

# Arquitetura de Comunicação para Cidades Inteligentes: Uma proposta heterogênea, extensível e de baixo custo\*

Edson Adriano M. Avelar, Lorena M. Avelar, Diego Dos Passos Silva, Kelvin Lopes Dias

Centro de Informática – Universidade Federal de Pernambuco (UFPE)  
Av. Jornalista Aníbal Fernandes, s/n – 50.740-560 – Recife – PE – Brasil

{eama, l1m, dps4, kld}@cin.ufpe.br

**Abstract.** *This paper proposes heterogeneous communication architecture for smart cities. The proposal includes the implementation of adaptive systems that allows communication between sensor networks (Sun SPOT / Xbee) based on ZigBee/IEEE 802.15.4 and WLAN based on IEEE 802.11 operating in mesh mode. Furthermore, the proposal also includes a message-oriented middleware for communication between the Sun SPOT technology and OpenWRT platform. A network testbed was built to monitor noise pollution, temperature and luminosity. Thus, it is possible to create an extensible, robust and low cost architecture, using open source software (OpenWRT) on Linksys routers and open platforms for sensors (Arduino / XBee / Sun SPOT).*

**Resumo.** *Este trabalho propõe uma arquitetura de comunicação heterogênea para cidades inteligentes. A proposta inclui a implementação de sistemas de adaptação que possibilita a comunicação entre redes de sensores (SunSPOT/Xbee) e redes IEEE 802.11 em modo mesh. Além disso, a proposta também inclui um middleware orientado a mensagem para comunicação entre o Sun SPOT e o sistema OpenWRT. Um testbed foi construído para o monitoramento de poluição sonora, temperatura e luminosidade. Desta forma, é possível criar uma arquitetura extensível, robusta e de baixo custo, utilizando software open source (OpenWRT) em roteadores Linksys e plataformas abertas para sensores (Arduino/XBee/Sun SPOT).*

## 1. Introdução

Todos os dias uma enorme quantidade de informações é oferecida ao cidadão por diversos meios. Grande parte dessas informações é apresentada implicitamente ou de forma bruta e incompreensível aos nossos olhos. Portanto, faz-se necessário o desenvolvimento de mecanismos capazes de capturar, tratar, disseminar e utilizar essas informações como base para tomada de decisões. O conceito de Cidades Inteligentes vai ao encontro desses anseios. Cidades Inteligentes são definidas como o uso de Tecnologias da Informação e Comunicação para verificar, analisar e integrar as informações-chaves de sistemas centrais em cidades [Kehua 2011].

Empresas e centros de desenvolvimento de tecnologia reconhecem a Internet como grande veículo de disseminação de informações e agregam cada vez mais dispositivos diferentes à rede, como televisores, geladeiras e até automóveis. Essa onipresença de fontes de dados agregados à rede é o objetivo da Internet das Coisas (IoT - *Internet Of Things*) [Tan and Wang 2010]. A IoT é um modelo de computação

\*Esta pesquisa foi financiada com recursos de projeto aprovado no Edital RNP/CTIC 2010 Cidades Inteligentes (Smart Cities).

baseado em ubiquidade no qual qualquer objeto ou coisa é uma fonte de dados que pode ser acessado de qualquer lugar e a qualquer momento.

Além da RFID (*Radio Frequency Identification*), as Redes de Sensores sem Fio (RSSF) também são de vital importância para a IoT e para a infraestrutura de comunicação das Cidades Inteligentes. Uma RSSF pode ser implementada através de *motes*, sistemas embarcados simples, de baixo custo, com habilidade de sensoriamento e limitada capacidade de processamento e comunicação [Al-Karaki and Kamal 2004]. Os principais *motes* do mercado são desenvolvidos seguindo o padrão IEEE 802.15 que provê especificações para as camadas básicas da pilha de protocolos, camada física (PHY) e de enlace/acesso ao meio (MAC). Para as camadas superiores, cada fabricante possui ou adota uma solução de protocolos específicos. Por isso, geralmente, *motes* de diferentes fabricantes não interagem entre si. Em alguns casos, não há interoperabilidade sequer entre *motes* de mesmo fabricante e séries diferentes.

Nos cenários futuros de Cidades Inteligentes e IoT devem-se considerar a possibilidade de projetos e aplicações que se beneficiem de redes heterogêneas de forma transparente e escalável. Em uma RSSF, a estação-base possui o papel fundamental de coleta de dados advindos dos nós sensores, bem como entrega desses dados ao usuário final, normalmente, conectado de forma direta a um computador. Na abordagem prevista neste trabalho será desenvolvido um modelo de estação-base híbrida capaz de obter dados de uma RSSF, composta por *motes* de diferentes fabricantes, e disseminá-los através de um *backbone mesh* baseado em uma solução de baixo custo que utiliza roteadores comerciais padrão IEEE 802.11. Características desejáveis nas redes *mesh* são auto-organização e autoconfiguração, o que possibilita a manutenção das conexões de forma automática, visando a inclusão de novos roteadores na rede para o aumento da área de cobertura da rede *mesh* [Akyildiz et al. 2005].

Este trabalho está organizado da seguinte forma: na Seção 2 são apresentados os projetos que realizam experimentos em Cidades Inteligentes. Na seção 3 é apresentada a arquitetura proposta pelo artigo. Na seção 4 são exibidos os resultados experimentais da arquitetura proposta e, por fim, a Seção 5 aborda as principais discussões e considerações finais obtidas no presente artigo.

## 2. Trabalhos Relacionados

Existem vários trabalhos na literatura que tratam de experimentos em cidades inteligentes. Porém, diferente deste trabalho, a grande maioria utiliza tecnologias de alto custo, prioritária e não extensível. A seguir serão discutidos os principais trabalhos.

A proposta descrita em [Smartsantander 2010] é uma plataforma aberta e em larga escala para testar tecnologias ligadas a Internet das Coisas. A infraestrutura do SmartSantander é formada pelas entidades: *IoT Node*, *Repeaters* e *Gateways*. O primeiro e segundo fazem a coleta dos dados e o terceiro é a porta de saída da rede para Internet. A ideia do projeto é fornecer em uma interface Web informações como temperatura, concentração de CO, luminosidade.

O Projeto SOFIA [Filipponi 2010] consiste em uma arquitetura para IoT e o *testbed* da solução foi implantado em um cenário de metrô. A arquitetura possui uma detecção de eventos anormais que simplifica o trabalho dos operadores e a comunicação entre os passageiros. O trabalho apresenta sua arquitetura com os seguintes elementos:

KP (*Knowledge Processors*) é o módulo central, o cérebro da arquitetura. Esse módulo faz consultas na base de dados e trata notificações, o SIB (*Semantic Information Broker*) é o módulo da base de dados e SSAP (*Smart Space Access Protocol*) é o protocolo de comunicação entre o KP e o SIB. O SOFIA é uma arquitetura orientada a eventos (*Event Driven Architecture*). Os eventos são descritos através de ontologias e seguem o padrão *publish/subscriber*.

O Projeto CitySense [Bers 2010] consiste em vários sensores que monitoram o tempo e a poluição do ar. Cada nó possui um *hardware* com grande capacidade de processamento, armazenamento e duas interfaces Wi-Fi, uma para plano de controle e outra para plano de dados. O Projeto Oulu Smart City [Ojala 2010] cria a interação entre espaços urbanos, o usuário e a tecnologia da informação. Uma grande rede é espalhada pela cidade e os usuários coletam dados por meio de smartphones usando a tecnologia IEEE 802.11 (Wi-Fi)

Os projetos Motescope [UC Berkeley 2010] e Flocklab [Beutel 2009] utilizam uma larga rede de sensores para monitoramento avançado de algumas funcionalidades, como o consumo de energia. O Motescope utiliza sensores heterogêneos e serve também para testar novos protocolos.

O projeto Friedrichshafen [German Telekom and City of Friedrichshafen 2010] engloba a utilização de tecnologias como PDAs, *smarphones* e sensores para avaliação experimental de serviços como *e-learning*, *e-health* e *e-tourism*.

Na literatura, existem outros projetos que usam redes de sensores em larga escala para teste com propósitos variados, testar redes de sensores, fazer experimentos em IoT, entre outros. São eles: o Motelab [Werner-Allen 2005], TutorNet [TutorNet 2009], Kansei [Arora 2006], SensLAB [SensLAB 2010], VineLab [Univ. of VA 2009] dentre outros.

Contudo, a maioria dos projetos apresentados utilizam equipamentos de alto custo e que não utilizam a tecnologia *mesh* nos gateways. Utilizando *mesh* é possível criar arquiteturas flexíveis e extensíveis, uma vez que para aumentar o alcance da arquitetura basta inserir outro nó *mesh* sem precisar reconfigurar toda a rede. O único projeto que utiliza rede *mesh* como tecnologia de *backbone* é o projeto CitySense, porém os equipamentos utilizados são compostos por sistemas fechados, que não podem ser modificados e possuem um custo bastante elevado. Neste artigo [Bers 2010], o autor menciona que um nó, com placa ALIX 2d2 e 4G de armazenamento, contendo um sensor Vaisala WXT520 custa acima de dois mil dólares.

A proposta deste trabalho é criar uma arquitetura extensível com roteadores de baixo custo (Linksys WRT54GL), *software open source* (OpenWRT) e plataformas heterogêneas para sensores também de baixo custo e abertas ((Arduino/XBee)/Sun SPOT) fornecendo um arcabouço simples, de baixo custo e robusto. A seguir, será apresentada a arquitetura proposta.

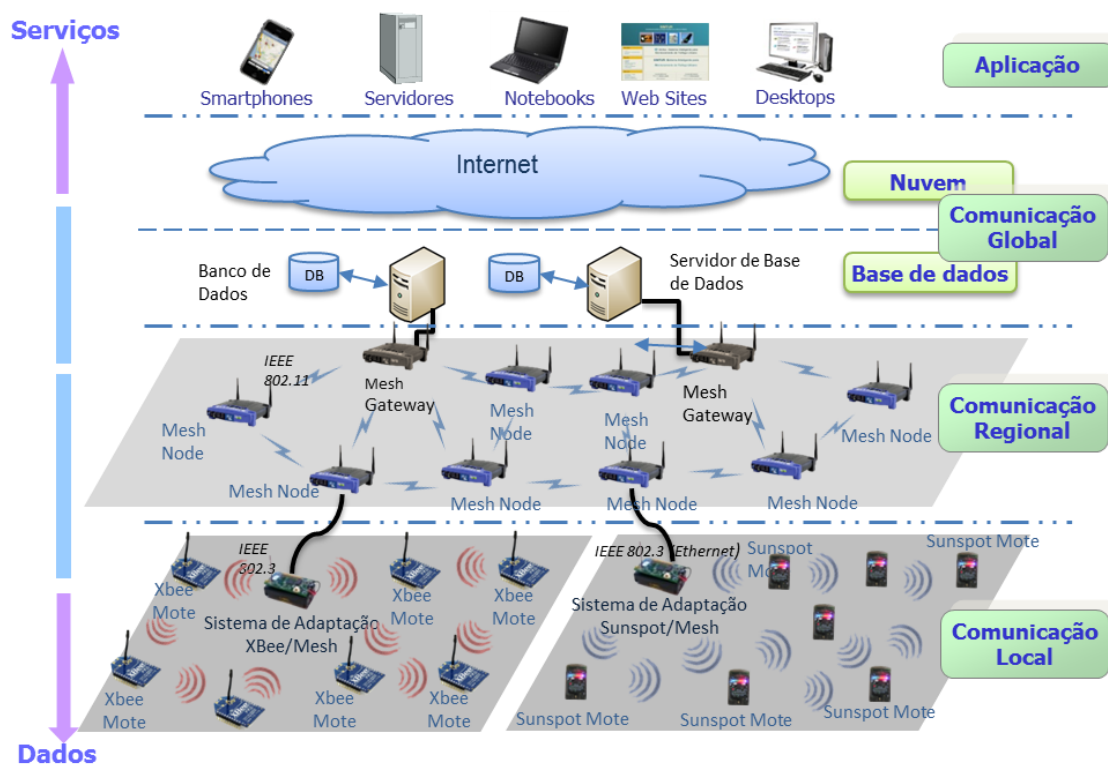
### **3. Arquitetura da proposta**

Os maioria dos experimentos em IoT seguem uma estrutura de duas ou três camadas [Gluhak 2011]. Estruturas com duas camadas possuem o primeiro nível formado pelos sensores e o segundo pelos servidores. A camada de sensores é onde os dados são coletados e a camada de servidores é onde os dados são armazenados e utilizados pelas

aplicações. A arquitetura de duas camadas é usada para testes em pequena escala. Já as estruturas com três níveis possuem uma camada intermediária entre a camada de sensores e de servidores. Ela é responsável por aumentar a escalabilidade da arquitetura, sendo geralmente usadas as tecnologias *Wi-Fi*, Fibra Ótica, Ethernet. Esta arquitetura é utilizada para grandes *testbeds*.

Uma das vantagens em se trabalhar com modelos em camadas é o melhor entendimento da proposta uma vez que os problemas serão divididos em grupos menores. Neste artigo, a camada de aplicação é acrescentada para melhor entendimento da proposta, mas não entra na classificação baseada em níveis.

A proposta deste trabalho, mostrada na Figura 1, é utilizar uma arquitetura de três-níveis onde o nível intermediário é desenvolvido utilizando a tecnologia *mesh* 802.11. É desenvolvido também um sistema embarcado de adaptação para que os sensores heterogêneos possam se comunicar com a rede *mesh* e vice-versa.



**Figura 1. Arquitetura da proposta e suas camadas.**

A divisão em camadas utilizada na arquitetura (Figura 1) tem o objetivo de simplificar a abordagem e criar uma arquitetura modular e de fácil extensão. A seguir, a descrição das camadas utilizadas.

- Camada de Comunicação Local:** As leituras dos sensores serão realizadas na camada de comunicação local. Esta camada também tem a função de transportar os dados dos sensores em uma ambiente local, ou seja, de curto espaço (cerca de 10m de sensor para sensor). Para utilizar uma abordagem heterogênea, serão utilizados dois tipos de redes de comunicação local, uma com a tecnologia XBee que implementa o ZigBee/IEEE 802.15.4 e outra com a tecnologia Sun SPOT

(Oracle). Serão utilizados vários sensores nesta camada (luminosidade, captação sonora e temperatura).

- **Camada de Comunicação Regional:** O objetivo desta camada é transportar os dados advindos dos sensores para a Internet. Ela é formada por roteadores com tecnologia IEEE 802.11 habilitados para comunicação *mesh*. Neste trabalho utilizou-se uma versão modificada do OpenWRT em roteadores Linksys para possibilitar roteamento *mesh*.
- **Camada de Comunicação Global:** Esta camada possibilita que os dados coletados sejam disponibilizados para a Internet. Para melhor entendimento, ela é subdividida em duas:
  - **Subcamada da Base de Dados:** Onde ficam armazenados todos os dados coletados pelos sensores. Nela residem os servidores de bancos de dados.
  - **Subcamada da Nuvem (Internet):** Se os dados da subcamada de base de dados precisarem ser publicamente disponibilizados para a Internet é utilizado a Camada da Nuvem para tal finalidade. Esta subcamada implementa questões de segurança e privacidade dos dados.
- **Camada de Aplicação:** Nesta camada estarão os clientes ou aplicações que utilizarão os dados coletados pelos sensores. Esses clientes podem ser páginas da Web, *smartphones* habilitados para receber os dados, empresas que contratem o serviço, outros servidores na rede ou qualquer computador que possua conexão com a Internet.

Cada camada preocupa-se apenas com as funções desempenhadas por elas, isso facilita o entendimento e o desenvolvimento das soluções tecnológicas de cada uma delas. Por exemplo, a aplicação que vai utilizar os dados da leitura dos sensores, não precisa saber como estes dados foram coletados ou como eles chegaram até o servidor na Internet.

### 3.1. Sistema Embarcado de adaptação

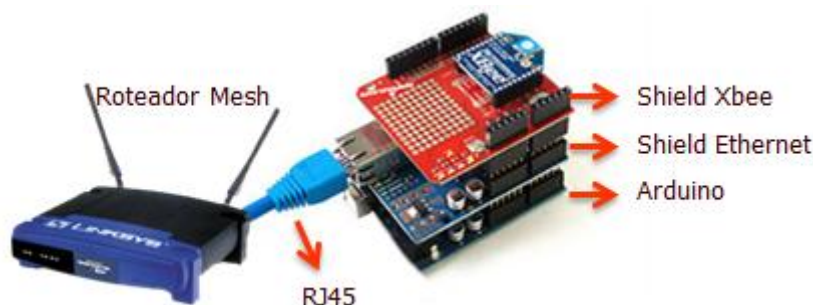
Para que as redes de sensores se comuniquem com o *backbone mesh*, será necessário desenvolver um sistema que faça a adaptação entre as redes XBee (ZigBee) e o *backbone mesh* IEEE 802.11. Para tal, foram desenvolvidos dois tipos de sistemas de adaptação, um para comunicação XBee/Mesh e outro para comunicação Sun SPOT/Mesh.

#### 3.1.1. Sistema XBee/Mesh

Este sistema é composto por uma placa Arduino com um *Shield Ethernet*. O Arduino é plataforma de desenvolvimento *open-source* que provê solução de *hardware* e *software* de baixo custo para computação física. Um *shield* é uma placa de expansão que adiciona mais funcionalidades ao Arduino. Existe *Shield* para diversos propósitos como: Controle de motores DC, matrizes de *leds*, GPS, 3G, áudio, Vídeo VGA, *joyticks* entre outros.

O *Shield Ethernet*, utilizado no sistema de adaptação, fornece ao Arduino uma interface IEEE 802.3 (*Ethernet*), que viabiliza a comunicação TCP/IP convencional com o roteador *mesh*. Outro *shield* utilizado é o *XBee Shield* juntamente com um módulo

XBee [Digi International 2009]. O módulo XBee é uma implementação proprietária do padrão ZigBee e o *Shield XBee* permite a comunicação deste módulo com a plataforma Arduino. A Figura 2 mostra o arcabouço que permite passar informações da plataforma XBee para o roteador *mesh*.

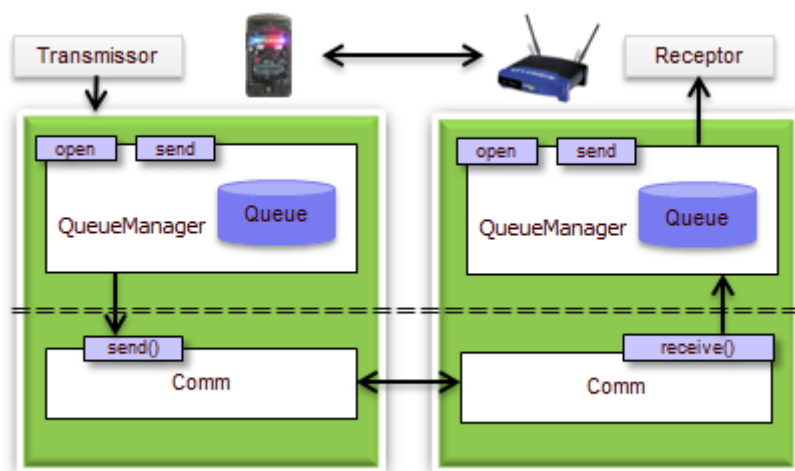


**Figura 2. Configuração física do sistema de adaptação XBee/Mesh.**

O arcabouço Arduino/XBee funciona como mote XBee para coleta e transmissão de dados e o sistema de adaptação (*Shield Ethernet + Arduino + Shield XBee*) serve como estação base e meio de comunicação com a rede *mesh* através da interface *Ethernet*. Além da arquitetura mostrada na Figura 2, o sistema de adaptação também é formado por um software instalado no Arduino que recebe os dados coletados da rede XBee, formata esses dados em XML e encaminha-os para o roteador *mesh* através do conector RJ45.

### 3.1.2. Sistema Sun SPOT/Mesh

O Sun SPOT e o *OpenWRT* são duas plataformas distintas que não possuem suporte a comunicação entre si. Com isso, desenvolveu-se um *middleware* orientado à mensagem que permite a comunicação entre as duas tecnologias supracitadas e facilita o desenvolvimento de aplicações dentro da rede de sensores. O mesmo não foi feito para o sistema XBee/Mesh porque o Arduino não possui um sistema operacional ou uma máquina virtual, o que impossibilita a criação de uma camada de infraestrutura para o *middleware*.



**Figura 3. Arquitetura do middleware utilizada pelo sistema de adaptação SunSPOT/OpenWRT.**

A Figura 3 exibe a arquitetura do *middleware* desenvolvido. Optou-se por um MOM (*Message Oriented Middleware*) porque as tecnologias sem fio possuem uma alta taxa de erros durante a transmissão, deste modo uma possível intermitência da conectividade gera a necessidade de armazenamento e posterior transmissão das mensagens (comunicação assíncrona). Neste caso, o transmissor é utilizado na plataforma Sun SPOT e o receptor no sistema *OpenWRT*.

O *middleware* da Figura 3 possui duas camadas, a camada de infraestrutura que está abaixo da linha tracejada e a camada de distribuição que está acima desta linha. A camada de infraestrutura é a camada responsável pela comunicação entre o transmissor e o receptor, portanto é a camada mais importante do sistema de adaptação. No Sun SPOT essa camada foi desenvolvida em Java e no *OpenWRT*, em C. Os dados são transferidos no formato XML, para garantir a interoperabilidades entre as duas linguagens.

A camada de distribuição trata as filas de mensagem do *middleware*, ela é responsável ainda por obter a informação da camada de infraestrutura e passar para a aplicação que está utilizando o *middleware*.

Uma das funções de um *middleware* é deixar transparente para a aplicação o funcionamento das camadas inferiores. A Figura 4 mostra o código da aplicação que envia a leitura do sensor de luminosidade do Sun SPOT para o *OpenWRT*. Percebe-se que não é necessário que a aplicação saiba como a mensagem é enviada, basta criar um gerenciador de filas, criar uma fila e enviar os dados. A camada de infraestrutura do *middleware* trata de toda a comunicação necessária.

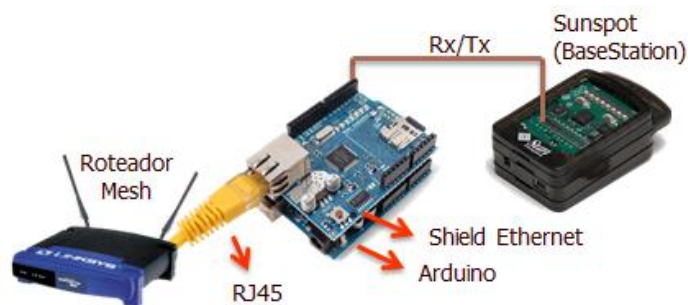
```
Message msg = new Message();
msg.setMessage(getSunspotLightSensor() );

QueueManager qm = new QueueManager();
qm.start();
qm.open("f1");

while (true) {
    qm.send("f1", msg);
}
```

**Figura 4. Exemplo de uma aplicação que utiliza o middleware.**

A Figura 5 mostra a configuração física do sistema de adaptação Sun SPOT/Mesh. Como a plataforma Sun SPOT não possui saída *Ethernet* para se conectar ao roteador *mesh*, foi necessário inserir uma placa Arduino com *Ethernet Shield* para intermediar esta comunicação. Porém, nesse caso, o Arduino funciona como um túnel e apenas repassa as informações da estação base da rede Sun SPOT para o roteador *mesh*.



**Figura 5. Configuração física do sistema de adaptação Sun SPOT/Mesh.**

O Arduino já possui bibliotecas que possibilitam a comunicação *Ethernet*. Com essa biblioteca é possível criar um IP, uma máscara e um *gateway* para o Arduino e em seguida iniciar uma comunicação *Ethernet* com um roteador. O sistema instalado no Arduino recebe os dados do Sun SPOT via interface serial, cria uma comunicação *Ethernet* com o *gateway mesh* e envia a informação formatada em XML.

### 3.2. Gateway Mesh

Este trabalho implementa, também, uma rede em malha sem fio (*mesh*) através de roteadores *Wi-Fi* que servem como *gateways* da RSSF heterogênea e formam um *backbone* sem restrição quanto ao consumo de energia.

Redes do tipo *mesh* possuem a vantagem de serem redes de baixo custo, fácil implantação e tolerantes a falhas. Destacam-se devido à grande maleabilidade topológica e interoperabilidade com outras redes e equipamentos. Os dados de sensoriamento advindos da RSSFs são disponibilizados aos roteadores *mesh*, que por sua vez suportam o envio desses dados pela Internet ou a um servidor específico.



**Figura 6. Roteador Linksys WRT54GL**

O roteador utilizado é o WRT54GL, da Linksys, em suas versões 1.0 e 1.1 que segue o padrão IEEE 802.11g na frequência não licenciada ISM (*Industrial, Scientific and Medical*) de 2,4GHz (Figura 6). O roteador sem fio possui 4 MB de memória *flash* e 32 MB de memória RAM, e pode rotear pacotes de clientes ligados a ele tanto pela interface sem fio como por suas quatro portas *Ethernet*. O roteador vem de fábrica com um sistema operacional da própria Linksys que possui uma interface de administração via Web. O roteador é adaptado e transformado em um roteador *mesh* através da instalação e configuração de uma distribuição especial do sistema operacional Linux denominada OpenWRT [OpenWRT Project 2012].

O OpenWRT permite a geração de imagens que venham a substituir *firmwares* de *hardware* especializados. O objetivo de substituir o *firmware* original dos roteadores sem fio (especificamente o Linksys WRT54G) é contornar as limitações impostas pelos fabricantes dos equipamentos. As ferramentas básicas do *OpenWRT* estão presentes em um arquivo binário executável denominado de *busybox*, as demais ferramentas estão disponíveis em módulos independentes. O OpenWRT utilizado opera com o *kernel* 2.4.30.

O *kernel* pode ser modificado e adaptado conforme as necessidades específicas. Por meio do *OpenWRT* nos roteadores, uma grande variedade de aplicativos poderão ser instalados. Em particular, a funcionalidade que permite o roteamento por múltiplos saltos é viabilizada por meio da instalação de um módulo específico para roteamento em redes *ad-hoc* sem fio, denominado OLSR (*Optimized Link State Routing*) [Clausen 2003].



O OLSR é um protocolo pró-ativo que trabalha de forma distribuída, baseado em tabelas que contém informações sobre topologia e estimativas de qualidades dos enlaces sem fio, trocados regularmente a fim de subsidiar tomadas de decisões quanto à configuração de rotas. Como o OLSR mantém rotas estabelecidas para todos os destinos da rede, funciona muito bem em redes onde o tráfego é aleatório e esporádico entre um grande número de nós, além de tornar a descoberta de rota mais rápida, sobretudo, para aplicações com requisitos temporais.

A qualidade dos enlaces é medida através da métrica ETX (*Expected Transmission Count*). A probabilidade de uma transmissão ser efetuada com sucesso é igual ao inverso do produto entre a taxa de recepção de ida ( $R_i$ ) e a taxa de recepção de volta ( $R_v$ ). As taxas de recepção  $R_i$  e  $R_v$  são obtidas através da quantidade de pacotes “HELLO” recebido pelos roteadores *mesh*.

$$ETX = 1/(R_i \times R_v)$$

Quanto mais alto for o valor do ETX, pior é a qualidade do enlace. Para uma rota com múltiplos saltos, o valor final do ETX é a soma do ETX de cada salto. Por exemplo, a comunicação entre um roteador A com um roteador C, passando pelo roteador B é expresso pela seguinte fórmula:

$$ETX_{ac} = ETX_{ab} + ETX_{bc}$$

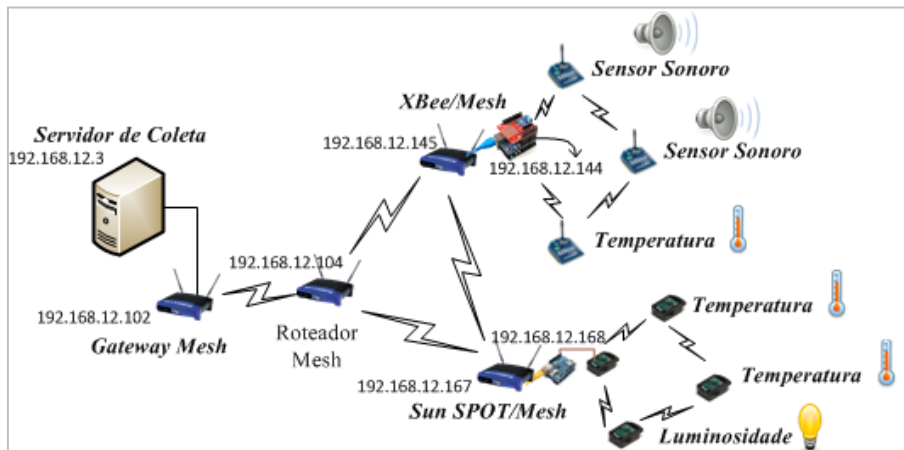
Outro módulo incluído na solução foi o serviço de autenticação WiFiDog. Esta ferramenta redireciona o cliente para uma página Web de autenticação quando o mesmo tenta o acesso. O WiFiDog também pode ser usado para auditoria da rede através de informações coletadas por seu sistema de contabilização. Para a operação do servidor WiFiDog são necessários os seguintes softwares: servidor Web Apache, a linguagem PHP e o banco de dados PostgreSQL.

A solução pode ser usada tanto em ambientes internos (*indoor*) quanto externos (*outdoor*). Em cenários *outdoor*, a solução *mesh* pode implementar um *backbone* sem fio, oferecendo acesso aos clientes móveis e distantes quilômetros do *gateway*. Como cada roteador possui um *switch* de 4 portas, a solução *outdoor* também oferece acesso aos clientes cabeados IEEE 802.3. Para estender a área de cobertura, antenas omnidirecionais com potência de 18.5 dB podem ser acopladas aos roteadores.

A utilização de redes *mesh* na arquitetura para Cidades Inteligentes trás vários benefícios, como confiabilidade, segurança e escalabilidade. Um nó *mesh* pode facilmente ser inserido na rede conforme a demanda para atender novos usuários. A seção seguinte mostrará alguns resultados experimentais obtidos no *testbed*.

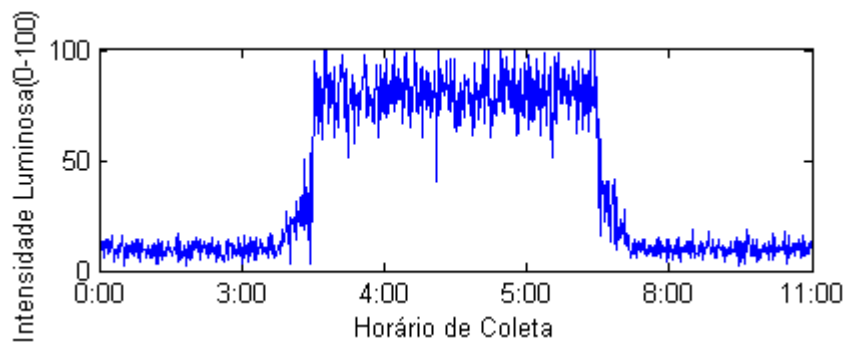
#### 4. Resultados Experimentais

Para validar a arquitetura e os sistemas de adaptação foi montado um *testbed* e foram realizadas algumas coletas ao redor do Centro de Informática da UFPE. A Figura 7 mostra a arquitetura dos experimentos para a validação da proposta, além do tipo de dados coletados e os IPs de cada nó, inclusive dos sistemas de adaptação. Na coleta, os sensores passaram 24h obtendo os dados do ambiente. A rede SunSPOT/Mesh coletou informações de temperatura e luminosidade e a rede XBee/Mesh coletou informações de temperatura e dos sensores sonoros.



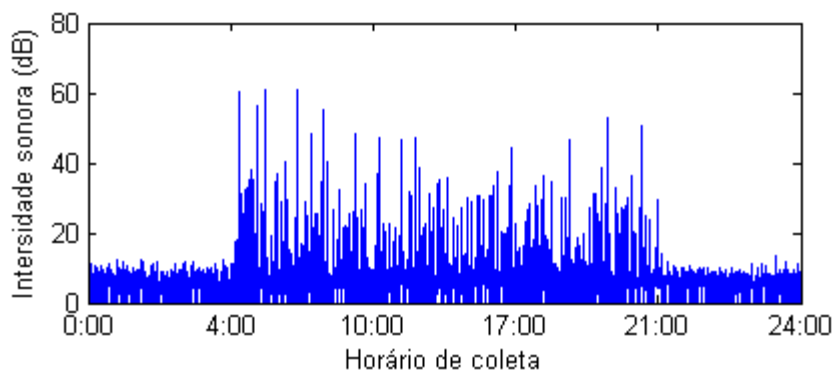
**Figura 7. Topologia do experimento**

Após as coletas, os dados foram analisados e alguns gráficos foram plotados. A Figura 8 mostra a intensidade luminosa coletada pela rede SunSPOT/Mesh durante vinte e quatro horas.



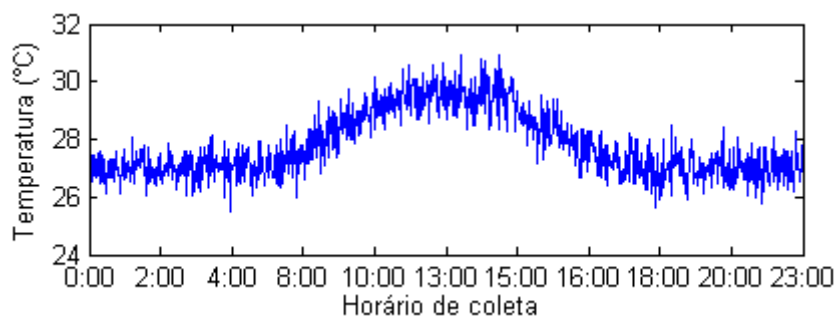
**Figura 8. Intensidade luminosa durante 24 horas**

A Figura 9 mostra dados de coleta do sensor sonoro. Essas informações foram coletadas próximo à Avenida Jorn. Aníbal Fernandes, dentro do Campus da UFPE. A flutuação de ruído é basicamente decorrente da passagem de veículos na região.



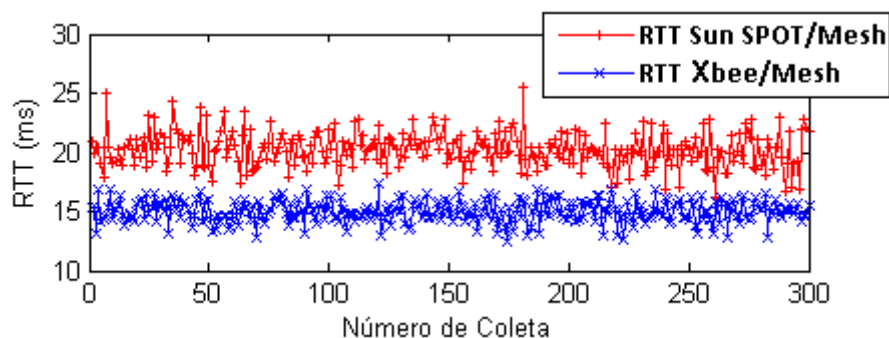
**Figura 9. Intensidade sonora durante 24 horas**

A Figura 10 mostra a variação da temperatura coletada pela rede XBee/Mesh durante 24 horas. Percebe-se que próximo às 13 horas a temperatura chega ao seu valor máximo.



**Figura 10. Temperatura durante 24 horas**

Para verificar o tempo gasto com a transmissão nos sistemas embarcados, verificou-se o RTT dos pacotes na rede Sun SPOT/Mesh e na rede XBee/Mesh. A Figura 11 mostra o resultado da coleta. Percebe-se que o RTT na rede Sun SPOT possui uma média superior (em torno de 20ms) em relação ao RTT na rede XBee (15ms). Isso acontece porque o Sun SPOT não possui interface de comunicação *Ethernet* e, dessa forma, é necessário inserir um elemento intermediário para realizar esta operação. Contudo, dada a baixa frequência na coleta dessas informações por parte de aplicações de monitoramento (ordem de minutos ou horas), estes valores de atraso podem ser considerados desprezíveis.



**Figura 11. RTT nos sistemas de adaptação**

## 5. Conclusão

Este artigo apresentou uma plataforma para Cidades Inteligentes baseada em RSSF e redes *mesh*. O objetivo principal é disponibilizar um modelo de estação-base híbrida capaz de obter dados de uma RSSF, composta por motes de diferentes fabricantes, e disseminá-los através de um backbone *mesh* baseado em uma solução de baixo custo que utiliza roteadores comerciais padrão IEEE 802.11.

Foram realizados alguns experimentos, a fim de validar a arquitetura, através da coleta de dados de sensores de temperatura, luminosidade e poluição sonora, próximo ao Centro de Informática da UFPE. Como trabalhos futuros estamos desenvolvendo um aplicativo Android para fornecer ao usuário, através da Internet, informações como a poluição sonora, concentração de CO, temperatura e umidade de uma determinada região. Além disso, pretende-se adicionar novos nós tanto na rede *mesh* quanto nas RSSFs, com objetivo de aumentar a cobertura e avaliar a escalabilidade da proposta, realização de testes em algumas regiões da cidade, bem como a inclusão de câmeras IP com o intuito de desenvolver um sistema integrado de monitoramento/controlado de tráfego urbano e de informações ambientais.

## Referências

- Akyildiz, I et al. (2005) "Wireless mesh networks: a survey", *Computer Networks*, Volume 47, Issue 4.
- Al-Karaki, J.N. and Kamal, A.E. (2004) "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol.11, no.6, pp. 6- 28.
- Arora, A. et al. (2006), "Kansei: A High-Fidelity Sensing Testbed" *IEEE Internet Computing*, vol. 10, pp. 35–47.
- Bers, J. et al. (2010) "Citysense: The Design and Performance of an Urban Wireless Sensor Network Testbed".
- Beutel, J. et al. (2009) "The Flocklab Testbed Architecture" *SenSys*, pp. 415–16.
- Clausen, T. and Jacquet, P. (2003) "Optimized Link State Routing Protocol (OLSR)," *IETF RFC 3626*.
- Digi International (2009), "XBee®/XBee-PRO® RF Modules, Product Manual v1.xEx - 802.15.4 Protocol. IEEE® 802.15.4 RF Modules".
- Filipponi, L. et al. (2010) "Smart City: An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors" *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pp.281-286, 18-25.
- German Telekom and City of Friedrichshafen (2010) "Friedrichshafen Smart City", <http://www.telekom.com/dtag/cms/content/dt/en/395380>, February.
- Gluhak, A. et al (2011) "A survey on facilities for experimental internet of things research," *IEEE Communications Magazine*, vol.49, no.11, pp.58-67.
- Kehua, S. et al. (2011) "Smart City and the Applications", In:*Internacional Conference on Eletronics, Communications and Controls (ICECC)*.
- Ojala, T. (2010), "Open Urban Testbed for Ubiquitous Computing", pp. 442–47.
- OpenWRT Project (2012) "OpenWrt Wireless Freedom", <http://www.openwrt.org/>, February.
- SensLAB (2010) "Very Large Scale Open Wireless Sensor Network Testbed", <http://www.senslab.info/>.
- Smartsatander (2010) "Smartsantander", <http://www.smartsantander.eu/>, February.
- Tan, L. and Wang, N. (2010) "Future Internet: The Internet of Things", *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol.5, pp.V5-376-V5-380, 20-22.
- Tutornet (2009) "Tutornet: A Tiered Wireless Sensor Network Testbed", <http://enl.usc.edu/projects/tutornet/>, pp. 19–28.
- UC Berkeley (2010) "WSN Testbeds at UC Berkeley" <http://www.millennium.berkeley.edu/sensornets/>, February.
- Univ. of VA (2009) "VineLab Wireless Testbed", [http://www.cs.virginia.edu/\\_whitehouse/research/testbed/](http://www.cs.virginia.edu/_whitehouse/research/testbed/).
- Werner-Allen, G. et al. (2005) "Motelab: A Wireless Sensor Network Testbed" *IPSN*.