CARLOS MAURÍCIO SERÓDIO FIGUEIREDO

AUTO-ORGANIZAÇÃO EM REDES DE SENSORES SEM FIO

Belo Horizonte
Fevereiro de 2007

CARLOS MAURÍCIO SERÓDIO FIGUEIREDO

Orientador: Antonio Alfredo Ferreira Loureiro

AUTO-ORGANIZAÇÃO EM REDES DE SENSORES SEM FIO

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Belo Horizonte
Fevereiro de 2007

CARLOS MAURÍCIO SERÓDIO FIGUEIREDO

Advisor: Antonio Alfredo Ferreira Loureiro

SELF-ORGANIZATION FOR WIRELESS SENSOR NETWORKS

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Belo Horizonte

February 2007

UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Auto-Organização em Redes de Sensores sem Fio

CARLOS MAURÍCIO SERÓDIO FIGUEIREDO

Tese defendida e aprovada pela banca examinadora constituída por:

Prof. Antonio Alfredo Ferreira Loureiro – Orientador
Departamento de Ciência da Computação – ICEx – UFMG

Profa. Linnyer Beatryz Ruiz – Co-orientador
Departamento de Engenharia Elétrica – DEE – UFMG

Prof. Djamel Fawzi Hadj Sadok
Centro de Informática da Universidade Federal de Pernambuco – CI – UFPE

Prof. Claudio Luis de Amorim
Universidade Federal do Rio de Janeiro – COPPE – UFRJ

Prof. Geraldo Robson Mateus
Departamento de Ciência da Computação – ICEx – UFMG

Prof. José Marcos Silva Nogueira
Departamento de Ciência da Computação – ICEx – UFMG

Belo Horizonte, Fevereiro de 2007

Agradecimentos

Agradeço à minha esposa e filha pelo apoio e compreensão incondicionais em momentos desafiadores. Aos meus pais pelos valores transmitidos e incentivo à formação. Agredeço ao prof. Loureiro pelo exemplo e pelo compartilhamento de conhecimentos e experiências. À FUCAPI e seus responsáveis pelo apoio e consciência de que devemos investir em educação para transformarmos nossas instituições.

Resumo

Auto-organização é um importante conceito para o desenvolvimento de sistemas de rede autônomos e de larga escala. Sua idéia principal é "a obtenção de um comportamento global a partir de interações locais entre os elementos do sistema", e isso leva a redes menos dependentes de controle centralizado, e que tendem a ser escaláveis, adaptáveis e, consequentemente, robustas.

Em particular, uma área de pesquisa bastante ativa é a de Redes de Sensores sem Fio (Wireless Sensor Networks – WSNs). WSNs possuem um grande potencial de serem empregadas em várias aplicações relevantes, e foram concebidas sob o paradigma da auto-organização devido a muitas características intrínsecas. Essas redes são formadas por uma grande quantidade de dispositivos sensores interconectados por um canal sem fio com a finalidade de realizar tarefas de sensoriamento de forma distribuída e cooperativa. Além da escala, que aumenta a complexidade de desenvolvimento e manutenção, essas redes tendem a ser muito dinâmicas, pois alterações topológicas podem ser freqüentes devido à destruição de elementos, ao esgotamento de energia, à inclusão de novos sensores, ou à intermitência de comunicação causada por interferências ou obstáculos. Ainda, muitas aplicações dessas redes requerem um funcionamento autônomo, pois podem ser aplicadas em locais remotos, inóspitos e de difícil acesso.

Embora muitos trabalhos na literatura abordem aspectos de auto-organização em WSNs, às vezes de forma implícita, não é provida uma visão mais geral e prática que possa ser usada para novos projetos e desenvolvimentos. Além disso, o projeto de funções auto-organizáveis pode não ser trivial, pois não há métodos formais ou modelos para o mapeamento dos comportamentos globais desejados para regras de interações locais. Assim, esta tese avança em dois aspectos particulares da aplicação do conceito de auto-organização a essas redes. No primeiro, um guia de projeto (guideline) é desenvolvido para nortear a modelagem e desenvolvimento de novas funções auto-organizáveis nessas redes, e sua vantagem consiste em um melhor entendimento e a especificação de funções auto-organizáveis através de aspectos e mecanismos gerais. No segundo, um esquema de gerenciamento dessas redes é proposto de forma a considerar um aspecto prático da operação de WSNs de que diferentes requisitos ou obje-

tivos globais podem ser requeridos por uma aplicação externa ou entidade de gerência. Assim, o esquema consiste em manter o funcionamento da rede de forma auto-organizável em um nível operacional, mas permitir a ação sobre as regras de interação locais por entidades centralizadas de mais alto nível que, por sua vez, têm a visão mais geral das necessidades externas e do desempenho da rede para o atendimento a esses diferentes objetivos.

Com base nesses aspectos gerais propostos, este trabalho ainda apresenta duas contribuições individuais abordando funções específicas de WSNs. Na primeira, a objetividade necessária às WSNs é mostrada através do desenvolvimento de uma solução de roteamento. Em particular, duas propostas de roteamento pró-ativo e reativo auto-organizáveis são combinadas em uma solução adaptativa híbrida, chamada Multi, que estende a aplicação de auto-organização em uma abordagem prática. Na segunda, a necessidade de considerar diferentes funções auto-organizáveis em um projeto integrado é apresentada com em uma solução, chamada RDC, que considera as funções de roteamento e controle de densidade.

Abstract

Self-organization is an important concept that has been applied to large-scale and autonomous network systems. Its main idea is the “achievement of a global behavior through local interactions”, which leads to networks that are less dependent on a central control, and that tend to be scalable, adaptable and, consequently, robust.

In particular, Wireless Sensor Networks (WSNs) form a very active research area. WSNs are expected to have a wide applicability, and they were conceived under the self-organization paradigm due to several intrinsic characteristics. These networks consist of sensor a high number of sensing nodes interconnected by a wireless channel to perform distributed and cooperative sensing tasks. In addition to the scale, which increases the maintenance complexity, these networks tend to be very dynamic, because topological changes may be frequent due to destruction or energy depletion of nodes, inclusion of new ones, or due to the intermittent wireless communication caused by interferences or obstacles. Also, several WSN applications require the adoption of a totally autonomous behavior due to the necessity of sensing in places that are remote, inhospitable or with difficult access.

Although several proposals in literature relate self-organization aspects in WSNs, sometimes implicitly, they do not provide any general and practical vision that can be used by new designs and developments. Additionally, the design process of self-organizing functions may not be trivial, because there are not formal methods of mapping desired global goals to local interaction rules. Thus, this thesis advances with two particular aspects of the concept application in the WSN domain. First, a design guideline is proposed to direct the design and the development of new self-organizing functions. Its advantage consists of a better understanding and the specification of self-organizing functions through general aspects and mechanisms. Second, a management scheme is presented, and it considers practical aspects of WSNs such that different requirements and global goals may be needed by an external application or management entity. Thus, the scheme allows the network operation in a self-organizing way in a lower level, and it also allows the action over local rules by centralized entities in a higher level, which has a wider network vision about the external needs and the

network performance according to different goals.

Based on the proposed general aspects, this work also gives two individual contributions regarding specific WSN functions. First, the objectivity needed in WSNs is shown through the development of a routing solution. In particular, a pro-active and a reactive self-organizing routing proposal are combined in a hybrid adaptive solution, called Multi, which extends the self-organization application in a practical approach. Second, the need of considering different self-organizing functions in an integrated design is presented in a solution, called RDC-Integrated, which considers the routing and density control functions.

Resumo Estendido

Com o intuito de facilitar o acesso a esta tese, originalmente escrita na língua inglesa, e seguindo as normas do programa de pós-graduação do DCC/UFMG, apresentamos nas seções seguintes um resumo estendido descrevendo os principais problemas identificados e soluções desenvolvidas.

Capítulo 1 - Introdução

Os sistemas computacionais atuais vêm, cada vez mais, sendo compostos por um grande número de componentes interconectados. Dessa forma, atividades de projeto e administração de tais sistemas estão se tornando muito complexas, principalmente quando realizadas manualmente por engenheiros e operadores. Nesse contexto, muita pesquisa tem surgido para o desenvolvimento de soluções autônomas, onde pretende-se criar sistemas computacionais com funcionalidades realizadas de forma automática e com o mínimo de intervenção humana.

Um importante conceito aplicado às redes dinâmicas e de larga escala é o conceito de auto-organização. Sua idéia é a criação de um comportamento global a partir de interações locais entre os elementos do sistema. Com essa característica, pode-se obter redes menos dependentes de um controle centralizado porque a inteligência do sistema é totalmente distribuída a todos os elementos do sistema. Assim, auto-organização torna-se um paradigma poderoso para a obtenção de sistemas adaptáveis e simples.

Particularmente, uma área de pesquisa bastante ativa e que demanda um funcionamento autônomo é a de Redes de Sensores sem Fio (RSSF). Tais redes são formadas por grandes quantidades de elementos sensores autônomos, normalmente empregados em cenários dinâmicos, e que se comunicam para a realização cooperativa de atividades de monitoramento. De fato, tais redes foram concebidas sob o paradigma da auto-organização. No entanto, as soluções existentes são muito particulares e não cobrem um visão mais geral e prática do conceito. Tal fato motivou o desenvolvimento deste trabalho.

O objetivo desta tese é estender a aplicação do conceito de auto-organização às RSSFs. Isso é feito com a proposta de modelos gerais de projeto que possam ser usados para um melhor entendimento do conceito da auto-organização e que possam guiar o projeto de novas funções auto-organizáveis. Adicionalmente, vários dos aspectos contidos nesse modelo geral são usados no desenvolvimento de soluções mais específicas, fornecendo contribuições mais direcionadas desta tese. Basicamente, as contribuições deste trabalho são:

- Uma visão geral sobre auto-organização e RSSFs – São apresentados conceitos e trabalhos relacionados relativos a sistemas auto-organizáveis diversos e, mais particularmente, às RSSFs. O objetivo principal dessa contribuição é a discussão da aplicação do conceito às RSSFs.

- Uma metodologia de projeto para RSSFs auto-organizáveis – Um arcabouço conceitual é proposto contendo importantes aspectos para o projeto de RSSFs auto-organizáveis. Particularmente, na literatura há a ausência de trabalhos abordando tais aspectos, e com o arcabouço proposto pode-se facilitar a atividade de projeto.

- Um esquema para o gerenciamento de RSSFs auto-organizáveis – Nessa contribuição, um modelo de gerenciamento é proposto de forma a colocar o aspecto totalmente descentralizado da auto-organização sob o contexto de entidades de gerência externas, autônomas ou não. Tal visão se faz necessária devido à característica "bottom-up" do conceito de auto-organização, mas que o atendimento aos requistos de aplicações e administradores se dá de forma "top-down".

- Roteamento adaptativo híbrido de RSSFs – Essa contribuição fornece uma solução de roteamento para RSSFs orientada a eventos composta de duas estratégias de criação de infra-estrutura auto-organizáveis, uma reativa e outra pró-ativa. Seguindo o modelo do esquema de gerenciamento, uma regra de adaptação autônoma é empregada no sink de forma a perceber o comportamento da rede e alterar a forma de criação da infra-estrutura com o objetivo de economizar energia.

- Uma solução integrada de roteamento e controle de densidade – Nessa contribuição, as importantes funções de roteamento e controle de densidade são integradas em uma única solução auto-organizável. Tal desenvolvimento mostra a necessidade do aspecto de integração discutido na metodologia de projeto auto-organizável para a obtenção de uma rede que funcione corretamente e de forma eficiente.

Capítulo 2 - Auto-organização: Uma Visão Geral

Auto-organização é um conceito bem conhecido na literatura e vem sendo estudado em diferentes áreas tais como Física, Química e Biologia. Sua definição diz respeito à formação de um comportamento global coerente a partir de interações locais entre os elementos do sistema, e esse aspecto de controle descentralizado leva à obtenção de sistemas mais adaptáveis, escaláveis e robustos.

Este capítulo, além de discutir a aplicação do conceito, apresenta vários exemplos de sistemas naturais auto-organizáveis. Particularmente, as características observadas nesses sistemas levaram à aplicação do conceito em diversos sistemas computacionais que vêm se tornando cada vez mais distribuídos e compostos por muitos elementos. Neste capítulo também são descritos exemplos de auto-organização no contexto de sistemas multi-agente, robótica e redes de computadores em diversas escalas, tais como as próprias RSSFs ou a Internet.

Capítulo 3 - Redes de Sensores sem Fio Auto-organizáveis

Uma rede de sensores sem fio (RSSF) é uma rede ad hoc formada por um grande número de nós sensores distribuídos em uma área de interesse com objetivo de monitoramento. Redes de sensores sem fio têm despertado o interesse da comunidade científica devido a sua aplicabilidade que abrange diversas áreas, tais como militar, ambiental, médica e industrial.

Redes de sensores diferem de redes tradicionais em vários aspectos. Algumas dessas particularidades são: Essas redes são formadas por um grande número de nós sensores; sensores possuem fortes limitações de energia, capacidade de processamento e memória; e muitas aplicações de RSSFs demandam características de auto-organização, ou seja, capacidade de se ajustar autonomamente às possíveis mudanças estruturais devido a intervenções externas, tais como mudanças topológicas (causadas por falhas, mobilidade ou inclusão de nós), reação a um evento de sensoriamento ou a uma solicitação feita por uma entidade externa (usuário ou sistema fixo).

Auto-organização vem sendo considerada nessas redes desde o surgimento dos primeiros trabalhos. Sob um ponto de vista mais prático, algumas funções fundamentais de RSSFs desenvolvidas sob o conceito de auto-organização são descritas, como por exemplo: funções básicas de comunicação (ex. roteamento e acesso ao meio), controle de densidade, agrupamento (*clustering*), localização, sincronização e segurança.

Capítulo 4 - Uma Metodologia de Projeto para RSSFs Auto-organizáveis

Após a discussão inicial apresentada neste trabalho, podemos ver que auto-organização tem um importante papel no desenvolvimento de sistemas computacionais autônomos e, em particular, redes de sensores sem fio. Enquanto alguns mecanismos básicos tratando de regras de interação locais para a obtenção de auto-organização são conhecidos, ainda há a necessidade de metodologias e/ou esquemas para auxiliar o projeto de tais sistemas.

Para o projeto de RSSFs auto-organizáveis, regras de interação locais precisam ser estabelecidas para que se atinja um comportamento global para toda a rede. Na prática, essa tradução “top-down” não é fácil devido à falta de métodos formais, uma linha de pesquisa ainda em aberto. No entanto, alguns aspectos gerais de projeto e idéias podem ser úteis a esse objetivo e esta seção apresenta um passo à frente em relação ao projeto de RSSFs auto-organizáveis.

Neste capítulo, é apresentada uma metodologia composta por:

- Aspectos gerais – Coloca a necessidade de se estabelecer regras de interações locais simples, codificadas em algoritmos distribuídos, para a obtenção de um comportamento global auto-organizável. Através de uma análise de trabalhos e soluções existentes na literatura, identificamos algumas características comuns desses algoritmos como o controle descentralizado, necessidade de estabelecer um ojetivo comum, necessidade de se ter um sistema dinâmico e se preservar os recursos escassos das RSSFs.

- Um arcabouço de projeto – Reúne características práticas comuns para o desenvolvimento de RSSFs auto-organizáveis. Basicamente, o modelo é composto por: (i) um cerne composto por informações locais, necessárias para a tomada de decisão distribuída, e mecanismos de interação locais, necessários à troca de informações entre indivíduos; e (ii) um nível de abstração de serviços construídos a partir do cerne, onde identificou-se os importantes serviços de informação de vizinhança e atribuição de papéis para facilitar o desenvolvimento de novas funções auto-organizáveis.

- Um guia de projeto – Propõe e descreve uma sequência de etapas para a busca de soluções auto-organizáveis de forma mais objetiva. De forma resumida, são descritas as seguintes etapas de projeto: (i) Especificação dos objetivos globais, para a especificação do comportamento do sistema; (ii) Mapeamento para comportamentos locais, para a definição das regras de interação e informações loais para a tomada de decisões; (iii) Integração de funções, para orientar o projeto conjunto de diferentes funções tornando-as mais eficientes; (iv) Implementação e avaliação, indicando a necessidade de simulação e experimentação para observação do comportamento do sistema; e (v) Revisão do projeto, indicando que pode ser necessário passar pelas etapas anteriores até se obter um comportamento global desejado a partir das regras locais.

Com o intuito de mostrar a aplicabilidade da metodologia proposta, um estudo de caso foi desenvolvido seguindo os aspectos identificados e descritos. O estudo de caso tratou do projeto de uma solução auto-organizável, criada a partir de um modelo de autômatos celulares, que cria padrões espaciais para ativação de nós de forma a permitir um sensoriamento amostral.

Diante de tais aspectos conceituais e do estudo de caso desenvolvido, as vantagens esperadas do uso da metodologia proposta são: (i) ela estende a discussão sobre a aplicação do conceito de auto-organização nas RSSFs focando no processo de projeto. Essa discussão também tende a favorecer um melhor entendimento do conceito; (ii) ela aponta para aspectos práticos importantes da implementação do conceito e isso pode ser muito útil ao servir de base para novos projetos; (iii) a metodologia assiste os projetistas no desenvolvimento de funções auto-organizáveis para RSSFs através da definição de importantes etapas de projeto, o que torna o processo de projeto mais dirigido.

Capítulo 5 - Um Esquema de Gerenciamento para RSSFs Auto-organizáveis

De forma prática, uma RSSF serve a um propósito, ela possui clientes (ex. um usuário, aplicação ou administrador) e mudanças dos objetivos globais, requisitos de QoS ou percepções globais do desempenho da rede (condições gerais não necessariamente relacionadas à percepção de cada indivíduo) por parte desses clientes podem demandar ações de gerenciamento de entidades externas, caracterizando uma abordagem “top-down”. No entanto, seguindo o paradigma da auto-organização, funções de RSSFs são executadas de forma descentralizada apenas através de interações locais entre os elementos, em uma abordagem “bottom-up”, e nenhum desses elementos está a par das necessidades externas ou da condição global da rede para executar tais mudanças de objetivos. Dessa forma, uma abordagem conjunta considerando características tradicionalmente centralizadas de gerenciamento e o comportamento descentralizado da auto-organização se faz necessária.

Adicionalmente, abordagens tradicionais de gerenciamento de redes são baseadas no controle centralizado sobre os indivíduos do sistema por entidades de gerência. Essa abordagem pode se tornar inviável para sistemas de larga escala e muito dinâmicos, principalmente por parte de administradores, porque tal controle centralizado aumenta a complexidade de processamento da entidade central e de comunicação na sua interação com todos os elementos da rede. E mesmo embora o conceito de Computação Autonômica tenha surgido e avanços relativos a ele tenham sido introduzidos no auto-gerenciamento de redes ad hoc e de sensores, essas soluções mantêm algumas funções auto-organizáveis e ainda aplicam outras através de gerentes autonômicos centralizados. Além disso, formas de ajustar as funções auto-organizáveis a diferentes objetivos e requisitos não são explicitadas.

Diante desse contexto, é apresentado um esquema de gerenciamento cujo modelo geral propõe uma visão intermediária em que as funções operacionais básicas dessas redes são realizadas de forma auto-organizável em um nível mais baixo. Nessa visão, entidades de gerenciamento centralizadas, que podem ser internas ou externas à rede e autonômicas ou não, controlam o comportamento executado por todos os indivíduos de forma a satisfazer diferentes objetivos globais.

De forma complementar, alguns aspectos práticos para a implementação da visão proposta também são relacionados. Esses aspectos são descritos nas seguintes etapas: (i) Mapeamento de Objetivos globais e regras locais; (ii) Definição de opções de mudança; (iii) Definição das entidades de controle; e (iv) Definição de políticas de gerenciamento.

Para demonstrar a aplicabilidade do esquema proposto, alguns estudos de caso são apresentados. Basicamente, são consideradas duas situações onde uma entidade de gerência pode atuar sobre o comportamento da rede:

- Atendimento a Diferentes Objetivos Globais – Considera a mudança de um objetivo externo por parte dos usuários da rede. Este estudo de caso considera funções auto-organizáveis para infra-estrutura de comunicação e controle de densidade. Nelas, basicamente, o processo de decisão de atividade dos nós é inserido no processo de construção de árvore de roteamento, que é iniciado pelo sorvedouro. A essa função de auto-organização pode-se associar como objetivos globais uma maior redundância de dados coletados, onde as regras de controle de densidade podem ser ajustadas ou até desabilitadas para manter mais nós em atividade, ou maior economia de energia, onde as regras são ajustadas para manter mais nós inativos. Como resultado, mostrou-se a mudança do comportamento global da rede e, consequentemente, das métricas de funcionamento desejadas, perante diferentes objetivos estabelecidos de forma centralizada.

- Auto-Adaptação da Função Auto-organizável – Considera a mudança de uma percepção global da rede que pode demandar um comportamento auto-organizável diferente. Este estudo de caso apresenta diferentes estratégias para a criação de infra-estrutura de roteamento de forma auto-organizável, sendo que uma regra adaptativa aplicada por uma entidade de gerenciamento autonômica muda a estratégia de roteamento adotada conforme sua percepção global da rede, causando um comportamento híbrido dos mecanismos citados. Como resultado, mostrou-se a viabilidade de uma entidade de gerência autônoma atuar globalmente na rede sem a necessidade de monitorar e atuar sobre cada elemento individualmente.

As vantagens do esquema proposto são: (i) ele permite a aplicação de funções auto-organizáveis em um nível mais baixo para um apropriado funcionamento da rede com todas as vantagens particulares dessa abordagem, ou seja, levando em consideração questões locais para seu funcionamento; (ii) o esquema permite a mudança do comportamento do sistema quando necessário considerando os objetivos e percepções globais da rede, não sendo voltado ao monitoramento e controle de indivíduos; (iii) o esquema explicita aspectos importantes de projeto de soluções de gerenciamento para RSSFs que podem guiar novos desenvolvimentos. (iv) ele apresenta uma visão complementar aos trabalhos da literatura para a aplicação de funções auto-organizáveis e seu gerenciamento e, assim, também avança no relacionamento entre os conceitos de auto-organização e auto-gerenciamento.

Capítulo 6 - Multi: Um Algoritmo Adaptativo Híbrido para Roteamento em RSSFs

O principal objetivo de uma RSSF é coletar e processar dados do ambiente onde ela se encontra e enviá-los para uma entidade externa, normalmente a partir de um nó sorvedouro (sink), para adequado processamento e armazenamento. Consequentemente, roteamento é uma função fundamental dessas redes e tem recebido muita atenção da comunidade científica.

Um consenso existente é que diferentes aplicações e cenários demandam diferentes características das RSSFs. Dessa forma, dado um determinado cenário, uma RSSF pode ser projetada para operar a solução de roteamento mais adequada, que pode ser definido a priori. No entanto, em alguns casos, variações nesses cenários podem ocorrer, inclusive de forma imprevisível. Tal situação pode fazer com que seja inviável a atuação de uma entidade de controle externa da rede de forma a agir dinamicamente sobre ela para mudar seu comportamento. Um exemplo claro, e que é explorado na solução apresentada neste capítulo, é o de um cenário orientado a eventos. Nesse caso, a rede pode permanecer longos intervalos de tempo com pouca ou mesmo sem a ocorrência de eventos, favorecendo estratégias de roteamento reativas. Mas em determinado instante, essa situação pode mudar e gerar constante tráfego na rede, favorecendo estratégias pró-ativas.

Diante desse cenário, torna-se interessante o projeto de soluções auto-adaptativas em que uma entidade monitora o comportamento da rede e atua sobre a função auto-organizável de roteamento de uma RSSF para levá-la ao funcionamento mais adequado. Tal solução foi introduzida como estudo de caso do esquema de gerenciamento proposto anteriormente, mas é melhor detalhada neste capítulo como uma contribuição individual desta tese.

Basicamente, a solução projetada, chamada Multi, consiste de um algoritmo de roteamento auto-organizável pró-ativo, e um reativo, com uma regra de adaptação aplicada de forma autônoma. Tais componentes são descritos a seguir:

- Algoritmo Pró-ativo (EF-Tree) – Algoritmo que constrói uma árvore de roteamento para toda a rede a partir de interações locais propagadas a partir do sink (flooding). Para fins de avaliação, foi escolhida uma regra de escolha de rotas onde cada nó simplesmente escolhe como pai o vizinho a partir do qual a mensagem de construção chegou primeiro. Tal procedimento de construção é repetido periodicamente para agir de forma pró-ativa a eventuais necessidades de reorganização introduzidas pela dinâmica da rede.

- Algoritmo Reativo (SID) – Algoritmo que permite a construção de rotas individuais para nós com necessidade de enviar dados. O algoritmo funciona a partir da descoberta de rotas feita pelo nó fonte enviando mensagens de controle em flooding. O sink, por sua vez, escolhe uma rota individual propagando mensagens de interação local pelo melhor caminho reverso ao fonte. Essa estratégia permite a economia de recursos em situações de tráfego eventual porque não precisa manter uma infra-estrutura para toda a rede, economizando mensagens de controle.

- Regra de Adaptação – Nesta solução é aplicada uma regra de adaptação, realizada no sink, que monitora a ocorrência de novas detecções de eventos na rede. A partir dessa percepção, um filtro de média móvel é utilizado para a estimativa de novas detecções. Se a estimativa prever novas detecções em um dado período de observação, o comportamento reativo dá lugar a um comportamento pró-ativo. O comportamento reativo é preferencialmente usado em cenários de eventos para permitir economia de energia em intervalos de inatividade, mas se novas detecções forem estimadas, é mais vantajoso tomar o comportamento pró-ativo, que constrói uma infra-estrutura para toda rede, do que o custo de descoberta de rotas para novas fontes de dados.

A solução proposta é avaliada em diversos cenários de simulação, onde a solução adaptativa híbrida é comparada com as soluções pró-ativa e reativa aplicadas de forma individual. Basicamente, observa-se que com poucas detecções de eventos, o algoritmo reativo permite maior economia de energia em relação ao pró-ativo. Conforme esse número de detecções aumenta, a situação se inverte favorecendo o algoritmo pró-ativo. Com a solução adaptativa híbrida do Multi, sem intervenção de entidades externas à rede, seu desempenho corresponde ao melhor dos desempenhos dos algoritmos independentes mesmo variando-se o número de detecções de eventos. Isso acontece porque Multi adota a melhor estratégia de roteamento para a situação monitorada da rede. Tais resultados se mostraram compatíveis quando comparado o Multi com soluções clássicas de roteamento como as versões do Direct Diffusion.

Capítulo 7 - Integração de Funções Auto-Organizáveis: Roteamento e Controle de Densidade

Espera-se que muitas funções auto-organizáveis sejam aplicadas simultaneamente para o funcionamentos das RSSFs. Como conseqüência, o projeto dessas funções deve ser considerado de forma conjunta para a obtenção de um funcionamento correto e eficiente. Esse aspecto é melhor descrito como um dos itens da metodologia de projeto apresentada nesta tese. A partir dessas idéias gerais, este capítulo apresenta um estudo de caso que considera a integração das funções de roteamento e controle de densidade, que são funções básicas para o funcionamento de RSSFs. Além disso, a solução obtida é também uma contribuição específica deste trabalho.

A integração das funções de roteamento e controle de densidade se justifica pelo impacto que uma pode causar na outra. Basicamente, a função de controle de densidade altera o estado dos nós entre ativo e inativo, proporcionando a economia de energia. No entanto, essa dinâmica introduzida na rede pode impactar na infra-estrutura de roteamento. Rotas previamente definidas pelo algoritmo de roteamento podem ser destruídas quando um nó passa do estado ativo para o inativo, podendo causar perda de dados. Adicionalmente, para que as duas funções sejam realizadas de forma auto-organizável, os nós da rede devem interagir localmente, trocando informações, para a formação de rotas ou execução do controle de densidade. Se as funções são integradas, pode-se economizar trocas de mensagens para interação local ao se juntar as informações necessárias a cada função. Com isso, proporciona-se uma economia de energia adicional.

A solução proposta considera a formação de infra-estrutura de roteamento baseada em árvore, com o algoritmo EF-Tree, previamente apresentado neste trabalho, e uma solução de roteamento bem referenciada na literatura, o OGDC. Para fins de comparação, foram adotadas as duas estratégias de integração também presentes na metodologia de projeto proposta, conforme descrito a seguir:

- Sincronização – Nesta solução, chamada RDC-Sync, a integração das funções é feita simplesmente considerando-se uma periodicidade de execução para cada uma e realizando a criação da infra-estrutura de roteamento logo após o controle de densidade. A idéia é realizar a função de roteamento imediatamente após a de controle de densidade, onde o roteamento é formado com o estado dos nós já definidos. Mas como as atividades são realizadas de forma distribuída, é difícil precisar a duração de cada função e isso é feito com base em estimativas. Como resultado, não se tem uma integração tão precisa.

- Projeto Integrado – Nesta solução, chamada RDC-Integrated, as duas funções são consideradas em um projeto integrado onde reune-se as informações necessárias a cada uma das funções nas mesmas mensagens de interação local. Dessa forma, criou-se uma regra de decisão local em que, ao mesmo tempo que o estado de atividade dos nós é definido, as rotas são refeitas. Com isso, a possibilidade de perda de dados por rotas quebradas é minimizada.

As soluções propostas foram implementadas e avaliadas em cenários de simulação. Comparações foram feitas entre as soluções desenvolvidas e um roteamento em árvore sem controle de densidade, que mostra a necessidade da função de controle de densidade para maior economia de energia. Da comparação entre as duas soluções de integração, observa-se o desempenho superior da solução RDC-Integrated em relação à taxa de perda de dados e consumo de energia. Sua desvantagem é a necessidade de reprojeto das funções de roteamento e controle de densidade em uma única solução, aumentando a complexidade do trabalho dos desenvolvedores.

Capítulo 8 - Considerações Finais

Auto-organização é um conceito importante e desafiador a ser aplicado aos atuais sistemas computacionais de grande escala, especialmente as RSSFs. Esse conceito é baseado na obtenção de um comportamento global através de interações locais entre os elementos do sistema e pode ser empregado para a obtenção de sistemas autônomos sem a necessidade de um controle centralizado.

Embora muitas soluções auto-organizáveis já existam para as RSSFs e outros sistemas computacionais, aspectos gerais de projeto ainda não são cobertos satisfatoriamente para serem aplicados com objetividade no desenvolvimento de RSSFs auto-organizáveis. Assim, este trabalho apresenta avanços nessa área de pesquisa promissora com duas contribuições em um nível mais conceitual. A primeira apresentou uma metodologia de projeto que reúne importantes aspectos práticos de projeto e um guia que podem ser úteis para o entendimento e desenvolvimento de novas funções auto-organizáveis. Já a segunda contribuição considerou um aspecto importante de RSSFs auto-organizáveis que é o atendimento de necessidades externas e requisitos globais.

Além dos aspectos mais conceituais relacionados à auto-organização de RSSFs, este trabalho proporcionou o desenvolvimento de soluções particulares para problemas específicos dessas redes. Esse foi o caso da solução integrada de roteamento e controle de densidade e da solução adaptativa híbrida de roteamento, soluções inovadoras e que representam contribuições mais individuais desta tese.

Adicionalmente, embora tenhamos focado em diversos exemplos e características particulares das RSSFs, acreditamos que essas contribuições sejam úteis a outros domínios tais como o de redes ad-hoc em geral, o de redes veiculares e o P2P. Assim, todo o conhecimento adquirido e gerado nesta tese abre muitas possibilidades de trabalhos futuros relevantes para o desenvolvimento de sistemas computacionais autônomos.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The current computer systems are expected to be composed by a high number of interconnected components, and these systems tend to become more complex due to the consequent increase of interaction possibilities among their components. This could be beyond the capacity of designers and administrators to predict and control these interactions, so operational functions such as configuration and optimization become very complex as well as massive. In this context, research advances have been emerged in the development of autonomous computer systems that treat this complexity with minimum interaction of human administrators. Generally, such solutions apply intelligent system components to perform the self-management of the whole computer system. This approach tends to be so promising in future networks that important IT companies have started initiatives in this direction such as IBM Autonomic Computing [Kephart and Chess (2003)], Sun N1 Grid Systems [Sun Microsystems Inc. (2005)], and Microsoft Dynamic Systems Initiative (DSI) [Microsoft Inc. (2005)].

Another important concept being applied to large-scale and dynamic network systems is the concept of self-organization [Heylighen (2002)]. Its main idea is the achievement of a global behavior through simple local interactions among the system elements. This characteristic leads to networks less dependent of a central control because the system intelligence is totally distributed in all the system elements. Thus, self-organization consists of a more powerful paradigm achieving highly adaptable, simple and lighter-weight systems, and which is complementary to technical initiatives cited above [Jelasity et al. (2006)]. In fact, this concept has been applied to several network domains [Zambonelli et al. (2005)], mainly those ones in which the network elements already present an autonomous feature such as the ad hoc networks.

Nowadays, a very active research field with such characteristics is the Wireless Sensor Network (WSN) [Clare et al. (1999); Akyildiz et al. (2002); Loureiro et al. (2003)]. These networks consist of many constrained sensor devices connected among themselves by a wireless medium to perform distributed sensing tasks. They are expected to be used in different applications [Arampatzis et al. (2005)] such as environmental and health monitoring, surveillance, and security.

Additionally to the scale, these sensor networks must be able to operate under very dynamic conditions, must adapt themselves to the environment and need to operate, in most cases, in an unattended mode (i.e., without external interference and control). These features lead to the need of deployment of systems which must be autonomous, and the concept of self-organization becomes the most appropriated in this intend.

In fact, WSNs were conceived under the self-organization paradigm, and in the development of these networks several proposals have applied the self-organization concept (sometimes implicitly) in specific functions such as communication, clustering and density control. However, these solutions did not cover a more general and practical view of the concept application that can be used for new designs and developments. Thus, these points have motivated this work.

1.2 Objectives

This work aims to extend the discussion on the application of the self-organization concept to the WSNs. The main objective is to provide general design models that can be used to guide the modeling and development of new self-organizing functions in these networks. This is done by the proposition of a design methodology, which relates common and practical design aspects of WSNs, and a management scheme, which puts in context the local and decentralized characteristic of the self-organization concept with the practical vision of adjusting the network behavior to global aspects.

Additionally, this work aims to apply the self-organizing concept in specific problems of WSNs. The goal is the development of practical solutions that show the importance of the proposed general design aspects, and that can be seem as individual contributions. These contributions are present in several case studies showing the application of the general models, and in particular, in two specific solutions regarding the routing function, which is a fundamental function of a WSN, and its integration with density control.

1.3 Contributions

Following the objectives described above, the contributions of this work are:

A general view about self-organization and WSNs. This work provides a general view about self-organization and WSNs. We divided this contribution in two parts. First, we present the self-organization concept, a discussion on its application in computer systems, examples of self-organization in nature that motivate its usage, and some applications of the concept in the Computer Science domain. In the second part, it explores the application of the concept in the WSN domain by introducing such networks, describing their characteristics that motivate the application of the concept, and discussing some important functions developed under this point-of-view. The goal of this general view is to present some fundamentals and related work regarding self-organization in WSNs.

A design methodology for self-organizing WSNs. The design process of self-organizing functions may not be trivial, and there is a lack of general methodologies to help this process. Thus, it is proposed a conceptual framework and a guideline that puts together some insights and important aspects for the design of self-organizing WSNs.

A management scheme for self-organizing WSNs. In practice, WSNs serve to a purpose and application and management goals can change along the time. This need requires a "top-down" approach as complementary with the "bottom-up" aspect of self-organization. Thus, it is proposed a management scheme to allow to external or higher-level management entities to change the network self-organizing behavior according to different global goals or network perceptions. This scheme can be seen as a step forward in the integration of self-organization and self-management concepts.

A hybrid adaptive routing solution for WSNs. Routing is a basic function in WSNs, and this contribution shows how different self-organizing strategies for routing can be autonomously adapted based on global network perceptions in order to achieve a more efficient self-organizing solution. The proposed solution is applied to event-driven scenarios, and it consists of a routing algorithm that combines both reactive and proactive strategies for infrastructure organization, and an event-detection estimation model for the adaptation between these strategies.

An integrated solution considering routing and density control. Different self-organizing functions are expected to be applied in WSNs. As a consequence, the design of these functions must be considered together in order to achieve a correct as well as efficient operation. This contribution presents an integrated solution considering both routing and density control functions in WSNs, and shows the need and the benefits of an integrated approach.

1.4 Thesis Outline

This thesis is organized as follows.

The general fundamentals and related work are presented in Chapters 2 and 3. Chapter 2 presents an overview of the self-organization concept and its application in the Computer System domain. Some examples found in nature and applications in computer systems are presented to illustrate and motivate its use. Chapter 3 presents a deeper discussion about the self-organization concept in WSNs. It is discussed the needs of self-organizing features, its meaning for WSNs, and several functions that are considered under this point-of-view. It is also discussed some alternative approaches to self-organization in WSNs and the advantages of each of them.

Design aspects of self-organizing WSNs are presented in Chapters 4 and 5. Chapter 4 proposes a design methodology that can guide the development of new self-organizing functions. It presents some general aspects, a framework containing practical implementation aspects, and a guideline relating important design phases. Chapter 5 proposes a scheme in which the designed self-organizing functions for proper operation of WSNs are put in context with high-level management functions. This contribution can be seen as a complementary view to traditional autonomic management approaches for WSNs.

The following chapters explore some practical applications of self-organizing WSNs. Chapter 6 presents and evaluates a hybrid adaptive routing solution, called Multi, and Chapter 7 shows a case study on the integration of routing and density control functions.

Finally, Chapter 8 concludes this thesis by discussing the general contributions present in this work, and complementary ideas that can be treated as future work.

Chapter 2

Self-Organization: An Overview

This chapter presents an introduction to the concept of self-organization and discusses its application in the computer science domain. Self-organization is a concept studied in natural systems and has relevant characteristics for obtaining autonomous computer systems. In this sense, we present some fundamentals in Section 2.1, 2.2 and 2.3. Examples of self-organization in natural systems, which can motivate its use, and the application of the concept in computer science are presented in Section 2.4. Finally, Section 2.5 presents some concluding remarks of this chapter.

2.1 The Self-Organization Concept

Self-organization is a well-known concept in the literature and it has been employed in different areas such as Physics, Chemistry and Biology (e.g. Haken (1983)). Several studies present the self-organization concept in different forms and its definition is sometimes imprecise, but its general idea seems to follow the definition of "spontaneous creation of a globally coherent pattern out of local interactions" [Heylighen (2002)].

The main characteristics of the distributed character of self-organization are:

- **Decentralized Control.** This is the general feature of self-organizing systems. It means that the system organization is achieved without any centralized decision making. The decision of the participation of each individual is taken by themselves according to local interactions with the environment and with other individuals. However, this individualist behavior can cause the emergence of a global coherent behavior.

- **Flexibility and Adaptability.** Due to the dynamic of interactions among individuals to perform and maintain the organization, these systems present the capacity to adjust to changes in the environment or in their elements.

- **Robustness.** As a consequence of the former characteristics, the system can accommodate to failures or interferences without the interaction of a centralized entity.
- **Scalability.** Due the distributed character of the concept in which an element interacts only with a restrict vicinity to achieve a global behavior, the system becomes very scalable.

2.2 Self-Organization in Computer Systems

The consequent characteristics of self-organization discussed above led to the application of the concept in several computer systems. This is specially true in very distributed systems that are composed of many elements, as described in Zambonelli et al. (2005) and in several examples later in this chapter.

However, a more precise view is needed. In computer systems, there is a desired global goal to be achieved through local interactions, because the system has a purpose. For instance, in Heylighen and Gershenson (2003) a self-organizing system is defined as a "system which creates its own organization". An organization is a "structure with function, where structure means that the components of a system are arranged in a particular order and function means that this structure fulfills a purpose".

In contrast to other areas, like Physics, Chemistry and Biology, in which the laws of the nature (chaotic dynamics) control the local interactions, in computer systems, and especially wireless networks, we need to devise simple local interaction rules, codified in distributed algorithms, to achieve the desired goal. This requirement is due to the efficiency necessary for the operation of such systems.

Under a practical view, the concept of self-organization is very related to the concept of distributed algorithms in "distributed message-passing systems" [Barbosa (1996)], in which algorithms running on several individuals of a communicating system exchange messages to accomplish a function. And they become closer in some practical systems when distributed algorithms are revisited under the term "localized algorithms" [Estrin et al. (1999)], which refers to interactions among individuals in a restrict vicinity to achieve a global collective behavior. Particularly, the localized algorithm term seems to refer to the implementation of distributed algorithms following the self-organization concept, and so it is unnecessary because it seems a new term for something existing before.

A natural question at this moment could be why not simply to use the distributed algorithm concept to build decentralized systems instead of using the self-organization term. And the answer is that with self-organization we are focused on a global property of the system, which can be a particular arrangement of its elements or a coherent behavior, and how it is achieved through these local interactions. Also, an important aspect of self-organization is that many complex behaviors for the system can be achieved through simple local rules, and this reduces the system complexity and computational requirements at all.

2.3 Alternatives to Self-Organization

Obviously, self-organization is not the only way to organize a system, and the discussion of alternatives can also help on the understanding of the concept.

Basically, alternatives of self-organization are centered on leadership. In such cases, there is a control hierarchy or at least a central entity that takes information about the system, decides what to be done, and sends commands to individuals about their tasks. The system organization is not done only through local interaction of individuals.

The control of the system by a leader can present different forms. For example, the leader can constantly monitor and define each element action, or it can build some kind of recipe containing sequential instructions that rigidly specify the temporal actions of the individual's contributions to the whole system behavior.

Examples in nature are known, as in the case of the queuing of ducklings, in which there is a mother who leads the queue and assures that all the ducklings are following her [Camazine et al. (2001)]. Also, the control by central entities is the traditional approach of computer systems, in which a human or even an autonomous entity defines and configures the participation of each element of a system.

Although the alternatives based on central control exist and are feasible, specially for computer systems, they present some drawbacks. The main drawback relies on scenarios where systems are large-scaled or very dynamic. In these cases, the central entity is supposed to present a very effective communication with all the elements, to be well informed about and interact with each one of them, and a sophisticated processing capacity to evaluate and define the participation of each element. By defining hard behavioral recipes for the individuals, which diminishes the monitoring and acting requirement, the system becomes less flexible because it looses its capacity to adjust to the environment dynamics, failures etc. Also, a clear problem is that the system becomes very dependent of the presence of the leader.

On the other hand, central control can lead to systems with more efficient organizations. As the central entity uses a global vision of the system and it can control each individual, it can define a better arrangement of these individuals to achieve a better or even an optimum performance. Obviously, it depends on the system size and the processing capacity of the central entity.

2.4 Examples of the Concept Application

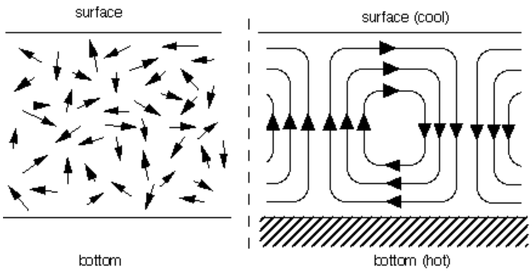
Some examples of the concept application are described as follows. They are divided in two classes, self-organizing systems in Nature, and in Computer Science.

2.4.1 Self-Organization in Nature

Nature has several examples of self-organizing systems, in which patterns and structures emerge spontaneously, i.e., without the influence of a central entity. These systems have attracted the attention of researcher for a long time and in different areas such as Physics, Chemistry and Biology [Ashby (1947); Kauffman (1993); Camazine et al. (2001)]. Their main goal is the understanding of the basic mechanisms achieving self-organizing behavior. To illustrate the application of this concept, we cite some examples as follows.

2.4.1.1 Physical Systems

In Physics, where the term seems to be first applied, one of its use refers to the structural phase transformation such as spontaneous magnetization and crystallization. As an example, Fig. 2.1(a) shows the magnetization case. Initial disordered spins of a potentially magnetic material, such as iron, are caused by random movements of molecules in the material, so their magnetic fields cancel each other. However, when its is affected by an external magnetic field or the temperature decreases, the spins align themselves to point in the same direction producing a overall magnetic field. This organization happens because spins pointing in opposite directions repel each other, and spins pointing in the same direction attract each other.



(a) Spontaneous Magnetization. (b) Bénard phenomenon.

Figure 2.1: Examples of self-organization in Physical Systems.

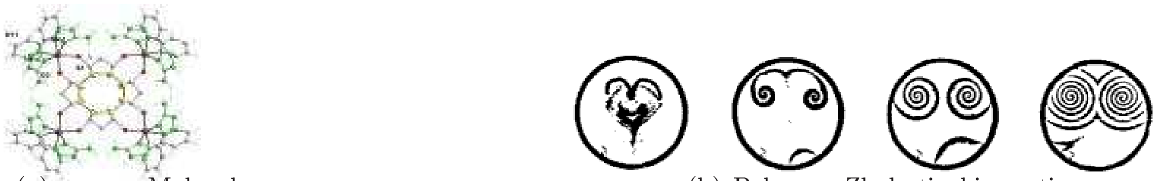
Another example which can be illustrated is the Bénard phenomenon in Fig. 2.1(b). Here, a liquid is heated uniformly from below and cooled uniformly at its surface. The warm liquid tends to move up to the surface and the cool liquid at the surface similarly tends to move to the bottom. The both opposite movements cannot occur at the same time, so a kind of coordination happens with the flows and a dynamical pattern of hexagonal cells emerges.

These examples can be found in Heylighen (2005). And many other examples could be cited such as equilibrium thermodynamics, and laser, superconductivity and Bose-Einstein condensation, in the quantum domain.

2.4.1.2 Chemical Systems

In Chemistry, a common example of self-organization includes self-assembly, which denotes a reunion of molecules reaching an equilibrium state without the help of an external entity, and producing larger-scale structures and patterns, as the examples of Fig. 2.2(a). Examples of self-assembled chemical structures include liquid crystals and organization of lipids into membranes.

Another example refers to the oscillating chemical reactions, such as the Belousov-Zhabotinski reaction. In such kind of chemical reaction a prolonged state of non-equilibrium is maintained leading to macroscopic temporal oscillations and spatial pattern formation. Fig. 2.2(b) shows waves of chemical activity propagating through a receptive liquid medium.



(a) Molecules self-assembly.

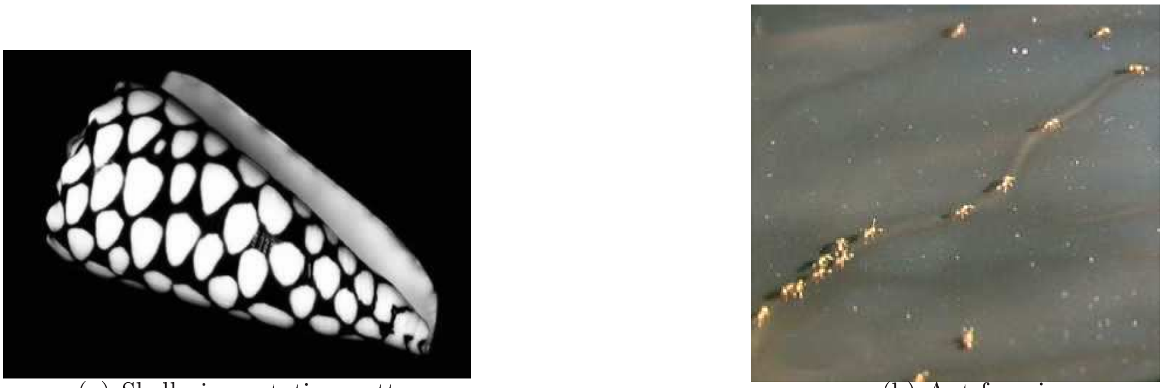
(b) Belousov-Zhabotinski reaction.

Figure 2.2: Examples of self-organization in Chemical Systems.

2.4.1.3 Biological Systems

In biological systems, there are many phenomena that have been described as "self-organizing". They range from the study of simple self-organized patterns found in the pigmentation of shells and animal skins (e.g., Fig. 2.3(a)), to the complex social behavior of animals. In this last case, we can find the most interesting examples of biological self-organizing systems because they deal with the creation of structures by social animals (e.g., social insects as bees, ants and termites), which are more complex then the inanimate objects studied in self-organizing systems of Physics and Chemistry, and this suggest that self-organization should be expected in human society too.

A very classical example of self-organization in social behavior is the ant foraging, as the example in Fig. 2.3(b). In the search of food, ants leave on the ground a trail of chemical messages called pheromones. This trail is followed by other ants in a short time, and when they return home from a food source they reinforce this trail, bringing other ants. When the food source is exhausted, the trail is not reinforced and it slowly dissipates. So, the ants restart the foraging process to build new paths from new food sources.



(a) Shell pigmentation pattern.

(b) Ant foraging.

Figure 2.3: Examples of self-organization in Biological Systems.

Many other examples of self-organization have also been studied, as the formation of lipid bilayer membranes, the self-maintaining nature of systems from the cell to the whole organism (homeostasis), the process of how a living organism develops and grows (morphogenesis), and the origin of life itself from self-organizing chemical systems.

2.4.2 Self-Organization in Computer Science

As described before, the characteristics of the recent computer systems match very well with the self-organization concept, and natural self-organizing systems reinforce this idea. Thus, many research areas of computer science have been applying such concept. Some examples are described as follows.

2.4.2.1 Multi-Agent Systems

A sub-field of Artificial Intelligence has been applying distributed processes in problem solutions. This is the area of multi-agent systems [Russel and Norvig (1995)]. Some of the studied problems deal with the cooperation among several agents, forming agencies, to solve problems that they cannot solve individually. These agent interactions clearly present self-organization characteristics.

An example of self-organization in Multi-Agent Systems is the coordination models [Ciancarini (2001)] that defines agent interactions (protocols and rules) for tasks such as life control of agents (creation and destruction), communication flows among agents, spatial and mobility control, and agent synchronization [Ciancarini et al. (2000)].

2.4.2.2 Cooperative Robotics

Many applications of multi-agent systems are done in the physical world through cooperative robotics [Balch and Parker (2002)]. In such applications, robots can self-organize to accomplish a given task. For example, bio-inspired swarms of robots are applied for field exploration in a coordinated motion or cooperative task accomplishment (See Swarm-Bots Project [Dorigo et al. (2004)]). Fig. 2.4 shows the self-organization of a swarm of robots to pass through an obstacle.

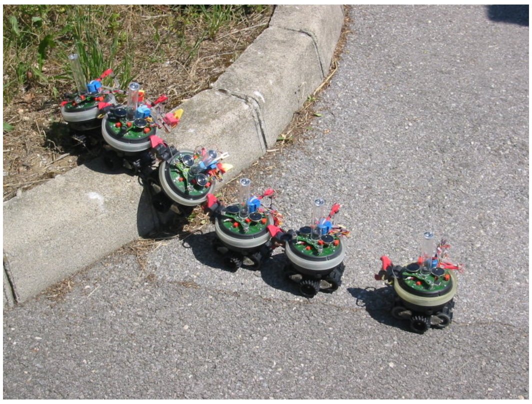


Figure 2.4: Exemple of self-organizing robots.

2.4.2.3 Networked Systems

The networked system area represents a great potential for the application of self-organizing principles and mechanisms. Recent and relevant work, such as in Zambonelli et al. (2005) and Prehofer and Bettstetter (2005), have been being applied from the micro scale, such as the micro-sensor networks, passing through the medium scale, such as the ad hoc networks of PDAs, to the global scale, consisting of P2P and the internet itself.

In addition to the sensor networks, which we explore in a deeper view in this work, the ad hoc networks have natural infrastructureless and dynamic characteristics, and self-organization has been considered since their initial propositions. Several works deal with self-organizing communication protocols (MAC, routing etc.), such as in Clare et al. (1999) and in Blazevic et al. (2001), but we can also cite application level approaches such as interaction models and spatial information sharing of TOTA Middleware [Mamei and Zambonelli (2004)].

In computational grids, which are composed by different computational elements such as machines and data repositories, we can cite the community formation (clusters) based on different criteria such as performance, trust and cost of usage [Lynden and Rana (2002)].

In P2P networks, which also have very dynamic characteristic, we can see the self-organization concept applied to form overlays [Castro et al. (2004)] to build and maintain logical relationship among their components.

Characteristics of self-organization are also found in the Internet TCP protocol, in which the decentralized mechanism of adaptive congestion control deals with global network congestion. Other studies such as the formation of Web Communities from web page analysis [Flake et al. (2002)] can be seen as examples of self-organization in global scale.

2.5 Chapter Remarks

Throughout this chapter, we presented a discussion about the self-organization concept, its consequent characteristics, how it is found in nature and how it is being applied in computer systems. In this last case, the great motivation is the scale and ubiquitous characteristics of the modern computer systems, which difficult their control by a central entity. Thus, self-organization becomes a solution to deal with these systems with simplicity, in which simple local rules performed by the system components make a global behavior emerge for the whole system.

An important observation for the subsequent development of this work is that self-organization is not synonymous of autonomous behavior. A system can be autonomous based on intelligent computational entities which control the system individuals. Also, this control entity can be present inside the network, out of sight from the system users. Thus, someone could figure out that this system is self-organizable. However, the self-organization concept requires the absence of any central control entity, independently from being internal or external to the system. A better discussion regarding the self-organization concept and the autonomous operation, specifically with the new paradigm of autonomic computing, is presented in Chapter 5.

In the next chapter, we will present a better discussion and related work referring to the self-organizing wireless sensor networks, which are the focus of this thesis.

Chapter 3

Self-Organizing Wireless Sensor Networks

This chapter presents an introduction to WSNs focusing on its self-organizing characteristics. Such networks have been attracting the attention of the scientific community due to its wide applicability potential. However, several envisioned applications depend on its autonomous operation capability, thus, the self-organization concept has emerged as a solution. In this chapter, we present an introduction to WSNs in Sections 3.1 and 3.2. Next, in Section 3.3, we discuss the need of self-organizing features and its meaning for WSNs, and in Section 3.4 we present common self-organizing functions of these networks pointing at some related work. Section 3.5 discusses alternative approaches to self-organization, some comparisons and refers some important related work to the WSN domain. Finally, Section 3.6 presents the concluding remarks.

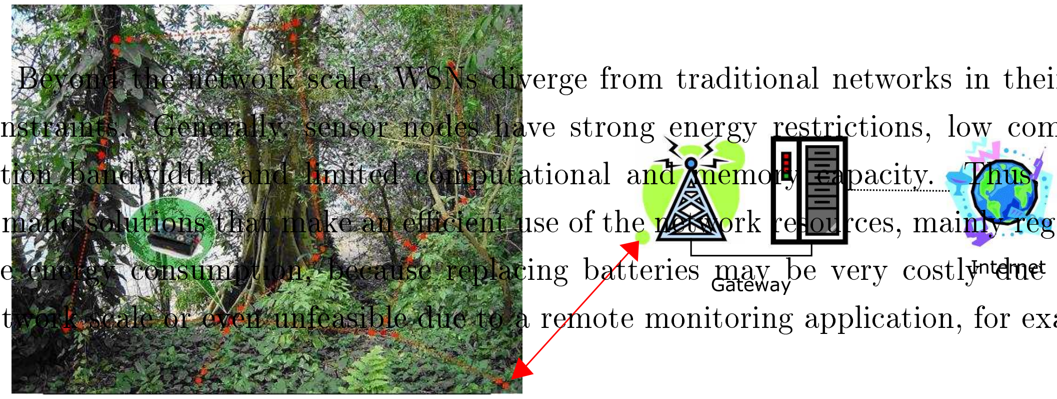
3.1 An Introduction to WSNs

Wireless sensor networks (WSNs) [Clare et al. (1999); Pottie and Kaiser (2000); Akyildiz et al. (2002); Loureiro et al. (2003)] are a particular case of ad hoc networks that differ from traditional networks in several features. They consist of low-cost and compact devices with sensing and processing capabilities, and they are connected among themselves by a wireless medium to perform distributed sensing tasks. When applied in high scales, these characteristics allow a wireless sensor network to create a powerful platform for processing data collected from the environment.

This type of network has become popular in the scientific community due to its applicability potential. They have been employed in several areas [Arampatzis et al. (2005)], such as environmental, industrial, health monitoring, surveillance, disaster recovery, military and security applications.

A basic example of WSNs architecture is illustrated in Fig. 3.1. Several sensing nodes are deployed in an area of interest. These nodes send their sensed data to a gateway (also called sink node), usually in a multi-hop way, to be further processed, stored and accessed by user applications.

Figure 3.1: Example of a WSN.



Beyond the network scale, WSNs diverge from traditional networks in their hard constraints. Generally, sensor nodes have strong energy restrictions, low communication bandwidth, and limited computational and memory capacity. Thus, WSNs demand solutions that make an efficient use of the network resources, mainly regarding the energy consumption, because replacing batteries may be very costly due to the network scale or even unfeasible due to a remote monitoring application, for example.

3.2 Sensor Platforms

As a typical sensor network platform we can exemplify the Mica2 sensor nodes [Crossbow Technology Inc. (2004)] (see Fig. 3.2). They were developed by researchers of Berkeley University and are traded by Crossbow Inc. A Mica2 node is a low-power device with a CC1000 wireless radio, a RISC 8-bit microprocessor ATMEGA 128L with 4KB of RAM and 128KB of flash memory, and interfaces for several sensing devices, such as thermometer and accelerometer. Applications are developed on the TinyOS operating system [TinyOS (2004)], which is open-source and very modular.

The Mica2 node has become the widest applied platform and a reference for sensor network applications. It also evolved to the MicaZ version in order to improve the communication with the IEEE 802.15.4 standard [IEEE (2004)]. In addition, there is still a lot of alternative platforms in development. We cite some examples as follows (More details are explained in Ruiz et al. (2004)).

(a) Mica2 sensor node.

(b) Mica2Dot sensor node.

Figure 3.2: Examples of Mica2 sensor nodes.

In the academic domain, we can exemplify: The Smart Dust Project [Dust (2006)], from the Berkeley University, whose goal is the reduction of the sensor scale to the order of one cubic millimeter; The μ-AMPs [μAMPS (2006)], from the MIT, which applies an adaptive energy control; The WINs Project [WINS (2006)], from the Rockwell Science Center and University of California, which introduces a more powerful processing and communication platform but with less energy concerns; and the BEAN Project [Vieira (2004b)], from our own research group (SensorNet [DCC/UFMG (2006)]) at Federal University of Minas Gerais, which applies common manufactured components and introduces a new event-driven operational system, the YATOS [Vieira (2004a)].

As a commercial platform, the MillenialNet [Millennial Net (2006)] is an expressive one. It introduces a complete sensor network solution that considers sensing applications, all the hardware and a self-organizable solution for communication with special elements called Routers.

3.3 Self-Organization in WSNs

In general, several WSN applications require the adoption of a totally autonomous behavior. It is due to the necessity of sensing in remote places, which can be inhospitable or of hard access, also in consequence of the network scale, which increases the network complexity due to the exponential number of possible interactions among nodes that can make the execution of management actions by administrators or central entities massive or even impossible.

In addition, the scenarios where WSNs are applied may be very dynamic. In these networks, topological changes are very frequent. For example, sensor nodes can be destroyed by the environment, they can have their energy depleted, new sensor nodes can be added to the network, or the wireless communication can be intermittent due to interferences or obstacles. Also, some sensing architectures assume mobile nodes, as the Robomote Project [Sibley et al. (2002)]. Thus, the algorithms and protocols for this kind of network must be able to enable network operation during its initialization, and both normal and exception situations.

As described in Chapter 2, self-organization has become an important concept being applied in large-scale and autonomous network systems. Its main idea is the achievement of a global behavior through local interactions, which leads to networks less dependent of central control and that tend to be robust, i.e., resistant to perturbations. Thus, this concept matches the WSN goals very well, due to the autonomic characteristic of the WSN elements, and their distributed and cooperative requirement to achieve a common goal efficiently.

In fact, self-organization has been considered in WSNs since their first works. For example, it is pointed in Clare et al. (1999) the need of communication infrastructure creation in a self-organizing way. Later, the application of self-organization concept in the WSN domain was reinforced in Collier and Taylor (2004), in which some general initial aspects are considered. And, in Figueiredo et al. (2005b), we have complemented this view discussing some existing mechanisms applied in practice.

In a more objective view, the goal of a self-organizing WSN is to maintain all the network functions (e.g., communication, and collaboration), adapting to perturbations (topological or environmental changes), saving resources (e.g., energy, bandwidth etc.), along its lifetime and without the need of centralized control.

3.4 Self-Organizing Functions in WSNs

Under a practical view, some fundamental functions of WSNs that have been applied under the self-organization concept are exemplified as follows.

3.4.1 Communication

Communication is a basic functionality in any data communication network and it was the first class of algorithms to be developed considering the self-organizing approach, especially in ad hoc and sensor networks (e.g. Sohrabi et al. (2000)). The problem is the creation of a communication infrastructure in a distributed manner and its maintenance in the presence of topological changes. Basically, the fundamental communication function deals with the establishment of links (link layer), mainly the channel access organization (MAC sublayer), and routes between nodes (network layer). There are many works regarding communication infrastructure, as we can find in general WSN surveys [Akyildiz et al. (2002); Ruiz et al. (2004)], or in specific ones, such as for MAC

[Naik and Sivalingam (2004); Macedo et al. (2005b)] and routing [Al-Karaki and Kamal (2004)]. But some examples are discussed as follows.

3.4.1.1 MAC Layer

In MAC layer, several works consist in organizing the channel access in collision-free links, mainly based on TDMA schemes. An advantage of this approach is the possibility of saving energy in these more restricted networks. Examples of such algorithms are SMACS (Self-organizing Medium Access Control for Sensor networks) [Sohrabi et al. (2000)] and S-MAC [Ye et al. (2002)], which establish TDMA schedules among neighbor nodes in a dynamic way through local negotiation.

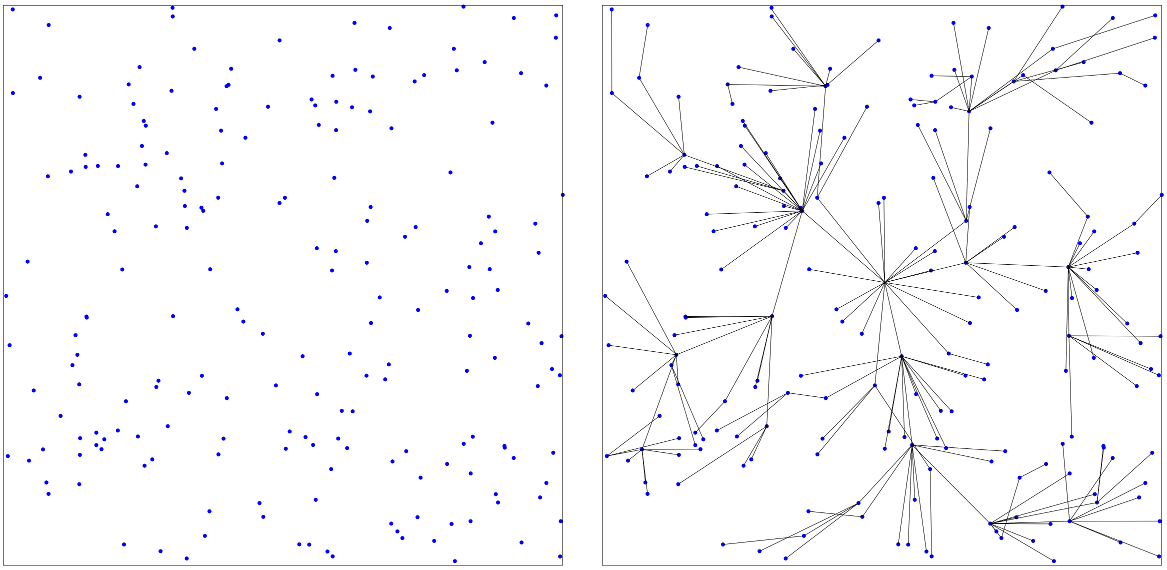
In most existing sensor networks, the channel access is done by a contention method, like the IEEE 802.11 MAC protocol. B-MAC [Polastre et al. (2004)] is an example for the Mica2 [Crossbow Technology Inc. (2004)] sensor node platform. This method does not require the maintenance of a special organization for channel access. When a node wants to transmit a frame it simply verifies if the channel is in use, controlling possible collisions. This process works as the channel access organization happens every time a node wants to transmit. An evolution of B-MAC that introduces radio power control in a self-configurable way was proposed as part of the SensorNet project in Correia et al. (2005).

3.4.1.2 Network Layer

The Network layer also concentrates several works dealing with routing infrastructure creation and maintenance [Al-Karaki and Kamal (2004)]. Basically, source and destination nodes interact with their neighbors and these interactions are propagated in a multi-hop way in the network for route discovery and maintenance. For example, AODV (Ad Hoc On-Demand Distance Vector Algorithm) [Perkins et al. (2003)] and Push Diffusion [Heidemann et al. (2003)] shares the behavior of data flooding from the sources for path discovery and directed requisitions from destination for specific backward path establishment.

Another simple and efficient solution adopted in WSNs is a routing tree formed through propagating local interactions from the destination (generally the sink). Some examples are SAR [Sohrabi et al. (2000)], TD-DES [Cetintemel et al. (2003)], One-Phase Pull Diffusion [Heidemann et al. (2003)] and some implementations in Woo et al. (2003). Fig. 3.3 illustrates a routing tree global pattern emerged from local interaction in which nodes choose their parents for data forwarding.

Particularly, in the SensorNet project, there are several works that deal with tree-based strategies for routing in WSNs, some of them with my direct participation. Some



(a) Before Self-Organization.

(b) After Self-Organization.

Figure 3.3: Tree Formation Example.

examples are: STORM [Nakamura et al. (2004)], which applies different strategies for tree construction giving different parent possibilities for data forwarding in the tree; Difuse [Nakamura et al. (2005d,c,b)], which applies data fusion methods for the failure detection and tree reconstruction needs; PROC [Macedo et al. (2006, 2005a)], which introduces a radio scheduling scheme for energy savings; and Multi [Figueiredo et al. (2004c,b)], which applies both reactive and proactive strategies for routing infrastructure construction. This last work is a relevant contribution of this thesis, and it is better presented in Chapter 6.

The establishment of these local interactions based on bio-inspired semantic behaviors, in particular the ant-foraging [Bonabeau et al. (1999)], is also present in works like the AntHocNet [Caro et al. (2005)] and Rumor Routing [Braginsky and Estrin (2002)].

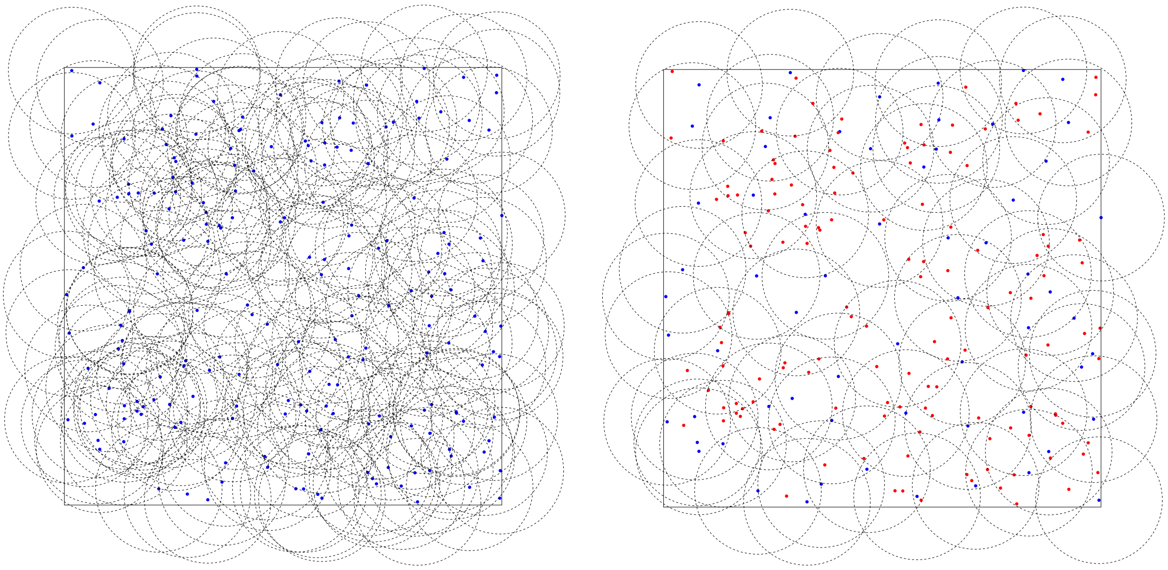
3.4.2 Density Control

Generally, sensor nodes are deployed in a high density and ad hoc way (i.e., without precise locations), causing many nodes sensing the same region and generating redundant data. This redundancy can lead to an unnecessary high traffic, increasing the bandwidth usage and energy consumption.

In particular, it is very important to pay attention to the communication aspects of a given algorithm or protocol for a WSN. The node transceiver consumes energy not only during transmission and reception, but also in channel listening (idle state). In general, the most expensive states in terms of energy consumption are transmission followed by reception, but the energy spent in the idle state cannot be neglected in the network lifetime. A possible strategy to save energy is to turn off the radio of redundant sensor nodes. When this happens, these nodes are no longer capable of communicating with other nodes, therefore the network topology and density are changed.

Thus, density control algorithms treat the redundancy of a wireless network in order to deactivate instantaneous unnecessary nodes and save its resources, mainly energy. The objective is to keep only a minimum set of active nodes guaranteeing the connectivity and/or coverage. The first guarantees that active nodes are connected through radio links, and the second maximizes the observability of the network area considering a sensing range for every node. Examples of density control algorithms considering connectivity are GAF [Xu et al. (2003)], SPAN [Chen et al. (2002)] and ASCENT [Cerpa and Estrin (2002)]. Examples considering coverage maintaining connectivity are OGDC [Zhang and Hou (2005)] and CCP [Wang et al. (2003)]. Particularly, an integrated solution for density control and routing is proposed in Siqueira et al. (2006a). It is part of the SensorNet project and it is better discussed as a contribution of this work in Chapter 7.

Fig. 3.4 shows an example of the emergence of a global pattern where the active nodes cover the sensoring area with a given density.



(a) Before Self-Organization.

(b) After Self-Organization.

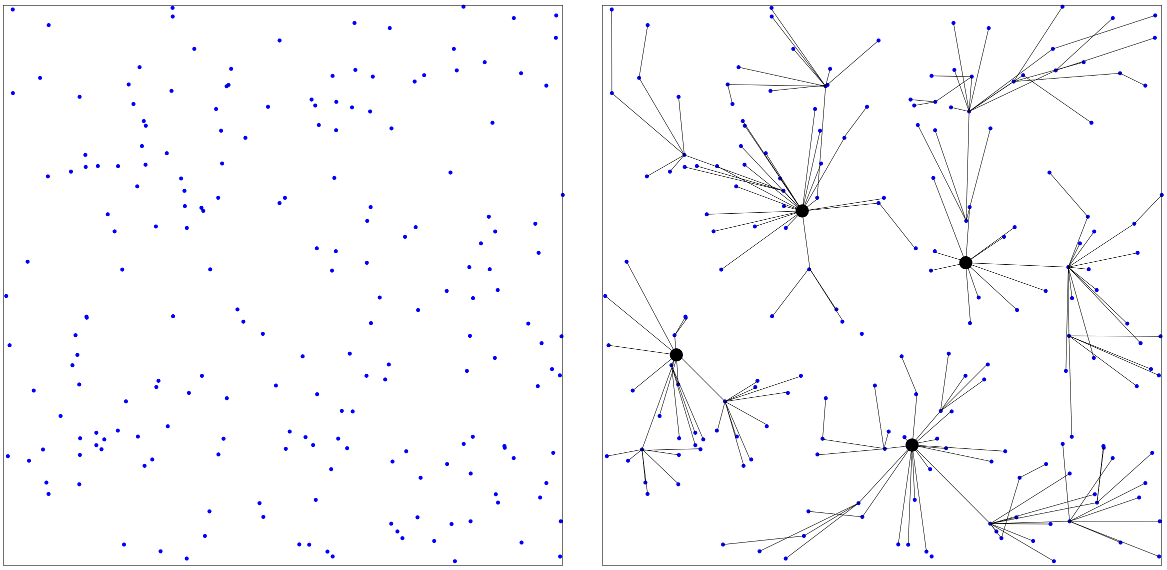
Figure 3.4: Density Control Example.

3.4.3 Clustering

The clustering of a network is the process of dividing it into hierarchical levels which are coordinated by a node with a special role called cluster-head. The organization of wireless networks in clusters has several advantages: it can group related nodes for some collaborative task; it can minimize the number of message transmissions since communication occurs among this small set of nodes, thus saving energy and bandwidth; the network complexity can be divided in subdomains; and finally, given all these points, it is possible to have a more scalable solution. In other words, a self-organizing clustering can support others centralized functions in other operational levels.

A classic clustering example for data dissemination in wireless sensor networks is LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol [Heinzelman et al. (2000)]. It is an algorithm that creates local clusters through the self-election of cluster-heads at periodical intervals. Many other proposals for clustering in ad hoc and sensor networks can be found in the literature, such as in Ramamoorthy et al. (1987) and Krishnan and Starobinski (2003), including more elaborated solutions for multi-layer organization and role-assignment, such as in Kochhal et al. (2003). All these referred clustering algorithms consist of proactive clustering approaches, and an interesting reactive one can be seen in Kwon and Gerla (2002), in which the creation and maintenance of clusters occur on-demand.

Fig. 3.5 shows an example of a cluster organization.



(a) Before Self-Organization.

(b) After Self-Organization.

Figure 3.5: Clustering Example.

3.4.4 Localization

Many applications and functions in ad hoc and sensor networks depend on the information of node localization. Some examples are routing, like in GPSR [Karp and Kung (2000)] where packets are forwarded to neighbors in the physical direction of the destination, in density control algorithms, like the GAF algorithm [Xu et al. (2003)], and applications that need to relate events with their locations.

Some proposals assume that information about node positioning is given by the GPS system. However, there are cases that the GPS signal is not available (e.g., indoor environment) or simply it is not possible to have a GPS receiver due to cost or size restrictions. An alternative solution is to use a distributed algorithm that allows nodes to find their relative positions inside the network area using range measurements between them to build a whole network coordinate system. An example algorithm with this functionality is called SPA (Self-Positioning Algorithm) [Capkun et al. (2001)]. It builds a local coordinate system for every network node from local interactions for distance estimative, and proposes a procedure to transform them in an uniform coordinate system for whole network. Other examples are found in Bulusu et al. (2004) and in de Oliveira et al. (2005). The latter as part of the SensorNet project.

3.4.5 Others

Many other functions in WSNs can also be treated under the self-organization paradigm. Some examples are security [Walters et al. (2005)], and synchronization [Ganeriwal et al. (2003)].

3.5 Alternative Solutions to Self-Organization

As described in Section 2.3 of the last chapter, centralized solutions are alternative to the self-organizing ones. Indeed, such solutions were the basis of traditional computer systems before the introduction of more dynamic and large-scale systems. And, the central control entities could be in form of human administrators or specialized management systems.

Although the WSNs, and other classes of ad hoc networks, were conceived under the self-organization paradigm, we can find several feasible proposals dealing with central control to perform some organizational functions. Generally, these proposals are used as complement to others still performed in a self-organizing way, mainly communication functions. In the literature, we find several proposals regarding centralized density control and node scheduling. Some examples are the node scheduling based on a Voronoi diagram [Vieira et al. (2003)], and other for density control such as in Slijepcevic and Potkonjak (2001) and in Cardei et al. (2002). They share the characteristic of using a model of the whole network to perform calculations and heuristic algorithms in order to establish a proper network organization.

A different approach uses optimization techniques. They take the advantage of a centralized model of the whole network to generate optimized organization solutions. Some examples are the pioneer works developed within the SensorNet project, such as optimization methods for density control and connectivity in Nakamura (2003), Nakamura et al. (2005e) and Menezes et al. (2005), in which combinatorial optimization methods are applied, or heuristic evolutionary approaches as in Quintão et al. (2004) and Quintão et al. (2005). Due to the high processing requirement of such solutions, they usually run in a special computational device outside the network, and depending on the processing time, they are not feasible to very dynamic networks.

Such a centralized approach is very common in management solutions, as we can see in the MANNA management architecture [Ruiz et al. (2003)], which is a pioneer work in the sensor networks domain and a fundamental work of the SensorNet project. A more specific example can be seen in Siqueira et al. (2006b), which applies centralized density control as a management function. However, as also described in Section 2.3, these solutions are expected to increase the need of communication efficiency of the centralized control entity with the whole network and its processing capacity. These requirements become harder when the sensor network is expected to be more dynamic and composed by huge amounts of nodes.

An intermediate approach deals exactly with the partition of the network in clusters, and the control of restricted network nodes by cluster-heads. This vision is also present in the management context, and it is also the basis of autonomic computing and self-management, as the proposal in Ruiz et al. (2005). Particularly, in this last citation, self-organization is used to form the cluster of nodes controlled by the cluster-head, but other functions such as density control are performed within the clusters in a centralized way. Thus, the complexity of a large network is divided, and the processing and communication capacity of cluster-heads is diminished for performing the central control. However, large networks can become very partitioned, and this can be undesirable by requiring other hierarchical levels, and the communication and processing requirements can still be higher than a solution entirely based on the self-organization concept. The discussion regarding the relation of management, and particularly self-management, and self-organization of sensor networks is extended in Chapter 5.

From this discussion above, we can conclude that there is room for all the alternative solutions due to the diversity of possible scenarios in WSN domain. For example, a particular solution will depend on the application requirements, the network architecture, the node platform constraints, the network scale, and the environment. But self-organization is an important concept to be present in almost all cases.

3.6 Chapter Remarks

This chapter presented a discussion about the WSNs, their need of self-organization features and several examples of important self-organizing functions. We discussed that WSNs present autonomous characteristics due to their scale and dynamic scenario applications, and self-organization really is an important concept to reach this view. However, its application must be very objective to accomplish WSN functions efficiently due to the constraints of the existing platforms.

Although self-organization has been considered since the initial proposals of WSNs, and, currently, there are several examples, we still have a lack of general paradigms or methodologies for the development of new self-organizing tasks or the deployment of them in WSN applications. In this sense, a step towards this direction is explored in this work in Chapter 4.

A disadvantage of the self-organization concept due to its decentralized characteristic is the difficulty of adjustment to external needs. As every computational system, a WSN serves to a specific purpose, and it must accomplish the user needs (e.g., administrators or specific applications) properly. Thus, by applying self-organizing functions a desired global behavior can be achieved in a totally distributed and autonomous way, but mechanisms to adjust the global behavior to different needs must be provided. This is the focus of the contribution described in Chapter 5.

Chapter 4

A Design Methodology for Self-Organizing WSNs

As described in the previous chapters, self-organization is an important concept in the development of autonomous computer systems, and specially WSNs. In practice, the design process of self-organizing tasks may not be trivial, and there is a lack of a general methodology that could be used in this context. Thus, this chapter presents a design methodology that reunites some insights that in some aspects refers to the associated related work, and propose a guideline to assist the design process of new self-organizing functions.

This chapter is organized as follows. Section 4.1 introduces the design problem and the contribution of this chapter. Section 4.2 presents some general aspects on the design of self-organizing WSNs, Section 4.3 presents some implementation aspects in a more practical view, and Section 4.4 proposes a design guideline. Finally, Section 4.6 presents some concluding remarks and possible future work in this promising research area.

4.1 Introduction

After the initial discussion presented in this work, we can see that self-organization plays an important role for obtaining autonomous computer systems, and specially in the WSN domain, which is the object of this thesis. While some basic mechanisms regarding local interaction rules to achieve self-organization seem to be well-known – as we can see in the existing solutions for WSNs presented in Section 3.4 or for other communication systems such as ad hoc and P2P networks in Section 2.4 – there is still the lack of methodologies and/or schemes to help in the design and development of such systems.

For the design of self-organizing WSNs, local interaction rules must be devised to achieve a desired global goal for the whole network. And in practice, this top-down translation is not easy due to the lack of formal models for this mapping. However, some general design aspects and insights can be helpful in this intent, and this chapter represents a step forward in the design process of self-organizing WSNs.

This chapter presents a design methodology composed by some general aspects, a design framework that unites important insights and practical characteristics of the application of self-organization in WSNs, and a design guideline that directs the search of self-organizing solutions in a more objective way. In order to show the applicability of the proposed methodology, this chapter also provides a case study built following the steps and insights of the proposed design guideline. This case study deals with the formation of self-organizing spatial patterns for sampled sensing.

The expected advantages of the proposed methodology are:

- It extends the discussion on the application of the self-organization concept in WSNs by focusing on the design process. This discussion is also expected to lead to a deeper understanding of the concept.

- It points to important practical aspects for the concept implementation. This can be very helpful for new designs by serving as examples or as a basis.

- It assists designers in the development of self-organizing functions for WSNs through the definition of important design steps. This can turn the design process more straightforward.

All the aspects presented in the proposed methodology are described in the following sections.

4.2 General Aspects

In contrast to other areas, such as Physics, Chemistry and Biology, in which the laws of the nature (chaotic dynamics) control the local interactions or it is obtained by an evolutionary process of million of years, in self-organizing communication networks, and mainly sensor networks, we need to devise simple local interaction rules, codified in distributed algorithms, to achieve a desired global goal.

In general, translating global goals into local interactions is not an easy task, mainly if efficiency is required due to network constraints such as energy, bandwidth and processing capacity. This translation aspect is still an open research area. However, we can identify general features of these algorithms as follows:

- **Decentralized Control.** There is no centralized entity that controls the participation of the system elements, and the global behavior must be mapped to local goals at each system element.

- **Common Goal.** The entire system must have a common goal to be achieved through the cooperation of the individuals. This common goal can be seen as the global behavior to be achieved from local interactions among the system elements, and can represent a particular arrangement or configuration of these elements, or a desired global performance.

- **Dynamic.** The system must be dynamic. The state of a system element must change based on its inputs (perceptions) and the local interactions with the other elements. Two important aspects related to obtaining dynamics are adaptation and exploration. Adaptation refers to the design of algorithms that adapt to changes in its environment. Basically, adaptation occurs when an element reacts to a perceived situation, such as a failure or an event, by changing its internal behavior or the way in which it interacts with the other system elements. A simple example of adaptation is multihop routing recovery in ad hoc and sensor networks. After a failure in an element of a path, the other nodes that use it have to react to discover an alternative route. Such an example was proposed in Nakamura et al. (2005d), and an example regarding event adaptation was proposed in Figueiredo et al. (2004b), which is part of this thesis. With the exploration aspect, the network elements must constantly try to increase its individual and consequently the whole system performance. Thus, the elements should explore different alternatives to perform their functions efficiently. A clear example of this aspect is found in the ant-foraging behavior, which is also applied to ad hoc routing (e.g., Caro et al. (2005)). Here, elements sometimes search for better paths towards the destiny.

- **Resource Preservation.** The nodes must preserve their limited resources and the self-organization cannot sacrifice the goal of the application running on the network. Thus, the dynamics of the system from local interactions must be objective and precise through simple established rules.

4.3 A Design Framework

In the following, we present a framework which joins common characteristics found in practical self-organization functions and that can turn easier and more comprehensive the design process of new self-organizing functions and applications. This framework is proposed by analyzing the existing related work (see Chapters 2 and 3).

Figure 4.1 shows a general view of our proposed framework. In the lowest level, we have the core of the framework that deals with local interactions and local information, which is also the core of the self-organization concept. This core uses the basic communication service for interacting with network elements. On top of the core, we introduce two basic services: Role-Assignment service and Neighborhood information abstraction. These services are present in the development of practical self-organizing functions in network systems, and can create an abstraction level for the development of new services or applications.

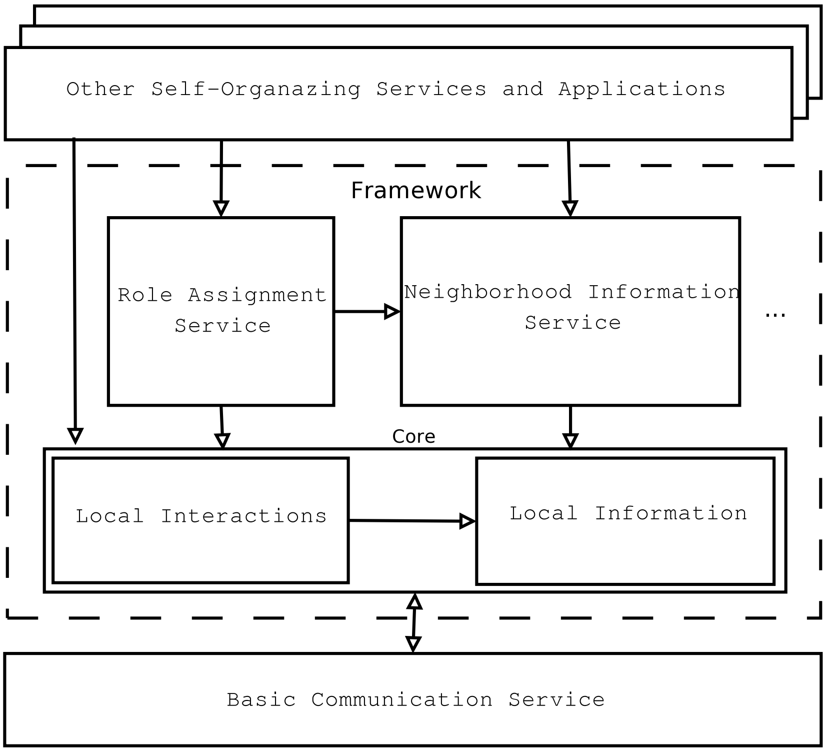

Other Self-Organazing Services and Applications
Framework
Role Assignment Service
Neighborhood Information Service
...
Core
Local Interactions
Local Information
Basic Communication Service

Figure 4.1: Self-Organizing Framework Overview.

4.3.1 Framework Core

The goal of self-organizing systems is to obtain a global behavior from local interactions. So, based on these interactions and on the information acquired through them, an element can establish its form of participation in the system and how it contributes with the global behavior. We discuss these ideas applied to WSNs as follows.

4.3.1.1 Local Information

In self-organizing systems, local decisions to actions (or reactions) may depend on the availability of information at a network element. Basically, we divided the local information set in two main classes:

- Self-dependent information. Many functions consider internal information of a node for the decision of its role or how it will respond to local interactions. It can be an internal state information, for example, the current role, or local information, such as node energy measurement, positional coordinates and sensed data. LEACH [Heinzelman et al. (2000)] is an example in which the role (cluster-head or common node) to be assumed by a node depends only on a probability function over its residual energy.

- Neighborhood acquired information. This type of information is dynamic and depends on message exchanges among neighbors and their carried data. Depending on the information acquired in a neighborhood, a node can take some reaction or change its state. For example, in the density control function (e.g., OGDC [Zhang and Hou (2005)]), a node can receive the positional information and state of its neighbors. Based on these data, it can verify if they cover its sensing area, and depending on the result, it can decide to assume an active or inactive state.

4.3.1.2 Local Interactions

In a communication network, local interactions occur among nodes in a neighborhood, i.e., among nodes which are reached directly, through message exchanges. In a wireless network, the neighbors of a node are the nodes that are in its radio communication range. Generally, these interaction messages inform the neighborhood about a node state and/or its local information in order to let them take their own decisions. Additionally, these interactions can be propagated iteratively in a hop-by-hop way through the network elements, like chain reactions. The same behavior is also seen in natural systems, such as chemical gradient reactions secreted by biological cells or propagating neural synapses in brain.

In the WSN context, messages carrying local and/or propagated information can be exchanged through mechanisms such as: Flooding, in which information is propagated for all connected nodes in an $n-$hop neighborhood without modification, for example, to inform neighbors of a node presence and its local information; Gradient, which is similar to flooding but information being propagated can be modified (e.g., accumulated) in every hop, and that can be used, for example, to establish cost functions in the network; Direct interaction, in which a direct neighbor that must receive the message is specified.

4.3.2 Service Abstraction Level

The goal of the framework abstraction level is to provide a set of common services that can be used to compose the design of higher level self-organizing activities or applications. Services in this level use the set of core primitives of the framework to abstract design particularities from higher levels. This level is envisioned to be totally extensible as experience and new implementation occur, but it focuses in the following two basic services.

4.3.2.1 Neighborhood Information Service

As described before, many local decisions taken by network nodes depend on acquired neighborhood information. Thus, a basic service that creates a list of neighbors and their acquired information through the interaction primitives can help in the design of several self-organizing functions. In fact, some proposals deal specifically with these services by implementing particular mechanisms and providing an API function set for easier sensor network programming. Some of them are Hood [Whitehouse et al. (2004)] and Abstract Regions [Welsh and Mainland (2004)], which abstract neighborhood information exchange. Thus, the implemented services or their applied mechanism can be used for new designs.

4.3.2.2 Role Assignment Service

Many self-organizing activities are associated with special roles which are assigned to network elements in order to establish a specific behavior. For example, the clustering function can be seen as a role assignment problem where the roles can be cluster-head or common node. In the same form, the density control function can be seen as the attribution of active or inactive roles to the nodes in order to accomplish a system goal, in this case, efficient area coverage. In a general way, the role assignment service is comprised of methods for role specification, by defining the different roles that can be assumed by a node; and role conditions, by describing the conditions (local or acquired neighborhood information) in which a role is established. The Generic Role-Assignment proposal [Frank and Römer (2005)] is a clear example in the WSN domain following these characteristics. Again, using an implemented service or its applied mechanism can helpful for new designs.

4.4 A Design Guideline

In order to assist system designers in the development of self-organizing functions for WSNs, we propose a guideline which directs the search of these self-organizing solutions by exploring the framework insights described before. The guideline starts from the definition of the desired global system behavior, and it goes through several steps to produce the local behavior of the network elements to fulfill the system requirements. These steps are shown in Fig. 4.2 and they are detailed in the following.

Its is important to stand out that this guideline will not generate solutions to be directly applied to the WSNs, and the experience and knowledge of the designer is fundamental for the generation of correct and efficient solutions.

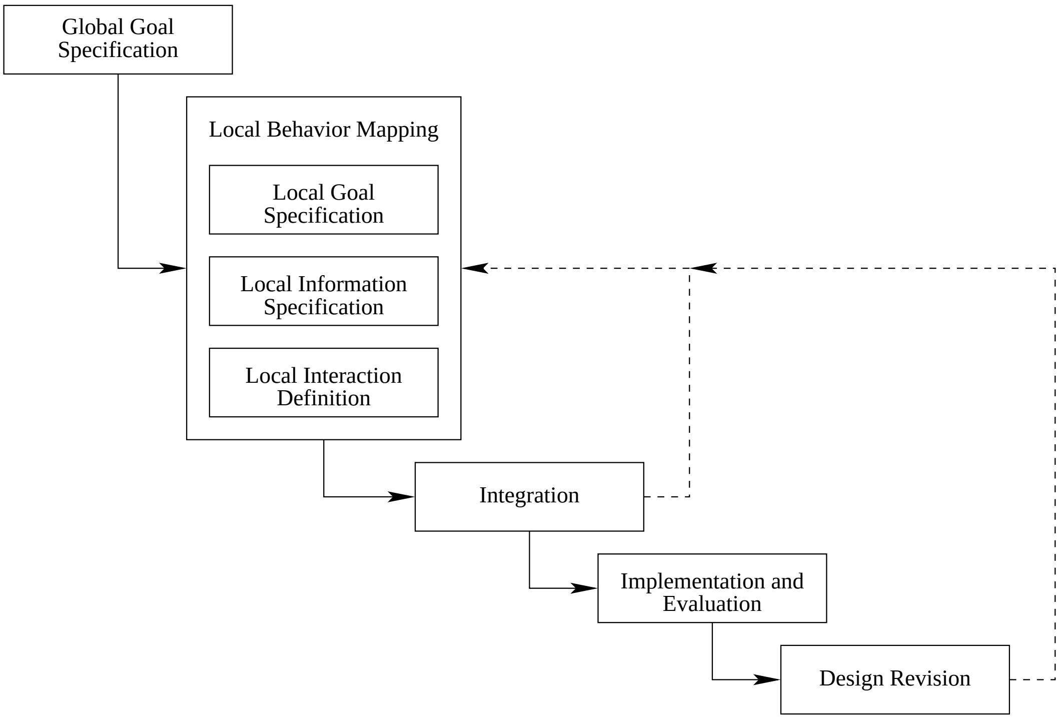

Global Goal Specification
Local Behavior Mapping
Local Goal Specification
Local Information Specification
Local Interaction Definition
Integration
Implementation and Evaluation
Design Revision

Figure 4.2: Design Guideline.

4.4.1 Global Goal Specification

The first step is the description of the global behavior to be obtained by the network. The more details can be provided in this phase the better the specification and characteristics to be considered in the subsequent steps.

As global behavior examples we can cite the formation of a tree structure connecting the network nodes from the sink, which is a very common applied infra-structure in WSNs, or a density control function, in which the network nodes form a subset of active nodes to perform the sensing task and maintain the network coverage by considering sensing ranges for the nodes.

Important details that can be surveyed for these examples could be: if their organization must be maintained pro-actively or only on-demand, in case of adaptation due to some specified situation as failures; or still if their organization must change along the network lifetime to share the resource consumption among all network nodes, in this case by requiring the introduction of a random factor in the organization process or the usage of memory and historical behavior.

4.4.2 Local Behavior Mapping

The main problem in the self-organizing design is the process of translating a desired global behavior in local interaction rules among its elements. And once they are established, not necessarily the global goal is achieved with its all performance requirements.

An important aspect in this intend is the use of nature-inspired solutions, in which a desired self-organizing behavior found in natural systems can be mapped to the design of computational systems. This nature-inspired engineering process is a promising technique to achieve intelligent autonomic systems and some examples are under the term *Swarm Intelligence* [Bonabeau et al. (1999)].

By considering a design "from scratch", the common characteristics found in the proposed framework of the last subsection may be very helpful. For example, if possible, the developer can think the self-organizing function under treatment as a role assignment problem, and then follow the described ideas to guide the design process, or even use some service already implemented. If the function can not be dealt this way, the developer can still try to use some known or available information mechanisms. In summary, the existing knowledge implicit in these services can help the design task.

Anyway, in order to better guide the design process or turn it more comprehensive, this "local behavior mapping step" is divided in the following sub-steps:

4.4.2.1 Local Goal Specification

The first step on the attempt for global to local behavior mapping is the specification of the local goals, i.e., the goals for each network element. Basically, a local goal relates the decision to be taken by an element with the conditions observed by it (from itself and/or from neighborhood). Besides, this decision can result in adjusting a parameter set, the definition of a state or the roles to be played by an element.

Following the examples cited in the first step, we can describe the local goal of an element as: in the tree organization case, setting a neighbor node as parent; or in case of the density control function, deciding for the inactive role if it is covered by active neighbors or active if otherwise. Again, some details can be important to generate solutions in compliance with the global system requirements. For example, if the distribution of energy among the nodes is important, the local goal in the tree organization could be to choose a neighbor node with the highest residual energy as parent, and in the density control case, to decrease the chance to be active according to the decrease of the energy level.

4.4.2.2 Local Information Specification

To support local decision and accomplish the local goal, all local information necessary must be specified. As described in the conceptual framework, these data can be self-dependent (internal information of a network element) or acquired from the neighborhood. The specification of the needed information will help in the definition of the interaction mechanisms (next step) to accomplish the local goals (previous step).

As an example, in the tree organization case, a node must know about the existence of neighbors and some metric about them (such as residual energy, position or number of hops to sink) for the choice of a parent. In the density control example, a node must acquire the position, state (if active or not) and the sensing range of the neighbors, and its own residual energy to check if its area is covered and choose its own state.

4.4.2.3 Local Interaction Definition

An interaction mechanism must be defined when neighborhood information is needed in a self-organizing function in order to a node take a decision and accomplish its local goal. This mechanism will establish the information which must be exchanged and how it is done. Due to restrict resources of WSNs, this interaction must be very objective and efficient, and its definition consists of a critical phase in the design task. Besides, it is the moment in which the designer's experience is most important.

Basically, local interaction mechanisms are comprised of:

- The definition of the messages and the information carried by them.

- How these messages are exchanged among the neighborhood. In this case, the temporal aspect of the interactions and/or their conditions is important (e.g. interact periodically, or only as a response to other interactions or detected condition).

- The establishment of the relation between the local interactions and local decisions. In other words, it is necessary to determine the moment of the interaction process the local decisions are taken.

All these sub-steps comprise an information service, as described in the conceptual framework. Thus, at this moment, the definition of a generic mechanism which will benefit other self-organizing functions or the consideration of an existing one can be useful.

As an example, in the tree infrastructure case of the last step, a common interaction mechanism found in the literature (as in Zhou and Krishnamachari (2003) and Figueiredo et al. (2004b)) defines a local interaction mechanism as a flooding started from the sink node in which each node broadcasts the building message with its own id, position, residual energy, and the number of hops obtained from the received messages incremented by one. Regarding the parent decision, it can be done during the flooding process after receiving some neighbor broadcasts (controlled by a random timer, for example) and before sending its own broadcast and continuing the process. By this way, the parent choice can be restricted, but this process permits the construction of the tree structure at once and avoids the formation of loops.

Following the density control example, local interactions are necessary to inform a node about the neighbor states. This could be done through broadcasts by the nodes in rounds. However, the nodes need information from each other to take a local decision, and so this decision can not be performed independently at the same time. Thus, a scheme such as choosing some random nodes to start local interactions in the active state, and proceeding with these interactions in a similar way but using the already propagated information for local decision must be introduced. This is exactly the idea present in the OGDC [Wang et al. (2003)] solution.

From these two examples, we can see that the relation between local interactions and local decisions is very important to obtain correct and efficient self-organizing functions.

4.4.3 Consideration of Integration Possibilities

As described in Chapter 3, many self-organizing functions are expected to be applied in WSNs. This fact is clear in several communication protocol layers such as MAC and routing, but other functions must also be considered for efficient WSNs, such as density control and clustering. As a consequence, integration aspects of different functions must be considered in order to achieve a correct as well as an efficient operation of these networks. Thus, this step is fundamental in the design of self-organizing WSNs.

As a clear example of the need of integration, we can consider the routing and density control functions. With a density control algorithm running independently, additional dynamics are introduced to the network topology alternating redundant nodes to active or inactive states to save energy. Thus, some previously established routes by a routing algorithm can be destroyed by the density control algorithm, and this will lead to data packet losses until the route recovery.

In this guideline, we present two general approaches that can guide the design of integrated self-organizing functions in WSNs, and they are described as follows.

4.4.3.1 Synchronization Approach

A very simple solution consists of a simple synchronization of the considered functions. In this case, the functions must execute in a given order satisfying their dependencies for a proper operation of the network. The advantage of this approach is that the self-organizing functions are considered with the minimum of modification, i.e., their deployment can be done as they were independently proposed.

For the previous routing and density control integration example, the routing infrastructure formation could be scheduled to execute just after the density control organization, so routes are reestablished after the topology alteration by the density control.

Obviously, such synchronization approach is only possible when the functions occur in well-defined periods and have a well-defined duration of execution. However, many organizational aspects of WSNs are treated in rounds, as the examples of LEACH [Heinzelman et al. (2000)] in clustering and OGDC [Zhang and Hou (2005)] in density control.

Even with the execution in rounds, this synchronization can be hard to implement. Due to distributed character of the self-organizing functions, their execution duration may be variable and cannot be determined without having a global view of the network. An alternative is the use of extra mechanisms for providing this support, which can increase the solution complexity, or a synchronization point can be estimated (or super-estimated to avoid problems), but in this case the network performance can be impacted with longer organization intervals. As a solution, a more aggressive approach for integration of self-organizing functions is discussed in the following.

4.4.3.2 Integrated Design Approach

The second approach consists of an attempt to integrate the design of several self-organizing functions. Generically, this approach optimizes different functions by unifying their goals and utilizing common information and interaction mechanisms. Ideally, when all these unifications can be done, we obtain a fully integrated solution in which several self-organizing functions can be considered as a single one.

The advantage of this approach is that it can lead to more efficient solutions which meet the design requirements of constrained WSNs. By unifying the local interaction mechanisms and the exchanged information among the network nodes for different functions we have a great reduction of resource consumption with less message exchange. Besides, in opposition to the last described approach, it also allows a more efficient interaction among the self-organizing functions, since they can be designed to work concurrently in perfect synchronization and without extra costs.

Regarding the same routing and density control example described before, the negotiation messages to inform node states (active or inactive) and position for coverage verification can flow from the sink to the network edges and be used directly to define a routing tree for data forwarding by considering only the active nodes. This example is better explored in Siqueira et al. (2006a), and it is presented in more details as part of this work in Chapter 7.

Of course, this approach turns the design process more complex by requiring from the designer the identification of integration opportunities. Besides, it is not possible to be used in all the cases. For example, in some localization functions some special negotiations using special elements (such as beacons) are necessary for the establishment of a global coordinate system, and these elements do not participate of others self-organizing functions. Thus, this function cannot be considered to be fully integrated with others and the synchronization approach would be complementary applied.

The issue of integrated design is already applied in WSNs under the term “cross-layer design”. Many proposals exist but mainly regarding aspects of communication protocols, and particularly regarding to MAC and routing protocols, as in Ding et al. (2003); Zorzi (2004); Sichitiu (2004); Cetintemel et al. (2003). Additionally, not necessarily the functions are fully integrated as one solution, but they can only consider a knowledge between layers. In our approach we discuss the need of such integration in a wider view considering self-organizing functions not only related to the network stack, and a fully integration is possible and can bring benefits.

In fact, integration aspects can be considered not only after the previous steps of the methodology, but also during the design of a new function, diluted in the previous steps. This can be done by considering previously designed functions and their characteristics that can be used by the new one, and again, requesting experience from the designer.

4.4.4 Implementation and Evaluation

After the design of the local behavior of the network elements, it is necessary to verify if the global behavior is achieved, and of course, the network must be put in operation. Thus, the system must me implemented to be evaluated.

A faster way to verify if the design will perform properly is through simulations. Simulations permit the test of specific and different conditions and its results can be used for design revision (as described in the next step) without the costs of a real implementation and experimentation, which can be performed only when the design is considered mature.

4.4.5 Design Revision

Due to the difficulties of mapping a global goal to a local behavior, many attempts can be necessary until a global goal is achieved considering a desired performance. Thus, the result of the last step can be used in a possible redesign in order to improve or correct the system performance.

This consequent cycle in the design task can be specially useful in the design of nature-inspired solutions or in the integration of different self-organizing functions. In the former case, nature and computer needs and requirements may not match very well, so repetitive design cycles may be necessary to adequate the nature-inspired solution. A similar idea is introduced in Zambonelli et al. (2005). In the latter case, previously designed functions can be reviewed to accommodate the integration of a new one.

4.5 Case study: Self-Organized Spatial Patterns for Sampled Sensing

In this section, we present a case study that forms self-organizing spatial patterns for sampled sensing. The idea is to activate only a subset of nodes for sensing, because not all the node information can be necessary for a given application in a given time. Also, the active nodes form a spatial pattern in order to provide a representative sample of the network state to the application. This view can be complementary to proposals that deal with density control for energy-efficient operation.

In fact, the design issues presented in this work are hard to evaluate and validate. But we believe that the case study following the proposed design guideline as an example can show its usefulness.

4.5.1 Global Goal Specification: Spatial Patterns

The desired global goal to be achieved by the network elements is a subset of active nodes following some spatial pattern. To be more specific, we consider two spatial patterns in this case study:

- Border Activation. This pattern can be useful in intrusion detection applications, in which only the border nodes can be sending data to inform if the network area is being invaded.

- Concentric Activation. Nodes are activated from concentric distances from the sink. With such a pattern, the sink may have a representative vision of the network without using all the network nodes. This pattern can also be dynamic through several steps, such as a radar monitoring, so the sensing load is distributed among the nodes and the collected information can complement that obtained from the last step.

4.5.2 Local Behavior Mapping

As described in this work, mapping a global goal to a local behavior is very challenging and there is not formal methods to do so. Also, the designer experience is important. From the framework insights (Section 4.3), which were discussed from the observation of real self-organizing function implementations, the spatial pattern of active nodes may be considered as a role-assignment problem. In this case, the roles to be achieved by the nodes are active or inactive sensing, but there is still the difficulty of determining the conditions that leads to the desired behavior.

Clearly, the information of the neighborhood position and state is necessary for the establishment of the role of an element. How to provide and how to use these informations is the goal of the subsequent steps. However, by observing self-organizing systems in nature, we see that a lot of them present an emergent spatial pattern formation (e.g. propagation of chemical reactions or skin pigmentation of animals [Camazine et al. (2001)]). In addition, we can found many works that propose Cellular Automata to model such self-organizing systems [Wolfram (1986)]. Thus, the behavior of a Cellular Automata inspired the construction of this case study.

4.5.2.1 Local Goal Specification

The goal of every node is to stay in an active sensing state if a given condition is verified. The applied rules that determine this condition for both considered examples are:

- Border Activation. This is a very simple and intuitive case. A node will stay active and send data to the sink only if it does not receive information of one of its immediate nodes after the information process. If there is some direction without node interactions, this means that the node is in a border.

- Concentric Activation. This case is not so intuitive, and we take the observed game-of-life behavior for cellular automata [Wolfram (1986)], which is similar to the desired organization by creating a propagation of state scheme from a node with a given fixed state, as the basic solution. In this case, the sink has the fixed state of 1, meaning active for sensing. So, if a node is in state 0 (inactive for sensing) and after the information process it perceives any neighbor in state 1, it also turns to the state 1. Otherwise, if the node is in state 1, after the information process it will return to the state 0.

4.5.2.2 Local Information Specification

As introduced before, for local decision on the role to be played by a node, the information of the neighbor nodes position and state is necessary for the establishment of the role of an element (neighborhood acquired information). This information can be maintained either in cache for the received messages, or in specific local variable that stores only the relevant information. By following the Cellular Automata Model, we can compose a grid that indicates the positional states of the neighborhood. Thus, we consider eight equal cells that store the data of the nodes in any of the directions (vertical, horizontal and two diagonals) from a given node. Particularly, in the case of concentric activation, the earlier state of the node is also necessary to define the next (self-dependent information).

4.5.2.3 Local Interaction Definition

After the latest steps, some mechanism must be implemented to perform information exchange among the nodes. In this moment, some existing information service might be useful, but we build a simple one following the cellular automata model as follows:

- Message definition. We simply put all the information described in the last step (current state and location of a node) in a broadcast message to inform all neighbor nodes.

- Message exchange. A cellular automata model works in rounds in which the state to be played in one round depends on the existing information of the last one. Also, we must accommodate the network dynamics (topology changes due to failures, node inclusion, battery depletion etc.), and change the network state along its lifetime. Thus, we define rounds in which every node sends the specified information message at the beginning.

- Local Decision. The decision of the roles of the nodes is done by applying the specified rules just after the reception of all the neighborhood information at the beginning of a round (which can be controlled by a timeout). Thus, this new state will last during the following round and will be informed at for the next.

4.5.3 Integration

The fundamental goal of a WSN is to collect information from the environment and to deliver it to a network observer through a sink node. A very common infra-structure for data collecting in WSNs consists of a tree-routing organization from the sink [Heidemann et al. (2003); Sohrabi et al. (2000); Woo et al. (2003)], in which each node determines its parent for data forwarding towards the sink.

A property of such a tree-building process is that the sink starts it by sending control messages in broadcasts, and every node in the network will propagate its own control message after its parent definition. Thus, we can use this common tree scheme as a basic information service to feed the neighborhood of a node with all the specified information. The advantage of this approach is the integration of a necessary infrastructure organization, in case the routing tree, with the new self-organizing function being developed in a single process (integrated design approach). This integration simplifies and permits resource savings in the organization functions by unifying control messages for tree construction and information exchange.

Usually, the tree building process is periodically repeated to support eventual network variations, so it matches with the round organization mechanism described before. Inner this process, a node decides its role only after the reception of all the neighborhood information, and this is easily implemented given a timer to execute the local decision after the reception of the first neighbor message. And this timer must be configured to execute after a period large enough to receive all neighborhood information. An additional fact is that this process is started by the sink, and then it is easier to maintain all the nodes synchronized through the rounds.

4.5.4 Implementation and Evaluation

The next step of the guideline consists of putting the designed functions in operation. In this case study we perform this step in a simulation environment, which turns easier and faster the validation of the design and permits rapid re-design activities in case of the need of adjustments.

The simulator used was the ns-2 Network Simulator [NS-2 (2004)]. The simulation parameters were based on the Mica2 Sensor Node [Crossbow Technology Inc. (2004)]: transmission power of 45.0 mW, reception and idle power of 24.0 mW, bandwidth of 19200 bits/s, and a communication radius of 40 m. As this platform uses a CSMA/CA like MAC layer protocol, we used the IEEE 802.11 implementation available in ns-2.

As the Mica2 nodes can work with a temperature sensor, the application chosen to be simulated was the temperature monitoring of a square area. In this application, the Mica2 nodes sense continuously and periodically report their results to the sink node if they are in the active state. The periodicity of report was set to 10 s in all cases, all the nodes beginning their sensing activities in random times between 0 and 10 s. The data packets sent have 32 bytes. For the tree-based information process, we considered control packets of 32 bytes and rounds of 100 s.

As a case study, we developed both examples considering a network of 100 nodes in a grid disposition (to simplify the application of rules considering location information), and the sink near the center. The nodes are 20m distant in the vertical and horizontal from each other, so nodes can communicate with 8 neighbors considering the four diagonal nodes.

The result of the *border activation* rule can be seen in Fig 4.3, and only the border nodes (filled nodes) send data to the sink through the routing tree.

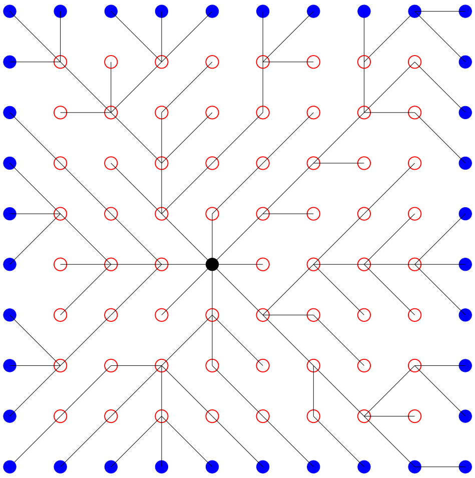


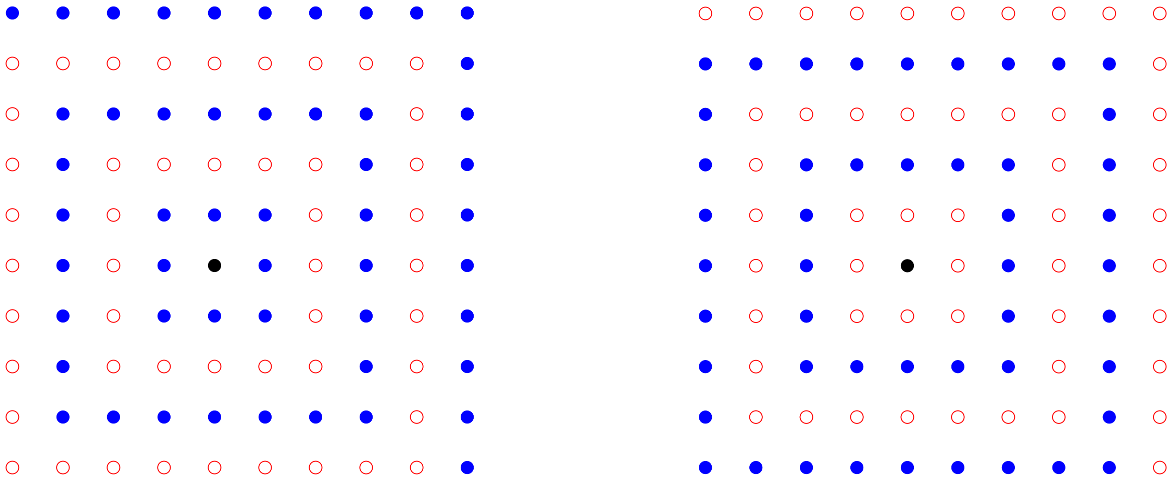
Figure 4.3: Border activation.

By applying the *concentric activation* rule, the patterns shown in Fig. 4.4 are formed and switched in every round (the tree formation was omitted in the figure). So, in every round sensing samples are taken from different distances from the sink. This dynamic activation distributes the data collection and consequently the resource consumption among the nodes.

In both cases, the amount of data in the network is reduced (only 36% of the nodes in the border activation and approximately 50% in the concentric sampling), which allows resource savings if the application does not need data from all the network nodes.

4.5.5 Design Revision

This case study was developed focused on simplicity, and the simulated scenarios were defined to turn its evaluation easier. However, in real application scenarios, we may find



(a) Instant n.

(b) Instant n+1.

Figure 4.4: Dynamic activation.

significant differences that can lead to a design revision. For example, the disposition of the nodes can not follow a regular grid, which is very difficult in practice, and a random disposition require different forms to deal with neighborhood state information related with positional characteristics. Thus, a design review can be necessary to deal with these differences, and particular issues of the presented case study will be explored as future work.

4.6 Chapter Remarks

This chapter presented important insights and practical characteristics of the application of the self-organization concept to WSNs focused on the design process. Besides some general aspects, we proposed a general design methodology that unites common characteristics found in self-organizing functions, and define a guideline that can guide the design process of new ones. Although the validation of the proposed methodology is difficult to obtain, it represents an original work as a complement to the exiting proposals in literature, and we believe that it can be helpful for guiding new designs. Additionally, a case study was discussed following such design aspects.

As future work we intend to complement the framework with other self-organizing mechanisms. Particularly, many aspects and the case study are based on a pro-active process, so we plan to explore reactive schemes which can be applied to on-demand self-organization cases such as tracking applications. Also, we intend to better explore the presented case study as an individual contribution, because WSNs have a spatial characteristic and spatial patterns can be useful in many applications.

Chapter 5

A Management Scheme for Self-Organizing WSNs: Acting over local rules

As described in Chapter 4, self-organization is an important concept being applied to WSNs to achieve robust autonomous operation, and the observed design aspects lead to solutions in which the network elements interact only among themselves, in a completely decentralized way, to accomplish a function for the proper network operation. However, application and management goals can change along the time, and a scheme for controlling these element interactions to adjust or change the achieved global behavior to a new requirement has not been explicitly addressed yet. This chapter presents a general scheme for managing this kind of network by focusing it in the self-organization characteristics. The proposed scheme is composed by a central idea, which consists of acting over local rules guiding the element interactions according to required goals and/or QoS metrics, and a methodology, which defines a generic procedure and relates important aspects for the design of the management solutions. We demonstrate the applicability of our scheme by presenting some case studies.

This chapter is organized as follows. Section 5.1 introduces the problem of adjusting self-organizing networks to different goals, the existing approaches, and the proposed solution. In Section 5.2 the proposed management scheme is presented. Next, in Sections 5.3 and 5.4, some practical case studies showing the applicability of the proposed scheme are presented. Section 5.5 briefly points to related work and their relation with the proposed scheme. Finally, the concluding remarks and possible future work are discussed in Section 5.6.

5.1 Introduction

In a practical way, a WSN serves to a purpose, it has clients (e.g., user, application or administrator), and the changing of global goals, QoS requirements, or global feeling of the network performance (general network conditions not necessarily regarding to each individual) by these clients can demand management actions from external entities, which is characterized as a "top-down" approach.

Traditional approaches for network management are based on centralized control over the system individuals by the management entities. Such an approach may be unfeasible to large scale and dynamic systems like the WSNs, mainly for human administration, because this central control increases the complexity of the processing performed by the management entity, and of its communication when interacting with all the network elements. Even though the emergence of the concept of Autonomic Computing [Kephart and Chess (2003)] and advances in the self-management of ad hoc [Shen et al. (2003)] and sensor [Ruiz et al. (2005)] networks have been introduced, these solutions maintain some self-organizing functions and apply other management functions using centralized autonomic managers. Also, it is not properly addressed how to adjust the self-organizing functions themselves according to changes of the application goals and requirements. On the other hand, by following the self-organization paradigm, which consists of a "bottom-up" approach, no individual in the network has the global vision of the network and an authority to define a different behavior to the other individuals to accomplish a different goal. Thus, a complementary approach is needed.

In this chapter, it is presented a general management scheme that consists of adjusting the global network behavior by acting over the local rules governing the self-organizing functions. This is done by central management entities according to required system goals and/or QoS metrics, and a global perception of the network. A methodology that defines basic procedures and relates important aspects for the design of such management solutions for WSNs is also provided. Additionally, the applicability of the proposed scheme is demonstrated by presenting some interesting case studies.

The advantages of the proposed scheme are: 1) It permits the application of self-organizing functions in lower operational levels with all their particular advantages, which considers only local characteristics. 2) It permits the changing of the system behavior when necessary by observing global goals and perceptions of the network, and it is not focused on individual monitoring and control of elements. 3) It turns explicit important aspects of the design of management solutions for self-organizing WSNs, which can guide other developments. 4) It presents a complementary view to the existing solutions in literature for the management of self-organizing networks, and which also advances in the relation between self-organization and self-management concepts.

5.2 A Scheme for the Management of Self-Organizing WSNs

In this section, the proposed scheme for managing self-organizing networks is described. First, it presents a general model of the scheme that introduces its main idea. Next, it presents the steps of the methodological approach for the design of management solutions.

5.2.1 General Model

As described before, a WSN may have to accomplish to different goals established by external entities like applications or administrators, and it also may have to adjust itself to different global network perceptions during its lifetime. By following the self-organization paradigm, the functions of this network are performed in a decentralized way only through the local interaction among the elements, but no individual in the network is aware of the external needs nor has the global vision of the network to define a different behavior.

In this context, it is introduced an intermediary view in which basic operational functions of the network are performed in a self-organizing way in a lower level, and centralized management entities (internal or external to network, and autonomous or not) control the local behavior played by all the individuals in order to satisfy different global goals. The proposed scheme is depicted in Fig. 5.1. Basically, it is represented at the bottom part the self-organizing functions that must be performed for the proper network operation and their correspondent local rules and parameters. The top represents the management entity that monitors the external needs and the network condition, and which may decide to change its self-organizing behavior.

For the design of management solutions following the proposed approach, it is also provided a methodology with generic procedures that relates important aspects. This methodology is presented in the following subsections.

5.2.2 Mapping Global Goals and Local Rules

There are several self-organizing functions that can be applied together for the proper network operation. Some examples are routing, clustering and density control (see Chapter 3). For each function there are several proposals in the literature that deal

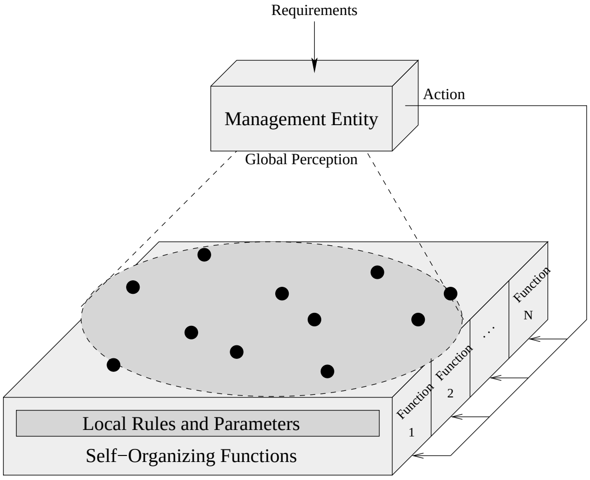


Figure 5.1: General model of the management scheme.

with particular solutions for different scenarios or requirements. Each solution is composed by its own set of local rules, which guide the local interactions to achieve the desired organization, and their parameters. As an example, routing rules could have a parameter specifying the routes validity and refresh rate, and a density control rule could specify the considered sensing range of the nodes which impacts on the achieved density for coverage maintenance.

When the network goals change, the parameters of the applied local rules or even the rules must be changed to achieve this new desired behavior. Thus, an important procedure in the design of management solutions is the relation of the different global goals of the network with the different local rules and parameters that can achieve the desired global behavior. In fact, there is no formal method for mapping local rules with global behaviors. This still is an opened research area of self-organizing systems. Thus, the experience and knowledge of the developer is important.

5.2.3 Defining the Changing Mechanism

Once different goals and rules are defined, they must be available for the action of the management entity. In this procedure, two important aspects are identified: turning different rule set available, and interaction mechanism for changing among them.

This work considers two basic approaches to turn rule sets available: Through **predefined rule set** or through **programmable rules**. The first case considers that there is a pre-defined set of rules embedded in the node to be set by the management entity. Thus, simple attribute-value messages sent to the network nodes can define which rules must be applied and set their required parameters. The advantage of this approach is the simplicity of the solution and the way that management entities interact with the network, but it is also very restrictive by considering that the appropriate alternatives are already known by the network designers. In the second case, programmable rules, the system changing is possible by reprogramming the nodes. This reprogramming is possible through dynamic code loading, in which the node system and so the implementation of local rules can be replaced (e.g., the in-network programming for the Mica2 platform [CrossBow Technology Inc. (2004)] and other works in Wang et al. (2006)), or through more flexible script-based solutions (e.g., TinyScript and Maté [Levis and Culler (2002)] virtual machine for Mica2, and Sensorware platform [Boulis et al. (2003)]), in which only scripts defining the rules are replaced and which represents a more efficient alternative. The advantage of the reprogramming approach is the possibility of rule redefinition or even implementation of new ones not planned before. However, it increases the system complexity by requiring higher management messages, because this messages must contain the coding of the new behavior, or by requiring interpreters to execute scripts.

Regarding the interaction mechanisms for changing the local rule set and parameters of the nodes, they usually are based on dissemination strategies. Basically, dissemination consists of sending the desired local rule configuration (parameter assignment case) or the new local rules (programmable rules case) from the management entity to all managed nodes. Some strategies are identified as follows.

A flooding scheme is a simple solution to reach all nodes, but it may add a significant overhead for constrained networks such as the WSNs. In these cases, other proposals considering energy-efficient dissemination, such as in Goussevskaia et al. (2005), are more appropriated. Robust mechanisms may also be required to keep the network in a consistent state, since a new node or a sleeping one can be (re)joined to the network with an older rule set, and solutions as Trickle [Levis et al. (2004)] for robust code update can be considered.

As an alternative solution, it is proposed the use of the local interaction mechanisms responsible for a self-organization functions for changing the network behavior. In self-organizing networks, local interactions occur through message exchanges, and many functions are implemented by propagating these interactions from special nodes, as sinks in WSNs, through the entire network. Thus, management entities can interact with these nodes to change the network behavior by embedding the rule configuration and/or their parameters in the local interaction process. By this way, when an element receives such interaction messages it adopts the rules defined by them. Obviously, this solution is only feasible with small definition messages (such as in parameter assignment case), otherwise a great overhead can be introduced in the self-organizing function.

5.2.4 Defining the Control Entity

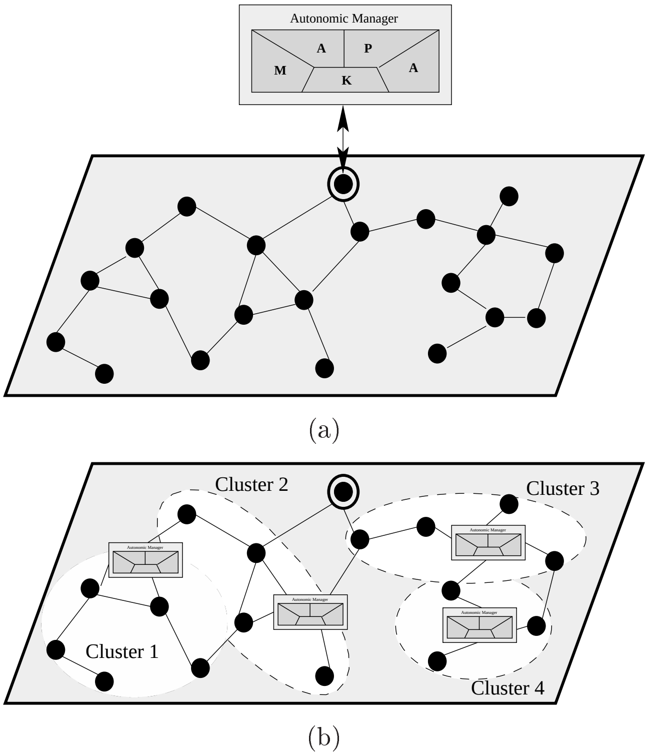
In a general way, any management entity can change directly the local rules of the nodes when desired. In order to do this, an administrator can control the network behavior through the implemented mechanisms as described before. However, more powerful solutions can be built by considering autonomic managers governed by high-level policies. Thus, in addition to the interaction mechanism with the managed elements, these management entities must monitor the global network condition (e.g., performance metrics or current state), and external requirements (e.g., administrative goals), to adjust the network to the proper behavior.

An important aspect of the employment of autonomic managers is their disposition in the network. Traditional manager/agent disposition models for the autonomic managers are feasible, and they are put in context as follows.

Global control. The simplest model consists of a single autonomic manager acting over the whole network establishing common organizational rules (see Fig. 5.2(a)). This is necessary when all the network elements must play the same set of rules, and the autonomic manager must monitor the whole network state. Examples in the WSN domain are the communication functions, in which every network element must play the same rules to form a common link and/or routing infra-structure, or the clustering function, in which a hierarchic organization is formed from the interactions among the initially common nodes. In this case, the interaction of the autonomic manager with the network can be performed by or through a special element, as the sink in the WSNs.

Partitioned control. In the second model, the autonomic managers can be established to act over self-organizing functions of subgroups (or clusters) of elements of the network (see Fig. 5.2(b)). In this case, the manager role can be played by the cluster-heads. The establishment of such hierarchical management domains is very common due to the subdivision of the network complexity, and it is the basis of the related work in Section 5.5. However, in the proposed scheme, instead of defining the individual role of every element in a cluster given a management task, an autonomic manager monitors and sets their interaction rules to achieve the desired behavior following the self-organization paradigm. In this way, different clusters can present different organizations according to their individual observed conditions. For example, the elements of a cluster can perform periodic local interaction to build a pro-active communication infra-structure if the cluster presents a high event activity, or otherwise, they can perform these interaction only if some condition is observed to build a reactive infra-structure.

Indeed, all these models can be combined in the creation of hierarchical levels treating different management and organizational functions. For example, a global



(a)

(b)

Figure 5.2: Possible alternatives for Autonomic Managers.

management entity can control the rules of all nodes to perform network clustering, and management entities from different clusters can control self-organizing functions like density control or communication according to particular observed conditions in their partitions.

5.2.5 Defining Management Policies

Policies are a well-known approach in network management that give a high-level and flexible way to realize management tasks [Sloman (1994); Westerinen et al. (2001)]. Policies can be specified as rules governing the choices on the behavior of a system, and flexibility is provided through their redefinition capacity.

Clearly, the objective and advantages of policies can be used to the development of self-organizing networks. And particularly policies have an important role in the deployment of autonomic managers. In the proposed scheme, policies must include functionalities to monitor, analyze, and act over the network. The monitoring phase uses both global data and state (network perception), as well as the requirements of external entities. The analysis phase may use techniques from data fusion and intelligent agents to detect situations where an action is needed. Some examples are accounting of network metrics, comparison with thresholds, prediction or inference techniques. And action over the network is performed through the changing mechanisms as described in Section 5.2.3.

Regarding the policy implementation, there are several proposals in the literature about languages (e.g., Damianou et al. (2001) and Dulay et al. (2001)) and frameworks (e.g., Kephart and Walsh (2004)) that can be applied to management solutions. Typically, policies are established by low-level specific rules that consider individual configuration parameters, such as action policies in the form of "IF(Condition) THEN(Action)" clauses. High-level abstractions, such as policies that consider the specification of system goals, may turn easier the administrator task, but require a more complex interpretation. Thus, these high-level policies are more feasible for management entities external to the WSNs, but they may not be for autonomic managers inside the network because common sensor node architectures are very constrained. For this last case, policies implemented like simple rules but dealing with decisions in the management level can use the implementation and dissemination (for policy redefinition) mechanisms presented in Section 5.2.3.

5.3 Case Studies On Changing to Different Global Goals

This section presents two case studies to show the feasibility of the proposed scheme. Basically, it considers a simple data collecting application, and it introduces particular instantiations of self-organizing mechanisms for the formation of a basic communication infra-structure and density control. The adopted mechanisms are focused on simplicity, necessary for several constrained WSN architectures, and they can be adjusted to different network goals according to management actions. We build the case studies following the steps described in the last section.

5.3.1 Mapping Global Goals and Local Rules

We consider two self-organizing functions for WSNs: Communication infra-structure creation and density control. They are described next:

Communication Infra-structure. A very common infra-structure for data collecting in WSNs consists of a tree-routing organization from the sink [Heidemann et al. (2003); Sohrabi et al. (2000); Woo et al. (2003)], in which each node determines its parent for data forwarding towards the sink. Basically, in this case study, the local interaction process to perform the tree construction is started by the sink through the broadcast of construction messages. This message is propagated by every node after some time waiting for parent options. These tree construction messages contain the local information of their senders for parent choosing given some metric. Following this organization process, there are several possible rules for choosing a parent, and two simple options are considered: choosing the neighbor with more energy, so the tree formation process can better distribute the energy consumption among the nodes, or choosing closer nodes, which can increase radio communication reliability and other performance metrics. In both options it is also considered the lower number of hops to the sink, which is defined by the increment of a counter in every construction message propagation after the parent definition by a node.

Density Control. Density control is a very important function of WSNs because it can diminish the network redundancy and then allow resource savings, mainly energy. Its main idea is to activate the minimum number of nodes for maintaining coverage and connectivity by considering that every node has an actuation area (sensing range). In this case study, this self-organizing function is implemented following the local interaction rules applied by the OGDC solution [Zhang and Hou (2005)] and the information mechanism described by RDC-Integrated [Siqueira et al. (2006a)], which integrates the activity decision in the tree construction process, for coverage verification and state decision by the nodes. Basically, the tree construction messages also contain the state (active or inactive) and the node locations of the senders for coverage evaluation and state decision by the nodes. Thus, when a node receives tree construction messages and decides to stay active it chooses one of the active originators as its parent and then proceeds with the tree information process. If a node is covered by its neighbors (detected through the tree construction messages already received), it decides to become inactive and it does not proceed with the tree organization process. The RDC-Integrated solution is also a specific contribution of this thesis, and it is better described and evaluated in Chapter 7.

Operation in rounds. Both organizational processes are periodically repeated to support eventual network variations, such as topological changes, link problems due to interference, traffic variations, and node energy degradation. This periodicity depends on how frequently topological changes occur. More dynamic networks need shorter rebuilding periods, and of course, it also depends on how frequent reorganization is required.

The self-organizing mechanisms described before can be associated with different global goals as described in Table 5.1.

Rule set	Parameter	Behavior	Goal
Tree Function	Parent by Energy	Parents with more energy	Better resource distribution
Tree Function	Parent by Distance	Parents are closer	Better network metrics
Density Control	Sensing range	Subset of active nodes	Density reduction and energy saving

Table 5.1: Mapping example.

5.3.2 Defining the Changing Mechanism

For changing the rules and parameters as a management function, all the considered rule options are programmed in the nodes to be set following the "pre-defined rules" model. Additionally, as all the considered organization processes start from the sink, it is adopted the scheme in which the tree construction messages themselves define the rules that will be applied by carrying the related parameters. Thus, the nodes adopt the self-organizing behavior according to the received control messages propagated by the sink, and the network organization can be adjusted in every round.

The parameters included in the tree control message are: *Tree_ Mode*, for the definition of the parent adoption rule for tree formation; *Density_ Control*, to define if the density control function will be enabled or not; and *Sensing_ Range*, to determine the considered sensing range for the nodes and then adjust the network density if the density control function is enabled.

5.3.3 Defining the Control Entity

As described before, the implemented self-organizing functions are performed through interaction messages propagated from the sink, and it is not considered the cluster formation. Thus, the model in which the management entity controls the whole network through the sink is applied. Additionally, the control entity monitors the global goals required externally to perform the changing of the network behavior.

5.3.4 Defining Management Policies

Policies can be defined to help in the decision of a management entity on what rules apply. In this sense, it is important the knowledge obtained from the first step, which is the mapping of local rules and global goals.

The case studies consider the action policy illustrated in Fig. 5.3 for the management of the communication function, and the policy of Fig. 5.4 for the management of the density control function. They show the adjustment of the network behavior according to some global goals that can be required by the application.

The communication policy considers two global goals, one to favor the energy consumption among the nodes in order to better distribute the data collection load, and then which uses the parent adoption rule by the energy, and other to increase the network performance, which uses the parent adoption rule by closer distances. The density control policy depends on the data resolution required by the application, because it alters the number of active nodes sensing the network in a given moment. Thus, it is set three levels for data resolution, the maximum level, which leads to all nodes active, the minimum data resolution, which leads to the minimum subset of active nodes and then can extend the network lifetime, and a medium level, which set the network density to a intermediary value. The behavior achieved by these policies are shown next.

```
if goal is EnergyDistribution then
  set_Tree_Mode(TMODE_ENERGY);
elseif goal is NetworkPerformance then
  set_Tree_Mode(TMODE_DISTANCE);
endif
```

Figure 5.3: Policy example for network performance.

```
if goal is MaximumDataResolution then
  set_Density_Control(OFF);
elseif goal is MediumDataResolution then
  set_Density_Control(ON);
  set_Sensing_Range(10);
elseif goal is MaximumLifetime then
  set_Density_Control(ON);
  set_Sensing_Range(20);
endif
```

Figure 5.4: Policy example for data resolution.

5.3.5 Simulation Results

In this subsection, the execution of the self-organizing mechanisms and the management policies described before are evaluated in a simulation environment. The simulation parameters and scenarios are described as follows.

Simulation Parameters. The simulator used was the ns-2 (Network Simulator 2) [NS-2 (2004)], and the simulation parameters were chosen based on the hardware of the commercially available Mica2 mote [Crossbow Technology Inc. (2004)]. Table 5.2 shows some important parameters used in the simulations. For the MAC layer, as the Mica2 nodes implement a CSMA/CA protocol, the IEEE 802.11 version available in ns-2 was used. Regarding the considered monitoring application, all the network nodes sense continuously and periodically report their results to the sink node. The periodicity of this report was set to 10s in all cases, all the nodes beginning their sensing activities in random times between 0 and 10s. The data packets sent have 32 bytes. For the tree-based organization process, it is considered control packets of 32 bytes and rounds of 100s.

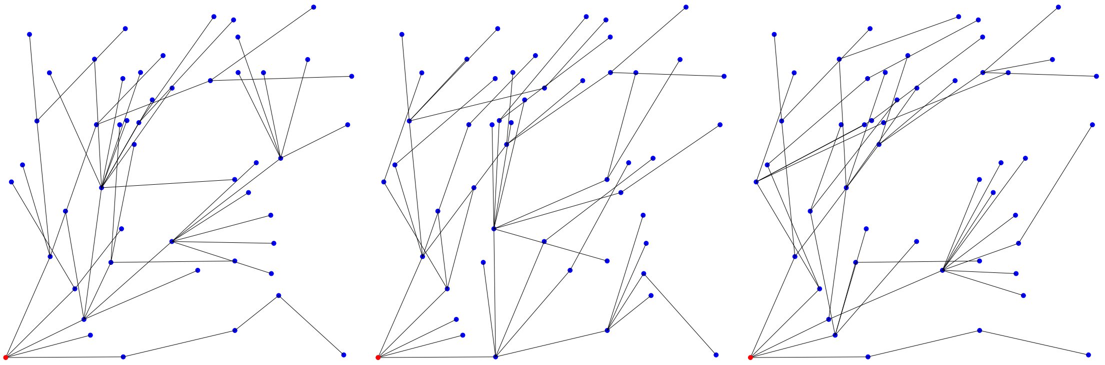
Parameter	Value
Transmission Power	45.0mW
Reception Power	24.0mW
Idle Power	24.0mW
Bandwidth	19200 bps
Communication Range	40m

Table 5.2: Mica2 simulation parameters.

Changing the Communication Rule. This scenario evaluates the consequent global behavior and its impact on the network performance according to the adoption of the different implemented communication rules by the management entity. We use a set of 50 nodes randomly distributed in an area of 100x100m^2 with the sink positioned in a corner. We set up a network with the initial goal of maximum network lifetime, which leaded the application of the parent adoption rule that considers the available energy. After 2000 s, this goal is changed to the network performance, which leaded to the change of the parent rule to the distance metric. The simulation results of this change are presented in Table 5.3, which presents them summarized after 33 runs. As we can see, although the total consumed energy difference is not significant between both cases, its variance is lower when the parent adoption rule considers the energy. This behavior happens because this tree organization tends to adopt different parents in every round and then better distribute the energy consumption among the nodes, inclusively peripheral nodes, as we can see in some samples in Fig. 5.5. When the goal is changed to the better network performance and the communication rule is changed to the distance metric, network metrics like the average packet delay between the sources and the sink, and the packet delivery ratio become better due to the creation of more directed paths to the sink, and the formation of closer links, which increases wireless communication reliability. However, it presents a higher energy variation due to the concentration of routes in some forwarding nodes, as we can see in some organization samples in Fig. 5.6.

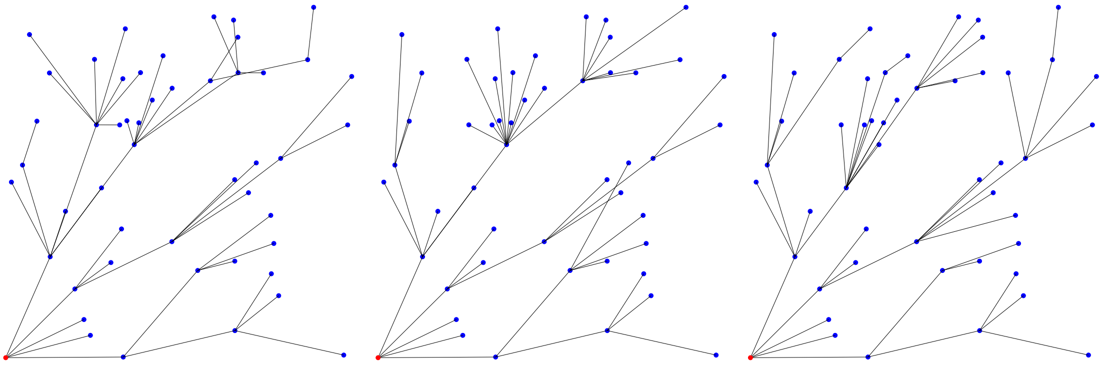
Global Goal	Energy Distribution 0 to 2000 s	Closer Links 2000 s to 4000 s
Total Energy (Joules)	2438.60	2438.25
Energy Variance (Joules)	0.20	0.29
Delay (seconds)	6.68	5.97
Delivery Ratio (%)	85.27	87.56

Table 5.3: Changing the communication rule.



(a) At 100 s. (b) At 200 s. (c) At 300 s.

Figure 5.5: Some tree samples with energy rule.



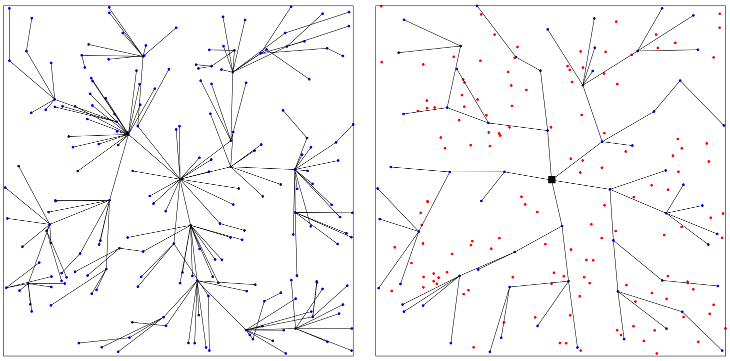
(a) At 2100 s. (b) At 2200 s. (c) At 2300 s.

Figure 5.6: Some tree samples with distance rule.

Changing the Density Control rule. This scenario evaluates the impact of adopting different density control rules by the management entity. This is done by considering the scenario described before, but a network of 100 nodes randomly distributed in an area of 200x200m^2 with the sink in its center, the maximum sensing range radius of 20m (the half of the communication radius), and activity decision timer values as shown in Zhang and Hou (2005). We set up a network with the initial goal of maximum data resolution. After 2000 s, this goal is changed to a medium data resolution, which enabled the density control function with the sensing range parameter set as the half of the maximum value, and after more 2000 s it is changed to the maximum lifetime goal, which caused the network to operate in its minimum density. This behavior corresponds to the policy of Fig. 5.4, and summarized simulation results after 33 runs are shown in table 5.4. As we can see, the density control function can be regulated to different global goals. As much as data resolution is important, more active nodes are maintained, and consequently more energy is spent. When the goal is the maximum lifetime, the minimum number of nodes remains active, which diminishes the monitoring resolution, but the network spends less energy. Fig. 5.7 illustrates some samples of network organization when the density control rule is off and when it is set with the maximum sensing range.

Global Goal	**Max. Data Resolution** **0 to 2000 s**	**Medium Data Resolution** **2000 s to 4000 s**	**Max. Lifetime** **4000 s to 6000 s**
Active Nodes	100	67	45
Total Energy (Joules)	7202.03	4420.62	2367.81

Table 5.4: Changing the density control rule.



(a) At 100 s.

(b) At 4100 s.

Figure 5.7: Some density control samples.

5.4 Case Studies On Self-Adaptation

In this section, it is presented two case studies showing that a global network perception can cause the changing of the self-organizing behavior of the network in order to achieve a better performance. This is shown in some developments regarding the routing function. The first case considers a scenario where different routing strategies are applied autonomously according to a policy definition. The second case deals with an adaptive parameter setting instead of a constant default value. In both cases, the goal is to achieve a better performance regarding energy consumption, a very limited resource in WSNs. Both cases are described in a more direct way as follows.

5.4.1 Changing the Routing Strategy

As described in Chapter 3, routing is an important function of WSNs treated under the self-organization paradigm. In the literature, there are several proposals implementing different mechanisms for routing infra-structure creation. A very common mechanism deals with the formation of a tree-routing organization in a pro-active manner, as described in the last case study. An alternative mechanism is the building of the routing infra-structure only for the nodes demanding traffic in a reactive strategy. While the first approach favors networks with constant traffic, the second approach is more appropriated for scenarios with eventual traffic because it builds the routing infra-structure only when necessary. However, there are situations where traffic variations may occur and different routing strategies may be more appropriated for different moments. This is the case of event-driven scenarios, in which the network may remain with a very low activity for days, favoring a reactive algorithm, but in a given moment a number of events may occur generating a traffic large enough to use a proactive algorithm. Thus, it is proposed a solution in which an adaptive rule can be applied by an autonomic management entity to achieve a hybrid behavior by switching between both described routing strategies according to the monitored network condition.

In this case study, both pro-active and reactive routing strategies are implemented in the nodes. The pro-active algorithm, called EF-Tree (Earliest-First Tree), is an implementation of a simple tree-based routing algorithm as described before, but the parent adoption strategy is based on the firstly tree-build message received from its neighbors. In the reactive algorithm, called SID (Source Initiated Dissemination), the routing process is triggered by source nodes only when data is available. The routing process starts with a data flooding on the network. When a data message arrives at the destination node (the sink), this node will respond with periodical requisition messages to the neighbor node that sent the first data message. And this response is recursively propagated in the same manner towards the data source node, which establishes a route and inhibits new floodings.

With these mechanisms, an autonomic agent present in the sink node can control which behavior to adopt in a given moment. In order to perform this changing, it is assumed a default reactive behavior due to the considered event-driven scenario, and this reactive behavior is suppressed by the pro-active one when the sink starts the tree building process and the nodes receive these messages, which leads to a new local rule set to be applied. The adaptive policy applied by the management entity is based on the traffic monitoring, and the number of sources sending data is counted. A threshold is used to evaluate this traffic, and if the traffic is lower than this threshold a reactive strategy is adopted, otherwise, a proactive strategy is taken. The rule that represents the decision of the management entity is shown in Fig. 5.8.

```
if Num_of_Sources > THRESHOLD then
  set_proactive_strategy();
else
  set_reactive_strategy();
endif
```

Figure 5.8: Routing strategy rule.

This case study represents a significative contribution in the WSN domain. It is better described and evaluated in Figueiredo et al. (2004b), and it is presented as a specific contribution of this thesis in Chapter 6, where we present an evolved solution.

5.4.2 Adaptive Routing Optimization

Often, routing algorithms for WSNs define different parameters that must be configured according to a given application and network situation to achieve a better performance. But if the network condition changes, a policy can detect it and set dynamically the better routing parameter. For example, in routing solutions presented in the Chapter 3, it is common the use of periodical updates to support topological changes caused, for instance, by failures. Pre-defined frequent updates can be set to a scenario with common failures, but this configuration is not appropriated to a stable scenario with a low failure rate due to the consequent communication overhead. If it is possible to determine the failure occurrence rate, a policy executed by an autonomic manager can dynamically change this routing parameter, or even define when to take the corrective action, which optimizes the routing performance.

In this case study, it is considered that a given application starts requiring continuous environmental measurements. Thus, a proactive routing strategy such as the previously described tree-routing is more adequate. Instead of setting a constant periodical rebuilding rate performed by the sink, it is proposed a solution that monitors the traffic and infers about failure occurrence to take this action only when necessary to save energy. This failure detection is possible by assuming a continuous traffic, thus, downsteps in the received traffic mean a failure possibility. The applied policy is shown in Fig. 5.9, in which the number of nodes sending data (counted as described in Rule 1) is observed in the periods of the data generation rate. A reduction of this number means a failure, which triggers a tree building to reorganize the routing infrastructure. A similar but evolved solution applying better data fusion techniques for failure inference is presented in Nakamura et al. (2005b).

```
if Num_of_Sources < old_Num then
  set_proactive_action();
endif
old_Num = Num_of_Sources;
```

Figure 5.9: Optimization rule.

5.4.3 Simulation Results

Experiments with both case studies on self-adaptive routing were performed using the same simulation parameters of the last case study. They are presented as follows.

Changing the Routing Strategy. In this evaluation, the number of sources generating data is varied along the simulation time of 4000 s, and they are randomly chosen following an uniform distribution. A policy equivalent to the routing rule of Fig. 5.8 was set to run in intervals of 10 s, the same period of data messages, and the traffic collected in an n-interval is compared with a static threshold in order to set the strategy activated in the $(n + 1)$-interval. Thus, it is expected that different thresholds lead to different results, as in Fig. 5.10, where it is shown the behavior of a network running with EF-Tree and SID algorithms alone, as well as the policy-based scheme with traffic thresholds (limit of the number of source nodes sending data to take an adaptive behavior) of 1, 2 and 3, called *PB-1*, *PB-2* and *PB-3*. This scheme autonomously adapts the routing mechanisms during the simulation, as described before. A threshold higher than 3 leads to a performance very close to SID, because proactive behavior is rarely taken with the simulated parameters, thus their results were omitted.

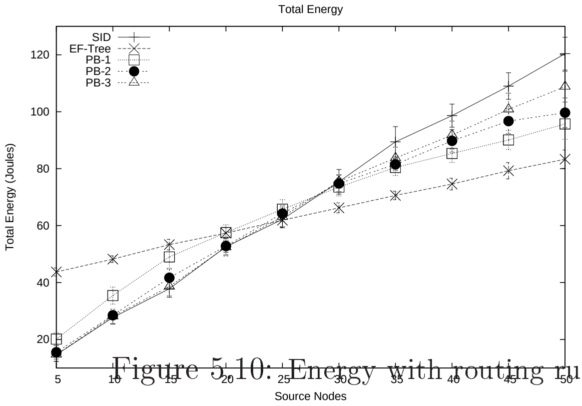


Figure 5.10: Energy with routing rule 1.

The results for PB-2 and PB-3 are closer to the SID algorithm when the number of source nodes is less than 30. When a low traffic is measured, such policies do not assume the proactive behavior. Some differences are due to the random nature of the traffic, because source node data generations can be concentrated in a time interval without implying in a high occurrence ratio. In these cases, the performance of EF-Tree and PB-1 are not so good because they assume an unnecessary proactive behavior. But, when traffic increases, the EF-Tree and PB-1 have a better performance than PB-2 and PB-3 because their proactive behaviors avoid the cost of creating a routing infrastructure for new sources. PB-1 never reaches the EF-Tree performance for situations of high traffic since it always starts with a reactive mode, before changing to a proactive behavior.

In the previous results, we can see that the chosen policy is not so precise in the traffic evaluation. But, this can be solved by the redefinition capacity of policies. For instance, a better solution regarding adaptive hybrid routing is presented as a evolution of this case study and an important contribution of this thesis in Chapter 6.

Adaptive Routing Optimization. In this case study, it is considered a continuous traffic scenario where the rule of Fig. 5.9 is applied to detect failures and rebuild the routing tree. The number of source nodes is fixed in 20, randomly chosen to send their data periodically towards the sink from the beginning to the end of the simulation. We varied the node failure ratio from 0 to 0.012 failures per second randomly distributed during the simulation. When a node fails, it stays inactive until the end of simulation. Fig. 5.11 shows the delivery rate improvement of the adaptive rule solution (EF-Tree adap) compared with the EF-Tree with fixed rebuildings (EF-Tree orig). It shows the advantage of failure detection to take an adaptive behavior. Fig. 5.12 shows the relative energy usage of the solutions. We note the better performance of the adaptive solution when the failure rate is low due to rare tree rebuildings. However, with higher failure rates, the energy usage between the adaptive and original solutions approximates because more tree rebuidings are needed by the latter to maintain the delivery rate.

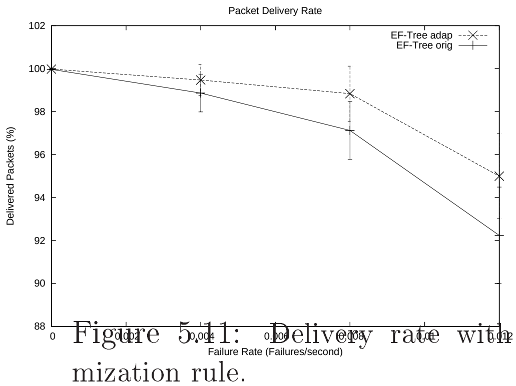


Figure 5.11: Delivery rate with optimization rule.

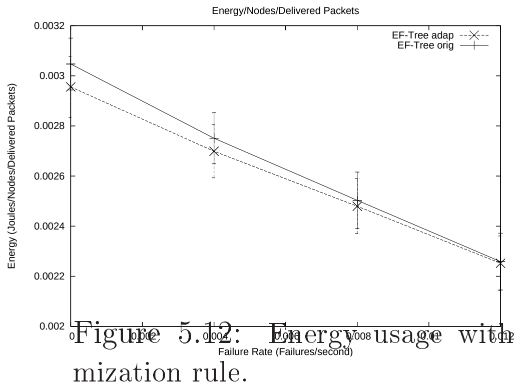


Figure 5.12: Energy usage with optimization rule.

5.5 Related Work

Two main domains of self-organized networks studied under the management point-of-view are the general ad hoc and sensor networks. Traditional approaches for the management of these networks deal with cluster-based solutions for the establishment of a cluster-head with centered responsibility over a group of managed nodes. This vision was introduced by the ANMP protocol [Chen et al. (1999)], and it was extended in proposals like the Guerrilla [Shen et al. (2003)], which proposes an adaptive grouping scheme for the establishment of nodes managed by autonomous managers governed by policies, and the MANNA architecture [Ruiz et al. (2003)], which is a pioneer work for WSNs that presents general aspects of functional, physical and information management, and includes the WSN functionalities as a third management dimension in addition to the functional area and management levels.

These proposals have much in common with the Autonomic Computing concept [Kephart and Chess (2003)], which is a technological vision for the replacement of human administrators by intelligent autonomic managers that control the network elements performing self-management. Particularly, WSN management was revisited according to the self-management framework in works like Ruiz et al. (2005). However, neither the existing proposed solutions or the autonomic computing concept properly address the relation between self-management and self-organization.

Clearly, both concepts of self-organization and self-management have much in common for obtaining autonomous computer systems. However, their relationship is not well understood, and their application together is being a topic of recent discussion [Jelasity et al. (2006)]. For example, by simply including autonomous managers in the system does not make it self-organized, because it still can realize central control over some managed elements. This is visible in the previously related management architectures in which a self-organizing function is performed to support the establishment of groups managed by autonomous managers, but other functions are still performed by the manager in a centralized way. However, self-management can be achieved through self-organization, i.e., management functions can be performed by self-organizing functions. In this last case, the advantage is that self-organization is a more powerful paradigm achieving highly adaptable, simple and lighter-weight systems only through local interactions.

In the presented context, the proposed scheme represents a complementary view that explores that both concepts of self-organization and self-management must be combined to achieve effective management solutions. By comparing this approach with the related work referred before like Guerrilla and Manna, it can reduce the complexity of the central control entities, because in these proposals the management entities still have to maintain up-to-date information for every node in a cluster, perform centralized evaluation of their state, and then emit individual commands. With the proposed scheme instead, basic network functions can be performed in a self-organizing way, and the management entities can modify the network behavior by considering only global aspects of the network operation. Also, this scheme shows how a self-organizing function like clustering itself could attend to different goals, and this is not considered in the referred works.

5.6 Chapter Remarks

In this chapter, it was presented a general scheme that allows to management entities to act over the local rules governing a self-organizing behavior. Although a central entity is introduced to perform some control over the network behavior, the scheme considers that operational functions still are accomplished following the self-organization paradigm, i.e., the management entity does not monitor and control the participation of each network individual in the realization of some organizational function. But, in a practical way, the scheme allows the changing of the network behavior to attend different goals which cannot be perceived by the individuals, for example, global network perceptions or external requirements. In the presented case studies, this need was shown when the self-organizing behavior is adjusted according to different goals required by an external management entity, or when different global perception of the network causes the changing of this self-organizing behavior in self-adaptive solutions. Particularly, all these case studies have a great potential to be explored as specific solutions for WSNs, and the improvement of the adaptive hybrid routing case is better described and evaluated as a specific contribution of this thesis in Chapter 6.

The proposed scheme presents a complementary view to the related work in literature. By adopting a fully self-organizing behavior in low-level operation, the importance and complexity of the management entity is reduced, but it still permits the control over the network when necessary. For the cases in which the presence of the management entity is more difficult (e.g. in remote and inhospitable areas), it is possible to set a default self-organizing behavior which is changed only when the interaction of the management entity is possible. This vision reinforces the complementary characteristics of self-management and self-organization concepts, and their coexistence in the operation of more efficient autonomous networks. Although this work has focused on several examples and the particular case studies on the WSN domain, i think that it can be useful in other self-organizing network domains (e.g., general ad hoc and P2P networks).

As future work we intend to better explore the presented case studies as individual contributions. We plan to improve their implementation and evaluation, and perform a quantitative comparison with other approaches present in the literature. Additionally, higher-level policies can simultaneously affect different self-organizing functions, and the construction of such solutions is also left as future work.

Chapter 6

Multi: A Hybrid Adaptive Routing Algorithm for Self-Organizing WSNs

In the previous chapters, it was discussed many self-organizing functions and design aspects to be considered in the WSN domain. Following the general design aspects discussed in Chapter 4, and the management scheme in Chapter 5, we can see that self-adaptation has an important role for controlling the system behavior in order to achieve more efficient self-organizing solutions. Particularly, this was shown in a case study (see Section 5.4) considering the routing function, which is a fundamental function in WSNs and is being considered under the self-organization concept since the first ad hoc and sensor network proposals.

This chapter explores the interesting hybrid and adaptive algorithm for routing in WSNs, called Multi, as a more practical and specific contribution of this thesis. Basically, the proposed Multi algorithm applies both reactive and proactive strategies for routing infrastructure creation in a self-organizing way, and an adaptive model to adjust its behavior autonomously in response to the variation of the network conditions. In particular, this solution is focused on dynamic scenarios like the event-driven, so it uses an event-detection estimation model to change between the strategies in order to save energy resources.

This chapter is organized as follows. Section 6.1 presents an introduction of the routing problem and Multi solution. Section 6.2 extends the discussion on routing in WSNs. Section 6.3 presents Multi, the hybrid adaptive algorithm, by describing its behavior, reactive and proactive components, and the applied adaptive model. Section 6.4 evaluates Multi by comparing it to its individual reactive and proactive components, and other proposals in literature. Section 6.5 presents a deeper discussion comparing Multi with other similar proposals. Finally, Section 6.6 presents the final considerations and discusses some future directions.

6.1 Introduction

The main objective of WSNs is to collect and process data from the environment and send it to be further processed and evaluated by an external entity connected to a sink node (or gateway). Consequently, routing towards the sink node is a fundamental function and different algorithms have been proposed [Al-Karaki and Kamal (2004)], as described before in Section 3.4.

Different applications and scenarios demand algorithms with different features. Thus, given a specific scenario, the WSN can be designed to operate with the most appropriated routing algorithm, which can be defined *a priori*. However, in some cases the variations of these scenarios can be constant or even unpredictable. For example, application requirements or traffic conditions may change along time, which favor different algorithms in different moments. But it might be unfeasible or undesirable to an external entity to act dynamically on the network to change its behavior, so the network must work in this way based on autonomic principles.

Hybrid adaptive approaches for routing are interesting solutions to deal with variable scenarios, and some protocols have been proposed for ad hoc networks (MANETs), such as ZRP [Haas and Pearlman (1998)] and SHARP [Ramasubramanian et al. (2003)]. However, they are not suitable for WSNs due to fundamental differences in the network, applications and traffic characteristics.

This work describes a new hybrid and adaptive algorithm for routing in WSNs, called Multi, that adapts its behavior autonomously in response to the variation of the network conditions. Its contribution concentrates on how to build this adaptive hybrid algorithm using traditional reactive and proactive routing strategies for WSNs, and how to build adaptive rules based on specific traffic characteristics of this kind of network. In particular, a special focus on event-driven scenarios is given, where it is proposed an event-detection estimation model that captures the seasonal and dynamic characteristics of events to change between reactive and proactive strategies in order to save energy resources. The advantages of the proposed solution are shown through simulations, in which comparisons with its independent reactive and proactive algorithms and with other well-known proposals in literature show improvements on energy consumption. To the best of my knowledge, there is not similar approach in the WSN domain.

6.2 Routing in WSNs

Routing in WSNs differs from traditional networks in many aspects. Essentially, energy efficiency is the main aspect considered in these networks due to its very constrained resources. Additionally, the basic goal of a WSN is to collect and process data from the environment and send it to be further processed and evaluated by an external entity connected to a sink node (or gateway). Consequently, routing (also called data dissemination) towards the sink node is a fundamental function and different algorithms have been proposed [Al-Karaki and Kamal (2004)], each one of them being more suitable for a different case or scenario due to their different features.

Basically, there are the following classes of protocols for WSNs:

Flooding and Gossiping [Hedetniemi and Liestman (1988)]. These are classical mechanisms to forward data in sensor networks that do not need to maintain any routing infrastructure or topology. In the flooding algorithm, every data is sent by broadcasting it to all its neighbors until it reaches the destination. Gossiping differs from flooding by choosing random nodes to forward the data. Although this approach disables the cost of route's creation and maintenance, it causes the data packet implosion problem, which represents an excessive cost for WSNs.

Proactive protocols. In this class, the routing infrastructure is created and constantly maintained no matter the network behavior. In general, this process is performed by the destination nodes. Examples of this approach are DSDV [Perkins and Bhagwat (1994)] for MANETs and various tree-based protocol for WSNs (e.g., One-Phase Pull Diffusion [Heidemann et al. (2003)] and some implementations by Woo et al. (2003)). This approach can result in an improved routing, but it has the disadvantage of a constant resource consumption.

Reactive protocols. In this class, the routing infrastructure is built only when a node wants to transmit a packet. AODV [Perkins et al. (2003)] is a well-known protocol for MANETs and Push Diffusion [Heidemann et al. (2003)] is an example for WSNs with such behavior. This approach saves resources in inactivity periods, but it has the overhead of path discovery for each originator node.

Another class of routing algorithms consist in the design of hybrid adaptive algorithms, which apply both reactive and proactive strategies that are chosen according to network conditions. Some protocols can be found in the literature for MANETs which compare both independent reactive and proactive approaches. For example, ZRP (Zone Routing Protocol) [Haas and Pearlman (1998)], the first protocol to apply reactive and proactive strategies in a hybrid solution, establishes a zone around every node where routing updates are performed proactively, and outside of these zones the protocol responds reactively. Its main goal is to reduce the routing overhead. SHARP [Ramasubramanian et al. (2003)] presents an extension of this approach in which zones can be dynamically determined only around the nodes with significant incoming data, and it also allows adaptation with other application-specific metrics, such as jitter and loss rate, in addition to routing overhead.

Such a hybrid adaptive approach has not been fully applied to the WSN domain and the referred MANET protocols are not suitable for sensor networks due to several particularities that must be considered for a proper solution in these networks. Unlike MANETs, in which communication is essentially many-to-many, in WSNs it is usually many-to-one (sources to sink or sources to cluster-head). This characteristic provides a broader view for a sink or cluster-head to perform an adaptive control of the nodes under its responsibility. The traffic characteristic also differs between these networks, and this fact must be considered in an adaptive model. While in MANETs the nodes can dictate how data is generated according to their independent applications and user needs, in WSNs this traffic can be dictated by an external application for the whole network according to different requisition models or queries [Tilak et al. (2002)]. Also, for event-driven applications, there may be a spatial and temporal correlation among traffic of different nodes. Again, all these characteristics can lead to a proper adaptive rule in the WSN domain, which is the goal of the Multi solution.

6.3 Multi: A Hybrid Adaptive Algorithm for Routing in WSNs

This section presents the design principles of the proposed hybrid routing algorithm for WSNs, its routing mechanisms, and the adopted adaptive rule.

6.3.1 An Overview

This work proposes a new hybrid and adaptive algorithm for routing in WSNs, called Multi, that adapts its behavior autonomously in response to the variation of the network conditions. Multi is built using traditional reactive and proactive routing strategies for WSNs, and it applies an adaptive rule based on specific requisitions and traffic characteristics of this kind of network. Multi's implementation addresses particular algorithms and scenarios, but the insights presented here could help in the design of other adaptive hybrid solutions for WSNs.

The adaptive control of Multi is performed by the sink node that interacts with external applications and monitors the traffic characteristic of the network, so many-to-one communication is assumed. Clustering, a common approach in WSNs [Kochhal et al. (2003)], is another approach that can be considered to achieve more scalability and resource savings as the number of nodes and the sensing area increases. Thus, the *cluster-heads* can be responsible for coordinating the activities of all nodes in its area, and in our case to realize the adaptive control. This model follows the scheme proposed in Chapter 5, where an autonomic entity controls the self-organizing behavior of the network, in case a proactive or reactive self-organization of the routing infrastructure, based on general perceptions and goals, and not based on information regarding individual nodes.

Regarding the application's requisition model, routing can be performed in continuous, event-driven or observer-initiated way [Tilak et al. (2002)]. Clearly, continuous and observer-initiated models are favorable for proactive protocols, because the traffic is predictable. Thus, the proactive behavior can be set with the definition of such requisition models. However, an event-driven model consists of a more complex situation because the event occurrence is unknown, and it is the case considered by Multi in this work.

In an event-driven condition, the network can stay inactive or with a low activity for long periods (e.g., months or years) until something happens. To extend the network lifetime, it is mandatory to reduce the energy consumption during the inactive periods. In such cases, the reactive routing approach is preferable because it avoids the proactive maintenance of the routing infrastructure and it is created only when necessary. However, in a given moment, several events can be detected generating a high traffic, which can be appropriate for proactive algorithms to avoid the path discovery cost. In these cases, a better adaptive model must be provided to detect these varying conditions in order to save energy, so it is an important part of Multi.

6.3.2 Routing Mechanisms

The proactive and reactive components of Multi are simple versions of classical approaches for WSNs and it is not the intention to consider them as new contributions, neither to compare them with other solutions in the literature. Our goal is to show how a hybrid solution can be built from them. In addition, an integrated forwarding rule is also presented for hybrid operation.

6.3.2.1 Proactive Component: EF-Tree

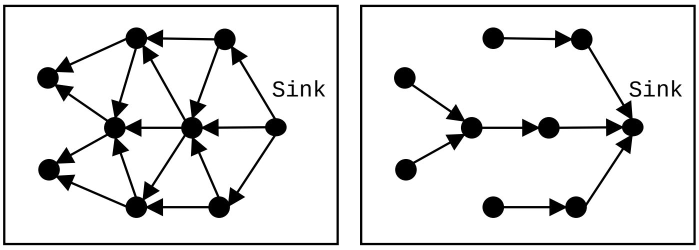
A simple and efficient structure for data dissemination is a routing tree that has been evaluated in some studies such as One-Phase Pull Diffusion [Heidemann et al. (2003)] and some implementations by Woo et al. (2003). Generally, the tree structure is created and maintained by a sink node in a proactive fashion, and a periodical rebuilding is used to support eventual topological changes, link problems due to interference or traffic variations, node energy degradation, etc. A specific implementation used in this work, called EF-Tree (Earlier-First Tree), is presented in the following.

- The sink node starts the process by broadcasting a control message as presented in lines 1 to 11 (Fig. 6.1(a)).

- When a node initially receives the building message, it identifies the sender as its parent and broadcasts the building message to all its neighbors. Messages received from other neighbors are discarded. This is shown in lines 22 to 31. Note that it is possible to define other possibilities to choose a parent node such as the node that belongs to a path with the highest amount of energy available. In particular, an interesting practical approach is presented by Woo et al. (2003) where the radio link quality is considered.

- Whenever a node has a data to be transmitted (sensed or forwarded by another node), it will send it directly to its parent. This is presented in lines 14 to 21 (Fig. 6.1(b)).

- The building process is periodically repeated so the network reflects eventual topological changes, such as failures, node movements and inclusion of new nodes. The function $TreeBuildTimer_i.timeout()$, in line 7, corresponds to this behavior. This periodicity depends on how frequent topological changes occur. More dynamic networks need shorter rebuilding periods.



(a) Tree building. (b) Routing towards the sink.

Figure 6.1: EF-Tree algorithm.

6.3.2.2 Reactive Component: SID

Another common approach for routing in ad hoc and sensor networks is to use on-demand algorithms, such as the AODV [Perkins et al. (2003)] and Push Diffusion [Heidemann et al. (2003)]. In this approach, the network may remain inactive until the communication process is started by sensors that have data to be sent, which is appropriated for event-driven scenarios. Obviously, the advantage of this approach is reinforced in situations where no management or control traffic is required, and

```
1: $EF - Tree.Init()$
2: begin
3:   if $n_i$ is the sink then
4:     $TreeBuildTimer_i = newTimer(TREE\_BUILD\_INTERVAL)$;
5:   end if
6: end
7: $TreeBuildTimer_i.timeout()$
8: begin
9:   {$n_i$ will build the tree}
10:  Send $TREE\_BUILD$ to all $n_k \in Neig_i$;
11: end
12: $EF - Tree.recv(msg_i)$
13: begin
14:   if $msg_i = DATA_k$ for some $n_k \in N$ then
15:     {Data will be forwarded to its parent}
16:     if $Parent_i \neq$ nil then
17:       Send $msg_i$ to $Parent_i$;
18:     else
19:       Drop $msg_i$;
20:     end if
21:   end if
22:   if $msg_i = TREE\_BUILD$ such that $origin_i(msg_i) = (n_j \rightarrow n_i)$ then
23:     {New Tree_Build Messages will be broadcasted}
24:     if not($msg_i \in ForwardedList_i$) then
25:       $Parent_i := n_j$;
26:       Send $msg_i$ to all $n_k \in Neig_i$;
27:       Put $msg_i$ in $ForwardedList_i$;
28:     else
29:       Drop $msg_i$;
30:     end if
31:   end if
32: end
```

Figure 6.2: EF-Tree Algorithm.

periodical wake-up schemes, such as in STEM [Schurgers et al. (2002)] or in B-MAC [Polastre et al. (2004)], can be used to save more resources while no event is detected.

It is proposed a simple implementation of an algorithm with such reactive behavior called SID (Source-Initiated Dissemination). It is very similar to the Push Diffusion and differs from it by the fact that it allows the source nodes to flood their data (e.g., when an event is detected) as long as a route is not available, and not only periodically as the Push Diffusion. These features make SID more suitable for event-driven scenarios, more reactive in the presence of network dynamics, but more sensitive in scenarios with intensive traffic. The SID protocol is presented as follows.

- A source node that detects an event broadcasts to all its neighbors the sensed data, its identification and a timestamp (Fig. 6.3(a)). This represents data identification. This process is shown in line 20 when the node is in the initial condition.

- Whenever a node receives the data sent by another node, it stores its data identification (source node identification and timestamp), as well as the sender identification. As the disseminated data is broadcasted, the node will receive it from all neighbors, however, it will register and forward only once (the first data received). This is shown in lines 14 to 26 when the *RequisitionList* is still empty. As described, the amount of memory necessary is in the order of the network size. Another criteria for path formation can be considered as in the EF-Tree.

- Similarly, data will arrive at the sink node from all of its neighbors. Then, the sink will send a control message requesting the data to be sent to the node from which it firstly received the data. This message identifies the data to be sent. This process is executed periodically by the sink while data is received, as represented by the function $RequisitionTimer_i.timeout()$ from lines 7 to 11.

- When a node receives this control message, it repeats this process identifying which node should send that data to it. This process is repeated until the source nodes are reached, as described from lines 27 to 31 (Fig. 6.3(b)).

- Once a source node receives the control message requesting its data, it will update its own table so subsequently it sends its data to the node that firstly requested it (line 18). Thus, data is disseminated through the path where the sink's requisition messages arrived (Fig. 6.3(c)).

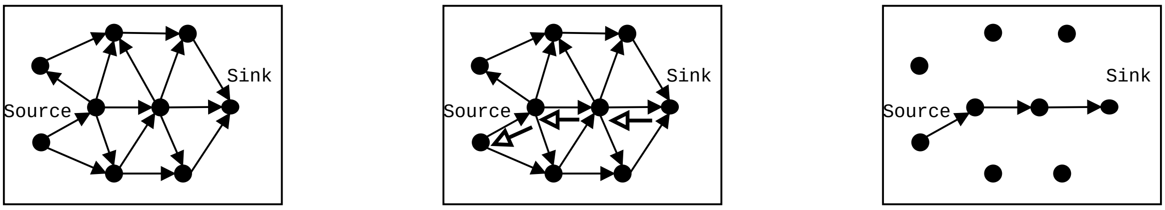
- In order to allow the network to adjust to eventual topological changes (due to failures, mobility or node inclusion), the requisition messages are periodically sent by the sink towards the sources while data is being received (condition in line 10). Once a node (source or intermediate) stops receiving requisition messages (condition in line 17), due to any topological change, the node will restart to send or forward data in broadcasts. Thus, if any path exists, data will reach the sink again and it will restart the requisition process.

- Once the events disappear, data will not be generated anymore and, consequently, the sink node will stop sending requisition messages to the sources. In absence of the periodical requisition messages of a specific data, the table entries will expire and the network will become inactive again.

6.3.2.3 Comparative Cost Analysis for EF-Tree and SID

In order to show the situations in which each algorithm has a better performance, a comparative cost analysis for them becomes necessary. Next, this analysis is presented regarding energy consumption, which is the basis for the Multi adaptive model.

When a path is formed by SID and EF-Tree algorithms, we have an approximated energy cost (in Joules) of data dissemination given by:



(a) Data diffusion. (b) Requisition messages. (c) Routing towards the sink.

Figure 6.3: SID algorithm.

```
$SID.Init()$
begin
   if $n_i$ is the sink then
      $RequisitionTimer_i = newTimer(REQ\_INTERVAL)$;
   end if
end
$RequisitionTimer_i.timeout()$;
begin
   {$n_i$ will send requisitions messages}
   Send $REQ_k$ to all $n_k \in ForwardedList_i$ such that $NOW - timestamp_k(msg_k) < REQ\_INTERVAL$;
end
$SID.recv(msg_i)$
begin
   if $msg_i = DATA_k$ from some $n_k \in N$ then
      {Data will be forwarded through the requisition path if valid, otherwise, it will be broadcasted}
      if not$(msg_i \in ForwardedList_i)$ then
         if Exists $msg_j \in RequisitionList_i$ such that $NOW - timestamp_j(msg_j) < REQ\_INTERVAL$
         then
            Send $msg_i$ to $n_j$ such that $origin_j(msg_j) = (n_j \rightarrow n_i)$;
         else
            Send $msg_i$ to all $n_k \in Neig_i$;
         end if
         Put $msg_i$ in $ForwardedList_i$;
      else
         Drop $msg_i$;
      end if
   end if
   if $msg_i = REQ_k$ for some $n_k \in N$ then
      {Requisition messages will be sent to the first sender of data}
      Send $REQ_k$ to $n_j \in ForwardedList_i$ such that $origin_k(DATA_k) = (n_j \rightarrow n_i)$ AND $NOW -$
      $timestamp_k(msg_k) < REQ\_INTERVAL$;
      Put $msg_i$ in $RequisitionList_i$;
   end if
end
```

Figure 6.4: SID Algorithm.

$$Cdata = Pt \cdot Sd \cdot Nh + Pr \cdot Sd \cdot Nh \cdot Nn \tag{6.1}$$

representing the cost of the data transmission by one source and the routers, and the cost of signal reception by neighbors. *Pt* and *Pr* are the transmission and reception power consumptions, respectively, in Joules/byte. *Sd* is the data size in bytes, *Nh* is the approximated number of hops from source to sink, and *Nn* is the approximated number of neighbors of a node.

The significant cost differences of the algorithms is given by the routing infrastructure formation. For the EF-Tree this energy cost is given by periodic broadcasts of tree construction messages, which is estimated by:

$$Ctree = (Pt \cdot Sc \cdot N + Pr \cdot Sc \cdot Nn \cdot N) \cdot \lceil (T/Ttree) \rceil \tag{6.2}$$

where Sc is the control packet size (in bytes), N is the number of nodes in the network, T is the execution time and $Ttree$ is the tree reconstruction period (both in seconds). The relation $\lceil (T/Ttree) \rceil$ represents the number of tree buildings in the execution time.

SID does initial broadcasts for each source and the sink sends requisitions messages towards source nodes for path definition. So we have:

$$\begin{aligned} Csid &= Ns \cdot (Pt \cdot Sd \cdot N + Pr \cdot Sd \cdot Nn \cdot N + \\ &\quad + (Pt \cdot Sc \cdot Nh + Pr \cdot Sc \cdot Nh \cdot Nn) \cdot \lceil (Tev/Tsid) \rceil) \end{aligned} \tag{6.3}$$

where Ns is the number of source nodes sending data, Tev and $Tsid$ are, respectively, the average traffic duration for every source and the sink's requisition messages interval (in seconds). The relation $\lceil (T/Tsid) \rceil$ gives the number of requisitions in the execution time.

Proposition 6.3.1 *There is a crossing point between the costs of EF-Tree and SID regarding the number of sources (Ns) in a time interval T.*

Proof. We can see that EF-Tree has a constant energy cost in function of the network lifetime, which is given by the multiplicative factor in function of T. SID, instead, has an energy cost in function of the number of sources (multiplicative Ns). A crossing-point in the energy costs of the algorithms is expected to occur as the number of source nodes increases in a given execution time. This point happens when the number of tree-building in this time is equal to the number of sources, because the terms in function of N (the great part of the energy cost) in $Ctree$ and $Csid$ get close assuming a little difference between Sd and Sc.

■

Thus, these facts make us conclude that it is not appropriate to use the EF-Tree algorithm during inactive periods of traffic which are expected in event-driven applications, and the SID algorithm should be used whenever the event detection ratio, which impacts in the number of sources, is low. These facts are explored on the proposed hybrid adaptive approach to achieve the better performance from both solutions.

6.3.2.4 Forwarding Rule

In a hybrid operation, the decision about the routing strategy chosen by the sink must be propagated to all network nodes and a broadcast control message might be used to switch the source nodes from one algorithm to another. In this work, this transition is simplified by assuming that all network nodes always respond normally to control messages as their original protocols determine, but a validity is maintained through a timestamp and a predefined period of time given by the periodicity of the reconstructions and requisitions of EF-Tree and SID, respectively. Thus, an integrated forwarding rule is applied to automatically adjust the network according to the sink decision: every data packet (generated or forwarded) is sent through a routing tree if it exists and it is valid (EF-Tree behavior); otherwise, data is sent to the neighbor from which a specific and valid requisition was received (SID behavior); otherwise, the data is sent by broadcast (Flooding) for path discovery.

6.3.2.5 Transition Analysis

It is important to analyze the impact of transitions in a hybrid adaptive algorithm. The presented instance of Multi is composed of operational changes between a reactive and a proactive behavior controlled by the sink and it is necessary to study what happens in these transitions and their impact on data routing.

The initial routing state of Multi is SID and it changes to EF-Tree routing state when a tree-build message is received. Multi stays in this last state while periodical tree-build messages are received. A timeout leads the algorithm back to the SID state. Figure 6.5 shows the state diagram of Multi in which the SID state was divided into two operation modes for data forwarding: SID-Ucast and SID-Bcast for unicast and broadcast, respectively, and the last case means data broadcast for route discovery. Along the time, different nodes in the network can be in different states. This may happen due to the following reasons: 1) The tree is under construction; 2) New nodes are added to the network in the EF-Tree mode; 3) The tree is not built for all the network due to packet losses; 4) The sink sends requisition messages, returning to the SID state, whereas part of the tree is still valid.

Proposition 6.3.2 *Considering the proposed forwarding rule, the communication between nodes in different routing states (due to some of the causes described before) is not disabled.*

Proof. This characteristic is shown in two parts:
From SID to EF-Tree state: Data generated or forwarded by a node in the SID-Ucast state can reach a node in the EF-Tree state, due to a tree construction for example

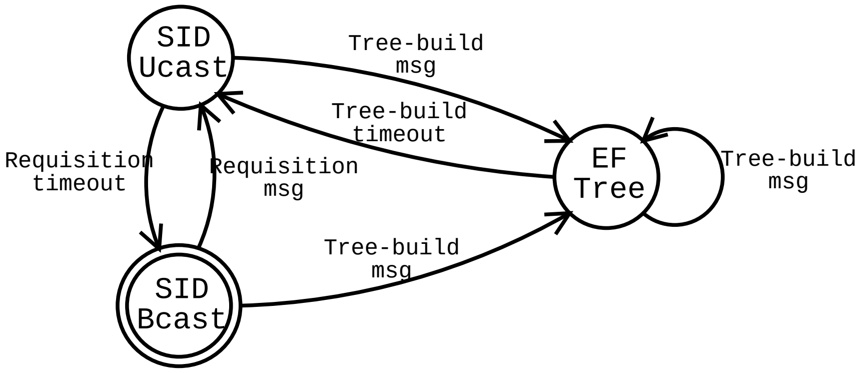


Figure 6.5: State diagram of Multi.

(first case). By the forwarding rule described before, the packet will follow the tree path normally. If a data packet is propagated by a flooding (SID-Bcast state), which occurs when the requisition message path is not built yet, this packet can reach the tree in more than one point and it will be forwarded with duplications until it reaches a tree-node which already received it. This last case is exactly what happens when new nodes are inserted, because they always start in the SID-Bcast state.

From EF-Tree to SID state: This situation happens when the sink node decides to return to the SID mode, sending requisition messages to source nodes that may be sending their data through a still valid tree. As the SID requisitions follow the reverse path of the last received data, they will follow the paths of the tree been used for each source. When the tree becomes invalid in a node which already received a requisition message, its reverse path will be used for data forwarding. Otherwise, the data will be forwarded by flooding, reaching the sink anyway, and the path for the source nodes will be built on the next requisition period.

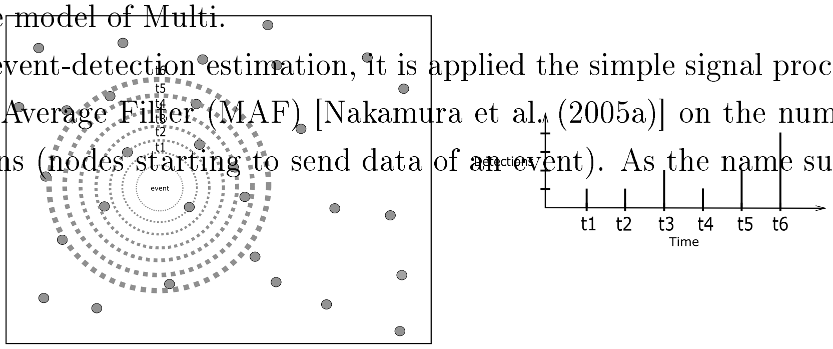
■

6.3.3 Estimation Model and Adaptive Rule

This section describes how adaptation is performed with the routing mechanisms described above in order to achieve performance improvements. In this work, the goal is to achieve energy savings.

In event-driven scenarios, a sensor node detects an event when its measured value represents a situation of interest. Once this occurs, it is assumed that the sensor node starts to generate data to the sink in order to inform and allow the monitoring of the events by some application.

In such scenarios, it is also expected the spatial and temporal correlation in event detections by sensor nodes. Additional to the influence area of events, that causes simultaneous detections in a proximity area, events may present a seasonal characteristic, where they can be distributed in time following an occurrence ratio, or a dynamic characteristic, where they can present an increasing actuation range or mobility (as the examples in Fig. 6.6). Thus, it is expect that an event with certain characteristic will not change abruptly. For example, a mobile or increasing event will be detected by new nodes following its velocity or increasing rate, or a seasonal occurrence characteristic will be maintained, and so its detection rate.



(a) Increasing event.

(b) Mobile event.

Figure 6.6: Examples of event occurrences and detections.

As described before, reactive algorithms are preferable in inactivity and low occurrence conditions. But by analyzing the reactive and proactive algorithms presented before it is observed that the cost of a data flooding for path discovery in SID is similar to the cost of a tree-building in EF-Tree, which establishes a routing infrastructure for all reachable nodes at once. Thus, if at least one source detecting events is expected to occur in the next time interval of route validity, the proactive behavior can be taken to avoid new discovery floodings and, thus, achieve energy savings. This is used by the adaptive model of Multi.

For event-detection estimation, it is applied the simple signal processing method of Moving Average Filter (MAF) [Nakamura et al. (2005a)] on the number of new node detections (nodes starting to send data of an event). As the name suggests, this filter computes the arithmetic mean of a number of input measures to produce each point of the output signal, and this can be translated in the following equation:

$$MAFout[i] = \frac{1}{m} \sum_{k=0}^{m-1} input[i+j], \tag{6.4}$$

where m is the filter's window size (number of input observations to be considered).

The MAF method was chosen due to its capacity to capture the behavior of the last m monitoring intervals, which is good for seasonal occurrence detection, and due to its characteristic of step response, which is good to detect correlated detection increases. It also filters the noise of measurements that can be caused by abrupt traffic variations due to packet losses, queue delays, or an occasional event distribution, improving the estimation. Other data fusion methods can be used for the node detection estimative, but MAF is simple and has low computational cost, necessary for constrained WSNs.

For the Multi's adaptation, as soon as a high traffic condition is detected and a proactive behavior is taken, more advantages can be achieved. Thus, the monitoring periodicity (and MAF supply) is configured as equal to the data generation rate of the source nodes (10 s in our simulations). The output of MAF is used as the estimate for the next period of observation. Therefore, as we want the estimate for the next route validity interval (ten times the data rate in our simulation cases), we can assume that it is 10× *MAFout*. Thus, the proactive behavior will be taken if this estimative is higher than or equal to 1 (i.e., a new estimated detection), and it corresponds to the *MAFout* ≥ 0.1 condition.

If Multi is already operating in the proactive mode with a valid routing infrastructure, this action is unnecessary and it is avoided until the validity expires. The resulting adaptive rule of Multi is depicted in Fig. 6.7.

6.4 Simulation and Evaluation

In this section, the algorithms described in this work are evaluated through simulation. First, it presents an evaluation of the *MAF* estimator to setup its parameters. Next, some specific scenarios assessing the performance of the Multi implementation described in this work are presented. They are used to compare Multi with SID and EF-Tree alone, with other well-know algorithms of the literature, which are Push Diffusion and One-Phase-Pull Diffusion, and with the version presented in Figueiredo et al. (2004b).

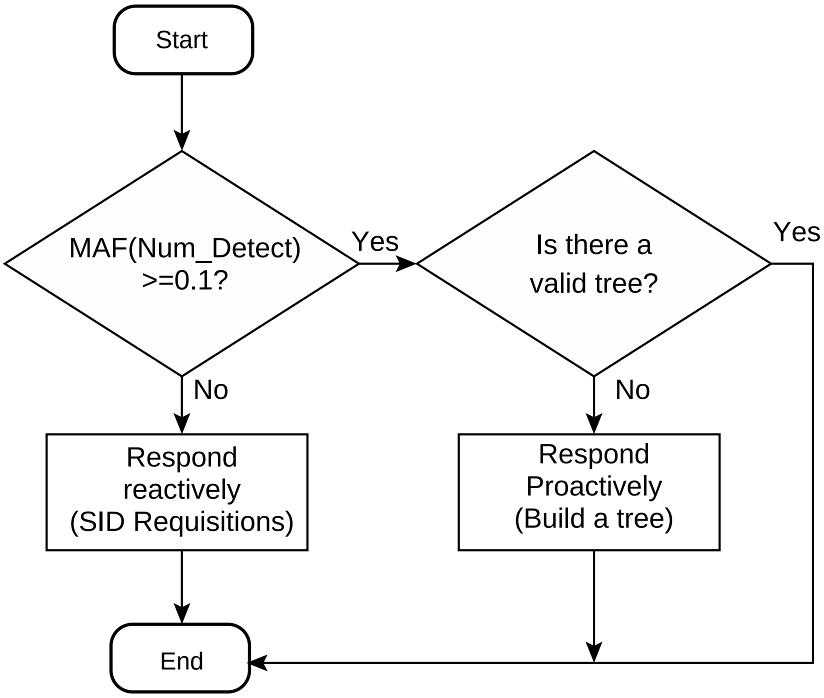

Start
MAF(Num_Detect) >=0.1?
Yes
Is there a valid tree?
Yes
No
No
Respond reactively (SID Requisitions)
Respond Proactively (Build a tree)
End

Figure 6.7: Multi's adaptive rule.

6.4.1 Parameters

The experiments were performed using the ns-2 Network Simulator [NS-2 (2004)] and the same Mica2 parameters described in the case studies in Chapter 5. As MAC protocol, again the IEEE 802.11 implementation available in ns-2 is used. But in fact, the power consumption relative to the channel listening (corresponding to idle power), which is very near to the reception power in the Mica2 nodes, cannot be ignored. As this consumption is very reduced in solutions like B-MAC [Polastre et al. (2004)], common to WSNs, and it is equal for all the evaluated algorithms, which results in a constant amount of energy added to all algorithms (verified through simulations), it is not considered for the comparative analysis.

In all simulations, it is considered a network size of 50 nodes randomly distributed in an area of 100 × 100 m^2 and only one sink. Both data and control messages have 20 bytes and are transmitted every 10 s and 100 s, respectively. The main metric evaluated was the energy consumption, which is a restrictive resource in a WSN. All experiments were executed 33 times with a confidence interval of 95% (vertical bars in graphics).

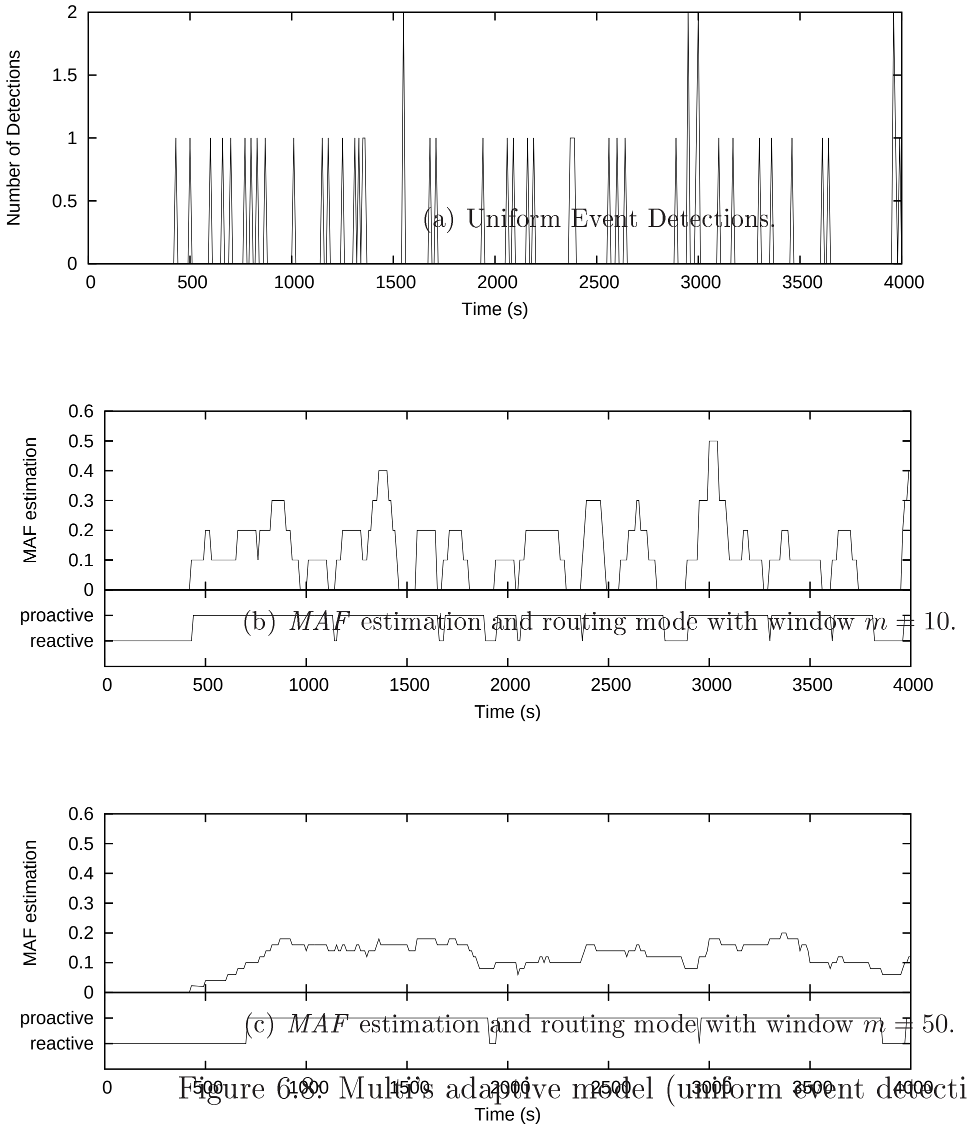
6.4.2 *MAF* estimator evaluation

Due to the absence of a generic event model for WSNs, this work represents the event-driven scenario by two types of detection distribution along time: uniform and normal distributions. In the first case, the variation on the detection ratio is represented by different number of sources generating data randomly along the simulation time. This distribution can represent the occurrence of uncorrelated event detections. In the second case, detections are distributed with a standard deviation around an average simulation time, and this distribution can represent the occurrence of correlated detections.

To better understand how the *MAF* estimator works, and how the window size m affects its performance it is shown in Figs. 6.8 and 6.9 some simulation cases following the previously described uniform and normal distribution considering 50 source nodes and window sizes of 10 and 50. The graphics show the detection measurements every 10 s, the resulting calculated *MAF* and the consequent operating mode of Multi (reactive or proactive mode).

In all cases we can note that the lower m, the faster the variation detection. On the other hand, the greater m, the smoother the estimation. For $m = 10$, we can see in Fig. 6.8(b) that *MAF* does not capture properly the seasonality of detection occurrence in an uniform distribution, causing several swaps between the reactive and proactive modes, and some of them undesirable. This can be seen in the change to reactive mode in the instants around 1100 s, due to an occasional empty interval just before it, which do not reflect the overall behavior of detections. In the sequence, a new detection happens, which represents an additional cost, and the Multi turns to proactive mode again. With $m = 50$ (Fig. 6.8(c)) these transitions are not so common and *MAF* has a better performance, however, it takes more time to respond from the initial condition (initial occurrences around 500 s). The impact of m is also observed with normal distribution in Figs. 6.9(b) and 6.9(c), but in this case, a lower m is better to respond faster to increases in the detections.

Based on more simulation cases with the adopted parameters, the $m = 20$ represented the best value to capture seasonal characteristics of uniform detections, which does not respond so slowly to variations of the normal distribution. This value represents two infrastructure building intervals (200 s) and it was used in the simulation scenarios presented as follows.



(a) Uniform Event Detections.

(b) MAF estimation and routing mode with window $m = 10$.

(c) MAF estimation and routing mode with window $m = 50$.

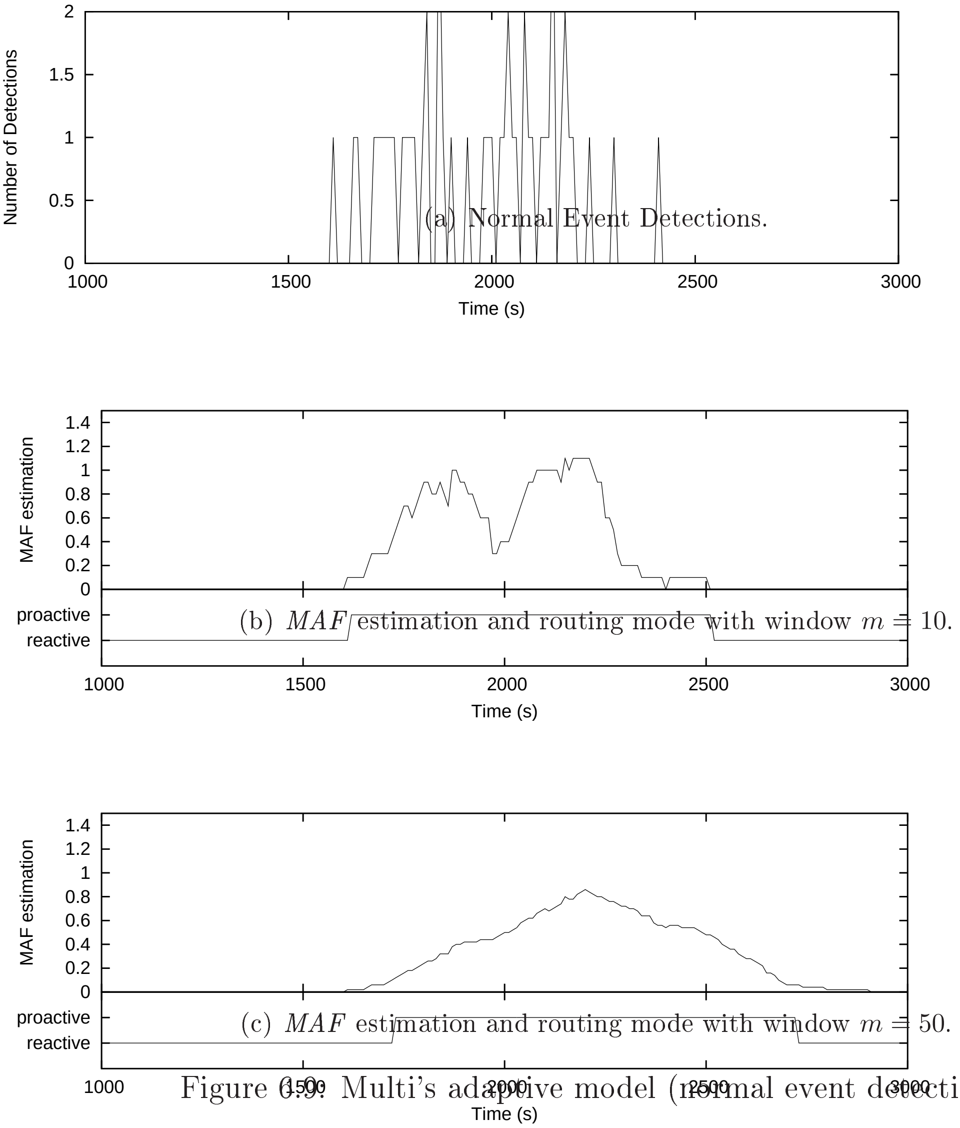
Figure 6.8: Multi's adaptive model (uniform event detection).

6.4.3 Multi Evaluation

6.4.3.1 Correlated detections

In the first scenario, sources generate data randomly with a normal distribution along the time to represent the occurrence of correlated detections of events. The average of this distribution is set in the middle of the simulation time of 4000 s and standard deviation of 100 s. The duration of the data generation was a random value between 1 s and 100 s.

Figure 6.10(a) shows the summary of the consumed energy in the entire network after the specified simulation time considering that the number of source nodes which



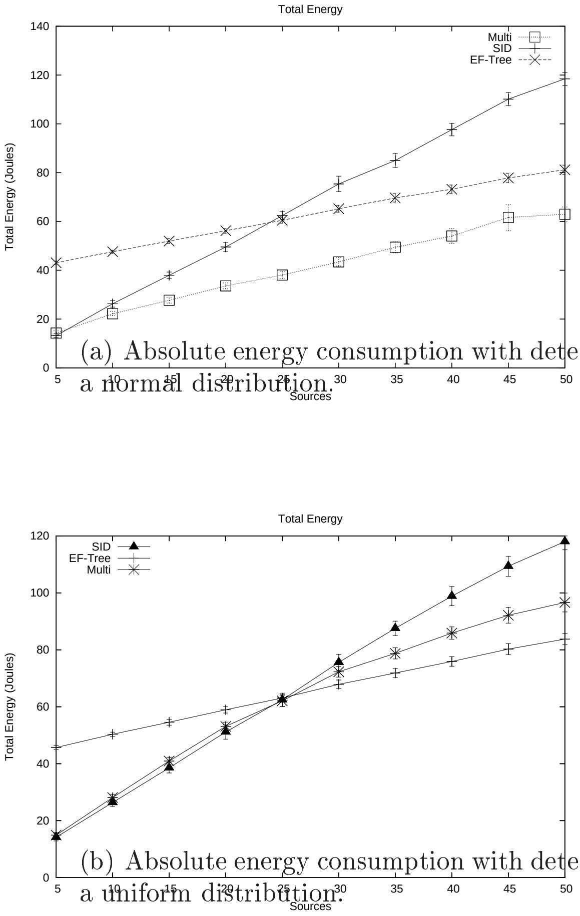
(a) Normal Event Detections.

(b) MAF estimation and routing mode with window $m = 10$.

(c) MAF estimation and routing mode with window $m = 50$.

Figure 6.9: Multi's adaptive model (normal event detection).

detected an event varied from 5 to 50 nodes, randomly chosen. In this simulation, with a not very intense traffic, all algorithms delivered nearly 100% of the packets (this graphic was omitted). Regarding the energy consumption, Multi outperforms the other algorithms almost to all number of sources. With the lowest number of sources (5 sources) Multi and SID consume almost the same energy, however, the difference between them becomes near 100% with 50 sources. The advantage of Multi is the adoption of the reactive behavior when the network is inactive, corresponding to energy savings related to EF-Tree, and the flooding avoidance of new sources when the proactive behavior is assumed, which represents a high cost to SID as the number of sources increases. This proactive behavior is taken when the MAF detects an increase



(a) Absolute energy consumption with detections in a normal distribution.

(b) Absolute energy consumption with detections in a uniform distribution.

Figure 6.10: Multi's performance under different traffic characteristics.

on the traffic, so its value becomes higher than the threshold 0.1.

6.4.3.2 Uncorrelated detections

The second scenario, based on the previous one, represents uncorrelated detections of events. The variation on detection ratio is represented by different number of sources generating data randomly along the simulation time of 4000 s with an uniform distribution.

In Fig. 6.10(b), SID and Multi outperform EF-Tree when the number of sources is small (less than 25). In this case, Multi operates just like SID, due to a low number of detections by Multi's MAF and consequently no proactive behavior is taken. This advantage happens because, with this reactive behavior, the routing infrastructure is created and maintained only when necessary and constant proactive behavior is unnecessary with a low amount of detections. However, when the traffic increases, EF-Tree begins to be more appropriated because it builds the routing infra-structure for all nodes at once and this proactive behavior is compensated by the initial floodings of SID for path discovery. In this case, Multi's performance becomes closer to the EF-Tree, outperforming SID. It happens because MAF detects the high traffic condition and the proactive behavior is taken. Again, all algorithms present a high packet delivery ratio and this graphic was omitted.

In this last case, with high number of detections, Multi and EF-Tree performances are not the same because as the simulation time was kept constant, the increase in the number of sources leads to a decrease in the inactivity time amortizing the cost of the EF-Tree from the start of simulation, while Multi always starts to operate as SID, changing to EF-Tree only when the MAF detects this traffic condition, so some initial data flooding happens. In a real event-driven scenario, we can expect to have longer inactivity periods than in the previous simulated scenario. Thus, the energy cost of the EF-Tree is shifted to a higher value due to its proactive characteristic which results in a better relative performance for both SID and, in special, Multi.

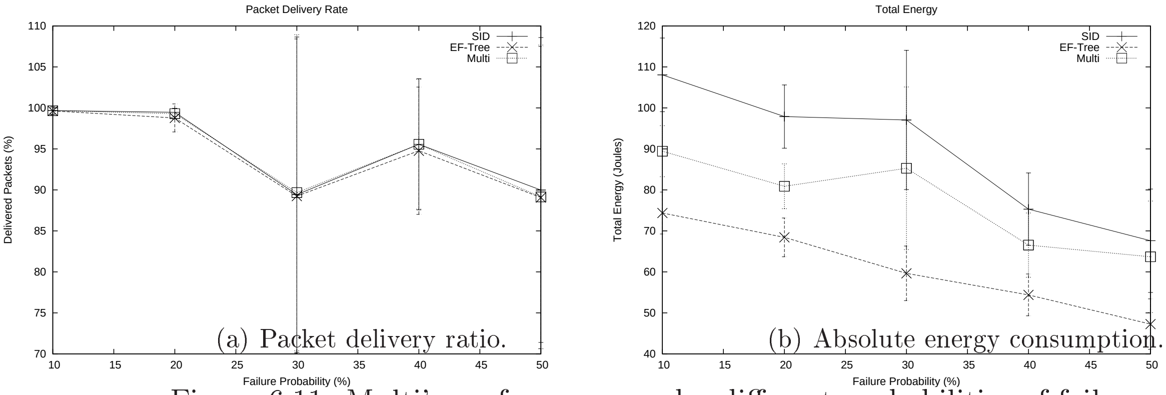
6.4.3.3 Robustness

To evaluate the resilience of Multi, the number of sources is fixed in 50 nodes with an uniform distribution in the simulation time, and varied the probability of node failures from 0 to 50% randomly during the simulation. When a node fails, it stays inactive until the end of simulation. As we can see in Fig. 6.11(a), all algorithms presented a similar performance regarding the delivery ratio (relation between received and data packet sent) because in all cases the time to recover from a failure is equal (10 times the data rate). Regarding the energy consumption, instead, the EF-Tree outperforms SID. This happens because in the SID algorithm sources have to flood messages for path discovery whereas the EF-Tree algorithm rebuilds the tree for the entire network at once. Multi's average performance stayed between EF-Tree and SID performances because it can operate in one of these two modes during the network lifetime.

6.4.4 Comparison with other algorithms

The evaluation of Multi performance against others contributions found in literature is performed by considering the Push and 1-Phase-Pull(1PP) versions of the Direct Diffusion [Heidemann et al. (2003)], classical solutions in literature. The same scenarios described before are used, and Push and 1PP Diffusion parameters such as data rate, control messages (interests and reforces) periods and packet sizes, are set to correspond to the Multi parameters. The simulation results are presented in Fig. 6.12 and 6.13.

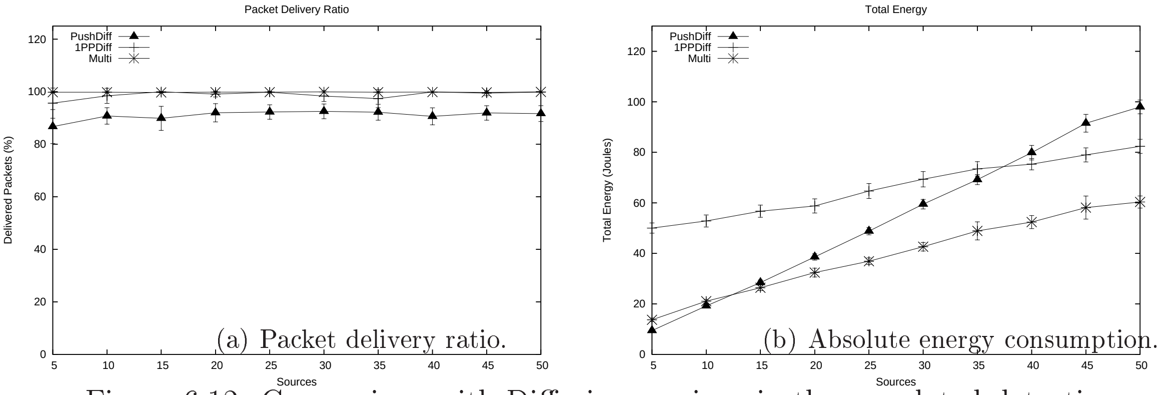
As we can observe, Push Diffusion and 1PP Diffusion behaviors are close to the SID and EF-Tree, respectively (see Fig. 6.10). It is due to their similar characteristics



(a) Packet delivery ratio.

(b) Absolute energy consumption.

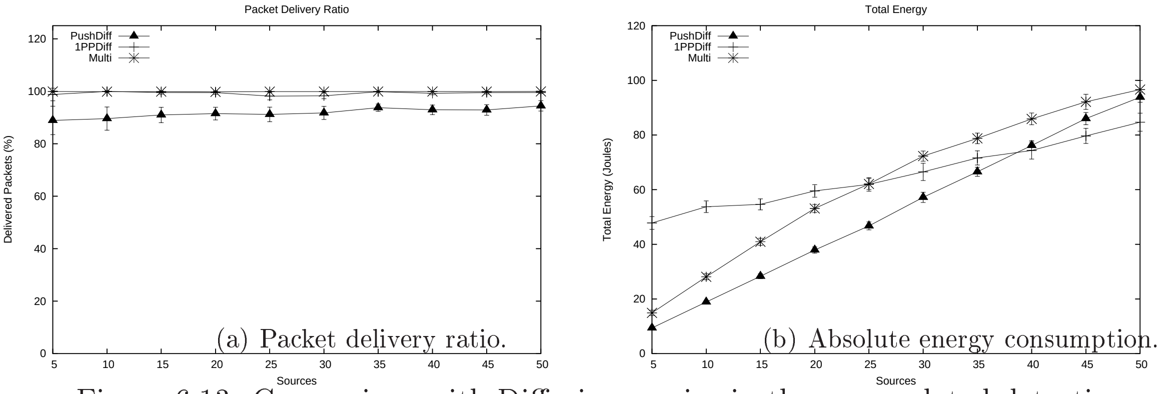
Figure 6.11: Multi's performance under different probabilities of failure.



(a) Packet delivery ratio.

(b) Absolute energy consumption.

Figure 6.12: Comparison with Diffusion versions in the correlated detection scenario



(a) Packet delivery ratio.

(b) Absolute energy consumption.

Figure 6.13: Comparison with Diffusion versios in the uncorrelated detection scenario

of infrastructure construction and maintenance. The main difference exists between SID and Push Diffusion. In Push, exploratory packets are sent in longer fixed periods, while in SID data are sent by broadcast until paths are built. This difference makes SID more appropriated for event-driven scenarios and Push Diffusion more susceptible to eventual packet losses, as we can see in the delivery ratio graphics. Obviously, less delivery packets results in lower absolute energy consumption, which is not an advantage in this case. A wider comparison among the related protocols can be found in Siqueira et al. (2006a). In summary, we can see that Multi's advantage is also observed against these classical algorithms by adapting to variable conditions.

Regarding the previously hybrid adaptive solution presented in Figueiredo et al. (2004b), which was also presented as a case study in Section 5.4, it uses a threshold on the number of sources as adaptive model. This model does not represent the event occurrence characteristic properly. For example, occasional distribution of event detections can lead to a concentrated measurement higher than the established threshold. This can lead to a proactive behavior that may not necessary correspond to an event occurrence characteristic. This solution is also evaluated by considering the uniform detection scenario described before, which better presents the variance in detections. We used thresholds of 1 (Multi-1) and 3 (Multi-3), because they leaded to the best performance in the lower and high traffic cases, respectively. As we can see in Fig. 6.14, Multi results with the new adaptive model are closer to the best performance of the others, thus it is a more adaptable solution to respond to event occurrence characteristics.

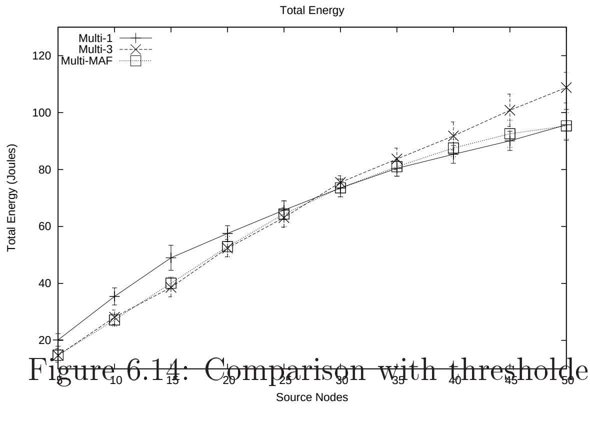


Figure 6.14: Comparison with thresholded version.

6.5 Related Work

Section 6.2 presented a discussion about the fundamental function of routing in WSNs with some related work found in literature [Al-Karaki and Kamal (2004)]. The described implementations of classical proactive and reactive routing approaches, respectively, EF-Tree and SID, were used as a basis to build our hybrid adaptive approach. They were better evaluated in Figueiredo et al. (2004c), in which comparisons with the traditional Diffusion solutions [Heidemann et al. (2003)] show a very similar performance. Thus, the evaluation results of Multi can also be extended to show its advantages against these traditional solutions.

Particularly to the reactive component, the recent LRDE [CHAE et al. (2006)] solution proposes a heuristic to reduce the energy cost of path discovery, and such an approach could be used by the SID component of Multi. In this case, the adaptive model of Multi should be adjusted to determine the better value of event-detection estimation for changing the routing strategy from the reactive to the proactive strategy. This solution can be evaluated as future work.

To the best of our knowledge, there is no other solution for WSNs considering hybrid adaptive behavior. However, this approach was first introduced in the MANET domain, and Section 6.2 already discussed some points about the particularities of the WSN domain. But now, after presenting Multi, some specific points are discussed regarding two classical solutions for MANETs: ZRP (Zone Routing Protocol) [Haas and Pearlman (1998)] and SHARP (Sharp Adaptive Hybrid Routing Protocol) [Ramasubramanian et al. (2003)]. The former maintains a fixed proactive zone around every network node, and outside this zone the protocol responds reactively. Regulating the zone size, the protocol can reduce the route discovery cost but the constant cost of the proactive zone is increased. As it was discussed in this work, this cost may represent an unnecessary resource consumption depending on the event occurrence. For instance, without any event detection, even a small proactive zone may represent an unnecessary cost. SHARP, instead, adjusts the proactive zone only around nodes with significative incoming data. In fact, the adaptation rule of SHARP only considers the number of sources and the respective cost to maintain a proactive zone of some radius around the destiny. It does not consider a temporal and spatial relation among sources which may occur in event situations in WSNs. Although the approaches share the same insight of the use of reactive and proactive behaviors, Multi solution differs by considering the traffic characteristic of WSNs to try to estimate event detections and to avoid completely the reactive path discovery with a proactive action. These are important aspects to optimize the routing maintenance overhead in these very resource constrained networks. In addition, Multi considers simple and feasible reactive and proactive routing solutions for WSNs.

6.6 Chapter Remarks

This chapter proposed an adaptive hybrid approach for routing in WSNs which allows the autonomous adaptation of the routing strategy to improve the energy efficiency when the network presents different variations of its traffic condition. This solution represents a practical solution in which the self-organization concept is applied in a very objective way, and it shows the advantage of adapting this self-organizing behavior according to general network perceptions following the autonomic control scheme.

Particularly, Multi unites the behavior of feasible reactive and proactive algorithms for WSNs, and adapts itself according to an event-detection estimation model to achieve energy savings. By observing the evaluation results, we can see that when operational conditions of the network are unpredictable, self-adaptive approaches like Multi should be more adequate than a single strategy algorithm, extending its applicability.

The future work includes the evolution of Multi. We intend to improve both reactive and proactive algorithms used, to evaluate other techniques for event estimation, and to include and evaluate other functions like data aggregation and node scheduling in the proposed solution. Additionally, with the applied approach it is possible the construction of new hybrid adaptive solutions including other routing algorithms, like those with geographical and hierarchical characteristics. The objective is to find other application requirements and network metrics in which an adaptive behavior among different routing strategies can bring a better performance.

Chapter 7

Integration of Self-Organizing Functions: A Case Study on Routing and Density Control

Many self-organizing functions are expected to be applied in a WSN, as it is shown by many examples in Chapter 3. As a consequence, the design of these functions must be considered together in order to achieve a correct as well as efficient network operation. This aspect was discussed as an important design phase in Section 4.4.3, and it was provided some general ideas about how this integration can be done. Taking these general ideas, we designed a case study considering the fundamental functions of routing and density control (see Section 3.4), and the obtained solutions can also be seen as specific contributions of this thesis. The obtained results confirm the need of integration aspects in the design of self-organizing WSNs. Additionally, an evolution of the particular solutions have originated a master thesis [Siqueira et al. (2006a); Siqueira (2006)] as part of the SensorNet Project.

This chapter is organized as follows. Section 7.1 introduces the routing and density control integration problem. In Section 7.2 the integrated approaches (RDC-Sync and RDC-Integrated) are described. Section 7.3 presents the simulation model and the evaluation results of the proposed solutions. In Section 7.4, the related work is presented. Finally, in Section 7.5, the conclusions are discussed and possible future work is described.

7.1 Introduction

As described in Section 3.4, routing and density control are two important functions to be considered for WSNs in the self-organization context. The former is important because the fundamental objective of WSNs is to report information from the environment to the network observer, generally through the sink node, and the routing function is the responsible to do it in an efficient manner. And the last is important because since WSNs may have high density, it is possible to make an scheduling of sleeping nodes (i.e., nodes which can turn off their devices such as radio, sensor etc.) without deteriorating the network coverage and connectivity, i.e., guaranteeing the application requirements and improving energy savings.

As discussed in Section 4.4.3, integration is an important aspect to be considered in the design of self-organizing WSNs in order to achieve either a correct and efficient operation. The main motivation for integrating routing and density control lies on the impact that the density control causes on the performance of the routing when both activities are considered independently. The main aspects are described as follows:

- **Increase of the Network Dynamics.** Density control changes the activity pattern of the sensor nodes, which are often put to sleep. Besides interrupting their sensing tasks, nodes in sleep mode have their radios turned off, and then they cannot participate on routing. Therefore, at any moment, the density control function may disable a router node, damaging the routing infrastructure and, by consequence, causing data losses while this infrastructure is not repaired.

- **Increase of the Network Traffic.** Even though density control reduces the number of nodes in activity and thus the traffic generated by the application and the routing protocol, it introduces its own traffic flow (control messages need to be exchanged in order to let nodes interact among themselves). These additional control messages may cause more drops, delays, and energy consumption, which can affect the functions themselves. For example, control packet losses can lead to an incomplete routing infrastructure or an inefficient density control.

Taking into account such discussed impacts, the integration of routing and density control functions is a natural approach. In Section 4.4.3 it was presented two approaches considering the integration of self-organizing functions, and they are applied in this section in two different solutions as case studies. Both solutions apply the classic solution of tree-based routing for WSNs, and it was used the EF-Tree implementation previously evaluated in the Multi proposal (see Chapter 6), and a density control algorithm called OGDC [Zhang and Hou (2005)], which is a fully decentralized solution that guarantees coverage with connectivity.

The first integrated solution presents a simple synchronization strategy for considering both density control and routing functions independently, called RDC-Sync. This solution tries to synchronize the considered functions in order to rebuild the routing infrastructure when the topology is changed. Its advantage is the simplicity of the solution requiring minimal changes in the individual algorithm implementations. The second solution presents a simple yet powerful approach for integrating both functions inside a single solution, called RDC-Integrated. This cross-layer vision is essential for reducing the costs of creating and maintaining the network infrastructures. Furthermore, it guarantees that the topology dynamics caused by density control (nodes going to sleep) are immediately considered by the routing function without incurring in loss of information. Performance evaluation of the RDC-Sync and RDC-Integrated algorithms demonstrates the importance of integrating density control and routing.

7.2 Solutions

In this section, the solutions designed to integrate routing and density control are described. They present a motivation for applying the integrated design, besides illustrating some aspects that take part in this kind of approach.

The first evaluated approach, called **Synchronized Routing and Density Control**, or simply **RDC-Sync**, is an integration alternative for reducing the topology dynamics problem, keeping an independent functioning of density control and routing processes. The second one, called **Integrated Routing and Density Control** protocol, or simply **RDC-Integrated**, is a fully integrated solution, in which the routing and density control protocols are combined in a single one.

In the following, a brief description of the considered density control algorithm is provided. The description of the tree-based routing algorithm can be seen in Section 6.3.2.1. Next, the strategies for integrating these algorithms are discussed.

7.2.1 Density Control Algorithm: OGDC

The Optimal Geographical Density Control algorithm (OGDC) [Zhang and Hou (2005)] is a density control algorithm, whose goal is to maintain complete coverage and connectivity of a wireless sensor network for as long time as possible. The idea is to make the sensor nodes temporarily inactive (i.e., in sleep mode) when they are not essential for guaranteeing coverage and connectivity of the network. Since an inactive node consumes significantly less power than an active node, this mechanism saves energy and prolongs the WSN lifetime.

The OGDC algorithm is a localized algorithm in the sense that each node uses only local information to carry out the density control process. The active nodes are carefully chosen based on the position of other neighboring nodes that are already active. In the density control process, only the nodes which contribute with maximum additional coverage to the existing active nodes are chosen to be active nodes themselves. The algorithm works in rounds, so at each round the density control process is repeated and this set of active nodes is recomputed.

During the operation of the algorithm, there are three states in which the nodes can be: UNDECIDED, OFF and ON. In the UNDECIDED state, the node has not yet decided if it will stay active or not in the present round, so it keeps its participation on the OGDC decision process. In the OFF or ON state, the node has already decided if it will stay active or not, so its participation in the OGDC process is finished in the present round. If a node goes to the ON state, it stays active until the next round, but if it goes to the OFF state, it becomes inactive, turning off its devices in order to save energy. In a round, the global OGDC process finishes when every node has entered the ON or OFF states.

The density control process works as follows. Each node begins in the UNDECIDED state, starting a volunteering process in which it probabilistically decides whether to volunteer to be a “starting node”. Each node that volunteers, after a back-off period, goes to the ON state and broadcasts a POWER-ON message. Each node that does not volunteer sets its back-off timer to a larger value. If it does not receive a POWER-ON message within the expiration of its back-off timer, it repeats the volunteering process, with the volunteering probability doubled. When a node receives a POWER-ON message, it checks if its entire coverage area (the area within its sensing range) is covered by its neighbors (which already sent POWER-ON messages). If the area is covered, the node sets its state to OFF. If not, it (re)sets its back-off timer to a value computed using a formula which ensures that the node whose presence is most beneficial to the network at that time (the one which is closest to the optimal) gets the lowest valued back-off timer. If the back-off timer expires, the node enters the ON state and broadcasts a POWER-ON message. If a node detects the occurrence of collisions, it also enters the OFF state by assuming a very dense network.

7.2.2 Synchronization Approach: RDC-Sync

The first alternative for integrating density control (OGDC) and routing (EF-Tree) processes is to implement them independently, but configured to act synchronized. This synchronization is made in order to let the routing infrastructure to be renewed whenever the density control algorithm changes the set of active nodes. So, in this approach the time periods of these protocols have to be configured with the same value. Furthermore, in each time period, the routing tree construction has to be adjusted to start immediately after the density control process is done.

Nevertheless, to adjust the start of the tree construction in EF-Tree with the finish time of OGDC in a round in practice is a hard task. It demands the use of a synchronization mechanism which could be complex and imply in a high extra cost for the network. For this reason, if the RDC-Sync is to be implemented in a real WSN, possibly the network designer would prefer to estimate a synchronization point. As a result, the performance of a real synchronization solution will depend on the estimation quality of the synchronization point. If OGDC finishes after this synchronization point, part of the tree construction will happen concurrently with OGDC, risking the quality of the tree. On the other hand, if OGDC finishes before this synchronization point, during the time interval between the finish time of OGDC and the synchronization point, information may be lost because the tree will not be up-to-date.

The estimation quality of the synchronization point can be measured by considering the best and worst possible cases for it. A real synchronization solution, then, will have performance lying between the best and the worst cases. The best possible synchronization case, i.e., the one that gives a superior performance limit, is that in which the synchronization point is set to the value immediately after the density control process is done in the whole network. It can only be obtained by considering that a global view of the network is available. The worst possible synchronization case, on the other hand, results in the inferior performance limit. The worst situation happens when the synchronization point is set to the beginning of the round, in which case the density control and routing tree process start simultaneously and, as a consequence, the routing infrastructure is built over a topology that is in the middle of a changing process. We refer to the best case as **RDC-Sync-B**, an to the worst case as **RDC-Sync-W**.

Regarding the implementation of RDC-Sync-B and RDC-Sync-W, we considered that the sink node do not participate on the OGDC process (because it can not be turned off), but contributes for the network coverage by keeping its sensor permanently turned on.

7.2.3 Fully Integrated Approach: RDC-Integrated

A better integration alternative is the one that combines the density control process of OGDC with the EF-Tree routing process, rather than putting one protocol on the top of the other, and this is the main idea of RDC-Integrated. The purpose of this solution is to incorporate the OGDC process inside the tree construction process. The result is a protocol that performs the density control using the tree construction messages, i.e., without incurring additional costs, and reduces the impact of the dynamic of density control because the process is executed at once.

To include the OGDC mechanism inside the tree construction process, some modifications to this algorithm are needed. First, the tree construction messages will serve as POWER-ON messages. Thus, the density control process is initiated only by the sink, i.e., the sink has to be pre-defined as the only starting node, rather than having the starting nodes chosen randomly in a probabilistic volunteering process. As a consequence, in RDC-Integrated the sink will be the first node to send a POWER-ON message and thus the following POWER-ON messages sent will flow from its position to the network boundaries. Once a node receives these tree construction messages, which also means a POWER-ON, it performs the activity decision process of the OGDC, and it only proceeds with the construction process if it decides to stay in the active mode. Since in OGDC the POWER-ON messages are originated by nodes which have entered the ON state and hence will stay active in that round, upon receiving a POWER-ON message, the node can include the message source as a candidate for being its parent in the tree. When data need to be routed, the parent is used as a path to reach the sink.

In this implementation, the source of the first POWER-ON message is chosen as the node's parent in the tree. This rule is the one considered in EF-Tree, but depending on the application requirements, different rules can be applied for electing the best candidate, such as the closer node, the node with higher energy stock etc.

7.3 Simulation and Evaluation

Simulations were performed to evaluate the performance of the integrated solutions and show practical results which motivate the use of integrated design. As described in the earlier chapters, we used the ns-2 (Network Simulator 2) [NS-2 (2004)] set with the parameters based on the Mica2 nodes [Crossbow Technology Inc. (2004)].

It is considered a continuous sensing application in which nodes periodically send their data to the sink node located at the center of the area. The periodicity of report was set to 20 s, all the nodes beginning their sensing activities in random times between 0 and 20 s. The data packets sent have 32 bytes. The sensing range considered was 20 m, which is the half of the communication range. As it is possible to connect a sensor on the gateway, we assume that the sink is also capable of sensing.

For the network routing, the EF-Tree protocol was applied. This protocol builds a routing tree with the sink as a root and which is periodically updated every 100 s. The control packets of EF-Tree have 32 bytes. For the MAC layer, as the Mica2 nodes implement a CSMA/CA protocol, the IEEE 802.11 version available in ns-2 was chosen to be used. For density control, the OGDC protocol was used, configured with a round time of 100 s. As in EF-Tree, the control packets of OGDC have 32 bytes. Regarding constants and timeout values for activity decision in the density control process, they are the same as shown used in Zhang and Hou (2005).

Concerning the network setting, the nodes were randomly distributed in the area, according to a uniform distribution and they are not mobile. Regarding the initial energy, it was set to 100 J. This value is sufficient for the network to live for more than 3,000 s, in all the network settings. The number of nodes and density were varied in the simulations, so their values will be described later.

The integrated solutions following the synchronized approach, RDC-Sync-B and RDC-Sync-W, which define the range of a practical solution (with the estimation of a synchronization point), and the fully integrated solution, RDC-Integrated, are evaluated and compared in two simulation scenarios. Their are also compared with a solution not applying density control, called RT, which only uses the EF-Tree routing and shows the importance of considering both functions. The purpose of the first scenario is to provide a performance comparison between these solutions when the network size is varied. The second scenario is designed to allow a comparative analysis along the simulation time.

7.3.1 Network Size Variation Scenario

In order to compare the performance of the chosen scenarios, simulations were conducted varying the number of nodes from 50 to 200. The area dimension was varied simultaneously in order to keep a fixed density of 5 nodes per 1,000 m^2. In this set, all the simulation experiments were executed for 3,000 s and repeated 33 times. Next, the results are shown, which present confident intervals of 95%.

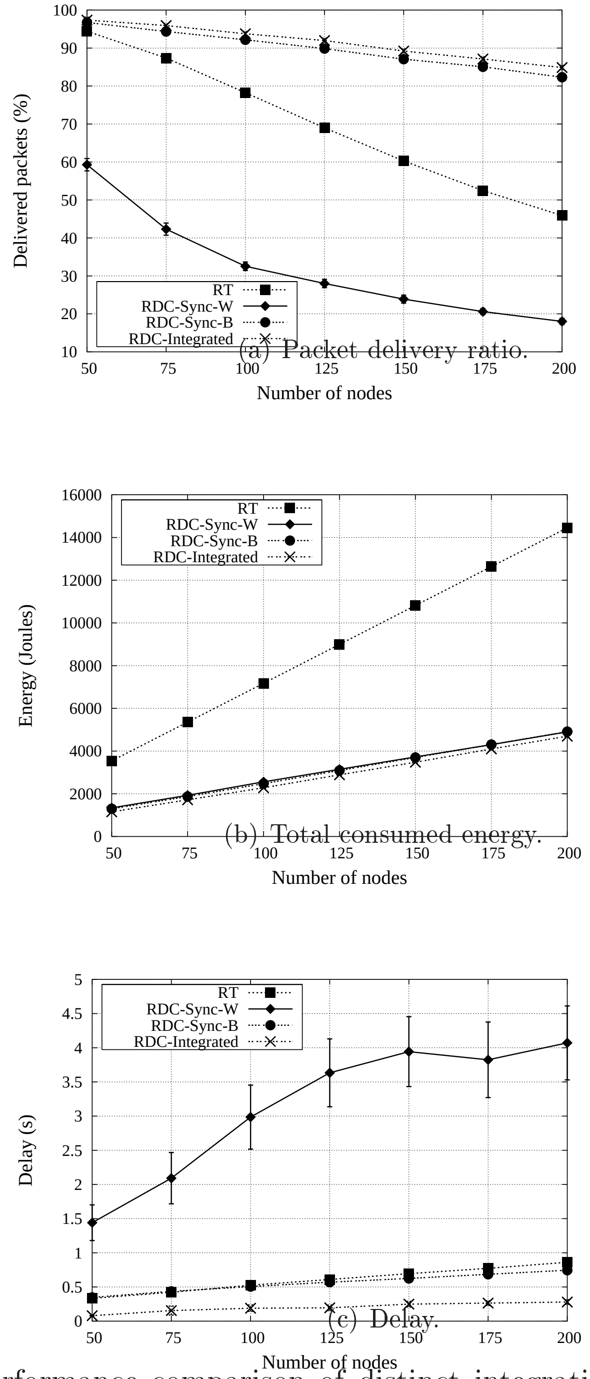
Fig. 7.1(a) shows the packet delivery ratio per network size, for all scenarios. We can see a better performance of RDC-Integrated, which is the only solution that completely avoids the invalidation of routes due to the dynamics introduced because of density control (when a node state is switched from active to inactive). In RDC-Sync-B, this effect of density control is only partially avoided because packets may be lost before the global synchronization is reached, i.e., before the density control process has finished in all nodes. In RDC-Integrated, the routes are computed inside the density control process and so this reduces the time when routes may be invalid. Nevertheless, although the difference between these solutions is small, the global view in RDC-Sync-B is hard to be achieved, thus we can expect real performance to lie between the results shown for RDC-Sync-B and RDC-Sync-W scenarios, which may be very poor (even lower than that of RT) depending on the estimation quality of the synchronization point. In RT (without density control), we can see a delivery ratio near that of RDC-Integrated for smaller network sizes, however, as the number of nodes is increased, it is negatively impacted by the high amount of redundant data generated in a very dense network.

Fig. 7.1(b) shows the energy consumption results of the four scenarios. Obviously, in the solutions that apply density control, less energy is spent due to the lower number of active nodes. In addition, these solutions reduce redundant data and the traffic as a consequence, saving even more energy. Moreover, if the density control is integrated with the routing construction, the energy efficiency is even better, as the results for RDC-Integrated in Fig. 7.1(b) show. The application of this strategy reduces the number of control messages by integrating the messages of density control and routing tree construction, and these functions are performed independently in the other approaches. In addition, as we considered the effective participation of the sink in the density control process, we have a little lower number of active nodes (see Fig. 7.2(c)), and this improves the energy savings.

In Fig. 7.1(c), we can see the impact of the evaluated solutions on the delay. RDC-Sync-W has the worst performance due to the high absence of valid routes. Data packets are retained in the routing queues for retrials, and this causes the increase on the delay of successfully received packets, which are in a few number in this case. RDC-Integrated has the best performance because in this solution, as soon as the topology is altered the routing tree is rebuilt and data can be correctly and immediately forwarded.

Regarding the resulting area coverage, we can see in Fig. 7.2(a) that the better global coverage is obtained with RT, which does not perform density control, and so all nodes remain active (see Fig. 7.2(c)). The other solutions also present a high area coverage, with a difference of 3%, but a lower performance is observed in RDC-Sync-W. This is explained by the impact of the routing tree construction that is performed concurrently with the density control negotiation, which can cause some collisions and control packet losses. As described in Section 7.2.1, in the OGDC scheme these collisions can make some nodes to become inactive, because it is believed that if the number of collisions is high, it means that the network is dense. Nevertheless, collisions may have other causes and, as a result, OGDC could negatively affect the global coverage. In addition, the packet losses can force a node to become active even if there are neighboring nodes covering its area.

If the coverage is measured in the sink (see Fig. 7.2(b)), we can also see a good performance in all solutions but RDC-Sync-W, due to its low delivery ratio. RT also has a low delivery ratio, but the coverage is maintained as a result of data redundancy. Nevertheless, when the network size is increased, the drawbacks of this strategy can be seen. Keeping less than 50% nodes (see Fig. 7.2(c)), the other solutions can manage to obtain almost the same coverage with reduced network traffic and energy consumption.



(a) Packet delivery ratio.

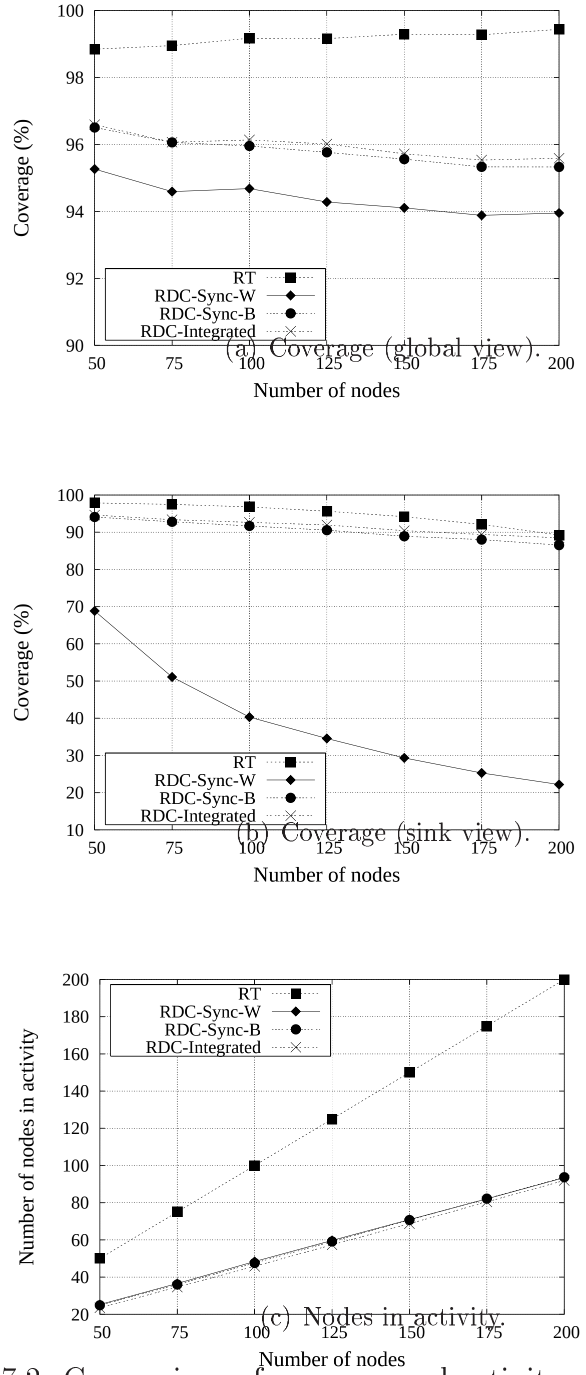
(b) Total consumed energy.

(c) Delay.

Figure 7.1: Performance comparison of distinct integration solutions per number of nodes.

7.3.2 Lifetime Evaluation Scenario

In order to verify the impact of each approach on the network coverage in time, simulations were conducted without limiting the simulation time, i.e., they were executed until all the sensor nodes have run out-of-energy. For these experiments, the number



(a) Coverage (global view).

(b) Coverage (sink view).

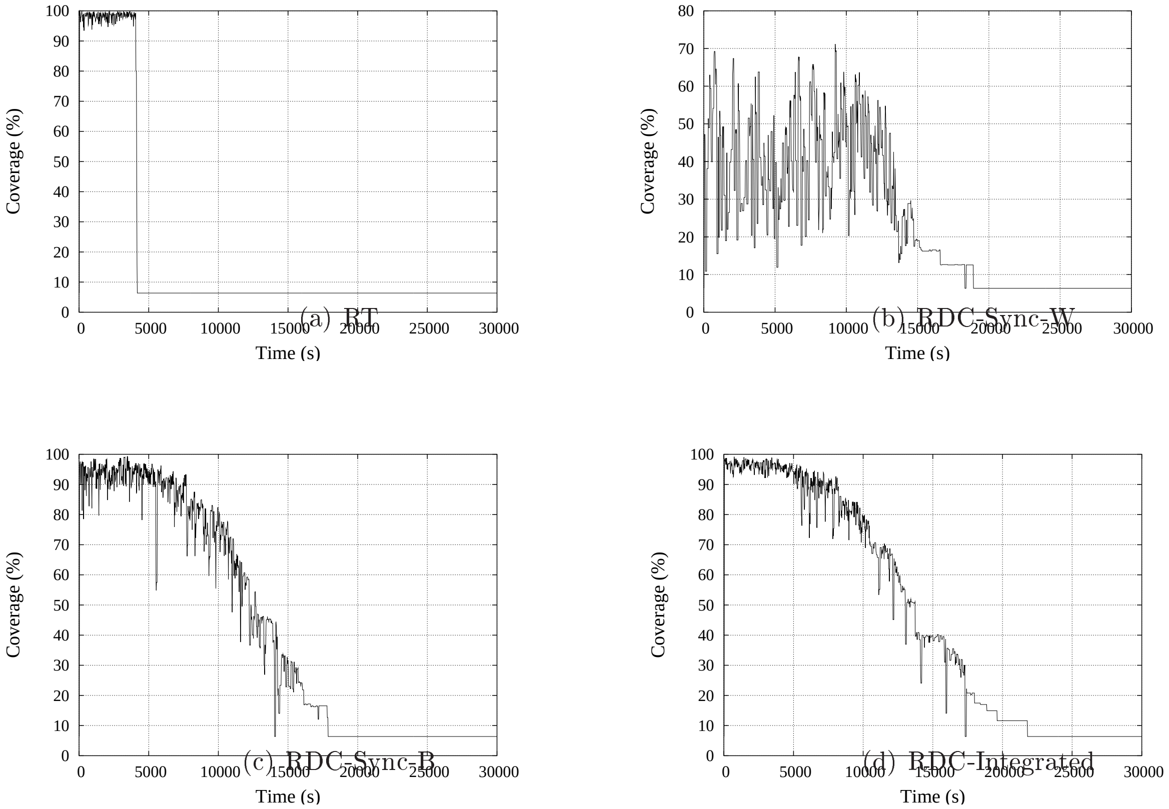
(c) Nodes in activity.

Figure 7.2: Comparison of coverage and activity per number of nodes.

of nodes was fixed in 100 with 5 nodes per 1,000 m^2. The results presented next show coverage in time for one simulation of each scenario.

Fig. 7.3 shows the coverage along the simulation time considering the sink view of the network, i.e., it considers only the data being received by the sink and so considers the packet delivery performance.

We can see the great benefit of density control on the network lifetime. While with RT the network completely loses its coverage before 5,000 s, the other solutions manage to prolong coverage until almost 20,000 s, i.e., an increase of four times in network lifetime. Nevertheless, RDC-Integrated presents a better performance, keeping the highest coverage at most of the time. This lifetime extension occurs due to its energy savings, as showed before. The differences among the integration strategies become clearer, and also the performance of the routing protocol. As we can notice, among the solutions with density control, RDC-Sync-W shows the worst performance, as expected. The delivery ratio makes the difference, as we can note mainly in the RDC-Sync-W performance with a low sink coverage and a high variance, and in the lower variance of the RDC-Integrated when compared with RDC-Sync-B.



(a) RT

(b) RDC-Sync-W

(c) RDC-Sync-B

(d) RDC-Integrated

Figure 7.3: Comparison of coverage (sink view) in time.

7.4 Related Work

Recently, a new research area has been opened for improving the performance of wireless networks, generically termed “cross-layer design”. The idea is to promote performance gains by jointly designing network protocols which would traditionally reside in different layers. When together, the protocols can interact and exchange important information. For integrating MAC and network (routing) layers in WSNs, for example, there are already some cross-layer proposals. In MINA [Ding et al. (2003)], it is presented a unified framework that encompasses network organization in groups, MAC and routing protocols. In Zorzi (2004), a contention-based MAC protocol is designed to be integrated with the GeRaF protocol, an on-demand geographical routing protocol. In Sichitiu (2004), a solution for random scheduling nodes is supported by interactions from the MAC and routing protocols. Another example of integration is shown in TD-DES [Cetintemel et al. (2003)], in which some application choices are determined in the network layer. In the presented case study, it is shown that an integrated approach is necessary not only for communication protocol stack, but also for other self-organizing functions.

Even though there are already many efforts on cross-layer design, few researches have yet shown concern on interaction between density control and routing, which is the focus of the present work. In ASCENT [Cerpa and Estrin (2002)], the authors observe that packet loss may occur when a node is set from the active to inactive state. They suggest that an improvement could be made by informing the network layer about ASCENT's state changes, so as to let the routing infrastructure be repaired. No solution was derived in detail and evaluated, though. Another work which deals more deeply with the cooperation between density control and routing is Tian and Georganas (2002) work. In this case, the proposed density control algorithm is evaluated in conjunction with LEACH [Heinzelman et al. (2000)], a cluster-based routing algorithm for WSNs. Their approach was to insert the density control phase just before the LEACH set-up phase, and so, at each round, inactive nodes do not participate on the cluster forming and communication infrastructure. Hence, their idea is to synchronize both protocols. In addition, LEACH-based solutions consider that nodes can communicate directly with the cluster-head, which turns easier the monitoring and controlling of the clustered nodes states but is often hard to be obtained in practice. The presented solution is different because a fully integrated and decentralized solution is provided following the self-organization concept.

7.5 Chapter Remarks

This chapter presented a case study on the integration of routing and density control, which are two important self-organizing functions for WSNs. It demonstrates the need of integration approaches for a correct and efficient network operation. But the proposed solutions can also be seen as interesting new contributions of this thesis.

It was proposed two solutions following the insights presented in Section 4.4.3: RDC-Sync, implementing the synchronization strategy, and RDC-Integrated, following the strategy of integrated design. Their results show that if the combination of routing and density control is not carefully designed, there is a risk of wasting the benefit of controlling the network redundancy because of an unsought information loss that could occur due to the impact of density control organization in the routing function. In addition, the results show that the fully integrated solution (RDC-Integrated) can present advantages by reducing the network overhead, consequently the energy consumption, while significantly reducing packet losses due to the dynamic introduced by density control.

Although the main purpose of presenting RDC-Sync and, mainly, RDC-Integrated in this work is to motivate the integrated design, they can be more explored as interesting individual proposals to be used in practice. In fact, an extension of this work was realized as a master thesis work [Siqueira (2006)], which better explored the density control and routing relation and better evaluated the presented solutions. As future work, we intend to integrate other self-organizing functions such as channel access scheduling or clustering in the RDC-Integrated process. The result of this work may be the obtainment of an integrated network architecture for the self-organization of WSNs that is more efficient than single solutions. Other possibility for future work could be the integration of different density control and routing solutions, mainly that ones with very distinct characteristics, such as on-demand solutions. Finally, the algorithms presented in this work could be implemented and evaluated in a real network, such as the Mica2 platform.

Chapter 8

Final Considerations

This chapter summarizes this thesis. It presents some general concluding remarks in Section 8.1, and describe some limitations of this work in Section 8.2. The current publications that came from the contributions of this thesis are cited in Section 8.3. Also, some ideas that can generate complementary work are presented in Section 8.4.

8.1 Concluding remarks

Self-organization is an important and challenging concept to be applied to the current large-scale computer systems, and specially to the WSNs. This concept is based on the achievement of a desired global behavior from local interactions among the system elements, and it can be used to achieve autonomic operation of these systems, i.e., an unattended operation without a central control. In fact, many WSNs applications demand on self-organization characteristics. This need is due to the scale, dynamic and unattended operation characteristics of such networks, and the self-organization concept matches these goals very well.

Although many self-organizing solutions already exist for WSNs and other computational systems, there still is a lack of general design aspects to be applied with objectiveness in the development of self-organizing WSNs. Thus, this thesis advances in this promising research area with two contributions in a conceptual level:

- **A Design Methodology.** This proposal reunites important practical design aspects and provides a guideline that can be useful in the understanding and in the development of new self-organizing functions for WSNs. It is not the purpose of this design methodology to generate closed and finished solutions. Actually, the experience of the developer is important to achieve effective self-organizing solutions. However, the methodology turns explicit important design aspects based on the study of common self-organizing functions, and this can help in new designs. Due to the absence of similar proposal in the literature, this methodology represents a step-forward in the designing of self-organizing systems.

- **A Management Scheme.** This contribution presents a scheme in which self-organization functions and centralized management entities are considered together for the development of effective autonomous solutions for WSNs. In a lower operational level, it considers that self-organizing functions can be used to achieve more powerful and simpler autonomous solutions, but in a higher level, it considers the practical aspect that the network behavior must be adjusted to different global goals defined by management entities and applications, or to global network perceptions. In this scheme, instead of controlling the network individuals, which is the approach of traditional management solutions, the management entities controls the network self-organizing behavior by changing the local rules played by the individuals. And this approach simplifies the management task by focusing it only on global aspects of network operation. As a complementary view to existing solutions in literature, this scheme turns explicit the need of adjusting the network self-organizing behavior to different goals, and it relates important aspects on the implementation of this vision.

With these more general contributions about self-organization in WSNs, it was envisioned some practical and interesting contributions with specific self-organizing functions of these networks:

- **Multi: A hybrid adaptive routing solution.** This contribution proposes a routing solution for WSNs that combines two self-organizing strategies for routing infrastructure formation, a reactive and a proactive, and an adaptation model performed by the sink based on global perceptions of the network to achieve energy saving. This solution follows the ideas present in the more general contributions in a more objective way, and it consists of an interesting routing solution to be applied in practice with significative results.

- **RDC-Sync and RDC-Integrated.** This contribution considers the important design aspect of function integration. The particular implementation proposes the integration of the routing function with density control, and both are fundamental functions for the proper operation of WSNs. It shows not only that integration is important to achieve a more efficient solution, but also to achieve correct operation, because different self-organizing functions can affect each other. This is the case when the density control function can damage the routing infrastructure organization by changing the state of active nodes. Although the proposed solution considers particular routing and density control instances, it represents an innovative solution to be used in practice, and the achieved significative results led to the extension of this work as a Master Dissertation [Siqueira (2006)].

In summary, this thesis advanced in general and specific aspects the concept of self-organization applied to the WSN domain. The advantages of treating this promising research area in a wider approach are the provision of different contributions regarding different aspects of self-organization in WSNs, and the obtainment of more possibilities of consequent work. In addition, some of the contributions present in this work, mainly those in the conceptual level, can easily be translated to other self-organizing domains such as the general ad hoc or mobile robot networks.

8.2 Limitations

Some contributions in this work are presented by considering general aspects of the application of the self-organization concept to the WSNs. Although it was presented particular case studies to show the usage and benefits of the proposed contributions, the validation of these general ideas are subjective. This observation is particularly true in the case of the design methodology. Maybe some validation methods of software engineering, in which several users consider the methodology and some metrics are taken from the design process, could be used to this purpose, but it is left as future work. This weakness is also present in the proposed management scheme. But in this case, as the contribution deals with a more specific and obvious problem, which is the need of adjusting the network behavior to different goals, the several case studies provide examples to show its applicability.

With the practical contributions present in this work, mainly in the case of Multi and RDC-Integrated, the validation of the solutions was based on simulation scenarios based on real platforms and assuming some common application characteristics. Although simulation approaches are very common and widely adopted in the research area of networks, and particularly in WSNs, it is necessary the evaluation of these solutions in real platforms considering practical aspects of implementation such as the memory usage, the required processing time to perform some function, energy consumption, and the radio communication reliability.

8.3 Publications

The contributions present in this thesis are reported in the following publications.

Introductory work regarding self-organization in WSNs is presented in Loureiro et al. (2003), which is an integrated effort of the SensorNet members presenting an overview of WSNs. Also, in Ruiz et al. (2004), the discussion on WSN architectures and existing protocols are extended as a collective work. Particularly, Figueiredo et al. (2005b) reports some representative self-organizing algorithms for sensor and ad hoc networks, and it is consequence of the work presented in Chapters 2 and 3.

Regarding the specific proposals present in this work, the design methodology is under revision in a new conference showing the advance of this research area in Figueiredo and Loureiro (2007). Some specific results of the management scheme is presented in Figueiredo et al. (2005a), and its wider view in under review in Figueiredo et al. (2007a). The first efforts on the Multi solution are published in Figueiredo et al. (2004c), which received the 2nd best paper award in its conference, and Figueiredo et al. (2004b). Extended evaluation of this routing solution and comparisons with other proposals in literature are presented in Figueiredo et al. (2004a). The extension of the Multi solution is under review in Figueiredo et al. (2007b). The RDC-Integrated and RDC-Sync solution are published in Siqueira et al. (2006a) and Siqueira (2006).

During the doctoral process, we have also collaborated with other studies with some insights and implementations. Related to routing solutions, the work by Nakamura et al. (2004) explores the tree routing solutions, and in Nakamura et al. (2005c,b,d) data fusion techniques are applied for routing recovery in WSNs. In a more general way, some helpful data fusion techniques to be applied in WSNs are discussed in Nakamura et al. (2005a). Other work dealing with data reduction techniques based on data stream algorithms is published in de Aquino et al. (2007b), and it has a version under review in de Aquino et al. (2007a).

8.4 Future Work

Through this work it is presented several contributions divided in chapters, and each chapter relates the particular intended activities as future work. In addition, it is also discussed before some limitations present in this work that can be treated as its sequence. Besides doing these improvements to the contributions present in this thesis, we present some general ideas pointing to future directions that can evolve the vision of self-organization in WSNs as follows:

Quantitative comparison between centralized and self-organizing solution. In Chapters 2 and 3, it is discussed that alternative approaches to self-organization are based on central control. While requiring a model of the entire network to perform its organization, these solutions demand more processing and communication reliability than self-organizing ones, but also they can provide either a more efficient or optimized result. It is fact that such centralized solutions can not be feasible for large scale and dynamic networks, but they can serve as a parameter to evaluate the self-organizing solutions. Thus, a future work can consider a methodology in which self-organizing functions are evaluated through a quantitative comparison of the network performance when considering a centralized and optimized solution.

Design of self-organizing evolutive solutions. The proposed solutions present in this work are based on local interactions considering local information received from the neighbors. This is the case of routing infrastructure formation and density control performed by the components of Multi and RDC-Integrated, as well as in other case studies. Also, the network reorganization is performed in rounds, and no information about the achieved network performance is considered between them. A common mechanism present in natural self-organizing systems and in some related work in the network domain is "exploration". With this mechanism, several attempts of a better organization, i.e., which leads to a better performance, are performed by the system elements, and so these elements can change their participation in the network organization or not depending on the attempt result. This behavior can lead to a network whose organization can be improved along time, although it causes resource consumption with unsuccessful attempts. An intention is to develop solutions following this approach, and price-based techniques can be used to evaluate the attempts. A particular solution can consider the density control function, not present in literature following this proposed approach, and existing optimized solution, as described in Chapter 3, can be used as a comparison basis according to the first idea.

Development of a formal method for the design of self-organizing functions. The design methodology proposed in Chapter 4 turns explicit several general aspects of the development of self-organizing functions. These aspects can be present in a formal language to the specification of new self-organizing functions. The advantage of this possible approach is that it can approximate the usage of formal verification, simulation and automatic code generation by considering a uniform way to specify the designed functions.

Bibliography

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cyirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422.

Al-Karaki, J. N. and Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11:6–28.

Arampatzis, T., Lygeros, J., and Manesis, S. (2005). A survey of applications of wireless sensors and wireless sensor networks. In *Proc. of the Mediterranean Control Conference (Med05)*, Limassol Cyprus.

Ashby, W. R. (1947). Principles of the self-organizing dynamic system. *Journal of General Psychology*, 37:125–128.

Balch, T. and Parker, L. E., editors (2002). *Robot Teams: From Diversity to Polymorphism.* AK Peters, Ltd.

Barbosa, V. C. (1996). *An Introdution to Distributed Algorithms.* The MIT Press.

Blazevic, L., Butty, L., Capkun, S., Giordano, S., Hubaux, J.-P., and Boudec, J.-Y. L. (2001). Self-organization in mobile ad-hoc networks: the approach of terminodes. *IEEE Communications Magazine.*

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems.* New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity.

Boulis, A., Han, C.-C., and Srivastava, M. B. (2003). Design and implementation of a framework for efficient and programmable sensor networks. In *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys'03).*

Braginsky, D. and Estrin, D. (2002). Rumor routing algorithm for sensor networks. In *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 22–31, Atlanta, Georgia, USA. ACM Press.

Bulusu, N., Heidemann, J., Estrin, D., and Tran, T. (2004). Self-configuring localization systems: Design and experimental evaluation. *Trans. on Embedded Computing Sys.*, 3(1):24–60.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-Organization in Biological Systems.* Princeton Univ Press.

Capkun, S., Hamdi, M., and Hubaux, J. (2001). Gps-free positioning in mobile ad-hoc networks. In *Proc. of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 9*, page 9008. IEEE Computer Society.

Cardei, M., Maccallum, D., Cheng, X., Min, M., Jia, X., Li, D., zhu Du, D., and Du, D. (2002). Wireless sensor networks with energy efficient organization. *Journal of Interconnection Networks*, 3:213–229.

Caro, G. D., Ducatelle, F., and Gambardella, L. (2005). Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications (ETT), Special Issue on Self Organization in Mobile Networking*, 16(2).

Castro, M., Costa, M., and Rowstron, A. (2004). Performance and dependability of structured peer-to-peer overlays. In *Proc. of the International Conference on Dependable Systems and Networks*, Firenze, I. IEEE CS Press.

Cerpa, A. and Estrin, D. (2002). Ascent: Adaptive self-configuring sensor networks topologies. In *Proc. of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, volume 3, pages 1278–1287, New York, NY, USA.

Cetintemel, U., Flinders, A., and Sun, Y. (2003). Power-efficient data dissemination in wireless sensor networks. In *Proc. of the 3rd ACM international workshop on Data engineering for wireless and mobile access (MobiDe'03)*, pages 1–8, New York, NY, USA. ACM Press.

CHAE, D.-H., HAN, K.-H., LIM, K.-S., AHN, S.-Y., and AN, S.-S. (2006). A localized route discovery for on-demand routing protocols in event-driven wireless sensor networks. *IEICE TRANSACTIONS on Communications*, E89-B.

Chen, B., Jamieson, K., Balakrishnan, H., and Morris, R. (2002). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks*, 8(5):481–494.

Chen, W., Singh, S., and Jain, N. (1999). Anmp: Ad hoc network management protocol. *IEEE JSAC*, 17(8):1506.

Ciancarini, P. (2001). Multiagent coordination: A computer science perspective. In *Proc. of the Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'01).*

Ciancarini, P., Omicini, A., and Zambonelli, F. (2000). Multiagent system engineering: The coordination viewpoint. *Intelligent Agents VI: Agent Theories, Architectures and Languages*, 1757:250–259.

Clare, L. P., Pottie, G. J., , and Agre, J. R. (1999). Self-organizing distributed sensor networks. In *Proc. of the SPIE Conf. on Unattended Ground Sensor Technologies and Applications*, pages 229–237.

Collier, T. C. and Taylor, C. E. (2004). Self-organization in sensor network. *Journal of Parallel and Distributed Computing*, 64(7):866–873.

Correia, L. H. A., Macedo, D. F., Silva, D. A. C., dos Santos, A. L., Loureiro, A. A. F., and Nogueira, J. M. S. (2005). Transmission power control in mac protocols for wireless sensor networks. In *Proc. of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '05)*, pages 282–289, New York, NY, USA. ACM Press.

Crossbow Technology Inc. (2004). Mica2 - wireless measurement system. [Online] Available: http://www.xbow.com/.

CrossBow Technology Inc. (2004). Mote in-network programming user reference. [Online] Available: http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/xnp.pdf.

Damianou, N., Dulay, N., Lupu, E., and Sloman, M. (2001). The ponder policy specification language. In *Proc. of the International Workshop on Policies for Distributed Systems and Networks (POLICY '01)*, pages 18–38, London, UK. Springer-Verlag.

DCC/UFMG (2006). Sensornet project - redes de sensores sem fio. [Online] Available: http://www.sensornet.dcc.ufmg.br.

de Aquino, A. L., Figueiredo, C. M., Nakamura, E. F., Buriol, L. S., and Loureiro, A. A. (2007a). Real time sensor networks applications design for data reduction algorithms. In *IEEE Symposium on Computers and Communications (ISCC'07) (Under review).*

de Aquino, A. L., Figueiredo, C. M., Nakamura, E. F., Buriol, L. S., Loureiro, A. A., Fernandes, A. O., and Jr., C. J. C. (2007b). A sampling data stream algorithm for wireless sensor networks. In *Proc. of the IEEE International Conference on Communications (ICC'07) (Accepted).*

de Oliveira, H. A. B. F., Nakamura, E. F., and Loureiro, A. A. F. (2005). Directed position estimation: A recursive localization approach for wireless sensor networks. In *Proc. of the 14th IEEE International Conference on Computer Communications and Networks (ICCCN'05)*, San Diego, USA.

Ding, J., Sivalingam, K., Kashyapa, R., and Chuan, L. J. (2003). A multi-layered architecture and protocols for large-scale wireless sensor networks. In *Proc. of the 58th IEEE Semiannual Vehicular Technology Conference (VTC 2003-Fall)*, volume 3, pages 1443–1447.

Dorigo, M., Tuci, E., Groß, R., Trianni, V., Labella, T., Nouyan, S., Ampatzis, C., Deneubourg, J.-L., Baldassarre, G., Nolfi, S., Mondada, F., Floreano, D., and Gambardella, L. (2004). The SWARM-BOTS project. In E. Şahin and Spears, W., editors, *Proc. of the First International Workshop on Swarm Robotics at SAB 2004*, volume 3342 of *Lecture Notes in Computer Science*, pages 31–44. Springer Verlag, Berlin, Germany.

Dulay, N., Lupu, E., Sloman, M., and Damianou, N. (2001). A Policy Deployment Model for the Ponder Language. In *Proc. of IEEE/IFIP Integrated network management*, pages 529–544.

Dust, S. (2006). Smart Dust: Autonomous sensing and communication in a cubic millimeter. `http://robotics.eecs.berkeley.edu/Epister/SmartDust`.

Estrin, D., Govindan, R., Heidemann, J., and Kumar, S. (1999). Next century challenges: Scalable coordination in sensor networks. In *Proc. of the 5th Annual International Conference on Mobile Computing and Networks (MobiCom'99)*, pages 263–270, Seattle, Washington, USA. ACM Press.

Figueiredo, C., dos Santos, A., Loureiro, A., and Nogueira, J. (2005a). Policy-based adaptive routing in autonomous wsns. In *Proc. of the 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM2005)*, Barcelona, Spain.

Figueiredo, C., Nakamura, E., and Loureiro, A. (2004a). Avaliacão de desempenho de algoritmos de disseminacão de dados para redes de sensores sem fio. In *Anais do VI Workshop de Comunicacão sem Fio e Computacão Móvel (WCSF'04)*, Fortaleza, CE, Brazil.

Figueiredo, C., Nakamura, E., and Loureiro, A. (2004b). Multi: A hybrid adaptive dissemination protocol for wireless sensor networks. In *Proc. of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors 2004)*, volume 3121 of *Lecture Notes in Computer Science*, pages 171–186, Turku, Finland. Springer.

Figueiredo, C., Nakamura, E., and Loureiro, A. (2004c). Protocolo adaptativo híbrido para disseminacão de dados em redes de sensores sem fio auto-organizáveis. In *Anais do 22o Simpósio Brasileiro de Redes de Computadores*. 2o melhor artigo.

Figueiredo, C., Nakamura, E., and Loureiro, A. (2005b). *Algorithms and Protocols for Wireless and Mobile Networks*, chapter Self-organization algorithms for wireless networks. Chapman & Hall/CRC Press.

Figueiredo, C. M. and Loureiro, A. A. (2007). A design methodology for self-organizing wireless sensor networks. In *IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07) (Under Review)*.

Figueiredo, C. M., Loureiro, A. A., and Ruiz, L. B. (2007a). A management scheme for self-organizing wsns: Acting over local rules. In *IEEE Symposium on Computers and Communications (ISCC'07) (Under review)*.

Figueiredo, C. M., Nakamura, E. F., and Loureiro, A. A. (2007b). An event-detection estimation model for hybrid adaptive routing in wireless sensor networks. In *Proc. of the IEEE International Conference on Communications (ICC'07) (Accepted)*.

Flake, G. W., Lawrence, S., Giles, C. L., and Coetzee, F. M. (2002). Self-organization and indentification of web communities. *IEEE Computer*, 35(3):66–71.

Frank, C. and Römer, K. (2005). Algorithms for generic role assignment in wireless sensor networks. In *Proc. of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, CA, USA.

Ganeriwal, S., Kumar, R., and Srivastava, M. B. (2003). Timing-sync protocol for sensor networks. In *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 138–149, New York, NY, USA. ACM Press.

Goussevskaia, O., Machado, M., Mini, R., Loureiro, A., Mateus, G., and Nogueira, J. (2005). Data dissemination based on the energy map. *IEEE Communications Magazine*, 43(7):134–143.

Haas, Z. J. and Pearlman, M. R. (1998). The performance of query control schemes for zone routing protocol. In *Proc. of the ACM SIGCOMM'98 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication*, pages 167–177, Vancouver, Canada.

Haken, H. (1983). *Synergetics: An Introduction. Nonequilibrium Phase Transition and Self-Organization in Physics, Chemistry, and Biology, Third Revised and Enlarged Edition.* ISBN 0387123563.

Hedetniemi, S. and Liestman, A. (1988). A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349.

Heidemann, J., Silva, F., and Estrin, D. (2003). Matching data dissemination algorithms to application requirements. In *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pages 218–229, Los Angeles, CA, USA. ACM Press.

Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the IEEE Hawaii International Conference on System Sciences*, pages 4–13.

Heylighen, F. (2002). The science of self-organization and adaptivity. *The Encyclopedia of Life Support Systems - EOLSS Publishers.*

Heylighen, F. (2005). Principia sybernetica web: Self-organization. [Online] Available: http://pespmc1.vub.ac.be/SELFORG.html.

Heylighen, F. and Gershenson, C. (2003). The meaning of self-organization in computing. *IEEE Intelligent Systems, Trends and Controversies. Self-Organization and Information Systems.*

IEEE (2004). Ieee 802.15 working group for wireless personal area networks. [Online] Available: http://ieee802.org/15/index.html.

Jelasity, M., Babaoglu, O., Laddaga, R., Nagpal, R., Zambonelli, F., Sirer, E. G., Chaouchi, H., and Smirnov, M. (2006). Interdisciplinary research: Roles for self-organization. *IEEE Intelligent Systems*, 21(2):50–58.

Karp, B. and Kung, H. T. (2000). Gpsr: greedy perimeter stateless routing for wireless networks. In *Proc. of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM Press.

Kauffman, S. (1993). *Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press.

Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50.

Kephart, J. O. and Walsh, W. E. (2004). An artificial intelligence perspective on autonomic computing policies. In *Proc. of the 5th International Workshop on Policies for Distributed Systems and Networks.*

Kochhal, M., Schwiebert, L., and Gupta, S. (2003). Role-based hierarchical self-organization for wireless ad hoc sensor networks. In *Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pages 98–107. ACM Press.

Krishnan, R. and Starobinski, D. (2003). Message-efficient self-organization of wireless sensor networks. In *IEEE WCNC 2003*, pages 1603–1608.

Kwon, T. and Gerla, M. (2002). Efficient flooding with passive clustering (pc) in ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 32:44–56.

Levis, P. and Culler, D. (2002). Maté: a tiny virtual machine for sensor networks. In *Proc. of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, pages 85–95. ACM Press.

Levis, P., Patel, N., Culler, D. E., and Shenker, S. (2004). Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proc. of the NSDI'04*, pages 15–28.

Loureiro, A. A. F., Nogueira, J. M. S., Ruiz, L. B., de Freitas Mini, R. A., Nakamura, E. F., and Figueiredo, C. M. S. (2003). Redes de sensores sem fio. In *Anais do 21o Simpósio Brasileiro de Redes de Computadores (SBRC'03)*, pages 179 – 226, Natal, RN, Brazil.

Lynden, S. and Rana, O. (2002). Coordinated learning to support resource management in computational grids. In Society, I. C., editor, *Proc. of the 2nd IEEE Conference on Peer-to-Peer Computing (P2P 2002)*, pages 81–89.

Macedo, D. F., Correia, L. H. A., dos Santos, A. L., Loureiro, A. A. F., and Nogueira, J. M. S. (2005a). A pro-active routing protocol for continuous data dissemination in wireless sensor networks. In *Proc. of the 10th IEEE Symposium on Computers and Communications (ISCC'05)*, pages 361–366, Washington, DC, USA. IEEE Computer Society.

Macedo, D. F., Correia, L. H. A., dos Santos, A. L., Loureiro, A. A. F., and Nogueira, J. M. S. (2006). A rule-based adaptive routing protocol for continuous data dissemination in wsns. *J. Parallel Distrib. Comput.*, 66(4):542–555.

Macedo, D. F., dos Santos, A. L., Correia, L. H. A., Nogueira, J. M., and Loureiro, A. A. F. (2005b). Taxonomy for medium access control protocols for wireless sensor networks. *Annals of Telecommunications*, 60(7–8):813–838.

Mamei, M. and Zambonelli, F. (2004). Programming pervasive and mobile computing applications with the tota middleware. In *Proc. of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004)*, pages 263–276.

Menezes, G., de Aquino, A., da S. CABRAL, R., and Nakamura, F. (2005). Uma abordagem paralela para os problemas de cobertura e conectividade em redes de sensores sem fio. In *Anais do Simpósio Brasileiro de Pesquisa Operacional (SBPO'2005).*

Microsoft Inc. (2005). Dynamic systems initiative overview. [Online] Available: http://www.microsoft.com/windowsserversystem/dsi/dsioverview.mspx.

Millennial Net (2006). Millennial net - wireless sensor networks. [Online] Available: http://www.millennial.net.

μAMPS (2006). μAMPS Projet. `http://www-mtl.mit.edu/research/icsystems/uamps`.

Naik, P. and Sivalingam, K. M. (2004). *Wireless sensor networks*, chapter A survey of MAC protocols for sensor networks, pages 93–107. Kluwer Academic Publishers, Norwell, MA, USA.

Nakamura, E., Figueiredo, C., and Loureiro, A. (2005a). *Algorithms and Protocols for Wireless and Mobile Networks*, chapter Information fusion algorithms for wireless sensor networks. Chapman & Hall/CRC Press.

Nakamura, E., Nakamura, F., Figueiredo, C., and Loureiro, A. (2005b). Fusão de dados como suporte à disseminacão de dados em redes de sensores sem fio auto-organizáveis. In *Anais do 23o Simpósio Brasileiro de Redes de Computadores (SBRC'05).*

Nakamura, E. F., Figueiredo, C. M., and Loureiro, A. A. (2004). Disseminassão de dados adaptativa em redes de sensores sem fio auto-organizáveis. In *Anais do 22o Simpósio Brasileiro de Redes de Computadores*, Gramado, RS, Brazil.

Nakamura, E. F., Figueiredo, C. M., and Loureiro, A. A. (2005c). Information fusion for data dissemination in self-organizing wireless sensor networks. In *Proc. of the 4th International Conference on Networking (ICN 2005)*, Reunion Island.

Nakamura, E. F., Nakamura, F. G., Figueredo, C. M., and Loureiro, A. A. (2005d). Using information fusion to assist data dissemination in wireless sensor networks. *Telecommunication Systems*, 30(1-3):237–254.

Nakamura, F., Quintão, F., Menezes, G., and Mateus, G. (2005e). An optimal node scheduling for flat wireless sensor networks. In *Proc. of the 4th International Conference on Networking (ICN'2005)*, volume 3420 of *Lecture Notes in Computer Science*, pages 475–483.

Nakamura, F. G. (2003). Dynamic planning for coverage and connectivity control in plain wireless sensor networks (in portuguese). Master's thesis, Computer Science Department of the Federal University of Minas Gerais, Belo Horizonte, MG, Brazil.

NS-2 (2004). The network simulator - ns-2. [Online] Available: http://www.isi.edu/nsnam/ns/.

Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad-hoc on-demand distance vector routing. RFC 3561.

Perkins, C. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM'94)*, pages 234–244.

Polastre, J., Hill, J., and Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proc. of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 95–107, New York, NY, USA. ACM Press.

Pottie, G. J. and Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58.

Prehofer, C. and Bettstetter, C. (2005). Self-organization in communication networks: Principles and design paradigms. *IEEE Communication Magazine*, 43(7):78–85.

Quintão, F., Nakamura, F., and Mateus, G. (2005). Algoritmo evolutivo para o problema dinâmico de cobertura aplicado ao gerenciamento de redes de sensores sem fio. In *Anais do Simpósio Brasileiro de Pesquisa Operacional (SBPO'2005)*.

Quintão, F. P., Mateus, G. R., and Nakamura, F. G. (2004). An evolutive approach for the coverage problem in wireless sensor networks (in portuguese). In *Proc. of the 24nd Brazilian Computer Science Society Congress (SBC'04)*, pages 440–450. 1st Place at the 23rd Undergraduate Scientific Research Contest (CTIC'04).

Ramamoorthy, C. V., Bhide, A., and Srivastava, J. (1987). Reliable clustering techniques for large, mobile packet radio networks. In *Proc. of the 6th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '87)*, San Francisco, CA, USA.

Ramasubramanian, V., Haas, Z., and Sirer, E. (2003). Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks. In *Proc. of the 4th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pages 303–314.

Ruiz, L., Braga, T., Silva, F., Assuncao, H., Nogueira, J., and Loureiro, A. (2005). On the design of a self-managed wireless sensor network. *IEEE Communication Magazine*, 43:95–102.

Ruiz, L., Nogueira, J., and Loureiro, A. (2003). Manna: A management archicteture for wireless sensor networks. *IEEE Communicaton Magazine*, pages 116–125.

Ruiz, L. B., Correia, L. H. A., Vieira, L. F. M., Macedo, D. F., Nakamura, E. F., Figueiredo, C. M. S., Vieira, M. A. M., Maia, E. H. B., Câmara, D., Loureiro, A. A. F., Nogueira, J. M. S., da Silva Jr., D. C., and Fernandes, A. O. (2004). Arquiteturas de redes de sensores sem fio. In *Anais do 22o Simpósio Brasileiro de Redes de Computadores*, pages 167–218, Gramado, RS, Brazil. Tutorial. ISBN: 85-88442-82-5.

Russel, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall.

Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, M. (2002). Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80.

Shen, C.-C., Srisathapornphat, C., and Jaikaeo, C. (2003). An adaptive management architecture for ad hoc networks. *IEEE Communications Magazine*, 41(2):108–115.

Sibley, G., Rahimi, M., and Sukhatme, G. (2002). "robomote: A tiny mobile robot platform for large-scale sensor networks. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA2002)*.

Sichitiu, M. L. (2004). Cross-layer scheduling for power efficiency in wireless sensor networks. In *Proc. of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*.

Siqueira, I., Figueiredo, C., Loureiro, A., Nogueira, J., and Ruiz, L. (2006a). An integrated approach for density control and routing in wireless sensor networks. In *Proc. of the 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006)*.

Siqueira, I., Ruiz, L., Loureiro, A., and Nogueira, J. (2006b). Coverage area management for wireless sensor networks. *International Journal Of Network Management*, 1.

Siqueira, I. G. (2006). Projeto integrado de controle de densidade e roteamento em redes de sensores sem fio. Master's thesis, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais.

Slijepcevic, S. and Potkonjak, M. (2001). Power efficient organization of wireless sensor networks. In *Proc. of the IEEE International Conference on Communications (ICC'01)*, volume 2, pages 472–476, Helsinki, Finland.

Sloman, M. (1994). Policy driven management for distributed systems. *Plenum Press Journal of Network and Systems Management*, 2(4):333–360.

Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G. (2000). Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27.

Sun Microsystems Inc. (2005). N1 grid - introducing just in time computing. [Online] Available: http://www.sun.com/software/n1gridsystem/docs.xml.

Tian, D. and Georganas, N. D. (2002). A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 32–41, Atlanta, GA, USA. ACM Press.

Tilak, S., Abu-Ghazaleh, N. B., and Heinzelman, W. (2002). A taxonomy of wireless micro-sensor network models. *ACM Mobile Computing and Communications Review (MC2R)*, 6(2):28–36.

TinyOS (2004). Tinyos - a component-based os for the networked sensor regime. [Online] Available: http://www.tinyos.net.

Vieira, L. F. M. (2004a). Middleware para sistemas embutidos e rede de sensores. Master's thesis, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte-MG, Brasil.

Vieira, M. A. M. (2004b). Embedded system for wireless sensor network. Master's thesis, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte-MG, Brasil.

Vieira, M. A. M., Vieira, L. F. M., Ruiz, L. B., Loureiro, A. A. F., Fernandes, A. O., and Nogueira, J. M. S. (2003). Scheduling nodes in wireless sensor network: A voronoi approach. In *Proc. of the 28th Annual IEEE Local Computer Networks (LCN'03)*, pages 423–429, Bonn/Königswinter, Germany.

Walters, J. P., Liang, Z., Shi, W., and Chaudhary, V. (2005). Wireless sensor networks security: A survey. Technical report, Wayne State University, USA.

Wang, Q., Zhu, Y., and Cheng, L. (2006). Reprogramming wireless sensor networks: challenges and approaches. *IEEE Network*, 20(3):48–55.

Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., and Gill, C. (2003). Integrated coverage and connectivity configuration in wireless sensor networks. In *Proc. of the 1st ACM Internation Conference on Embedded Networked Sensor Systems (SenSys'03)*, pages 28–39, Los Angeles, California, USA. ACM Press.

Welsh, M. and Mainland, G. (2004). Programming sensor networks using abstract regions. In *Proc. of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004)*.

Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., and Waldbusser, S. (2001). Rfc 3198: Terminology for policy-based management.

Whitehouse, K., Sharp, C., Brewer, E., and Culler, D. (2004). Hood: A neighborhood abstraction for sensor networks. In *Proc. of the International Conference on Mobile Systems, Applications, and Services (MOBYSYS 2004)*.

WINS (2006). Wireless Integrated Network Sensors (WINS). `http://www.janet.ucla.edu/WINS/`.

Wolfram, S. (1986). *Theory and Applications of Cellular Automata*. World Scientific.

Woo, A., Tong, T., and Culler, D. (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. of the 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 14–27. ACM Press.

Xu, Y., Bien, S., Mori, Y., Heidemann, J., and Estrin, D. (2003). Topology control protocols to conserve energy in wireless ad hoc networks. In *CENS Technical Report 0006*.

Ye, W., Heidemann, J., and Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks. In *Proc. of the IEEE Infocom'02*, pages 1567–1576, New York, NY, USA. USC/Information Sciences Institute, IEEE.

Zambonelli, F., Gleizes, M., Mamei, M., and Tolksdorf, R. (2005). Spray computers: Explorations in self-organization. *Journal of Pervasive and Mobile Computing*, 1(1):1–20.

Zhang, H. and Hou, J. C. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *International Journal of Wireless Ad Hoc and Sensor Networks*, 1(1–2):89–124.

Zhou, C. and Krishnamachari, B. (2003). Localized topology generation mechanisms for self-configuring sensor networks. In *Proc. of the IEEE Wireless Communication Symposium (Globecom 2003)*, San Francisco, USA.

Zorzi, M. (2004). A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks. In *Proc. of the IEEE Internation Conference on Communications (ICC 2004)*, volume 27, pages 3481–3485.