consumida. Assim, o projeto de qualquer solução para esse tipo de rede deve levar em consideração consumo, o modelo de energia e o mapa de energia da rede.

Tarefas colaborativas. O objetivo principal de uma RSSF é executar alguma tarefa colaborativa onde é importante detectar e estimar eventos de interesse e não apenas prover mecanismos de comunicação.

3. LIMITAÇÃO DE ENERGIA

Os recursos de processamento, armazenamento e energia disponíveis nos sensores são bastante limitados [2]. Em geral, sensores são alimentados por baterias não recarregáveis. Devido à sua enorme quantidade e à sua instalação em locais de difícil acesso, os sensores devem operar sem assistência humana por longos períodos de tempo. A constante substituição de sensores que tenham suas baterias esgotadas é indesejável e inviabiliza parte dos benefícios do uso de RSSFs, como a instalação *ad hoc* e a não perturbação do ambiente a ser monitorado. Como muitas aplicações das RSSFs são implementadas para durarem muito tempo, a limitação de energia disponível torna-se um dos desafios mais importante para esses tipos de aplicações em Redes, problema este que é agravado pela falta de um modelo de medição de energia adequado para ajudar no monitoramento dos níveis de energia nos sensores individualmente. Portanto, é importante manter os sensores ativos o maior tempo possível. O tempo de vida operacional das RSSFs é severamente limitado pela capacidade de bateria de seus nodos. Economia de energia torna-se, então, uma questão crucial nessas redes. Sendo assim, todas as etapas de projeto e funcionamento de RSSFs devem levar em conta o consumo de energia e procurar otimizá-lo.

Muitas das aplicações feitas para as RSSFs, são implementadas objetivando um longo uso da rede ao longo do tempo, chegando às vezes a durar por mais de 5 anos. Em razão disto, muitas das tarefas executadas por estes sensores ficam prejudicadas devido ao limite de energia disponível. Com uma medição mais adequada do nível de energia nos sensores individualmente, as aplicações teriam a possibilidade de melhor gerenciar seus recursos, podendo assim tomar uma decisão mais apropriada do que deve ou pode ser feito na rede, de modo a prolongar o tempo útil de bateria e por consequência de cada sensor.

3.1. Modelo de Energia

O modelo de energia de um nodo sensor pode ser visto como um provedor de energia, que depende de uma bateria com capacidade finita, e de elementos consumidores. Os consumidores de energia são os modelos de radio, processador e dispositivos de sensoriamento. Cada consumidor notifica ao provedor seu consumo de energia, que por sua vez informa a quantidade de energia disponível [1].

Sabendo que o consumo de energia em baterias de sensores de redes sem fio é hoje talvez o maior desafio desses sistemas, o conhecimento das reservas de energia disponível em cada parte da rede é uma informação importante. A forma mais natural de descobrir o nível de energia de um RSSFs é através da construção de um Mapa de Energia de cada área da RSSFs onde periodicamente cada nodo sensor envia para o nodo sorvedouro sua energia disponível. Entretanto esse envio da quantidade de energia, pelos sensores ao sorvedouro, deve ser evitado por ser um gasto extra de energia. Técnicas mais eficientes em termos de consumo de energia devem ser propostas para obter a quantidade de energia disponível em cada parte da rede ou em cada nodo da rede.

3.2. Modelo de Dissipação de Energia

Foram propostos dois modelos para representar a dissipação de energia nos nodos sensores [4]. Um chamado de *modelo de dissipação uniforme*. Durante um evento de sensoriamento, cada nodo n na rede tem uma probabilidade p de iniciar uma atividade de sensoriamento local, e cada nodo dentro de um círculo de raio r com centro em n consome uma quantidade fixa de energia e . O outro modelo é chamado *modelo de dissipação baseado em hotspot*. Neste modelo, existem h *hotspots* fixos e uniformemente distribuídos na área de sensoriamento. Cada nodo n tem uma probabilidade $p = f(d)$ de iniciar uma atividade de sensoriamento local e todo nodo dentro do círculo de raio r centrado em n consome uma quantidade fixa de energia e , onde f é uma função de densidade e d é a distância de n ao *hotspot* mais próximo. A principal desvantagem destes modelos é que eles não levam em consideração que a escassez de energia destas redes irá influenciar o seu comportamento. Por exemplo, para conservar energia, alguns nodos devem dormir por algum intervalo de tempo. Outros problemas incluem a suposição que todos os nodos que trabalham em um evento irão gastar a mesma quantidade de energia e que todos os eventos têm o mesmo raio de influência. Em [3], é proposto outro modelo de dissipação de energia para os nodos sensores. Neste modelo, os sensores possuem vários modos de operação com diferentes níveis de ativação e conseqüentemente com diferentes níveis de consumo de energia e, assim que possível, os sensores devem ir para um modo que consome menos energia. Neste modelo, cada nodo tem quatro modos de operação:

Modo 1: sensoriamento desligado e rádio desligado;

Modo 2: sensoriamento ligado e rádio desligado;

Modo 3: sensoriamento ligado e rádio recebendo ou ouvindo o canal;

Modo 4: sensoriamento ligado e rádio transmitindo.

O comportamento de cada nodo sensor pode ser descrito pelo diagrama da figura 22. No início, cada nodo vai para o modo 1 com probabilidade *sleep-prob* ou para o modo 2 com $(1-sleep-prob)$.

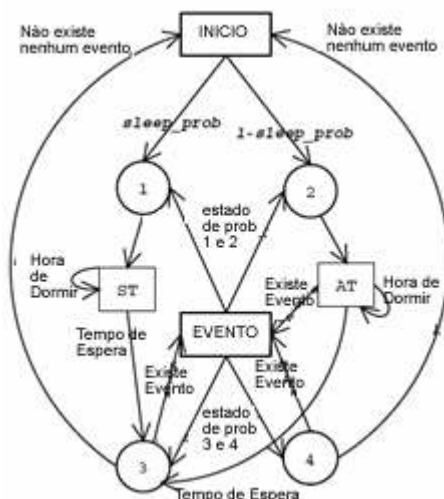


Figura 2 – Figura que mostra as fases (modos) de sensoriamento

4. PROTOCOLOS DE COMUNICAÇÃO

Os protocolos de roteamento *ad hoc* podem ser usados como protocolos para redes de sensores. Entretanto, eles geralmente não são bons candidatos para as RSSF, pois os sensores têm baixa carga de bateria e baixa disponibilidade de memória. Como o tamanho da tabela de roteamento cresce proporcionalmente com o tamanho da rede e as RSSF são projetadas para comunicação fim a fim, e ainda reagem inapropriadamente havendo movimentação, suas requisições de endereçamento podem ser inapropriadas para as RSSF e protocolos de roteamento para redes *ad hoc* não suportam disseminação cooperativa. Mais especificamente, protocolos de roteamento *multihop* suportam a criação e manutenção de caminhos para roteamento de pacotes da fonte para o destino. Como protocolos de roteamento *ad hoc* não suportam agregação ou fusão de dados, eles podem não ter bom desempenho em aplicações de redes de sensores. De uma perspectiva operacional, é interessante comparar o protocolo de roteamento *ad hoc* e a taxonomia de RSSF. Aparentemente, protocolos pró-ativos são mais apropriados para continuar o envio de dados, já que eles mantêm caminhos através da rede. A função de atualização do estado do *link* nestes protocolos pode ser vista como uma forma de continuidade de envio de dados. Os protocolos reativos parecem ser mais portáteis para disseminação de informação orientada a evento ou baseada em consulta.

A divisão da usabilidade desses protocolos pode ser em camadas como é sugerido pela arquitetura TCP/IP e mostrado na tabela abaixo.

Tabela 3. Divisão em Camadas dos Protocolos das RSSF

Camada	Protocolos
Transporte	PFSQ, ESRT, RMST
Rede	DD, SPIN, SAR, MULTI, STORM, PROC, TinyOSBeaconing, LEACH, LEACH-C, TEEN, PEGASIS, ICA, GEOMOTE, GEAR, GPSR
Enlace	S-MAC, ARC, T-MAC, B-MAC, DE-MAC, TRAMA
Física	Transmissão em Frequência de Rádio, Ótica e Infravermelho

Os protocolos de roteamento para RSSF podem ser classificados em protocolos planos, hierárquicos e geográficos. Nos protocolos de roteamento plano, todos os sensores são considerados iguais, ou seja, a atividade de roteamento é tratada de forma idêntica por todos os sensores da rede. Já no roteamento hierárquico são definidas duas classes distintas de sensores: sensores fontes e sensores líderes de grupo,

denominados cluster-head. Os sensores fonte têm a função de coletar e enviar os dados observados para o sensor líder de seu grupo. Cabe a este sensor a decisão de executar uma fusão ou agregação dos dados recebidos antes de reenviá-los. Os protocolos de roteamento geográfico utilizam informações geográficas obtidas através de um sistema de GPS (Global Positioning System) ou através de um sistema local, válido somente para os sensores da rede.

Tabela 4. Classificação dos Protocolos das RSSF

Classificação	Protocolos
Planos	DifusãoDirecionada (DD), SPIN, SAR, STORM, PROC, TinyOSBeaconing
Hierárquicos	LEACH, LEACH-C, TEEN, PEGASIS
Geográficos	GEOMOTE, GEAR, GPSR

Abaixo detalhamos os principais protocolos de comunicação usados pelas redes de sensores sem fio, devido à preocupação com o gerenciamento da energia disponível nas baterias dos sensores:

4.1. LEACH

LEACH é um protocolo eficiente em energia para RSSF's projetadas com mecanismo de envio de dados contínuo e sem mobilidade [5]. LEACH usa uma arquitetura *clustering* onde os nós membros enviam seus dados para o *cluster-head* local. *Cluster-heads* agregam dados de cada sensor e enviam esta informação para o nó observador. LEACH usa rotação do *cluster-head* para distribuir carga de energia. Uma vez que os *clusters* são formados, os membros do *cluster* usam TDMA para comunicar com o *cluster-head*. Entretanto, LEACH é portátil para redes onde todo nó tem dado para enviar em intervalos regulares. Entretanto, ele precisa ser estendido para modelos orientados a evento bem como para sensores móveis.

4.2. Directed Diffusion

Directed Diffusion (DD) é um protocolo *data-centric*, onde os nós não são endereçados por seus endereços na rede, mas sim pelos dados que monitoram [6]. Os dados são nomeados por pares de atributo-valor. No *Directed Diffusion* o interesse é expresso pelos nós observadores em termos de uma consulta que se difunde pela rede usando interações locais. Uma vez que um nó sensor que satisfaz a consulta (nó fonte) é alcançado, aquele nó começa a transmitir dados para o nó sumidouro, novamente usando interações locais. A ausência de noção de um identificador global (por exemplo, um endereço IP) torna a difusão orientada eficiente para redes com mobilidade. O protocolo *DD* é aplicável para redes orientadas a eventos e orientadas a consulta. As interações localizadas permitem ao protocolo ser escalável para redes grandes. O protocolo *DD* escala como uma função do número de interesses ativos presentes na rede.

4.3. Publish/Subscribe

O modelo *publish/subscribe* [4] foi proposto para redes móveis por Huang e Garcia-Molina. Neste modelo a comunicação é tipicamente anônima, inerentemente assíncrona e *multicast* por natureza. Do ponto de vista da aplicação, o modelo *publish/subscribe* captura o relacionamento entre o observador e o fenômeno para algumas aplicações. Neste modelo a comunicação não é fim a fim, mas anônima com formação de grupo *multicast* específico da aplicação. Em relação à implementação, a comunicação assíncrona ajuda a preservar energia e aumentar a vida útil da rede. Ratnasamy apresenta uma classificação alternativa de redes de sensores baseada no modelo de disseminação de dados. Ele propõe que a disseminação de dados possa ser feita no mínimo de três formas: armazenamento externo, onde passa todos os dados para o observador e o deixa processar esta informação; armazenamento local, onde a informação sobre o evento é armazenada localmente pelos sensores e armazenamento orientado a dados, onde os dados são armazenados pelo nome e consultas são direcionadas pelo nome para o sensor correspondente. A escolha do modelo influencia os padrões de comunicação dentro da rede.

4.4. Protocolo TEEN

O TEEN (Threshold Sensitive EnergyEfficient sensorNetwork) é um algoritmo de roteamento hierárquico similar ao LEACH exceto pelo fato de que os nós sensores podem não possuir dados a serem transmitidos de tempos em tempos. Os autores deste protocolo propõem classificar as redes de sensores em redes pró-ativas e redes reativas. Uma rede pró-ativa monitora o ambiente continuamente e possui dados a serem

enviados a uma taxa constante. Em uma rede reativa os nós somente enviam dados quando a variável sendo monitorada se incrementa acima de certo limite. TEEN utiliza a estratégia de formação de líder do LEACH, mas adota uma estratégia diferente na fase de transmissão de dados. Ele faz o uso de dois parâmetros chamados Hard Threshold (Ht) e Soft Threshold (St) para determinar a necessidade de transmissão do dado coletado. Se o valor exceder (Ht) pela primeira vez, ele é armazenado em uma variável e transmitido durante o intervalo (slot) de tempo alocado a transmissão do nó. Em seguida, se o valor monitorado exceder o valor armazenado por uma magnitude de St o nó transmite o dado imediatamente. O valor enviado é armazenado para comparações futuras.

4.5. Protocolo PEGASIS

O PEGASIS (Power-Efficient Gathering in Sensor Information Systems) é um protocolo para RSSF baseado no conceito de correntes. Cada nó troca informações apenas com os vizinhos mais próximos formando uma corrente entre os nós, e apenas um nó é escolhido a cada momento para transferir as informações coletadas ao nó gateway (ver figura 8). Portanto, o número de trocas de mensagens será baixo e a comunicação será realizada entre nós próximos uns dos outros. Espera-se com isso que a energia gasta seja menor, se comparada a outros protocolos que requerem muitas trocas de mensagens para eleger líderes e formar grupos, e protocolos em que os nós constantemente trocam mensagens com o nó gateway de forma direta (o gateway geralmente se encontra distante dos nós). Isto implica um tempo de vida maior para cada nó e um consumo menor da largura de banda da rede. O PEGASIS assume o seguinte:

- O nó gateway (estação base) situa-se estacionado à uma distância fixa da rede;
- Os nós são capazes de transmitir dados diretamente para o nó gateway e para qualquer outro nó;
- Cada nó possui informação de localização dos outros nós;
- Os nós são homogêneos e com o nível de energia uniforme;
- Os nós não são móveis. A cada round um nó é escolhido para transmitir a informação a estação base;

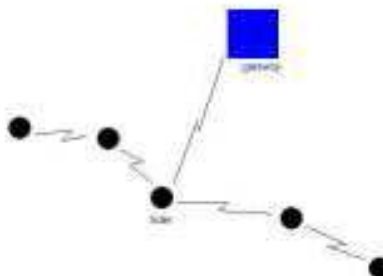


Figura 3 – Figura que mostra o funcionamento do Pegasis

4.6. Protocolos da Camada de Enlace

Os requisitos da camada de enlace são diferentes para os diferentes tipos de RSSF apresentados. Por exemplo, os nós de uma RSSF sob demanda podem permanecer com os transceptores inativos por longos períodos de tempo e que repentinamente tornam-se ativos quando algum fenômeno é detectado. Se a RSSF for densa, vários nós sensores na área de ocorrência do evento estarão acessando o meio ao mesmo tempo para transmitir os dados. As características particulares das RSSFs e sua dependência da aplicação motivam um controle de acesso ao meio (MAC - Medium Access Control) que é diferente do tradicional tal como o IEEE 802.11 [IEEE802.11, 2003]. Quase sempre a conservação de energia e a auto-organização são objetivos primários.

Abaixo é apresentada uma comparação entre os principais protocolos de acesso ao meio projetados para RSSFs:

Protocolo	Características	Tipo de RSSF	Vantagem	Desvantagem
S-MAC	Ciclo de operação reduzido.	Dirigidas a eventos, com coleta periódica, insensível a latência da rede e com baixa	Prolonga consideravelmente o tempo de vida da rede.	Aumento da latência devido ao período sleep.

		taxa de envio de mensagens.		
T-MAC	Ciclo de operação reduzido com incremento de um temporizado.	Dirigidas a eventos, com coleta periódica, insensível a latência da rede e com baixa taxa de envio de mensagens.	Consegue um consumo mais eficiente que o S-MAC, devido ao incremento do temporizado.	É extremamente limitado com relação a largura de banda da rede.
ARC	Evita colisões adicionando um atraso aleatório antes do tempo de escuta.	Coleta periódica de dados, sensível a latência com baixa ou alta taxa de envio de mensagens.	Provê uma distribuição justa e mantém uma razoável largura de banda.	Não é eficiente para redes com um tráfego alto.
DE-MAC	Ciclo de operação reduzido utiliza um algoritmo distribuído para balanceamento da carga na rede.	Coleta periódica, insensível a latência da rede e com baixa taxa de envio de mensagens.	O balanceamento de carga serve para obter um tempo de vida uniforme para todos os nodos da RSSF.	O algoritmo de balanceamento pode comprometer no caso de uma rede dirigida a eventos.
TRAMA	Utiliza um algoritmo de eleição distribuído para evitar colisões em comunicações unicast, broadcast ou multicast, alterna acessos entre aleatórios e escalonados.	Insensíveis à latência, com alta taxa de entrega.	Acomoda mudanças de topologia da rede, adição de nodos a rede e tolerância a falhas.	Devido seu algoritmo de eleição há uma grande troca de informações na rede, desfavorecendo a economia de energia.

5. RESIDUAL ENERGY SCAN

O Residual Energy Scan é um método de obter a distribuição geográfica dos recursos da rede ou das atividades dos sensores, como a energia. Ele provê uma visão abstrata da distribuição de energia ao invés da informação detalhada da energia residual de cada nó.

Ele agrega dados, reduzindo a quantidade de dados transmitidos, mas introduz uma distorção na medida. Existe um *tradeoff* entre a quantidade de dados transmitidos e a sua precisão. O problema com o **Residual Energy Scan** é o fato do roteamento tender a sobrecarregar os nós pais. O roteamento é feito de maneira que todas as mensagens do nó são enviadas para outro nó, chamados de nó pai, no qual está mais próximo do ponto de acesso, e assim por diante. Os nós pais tendem a receber e transmitir mais mensagens do que seus descendentes, levando a um consumo de energia não balanceado.

Portanto, os nós que estão próximos dos pontos de acesso irão receber e transmitir mais mensagens do que os nós mais distantes. Isto pode causar uma morte prematura da rede, pois mesmo que existam nós com energia suficiente para operar, suas mensagens não chegarão ao ponto de acesso. Outro problema com o **Residual Energy Scan** é que ele não é tolerante a falha. Suponha que aconteça uma disfunção no nó sensor durante uma pesquisa. Os dados enviados pelos nós descendentes não estarão disponíveis para o ponto de acesso e esses nós sensores ficarão órfãos. Em RSSF, uma falha é um caso comum, não uma exceção.

6. CYGWIN

O Cygwin é uma coleção de ferramentas de software livre originalmente desenvolvidas pela Cygnus Solutions de maneira a permitir que várias versões do Microsoft Windows possam agir como um sistema Unix. Essa ferramenta foi utilizada no princípio da pesquisa quando ainda pensávamos em utilizar o Windows como Sistema Operacional. Como tivemos vários problemas de configuração ao tentar executar o TinyOS, migramos para o Linux de distribuição Debian.

7. TINYOS E TOSSIM

TinyOS[7] é um sistema operacinal *open-source*, projetado para sensores das RSSF. Ele caracteriza-se por ter uma arquitetura baseada em componentes, o que permite rápida inovação e implementação no tamanho dos códigos, já que restrição de memória é uma característica inerente ao sensor. A biblioteca de componentes do TinyOS inclui protocolos de rede, serviços distribuídos, drivers de sensores e ferramentas de aquisição de dados, e todos estes componentes podem ser utilizados para customizar uma determinada aplicação.

O Tossim[8] é um simulador de um sensor de uma RSSF, o que permite o desenvolvimento de aplicações mesmo sem fazer uso de um sensor físico real. O tossim simula a rede TinyOS a nível binário e permite ainda a experiência de se trabalhar com protocolos de baixo nível adicionalmente a sistemas de alto nível.

A instalação do TinyOS consistiu nos seguintes passos:

- Fazer o Download dos arquivos e pacotes necessários à instalação como os compiladores nativos e as ferramentas do TinyOS.
- Instalação e Configuração da árvore de arquivos do TinyOS.
- Configuração das variáveis ambientes utilizando o script abaixo:
- Instalação do Graphviz, um visualizador gráfico do Linux.

8. NESC

A linguagem de programação nesC[9] é uma extensão da linguagem de programação C que foi projetada para implementar os conceitos de estruturas e execução do modelo do TinyOS. As características da linguagem nesC são a separação entre a construção e a composição; especificação do comportamento dos componentes setado nas *interfaces*; relacionamentos bidirecionais entre os *commands* (provedores da interface), e os *events* (usuários das interfaces); os componentes são *linkados* estaticamente através de suas relações; e é projetado sob a expectativa que o código será gerado pela compilação do programa inteiro.

9. SOFTWARE

Foi desenvolvido uma software que realiza a medição do nível de energia num sensor de uma RSSF individualmente. O desenvolvimento foi feito utilizando-se da linguagem NesC, sobre o ambiente TinyOS e através do simulador TOSSIM, visto que não dispúnhamos de um dispositivo físico real.

10. CONCLUSÕES

Dentre os resultados que obtivemos ao longo da pesquisa destacamos os relatórios que geramos relatando as experiências adquiridas e as dificuldades enfrentadas. Visto que a área de pesquisa é uma área relativamente nova, as fontes de informações são poucas e algumas delas contêm informações ambíguas ou totalmente falhas. Devido a isto, tivemos problemas com a instalação e configuração do ambiente de trabalho.

Também foi desenvolvido um protótipo de software que lê o nível de energia em uma bateria de um sensor de rede sem fio.

11. REFERÊNCIAS

- [1] PEREIRA, M. R.; AMORIM, C. L.; CASTRO, M. C. S. “Tutorial sobre Redes de Sensores”, Programa de Engenharia de Sistemas e Computação (COPPE/UFRJ), Rio de Janeiro.
- [2] LOUREIRO, A. A. F.; NOGUEIRA, J. M. S.; RUIZ, L. B.; MINI, R. A. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S. “Redes de Sensores Sem Fio”, Departamento de Ciência da Computação (UFMG), Belo Horizonte.
- [3] MINI, R. A. F.; NATH, B.; LOUREIRO, A. A. F. Prediction-based approaches to construct the energy map for wireless sensor networks. *In* 21º Simpósio Brasileiro de Redes de Computadores, Natal, RN, Brasil, Maio 2003.

- [4] ZHAO, Y. J.; GOVINDAN, R.; ESTRIN, D. Residual energy scans for monitoring wireless sensor networks. *In* IEEE Wireless Communications and Networking Conference (WCNC'02), Orlando, FL, USA, March 2002.
- [5] HEINZELMAN, W.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-Efficient Communication Protocols for Wireless Microsensor Networks, Proc. Hawaaiian Int'l Conf. on Systems Science, January 2000.
- [6] ESTRIN, D.; HEIDEMANN, J.; GOVIDAN, R. Next Century Challenges: Scalable Coordination in Sensor Networks. *In* Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99), August 1999, Seattle, Washington.
- [7] HILL, J.; SZEWCZYK, R.; WOO, A.; HOLLAR, S.; CULLER, D.; PISTER, K. "System Architecture Directions for Networked Sensors". *In* Proceedings of Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), November 2000.
- [8] LEVIS, P.; LEE, N.; WELSH, M.; CULLER, D. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. *In* Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003).
- [9] GAY, D.; LEVIS, P.; BEHREN, R. V.; WELSH, M.; BREWER, E.; CULLER, D. The nesC Language: A Holistic Approach to Networked Embedded Systems, *In* Proceedings of Programming Language Design and Implementation (PLDI) 2003, June 2003.

12. AGRADECIMENTOS

Agradecemos em primeiro lugar a Deus por ter nos dado força e saúde para a realização desta pesquisa, apesar das muitas dificuldades encontradas ao longo da caminhada. Também agradecemos ao doutorando do Centro de Informática da UFPE Klaus Cavalcante que nos ajudou bastante na realização deste trabalho.