

## Projeto Cross-layer entre as Camadas MAC, Roteamento e Aplicação em Redes de Sensores sem Fio

Eduardo Souto<sup>1,2</sup>, Rafael Aschoff<sup>1</sup>, Djamel Sadok<sup>1</sup>, Judith Kelner<sup>1</sup>

<sup>1</sup>Grupo de Pesquisa em Redes e Telecomunicações do Centro de Informática –  
Universidade Federal de Pernambuco (UFPE)  
Caixa Postal 50.740-540 – Recife – PE – Brasil

<sup>2</sup>Universidade Federal do Amazonas (UFAM)  
Manaus – AM – Brasil.

{esouto, rafael.aschoff, jamel, jk}@gpert.ufpe.br

**Abstract.** *This work proposes a join design across MAC, routing, and application layers to improve the performance of energy-constrained wireless sensor networks. The performance of the cross-layer project proposed is evaluated through simulation. We compared our cross-layer optimized networking protocol with traditional protocol (i.e. without cross-layer optimization). The results show that the proposed approach significantly improves the communication efficiency, the energy consumption, and outperforms the traditional layered approach architecture.*

**Resumo.** *Este trabalho propõe um projeto cross-layer entre as camadas MAC, de roteamento e de aplicação para melhorar o desempenho das redes de sensores sem fio. O desempenho do projeto cross-layer proposto é avaliado através de simulação e comparado com a abordagem tradicional, ou seja, sem otimização cross-layer. Os resultados de simulação mostram que o projeto integrado apresenta melhores índices de eficiência na utilização da largura de banda, na comunicação do protocolo de roteamento e no tempo médio de vida dos nós sensores quando comparados com a abordagem tradicional.*

### 1. Introdução

Atualmente, a maioria dos esforços de pesquisa em redes de sensores tem sido dirigida ao desenvolvimento de novas plataformas de hardware e de um grande número de protocolos e algoritmos para rede, comunicação e processamento da informação. Essas soluções têm sido tradicionalmente projetadas de acordo com a arquitetura em camadas, ou seja, a funcionalidade da rede é essencialmente tratada em cada camada como uma caixa preta: as camadas superiores somente necessitam conhecer as interfaces para as camadas inferiores, mas não detalhes de como as interfaces são implementadas. Claramente, a arquitetura em camada fornece modularidade, a qual contribui para a simplicidade e escalabilidade do sistema inteiro. Adicionalmente, isto tem sido o fator chave para o sucesso de muitos sistemas de rede, incluindo a Internet.

Enquanto a abordagem em camadas tem sido usada com bastante sucesso em redes cabeadas, em redes sem fio esta tem se mostrado cada vez mais inadequada. Em redes sem fio existe um acoplamento natural entre as diferentes camadas. Por exemplo, a capacidade do enlace de rádio depende dos níveis de sinais e interferência e, portanto,

depende da potência e do escalonamento de transmissão dos outros enlaces. Para complicar ainda mais este contexto, mesmo se o escalonamento e a potência atribuída a cada enlace fossem conhecidos, o canal sem fio ainda possui variações devido a outros problemas de transmissão como o “*multipath fading*” [Liu 2003]. Essas características são bem diferentes nas redes cabeadas onde a capacidade de cada enlace é fixa e conhecida. Portanto, as taxas de dados que podem ser suportadas pelas aplicações em redes sem fio estão diretamente associadas à maneira como os recursos nas camadas física e MAC (tais como modulação, codificação, controle de potência, entre outros) são escalonados.

Devido a esse forte acoplamento, os pesquisadores têm argumentado que a otimização dos sistemas de redes pode ser obtida através do conhecimento de determinadas informações disponíveis em cada camada, a qual tem sido denominada de “*otimização cross-layer*” [Santhanam e Cruz 2003][Chen 2005]. De fato, a lógica é bastante simples, o processo de otimização pode obter um melhor desempenho quando manipula de maneira integrada as variáveis de controle de duas ou mais camadas, do que quando são otimizadas separadamente. Entretanto, é necessário um cuidado especial no desenvolvimento de soluções *cross-layer*. O forte acoplamento entre as camadas pode facilmente tornar o sistema mais complexo, imprevisível e frágil. Mudanças em uma camada podem potencialmente serem prejudiciais ao sistema, caso o impacto de tais mudanças sobre o comportamento do sistema não seja cuidadosamente considerado [Kawadia e Kumar 2005].

Em redes de sensores, a maioria dos protocolos de comunicação propostos explora a natureza colaborativa dos nós sensores e, geralmente, visa a utilização eficiente dos recursos, principalmente em relação ao consumo de energia e processamento. Comum a todas essas soluções é o fato de que estas redes também são projetadas sobre a tradicional arquitetura de protocolos em camada, ou seja, a maioria desses protocolos de comunicação é individualmente desenvolvida para as diferentes camadas de rede. Enquanto estes protocolos podem alcançar um bom desempenho em termos das métricas relativas a cada uma dessas camadas, estes não são otimizados de maneira integrada para maximizar o desempenho da rede inteira, enquanto minimizam a quantidade de energia consumida [Akyildiz 2006]. Considerando os recursos escassos das redes de sensores, o projeto integrado das camadas de rede, isto é, o projeto *cross-layer*, se apresenta como uma alternativa promissora para a ineficiência das tradicionais arquiteturas de protocolos em camadas.

Este trabalho explora a forte interação existente entre as camadas MAC, de roteamento e de aplicação e propõe um projeto integrado entre estas camadas. O desempenho do projeto integrado proposto é avaliado através de simulação. Os resultados são obtidos e comparados com a abordagem tradicional, ou seja, sem otimização *cross-layer*. Os resultados mostram que a comunicação direta entre os protocolos pode contribuir para a melhoria de desempenho e do tempo de vida das redes de sensores.

O restante deste artigo está organizado como segue. A Seção 2 apresenta algumas das propostas *cross-layer* existentes para redes de sensores. A Seção 3 descreve os protocolos usados e apresenta o projeto *cross-layer* proposto. A Seção 4 apresenta detalhes da simulação e os resultados. Por fim, a Seção 5 apresenta as conclusões finais do trabalho.

## 2. Trabalhos Existentes

Recentemente tem surgido um grande número de propostas de projeto *cross-layer* para redes de sensores. Pesquisadores têm estudado aspectos específicos para melhorar o desempenho dessas redes e, abordando a comunicação direta entre camadas, adjacentes ou não, têm apresentado várias propostas de projeto *cross-layer*.

Algumas destas propostas exploram o projeto integrado entre as camadas MAC e Física. Em [Haapola 2005] foi proposto o emprego de um modelo de energia sob três protocolos MAC. Os autores fornecem uma análise do consumo de energia e concluem que a comunicação *single-hop* pode ser mais eficiente que a *multi-hop* quando modelos de rádio reais são adotados. Embora este trabalho apresente resultados interessantes, a análise foi baseada apenas em uma topologia de rede linear, uma avaliação em cenários mais realísticos é necessária.

Outras propostas consideram a integração entre as camadas MAC e de roteamento. Em [Sichitiu 2004] foi proposto um esquema de conservação de energia baseada no escalonamento dos nós sensores. A idéia é bastante simples, no esquema proposto, os nós criam dinamicamente um escalonamento *on-off* em que os nós permanecerão ativos durante certo período e inativos o restante do tempo. O algoritmo de roteamento utiliza esse escalonamento para encaminhar os dados para a estação base. Claramente, essa abordagem só poderá ser utilizada em redes onde os nós sensores estejam sincronizados. Outro exemplo desta integração foi proposto em [Skraba 2004]. Neste esquema, um algoritmo de roteamento geográfico utiliza informações implicitamente distribuídas da localização de cada nó, obtidas na camada de enlace, para tomar as decisões de roteamento.

Além das soluções que consideram a interação entre duas camadas, os trabalhos em [Madan 2005] [Cui 2004] mostram que a otimização integrada entre múltiplas camadas (enlace, MAC e roteamento) é uma abordagem viável e benéfica para implementar eficiência no consumo de energia das redes de sensores.

## 3. Proposta de Projeto Integrado MAC, Roteamento e Aplicação

Esta seção explora a forte interação existente entre as camadas de roteamento e de acesso ao meio (MAC) e em conjunto com a aplicação propõe um projeto integrado entre estas camadas. O objetivo é mostrar que a comunicação direta entre os protocolos pode contribuir para a melhoria de desempenho e do tempo de vida das redes de sensores.

No projeto integrado proposto, as rotas são estabelecidas tomando como base as informações de energia dos nós (camada física) e da conectividade dos nós (camada de enlace) para construir rotas mais eficazes. A existência dessas rotas é conhecida pela aplicação que portanto, só envia mensagens quando certa do caminho fim-a-fim. Esse trabalho estende o protocolo de roteamento adaptativo OPER (*On-Demand Power-Efficient Routing Protocol*) [Gomes 2006] [Gomes 2005] para possibilitar o compartilhamento de informações entre as camadas. A interação com a camada de enlace é baseada no protocolo B-MAC (*Berkeley MAC*) [Polastre 2004] que além de oferecer um conjunto de serviços pré-definidos para as camadas mais altas, permite a reconfiguração “*on-the-fly*” do protocolo e está disponível na plataforma real de sensores Mica2 [Crossbow Technology]. Uma breve descrição destes protocolos é

apresentada a seguir. É importante salientar que não é o foco desse trabalho avaliar o desempenho individual do protocolo B-MAC ou do protocolo OPER.

### 3.1. O Protocolo B-MAC

B-MAC é um protocolo de controle de acesso ao meio baseado em detecção de portadora, semelhante ao CSMA. Sua principal meta é reduzir o consumo de energia enquanto melhora a utilização do canal e evita colisão.

O B-MAC foi projetado para tornar mais flexível o controle do protocolo MAC pelos serviços das camadas mais altas. O protocolo fornece um conjunto de interfaces que permite aos serviços de rede ajustar seus mecanismos de funcionamento. Os principais mecanismos do B-MAC são o CCA (*Clear Channel Assessment*), usado para evitar colisão e o LPL (*Low Power Listening*) cuja principal função é evitar o desperdício de energia pelo rádio. A Figura 1 exhibe algumas das interfaces fornecidas pelo protocolo B-MAC.

<pre> Interface MacControl {     command result_t EnableCCA ();     command result_t DisableCCA ();     command result_t EnableAck ();     command result_t DisableAck ();     command void* HaltTx (); } </pre>	<pre> Interface LowPowerListening {     command result_t SetListeningMode (uint8_t mode);     command uint8_t GetListeningMode ();     command result_t SetTransmitMode (uint8_t mode);     command uint8_t GetTransmitMode ();     command result_t SetPreambleLength (uint16_t bytes);     command uint16_t GetPreambleLength ();     command result_t SetCheckInterval (uint16_t ms);     command uint16_t GetCheckInterval (); } </pre>
--	---

**Figura 1. Exemplos de interfaces que permitem um controle flexível do B-MAC pelos serviços das camadas mais altas.**

Pela exposição de um conjunto de mecanismos configuráveis, os protocolos construídos acima do protocolo B-MAC podem tomar decisões de políticas locais para otimizar o consumo de energia, latência, vazão ou confiabilidade.

#### Processo de Amostragem do Preâmbulo

O protocolo B-MAC procura ser eficiente em energia empregando um esquema adaptativo de amostragem do preâmbulo para reduzir a utilização do rádio (*duty cycle*) e minimizar a escuta ociosa, o qual é denominado de LPL (*Low Power Listening*).

Neste esquema, o nó ao acordar liga o rádio e checa por atividade. Se alguma atividade é detectada, o nó permanece ativo pelo tempo necessário para receber o quadro que está chegando. Após a recepção, o nó retorna a dormir. Se nenhum quadro for recebido, um *timeout* forçará o nó a voltar a dormir. O CCA é usado não só para encontrar o canal livre para uma transmissão, mas também para determinar se existe atividade durante o período LPL.

Para receber os dados de forma confiável, o tamanho do preâmbulo é casado com o intervalo em que o canal é checado por atividade. Se o canal é checado a cada 100 ms, o preâmbulo deve ter ao menos 100 ms para que o nó acorde, detecte atividade no canal, receba o preâmbulo e então receba a mensagem. A escuta ociosa ocorre quando o nó acorda para amostrar o canal e não existe nenhuma atividade. O intervalo em que o canal é checado e o tamanho do preâmbulo são exemplos de parâmetros configuráveis através das interfaces disponibilizadas pelo protocolo B-MAC.

### Processo de Amostragem do Canal

Para efetivamente evitar colisões, o protocolo B-MAC é capaz de determinar precisamente se o canal está livre através de uma heurística chamada de CCA. Essa heurística é responsável por verificar se existe atividade no canal e por retornar a informação para a aplicação. O CCA emprega a técnica de julgamento de canal baseado em uma estimativa de ruído do canal, obtida através do algoritmo que mede a força do sinal recebido (RSSI - *Received Signal Strength Indicator*). Após o processo de amostragem do canal, um limiar é estabelecido para decidir se o canal está ou não pronto para ser usado.

Esta heurística funciona da seguinte forma: uma vez que o ruído do ambiente muda em função do mesmo, o B-MAC emprega um controle de ganho automático para estimar o ruído do canal. Amostras da força do sinal são obtidas frequentemente e armazenadas em uma fila FIFO. A média dessas amostras é usada para estimar a qualidade do canal. Uma vez que uma boa estimativa de ruído no canal está estabelecida, o processo de monitoramento da força do sinal é iniciado através da transmissão de quadros especiais pelo rádio. O procedimento seguinte consiste em comparar se o sinal recebido está acima ou abaixo de certos limiares estabelecidos durante a amostragem do canal. Tal informação indicará se o canal está ou não livre para transmissão.

O B-MAC também fornece um suporte opcional para confirmação. Caso este suporte esteja habilitado, o B-MAC imediatamente transfere um código de confirmação após receber um pacote *unicast*.

### 3.2. O Protocolo de Roteamento OPER

O OPER é uma família de protocolos de roteamento sob demanda para redes de sensores sem fio. O objetivo da família OPER é realizar a disseminação dos dados coletados pelos nós sensores levando em consideração a quantidade de energia dos nós que compõem uma rota. Esta seção descreve o funcionamento do protocolo OPER. Detalhes de suas variações podem ser obtidos em [Gomes 2006][Gomes 2005].

#### 3.2.1. O Algoritmo OPER

O OPER combina características como o uso do número de seqüência do destino, para conceber rotas livres de *loops* e técnicas de descoberta de rotas sob demanda semelhante às utilizadas pelos protocolos DSR e AODV.

Para garantir a economia de banda e energia, o processo de aquisição de rotas implementado no OPER atua sob demanda. O processo de descoberta de rota é iniciado sempre que uma fonte necessita de uma rota para um destino, que tipicamente envolve a inundação de mensagens de requisição de rotas e a espera por mensagens de resposta de rota. O processo de manutenção de rotas é realizado através de mensagens de erro de rota, que permitem detectar e informar a queda de um enlace, como também pelo tempo de expiração de rotas que exige novos pedidos de rotas.

O protocolo utiliza ainda um mecanismo de detecção de vizinhança para detectar a queda de um enlace. Neste mecanismo, mensagens de broadcast local, chamadas de *hello*, são enviadas periodicamente para confirmar a conectividade local entre vizinhos. O não recebimento de uma mensagem de *hello* de um vizinho durante certo período de tempo, indica a queda de um enlace.

### 3.2.2. As Fases do OPER

O algoritmo OPER é composto pelas fases de descoberta de vizinhança, descoberta de rotas e manutenção de rotas.

#### Descoberta de Vizinhos

O processo de descoberta de vizinhos tem como objetivo permitir que cada nó possa criar sua lista de vizinhança e através desta, saber com quais nós pode se comunicar diretamente. Esse procedimento é realizado através da troca de mensagens de *hello*. Quando um nó recebe uma mensagem de *hello*, deve verificar se a origem dessa mensagem já se encontra na sua lista de vizinhança. Se for encontrado algum registro, isso indicará que a conectividade entre os nós foi mantida. Caso não exista nenhuma entrada, trata-se de um novo vizinho e, portanto, o identificador da fonte dessa mensagem deverá ser adicionado à lista de vizinhança.

Se dentro de um período de tempo, definido de acordo com as características da aplicação, um nó não receber nenhuma mensagem de *hello* de algum dos seus vizinhos, deverá supor que a conexão com este vizinho está perdida. Neste caso, o nó deverá retirá-lo da sua lista de vizinhança e realizar o processo de notificação de falha no enlace que será apresentado posteriormente.

#### Descoberta de Rotas

O processo de descoberta de rotas é iniciado quando um nó sensor necessita se comunicar com outro nó o qual não tem nenhuma informação de rota em sua tabela de roteamento. O fato da formação de rotas ocorrer apenas quando necessária é o que torna o algoritmo OPER “sob demanda”. Para descrever o processo de descoberta de rota do OPER, considere a rede de sensores da Figura 2, em que um nó A deseja transmitir para o nó F (nó sorvedouro).

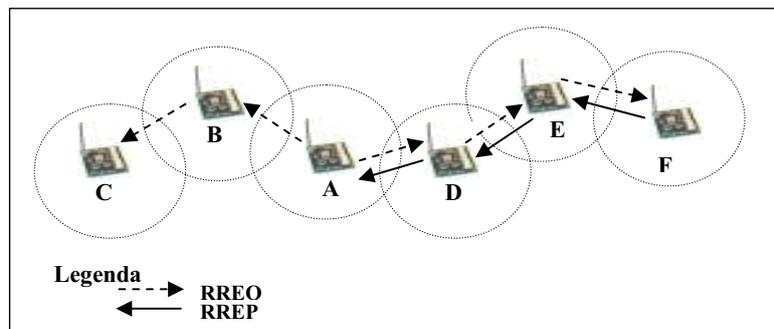


Figura 2. O processo de descoberta de rotas no OPER.

Os círculos na Figura 2 ilustram a faixa de comunicação de cada nó. Devido a limitações da potência de transmissão, cada nó só pode se comunicar diretamente com os vizinhos mais próximos. Para localizar F, o nó A envia uma mensagem de requisição de rota (RREQ - *Route REQuest*) em *broadcast*. Nós vizinhos a A, neste caso os nós B e D, verificam se possuem uma entrada em suas tabelas. Se B ou D conhecem uma rota até o destino, transmitem uma mensagem de resposta de rota (RREP - *Route REPlay*) informando como chegar ao destino. Caso B ou D não conheçam uma rota até o destino, a mensagem RREQ é retransmitida. Antes da retransmissão algumas informações são extraídas para que seja possível construir a rota inversa até a fonte.

Se uma resposta para a requisição de rota não for recebida dentro de um determinado período de tempo pré-estabelecido, o nó poderá enviar uma nova

mensagem de requisição, desde que a quantidade de tentativas não supere a quantidade máxima estipulada, caso contrário o destino será assinalado como inalcançável.

A métrica de energia usada para avaliar as rotas é um importante componente do protocolo. Dependendo da métrica, as características do protocolo podem mudar substancialmente. Nas diferentes implementações do protocolo OPER, essa métrica pode incluir informações sobre o número de nós, quantidade de energia dos nós ao longo do caminho, o custo do uso do caminho, entre outras.

Para selecionar o melhor caminho e minimizar o total de energia necessária para rotear pacotes, maximizando o tempo de vida de todos os nós sensores, o esquema CMMBCR (*Conditional Min-Max Battery Cost Routing*) semelhante ao proposto em [Toh 2001] será utilizado. A intuição por trás dessa métrica é simples, usando somente os nós que tenham carga de bateria acima de um limiar  $\gamma$ , encontrar a rota com o máximo de energia residual.

Observe que nesse ponto do algoritmo foi introduzido o primeiro mecanismo para permitir o controle de energia dos nós da rede, realizado através da aceitação seletiva das requisições de rota. O objetivo deste mecanismo é evitar que nós sensores, com pouca energia, sejam usados como intermediários para o estabelecimento de novas rotas.

### **Manutenção de Rotas**

Uma vez estabelecida uma rota entre dois nós da rede é necessária a sua manutenção através da verificação de falhas de conectividade. Ao detectar-se uma falha de conexão quatro etapas são seguidas para permitir a adoção de medidas corretivas: primeiro verifica-se quais destinos foram afetados; as rotas que utilizavam esse destino são então invalidadas; determina-se quais os vizinhos afetados; por fim, envia-se uma mensagem de erro de rota (RERR - *Route ERROR*) para estes vizinhos informando a falha.

A manutenção da conectividade local é realizada através do mesmo mecanismo usado pelo processo de descoberta dos vizinhos, ou seja, através da troca periódica de mensagens de *hello*. Cada nó utiliza a atualização da lista de vizinhança para verificar se os nós inativos fazem parte de alguma rota ativa em sua tabela de roteamento. O procedimento de manutenção é realizado da seguinte forma: quando qualquer vizinho se torna inacessível, o nó verifica sua tabela de roteamento para ver quais destinos possuem rotas que utilizam o vizinho agora inativo. Após retirar de sua tabela as rotas inválidas, o nó comunica o ocorrido aos seus vizinhos ativos que, em seguida, comunicam aos seus próprios vizinhos e assim por diante, até todas as rotas que dependem do nó inativo sejam invalidadas.

### **3.3. Interações Cross-layer Propostas**

A camada de roteamento e a camada MAC possuem algumas funcionalidades que se sobrepõem: ambas possuem diferentes requisitos para a bidirecionalidade do enlace, podem implementar testes para bidirecionalidade e mecanismos de confirmação de mensagens e podem detectar problemas no enlace.

A principal meta é garantir que enlaces unidirecionais não sejam usados pelo protocolo OPER, quando o protocolo MAC requerer enlaces bidirecionais. Isto é importante porque pode evitar transmissões desnecessárias de mensagens e, conseqüentemente, o desperdício de energia dos nós sensores.

O projeto *cross-layer* proposto neste trabalho é principalmente baseado na troca das informações de conectividade da camada de enlace para a camada de roteamento. Para garantir que os quadros foram corretamente entregues, o protocolo B-MAC implementa um mecanismo de confirmação de mensagens (*Acknowledgment – Ack*). O princípio é bastante simples: cada vez que um nó recebe um quadro, este envia imediatamente de volta um *Ack* para o transmissor indicando que o quadro foi entregue sem erro. O propósito do projeto *cross-layer* é baseado no compartilhamento das confirmações ocorridas no nível de enlace com o protocolo de roteamento OPER. Tendo acesso a essa informação, o protocolo OPER pode:

1. Atualizar de forma mais rápida a tabela de vizinhança dos nós;
2. Inferir sobre possíveis problemas de conectividade entre vizinhos. Por exemplo, sucessivas retransmissões na camada de enlace podem representar problemas no enlace ou no nó receptor;
3. Baseado em 1 e 2, eliminar a mensagem de *hello*;
4. Decidir se o enlace é simétrico para garantir o sucesso das transmissões de mensagens RREP.

Portanto, o protocolo OPER pode ser otimizado para não usar as mensagens de *hello* e sim, obter informações de vizinhança diretamente da camada MAC usando uma interação *cross-layer* específica. Esta integração possibilita ao protocolo de roteamento reagir melhor e mais rápido às mudanças dinâmicas da rede.

Por fim, o projeto proposto neste trabalho também oferece uma simples integração entre a camada de aplicação e o protocolo de roteamento OPER. Quando não existe um caminho fim-a-fim disponível para o destino, todas as mensagens para aquele destino são enfileiradas até que o caminho esteja disponível. Dependendo da taxa em que os dados são gerados e do tamanho do buffer, isto pode facilmente conduzir à perda de pacotes. Além disso, o atraso associado pode ser problemático principalmente para aplicações sensíveis ao atraso. Para resolver este problema, a aplicação consulta o protocolo de roteamento para saber se existe uma rota para um destino antes de repassar a mensagem para a pilha de protocolo. Caso exista, a mensagem segue o fluxo normal, ou seja, é repassada para a camada de roteamento. Caso contrário, a aplicação solicita que seja estabelecida uma nova rota. O propósito dessa interação resulta na melhoria da taxa de entrega de pacotes e no atraso médio fim-a-fim.

## 4. Simulação e Resultados

O desempenho do projeto integrado proposto é avaliado através de simulação. Esta seção descreve as métricas de desempenho, o ambiente de simulação e os resultados experimentais.

### 4.1. Métricas de Desempenho

Para analisar e comparar o desempenho do projeto integrado entre o protocolo MAC e o algoritmo de roteamento proposto serão usadas as seguintes métricas:

- Taxa de Entrega: definida como a relação entre o número total de mensagens recebidas pelo nó sorvedouro e a quantidade de mensagens enviadas pelos nós sensores. Esta métrica fornece um indicativo da eficiência de utilização da largura de banda da rede;

- Número de Mensagens de Controle: definida como a quantidade de mensagens de controle necessárias para estabelecer rotas entre os nós sensores e o nó sorvedouro. Esta métrica fornece um indicativo sobre a eficiência de comunicação do protocolo de roteamento;
- Eficiência no Consumo de Energia: definida como a quantidade de mensagens transmitidas com sucesso sobre o total de energia dissipada pelos nós sensores. Esta métrica indica a eficiência no consumo de energia;
- Número Total de Nós Vivos: indica o tempo de vida geral da rede. Esta métrica dá uma idéia da área de cobertura da rede em função do tempo.

#### 4.2. Ambiente de Simulação

Para avaliar o desempenho da abordagem proposta foram realizadas diversas simulações em redes de sensores compostas por 50, 100, 150 e 200 nós sensores estacionários e homogêneos, distribuídos randomicamente em áreas de  $50 \times 50 \text{m}^2$  para as duas primeiras redes e  $100 \times 100 \text{m}^2$  para as duas últimas. A faixa de transmissão de um nó sensor é circular com um raio de 10m.

Os experimentos foram realizados no simulador TOSSIM [Levis 2003], em conjunto com o módulo de cálculo de consumo de energia denominado PowerTossim [Shnayder 2004], utilizando o modelo de energia do Mica2. O algoritmo de roteamento OPER foi implementado e incorporado a pilha de protocolos implementada no sistema operacional TinyOS versão 1.15 [TinyOS Project]. O tempo de simulação foi de 600s. Durante o “*startup*” das simulações, os nós são ativados aleatoriamente, sendo o período de ativação randômico no intervalo de 10, 20, 30 ou 40 segundos para redes compostas por 50, 100, 150 e 200 nós respectivamente.

A quantidade de energia inicial de cada nó sensor foi definida como 3000  $\mu\text{Joules}$ . O processamento “*in-network*” é realizado por cada nó participante de uma rota e, portanto, todo dado é encaminhado, em múltiplos saltos, para uma estação base localizada sempre no centro da rede. É assumido que a estação base tem energia suficiente para atender as solicitações de qualquer nó na rede.

Foram realizados experimentos com diferentes taxas de dados. Os resultados apresentados neste trabalho correspondem a uma aplicação que envia uma mensagem de dados de 36 bytes. O intervalo de envio de mensagens é escolhido aleatoriamente, para cada transmissão, no intervalo de 0 a 30s ou 0 a 60s para os cenários de alta e baixa carga respectivamente. Em cada cenário foram escolhidos aleatoriamente 50% do total de nós da rede para serem responsáveis pela geração de dados. A Tabela 1 fornece os valores para os parâmetros essenciais do protocolo OPER. Esses valores foram escolhidos visando minimizar o congestionamento da rede.

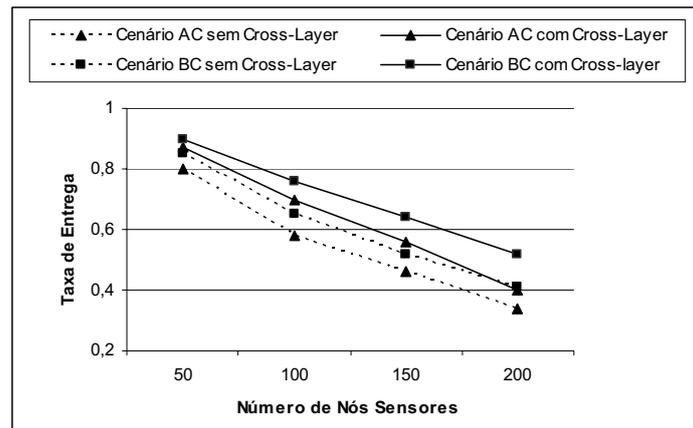
**Tabela 1. Parâmetros usados pelo OPER nas simulações.**

Parâmetro	Valor	Descrição
Hello_Interval	40 segundos	Intervalo de envio de uma mensagem de <i>hello</i>
Route_Exp_Time	120 segundos	Tempo de validade da rota
OPER_Max_Retransmission	4	Número máximo de tentativas de retransmissão
Energy_Threshold	10% da capacidade da bateria inicial	Limiar de energia $\alpha$ que decide se o nó irá participar da rota

Para cada experimento, 30 diferentes simulações foram executadas. Os gráficos plotados, a seguir, representam a média dos resultados para todos os experimentos com um intervalo de confiança de 95%.

### 4.3. Resultados

O primeiro experimento compara a taxa de entrega de pacotes obtida pela abordagem tradicional (sem otimização *cross-layer*) com a abordagem *cross-layer* proposta neste trabalho. A Figura 3 apresenta a taxa de entrega para os cenários de alta carga (AC) e baixa carga (BC) variando o número de nós sensores.



**Figura 3.** Taxa de entrega usando a abordagem sem e com *cross-layer* para os cenários de alta carga (cenário AC) e baixa carga (cenário BC) para redes compostas por 50, 100, 150 e 200 nós sensores.

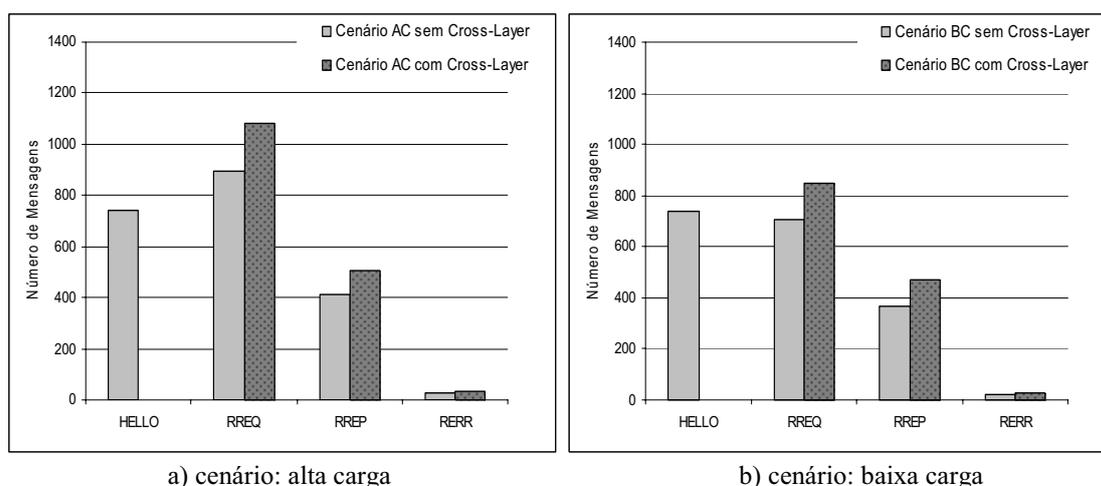
Como observado na Figura 3, o uso da abordagem *cross-layer* apresenta melhores percentuais de entrega das mensagens nos dois cenários avaliados (alta e baixa carga). Para a rede formada por 50 nós, os ganhos obtidos com a otimização *cross-layer* são relativamente pequenos, apenas 8% para o cenário AC e de 5% para o cenário BC, se comparados com a abordagem tradicional. Entretanto, nota-se claramente uma melhora significativa na taxa de entrega de mensagens de 0,58 para 0,7, de 0,46 para 0,57 e de 0,34 para 0,4, representando uma melhoria de aproximadamente 17%, 18% e 15% para as redes com 100, 150 e 200 nós no cenário de alta carga. Enquanto que, no cenário de baixa carga, os ganhos foram de 14%, 18% e 21 % para as redes com 100, 150 e 200 nós sensores quando comparados com a abordagem tradicional.

Nos experimentos de alta carga, uma mensagem é enviada num intervalo randômico de até 30 segundos criando uma situação de alta contenção. Conseqüentemente, com o aumento do número de nós, a média de entrega da rede cai devido a maior disputa pelo meio e a capacidade de transmissão da B-MAC. Tal fato pode ser observado através da análise dos resultados dos experimentos realizados com 200 nós sensores, onde a taxa de entrega obtida foi de apenas 0,34 usando a abordagem tradicional e 0,40 usando a abordagem *cross-layer*.

Nos experimentos de baixa carga, o melhor desempenho da abordagem *cross-layer* pode ser explicado através de um conjunto de fatores. Primeiro, a taxa de transmissão é de uma mensagem em até 60 segundos, ou seja, existe uma menor disputa pelo meio. Mesmo para uma rede com 200 nós, a taxa de entrega obtida foi 21% superior à abordagem tradicional. Segundo, na abordagem *cross-layer* a aplicação

consulta a camada de roteamento para saber se existe uma rota para o destino desejado. Caso a resposta seja positiva, a aplicação pode enviar o dado imediatamente para o destino. Caso contrário, a aplicação mantém em seu buffer o dado a ser transmitido até que uma rota seja estabelecida ou que a validade do dado expire. Nos experimentos, o tempo de validade de um dado na camada de aplicação foi de 2000 milissegundos. O processo de consulta por rota e de utilização de buffer é importante porque resulta na melhoria da taxa de entrega de pacotes e na diminuição do atraso médio fim-a-fim, pois diminui a perda de pacotes e, conseqüentemente, o processo de retransmissão. Este processo também poderia ser usado pela aplicação para realizar ajustes no controle do tamanho do *buffer*, na política de descarte e/ou nas estratégias de agregação de dados. Terceiro, a eliminação da mensagem de *hello* possibilitou uma melhoria no uso do canal. Por último, mas não menos importante, a atualização da lista de vizinhança pela camada de enlace proporcionou à abordagem *cross-layer* uma resposta mais rápida e eficiente às mudanças de topologia.

Enquanto a mensagem de *hello* permite que os nós aprendam sobre as mudanças de vizinhos, esta também cria um custo extra de controle e um aumento no consumo da largura de banda. A Figura 4 mostra a média de mensagens de controle usadas nos experimentos com redes com 100 nós sensores. Como o número total de mensagens de controle aumenta linearmente com o tamanho da rede, os gráficos dos experimentos com 50, 150 e 200 nós não serão plotados.



**Figura 4. Quantidade média de mensagens de controle usadas em redes com 100 nós, durante 600 segundos de simulação, para os cenários de alta e baixa carga.**

Na abordagem *cross-layer*, a eliminação da mensagem de *hello* representa em média a diminuição de 743 mensagens para o cenário de alta carga e de 738 mensagens para o cenário de baixa carga. Como mostrado na Tabela 2, a mensagem de *hello* corresponde a 35,8% do fluxo de mensagens de controle no cenário de alta carga e de 40,2% no cenário de baixa carga. Como não existe um canal exclusivo para o tráfego das mensagens de controle, as mensagens de *hello* acabam não só competindo com as outras mensagens de controle como também com o tráfego de dados. Além disso, estas também interferem na disputa interna pelos recursos do nó (rádio e CPU). Estas afirmações foram comprovadas pelos experimentos realizados sem a mensagem de *hello*, onde ocorreu um aumento do número de mensagens RREQ e RREP. Nos experimentos com 100 nós sensores, a eliminação da mensagem de *hello* e o aumento

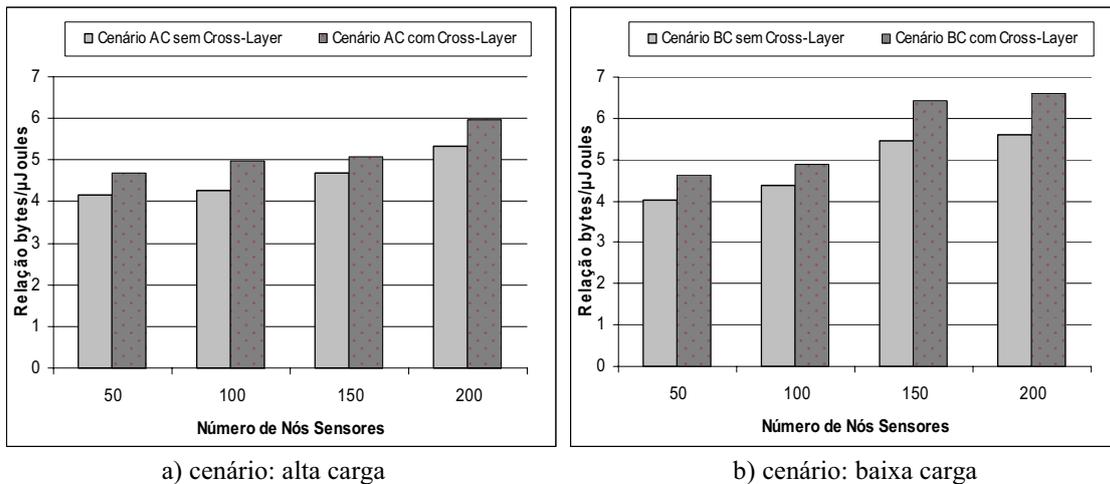
das mensagens RREQ e RREP representam uma diminuição média de apenas 284 e 243 mensagens de controle para os cenários de alta e baixa carga respectivamente.

Portanto, a eficiência obtida na entrega de mensagem mostrada na Figura 3 é resultante não só da diminuição das mensagens de controle, mas principalmente pela eficiência obtida através dos processos de descoberta e manutenção de vizinhança, alimentados com informações fornecidas pela camada B-MAC (abordagem com *cross-layer*).

**Tabela 2. Total de mensagens de controle e o percentual representativo da mensagem de *hello* para simulações com 50, 100, 150 e 200 nós sensores.**

# Nós	Alta Carga			Baixa Carga		
	Total Msg	Total Hello	% Hello	Total Msg	Total Hello	% Hello
50	1339	506	37,7	1122	507	45,1
100	2075	743	35,8	1833	738	40,2
150	3384	1149	33,9	3165	1140	36,0
200	4184	1240	29,6	3944	1238	31,3

Outra avaliação importante é a energia consumida. Os valores apresentados nos gráficos da Figura 5 representam a média da relação entre bytes eficientemente transmitidos por Joules consumidos para as redes com 50, 100, 150 e 200 nós sensores.



**Figura 5. Relação entre mensagens de dados transmitidas (em bytes) com sucesso por energia consumida para as redes com 50, 100, 150 e 200 nós sensores após 600 segundos de simulação, para os cenários de alta e baixa carga.**

A Figura 5 mostra que a abordagem *cross-layer* obtém um menor consumo de energia por byte enviado em todos os cenários avaliados. Os ganhos obtidos com o projeto *cross-layer* representam 11,8%, 14,4%, 8,1% e 10,2% para redes com 50, 100, 150 e 200 nós sensores no cenário de alta carga e de 12,8%, 10,6%, 14,8% e 14,7%, para redes com 50, 100, 150 e 200 nós sensores no cenário de baixa carga. Pode-se atribuir esses ganhos às duas principais otimizações realizadas. Na abordagem *cross-layer*, a camada de aplicação possui ciência de rota, o que possibilitou a melhoria da taxa de entrega da rede e, conseqüentemente, diminuição no consumo de energia. Nos experimentos realizados, o modelo de energia utilizado no cálculo de rotas também leva em consideração a quantidade de vizinhos. Desta forma, o processo de seleção de rota

tornou-se mais eficiente com a atualização da tabela de vizinhança sendo realizada através das informações advindas da camada de enlace.

O melhor desempenho da abordagem *cross-layer* proposta também pode ser comprovado avaliando o número total de nós que permanecem vivos durante o tempo de simulação (1500 segundos), como mostrado na Figura 6. Além de transmitir uma maior quantidade de dados, os nós permanecem vivos por uma maior quantidade de tempo.

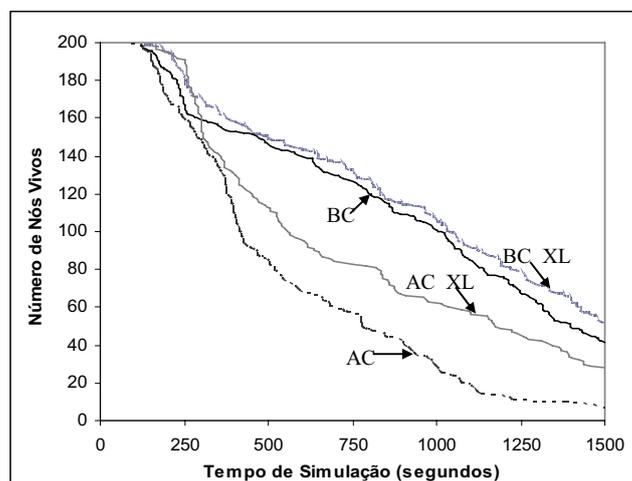


Figura 6. Número de nós sensores vivos para uma rede com 200 nós sensores após 1500 segundos de simulação, para os cenários de alta (AC) e baixa carga (BC) com e sem *cross-layer* (XL).

## 5. Conclusão

Este artigo apresentou o projeto integrado entre as camadas de roteamento, MAC e aplicação. As interações entre as camadas possibilitaram uma melhoria de desempenho em todos os cenários avaliados. Os resultados de simulação mostram que o projeto integrado apresenta melhores índices de eficiência na utilização da largura de banda, na comunicação do protocolo de roteamento e no tempo médio de vida dos nós sensores quando comparados com a abordagem tradicional.

O uso das informações de vizinhanças diretamente da B-MAC possibilitou ao protocolo OPER reagir melhor e mais eficiente às mudanças dinâmicas da rede. Essa forte interação entre a camada de roteamento e MAC mostrou que o projeto integrado entre camadas é uma escolha promissora para superar os limites impostos pelos escassos recursos das redes de sensores.

## Referências

- X. Liu, E. K. P. Chong, and N. B. Shro, "A Framework for Opportunistic Scheduling in Wireless Networks," in *Computer Networks*, March 2003, vol. 4, pp. 451-474.
- A. Santhanam, R. Cruz, "Optimal routing, link scheduling and power control in multihop wireless networks," in *Proc. of IEEE INFOCOM*, San Francisco, California, USA, Apr. 2003, vol. 1, pp. 702-711.
- L. Chen, S. H. Low, and J. C. Doyle. "Joint Congestion Control and Media Access Control Design for Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, USA, March 2005, vol. 3, pp. 2212-2222.

- V. Kawadia and P.R. Kumar, "A Cautionary Perspective on Cross Layer Design," in IEEE Wireless Communication Magazine, February 2005, vol. 12, no. 1, pp. 3-11.
- I. F. Akyildiz, M. C. Vuran, and O. B. Akan, "A Cross-Layer Protocol for Wireless Sensor Networks," in Proc. Conference on Information Sciences and Systems, Princeton, NJ, USA, March 2006, pp. 1102-1107.
- J. Haapola, Z. Shelby, C. Pomalaza-Raez, P. Mahonen, "Cross-layer Energy Analysis of Multi-hop Wireless Sensor Networks," in EWSN '05, January 2005, pp. 33-44.
- M. L. Sichitiu, "Cross-layer Scheduling for Power Efficiency in Wireless Sensor Networks," in Proc. IEEE Infocom, USA, March 2004, vol. 3, pp 1740-1750.
- P. Skraba, H. Aghajan, A. Bahai, "Cross-layer Optimization for High Density Sensor Networks: Distributed Passive Routing Decisions," in Proc. Ad-Hoc Now'04, Vancouver, Canada, July 2004, vol. 3158, pp. 266-279.
- R. Madan, S. Cui, S. Lall, and A. J. Goldsmith, "Cross-layer Design for Lifetime Maximization in Interference-limited Wireless Sensor Networks," in Proc. IEEE Infocom, CA, USA, March 2005, vol. 3, pp 1964 - 1975
- S. Cui, R. Madan, S. Lall, A. J. Goldsmith, "Energy Minimization and Delay Analysis in TDMA-based Sensor Networks," in IEEE Trans. on Wireless Communications, April 2004.
- R. Gomes, E. Souto, Kelner, D. Sadok, "Evaluation of Energy Heuristics to On-Demand Routes Establishment in Wireless Sensor Networks," in Proceedings of Third Annual Conference on Wireless On-demand Network Systems and Services, Les Ménuieres, France, January 2006, pp. 179-184.
- R. Gomes, E. Souto, J. Kelner, D. Sadok, "Evaluating Energy Mechanisms to Routing in Wireless Sensor Networks," in Proceedings of Mobility Conference (Mobility 2005), China, November 2005, vol. 01, pp. 848-855.
- J. Polastre, J. Hill, D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, USA, November 2004, pp. 95-107.
- Crossbow Technology Inc. Available in: <http://www.xbow.com/>
- C. K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks," in IEEE Communication Magazine, June 2001, vol. 39, no. 6, pp. 138-147.
- P. Levis, N. Lee, M. Welsh e D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," in Proc. of First International Conference on Embedded Networked Sensor Systems, CA, EUA, 2003, pp. 126-137.
- V. Shnayder, M. Hempstead, B. Chen, G. Allen e M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications," in 2nd Conference on Embedded Networked Sensor Systems, Baltimore, EUA, 2004, pp. 188-200.
- TinyOS Project. Available in: <http://www.tinyos.net/>