

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

Rafael de Lucena Valle

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

Florianópolis

2013

Rafael de Lucena Valle

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

Trabalho de conclusão de curso submetido
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

RESUMO

Redes de sensores são utilizadas para a captação, processamento de informação e atuação sobre um ambiente, tornando-as importantes em aplicações de controle, telemetria e rastreamento.

Os nós destas redes possuem processadores, transmissores e receptores simplificados, que possuem restrições de memória, processamento, energia e comunicação. Contudo estes dispositivos apresentam um custo baixo de equipamentos, tornando mais interessante a implantação destes sistemas.

O protocolo HTTP, um protocolo de aplicação muito utilizado na atualidade, 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.

Este trabalho propõe 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 WSN IPv6 6LoWPAN GPRS CoAP IoT

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associa��o Brasileira de Normas T�cnicas
CoAP	Constrained Application Protocol
EPOS	Embedded Parallel Operating System
HTTP	Hipertext Transfer protocol
IETF	Internet Engineering Task Force
M2M	Machine-to-Machine
REST	Representational State Transfer
UDP	User Datagram Protocol
6LoWPan	IPv6 over Low power Wireless Personal Area Networks
IoT	Internet of Things

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVOS	9
1.1.1	Objetivos Específicos	9
1.2	JUSTIFICATIVA	9
1.2.1	Metodologia	11
1.3	ORGANIZAÇÃO	11
2	REVISÃO BIBLIOGRAFICA	13
2.1	REDES DE SENSORES SEM FIO	13
2.2	ARQUITETURA ORIENTADA A SERVIÇOS	13
2.3	REST	13
2.4	CONSTRAINED APPLICATION PROTOCOL	14
2.4.1	Formato das mensagens	14
2.4.2	Camada de Reposta e Requisição	15
2.4.3	Recursos	16
2.5	EMBEDDED PARALLEL OPERATING SYSTEM	16
2.6	TRABALHOS RELACIONADOS	16
2.6.1	Contiki	16
2.6.2	LibCoap	17
2.6.3	TinyOS	17
3	PROPOSTA	19
3.1	METAS	19
3.2	CRONOGRAMA	20
3.3	RESULTADOS PARCIAIS	21
3.3.1	Atividades pendentes	21
4	CONSIDERAÇÕES FINAIS	23

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.

Nós que participantes 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

O objetivo geral desse trabalho é descrever e implementar webservices em uma rede sensores sem fio, que farão a aquisição dos dados do ambiente e disponibilizarão as informações captadas na Internet.

1.1.1 Objetivos Específicos

Os objetivos específicos do trabalho são relacionados ao desenvolvimento de duas aplicações e uma biblioteca.

- Implementar uma aplicação de redes de sensores sem fio.
- Desenvolver o firmware do gateway GPRS/Zigbee, que será responsável em disponibilizar as informações captadas da rede para Internet.
- Portar o protocolo CoAP para o EPOS.
- Monitoramento de ambientes e a integração da rede de sensores sem fio com a Internet.

1.2 JUSTIFICATIVA

Os mecanismos de confiabilidade na transmissão, técnicas para se manter uma conexão do TCP e 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 mantem 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 é feito 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 proprietários, 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.

A distribuição da informação para Internet será feita através de um gateway. Este gateway é composto por um EposMoteII e um módulo GPRS. Em lugares aonde o 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.

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

1.2.1 Metodologia

Inicialmente será feito um levantamento bibliográfico sobre os tópicos escolhidos e as tecnologias utilizadas. Para uma validação inicial de modelo do sistema, serão feitas simulações dos nós da rede e do gateway. As simulações serão feitas nas ferramentas OMNeT++, biblioteca e framework de simulação de redes, e o Cooja, simulador do Contiki.

Será feito um levantamento dos componentes necessários para o desenvolvimento do gateway 6LowPan/GPRS utilizando o mote EPOSMote II e um modem GPRS, que será definido no decorrer do trabalho. Então será desenvolvido o esquemático para fabricação do gateway.

Após a validação do modelo, será iniciada a prototipação do hardware e a implementação do protocolo de aplicação CoAP no EPOS. O desenvolvimento do protocolo será orientado a testes, aonde o código escrito apenas satisfaz as condições necessárias para validar um comportamento desejado da aplicação.

Para testes de integração, as aplicações serão executados na plataforma de sensores sem fio EPOS Mote II utilizando o EPOS com o CoAP desenvolvido.

Nos testes de integração do gateway, será utilizada 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.

1.3 ORGANIZAÇÃO

Para infra-estrutura de software será necessário implementar o protocolo CoAP no EPOS.

Desenvolver o protocolo CoAP utilizando a pilha UDP do EPOS.

2 REVISÃO BIBLIOGRAFICA

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.

2.1 REDES DE SENSORES SEM FIO

Redes de sensores sem fio são utilizados para capturar, processar e comunicar dados captados do ambiente. Geralmente tais redes possuem centenas ou milhares de sensores e possuem as seguintes características: pouca memória, pouco alcance do rádio, baixa capacidade de processamento e bateria, e custo reduzido. Um nó pertencente a esta rede geralmente é um dispositivo especificamente desenvolvido para um propósito, que possui poucos recursos computacionais e energéticos e se comunicam entre seus semelhantes.

A conservação de energia é um dos objetivos das redes de sensores sem fio, pois não estão ligados diretamente a fonte de energia. 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.

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. (??)

2.3 REST

Os servidores não salvam estado das aplicações, e fazem uso de métodos HTTP explicitamente, são eles:

- GET: para receber um recurso do webserver.
- POST: criar um recurso no webserver.
- PUT: mudar o estado de um recurso do webserver.

- **DELETE**: remover o recurso ou alterar para um estado vazio.

A identificação de recursos na rede é feita por estruturas de diretórios, as URIs. De acordo com essa definição uma URI é uma árvore com ramos subordinados e superordinados conectando os nós.

Para transferência de dados utiliza-se formatos genéricos que enfatizam simplicidade e usabilidade pela internet, como XML e JSON.

2.4 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.

As vantagens de utilizar um protocolo compatível com o HTTP são: a facilidade de integração e o reuso de aplicações. CoAP possui compressão é apenas uma compressão do HTTP, mas um subconjunto REST otimizado para M2M, possui suporte para descoberta de recursos, suporte a multicast e troca de mensagens assíncronas com simplicidade e baixo overhead.

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 levando em consideração as restrições energéticas e altas taxas de falha na transmissão dos pacotes.

A comunicação entre os pontos no CoAP é de forma assíncrona usando o UDP, a confiabilidade é opcional e feita através de um mecanismo de retransmissão exponencial. Possui 4 tipos de mensagem: Confirmável, Não-Confirmável, Confirmação (ACK) e Reset.

2.4.1 Formato das mensagens

```

0 1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
0 1 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
—Ver— T — TKL — Code — Message ID — +-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ — Token (if any, TKL bytes)
... +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
— Options (if any) ... +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+ —1 1 1 1 1 1— Payload (if any) ... +-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ O Formato do pacote
CoAP (??)

```

Os campos do pacote CoAP são definidos:

- Versão: 2-bit Indica a versão do CoAP, implementações devem utilizar este campo com (1).
- Tipo: 2-bit Campo responsável para definir o tipo da mensagem: Confirmável (0), Não-Confirmável (1) , de Confirmação (2) ou Reset (3).
- Tamanho do Token: 4-bit, o tamanho do Token pode variar entre 0 e 8 bytes. Tamanhos entre 9 a 15 são reservados e não devem ser usados.
- Código: 8-bit separados em 3-bit mais significativos para classes e 5-bits menos significativos para detalhe. As classes podem indicar uma requisição (0), uma resposta de sucesso (2), e uma resposta de erro do cliente (4), ou uma resposta de erro do servidor (5), as outras classes são reservadas. Em um caso especial o código 0.00 indica uma mensagem vazia.
- ID da mensagem: 16-bit in network byte order. Usada para deduplicação de mensagens e para confirmação ou reset de mensagens.

As regras para definição do ID de mensagem são:

Codificação das opções

A transmissão de mensagens é controlada basicamente pelos parâmetros:

- ACK TIMEOUT:
- ACK RANDOM FACTOR:
- MAX RETRANSMIT:
- NSTART:
- DEFAULT LEISURE:
- PROBING RATE: Período

2.4.2 Camada de Reposta e Requisição

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

O servidor irá ignorar mensagens que chegam por multicast quando não puder responder nada de útil. Quando possuir uma informação suficientemente nova pode responder na própria mensagem de confirmação (ACK). Essa técnica é chamada de "Pigbackend" um mecanismo de transmissão para mensagens confirmadas.(??)

2.4.3 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.(??)

2.5 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 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.

(??)

2.6 TRABALHOS RELACIONADOS

2.6.1 Contiki

Contiki Sistema Operacional criado por Adam Dunkels em 2000, escrito em C, código aberto para sistemas com restrição de recursos que se comunicam em uma rede. Foi desenvolvido para ser um sistema operacional para Internet das coisas. Possui uma camada de abstração RESTful para web services chamada Erbium, que implementa o protocolo CoAP.

Cada processo no Contiki possui bloco de controle, que contém informações de tempo de execução do processo e uma referência para uma protothread, na qual o código é armazenado na ROM. Protothread é uma combinação entre eventos e threads, possuem comportamentos de bloqueio e espera, que permite o intersequenciamento e dos eventos herdando o baixo overhead de memória por não necessitarem de salvamento de contexto.

Na parte de implementação, estas técnicas são maneiras de estruturar o código de uma forma que permita que o sistema rode outras atividades enquanto o código está esperando para algo acontecer. Cada protothread consome 2 bytes de memória, esses dois bytes são utilizados para armazenar a continuidade local, uma referência utilizada em um pulso condicional durante a execução da thread. Essa marcação é feita utilizando o resultado da macro `—LINE—` do compilador. É um método similar ao mecanismo de Duffy e Co-rotina em C. (??)

O transceiver sem fio é um dos componentes que mais consome energia quando ligado escutando o ambiente, assim uma das estratégias utilizadas é manter o mínimo de tempo possível ligado, mas o suficiente para manter a troca de mensagens entre a rede. 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.

2.6.2 LibCoap

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

2.6.3 TinyOS

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

3 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. Este componente esta 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.

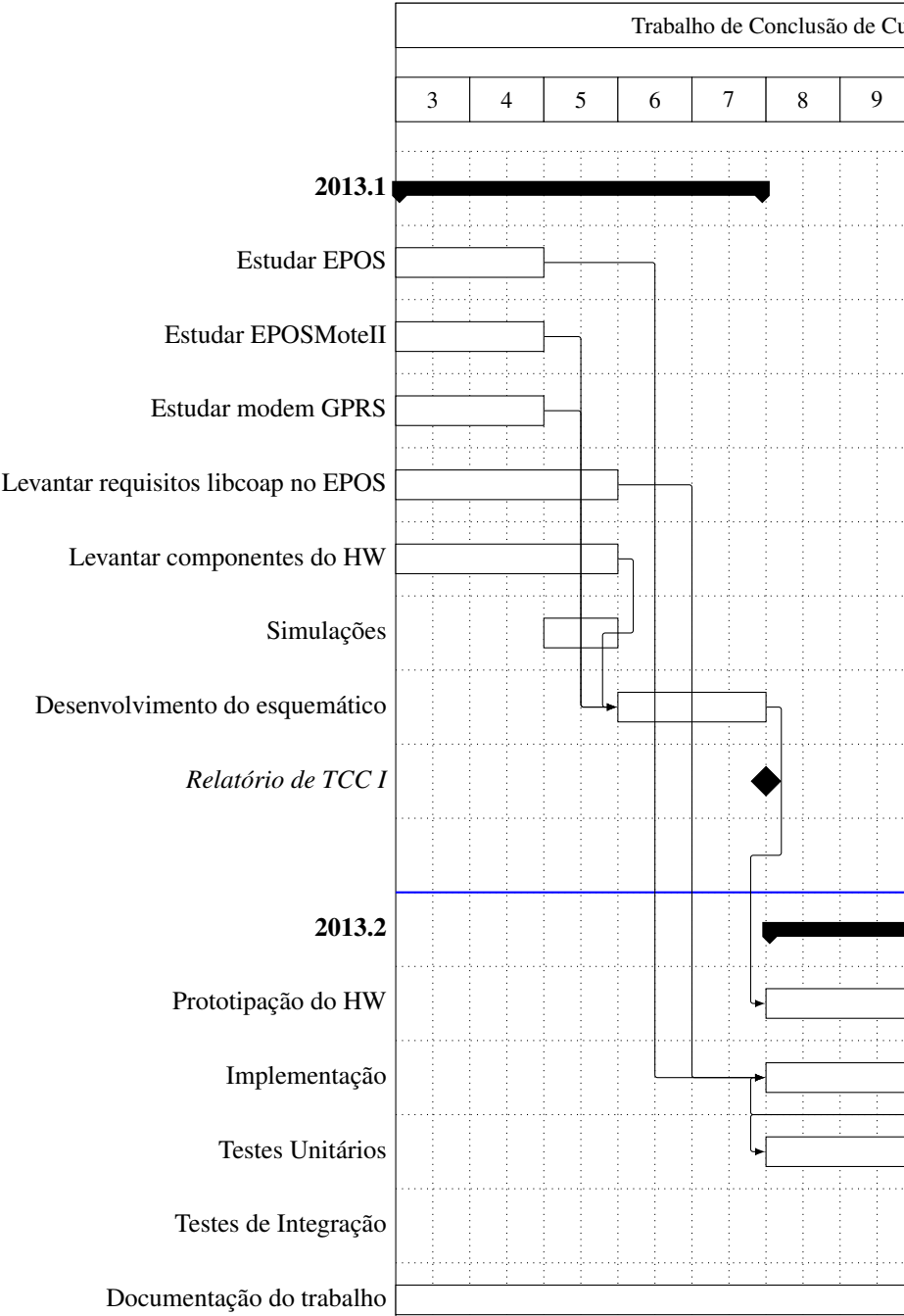
As funções a serem desenvolvidas na aplicação do gateway são:

- Configuração SMS
- Envia mensagem SMS
- Recebe mensagem SMS
- Configuração contexto PDP
- Configuração GPRS
- Configuração TCP/IP
- Manter uma conexão TCP/IP

3.1 METAS

Entregar um módulo simplificado do protocolo CoAP no primeiro semestre. Testar as funcionalidades no modem GSM/GPRS. Desenvolver a aplicação no EPOS no segundo semestre e fazer testes de integração.

3.2 CRONOGRAMA



3.3 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 Resposta 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.

3.3.1 Atividades pendentes

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

4 CONSIDERAÇÕES FINAIS

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.

O gateway ZigBEE/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, que será desenvolvida em paralelo por um colega do laboratório.

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.