

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Rafael de Lucena Valle

**IMPLEMENTAÇÃO DO PROTOCOLO COAP PARA O  
MONITORAMENTO EM REDES DE SENSORES SEM FIO**

Florianópolis

2013



Rafael de Lucena Valle

**IMPLEMENTAÇÃO DO PROTOCOLO COAP PARA O  
MONITORAMENTO EM REDES DE SENSORES SEM FIO**

Trabalho de conclusão de curso submetida  
ao Bacharel em Ciências da Computação  
para a obtenção do Grau de Bacharel em  
Ciências da Computação.

Orientador: Prof. Dr. Antônio Augusto  
Fröhlich

Coorientador: Prof. M.Sc. Arliones Hoel-  
ler Jr

Florianópolis

2013



Rafael de Lucena Valle

**IMPLEMENTAÇÃO DO PROTOCOLO COAP PARA O  
MONITORAMENTO EM REDES DE SENSORES SEM FIO**

Esta Trabalho de conclusão de curso foi julgada aprovada para a obtenção do Título de “Bacharel em Ciências da Computação”, e aprovada em sua forma final pelo Bacharel em Ciências da Computação.

Florianópolis, 15 de julho 2013.

---

Prof. Dr. Roberto Cislighi  
Coordenador

**Banca Examinadora:**

---

Prof. M.Sc. Arliones Hoeller Jr  
Coorientador

---

Prof. M.Sc. Arliones Hoeller Jr

---

Prof. Dr. Antônio Augusto Fröhlich

---

Prof. Dr. Eng. Rafael Luiz Cancian

---

Prof. Dr. Frank Siqueira

## **AGRADECIMENTOS**

Inserir os agradecimentos aos colaboradores á execução do trabalho.





## RESUMO

Redes de sensores são utilizadas para a captação, processamento de informação e atuação sobre um ambiente, tornando-as importantes para controle, telemetria e rastreamento de sistemas. Os nós das redes geralmente são computadores e rádios simplificados, que possuem restrições de memória, processamento, energia e comunicação, mas um custo relativamente baixo de equipamentos, tornando interessante a implantação destes sistemas. O protocolo HTTP foi desenvolvido pensado em computadores de propósito geral, onde essas restrições não existem. Um protocolo leve como CoAP pode tornar viável a criação de aplicações web em redes de sensores sem fio por um baixo custo.

É proposto uma infraestrutura de comunicação entre redes de sensores sem fio e a Internet, utilizando protocolos leves entre os nós sensores e um gateway GPRS para áreas sem acesso à WIFI, aproveitando a vasta abrangência da tecnologia de telefonia. Com a Utilização do CoAP é esperado uma redução de consumo de energia e memória, em relação a outros protocolos de aplicação existentes.

**Palavras-chave:** internetworking wireless sensor networks IPv6 6LoWPAN GPRS CoAP



## **LISTA DE FIGURAS**



## **LISTA DE ABREVIATURAS E SIGLAS**



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>15</b>
1.1	OBJETIVOS .....	15
1.1.1	Objetivo Geral .....	15
1.1.2	Objetivos Específicos .....	15
<b>2</b>	<b>DESENVOLVIMENTO .....</b>	<b>17</b>
2.1	EXPOSIÇÃO DO TEMA OU MATÉRIA .....	17
2.1.1	Metodologia .....	17
2.1.2	Justificativa e Motivação .....	17
2.1.3	Organização do Trabalho .....	18
2.2	REVISÃO BIBLIOGRÁFICA .....	18
2.2.1	Redes de sensores sem fio .....	18
2.2.2	Arquitetura orientada a serviços .....	19
2.2.3	Embedded Parallel Operating System .....	19
2.2.4	REST .....	19
2.2.5	Constrained Application Protocol .....	19
2.2.6	Trabalhos Relacionados .....	21
2.2.7	Proposta .....	22
2.2.8	Metas .....	23
2.2.9	Resultados parciais .....	23
2.2.10	Tecnologias utilizadas .....	23
2.2.11	Atividades pendentes .....	24
2.2.12	Considerações Parciais .....	24
<b>3</b>	<b>CONSIDERAÇÕES PARCIAIS .....</b>	<b>25</b>





## **1 INTRODUÇÃO**

Redes de sensores são utilizadas para a captação, processamento de informação e atuação sobre um ambiente, tornando-as importantes para controle, telemetria e rastreamento de sistemas. Os nós que participam destas redes geralmente são computadores e rádios simplificados, que possuem restrições de memória, processamento, energia e comunicação, mas um custo relativamente baixo de equipamentos. O maior consumo de energia neste tipo de aplicação é o rádio, portanto os desafios dos algoritmos de roteamento nesta área são em manter os rádios o mínimo período de tempo possível e manter o nó comunicável pela rede.

### **1.1 OBJETIVOS**

Descrição...

#### **1.1.1 Objetivo Geral**

O trabalho será descrever e implementar webservices em uma rede sensores sem fio, que farão a aquisição dos dados do ambiente e disponibilizarão as informações captadas. A comunicação entre os nós sensores será feita utilizando um protocolo de aplicação padrão, apropriado para redes de sensores sem fio, viabilizando o desenvolvimento de webservices para os nós da rede. A distribuição da informação para Internet será feita através de um gateway. Os objetivos são: o monitoramento de ambientes e a integração da rede de sensores sem fio com a Internet em lugares aonde não existe o acesso a rede cabeada ou sem fio, como lugares afastados, na área rural, por exemplo. GPRS possui a maior cobertura dentre as tecnologias de transmissão de telefonia no Brasil, atingindo cerca de 5477 municípios.

#### **1.1.2 Objetivos Específicos**

Implementar uma aplicação de redes de sensores sem fio, que utilize um gateway GPRS/Zigbee que será responsável em disponibilizar as informações captadas pela rede para Internet, além de oferecer uma simples aplicação web que será possível para o usuário configurar os parâmetros da rede. Na infra-estrutura de software será necessário implementar o protocolo

CoAP no EPOS. Desenvolver o protocolo CoAP utilizando a pilha UDP do EPOS ou portar a biblioteca libcoap, uma biblioteca de código aberto desenvolvida em C que implementa o protocolo de aplicação leve para dispositivos com restrições de recursos como poder computacional, alcance de rádio, memória, vazão ou tamanho dos pacotes de rede. O gateway 6LowPan/GPRS terá a capacidade de fazer a ponte entre a rede de sensores e a Internet, utilizando um modem GPRS. Será projetado utilizando um mote, um modem GPRS além do circuito necessário para integração dos dois módulos. Será a parte de hardware do projeto.

## **2 DESENVOLVIMENTO**

### **2.1 EXPOSIÇÃO DO TEMA OU MATÉRIA**

#### **2.1.1 Metodologia**

O desenvolvimento do protocolo será orientado a testes, aonde o código escrito apenas satisfaz as condições necessárias para validar um comportamento desejado da aplicação. Apesar de desafiador, por ser .... X demonstra que com certas ferramentas, e utilizando técnicas em nível de linkagem para testar o módulo de cada objeto, co TDD em sistemas embarcados é possível.

Será utilizado uma placa de desenvolvimento em conjunto com um módulo M95 da Quectel disponibilizada pelo Laboratório, para serem feitos os testes de envio de mensagens em diversos protocolos, inclusive testes com comandos proprietários adicionais do modem.

#### **2.1.2 Justificativa e Motivação**

Os mecanismos de confiabilidade na transmissão e as técnicas para se manter uma conexão do TCP e os rearranjos que são feitos para garantir a ordem das mensagens recebidas não são adequados para um dispositivo que possua restrições de energia, pois podem fazer que fiquem com seus transmissores, ligados por mais tempo para manter a conexão ou até mesmo para reenvio de mensagens. O maior consumo de energia de um nó sensor é no envio e recebimento de dados, quando mantém seu transmissor ligado.

Assim faz-se uso do UDP, um protocolo que não mantém conexão, os dados são recebidos fora de ordem e o envio é feito de uma mensagem por vez, sem o uso de streammings do TCP, que fazem que quem receba a mensagem precise montá-la e garantir que nenhuma das peças está corrompida. Também também é feita uma redução do tamanho do cabeçalho do pacote. Estas características demonstram uma alternativa interessante para estes equipamentos restritos. Testes feitos em implementações de sistemas operacionais similares ao EPOS, como Contiki e TinyOS, utilizando o protocolo CoAP demonstram redução no consumo de energia e memória em relação ao HTTP.

A falta de padronização dos protocolos afeta o desenvolvimento de uma rede pública ubíqua de uma cidade inteligente por exemplo, a falta de um padrão de comunicação. A maioria das empresas utiliza protocolos pro-

prietários, que se comunicam apenas com os produtos da própria empresa.

O protocolo HTTP foi desenvolvido para comunicação de computadores de propósito geral, onde as restrições citadas não são comuns. Em relação ao tamanho do pacote HTTP é um problema, já que redes que trabalham nessa frequência possuem uma restrição de 128 bytes. Além do tamanho do pacote HTTP, manter uma conexão TCP é custosa, já que os nós sensores que precisam manter seus rádios desligados o maior tempo possível.

Um protocolo leve como CoAP pode tornar viável a criação de aplicações web em redes de sensores sem fio por um baixo custo. Neste trabalho é proposto uma infraestrutura de comunicação entre redes de sensores sem fio e a Internet, utilizando protocolos leves entre os nós sensores e um gateway GPRS para áreas sem acesso à WIFI, aproveitando a vasta abrangência da tecnologia de telefonia. Com a utilização do CoAP é esperado uma redução de consumo de energia e memória, em relação a outros protocolos de aplicação existentes.

O Sistema operacional EPOS possui implementação das camadas de transporte UDP e TCP, porém não possui nenhum protocolo de aplicação desenvolvido.

### **2.1.3 Organização do Trabalho**

## **2.2 REVISÃO BIBLIOGRÁFICA**

### **2.2.1 Redes de sensores sem fio**

Avanços recentes nas tecnologias de sistemas eletrônicos, semicondutores, sensores, microcontroladores e rádios tornaram possível o desenvolvimento de redes de sensores de baixo custo e baixo consumo uma realidade. São utilizados para capturar, processar e comunicar dados captados do ambiente. Geralmente tais redes possuem centenas ou milhares de sensores tornando-as importantes para controle, telemetria e rastreamento de sistemas.

Características destas redes são: pouca memória, pouco alcance do rádio, baixa capacidade de processamento e bateria, e custo reduzido. A conservação de energia é um dos objetivos das redes de sensores sem fio, deve-se minimizar o consumo em todos os níveis do sistema, da aplicação até o nível de hardware. Redes de sensores são utilizadas para a captação, processamento de informação e atuação sobre um ambiente.

Um nó pertencente a esta rede geralmente é um dispositivo especifica-

mente desenvolvido para um propósito, que possui poucos recursos computacionais e energéticos e se comunicam entre seus semelhantes.

### **2.2.2 Arquitetura orientada a serviços**

Vantagens em novos processos de negócios podem ser definidos de uma forma similar, reutilizando os serviços e acrescentando metainformação para uso de conectores genéricos. (??)

### **2.2.3 Embedded Parallel Operating System**

Sistema Operacional Multithread com suporte a preempção, foi desenvolvido em C++ e faz uso intenso de programação orientada a aspectos utilizando templates. Possui abstrações infladas para entidades temporais como relógio, alarme e cronometro, biblioteca com estruturas de dados e sequenciadores que permite o uso de ferramentas para geração automatizada de abstrações de sistemas. A portabilidade é atingida utilizando entidades chamados de Mediadores de Hardware que fornecem interfaces simples para acesso as funções específicas de arquitetura. Estas interfaces são utilizadas por entidades abstratas como alarmes e threads periódicas.

Projetado utilizando o método ADESD, um método para projeto de sistemas embarcados orientados á aplicação. Esta metodologia guia o desenvolvimento paralelo de hardware e software além de manter certa portabilidade. O EPOS possui porte para as seguintes arquiteturas: MIPS, IA32, PowerPC, H8, Sparc e AVR.

(??)

### **2.2.4 REST**

### **2.2.5 Constrained Application Protocol**

Um dos principais objetivos do CoAP é ser uma alternativa de um protocolo web genérico para redes com dispositivos com restrição de energia e memória.

Possui

A IETF estabelece as condições mínimas para o desenvolvimento de um protocolo de aplicação compatível com HTTP, mas focado em aplicações aonde energia e memória são escassas. O protocolo CoAP foi projetado



posta Mensagens confirmadas Mensagens não-confirmadas usa um contador de timeouts e a transmissão da mensagem de tempos em tempos define por uma função que varia de acordo com o número de tentativas. A função é:

Um servidor que atende uma requisição confirmável pode respondê-la em conjunto da mensagem de confirmação. Essa técnica é chamada de Pigbackend um mecanismo de transmissão para mensagens confirmadas.(??)

Recursos Os recursos são identificados por uma URI, e os métodos são implementados de forma similar ao HTTP.

A descoberta de serviços no protocolo CoAP é feita através de socket Multicast. A descoberta de recursos é feita quando um servidor recebe uma requisição GET para o recurso /well-know/core. O servidor CoAP deve responder no formato CORE link Format.(??)

## 2.2.6 Trabalhos Relacionados

LibCoap é uma biblioteca implementada em C do protocolo CoAP, possui 292K de tamanho compilada estaticamente em sua versão 4.0.1

Contiki Sistema Operacional desenvolvido em C, código aberto para sistemas com restrição de memória e (que lidam com temporização)? foi desenvolvido para ser um sistema operacional para a Internet das coisas. Possui uma camada de abstração RESTful para web services chamada Erbium, que implementa o protocolo CoAP. Um processo no Contiki possui bloco de controle que contém informações de tempo de execução do processo, o nome do processo, estado do processo e um ponteiro para a thread do processo. O código que reside na thread do processo é armazenado na ROM. No contiki os processos são implementados como phrotothreads, uma combinação entre eventos e threads. São executadas no kernel dirigido a eventos do Contiki. Como threads, possuem comportamentos de bloqueio e espera, que permite o intersequenciamento entre as protothreads. Dos eventos herdam o baixo overhead de memória e por não necessitarem de salvamento de contexto na região de memória da Stack. Uma protothread é uma maneira de estruturar o código, de forma que permita que o sistema rode outras atividades enquanto o código está esperando para algo acontecer. Permite que o código C execute de uma maneira similar a threads, porém sem o overhead de memória. A maior vantagem das protothreads sobre as threads comuns é o baixo consumo de memória, todas utilizam a mesma Stack e a troca de contexto é feita por (TODO) Stack rewinding. Cada protothread consome 2 bytes de memória, esses dois bytes são utilizados para armazenar a continuidade local, uma referencia utilizada para um pulo condicional ser feito na execução da thread. Essa marcação é feita utilizando o resultado da macro -LINE- do

compilador, que é o número da linha do código no momento da compilação. Com esta marcação próxima vez que a thread for chamada ele utilize o switch para aquele número de linha, que é aonde o teste para verificar se a condição passada pela função `PT THREAD WAIT UNTIL` é verificada.

Criadores: Adam Dunkels Protocolos de Roteamento

Dispositivos de baixo consumo que possuem restrição de energia e precisam ficar ligados durante anos utilizando uma única bateria, precisam consumir o mínimo de energia possível. O transmissor e receptor sem fio é um dos componentes que mais consome energia e é preciso mantê-lo o mínimo de tempo possível, mas também precisa ficar ligado para comunicação com seus vizinhos. O Contiki propõe uma estratégia de ciclos de trabalho que consegue manter um nó comunicável em uma rede, porém com seu rádio desligado em aproximadamente 99% do tempo. Utiliza mecanismos assíncronos com precise timing e fast sleep optimization (TODO)

O protocolo de roteamento padrão ContikiMAC os nós dormem a maior parte do tempo e acordam periodicamente para verificar a atividade do rádio. Se um pacote de rede é detectado o receptor mantém-se acordado para receber o próximo pacote

TinyOS Sistema Operacional projetado para sistemas embarcados com comunicação sem fio e restrições energéticas. Foi desenvolvido em nesC, uma linguagem código aberto que é uma extensão do C. é um sistema operacional event-driven desenvolvido para redes de sensores que possuem recursos limitados. Possui uma implementação do CoAP baseada na libCoAP.

Possui licença BSD

Protocolos de Roteamento

Arduino Protocolos de Roteamento

### 2.2.7 Proposta

Será implementada uma biblioteca que utiliza a camada UDP do EPOS para dar suporte ao protocolo CoAP.

O trabalho também consiste na implementação de uma aplicação para gateway GPRS/Zigbee utilizando o EPOS e um componente de hardware que será acoplado ao EposMoteII que está sendo desenvolvido em paralelo por um colega de laboratório. O desenvolvimento da aplicação no EPOS que será responsável pelo roteamento de mensagens para Internet utilizando a tecnologia GPRS, provida por um módulo GSM/GPRS da Quectel o M95.

As principais funções deste gateway é receber os dados da rede de sensores e encaminhá-las para um servidor remoto que armazenará essas informações e exibirá de forma conveniente para o usuário final.



As funções a serem desenvolvidas na aplicação do gateway são: Configuração SMS Envia mensagem SMS Recebe mensagem SMS  
Configuração contexto PDP Configuração GPRS Configuração TCP/IP

### 2.2.8 Metas

Entregar um módulo simplificado do protocolo CoAP no primeiro semestre. Testes efetuados no modem GSM

Entregar o firmware do gateway na segunda parte do ano A aplicação desenvolvida no EPOS precisará de um buffer para o recebimento de dados da rede ZigBEE que será enviado para rede via GPRS. Duas threads, uma produtora que ficará escutando o rádio ZigBEE e alimentando num formato de dados ainda não especificado. Outra consumidora que será responsável em utilizar estes dados na rede de sensores e encaminhá-los pra Internet usando a extensão GPRS do EPOSmote II.

### 2.2.9 Resultados parciais

A parte de validação de um pacote CoAP foi feita com TDD e está disponível no site: (TODO) Ao receber uma mensagem de confirmação, remove da lista a mensagem que não havia sido confirmada utilizando o id. Ao receber uma mensagem confirmável, envia uma mensagem de confirmação. Repassar mensagem para controle de Requisição e Reposta Adicionar a lista de mensagem recebidas.

Enviar mensagem não confirmável Enviar mensagem confirmável e adicionar na lista de confirmações pendentes. Reenviar a mensagem que está na lista de confirmação pendente e reconfigurar o próximo reenvio.

Modelagem do sistema Testes no Modem

Enviar Mensagem Receber Mensagem

Criar socket TCP Enviar mensagem via socket Receber mensagem via socket

Fazer requisição HTTP para um webserver, foi possível utilizando os comandos proprietários do modem.

### 2.2.10 Tecnologias utilizadas

C++ Linguagem de programação com suporte a programação genérica e orientação a objetos. Foi criada por Bjarne Stroustrup em 1970 como uma

extensão da linguagem C.

CPPUTEST é um utilitário para testes unitários em C++, focado em aplicações embarcadas e foi desenvolvido por xxx

Autotools/CMake Ferramenta utilizada no ambiente GNU para encontrar bibliotecas e montar os projetos com código C e C++.

A linguagem de programação utilizada é o C++, que API do EPOS de alarme, chronometro, threads Framework de testes CPPUTEST CMake ou Autotools?

API assíncrona UDP do EPOS

NewLib

GDB

Git

Vim

### **2.2.11 Atividades pendentes**

Camada de Requisição e Resposta do CoAP Implementação da aplicação gateway GPRS/Zigbee no EPOS Discussões interessantes:

### **2.2.12 Considerações Parciais**

Um objeto interessante a ser desenvolvido é um cronômetro decrescente e configurável, que execute uma função passada pelo usuário seja executada no final do decremento de forma assíncrona. Provavelmente bloqueante, pois o buffer de mensagens confirmáveis, que precisam ser confirmadas será único.

### **3 CONSIDERAÇÕES PARCIAIS**



## **ANEXO A – Código Desenvolvido**

