Compressão Adaptativa de Arquivos HTML em Ambientes de Comunicação Sem Fio

Rainer R. Couto , Ricardo A. Rabelo , Antonio A. F. Loureiro

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Av. Antonio Carlos, 6627
Belo Horizonte, MG, Brasil

{rainerpc,rabelo,loureiro}@dcc.ufmg.br

Abstract. *The increasing development of mobile computing technologies has allowed users to access the Internet any time and anywhere. However, as the resources of mobile devices are scarcer than their counterparts in wired network, we need to adapt HTML data in order to provide an efficient access to Web documents in wireless environments. In this work we propose a model which predicts how and when a file should be compressed before its transmission, aiming to minimize the impact over energy consumption and to improve response time. Our experiments and simulations with IEEE 802.11 and Bluetooth prove the efficiency of this model when compared with simplistic approaches adopted in literature.*

Resumo. *O atual desenvolvimento das diversas tecnologias da computação móvel permite a um usuário acessar a Internet de qualquer lugar e a qualquer instante. Entretanto, para prover um acesso eficiente nesse ambiente, necessitamos de algum processo de adaptação do conteúdo da Web, uma vez que os recursos dos dispositivos móveis são inerentemente escassos. Nesse trabalho propomos um modelo para adaptação de conteúdo HTML que prevê, dinamicamente, quando um arquivo deve ser comprimido antes de sua transmissão. Nossos experimentos e simulações com os ambientes Bluetooth e IEEE 802.11 comprovam a eficácia e a factibilidade desse modelo quando comparado às técnicas atuais de adaptação por compressão.*

1 Introdução

As diversas classes de dispositivos utilizados na computação móvel apresentam características diferentes entre si [1]. Entre todas essas características, as mais relevantes para a distinção dos elementos móveis são: capacidade de processamento, capacidade de transmissão e gasto de energia. A capacidade de processamento é notadamente inferior à dos elementos da rede fixa. A capacidade de transmissão e o gasto de energia são regulados pela implementação e pela forma de comunicação como, por exemplo, os protocolos de acesso ao meio. Dispositivos projetados para o protocolo 802.11 possuem uma grande capacidade de transmissão e um gasto maior de energia quando comparados aos elementos projetados para o protocolo Bluetooth.

Tais propriedades representam restrições no projeto de aplicações para estes dispositivos [2]. Por exemplo, as aplicações devem ser desenvolvidas visando minimizar o gasto dos recursos escassos, como energia e largura de banda.

A chave para a economia destes recursos escassos é a *adaptabilidade* [1, 2, 3]. A adaptação consiste na alteração do comportamento das aplicações de acordo com as condições

do ambiente. A adaptabilidade se mostra mais eficiente quando tenta analisar os recursos disponíveis em tempo real e, a partir daí, utiliza métodos mais econômicos para a execução das aplicações.

Dentro dos diferentes ambientes onde a computação móvel pode estar presente, o mais comum e abrangente é a WWW. O formato mais geral de circulação na Web é o HTML e suas variantes (ASP, JSP, PHP, XML, DHTML, entre outros). O termo variantes aqui não se aplica ao aspecto comportamental ou funcional dos demais formatos, e sim no sentido de que todos eles se traduzem em efeitos de formação de documentos no cliente. Além disso, todos esses formatos se caracterizam por serem verbosos, o que acarreta uma maior carga de transmissão de dados. Portanto, daqui em diante, a utilização do termo HTML estará, arbitrariamente, fazendo menção também a todas as suas citadas formas de variação.

Realizar a compressão desses dados antes de suas respectivas transmissões parece ser uma solução óbvia de adaptação. Este trabalho explora quais são as condições adequadas nas quais a compressão deve realmente ocorrer e como ela deve ser feita. Os objetivos deste trabalho, portanto, são:

- Descobrir qual método de compressão é mais apropriado a ser utilizado em função das condições do ambiente no momento da transmissão;
- Propor um modelo capaz de prever quando a compressão deve ser usada;
- Avaliar quantitativamente o ganho com o uso do modelo proposto em dois ambiente de comunicação (Bluetooth e IEEE 802.11) através de simulação e experimentação.

Este trabalho está estruturado da seguinte forma: na seção 2 apresentamos alguns trabalhos relacionados; na seção 3 analisaremos as vantagens e desvantagens de diferentes métodos de compressão e qual deles é o mais adequado para a utilização em ambientes móveis. Na seção 4 propomos um modelo que tenta prever quando um arquivo HTML deve ser comprimido antes de sua transmissão em meio sem-fio. A seguir, na seção 5, analisamos dois protocolos de comunicação em ambientes sem-fio, Bluetooth e IEEE 802.11, que são utilizados nas simulações realizadas para validar o modelo. Os resultados são apresentados nas seções 6 e 7.

2 Trabalhos Relacionados

Vários são os trabalhos que estudam a adaptação de conteúdo HTTP para ambientes móveis. [4] utiliza compressão de dados e *delta-enconding* - uma técnica que propõe a atualização de documentos em *cache* enviando-se apenas a parte modificada - para minizar a quantidade de dados transmitidas e o tempo de resposta. O trabalho propõe modificações específicas para o protocolo HTTP. [5] descreve uma tecnologia de *software*, WebExpress, que facilita a tarefa de navegação Web em ambientes sem-fio. WebExpress é implementado como um *middleware* que intercepta uma seqüência de requisições HTTP e aplica uma série de transformações sobre a mesma, a fim de reduzir o custo de transmissão e o tempo de resposta da comunicação. As otimizações incluem *caching*, redução de protocolo e eliminação de cabeçalho HTTP redundante. [6] propõe uma arquitetura de *middleware* para adaptação de conteúdo Web para ambientes sem-fio. Explorando o uso de *proxies* da Web, o trabalho propõe o uso de módulos que adaptam o tráfego ponto-a-ponto no caminho entre cliente e servidor. É utilizado um modelo adaptativo simples para compressão, tanto para imagens (compressão com perda) quanto para texto (compressão sem perda) que considera somente a largura de banda do meio de transmissão para determinar se a compressão deve ser realizada.

Muitos trabalhos também procuram propor modificações em camadas inferiores para otimização de tempo de resposta e consumo de energia para dispositivos móveis. [7] apresenta um modelo de compressão de cabeçalhos para protocolos TCP/IP. Esse modelo inclui uma linguagem simples, independente de plataforma, para especificação de propriedades de cabeçalhos e uma aplicação para geração de código para plataformas específicas. [8] investiga a interação entre o protocolo de economia de energia do 802.11 e o desempenho de tranferências de arquivos Web utilizando-se TCP. Para superar os problemas desse modelo é proposto um protocolo que se adapta dinamicamente à atividade da rede. Os resultados indicam um ganho em economia de energia e tempo de resposta.

Esse trabalho propõe uma nova técnica para decisão de quando um arquivo deve ser comprimido antes de sua tranmissão em um ambiente sem-fio. Ao contrário dos trabalhos anteriores, que visaram construir *softwares* estáticos ou modificar as tecnologias de transmissão existente, o modelo proposto é flexível e pode ser utilizado, por exemplo, em arquiteturas de *middleware* para melhorar o desempenho do sistema.

3 Métodos de compressão

Dentre os arquivos encontrados na WEB e que podem ser considerados como de informação textual, a presença dos arquivos HTML é superior a 97% [9]. Arquivos HTML são puramente textuais, uma vez que trazem apenas tags HTML embutidas junto ao texto a ser exibido. Essas *tags* são usadas para a formatação e não utilizam código binário, o que facilita a leitura humana tendo como revés serem extremamente verbosas e pouco eficientes quanto ao espaço ocupado.

Para analisar os resultados a serem obtidos sobre os arquivos HTML, focalizamos o estudo na compressão de arquivos texto tomando-se como referência métodos hoje conhecidos como *genéricos*, devido a serem passíveis de aplicação em qualquer mídia. Porém, tais métodos foram desenvolvidos com base em estudos sobre arquivos texto e é sobre essas mídias que encontram seu melhor desempenho. Receberam destaque neste estudo os métodos de compressão via modelo estatístico (Huffman, codificação aritmética e outros [10]) e via dicionários (modelos lz77, lz78 e variações [10]).

A tabela 1 descreve as características de cada método quanto à simetria de processamento (se o tempo gasto para descompressão é o mesmo para compressão) e se a descompressão pode ser feita a partir de qualquer ponto do arquivo comprimido ou não.

Método	Características
Huffman(estático)	Simétrico/a partir do início
Huffman(adaptativo)	Simétrico/qualquer ponto
Codificação aritmética (em geral)	Simétrico/a partir do início
LZW	Simétrico/a partir do início
LZSS	Assimétrico/a partir do início

Tabela 1: Simetria e restrição quanto ao ponto de início de descompressão dos métodos de compressão.

Métodos assimétricos têm a vantagem de oferecem uma descompressão mais rápida. Isso é benéfico pois, em geral, comprime-se uma vez e descomprime-se várias. Em outras palavras, um servidor armazena uma cópia comprimida que será recuperada e descomprimida por vários clientes diferentes. Os métodos que permitem descompressão a partir de qualquer

ponto são vantajosos à medida que se faz uma transmissão do tipo *streaming*, pois erros podem ser desconsiderados e pode-se continuar o processo sem reiniciar o envio de todo o arquivo. Para envios através de pacotes de dados (*frames*) essa vantagem é minimizada sensivelmente.

Esse estudo realizou a comparação entre esses vários métodos. Os métodos considerados de melhor performance foram: PPM (variação da codificação aritmética), LZSS (variação do lz77), LZW (variação do lz78).

3.1 Análise dos métodos de compressão

Para avaliarmos qual método seria o mais adequado para comprimir arquivos a serem transmitidos em um meio sem fio, fizemos uma análise da taxa de compressão obtida por cada método sobre um conjunto de arquivos. O dois conjuntos escolhidos foram: canterbury_corpus [11] (arquivos grandes) e calgary_corpus [12] (arquivos pequenos e médios). Para se verificar a performance com os arquivos mais abundantes na Web foram utilizados pouco mais de 1500 arquivos do tipo HTML recolhidos na *web* brasileira através de um robô implementado no projeto SIAM [13].

Visando simular o ambiente real de execução onde se daria a descompressão, a bateria de testes foi realizada em um *notebook* com processador Pentium 133 e 24 Mbytes de memória principal. O gráfico 1 representa a taxa de compressão obtida para cada método para arquivos de tamanho até 30KB.

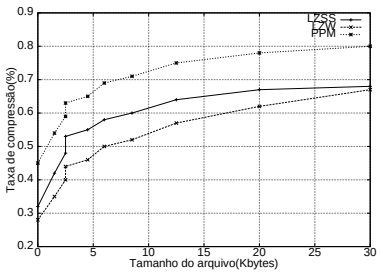


Figura 1: Taxa de compressão dos métodos PPM, LZW e LZSS para arquivos menores que 30Kbytes.

Como esperado o melhor resultado é do PPM (aritmético). Podemos notar que todos métodos vão melhorando de rendimento com o aumento do tamanho dos arquivos. Para arquivos de tamanho maior que 30KB, os métodos de melhor desempenho são: PPM, LZW, LZSS. Porém, arquivos HTML, em geral são de tamanho reduzido (mais de 90% do 1500 arquivos HTML utilizados têm até 2 KB). Com isso, entra-se numa faixa de performance anômala onde, como visto, o LZSS sai-se ligeiramente melhor que o LZW. O LZW possui uma inércia maior do que LZSS para montar um dicionário eficiente e, por isso, possui um desempenho pior para arquivos pequenos.

Os resultados obtidos mostraram melhores tempos de compressão para os métodos na seguinte ordem: LZSS, LZW, PPM. A boa taxa de compressão do método PPM é conseqüência de um processo complexo, ocasionando um tempo maior para realização da compressão.

Com importância ainda maior na comparação surge o tempo de descompressão. Uma análise detida ao gráfico de taxas de compressão mostra que a diferença obtida entre LZW e LZSS é pequena. Por outro lado o LZW tem descompressão simétrica enquanto o LZSS é assimétrico. Devido a isso, o LZSS tem uma descompressão muito mais rápida que o LZW, o

que garante um resultado geral muito melhor para o primeiro. Mais importante ainda é que esse resultado foi uniforme em todas as condições testadas, ou seja, mostrando ser desnecessária a adaptação quanto ao método de compressão em função das condições de momento do ambiente.

4 Modelo

Feito o estudo de qual método de compressão é o ideal, passamos para a construção do modelo capaz de prever quando a compressão deve ser usada. O primeiro passo foi avaliar sob quais condições se torna vantajoso usar a compressão na transmissão de arquivos em ambientes sem-fio. É fácil ver que em ao menos dois aspectos o uso de compressão é válido: quando ocorre a diminuição do tempo de transmissão ou do gasto de energia.

Vários fatores devem ser considerados na construção de um modelo que possa prever quando comprimir: a largura de banda efetiva no instante de avaliação, a percentagem de pacotes com erro do canal, o tamanho do arquivo a ser transmitido, a taxa de compressão do arquivo, o dispositivo no qual ocorrerá a descompressão e, para o cálculo de energia, a energia nominal gasta em cada operação.

Para predizer quando a compressão trará um ganho, tanto do tempo quanto da energia, criamos o modelo que pode ser visto na figura 2. Os dois módulos de avaliação de tempo e energia, T e E respectivamente, recebem como parâmetros de entrada os dados citados acima. Cada um utiliza um modelo analítico próprio e retorna um valor, V_t ou V_e respectivamente, que indica se a compressão deve ou não ser usada em cada caso. Esses valores são passados a outro módulo, C, que faz a decisão final, D_c, tentanto conciliar os dois resultados individuais. Aos módulos T e E são associados dois pesos, W_t e W_e, que representam as confiabilidades dos valores V_t e V_e.

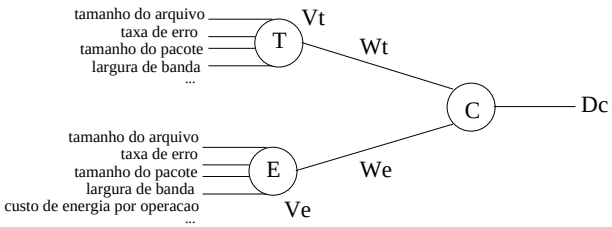


Figura 2: Esquema do preditor

Os valores retornados pelos módulos T e E são binários: 1 quando há ganho de tempo/energia e 0 caso contrário. Os graus de confiabilidade associados a cada termo variam entre 0 e 1, sendo 1 o maior grau de confiança. Portanto:

$$W_t, W_e \in [0, 1]$$

A decisão final, D_c, sobre o uso de compressão é feita utilizando-se a média simples dos valores. Assumindo que a decisão esteja feita, faz-se a transmissão usando-se compressão ($D_c \geq 0.5$) ou não ($D_c < 0.5$). Isso significa que a compressão será realizada quando ambos os parâmetros apresentarem um grau de confiança alto, ou quando um grau de confiança é suficientemente alto para contrabalancear o outro.

$$D_c := \begin{cases} 0 & \text{, se } W_t = W_e = 0 \\ \frac{W_t \cdot V_t + W_e \cdot V_e}{2.0} & \text{, caso contrário} \end{cases}$$

O próximo passo é a realimentação. O principal dado recolhido é o tempo total gasto para o processo de trasmissão e, caso seja necessário, o tempo de descompressão do arquivo. Também são recolhidos: a quantidade de bits transferidos, a quantidade de bits errados encontrados e a quantidade média de bits errados agrupados. Todos este dados são utilizados para recalcular os novos valores de largura de banda efetiva, percentagem de pacotes com erro e novos valores dos pesos W_t e W_e.

A largura de banda efetiva do canal é calculada com base na última transmissão realizada e no valor corrente da largura de banda,

$$bw = \alpha \times bw + (1 - \alpha) \times \frac{Tam_i}{tt_i}$$

onde Tam_i é a quantidade de informação (em bits) transmitida e tt_i é o tempo total gasto na última transmissão. Depois de uma série de experimentos, decidimos por utilizar $\alpha = \frac{7}{8}$ em nosso modelo.

A percentagem de pacotes com erro é calculada da seguinte maneira. Primeiramente calculamos o número de pacotes entre uma rajada de erros e a próxima através da fórmula:

$$Pac = \frac{[10^{-(Tx)} \cdot \frac{B_e}{8}]}{T_m}$$

onde T_x é um expoente que indica a taxa de bits errados transmitidos no canal durante as últimas N transmissões (valores comuns estão em torno de -5, significando 1 bit errado a cada 10^5 bits transmitidos), B_e é a quantidade média de bits com erro agrupados durante as últimas N transmissões (valores comuns variam entre 3 e 10), T_m é o tamanho do pacote (em bytes - motivo pelo qual utiliza-se o denominador 8 na fórmula)e Pac indica o número médio de pacotes entre uma rajada de erro e outra. Este último valor é utilizado na fórmula

$$P_{err} = \frac{(1 - D_i)}{Pac} + \frac{2 \cdot D_i}{Pac} = \frac{1 + D_i}{Pac}$$

onde D_i é a probabilidade de se ter 2 pacotes consecutivos com erro devido a uma mesma rajada de erros e P_{err} é a percentagem de pacotes com erro.

Os novos valores de W_t e W_e são recalculados com base nos valores reais obtido na transmissão e com os valores estimados para esta transmissão. Se realmente houve um ganho devido a predição dos modelos de tempo e de energia, esses valores são reforçados (aumentados); caso contrário, há uma penalização daquele peso que trouxe uma predição errada. Isso é descrito na seguinte fórmula.

$$w := \begin{cases} w + \frac{(1-w)}{2} & \text{, se a predição foi correta} \\ w - \frac{w}{2} & \text{, caso contrário} \end{cases}$$

A verificação da predição é feita da seguinte maneira (os subscritos nc e c indicam valores referentes a não-compressão e a compressão dos dados): a predição anterior à transmissão calcula dois valores esperados para cada fator de otimização - tempo e energia. Dessa forma temos dois tempos esperados de transmissão (um sem compressão, TE_{nc}, e outro com compressão, TE_c) e dois gastos esperados de energia (EE_{nc} e EE_c). O valor retornado para o uso de compressão é baseado na diferença entre esses dois valores. Se o ganho é positivo, retorna-se 1 ($V_t = 1$), caso contrário, 0 ($V_t = 0$). Feita a transmissão, compara-se o valor de tempo

total, TT, e gasto de energia total, ET, com os valores esperados para o caso não utilizado. Por exemplo, se foi feita a compressão, compara-se TT com TE_{nc} e ET com EE_{nc}. Se realmente o valor real for menor que o tempo estimado há um ganho, e a confiabilidade do parâmetro é reforçada. Se ocorreu o contrário - o valor real é maior que o estimado - a confiabilidade do parâmetro é penalizada, indicando que o módulo de predição não está conseguindo captar a condição correta do sistema.

Enfim, resta descrever como o cálculo dos valores esperados é feito. Primeiramente, analisaremos o tempo. O tempo total esperado para o envio de um arquivo sem compressão é determinado com sendo o tempo para se enviar um pacote e receber seus *acknowlegements* mais o tempo de se reenviar os pacotes com erro. Para o uso de compressão devemos adicionar o tempo deste processo. Portanto:

$$\begin{aligned} TE_{nc} &= Tsend_{nc} + Tresend_{nc} + Tack_{nc} \\ &= \frac{tam_{nc}}{bw} + P_{err} \cdot \frac{tam_{nc}}{bw} + (1 + P_{err}) \cdot \frac{tam_{nc}}{tam_{pkt}} \cdot \frac{tam_{ack}}{bw} \\ TE_c &= Tsend_c + Tresend_c + Tack_c + Tdesc \\ &= \frac{tam_c}{bw} + P_{err} \cdot \frac{tam_c}{bw} + (1 + P_{err}) \cdot \frac{tam_c}{tam_{pkt}} \cdot \frac{tam_{ack}}{bw} + k \cdot tam_{nc} \\ \Delta TE &= (1 + P_{err}) \cdot \frac{\Delta tam}{bw} + (1 + P_{err}) \cdot \frac{\Delta tam}{tam_{pkt}} \cdot \frac{tam_{ack}}{bw} - k \cdot tam_{nc} \\ &= (1 + P_{err}) \cdot \frac{\Delta tam}{bw} \cdot (1 + \frac{tam_{ack}}{tam_{pkt}}) - k \cdot tam_{nc} \end{aligned}$$

onde tam é o tamanho do arquivo, tam_{pkt} é o tamanho do pacote, tam_{ack} é o tamanho do pacote de *acknowledgement* e k é uma constante que depende do método de compressão, da máquina onde será feita a descompressão e do tamanho do arquivo.

A energia total gasta no processo de envio sem compressão é a energia gasta na recepção dos pacotes e envio dos *acknowlegements* por parte dos clientes. O uso de compressão implica em gasto de energia com o processo de descompressão por parte do cliente. A energia de se enviar e comprimir um arquivo é nula pois supomos que o modo transmissor está conectado a um fornecedor de energia constante (rede fixa). Dessa forma, temos:

$$\begin{aligned} EE_{nc} &= Esend_{nc} + (Ereceive_{nc} + Eack_{nc}) \\ &= 0 + [P_r \cdot (1 + P_{err}) \cdot \frac{tam_{nc}}{bw} + P_s \cdot (1 + P_{err}) \cdot \frac{tam_{ack}}{bw} \cdot \frac{tam_{nc}}{tam_{pkt}}] \\ EE_c &= Esend_c + (Ereceive_c + Eack_c + Edesc) \\ &= 0 + [P_r \cdot (1 + P_{err}) \cdot \frac{tam_c}{bw} + P_s \cdot (1 + P_{err}) \cdot \frac{tam_{ack}}{bw} \cdot \frac{tam_c}{tam_{pkt}} + P_d \cdot k \cdot \frac{tam_c}{bw}] \\ \Delta EE &= P_r \cdot (1 + P_{err}) \cdot \frac{\Delta tam}{bw} + P_s \cdot (1 + P_{err}) \cdot \frac{tam_{ack}}{bw} \cdot \frac{\Delta tam}{tam_{pkt}} - P_d \cdot k \cdot tam_{nc} \end{aligned}$$

onde $P_r, P_d e P_s$ são os consumos de potência para as operações de recepção, envio e processamento dos dados.

Como dito na seção 3, um estudo foi realizado sobre uma extensa base de arquivos de texto e dela retiramos dados analíticos que foram usados na formulação deste modelo. Esses dados são utilizados para determinar os valores da constante k, presente nas fórmulas acima.

Nas fórmulas acima utilizamos uma previsão do tempo de transmissão com o arquivo comprimido e uma previsão do tempo de descompressão do arquivo. Para descobrir qual será o tamanho aproximado de um arquivo comprimido, construímos a tabela 4 que determina a taxa média de compresão em função do método utilizado e do tamanho original do arquivo.

O tamanho comprimido de arquivo pode, então, ser calculado através da fórmula

$$Tam_c = F_c(Tam_{nc}, Met) = K_{mt} \cdot Tam_{nc}$$

na qual o parâmetro K_{mt} se refere ao método de compressão LZSS que, como visto na seção 3,

	K_{mt}			
Faixa(KB)	LZSS	LZW	GZIP	PPM
0-1	0.68	0.72	0.60	0.55
1-2	0.58	0.65	0.56	0.46
2-3	0.52	0.60	0.51	0.41
3-4	0.47	0.56	0.46	0.37
4-5	0.45	0.64	0.41	0.35
5-7	0.42	0.50	0.34	0.31
7-10	0.40	0.48	0.32	0.29
10-15	0.36	0.43	0.29	0.25
15-25	0.33	0.38	0.27	0.22
+25	0.32	0.33	0.25	0.20

Tabela 2: Taxa de compressão de um arquivo texto em função de seu tamanho original de do método de compressão.

Método	K_t
LZSS	4×10^{-6}
LZW	6×10^{-6}
PPM	2×10^{-4}
GZIP	4×10^{-6}

Tabela 3: Constante de tempo de descompressão em função do método

foi o método escolhido para se realizar a compressão.

A previsão de tempo de descompressão é determinada por três fatores: o tamanho do arquivo comprimido, o método de descompressão e a máquina cliente na qual será realizada o processo.

$$\begin{aligned} Tdesc &= F\_d(Tam_c, Maq, Met) = K \cdot Tam_c \\ Tdesc &= K \cdot K_{mt} \cdot Tam_{nc} \\ Tdesc &= K_t \cdot K_{cl} \cdot K_{mt} \cdot Tam_{nc} \end{aligned}$$

K_{cl} é uma constante relacionada à máquina onde será feita a descompressão (nesse trabalho utilizamos a constante 1 para a máquina de teste). K_t é uma constante que depende do método de descompressão. Esses valores foram obtidos empiricamente e estão presentes na tabela 4.

5 Protocolos de Comunicação

Para validar o modelo precisamos avaliar seu desempenho em diferentes condições, ou seja, para que este modelo seja realmente adaptável, ele deve apresentar uma boa performance, independente do ambiente no qual será utilizado. Neste estudo, consideramos que o fator principal que diferencia os ambientes de comunicação sem-fio é o protocolo da camada MAC. A seguir daremos uma breve descrição dos dois ambientes de transmissão utilizados neste trabalho, o IEEE 802.11 e o Bluetooth.

5.1 Tipos de protocolos MAC

Os protocolos da camada de acesso ao meio para redes sem-fio foram desenvolvidos visando cobrir os requisitos de diferentes tipos de aplicações e dispositivos. Desta forma, cada protocolo possui características próprias que levam em consideração os diversos fatores das redes de maneira diferenciada.

5.1.1 IEEE 802.11

O padrão de comunicação IEEE 802.11 foi criado em 1999 para suportar a comunicação em Redes Locais Sem Fio, *Wireless Local Networks (WLANs)*. A especificação define uma camada de acesso ao meio, camada MAC, e diversas camadas físicas, tornando possível acessar o meio de três formas possíveis: FHSS (Frequency Hopping Spread Spectrum), DSSS (Direct Sequence Spread Spectrum) e infra-vermelho.

No protocolo 802.11, a unidade de arquitetura é um BSS, *Basic Service Set*. Uma BSS é definida como um grupo de estações comunicantes sob controle de uma função de coordenação (*Distributed Coordination Function -DCF*), que é responsável por determinar quando um dispositivo pode enviar/receber dados. As estações podem se comunicar diretamente (ponto-a-ponto) ou com o suporte de uma infra-estrutura. Redes que se comunicam com a primeira forma são conhecidas como redes *ad-hoc*; com a segunda forma, redes *infra-estruturadas*. Essas últimas utilizam estações-base para interconectar os dispositivos em prover suporte à mobilidade sobre diferentes áreas.

Existem duas taxas de comunicação, 1Mbps e 2Mbps. Os padrões 802.11a e 802.11b alteraram a especificação para prover taxas de 5.5 e 11Mbps (802.11b), chegando até 54Mbps (802.11a). O 802.11a utiliza um esquema especial de multiplexação para atingir altas taxas de comunicação, o que torna impossível a comunicação entre dispositivos 802.11a e 802.11b.

5.2 Bluetooth

O Bluetooth tem como principal objetivo prover um mecanismo que permite a interligação dos dispositivos móveis de baixa potência através de enlances de rádio. O principal objetivo dessa arquitetura é permitir que dispositivos como PDAs, *pagers*, *modems*, telefones celulares e computadores portáteis sejam interligados sem a necessidade de cabos, de forma rápida e fácil, permitindo a troca de voz e dados.

Bluetooth opera na faixa de frequência de 2.4 Ghz, conhecida como ISM, e foi especificado com ênfase em robustez e baixo custo. Sua implementação permite o surgimento de pequenas redes de acesso pessoal, conhecidas como *WPANs*, (*Wireless Personal Area Network*), formada por dispositivos de baixa capacidade computacional. O alcance de transmissão varia de 10 centimetros a 10 metros, mas a especificação prevê casos de 100 metros, aumentando-se a potência de transmissão.

Cada dispositivo Bluetooth pode ser mestre ou escravo a qualquer tempo, mas não de maneira simultânea. Os mestres são responsáveis pelo gerenciamento da conexão, iniciando a transferência de dados, enquanto o escravo segue essa coordenação do canal. Uma dos maiores desafios em redes Bluetooth é a configuração e reconfiguração das redes, ou seja, o que acontece quando um nodo entra ou sai das áreas de comunicação.

6 Experimentos

Nessa seção descrevemos os resultados de nossos experimentos.

6.1 802.11

Para realizarmos os experimentos reais construímos o seguinte cenário: um sistema embutido comunicando-se através de um canal *wireless* com um servidor de arquivos. A recuperação

de arquivos é feita através do protocolo HTTP. Esse cenário assemelha-se a um usuário navegando na *Intranet* de um *shopping*, comunicando-se com o servidor do estabelecimento com um dispositivo móvel de pequeno porte. Utilizamos como sistema embutido o modelo DIMM-PC/486-I da Jumptec [15], que possui o processador 486 com memória de 16MB e velocidades de 33 e 66Mhz. As placas de comunicação utilizadas foram WaveLan 802.11 da Lucent [16]. Os gastos de energia com recepção, transmissão e compressão foram retirados da especificação de cada componente do sistema e estão presentes na tabela 4.

Operação	Corrente (mA)	Voltagem (V)	Potência (W)
Transmissão	330	5	1.65
Recepção	280	5	1.40
Descompressão	510	5	2.55 (66Mhz)
	267	5	1.33 (33Mhz)

Tabela 4: Gastos de energia

Para obtermos os valores das constantes presentes no modelo, executamos novamente os testes de compressão sobre o sistema embutido. Para isso montamos outra amostra de 1500 documentos retirados da *web* brasileira e a submetemos aos testes de compressão. As constantes de razão de compressão permaneceram inalteradas, já que utilizamos o mesmo método de compressão (LZSS) das experiências anteriores. O tempo de descompressão apresenta uma variação linear com o tamanho do arquivo, como já era esperando. O gráfico da figura 3 representa esse comportamento para os arquivos de tamanho inferiores a 15KB. A nova constante K_t é de $1,875 \times 10^{-5}$, obtida por regressão.

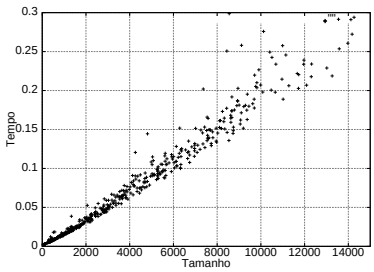
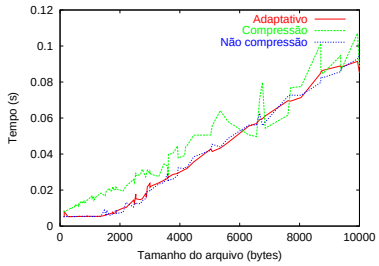


Figura 3: Relação entre tempo de descompressão e tamanho do arquivo.

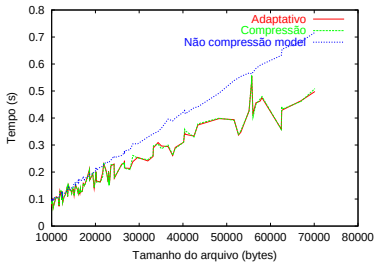
A experimentação foi feita da seguinte maneira: adaptamos o protocolo HTTP com novos campos de cabeçalho, de forma que os dados relativos à última transferência de arquivo possam ser transmitidos em *piggyback* com uma nova requisição. Esses dados são: tempo total gasto na recepção dos dados, energia total gasta na recepção dos dados, valor da constante de descompressão da máquina (K_t) e valores de potência para cada operação. Assim, o servidor fica encarregado de manter uma sessão, que contém os dados referentes ao modelo. Essa é uma boa opção para arquiteturas com *middleware* ou para os servidores que já mantém uma sessão para cada cliente conectado, pois acarretaria apenas na inserção de novos dados àqueles já existentes. O cliente fica responsável apenas por armazenar os dados referentes à última transferência com aquele servidor, o que pode ser obtido com uma pequena *cache*. As alterações do HTTP também podem ser obtidas facilmente, já que a versão 1.1 desse protocolo permite extensões em seu cabeçalho [17].

As figuras 4(a) e 4(b) representam o tempo de resposta em relação ao tamanho do arquivo para largura de banda de 1Mb e para todos os modelos. Podemos notar que o modelo

proposto adapta-se ao melhor caso em todas as situações. Na figura 4(a), a não compressão é a melhor situação para arquivos menores que 10 Kbytes; a partir desse limite (figura 4(a)), comprimir torna-se a solução mais vantajosa. De acordo com [18], a maior parte dos arquivos acessados na Internet por clientes móveis são menores 3 Kbytes e para clientes *off-line*, 6Kbytes, e, portanto, a compressão indiscriminada não seria uma boa política nesse caso.



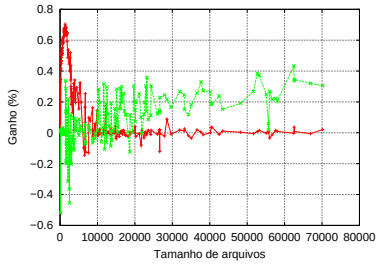
(a) Tamanhos menores que 10Kbytes.



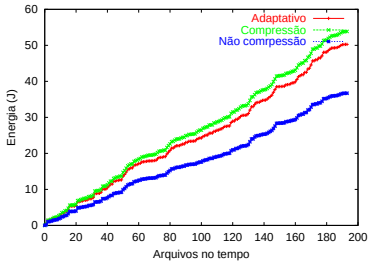
(b) Tamanhos maiores que 10Kbytes.

Figura 4: Tempo de resposta para 6 adaptativo, não comprimido e comprimido

A figura 5(a) representa o ganho percentual do modelo adaptativo sobre os outros, para todos os arquivos testados. Comparando-se ao modelo de compressão, podemos notar ganhos entre 60% e 70% para tamanhos entre 1 e 2Kbytes, reduzindo-se à medida que nos afastamos desses limites. Comparando-se ao modelo de não-compressão, há um aumento contínuo do ganho percentual a partir de 10Kbytes. Para tamanhos médios de 50Kbytes, os ganhos são entre 20% e 30% no tempo total de resposta. A instabilidade dessa última comparação para arquivos pequenos deve-se à grande susceptibilidade à taxa de erro e atrasos nos dispositivos para arquivos de tamanho pequeno.



(a) Ganho percentual.

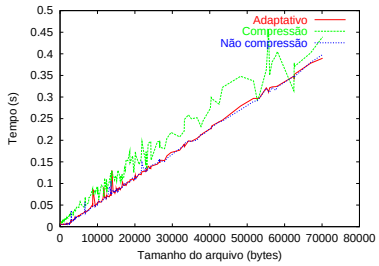


(b) Energia gasta acumulada.

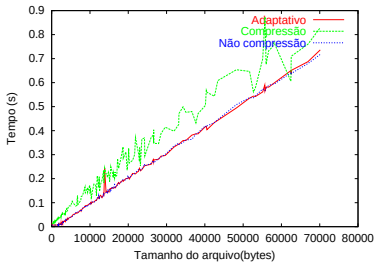
Figura 5: Modelos adaptativo, compressão e não compressão.

O gráfico da figura 5(b) representa a energia gasta acumulada em cada modelo. O modelo implementado dá pesos iguais a cada parâmetro - tempo e energia - e a compressão pode ser realizada mesmo quando o ganho seja em apenas um deles. Para essa arquitetura, mesmo a compressão sendo mais cara que a transmissão, há um ganho no tempo quando a largura de banda é pequena. Isso explica o gasto de energia do modelo adaptativo do gráfico.

Os próximos gráficos demonstram como a decisão de compressão depende tanto do dispositivo móvel, quanto da largura de banda e o tamanho de arquivo.



(a) Largura de banda 2Mbps; Freqüência do processador 66MHz.

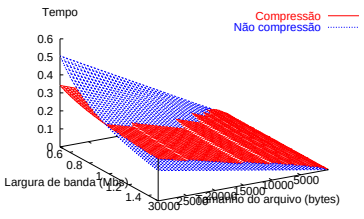


(b) Largura de banda 1Mbps; Freqüência do processador 33MHz.

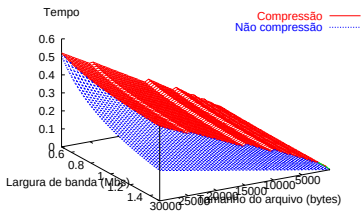
Figura 6: Tempo de resposta para os modelos adaptativo, não compressão e compressão.

Os dados da figura 6(a) representam a performance dos modelos quando alteramos a largura de banda do canal para 2Mbps. Pode-se notar que os ganhos da compressão desaparecem por completo, uma vez que o atraso de descompressão não é compensado pela economia no tempo de transmissão. O mesmo acontece quando a freqüêcia do processador do dispositivo é modificadada para 33MHz e a largura de banda do canal é mantida a 1Mbps (figura 6(b)) . É interessante notar que, para este último experimento, necessitou-se apenas de duas alterações no modelo adaptativo: a constance K_{cl} foi modificada 2, já que diminuímos pela metade a freqüência do processador, e as constantes de potências consumidas assumiram os valores adequados (tabela 4), o que demonstra a flexibilidade do modelo.

As fórmulas do modelo analítico (seção 4) podem ser utilizadas para se construir gráficos do tempo de resposta em função do tamanho do arquivo, largura de banda e freqüência do processador. Utilizando os dados do experimento acima e fazendo algumas suposições (como taxa de erro igual a 0), obtém-se os gráficos das figuras 7(a) e 7(b). Esses gráficos nos dão uma visão analítica do modelo, que está representado na figura 8.



(a) Largura de banda 1Mbps; Freqüência do processador 66MHz.



(b) Largura de banda 1Mbps; Freqüência do processador 33MHz.

Figura 7: Tempo de resposta para os modelos de não compressão e compressão, utilizando o modelo analítico.

Na figura 8, a área descrita pela função define onde a compressão deve ser utilizada. A forma da área é função, principalmente, do método de compressão e do tipo de mídia estudado -

definido pelo parâmetro K_{mt}. O deslocamento da área é função, principalmente, do dispositivo - o produto $K_t \times K_{cl}$. A taxa de erro de pacotes, os tamanhos dos pacotes podem alterar, em menor escala, ambas características da função. O gráfico para consumo de energia pode ser obtido de maneira semelhante.

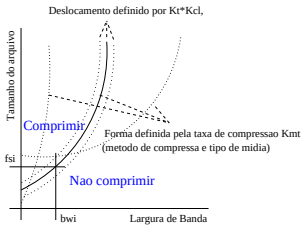


Figura 8: Representação gráfica do modelo analítico.

Esta representação analítica do modelo pode ser utilizada para se obter uma versão simplificada do modelo, que executa no próprio dispositivo. Para isso, basta estipular alguns valores de largura de banda, bw_i, e, para cada um, obter o valor correspondente de tamanho de arquivo, fs_i, a partir do qual a compressão passa a ser válida. Esses valores podem ser utilizados para se 0 uma tabela, armazenada no próprio cliente, e que é consultada a cada requisição de arquivo. O servidor, nesse caso, fica responsável por analisar o limite de arquivo transmitido, e realizar a compressão quando possível. É claro que esta alteração implica em mais gasto de energia pelo cliente e diminui a precisão do modelo.

6.2 Bluetooth

Neste experimento construímos um cenário semelhante ao anterior, exceto pela configuração das máquinas comunicantes. O cliente opera em uma processador PentiumII, 400Mhz, com 256Mbytes de memória, executando Suse Linux 8.1 (kernel 2.4.18) e utiliza um placa Ericsson [19] para comunicação. O servidor executa em uma máquina semelhante, porém utiliza um placa Widcomm para o suporte a Bluetooth. Utilizamos metade do lote de arquivos da seção anterior e tivemos que 1 o parâmetro K_t para a máquina cliente.

O protocolo Bluetooth garante uma interferência mínima no canal de comunicação através de seu modelo de salto de frequência, o que, por sua vez, limita a largura de banda máxima em 723Kbps. Esse limite da largura de banda provê um cenário ideal para a utilização da compressão prévia dos arquivos, uma vez que a largura de banda é basta restrita. O gráfico da figura 9 ilustra a discrepância entre os tempos de resposta para o modelo de não compressão e o modelo adaptativo que, nesse cenário, optou por comprimir todo o lote de arquivos.

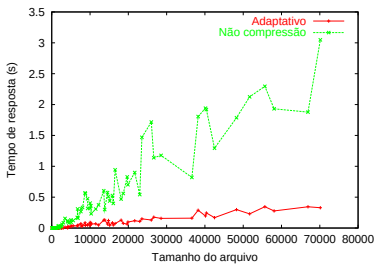


Figura 9: Tempo de resposta para o protocolo Bluetooth.

7 Simulações

Nesta seção apresentamos alguns resultados analisados através de simulação. Para verificar a validade do modelo proposto utilizamos o simulador NS [20]. Nas diversas simulações realizadas, variamos os seguintes parâmetros: a tecnologia de comunicação, o número de usuários, a largura de banda do meio e o modo de transmissão. As tecnologias de comunicação estudadas foram o Bluetooth e o 802.11; os modos de transmissão foram divididos em três modelos: transmissão sem compressão, transmissão com compressão e transmissão com o modelo adaptativo.

O módulo de compressão e descompressão foi implementado a partir dos resultados experimentais obtidos na primeira fase deste trabalho. Criamos um protocolo de transmissão com janela de tamanho 1, no qual deve-se esperar uma confirmação de transmissão a cada pacote enviado. Implementamos, portanto, um agente, denominado *Compression*, presente tanto no servidor quanto no cliente, capaz de realizar todo o controle de fluxo e garantia de entrega, além de prever quando comprimir os arquivos antes de suas respectivas transmissões.

Primeiramente analisamos o desempenho do modelo no ambiente 802.11. A rede simulada é composta de um único servidor S e vários clientes C_i que fazem requisições de arquivos a S, como pode ser visto na figura 7. Simulamos cenários com 1, 5, 10, 50 e 100 clientes, em ambientes com taxa de erro equivalente a 0,5%, onde os nodos não apresentam mobilidade. Os tamanhos de arquivos foram gerados através de uma distribuição exponencial de média igual a 3Kbytes. O intervalo entre o fim de uma requisição e a próxima segue uma distribuição exponencial com média de 30s. O tempo de simulação foi de 3600s e cada simulação foi realizada 33 vezes.

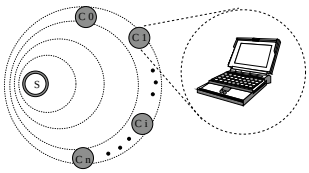
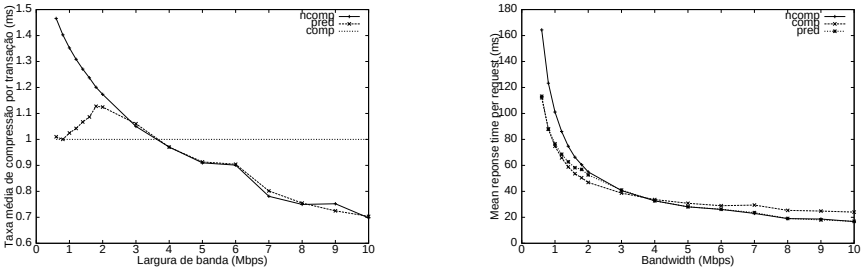


Figura 10: Esquema da rede simulada.

Os gráficos das figuras 11(a) e 11(b) demonstram o desempenho do modelo adaptativo sobre os outros modelos, em forma de 0 e tempo, para o cenário de 1 único cliente e largura de banda variável. O modelo adaptativo apresenta boa performance para larguras de banda muito pequenas e muito altas (não compressão foi definido como 1 no gráfico 13(b)).

O gráfico da figura 7 ilustra o modelo adaptativo com uma pequena alteração: modificamos a fórmula de decisão para que o energia restante no dispositivo fosse analisada. A energia restante no dispositivo foi classificada em 5 classes (80-100%, 60-80%, 40-60%, 20-40%, 0-20%) e para cada uma definiu-se um peso diferente e crescente, 0 proporcionalmente o peso do parâmetro de tempo. O gráfico 7 demonstra a performance decrescente do sistema para um cenário de 50 nodos, no qual cada um inicia com energia igual a 100%, uma vez que à medida que a energia se esvai, a compressão deixa de existir.

Em seguida, foi simulado o ambiente Bluetooth. O cenário é semelhante ao anterior, no qual o mestre é um servidor de arquivo e os 0 são clientes da aplicação. Sob o ambiente Bluetooth a compressão se faz válida para qualquer número de clientes, uma vez que a largura de banda, que já é pequena, se torna mais escassa ainda. Os gráficos das figuras 13(a) e 13(b) ilustram a performance do modelo adaptativo sobre os outros modelos (não compressão foi definido como 1 no gráfico 13(b)).



(a) 0. (b) Tempo.

Figura 11: Comparação dos modelos adaptativo, compressão e não-compressão para cenário de 1 cliente.

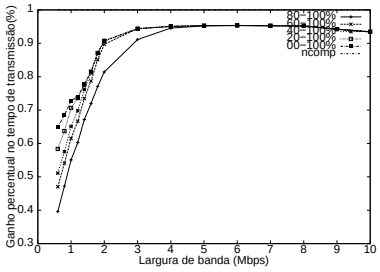
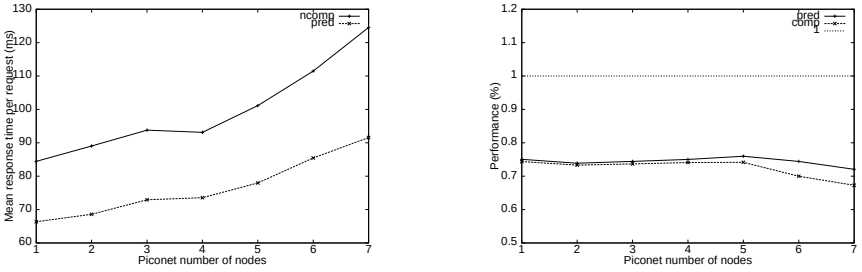


Figura 12: Performance do modelo adaptativo com peso variável para o parâmetro de energia.



(a) Tempo de resposta. (b) 0

Figura 13: Comparação de modelos adaptativo, não-compressão e compressão para o ambiente Bluetooth.

8 Conclusão

Neste trabalho apresentamos uma metodologia para adaptação, através de compressão, de arquivos HTML para dispositivos móveis. Ao contrário dos trabalhos existentes na literatura, nosso trabalho propõe uma técnica que prevê, dinamicamente, quando um arquivo deve ser comprimido. As informações avaliadas - tamanho do arquivo, algoritmo de compressão, largura de banda, consumo de energia e capacidade do próprio dispositivo móvel - garantem uma melhor precisão sobre quando a compressão deve ser aplicada. Os teste experimentais e simulados com os ambientes IEEE 802.11 e Bluetooth comprovam a eficiência deste método quando comparados a técnicas mais simples de adaptação.

Referências

[1] George H. Forman and John Zahorjan. The Challenges of Mobile Computing. *IEEE Computer*, 27(4):38–47, Abril 1994.

[2] R. H. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications*, 1:6–17, 1994.

[3] Malcolm McIlhagga, Ann Light, and Ian Wakeman. Towards a Design Methodology for Adaptative Applications. In *Fourth Annual International Conference on Mobile Computing and Networking MOBICOM'98*, Dallas, Texas, USA, 1998.

[4] Jeffrey C. Mogul, Fred Douglis, Anja Feldmann, and Balachander Krishnamurthy. Potential Benefits of Delta Encoding and Data Compression for Http. In *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication ACM SIGCOMM '97*, Cannes, France, 1997. ACM Press.

[5] Barron C. Housel, George Samaras, and David B. Lindquist. WebExpress: a Client/Intercept Based System for Optimizing Web Browsing in a Wireless Environment. *Mobile Networks and Applications*, 3(4):419–431, 1999.

[6] Jesse Steinberg and Joseph Pasquale. A Web Middleware Architecture for Dynamic Customization of Content for Wireless. In *WWW2002*, Honolulu, Hawaii, USA, 2002.

[7] Jeremy Lilley, Jason Yang, Hari Balakrishnam, and Srinivasan Seshan. A Unified Header Compression Framework for Low-Bandwidth Links. In *Sixth Annual International Conference on Mobile Computing and Networking*, Boston, Massachusetts, USA, 2000. ACM Press.

[8] Ronny Kranshinsky and Hari Balakrishnam. Minimizing Energy for Wireless Web Access with Bounded Slowdown. In *Eighth Annual International Conference on Mobile Computing and Networking MOBICOM'02*, Atlanta, Georgia, USA, Setembro 23-28 2002. ACM Press.

[9] E. Alonso, E. S. de Moura, P. Golgher, A. Silva, R. Barra, A. Laender, B. Ribeiro-Neto, and N. Ziviani. Um Retrato da Web Brasileira. In *SEMISH'2000*, Curitiba, Paraná, Brazil, July 2000.

[10] Mark Nelson and Jean loup Gailly. *The Data Compression Book*. M&T Books, 2nd edition, 1995.

[11] Tim Bell. Canterbury Corpus. http://corpus.canterbury.ac.nz/.

[12] Ian Witten and Tim Bell. Calgary Corpus. http://links.uwaterloo.ca/calgary.corpus.html.

[13] Sistemas de Informação em Ambientes Móveis. http://www.dcc.ufmg.br/siam/.

[14] Ieee 802.11. http://grouper.ieee.org/groups/802/11/.

[15] Jumptec. http://www.jumptec.de/.

[16] Lucent technologies. http://www.lucent.com/.

[17] Jeffrey C. Mogul. Clarifying the Fundamentals of Http. In *WWW2002*, Honolulu, Hawaii, USA, 2002.

[18] Atul Adya, Paramvir Bahl, and Lili Qiu. Analyzing the Browse Patterns of Mobile Clients. In *Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement Workshop*, San Francisco, California, USA, 2001.

[19] Ericsson. Bluetooth development kit. http://www.ericsson.com/bluetooth/.

[20] Network Simulator. http://www.isi.edu/nsnam/ns.