

Uma Arquitetura para Auto-Organização em Redes em Malha Sem Fio

Lucas Guardalben, João Bosco M. Sobral e Clytia H. Tamashiro

Abstract—The problem of self-organization in wireless mesh networks is becoming important due to the growth in number of nodes in a network. The specific problem of manual configuration of individual nodes it has been either impossible or very expensive. In this work we propose an alternative architecture in wireless mesh network self-organized based on the Mobile Mesh Protocol and Agent technology to identify the main components involved towards the construction of a prototype. We present a formal specification of the architecture in Object-Z in order to provide a model in high level abstraction to show the applicability of the architecture.

Keywords—wireless mesh networks, self-organization, agents.

I. INTRODUÇÃO

Nos últimos anos, as redes em malha sem fio (*Wireless Mesh Networks*) tem atraído considerável atenção da indústria [1] [2] e da comunidade científica devido a gama de aplicações que podem ser desenvolvidas com bases nessas redes, principalmente em ambientes metropolitanos. Redes em malha sem fio comunicam-se através de múltiplos saltos e consistem basicamente de nodos-clientes, que suportam mobilidade ou não, e nodos-roteadores. Ambos os nodos possuem interfaces sem fio para comunicação entre eles. Os nodos-roteadores são responsáveis por formar a malha da rede, de uma forma fixa, autônoma ou pré-determinada, eles podem trabalhar como *gateways* ou *bridges*, permitindo a comunicação de diferentes tipos de redes [3], são usualmente colocados em altitudes elevadas para evitar interferências físicas na área de cobertura de suas antenas. Redes em malha sem fio é uma tecnologia emergente, e possuem vantagens em relação a maximizar sua área de cobertura, devido a possibilidade de introdução de nodos-roteadores com função de *gateways* conectados a rede cabeada afim de estender as aplicações para a malha sem fio, e também o seu baixo custo de implementação, pois pode utilizar tecnologias sem fio disponíveis no mercado para sua construção.

Geralmente as redes em malha sem fio são aplicadas em ambientes metropolitanos e podem consistir de uma grande quantidade de nodos. Devido a dificuldade em possuir uma infra-estrutura de comunicação conseqüentemente surgem dois problemas: heterogeneidade dos equipamentos envolvidos e a auto-organização (*self-organization*) (gerenciamento da topologia, descoberta otimizada de rotas e em alguns casos específicos a eleições de nodos-roteadores líderes [4]). A

capacidade de auto-organização permite que a rede não perca sua autonomia, por causa da mínima intervenção manual do administrador para configuração de cada nodo-roteador. Uma outra vantagem de redes auto-organizadas é a redução do custo de instalação e manutenção dos nodos envolvidos na rede, uma vez que os nodos-clientes e nodos-roteadores entram na rede de uma forma transparente para os usuários. No entanto, a tecnologia atual para redes em malha sem fio somente permite parcialmente realizar o objetivo da auto-organização.

Devido à natureza complexa das redes em malha sem fio, novos paradigmas são necessários para projeto, gerenciamento e manutenção. Este artigo propõe uma arquitetura para auto-organização, usando tecnologia de agentes fixos e móveis [5]. A arquitetura proposta é baseada no protocolo Mobile Mesh [6] e na construção de módulos para auto-otimização (*self-optimization*), auto-cura (*self-healing*), auto-proteção (*self-protection*) e auto-configuração (*self-configuration*) da rede, e introduz três tipos de agentes de software, responsáveis pela descoberta da topologia, descrição global da topologia e monitoramento. Além disso, duas melhorias são propostas nas fases de auto-organização, e diferenciadas das encontradas no protocolo Mobile Mesh, e juntamente com a especificação parcial em linguagem formal Object-Z [7] de algumas classes propostas neste trabalho.

Este artigo está organizado da seguinte forma: Na seção II estão os trabalhos relacionados. A seção III apresenta uma visão geral da arquitetura, juntamente com as camadas que a compõem. A seção IV é apresentada a especificação parcial em Object-Z, da classe do agente de descoberta e do módulo de auto-otimização e na seção V contém a conclusão e os trabalhos futuros.

II. TRABALHOS RELACIONADOS

De forma a mostrar o estado-da-arte sobre as arquiteturas e abordagens auto-organizadas em redes em malha sem fio, são considerados alguns trabalhos.

Devido ao crescimento em número de dispositivos e serviços que fazem parte das redes sem fio atuais, alguns esforços estão sendo feitos para que o processo de instalação, gerenciamento e manutenção dessas redes não se tornem uma dor de cabeça para o administrador da mesma. Algumas pesquisas no sentido de que a rede possua autonomia em suas decisões estão sendo desenvolvidas.

Em [8], os autores apresentam um projeto de computação autônoma que demonstra que a capacidade de auto-gerenciamento pode ser alcançada de maneira evolutiva, e prescreve-se as iniciativas para aplicação dessa capacidade

Lucas Guardalben, João Bosco M. Sobral e Clytia H. Tamashiro, Universidade Federal de Santa Catarina - UFSC, Programa de Pós-Graduação em Ciência da Computação Grupo de pesquisa DMC-NS (Distributed Mobile Computing and Network Security) Florianópolis-SC, E-mails: guardalben,bosco,clytia@inf.ufsc.br.

em ambientes heterogêneos. A complexidade em configurar as políticas de gerenciamento é uma desvantagem eminente nesta abordagem.

Algumas arquiteturas são propostas para redes em malha sem fio, em Raniwala et al. [9] apresenta uma arquitetura multi-canais chamada *Hyacinth*, esta arquitetura é composta por nodos sem fio equipados com múltiplas interfaces de rede (NIC's) no padrão 802.11. Os autores demonstram um modelo inteligente para escolha dos melhores canais utilizando múltiplos rádios. O alto custo de implantação dessa abordagem em larga escala inviabiliza o projeto, devido a necessidade de hardware dedicado para agregar o tráfego da rede. Além de não enfatizar a auto-organização em nível de camada de rede.

Já neste trabalho [10], Wu considera que as redes em malha sem fio são boas candidatas a resolver o problema da última milha para serviços de banda larga. As redes em malha sem fio se deparam com o problema da escalabilidade devido às características de auto-organização e conexões através de múltiplos saltos. Para este contexto o autor propõe uma arquitetura hierárquica para redes em malha sem fio. Esta abordagem consiste em dividir dentro de pequenas sub-redes, o qual é operado independentemente e são conectadas através de um *backbone* confiável. Um problema encontrado em abordagens utilizando arquiteturas hierárquicas, é que o *cluster head* eleito para ser o nodo cabeça dentro de um subgrupo de redes, se torna um ponto único de falha, além de necessitar políticas bem definidas para eleição de um novo nodo cabeça em caso de falhas. Neste caso, abordagens utilizando arquiteturas híbridas são mais tolerantes a falhas.

A auto-organização é um assunto promissor para as próximas gerações de redes sem fio, neste sentido muitas lacunas devem ser preenchidas em direção a auto-organização completa da rede. Analisando os trabalhos relacionados, conclui-se que somente algumas soluções enfatizam o problema da auto-organização em redes em malha sem fio. Neste caso, encoraja-nos a desenvolver uma arquitetura híbrida para prover uma primeira experiência a baixo custo de instalação e manutenção e com políticas de gerenciamento simples e bem definidas, juntamente com as principais capacidades de: auto-otimização, auto-cura, auto-configuração e auto-proteção da rede.

III. ARQUITETURA PARA AUTO-ORGANIZAÇÃO

O termo auto-organização foi primeiramente reportado em sistemas biológicos e adaptado para diferentes áreas de aplicação[11], tais como: engenharia, telecomunicações e computação. Neste artigo é considerada a seguinte definição:

Os nodos sem fio auto-organizados são entidades responsáveis por executar tarefas requeridas pela aplicação de uma forma eficaz e sem intervenção humana, obtendo como benefício a redução do custo de instalação. [4].

Em geral, redes podem ser auto-organizadas conforme o modelo de referência ISO/OSI em duas camadas [12]: aplicação e rede. E também algumas topologias suportam modelos auto-organizados como: plana, hierárquica e híbrida. Neste trabalho é considerado a topologia híbrida, e a auto-organização na camada de rede. A arquitetura consiste de

agentes que coletam informações a respeito dos nodos-clientes e nodos-roteadores e juntamente com módulos criados para dinamizar as funções da rede. Esses módulos são denominados auto-x (*self-x*) [13] [8], que são **otimização, cura, proteção, configuração**. O protocolo escolhido para formar a base de conectividade da rede em malha sem fio foi o Mobile Mesh, o qual é formado por três sub-protocolos com funções específicas: descoberta de nodos-clientes e roteadores, roteamento e descoberta de nodos-roteadores de borda. Os três sub-protocolos são descritos a seguir:

A. Mobile Mesh Link Discovery Protocol (MMLDP)

MMLDP é baseado em um protocolo tradicional de "Hello". Cada interface de comunicação roda o protocolo e envia mensagens periodicamente contendo seu endereço IP (*Internet Protocol*) e os endereços de seus vizinhos, isso ocorre em um certo período de tempo. Com esse processo os nodos mantêm atualizadas as informações dos enlaces da rede, juntamente com as informações de estado de conectividade dos enlaces dos seus vizinhos;

B. Mobile Mesh Routing Protocol (MMRP)

MMRP é baseado na técnica de roteamento denominada "link-state" e permite construir rotas de menor custo entre o nodo origem e o nodo destino, utilizando informações dos nodos pertencentes à rede. As informações são encapsuladas em pacotes denominados *Link State Packets* (LSP). Um pacote LSP é dividido da seguinte forma [14]:

- **Versão e Tipo de Mensagem:** Denota a versão do protocolo (versão atual 1.0) e o tipo de mensagem, o qual garante que mensagens que não sejam do tipo LSP sejam identificadas;
- **Identificador de Rota *Route-id*:** é um número único de 32 bits, que identifica a origem do LSP. Esse número é o endereço IP da interface de rede;
- **Número Sequencial:** Este campo é usado para distinguir as mensagens LSP's mais recentes das mais antigas. E em tempos em tempos o nodo gera um novo pacote LSP.
- **Idade:** usado para configurar o tempo máximo de vida de uma mensagem LSP na rede;
- **Contador de Saltos (hops):** este campo é responsável por manter atualizado o número de saltos de uma mensagem LSP até chegar ao nodo destino;
- **Lista de Interfaces Locais:** indica quantas interfaces vizinhas ao nodo origem está participando da rede, incluindo a interface do próprio nodo origem;
- **Rotas Externas:** este campo mostra a lista de rotas externas que podem ser adicionadas a tabela de roteamento. Este campo se aplica aos roteadores de borda.

Cada nodo mantém uma base de dados contendo a mais recente mensagem LSP de cada nodo origem. Quanto um nodo recebe uma mensagem LSP ele apresenta a seguinte lógica de processo, conforme é observado na Fig. 1.).

Para aumentar a escalabilidade da rede, o módulo MMRP dispõe de uma técnica denominada *olho de peixe*, o nodo percebe com alta definição os seus nodos vizinhos e conforme

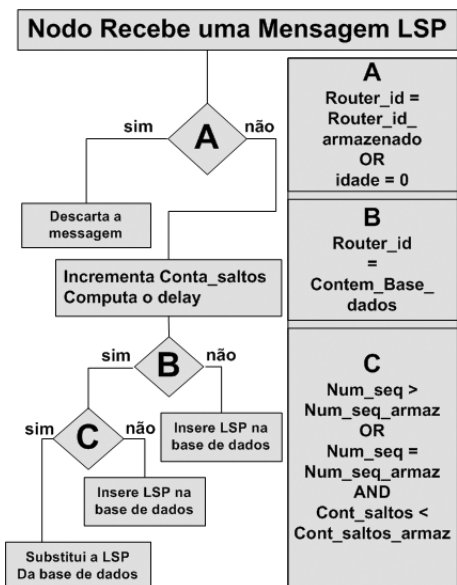


Fig. 1. O nodo recebe uma mensagem LSP e testa se o identificador de rota é o mesmo do pacote que já está armazenado no nodo, ou se a idade é igual a zero, se isso for verdade o nodo descarta a mensagem. Caso contrário o nodo incrementa o campo contador de saltos e estima o tempo do atraso imposto antes de enviar uma mensagem LSP. Se o nodo não possui uma mensagem LSP armazena em sua base de dados, o nodo insere a mensagem LSP dentro da sua base de dados. O número sequencial é usado para testar se a mensagem LSP recebida é maior ou igual do que a mensagem já armazenada na base de dados do nodo que recebeu o pacote e se também contador de saltos recebido de uma mensagem LSP é menor que o contador de saltos armazenado. Se isso for verdade então a mensagem que foi recebida substitui a mensagem mais antiga da base de dados. Esse processo garante que mensagens mais atuais sejam gravadas na base de dados, garantindo assim uma tabela de roteamento sempre atualizada

aumenta a distância entre eles a sua definição vai diminuindo. Esta técnica permite controlar as mensagens de LSP de forma a não inundar a rede.

C. Mobile Mesh Border Discovery Protocol (MMBDP)

Este sub-protocolo é utilizado em cenários onde há necessidade de conexão com a rede cabeada, e provê um mecanismo para descoberta dos nodos usando BorderAdvertisement e com função para conexão a rede cabeada. Isto permite aumentar a capacidade da rede.

Os parâmetros dos subprotocolos devem ser configurados de forma a otimizar o processo de cômputo da rotas. Os fatores que podem influenciar os parâmetros são: o tamanho da rede, taxa de mobilidade, distância de transmissão, força do sinal de transmissão, sobrecarga da rede, jitter e atraso. O ato de configurar parâmetros em redes complexas necessita de um maior acompanhamento do administrador da rede de forma que os valores não comprometam o desempenho da mesma. Na maioria dos casos isso não acontece, o administrador da rede por falta de um conhecimento macro da rede acaba deixando valores padrões que em determinados estágios da rede podem influenciar drasticamente em seu desempenho. Com a baixa intervenção do administrador, a rede se torna mais autônoma e auto-organizada, diminuindo assim custos de sua manutenção e instalação. Deixando a cargo do administrador da rede a tomada de decisão somente em níveis mais críticos.

Para melhor explicar a lógica de interação envolvida neste trabalho, desenvolveu-se a arquitetura baseada em camadas. Abordagens baseadas em camadas garantem flexibilidade e simplicidade no desenvolvimento, no sentido que todas as funções envolvidas são bem definidas e tratadas de forma independente, uma vantagem em relação a detecção de falhas isoladas. O modelo de interação entre as camadas pode ser observado na Fig. 2.

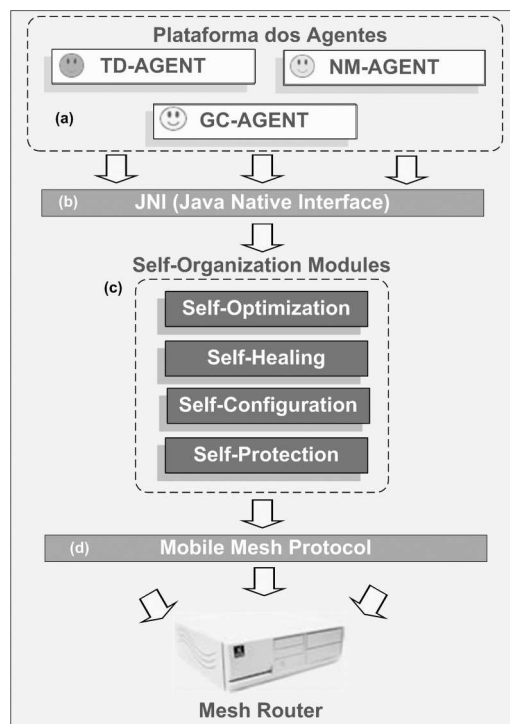


Fig. 2. Arquitetura de Auto-Organização para um nodo-roteador

Os agentes que compõem a camada superior possuem um papel específico na arquitetura e são classificados em:

- (A) **TD-AGENT**: são agentes móveis que são criados nos nodos-roteadores e a cada pedido de associação de um nodo-cliente para um nodo-roteador, o agente migra para o nodo-roteador e inicia o processo de aquisição de informações: (IP, MAC, ativo ou não, nível da bateria e se possui mobilidade) do nodo-cliente. Os nodos clientes podem ser tanto móveis quanto fixos, e diferenciam-se pelo tipo do dispositivo: classe de dispositivos móveis (PDA's, telefones celulares, laptops), classe de dispositivos fixos (computadores desktops, nodos-roteadores ou nodos-roteadores de borda).
- (A) **NM-AGENT**: são responsáveis por monitorar funções específicas da rede. São agentes estáticos nos nodos-roteadores que servem como suporte ao processo de auto-otimização da rede.
- (A) **GC-AGENT**: são agentes que são responsáveis pela comunicação entre os nodos-roteadores para formar a descrição global da topologia e determinar a densidade da rede. Esses agentes são estáticos nos nodos-roteadores e trocam mensagens entre si para auxiliar na configuração dos parâmetros do sub-protocolo MMRP.

(B) Java Native Interface (JNI): esta camada é responsável por intermediar a plataforma JADE-LEAP [15] (desenvolvida em linguagem Java) com o protocolo Mobile Mesh (desenvolvido em linguagem C/C++).

(C) Auto-Otimização: Este módulo é responsável por otimizar os parâmetros do sub-protocolo MMRP.

(C) Auto-Cura: este módulo é responsável por detectar, localizar e reparar falhas de forma transparente para o usuário;

(C) Auto-Configuração: responsável por configurar, o endereço IP(Internet Protocol) de cada nodo-roteador ou nodo-cliente que é associado a rede;

(C) Auto-Proteção: responsável por alertar o administrador da rede, sobre alguns possíveis ataques a segurança da rede;

(D) Protocolo Mobile Mesh: Com as informações obtidas através dos agentes de descoberta(TD), monitoramento (NM) e comunicação (GC), elas serão utilizadas para alimentar o arquivo de configuração (mmrp.conf) do sub-protocolo de roteamento MMRP. Entre os parâmetros do protocolo de roteamento que podem ser otimizados estão:

- **métrica:** O custo associado a cada interface;
- **alpha:** este parâmetro controla a soma do atraso antes do envio de uma mensagem LSP, onde a variação do valor é obtida através do intervalo (1):

$$(1.0 \leq \text{valor} \leq 2.0); \quad (1)$$

A estimativa do atraso é obtida através da formula (2):

$$\text{Max}(\text{minTxDelay}, \text{Min}(\text{maxTxDelay}, (\alpha^h))) \quad (2)$$

Onde h é a quantidade de saltos que uma mensagem LSP usa para encontrar o nodo destino;

- **minTxDelay:** o atraso mínimo, em segundos, que seria colocado antes de enviar um LSP ao nodo vizinho;
- **maxTxDelay:** o atraso máximo, em segundos, que seria colocado antes de enviar um LSP ao nodo vizinho;
- **maxAge:** o número máximo de segundos que uma mensagem LSP possa estar ativa na rede, antes de ser descartada;
- **updatePeriod:** o número de segundos entre envio do LSP do nodo origem para atualizar a tabela de roteamento.

Algumas fases são indentificadas em direção a auto-organização da rede, dentre elas: **Descoberta, Organização, Manutenção e Reorganização**, o qual são detalhadas em [12]. Para uma primeira experiência enfatiza-se as duas primeiras.

Neste contexto é proposto uma melhora nos métodos de descoberta e organização originalmente proposto pelos sub-protocolos MMLDP e MMLDBP e MMRP, a qual são descritas a seguir:

- **Descoberta e Descrição Global da Rede:** uma das desvantagen aparentemente no método de reconhecimento dos sub-protocolos MMLDP e MMLDBP, é que utilizam a técnica de inundação de mensagens afim de conhecer os nodos envolvidos na rede. Esta técnica muitas das vezes pode gerar um grande acúmulo de mensagens de reconhecimento, reduzindo assim a performance de algumas aplicações. A melhora se baseia no pré-conhecimento do

cliente que deseja se registrar na rede, ou seja, quando um cliente está tentando associação a um nodo-roteador. O agente (TD) migra para o suposto cliente e verifica qual o tipo de dispositivo (móveis ou fixos) que estão pedindo para se registrar, com esse processo evita-se que os nodos-roteadores façam inundação de mensagem de descoberta de uma forma desordenada na rede, aumentando a quantidade de pacotes e supostamente reduzindo a vazão ¹. O GC_AGENT é responsável por manter sempre atualizada a tabela de roteamento em caso de queda de algum nodo-roteador, desta forma há sempre comunicação direta entre os nodos-roteadores.

- **Organização:** antes de enviar uma mensagem LSP o sub-protocolo estima um tempo de atraso de forma empírica, isso é uma desvantagem pois é difícil estimar um tempo ideal para se calcular o atraso, podendo influenciar na hora de atualizar a tabela de roteamento. A melhora consiste em estimar o tempo de atraso utilizando informações advindas do agente NM_AGENT responsável pelo monitoramento do atraso e a sobrecarga em determinados caminhos da rede.

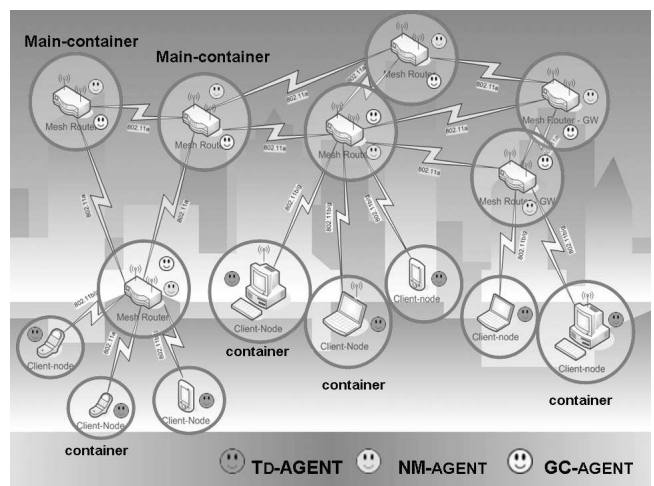


Fig. 3. Cenário para Aplicação da Arquitetura

Na Fig. 3. Pode ser observado um cenário metropolitano afim de exemplificar a aplicação da arquitetura. Para implementar os agentes, usou-se a plataforma JADE-LEAP. Nos nodos roteadores foram instalados *main-containers*. Já nos nodos-clientes são executadas instâncias de JADE-LEAP denominada *container*, o qual são subordinados ao *main-container* mais próximos. Isso permite ter flexibilidade na migração dos agentes e padronização da comunicação entre os mesmos.

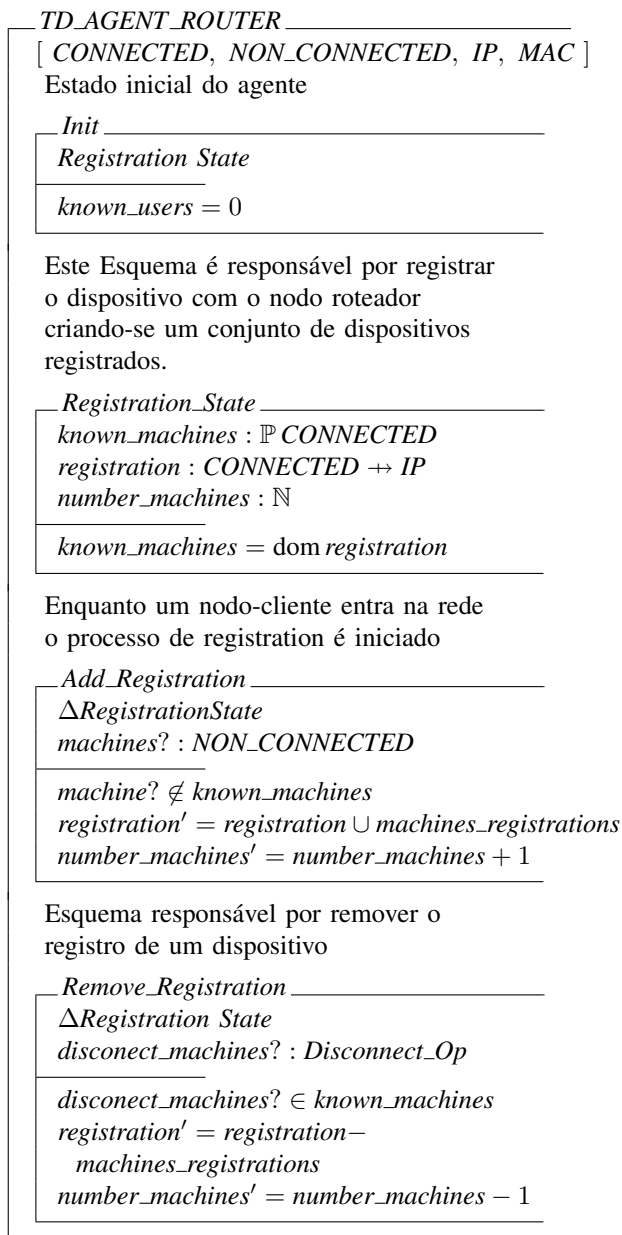
IV. ESPECIFICAÇÃO DA ARQUITETURA

A fim de ter uma abstração em alto nível da arquitetura, utilizou-se a linguagem de especificação formal Object-Z. A arquitetura é definida basicamente pela interação do sub-protocolo MMRP e os agentes de captura de informações da rede, trabalhando juntamente com os módulos auto-x.

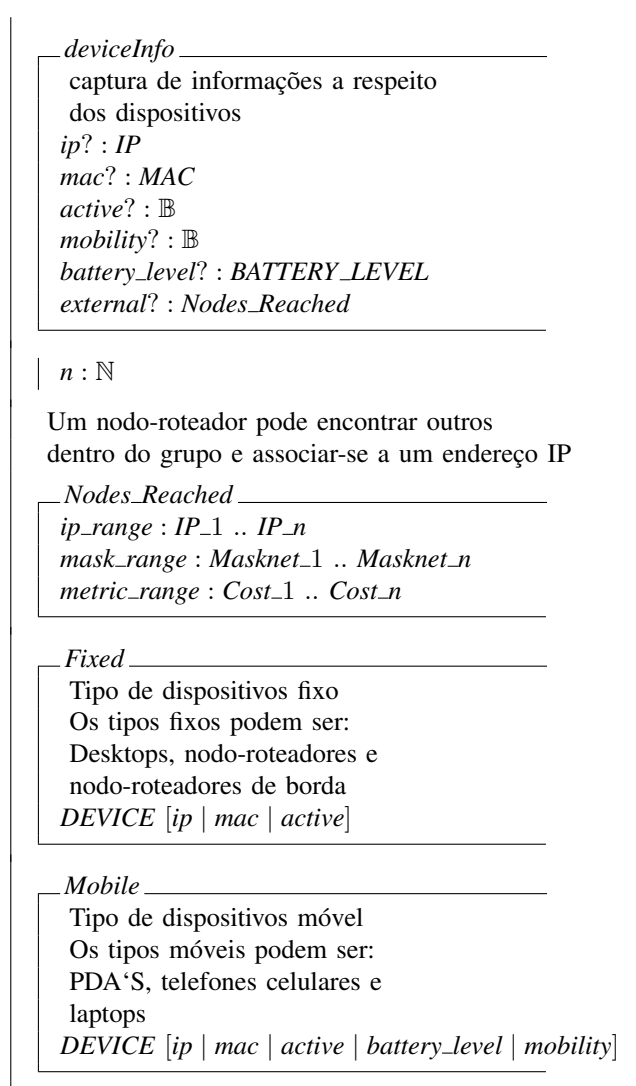
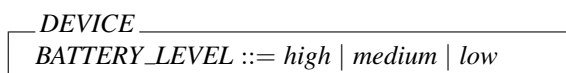
¹É a quantidade de requisições processadas, por um certo período de tempo.

A especificação em primeira instância demonstra como o método de registro (*registration*) conhece o cliente que está tentando se associar a um nodo-roteador. Neste processo o nodo-roteador tem um conjunto maior de informações que são captadas dos nodos clientes. Ao contrário dos sub-protocolos (MMLDP e MMLDBP) do Mobile Mesh, que possuem como referência somente o endereço da interface (IP). A idéia chave é conhecer o cliente na hora que ele irá se associar ao nodo-roteador. Uma versão parcial da especificação é apresentada a seguir:

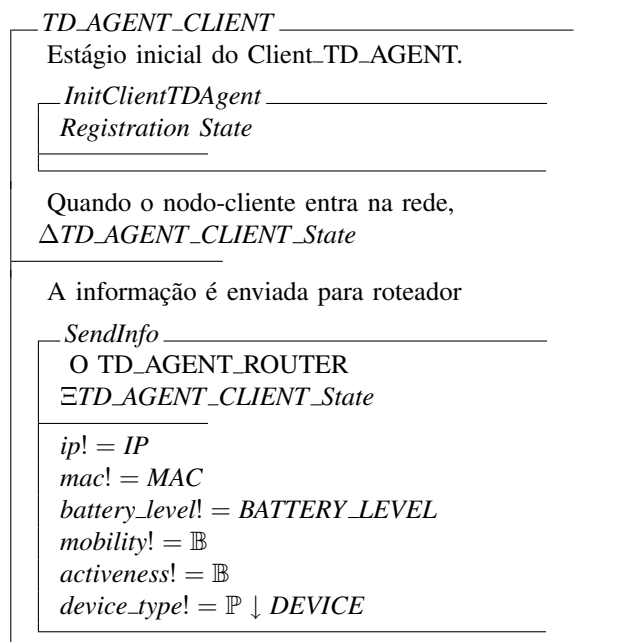
A classe TD_AGENT_ROUTER recebe as informações oriundas da classe TD_AGENT_CLIENT.



Classe DEVICE é responsável por conhecer o tipo de dispositivo a se registrar.



A classe TD_AGENT_CLIENT informações capturadas dos dispositivos e as envia para a classe TD_AGENT_ROUTER.



Classe responsável pela otimização das rotas através de informações reais do agente de monitoramento.

Self_Optimization

MMRP

$\alpha? : \mathbb{R}$
 $delay? : NM_AGENT.delay$
 $min_Tx_delay? : NM_AGENT.mindelay$
 $max_Tx_delay? : NM_AGENT.maxdelay$
 $update_period? : \mathbb{N}$
 $max_age : \mathbb{N}$
 $router_id : \mathbb{N}$
 $age : \mathbb{N}$
 $sequence_number : \mathbb{N}$
 $hop_count : \mathbb{N}$

$(router_id = router_id_storage) \vee$
 $(age = 0)$
 $discard_msg' = true$
 $\neg (router_id = router_id_storage) \vee$
 $(age = 0)$
 $hop_count : hop_count + 1$
 Computa o atraso, esta informação é vinda do agente de monitoramento, ao invés de ser estimado pelo MMRP.
 $delay : Max(mideTx_delay,$
 $Min(max_Tx_delay, (\alpha^h op_count)))$

V. CONCLUSÃO

Neste trabalho é proposta uma arquitetura para auto-organização em redes em malha sem fio utilizando como base o protocolo Mobile Mesh juntamente com a tecnologia de agentes de software para ajudar os módulos de auto-organização na tomada de decisão. A análise do problema foi concretizada através da especificação formal em Object-Z, no sentido de posteriormente facilitar a implementação da arquitetura em uma linguagem orientada a objeto como o JAVA. Revisando a literatura especializada ainda existem poucas pesquisas em auto-organização para redes em malha sem fio. É aceito neste contexto que a auto-organização da rede pode reduzir a intervenção do administrador para sua configuração, gerenciamento e manutenção, deixando assim a cargo do administrador processo de decisões de mais alto nível. A arquitetura proposta envolve a definição de camadas que dão suporte a agentes de software que auxiliam nas tarefas de descoberta da topologia, descrição global da topologia e monitoramento. Como primeira experiência são enfatizadas duas vantagens em relação aos sub-protocolos MMLDP, MMLDBP e MMRP: (1) a especificação da descoberta e descrição global da topologia, baseado no pré-conhecimento dos nodos clientes a serem registrados na rede. (2) e a estimativa de atraso e latência para agilizar a descoberta e atualização das rotas baseada no agente de monitoramento. Alguns aspectos práticos são visíveis para se construir uma rede em malha sem fio. Os nodos roteadores podem ser montados através de componentes

off-the-shelf das mais conhecidas tecnologias existentes no mercado tais como: computadores (PC's e roteadores mesh), e o uso de software livre (Linux, Java, Mobile Mesh) juntamente com dispositivos comercialmente disponíveis (PDA, laptops, telefones celulares). O padrão de redes sem fio mais utilizado é o 802.11 e suas variantes, este padrão possui uma série de limitações, quando a quantidade de dispositivos aumenta na rede. Padrões alternativos como 802.16 (WiMAX), estão sendo testados e utilizados em redes em malha sem fio para constituírem o *backbone* da malha de roteadores. Para uma primeira experiência optou-se em utilizar as principais capacidades da auto-organização (otimização, cura, configuração e proteção), definindo uma arquitetura mais simplificada para alcançar o objetivo da auto-organização em redes em malha sem fio.

Neste trabalho, observa-se a necessidade de validar a arquitetura proposta empregando-a em um ambiente real. Entretanto, deve ser proposta uma extensão para o protocolo Mobile Mesh para operar com múltiplas interfaces e múltiplos canais, a fim de melhor suportar a demanda de tráfego existente na rede. Além disso, para aumentar a qualidade da entrega dos pacotes é necessária uma métrica mais eficiente da que atualmente é utilizada pelo protocolo Mobile Mesh, pois é utilizada contagem dos saltos (*hop count*) para definição da métrica de roteamento, em muitos casos, o menor caminho não é o melhor caminho a ser seguido pelo pacote, com isso deve ser levado em consideração alguns fatores: como perda de pacotes e a sobrecarga da rede.

REFERÊNCIAS

- [1] Microsoft, "Self-Organizing Neighborhood Wireless Mesh Networks," 2005. [Online]. Available: <http://research.microsoft.com/mesh/>
- [2] Motorola, "Multi-Radio Broadband Solution," 2005. [Online]. Available: <http://www.motorola.com/mesh/pages/products/motomesh.htm>
- [3] I. F. Akyldiz, X. Wang, and W. Wang, "Wireless Mesh Networks: a survey," *IEEE Computer Networks*, vol. 47, p. 445 a 487, March 2005.
- [4] S. Vasudevan, "Self-Organization in Large-Scale Wireless Networks," Ph.D. dissertation, University of Massachusetts - Dept Computer Science, September 2006.
- [5] G. Weiss, *Multiagent System: A Modern Approach to Distributed Artificial Intelligence*. London: MIT Press, 1999.
- [6] M. Corporation, "Mobile Mesh," 2005. [Online]. Available: http://www.mitre.org/work/tech_transfer/mobilemesh/
- [7] R. Duke, P. King, G. Rose, and G. Smith, "The Object-Z Specification Language," Department of Computer Science - University of Queensland 4072 - Australia, Tech. Rep., 1991.
- [8] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," in *IEEE Computer Networks* 36, vol. 1, 2003, pp. 41-50.
- [9] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network," *24th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)*. *Proceedings IEEE*, vol. 3, pp. 2223-2234, March 2005.
- [10] X. Wu, "Hierarchical MESH Architecture: Toward Practical Applications," *iwnas*, pp. 69-72, 2006 *International Workshop on Networking, Architecture, and Storages (IWNAS'06)*, 2006.
- [11] M. Eigen and P. Schuster, *The Hypercycle: A principle of Natural Self Organization*. Berlin: Springer-Verlag, 1979.
- [12] H. Tang and H. Tianfield, "Self-Organizing Networks of Communications and Computing," in *International Transactions on Systems Science and Applications*, vol. 1, no. 4, 2006, pp. 421-431.
- [13] O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. van Moorsel, and M. van Steen, *SelfStar Properties in Complex Information Systems*. Springer, 2005.
- [14] K. Grace, "Mobile mesh routing protocol," Mitre Corporation," draft-grace-manet-mmrip-00.txt - work in progress, September, 2000.
- [15] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley, 2007.