

Um Elemento Autônomo para Redes de Sensores Sem Fio

Thais R. M. Braga¹, Fabrício A. Silva¹, Linnyer B. Ruiz²,
José Marcos S. Nogueira¹, Antonio A. F. Loureiro¹

¹ Departamento de Ciência da Computação

² Departamento de Engenharia Elétrica

Universidade Federal de Minas Gerais

Av. Antônio Carlos, 6627

31279-010 Belo Horizonte, MG, Brasil

{thaisrb, fasilva, linnyer, jmarcos, loureiro}@dcc.ufmg.br

Abstract. *Autonomic Networks control and supervise themselves without direct human intervention. In the definition and development of these networks, the smallest part to be built is the Autonomic Element (AE). The design aspects and the functionalities involved with the modelling of an AE influence the performance of managed elements, network and provided services. This article presents the proposal of an Autonomic Sensor Element (ASE) for Wireless Sensor Networks (WSNs). The challenge in this case is to design ASEs that can be executed in environments with severe hardware, software and data communication restrictions and that still can promote productivity of these resources and quality of the services provided by the sensor nodes. Particular, this work creates an instance of an ASE model considering the Mica Motes 2 sensor nodes platform. The results have shown the relevance of applying autonomic computing in WSNs.*

Resumo. *Redes Autônomicas controlam e supervisionam a si próprias sem intervenção humana direta. Na definição e desenvolvimento destas redes, a menor parte a ser construída é o Elemento Autônomo (EA). Os aspectos de projeto e as funcionalidades envolvidas na modelagem de um EA influenciam no desempenho dos elementos gerenciados, da rede e dos serviços provisionados. Este trabalho apresenta a proposta de um Elemento Sensor Autônomo (ESAs) para Redes de Sensores Sem Fio (RSSF). O desafio neste caso é projetar ESAs que possam ser executados em ambientes com restrições severas de hardware, software e comunicação de dados e que ainda assim possam promover a produtividade destes recursos e a qualidade dos serviços provisionados pelos nós sensores. Em particular, o trabalho instância um modelo de ESA considerando a plataforma de nós sensores Mica Motes 2. Os resultados mostram a relevância de se aplicar computação autônoma em RSSFs.*

1. Introdução

Redes de Sensores Sem Fio (RSSFs) são consideradas um tipo especial de redes ad hoc formadas, na maioria dos casos, por centenas a milhares de elementos de rede (nós sensores) depositados em ambientes remotos onde a manutenção por técnicos e a administração local são impraticáveis. O controle e a supervisão dos elementos dessas redes é uma tarefa complexa uma vez que a obtenção de dados referentes à energia, localização e a qualidade de serviço implica no aumento do tráfego e conseqüentemente um aumento no consumo de energia. Este trabalho propõe a aplicação do paradigma de computação autônoma em RSSFs com objetivo de promover a produtividade dos recursos dessas redes e a qualidade dos serviços providos. Em particular, o trabalho propõe um modelo de elemento

autonômico a ser embutido nos elementos dessas redes que poderá ser instanciado para diferentes plataformas de nós sensores.

A computação autonômica define sistemas computacionais que são capazes de gerenciar a si próprios com nenhuma ou mínima intervenção humana [Kephart and Chess 2003]. A implementação dos conceitos ligados à computação autonômica em redes de computadores leva a criação de um novo conceito: as redes autonômicas. Esse tipo de rede é capaz de realizar auto-gerenciamento de seus elementos e das conexões entre eles. Os serviços e funções de gerenciamento da rede são executados sem envolvimento de um gerente humano e de forma transparente para o usuário da mesma. Além disso, a rede é capaz de aprender com as ações praticadas por seus elementos e análise dos resultados obtidos. A execução automática de tarefas e a possibilidade de aprendizado caracterizam o aspecto autonômico deste tipo de rede.

Elementos Autonômicos (EAs) são a menor parte de um sistema autonômico e podem ser vistos como sistemas individuais que contém recursos e provêm serviços para humanos e/ou outros EAs. É possível se ter EAs embutidos em diferentes tipos de elementos gerenciados, como dispositivos de hardware (discos e CPUs por exemplo), unidades de software e até mesmo em coleções de componentes, como um desktop. Um EA contém um único gerente que representa e monitora o(s) elemento(s) gerenciado(s). O projeto de EAs particulares para cada dispositivo ou coleção de componentes pode ser considerado um dos principais desafios de pesquisa da computação autonômica [Kephart 2005]. Este trabalho trata do desafio de se propor um elemento autonômico para RSSFs: o Elemento Sensor Autonômico (ESA). A definição dos serviços autonômicos e suas respectivas funções foi realizada através da observação das particularidades das RSSFs e seus componentes, e considerando a proposta da arquitetura Manna [Ruiz 2003] para auto-gerenciamento deste tipo de rede. O formato genérico de um EA descrito em [Kephart and Chess 2003] foi utilizado como base para o modelo do ESA.

O restante deste artigo está estruturado como a seguir: a Seção 2. discute os principais aspectos relacionados com redes autonômicas. A Seção 3. descreve os trabalhos relacionados encontrados na literatura. O modelo proposto neste trabalho para um ESA está apresentado na Seção 4.. O projeto de simulação e a análise dos resultados obtidos estão descritos respectivamente nas Seções 5. e 6.. A Seção 7. contém os comentários finais.

2. Redes Autonômicas

A computação autonômica é um paradigma amplo, que define a capacidade de auto-gerenciamento para diversos tipos de sistemas computacionais. No entanto, as máquinas e os ambientes computacionais atuais não estão isolados e assim não será suficiente manter o foco somente em partes de um sistema, como unidades de hardware, software ou mesmo um conjunto de componentes. É necessário que a rede que realiza a comunicação entre os componentes ou sistemas também implemente os aspectos da computação autonômica. Neste caso, os elementos de rede e as próprias redes podem ser vistas como sistemas computacionais autonômicos ou redes autonômicas.

Esse cenário de expansão tecnológica e o aumento de complexidade, heterogeneidade, ubíqüidade, conectividade e integração são alguns dos motivos que levam à necessidade de desenvolvimento de redes autonômicas, as quais devem utilizar novas funcionalidades, serviços e paradigmas de gerenciamento.

As redes autonômicas devem ser capazes de diagnosticar e localizar falhas em sua estrutura e agir sobre elas de forma que possam ser eliminadas ou isoladas, causando o menor impacto possível no provisionamento de serviços para o usuário. Essa capacidade do sistema em se recuperar de rotinas e eventos extraordinários é chamada

auto-cura. Além disso, essas redes devem implementar mecanismos de proteção, que garantam a manutenção da segurança e integridade de seus elementos (auto-proteção). A tarefa de configuração de rede de computadores tem se tornado cada vez mais difícil e trabalhosa. Redes heterogêneas, com grandes dimensões e uma natureza dinâmica tornam a configuração algo ainda mais complexo. Assim, é necessário que as redes autonômicas implementem a auto-configuração, sendo capazes de realizá-la sempre que necessário e ainda em condições variáveis e imprevisíveis. Finalmente, a auto-otimização é também um aspecto importante. As redes devem realizar monitoração e ajuste constante de elementos e enlaces, procurando maneiras de otimizar seus desempenhos.

A menor parte de uma rede autonômica são seus elementos de rede. Cada elemento é capaz de se monitorar e controlar, possuindo um EA embutido responsável por implementar serviços e funções de gerenciamento. Vale lembrar que o hardware ou software de um componente da rede pode ser um sistema autonômico independente, possuindo capacidade de auto-gerenciamento. Cada EA atua como um gerente, responsável por promover a produtividade dos recursos e a qualidade dos serviços providos pelo componente da rede no qual está instalado. O gerente executa um laço de controle contínuo que realiza monitoração e análise de dados internos e externos ao elemento gerenciado, e também planejamento e execução de ações corretivas ou que otimizem o funcionamento deste elemento. Além disso, o EA possui um módulo que é uma base de conhecimento onde armazena dados, regras, limites, dentre outros. A definição de um EA descrita acima é um modelo genérico proposto em [Kephart and Chess 2003].

As funções que compõem cada um dos serviços do laço de controle contínuo de um EA, sua base de conhecimento, as etapas presentes em seu ciclo de vida e a forma como elas são executadas, bem como o modelo de relacionamentos entre EAs de uma rede autonômica, devem ser definidos de forma específica e particular para cada tecnologia e tipo de rede. A definição desses aspectos é apontada por Jeffrey O. Kephart¹ em [Kephart 2005] como um dos principais e fundamentais desafios de pesquisa ligados a área de computação autonômica. Apesar da necessidade de se definir o contexto específico dos EAs de cada ambiente computacional, é interessante que todos sejam desenvolvidos de acordo com um modelo genérico por questões de padronização e interoperabilidade.

2.1. Redes Sem Fio Autonômicas

Atualmente, as redes sem fio têm sido cada vez mais utilizadas. Novos hardwares, protocolos, serviços e sistemas operacionais, têm sido desenvolvidos para esse tipo de rede. Além disso, os usuários esperam que as redes sem fio sejam integradas às tecnologias tradicionais, tais como a Internet e as redes de computadores com fio locais (LANs), aumentando sua utilidade, abrangência e complexidade. Tipicamente, os elementos das redes sem fio possuem alguns tipos de restrição de recursos. O mais comum é a restrição de energia, devido à utilização de uma fonte finita desse recurso. No entanto, ainda é possível observar restrições de memória, largura de banda, alcance, dentre outros. Desta forma, é evidente a necessidade de utilização de uma ferramenta de gerenciamento específica para as redes sem fio, capaz de controlar e resolver questões de compatibilidade, qualidade de serviço, utilização racional de recursos e conectividade. Os recursos oferecidos pela computação autonômica se encaixam com as demandas por serviços e funções de gerenciamento necessários para as redes sem fio. Algumas soluções autonômicas já foram discutidas e desenvolvidas para redes tradicionais. No entanto para um ambiente tão cheio de restrições e falhas como o das redes sem fio, o qual poderia obter grandes benefícios com o uso de serviços autonômicos, foram encontrados poucos trabalhos disponíveis na literatura (ver Seção 3.).

¹Jeffrey O. Kephart é um importante pesquisador da área de computação autonômica. Ele trabalha com pesquisas nessa área no *IBM Thomas J. Watson Research Center*

As RSSFs são um tipo especial de redes sem fio e ad hoc, dependentes da aplicação e frequentemente projetadas para operar em ambientes inóspitos ou hostis. Considerando que as RSSFs possuem todos os desafios descritos acima para as redes sem fio, além de outros ligados as redes ad hoc (organização e manutenção por exemplo), redes móveis e aqueles relacionados às suas características particulares, percebe-se que a tarefa de gerenciar esse tipo de rede não é trivial. Portanto, mais uma vez a implementação de recursos autonômicos é de fundamental importância. Para utilizar efetivamente o conceito de redes autonômicas em RSSFs é necessário projetar um elemento sensor autonômico. O ESA será a menor parte em uma RSSF autonômica, e deve estar embutido em cada nó sensor da rede.

3. Trabalhos Relacionados

Existem alguns trabalhos disponíveis na literatura que tratam do uso de serviços ou aspectos autonômicos em RSSFs. Todos esses trabalhos demonstram os benefícios da aplicação de alguma característica particular do paradigma de redes autonômicas. No entanto, nenhum deles apresenta um modelo para tornar o elemento de rede das RSSFs (nó sensor) autonômico. Em [Marsh et al. 2004] os autores discutem como características autonômicas podem ser incorporadas em RSSFs utilizando a deposição de tecnologias móveis e ágeis baseadas no uso de múltiplos agentes. Uma RSSF contendo três nós sensores foi utilizada no experimento realizado neste trabalho. Cada nó dessa rede foi programado de forma diferente, e os resultados mostraram que o nó autonômico foi capaz de detectar eventos com mais eficiência, além de enviar um número maior de dados para o Ponto de Acesso (PA).

No artigo descrito em [Pujolle and Chaouchi 2005], os autores apresentam uma arquitetura autonômica (*Autonomic-Oriented Architecture* - AoA) definida para dar suporte de auto-organização e auto-gerenciamento em redes de sensores. A arquitetura AoA é definida como um conjunto de 4 planos de abstrações, sendo eles os planos de gerenciamento, conhecimento, controle e dados. Não é o objetivo do artigo propor novos algoritmos, protocolos ou esquemas para o plano de controle, mas sim prover uma seleção dinâmica dos melhores algoritmos, protocolos e valores de parâmetros a qualquer momento. Outros artigos disponíveis na literatura, tais como [Liu and Martonosi 2003, Ruiz et al. 2005] também utilizam aspectos de computação autonômica para implementação de algum serviço ou função, como auto-organização, auto-manutenção e auto-configuração em RSSFs. Este trabalho difere dos artigos citados acima por não apresentar soluções autonômicas para serviços específicos, e sim o projeto de um ESA capaz de possibilitar a implementação do paradigma de redes autonômicas nos componentes das RSSFs.

Em relação à utilização do paradigma de redes autonômicas no desenvolvimento de soluções para redes tradicionais, existem alguns artigos disponíveis na literatura que tratam do desenvolvimento de técnicas e algoritmos para auto-organização [Eymann et al. 2003], auto-configuração e distribuição e reforço de aprendizado [Littman et al. 2004], por exemplo. No entanto, todas essas abordagens foram definidas e projetadas para redes tradicionais, não sendo adequadas ou mesmo diretamente aplicáveis em redes sem fio e em particular em RSSFs, uma vez que não consideram as características intrínsecas desses tipos de rede.

Em [Ruiz 2003], uma arquitetura para o gerenciamento de RSSFs chamada Manna é especificada. Essa arquitetura, pioneira em sua proposta, define a utilização das duas dimensões de gerenciamento tradicionais (Áreas Funcionais e Níveis de Gerenciamento) e propõe uma nova dimensão (Funcionalidades das RSSFs) para o gerenciamento dessas redes. A proposta de um ESA para RSSFs complementa o objetivo da arquitetura Manna, que é o de permitir que os componentes dessas redes gerenciem a si próprios, com o mínimo de intervenção humana.

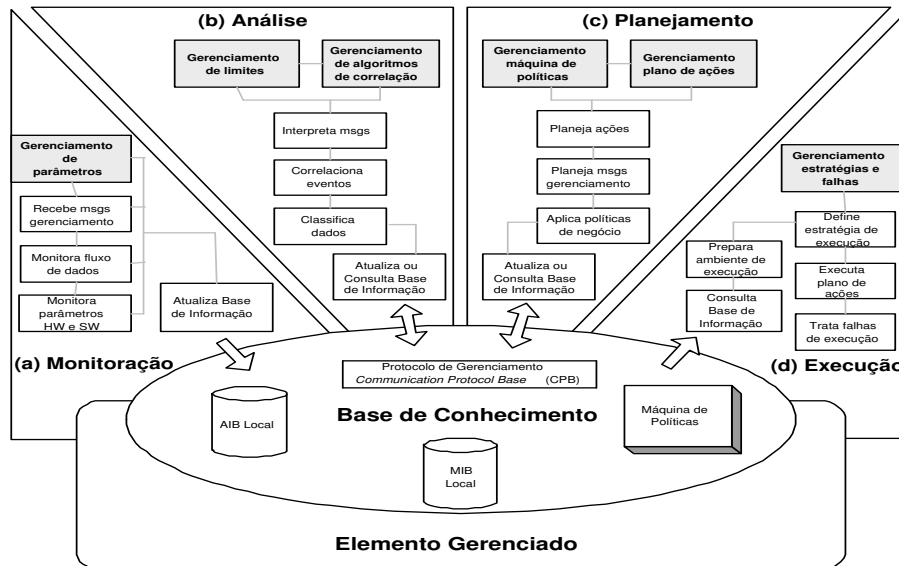


Figura 1. Visão geral dos serviços e funções que compõem o Elemento Sensor Autônomo.

4. Elemento Sensor Autônomo

Nesta seção, apresentamos uma proposta de um elemento autônomo para RSSFs, chamado Elemento Sensor Autônomo (ESA), capaz de ser executado em ambientes com restrições severas de hardware, software e comunicação de dados e ainda assim promover a produtividade destes recursos e a qualidade dos serviços provisionados pelos nós sensores. Os EAs podem ser vistos como a menor parte de um sistema autônomo e, apesar de terem seus papéis de gerentes bastante definidos, as tarefas desempenhadas e a forma como as mesmas são implementadas diferem de um ambiente para outro. Em redes de computadores tradicionais, um EA pode ser responsável por diversas funções de gerenciamento, implementando-as de forma bastante sofisticada e complexa. Por outro lado, uma vez que EAs em RSSFs estão embutidos em dispositivos com recursos limitados, eles precisam ser robustos, otimizados, flexíveis e devem observar rigorosamente o compromisso entre complexidade e precisão.

De acordo com o formato de um EA genérico proposto em [Kephart and Chess 2003], um elemento autônomo é constituído de cinco partes: monitoração, análise, planejamento, execução e uma Base de Conhecimento (BC). No modelo do ESA proposto neste trabalho, as primeiras quatro partes são vistas como serviços de gerenciamento, executados de forma autônoma através de um conjunto de funções. A BC pode apresentar diferentes componentes, dentre eles uma base de informações de gerenciamento (MIB) local e uma máquina de políticas. Os serviços são conectados através de um laço autônomo de execução infinito. Todos os serviços interagem de alguma forma com a BC. A seguir estão descritos individualmente cada um dos serviços providos pelo ESA e suas respectivas funções, além da BC e seus componentes. As funções demarcadas em cinza na Figura 1 indicam pontos em que algoritmos de aprendizagem devem ou podem ser incluídos. Embora existam limitações físicas, algoritmos simples, otimizados e eficientes podem ser utilizados e o ESA pode decidir executá-los ou não de acordo com a disponibilidade de recursos.

4.1. Serviço de Monitoração

Um sistema autônomo não pode gerenciar o que ele não sabe que existe. Neste caso, o auto-conhecimento e auto-consciência são conceitos importantes e devem ser providos

pelo modelo de elemento autônômico utilizado. O auto-conhecimento indica que um elemento gerenciado deve conhecer a si próprio e a seus componentes, enquanto a auto-consciência propõe que ele deve também conhecer a vizinhança e o ambiente que o cerca.

O serviço de monitoração (Figura 1(a)) de um elemento autônômico é responsável pela implementação dos dois conceitos descritos acima. Tal serviço pode ser projetado de diversas maneiras, dependendo do sistema autônômico ao qual pertence. Uma vez que os componentes das RSSFs possuem grande limitação de recursos, dentre eles energia, memória e processamento, cada serviço que o ESA embutido possa prover deve ser cuidadosamente planejado e gerenciado. Em particular, o serviço de monitoração deve se ater a um conjunto mínimo de parâmetros que represente de forma satisfatória e momentânea o estado do elemento gerenciado, seus componentes e seu ambiente. Cada parâmetro pouco relevante representa um consumo de recursos desnecessário. É importante ressaltar que o conjunto ótimo de parâmetros monitorados pelo ESA é extremamente dependente da aplicação para a qual a RSSF foi projetada, assim como do contexto ou momento no qual a rede se encontra.

Os parâmetros monitorados pelo ESA podem ser divididos em duas categorias: internos e externos. Parâmetros internos estão ligados ao conceito do auto-conhecimento e refletem o estado atual do hardware e software do elemento gerenciado. Parâmetros externos são aqueles ligados à auto-consciência e constroem a visão do elemento gerenciado em relação às condições da rede, suas conexões com vizinhos e o estado destes.

A coleta dos parâmetros internos é realizada por funções de acesso direto que o ESA possui aos componentes de hardware e software do elemento gerenciado. Em relação aos parâmetros externos, duas funções principais devem ser implementadas: monitoração do fluxo de dados e recepção de mensagens de gerenciamento. As RSSFs, quaisquer que sejam suas aplicações e formas de sensoriamento e disseminação de dados, produzem um fluxo de dados coletados. Esse fluxo é roteado em um esquema de comunicação multi-saltos até o Ponto de Acesso (PA) da rede. Assim, cada nó sensor da rede não só produz um fluxo de dados, como é responsável por disseminar outros fluxos, criados por outros nós da rede, e que passam por ele para alcançar o PA. A análise dos múltiplos fluxos de dados que são criados ou passam por um nó sensor, pode fornecer parâmetros externos importantes para a auto-consciência do ESA. Além disso, em algumas ocasiões, os diversos ESAs presentes na rede podem desejar trocar mensagens de gerenciamento entre si ou receber mensagens vindas do PA. Dentre alguns motivos para envio de mensagens estão negociação de serviços e troca de informações. Neste caso, o ESA deve monitorar o recebimento destas mensagens. Neste ponto, é necessário que uma análise dos dados monitorados seja realizada para que conhecimento seja gerado.

4.2. Serviço de Análise

A verificação dos contextos interno e externo de um elemento autônômico será realizada pelo serviço de análise. É responsabilidade deste serviço transformar os dados de entrada obtidos pelo serviço de monitoração, em informações úteis que possam ser analisadas, e que levem a conclusões sobre diversos aspectos, como desempenho ou falhas por exemplo, ligados ao elemento gerenciado. As funções definidas para o serviço de análise do ESA podem ser vistas na Figura 1(b). O conceito autônômico ligado a este serviço é o auto-diagnóstico.

Até o momento, o ESA possui novos dados, mas não novas informações, com as quais poderia analisar a situação do elemento gerenciado. A primeira função a ser executada é a classificação dos dados, transformando-os em eventos. É necessário que o ESA possua valores limites (*thresholds*) armazenados em sua BC, com os quais classificará os eventos utilizando alguma escala. O serviço pode por exemplo, possuir alguns limites aplicáveis a vários tipos de dados, e assim definir se um determinado valor está aceitável ou inaceitável, ou ainda alto, médio ou baixo. Os limites devem ser ajustados sempre que

necessário, e ainda aqueles que não são mais utilizados podem ser descartados. Cabe ao próprio serviço de análise realizar o gerenciamento destes limites.

Em seguida, a função de correlação dos eventos identificados será executada. Existem vários algoritmos de correlação disponíveis na literatura. Alguns deles são considerados simples, tais como a correlação baseada em regras, enquanto outros podem ser bastante complexos (redes bayesianas ou redes neurais). O uso de determinado algoritmo será definido pelo ESA com base na adequação do mesmo à situação atual do elemento gerenciado. Caso julgue necessário, o ESA pode manter mais de uma opção de algoritmo armazenados, utilizando-os nos momentos oportunos. Ele pode também descartar opções que já não são mais utilizadas e identificar a necessidade de requisitar o recebimento de um novo código, contendo um algoritmo mais adequado. O gerenciamento dos algoritmos de correlação é mais uma função de responsabilidade do serviço de análise.

Uma vez que o ESA pode ter recebido mensagens de gerenciamento de outros ESAs da rede ou mesmo do PA, as quais foram recebidas pelo serviço de monitoração, ele deve conter uma função de interpretação destas mensagens. O serviço de análise contém esta função, e utiliza conhecimentos sobre o protocolo de gerenciamento utilizado pela rede armazenados na BC, para executá-la. De acordo com o tipo e conteúdo das mensagens, novas classificações e correlações podem ser necessárias.

Uma importante discussão que surge neste ponto diz respeito a quais aspectos do elemento gerenciado serão observados e em quais níveis, ou seja, quais áreas funcionais e níveis de gerenciamento serão considerados. Além disso, como o modelo do ESA está baseado no gerenciamento tridimensional proposto pela arquitetura Manna, é ainda necessário definir quais funcionalidades das RSSFs devem ser analisadas. Em [Ruiz 2003] está definido um cubo que mostra as interseções entre as três dimensões de gerenciamento utilizadas por essa arquitetura. Cada interseção define uma célula, que é na verdade uma perspectiva sobre o gerenciamento de uma RSSF que indica área(s) funcional(ais), nível(is) de gerenciamento e funcionalidade(s) consideradas. Assim sendo, o ESA deve definir quais perspectivas serão consideradas. Podem ser escolhidas uma ou um conjunto de perspectivas, de acordo com cada situação em particular. Além disso, este conjunto pode ser modificado sempre que necessário. Finalizando a etapa de análise, uma função de atualização da BC deve ser executada.

4.3. Serviço de Planejamento

O serviço de planejamento de um EA possui como principal objetivo escolher ou elaborar um plano de ações a serem seguidas, de acordo com o resultado da análise realizada pelo serviço anterior. Mais uma vez, um conjunto de funções associadas a este serviço pode ser definido de acordo com o grau de precisão e autenticidade desejado e o nível de complexidade permitido. O EA pode implementar conceitos bastante sofisticados de inteligência artificial por exemplo, caso ele possua recursos computacionais compatíveis. Portanto, o projeto e desenvolvimento das funções que compõem o serviço de planejamento do ESA devem ser realizados considerando o compromisso entre complexidade e precisão. A Figura 1(c) ilustra o serviço de planejamento do ESA.

O gerenciamento baseado em políticas (PBNM) é utilizado especificamente pelo serviço de planejamento. Políticas podem ser definidas como um conjunto de considerações projetadas para guiar decisões. Elas definem objetivos e limites que governam as ações dos elementos autônômicos. Inicialmente, uma função busca na BC as informações ou conclusões obtidas sobre o conjunto de perspectivas consideradas pelo serviço de análise. Essas conclusões serão entregues a uma outra função que vai aplicá-las às políticas correspondentes. As políticas vão mapear as conclusões em uma ou mais ações, as quais serão executadas posteriormente com o intuito de ajustar, melhorar ou consertar algo, em uma ou mais perspectivas do ESA. Identificadas as ações, uma função será responsável por organizá-las em um conjunto, montando assim o plano de

ações do ESA. Outra função planejará mensagens de gerenciamento, caso seja necessário enviar alguma. As ações utilizadas pelo ESA podem estar relacionadas a vários conceitos autônômicos, dentre eles auto-recuperação, auto-otimização, auto-proteção, auto-organização, auto-manutenção, auto-sustento e auto-serviço [Ruiz 2003].

O gerenciamento da máquina de políticas e das ações que o ESA pode executar são responsabilidades também do serviço de planejamento. Neste caso, ele deverá sempre manter um conjunto ótimo de políticas e ações, descartar os itens desnecessários e identificar o momento em que atualizações são necessárias. Novas ações ou políticas são inseridas por este serviço na BC do ESA. Especificamente em relação às políticas, a função de gerenciamento da máquina deve ser capaz de receber políticas de negócio, que são as políticas de alto nível especificadas pelo observador da rede, e transformá-las em políticas mais simples ou mesmo SLAs (*Service Level Agreements*) diretamente aplicáveis.

4.4. Serviço de Execução

O serviço de execução está relacionado a um dos principais aspectos do paradigma da computação autônômica: a auto-configuração. É através deste aspecto que o EA consegue atuar sobre o hardware ou software do elemento gerenciado, buscando configurá-los da melhor maneira possível para as necessidades deste elemento e às condições do ambiente que o cercam.

No modelo de ESA proposto neste trabalho, conforme mostra a Figura 1(d), o primeiro passo do serviço de execução é consultar a BC para recuperar o plano de ações definido pelo serviço de planejamento. De acordo com as perspectivas analisadas e o plano de ações escolhido, a auto-configuração pode ser realizada em diferentes níveis de gerenciamento e para uma ou mais áreas funcionais. Uma outra função deve preparar o ambiente de execução, de acordo com as ações. Pode ser necessário recuperar dados ou informações, preparar determinado hardware, executar algum componente de software, dentre outras atividades. Antes de chamar a função que efetivamente executa o plano de ações, uma outra função responsável por definir a melhor estratégia de execução é invocada. Principalmente no caso em que uma ação possa interferir no resultado de outra, é importante que o ESA identifique qual a melhor ordem de execução para as ações de um plano.

A função de execução do plano executa cada ação na ordem em que foram planejadas. Se as execuções forem todas bem sucedidas, o ESA completa uma iteração do laço autônômico. As mudanças realizadas devem causar algum impacto sobre o elemento gerenciado, o ambiente no qual ele se encontra e/ou em seus vizinhos. Estas mudanças serão percebidas pelo ESA, através do serviço de monitoração, na próxima iteração do laço. Caso uma ou mais execuções apresentem problemas, o ESA deve invocar uma função de tratamento de falhas. Esta função procurará formas de contornar os problemas ocorridos. Mecanismos sofisticados de identificação e correção de falhas podem levar a bons resultados, mas o consumo de recursos associados seria proibitivo. É interessante que esta função implemente estratégias simples e eficientes.

4.5. Base de Conhecimento (BC)

A BC armazena dados, informações, políticas, limites, dentre outros itens. O principal compromisso observado pelo ESA neste caso é entre consumo de espaço de armazenamento e precisão. Os nós sensores possuem uma quantidade limitada de espaço em memória. Além disso, a restrição no consumo de energia requer que mesmo as operações de leitura e escrita em memória sejam bem planejadas. Por outro lado, é interessante que o ESA possa contar com todos os itens necessários em sua BC. O ESA deve portanto manter um conjunto ótimo, contendo apenas os itens realmente necessários. Este conjunto

pode variar durante o tempo de vida da rede. Assim, os ítems desnecessários são descartados e outros novos, se for o caso, inseridos. Conforme ilustrado pela Figura 1, a BC possui quatro componentes: uma base de informações de gerenciamento local (MIB - *Management Information Base*), uma base de informações da aplicação (AIB - *Application Information Base*), uma máquina de políticas e um módulo correspondente ao protocolo de gerenciamento utilizado (CPB - *Communication Protocol Base*).

MIB local: A MIB de um EA não é estática, ou seja, seus componentes devem ser constantemente gerenciados e renovados caso haja necessidade. No caso do ESA, sua MIB deve ser dinâmica e otimizada. Apenas os componentes realmente importantes para o contexto atual da RSSFs devem ser mantidos. Cada aplicação, em cada momento específico do tempo de vida da rede, apresentará um conjunto particular. Tanto dados como informações coletadas ou produzidas e utilizadas pelos serviços de monitoração, análise e planejamento são armazenadas na MIB do ESA. Além disso, a MIB armazenará limites, mapas, eventos gerados, resultados das correlações de eventos, além do conjunto e do plano de ações.

AIB local: é uma base de informações que apresenta estrutura de armazenamento e representação de dados similar a uma MIB, mas que possui apenas conteúdo relacionado com a aplicação para a qual a RSSF foi projetada. Dado que as RSSFs são redes dependentes da aplicação, é importante que a entidade de gerenciamento, conheça e utilize detalhes da mesma, para realizar um controle satisfatório, coerente e otimizado dos elementos gerenciados. A AIB deve conter apenas limites e representações de dados, tais como mapas, específicos da aplicação.

Máquina de políticas: esta máquina é responsável por armazenar e gerenciar as políticas de negócio especificadas pelo projetista da rede [Chadha et al. 2002]. A máquina de políticas do ESA, armazenada em sua BC, é utilizada apenas pelo serviço de planejamento. Ela deve ser uma versão compacta das máquinas de políticas utilizadas em redes tradicionais. Somente os elementos mais necessários, implementados de maneira simples e otimizada, devem ser mantidos. Assim também devem ser as próprias políticas, algoritmos e SLAs utilizados.

Protocolo de comunicação: os EAs devem compartilhar uma maneira de trocar mensagens. A especificação desta maneira é dada pelo protocolo de gerenciamento utilizado. O ESA deve armazenar em sua BC, de uma forma compacta, as especificações do protocolo de comunicação utilizado na rede. Caso o protocolo utilizado seja atualizado ou mesmo modificado, caberá ao ESA se adaptar à nova situação.

5. Projetando um ESA para a Plataforma Mica2

Neste trabalho, além de propor o elemento autônomo para RSSFs (ESAs), instanciamos o modelo proposto para a plataforma de nós sensores Mica Motes 2 [Hill and Culler 2002]. Como caso de estudo, esse modelo foi instanciado considerando RSSFs planas (sem formação de grupos) e homogêneas (todos os nós possuem mesmas características de hardware). O desempenho dessas redes foi comparado ao de outras RSSFs com mesma configuração, mas que não utilizam serviços e funções de gerenciamento. Três tipos de cenários de simulação foram construídos, sendo dois deles compostos por RSSFs autônomas, cujos nós possuem uma instância de um ESA embutido e um contendo uma RSSF não-gerenciada. Neste último caso, os elementos de rede apenas sensoriam e disseminam seus dados, utilizando sempre suas configurações iniciais. Entre os cenários autônomos, dois modelos foram projetados e implementados: um modelo localizado no qual os nós sensores não trocam mensagens de controle entre si, e o outro distribuído, ou seja, com troca de informações entre os elementos da rede. As implementações foram feitas utilizando-se o simulador NS-2 (*Network Simulator - 2*) [Network Simulator 2005].

5.1. Instância de um Elemento Sensor Autônomo

A instância do ESA contém implementações simples de cada um dos quatro serviços definidos para um ESA, além de uma base de conhecimento que possui MIB, AIB, máquina de políticas e CPB. A Figura 2 ilustra o funcionamento geral da instância de ESA projetada.

O serviço de monitoração contabiliza alguns parâmetros de hardware e software, tais como energia residual, número de dados relevantes coletados (dados que apresentem valores acima da média esperada), número de dados relevantes coletados pelos vizinhos, porcentagem de perda de mensagens e quantidade de bytes transmitidos. Também são monitorados os intervalos de sensoriamento e disseminação. A monitoração é periódica, sendo realizada a partir da análise do fluxo de dados gerados pelo próprio nó e/ou recebidos de seus vizinhos, e leitura de informações do hardware e software. Nos cenários que possuem RSSFs autônomas distribuídas, os ESAs de nós vizinhos são capazes de trocar mensagens de controle entre si. Tais mensagens contêm dados contabilizados e de configuração e são trocadas periodicamente. Neste caso, mais um parâmetro pode ser monitorado: porcentagem de vizinhos com configurações semelhantes. Caso um nó perceba que suas configurações estão muito próximas a de seus vizinhos, ele pode decidir se retirar de serviço temporariamente. Essa mudança de estado administrativo requer reconfiguração de hardware, pois necessita que o nó modifique os estados de componentes, tais como rádio, processador e sensores.

Todas as informações obtidas com este serviço são armazenadas na MIB e AIB do ESA. O serviço de análise é executado em seguida. Nele, as informações obtidas são convertidas em eventos, utilizando valores de limites inferiores e superiores, previamente definidos e armazenados na AIB do ESA. O valor de cada parâmetro monitorado é classificado como alto, baixo ou médio e cada classificação representa um evento. Os eventos são utilizados em um algoritmo de correlação, o qual apresentará conclusões sobre o desempenho do nó sensor, ou seja, verificará se o nível de qualidade de serviço (QoS) desejado está ou não sendo provido.

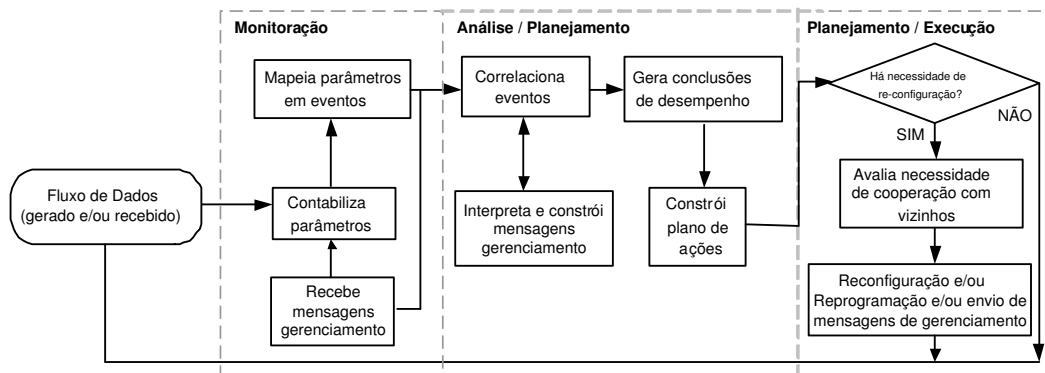


Figura 2. Instância do modelo de ESA implementada para as simulações deste trabalho.

De acordo com o resultado do serviço de análise, o serviço de planejamento deve verificar quais ações deverão ser tomadas. Para simplificar o projeto da instância de ESA utilizada neste trabalho, as opções de ações se restringiram ao aumento ou diminuição dos intervalos de sensoriamento e disseminação. Existe ainda a opção de mudança de estado administrativo (ativo ou inativo) para as RSSFs autônomas distribuídas. As mensagens de controle trocadas pelos nós desses cenários são planejadas por este serviço. As opções de ações disponíveis ficam armazenadas na MIB do ESA. Finalmente, o serviço de

execução recebe o plano de ações a serem executadas (incluindo o envio de mensagens) e o realiza de acordo com a melhor estratégia de execução possível.

5.2. Aplicação e Modelo Computacional

Neste trabalho foi escolhida a aplicação de monitoração de monóxido de carbono (CO) em ambientes fechados como estudo de caso. O modelo computacional utilizado define as características tanto da RSSF como dos elementos que a compõe. Os nós sensores que compõem a rede conhecem sua localização, e a área de sensoriamento destes nós será considerada igual à de comunicação, sendo que ambas podem ser aproximadas por um círculo. Além disso, os elementos da rede são estáticos e possuem um relógio global aproximadamente sincronizado. Os canais de comunicação não são confiáveis e pacotes podem ser perdidos e congestionamentos, bem como atrasos, podem ocorrer. Cada nó sensor considera como vizinhos aqueles que alcança através do roteamento utilizando apenas um passo.

Requisitos de Memória: A utilização de memória em relação à instância do modelo de ESA proposto neste trabalho segue a seguinte formulação:

$$Total_{Memória} = Cod_{App} + Cod_{ESA} + BC \leq X \text{ KB}$$

onde Cod_{App} é a quantidade de bytes gasta pelo código da aplicação utilizada pela RSSF, Cod_{ESA} a quantidade necessária para o código contendo a implementação dos 4 serviços da instância do ESA ($Cod_{ESA} = Cod_{Monitoração} + Cod_{Análise} + Cod_{Planejamento} + Cod_{Execução}$), BC é o número de bytes para a Base de Conhecimento do ESA ($BC = MIB + AIB + CPB + Máquina \text{ Políticas}$) e $X \text{ KB}$ é o número de bytes que o nó sensor possui disponível em sua memória de programa. O código de uma aplicação simples desenvolvido para a plataforma Mica2 necessita de aproximadamente 20 KB . A quantidade de bytes para a memória de programa depende da arquitetura do nó sensor utilizada. O Mica2 possui 128 KB disponíveis para essa memória. Neste caso, Cod_{ESA} e BC poderiam utilizar no máximo aproximadamente 100 KB da memória de uma plataforma como a do Mica2.

5.3. Detalhes de Simulação

Para realizar o roteamento nos cenários simulados, foi utilizado um algoritmo chamado EFTREE [Figueiredo et al. 2004]. Este algoritmo, que é específico para RSSFs planas, constrói uma árvore de roteamento com os nós da rede, a partir do PA da mesma. Os nós sensores retirados de serviço temporariamente pela reconfiguração de hardware dos cenários autônomicos distribuídos, continuam a realizar roteamento dos dados recebidos por seus vizinhos. Para verificar o comportamento dos nós mediante o aumento inesperado do valor médio do parâmetro sensoriado, a ocorrência de um evento durante a simulação foi planejada. Neste momento, alguns nós da rede passam a coletar dados com valores acima do esperado. A Tabela 1 apresenta cada um dos cenários implementados. As características destes cenários (rede, nós sensores, eventos, dentre outros aspectos), descritas a seguir, foram definidas de acordo com a aplicação e o modelo computacional utilizados. O número máximo de nós sensores simulados foi mantido em 200 devido a restrições computacionais (são gastas em média 3 horas para executar todos os cenários uma única vez). Diferentes valores foram testados para os intervalos de monitoração e troca de mensagens de controle. Foram escolhidos os valores que melhor representavam o comportamento da rede (melhor custo-benefício). Esses resultados foram omitidos por questão de espaço.

- **Simulação:** Tempo de simulação: 10.000s; Número de simulações: 33.

- **Rede:** Número de nós: 100 ou 200; Organização: plana; Configuração: homogênea; Tamanho da Área (X e Y): 200m x 200m (para 100 nós) e 282m x 282m (para 200 nós); Densidade: 1 nó a cada $400m^2$; Localização PA: perímetro; Distribuição dos nós: uniforme.

Tabela 1. Cenários implementados. A densidade das redes utilizadas foi sempre mantida em 1 nó/400m², ou seja, um nó a cada 400 metros quadrados.

Cenário	Tipo	# Nós
1	Autonômica Distribuída	100
2	Autonômica Localizada	100
3	Não gerenciada	100
4	Autonômica Distribuída	200
5	Autonômica Localizada	200
6	Não gerenciada	200

- **Elementos de Rede:** Intervalo inicial de sensoriamento: 20s; Intervalo inicial de disseminação: 40s; Intervalo de monitoração (ESA): 100s; Intervalo de mensagens de controle: 200s; Intervalo de desligamento de redundantes: 300s; Consumo tx: 0.036W; Consumo rx: 0.024W; Consumo com sensoriamento: 0.0021W; Consumo com processamento: 0.024W; Largura de banda: 19.2 Kbps; Tipo de sensoriamento: programado; Tipo de disseminação: programada; Pilha de protocolos: MannaNMP [Silva et al. 2005]/EFTREE/802.15.4; Mobilidade: não; Capacidade bateria: 1J; Alcance de sensoriamento e comunicação: 40m; Tamanho mensagem: 64 bytes.

- **Aplicação:** Parâmetro sensoriado: monóxido de carbono (CO); Valor médio: 43mg/m³; Máximo valor permitido: 60mg/m³; Número de eventos: 1; Duração do evento: 1000s; Início do evento: 100s; Localização do evento: metade esquerda da área monitorada; Valor sensoriado durante evento: 80mg/m³;

Em relação à classificação de eventos, os parâmetros monitorados ligados à quantidade de dados relevantes utilizam porcentagens para distinção entre alto (maior ou igual a 80%), médio (entre 20% e 80%) e baixo (menor ou igual a 20%). Para os limites ligados aos intervalos, definiu-se que serão considerados altos (grandes) se estiverem 4 vezes maiores que o valor inicial, baixos (pequenos) se estiverem 4 vezes menores e médios se estiverem no intervalo entre alto e baixo. Para a quantidade de bytes enviados, o envio de menos do que 500 bytes é considerado baixo, acima de 15000 bytes alto e entre estes dois valores, médio. Para a reconfiguração utilizamos como padrão o acréscimo ou decréscimo de 10 segundos nos intervalos até que os valores máximos ou mínimos sejam atingidos. Para o sensoriamento, foi permitido intervalo máximo de 120 segundos e mínimo de 10 segundos. Em relação à disseminação, máximo de 180 segundos e mínimo de 15 segundos.

As RSSFs autonômicas distribuídas contam ainda com um outro limite, utilizado para verificar se o parâmetro de porcentagem de vizinhos com configurações semelhantes está alto (igual ou maior a 80%), baixo (igual ou menor a 50%) ou médio (entre 50% e 80%). Além disso, foi necessário utilizar dois limites que definem quando as informações de um certo nó estão ou não bastante semelhantes às de um vizinho. Neste trabalho, caso 60% dos dados de um nó estejam diferentes dos de um vizinho em apenas 20%, considera-se que as informações destes dois nós são similares. Os cenários 2 e 5 possuem RSSFs autonômicas localizadas cujos nós não são capazes de realizar mudança de estado administrativo.

6. Análise de Resultados

6.1. Consumo de Energia

A primeira linha da Tabela 2 mostra o valor médio do consumo total de energia dos nós comuns em cada um dos cenários. Pode-se observar que o consumo de energia dos nós

nos cenários 1 e 4 (autônomico distribuído) é menor do que aquele dos demais cenários. As redes dos cenários 2 e 5, autônicas localizadas, consumiram mais energia do que as dos cenários 1 e 4, uma vez que não eram capazes de fazer reconfigurações de hardware (desligar ou modificar componentes, para mudança de estado administrativo), mas ainda conseguiram apresentar um resultado melhor do que as redes não gerenciadas dos cenários 3 e 6.

Tabela 2. Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço (sensoriamento, comunicação e processamento).

Cenário	1	2	3	4	5	6
Consumo total (J)	0,64	0,96	0,99	0,61	0,91	0,99
% Comunicação	95,06	74,92	90,85	98,48	83,65	91,83
% Processamento	4,79	24,44	8,14	1,49	16,05	7,46
% Sensoriamento	0,15	0,64	1,01	0,03	0,30	0,71

Não houve um impacto significativo no consumo de energia devido ao aumento do número de nós da rede. Isso se deve principalmente ao fato de que a densidade da rede, para todos os cenários, foi mantida constante. A Tabela 2 mostra ainda a porcentagem média de consumo de energia dos nós comuns por serviço. As redes autônicas distribuídas apresentam o maior consumo com comunicação, devido às mensagens de gerenciamento trocadas. No entanto, tais redes apresentam um consumo médio aproximadamente 40% menor do que aquele dos demais cenários. Os cenários 2 e 5 consumiram menos com comunicação, pois enviaram menos mensagens de dados do que os cenários 3 e 6 e seus nós não realizaram troca de mensagens de gerenciamento.

6.2. Informações Enviadas pelos Nós Comuns

A Tabela 3 mostra a quantidade total de mensagens, dados e dados relevantes enviados pelos nós de cada cenário durante o tempo de simulação. Considerando tanto os cenários contendo 100 nós, quanto os que contém 200, as redes autônicas conseguem enviar um número total de mensagens menor por aumentarem seus intervalos de sensoriamento e disseminação em momentos adequados. No entanto, quando a análise do fluxo detecta a presença de um evento, tais intervalos são diminuídos, o que faz com que um grande número de amostras (dados) cheguem ao observador.

Tabela 3. Número de mensagens e dados enviados. Uma mensagem pode conter um ou mais dados sensoriados. Um dado relevante é aquele coletado durante a ocorrência de um evento.

Cenário	1	2	3	4	5	6
Mensagens	5.116,8	6.536,3	9.983,1	8.288,9	10.632,6	16.408,7
Dados	8.175,5	10.256,6	19.866,4	13.731,4	17.070,5	32.617,6
Dados relevantes	3.083,0	4.384,6	2.474,4	5.904,3	7.311,3	4.627,8

As RSSFs autônicas distribuídas (cenários 1 e 4) enviaram um número menor de mensagens e dados do que as redes autônicas localizadas, uma vez que durante o tempo de vida da rede existia sempre um conjunto de nós fora de serviço. As redes não gerenciadas enviaram sempre um número maior de mensagens, mas um número menor de dados relevantes. Os cenários 2 e 5 apresentaram resultados interessantes para esta métrica, visto que seus nós enviaram um número maior de dados relevantes do que os

nós dos demais cenários. Vale ressaltar que os nós destes cenários (2 e 5) conseguem enviar mais dados relevantes, enviando menos mensagens do que os nós das redes não gerenciadas (cenários 3 e 6).

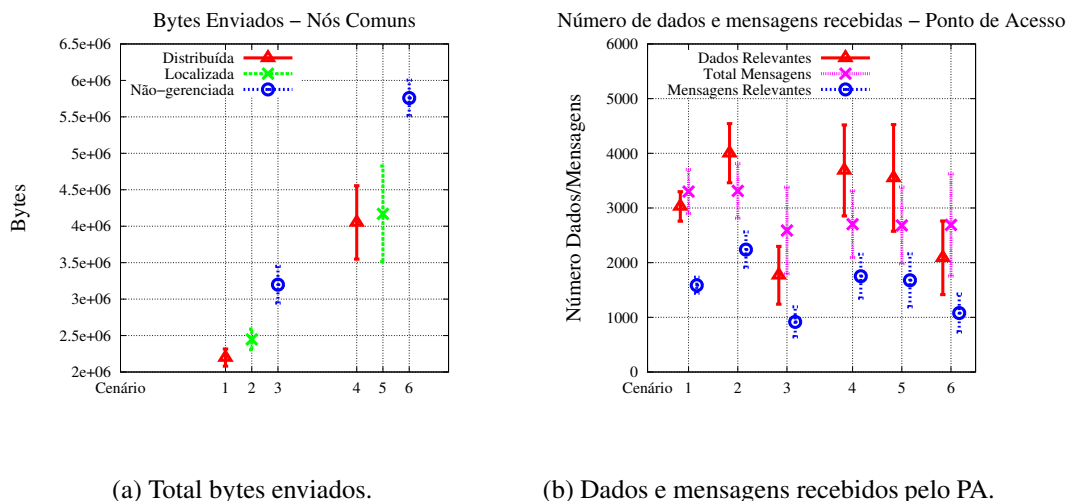


Figura 3. Médias e desvios padrão do total de bytes enviados pelos nós comuns e número de dados e mensagens recebidas pelo PA.

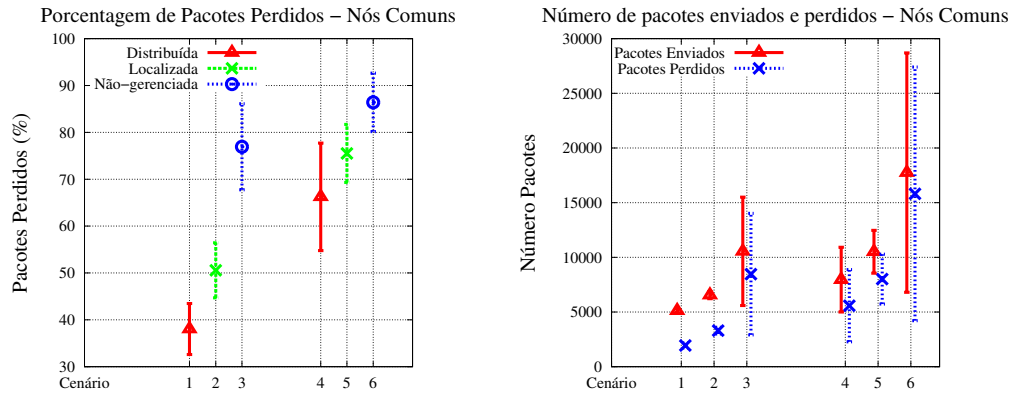
As redes autônomicas enviam grande número de dados e mensagens apenas nos momentos em que eventos ocorrem. No entanto, nos momentos em que nada ocorre no ambiente monitorado, os nós enviam um número menor de mensagens, as quais contém poucos dados sensorizados. Nenhum dado coletado é filtrado ou selecionado. O que muda nas redes autônomicas são os valores das taxas de sensoriamento e disseminação, que são ajustadas de acordo com a aplicação e com as condições internas e externas detectadas pelo ESA. A Figura 3(a) mostra que ao final, a quantidade de bytes enviados pelos nós dos cenários autônomicos é menor do que a dos cenários das redes não gerenciadas. Em particular, as redes distribuídas apresentaram um menor número de bytes enviados, devido à utilização das técnicas de reconfiguração de hardware.

6.3. Informações Recebidas pelo PA

O gráfico da Figura 3(b) mostra o número de dados, mensagens e mensagens relevantes recebidos pelo PA. A grande quantidade de dados e mensagens enviadas pelos nós sempre ativos da rede autônômica e localizada do cenário 2 não sobrecarregou o meio de comunicação, e permitiu uma maior entrega, principalmente de informações relevantes. No entanto, no cenário 5, o número alto de mensagens fez com que colisões e perdas ocorressem, e neste caso o cenário distribuído apresentou resultados ligeiramente melhores, apesar de ter enviado menor quantidade de dados e mensagens. Em geral, as redes autônomicas obtiveram sempre melhor resultado do que aquelas não gerenciadas, devido à utilização estratégica dos recursos dos nós e da rede como um todo.

6.4. Número de Pacotes Perdidos

A implementação dos serviços de gerenciamento também causa impacto sobre uma outra métrica importante: o número de pacotes perdidos. A utilização de técnicas de reconfiguração de hardware e reprogramação de software influenciam de forma positiva nesta métrica, conforme mostra o gráfico da Figura 4(a). A Figura 4(b) mostra a relação entre o número de pacotes enviados e o número de pacotes perdidos em cada um dos



(a) Porcentagem de pacotes perdidos.

(b) Número de pacotes enviados e descartados.

Figura 4. Média e desvio padrão relativos às métricas de envio e perda de pacotes.

cenários. Os cenários autônômicos enviam um número menor de pacotes, o que se reflete em menor perda, consumo de energia e número de bytes enviados.

As redes não gerenciadas dos cenários 3 e 6 apresentaram os piores resultados. O grande número de pacotes enviados continuamente levaram ao congestionamento do meio de transmissão, e conseqüentemente colisão e perda. O cenário 1 possui porcentagem de perda menor do que os cenários 2 e 3, assim como o cenário 4 em relação aos cenários 5 e 6. Ao relacionarmos a métrica de mensagens e dados enviados à porcentagem de pacotes perdidos, pode-se perceber que apesar das redes autônômicas localizadas enviarem um número maior de mensagens e dados relevantes (ver Seção 6.2.), a perda de pacotes nestes cenários é maior do que aquela nos cenários 1 e 4, indicando sobrecarga no meio de transmissão. O impacto desta sobrecarga pode ser visto no gráfico 3(b) da Seção 6.3.. Para os cenários com 200 nós sensores, o PA recebeu menos dados e mensagens relevantes do cenário 5 do que do cenário 4, apesar do primeiro ter enviado quantidades maiores do que o último.

7. Conclusão

Redes autônômicas são aquelas capazes de gerenciar seus componentes e os enlaces entre eles com pouca ou nenhuma intervenção humana. Soluções de hardware e software ligadas a este paradigma têm sido recentemente desenvolvidas pela comunidade científica. As RSSFs são redes com características específicas que possuem todas as restrições computacionais das redes sem fio, móveis e ad hoc, além de outras decorrentes de suas particularidades. A implementação do paradigma de redes autônômicas em RSSFs será freqüentemente a única forma de prover controle e supervisão para esse tipo de rede. Neste trabalho foi proposto o Elemento Sensor Autônômico (ESA) baseado no modelo genérico apresentado em [Kephart and Chess 2003] e nos conceitos da arquitetura de gerenciamento de RSSFs chamada Manna. Uma avaliação foi realizada considerando a implementação do ESA proposto para a plataforma Mica Motes 2. Os resultados obtidos mostram que é interessante embutir serviços e funções autônômicas em uma RSSF, através de um ESA, o qual é capaz de gerenciar os recursos de um nó sensor, utilizando-os de forma estratégica e otimizada. O desenvolvimento de uma solução autônômica nível 5 é ainda uma visão, uma vez que a ciência atual não dispõe das bases necessárias. No entanto, o surgimento de novas propostas, idéias e algoritmos permite um avanço gradual em direção a esse tipo

de solução.

Agradecimento

O desenvolvimento e os estudos descritos neste trabalho foram viabilizados pelo apoio financeiro concedido pela Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

Referências

- Chadha, R., Lapiotis, G., and Wright, S. (2002). Policy-based networking. *IEEE Network*, 16(2):8–9. (Guest Editorial).
- Eymann, T., Reinicke, M., Ardaiz, O., Artigas, P., Freitag, F., and Navarro, L. (2003). Self-organizing resource allocation for autonomic networks. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, page 656, Prague, Czech Republic. Lectures Notes in Computer Science - ISBN:3-540-40806-1.
- Figueiredo, C. M., Nakamura, E. F., and Loureiro, A. A. (2004). Multi: A hybrid adaptive dissemination protocol for wireless sensor networks. In *Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algo-sensors'04)*, volume 3121, pages 171–186, Turku, Finland. Springer.
- Hill, J. and Culler, D. (2002). Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24. ISSN: 0272-1732.
- Kephart, J. O. (2005). Research challenges of autonomic computing. In *27th International Conference on Software Engineering (ICSE'05)*, pages 15–22, St. Louis, MO, USA.
- Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1):41–50.
- Littman, M. L., Ravi, N., Fenson, E., and Howard, R. (2004). Reinforcement learning for autonomic network repair. In *Proceedings of the 1st International Conference on Autonomic Computing (ICAC'04)*, pages 284–285, New York, USA.
- Liu, T. and Martonosi, M. (2003). Impala: A middleware system for managing autonomic parallel sensor systems. In *Proceedings of the 9th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'03)*, pages 107–118, San Diego, CA, USA.
- Marsh, D., Tynan, R., O’Kane, D., and O’Hare, G. M. P. (2004). Autonomic wireless sensor networks. *Engineering Applications of Artificial Intelligence*, pages 741–748. Elsevier Publishers.
- Network Simulator (2005). UCB/LBNL/VINT network simulator (ns-2). Available at <http://www.isi.edu/nsnam/ns>.
- Pujolle, G. and Chaouchi, H. (2005). An autonomic oriented architecture for wireless sensor networks. *Annals of Telecommunications - Sensor Networks*, 60(6/7):819–830.
- Ruiz, L. B. (2003). *Manna: A Management Architecture for Wireless Sensor Networks*. PhD thesis, Computer Science Department of the Federal University of Minas Gerais, Belo Horizonte, MG, Brazil.
- Ruiz, L. B., Braga, T. R. M., Silva, F., ao, H. P. A., Nogueira, J. M. S., and Loureiro, A. A. F. (2005). On the design of a self-managed wireless sensor network. *IEEE Communications Magazine*, 43(8):95–102.
- Silva, F., Ruiz, L. B., Braga, T. R. M., Nogueira, J. M. S., and Loureiro, A. A. F. (2005). Defining a wireless sensor network management protocol. In *Proceedings of the 4rd Latin American Network Operation and Management Symposium (LANOMS'05)*, Porto Alegre, RS, Brazil.