Linnyer Beatrys Ruiz

MANÁ: Uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio

Tese apresentada ao Curso 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.

Universidade Federal de Minas Gerais
Dezembro de 2003

Ao MANNarido

Camillo

aos MANNApais

Sidney e Julita

e aos MANNamigos

Edênia, Kalina, Thais, Fabrício, Rainer e Claudine,

com carinho...

Agradecimentos

Meu precioso Jesus: “Agradeço pela força, refúgio, consolo e pelo Maná de todos os dias. Sou grata pela vida e amigos abundantes. Obrigado pela excelência!”

Meu orientador e amigo José Marcos Silva Nogueira: “Agradeço pelo incentivo, dedicação e exemplo. Sou grata pela oportunidade de trabalhar com você e honrada por tê-lo tido como meu orientador”.

Ao Prof. Antonio Alfredo Ferreira Loureiro: “Sou grata por sua amizade, paciência, dedicação, ensinamentos e pelo “sim” para este trabalho”.

Ao MANNA Racing Team: Thais, Fabrício e Isabela. “Agradeço a honra de tê-los como amigos e pela dedicação de todos os dias (isto inclui os sábados e domingos)”. Thadeu e Mauro Jr., quero agradecer o apoio. Aos irmãos Flip e Flop pelas emoções dos últimos anos. À “amiga-chefe” Ana Paula Silva quero agradecer pela alegria, serenidade e por tantas coisas boas que pude aprender com você.

À PUCPR: “Agradeço pelo apoio nesta empreitada”. Sou grata ao Prof. Flávio Bortolozzi, Prof. Robert C. Burnett, Prof. Celso A . Kaestner, Prof. Julio Nievola, Prof. Chu Chia Gean, Prof. Manoel Camillo Penna, Prof. Josuê, Prof ª. Maria Julia, Prof ª. Avani, Roberto, Alice, ao pessoal do DRH e da secretaria de exatas. Meu muito obrigado ao Wilson do HSBC agência PUC.

Ao CNPq: “Agradeço ao Governo Federal pelo incentivo”.

Aos professores Liane Tarouco (UFRG), Elias Procópio Duarte Jr.(UFPR) e Geraldo Robson Mateus (UFMG): “Agradeço pelas contribuições ao projeto desta tese”.

Ao Prof. Mario F. Montenegro Campos (UFMG): “Bendito o que vem em nome do Senhor!”.

Aos professores Claudionor N. Coelho Junior (UFMG) e Henrique Pacca L. Luna (UFAL): “Agradeço as lições de empreendedorismo, liderança, qualidade de vida e determinação”.

Ao Prof. Carlos Alberto Malcher Bastos (UFF): “Grata pelo apoio, atenção, oportunidades e direções apontadas”.

Ao Prof. Mehmet Ulema (Manhatan College): “Minha gratidão pelo incentivo a este trabalho”.

Às professoras Ana Paula Olegário e Susan Strickland: vocês são preciosas!

À UFMG: sou grata pela oportunidade de ter convivido com pessoas cujo exemplo não poderei esquecer. Agradeço esta instituição na pessoa da Túlia, Renata, Emília, Gilberto,

Cida, Helvécio, Belkiz, Lizete, Luciana(s), Sônia, Gilmara, Sheila, Claudinha, Gustavo, Alexandre(s) e aos professores, Clarindo de Pádua, Arnaldo A. Araújo, Wong Hao Chi, Antônio O. Fernandes, Alberto H. F. Laender, Newton José Vieira, José Monteiro da Mata e Antônio Mendes.

Às minhas amigas Kalina, Pati Correa, Pati Aguiar, Ana Luiza Bessa, Daniela Alvim, Christiane Marie, Alessandra Bomura, Adriane Loper, Sandra Calixto, Ana Carolina Fortes e Elaine Pimentel: muitas e muitas vezes vocês me surpreenderam. Dei muito trabalho mas Deus deu-me vocês. Obrigado. "Em todo tempo ama o amigo e na angústia se faz o irmão" (Provérbios 17:17).

Às minhas amigas Claudine e Edênia: "...mas há um amigo mais chegado do que um irmão" (Prov 18:24). Sou grata porque Deus as usou para soprar sobre mim como vento e tirar toda cinza. "E soprou o vento do Espírito Santo e o fogo recomeçou"!

Às minhas irmãs (Jô e Kennya) e meu cunhado (Paulo): "Agradeço pelas orações, telefonemas e palavras de apoio".

Aos meus sobrinhos (Natália, Gabriel e Pedro): uma geração de adoradores ultra jovens capazes de mudar a história de uma família. Um trio sensacional!

Às minhas queridas Laíze, Lara, Ana(s), Silmara e Heloice: "Sou grata por participarem da minha família".

Ao Vô Mané, Vó Lady e Vó Loura: "Agradeço os conselhos, carinho e atenção".

Ao meu pais (Sidney e Julita): "Tem que dar certo, bicho do Paraná!". E deu! Amo vocês.

À família "Ruiz e Silva" por me fazerem rir de tanto chorar e chorar de tanto rir! Obrigado Tuti e Tatu...

Artur e Vera: sou grata pelas lições de amor e cuidado. Cirino, Vânia, Lisnara e Edson: agradeço por orarem pelos meus sonhos e darem cobertura às minhas decisões. Erasmo, Cláudia, Léia e Josuel: sou grata por me receberem como família em BH. Waldomiro e Graça Piza obrigado pelo incentivo, adoção e carinho.

D. Elisa, Sônia, Sandra, Jivago, Hiago e Elise: Grata pelo amor incondicional!

Aos amigos Sylvie, Silvio Jamil, Tati, Cristiano, Aletéia, Xandi, Décio, Deffo, Raquelzinha, Henri, Aline, SergioOl, Carlos Frederico, Ruiter, Hervaldo, Lúcio (ATM), Sica, Romanelli, Lucila, Wesley, Benício, Júnia, Elenice, Renan, FePaixão, Marciocm, Valdeci, Laudares, Polai, Lula, Kissia, Gabi, Rêmulo, Emanoel, Valdo, Marcelo, Melissa, Simone, Umberto, Teixeira, Rafael Kelles, Loius, Paulo Marinheiro Paraíba, Guilherme Pereira, Andreia Iabrudi, Dilu, Damacedo, Pável, Autran, Lizandro Grenville e Luciano Gaspary: grata pelo carinho, atenção, piadas, discursos, etc. Momentos preciosos para mim.

Ao grupo tBb de Redes de Sensores: crescemos em progressão geométrica!!! Obrigado

Wagner,...

Ao amigo Manoel Palhares: amigos assim, queremos aos montes. Obrigado por dividir sua família conosco. Acredite, isso tornou nossa chegada mais suave.

Ao amigo e "suplente": José Pio. Grata pelos preciosos conselhos, pela amizade, pela generosidade e pelo cuidado que dedicou a mim e ao meu marido.

Aos amigos Rainer e Rabelo: eh! Uh! Hum!

Aos amigos do Coreu (Bairro Coração Eucarístico): Marcelo, Hugo, Cosme, Kelly,...Obrigado.

Ao meu MANNArido Camillo: "La vie en rose!"

Aos meus filhos que ainda não nasceram: saibam que já são amados, preciosos e queridos!

Resumo Estendido

O documento desta tese foi originalmente redigido em inglês com título: "*MANNA: A Management Architecture for Wireless Sensor Networks*". Para estar em conformidade com as normas da Universidade Federal de Minas Gerais, este resumo em português faz uma exposição abreviada de cada um dos capítulos que compõe esta tese.

Resumo do Capítulo 1 – Introdução

Um dos objetivos desta tese é estudar o problema do gerenciamento em Redes de Sensores Sem Fio (RSSFs), procurando entender as necessidades, os requisitos e as questões relacionadas ao tema e identificando as diferenças em relação ao gerenciamento de outras redes. Delinear soluções de gerenciamento para RSSFs é o objetivo principal.

RSSFs é um tema recente de pesquisa e que se encontra na fronteira tecnológica. Até o momento, não foram encontrados na literatura trabalhos que proponham uma solução de gerenciamento para tais redes ou discutam as diferenças entre o gerenciamento de redes tradicionais e o gerenciamento de RSSFs.

Tomando como pressuposto básico que o gerenciamento de RSSFs deve ser simples, aderente às peculiaridades dessas redes, incluindo também o seu dinamismo, e eficaz no uso dos recursos escassos, esta tese propõe um arcabouço para o gerenciamento de tais redes.

O arcabouço proposto introduz uma organização baseada em três dimensões de gerenciamento. Duas dessas dimensões, áreas funcionais e níveis de gerenciamento, têm sido usadas no gerenciamento tradicional e são redefinidas sob a perspectiva das RSSFs. A terceira dimensão, chamada "funcionalidades de RSSFs" é proposta por esta tese. Uma lista de funções de gerenciamento é estabelecida a partir dessa organização tridimensional

que também é útil na definição da informação e no desenvolvimento de serviços e aplicações de gerenciamento.

O arcabouço proposto inclui uma arquitetura de gerenciamento chamada MANNA[1]. Esta arquitetura é baseada no paradigma de computação autonômica [40] que permite definir soluções de auto-gerenciamento (*self-management*). A arquitetura MANNA propõe que as RSSFs sejam auto-gerenciadas e que para isto utilizem funções e serviços automáticos, isto é, executados com o mínimo de interferência humana. O principal objetivo da arquitetura de gerenciamento é promover a produtividade dos recursos e a qualidade dos serviços providos.

Não é objetivo desta tese a implementação de um sistema de gerenciamento. Todavia, com o propósito de mostrar como o arcabouço, em particular a arquitetura MANNA, pode ser aplicado e atingir seus objetivos, uma solução de gerenciamento foi construída para uma aplicação definida como estudo de caso.

Como contribuições, esta tese apresenta:

- Um arcabouço de gerenciamento para RSSFs que provê a distinção entre funcionalidades de gerenciamento e funcionalidades da aplicação. Este arcabouço inclui: uma arquitetura de gerenciamento chamada MANNA, que é construída a partir de uma arquitetura de informação, uma arquitetura funcional e uma arquitetura física (cada uma dessas sub-arquiteturas relacionadas a diferentes aspectos da solução de gerenciamento) e uma organização tridimensional para o gerenciamento de RSSFs. Duas dessas dimensões, áreas funcionais e níveis de gerenciamento são comuns no gerenciamento de outras redes mas foram redefinidas sob a perspectiva de RSSFs. A terceira dimensão "funcionalidades de RSSFs" é também uma proposta desta tese. Esta nova dimensão é baseada no modelo funcional e na caracterização das RSSFs que também

[1]MANNA - tradução para a língua inglesa do termo Maná. A palavra MANNA tem origem no hebraico *Man hu*: pão do céu ou "Que é isto?" - alimento fornecido por Deus aos israelitas durante 40 anos no deserto. Era como uma semente miúda que era lançada do céu e cobria a terra como geada. Tinha gosto de bolo de mel (Êxodos 16:31). Sinal de confiança. Coisa excelente, vantajosa. [Novo Aurélio Século XXI: o dicionário da língua portuguesa. Aurélio Buarque de Holanda Ferreira, 3a. Edição, Editora Nova Fronteira, 1999].

são contribuições desta tese;

- Uma abordagem original para desenvolver serviços e aplicações de gerenciamento considerando diferentes tipos de RSSFs;
- Um esquema para definição de funções de gerenciamento de RSSFs, assim como uma lista de funções de gerenciamento e serviços que podem ser executados de forma automática, semi-automática ou manual;
- A proposição de novos modelos para representação dos estados de RSSFs;
- Um modelo funcional que permite caracterizar as RSSFs;
- Um modelo genérico de informação;
- Um arcabouço para simulação de RSSFs chamado MANNASim construído a partir da ferramenta Network Simulator (NS-2) e que permite o desenvolvimento de aplicações de RSSFs;
- A aplicação dos conceitos do paradigma de auto-gerenciamento (*self-management*) em RSSFs.

Algumas das contribuições e resultados deste trabalho foram publicados em paralelo ao seu desenvolvimento [52, 53, 54, 79, 80, 81, 82, 83, 84, 85, 86, 92, 93, 103, 104] e outros foram submetidos e estão em processo de avaliação. Em alguns casos por uma questão de organização, os artigos e documentos apresentam mais detalhes do que os contidos na tese.

Construir partes do sistema e desenvolver soluções para problemas específicos em RSSFs não são tarefas triviais. Integrar estas partes ou soluções de maneira a promover a produtividade da rede e a qualidade dos serviços providos é um desafio ainda maior. Esta tese trata deste desafio.

É notável o progresso da área de RSSFs. Contudo, quando este trabalho teve início, havia poucas publicações sobre o assunto e muitos resultados de pesquisa não estavam disponíveis. Este cenário impôs muitos desafios e dificuldades a serem superados que

foram vistos como oportunidades de pesquisa. Era necessário entender as RSSFs e suas características especiais para então propor soluções de gerenciamento e avaliar como estas soluções contribuiriam para o funcionamento da rede.

O texto da tese está organizado da seguinte maneira. O capítulo 2 apresenta uma visão geral sobre RSSFs e nós sensores sem fio, trata das principais diferenças entre as RSSFs e outras redes, e introduz algumas aplicações e trabalhos relacionados. O capítulo 3 discute os desafios do gerenciamento de RSSFs e propõe uma organização para o gerenciamento baseada em três dimensões: áreas funcionais, níveis de gerenciamento e funcionalidades. A arquitetura MANNA é apresentada no capítulo 4. Este capítulo propõe um esquema para definição de funções de gerenciamento e um outro esquema para definição de serviços e aplicações de gerenciamento usando a arquitetura MANNA. O capítulo também apresenta as sub-arquiteturas da MANNA: informação, funcional e física. No capítulo 5, uma aplicação para monitoração da qualidade do ar usando RSSFs é desenvolvida como estudo de caso. Alguns serviços e funções de gerenciamento são implementados com objetivo de prover soluções de gerenciamento para esta aplicação e mostrar como a arquitetura proposta cumpre seus objetivos. O capítulo 5 também apresenta um arcabouço para simulação de RSSFs chamado MANNASim. O capítulo 6 conclui a tese discutindo os resultados obtidos e os trabalhos futuros. Um apêndice apresenta um estudo de caso de RSSFs autonômicas utilizando o serviço de auto-diagnóstico.

Resumo do Capítulo 2 - Redes de Sensores Sem Fio

O capítulo 2 propõe um estudo sobre RSSFs, incluindo uma visão geral de suas características básicas na seção 2.1. RSSFs têm sido viabilizadas pela rápida convergência de três tecnologias: circuitos integrados, comunicação sem fio e micro sistemas eletro-mecânicos. Uma RSSF pode ser usada para monitorar e controlar um ambiente. Este tipo de rede tende a ser formada por centenas ou milhares de dispositivos autônomos chamados nós sensores.

Os nós de uma RSSF podem ser lançados sobre áreas remotas (reservas ambientais,

oceanos, vulcões, rios, florestas, etc.) e sem a intervenção de técnicos ou operadores formam uma rede sem fio *ad hoc* (ver figura 2.1) que coleta dados sobre os fenômenos de interesse, realiza processamento local, e dissemina as informações para um ponto de acesso através do qual a rede comunica-se com outras redes ou com usuários.

Algumas características especiais das RSSFs são:

- Fluxo de dados predominantemente unidirecional: os dados são disseminados dos nós sensores em direção ao ponto de acesso utilizando nós intermediários como roteadores (ver figura 2.2). Em RSSFs que utilizam rádio freqüência para transmissão, a atividade de maior consumo de energia é a transmissão de dados. A energia consumida com a transmissão via rádio pode variar com o quadrado do alcance de transmissão. Uma forma de economizar energia é se utilizar alcance pequenos de transmissão. Assim, os nós coletam dados e se usam os nós intermediários para retransmissão desses dados até a entrega ao ponto de acesso;

- Topologia dinâmica: mesmo que os nós não sejam móveis, eles podem ocasionar alterações na topologia quando saem de serviço por problemas tais como quebras e defeitos resultantes da deposição, falta de energia, ameaças e ataques à segurança, problemas de calibração dos dispositivos sensores, falhas nos componentes e falhas de comunicação;

- Dependência da aplicação: os parâmetros de configuração, operação e manutenção das RSSFs variam com o tipo de aplicação. Qualquer projeto (hardware ou software) ou solução proposta para estas redes ou seus elementos deve levar em consideração essas características e restrições, assim como as características do ambiente onde tais redes serão aplicadas. Isto determina o desenvolvimento de diferentes arquiteturas de nós sensores ou diferentes configurações para arquiteturas existentes, assim como o desenvolvimento de soluções de gerenciamento compostas por diferentes serviços e funções;

- Grande número de elementos de rede distribuídos em áreas remotas ou inóspitas que

operam sem intervenção humana direta;

- Apresentam restrições severas de energia, e devem possuir mecanismos para auto-configuração e adaptação devido a problemas como falhas de comunicação, variações nas condições ambientais e perda de nós (ver figura 2.3). Para as RSSFs, falhas não são exceções mas acontecimentos comuns;

- Tendem a serem autônomas e requerem um alto grau de cooperação entre os elementos de rede para executar um objetivo comum. Na maioria das aplicações de RSSFs, os elementos de rede executam tarefas comuns enquanto que nas redes tradicionais os elementos executam aplicações diferentes.

- Podem ser organizadas em grupos de nós. Para cada grupo deve existir um nó líder que recebe os dados coletados dos nós comuns, realiza o processamento local e envia as informações resultantes para o ponto de acesso. A comunicação entre o líder e os nós comuns pode ser realizada em multi-saltos ou diretamente (ver figura 2.4).

O projeto de uma RSSF é influenciado por muitos fatores que incluem tolerância a falhas, escalabilidade, custo de produção, ambiente operacional, topologia da rede, restrições de hardware, meio de transmissão e consumo de energia. Cada um destes fatores exige requisitos específicos na concepção e projeto dos nós, assim como em todas as camadas da pilha de protocolos de comunicação.

Estas características tornam as RSSFs diferentes das redes tradicionais. Isto significa que algoritmos distribuídos tradicionais, como protocolos de comunicação e eleição de líder, assim como paradigmas de gerenciamento, devem ser revistos para esse tipo de rede antes de serem usados diretamente. A energia é um recurso crítico e assim, todas as operações executadas na rede devem ser eficientes em energia, incluindo as tarefas de gerenciamento. Outro aspecto que deve ser considerado é que as RSSFs estarão integradas a outras redes como por exemplo a Internet. Uma solução de gerenciamento que separe as funcionalidades, promova a integração das soluções propostas e utilize um modelo genérico de informação pode facilitar o planejamento, desenvolvimento e implementação das RSSFs,

além de promover a produtividade da rede e a qualidade dos serviços providos por ela.

Os elementos de uma RSSFs, os nós sensores, tendem a ser projetados com pequenas dimensões (cm^3 ou mm^3) e esta limitação de tamanho acaba impondo limitações nos recursos de seus componentes quais sejam, unidade de comunicação sem fio (transceptor), unidade de energia, unidade de sensoriamento (compostas por diferentes dispositivos sensores) e unidade de computação (memória e processador) (ver figura 2.5). O componente lógico de um nó sensor é o software que executa no processador. Em alguns casos, uma RSSF também pode ser composta de dispositivos chamados atuadores que permitem ao sistema controlar parâmetros do ambiente monitorado [93]. Apesar dos nós individualmente possuírem pouca capacidade computacional e de energia, um esforço colaborativo entre os mesmos permite a realização de uma tarefa maior. A seção 2.2 apresenta as principais características da arquitetura de nós sensores sem fio.

É importante salientar que a tecnologia para projetar e construir nós sensores sem fio está comercialmente disponível e tende a se tornar cada vez mais acessível com a produção em larga escala de diferentes tipos de micro-sensores [1, 3, 4, 5, 6, 62]. A figura 2.6 apresenta alguns exemplos de nós sensores sem fio resultantes de pesquisas em diversas instituições, como o COTS Dust e o Smart Dust [5] da Universidade da California, Berkeley, WINS [6] (*Wireless Integrated Network Sensors*) da Universidade da California, Los Angeles e JPL Sensor Webs [3] do *Jet Propulsion Lab* da NASA.

Os principais obstáculos ao desenvolvimento de um arcabouço de gerenciamento para RSSFs decorreram da novidade, da interdisciplinaridade do tema e da dificuldade associada ao entendimento dos detalhes dessas redes. Estas dificuldades foram superadas e resultaram em contribuições na forma de mini-cursos.

Os resultados do desenvolvimento da fase de revisão bibliográfica, apresentada no capítulo 2, foram publicados como parte do mini-curso da Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação de 2002 [52]. A partir da identificação de problemas em RSSFs e da definição dos primeiros requisitos de gerenciamento, outro mini-curso foi publicado no Simpósio Brasileiro de Redes de Computadores (SBRC) 2003 [53]. Durante o trabalho de pesquisa sobre o gerenciamento de serviços e a

definição de requisitos de QoS (*Quality of Service*) para RSSFs, um mini-curso referente ao tema foi publicado no Workshop de Comunicação Sem Fio e Computação Móvel de 2003 [85]. Um tutorial sobre *middleware* e tolerância a falhas em RSSFs foi publicado no Workshop de Segurança e Tolerância a Falhas do Simpósio Brasileiro de Redes de Computadores de 2003 [54].

Apesar da rápida expansão, as RSSFs e suas aplicações vêm sendo projetadas e desenvolvidas sem considerar uma solução integrada de gerenciamento. Na maioria dos casos, os pesquisadores desenvolvem soluções para problemas específicos, fazendo suposições sobre o contexto sem considerar a integração com outros trabalhos. Além disso, as funcionalidades da aplicação são confundidas com as funcionalidades de gerenciamento, não havendo um mecanismo que possa propor a distinção entre elas. Embora isso possa não ser um problema para redes pequenas, provavelmente será para RSSFs formadas por centenas ou milhares de nós nas quais há a necessidade de que as redes e seus elementos se reconfigurem e se adaptem ao seu próprio estado e às condições ambientais onde estão operando sem intervenção humana.

Dadas as características particulares das RSSFs, fica claro que existem diferenças significativas entre o gerenciamento tradicional e o gerenciamento de RSSFs. As RSSFs tem características particulares que as diferenciam em muitos aspectos de outra redes, inclusive no que diz respeito ao gerenciamento. A seção 2.3 procura apontar algumas dessas diferenças.

O potencial de observação e controle do mundo real permite que as RSSFs se apresentem como uma solução para diversas aplicações: monitoramento ambiental, gerenciamento de infra-estrutura, segurança pública e de ambientes em geral, transporte e controle militar [6, 8, 29, 50, 58, 99]. Esta gama de aplicações tem estimulado ainda mais o desenvolvimento desses dispositivos e atraído a atenção da comunidade acadêmica. A seção 2.4 descreve algumas aplicações que utilizam RSSFs ou que são aplicações em potencial para tais redes.

Como não foram encontrados na literatura trabalhos relacionados diretamente ao gerenciamento de RSSFs, a seção 2.5 apresenta um resumo sobre os principais tópicos de pesquisa. Alguns destes tópicos poderiam ser usados pela arquitetura de gerenciamento

proposta mas não estão diretamente relacionados ao tema gerenciamento. Uma conclusão para este capítulo é oferecida na seção 2.6.

Resumo do Capítulo 3 - Uma Organização para o Gerenciamento de RSSFs

O capítulo[2] 3 discute os desafios impostos ao gerenciamento de RSSFs e propõe uma organização tridimensional para esse gerenciamento.

Sendo as RSSFs formadas por nós sensores autônomos e que operam em áreas remotas sem intervenção humana direta, a seção 3.1 propõe o uso do paradigma de computação autonômica no desenvolvimento RSSFs, o que define RSSFs auto-gerenciadas. Uma RSSF autonômica ou auto-gerenciada é responsável por configurar e reconfigurar a si própria sem intervenção humana direta. Conforme o caso, uma RSSF autonômica deve organizar-se em grupos (auto-organização), adaptar-se a mudanças no ambiente e mudanças em sua topologia e conectividade (auto-configuração). Uma RSSF autonômica deve otimizar seu funcionamento e monitorar seus componentes para configurá-los às diferentes densidades de nós e cargas de trabalho para atender aos requisitos de qualidade de serviço. Ela deve implementar serviços de auto-diagnóstico para detectar problemas ou problemas em potencial tais como áreas descobertas decorrentes da baixa densidade ou desperdício de energia e perdas de informação em função da alta densidade de nós. Uma RSSFs autonômica deve recupera-se dos problemas e eventos extraordinários que causem mal funcionamento de seus componentes (auto-cura). Uma RSSF autonômica deve detectar, identificar e proteger-se contra várias ameaças (internas e externas) para manter sua segurança e integridade (auto-proteção). Uma RSSF autonômica deve conhecer seu ambiente e o contexto onde realiza duas atividades e agir de acordo com os requisitos de qualidade que foram estabelecidos (auto-consciência). Uma RSSF autonômica deve conhecer a si própria, assim

[2]O conteúdo deste capítulo foi resumido para publicação como artigo no IEEE Communications Magazine de fevereiro de 2003 [84] e será publicado em 2004 como um capítulo III do livro "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems" editado por Mohammad Ilyas and Imad Mahgoub, editora CRC Press.

como seus limites de operação (auto-conhecimento). Uma RSSF produz e transporta os próprios dados (auto-serviço). Ela deve negociar a realização de um serviço a partir de três níveis de qualidade: qualidade de sensoriamento, qualidade de processamento e qualidade de disseminação.

A tarefa de construir e desenvolver soluções autônomas de gerenciamento em ambientes onde existem centenas ou milhares de nós com características particulares é uma tarefa complexa. Esta tarefa torna-se ainda mais difícil devido às restrições de recursos das RSSFs. Uma boa estratégia para lidar com esta complexidade é usar dimensões de gerenciamento que permitam diferentes níveis de abstração. Neste sentido, esta tese propõe uma organização em três dimensões como parte do arcabouço de gerenciamento (ver figura 3.1). Uma dimensão é composta pelos níveis de gerenciamento (gerenciamento de negócio, gerenciamento de serviços, gerenciamento de rede, gerenciamento de elemento de rede) (ver figura 3.2) e outra dimensão é composta pelas áreas funcionais de gerenciamento (gerenciamento de configuração, gerenciamento de falhas, gerenciamento de desempenho, gerenciamento de segurança e gerenciamento de contabilização) (ver figura 3.3). Estas duas dimensões foram definidas e são utilizadas na organização do gerenciamento tradicional (seção 3.2). Contudo, elas são redefinidas sob a perspectiva de RSSFs. A terceira dimensão de gerenciamento, chamada "funcionalidades de RSSFs" é proposta por esta tese.

Como mencionado, uma RSSF é um tipo de sistema dependente da aplicação, isto é, os parâmetros de configuração, operação e manutenção variam com o tipo de aplicação definida. O tipo de aplicação influenciará diretamente nas funções exercidas pelos nós da rede, assim como na arquitetura desses nós (processador, memória, dispositivos sensores, fonte de energia, transceptor), na quantidade de nós que compõem a rede, na distribuição inicialmente planejada para a rede, no tipo de deposição dos nós no ambiente, na escolha dos protocolos da pilha de comunicação, no tipo de dado que será tratado, no tipo de serviço que será provido pela rede e conseqüentemente no tempo de vida dessa rede. No processo de desenvolvimento do arcabouço de gerenciamento, estudamos um conjunto de aplicações encontradas na literatura e propusemos um modelo funcional, que permite caracterizar

Configuração		
Composição	Homogênea	Rede composta de nós que apresentam a mesma capacidade de hardware. Eventualmente os nós podem executar software diferente.
	Heterogênea	Rede composta por nós com diferentes capacidades de hardware.
Organização	Hierárquica	RSSF em que os nós estão organizados em grupos (*clusters*). Cada grupo terá um líder (*cluster-head*)que poderá ser eleito pelos nós comuns. Os grupos podem organizar hierarquias entre si.
	Plana	Rede em que os nós não estão organizados em grupos.
Mobilidade	Estacionária	Todos os nós sensores permanecem no local onde foram depositados durante todo o tempo de vida da rede.
	Móvel	Rede em que os nós sensores podem ser deslocados do local onde inicialmente foram depositados.
Densidade	Balanceada	Rede que apresenta uma concentração e distribuição de nós por unidade de área considerada ideal segundo a função objetivo da rede.
	Densa	Rede que apresenta uma uma alta concentração de nós por unidade de área.
	Esparsa	Rede que apresenta uma baixa concentração de nós por unidade de área.
Distribuição	Irregular	Rede que apresenta uma distribuição não uniforme dos nós na área monitorada.
	Regular	Rede que apresenta uma distribuição uniforme de nós sobre a área monitorada.
Tamanho	Pequena	Rede composta de uma centena de elementos de rede.
	Média	Rede composta de centenas a mil elementos de rede.
	Grande	Rede composta por milhares de elementos de rede.

Table 1: Caracterização das RSSFs segundo a configuração.

as RSSFs. A classificação de uma RSSF depende de seu objetivo e área de aplicação. O modelo funcional foi inicialmente publicado em [83] e serviu de base para o desenvolvimento da nova dimensão de gerenciamento.

De acordo com o modelo funcional desenvolvido, as RSSFs podem ser classificadas segundo a configuração (ver tabela 1), o sensoriamento (ver tabela 2) e segundo o tipo de comunicação (ver tabelas 3 e 4). Uma RSSF também pode ser diferente segundo o tipo de processamento que executa (ver tabela 5).

A nova dimensão de gerenciamento permite a caracterização da rede, facilitando a iden-

Sensoriamento		
Coleta	Periódica	Os nós sensores coletam dados sobre o(s) fenômeno(s) em intervalos regulares. Um exemplo são as aplicações que monitoram o canto dos pássaros. Os sensores farão a coleta durante o dia e permaneceram desligados durante a noite.
	Contínua	Os nós sensores coletam os dados continuamente. Um exemplo são as aplicações de exploração interplanetária que coletam dados continuamente para a formação de base de dados para pesquisas.
	Reativa	Os nós sensores coletam dados quando ocorrem eventos de interesse ou quando solicitado pelo observador. Um exemplo são as aplicações que detectam a presença de objetos na área monitorada.

Table 2: Caracterização das RSSFs segundo o sensoriamento.

Classificação segundo a Comunicação		
Disseminação	Programada	Os nós disseminam em intervalos regulares.
	Contínua	Os nós disseminam os dados continuamente.
	Sob eventos	Os nós disseminam os dados quando ocorre um evento pré-determinado.
	Sob Demanda	Os nós disseminam os dados em resposta à consulta do observador e à ocorrência de eventos.
Tipo Conexão	Simétrica	Todas as conexões existentes entre os nós sensores, com exceção do nó sorvedouro têm o mesmo alcance.
	Assimétrica	As conexões entre os nós comuns têm alcance diferente.
Transmissão	Simplex	Os nós sensores possuem transceptor que permite apenas transmissão da informação.
	Half-duplex	Os nós sensores possuem transceptor que permite transmitir ou receber em um determinado instante.
	Full-duplex	Os nós sensores possuem transceptor que permite transmitir ou receber dados ao mesmo tempo.

Table 3: Caracterização das RSSFs segundo a comunicação (Parte A).

tificação de requisitos de gerenciamento que são dependentes da aplicação. Por exemplo, os requisitos de gerenciamento de uma RSSF que monitora fenômenos de temperatura diferem-se dos requisitos de gerenciamento de uma RSSF que monitora imagens de vídeo, sejam eles latência, precisão, largura de banda, exposição, área de cobertura e processamento de sinais. A nova dimensão é composta pelas funcionalidades de sensoriamento, processamento, comunicação, manutenção e configuração, permitindo caracterização das RSSFs e a especificação de modelos funcionais para as aplicações. Por essa razão, a organização tridimensional é considerada na definição das funções e da informação de geren-

Classificação segundo a Comunicação		
Alocação de Canal	Estática	Neste tipo de rede se existirem "n" nós, a largura de banda é dividida em "n" partes iguais na freqüência (FDMA – *Frequency Division Multiple Access*), no tempo (TDMA – *Time Division Multiple Access*), no código (CDMA – Code Division Multiple Access), no espaço (SDMA – *Space Division Multiple Access*) ou ortogonal (OFDM – *Orthogonal Frequency Division Multiplexing*). A cada nó é atribuída uma parte privada da comunicação, minimizando interferência.
	Dinâmica	Neste tipo de rede não existe atribuição fixa de largura de banda. Os nós disputam o canal para comunicação dos dados.
Fluxo de Informação	*Flooding*	Neste tipo de rede, os nós sensores fazem *broadcast* de suas informações para seus vizinhos que fazem *broadcast* desses dados para outros até alcançar o ponto de acesso. Esta abordagem promove um alto *overhead* mas está imune às mudanças dinâmicas de topologia e a alguns ataques de impedimento de serviço (DoS – *Denial of Service*).
	Multicast	Neste tipo de rede os nós formam grupos e usam o *multicast* para comunicação entre os membros do grupo.
	Unicast	Neste tipo de rede, os nós sensores podem se comunicar diretamente com o ponto de acesso usando protocolos de roteamento multi-saltos.
	Gossiping	Neste tipo de rede, os nós sensores selecionam os nós para os quais enviam os dados.
	Bargaining	Neste tipo de rede, os nós enviam os dados somente se o nó destino manifestar interesse, isto é, existe um processo de negociação.

Table 4: Caracterização das RSSFs segundo a comunicação (Parte B).

Classificação segundo o Processamento		
Cooperação	Infra-estrutura	Os nós sensores executam procedimentos relacionados à infra-estrutura da rede como por exemplo, algoritmos de controle de acesso ao meio, roteamento, eleição de líderes, descoberta de localização e criptografia.
	Localizada	Os nós sensores executam além dos procedimentos de infra-estrutura, algum tipo de processamento local básico como por exemplo, tradução dos dados coletado pelos sensores baseado na calibração.
	Correlação	Os nós estão envolvidos em procedimentos de correlação de dados como fusão, supressão seletiva, contagem, compressão, multi-resolução e agregação.

Table 5: Caracterização das RSSFs segundo o processamento.

ciamento, no desenvolvimento de serviços e aplicações de gerenciamento. A seção 3.3 apresenta a organização tridimensional proposta por esta tese.

Uma conclusão para este capítulo é oferecida na seção 3.4. As contribuições deste capítulo foram parcialmente publicadas [79, 82, 83, 84, 85].

Resumo do Capítulo 4 - Arquitetura MANNA

A arquitetura MANNA foi proposta[3] para prover soluções de gerenciamento para diferentes tipos de RSSFs. No capítulo 4, uma visão geral da arquitetura é apresentada na seção 4.1.

O capítulo define um esquema para se construir soluções de gerenciamento a partir da definição de serviços e funções e da utilização de modelos. Funções de gerenciamento representam a menor parte funcional de um serviço de gerenciamento. A especificação de serviços de gerenciamento consiste em definir quais, quando e com quais dados as tarefas de gerenciamento serão executadas. Entretanto, para as RSSFs, a arquitetura MANNA estabelece que o gerenciamento não se esgota nas funções de gerenciamento, sendo necessário transcendê-las. O gerenciamento não pode executar qualquer serviço ou função sobre a RSSF sem conhecer as condições da rede, isto é, sem conhecer o seu estado. Assim, as condições para execução de serviços e funções de gerenciamento são dependentes do estado da rede, representado por modelos de rede que neste texto também serão chamados de "mapas'.

A figura 4.3 apresenta o esquema para a construção do gerenciamento, iniciando pela definição dos serviços de gerenciamento que são executados por um conjunto de funções de gerenciamento. As condições para execução de uma função são obtidas a partir de modelos, os quais representam o estado da rede sob determinado nível de abstração. Um serviço de gerenciamento pode utilizar uma ou mais funções de gerenciamento. Diferentes serviços podem utilizar funções em comum, as quais utilizam modelos para recuperar um estado da rede, considerando um dado aspecto. Além de utilizar as informações sobre o estado da

[3]Esta proposta foi publicada no IEEE Communications Magazine de Fevereiro de 2003 [84].

rede, algumas funções podem ser definidas para gerar e atualizar mapas. Os serviços e as funções de gerenciamento utilizam e produzem informação de gerenciamento.

De acordo com o modo como são implementados, os serviços e funções de gerenciamento podem ser: *manuais*, quando executadas fora do sistema de gerência, *semi-automáticos*, quando executadas por um humano auxiliado por um sistema de software que fornece um modelo da rede no período ou invocadas pelo sistema de gerência, e *automáticos*, quando executadas por algum software invocado automaticamente após o processamento de informações obtidas a partir de um ou mais modelos de rede. Neste último caso, quando os serviços de gerenciamento são executados automaticamente sem intervenção humana, a RSSF passa a ser um sistema de computação autonômico [40], isto é, auto-gerenciado.

A seção 4.2 apresenta uma lista de funções de gerenciamento obtidas a partir da organização tridimensional proposta no capítulo 3. A seção 4.3 apresenta o esquema para se desenvolver soluções de gerenciamento a partir da definição de serviços e funções e da utilização de modelos.

Alguns destes serviços e funções, assim como estratégias de atualização dos mapas foram implementados e testados durante o desenvolvimento da tese. Um esquema para construção e atualização de mapas para RSSFs planas utilizando algoritmos distribuídos foi publicado em [92]. Outro esquema para atualização de mapa de energia tolerante a falhas e utilizando métodos de fusão por área foi publicado em [104]. Além disso, uma função de gerenciamento para identificação de nós redundantes usando Diagramas de Voronoi foi publicado em [103], um serviço para manutenção da área de cobertura centralizado foi definido e publicado em [79].

Os modelos de rede provem uma visão abstrata do sistema através da qual, dado um certo objetivo, é possível omitir todos os aspectos não relevantes. Exemplos de modelos de rede definidos pela arquitetura MANNA são: mapa da área de sensoriamento, mapa da área de cobertura de comunicação, modelo de comportamento, modelo de dependência, mapa de topologia, mapa de energia, modelo de conectividade, modelo de agregação, modelo de custo, modelo de estimativas, modelo de consumo de energia e modelo comportamental.

Embora as RSSFs sejam dependentes da aplicação, a arquitetura MANNA provê flexi-

bilidade, isto é, ela permite o gerenciamento de qualquer tipo de RSSF. Com este intuito, a arquitetura MANNA propõe três arquiteturas de gerenciamento – funcional, física e de informação – que em seu desenvolvimento e implementação também levam em conta as três dimensões de gerenciamento definidas no capítulo 3. A arquitetura de informação (seção 4.4) é proposta para garantir soluções comuns para o gerenciamento através da definição de uma modelo genérico de informação[4] e estratégias para obter esta informação. Esta arquitetura define dois tipos de informação para RSSFs, estáticas (representadas através de classes de objetos) e dinâmicas (representadas através de modelos de rede citados acima). A arquitetura funcional (seção 4.5) é proposta para planejar os locais na rede onde as entidades de gerenciamento (gerentes e agentes) podem ser executadas e por quais serviços e funções de gerenciamento cada uma delas será responsável. A arquitetura física (seção 4.6) descreve as interfaces que podem ser utilizadas para troca de informação entre as entidade de gerenciamento. Ela não define ou desenvolve protocolos de comunicação mas sugere quais perfis podem ser mais adequados ao propósito da solução de gerenciamento.

A seção 4.7 trata dos aspectos envolvidos no desenvolvimento de uma aplicação de gerenciamento, incluindo uma discussão sobre os tipos de gerenciamento: centralizado, hierárquico e distribuído. A seção 4.8 apresenta um exemplo do uso de algumas funcionalidades da arquitetura MANNA. Uma conclusão para este capítulo é oferecida na seção 4.9. As contribuições deste capítulo foram parcialmente publicadas [79, 83, 92, 103, 104].

[4]O modelo de informação genérico e um modelo funcional para as RSSFs foram publicados no GRES – Colloque Francophone sur la Gestion de Reseaux et de Services 2003 [83].

Resumo do Capítulo 5 - Desenvolvendo uma Aplicação para RSSFs Contínuas

Neste capítulo[5], experimentos são realizados para mostrar como o arcabouço de gerenciamento proposto por esta tese pode ser usado no desenvolvimento de uma solução de gerenciamento. Uma aplicação para monitoração da qualidade do ar foi definida como estudo de caso. A monitoração da qualidade do ar envolve a percepção e processamento de muitos parâmetros. Para simplificar, a rede definida realiza apenas o sensoriamento de temperatura e concentração de monóxido de carbono. A rede continuamente coleta dados do ambiente, realiza o processamento e dissemina estes dados em direção ao observador. Trata-se de uma rede de sensoriamento e disseminação contínua.

Diferentes cenários foram desenvolvidos considerando diferentes configurações de redes em termos de composição (homogênea e heterogênea) e organização (plana e hierárquica). As soluções de gerenciamento propostas consideraram as diferentes configurações na especificação das arquiteturas de informação, funcional e física. Os experimentos realizados estão descritos na seção 5.1. Uma aplicação de gerenciamento é construída selecionando alguns serviços e funções de gerenciamento e a informação necessária para sua execução. As escolhas foram realizadas utilizando a organização tridimensional proposta no capítulo 3.

Para a realização destes experimentos, um ambiente de simulação foi desenvolvido e também é considerado uma contribuição desta tese. Este ambiente é chamado de MANNASim e foi construído a partir das funcionalidades da ferramenta de simulação Network Simulator (NS-2) [94]. A seção 5.2 apresenta uma visão geral deste ambiente que ainda está em fase de desenvolvimento para contemplar outras funcionalidades. Atualmente, o MANNASim é um projeto de software livre financiado pelo CNPq.

Os principais serviços de uma RSSFs são sensoriamento, processamento e disseminação. Para a aplicação definida em nossos experimentos, monitoramos a QoS utilizando

[5]Este capítulo foi publicado no Workshop de Comunicação Sem Fio e Computação Móvel 2003 [80] e aceito para publicação no IEEE/IFIP Network Operations and Management Symposium 2004 [81]. Outros resultados de experimentos utilizando a arquitetura MANNA foram publicados no IEEE Latin American Network Operations and Management Symposium 2003 [79].

as métricas de área de cobertura, precisão, atraso, mensagens perdidas e consumo de energia) e configuração de parâmetros de operação da rede (utiliza-se este serviço nas redes homogêneas para estabelecer um compromisso entre o sensoriamento e a comunicação, isto é, os nós próximos ao nó sorvedouro são programados para deixar de realizar o sensoriamento quando atingirem determinado nível de energia residual. Nas redes hierárquicas heterogêneas, o serviço de configuração altera a potência de transmissão, isto é, o alcance da comunicação dos nós líderes, também em função da distância da estação base).

Na ocorrência de áreas com alta densidade de nós sensores, podem ocorrer áreas de intersecção de sensoriamento, redundância de dados, interferência na comunicação, e desperdício de energia. Neste caso, o serviço de gerenciamento identifica os nós redundantes e os retira administrativamente de serviço, isto é, desliga os nós por um período de tempo. Quando os nós principais saem de serviço, gerando áreas esparsas, o serviço tenta ativar os nós *backups*, se existirem. O serviço é realizado automaticamente promovendo a produtividade dos recursos e tirando proveito da alta densidade) e monitoração de QoS (uma RSSF é usuária de si mesma, isto é, ela produz, processa e entrega sua informação).

As funções de gerenciamento selecionadas da lista definida na seção 4.2 e usadas nos experimentos, sem qualquer ordenação particular, são: definição da área monitorada, distribuição dos nós, auto-teste dos nós, auto-organização, controle de densidade, descoberta do mapa de topologia, agregação, geração do mapa de energia, geração do mapa de produção, escalonamento das operações de gerenciamento, controle do estado operacional dos nós, controle do estado administrativo dos nós, geração do mapa da área de cobertura.

Os principais serviços de gerenciamento selecionados da lista apresentada na seção 4.3 foram: planejamento da rede (este serviço contempla todas as funções de gerenciamento que antecedem a deposição dos nós na área monitorada), manutenção da área de cobertura (este serviço executa funções de monitoração da área de cobertura identificando áreas de intersecção de sensoriamento e áreas descobertas.

Um segundo objetivo dos experimentos realizados é mostrar como a configuração da rede influencia nas métricas de atraso, mensagens perdidas e energia consumida. Selecionar o nível de detalhe ou nível de abstração para uma simulação é uma tarefa difícil. Poucos de-

talhes podem produzir simulações que são incorretas ou produzem resultados que conduzem a uma impressão errada. Por outro lado, um nível mais detalhado de simulação pode implicar em maior tempo de desenvolvimento, simulação e análise de resultados. A seção 5.3 apresenta justificativas sobre as decisões tomadas no desenvolvimento das simulações e as condições que foram assumidas como verdadeiras na condução dos experimentos.

As seções 5.4 e 5.5 avaliam o impacto da solução de gerenciamento proposta sobre a RSSFs definida como estudo de caso. Os resultados das simulações mostraram que o gerenciamento pode melhorar o desempenho de RSSF com várias configurações e fornecer ao observador informações relevantes, sem custo adicional de consumo de energia para a rede. Uma conclusão para este capítulo é oferecida na seção 5.6. As contribuições deste capítulo foram publicadas em [80, 81].

Resumo do Capítulo 6 - Conclusão

Nesta tese, desenvolvemos um arcabouço para o gerenciamento de RSSFs. Este arcabouço traz contribuições para a área, além de bases técnicas para a evolução desse tipo de tecnologia no aspecto de gerenciamento. Tal como definido no texto, alguns princípios nortearam a concepção do arcabouço proposto, quais sejam (1) simplicidade, (2) aderência às peculiaridades dessas redes, incluindo também o seu dinamismo, e (3) eficácia no uso dos recursos escassos.

Apesar da rápida expansão, até o momento, as RSSFs e suas aplicações vinham sendo projetadas e desenvolvidas sem considerar uma solução integrada de gerenciamento. As funcionalidades da aplicação eram confundidas com as funcionalidades de gerenciamento, não havendo um mecanismo que pudesse propor a distinção entre elas. Embora isso possa não ser um problema para redes pequenas, é certamente para RSSFs formadas por centenas ou milhares de nós nas quais há a necessidade de que as redes e seus elementos se reconfigurem e se adaptem ao seu próprio estado e às condições ambientais onde estão operando sem intervenção humana. Outro aspecto que deve ser considerado é que em pouco tempo as RSSFs estarão integradas a outras redes como por exemplo a Internet.

Uma solução de gerenciamento que separe as funcionalidades, organize o gerenciamento e utilize um modelo genérico de informação pode facilitar a integração. A utilização do paradigma de autogerenciamento, tal como proposto pela arquitetura MANNA, também mostra-se adequada às características específicas dessas redes.

Os principais obstáculos ao gerenciamento das RSSFs decorrem da novidade e interdisciplinaridade do tema e da dificuldade associada ao entendimento dos detalhes dessas redes. No processo de desenvolvimento deste trabalho, procuramos organizar o conhecimento sobre as RSSFs propondo um modelo funcional, que permite caracterizar essas redes, e uma lista de serviços e funções de gerenciamento. O modelo funcional desenvolvido foi usado como base para a nova dimensão de gerenciamento chamada "funcionalidades de RSSFs". Duas outras dimensões de gerenciamento compõem a organização tridimensional proposta por esta tese, sendo elas áreas funcionais de gerenciamento e níveis de gerenciamento. Os serviços e funções de gerenciamento que compõem as listas providas neste documento foram obtidos a partir do uso dessa organização tridimensional.

A arquitetura MANNA proposta no arcabouço, estabelece uma separação entre as funcionalidades das RSSFs e as funcionalidades do gerenciamento através do uso da organização tridimensional e de três arquiteturas que compõem o sistema de gerenciamento quais sejam, arquitetura funcional, arquitetura física e arquitetura de informação. Isto tornará possível a integração das atividades de organização, administração e manutenção para este tipo de rede.

Entendemos que o arcabouço proposto neste trabalho é uma contribuição relevante para a área, uma vez que não havia na literatura qualquer proposta relacionada ao tema. Durante o processo de desenvolvimento muitos desafios foram superados e muitos ainda prevalecem como, por exemplo o uso de uma pilha de protocolos que seja adequada às RSSFs. Este tópico, pilha de protocolos, não está diretamente relacionado ao tema gerenciamento mas os efeitos de se utilizar protocolos inadequados nos experimentos foram identificados nos resultados. Este exemplo ilustra o tipo de dificuldade que tivemos que enfrentar, uma vez que quando iniciamos o desenvolvimento, algoritmos de roteamento e controle de acesso ao meio específicos para RSSFs não estavam disponíveis ou não haviam

sido publicados. Ainda hoje, existem poucos algoritmos propostos nesta área. Um outro exemplo diz respeito ao ambiente de simulação. Quando o desenvolvimento teve início e até o momento, nenhuma ferramenta de simulação específica para RSSFs foi encontrada disponível para uso. Assim, para realizar os experimentos, tivemos que construir um ambiente de simulação a partir do simulador Network Simulator (NS-2) [94]. A módulo MANNASim é uma contribuição desta tese e estará disponível em pouco tempo. Outros pesquisadores poderão utilizá-la reduzindo o tempo e o esforço no desenvolvimento tanto de aplicações como de soluções de gerenciamento.

Ao final do trabalho, também percebemos que algumas decisões demandaram tempo e esforço em direções equivocadas. Em muitas ocasiões tentamos superar dificuldades assumindo responsabilidades além do necessário. Nessas ocasiões não tínhamos visão, experiência, ou referência a qualquer outro trabalho na literatura que indicasse a relação custo benefício de tal decisão. Por outro lado, a maioria das decisões foram acertadas, e com isso, conseguimos chegar à proposição de uma arquitetura que foi apresentada em detalhes no texto. Um dos maiores objetivos da arquitetura MANNA é promover a produtividade dos recursos e a qualidade dos serviços. Os experimentos realizados com a arquitetura MANNA mostraram que a solução é viável embora sua implementabilidade não tenha sido completamente testada. A implementação de uma solução completa de gerenciamento demandaria tempo além daquele definido para o desenvolvimento desta tese.

Este trabalho pode ser estendido de várias formas. Algumas extensões imediatas seriam: (1) ampliar o conjunto de experimentos para avaliar a escalabilidade das soluções propostas; (2) desenvolver soluções de gerenciamento para outros tipos de RSSFs (por exemplo, dirigidas a eventos, sob demanda e programadas); (3) desenvolver e integrar novos serviços e funções automáticas aos cenários já desenvolvidos (4) implementar gerenciamento hierárquico (usando o conceito de gerente de gerentes) e gerenciamento distribuído (usando o conceito de gerente-para-gerente); (5) especificar políticas de gerenciamento e aplicar o paradigma de gerenciamento baseado em políticas no arcabouço proposto; (6) utilizar perfis de protocolos específicos para RSSFs; (7) projetar e avaliar mecanismos de construção e atualização dos modelos (mapas) de rede; (8) utilizar mobilidade de código

para migração dos agentes ou atualização dos serviços; (9) ampliar o modelo de informação genérico; (10) avaliar o arcabouço em cenários reais utilizando nós sensores Mica-Motes e (11) desenvolver funções objetivo a serem utilizadas no contexto da arquitetura.

Linnyer Beatrys Ruiz

MANNA: A Management Architecture for Wireless Sensor Network

A thesis submitted to the Department of Computer Science in partial fulfillment of the requirements for the Degree of Doctor of Science.

Federal University of Minas Gerais
Brazil
December, 2003

To my husband

Camillo

To my parents

Sidney and Julita

and my friends

Edênia, Kalina, Thais, Fabrício, Rainer, and Claudine,

with love...

Abstract

Wireless sensor networks are becoming an increasing technology that will be used in a variety of applications such as environmental monitoring, infrastructure management, public safety, medical, home and office security, transportation, and military systems. Wireless sensor networks will also play a key role in pervasive computing where computing devices and people are connected to the Internet. However, until now, wireless sensor networks and their applications have been developed without considering a integrated management solution.

This thesis proposes a management architecture for wireless sensor networks called MANNA. The proposed architecture establishes a separation between both sets of functionalities, i.e., application and management through a proposition of three architecture (information, functional, and physical) and using three management dimensions (management functional areas, management levels, and WSN functionalities). This will enable the integration of organizational, administrative, and maintenance activities for this kind of network. The adoption of a strategy based on the traditional framework of functional areas and management levels will allow management integration in the future. One of the major goals of the management architecture is to promote the productivity of the network resources and the quality of the service provided.

The task of building and deploying management solutions in environments where there will be tens of thousands of network elements with particular features and organization, is very complex. This task becomes worse due to physical restrictions of the sensor nodes, in particular energy and bandwidth restrictions.

Contents

List of Figures

List of Tables

Chapter 1

Introdution

Wireless Sensor Networks (WSNs) consisting of a large number of sensor nodes deployed over an area and integrated to collaborate through a wireless network, encourage several novel and existing applications such as environmental monitoring, health care, infrastructure management, public safety, medical, home and office security, transportation, and military [8, 29, 50, 58]. These applications have been enabled by the rapid convergence of three technologies, namely digital circuit, wireless communication, and Micro Electro Mechanical System (MEMS). These technologies have enabled very compact and autonomous sensor nodes, each containing one or more sensor devices, computations and communication capabilities, and power supply. The physical dimensions of sensor nodes tend to be small (e.g., cm^3 or mm^3) and the size limitation ends up restraining the power supply capacity and computational resources of the sensor nodes.

Some of the applications foreseen to sensor networks will require a large number of devices, about tens of thousands sensor nodes. Traditional methods of sensor networking represent an impractical, complex, and expensive demand on cable installation. WSNs promise several advantages over traditional sensing methods in many ways: better coverage, higher resolution, fault tolerance, and robustness. The ad hoc nature and deploy–and–leave vision make them even more attractive in military applications and other risk-associated applications such as catastrophe, toxic zones, and disaster [8, 29].

Until now, WSNs and their applications have been developed without considering an integrated management solution. Several interesting works may be found in the literature about specific issues in wireless sensor networks (see Section 2.5). Some of these proposals aim at the specific function for determined WSN application, for example routing in flat networks that do environmental monitoring. To the best of our our knowledge, it has not been found in the literature any work that proposes a management architecture for WSNs. The same is true for a generic information model that includes management (managed object classes and WSN models) and support (support object classes) information.

The task of building and deploying management solutions in environments where there will be tens of thousands of network elements with particular features and organization is not trivial. To make things worse, due to physical restrictions of unattended sensor nodes, especially, energy and bandwidth restrictions, this task becomes difficult.

1.1 Objectives

This thesis aims to study the WSNs management problem in order to understand the needs, requirements and open questions about this theme as well as to identify the differences between WSNs management and traditional management In order to contribute to the progress of this and other correlated areas, this thesis proposes a framework for WSNs management. This framework must be simple, adherent to network idiosyncrasies including its dynamic behavior, and efficient in the use of scarce resources.

The proposed framework introduces a novel organization for WSNs management considering the two well-known management dimensions, namely, management levels, management functional areas, and a novel dimension called WSN functionalities. This innovative dimension is a proposal of this thesis as well. The traditional management dimensions are revisited from a WSN perspective. The framework includes a scheme to obtain management functions and a scheme to build management services and applications from the three management dimensions. A list of these management functions and services are provided.

The WSN management framework also includes the proposition of a management archi-

tecture called MANNA[1]. This architecture is based on the paradigm of self-management which is management based on the use of automatic functions and services with a minimum of human interference.

The implementation of a management system is not objective of this thesis. Its goal is to propose a management framework for WSNs. In this direction, some experiments are presented to show how the proposed architecture achieve its objectives. One of the major goals of a management architecture is to promote network resources productivity and the quality of the service provided.

1.2 Contributions

There are several significant differences in the management of traditional networks and WSNs and until now, no work has addressed these differences [84]. In this sense, this work presents a contribution to the field, since it proposes a WSNs management architecture. The contributions presented in this thesis are:

- A management framework which allows to differentiate application functionalities from management functionalities of the WSNs. This management framework includes: a management architecture for WSNs called MANNA which is organized in information, functional, and physical architectures, and the proposition of an organization for WSNs management from three management dimensions, namely management levels, management functional areas and a novel management dimension called "WSN functionalities". The traditional management dimensions are rethought for WSNs;

- A novel approach to build management services and management applications considering the kind of WSN application;

[1]MANNA from Hebrew *Man hu*: "What is this?" – food that God provided for its people during the journey through the wilderness (Exodus 16:31 in the Bible). An unexpected and very welcome gift or advantage. A signal of confidence.

- An approach to define management functions for WSNs as well as a list of management functions and services which can be performed automatically, semi-automatically, and manually;
- A proposition of new WSN models to represent network states in different perspectives;
- A WSN characterization through a functional model;
- An information model for WSNs;
- A framework for developing WSN applications and management solutions based on the Network Simulator Tool (NS-2), called MANNASim;
- The use of a new paradigm called autonomic management;

Some of the contributions and results of this work have been published in parallel with its development [52, 53, 54, 79, 80, 81, 82, 83, 84, 85, 86, 92, 93, 103, 104]. In some cases, the documents presented aspects in more details than this thesis, for a matter of space.

The progress of the WSNs area is notable. When this work began, there were few works published and various proposals still bide their time to be published. Management of WSNs is a new research area that only recently started to receive attention from the research community. Thus, when this work has started, there was no specific work in this theme. Such a scenario imposes some difficulties to perform this project as well as good opportunities.

1.3 Thesis Organization

This thesis has an approach to a perspective on emerging wireless sensor management, covering management issues and technologies, proposing a network management architecture, an information model and some schemes establishing a management functions list, building a simulation framework and presenting new research opportunities.

This thesis is structured into six chapters and one appendix. Chapter 2 presents an overview of WSNs, their characteristics and potential applications, shows how WSNs differs from other kinds of networks; and offers an overview about the main research topics in WSNs.

Chapter 3 discusses the management challenges for WSNs and introduces a novel organization for WSNs management. The novel organization propose the use of the three management dimensions. One composed by the management levels, one by the management functional areas, and a novel dimension composed by WSN functionalities. All three dimensions are explored from a WSN perspective.

MANNA architecture and its functionalities are presented in Chapter 4. This chapter introduces the three architectures that composes the MANNA: information, functional and physical architectures. Two schemes are also proposed in this chapter: a scheme to define management functions and a scheme to build management services and management applications using the MANNA architecture.

The implementation of a management system is not one of the objectives of this thesis. Its goal is to propose a management framework for wireless sensor networks. Aiming to show how the architecture proposed can achieve its objectives, some simulations were carried out. In this direction, a WSN application which does the monitoring of air quality in urban area is developed in Chapter 5. In order to evaluate some aspects of MANNA, a management application is created using some management services and functions, WSN models, and some managed object classes. The three architectures (information, functional and physical) are established according to the application type because the management application to be built depends on the kind of application being managed. The experiments were conducted for different WSN configurations (flat, hierarchical, homogeneous, heterogeneous, regular and irregular) and number of sensor nodes considering continuous WSNs. All experiments were performed using the MANNASim, a module for simulation based on the Network Simulator (NS-2). The MANNASim is a result of this thesis as well. Chapter 5 also presents the limitations, difficulties, and assumptions involved in the experiments conduction as well as it describes and discusses the results obtained in the

experiments; and finally, the MANNASim module is described.

Chapter 6 concludes the thesis, discussing the results obtained and future research.

In Appendix A, a fault management application is developed to evaluate a self-diagnostic scheme. All experiments were performed considering an event-driven WSN. This appendix presents the experiments as well as the results obtained.

Chapter 2

Wireless Sensor Networks

This chapter provides a prospective study on wireless sensor networks, including an overview of its basic characteristics (Section 2.1) and sensor node architectures (Section 2.2). The main differences among WSNs and other networks are addressed in Section 2.3. WSN and sensor nodes architecture are completely dependent on the purpose of the application. To illustrate the "application-dependent" characteristic of WSNs, Section 2.4 provides some applications of WSNs.

It has come to our knowledge that the MANNA architecture [84] is the only integrated management solution for WSNs that has been proposed in the literature. Thus, Section 2.5 aims to present an overview of the main research topics in WSNs. Some of these topics could be used to perform some management services proposed by MANNA but they are not directly related to the management architecture field. Finally, Section 2.6 a conclusion concerning the topics presented.

2.1 Wireless Sensor Networks Overview

The increasing sophistication of monitoring and controlling systems with multiple sensors has recently generated a great deal of interest in the development of WSNs. This provides distributed network access to sensors, actuators, and processors embedded in a variety

of equipment, facilities, and environments, representing a significant improvement over traditional sensors. WSNs aims to collect data and sometimes control an environment. This kind of network may consist of hundreds to thousands of sensor nodes that have the capability of sensing, processing and communicating using a wireless medium [53, 52].

The sensor nodes are deployed over an area as illustrated in Figure 2.1(B). They are able to discover their locations (see Figure 2.1(C)) and organize themselves as a wireless network (see Figure 2.1(D)). The node deployment can be done, for example, by dropping a large number of sensor nodes from an airplane in a certain area or placing them in this area by hand or using a robot. Figure 2.1 illustrates a WSN life-cycle phase called "network self-boot up" [78]. A WSN must be able to operate under very dynamic conditions. Moreover, it usually works unattended in remote areas.

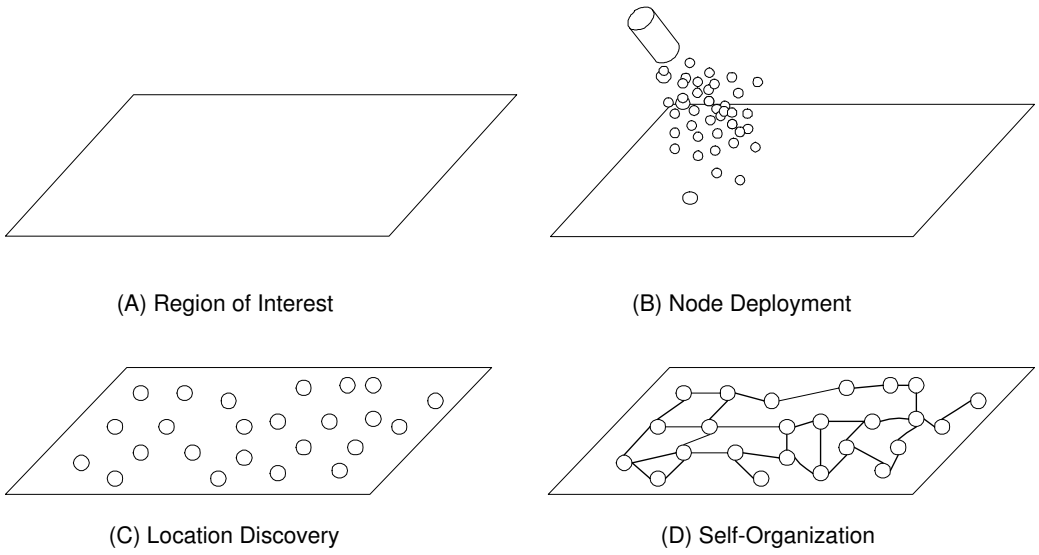


Figure 2.1: WSN boot up.

Once the network is formed and the sensor nodes are operating, most sensor nodes will be able to sustain a steady state of operation, that is to say, their energy reservoirs will be nearly full, and they will be able support all the sensing, processing, and communication

tasks required. In this mode, sensor nodes will constitute a multi-hop network. The sensor nodes begin to establish routes by which information is passed to one or more *sink nodes* (see Figure 2.2). Sink nodes are typical sensor nodes that usually differ from other types of sensor nodes in the following aspects: they have more energy, longer radio range and do not perform sensing. Furthermore, the sink node may be a data gatherer mobile node. In the literature, sink nodes are also called *monitoring nodes* [64]. Any other entity (non node) required to perform the functionalities of a sink node will be called *Base Station* (BS) in this work. The main difference between a sink node and a base station is that base station has no resources limitation. Sink nodes and BSs can serve the purpose of collecting information from the network and sending it to one or more external entities called observers. In this case, the sink node or the base station are performing the *access point* role. An access point can be composed of gateway functions to connect the WSN to the outside world [83]. Section 3.3.2 characterizes the WSNs and introduces a functional model containing an explanation about meaning of the main terms used in WSNs.

Because sensor data is intrinsically associated with the physical context of the phenomenon being sensed, spatial coordinates are often a natural way to name data [67]. Besides addressing (naming) purpose, the node location can be employed by routing protocols that use spatial addresses and by signal processing algorithms (e.g., beamforming) that are used for tasks such as target tracking. In some applications, the resource constraints of WSNs can be better met by an attributed-based naming system than by traditional approaches such as IP-addressing. Application-dependent systems such as WSNs can name and route data directly in application-level terms [34]. Thus, in some cases sensor nodes may not have global identification (ID) because of large overhead and large number of sensor nodes.

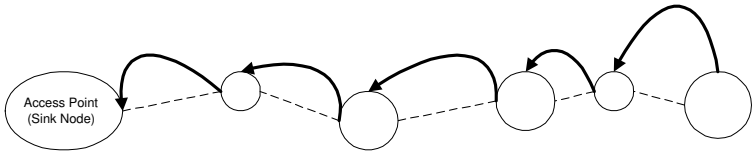


Figure 2.2: Multi-hop communication.

In the case of ad hoc deployment, the sensor nodes should be able to cope with the resultant distribution and form connections among them. The limited available energy and small form of the sensor nodes impose a limit on the radio transmission range and suggest small multi-hop transmissions schemes (see Figure 2.2). Although the multi-hop network can operate in both sensor-to-sink or sink-to-sensor (broadcast or multicast) modes, the bulk of traffic will happen in sensor nodes near to sink node. This is due to the fact that disseminated data from all source nodes to sink node use intermediate nodes, putting a significant strain on the energy resource of the sensor nodes near the sink and making that neighborhood more susceptible to energy depletion and failure. This situation is called energy wave problem. Figure 2.3 illustrates the energy wave problem which occurs due to multi-hop communication scheme. The dark region represents unavailable sensor nodes due to energy problem. However, sensor nodes may fail due to other reasons such as mechanical failure [97].

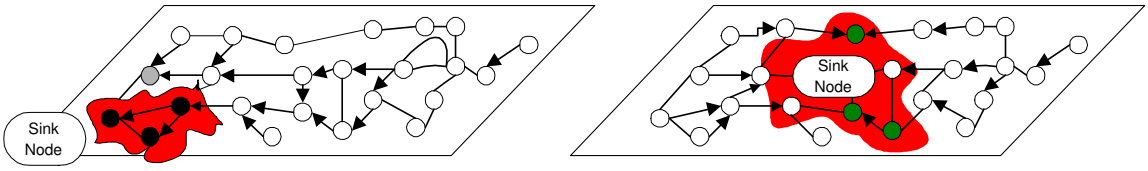


Figure 2.3: Energy wave problem.

When sensor nodes fail, the medium access control (MAC) and routing protocols must accommodate the formation of new links and routes to the access point(s) (sink nodes) [104]. This may require actively adjusting transmission powers and signaling rates on the existing links to reduce energy consumption, or rerouting packets through regions of the network where sensor nodes have more energy left. In the cases illustrated in Figure 2.3 the network is partitioned and the sink nodes become isolated. Solutions could be incremental sensor nodes deployment, sink nodes displacement (in case of mobile sinks) and backup nodes activation (if there are some) [103].

The failure of sensor nodes should not affect the overall task of the WSNs. This property is related to the dependability and fault tolerance topic [14, 98, 106]. Fault tolerance is the ability to sustain network functionalities without any interruption despite sensor nodes failures. In the context of WSNs, dependability and fault tolerance are discussed in [39, 54, 104].

Communication is the major energy consumer in wireless networks, especially data transmission. The transmission power required by communication between nodes is dependent on distance. For example the ground-to-ground transmission costs[1] 3 Joules of energy to transmit 1Kb of data a distance of 100m. On the other hand, a general-purpose processor with the modest specification for 100 million instructions per second per Watt (MIPS/W) processing capability executes 300 million instructions with the same amount of energy [75]. This example suggests placing nodes closer to each other in order to reduce energy consumption and local processing of data to reduce the amount of data to be transmitted [76]. To reduce the amount of power spent on long distance radio transmission and to minimize the energy wave problem, the sensor nodes can also be clustered [48, 74].

The clustering algorithms can include cluster-head (leader) election mechanisms such that each sensor node is associated with a cluster-head as its leader. The cluster-head–common-node relationships are established between sensors that are able to communicate with each other. The communication between common-nodes and cluster-heads can be multi-hop (as illustrated in Figure 2.4 (A)) or single-hop (as showed in Figure 2.4 (B)).

The large use of WSNs depends on the design and development of a scalable, low-cost, sensor nodes. WSN and sensor nodes architecture are completely dependent on the purpose of the application. The section below presents the main components of sensor nodes that can be applied to WSNs.

[1]Watt is a unit for measuring electrical power. Joule is a unit of energy or work. 1Watt = 1 Joule/second. Watt and Joule are represented by “W” and “J” respectively. MIPS means millions of instructions per second.

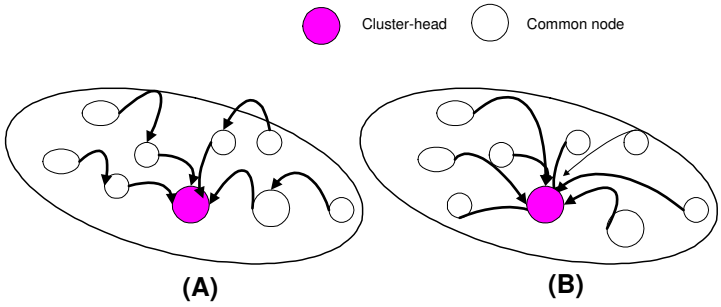


Figure 2.4: Cluster communication scheme.

2.2 Wireless Sensor Node Architecture

A wireless sensor node[2] is composed basically of a *power supply*, *computational module* (processor and memory), *transceiver*, and *sensor unit* (Figure 2.5). The physical and logical components of a wireless sensor node are presented below.

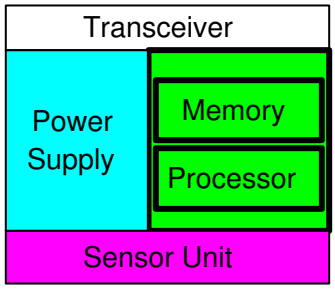


Figure 2.5: Components of sensor node.

Power Supply. The most widely used power supply in sensor nodes is the battery. The choice of the battery type is important since it can affect the design of the sensor node. Batteries are classified in the following types [88]: linear – the battery is considered to be a bucket of energy that is linearly drawn from this bucket by the energy consumers; dependent model– it considers the rate in which the energy is drawn from the battery to compute the remaining battery lifetime at high discharge rates and the capacity of the

[2]More details about main node architectures were published in a Technical Report of the Computer Science Department of UFMG (Universidade Federal de Minas Gerais), Brazil [93].

battery is reduced; and relaxation model – it takes into account a phenomenon seen in real-life batteries where the battery's voltage recovers if the discharge rate is decreased.

Computational Module. Composed by processor and memory. It permits sensor node to process local data. Developing a node for ultra-low power represents a critical challenge. In the case of processors, *low power* is a quality of a device that consumes low energy per clock. A device that consumes low energy per instruction is called *energy-efficient*. For example, the ATMega128L@4MHz processor consumes 16.5mW and its efficiency is 242 MIPS/W, spending 4nJ/instruction. The ARMTumb@40MHz processor consumes 75mW and its efficiency is 480 MIPS/W, spending 2.1nJ/instruction [93]. Many different types of computational modules (with processors or micro-controllers) can be integrated into a sensor node. Examples are the AT90LS8535 (4MHz, 35 pines, consumes 19.2mJ/s in active mode, 5.7mJ/s in idle mode, and 3μJ/s in sleep mode, 8 bits and 512B of RAM), and the Intel StrongARM1100 (133MHz, 32 bits, 150 MIPS/W, 16KB instruction cache, 8KB data cache, 128KB of SRAM, and 1 MB of flash memory) [93].

Transceiver. The transceiver connects the node to the network. The main types of transceivers are: radio frequency (RF), infrared and optical. Each technique has its advantages and disadvantages. An example of transceiver radio frequency is the TR1000 which has 916MHz or 433MHz of frequency, with transmission rate of 50 Kbps and ranges from 30 to 90 meters. An optical transceiver using a laser module and a Corner Cube Reflector (CCR), which has 0.5 x 0.5 x 0.1mm^3, can transmit at a rate of the 10Kbps consuming 1μWatt to 1Km of range [93].

Sensor Unit. Sensor unit can be composed of one or a group of sensors which are devices that produce electrical response to a change in physical conditions. Sensing devices generally have widely different theoretical and physical characteristics. Besides, they can have different design, manufacturing, modelling, and signal processing. Thus, numerous models of varying complexity can be constructed based on application needs and device features (e.g. pressure, light, humidity, luminosity, acceleration, mechanical stress, audio, video, temperature, angular rate, force, acoustic, hysteresis, ultrasonic, flow meter, optoelectronic/photonic, ionizing radiation, surface plasmon resonance, viscosity, proximity,

pH, gas, radiative, altitude, chemical, biological, microbalance, medical, and so on).

Software. It is used to represent a set of programs and procedures which becomes an autonomous system capable of performing the information processing, relaying or routing, and management tasks. As previously seen, sensor nodes have strong hardware and software restrictions in terms of processing power, memory capacity, battery lifetime, and communication throughput. These are typical characteristics of mobile and wireless devices and not of wired network elements. Thus, software designed for sensor nodes must consider those limitations [97], whereas an element for a wired network may have other restrictions such as performance and response time.

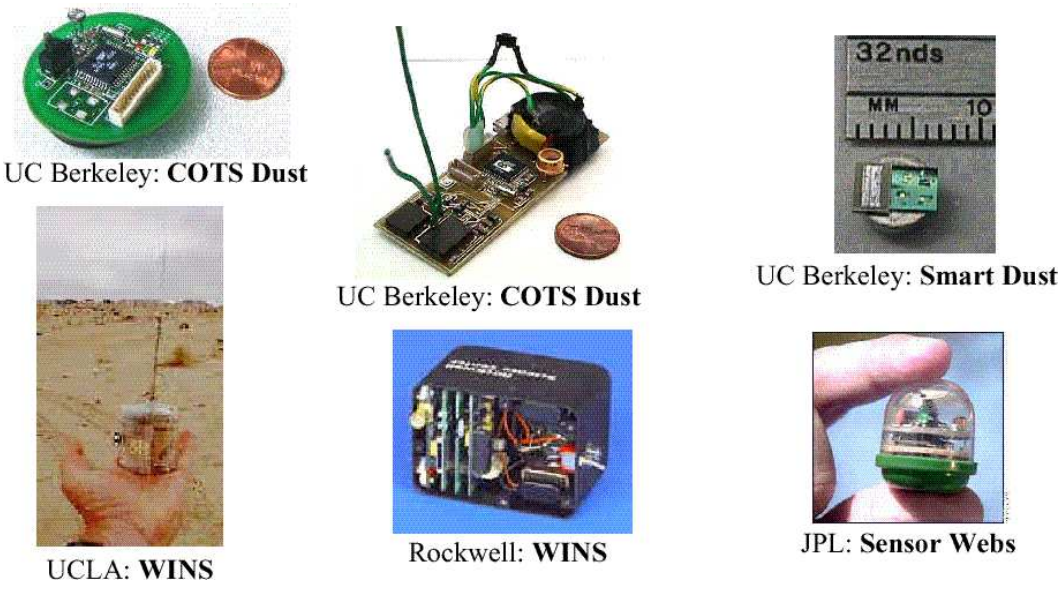


Figure 2.6: Wireless sensor nodes projects.

A sensor node may also have additional-dependent components such as location finding system, power generator, and mobilizer [7]. All of these units are expected to have small dimensions, consume ultra low power, operate in high volumetric densities, have low production cost, be disposable and autonomous, operate unattended and be adaptive to the environment [7, 76]. These design factors are addressed by many researchers. Figure 2.6

shows some wireless sensor nodes such as Smart Dust [5] from University of California, Berkley, WINS (Wireless Integrated Network Sensors) [6] from UCLA and Rokwell and JPL *Sensor Webs* [3] by NASA's Jet Propulsion Lab.

The mobile and static wireless sensor nodes have the ability to gather spatial as well as temporally dense data over vast geographical areas. The cost of a single node is very important to justify the overall cost of the network. In some cases, the network cost is more expensive than the cost of the traditional wired sensor networks, although the WSN is cost-justified by benefits. The cost of a sensor node should be much less than US$1 for the WSNs to be feasible but currently they are more expensive. The cost of a Mica-Motes [1], for instance, is about US$200.

2.3 Differences Among WSNs and Other Kinds of Network

A WSN differs from other types of networks basically in the following aspects: number of elements, ad hoc and unattended elements deployment, hardware and software restrictions, unattended operations, addressing, and routing. The number of sensor nodes in WSNs can be several orders of magnitude higher than the number of nodes in an ad hoc network. In general, dense deployment allows greater sensing task and also fault tolerance through a high level of redundancy. Due to sensor nodes deployment in environments where the nodes may be lost or destroyed, and in cases where sensor nodes cannot be carefully positioned relative to each other and the environment, an alternate strategy to achieve coverage is to deploy greater density of elements. In some contexts, even if the elements are uniformly placed in three dimensional-space, environmental conditions might be such that the coverage is not uniform due to obstacles and other sources of noise. Another motivation for using a large number of sensor are the cases where the incremental cost of deploying a node during initial deployment is much lower than the incremental cost of deploying new nodes or renewing node resource.

Sensor nodes have strong hardware and software restrictions in terms of processing power, memory capacity, battery lifetime, and communication throughput. In traditional mobile networks, energy consumption is of secondary importance as the battery packs can be replaced when necessary. However, in WSNs the main physical restriction is the available energy, since batteries are not recharged during the operation of a sensor node because of operations in hostile or remote environment and the number of nodes. All activities performed by the node must take into account energy consumption. As a result, the design of software for wireless sensor nodes must consider these limitations whereas its wired network element counterpart have restrictions such as performance and response time.

In computer networks, the replacement of faulty components or resources by technicians is an ordinary operation. The network tends to follow a well-established planning of available resources and the location of each of its elements is well-known. In a WSN this is not often the case, since the nodes are ad hoc and deployed unattended, and the network is planned to have nodes discarded, lost, and out of operation temporarily or permanently. The topology changes very frequently, even if the nodes are stationary after deployment. In this scenario, faults are common, which is not expected in a traditional network. In fact, the initial configuration of a WSN can be quite different from what is expected in the case of nodes thrown in the ocean, in a forest and other remote environments. Dynamic environmental conditions require the network to adapt over time to state changing and unpredictable environmental situations. Unattended operation requires automatic configuration and reconfiguration (self-configuration). Ad hoc deployment requires the system to identify and cope with the resulting topology and connectivity of nodes (self-organization) [27].

Inherent to the design of most distributed systems today is the assumption that each node has a unique network address. This address appears in every packet to identify its source and destination [25]. Depending on the WSN application, it may or may not be interesting to identify uniquely each node in the network. The cost of an address in an energy-constrained network can be considerably high if the address space is under-

utilized and the address itself accounts for a significant portion of the total number of bits transmitted.

A common alternative in WSNs is to use attributed-base naming. Data is named by attributes and applications request data matching a certain attribute value. Furthermore, observers may be interested in a value associated to a given region and not to a particular node. For instance, any observer may be interested in the temperature at the top of a mountain. WSNs are typically data-centric, which is not a common characteristic of traditional computer networks.

In most of WSNs, data flow is predominantly unidirectional, that is, data flows from node source (producer) to access point (sink node, monitoring node or base station). Sensor nodes usually do not have a direct communication channel to sink nodes, which demands intermediate nodes to act as routers to send communication messages. In this architecture, each sensor node is also a potential router. The links can be formed by radio, infrared, or optical media. The protocol stack must combine power and routing awareness, integrate data with networking protocols, communicate power efficiently through the wireless medium, and promote cooperative efforts among sensor nodes. Although many protocols and algorithms have been proposed for traditional wireless ad hoc networks, they are not well suited to the specific features and application requirements of WSNs [7].

WSNs are heavily dependent on the purpose of the application. They are employed in specialized tasks and their nodes cooperate among themselves to perform a huge task. The following sections introduce the main applications and the work proposed for WSNs.

2.4 WSNs Applications

WSNs have the advantage of spanning a large geographical area and being able to detect and track events collaboratively. A number of high profile applications for WSNs have been proposed [8, 29, 50, 58] to monitor the environment, detect, classify and locate specific events, and track targets over a specific region. All WSN parameters can vary depending on the application considered. For example, the deployment can be either predetermined

- when the environment is sufficiently known and under control, such that the sensors can be strategically hand-placed; or undetermined a priori - when the environment is unknown or hostile, in which case the sensors may be airdropped from an aircraft or by other means. Despite the diversity of applications, the following features are true for all of them: low power, low cost, wireless and ad hoc, the most important being the former. Among various applications for WSNs, the most interesting are described in the following paragraphs.

- **Tracking a chemical plume.** Each sensor by itself has limited information such as whether certain chemical element exists or not at the sensing spot, whereas global information such as the shape of the plume and its motion need to be determined collaboratively by many sensors. In addition, because of the limited node energy reserves, such processing and communication must achieve energy efficiently. Jie Liu [51] addresses this type of problem using physical constraints to dynamically define sensor collaboration regions. In this work, he developed a laboratory testbed of a two-dimensional, wireless connected sensor field using 16 Berkeley motes [1] sensor nodes. They present a shadow edge detection and power management scheme using a dual-space transformation.

- **Disaster area surveillance.** Several thousand sensors are thrown from an airplane and rapidly deployed in a disaster area. The sensors communicate and coordinate to form an ad hoc communication network. Emergency response teams can disseminate concurrent queries into the WSNs to collect information in the disaster area. The queries are automatically routed to the most appropriate sensors, and replies are collected and sent to the designated reporting points. The disaster area can also be monitored to alert emergency response teams for changing situations. Chavalit Srisathapornphat *et al* [91] model a sensor network as a collection of massively distributed objects, and define a middleware that allows applications to issue queries and collect replies. If the region affected by the disaster includes fire, the sensor nodes are destroyed and the remaining nodes inform safe evacuation paths, for example.

- **Civil applications.** There are many of varied nature. One example is pollution

detection along beaches with sensor nodes distributed along the shoreline. WSNs can also be spread throughout the exhaust system of an urban area to detect level of air quality. WSNs can also be used where traditional methods represent impractical or expensive solutions, for instance in the use of WSN in aircrafts to avoid cable weight. They can also be used in each item of an inventory in a factory warehouse or office complex, attached to walls, or embedded in floors and ceilings, tracking the location history and use of items. The WSNs can automatically locate items, report on those needing servicing, analyze long-term correlations between workflow and wear, report unexpected large-scale movements of items or significant changes in inventory levels [28].

- **Intelligent Transportation Systems.** Kanaian [47] developed a wireless sensor package that counts passing vehicles, measures the average roadway speed, and detects ice and water on the road. Clusters of sensors can transmit this information in near real-time to wired base stations for controlling and predicting traffic, and in clearing road hazards. The nodes cost much less than US$30 to manufacture, and can be installed without running wires under the road. The devices notice vehicles by detecting the perturbations caused by vehicles in the magnetic field of the Earth. Another example would be the use of sensors attached to every vehicle in large metropolis has one or more attached sensors. These sensors are capable of detecting their location, vehicle size, speed and density, road conditions and so on. As vehicles pass each other, they exchange information summaries. These summaries are diffused across sections of the metropolis. Drivers can plan alternate routes, estimate trip times, and be warned of dangerous driving conditions [28] .

- **Habitat monitoring.** This type of application provides a rich collection of sensing modalities and environmental conditions. Cerpa *et al* [15] proposed an application which the goal is to support data collection and model the development of complex ecosystems. According them, scientists and environmental impact monitoring authorities would like to monitor soil and air chemistry, as well as plant and animal

species populations and behavior. For the latter, the primary modalities are imaging and acoustics to localize, identify and track species or phenomenon based on implicit signals (acoustic and seismic), or explicit signals (RF tags). These facilities must be deployable in remote locations that lack installed energy and communication infrastructures, motivating the need for low-power wireless communication [27, 15].

The strategy for node cooperation has significant consequences in terms of communication bandwidth and energy consumption. Estrin *et al* [27], give an example that considers the task of identifying bird species surveyed by several cameras. If it is to be accomplished through image analysis, the observer's access to the video is a very costly process. Alternatively, it could be possible to stream audio to a central location (cluster-head, sink node or monitoring node), which then performs signal processing to identify and stream back only those streams that are most likely to contain a target species. According Estrin *et al*, while this reduces the communication overhead greatly, it still suffers from communication latency and lacks scalability due to the need to stream audio through a central processing point.

Another alternative is to distribute the problem further, hosting the audio signal processing software on the nodes, and developing algorithms that require only local cooperation to make a decision to capture images. This approach can be scalable in that no long-range streaming of audio or video is necessary, resulting in more efficient use of communication bandwidth and limited energy resources but all this depends on the processing capabilities. Mainwaring *et al* [56] provide an in-depth study of applying wireless sensor networks to real-world habitat monitoring. A set of system design requirements is developed for covering the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and control. The deployed network consists of 32 Mica-Motes nodes on a small island off the coast of Maine streaming useful live data onto the web. The network represents a substantial advance over traditional invasive methods of monitoring. In the Great Duck island, seabird colonies are notorious for their sensitivity to human disturbance.

Research in Maine suggests that even a 15 minutes visit to a cormorant colony can result in up to 20% mortality among eggs and chicks in a given breeding year.

- **Monitoring forests, volcanos, twisters, and so on.** An application for environment centered sensing that wants the WSN to report the occurrence of a critical event (forest fire, volcano eruption, spinning column of air) with minimum delay, also providing information about its location.

- **Small scales ecosystem monitoring.** Remote sensing from satellite and airborne sensor has proved to be a powerful tool for studying "large" biodiversity (e.g. spatial complexity of dominant plant species). While many scientists and land managers attempt to study biodiversity using top-down remote sensing tools, the vast majority of the biodiversity and resulting biocomplexity within an ecosystem exists at very small scales. Besides, it is not readily observable with even the best airborne and satellite-based sensors [15]. WSNs offer opportunities to answer the key questions posed by biocomplexity.

- **Tracking enemy in military applications.** In this type of application, a WSN is deployed in a field to monitor movements of enemy tanks which are considered targets. The movement of tanks is detected by the seismic sensor (geophones) on the sensor nodes and the target detection is propagated back to the control center where the information can be further analyzed by observers (a central server and human operators) [70].

- **Helping fight against nuclear terrorism and other threats.** Researchers are focusing on systems for detecting and tracking threats. This kind of system has several denominations such as correlated sensor networks, wide-area tracking system, sensor or network fabrics; but the concept behind them is the same: an easy-to deploy system with number of wireless sensors (for instance, seismic, magnetic, pressure, acoustic, nuclear, or particle-counting) tied together with a communication network, and a scheme for fusing the data (that is, converting the data into forms

easily interpreted by users). Such correlated sensor networks can help detect a nuclear terrorist attack, track the movement and characteristics of a wildfire, assist military operations in taking out a target, determine earthquake damage to large structures such as bridges, and even protect the important authorities. The Wide-Area Tracking System (WATS) of the Livermore National Laboratory is one example of a correlated sensor network. Livermore researchers have been working on many applications. For instance a prototype for detecting and tracking a ground-delivered nuclear material. Another example of correlated sensor network development involves a concluded project called Joint Biological Remote Early Warning System (JBREWS) which uses biodetectors to provide U.S. field troops with early warning off a biological attack [38].

- **Human-embedded smart sensor network.** Implanted biomedical devices have the potential to revolutionize medicine. Smart sensors, which are created by combining sensing materials with integrated circuitry, are being considered for several biomedical applications, such as glucose level monitors or retina prosthesis. These devices require the capability to communicate to an external computer system (base station) via a wireless interface. The limited power and computational capabilities of smart sensor-based biological implants present challenges to the researchers in several aspects of wireless networking. This is due to the need of having a bio-compatible, fault-tolerant, energy-efficient, and scalable design. Furthermore, embedding these sensors in humans imposes additional requirements. For example, the wireless networking solutions should be ultra-safe and reliable, work trouble-free in different geographical locations (although implants are typically not expected to move, they should not restrict the movements of their human host), and require minimal maintenance. This requires few and specialized sensor nodes. Schwiebert *et al* [90] describe the potential of biomedical smart sensors. They explain the challenges of human-embedded smart sensor array for wireless networking and a preliminary approach for wireless networking of a retina prosthesis.

- **Planetary Exploration.** WSNs can replace spacecrafts that have been orbiting other planets such as Mars. A spacecraft orbiting Mars has detected large quantities of water-ice just below the surface of the planet. The American Space Agency NASA has invested in research about WSNs for interplanetary discovery, called Sensor Webs [3].

Several aspects of WSNs applications presented in this section pose design challenges and research opportunities. Different proposals to solve these new challenges can be found in the literature. Nevertheless, WSNs management is still an open issue. The following section offers an overview of the main concepts in the WSNs field.

2.5 Related Work

To the best of our knowledge and in accordance with [102], the framework proposed in this thesis and published in [84] is the only integrated management solution for WSNs that has been proposed in the literature so far. Thus, this section presents an overview of some research topics in WSNs. Even though these topics are not directly related to management architecture or solution theme, they can be used to perform some management services proposed by MANNA (see Chapter 4).

WSNs is likely to provide one of the missing connections between the Internet and the physical world. The implementation of WSNs needs to satisfy the constraints imposed by factors such as fault tolerance, scalability, cost, hardware, topology dynamics, unattended operations, and power consumption. There has been a number of research projects and efforts in all levels of development and usage of WSNs, including the topics described below.

-*Localization* is the mechanism whereby a sensor node estimates its spatial coordinates. A list of published work on this topic include [12, 13, 27, 30, 66, 87]. Procedures for algorithmic location discovery can be classified in two large groups: those used in fixed infrastructure wireless systems and those used in wireless ad hoc systems. In the first

group, the most notable location discovery system includes Automatic Vehicle Location (AVL) [77] and Global Position System (GPS). The second group has only recently become the focus of study [59]. However, until now, there has been no agreement on the feasibility, efficient-energy and scalability of these methods [76].

-*Self-organization* is the property which the wireless sensor nodes must have to organize themselves to form the network [18, 55, 97]. The efficiency of this organizational process can be heavily dependent on the particular deployment of the network, the accuracy in the location discovery, and the degree and accuracy of the information that is programmed into the nodes [84]. For example, if all nodes are powered up simultaneously, their attempts to find one another will be subject of heavy contention [79]. In general, previously published works present self-organization algorithms that work in the boot up phase or time of the network (see Figure 2.1). Nevertheless, it is not clear if these algorithms can adapt themselves dynamically to network changes. Other open issue is the small number of nodes used in experiments considering that a WSN is usually composed of hundreds to thousands of sensor nodes. In the majority of these work, the sensor nodes are assumed not to be mobile.

-*Topology Discovery.* Deb *et al* [20] describe a topology discovery algorithm (TopDisc) for WSNs which will can be used in application to network management as proposed in [69]. The algorithm finds a set of distinguished nodes, using whose neighborhood information and building the approximate topology of the network. TopDisc forms a Tree of Clusters (TreC) rooted at the sink node, which initiates the topology discovery process. However, the work in [69] is still a preliminary investigation to define, in future, a protocol similar to the protocol SNMP (Simple Network Management Protocol) [108]. Other work about topology are [89] and [16].

-*Network density* can be expressed in terms of the number of nodes per nominal coverage area. Thus, if N nodes are scattered in a region of area A, and the nominal range of each node is R, the network density $\mu(R)$ is $\frac{N.\pi.R^2}{A}$. Note that in the equation the range R can be either the range of a particular sensor or the radio transmission range (idealized with circular propagation). In each case, the associated network density will be different [67].

- *Exposure* is a measure of how well an object, moving on an arbitrary path or not, can be observed by a sensor network over a period of time. Meguerdichian *et al* [59] provide formal, yet intuitive, formulations to establish the complexity of the problem and develop practical algorithms for exposure calculation. They studied how errors in location discovery impacts the calculation of exposure and how one can statistically predict the required number of sensors for a targeted level of exposure. Exposure is directly related to the coverage area of network.

Most of the work proposed for density, exposure, and coverage in the literature are either theoretical or define an algorithm which treats this problem in initial phases of the network. The management solution proposed by this thesis could use these functions during the boot up time of the network, according to the application type. A key problem in such a maintenance scheme could be used to select which node to shutdown and which node to turn on at any given instant. In this direction, we have proposed a method to schedule sensor nodes in [103]. In our research, in order to control the network density a management function was defined based on a criterion employed to decide which nodes should be turned on or off. The management function is part of the management service which can take the sensor node out of service temporarily to perform coverage maintenance. The proposed solution is based on Voronoi diagrams which decompose the space into regions around each node, to determine which sensor node could be administratively put out of service. The results show that the use of control density management function can save energy without losing the sensing area. This schema is used in the MANNA architecture (see Section 4.3). A Voronoi diagram has already been applied to solve other problems in a wireless sensor network. Meguerdichian *et al* [58] proposed an algorithm for calculating the maximal breach and maximal support paths in a sensor network based on a Voronoi diagram.

-*Energy* is a critical resource in WSNs. The rate of energy consumption is related to the power. If the power is not managed efficiently, the lifetime of the power supply (batteries) will be shortened and the longevity of the network will suffer. The goal of being energy-efficient can be translated into the problem of optimizing the number of operations needed

to be performed. Bhardwaj *et al* [9] bring a perspective on energy usage establishing upper bounds in the lifetime of a WSN. Mini *et al* [64] propose mechanisms to construct the energy map of a WSN energy-efficiently. This map is constructed using a prediction based model of the dissipated energy in each node. Goel *et al* [31] propose a new paradigm of operation in sensor networks called PREMON (PREdiction-based MONitoring). The PREMON paradigm prevents a sensor from unnecessarily transmitting all the readings that can be successfully predicted at the monitoring entity, thereby saving energy. This saving is obtained at the cost of extra computations. A way of obtaining aggregated information and energy data is the Residual Energy Scan defined by Zhao *et al* in [111]. Instead of providing detailed information about the residual energy at individual sensors, the scan provides an abstract view of the energy resource distribution. The routing process tending to overload the parent node is a problem with the Residual Energy Scan. Routing is done in a way that every message originated in a node is sent to another node, called the parent node which is closest to the sink node, and so forth. The parent node tends to receive and transmit more messages than its descendants, consequently leading to an unbalanced consumption of energy in the network. Therefore, nodes closer to the access point (sink node) will receive and transmit more messages than nodes that are far away. This can cause a premature death of the network, even though there may exist nodes with enough energy to execute services. However their messages will not reach the access point (see Figure 2.3).

Another problem with the Residual Energy Scan is that it is not fault-tolerant to a node failure during a scan. Suppose there is a node malfunctioning during a search. The data sent by the node's descendants will not be available to the access point and these nodes will become orphans. Note that in a WSN a node failure is a common case, not an exception. We propose a solution to the two problems identified in [111]. In this work, the principles of Smart Sink and Stepfather are defined and applied to the spanning routing tree algorithm [111] to extend the WSN lifetime and make it fault-tolerant. Smart Sink is a sink node which has extended functionalities, and fixes the spanning routing tree, spending one message per node. Stepfather consists of a sensor node that has a list

of potential parents (called stepfather's list) for routing messages. In [104], we define the stepfather discovery algorithm which is used to build the stepfather list. When a node loses its father, it can substitute it by a node in the stepfather's list. This mechanism introduces fault tolerance to the spanning routing tree algorithm, correcting the tree dynamically. Simulation results show that the WSN lifetime can increase three times compared to the previous work. In [92], we propose another solution to obtain states of the network, for example energy map. In our work, the algorithms Distributed Snapshot and Broadcast and Propagation of Information with Feedback (PIF) were adapted to WSNs and applied to generate the energy map of a WSN. This map shows the behavior of this network and can be used in the MANNA architecture to represent the network state.

-*Quality of Service.* Due to the basic characteristics of the ad hoc and sensor networks the term QoS (Quality of Service) has very different meanings in each of them. In ad hoc networks the main purpose of providing QoS is to guarantee a high throughput, low delay, and jittering, in resume, to maintain the communication efficiency even with the nodes mobility. In contrast, there are WSNs in which the main purpose is to save energy and, when the application requires, the quick deliver of a high priority message. Many authors have proposed solutions to QoS in ad hoc networks. The most referenced have been CEDAR (Core-Extraction Distributed Ad hoc Routing algorithm) [96]. Sudeept Bhatnagar *et al* [10] introduced the concept of service differentiation based on data prioritization and argued that service differentiation is inherently required in sensor network. They presented a simple forwarding algorithm called Adaptive Forwarding Scheme (AFS) which allow to control the reliability of a sensor network's communication. They assume that the sensor network does not have any acknowledgment mechanism for reliable packet delivery and that the network is formed by random deployment of sensor in a field. On WSNs this field is budding. In the scope of this thesis we have some published work about quality of service in WSNs [79, 81, 82, 85].

-*Dissemination and communication.* Energy-efficient data dissemination is among the first set of research issues being addressed [50, 95, 110]. The main algorithms proposed in the literature are LEACH (Low-Energy Adaptive Clustering Hierarchy) [35], PEGASIS

(Power-Efficient Gathering in Sensor Information Systems) [49], SPIN (Sensor-Protocol Information Negotiation) [36], and DD (Directed Diffusion) [42]. These protocols are still being improved and other new protocols are being developed to address higher topology changes, higher scalability, and energy efficiency. Other work in the field are [15, 36, 50, 90, 99].

-*Topology Discovery.* Deb *et al* [20] describe a topology discovery algorithm (TopDisc) for WSNs which will can be used in application to network management. The algorithm finds a set of distinguished nodes, using whose neighborhood information and building the approximate topology of the network. TopDisc forms a Tree of Clusters (TreC) rooted at the sink node, which initiates the topology discovery process. However, this work [20] is still a preliminary investigation to define, in the future, a protocol similar to the protocol SNMP (Simple Network Management Protocol).

Other topics about WSNs such as link protocols [65, 107, 110], data gathering [50], hardware architecture [1, 2, 3, 4, 6, 9, 37, 63], operating systems [37], build naming [24, 25, 34], synchronization [26], and so on have been published.

2.6 Conclusion

This chapter attempts to show that the WSNs present many and drastic different challenges. The number of sensor nodes in WSNs can be several orders of magnitude higher than the nodes in an ad hoc network. In WSNs, sensor nodes are densely deployed, limited in power, in computational capacities and in memory, and are prone to failures. The topology of WSNs changes very frequently. Sensor nodes may not have global identification (ID) because of the large overhead and number of sensors. The position of sensor nodes cannot be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relied operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Another unique feature of WSNs is the cooperative effort among sensor nodes. The deployment of these networks requires wireless ad hoc network techniques. Although many protocols and algorithms have

been proposed for traditional wireless ad hoc networks, they are not adequate to the unique features and application requirements of WSNs. Several interesting work may be found in the literature about specific topics in wireless sensor networks (see Section 2.5). Some of these proposals treat specific functions for determining WSNs, for instance, routing in flat networks that do environmental monitoring.

Until now, WSNs and their applications have been developed without taking into consideration integrated management solution. In [84] we propose a management framework, in particular an architecture for wireless sensor networks management, called MANNA. In the following chapters, we present the MANNA architecture and its management dimensions aiming at integrated management solutions.

Chapter 3

A Novel Organization for WSN Management

This chapter[1] focuses on the problem of managing WSNs and proposes an organization to be used in the design of management solutions for different WSNs. This novel organization is a proposal of this thesis as well. Managing WSNs is a task significantly harder than managing other networks because of the reasons presented in Chapter 2. All of these distinguishing characteristics will potentially affect the management solution design. The management of large networks requires powerful abstractions which permit the identification of management functions in different levels. Section 3.1 discusses the self-managing paradigm chosen as an approach to manage WSNs. Section 3.2 presents the management functional areas and management levels as defined for traditional management. Section 3.3 discusses the two traditional management dimensions under WSNs perspective and proposes a novel management dimension called WSN functionalities. Clearly, the management functional areas and the management levels must be rethought for WSNs. In this sense, the following sections are contributions to the field, since such discussion was not found in the literature. The contributions of this chapter have been published in [79, 81, 82, 83, 84].

[1]The contents of this chapter will be published in 2004 as a chapter in the book entitled "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems. Edited by Mohammad Ilyas and Imad Mahgoub. CRC Press" [82].

3.1 WSN Management

One of the major goals of network management is to promote network resources productivity and to maintain the quality of the service provided. Given the discussion in Chapter 2, where some unique characteristics of WSNs were outlined, what would be the challenges for managing these more complex and dynamic networks? It is clear that there are several significant differences between the management of traditional networks and WSNs. Probably, the fundamental issue about the management of a WSN is concerned with how the management can promote both plant and resources productivity, and how it integrates, in an organized way, functions of configuration, operation, administration, and maintenance of all elements and services. Energy is a critical resource in WSNs. Thus, all operations performed in the network should be energy-efficient, including the management tasks.

This thesis proposes that a WSN must be self-managed. A self-managed WSN is responsible for configuring and reconfiguring itself under varying (and in the future, even unpredictable) conditions. System configuration ("nodes setup" and "network boot up") must occur automatically. Moreover, dynamic adjustments need to be done for the current configuration to best handle changes of the environment and of itself.

A self-managed WSN must look for ways to optimize its functioning; it will monitor its components and fine-tune its workflow to achieve predetermined system goals; it must perform something akin to healing – it must be able to recover from routine and extraordinary events that might cause some of its parts to malfunction. The network must be able to discover problems or potential problems, such as uncovered areas, and then find an alternate way of using resources or reconfiguring the system to keep it functioning smoothly. In addition, it must detect, identify and protect itself against various types of attacks in order to maintain the overall system security and integrity.

A self-managed WSN must also know its environment and the context surrounding its activity, and act accordingly. The management entities must find and generate rules to perform the best management of the current state of the network. A self-managed WSN with such characteristics, can be called an autonomic system [40]. An autonomic system is

an approach to self-managed computing systems with minimum human interference. This term derives from the autonomic nervous system of the human body, which controls key functions without human awareness or involvement.

The processors in autonomic systems use algorithms to determine the most efficient and cost-effective way to distribute tasks and store data. Along with software probes and configuration controls, computer systems will be able to monitor, adjust and even repair themselves without requiring technology staff – at least, that is the goal [40].

Therefore the WSN management must be autonomic, i.e., self-managed and robust to changes in network states while maintaining the quality of service. In the scope of this thesis, the term autonomic means the capabilities of self-discovery, self-configuration, self-organization, self-diagnostic, self-healing, self-maintenance, self-optimization, self-protection, self-service, and self-awareness which a WSN has. This capabilities are described in Section 4.3. The scheme to define automatic services that can be used in self-management is introduced in Chapter 4. Depending on the WSN application, it may be interesting or not to implement a certain management solution. The computational and energy costs of autonomic processes can be very expensive for some WSN architectures, even for services and functions performed semi-automatically or manually.

The task of building and deploying autonomic management systems, in environments where there will be tens of thousands of network elements with particular features and organization, is very complex. This task becomes even worse due to the physical restrictions of the sensor nodes, in particular, energy and bandwidth restrictions. The management application to be built also depends on the type of the application being monitored. A good strategy is to deal with complex management situations by using management dimensions. The following section presents the two management dimensions used in traditional management.

3.2 Management Dimensions

In general, for traditional networks, management aspects are clearly separated from network common activities, i.e., from the services they provide to their users. It is also said that there is an overlapping of management and network functionalities, although the implementation can be thought independently. This separation can be achieved by using two traditional management dimensions, called management functional areas [44] and management levels [45]. Nevertheless, carrying out these two traditional management dimensions will require new approaches based on the WSN characteristics.

The requirements to be satisfied by systems management activities can be categorized into functional areas. These facilities are known as the Specific Management Functional Areas (SMFAs): fault management, configuration management, performance management, accounting management, and security management. This has proven to be a helpful way of partitioning the network management problem from an application point of view [44].

- Fault management involves discovering, isolating, and fixing problems in the network. This functional area is responsible for ensuring smooth and continued operation of the network.

- Configuration management involves the initialization and shutdown of the network. It also involves maintaining, adding, and updating new network components. Part of the function of configuration involves defining relationships between network entities.

- Security management involves controlling access to network components and information. This component is also responsible for implementing encryption and decryption schemes for secure end-to-end communication.

- Performance management involves collecting network statistics and tuning the network to improve performance.

- Accounting management involves tracking network utilization by various users and groups. This information can be very useful in network configuration and allocation

of network resources to the various groups in an organization.

To deal with the complexity of management, the management functionality with its associated information can be decomposed into a number of logical layers, namely business management, service management, network management, and network element management. The architecture that describes this layering is called the Logical Layered Architecture (LLA) [45]. The business management layer is responsible for the management of the whole system. This layer has a broad scope, with the communication management being just a part of it. Business management can be seen as a goal setting, rather than goal achieving. For this reason, business management can be better related to strategic and tactical management instead of to operational management. The service management layer is concerned with the management of those aspects that may directly be observed by the users of the network. These users may be both final users (customers) and other service providers (administrations). The responsibility of the network management layer is to manage the functions related to the interaction between multiple elements. At network management level, the internal structure of the network elements is not visible. Element management layer is used to manage each network element individually. This layer deals with specific management functions and hides these functions from the layer above, the network management layer [45]. Thus, the management activities can be clustered into layers and later decoupled by introducing manager and agent roles. A logical layer reflects particular aspects of management and implies the clustering of management information supporting this aspects. Typically, there is an interaction between adjacent layers but due to operational and management considerations, other interactions may also occur between non-adjacent layers.

The use of the management dimensions is a good strategy to deal with complex management situations become it decomposing a problem into smaller sub-problems in successive refinement steps and to provide a separation between application and management functionalities through a management architecture. As a result, this makes possible the integration of organizational, administrative, and maintenance activities for a given net-

work.

This thesis proposes that WSN management be simple, adherent to network idiosyncrasies (including its dynamic behavior) and efficient in the use of its scarce resources. The adoption of a strategy based on the traditional framework of functional areas and management levels will allow management integration in the future. However, for WSN management it is necessary to go further. Using management functional areas and management levels is not enough because WSNs are application-specific. Therefore, a novel management dimension called WSN functionalities is proposed.

The next section discusses how the traditional management dimensions can be applied in the WSNs management and introduces a novel management dimension, considering the general aspects of the different types of the networks.

3.3 Dimensions for WSN Management

WSNs are embedded in applications to monitor the environment and act upon it. Thus, the management application should try to be "compatible" with the type of application being monitored. This thesis establishes that in order to have a better development of the WSN management services and functions it is necessary to characterize the WSN and establish a novel management dimension. Therefore, looking at the characteristics of various WSN applications, five main WSN functionalities are established: configuration, sensing, processing, communication, and maintenance. These functionalities define a novel dimension for the management, as depicted in Figure 3.1. Configuration is the first functionality before the network starts sensing the environment, processing and communicating data. Maintenance treats specific characteristics of the WSN applications during the entire network lifetime.

In this way, the WSN management will have an organization that comes from abstractions offered by management functional areas, management levels, and WSN functionalities (configuration, sensing, processing, communication, and maintenance). The novel dimension introduced can be observed in the upper part of Figure 3.1 together with the functional

management areas and the management levels on each side of the cube. The intersection of the three planes define a cell. Each cell contains a set of management functions.

The coordination among the three planes can be based on policies. Policy-Based Network Management (PBNM) [19] is a feasible alternative because it allows the manager to set actions to be carried out by the network without worrying too much about network details. Managers can define suitable actions in due time and still have a global or local view of the network. PBNM helps to manage complex networks such as WSNs. Managers will only inform what is expected, not how it should be obtained. Agents will be intelligent to decide how, when and what to do. Automatic services and functions can be executed towards self-management if there are appropriate conditions such as residual energy level.

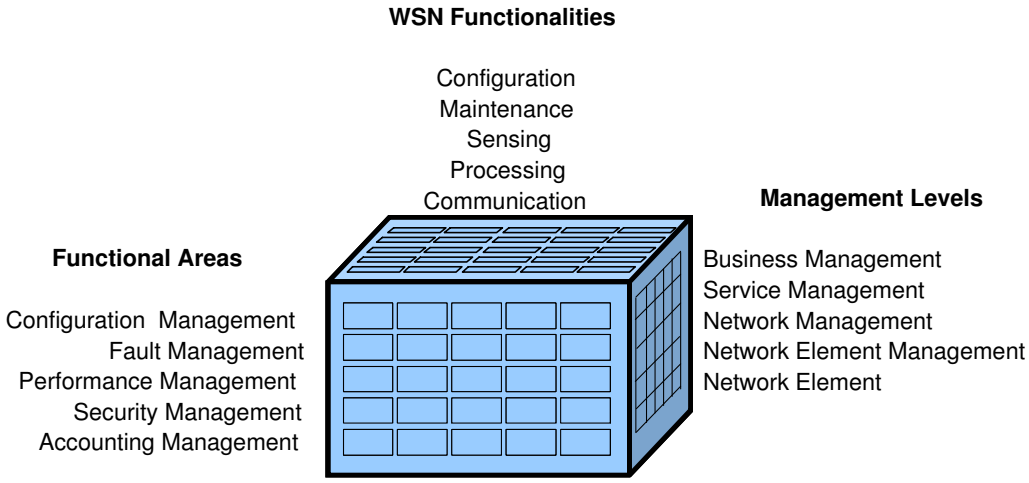


Figure 3.1: Management dimensions for WSNs.

The three management dimensions must be considered in the definition of a management function, in the establishment of an information model, in service composition, and in the development of a management application. For instance, management functions must be in one or more cells of the cube. For example, the intersection of the planes of performance management, network management and sensing must contain all the functions referring QoS sensing in the network level. A function may have several purposes

and so it may be found in one or more cells of the cube. There are several different views of management information which may be defined for management purposes. The three management dimensions can be used to define the levels of abstraction to obtain management information. For example, the intersection of network management level, configuration management, and communication is concerned with the information that is required to manage how network element entities (both physically and logically) are related and configured to provide and maintain area-to-area connectivity.

In the next sections, the WSN management is introduced from the perspective of management level, WSN functionalities, and management functional areas. As mentioned before, there are several significant differences between the management of traditional networks and WSNs and until now, no work has addressed these differences. In this sense, the following sections are contributions to the field, since such discussion was not found in literature.

3.3.1 Management Levels

Many traditional management systems use the LLA model in a bottom-up approach. However, in WSN management, the LLA model is used in a top-down approach. After analyzing the business level issues, the necessities of the lower levels become clear (see Figure 3.2). Similarly, it is only after defining the application, including the corresponding requirements on the service layer, that it is possible to plan the network, the network element management layers, and the network elements. This is a key observation when one reasons about WSN management. Now, we introduce a discussion concerning WSN management from the perspective of management levels.

3.3.1.1 Business Management

Requirements that allow the characterization of a sensor network come from the objectives defined in the business management layer. As WSNs depend on applications, business management deals with service development and the determination of cost functions. It

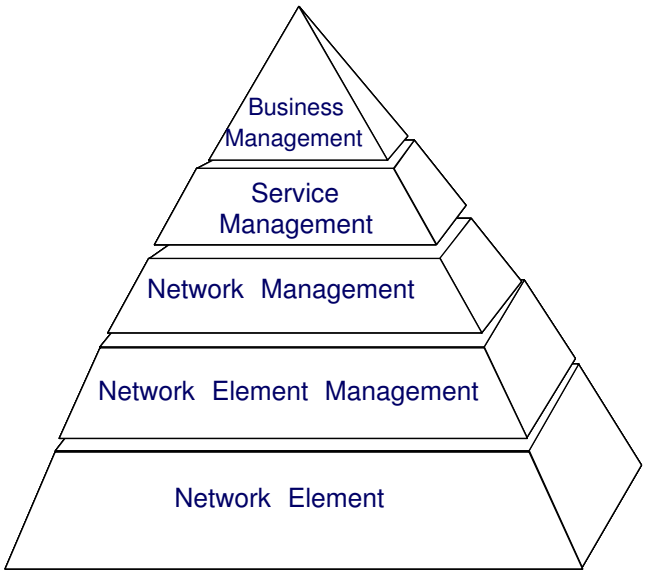


Figure 3.2: Management levels.

represents a sensor network as a cost function associated with network set up, sensing, processing, communication, and maintenance. The WSN applications have enormous potential benefits for society as a whole and represent new business opportunities. Instrumenting environments (as discussed in Section 2.4) with numerous networked sensor nodes can enable long-term data collection at scales and resolutions that are difficult, if not impossible, to obtain otherwise. In the future, we can expect to have the Internet end-points equipped with a variety of sensors to monitor both the network and its own state, and fairly sophisticated computing capabilities to allow them to function as decision elements and not just as repeaters. As more and more aspects of society are connected to networks, their sensory components become more and more prominent.

3.3.1.2 Service Management

A WSN is used to monitor and, sometimes, to control an environment. The WSN service management[2] introduces new challenges due to scarce network resources, dynamic topology,

[2]The content of this section was also published as a part of a paper in the Latin American Network Operations and Management Symposium, September 2003 [84] and as a tutorial [85].

traffic randomness, energy restrictions, and the large amount of network elements. WSN services are concerned with functionalities (see Figure 3.1) associated with the application objectives. Basic WSN services are sensing, processing, and data dissemination [79]. These services are application specific, that is, different protocol profiles for a certain service may be specified for different applications. A service, as a function of the network and the supporting applications, is characterized by a set of parameters which determine the service level. Bandwidth, data rate, throughput, response time, delay, message loss rate are examples of qualifiers of dissemination service. There are two main issues associated with WSN service management: Quality–of–Service (QoS) and Denial–of–Service (DoS).

Quality of Service. QoS architectures can only be effective and provide guaranteed services if QoS elements can be adequately configured and monitored. Mechanisms can be defined to help management applications to deal with QoS elements. Besides, such mechanisms must allow the replacement of the current device-oriented management approach with a network-oriented or cluster-oriented approach. Thus, in addition to the elements management (physical and logical resources), management applications must also be in charge of QoS aspects. This thesis introduces the components involved in QoS support for WSNs: QoS models, QoS Sensing, QoS Processing, and QoS Dissemination. This components are defined below. The larger the number of monitored QoS parameters, the larger the energy consumption and the lower the network lifetime. The QoS components introduced in this thesis are presented in the next paragraphs.

QoS Model. A QoS model specifies an architecture in which some services can be provided in WSNs. All other QoS components, such as QoS Sensing, QoS Processing, and QoS Dissemination (e.g. signaling, QoS routing, and QoS MAC) must cooperate to achieve this goal. A management application can establish the QoS model and control the QoS signaling to coordinate the behavior of other components. QoS-related tasks must be performed by using network management functions.

QoS Sensing. QoS sensing considers the sensor device calibration, environment interference monitoring, and exposure (time, distance, and angle between a sensor device and a phenomenon). As discussed in Section 2.5, the coverage area is defined as a measure of

QoS for a WSN. In the worst-case coverage, attempts are made to quantify the quality of service by finding areas of low observability to sensor nodes and detecting breach regions. In the best-case coverage, the management application has to find areas of high observability to sensors and identify the highest accuracy. A denser network will lead to a more effective sensing because of the higher accuracy of the network (e.g. areas of overlapping, and redundant information) and better fault tolerance. On the other hand, this will lead to a larger number of collisions and potentially to congestions, increasing latency and reducing energy efficiency [103]. Congestion control must not only be based on the capacity of the network, but also on the accuracy level required by the observer. The traffic in a WSN is different from conventional networks, in that it is a collective communication operation with redundancy. Consequently, the management application has the flexibility of reaching the performance demands by controlling the reporting rate of sensors, controlling the virtual topology of the network (by scheduling some sensors [103]), or optimizing the collective reduction in the communication operation (by data aggregation). The provision of QoS can rely on resource reservation. When an active node goes out of service due to operational problems, the management application activates a redundant node defined by any type of resource reservation scheme. In case of a low density of sensors, the network coverage area can be committed, thus affecting the quality of the service. This thesis applies a resource reservation scheme.

QoS Dissemination. Reliable data delivery is still an open issue in the context of WSNs. QoS Dissemination in WSNs is a challenging task due to constraints, mainly energy and the dynamic topology of WSNs. This thesis proposes two components for QoS dissemination: QoS routing and QoS MAC. QoS routing finds a path which satisfies a given QoS requirement, and QoS MAC solves the problem of medium contention that supports reliable unicast communication [109]. In order to support QoS, link state information such as delay, bandwidth, cost, loss rate, and error rate in network may be available and manageable. One of the objectives of the management application is to get and control link state information in WSNs for monitoring QoS. This is very difficult due to the fact that the quality of a wireless link can change with the circumstances, such as residual energy,

node distribution, density (all these change along the network lifetime), and interference. Configuration characteristics such as coverage area, density, network organization, node deployment (distribution), latency, and communication range may degrade or deny the service.

QoS Processing. The processing quality depends on the robustness and complexity of the algorithms being used, and the processor and memory capacities. The computing paradigm changes from one based on computational power to one driven by data. The way we measure processing performance changes from processor speed to the immediacy and accuracy of the response and energy consumption. Individual computers become less important than lower granularity and dispersed computing attributes. In many applications, the objective is not simply to perform a small number of high quality sensing operations, but rather to complete a large number of computations over longer timescales. In this context, correlating data from the maximum number of sensors is a good alternative. Each sensor node may be able to process local data using a correlation algorithm (data fusion, selective suppression, compression, clustering, filtering, counting, so on) to correlate the collected data, transforming it to information. If collaborative signal and processing algorithms run at each sensor node, the processing can occur at different layers in the protocol stack for such a cluster-based system. The sensor layer is responsible for collaborative signal processing. This process can include data correlation or beamforming, as well as the parameters distributed detection/estimation. This information is placed in a very small data packet that is sent to all other nodes in the cluster. Upon receiving of the packet, the other nodes update their tentative decisions. These decisions may be then broadcasted to all nodes in the cluster. The number of iterations depends on the distributed algorithm being used, and the possible achievement of convergence. The metrics quantifying the signal processing performance can be the probability of detection, false alarm, and error, the number of iterations, channel accesses, and bits transmitted, and total amount of energy expended [23]. Thread-task level metrics include average power expended in a given time period to complete a thread (task), power expended in transmitting control messages and information packets, and task completion time. Diagnostic metrics, which characterize

network behavior at packet level, include end-to-end throughput (average successful transmission rate) and delay, average link utilization, and packet loss rate. To achieve optimal performance in a WSN, it is important to consider the interactions among the algorithms operating at different layers of the protocol stack.

The network quality of service can be measured by the energy consumed to perform a service with a specific quality level. In most WSNs, energy consumption is one of the main metrics. However, there are situations in which, when certain events occur, the network must apply the maximum possible of the available energy possible in information delivery. As an example of this situation we can mention WSNs deployed over havoc of a cave where someone wants to get as much information as it can in as little time as possible. In this kind of application, the extension of the network lifetime is not the goal. However, without proper management mechanisms, the network can suffer from the implosion problem (congestions, collisions, and data losses in the network).

Denial of Service. Any situation that diminishes or eliminates the capacity of the network to perform its expected job is called DoS (Denial of Service). A network is subject to different types of threats: internal or external, and malicious or accidental. An attack occurs when malicious threats succeed. Some examples of incidental threats are hardware failures, software bugs, resource exhaustion, and unexpected environmental conditions. DoS aspects will be discussed in Section 3.3.3.4.

3.3.1.3 Network Management

This layer aims to manage a network as a whole, which is typically distributed over an extensive geographical area. In the network management level, relationships among sensor nodes are to be considered. It is known that individual nodes are designed to sense, process and communicate data, contributing to a common objective. In this way, nodes can be involved in collaboration, connectivity, and aggregation relationships. Building self-organizing WSNs is difficult because of the following main reasons: many different types of sensors with a range of capabilities must be developed with different application

requirements, the use of data-centric network protocols (such as directed diffusion [42]), the network must be extensible to new types of sensor nodes and services; the network must react rapidly to changes in the topology, task, degradation, and mobility.

A WSN is composed of interconnected managed objects (physical or logical ones), capable of exchanging information. In these cases, the network is basically composed of two parts: physical-logical resources and services. The service performance depends on the physical resource capabilities.

3.3.1.4 Network Element Management

In this layer, the functions referring to the management of individual network elements or network element groups (clusters) are defined. Managed network elements represent the sensor and actuator nodes or other WSN entities, which perform management functions, providing sensing, processing, and disseminating services. The basic functions of a WSN management network element are power management (how a sensor node uses its power), mobility management (plans, runs and registers the movement of sensor nodes), state management (how a sensor node manages the three management states defined for a node: operational, administrative, and usage), and task management (how a sensor node balances and schedules the sensing, processing, and disseminating tasks given a specific network state). Each sensor node must be autonomous and capable of organizing itself in the overall community of sensor nodes to perform coordinated activities with global objectives. When placed in an environment, the sensor nodes should immediately recognize their own capabilities and functions (self-test) and those of other sensor nodes, and work together as a community system to perform cooperative tasks and networking functionalities. WSNs need to be self-organizing and some sensor nodes may provide networking service system service, and resources to other sensor nodes. Others may detect the presence of these nodes and request services from them. The characteristics of sensor nodes necessary for creating self-organizing sensor networks are agility, self-awareness, self-configurability and autonomy. Sensor nodes with these features will have capabilities for self-assembling im-

promptu network degradation, mobility of sensor nodes, and changes in task and network requirements. Nodes are aware of their own capabilities and those of other nodes around them which may provide the services that they need.

Considering that some applications may require networks with a large amount of sensor nodes, a network element can deal with a single node component or a group of nodes. In such case, a manageable element can be a cluster of nodes or a cluster-head node, rather than an individual node. The design of a sensor node is motivated by the need to create an inexpensive device with a small form factor and low power dissipation.

3.3.1.5 Network Element

The network element represents the physical and logical components of a managed element. Physical resources include sensor or actuator nodes which include power supply, processor, memory, sensor device, and transceiver. Logical resources include communication protocols, application programs, correlation procedures, and network services. The main physical restriction of a WSN is the available energy, since in general batteries are not recharged during the operation of a sensor node. All activities performed by the node must take into account the energy consumption. Energy consumption patterns of individual nodes and the entire network must be characterized and profiled. This process yields a better understanding of where to apply trade-offs in the design of the management solution. The most widely used power supply in a WSN is the battery. Understanding a node capability allows the function management to be more efficiently structured and fine-tuned. Aiming to show more details about sensor nodes architectures, this thesis offers an overview of them in Section 2.2.

3.3.2 WSN Functionalities

This section introduces the novel proposed dimension for WSNs management, which consists of configuration, sensing, processing, communication, and maintenance functionalities. This novel dimension is obtained from the functional model defined in [83], which presents

a scheme to characterize WSNs considering that they are application-dependent. These WSN functionalities can be observed in the upper part of Figure 3.1. A management solution depends on the features of the network. Thus, this solution must also be proposed considering the type of the network. For this reason, WSN functionalities are serviceable in the development of the management application.

3.3.2.1 Configuration

This functionality involves procedures related to the planning, placement, boot up and self-organization of a WSN. The configuration functionality (pre-deployment) is related to the definition of WSN application requirements, the determination of the monitoring area (shape and dimension), the environment characteristics, the choice of nodes, the definition of the WSN type, and the service provision. In the deployment phase, sensor nodes can be placed on a certain area, one by one, by being dropped from a plane, rocket, or missile, or by a human or robot. The placement approach must also take into account the high cost and the difficulty in re-deploying nodes. This is chiefly due to the limited life span of nodes, and to the fact that their power sources are, in general, non-replaceable [60]. Another problem is the optimal location of the access point (sink node or base station). An inefficient configuration management may adversely affect the overall performance.

WSNs are application-specific, which causes that the configuration functionality to change from one WSN to another. Here we discuss the configuration considering the possible types of WSNs and the other two management dimensions.

Considering the *network management level* and the management functional areas based on the configuration functionality, WSNs can be classified as described in the following paragraphs. A WSN is said to be *homogeneous* when all nodes have the same hardware capabilities (processor, memory, battery, and communication device features). When the WSN is comprised of nodes with different capabilities it is said to be *heterogeneous*.

A WSN is said to be *hierarchical* when the nodes are organized in groups (see Figure 2.4). The groups can be organized in different hierarchies. Each group has its own

leader and belongs to an hierarchical level. When the nodes are organized in only one level, the hierarchy of the groups is one. In heterogeneous hierarchical networks, the nodes with the highest capabilities can assume the leadership during all the network lifetime. If the network is homogeneous hierarchical, it can have a leader election process. New elections should occur motivated by different parameters, such as minimal level of residual energy of the current leader. Is is also possible to establish some criteria to determine which nodes are able to vote and which ones can be elected. A possible alternative to the leader election process is an indication process. In this case, the first leader of each group is indicated by the management entity which has a global view of the network. As each leader has a local view of its group, it indicates its successor. A WSN is said to be *flat* when its nodes are not organized in groups.

In homogeneous hierarchical networks, when the leaders receive the information from the common-nodes, they can perform some processing (e.g., fusion, aggregation, counting, selective suppression, and others) and disseminate the resulting information using multi-hop communication to the base station. In some cases, if the hardware allows, the leaders radio range can be adjusted (increased) to transmit data in a single-hop to the BS, thus, increasing energy consumption. A WSN is *static* when nodes are stationary, and *mobile* otherwise. Note that the topology may change dynamically even when nodes are stationary since new nodes can be added to the network or existing nodes can become unavailable. A WSN is *symmetric* concerning signal transmission when each transceiver has the same transmission range, and *asymmetric* otherwise. A WSN is said to be *regular* concerning nodes placement when its nodes are placed in a grid. It is classified as *irregular* when its nodes are randomly distributed and present different densities in the monitored area. A WSN is *balanced* when its nodes present a uniform distribution in the monitored area. A WSN can also be *sparse* or *dense* depending on the number of nodes per area unit. The number of network elements permit the characterization of a WSN as *large* (composed of a thousand network elements or more), *medium* (composed of a hundred to thousand network elements), and *small* (composed of up to a hundred network elements).

Considering the *network element management level* and the management functional

areas based on the configuration functionality, the sensor nodes in a WSN are spread over a region and communicate among themselves using point-to-point wireless communication, forming an ad hoc network. The nodes are autonomous when they are able to execute location discovery and self-configuration tasks without human intervention. To send out information, sensor nodes are equipped with a wireless communication device, *transceiver*.

A *wireless sensor node* is also comprised of *sensor unit*, *battery*, *memory*, and *processor*. The size of a node is an important consideration. Nodes need to have small dimensions factors so that they may be located unobtrusively in the environment targeted for monitoring. The restriction in size is closed related to the amount of energy available in the node. A rugged and robust construction is required if nodes are being dispersed in an inhospitable terrain such as a forest. *Software* developed to run in a wireless sensor node must take into account its hardware restrictions. Due to the limited energy capacity, nodes are expected to be thrown away once their energy supply is exhausted. The system can have levels of redundancy built into it so that it can allow failures or to increase accuracy. This can be achieved by using more sensor nodes than what is strictly necessary to cover an area. Additionally, due to environment nature, logistics, and deploying costs, the deployment of sensors can be an one-time operation. Therefore, after nodes have been distributed in the field, human intervention is not an option.

Basically there are three different types of sensor nodes: *common-nodes* responsible for collecting sensing data, *sink nodes* (monitoring nodes), responsible for receiving, storing, and processing data from common-nodes, and *cluster-heads*, which are the leaders of group responsible for receiving, storing, and processing data cluster nodes. WSNs can also include *actuators* that enable control or actuation on a monitored area. In a hierarchical network, it is common to have a *Base Station* (BS) (see Section 2.1).

Considering the *service management level* and the management functional areas, a WSN comprises three entities: *observer*, *phenomenon*, and *environment*. The observer is a network entity or a final user that wants to have information about data collected, processed, and disseminated by sensor nodes. Depending on the type of application, the observer may send a query to the WSN, and receive a response from it. These queries can

be sent with or without *fidelity*. Fidelity is a property which permits to select only a subset of nodes sending data according to determined resolution. The translation of the query can be performed by the application software or sensor nodes. The WSN may participate in synthesizing the query (e.g., filtering some sensor data or summarizing several measurements into one value), but these procedures are related to the processing functionality. The phenomenon is the entity of interest to the observer that is being sensed and can be optionally analyzed or filtered by the WSN. The observer is interested in monitoring a phenomena under some latency and accuracy restrictions. A sensor element generates data about one or more phenomenon such as temperature, pressure, electromagnetic field, or chemical agents.

3.3.2.2 Sensing

The lowest level of the sensing application is provided by the autonomous sensor nodes. An important operation in a sensor network is data gathering. Sensing functionality depends on the type of the phenomenon. Thus, WSNs can be classified in terms of the data gathering required by the application as *continuous*, when sensor nodes collect data continuously along the time, and *reactive* when it answers to an observer's query or gather data corresponding to specific events occurring in the environment and *periodic* when nodes collect data according to conditions defined by the application. Some approaches can coexist in the same network; we refer to this model as the hybrid collect model. An example of a continuous phenomenon is temperature, and an example of an application where the phenomenon itself is moving is animal detection. Other examples of phenomena are video, audio, pressure, mechanical stress, humidity, soil composition, luminosity, seismic activity, and chemical elements. Regardless of the gathering being continuous or not, WSNs are defined based on how the data will be transmitted to the observer. The sensing encloses the exposure (time, distance and angle of phenomenon exhibition at the sensor), calibration and sensing coverage. Depending on the density of the phenomenon, all sensor nodes being active all the time may be inefficient. A model which is well-suited to this case is the

Frisbee model [15]. In this model, a set of nodes are maintained actives in a defined zone. The zone might be circular, with its radius proportional to the speed of the phenomenon that is being tracked by the active nodes. The nodes out of the zone are maintained inactive saving energy and minimizing congestion and collision. However, redundancy (overlap in the sensor coverage) should be utilized in such a way that fault tolerance in the communication network is available, and better accuracy can be found [103]. Nevertheless, the sensors can be mobile. In this case, the sensors are moving with respect to each other and to the observer as well, and they have direction, orientation and acceleration.

3.3.2.3 Processing

The sensor node memory and processor form the computational module. The computational module is a programmable unit that provides computation and storage for other nodes in the system. Depending on the system communication constraints, algorithms must be developed to allow individual nodes or clusters of nodes to share and process data efficiently. The computational module performs basic signal processing (e.g., simple translations based on calibrating data or threshold filters), and dispatches data according to the application. The processing can also involve correlation procedures such as data fusion. The correlation consists of the conceptual interpretation of multiple data, leading to the attribution of a new meaning to the original data. It generally has the goal of reducing the number of data transferred to the manager of the network management system. In the sensor network management, the correlation may be applied to any of the five management functional areas and may be done at several levels of the configuration, from the individual network elements to the maximum level, which involves all the network. When the correlation takes place at a lower level, it is generally made up of simpler and, consequently, faster processes. Due to the large volume of information involved, and the energy restraint, the correlation is useful in sensor network applications. Several types of correlations may be identified [61], according to the operations performed on the data. The most used is data fusion and aggregation but there are some other important operations, which are detailed

as follows:

- Aggregations are operations which summarize current data values in some or all sensor nodes of a WSN. Computing aggregates in-network are attractive from a network performance and longevity standpoint: extracting all data over all time from all sensors consumes large amount of time and power as each individual sensor's data is independently routed through the network. Previous results have shown [33] that aggregation or data fusion (described below) dramatically reduces the amount of data routed through the network, increasing throughput and extending the lifespan of battery-powered sensor networks as less load is placed on power-hungry radios. Aggregation is essential for wireless sensor networks where energy resources are limited.

- Data fusion combines one or more data packets received from different sensors to produce a single packet (data fusion). It helps reducing the amount of data transmitted between the sensor nodes and the observer, allowing the design of a network which delivers the required data while meeting energy requirements. Other possible tasks are security processing and data compression.

- Compression consists of detecting, from the observation of the data received in a given time-window, multiple occurrences of the same data, substituting the corresponding data for a single data, possibly indicating how many times the event occurred during the observation period.

- Selective suppression is a temporary inhibition of management data referring to a given event, according to certain criteria which is continuously evaluated by the correlation system related to the dynamic context of the network management process. The suppression criterion is generally linked to the information obtained through the network models (see Section 4.4).

- Filtering consists of suppressing a given management event, depending on the values of a set of parameters previously obtained through the network models.

- Counting consists of generating new management data each time the number of occurrences of a given type of event surpasses a previously established threshold.

- Scaling is an operation in which, depending on the operational context obtained through the network models, a management event is replaced with another one with one of its parameters (for example, *severity*) assuming a higher value.

- Temporal relationship is an operation in which the criteria for correlation depend on the order or the time at which data is generated or received. Several temporal relationships may be defined, utilizing concepts such as: AFTER, FOLLOW, BEFORE, PRECEDE, DURING, START, FINISH, COINCIDE, OVERLAP.

- Spatial relationship is an operation in which the criteria for correlation depend on the location at which data is generated or received.

3.3.2.4 Communication

Individual nodes communicate and coordinate among themselves. We propose two types of communication: *infrastructure* and *application*. Infrastructure communication refers to the communication used to configure, maintain, and optimize operation. The configuration and topology of the sensor network may rapidly change in a hostile environment, a large workload, and nodes that fail routinely. Conventional protocols may be inadequate to manage such situations and, thus, new protocols are required to promote WSN productivity. In a static sensor network, an initial phase for the infrastructure communication is needed to boot up the network. Moreover, additional communication is needed to perform reconfiguration. If the sensors are mobile, additional communication is needed for path discovery/reconfiguration.

Application communication (dissemination) relates to the transfer of sensed data or information obtained from it. The amount of energy spent in transmitting a packet has a fixed cost related to the hardware and a variable cost that depends on the distance of transmission. Receiving a data packet also has a fixed energy cost. Therefore, in order to

save energy, short distance transmissions are preferred. Since the access point (sink node or the BS) may be located far away, the cost to transmit data from a given node to the access point may be high. In a homogeneous flat WSN, sensor nodes can form a multi-hop network by forwarding each other's messages, which can provide different connectivity options. In a heterogeneous hierarchical WSN, the cluster-heads can form a single-hop network for reporting aggregated data to the BS. Within a cluster, measured data is sent to the cluster-head by the sensor nodes, which are under its control. All nodes in a cluster are identical, except those in the heterogeneous WSN, where the cluster-head has a larger transmission capacity.

WSNs can be classified in terms of the application data delivery requirements as *continuous*, when sensor nodes send their data to an observer continuously along the time, and *on-demand* when it answers to an observer's query. A WSN is *event-driven* when sensor nodes send data referred to specific events occurring in the environment and *programmed* when nodes send data according to conditions defined by the application. Some approaches can coexist in the same network; we refer to this model as the hybrid model. The cost of sending data continuously may lead to a faster consumption of the scarce network resources consequently shortening the network lifetime. Multi-hop wireless capabilities will enable communication and coordination among autonomous nodes in unplanned environments and configurations. At the same time, wireless channels present challenges in dynamic operating conditions, power constraints for autonomously-powered nodes, and complicated interactions between high level behavior and low level channel characteristics (e.g., increased synchronized communication will significantly degrade channel characteristics).

For any of the aforementioned models, we can classify the communication approach as: *flooding* (sensors broadcast their information to their neighbors, which in turn broadcast this data until it reaches the observer), *gossiping* (sensors send data to one randomly selected neighbor), *bargaining* (sensors send data to sensor nodes only if they are interested), *unicast* (sensor can communicate to the sink node, cluster-head or BS directly), or *multicast* (sensors form application-directed groups and use multicast to communicate among group members). A major advantage of flooding or broadcast is the lack of a complex

network layer protocol for routing, addressing and performing location management.

In a WSN, each sensor node puts its information into a common medium. This requires careful attention to protocols design. In *master-slave* protocols, one node gives the commands and another node or a collection of nodes carry them out. The cluster-head is usually the master and the common-nodes (sensors and actuators) are the slaves. This protocol allows tight traffic control because no node is allowed to transmit unless requested by the master, and no communication is allowed between slaves except through the master (e.g., medium control access protocol using a channel-fixed allocation scheme). If the master outs of service, its slaves will also out of service. In a *peer-to-peer* network, all nodes are created equal. A node can be a master one moment and then be reconfigured at another time. Peer-to-peer configurations offer the greatest flexibility, but they are the most difficult to control. Any node can communicate directly to any other node.

3.3.2.5 Maintenance

The maintenance functionality which is used in WSNs can configure, protect, optimize and heal themselves without intervention of human operators. Maintenance detects failures or performance degradations, initiates diagnostic procedures and carries out corrective actions in the network. Its ability to discover changes in the network state enables the self-management to adapt and optimize the network behavior. Beyond corrective maintenance, there are other types of maintenance: *adaptive* (the system should adapt itself to meet the changes), *preventive* (the system should learn to anticipate the impact of those changes), and *proactive* (as the system gets smarter, it should learn to intervene so as to preempt negative events). The preventive maintenance does not, however, prevent the network from reacting to unpredictable changes in the environment. An example of maintenance is controlling the density of nodes in a WSN. In case of high node density, the maintenance functionality can turn off temporally some nodes.

A WSN state (for instance the topology, energy level, coverage area) of the network changes frequently. In the case of static networks, changes occur because nodes may be-

come unavailable during operation. This dynamic behavior must be observed. The maintenance depends on the knowledge of the network state. Thus, maintenance functionality is needed to keep the network operational and functional, to ensure robust operation in dynamic environments, as well as to optimize the overall performance. The maintenance provides dependability whose main attributes are reliability, availability, safety, security, testability and performability. WSNs have important characteristics depending on the application. Some of them are planning, deployment, coverage, accuracy, fidelity, density, self-organization, adaptation and location. The points described in this section play an important role in the definition of the management services and functions.

3.3.3 Management Functional Areas

The WSN management considers that the fault, security, performance and accounting management functional areas are extremely dependent on the configuration functional area. In WSNs, all operational, administrative and maintenance characteristics of the network elements, network, services, business, and the adequacy performed in the activities of configuration, sensing, processing, communication, and maintenance (as shown in Figure 3.1) are dependent on the configuration phase of the WSN. An error in the configuration or a forgotten requisite during the planning may compromise all the functionalities of all other areas. This concept is depicted in Figure 3.3 where the configuration functional area plays a central role. As mentioned before, there are several significant differences in the management of traditional networks and WSNs. In this sense, management functional areas must be rethought considering the WSNs features.

3.3.3.1 Configuration Management

Configuration management is a functional area of high relevance in WSN management. As the objective of a sensor network is to monitor (acquisition, processing, and delivery of data) and, sometimes to control an environment, any problem or situation not anticipated in the configuration phase can affect the service provided. The configuration management

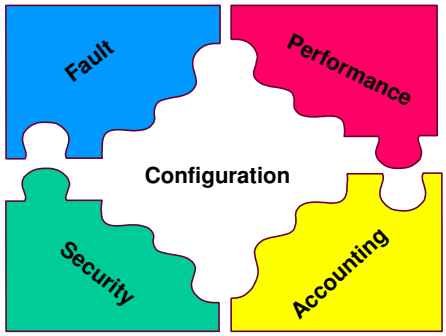


Figure 3.3: The role of configuration management.

must provide basic features such as self-organization, self-configuration, self-discovery, and self-optimization. Some management functions which we have defined for network level configuration management are: requirements specification of the network operational environment; environmental variations monitoring; size and shape definition of the region to be monitored; node deployment – random or deterministic; operational network parameters determination; network state discovery; topology discovery; network connectivity discovery; node density controlling; synchronization; network energy map evaluation; coverage area determination; and integration with an observer. Some management functions that we have defined for network element level configuration management are: node programming; node self-test; node location; node operational state; node administrative state; node usage state; node energy level; and so on.

3.3.3.2 Fault Management

Faults in WSNs are not an exception and tend to occur frequently, thus fault management is a critical function. This is one of the reasons that make WSN management different from traditional network management. Faults happen all the time due to energy shortage, connectivity interruption, environmental variations, and so on. In general, sensor networks must be fault-tolerant, robust and must survive despite faults in individual nodes, in the network or even in services provided. In addition to events caused by energy problems,

other events can happen in a wireless sensor network related to communication, quality of service, data processing, physical equipment fault, environment, integrity , operational and time-domain violation, and security. Therefore, even if a node has an adequate energy level to perform its function, it may decide not to do that because of other reasons.

Fault management must provide basic functionalities such as self-maintenance, self-healing, and self-protection. Mechanisms for recovering from node failures must be thought in terms of the network, so that it be self-healing. Several characteristics of sensor networks make us believe that faults, common in traditional computer networks, will be even more common in this kind of network. First, large-scale deployment of cheap individual nodes means that node failures from manufacturing defects will not be rare. Second, attacks by adversaries will be likely because these networks will often be embedded in critical applications, and deployed in open spaces or enemy territories, where adversaries can not only manipulate the environment (so as to disrupt communication by jamming it), but also have physical access to the nodes. Finally, ad hoc wireless communication by radio frequencies means that adversaries can easily put themselves in the network and disrupt infrastructure functions (such as routing) which are taken by the individual nodes themselves.

In the majority of applications, fault detection is vital not only for fault tolerance, but also for security. If, in addition to detecting a failure, we can also determine (or gather indicatives) that it has malicious origins, then we can alert the observer to an attack, for instance setting off an alarm.

3.3.3.3 Performance Management

The challenge in performance management is to perform this task without adversely consuming network resources. There is a trade-off to be considered: the higher the number of managed parameters, the higher the energy consumption and the lower the network lifetime. On the other hand, if enough parameter values are not obtained, it may not be possible to manage the network appropriately.

The configuration (in terms of sensor capabilities, number of sensors, density, nodes distribution, self-organization, and data dissemination) plays a significant role in determining the network performance. Performance management must consider the self-service characteristic. The performance of the network and the provided service are best measured with parameters such as the accuracy and delay requirements of the observer, and consumed energy. The accuracy indicates the reliability or exactness of a result. It can also be defined as the fraction of valid results from all the results obtained. The accuracy of a measurement at a network element (sensor) is specific to the physical transducer and to the nature of the phenomenon. The accuracy at the network level depends on the delay in data delivery due to network congestion, route length, duty cycle of the sensors, or data aggregation processing. As for the accuracy at the service level it depends on the metric chosen by the application for establishing the coverage area and the amount of energy to be spent in gathering and disseminating data. At the observer, it is likely that multiple samples may be received from different sensor nodes with different data quality.

Thus, additional performance metrics include coverage area, exposure, goodput (the ratio of the total number of packets received by the observer to the total number of packets sent by all sensors over a period of time [101]), cost of sensors, scalability, and produced data quality. In some applications, besides the information about some features of the phenomenon, it might be necessary to know where (sensor location), when (data–time) and how (sensor calibration and exposure) data were generated to manage the WSN performance.

Regardless of the application, there are certain critical features that can determine the efficiency and effectiveness of a sensor network [100]. These features can be categorized into quantitative and qualitative features. Quantitative features include network settle time, network join time, network depart time, network recovery time, frequency of updates (overhead), memory requirements, and network scalability. Qualitative critical features, on the other hand, include knowledge of nodal location, topology changes effects, adaptation to radio communication environment, power consciousness, single or multichannel, and network security preservation.

3.3.3.4 Security Management

Security functionalities for WSNs are intrinsically difficult to be provided because of their ad hoc organization, intermittent connectivity, wireless communication and resource limitations. A WSN is subject to different safety threats: internal, external, accidental, and malicious. As a result, information or resource can be destroyed, information can be modified, stolen, removed, lost, or disclosed, and service can be interrupted. Even if the WSN is secure, the environment can turn it vulnerable. Security management must provide self-protection (confidentiality, integrity, reliability, disposability, privacy, authenticity, and integrity). Determining if a fault or a collection of faults is the result of an intentional DoS attack presents a concern of its own, a point that becomes even more difficult in large-scale deployments, which may have higher nominal failure rates of individual nodes than in small networks. The robustness against physical challenges may prevent some classes of DoS attacks. Each layer of the protocol stack is vulnerable to different DoS attacks and has different options available for its defense.

3.3.3.5 Accounting Management

Accounting management includes functions related to the use of resources and corresponding reports. It establishes metrics, quotas and limits that can be used by functions of other functional areas. These functions can trace the behavior of the network, and even make inferences about the behavior of a given node. Furthermore, accounting management must consider self-sustaining. In a WSN, there is an energy producer (the battery) and some energy consumers (the transceiver, computation module, and sensor devices). Operations of the application or management can be measured or counted in terms of energy consumption. Given the node characteristics, the average sensor lifetime determines the cost of running a sensor network. One way of having a reduction in total energy consumption is to cut down high-energy operations at the cost of an increase in the number of low-energy operations. The measured cost can be amortized using prediction models [31]. Some functions related to accounting management are: discovery, counting, storing, and

parameter data reporting; network inventory; communication costs determination; energy consumption; and traffic checking.

3.4 Conclusion

This chapter has proposed the use of the paradigm known as self-management as the basis for the management framework proposed by this thesis. In Section 3.1 the characteristics of self-management were discussed for WSNs. The WSN management must be autonomic, i.e, self-managed (self-organizing, self-healing, self-optimizing, self-protecting, self-sustaining, and self-diagnostic), with a minimum of human interference, and robust to changes in the network states while maintaining the quality of the services.

This chapter has also discussed the management challenges for WSNs and proposed the organization of the WSN management into three dimensions, one composed of management levels, one composed of management functional areas, and one composed of functionalities. The idea of functionality for WSNs management is a proposal of this thesis as well. All three dimensions are explored from a WSN perspective. The management functional areas and the management levels were rethought considering the particular characteristics of WSNs. Some management functions are given in each management functional area and in the following chapter, we will explain how to define these functions. The following chapter also presents the MANNA architecture and its functionalities.

Chapter 4

The MANNA Architecture

The MANNA architecture is presented in this chapter[1]. The MANNA architecture, which was proposed to provide a management solution for different WSN applications. It provides a separation between both sets of functionalities, i.e., application and management, making possible the integration of organizational, administrative, and maintenance activities for this kind of network. The approach used in the MANNA architecture works with each functional area, each management level, and proposes a novel management dimension called WSN functionalities (configuration, sensing, processing, communication, maintenance) as presented in last chapter. The principles and characteristics of the management architecture are presented in Section 4.1.

A new scheme to define management functions and a list of these management functions are presented in Section 4.2. An approach to develop management services is presented in Section 4.3. As a result, the MANNA architecture provides a list of management services and functions independent of the technology adopted which are presented in this chapter. The MANNA architecture establishes some automatic services, which feature self-managing (self-organizing, self-healing, self-optimizing, self-protecting, self-sustaining, self-diagnostic, and so on) with a minimum of human interference. The MANNA archi-

[1]The subject of this chapter was published in the IEEE Communications Magazine, February 2003 [84]. and will be published as chapter in the book entitled "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems" [82].

tecture intends to be robust adapting to changes in the network state and establishes some services to maintain the quality of the services provided. This chapter also presents the three sub-architectures of which the MANNA architecture is composed. Information architecture is described in Section 4.4. Functional and physical architectures are described in Section 4.5 and in Section 4.6 respectively. Section 4.7 presents a discussion about how to build management applications considering the management dimensions presented in Chapter 3. Finally, in Section 4.8 we consolidate all these concepts through an example.

4.1 The MANNA Architecture Overview

The design of a WSN management architecture must follow some principles. Within the scope of this thesis, WSN management must be simple, adherent to network idiosyncrasies (including its dynamic behavior) as well as efficient in the use of the scarce resources. If, on one hand, it is interesting to conceive open architectures, on the other hand, adhering to standards can be a complicating factor.

As seen in the last chapter, the MANNA architecture is based on the paradigm of self-management which allows the definition of autonomic WSNs. The MANNA architecture uses the three management dimensions (management functional areas, management levels, and WSN functionalities) to define management functions, build management services, establish an information model, and design management applications. This will make possible the integration of organizational, administrative, and maintenance activities for this kind of network. The adoption of a strategy based on information models and the traditional framework of management functional areas and management levels will make possible management integration in the future.

The MANNA architecture establishes a separation between applications and management, through the proposition of three architectures:

- Information architecture. It is proposed to ensure common solutions for management through the definition of an information model and a strategy to obtain management

information. An information model provides a foundation for understanding the characteristics of the network and promotes uniformity in dealing with the various aspects of resource management. There are several different viewpoints of management information which may be defined for management purpose. In the information architecture, these viewpoints are the three management dimensions. Orthogonality among the three dimensions should be maintained in the descriptions of the information models to avoid redundancy in it (see Section 4.4).

- Functional architecture. It is proposed to plan possible locations for management entities (managers and agents), and management services and functions they can perform and how the latter will be performed. The information and conditions to perform services and functions are obtained from WSN models defined in the information architecture. The management choice depends on the three management dimensions (see Section 4.5).

- Physical architecture. It is proposed to provide interfaces between management entities. It does not define a protocol stack for these interfaces, but provides protocol profiles that may be adequate for each application type (see Section 4.6).

One of the major goals of a management architecture is to promote network resources productivity and the quality of the service provided. A management solution depends on the feature of the network. There are WSNs in which only a few management services or functions can be implemented. In other cases, the self-management solution cannot be performed because of restrictions in the computation and resources. Depending on the WSN application, it may be interesting or not to use certain management functions which also can be implemented as automatic, semi-automatic, or manual. The MANNA architecture framework provides this flexibility. The management solution can be obtained from the composition of the management services and the definition of management policies which can be performed through a centralized, distributed, and hierarchical approaches.

The following sections present the management framework introduced by MANNA which includes information, functional and physical architectures; the use of the three

management dimensions in definition of management services and the construction of the management services, a list of management functions and services which can be executed automatically, semi automatically, and manually to provide self-management; an information model for WSNs which includes managed object classes (to represent static information) and WSN models (to represent network states); and the three architectures of which MANNA is composed of (information, functional and physical). The next sections present how the MANNA architecture establishes management functions and services and how it works in order to develop of management solutions.

4.2 Defining Management Functions

The management functions represent the lowest granularity of functional portions of a management service, as perceived by users. A scheme to design management functions consists in dealing with each management functional area and each management level considering the functional model of the network and establishing what are the management tasks found in the intersection of the three dimensions (see Figure 4.1). Thus, one or more management functions can fit into one or more cells of the cube (see Figure 3.1).

As result, a partial list of the management functions, in no particular order, is given in the following.

Environment requirements acquisition function: consists in obtaining requisites about the environmental conditions of the area to be monitored. The propagation of signals and the behavior of electronic components are susceptible to environmental conditions. The physical effects are difficult to predict and may lead to inaccurate measurements, thus affecting the quality of the service. Consequently, a requisite provision function of the environment is necessary to the network planning.

Monitored area definition function: consists in establishing the size and the form of the region to be monitored.

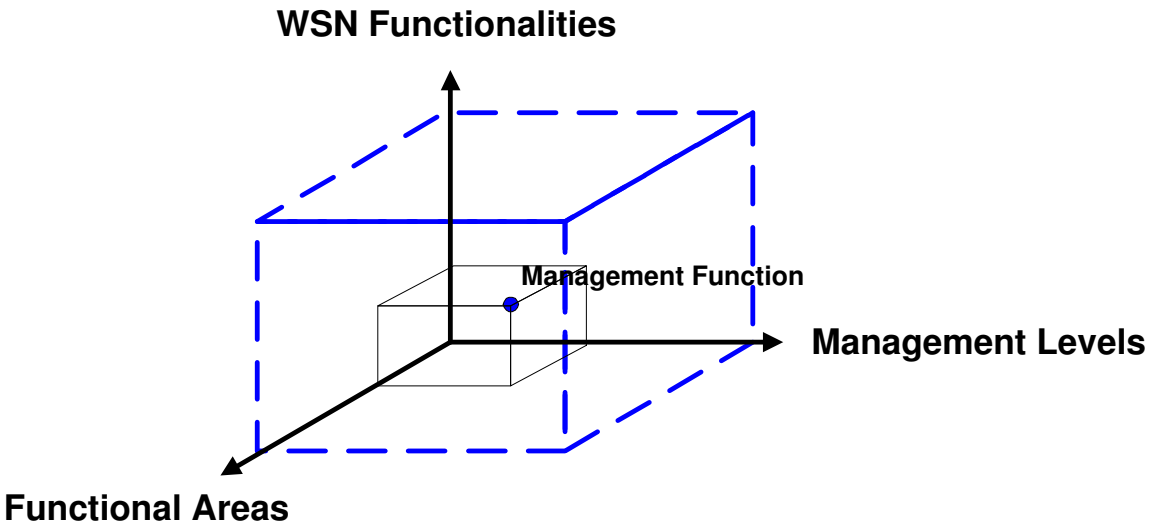


Figure 4.1: Intersection of the management dimensions.

Environment monitoring function: consists in supervising variations in the environment which extrapolate defined thresholds.

Node definition function: consists in defining the node architectures to be used in the network to perform the defined services . The accuracy of the sensing hardware or transducer will affect the accuracy of the sensing at the observer. The size of the memory affects the buffering space at the sensors and the ability of the network to handle transient bursts in traffic. The battery size determines the amount of energy available at the sensor node and affects the network lifetime. The capabilities of the embedded processor determine the level of optimization that is possible at the sensor nodes without introducing excessive power loss or intolerable delay levels. The characteristics of the transceiver determine the network transmission range and the transmission channel capacity. Improving the characteristics of any of these devices may increases cost and dimension factor of sensor nodes.

Number of nodes definition function: a higher sensor nodes density offers the potential for a better connected network with more efficient paths between the sensor

nodes and the observers. However, increasing the number of sensor nodes in turn results in a higher number of sensor nodes reporting their results per time unit. If this increased load exceeds the capacity of the network in terms of the number of nodes accessing the shared wireless medium and the number of intermediate nodes in the network, increasing the number of active sensor nodes may adversely affect the network performance.

Node deployment definition function: consists in determining the location and the way in which the nodes will be placed in the monitored region. Basically, there are three common deployment strategies: random deployment, in which the sensor nodes are "sprayed" with a uniform distribution within the field; regular deployment, in which the sensor nodes are placed with some regular geometric topology in the sensor node field for example, a grid; planned deployment, in which sensor nodes deployment is planned according to some criteria (for example, biased to provide higher sensor node density in areas where the phenomenon is concentrated).

Network operating parameters configuration function: consists in attributing values to the network parameters and to the nodes. Some parameters may be configured while others are characteristics of the actual network. As examples of parameters we can mention: number of nodes, types of nodes, node capacity, type of channel, type of propagation, presence of actuators, type of access control protocol, type of the mechanism for information dissemination, type of routing scheme, type of synchronization, number of active nodes, target speed, quality metrics, traffic parameters, network range, and so on.

Node deployment function: after the definition, the nodes can be deposited in a random or uniform-distributed environment.

Topology map generation function: consists in discovering the topology of the network. The topology describes the connection that may exist and expresses the relationships among the sets of nodes.

Network connectivity discovery function: consists in discovering the connectivity of the network. The connectivity represents the association between two network points at a given instant.

Correlation discovery function: consists in discovering the relationship between a compound object and its immediate components. This function may be used to determine the aggregational relations which exist in the network, for example the existing clusters.

Cooperation discovery function: consists in obtaining the cooperating relations. These relations may be created, activated and terminated (normally or abnormally) among the network components. The components involved may, by their own initiative or activated by foreign actors, adjust their behavior or share resources in order to reach common goals.

Synchronization function: consists in the execution of synchronizing functionalities that may be used in functions such as criptography, to coordination and plan future events, to order events stored in the log during the system debugging process, in order to remove redundant messages. Due to energy restrictions in sensor networks, the synchronization method should consume little energy, making it different from other conventional methods of distributed systems [74].

Energy map generation function: consists in obtaining the energy map of the network. The energy level in the nodes and in the network may be visualized considering the region or the time interval.

Nodes density calculation function: consists in discovering the quantity of nodes per monitored area.

Network coverage area definition function: consists in strategic planning for the establishment of the covered network area, considering the area type (internal or external), dimensions, environmental conditions, conditions of node disposition, and so

on.

User interface function: consists in executing of functions to interface with observers.

Node programming: consists in programming the sensor nodes to perform application tasks, power management, mobile management, and so on.

Self-test function: consists in running tests done by the nodes themselves.

Node localization function: consists in discovering the nodes location. This function allows the utilization of different methods of global or relative localization (see Section 2.5).

Node operating state control function: due to the different activities and the energy level in the nodes and in the network, the network nodes may present different operational states: normal, major, minor, critical, and inactive. The normal, major, minor, and critical states correspond to an active node. Thresholds are used to indicate state changes. An event is used to indicate when the energy level of a power supply reaches a certain threshold.

Node administrative state control function: there may be moments in which it is desirable for a node to change its administrative state for the interest of the application, for instance in the case of two sensor nodes presenting an intersection in the covered area. In this situation, the application may remove one of these nodes from service. Other actions may be blocking the node for collecting and activating it for communication or blocking it for all kinds of activity. Conditions for the node to return operating regularly may be defined, with the destruction of a neighboring node. The different administrative states are: locked when the network element is out of service (turned off or sleeping), unlocked when the network element is in service, and in-unlocked when the element is apt to operate but is waiting to integrate into the network.

Component	Usage State	Usage State	Usage State	Usage State	Usage State
Communication unit	transmitting	receiving	idle	inactive	sleep
Processing unit	active	waiting	idle	inactive	sleep
Sensing unit	active	waiting	idle	inactive	sleep

Table 4.1: Example of usage states.

Node usage state control function: consists in controlling which elements of the node are in use. Node usage state is a result of the combination of its components states. It is possible that there are some states for each node component. For example, a processor can be in an active, inactive, idle, and locked (sleep) mode. The transceiver can be transmitting, receiving, hearing, idle, locked (sleep), or inactive. Table 4.1 shows the possible states and the results of each combination.

Energy level discovery function: each node may notify its energy residual level.

Leader election function: consists in algorithms to choose or elect leaders (cluster-heads) such that each sensor node will be associated with at least one cluster-head as its leader in a hierarchical WSN. The leader election algorithm may provide the changing of the leaders following different criteria, for example residual energy, as well as, it must guarantee a good leader distribution (physical localization) and a balanced number of common-nodes per group.

Invitation to form cluster: the leader nodes send invitation to other nodes to form a cluster.

Listening for invitation: the nodes listen to other nodes for invitations to form a cluster. If the node does not hear anything within the network discovery time-out, it assumes that it is the first node and begins to send invitations for other nodes to join it. The network bootup latency specification determines the frequency of these invitations.

Response to invitation: once a node hears an invitation to join the network, it transmits a response. It is possible that multiple nodes will hear the same invitation and then

they will be part of the same cluster.

Calibration: typical traditional single-sensor calibration relies on providing a specific stimulus with a known result, thus creating direct input-to-output mappings. The calibration for any sensor is subject to specific ranges and operating conditions which are reported in the specifications of the sensor. This type of calibration is often performed at the manufacturer, at the production stage and/or manually in the field. With large-scale sensor networks, manual, single-sensor calibration schemes will not work well. In addition to the obvious scaling issues, the following problems also hinder such methods: limited access to the sensors, complex environmental effects on the sensors (dust on light sensor or salts on a nitrate sensor) and sensor drift (age, decay, damage, etc)

Power management: consists in a plane to manage how a sensor node uses its power.

Mobility management: consists detecting, planning, running, and registering the movement of sensor nodes.

Task management: consists in balancing and scheduling the sensing, processing and disseminating tasks given in a specific region. Task management can include micro task of the management.

Coverage area supervision function: consists in supervising the alterations in form and size of the monitored area.

Priority of action definition function: allows the establishment of priorities for operational actions depending on the state of the network.

Management operation schedule function: makes possible the establishment of a plan for the managing operations.

QoS Monitoring Function: consists in monitoring the quality of sensing, processing, and disseminating services.

We defined some functions that allow us to obtain characteristics which can determine the efficiency and effectiveness of a WSN. Some of these quantitative functions defined to obtain parameters are presented in [100]:

Network settle time function: consists in obtaining the time required for a collection of nodes to automatically organize itself and transmit the first message reliably.

Network join time function: consists in acquiring the time necessary for an entering node or group of nodes to become integrated into an ad hoc network.

Network depart time: consists in obtaining the time required for the network to recognize the loss of one or more nodes, and reorganize itself to route around the departed nodes.

Network recovery time function: consists in obtaining the time required for a colapsed portion of the network (due to traffic overload or node failures) to become functional again once the load is reduced or the nodes become operational.

Frequency of updates (overhead) function: consists in defining the number of control packets required in a given period of time to maintain normal network operation.

Memory requirement function: consists in computing the requisites of storage space in bytes, including routing tables and other management tables.

Network scalability function: consists in finding the network threshold, which is the number of nodes the network may escalate and confidently preserve the communication.

Energy consumption per task function: consists in discovering or predicting the energy consumption rate per task.

As previously discussed, the distributed management architecture MANNA is based on two paradigms: policy-based management and autonomic management. Also, the performance of WSNs and the management application depends on the routing and medium

access control of the underlying network. Thus, the qualitative features [100] to define policies regardless of the application involves:

Knowledge of node locations: Does the routing algorithm require local or global knowledge of the network?

Effect of topology changes: Does the routing algorithm need complete restructuring or only incremental updates?

Adaptation on radio communication environment: Do nodes use estimated knowledge of fading, shadowing, or multiuser interference on links in their routing decisions?

Power consciousness: Does the network employ a routing mechanism that considers the remaining energy of nodes?

Single or multichannel: Does the routing algorithm uses a separate control channel? In some applications, multichannel performance may cause the network to be vulnerable to countermeasures.

Bidirectional and unidirectional links: Does the routing algorithm perform efficiently on unidirectional links, e.g., if bidirectional links become unidirectional?

Preservation of network security: Do routing and MAC layer policies support the survivability of the network, in terms of low probability of detection, low probability of interception, and security?

QoS routing and handling of priority messages: Does the routing algorithm support priority messaging and reduction of the latency for delay sensitive real-time traffic?

In the majority of the management applications, the MANNA architecture uses automatic services and functions performed by a management entity invoked as a result of information acquired from a WSN model. Management services and functions can also be

semi-automatic, when performed by an observer assisted by a software system that provides a network model or invoked by a management system, and manual, when performed outside the management system.

Six possible states are defined for a function:

- ready, when the necessary conditions to carry out a function are satisfied;
- not-ready, when the necessary conditions to carry out a function are not met;
- running, when the function is being performed;
- done, when the function performed well;
- cancelled, when a cancellation occurs;
- failed, when a failure occurs during function execution.

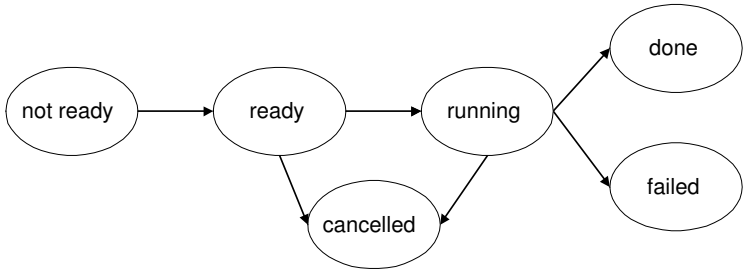


Figure 4.2: Possible states of a function.

The above management function list will be helpful in the development of the information model.

4.3 Defining Management Services

The purpose of this approach is to provide a description of the processes leading towards the definition of the management services. The definition of the management services[2] is

[2]Note that the term management service is different from the service management functional area.

a task that consists in finding which activities or functions must be performed, when and with which data they should be performed. Management services are carried out by a set of functions, and they need to succeed to conclude a given service. The input data for each function is obtained from management information base or in the WSN models (see Section 4.4). The WSN models, defined in the MANNA architecture, represent aspects of the network and serve as a reference for management. These models provide abstract views of the system, through which is possible to hide all non-relevant aspects given a certain objective. The conditions for performing a service or function are described by rules and are obtained from the WSN models.

Figure 4.3 represents a scheme to develop management solutions, starting at the definition of management services and functions that use models to achieve their goals. A management service can use one or more management functions. Different services can use common functions that use models to retrieve a network state concerning a given aspect. Therefore, the management functions use and generate management information as well. For example, the conditions to perform a service or a function can be based in a "budget",

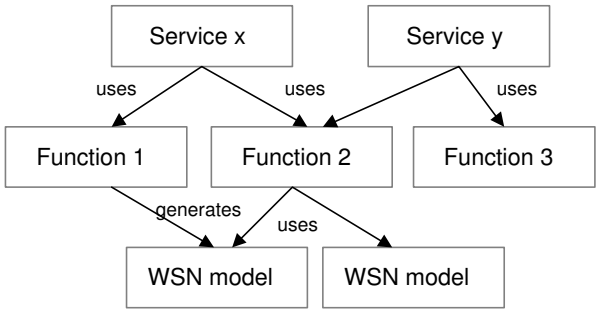


Figure 4.3: Services, functions and WSN models.

that is, the amount of energy that the active sensor nodes or the network can spend during the execution of the service. If there is not enough energy in the current energy map to run the algorithm of the function, the management application can reschedule the functions. The information about the remaining available energy in each part of the network is represented by a WSN model called energy map. The energy map can also enable the

management application to determine if any part of the network is about to suffer system failures due to energy depletion. The knowledge of low-energy areas can aid in deciding whether or not to wake up backup nodes and to incrementally deploy new sensors nodes when they can be placed selectively on those areas that are short of resources. Taking into account energy consumption, it would be interesting to determine the energy consumption associated with each management function, including those which build and update WSN models.

Management services can be clustered and performed by distinct functions in different manners (automatic, semi-automatic, and manual) according to the WSN application. Some of the management services sets are described in the following. In the service definitions, the term "entity" means a node, a cluster or a network.

Planning. This service involves the design of the network and the decisions about node architectures, number of nodes, type of deployment, and so on. It is the management service performed before the network boot up time to decide how the monitoring network is to be placed and done. Examples of management functions which are included in this service are: environment requirements acquisition function, monitored area definition function, environment monitoring function, node definition, number of nodes, and nodes deployment definition function.

Placement. Is the management service that includes all functions related to sensor nodes deployment on a certain region.

Self-organization. Refers to the management service used to achieve the necessary organizational structures without requiring human intervention. The efficiency of this organizational process can be heavily dependent on the particular deployment of the network and the degree and accuracy of the information that is pre-programmed in the nodes. The self-organization service is composed of initialization routines, network discovery routines, node type announcement, program/command injection/exchange, topology learning and position determination routines, nodes scheduling, initial traffic determination routines, routing routines, medium control access routines, network time distribution routines, and dynamic connection establishment/disestablishment routines. Self-organization means to

dynamic adaptation to environmental conditions and states of the network variation in order to maintain its operation.

Node setup. Upon power up, a node will perform a number of initialization routines, such as internal node self-test health status determination, and built-in calibration. It will also launch any procedures that have been pre-programmed to reflect specific mission requirements and expectations.

Self-configuration. Is the management service that changes the parameters of configuration to adapt itself dynamically to the changing conditions or states of the network. It configures and reconfigures itself under varying (and in the future, even unpredictable) conditions. System configuration or "setup" must occur automatically, as well as dynamic adjustments to the current configuration to best handle changing environments.

Self-diagnosis.[3] Is the management service that qualifies the network to monitor itself and find faulty or unavailable nodes.

Self-protection. Is the management service that anticipates, detects, identifies and protects the entity against threats and attacks. When an attack happens, these services perform intrusion detection routines to reach securite and safe states.

Self-healing. Is the management service that prevents disruptions or that acts to recover the network or the node after the self-diagnostic (if possible). It enables the entity to recover from problems that might have happened. It must be able to discover potential problems and then find an alternate way of using resources or reconfiguring the entity to keep in normal operation.

Self-optimization. Is the management service that tunes resources and balances tasks to maximize the use of resources, minimizes latency, and maintains the quality of service. An entity always looks for ways to optimize its job.

Self-service. Is the management service that enables a entity to provide sensing, processing, and disseminating services, anticipating the optimized resources needed while keeping its complexity hidden. It must marshal resources to shrink the gap between the ap-

[3]An approach using self-diagnostic is presented in appendix A.

plications business or service goals (QoS sensing, QoS processing, and QoS disseminating), and the implementation necessary to achieve these goals.

Self-awareness. Is the management service that allows the entity to know its environment and the context surrounding its activity, and act accordingly. It will find and generate rules for best interacting with neighboring entities. It will tap available resources, even negotiate their under-utilized elements used by other entities, changing both itself and its environment in the process – in a word, adapting.

Self-knowledge. Is the management service that qualifies a entity to "know itself". For example, a entity that governs itself must know what are its components, current state, ultimate capacity, and all connections to other entities. It will need to know the extent of its resources, which ones can be borrowed or lent, and which ones can be shared.

Self-sustaining. Is the management service that uses budget schemes to prevent energy waste and promote rational use of energy in order to survive.

Self-maintenance.[4]. Is the management service that enable an entity to monitor its constituent parts and fine-tune itself to achieve pre-determined entity goals. One of the main examples of the maintenance services is "coverage area maintenance management services" which uses the density control function to identify which nodes can be administratively put out of service in order to reduce congestion, collision and energy waste. Other examples of management services in this set are: service negotiation, QoS maintenance, mobile management, scheduling task, key management and differentiation of services.

The MANNA architecture also proposes three architectures: information, functional and physical. The following sections discuss how the MANNA architecture can cope with different kinds of networks and present the functional, information, and physical architectures.

[4]An approach to perform this service was published in the IEEE Local Computer Management 2003 [103] and IEEE LANOMS – Latin American Network Operations and Management Symposium 2003 [79].

4.4 Information Architecture

To ensure common solutions for WSN management, the MANNA architecture defines an information model[5]. An information model provides a foundation for understanding the interrelationships between the resources and attributes, and may, in turn, promote uniformity in dealing with the various aspects of resource and attributes management.

The definition of management information must take into consideration all three dimensions, namely management levels, management functional areas, and WSN functionalities. Orthogonality among the three dimensions should be maintained in the descriptions of the information model to avoid redundancy in it.

In WSN management, there are two kinds of management information: static and dynamic. Static management information describes the configuration of services, network and network elements. Dynamic management information describes the information that changes frequently. In the MANNA architecture, static management information is object-oriented based and dynamic management information is described by WSN models (see Figure 4.3).

4.4.1 Static Information

There are two types of object classes which represent resources under the three different dimensions: managed object class and support object class. The managed object class directly relates with the network components and with the network itself. The support object class plays the role of supporting the management functions, i.e., making available necessary information to them.

The specification of an object class is done through pre-defined syntactic structures called templates, based on the Abstract Syntax Notation.1 (ASN.1) language, which is used to describe the objects and their characteristics.

The object classes may be inherited or reused from standard objects. The reuse allows

[5]The contents of this section were published in the "Colloque Francophone sur la Gestion de Reseaux et de Services" [83].

future management integration. Some object classes and their new attributes, based on WSN characteristics, are listed below.

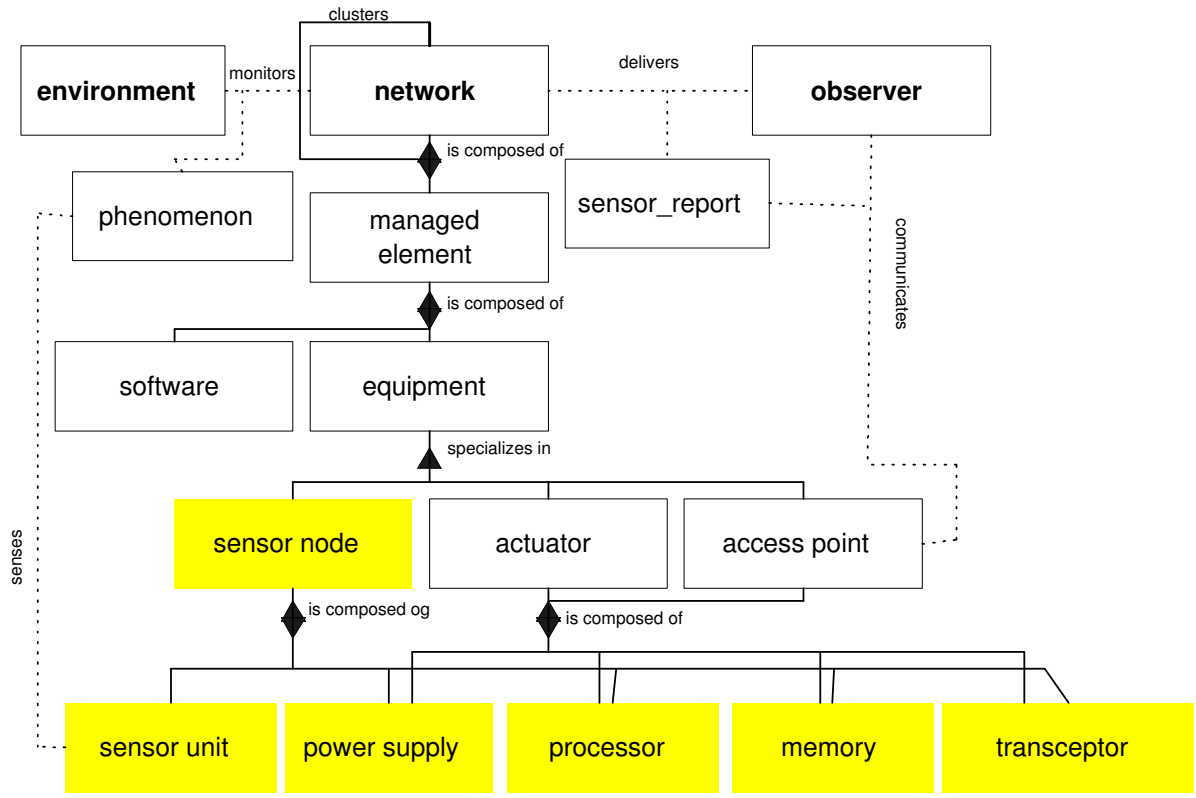


Figure 4.4: Simplified class diagram.

Support Object Classes. These classes can be programmed by the agent or can be presented in the management application. They are mostly derived from the OSI reference model. Some support object classes are: log, state change record, attribute change value record, event record, event forwarding discriminator, management operation schedule, information log, management log, energy level severity assignment profile, current remaining energy level summary control, monitored object, current data object, history data object, threshold data object, scanners, and so on.

Managed Object Classes. The RFC 3433 [11] describes managed objects for extending the Entity MIB (RFC 2737) to provide generalized access to information related

to physical sensor devices, which are often found in networking equipment (such as chassis temperature, fan RPM, and power supply voltage). The objects of the RFC 3433 are used in generic information model proposed for WSNs. Other object classes are defined and presented bellow:

Network. It is composed by interconnected managed objects (physical or logical ones) capable of exchanging information. Examples of new attributes for this class are the network identifier, the composition type (homogeneous or heterogeneous), organization type (flat or hierarchical), organization period, mobility (stationary, stationary nodes and mobile phenomenon, mobile node or mobile phenomenon), data delivery (continuous, event-driven, on-demand, programmed or hybrid), type of access point (sink node or base station) localization type (relative or absolute), control (open or close), mission (critical or common) node distribution (regular, irregular, balanced, sparse or dense), and node deployment (it is affected by many factors, some of them being the sensor node capabilities of individual nodes, radio propagation characteristics, and the topology of the region). Other constraints may include a degree of overlapping in the sensor coverage of two nodes so that they may collaborate.

Managed Element. It represents the sensor node and actuator nodes or other WSN entities, which perform functions on managed elements and provide sensing, processing, and communicating services. Examples of new attributes of this class are: localization type (relative or absolute), element type (common-node, sink node, gateway or cluster-head), minimum energy limit, and mobility (direction, orientation and acceleration). The problem is where to place the base station or sink node. Some approaches use a combination of computational geometry, Computer Aided Design, and numerical optimization methods.

Equipment. It represents the physical components of a managed element. In this case, this class represents the physical aspects of the sensor node constitution, which is composed of memory, processor, sensor device, battery, and transceiver. The equipment class can be specialized in object classes. For instance, (1) battery type (linear, discharge rate dependent model, relaxation model, battery capacity, remaining energy level, energy density, and max current); (2) computational module composed of processor and memory

(clock, state of use, available memory, endurance, AD channel, operating voltage and IO pins); (3) sensor element (sensor type, current consumption, voltage range, minmax range, accuracy, temperature dependence, version, current state, exposure, collect type – periodic, continuous, reactive and real time); and (4) transceiver (type, modulation type, carrier frequency, operating voltage, current consumption, throughput, receiver sensitivity and transmitter power).

System. It is used to represent the hardware and software, which constitute an autonomous system capable of processing and/or transferring information. Examples of new attributes are: operating system type, version, code length, complexity, total MIPS per available MIPS, and synchronization type (mutual exclusion and synchronization of processes). A notification of change in an attribute value must be reported upon the event occurrence, such as a software upgrade.

Environment. It represents the environment where the WSN is operating. Examples of new attributes are: environment type (internal, external or unknown), noise ratio, atmospheric pressure, temperature, radiation, electromagnetic field, humidity, and luminosity. The environment can present static and dynamic features.

Connection. It represents the actual connections and it is expressed as an association between particular points. The direction of the connectivity can be unidirectional (asymmetric) or bi-directional (symmetric). If an instance of this class is unidirectional, the point "a" will be the origin and the terminal point "z" will be the destination. The operational state will indicate the capacity to load a signal. An example of an attribute for this class is the communication direction (simplex, half duplex or full duplex). The network topology describes the connections, which may exist, and it is expressed as relationships among a set of points.

WSN Observer. It represents the entity that requires the WSN services. It may be a human user applying for the use of services via some human-machine communication scheme or it may be some computer-based organizational system.

WSN Goals. WSN goals acquired by carrying out WSN activities and using WSN services, are the benefits provided to the users. They can be defined as accuracy, latency,

fidelity, etc.

WSN Management Context. The WSN context defines the environment where WSN management services are carried out. The definition includes the description of the entity responsible for managing the network, what is managed and how it can be managed. Figure 4.5 illustrates the management context set which includes policies, resources, functions, and services used to define who manages, what is to be managed, how they are managed, what is required, and what the benefits are. The WSN management context shall be described by using three dimensions: management functional areas, management levels, and WSN functionalities (see Figure 3.1).

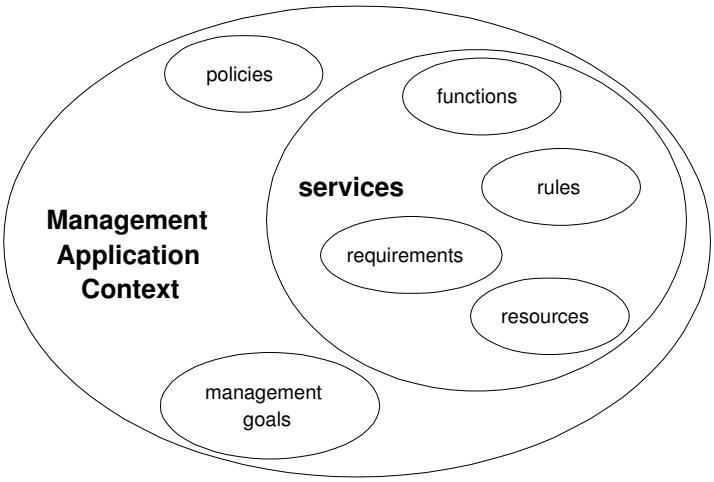


Figure 4.5: Management context.

4.4.2 Dynamic Information

Dynamic management information is described by WSN models and has to be updated frequently. The acquisition of this information has a cost in terms of energy consumption. Therefore, an important aspect is to determine the adequate moment, frequency, and fidelity for updating this information. Furthermore, the collected information may not be valid at the moment it is processed by the management entity due to delays, omissions,

and uncertainty present in WSNs. Static information can be necessary in order to obtain the WSN models.

In a WSN, the network conditions may vary dramatically along the time. In this case, the use of models established by MANNA is of fundamental importance for management, although its updating cycle can be extremely dynamic and complex. Based on the information obtained with these models, services and functions are carried out according to management policies.

In the following, some network models are presented. They always represent dynamical aspects of the network. The dynamic information represented in the WSN models can or cannot be stored in Management Information Base (MIB). Some WSN models can be obtained from the combination of other models using management information stored in MIB. Some of the WSN Models[6] (maps) are given below:

Network topology map. It represents the topology map and the reachability of the network;

Residual energy.[7] It represents the remaining energy in a node or in a network. Using the energy map, the management application can determine if any part of the network is about to suffer system failures in the near future due to depletion of energy.

Sensing coverage area map.[8]. It describes the actual sensing coverage map of the sensor elements;

Communication coverage area map.[9] It describes the actual communication coverage map from the range of transceivers;

Cost map. It represents the cost of energy necessary for maintaining the desired performance levels;

[6]A work about approaches to obtain WSN models was published in the IEEE Workshop on Future Trends of Distributed Computing Systems, May 2003 [92].

[7]A fault-tolerant approach to obtain energy map was published in the Workshop de Comunicação sem Fio e Computação Móvel, October 2003 [104].

[8]A coverage map was used in a work published in the IEEE Latin American Network Operations and Management Symposium 2003 [79].

[9]An approach to obtain sensing coverage map and communication coverage map was published in the IEEE Local Computer Network 2003 [103].

Production map. It represents the nodes that are producing and delivering their data;

Usage standard map. It represents the activity of the network. It can be delimited for a period of time, for quantity of the data transmitted to each sensor unit, or for the number of movements made by the target;

Dependence model. It represents the functional dependency that exists among the nodes;

Structural model. It represents the aggregation and connectivity relations among network elements;

Cooperational model. It represents relations of interaction among network entities.

Audit map It represents records which permit the verification of whether a security violation is happening or happened.

Coverage area map There are some possibilities to determine the coverage area map using the sensing and communication maps, as shown in Figure 4.6, when considering sensor range and radio range: sensor range greater than, less than, or equal to radio range.

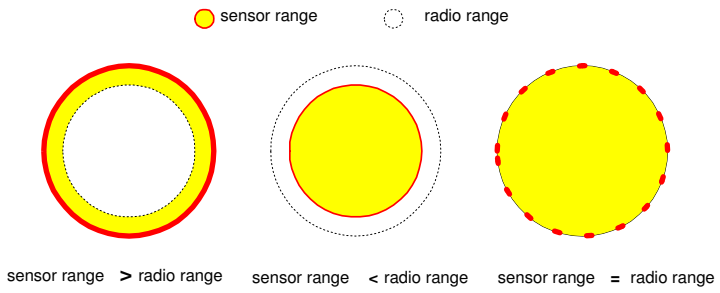


Figure 4.6: Sensor and radio range possibilities.

4.4.3 Issues Concerning Management Information Base Implementation and Usage

The description of objects present in the information model and the relationship among them are specified in the management information base. In a WSN, to update a MIB with the current network state may require the measurement of various parameters. In general,

the collection of these parameters can have spatial and temporal errors. This is called the uncertainty problem.

To have a higher precision in the network state, probabilistic measures should be taken with a higher granularity [21]. As in any probing, this would take a finite amount of system energy and could modify the network state. This is called the probe effect. In this way, a better precision in the management information requires the modification of the state.

The MANNA architecture proposes the scope limitation as a method for reducing the uncertainty and the energy consumption while updating the MIB. Spatial limitation consists in defining a physical space inside which data will be considered for management. Temporal limitation defines a time window (fixed or sliding) inside which the collected data are considered. Functional limitation selects the data of a certain functional network segment for management, for example, the data of a group of nodes or a group leader.

The following section presents the functional architecture. From the management point of view, the MANNA functional architecture establishes the circumstances in which a manager receives event notifications and how it can get the information (monitoring). It also makes clear what kind of influence the management system has over the WSN resources and how to control them.

4.5 Functional Architecture

The functional architecture describes the distribution of the network management functionalities among manager, agent, and MIB. In the architecture, it is possible to have a diversity of manager and agent locations. The management choice depends on the functional areas involved, the management level considered, and the application running in the WSN, i.e., depends on the network functionalities (Figure 3.1). This architecture introduces the organizational concept of a management "domain". A domain is an administrative partition of the network for the purpose of network management. Domains may be useful for reasons of scale, security, or administrative autonomy. Each domain may have one or more man-

agers monitoring and controlling agents that belongs to that domain. In addition, both managers and agents may belong to more than one management domain. Domains allow the construction of both strict hierarchical and fully cooperative, and distributed network management systems.

There have been basically two models for network management - distributed and hierarchical/centralized. The WSN management can be centralized, distributed or hierarchical. In a centralized management network, there is a single manager, which collects information from all agents and controls the entire network. A distributed management network has several managers, each one responsible for a subnetwork which communicates with other managers. In a hierarchical management network, there are intermediate managers to distribute the management tasks.

4.5.1 WSN Manager

In distributed network management there is no central network manager, and managers act as peers. This architecture will be called Manager to Manager (M2M). In centralized management model, network management is controlled form a single point, the central manager. This manager may manage network resources from a centralized computing environment. There may be a centralized manager or a hierarchy of managers controlled by a super manager called Manager of Managers (MoM).

The management alternative to be chosen depends on the application running on the WSN. In any case, it may be important to have a manager entity located externally to the WSN. The external manager has a global view of the network and can perform complex tasks (automatic services and functions) that would not have been possible inside the network. However, this manager can be the only one manager (centralized management) or it can collaborate with another ones localized inside the network (decentralized management).

4.5.2 WSN Agents

The development of a functional architecture raises the question of what is the most adequate location for an agent, given a particular kind of WSN. A possible alternative to the agent location is to place it close to the manager, i.e., external to the network. Nevertheless, this may cause management to be isolated and makes it difficult to integrate it with other management systems in the future.

In the following sections, some possible configurations are explored for different WSN organizations (flat and hierarchical) and compositions (homogeneous and heterogeneous):

4.5.2.1 Agents in Homogeneous Flat WSNs

A flat WSN has at least one sink node to provide network access. All network nodes have the same hardware configuration. Some possible alternatives for homogeneous flat networks, considering agent location are:

1. Agents inside the network and external manager (Figure 4.7(a)).

2. Agent in the sink node and external manager (Figure 4.7(b)).

3. Agents and manager inside the network. The two possibilities for manager organization are hierarchical (MoM – Manager of Managers) (Figure 4.7(c)), where there is an external manager and another one in the sink) and distributed (M2M – Manager to Manager) (Figure 4.7(d)).

In any of these proposals, the main concern is the large amount of traffic that may be generated in response to operation requests and notification emissions. Another alternative is to place managers inside the network allowing them to communicate among themselves. This defines a distributed management.

In case of having agents as part of common-nodes, some questions remain such as how to distribute the agents, how to define domains for the agents, and how to deal with nodes with more than one agent.

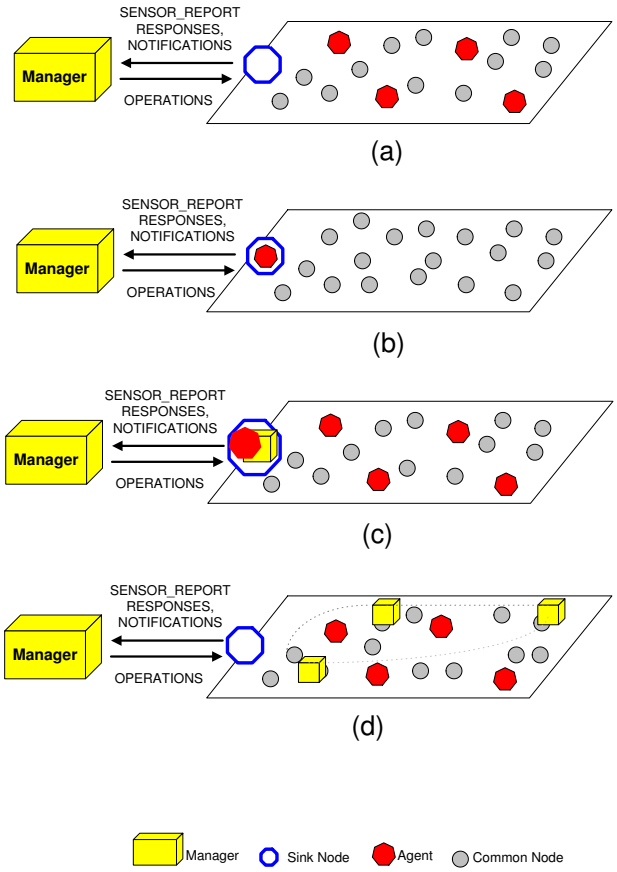


Figure 4.7: Manager and agent location in flat WSNs.

4.5.2.2 Agents in Heterogeneous Flat WSNs

In a heterogeneous WSN, nodes differ in their physical hardware capabilities. Agents can be placed in more powerful nodes, as long as they may present adequate location in the network. The sink node can host an intermediate manager or even present no management functions at all. To establish a distributed management, we can place agents in less powerful nodes and managers in more powerful ones.

4.5.2.3 Agents in Homogeneous or Heterogeneous Hierarchical WSNs

In this kind of network, there is no sink node. A cluster-head node is responsible for sending data to a base station. It also communicates with the observer. The cluster-head

may also perform the correlation of management data. This computation may decrease the information flow and, consequently the energy consumption. The correlation may also allow a data multi-resolution where differences are filtered and a higher precision is obtained.

Some possible alternatives for a hierarchical WSN, considering the agent location are:

1. Agents in cluster-heads and an external manager (Figure 4.8(a)).
2. Agents in the base station and external manager (Figure 4.8(b)).
3. Agents inside the network and an intermediate manager (MoM – Manager of Managers) (Figure 4.8(c)).
4. Agents and distributed managers inside the network (M2M – Manager to Manager) (Figure 4.8(d)).

The following section presents a discussion about protocol profiles, which can be used in the exchange of information among management entities. In the MANNA architecture, this issue is addressed at the physical architecture.

4.6 Physical Architecture

The physical architecture defines how the management information is exchanged between management entities. It can be seen as the implementation of the functional architecture. In doing so, physical aspects such as the management protocol, the physical location of agents, agent functionalities, implemented management service, and supported interfaces for WSNs are defined. The interface among management entities should use a light-weight protocol stack. The MANNA architecture does not define a protocol stack for these interfaces, but provides protocol profiles which may be adequate for each application type. Below, we discuss the main aspects relation to each layer of protocol stack. Some details about the protocols developed for WSNs were presented in Section 2.5.

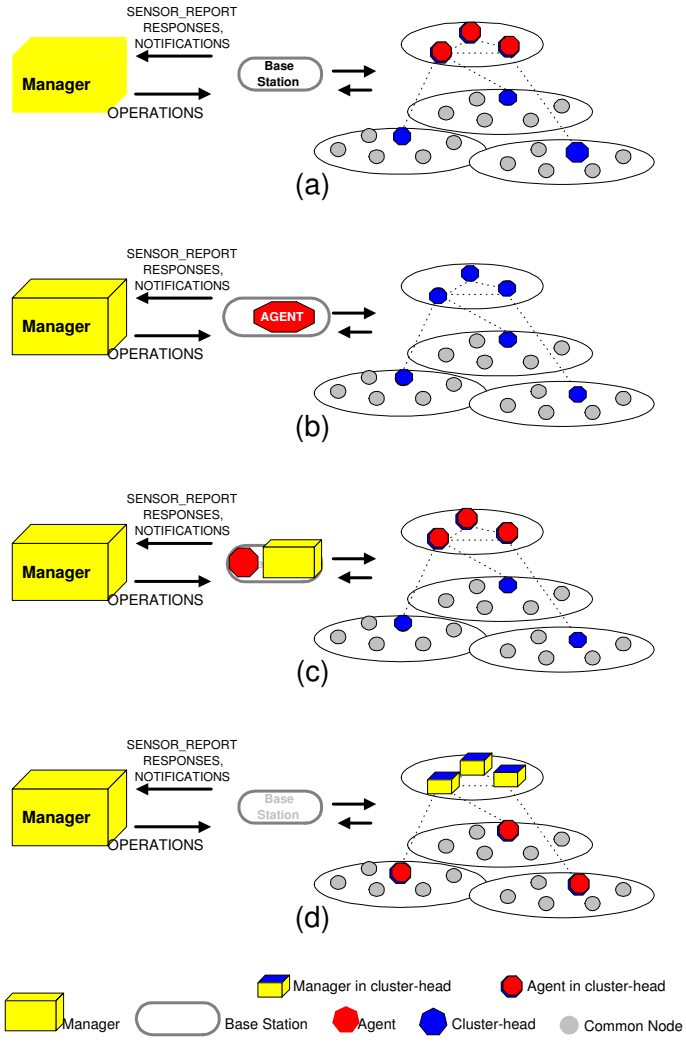


Figure 4.8: Agent location in hierarchical WSNs.

Application layer. Although the Simple Network Management Protocol (SNMP) [108], the Common Management Information Protocol (CMIP) [43], the Web Based Management (WBM) [22] and the Ad hoc Network Management Protocol (ANMP) [17] allow management in a decentralized and event-oriented way, the structure of managed components is always rather rigid. In these paradigms, the management intelligence always resides in the management instance, while the information is generated at the managed instances. An alternative method would be the delegation of the

management functionalities to the managed systems. A solution for supporting this feature in the implementation of the physical architecture is the Management by Delegation (MbD) [32]. Other alternatives are "intelligent agents" and mobile agents. In the model of mobile agents, data stays at the local place while the processing task is moved to the data locations. Management functions are performed locally and only the resulting data are sent to the manager.

By transmitting the code instead of data, the mobile agent model offers several important benefits: reduction in network bandwidth requirements, which is especially important for real-time applications and when communication uses low-bandwidth wireless channels; agents can migrate to another node when the hosting node is compromised; network scalability is supported; agents can migrate to regions of interest independently of the movement of nodes, if they are mobile; extensibility is supported, that is, mobile agents can be programmed to carry out task-adaptive processes, which extend the capability of the system; more stability is achieved because mobile agents can be sent when the network connection is alive and return results when the connection is re-established along with the network data; it reduces the delay in the management actions; managers are not required to instruct agents all the time; the main management part does not reside in the manager; and agent cloning offers means for robustness and fault tolerance.

Transport layer. For all protocols described in the application layer, the correct reception of data messages is not assured [105]. Unlike traditional networks (e.g., IP networks), reliable data delivery is still an open research question in the context of WSNs.

Network layer. It should be designed considering power efficiency, and that WSNs are mostly data-centric. Data aggregation is useful only when it does not hinder the collaborative effort of sensor nodes. Energy efficient routes can be found based on the available power in the nodes and the energy required for transmitting data in the link along the route.

Data Link Layer. Data link layer is responsible for the multiplexing of data streams, data frame transmission and reception, medium access and error control. Medium access control has two goals: creating the network infrastructure to establish communication

links for data transfer and giving the sensor network self-organizing ability; and sharing communication resources between sensor nodes fairly and efficiently. Simple error control codes with low complexity encoding and decoding might present the best solutions for sensor networks. Open research topics for MAC protocols in WSNs are the determination of low bounds on the energy required for sensor network self-organization, error control coding schemes, and power-saving modes of operation [68].

In most of the existing or proposed ad hoc networks, channel access is done by two different methods: contention or explicit organization in time/frequency/code domains. The MAC-layer design for 802.11 [41] is an example. The second class of channel access schemes, which are called "organized" channel access, attempts to determine network radio connectivity first, that is, the discovery of the radio neighbors of each node and the assignment of collision-free channels to links. The task of channel assignment (i.e., TDMA, slots, frequency bands, or spread spectrum codes) to links between radio neighbors so that they do not collide is a hard problem. The contention-based channel access schemes are clearly not suitable for sensor networks, due to their requirement for radio transceivers to monitor the channel all time. This is a particularly expensive proposition for the low radio ranges of interest for WSNs, where transmission and reception have almost the same energy cost.

Turning off the radios when no information is to be sent or received can be interesting [97]. The organized methods of channel access require that nodes in the network to be synchronized with each other at some level. In organized schemes, usually a period of time is set aside for neighbor discovery. If a centralized channel assignment algorithm is to be used, all the connectivity information (along with any bandwidth requirements for specific links) is sent to a single node in the network for schedule calculation. There are distributed assignment methods where nodes exchange connectivity data only with some local neighborhood. This network-wide synchronization is again expensive for WSNs, because it requires extensive message exchange over the air to synchronize all nodes [97].

Physical Layer. It is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption. The 915 MHz ISM (Industrial Scientific and Medical) band has been widely suggested for sensor networks. All signals decay with

distance as the wavefront expands. For example, in free space, electromagnetic waves decay in intensity with the square of the distance; in other media, they are subject to absorption and scattering effects that can induce even steeper declines in intensity with distance. Many media are also dispersive and obstructions can render electromagnetic sensor useless. Propagation is influenced by surface roughness, the presence of reflecting and obstructing objects, and antenna elevation.

4.7 Building Management Applications

Management architectures (information, functional, and physical) define how the management entities receive and analyze information and react to it, how the information is represented, which services and functions will be executed and how the information is exchanged through the communication interface. The type of management (centralized, hierarchical (MoM), and distributed (M2M)) is also established.

Centralized management for WSNs, as well as for traditional ad hoc networks, is not always appropriate. One important reason is the traffic concentration problem, caused by a central manager that receives and originates management traffic. In addition, the response implosion problem may happen when there is a high volume of incoming replies triggered by management operations or events. In case of WSNs, there will always be one access point (sometimes more than one) through which data go to the observer or to the management application. The access point represents a sink node or a base station that can make use of a gateway to communicate with the external environment.

One possibility to solve the response implosion problem for the management and application is to select only a subset of nodes sending data. This solution is known as fidelity. In case of management, some agents are selected to send replies back. This approach may be suitable for densely populated sensor networks with a large number of sensor nodes, where missing information from some nodes can be ignored with an acceptable accuracy. The accuracy of the calculation might significantly degrade in a sparse sensor network, or a network with a small number of nodes not collecting enough replies. However, the number

of replies may not be small enough to be received without taking into account the response implosion problem. Other solution is to make a scheduled response approach [46].

A management solution depends on the characteristics of the network. There are WSNs where only a few management functions can be implemented. In other cases, the management functions must be semi-automatic or manual because of restrictions in the computation. The MANNA architecture is built to provide a management solution to different WSNs applications. Depending on the WSN application, it may be interesting or not to use certain management services, which also can be implemented as automatic, semi-automatic, and manual.

A management solution must also be proposed considering the type of the dissemination: continuous, on-demand, programmed and event-driven (see Section 3.3.2.4). In a continuous monitoring scheme, agents are programmed to send monitoring data continuously to a manager. In an on-demand scheme, a manager sends a query to one or more agents and receives data back from these agent nodes. In an event-driven monitoring scheme, agents are programmed to send data to a manager only when an event happens and a local condition is satisfied.

There are pros and cons of using each of these management solutions. In a continuous monitoring scheme, when the management application stops receiving data from a given node, this may be an indication of a problem, mainly if the previous sensor condition was normal. The cost of sending data continuously may lead to a more rapid consumption of the scarce network resources, thus, shortening its lifetime. In an on-demand and programmed scheme, the monitoring node can become aware of a problem in the network after sending a query to a node. The cost of having this information is proportional to the number of queries sent or the number of programmed responses. Finally, the design of an event-driven monitoring scheme makes some assumptions about how events are generated. If they happen unpredictably way then, again, there is the problem of network resources consumption . On the other extreme, if a node does not report an event, it may be either an indication of a failure or the event did not happen at all. In both cases, the management application cannot differentiate them. The same is true for on-demand networks. In normal

situations, an event-driven scheme only sends an event to the sink node when the event happens. This is the minimum possible cost associated to an event when it has to be sent to the management application.

In energy-constrained WSNs, event-driven networks represent an attractive option when compared to continuous networks, because they typically send and receive far fewer messages. This results in a significant energy saving, since message transmissions are much more energy-intensive than sensing or (CPU) processing intensive.

In terms of failure detection, event-driven networks present challenges not found in continuous and programmed networks. Under normal conditions, a management application of a continuous network receives sensing data at regular intervals. This stream of data not only delivers the content we are interested in, but also works as an indicative of how well the network is operating. If the management application receives data from every single node, then we know that all is well (of course, assuming that the messages are authenticated, and cannot be spoofed). If, however, the management application stops receiving data from certain nodes or entire regions of the network, we know that a failure has occurred.

A scheme to develop a management application includes: (1) the identification each area of management activity which is to be supported by the management application in the form of a list of management services. (2) For each management service, identify the management goals. A list of management services with a brief textual description of each service is provided. (3) the development of the management context. (4) a list of roles, resources and functions associated with a given management service. (5) Also specify their relationships, where possible in the form of scenarios. An example of a list of descriptions of management roles, resources and management functions (or function set/group) for that part of the management service selected is presented in Section 4.8. (6) For each management function in the function list, check if it is supported by one or more object classes. A function in the function list is supported by one or more object classes when the monitoring part of the function can obtain all the necessary information from the objects; and/or the control part of the function has the necessary influence over the objects. If a

management function is not supported by one or more of the existing object classes, then new object classes may be defined or existing object classes may be extended, e.g. by specialization, thereby creating a subclass. Some object classes are defined only for the purpose of creating subclasses. Therefore, not all object classes have a relationship with one or more management functions. (7) the determination of the type of management and who will perform the management services and functions as well as if these services and functions will be automatic, semi-automatic, and manual.

4.8 Putting It All Together

Consider that a management entity has just received the location and energy messages. It calculates the sensing and communication range area maps and detects the existence of high node density, because there are lots of intersections among the sensing range of the nodes. The management entity faces a redundancy problem of the sensing data received. On one hand redundancy provides a mechanism for fault tolerance and multi-resolution (gives better accuracy), but on the other hand, it represents a waste of resources.

This redundancy problem can be detected by the MANNA architecture using the WSN models, in particular, the "Topology Map", "Energy Map", "Communication Coverage Area Map", and "Sensing Coverage Area Map". Based on these maps, maintenance services may be performed. These services are automatic and executed by a set of functions. These functions use and generate the management information. In this case, one of the functions invoked is the "Node Administrative State Control Function".

This function represents the intersection of the three abstraction dimensions for the Configuration Functional Area, Network Element Management Level and Sensing Functionality. The function allows putting the redundant nodes in the administrative state locked. For this, the agent assigns the value "locked" for the administrative state attribute of the objects (present in the MIB) which represents such nodes, acts over the nodes and removes them from the sensing, processing and dissemination services. Figure 4.9 shows an UML diagram that represents the process just described.

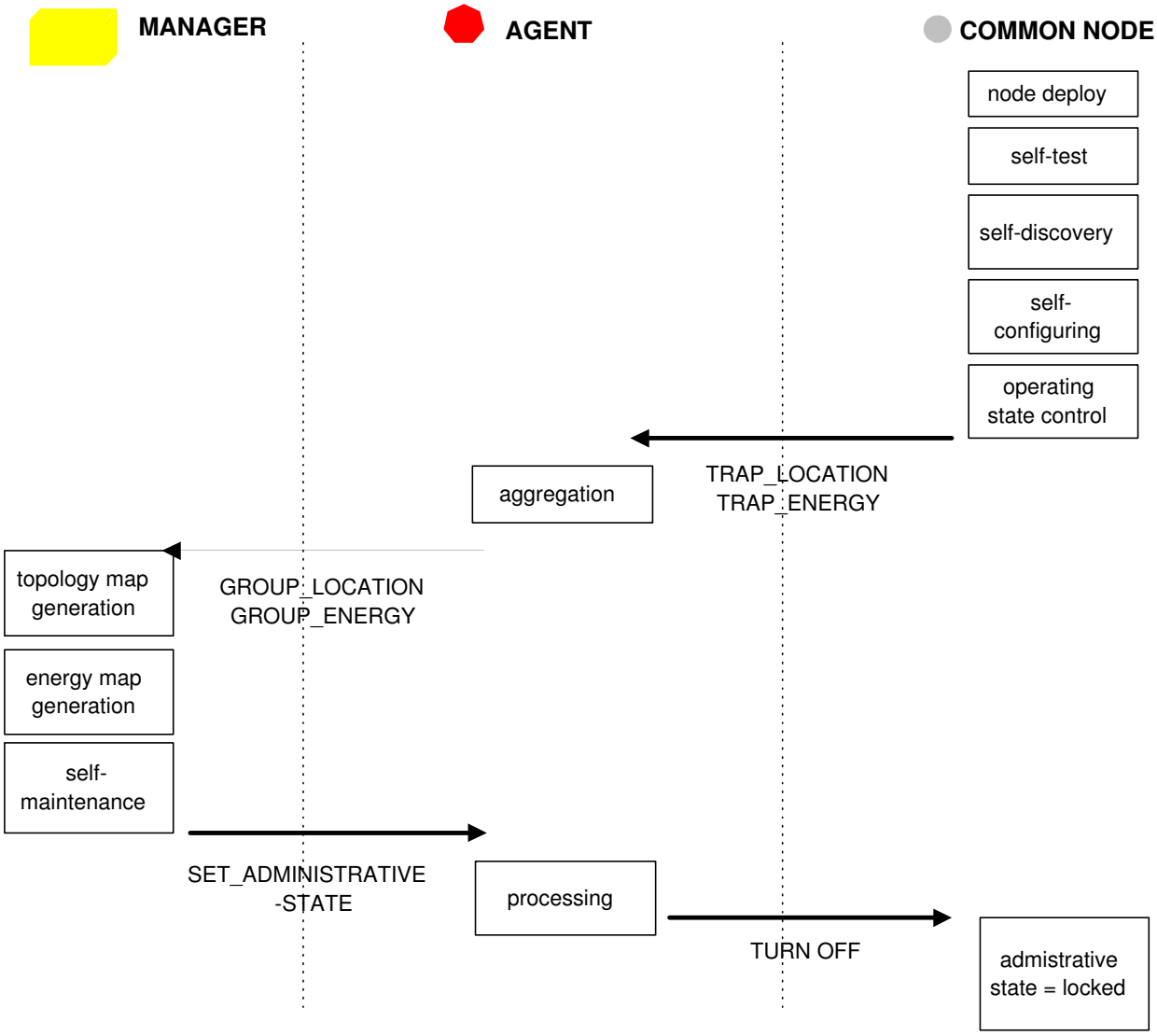


Figure 4.9: Applying the MANNA architecture: an example.

4.9 Conclusion

WSN management must be simple, adherent to network idiosyncrasies, including its dynamic behavior, and efficient in its use of scarce resources. The approach used in the MANNA architecture works with each functional area, each management level, and each WSN functionality to obtain management functions, to build management services, to

identify management information, and to define the type of management (centralized, hierarchical, and distributed). This chapter also shows that the MANNA architecture establishes a separation between application and management through a proposition of three architectures (information, functional, and physical) and using three management dimensions (management functional areas, management levels, and WSN functionalities). This will make possible the integration of organizational, administrative, and maintenance activities for this kind of network. The adoption of a strategy based on the traditional framework of functional areas and management levels will allow management integration in the future.

One of the major goals of management architecture is to promote network resources productivity and the quality of the service provided. This thesis does not aim to implement a management system for WSNs. Its goal is to propose a management framework for wireless sensor networks. Aiming to show how the architecture proposed can achieve its objectives, in next chapter, an application and a management solution using the MANNA architecture are built to show how these solutions can be obtained.

Chapter 5

Developing Management Solution for Continuous WSNs

This chapter[1], presents experiments that were conducted to show how the proposed management framework can be used in the development of management solutions. An application which continuously does data sensing, processing, and dissemination was defined as a case study. Some management services and functions were chosen from the lists provided in Chapter 4 as well as some WSN models and managed object classes. Scenarios were developed with different WSN configurations in terms of organization (flat and hierarchical) and composition (homogeneous and heterogeneous). Architectural decisions were made considering the type of the network. A second goal of these experiments is to show how the network configuration influences metrics such as delay, message loss, and energy consumption. The implementation of a management system is not one of the objectives of this thesis. The experiments are described in Section 5.1. To perform these experiments, a simulation tool has been developed and is presented in Section 5.2. Section 5.3 presents some assumptions that were made regarding the simulations. The results presented in Section 5.4 show the impact of the management solution on the WSN application developed

[1]The experiments and results of this chapter were accepted and will be published at the IEEE/IFIP Network Operations and Management Symposium 2004 [81] and were published at the Workshop de Comunicação Sem Fio e Computação Móvel 2003 [80].

as a case study, and the cost-benefit relation of different network organizations simulated.

5.1 Experiments

As mentioned before, one of the major goals of a management architecture is to promote productivity of the network resources and the quality of the service provided. Probably, the important issue about the management of a WSN concerns how management can promote plant and resources productivity. This section defines a set of experiments in order to show how the MANNA architecture is used in management solution development. The management solution must consider the type of WSN (see Section 4.7). In this sense, a WSN application to monitor the air quality is defined as a case study. The air quality monitoring involves various parameters. To simplify, only temperature and carbon monoxide concentration level are monitored using a network composed by approximately 188 sensor nodes. The number of nodes used in these experiments is limited by computational requirements of the simulation tool. To run the simulations, a Pentium IV 1.4 GHz computer with 1 GB of RAM memory was used. It took about 23 hours to simulate all scenarios just once, each one having 188 sensor nodes on average and 125 seconds of network lifetime. Given that the scenarios are evaluated from 33 to 52 times, it took 764 hours (32 days) to run the whole experiment with the computer dedicated only to these simulations. Due to these computational constraints, it was difficult to perform simulations with a larger number of nodes or with networks with longer lifetimes.

The parameters of a real sensor node – the Mica-Motes [1] are used in the simulation scenarios. Regarding the propagation of information from nodes to observer, the WSN defined as a case study is continuous (see Section 3.3.2), that is, data is continuously sensed, processed, and sent to the observer. The cost of sending data continuously may lead to a more rapid consumption of scarce network resources and, thus, shorten its lifetime. This is an important kind of WSN used in different applications (see Section 2.4) and the results show that the use of some management services proposed by MANNA (see Section 4.3) can improve the performance metrics depending on the WSN organization and composition.

As mentioned in Section 3.3.3.3, in WSNs management there is a trade-off to be considered: the highest the number of managed parameters, the highest the energy consumption and the lowest the network lifetime. On the other hand, if parameter values are not obtained, it may be not possible to manage the network appropriately. The WSN management challenge is to perform its tasks without adversely consuming network resources. Figure 5.1 illustrates the trade-off among energy consumption, latency (delay), and quality (message delivery, coverage area, accuracy, and so on).

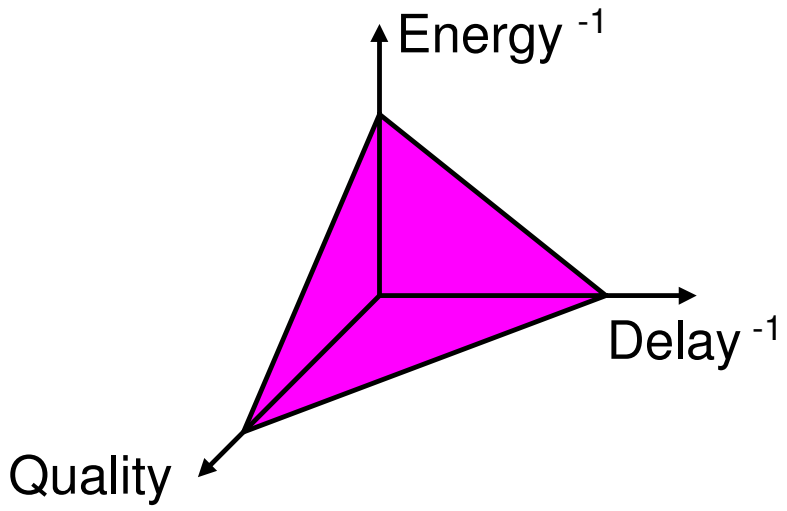


Figure 5.1: Management trade-off.

The following sections present the simulation approach and information, functional, and physical architectures developed as a case study of the use of the framework proposed in this thesis. For each configuration defined, a management solution is discussed.

Simulations are performed to show how the management solution can promote the network productivity and to evaluate different configurations in terms of the organization and composition, varying the initial number of nodes. Thus, the experiments are conducted in order to:

1. Evaluate the impact of the network configuration over the WSN performance and service;

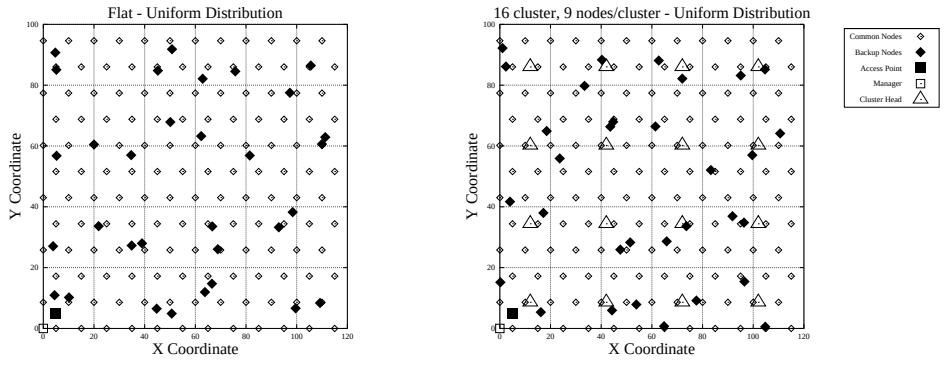
2. Evaluate the impact of the management application on the WSN defined as a case study.

In this sense, the next section describes the scenarios defined in simulation context.

5.1.1 Simulation Approach

In order to evaluate the impact of the network configuration and the introduction of management in different WSN organizations (flat, hierarchical, homogeneous, and heterogeneous), experiments were conducted based on a distinct set of simulation scenarios (see Table 5.1). The metrics used to analyse the behavior of the network are energy consumption, delay and message loss. In Table 5.1, the letters "CH" means "Cluster-Head" and "CN" means "Common-Nodes" in cluster. This table also presents the scenarios where the best access point location is evaluated when the WSN does continuous data collecting and dissemination. Column "AP local" shows two simulated options which are the access point in the middle (see Figure 5.3(B)) and in the perimeter of the monitoring area (see Figure 5.3(A)).

The scenarios were simulated using MANNASim, a framework developed in the context of this thesis, described in Section 5.2. Each simulation was ran for 125 seconds and repeated at least 33 times. In the evaluated scenarios, the following variables were used:



(a) Scenario 1 and 2 (b) Scenario 5, 6, 9, and 10

Figure 5.2: Nodes distribution in the scenarios flat and hierarchical WSNs.

- **Network.** It has 180 common-nodes (in average), uniformly distributed (see Figure 5.2) upon the monitored area (115 × 95 m). Besides common-nodes, there are cluster-head nodes, and only one access point (sink node in flat networks and base station in hierarchical networks). The network is dense (containing 20% of redundant nodes). In scenarios 5, 6, 7, 8, 9 and 10 the network is organized in 16 clusters. Each cluster has a cluster-head and 9 common-nodes. The protocols used are: IEEE 802.11 [41], AODV (Ad hoc On Demand Distance Vector routing) [73], UDP [108] and MannaNMP[2] (MANNA Network Management Protocol).

- **Nodes.** The initial energy of each common-node is 1 Joule. Communication range is 15 m. Bandwidth is 100 kbps, transmission energy consumption is 0.036 Joule/s, reception energy consumption is 0.0054 Joule/s, energy consumption in sleep mode is 0.000003 Joule/s. Energy consumption in processing in active state is 0.0165 Joule/s, in idle state is 0.0048 Joule/s, and in sleep state is 0.00006 Joule/s. These parameters are based on Mica-Motes characteristics [1]. In heterogeneous scenarios, the cluster-head was simulated using WINS [6] parameters (communication range is 140 m, transmission energy consumption is 1.176 Joule/s, reception energy consumption is 0.588 Joule/s, energy consumption in sleep mode 0.001 Joules/s, energy consumption in processing in active mode is 0.300 Joule/s, in sleep mode is 0.0008 Joule/s, and in idle mode is 0.040 Joule/s. The initial energy of each cluster-head is 20 Joules).

Figure 5.2 illustrates the distribution of the nodes in the cases where the access point is localized in perimeter of the network.

The configuration management (in terms of the sensor node capabilities, number of sensor nodes, density, distribution, self-organization, and data dissemination) plays a significant role in determining the performance of the network. As mentioned in Section 3.3.3, in WSNs all operational, administrative and maintenance characteristics of the network elements, the network, the services, and business, and the adequate performance in the

[2]MannaNMP: MANNA Network Management Protocol. A simple protocol created at the application level.

Table 5.1: Simulation scenarios.

Scenario	Organization	Composition	Clusters	AP local	Management
1	Flat	Homogeneous	no	Perimeter	yes
2	Flat	Homogeneous	no	Perimeter	no
3	Flat	Homogeneous	no	Center	yes
4	Flat	Homogeneous	no	Center	no
5	Hierarchical	Homogeneous	16CH/9CN	Perimeter	yes
6	Hierarchical	Homogeneous	16CH/9CN	Perimeter	no
7	Hierarchical	Homogeneous	16CH/9CN	Center	yes
8	Hierarchical	Homogeneous	16CH/9CN	Center	no
9	Hierarchical	Heterogeneous	16CH/9CN	Perimeter	yes
10	Hierarchical	Heterogeneous	16CH/9CN	Perimeter	no

activities of configuration, sensing, processing, communication, and maintenance are dependent on the configuration of the WSN (see Figure 3.3). The definition of the scenarios presented in Table 5.1 is motivated by this dependence. The evaluation of the impact of management on a WSN application, also enables the verification of the influence of the configuration in other areas and functionalities.

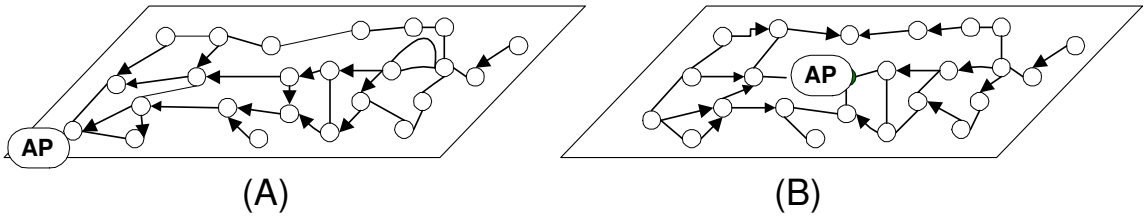


Figure 5.3: Example of access point location.

5.1.2 Information Architecture

In Section 4.4 a generic information model was defined. In the experiments here presented, some managed object classes of the model are used: network, managed element, equipment, WSN observer, phenomenon, WSN context management and WSN goals. Some WSN models defined by MANNA were chosen to represent network states. Some maps (WSN

models) are used such as network topology, residual energy, sensing coverage area map, communication coverage area map, cost map, and production map. In the experiments we use a naive approach to build the maps. The management application takes advantages of characteristics of continuous network to update these maps. As the network continuously sends data to the access point, the management application uses this data to infer the network state. The information model provided by the MANNA architecture is described in [83].

5.1.3 Functional Architecture

As mentioned in Section 4.5, the functional architecture describes the distribution of management functionalities of the network by identifying the network places where managers and agents can be implemented and which management functions will be executed. The management functions chosen are joined into services of a management application. In the simulated scenarios of this chapter, a manager is located outside of the WSN and could be remotely connected. In the functional architectures of the scenarios 1, 3, 5, 7, and 9, the management application has a global network view and can process algorithms that would be impossible to be carried out by the processors of the Mica-Motes [1] common-nodes. In the following sections, a management solution as well as the the location of the agents are discussed for each proposed scenario.

5.1.3.1 Managing Homogeneous Flat Continuous WSN

The flat WSN (scenarios 1 to 4) has one sink node as the network access point. In Section 4.5.2, options about the location of the agents are discussed. In these experiments, the sink node implements the agent that should be able to answer the management operations and access the network resources. This functional architecture is illustrated in Figure 4.7(b). The agent is implemented in the network level and may have some intelligence.

In the homogeneous flat network without management (scenarios 2 and 4), the nodes

collect temperature and carbon monoxide level data and send them to the sink node using a SENSOR-REPORT message. In this case, a SENSOR-REPORT message contains the node identification, carbon monoxide concentration level data, temperature data, and collect time. Redundant nodes produce redundant data generating traffic, leading to collisions, message loss and energy waste.

When management functionalities are used in the network, initially the nodes discover their location (self-discovery management service), collect data just described and send them to the sink node using a SENSOR-REPORT message. In the initial phase of the network, the SENSOR-REPORT also contains management data such as the node location and the operational state (energy level) information (self-knowledge management service). The management application receives this information and produces the topology, energy, and coverage area maps. The manager performs automatically the coverage area maintenance management service which will be described in the next section. This service uses management functions to calculate the network density and to identify redundant nodes. This management service uses other functions to put the redundant nodes administratively out of service. These nodes are called backups. The backup nodes ask to come back to the network when the "sleep time" is over. The active nodes send a TRAP message (described in Section 5.1.4) when the operational state (residual energy) becomes critical. Some of management functions performed are: environment requirements acquisition function, monitored area definition function, environment monitoring function, node definition function, number of nodes definition function, node deployment definition function, network operating parameters configuration function, node deployment function, topology map generation function, network connectivity discovery function, energy map generation function, nodes density calculation function, node operating state control function, node administrative state control function, and so on.

In some cases, the management can also give priority to the multi-hop communication over the sensing in order to avoid that the nodes close to the sink leave the network and stop the retransmission of information coming from more distant nodes (see Figure 2.3). For this, the management configures the nodes so they turn off their sensor device and

only relay data, when their residual energy reach a certain level (see Figure 5.7). In the context of management, this means service negotiation to maintain minimal requirements of the QoS sensing, QoS processing and QoS dissemination. This service is explained in Section 5.1.3.4.

5.1.3.2 Managing Homogeneous Hierarchical Continuous WSN

In homogeneous hierarchical networks all nodes have the same hardware capacity and they need to self-organize into clusters in the initial phase by electing a leader called cluster-head. The cluster-heads receive data from common-nodes of their clusters and send it to the access point. In this case, a base station is the access point. In the functional architecture defined for this network, the agents run in cluster-heads. This functional architecture is illustrated in Figure 4.8(a). The self-organization of this kind of network is complex. Algorithms for leader election are needed so that they contemplate the election of new leaders following different criteria, as mentioned in Section 3.3.2.1. The network performance depends on the leader position in relation to the common-nodes of its group and the base station. The communication scheme between leaders and base station is also relevant (single-hop or multi-hop). The leader election algorithm must guarantee a good leader distribution (physical location) and a balanced number of common-nodes per group. The LEACH algorithm proposed in [35] does not guarantee this two basic conditions. Several leaders can be elected in the same region and groups with only two nodes can occur. Furthermore, self-organized localized algorithms may consume a lot of energy. For that reason, in this work, an indirect election is proposed, where the management application, that has a global network view, elects the leaders. The agents are installed in cluster-heads but when new elections happen, strategies are defined to move the agents and their data. Besides of the management functions used in homogeneous flat WSN (see Section 5.1.3.1), a management solution for homogeneous hierarchical WSN includes leader election function, invitation to form cluster, listening for invitation, discovery function, response to invitation, cooperation discovery function, and so on. In homogeneous hierarchical networks, to have

a single-hop communication between the leader and the BS it is necessary that the elected leader changes its radio configuration, increasing its range and, consequently, the energy consumption. Since cluster-heads consume more energy, they have a shorter lifetime than common-nodes. Thus, new leader elections are necessary, which consume more energy. Supposing that the initial organization phase was successfully performed, let us analyze the leader behavior. The leader receives data from its group, performs some processing and sends the result to the BS. If the communication with the BS is multi-hop, all nodes in the path will be affected by the traffic. If nodes in that path leave the network for any reason, the information may not be delivered. When information coming from the leader is lost because of either congestion, collision or route absence, all information of a whole group is lost. The same is true about the agents which run in cluster-heads. Thus, a natural question is why use this kind of organization if it does not have good perspectives about energy consumption. One reason is to decrease the traffic using some correlation scheme (e.g., data fusion), replacing all the common-nodes messages by one single message composed by the relevant information processed by the leader. The point is, how can the observer know whether the average temperature in the region corresponds to the whole monitored region? How does lost information of a whole group affect the results? How would it be possible to know and manage those situations if there is no management functions and services? Observing the simulation results, an answer to the following question is expected: Considering a continuous WSN, in what aspects can the hierarchical organization be better than the flat organization? To this kind of network the best location for the access point (AP) is also observed considering two options: access point in center and access point in perimeter of the network (see Figure 5.3). The service negotiation to maintain minimal requirements of QoS sensing, QoS processing, and QoS dissemination is very complex for this network.

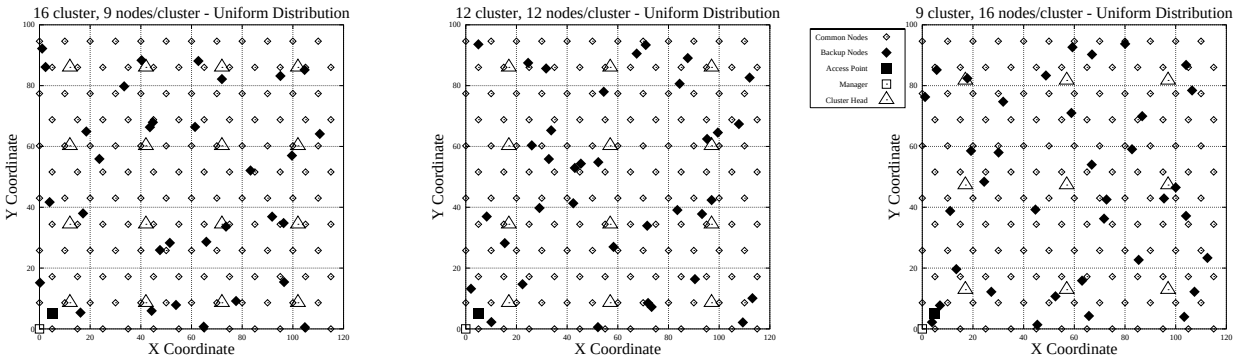
5.1.3.3 Managing in Heterogeneous Hierarchical Continuous WSN

A heterogeneous hierarchical network has some nodes with greater capacity, which become the leaders of groups through all the network lifetime. With the exception of the leader election function, all functions used in Section 5.1.3.2 can be used in this scenario. In general, the leaders invite common-nodes to participate of their groups. The common-nodes accept the invitation and start to send their information to that leader, which can process the information and then send it to the access point. The cost-benefit relation of this organization can be discussed considering the following questions: What is the cost of building a network with more powerful nodes? How to get a good leader distribution in the network deployment? How to guarantee that the leaders fall in the planed places? What should be done if some leader leaves the network because of any problem rather than energy problems? What is the best organization in terms of number of nodes per group? How many leaders should be launched? To answer some of these questions, three different scenarios (see Table 5.2) for heterogeneous hierarchical network are simulated. The common-nodes still are Mica-Motes and the cluster-heads are defined as WINS [6]. The base station is located in the network perimeter, since the cluster-heads can directly transmit (single-hop) to the BS. Besides the management services executed by other network organizations, we have introduced a new configuration service that sets the radio range of a leader according to its distance to the base station. Figure 5.4 illustrates scenarios of the heterogeneous hierarchical WSNs comprised of common-nodes, cluster-heads and a base station. In this case, common-nodes are less powerful than cluster-heads.

The results of the experiments with scenarios presented in Table 5.1 are presented in Section 5.4 and the results of the experiments with different heterogeneous hierarchical WSNs are presented in Section 5.5.

5.1.3.4 Main Management Services Implemented

In this chapter, we develop a WSN application with some different configurations and a corresponding management application for each one. We used the three dimensions to



(a) Scenario A and B (b) Scenario C and D (c) Scenario E and F

Figure 5.4: Scenarios of heterogeneous hierarchical WSNs.

Table 5.2: Heterogeneous hierarchical network scenarios.

Scenario	**Number of Leaders**	**Nodes per Group**	**Backup Nodes per Group**	**Management**
A	16	9	2	Yes
B	16	9	2	No
C	12	12	3	Yes
D	12	12	3	No
E	9	16	4	Yes
F	9	16	4	No

choose the management functions and identify management information to be used. The management functions are combined to form management services. The main management services of MANNAarchitecture performed are:

Network planning. This management service contemplates all the management functions that have to be performed before nodes are deployed in the monitoring area. For example, it calculates the number of necessary nodes to perform the distributed sensing and investigates the best organization (flat or hierarchical, heterogeneous or homogeneous) for the network and the location of the access point.

Coverage area maintenance. This management service performes automatic services of coverage area maintenance, identifying areas of sensing redundancy [103] and

areas not covered. The network proposed as case study is initially dense, that is, there is a high number of nodes per area. A denser network will lead to a more effective sensor network because of the higher accuracy in the network (areas of overlapping sensing, and redundant information) and fault tolerance (see Figure 5.5). On the other hand, this will lead to a larger number of collisions and potentially to congestion in the network, increasing latency and reducing energy efficiency. Congestion control must not only be based on the capacity of the network, but also on the accuracy level required at the observer. The traffic in a WSN is different from conventional networks. It is a collective communication operation with redundancy. The management application has the flexibility to negotiate services, that is, of meeting the performance demands by controlling the reporting rate of the sensors, controlling the virtual topology of the network (by scheduling of sensors), or optimizing the collective reduction communication operation (by data aggregation). In this sense, the self-managed WSN performs service negotiation using some service qualifiers defined in Section 3.3.1.2. In some applications, besides the information about some feature of the phenomenon, it might be necessary to know where (sensor location), when (data–time) and how (sensor calibration and exposure) to manage the WSN performance. In these experiments, the coverage area map is used to recognize overlapping sensing and uncovered area and when used together with production map it is possible to determine where, when, and how the data was produced. Figure 5.5(A) illustrates the coverage area of the nodes and the overlapping areas are identified as is illustrated in Figure 5.5(B).

The network density control management function depends on the application. Within the scope of this experiment, the management service identifies the redundant nodes and turns them off for a period of time, that is, the node is out of service by administrative decisions. When the main nodes become inactive, generating sparse areas, the service tries to turn the backup nodes on, if there is any. This management service uses management functions that identify redundant sensor nodes. Figure 5.6 shows that the sensor node “A” is redundant and it can be turned off in order to do density control.

QoS monitoring. WSNs are self-service networks, since they produce, process and deliver their own information. The main services of a WSN are sensing, processing and

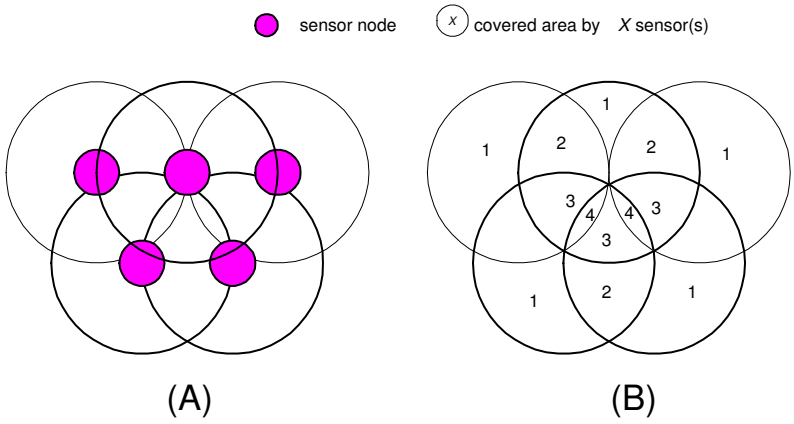


Figure 5.5: An example of covered area in dense WSN.

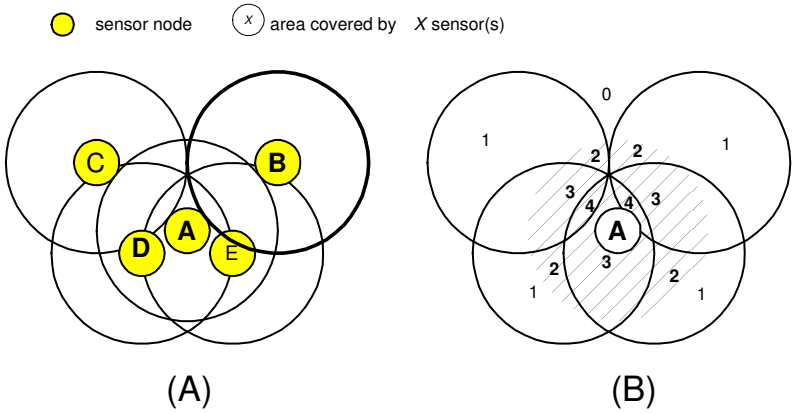


Figure 5.6: Backup nodes: node A is redundant.

data dissemination. As mentioned in Section 3.3.1.2, the dissemination quality can be characterized by latency, delay and by the number of lost messages. The network service quality also must be determined by the energy consumption, i.e., the amount of energy consumed by the performance of certain services with some quality level. In the management application defined in these experiments, the QoS is monitored using the coverage area, delay, lost messages and energy consumption metrics. Accuracy could be used when the observer needs to know how many nodes participated in the construction of those values and what is the percentage of the area covered by the network in that moment.

Network operation parameters configuration. This management service uses management functions to change some parameters of the network elements depending on

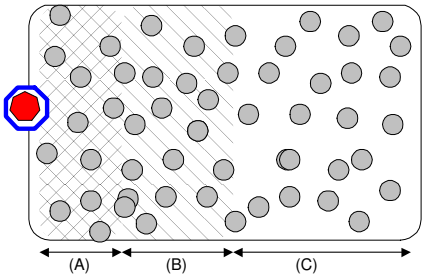


Figure 5.7: An scheme of configuration of nodes per area.

the network state. In the simulated scenarios in these experiments, this function was used in flat networks to establish a trade-off between the sensing and the communication. Nodes near the sink are programmed to turn the sensor off when they reach a given value of the residual energy level. In this case, the nodes stop sensing and disseminating their own information, and just relay information from distant nodes.

Figure 5.7 illustrates a scheme of configuration of nodes as a function of distance between them and the sink node. For example, sensor nodes in region (A) of Figure 5.7 stop sensing when they have 8% of residual energy and keep only relaying the information that comes from other nodes. Sensor nodes in region (B) of the Figure 5.7 stop in 3% and nodes in other regions keep sensing until their energy is not finished. In homogeneous hierarchical networks, a scheme like this was also implemented. In heterogeneous hierarchical network, the configuration service changes the transmission power of the group leaders considering their distances from the base station. However, the hardware of the node should allow such configuration, as in the WINS [6] nodes. In this kind of network, the communication between cluster-heads and the base station is single-hop. The management service decreases the communication range of the leaders that are closer to the sink, reducing the area of interference and the energy consumption.

Other management services used but not described in this section are: placement, self-discovery, and self-knowledge, among others.

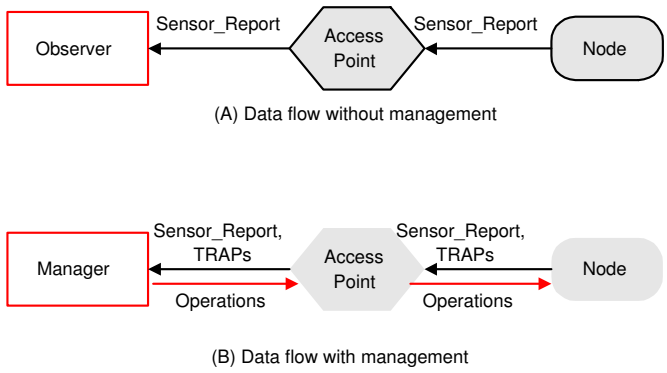


Figure 5.8: Information flow across a WSN.

5.1.4 Physical Architecture

Both reliable data delivery and the naming scheme (addressing) are open issues in the WSN context. The protocols proposed for WSN described in Section 2.5 are not applied in these cases and no appropriate routing and media access control algorithms for continuous WSNs are available yet. Thus, the protocol stack used in this work is comprised of UDP and AODV, and IEEE 802.11. Furthermore, the communication between common-nodes and cluster-heads uses a lightweight protocol called MNMP, which has been designed in the context of this thesis. Other algorithms are being evaluated and modifications are being proposed in a way to develop the physical architecture of the MANNA architecture. The interfaces between management entities must use a protocol stack that is lightweight and adequate to the type of the WSN. As mentioned in Section 4.6, the MANNA physical architecture does not define a protocol stack to those interfaces but provides protocol profiles adequate to each different application. The main management operations defined by MNMP used in these experiments are presented in the following:

- **SET**. Operation used to set or change values of attributes of the managed objects. In these experiments, the manager uses this operation to change the administrative state of the nodes. For example, when the density control management function discovers redundant nodes in dense areas, the manager turns them out of service, by changing their administrative state and attributing a period of time for them to stay

with their radios off.

- **TRAP-DELETE**. Operation used to notify events. In these experiments, when a node reaches the critical level of residual energy, it sends a TRAP-DELETE to warn the management application. The management application updates the energy map, updates the coverage area map and activates the backup nodes, if there are any. The management treats the uncovered area problem, decreasing the loss and using the calculated redundancy to extend the network lifetime but keeping the QoS.

- **TRAP-CREATE**. Operation used to notify object creation. In these experiments, the management finds and turns off the network redundant nodes using the initial coverage area map. In these cases the radio is completely off, and thus, the management application has to assign a period for the nodes to wake up and ask to come back to service. The node performs this function by sending a TRAP-CREATE.

- **GET**. Operation used to obtain attribute values. This operation is not used in these experiments.

In the scenarios without management defined in the experiments, the application flow is always unidirectional, from source nodes to an access point (see Figure 5.8(A)). In scenarios with management, the flow is bidirectional, the SENSOR-REPORT and TRAPS come from the nodes and management entities to the management application and the SET operations go from management application to agents (see Figure 5.8(B)). SENSOR-REPORT is the application message; TRAPs and SETs are management messages.

To perform these experiments of this thesis, a simulation tool has been developed. The next section presents the main characteristics of this tool and the motivation to develop it.

5.2 MANNASim Framework

This section introduces the ongoing efforts in the development of MANNASim, a simulation framework that introduces new modules for design, development and analysis of different

WSN applications and management applications. MANNASim inherits the core features of the Network Simulator (NS-2) [94], version 2.26, and builds up new features that include ability to use different protocol profiles for different WSN applications.

The goal is to develop a detailed simulation framework, which can accurately model different sensor nodes and applications while providing a versatile testbed for algorithms and protocols. The numerous challenges make the study of real deployed sensor networks very difficult and financially infeasible. At the current stage of the technology, a practical way to study WSNs is through simulations that can provide a meaningful perspective of the behavior and performance of various algorithms.

Some simulation tools were studied but despite their effectiveness, these tools are currently not equipped for capturing all the aspects of interest in WSNs and management. Park et al. [70] created a simulation framework for sensor networks called SensorSim. They created two types of models. The first one is the sensor function model which represents the software abstract of a software (protocol stack, middleware, user application, and the sensor protocol stack). The second type of model is the power model which simulates the hardware abstract (CPU, radio, sensors). The simulated micro WSNs consists of 7 nodes randomly placed of a flat terrain for a specific application (movement of tanks in military application) and the code is not completely available. Existing simulators for wireless networks are Opnet, NS-2, Parsec, and SSF but no one presents specific modules for WSNs.

Network Simulator – NS-2. Network Simulator is an event driven, object oriented simulator that enables the simulation of a variety of IP networks. It supports, among other features, network protocols such as TCP and UDP, unicast, multicast, hierarchical routing, queue management mechanisms such as CBQ, RED and Drop Tail, traffic generators and mobility. *NS* is written in C++ language with an OTcl interpreter as a front-end. It supports two class hierarchies, one in C++ language (also called the compiled hierarchy), and similar within the OTcl interpreter (called the interpreted hierarchy). These hierarchies are closely related to each other, which means that every C++ class has a correspondent

in OTcl. All compiled objects are made available to the OTcl interpreter through a linkage that creates a matching interpreted object for each compiled object created. As a general rule, OTcl is used to describe the network topology being simulated, and to dynamically configure components during simulation. Packet processing, event scheduling and basic network components are written and compiled in C++. To create simulation scenarios in OTcl scripts, the steps below must be followed: setting the options for simulator configuration, creating a simulator instance, setting up the topology, setting up the traffic, setting up movement of other dynamic changes during the simulation, tracing the events [94].

MANNASim Overview. The simulator tool MANNASim was built considering the object class defined at the generic information model proposed by the MANNA information architecture (see Section 4.4). As well as in the establishment of the novel management dimension, many WSNs applications were investigated in the direction of extracting a generic simulation model that allows to carry out experiments considering different types of WSN applications. MANNASim is comprised of two modules: SIMappli and SIManna. The first module allows the simulation considering different types of applications and does not implement any management functionality. The second one is used when a management solution is necessary to the desired application.

The first step taken in the implementation of the simulator was the implementation of a node specific to WSNs, the sensor node. Since NS-2 already possesses an object class that represents a mobile node with wireless communication capability, the new node was implemented extending the mobile nodes class. To this new node, new characteristics were added such as sensing and processing energy consumption, “wake up” and “sleep” functions, self-test performance (still to be implemented) and control of components usage state such as sensor devices and processor. A subclass of the existing energy model [94] was also created; it implements a battery class that can be used to implement the different existing battery models. Next, specialized classes that describe the behavior of each node type found in a WSN were modelled and implemented. These behaviors were implemented in the application layer, since no restriction may be imposed to the user regarding the

desired protocol stack. Thus, each developed class that models a node from MANNASim inherits from NS's application class [94]. Common-nodes, leader nodes and access points were created also.

To create simulation scenarios using MANNASim, the user must set up the desired parameters of the sensor nodes, and then create node instances and their applications using OTcl language. Some of the parameters that may be configured are the sensing and dissemination types. There are no restrictions regarding the scenarios configuration that may present different compositions, organizations, hierarchical levels, number of nodes, number of access points, and so on. Each common-node may have several data generators attached to it, one to each parameter measured by the application. These data generators generate new data according to the sensing type and send them to the processing class. The data processing is performed according to the type of the processing class and the results are disseminated according to the chosen dissemination type. NS models the data application format through an abstract class called AppData. MANNASim has a specialized class that models the sensed data, that inherits from AppData and implements specific methods and attributes (see Figure 5.9). The main classes are described below.

-*SensorNode class*: represents the sensor node. It is derived from MobileNode NS'class. The attributes which characterize this class are: app (list of applications), sensingPower (energy consumption in sensing mode), processingPower (energy consumption in processing mode), processorInstructionsPerSecond (number of instructions per second), sensorUseState (usage state of the sensor devices), processorUseState (usage state of processor), transceptorUseState (usage state of transceptor). The methods are: selfTest (algorithm which runs self-test), sleep (algorithm which turns off the node's applications), and wake-up (algorithm which turns on the node's applications).

-*Battery class*: represents the power supply of the node. It is derived from EnergyModel NS'class. The main methods are: DecrSensingEnergy (algorithm which decreases the energy level related to sensing activity), DecrProcessingEnergy (algorithm which decreases the energy level related to processing activity), setNodeOn (set the "node-on" boolean variable to true), and setNodeOff (set the "node-on" boolean variable to true).

-*SensorBaseApp class*: represents the type of the applications for common-nodes, cluster-heads, managed nodes. It is derived from Application NS'class. The attributes are: gen (list of all data generators), info (list which contains all collected data), sensor-node (represents the node which is connected to the application), disseminating-type (type of dissemination of the application), disseminating-interval (represents the interval to data dissemination), destination-id (application messages destination address), dissTimer (event scheduler). The methods are: start, stop, disseminateData, processSensedData, insertData (puts data in info), insertNewGenerator (puts data generator in gen), and getExpireTime (indicates the disseminating-interval value from the disseminating-type).

-*ClusterHeadApp class*: represents the leader of group application. It is derived from SensorBaseApp class. The main attribute is child-list (list of nodes which belong to the group). The main methods are: process-data(in charge of to receive messages), insert-child (inserts a new child in child-list), remove-child (delete a child of child-list), and search-child (search a child in child-list).

-*CommonNodeApp class*: represents the common-nodes application which performs data dissemination using the disseminateData method, processing using processSensed-Data method and other functions using CommonNodeApp methods. It is derived from SensorBaseApp.

-*AccessPointApp class*: represents the access point application. The outside-network attribute contains the address to exchange information with external entities using forward-data (to send data) and process-data(to receive data).

-*DataGenerator class*: represents the data generators which simulate the sensing task. The attributes are: app (list of applications which are related to data generator), sen-sTimer (timer of the event scheduler), sensing-interval (interval to generate sensing event), sensing-type (type of sensing). The main methods are: insertNewapp (puts new applica-tion in app), start, stop, generateData, getFromNormalDistribution (generates data normal distribution), getExpireTimer (indicates the sensing-interval to scheduler the sensTimer), and insertInterference (inserts interference in data gathering).

-*TemperatureDataGenerator*: is an example of a data generator. Temperature AppData

class represents the data type of this class. The attributes are: avg-measure (average value of temperature) and std-deviation (standard deviation of the value).

-*SensingTimer class*: implements the sensing events scheduler. It is derived from TimerHandler NS's class.

-*DisseminatingTimer class*: implements the processing events scheduler. It is derived from TimerHandler NS's class.

-*TemperatureAppData class*: determines the type of data which is provided by data generator. It is derived from AppData NS's class.

-*SensedData class*: represents sensing applications. It is derived from AppData NS's class. The attributes are: msgType (type of sent message), eventType (type of event), node-id (source node identification), infoRepository (data log). The methods are: insert-NewData (inserts new data in the infoRepository), existsData(used to get inforepository status), and getData(used to get stored data in infoRepository).

5.3 Assumptions

Selecting the correct level of detail (or level of abstraction) for a simulation is a difficult task. Few details can produce simulations that are misleading or incorrect. However adding too many details requires more time to implement, debug, and later change; it slows down simulation and can distract from the research problem at hand. Designing simulations to show how a management solution can be provided from the framework proposed by this thesis and, thus to study the effects or impact of this management solution on WSN application involves making choices of the detail level to be used. Choices about details are difficult for the purpose of this thesis. Low-level details can have a great effect on performance and detailed simulations can be very expensive (computational resources and time simulation). Simulation run-time is adversely affected by details. In this sense, some assumptions have been defined here. The challenge is to identify what level of detail does not affect the answer to the design question at hand. For example, there are no wireless network simulators or other work that consider details of CPU instruction set or

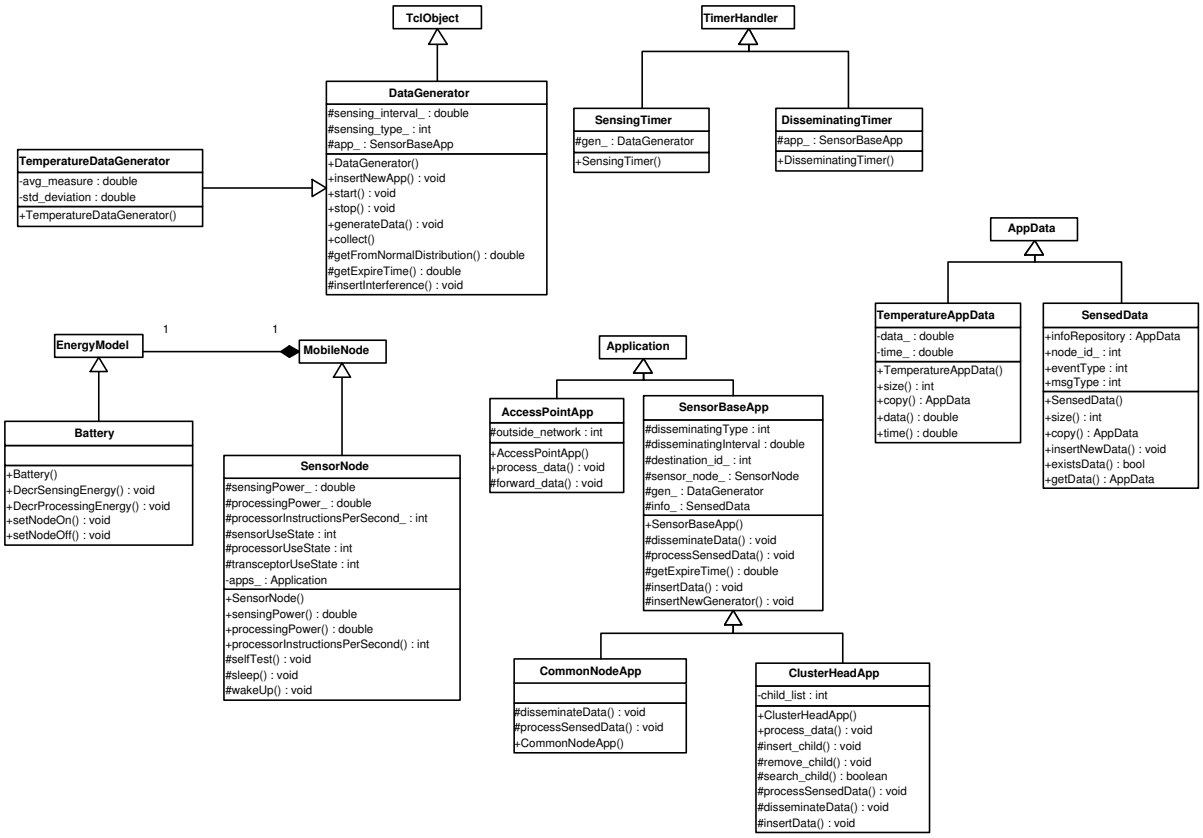


Figure 5.9: Initial class diagram of MANNASim.

memory[70].

The issue in designing WSNs management solution is one of the objectives of this thesis. However, the management solution depends on WSN application. The issues in designing a WSN include: selection of the collaborative signal processing algorithms running at each sensor node, selection of the multi-hop networking algorithms and medium access control algorithms to link, optimal matching of sensor requirements with communication performance, definition about number of nodes to cover the monitored area, type of organization, type of composition, and so on. In military networks, additional issues are resistance to jamming (reliability of data), integrity, privacy, and robustness.

To build the scenarios and perform the experiments some assumptions were done. The main assumptions are described in the following.

- It is assumed that each node knows its own location and nodes are not moving. These are common assumptions for many sensor network applications.

- For simplicity and convenience, the sensing area of a node is a circle with a nominal radius r centered at the location of the node itself. In the scenarios description, the sensor nodes are deployed in a two-dimensional Euclidean plane. However, it is possible to extended to a three-dimensional space without much difficulty.

- It is also assumed that the neighboring nodes should be roughly time synchronized in the order of seconds. The last assumption is that nodes can directly communicate with the neighboring nodes within a radius larger than r (r is nominal sensing radius). This is a typical case in all systems in literature. The chosen protocol works as long as the nodes are able to communicate directly or indirectly with each other within the distance of r.

- To achieve fault tolerance higher densities are required. Thus in contrast to traditional ad hoc based protocols, WSNs protocols need to sustain large number of nodes at high densities.

- Functionalities of the Network Simulator tool are used to promote energy efficiency by allowing sensor nodes or some of its devices to sleep when the management application put them out of service. The lifetime of a sensor network is composed of a configuration phase and a maintenance phase. During the configuration phase, each sensor node finds its own position and sends its location and energy to the management application. After that, active nodes enter into a sensing phase and start to sense environmental events and redundant nodes are turned off to become back up nodes.

- The performance of a WSN depends on the routing of the underlying ad hoc network considering the application feature. As there are no available routing algorithms with the needed requirements we used AODV [73] in our experiments. We performed tests

with DSR [72] whose algorithms include dynamic source routing protocol (DSR) and the AODV. In the DSR algorithm, each packet to be routed carries in its header a complete ordered list of nodes through which the packet must pass. This is a key aspect in the algorithm since intermediate nodes do not need to keep up-to-date routing information. AODV is an example of demand-driven system that eliminate most of the overhead associated with table update in high mobility scenarios. However, it has high energy cost during route setup (path discovery).

- Wireless networking protocols such as 802.11 [41] (used in the physical architecture defined) are not directly applicable to multi-hop sensor networks because they require all nodes to listen at all times. The main sources of energy waste have been identified: collision (when a transmitted packet is corrupted it has to be discarded, leading to the necessity of re-transmissions, which increases energy consumption, collision increases latency as well), overhearing (a node picks up packets that are destined to other nodes), and control packet overhead. Sending and receiving control packets consume energy too, and less useful data packets can be transmitted. The last major source of inefficiency is the idle listening, i.e., listening to receive possible traffic that is not sent. If nothing is sensed, nodes are idle mode most of the time.

- We used centralized management solutions due to simplicity, computational capacity extension, and global vision. The fact that the manager entity is located outside the WSN allows the processing to be done by automatic services without incurring in energy consumption. However, the network can suffer of the implosion problem when all nodes send data in same time. In these experiments, it is used a scheduling scheme to avoid this problem.

- The decisions about nodes deployment are based in [59]. In this work, sensor nodes are placed at the vertices of equally spaced triangular grid. Numerous other placements can be constructed. However, in [59] experiments with triangle based deployment scheme seemed to provide the best level of exposure following the square

scheme. Finding the optimal placements of sensor nodes to guarantee exposure coverage levels is an interesting and challenging problem. For example, more uniform coverage levels may be beneficial, suggesting the use of more uniform sensor deployment schemes such as the triangular and hexagonal deployment schemes. But it cannot expect the sensor field to be deployed in a regular fashion (i.e., array, two-dimensional lattice). More importantly, uniform deployment does not correspond to uniform connectivity owing to unpredictable propagation effects when nodes, and therefore antenna, are close to ground or other surfaces such as obstacles, trees, etc. If we deploy few nodes, the distance between neighboring nodes will be too great and the packet loss rate will increase or the energy required to transmit the data over longer distances will be prohibitive. If we use all deployed nodes simultaneously, the system will be expending unnecessary energy at the best case, and at the worst case the nodes may interfere with one another by congesting the channel. In the process of finding an equilibrium, we are using [103].

- It is assumed that the data rate is one of the major factors that contributes to transceiver power consumption where higher data rate will lead to higher power consumption. Five different modes of operation that the radio may engage are considered: transmit, receive, sleep, idle and off. In transmit mode, a packet is being transmitted and both transceiver and amplifier are operating to process the packet. In receive mode, a packet is actively being received and the amplifier has a different amount of energy consumption most likely less than that of the transmit mode. In idle mode, there is no packet being received but radio monitors the air for any signal. In sleep mode, both the transceiver and amplifier are turned off. Any signal from the air will not be picked up. The off mode is similar to sleep but the radio cannot go back to any other mode [71]. Given the different modes of operation, different power management schemes can be implemented to conserve energy.

- To know the amount of available energy in the network, a naive solution was constructed where each node is programmed to send its energy level to the agent. This

technique has to be designed to gather information about the available energy in each part of a sensor network (in the clusters). In the future, a prediction-based approach to construct the energy map will be used. Basically, each node sends to the agent its available energy and its power consumption. The monitoring node uses this information to update locally the information about the available energy in each sensor node.

- The power usage of the processing will depend mostly on the clock speed and the operation mode of the CPU. One way to measure this power usage is to count the number of clock cycles for different tasks (e.g., route updates, signal processing). A rough estimative of the clock cycles can be assigned to each task. The battery is modelled as a bucket of energy which is drained at a rate equal to the total aggregate power of sensors, processor, and radio.

5.4 Simulation Results

This section presents the results for the performance metrics delay, message loss, energy consumption, and production. In order to investigate the influence of the configuration in terms of the organization (flat and hierarchical), composition (homogeneous and heterogeneous), number of nodes, density, and access point location, all experiments are conducted in scenarios with and without management services.

The fact that the manager entity is located outside the WSN allows the processing to be done by this management function without incurring in energy consumption. The manager implements most of the management intelligence because of its capacity to process the automatic services. The manager has also a global view of the network and can take decisions with much more authority. Recall that we are dealing with a WSN with continuous collection and dissemination. For other applications the management solution designed and the results might be completely different.

5.4.1 Delay

Figure 5.10 presents the average delay in message delivery of the simulated scenarios presented in Table 5.1. In the homogeneous flat networks (scenarios 1 to 4), the lowest delay occurs in scenario 3, the homogenous flat network with access point in the middle of the monitored area that implements the management services described in Section 5.1.3.4. For scenarios 1 and 2 (access point in the perimeter) the delay is higher because of the distance to reach the access point (see Figure 5.3). In scenario 1, the bidirectional flow also contributes to increase the delay (see Figure 5.8). In homogeneous hierarchical network scenarios (5, 6, 7 and 8), the lower delay occurs when the access point is in the network perimeter. The best results obtained of a homogeneous hierarchical networks occurs when the management services are implemented (scenario 5). Comparing the flat networks with the hierarchical homogeneous networks we verify that the delay is higher when there is a group formation with a node of the group being the group leader (scenarios 5 to 8). Comparing the homogeneous networks (flat and hierarchical), the best organization, considering the delay, is scenario 3 which implements the management services and has the access point located in the center of the monitored area. Observing all the delay results, we verify that the best choice is to use a heterogeneous hierarchical network (scenarios 9 and 10). However, we must consider the investment necessary when acquiring higher capacity nodes. We also must consider the cost of positioning the leader in such a way that the network has a uniform distribution. For heterogeneous hierarchical networks (scenarios 9 and 10) the delay is lower in scenario 9 (Table 5.5), which implements the management services. Other heterogeneous hierarchical WSN scenarios are studied (Table 5.2) and the results are presented in following section.

5.4.2 Lost Messages

Figure 5.11 illustrates the average percentage of lost messages for the simulated scenarios. Observing the graph, we can see that the lowest percentage of messages loss occurs in scenario 9 (heterogeneous hierarchical network with management). In homogeneous networks

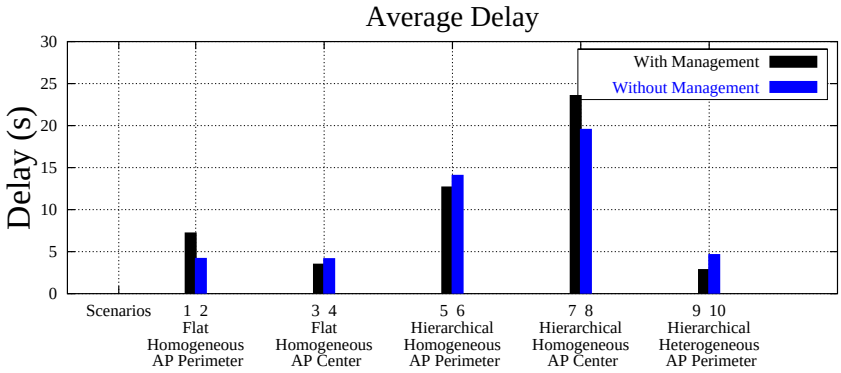


Figure 5.10: Average delay in the proposed scenarios.

(scenarios 1 to 8) the best architecture, considering lost messages, is the one implemented in scenario 3 (homogeneous flat network with management services and access point located in the center of the monitoring area). Comparing scenarios without management, the worst cases occurs in scenarios of the homogeneous hierarchial WSNs (scenario 6 and 8). This behavior is expected because of the characteristics of this type of the network presented in Section 5.1.3.2. Other management services must be implemented for homogeneous hierarchical to try to reduce the message loss. We can observe that management has a positive effect in all simulated scenarios, considering this metric. Since the density is controlled, the congestion and the collision are minimized. The average of lost messages could be reduced even more if specific media access control protocols for this type of WSN were already available. The management service of parameters configuration also influence the performance of management for this metric. As seen, in homogeneous network, the nodes closer to the access point tend to consume more energy, leaving the network before others. The management application configures the nodes closer to the access point to privilege the dissemination when its residual energy reaches 8% of its total capacity. Thus, nodes far from the access point can find a path to deliver their data. The hierarchical homogeneous network looses more messages because of the flow of messages when the manager indicates the leaders of the groups, the group formation and the fact that the leaders aggregate the messages of all common-nodes in the group making a larger message. Regarding lost messages, management has shown to be productive since it manages the

redundant nodes and privileges the communication in nodes closer to the access point.

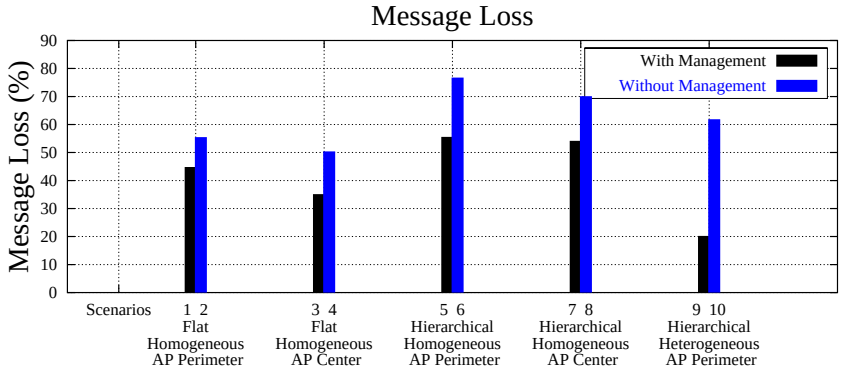


Figure 5.11: Message loss in the proposed scenarios.

5.4.3 Energy

Figure 5.12 shows the average energy consumption considering common-nodes and cluster-head energy consumption. The graph shows that the energy consumed by management services is not significant. Regarding the access point location, there is no consumption difference when management functions are implemented. The average energy consumption in scenarios of homogeneous network with and without (scenario 1 to 8) management are similar. A difference can be noticed between the comparison of homogeneous hierarchical networks (scenarios 1 to 8) and heterogeneous hierarchical networks (scenarios 9 and 10). The heterogeneous networks consume more energy since they have leaders with a higher radio range and transmit the information in a single-hop to the base station. The management improves the productivity of heterogeneous hierarchical networks (scenario 9) since it controls the common-nodes density in the groups and also configures the radio range of the leaders in relation to the distance of the base station (as observed for delay and message loss metric).

The majority of the management services executed are automatic, that is, without human interference. These results show that, besides the bidirectional flow, management can improve the productivity, decreasing the average delay in some scenarios (3, 5 and 9) and minimizing the average amount of lost messages in all scenarios (1, 3, 5, 7 and

9) without incurring in negative influences in energy consumption. Another advantage of the management is the accompaniment of the quality of service, represented by the management of data production.

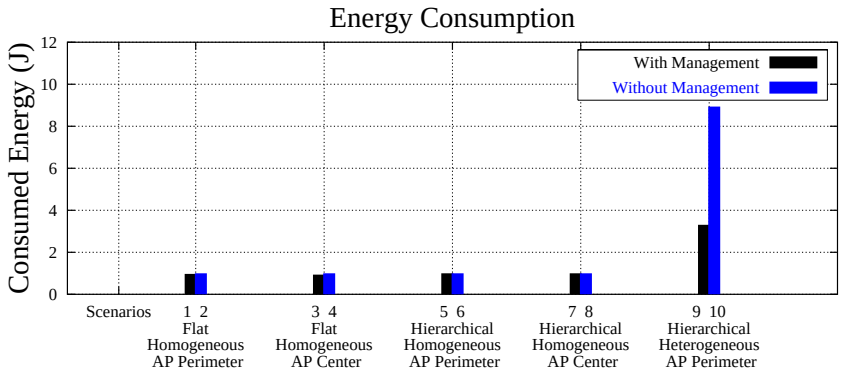


Figure 5.12: Energy consumption in the proposed scenarios.

5.4.4 Production

Considering the service aspects of WSNs, we evaluate the production defined as the number of nodes that produce and that achieve a well succeed information dissemination. Observing Table 5.3, we note that at 13 seconds of simulation, 77% of the nodes of scenario 1 deliver their information to the observer, that is, the temperature and carbon monoxide level averages that were produced by 77% of the nodes. The remaining 23% had their data lost or delayed. Between 13 and 23 seconds of the simulation, the management application generates a bidirectional flow inside the network. The effect of this traffic can be noticed at instant 23 of scenario 3 and at instant 33 of scenario 1, when the percentage of nodes that can deliver their information decreases to 63% and 74%, respectively. Following, at 53 seconds of simulation, the nodes start to leave the network due to energy problems. The network of the scenario 1 (with the access point in the perimeter) stops its activities at 63 seconds of simulation. The network of the scenario 3 stops its activities at 83 seconds of simulation. Therefore, the access point has a better location in the center when talking about flat networks.

The information in Table 5.3 and Table 5.4 is available to the manager. In scenarios

Table 5.3: Number of nodes producing in the homogeneous flat WSN.

Scenario	**13s**	**23s**	**33s**	**43s**	**53s**	**63s**	**73s**	**83s**
1	77%	80%	74%	88%	71%	0	0	0
3	78%	63%	69%	71%	71%	68%	17%	0

without management, the observer can not know how, when and were the data is produced. Observing Table 5.4, we note that the heterogeneous hierarchical network (scenario 9) has a greater production time. On the other hand, the homogeneous hierarchical networks (scenarios 5 and 7) stop producing even before the flat networks. At 52 seconds of simulation, the homogeneous hierarchical network with access point located in the middle of the network is out of service. At 77 seconds of simulation, the hierarchical homogeneous network with access point in the perimeter stops its activities. At 102 seconds of simulation, the heterogeneous hierarchical network is still producing, with 85% of the nodes taking part in the temperature and carbon monoxide level averages construction. Because the network productivity analysis is a service offered by the management, this information is not available to scenarios 2, 4, 6, 8 and 10.

Table 5.4: Number of nodes producing in the hierarchical homogeneous and heterogeneous WSN.

Scenario	**27s**	**52s**	**77s**	**102s**
5	18%	25%	0	0
7	7%	0	0	0
9	53%	93%	97%	85%

Regarding energy consumption, this metric is similar for all scenarios of homogeneous WSNs. However, the lifetime is different.

5.5 Results of Different Heterogeneous Hierarchical Scenarios

To answer the questions proposed in the Section 5.1.3.3, we have simulated different heterogeneous hierarchical network scenarios, considering three different groups (cluster) sizes. Table 5.2 presents the six simulated scenarios. In this section we evaluate the use of management and different configuration effects (number of nodes per cluster, number of leaders per network and number of backup nodes) considering the following metrics: delay, message loss, and energy consumption.

Delay for different configurations of a heterogeneous hierarchical network. Observing the graph of Figure 5.13, we notice that scenario F (heterogeneous hierarchical network organized in 9 groups with 16 common-nodes per group and 4 backup nodes) has the smallest average delay. Nevertheless, the average delay does not differ much from scenario E (with management). This result shows that the network with 9 leaders and 16 common-nodes has a better result towards the delay metric. When we compare all scenarios, we see that in most cases the management reduces the delay.

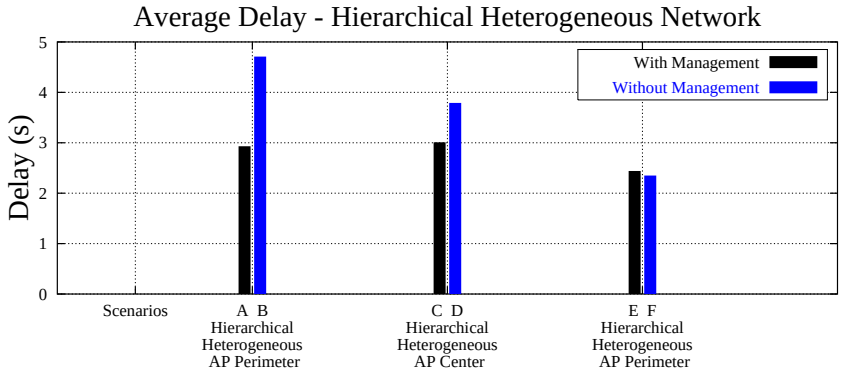


Figure 5.13: Average delay in heterogeneous hierarchical scenarios.

Average number of lost messages in different configurations of heterogeneous hierarchical networks. Observing the graph of Figure 5.14, we verify that management reduces the number of lost messages and that the best scenarios with management are A and E. The best scenario without management is F. Observing the graph, the worst case

among the scenarios without management to the message loss metric is scenario B.

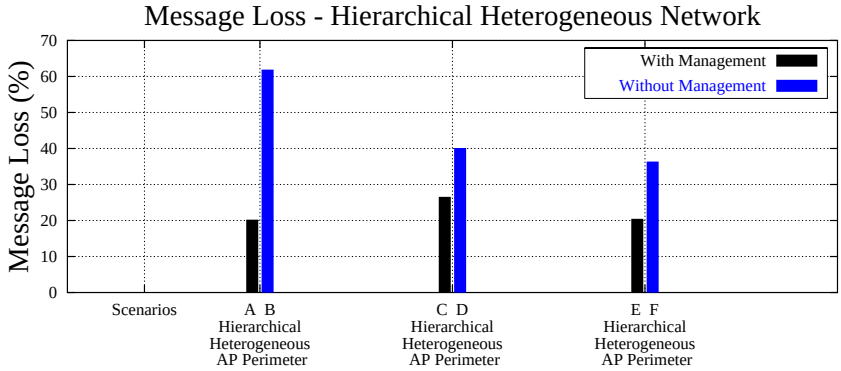


Figure 5.14: Message loss in heterogeneous hierarchical scenarios.

Energy consumption in different configurations of a heterogeneous hierarchical network. Observing the graph of Figure 5.15, we verify that the average energy consumption of the common-nodes is almost the same for different organizations with and without management. In this graph, only the energy of common-nodes is considered. The management contributes to the energy economy metric when we observe Figure 5.16, that presents the energy consumption of the leader nodes. The management service that reconfigures radio range according to the distance from the BS, contributes to energy consumption decrease. The graph of Figure 5.17 shows the energy economy achieved with the inclusion of the management services (Section 5.1.3.4).

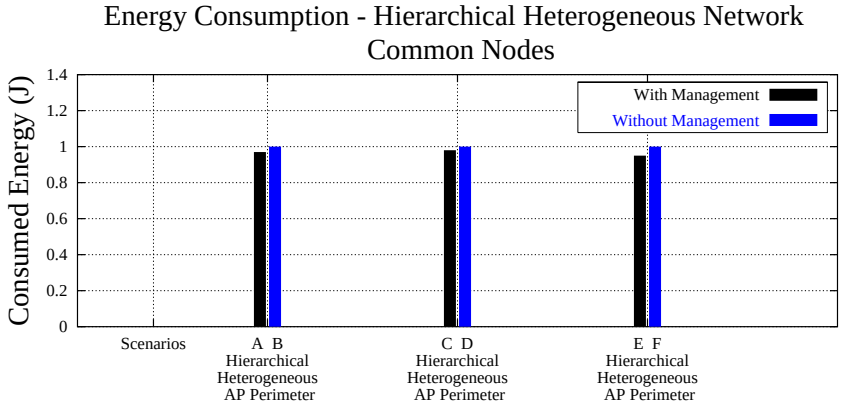


Figure 5.15: Energy consumption in common-nodes in heterogeneous hierarchical scenarios.

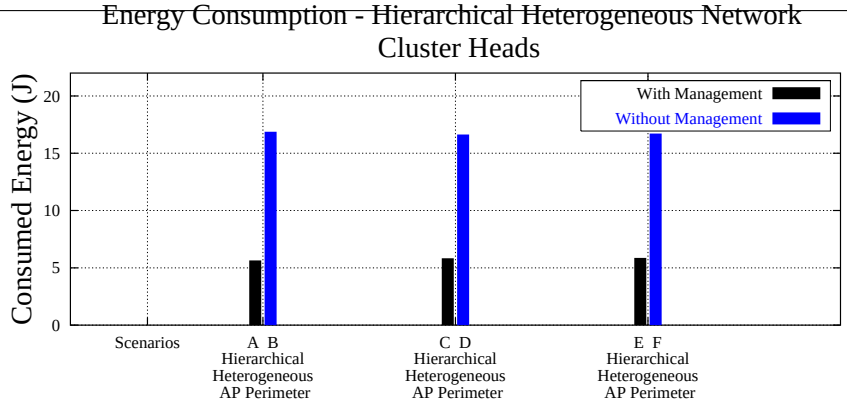


Figure 5.16: Energy consumption in cluster-heads in heterogeneous hierarchical scenarios.

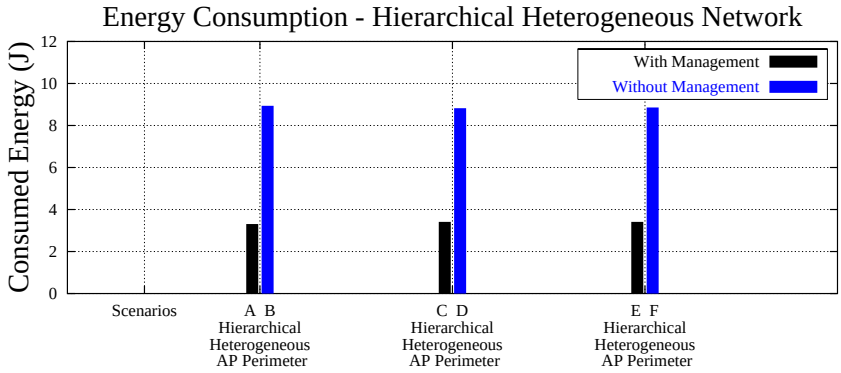


Figure 5.17: Energy consumption in heterogeneous hierarchical scenarios.

5.5.1 Some Considerations about the Results

Observing Table 5.5, we can notice that the best scenario for continuous WSNs is 9A—hierarchical heterogeneous network with 16 leaders, 9 common-nodes, 2 backup nodes per group and implementing management services and functions. However there is a cost related to this kind of network, that is, nodes with greater hardware capacity have higher cost. Scenario 9A is the best in terms of number of lost messages and energy consumption. Regarding the delay metric, the best architectures are the ones in scenarios 9E and 10F, when comparing all scenarios (Tables 5.1 and 5.2). Considering only the homogeneous networks scenarios (scenarios 1 through 8), the best configuration is achieved by the homogeneous flat network with access point located in the middle of the network (scenario 3). Scenario 7 holds the worst configuration in terms of delay. For energy consumption, practically all homogeneous networks scenarios have the same consumption. Recalling the results we had obtained in terms of production (Tables 5.3 and 5.4), we verify that the

Table 5.5: Summary of results.

Scenario	Avg. Delay (s)	Std. Deviation	Avg. Msg Loss (%)	Std. Deviation	Avg. Energy Consumption (J)	Std. Deviation
1	7.29	0.45	44.85	2.47	0.97	0.01
2	4.26	0.36	55.49	3.06	1	0.00
3	3.57	0.39	35.15	2.55	0.94	0.02
4	4.23	0.41	50.38	2.24	1	0.02
5	12.76	0.97	55.59	4.21	1	0.00
6	14.14	0.74	76.74	1.62	1	0.01
7	23.64	2.01	54.18	2.87	1	0.00
8	19.61	1.33	70.10	5.18	1	0.00
9A	2.93	0.24	20.24	1.23	3.31	0.18
10B	4.71	0.34	61.90	2.9	8.94	0.87
9C	3.01	0.46	26.56	4.84	3.41	0.1
10D	3.79	0.67	40.13	7.2	8.82	0.4
9E	2.44	0.17	20.48	1.73	3.41	0.1
10F	2.35	0.20	36.38	2.93	8.86	0.16

homogeneous hierarchical network with access point in the middle (scenario 7) stops providing its services at 52 seconds of simulation. Scenario 1 stops its production at 63 seconds and scenario 5 stops its production at 77 seconds of simulation, while the network in scenario 3 presents 17% of its nodes producing at 73 seconds. The network in the scenario 9 still has 85% of its nodes producing at 102 seconds. Notice that only for networks that implement the management QoS monitoring service is possible to advertise the observer about the precision of the temperature an carbon monoxide level averages obtained.

5.6 Conclusion

In this chapter, we describe case study for management of different configurations (in terms of organization and composition) of WSNs. The application continuously performs data sensing, processing, and data dissemination. We used the three management dimensions in the choice of the management functions and in the development of the functional, information and physical architectures. We decided to use a centralized management approach

using a naive approach to up date the maps (WSN models) which represent the network states. Schemes to update this map were defined according to type of the network, in this case continuous. The management functions were joined in management services which are performed automatically. Other management functions and services could be used and other architectures could be implemented. Smart schemes could be adopted to build and update the maps. However, it is not objective of this chapter to develop a complete management solution but to present how the framework proposed in this thesis can be used. We tried to perform these experiments in order to evaluate scalability but the computational limitation described in Section 5.1 became this experiment unfeasible.

Resources of a wireless sensor network are critical and the results show that the management solution proposed can promote the productivity of these resources and the quality of the services provided, as well as define other schemes to other kinds of wireless sensor networks. As stated earlier, some principles have guided the conception of the management application. Among the various management services and functions proposed by the MANNA management architecture, we have chosen some of them to evaluate the management effects over the WSNs. In the WSNs the energy management is probably the main aspect to be considered, once the network lifetime depends on its rational use. Effectively, other management aspects should be also considered like, for example, the processing capacity limitation, the variations in environment conditions, the thin bandwidth used for transmission, delay and number of lost messages. The simulation results have shown that the management solution can improve the performance of the various WSNs configurations and give the observer relevant information, without additional network energy consume or cost. The experiments presented here presents contributions to the field and technical basis for the evolution of this kind of technology in the aspect of management.

Chapter 6

Conclusion

This chapter presents a summary of the efforts undertaken with regard to the proposition of a management framework for wireless sensor networks by outlining the various chapters of this document. Conclusions and observations from this study are presented. A discussion of future work is also presented.

In this thesis we have developed a framework for managing WSNs. This framework brings contributions to the field besides technical bases for the evolution of this type of technology concerning management.

As defined in the text, some principles were considered in the conception of the framework proposed, being (1) simple, (2)adherent to network idiosyncrasies (including its dynamic behavior) and (3) efficient in the use of its scarce resources.

In spite of its rapid expansion, up to now WSNs and their applications have been projected and developed without considering an integrated management solution. The application functionalities were confused with the management functionalities, and there were no mechanisms to promote distinction between them. Although this may not be a problem for small networks it certainly is for WSNs formed by hundreds or thousands nodes where there is the need of having the network and its elements to self-configure and adapt to their own state and to environmental conditions where they are operating without human intervention. Another aspect that should be considered is that the WSNs will soon be integrated to other networks, like the Internet for example. A management solution that

is capable of separating functionalities, organize management and use a generic information model may facilitate this integration. The use of the self-management paradigm has also shown itself to be adequate for the specific features of these networks.

The main obstacles to the WSNs management come from the innovation and inter-disciplinarity of the theme and from the difficulty associated to the understanding of the networks itself. During the development of this work we aimed at organizing the knowledge about WSNs proposing a functional model that allows characterization of the networks and, a list of management services and functions. The functional model developed was used as base for a novel dimension in management called "WSN functionalities". Two other management dimensions compose the three-dimension organization proposed by this thesis: functional areas and management levels. The management services and functions that compose the lists provided in this document were obtained from the use of this three-dimensional organization.

The MANNA architecture proposed in the framework establishes a distinction between the functionalities of WSNs and the functionalities of the management through the use of the three-dimension organization and three architectures that compose the management systems: functional architecture, physical architecture and information architecture. This will enable the integration of the activities of organization, administration and maintenance for this type of network.

We understand that the framework proposed in this thesis is a relevant contribution for the field, once it has not been found in the literature any work that proposes integrated management solutions for WSNs. During the development process many challenges were overcame and many still rule as, for example the use of a protocols stack adequate to WSNs. This topic, protocol stack, is not directly related to the management theme itself, but the effects of using an inadequate protocol in the experiments were identified in the results. This example illustrates the type of difficulty that we had to face. Once we had begun the development, routing algorithms and medium access control algorithms to specific environments for WSNs were not available or were not issued. Until today, there are few algorithms proposed in this area. Another example talks about the simulation

environment. Ever since the development began and until now no simulation tool specific for WSNs was found available for use. Therefore, to execute the experiments we had to build the simulation environment. The MANNASim tool is a contribution of this thesis and will be available soon. Other researchers will be able to use it reducing the time and effort in developing of not only applications but management solutions.

At the end of this work we also noticed that some decisions demanded time and effort in mistaken directions. In many occasions we tried to overcome difficulties taking responsibilities that were beyond the necessary. In these occasions we had no vision, experience or any other work reference in literature that could indicate the cost-benefit of such decisions. On the other hand, most decisions were right and with that we managed to reach the point of proposing the architecture that was presented in details in the text. One of the objectives of the MANNA architecture is to promote the productivity of the resources and the quality of services. The experiments done with MANNA show that the solution is feasible in spite of its implementation has not been yet completely tested. The implementation of a complete management solution would demand time beyond that first established for the development of this thesis.

This work can be extended in various ways. Some immediate extensions would be: (1) amplify the group of experiments to evaluate the scalability of the proposed solutions; (2) develop management solutions for other types of WSNs (for example, event-driven, on demand and programmed); (3) develop and integrate new services and automatic functions to the scenarios already developed; (4) implement hierarchical management (using the concept manager of managers) and distributed management (using the concept managers-to-managers); (5) specify management policies and apply the policy-based network management paradigm of the proposed framework; (6) use protocol profiles specific for WSNs; (7) project and evaluate mechanisms to build and update network models (maps); (8) use code mobility to migrate agents or update services; (9) amplify the generic information model; (10) evaluate the framework in real scenarios using Mica-Motes sensor nodes; (11) develop functions to be used in the context of the architecture.

References

[1] The commercialization of microsensor motes. Available in http://www.sensorsmag.com, January 2002.

[2] DARPA. Sensor Information Technology (SensIt). Available in http://dtsn.darpa.mil/ixo, February 2002.

[3] JPL Sensor Webs. Available in http://sensorwebs.jpl.nasa.gov/, February 2002.

[4] μAMPS Projet. Available in http://www-mtl.mit.edu/research/icsystems/uamps, March 2002.

[5] Smart Dust – autonomous sensing and communication in a cubic millimeter. Available in http://robotics.eecs.berkeley.edu/epister/smartdust, February 2002.

[6] Wireless Integrated Network Sensors (WINS). Available in http://www.janet.ucla.edu/wins/, March 2002.

[7] Ian Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communication Magazine*, 40(8):102–114, August 2002.

[8] B. R. Badrinath, M. Srivastava, K. Mills, J. Scholtz, and K. Sollins. Special issue on smart spaces and environments. *IEEE Personal Communications*, October 2000.

[9] M. Bhardwaj, A. Chandrakasan, and T. Garnett. Upper bounds on the lifetime of sensor networks. pages 785 – 790. IEEE International Conference on Communications, 2001.

[10] Sudeept Bhatnagar, Budhaditya Deb, , and Badri Nath. Service differentiation in sensor networks. In *Fourth International Symposium on Wireless Personal Multimedia Communications (WPMC'01)*, Aalborg, Denmark, September 2001.

[11] A. Bierman, D. Romascanu, and K.C. Norseth. RFC 3433 on Entity Sensor Management Information Base (draft-ietf-entmib-sensor-mib-02.txt). Available in ftp://ftp.rfc-editor.org/in-notes/rfc3433.txt, February 2002.

[12] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[13] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Adaptive beacon placement. In *21th International Conference on Distributed Computing Systems (ICDCS-21)*, pages 489–498, Phoenix, Arizona, USA, April 2001.

[14] Rafael Saldanha Campello and Raul Fernando Weber. Sistemas de detecção de intrusão. In *19o. Simpósio Brasileiro de Redes de Computadores*, volume I, pages 1–43, Florianópolis SC, Brasil, Maio 2001.

[15] Alberto Cerpa, Jeremy Elson, Michael Hamilton, Jerry Zhao, Deborah Estrin, and Lewis Girod. Habitat monitoring: application driver for wireless communications technology. In *ACM SIGCOMM Workshop on Data Communication in Latin America and the Caribbean*, pages 20–41, San Jose, Costa Rica, April 2001.

[16] Alberto Cerpa and Deborah Estrin. ASCENT: adaptive self-configuration sensor networks topologies. Technical Report UCLA/CSD-TR 01-0009, Departament of Computer Science, University of California Los Angeles and USC/Information Science Intitute, May 2001.

[17] W. Chen, N. Jain, and S. Singh. ANMP: ad hoc network network management protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1506–1531, August 1999.

[18] Loren P. Clare, Gregory J. Pottie, and Jonathan R. Agre. Self-organizing distributed sensor networks. *SPIE - The International Society for Optical Engineering*, pages 229–237, April 1999.

[19] Joel Conover. *Policy-Based Network Management.* Network Computing, November 1999.

[20] Budhaditya Deb, Sudeept Bhatnagar, and Badri Nath. A topology discovery algorithm for sensor networks with applications to network management. In *Short Paper in IEEE CAS Workshop*, September 2002.

[21] Budhaditya Deb, Sudeept Bhatnagar, and Badri Nath. A topology discovery algorithm for sensor networks with applications to network management. Technical Report DCS-TR-441, Department of Computer Science, Rutgers University, May 2002.

[22] Distributed Management Task Force (DMTF). Web-based Management. Available in http://www.dmtg.org, March 2002.

[23] Robert E. Van Dyck and Leonard E. Miller. Distributed sensor processing over an ad hoc wireless network: Simulation framework and performance criteria. In *IEEE Milcom Military Communications Conference*, pages 1285–1286, McLean VA, USA, October 2001.

[24] Jeremy Elson and Deborah Estrin. An address-free architecture for dynamic sensor networks. Technical Report 00-274, University of Southern California, January 2000.

[25] Jeremy Elson and Deborah Estrin. Random, ephemeral transaction identifiers in dynamic sensor networks. 21st International Conference on Distributed Computing Systems (ICDCS-21), Phoenix, Arizona, USA, April 2001.

[26] Jeremy Elson and Deborah Estrin. Time synchronization services for wireless sensor networks. In *International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel Computing Issue in Wireless Networks and Mobile Computing*, pages 1965–1970, San Francisco CA, USA, April 2001.

[27] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), May 2001.

[28] D. Estrin, R. Govindan, and J. Heidemann. Scalable coordination in sensor networks. Technical Report 99-692, University of Southern California, January 1999.

[29] D. Estrin, R. Govindan, and J. Heidemann. Embedding the internet. *Communications of the ACM*, 43(5):39–41, May 2000.

[30] J. D. Gibson. *The mobile communications handbook*. CRC Press, 1996.

[31] S. Goel and T. Imieli. Prediction-based monitoring in sensor networks: Taking lessons from mpeg. Technical Report DCS-TR-438, Rutgers University, June 2001.

[32] G. Goldzmidt and Y. Yemini. Distributed management by delegation. *Proceedings of the 15th International Conference on Distributed Computing System*, pages 333–340, June 1995.

[33] D. Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House,

Boston,MA, 1992.

[34] John S. Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan. Building efficient wireless sensor networks with low-level naming. In *Symposium on Operating Systems Principles*, pages 146–159, 2001.

[35] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd International Conference on System Sciences (HICSS00)*, pages 4–7, January 2000.

[36] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Mobile Computing and Networking*, pages 174–185, 1999.

[37] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, and Kristofer S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, November 2000.

[38] Robert Hills. Sensing for danger. Technical Report UCRL-52000-01-7/8, Lawrence Livermore National Laboratory, http://www.llnl.gov/str/JulAug01/Hills.html, August 2001.

[39] G. Hoblos, M. Staroswiecki, and A. Aitouche. Optimal design of fault tolerant sensor network. In *IEEE International Conference Control Applications*, pages 467–472, Anchorage Alaska, USA, September 2000.

[40] IBM. Autonomic Computing – Creating Self-maning Computing Systems. Disponível em http://www-3.ibm.com/autonomic/index.shtml, 2003.

[41] IEEE 802.11. CSMA-CA clarrier sense multiple access with collision detection. Available in http://grouper.ieee.org/groups/802/11/ , June 2001.

[42] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the sixth annual international conference on Mobile computing and*

networking, pages 56–67, 2000.

[43] International Organization for Standardization. *ISO/IEC ITU-T X.711 Information Tecnology – Open System Interconnection – CMIP*, Specification 1991.

[44] International Telecommunication Union (ITU). *CCITT Recommendation X.700, Management framework for Open Systems Interconnection (OSI) for CCITT applications*, May 1992.

[45] International Telecommunication Union (ITU). *ITU-T M.3010 – Principles for a Telecommunications management network*, May 1996.

[46] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[47] Ara N. Knaian. A wireless sensor network for smart roadbeds and intelligent transportation systems. M. eng. thesis, MIT EECS Department and The MIT Media Lab, June 2000.

[48] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. In *IEEE Journal on Selected Areas in Communications*, volume 15, October 1997.

[49] Stephanie Lindsey, Cauligi Raghavendra, and S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *International Conference on Communications*, 2001.

[50] Stephanie Lindsey, Cauligi Raghavendra, and Krishna Sivalingam. Data gathering in sensor networks using the energy delay metric. In *International Workshop on Parallel and Distributed Computing: Issues in Wireless Networks and Mobile Computing*, April 2001.

[51] Jie Liu, Patrick Cheung, Feng Zhao, and Leonidas Guibas. A dual-space approach to tracking and sensor management in wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 131–139. ACM Press, 2002.

[52] Antonio A. Loureiro, José Marcos S. Nogueira, Linnyer B. Ruiz, and Raquel A. Mini. Rede de sensores sem fio. pages 193–234. XXI Jornada de Atualização em Informática

do Congresso da Sociedade Brasileira de Computação, julho 2002.

[53] Antonio A. Loureiro, José Marcos S. Nogueira, Linnyer B. Ruiz, Eduardo Nakamura, Carlos Maurício Seródio, and Raquel Mini. Redes sensores sem fio. pages 179–226. Simpósio Brasileiro de Redes de Computadores (SBRC), maio 2003.

[54] Antono A. Loureiro, Linnyer B. Ruiz, Fernanda P. Franciscani, Rainer R. P. Couto, and José Marcos S. Nogueira. Middleware para redes de sensores sem fio. pages 89–115. Simpósio Brasileiro de Redes de Computadores (SBRC), maio 2003.

[55] Haiyun Luo, Paul Medvedev, Jerry Cheng, and Songwu Lu. A self-coordinating approach to distributed fair queueing in ad hoc wireless networks. In *IEEE INFOCOM - Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1370–1379, April 2001.

[56] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor network for habitat monitoring. WSNA 02 - Wireless Sensor Network Appication, September 2002.

[57] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *ACM Mobicom*, pages 255–265, August 2000.

[58] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. Coverage problems in wireless ad hoc sensor networks. In *IEEE INFOCOM - Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1380–1387, April 2001.

[59] Seapahn Meguerdichian, Farinaz Koushanfar, Gang Qu, and Miodrag Potkonjak. Exposure in wireless ad hoc sensor networks. volume 7, pages 139–150, Italy, July 2001. ACM SIGMOBILE- The seventh annual international conference on Mobile Computing and Networking.

[60] Shashank Mehrotra. Distributed algorithms for tasking large sensor network. *Thesis submitted to the Faculty of Virginia Polytechnic Institute and State Univesity*, July 2001.

[61] Dilmar Malheiros Meira. *A Model for Alarm Correlation in Telecommunications Network*. PhD thesis, Federal University of Minas Gerais, 1997.

[62] Millennial Net. Millennial net: Wireless sensor networks. http://www.millennial.net, fevereiro 2004.

[63] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. An architecture for a power-aware distributed microsensor node, 2000.

[64] Raquel A. F. Mini, Badri Nath, and Antonio A. F. Loureiro. A probabilistic approach to predict the energy consumption in wireless sensor networks. *IV Workshop de Comunicação sem Fio e Computação Móvel*, October 2002.

[65] Jeffrey Monks, Vaduvur Bharghavan, and Wen mei W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *IEEE INFOCOM - Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 219–228, April 2001.

[66] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *In Proceedings of INFOCOM - Annual Joint Conference of the IEEE Computer and Communications Societies*, San Francisco, CA, 2003. Available in citeseer.nj.nec.com/niculescu03ad.html.

[67] L. Nirupama, B. Deborah, and E. Deborah. Scalable coordination for wireless sensor networks: Self-configuring localization systems. International Symposium on Communication Theory and Applications (ISCTA), July 2001.

[68] National Chiao Tung University. Department of Computer, Information Science. Mobile Computing, and Broadband Networking Lab. Wireless sensor network. Available in http://pds.cis.nctu.edu.tw, October 2002.

[69] Dataman Lab of Department of Computer Science. Rutgers University. Available in http://www.research.rutgers.edu/ bdeb/sensor-networks.html. November 2003.

[70] S. Park, A. Savvides, and M. B. Srivastava. Sensorsim: a simulation framework for sensor networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 104–111, Boston, MA, United States, 2000.

[71] Sung Park, Andreas Savvides, and Mani B. Srivastava. Simulating networks of wireless sensors. In *Proceedings of the 2001 Winter Simulation Conference*, 2001.

[72] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *IEEE Computer Communications*, 3:1405–1413, 1997.

[73] C. E. Perkins and E. M. Royer. Ad hoc on demand distance vector (AODV) routing. In *Internet Draft*, August 1998.

[74] G. J. Pottie. Wireless sensor networks. *Information Theory Workshop*, pages 139–140, June 1998.

[75] G. J. Pottie and W. J. Kaiser. Wireless sensor networks. *Communication ACM*, (5):51–58, May 2000.

[76] Praveen Rentala, Ravi Musunuri, Shashidhar Gandham, and Udit Saxena. Survey on sensor networks. Technical Report UTDCS-10-03, University of Texas at Dallas.

[77] S. Riter and J. MacCoy. Automatic vehicle location - an overview. In *IEEE transaction on Vehicular technology*, volume VT26, February 1977.

[78] Linnyer B. Ruiz. Manna: A management architecture for wireless sensor networks. Ph.D. Proposal of Computer Science Department, Federal University of Minas Gerais, August 2002.

[79] Linnyer B. Ruiz, Thais R.M. Braga, Fabrício Silva, José Marcos S. Nogueira, and Antonio A.F. Loureiro. Service management for wireless sensor networks. *IEEE LANOMS – Latin American Network Operations and Management Symposium*, pages 55–62, September 2003.

[80] Linnyer B. Ruiz, Thais R.M. Braga, Fabrício Silva, José Marcos S. Nogueira, and Antonio A.F. Loureiro. Sobre o impacto do gerenciamento no desempenho das redes de sensores sem fio. *V WCSF - Workshop de Comunicação Sem Fio e Computação Móvel*, pages 199–210, Outubro 2003.

[81] Linnyer B. Ruiz, Fabrício, Thais R.M. Braga, Silva, José Marcos S. Nogueira, and Antonio A.F. Loureiro. On impact of management on wireless sensor networks. *IEEE NOMS – Network Operations and Management Symposium*, pages 657–670, April 2004.

[82] Linnyer B. Ruiz, José Marcos S. Nogueira, and A. A. Loureiro. *Handbook of Sensor*

Networks: Compact Wireless and Wired Sensing Systems, volume 1, chapter Sensor Network Management. Edited by Mohammad Ilyas and Imad Mahgoub, CRCPress, June 2004.

[83] Linnyer B. Ruiz, José Marcos S. Nogueira, and Antonio A. F. Loureiro. Functional and information models for the manna architecture. *GRES03 - Colloque Francophone sur la Gestion de Reseaux et de Services*, pages 455–470, February 2003.

[84] Linnyer B. Ruiz, José Marcos S. Nogueira, and Antonio A.F. Loureiro. MANNA: A management architecture for wireless sensor networks. *IEEE Communications Magazine*, 41(2):116–125, February 2003.

[85] Linnyer B. Ruiz, Kalina Ramos Porto, José Marcos S. Nogueira, and A. A. Loureiro. Qualidade de serviço em redes de sensores sem fio. *V WCSF - Workshop de Comunicação Sem Fio e Computação Móvel*, Outubro 2003.

[86] Linnyer Beatrys Ruiz. Manna: Uma arquitetura para o gerenciamento de redes de sensores sem fio. RT.DCC/UFMG 005/2002, Departamento de Ciência da Computação da Universidade Federal de Minas Gerais, Agosto 2002.

[87] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad hoc networks of sensors. *In The seventh annual international conference on mobile computing and networking*, pages 166–179, July 2001.

[88] Andreas Savvides, Sung Park, and Mani B. Srivastava. On modeling networks of wireless microsensors. pages 318–319. In Joint international conference on Measurement and modeling of computer systems, Cambridge, MA, United States, June 2001.

[89] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Optimizing sensor networks in the energy-latency-density design space. In *IEEE Transactions on Mobile Computing*, volume 1, 2002.

[90] Loren Schwiebert, Sandeep K. S. Gupta, and Jennifer Weinmann. Research challenges in wireless networks of biomedical sensors. In *Mobile Computing and Networking*, pages 151–165, 2001.

[91] Chien-Chung Shen, Chavalit Srisathapornphat, and Chaiporn Jaikaeo. Sensor Information Networking Architecture and Applications. *IEEE Personel Communication*

Magazine, 8(4):52–59, August 2001.

[92] Ana Paula Ribeiro Silva, Fernando Teixeira, Rafael Lage, Linnyer B. Ruiz, A. A. Loureiro, and José M. Nogueira. Using distributed snapshot algorithm in wireless sensor networks. *The 9th IEEE Workshop on Future Trends of Distributed Computing Systems*, pages 31–27, May 2003.

[93] Fabrício Silva, Thais R.M. Braga, Linnyer B. Ruiz, and José Marcos S. Nogueira. Tecnologia de nós sensores sem fio. Relatório Técnico RT.DCC/UFMG 006/2003, Departamento de Ciência da Computação, janeiro 2003.

[94] NS-2 Network Simulator. Available in http://www.isi.edu/nsnam/ns, July 2003.

[95] A.and A. Wang Sinha and A. Chandrakasan. Algorithmic transforms for efficient energy scalable computation. pages 31–36. Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED), 2000.

[96] Raghupathy Sivakumar, Prasun Sinha, and Vaduvur Bharghavan. Cedar: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8), August 1999.

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

[98] J. M. Souza and M. R. B. Martini. Avaliação de confiabilidade de sistemas. In *Simpósio em Sistemas de Computadores Tolerantes a Falhas*, volume II, pages 24–28, Campinas SP, Brasil, Agosto 1987.

[99] Mani B. Srivastava, Richard R. Muntz, and Miodrag Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving enviroments. pages 132–138. Mobile Computing and Networking, July 2001.

[100] Madhavi W. Subbarao. Ad hoc networking critical features and performance metrics. Technical report, Wireless Communications Technology Group, NIST, September 1999.

[101] S. Tilak, N. Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM Mobile Computing and Communications Review*

(MC2R), 6(2), April 2002.

[102] Mehmet Ulema. Tutorial: Management of next generation wireless networks and services. *IEEE LANOMS – Latin American Network Operations and Management Symposium*, September 2003.

[103] Marcos A. Vieira, Luiz Filipe Vieira, Linnyer B. Ruiz, , Antonio A. Loureiro, Antonio O. Fernandes, and José Marcos S. Nogueira. Scheduling nodes in wireless sensor network: A voronoi approach. *IEEE LCN – Local Computer Network*, pages 423–429, October 2003.

[104] Marcos A. Vieira, Luiz Filipe Vieira, Linnyer B. Ruiz, Antonio A. Loureiro, , and Antonio O. Fernandes. Como obter o mapa de energia em redes de sensores sem fio? uma abordagem tolerante a falhas. *V WCSF - Workshop de Comunicação Sem Fio e Computação Móvel*, pages 183–189, outubro 2003.

[105] Chieh-Yih Wan, Andrew Campbell, and Lakshman Krishnamurthy. PSFQ: A reliable transport protocol for wireless sensor networks. *WSNA - Wireless Sensor Network Application*, September 2002.

[106] Taisy S. Weber, Ingrid Jasch-Porto, and Raul Weber. Tolerância a falhas: conceitos e técnicas, aplicações, arquitetura de sistemas confiáveis. Notas de aula. Porto Alegre: Instituto de Informática, UFRG, Julho 1996.

[107] Alec Woo and David E. Culler. A transmission control scheme for media access in sensor networks. In *Mobile Computing and Networking*, pages 221–235, 2001.

[108] W.Stallings. *SNMP, SNMPv2, SNMPv3, ROMON, and ROMON2: Pratical Network Management*. Addison-Wesley, 3rd edition, 1998.

[109] Kui Wu and Janelle Harms. QoS support in mobile ad hoc networks. *Crossing Boundaries- the GSA Journal of University of Alberta*, 1(1):92– 106, November 2001.

[110] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. USI/ISI Technical Report ISI-TR-543, September 2001.

[111] Y. Zhao, R. Govindan, and D. Estrin. Residual energy scans for monitoring wireless sensor networks. SC Computer Science Department, May 2001. Technical Report 01-745. Available in citeseer.nj.nec.com/zhao01residual.html.

Appendix A

Using Self-Diagnostic Management Service, a Case Study

In this appendix experiments are conducted to show how the management framework proposed in this thesis can be used to develop a self-managed WSN. In particular, we show the development of a self-diagnostic management service for an event-driven WSN. It is not objective of this appendix to develop a complete self-managed system but to use the paradigm called autonomic management in the WSN management. We also evaluate the the efficacy of the management architecture performing self-diagnosis and the impact of some automatic services defined by MANNA architecture on an event-driven WSN. For both objectives, an event-driven WSN, which performs temperature monitoring is proposed, as a case study and an unexpected event set, which makes some nodes unavailable, happens at the middle of the simulation time. This event set puts the nodes confined in a predefined region out of service. The event is like a car passing over the network, a spot of fire that burns the nodes, or another external event which could ruin some nodes. The event-driven WSN is heterogeneous and hierarchical. The sensor nodes only disseminate their data when the temperature of the monitored area surpasses predefined thresholds.

An event-driven WSN has characteristics that differ from the continuous WSN simulated in Chapter 5. When the observers do not receive any information from the network,

they may suppose that no event has happened. However, in some cases, the network or part of it may be unavailable due to energy problem or other types of failures or attacks. Section A.1 discusses these differences. The experiments considering a management solution are presented in Section A.2. Section A.3 presents the management application. Section A.4 presents the results. Section A.5 presents the appendix conclusion.

A.1 Fault Detection in Event-Driven WSNs

In terms of failure detection, event-driven networks present challenges that are not faced by the continuous networks. Under normal conditions, an observer of a continuous WSN receives sensing data at regular intervals. This stream of data not only delivers the content the observer is interested in, but it also works as an indicative of how well the network is operating. When the management application receives data from every single node, then the observer knows that everything is doing fine (of course, assuming that the messages are authenticated, and cannot be spoofed). If, however, the management application stops receiving messages (SENSOR-REPORT) from part or an entire region of the network, the observer knows that a failure has occurred. This is not the case in event-driven WSNs. In event-driven WSN without management, when the observer does not receive any data, it supposes that no event has happened. However, this could not be the case because the nodes can be unavailable or out of service for different reasons. A management solution based on MANNA uses WSN models (maps) to supervise the network states. In case of these experiments, the management solution proposed uses automatic management services to supervise the network. This service is called self-diagnostic.

In [57], a routing scheme is proposed where nodes police each other in order to detect faults and misbehavior. More specifically, nodes observe the behavior of the neighbor they are currently routing packets to, and can determine whether the message it sent was forwarded or not. If the message was not forwarded, the node concludes that its neighbor has failed and chooses a new neighbor to route to. This scheme does not help in cases where a whole region is compromised.

A.2 Description of Experiments

For our study, we have conducted a set of experiments taking into account distinct simulation scenarios. We have defined a WSN application and some management functions, as mentioned before, and evaluated the performance of the system using the Network Simulator (NS-2) [94], version 2.1b8a. Each simulation was run during 100 seconds and repeated at least 33 times.

In our application, temperature is the application parameter. Although the nodes sense the temperature continuously along the time, data is sent only when the minimum or the maximum collected value differs 2% from the last sent data. This brings the event-driven property to the sensing application. In order to simulate the temperature behavior of the environment, random numbers were generated following a normal distribution, taking into account standard deviation of 1 from an average temperature of 25° Celsius.

Figure A.1 illustrates the distribution of the nodes in the monitored area. It illustrates an hierarchical network comprised of common-nodes, cluster-heads and a base station. Common-nodes have less hardware capacity than cluster-heads and take part of the group that has the nearest cluster-head as leader. Communication among nodes is single-hop. We notice that in this scenario the nodes are not placed at the vertices of equally spaced triangles as in the scenarios used in Chapter 5 (see Figure 5.4). In the evaluated scenarios, the following variables are used:

- **Network.** It is composed of 144 common-nodes and 16 cluster-heads, which are uniformly distributed upon the monitored area (125 m $\times$ 120 m). Each cluster has a cluster-head and 10 common-nodes (in average). The MAC protocol used is IEEE 802.11 and no routing algorithm is implemented. It is assumed that the nodes are not mobile. Message size is 64Bytes. Energy level is considered to be critical when reaches 1 Joule. The communication is single-hop and the communication between cluster-heads and base station uses UDP and IEEE 802.11, but between common-node and cluster-head is used the MNMP (see Section 5.1.4) .

- **Nodes.** The initial energy of each common-node is 5 Joule. Communication range is 35 m. Bandwidth is 100 kbps, transmission energy consumption is 0.66Joule/s, reception energy consumption is 0.2W. Energy consumption in processing is not considered. The cluster-head is simulated with WINS [6] parameters (communication range is 140 m, transmission energy consumption is 1.176 Joules/s, reception energy consumption is 0.588 Joule/s, energy consumption in processing is in active mode: 0.00165 Joulte/s, in sleep mode: 0.000006 Joule/s, and in idle mode: 0.00048 Joule/s. The initial energy of each cluster-head is 50 Joules).

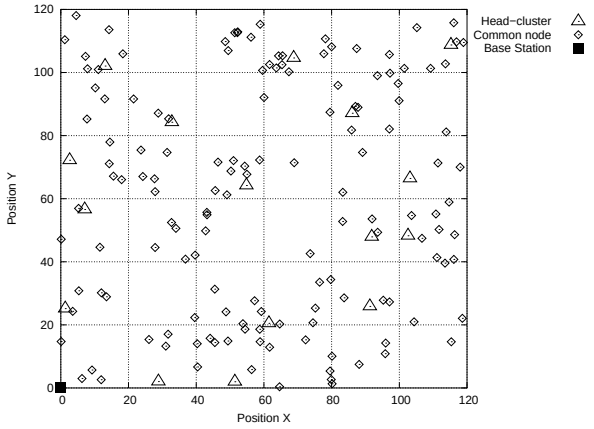


Figure A.1: Scenarios of the heterogeneous hierarchical WSNs.

A.3 Management Application for Self-Diagnostic Event-Driven WSNs

In the simulated scenarios the manager is located outside of the WSN, it has a global network view and can process algorithms that would be impossible to be carried out by the processors of the common-nodes. The agents are performed in cluster-heads. However, in these experiments, the number of nodes per cluster may vary. In general, there are ten common-nodes in each cluster. Each cluster has a cluster-head with more powerful hardware.

The management application is divided into two phases: installation and operation. The installation phase occurs as soon as the nodes are deployed in the network. In this phase, each node finds out its position (self-discovery) in the area and reports it to the agent located in the cluster-head. The agent aggregates this information together with the information gathered from the other nodes of its group, and sends a TRAP to the manager. The common-nodes also inform their energy level that the agent aggregates in an TRAP of energy and sent to the manager (self-knowledge).

The management application develops the WSN models (maps) based on both local information and the data sent by the agents, i.e., the topology map and the energy map. These two models are used to build the coverage area map, which the manager uses to monitor the sensing and communication coverage area. In the operation phase, while the sensor nodes are performing their functions, i.e., collecting, processing, and sending temperature data, management activities take place. Among them, energy level monitoring plays a central role. Each node checks its energy level and sends a message to the agent whenever there is an operational state change. This information is transmitted to the manager via another TRAP. Any information the manager receives, it uses to update the maps and performs self-diagnosis. Also, operations can be sent to the agents in order to update the maps (WSN models) and thus, to perform the self-diagnostic services. To update the maps and supervise the real state of the network, the manager sends GET operations in order to retrieve the node state. However, the conditions to perform queries using GETs are obtained from WSN models. For example, observing the residual energy in the energy map to know if there is sufficient energy to perform this operation or using other maps to determine the strategy to perform management operations. The challenge is to determine the frequency to perform queries using GETs. In these experiments, we define that during the simulation time, the manager sends three GETs. The GET-RESPONSEs are used to build the WSN audit map. If an agent or a node does not answer to a GET operation, the manager consults the energy map to verify if it has residual energy. If so, the manager detects a failure and sends a notification to the observer. In this way, the MANNA architecture provides failure detection in event-driven WSN using self-diagnosis

(management service). Other automatic services could be implemented to provide self-healing, self-protection, and so on. However, our objective is to show how use the automatic management paradigm in WSN.

A.4 Results about Self-Diagnostic Services

The monitoring scheme to be chosen depends fundamentally on the kind of application being monitored. Thus, the management requirements also change among WSNs. In these appendix, we develop a management solution using the MANNA architecture considering an event-driven wireless sensor network.

In order to evaluate the results, we have considered two sets of experiments with two distinct goals. The first set aims at identifying the efficacy of the management architecture in detecting failures. The second one aims at evaluating the impact of management functions over the WSN, analyzing the cost of management introduction. For both sets we have simulated an unexpected event to happen at the middle of the simulation time. This event puts the nodes confined in a predefined region out of service. We could think of this event as a car passing over the network, a spot of fire that burns the nodes, or another external event which could ruin the nodes.

A.4.1 Failure Detection Efficacy

For this set of experiments, we have modified the region where the nodes are ruined occurred in terms of location and dimension. Table A.1 presents the description of the simulated scenarios. For these experiments, we simulated an event which harms the nodes at 45 seconds of simulation time, putting them out of service until the end of the simulation. The management application was programmed to start the self-diagnostic sending GETs operations at times 25, 50, and 75 seconds and to report the results at times 50, 75, and 100 seconds, respectively. So, by the time that the unexpected event occurs, there was time enough (20 seconds) for the self-diagnostic management service to have come to a

conclusion regarding the availability of the nodes. This means that only the reports in 75 and 100 seconds would have to contain any conclusion regarding this event. Thus, the report at 50 seconds shows the results obtained before the event occurrence.

Table A.1: Description of the simulated scenarios for the second set.

Scenario	**Description**
1	32 nodes (20% of the network, composed of 3 cluster-heads and 29 common-nodes) located at the center of the network are harmed (see Figure A.2). These nodes have x and y coordinates between 30 and 90.
2	41 nodes (25.63% of the network, composed of 4 cluster-heads and 37 common-nodes) located near the BS are harmed (see Figure A.3). These nodes have x and y coordinates between 0 and 60.
3	39 nodes (24.37% of the network, composed of 4 cluster-heads and 35 common-nodes) located far from the BS are harmed (see Figure A.4). These nodes have x and y coordinates between 60 and 120.
4	14 nodes (8.75% of the network, composed of 1 cluster-head and 13 common-nodes) located at the center of the network are harmed (see Figure A.5). These nodes have x and y coordinates between 40 and 80.
5	62 nodes (38.75% of the network, composed of 6 cluster-heads and 56 common-nodes) located at the center of the network are harmed (see Figure A.6). These nodes have x and y coordinates between 20 and 100.

The results, shown in histograms, present the total number of nodes failures detected by self-diagnostic management service for each scenario, comparing with the number of genuine (forced) failures. The number of detected failures that were not real failures (false positives) and the number of failures not detected are also presented. Just as an illustration, Figure A.7 demonstrates the results obtained for one simulation, regarding scenario 1.

Figure A.8 shows the efficacy of the detection mechanism for scenario 1. The numbers in the x axis represent the points in time when management applications report the

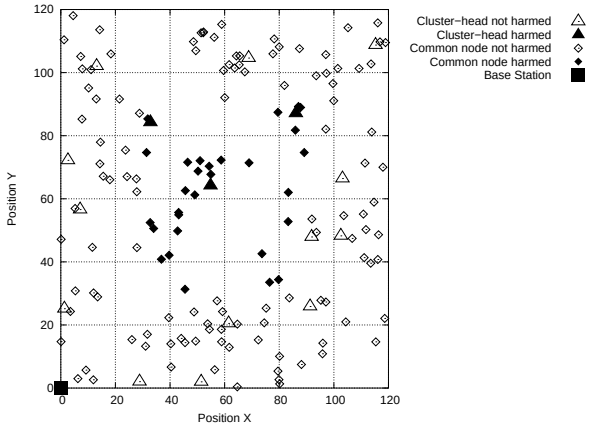


Figure A.2: Nodes harmed/not harmed in scenario 1.

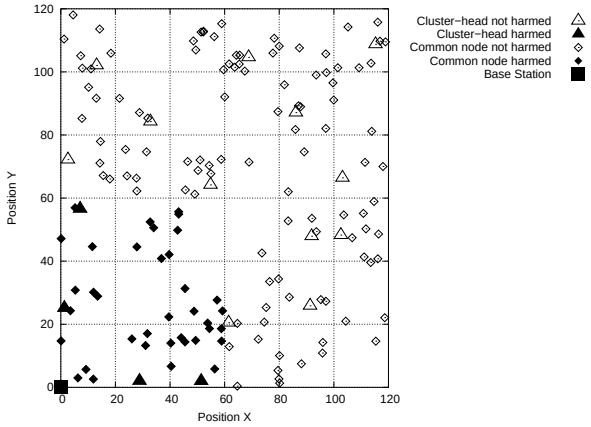


Figure A.3: Nodes harmed/not harmed in scenario 2.

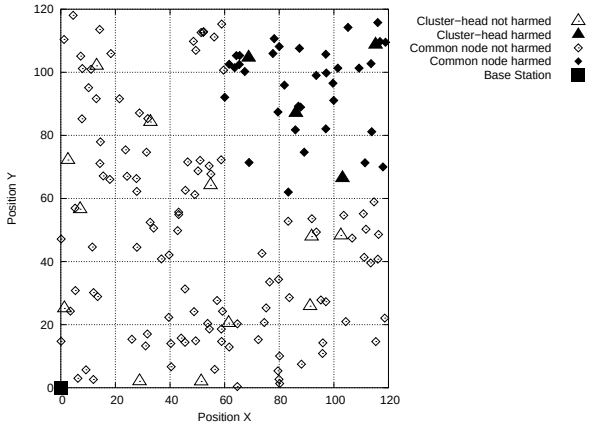


Figure A.4: Nodes harmed/hot harmed in scenario 3.

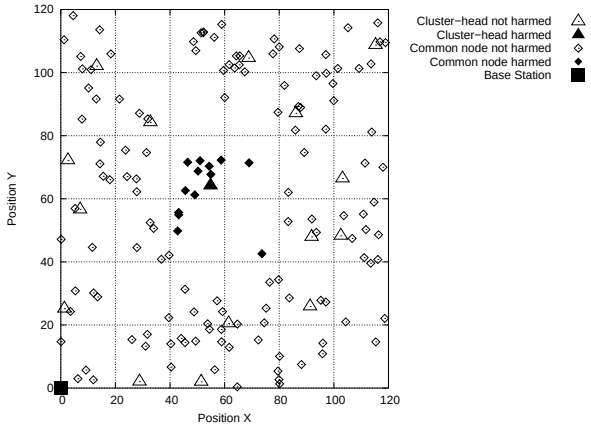


Figure A.5: Nodes harmed/not harmed in scenario 4.

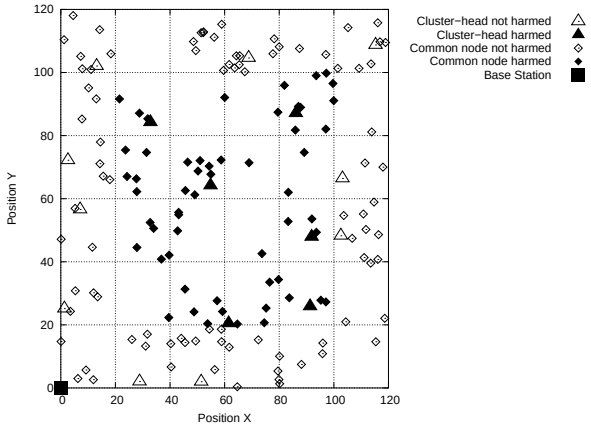


Figure A.6: Nodes harmed/not harmed in scenario 5.

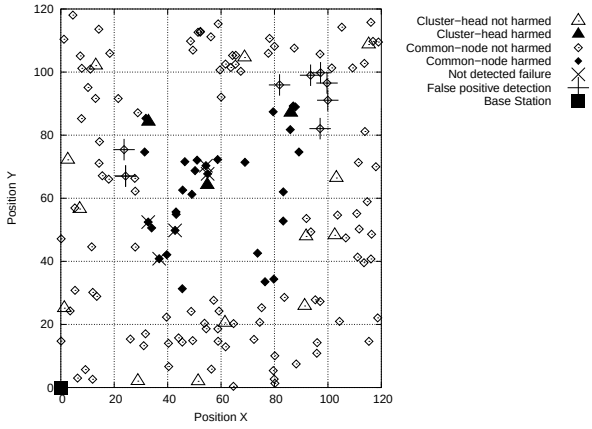


Figure A.7: Result for a case of failure detection.

availability of the nodes in the network.

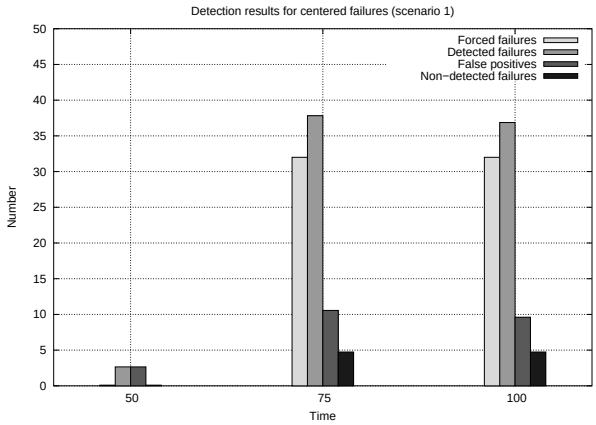


Figure A.8: Detection efficacy for scenario 1.

We can observe in Figure A.8 that there were some failure detections in time 50 seconds, although at this time the destruction of the nodes could not yet be perceived. Drops of GETs or GET-RESPONSEs operations cause the self-diagnostic to be misled and, consequently, produce false positives. This problem also occurs at points 75 and 100 for the same reason, representing 27.93% and 26.06% of the detections, respectively. The quantity of false positives at these points is considerably higher than the quantity for point 50 due to the harm of some cluster-heads where the agents run. What happens is that after the unexpected event occurs, some common-nodes, which were not harmed, loose their cluster-heads if they are located inside the damaged region. As a consequence, these common-nodes stop receiving the GETs operation from the manager, since they are sent to them through the agents. As a result, the manager does not receive answers from these common-nodes provoking false positives. Regarding scenario 1, the number of "orphan" nodes was 8.

Besides false positives, the results in Figure A.8 also show the amount of non-detected failures, representing 14.81% of the failures in both points 75 and 100. The manager cannot recognize a failure if it does not have knowledge of the damaged node. This may be caused by drops of messages that contains position information at the initial phase of the network.

Figure A.9 shows the results for scenario 2. We can see that the results for point

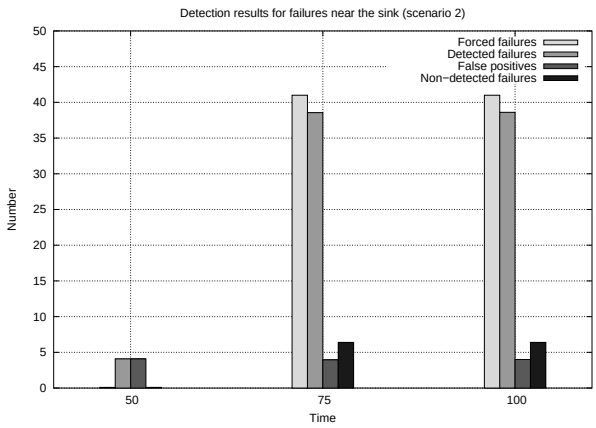


Figure A.9: Detection efficacy for failures near the BS.

50 are almost the same as the results of scenario 1. As mentioned before, at that point the unexpected event had not yet been perceived, meaning that the results seem to be independent from the chosen region. However, as far as points 75 and 100 are concerned, it is possible to observe considerable dissimilarities. The number of false positives decreases to 10.26% (point 75) and 10.36% (point 100) of the detections. The reason is that in this experiment the number of orphan nodes is only 4, i.e., two times less than the number of orphan nodes for scenario 1.

Figure A.9 also shows the results for non-detected failures. Comparing with the results of scenario 1, the amount of non-detections is similar, representing 15.59% of the failures. This shows that the number of initial messages drops in the center is similar to the region near the base station.

Figure A.10 shows the results for scenario 3. We can notice that the amount of false positives in point 50 is smaller when compared to the previous results. However, as stated before, this result is independent from the chosen region. Regarding points 75 and 100, a slight decrease in the number of false positives when compared to scenario 1 is produced. In terms of percentage of detections, this amount represents now 21.48% and 21.67% for points 75 and 100, respectively. The number of orphan nodes for this experiment is 7, very similar to the number produced for scenario 1, leading thus to another cause for the decrease. Through the logs of the simulations, it is possible to notice that the highest

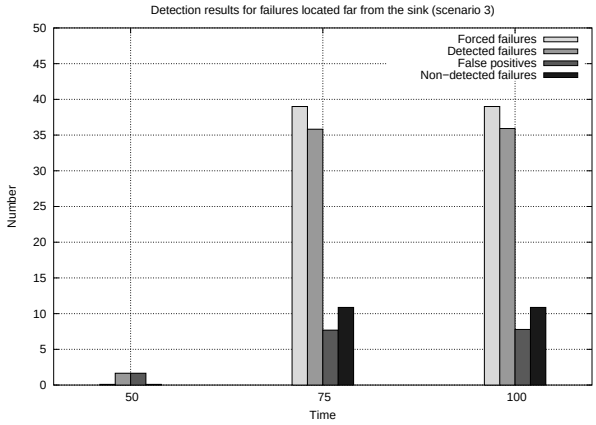


Figure A.10: Detection efficacy for failures far from the BS.

quantity of drops generally takes place in regions far from the BS. This is due to the wireless propagation behavior. The farthest the source of a message from the destination, the lower the probability of this message being delivered. For this reason, most of the false positives are nodes far from the BS (since their agents are also located far from the BS). Therefore, when these nodes are damaged the number of false positives is likely to decrease.

Figure A.10 also presents the results for non-detected failures. Comparing with the previous results, the amount of non-detections is higher, representing 27.87% of the failures. This clearly shows that the initial message drop is higher at regions far from the base station, as mentioned before. As a result, most of the nodes, which the manager does not know, are located far from it.

Figure A.11 shows the results for scenario 4. It can be noticed that the results for point 50 are almost the same as the results shown in Figure A.8 for scenario 1. As mentioned before, the results are independent from the chosen region. On the other hand, in points 75 and 100 a great difference can be perceived in the number of false positives, which decreases as expected, since the number of orphan nodes is smaller. The percentage of false positives in comparison with detections is now 14.24% and 13.95%.

Figure A.11 also presents the results for non-detected failures. Comparing to the results of scenario 1, the amount of non-detections is lower. However, in terms of percentage, it

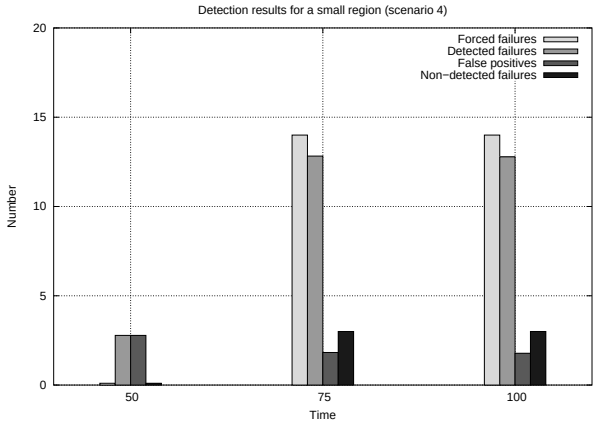


Figure A.11: Detection efficacy for less failures.

represented 27.87% of the failures, i.e., a higher result. This is due to the fact that the chosen region, as seen in Figure A.1, almost coincides with a specifc group and if an initial message from this group is lost, the manager lacks the knowledge of the whole group.

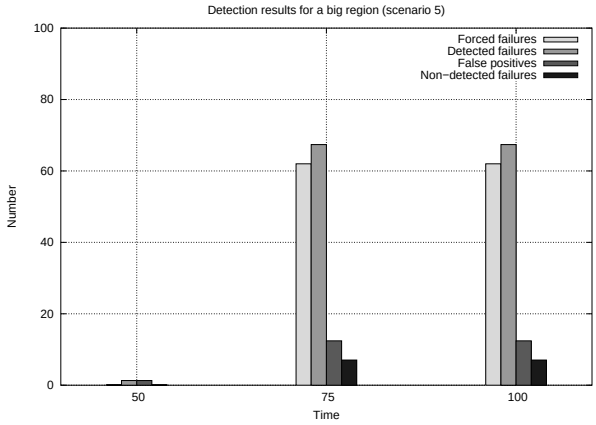


Figure A.12: Detection efficacy for more failures.

Figure A.12 shows the results for scenario 5. It can be noticed that the results for point 50 are almost the same shown before. Nevertheless, in points 75 and 100 some differences can be observed. Regarding the false positive results and comparing with the results shown in Figure A.8 (scenario1), the amount of false positives has increased slightly considering the enlarged region. Providing that the number of orphan nodes is a lot higher, a greater increase would be expected. However, since the undamaged region is smaller, there are

less drops of GETs and GET-RESPONSEs operations not due to unavailability. This leads to the reduction of the number of false positives. The quantity of false positives shown in Figure A.12 are, thus, a result of two opposed factors. As a consequence, as far as the percentage in relation to detections is concerned, there was a decrease (18.45% for both points). This shows that the detection mechanism seems to scale well.

Figure A.12 also presents the results for non-detected failures. In comparison to the results of scenario 1, the amount of non-detections is higher. This was expected since the number of damaged nodes is higher as well as their probability of being unknown to the manager. In terms of percentage, it represented 11.36% of the failures, i.e., a lower result. This shows again a good scalability.

Concerning these experiments and the results for point 50, we could note a small fixed number of false positives. Analyzing the behavior of the figures at point 75, it is possible to notice that the percentage of false positives regarding to the detections vary from 10% to 28% whereas the non-detected failures vary from 11% to 28% of the forced failures. Time 100 seconds presented almost the same results, meaning that the time has worthless influence. The main reasons for these high portions of false positives and non-detections were message drops and the creation of orphan nodes after the occurrence of an unexpected event. Sensor nodes have to communicate via wireless channels and message drops will exist. The problem could be reduced by sending redundant information or using acknowledge schemas. However, the benefits of these solutions have to be investigated because they would improve energy consumption, which is undesirable for WSNs. The problem of orphan nodes, on the other hand, could be solved by the use of an adoption schema – assigning undamaged or redundant cluster-heads to the orphan nodes.

A.4.2 Evaluating the Impact of Management

Table A.2 shows the three scenarios considered in the first set of experiments regarding to management functions.

For this set of experiments, we have considered the unexpected event to cause the

Table A.2: Description of the simulated scenarios - first set of experiments.

Scenario	Description
1	WSN with some management functions and self-diagnostic service
2	WSN with management, but without self-diagnostic service
3	WSN without management

failure of 32 nodes located at the center of the network (which have x and y coordinates between 30 and 90). This event happens at 45 seconds of simulation.

In order to evaluate the management impact over the WSN, we simulated scenarios 1, 2, and 3, and chose three metrics to analyze the results. The first one was the delivery rate, which measures the ratio of messages received by the nodes in the network to messages sent by the nodes, during the simulation time. This metric computes the ability of the network to deliver messages at their destinations. The second metric chosen was the average energy consumption, which measures the ratio of total dissipated energy by the nodes to the number of nodes in the network. This metric defines the cost of transmitting and receiving packets per node and sensing. The energy consumption of the processing is not considered. The third metric chosen was the total amount of messages generated by the nodes in the network. This metric shows the traffic imposed by the nodes tasks.

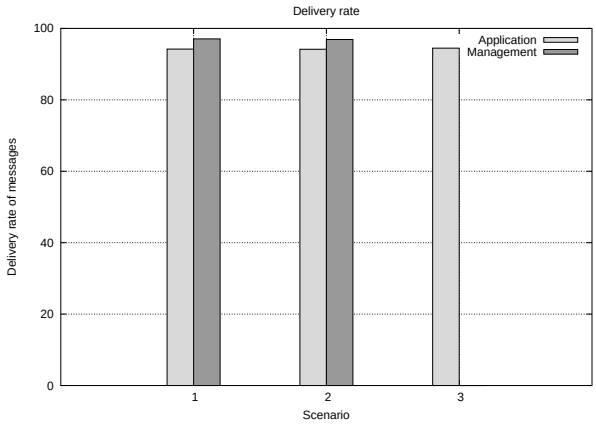


Figure A.13: Delivery rate of messages in the WSN for Scenarios 1, 2, and 3.

Figure A.13 shows the delivery rate for sensing application and management messages. It is observed that for scenarios 1 and 2 the delivery rate for management messages and application messages are similar. This is expected since they are transmitted in the same wireless environment to and from the same nodes. We can also notice that the introduction of the self-diagnostic service had no influence on this metric. When other routing algorithm are used, the network has other behavior for this metric.

Another result exhibited in Figure A.13 concerns the delivery rate of application messages. The introduction of management had little impact on the sensing application.

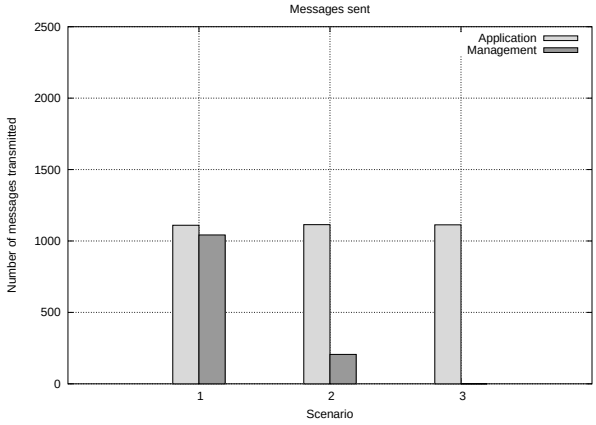


Figure A.14: Number of messages transmitted by nodes in the WSN.

Figure A.14 shows the traffic of messages in the WSN. Comparing the results for scenarios 2 and 3, we can notice that the management application contributed with a increase (18.49%) to the WSN traffic. This is due to the fact that, like the sensing application, management was implemented as event-driven. However, the number of sent messages almost doubled (increase of 93.33%) when management with self-diagnostic is concerned. This is an expected result since GETs operations have to be sent to all nodes in the network and be responded by them. Fortunately, as shown before, this is not a problem since the delivery rate of application messages is not greatly impacted.

Figure A.15 shows the energy consumption of cluster-heads and common-nodes for scenarios 1, 2, and 3. It is observed that, as far as self-diagnostic is not concerned, the energy consumption increased with management in 18% for cluster-heads and 29.45% for

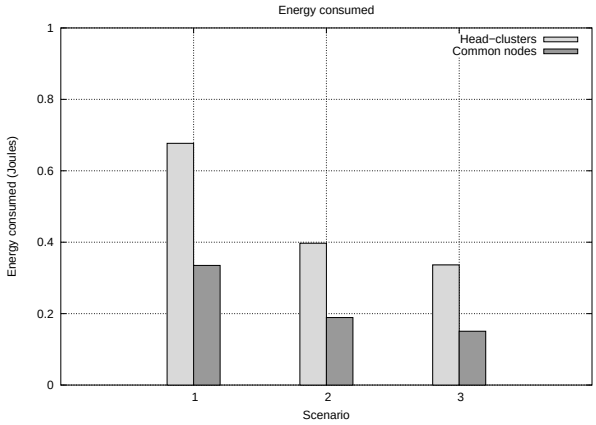


Figure A.15: Energy consumption of nodes in the WSN.

common-nodes. But when the detection mechanism was taken into account, management caused an increase of 101.2% and 129.45% in the energy consumption for cluster-heads and nodes, respectively. This result was expected since the act of transmitting and receiving messages are the most determinant activities for energy consumption according to the simulated energy model.

A.5 Conclusion

This appendix showed how the autonomic management paradigm is used in the automatic management services defined by MANNA architecture. In particular, we developed a simple management solution that use of self-diagnostic to identify unavailable nodes in the network. Other automatic management functions and maps could be used in these experiments. However, our objective is to show how to use the proposed framework to develop a self-managed WSN. From the experiments presented above, we can see that the solution proposed achieves a reasonable detection rate, and that it incurs an overhead that is acceptable for mission-critical applications. The results shows that the introduction of management with self-diagnostic in the WSN was responsible for a great increase in the number of messages transmitted in the network. Although the delivery rate of the sensing application messages was not affected, the energy consumption of the network

grew considerably. In spite of that, the experiments also show that the number of harmed nodes and their location does not influence much the efficacy of the detection mechanism proposed. The main conclusion we could draw about our approach is that its cost is fixed and its efficacy is the same, independent from the failures that take place. Although one might think at first sight that the cost introduced by management is high enough to be paid for, this could be worth, since failures are a common fact in wireless sensor networks.