



AVALIAÇÃO DE CAPACIDADE E CONSUMO DE ENERGIA DE REDE
MÓVEL AD HOC CENTRADA EM INTERESSE

Rafael Cruz Salles

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Claudio Luis de Amorim

Rio de Janeiro
Setembro de 2014

AVALIAÇÃO DE CAPACIDADE E CONSUMO DE ENERGIA DE REDE
MÓVEL AD HOC CENTRADA EM INTERESSE

Rafael Cruz Salles

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Claudio Luis de Amorim, Ph.D.

Prof. Felipe Maia Galvão França, Ph.D.

Prof. Noemi de La Rocque Rodriguez, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2014

Salles, Rafael Cruz

Avaliação de capacidade e consumo de energia de rede móvel ad hoc centrada em interesse/Rafael Cruz Salles. – Rio de Janeiro: UFRJ/COPPE, 2014.

IX, 66 p.: il.; 29,7cm.

Orientador: Claudio Luis de Amorim

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2014.

Referências Bibliográficas: p. 55 – 61.

1. Protocolo de comunicação.
 2. REPA.
 3. Endereçamento por interesses.
 4. Raspberry Pi.
 5. MANET.
- I. Amorim, Claudio Luis de.
II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

AVALIAÇÃO DE CAPACIDADE E CONSUMO DE ENERGIA DE REDE
MÓVEL AD HOC CENTRADA EM INTERESSE

Rafael Cruz Salles

Setembro/2014

Orientador: Claudio Luis de Amorim

Programa: Engenharia de Sistemas e Computação

Parte do consumo energético associado a equipamentos móveis deve-se ao processamento dos programas em execução. Visto que grande parte dos sistemas móveis opera sob o uso de baterias, torna-se importante uma avaliação do consumo energético do software utilizado ou em desenvolvimento, de forma a aumentarmos o tempo necessário entre recargas.

Avaliamos nesta dissertação o consumo energético e as capacidades de transmissão e recepção de um protocolo orientado a interesses (REPI) que possui aplicações nas áreas de Redes Móveis Ad Hoc (do inglês MANETS) e de sensoriamento. A fim de compararmos a implementação do protocolo REPI com um protocolo de comunicação amplamente utilizado, realizamos também a avaliação da pilha de protocolos TCP/IP. Para realização de tais avaliações utilizamos um ambiente de código aberto e de baixo custo (Raspberry PI).

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CAPACITY AND ENERGY CONSUMPTION EVALUATION OF AN
INTEREST CENTRIC MOBILE AD HOC NETWORK

Rafael Cruz Salles

September/2014

Advisor: Claudio Luis de Amorim

Department: Systems Engineering and Computer Science

Part of the energy consumption of mobile devices is related to the processing of the running programs. As most of mobile devices run on batteries it's important to evaluate energy consumption of the program in development or being executed so that we can increase the time needed between recharges.

In this dissertation we evaluate the energy consumption and the transmission and reception capacities of an interest centric protocol (REPI) with applications on sensor and Mobile Ad hoc Networks (MANETS). In order to compare the REPI protocol with a widely used one we also evaluated the TCP/IP. The evaluation was done on an open source and low cost testbed (Raspberry PI).

Sumário

Lista de Figuras	viii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivo	2
1.3 Contribuições	2
1.4 Organização	3
2 Conceitos básicos	4
2.1 Eficiência energética em sistemas de tecnologia da informação	4
2.2 Protocolo TCP/IP e suas limitações	7
2.3 Rede endereçada por Interesse	9
2.3.1 Exemplos de arquiteturas	10
2.4 Protocolo REPI	12
2.5 Discussão	18
3 Avaliação Experimental	20
3.1 Objetivo	20
3.2 Trabalhos relacionados	20
3.3 Ambiente Experimental	21
3.3.1 Bancada	21
3.3.2 Topologia	22
3.3.3 Sistema Operacional	23
3.3.4 Software coletor de dados	23
3.3.5 Codificação	23
3.4 Experimentos	24
3.4.1 TCP/IP – Experimento 1 - Influência do número de repetições nas taxas de transmissão e recepção	25
3.4.2 TCP/IP – Experimento 2 - Consumo energético e capacidade de transmissão	28

3.4.3	TCP/IP – Experimento 3 - Consumo energético e capacidade de recepção	32
3.4.4	REPD – Experimento 1 - Influência do número de envios nas taxas de transmissão e recepção	36
3.4.5	REPD – Experimento 2 - Consumo energético e capacidade de transmissão	40
3.4.6	REPD – Experimento 3 - Consumo energético capacidade de recepção	44
3.5	Discussão	46
4	Conclusão	53
4.1	Trabalhos futuros	54
	Referências Bibliográficas	55
A	Plataforma Raspberry PI	62
A.1	História	62
A.2	Plataformas computacionais de código livre	63
A.3	Utilização da plataforma Raspberry PI pela comunidade científica . .	64

Lista de Figuras

1.1	Usuários próximos interessados em um mesmo conteúdo musical. Atualizado diariamente.	2
2.1	Utilização energética. Extraído de [1]	7
2.2	Consumo energético do transmissor TCP/IP. Extraído de [1]	7
2.3	Vazão do transmissor TCP/IP. Extraído de [1]	8
2.4	Cabeçalho REPI proposto e extraído de [2]	12
2.5	Equivalência REPI e modelo OSI. Proposto e extraído de [3]	13
2.6	REPI: PA composto de dois campos de características	15
2.7	REPI: Nó A envia mensagem com seu PA: [1;5;Futebol]	15
2.8	REPI: Nó B encaminha a mensagem por haver casamento	15
2.9	REPI: Nó C encaminha a mensagem por haver casamento	16
2.10	REPI: Nó D aceita a mensagem, mas não a encaminha	16
2.11	Serviço REPAD. Proposto e extraído de [3]	17
3.1	Principais elementos da bancada de testes	22
3.2	TCP/IP. Exemplos de taxas médias obtidas para os diversos tamanhos de pacotes (S) (1a coluna)	26
3.3	TCP/IP. Influência do número de repetições na diferença entre as taxas médias de recepção e transmissão	27
3.4	TCP/IP. Taxas médias de recepção para 3, 48 e 96 mil repetições	27
3.5	TCP/IP. Com suficientes repetições, taxas médias de transmissão e recepção possuem diferença inferior a 1%	29
3.6	TCP/IP. Exemplos de medições obtidas no aplicativo coletor	29
3.7	TCP/IP. Consumo por mensagem transmitida	30
3.8	TCP/IP. Transmissão de 1400 bytes - Tamanho do pacote vs consumo	30
3.9	TCP/IP. Fator de multiplicação – Consumo energético necessário ao envio de 1400 bytes	31
3.10	TCP/IP. Vazão do transmissor - KBytes por segundo	31
3.11	TCP/IP. Capacidade de transmissão - Mensagens por segundo	32
3.12	TCP/IP. Consumo energético por mensagem recebida	33

3.13	TCP/IP. Recepção de 1400 bytes - Tamanho do pacote vs consumo energético	34
3.14	TCP/IP. Fator de multiplicação – Consumo energético necessário à recepção de 1400 bytes	34
3.15	TCP/IP. Vazão de recepção - KBytes por segundo	35
3.16	TCP/IP. Capacidade de recepção - Mensagens por segundo	35
3.17	REPD. Taxa média de todas as amostras de taxa de recepção	36
3.18	REPD. Variação entre as amostras da taxa de recepção	37
3.19	REPD. Influência do percentual recebido na taxa de recepção	38
3.20	REPD. Variação média das taxas, em relação à mínima, para diversos percentuais de recepção	38
3.21	REPD. Taxas de transmissão e recepção confiáveis quando recebimento superior a 98%	39
3.22	REPD. Taxas de recepção quando recebimento superior a 99%	39
3.23	REPD. Consumo energético por mensagem transmitida	41
3.24	REPD. Consumo para envio de 1400 bytes	42
3.25	REPD. Fator de multiplicação – Consumo energético necessário ao envio de 1400 bytes	42
3.26	REPD. Vazão de transmissão - KBytes por segundo	43
3.27	REPD. Capacidade de transmissão - Mensagens por segundo	43
3.28	REPD. Consumo por mensagem recebida	45
3.29	REPD. Consumo energético para recepção de 1400 bytes	46
3.30	REPD. Fator de multiplicação – Consumo energético necessário à recepção de 1400 bytes	46
3.31	REPD. Capacidade de recepção - KBytes por segundo	47
3.32	REPD. Capacidade de recepção - Mensagens por segundo	47
3.33	Impacto resultante da análise de pacotes através de filtros. Ambiente com um único processador. Extraído de [4]	49
A.1	Conexões Raspberry Pi Modelo B - Extraído de [5]	64
A.2	Visão geral da futura rede aeronáutica da região Ásia-Pacífico - Extraído de [6]	65
A.3	PiCloud: Infraestrutura para computação na nuvem. Esquerda: seus 56 nós. Direita: ferramenta de gerenciamento - Extraído de [7]	66

Capítulo 1

Introdução

1.1 Motivação

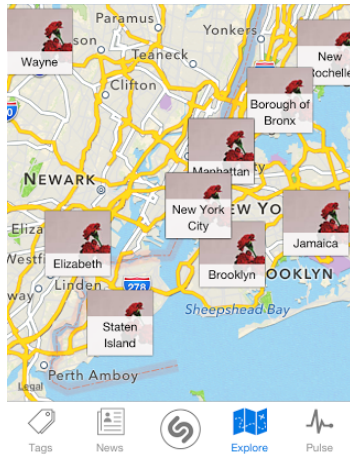
O processo de desenvolvimento de sistemas de sensoriamento e de sistemas móveis envolve a seleção do hardware adequado e o desenvolvimento de software. Um ponto importante no desenvolvimento de tais sistemas é o consumo energético. Microsensores precisam minimizar o consumo energético de forma a operarem por longos períodos sem necessidade de troca de sua fonte de energia. Dispositivos móveis portáteis como o celular necessitam cada vez mais de baterias menores e com maior tempo entre recargas.

Durante o desenvolvimento de software para esses sistemas é comum a ideia de que o consumo energético é algo a ser avaliado unicamente no momento da seleção do hardware a ser utilizado, no entanto, a avaliação de consumo energético do código desenvolvido pode contribuir, de forma significativa, para sua redução.

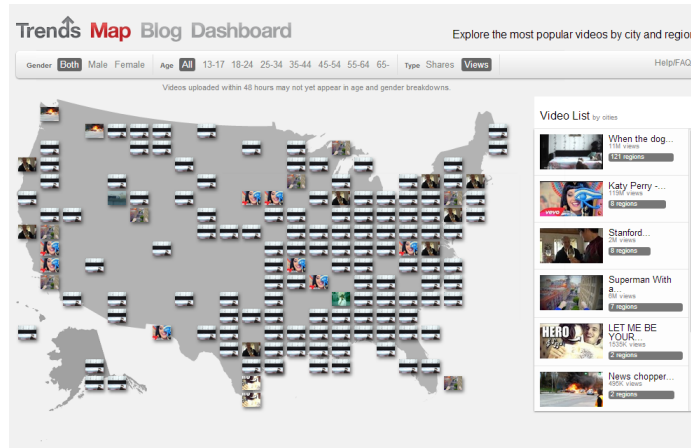
A popularização dos equipamentos móveis, em especial, o celular, proporciona um grande ambiente para a formação de rede ad hoc, com constantes entradas e saídas de nós. Uma dificuldade é que em uma rede como essa prevalece o anonimato, tornando difícil o endereçamento, assim como o estabelecimento de comunicação em um modelo cliente-servidor. Nesse cenário verificamos que uma arquitetura par-a-par mostra-se bastante adequada.

Em uma arquitetura par-a-par (P2P) temos aplicações colaborativas, distribuindo e consumindo conteúdo de maneira que a informação passa a ser difundida de forma descentralizada, mas que requer endereçamento centralizado. Uma exigência atual de alguns programas conhecidos que se utilizam de tal arquitetura (ex.: Bit Torrent e Kazaa) é a necessidade de um servidor previamente conhecido que atualize e distribua, aos nós recém-chegados, a lista de nós detentores do conteúdo desejado.

Outra característica importante, que diz respeito ao conteúdo acessado nessa atual rede móvel formada por celulares, é a rápida difusão (e o conseqüente acesso)



(a) SHAZAM [8]



(b) YOUTUBE [9]

Figura 1.1: Usuários próximos interessados em um mesmo conteúdo musical. Atualizado diariamente.

de um mesmo conteúdo, em curto intervalo de tempo, em uma mesma região. Seja através de uma propaganda divulgada por um avião na praia ou por um e-mail recém chegado, muitas vezes vemos um interesse comum, regionalizado, que leva pessoas fisicamente próximas a acessarem um mesmo conteúdo, em tempo relativamente próximo. Embora dados instantâneos a respeito de tais acessos não sejam amplamente divulgados por grandes empresas da Internet, alguns serviços nos mostram claramente que essa grande rede, formada por desconhecidos, poderia se beneficiar, de maneira descentralizada, do compartilhamento através de uma rede P2P e da difusão móvel que seus integrantes propiciam.

1.2 Objetivo

Dado o exposto, este trabalho objetiva avaliar experimentalmente o consumo energético e a capacidade do protocolo orientado a interesses REPI. Para isso utilizamos a atual implementação através do serviço REPD, desenvolvido no Laboratório de Computação Paralela e Sistemas Móveis da Universidade Federal do Rio de Janeiro (COMPASSO - COPPE/UFRJ). Tal serviço propõe uma alternativa para o desenvolvimento de aplicações P2P em redes ad hoc como as exemplificadas acima.

1.3 Contribuições

As principais contribuições deste trabalho são:

- Avaliação de capacidade e consumo energético do protocolo REPI, através de sua implementação como serviço (REPD).

- Avaliação de capacidade e consumo energético do protocolo TCP/IP.

As avaliações se deram em sistema operacional Linux, utilizando placas Raspberry PI.

1.4 Organização

O capítulo 2 apresenta conceitos e trabalhos relacionados à eficiência energética em sistemas de informação, ao protocolo TCP/IP, a redes endereçadas por interesse e ao protocolo orientado a interesses que analisamos (REPI). Tais conceitos serão de utilidade em nossos experimentos.

O capítulo 3 apresenta avaliações do serviço REPD (uma implementação do protocolo REPI) e da pilha TCP/IP na plataforma Raspberry PI. Aqui descrevemos o ambiente experimental, as avaliações realizadas, seus objetivos e resultados.

Concluimos o trabalho no capítulo 4, através de uma ligação entre as discussões realizadas e os resultados encontrados ao longo do trabalho.

No Apêndice A, apresentamos e justificamos a escolha do ambiente Raspberry PI, através de um breve resumo sobre sua história e alguns trabalhos de interesse.

Capítulo 2

Conceitos básicos

Neste capítulo encontram-se alguns dos fundamentos necessários ao melhor entendimento deste estudo. O objetivo é apresentarmos trabalhos e conceitos que serão utilizados nos demais capítulos.

Inicialmente, abordamos questões referentes à eficiência energética no contexto de sistemas de informação.

Posteriormente, descrevemos limitações do protocolo TCP/IP quando utilizado em redes sem fio. Também destacamos a dificuldade de sua substituição, visto seu amplo uso.

Em seguida, descrevemos o conceito de redes endereçadas por interesse, um modelo de comunicação na Internet. Para tal modelo, classificamos e descrevemos os principais grupos de problemas a serem abordados. Aqui também apresentamos alguns exemplos de arquiteturas orientadas a interesse e relatamos de que forma propõem solução aos grupos de problemas levantados.

Na sequência, apresentamos a evolução do protocolo REPI desde sua primeira proposta descrita em [2]. Descrevemos os trabalhos relacionados que colaboram ou colaboraram para seu desenvolvimento. Ao longo de tais trabalhos diversos acrônimos surgiram, então, procuramos organizá-los de forma a facilitar eventual desambiguação que se faça necessária.

Por fim, realizamos uma síntese do capítulo, discutindo os pontos principais apresentados.

2.1 Eficiência energética em sistemas de tecnologia da informação

Em 2008, observamos que um dos grandes desafios apresentados pela Academia Nacional de Engenharia Americana diz respeito à redução do custo energético e da emissão de CO₂ [10].

No passado, o desenvolvimento de novas tecnologias de comunicação aplicadas a sistemas da informação (do inglês ICT - Information and Communication Technologies) tinha como foco principal a redução de custos e o aumento de performances. Como relatado em [11], as emissões de CO₂ referentes a tais sistemas têm aumentado a taxa de 6% ao ano, com previsões de que representem 12% dos totais de emissões no ano de 2020. Em referência ao uso de dispositivos móveis, previsões apresentadas em estudo [12] realizado pela Rice University, Houston, TX, USA, indicam que, em 2020, a emissão mundial de CO₂ proveniente de tais dispositivos deve quadruplicar em relação a 2009, atingindo 6,58 mega toneladas de CO₂. Tal estimativa, de maneira conservadora, considerou que cada celular é recarregado uma única vez a cada dois dias.

Ao contrário do que ocorre na indústria energética, onde um dos subprodutos acaba sendo o CO₂, em ICT verificamos que tal emissão se dá indiretamente, através do consumo elétrico necessário aos equipamentos utilizados e também ao resfriamento dos mesmos.

Pelas considerações apresentadas é de interesse a pesquisa por soluções que procurem diminuir o consumo energético.

No contexto de redes conectadas fisicamente por fios ou fibra ótica, verificamos que os recentes protocolos passaram a incorporar eficiência energética ao seu desenvolvimento, como descrito na especificação do protocolo 802.3az [13].

Aplicado à comunicação sem fio em [14] é realizado estudo do consumo energético do protocolo Bluetooth. Em seu trabalho o autor destaca que os resultados obtidos através de métodos simulados de avaliação de consumo energético costumam apresentar pouca precisão ou não consideram ambientes fechados com alta possibilidade de reflexão do sinal eletromagnético. A fim de obter resultados mais precisos, o trabalho aponta a necessidade de construção de um sistema mais eficiente para medição de consumo de energia, capaz de representar os efeitos das diversas variáveis ambientais (localização, obstáculos, ruídos eletromagnéticos externos, etc.). Na citada dissertação o autor desenvolve um hardware simples e de baixo custo capaz de medir o consumo, apresenta um modelo de consumo de energia e constrói um sistema capaz de avaliar experimentalmente tal consumo.

O hardware desenvolvido nesse trabalho (UMDC-Unidade de Medida de Consumo) realiza a medição da corrente que percorre o adaptador Bluetooth, convertendo-a de sinal analógico em digital. Posteriormente transmite serialmente tais medições a um computador.

O modelo de consumo (SMCEESF) proposto no trabalho acima utiliza-se da variação de corrente do equipamento sem fio como principal mecanismo de observação das variações energéticas em diferentes tipos de cenários e ambientes avaliados. Para o protocolo Bluetooth, o modelo avalia o consumo total em função da soma dos

consumos dos seguintes estados: ativo sem conexão, scan, page , inquiry, ativo e conectado, transmissão e recebimento.

Utilizando-se do sistema desenvolvido o autor realiza avaliações experimentais em seis cenários de interesse à caracterização do consumo energético do protocolo Bluetooth.

Em relação à eficiência dos protocolos de comunicação, verificamos em [1] o comportamento do protocolo TCP/IP em três diferentes sistemas operacionais (FreeBSD 4.2, FreeBSD 5 e Linux 2.4.7). Utilizando adaptador 802.11b os autores analisam em plataformas PC (Intel) e Pocket PC (ARM) os custos energéticos relativos ao envio e à transmissão. Inicialmente, realizou-se o experimento em topologia onde os erros de transmissão e recepção são inexpressivos e obtiveram-se custos relativos aos seguintes processamentos envolvendo o protocolo TCP/IP: cálculo de checksums, cópia dos dados do adaptador para o kernel, cópia dos dados do kernel para o modo usuário e custos relativos a demais processamentos. Posteriormente, introduziu-se perda na transmissão, realizando-se também a análise do custo relativo à esgotamento de tempo (TO), resposta ao recebimento de três ACKS (detecção de perda) e resposta a ACKS esperados (manutenção correta da janela de transmissão). As medições referentes ao consumo energético foram realizadas com a utilização de multímetro (Agilent 34401A).

Dos resultados apresentados, alguns são de especial interesse ao presente estudo e serão referidos ao analisarmos os resultados obtidos neste trabalho. São eles:

- Utilização energética quando não há perda: verificamos na tabela 2.1 a distribuição dos custos energéticos associados a operações realizadas na comunicação TCP/IP. Observamos que os experimentos relatam que grande parte (mais de 72%) dedica-se a operações relativas à cópia da informação entre o adaptador e o kernel. Percentual expressivo da energia é utilizado na transferência da informação do kernel à aplicação.
- Consumo energético do transmissor: para o dispositivo analisado observamos no gráfico 2.2 o consumo energético relativo à transmissão de pacotes TCP/IP de diversos tamanhos.
- Vazão do transmissor: no gráfico 2.3 a vazão do protocolo TCP/IP é apresentada para pacotes de diversos tamanhos.
- Otimização do transmissor através de "zero copying": consiste em reduzir a cópia de informação para o adaptador visto o alto de consumo energético associado a tal operação (tabela 2.1). Duas formas são avaliadas: reaproveitamento do buffer de envio em caso de perda de pacotes e envio de mais de um pacote ao adaptador. O primeiro exige mudanças no driver de forma que

contemple mecanismo de retransmissão que reaproveite o pacote já existente no buffer do adaptador. O segundo caso exige mudanças no driver a fim de que possam ser informados os limites (início e fim) de cada pacote a transmitir. Para a primeira otimização, os autores indicam redução de até 10 % no consumo, para o caso de transmissão com maiores percentuais de erros na comunicação (10%)

	BSDv4.2	BSDv5	iPAQ
NIC-to-Kernel	78%	77%	72%
Kernel-to-user	16%	15%	15%
TCP Proc.	6%	8%	13%
Metric	Energy	Energy	Energy

Figura 2.1: Utilização energética. Extraído de [1]

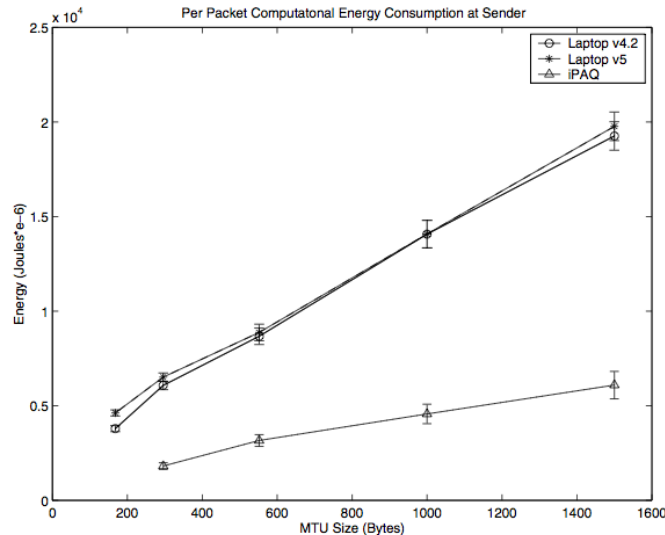


Figura 2.2: Consumo energético do transmissor TCP/IP. Extraído de [1]

Outras propostas, como a troca de conexão (por exemplo, alternância entre Wi-Fi e 3G) e a pesquisa de materiais que produzam dispositivos com menor consumo, contribuem para melhor eficiência energética, entretanto, fogem ao escopo principal do presente trabalho.

2.2 Protocolo TCP/IP e suas limitações

Como relatado por HANDLEY [15], grande parte da evolução da Internet ocorreu em consequência da imediata necessidade de mudança ou avanço.

Diante da rápida penetração da Internet, sua arquitetura em camadas passou a ser amplamente difundida. Inúmeros protocolos e aplicações passaram a ser criados

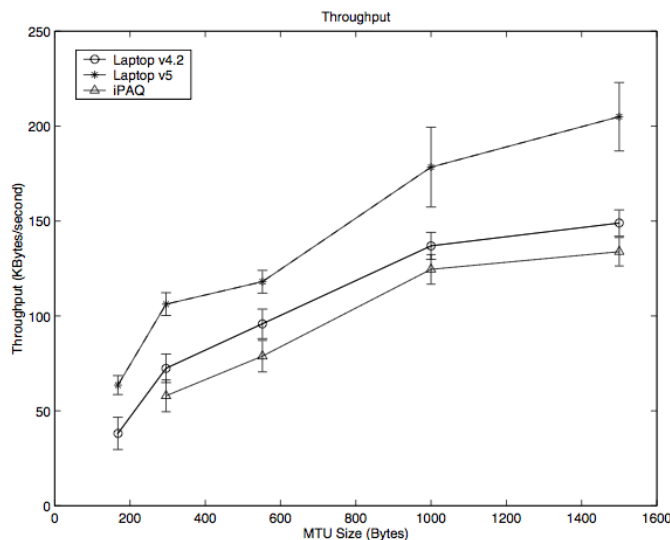


Figura 2.3: Vazão do transmissor TCP/IP. Extraído de [1]

com base no TCP/IP. Funcionalidades decorrentes das necessidades de segurança, mobilidade e escalabilidade no provimento de informações passaram a ser resolvidas, em grande parte, na camada de aplicação, tendo como base o TCP/IP.

Originalmente tal protocolo foi concebido para operar em redes envolvendo nós interligados por fios, onde congestionamento e atrasos inesperados são os principais motivos de perda de pacotes.

A fim de evitar tais problemas, o transmissor utiliza-se de mensagens de confirmação de recebimento, efetuando retransmissão de eventuais pacotes perdidos e garantindo entrega. Média do tempo de entrega e desvio de tal tempo são calculados durante toda a transmissão.

Caso o transmissor receba mensagens duplicadas de confirmação ou o tempo máximo de recebimento seja excedido, é identificada a perda de pacotes.

Como prevenção a tais perdas, o tamanho da janela de envio é diminuído e o intervalo entre transmissões aumentado, diminuindo-se a carga nos links intermediários.

Embora funcione muito bem na prevenção de congestionamento, um dos principais problemas verificados na transmissão sem fio é a elevada taxa de erros de bit (BER) [16], quando comparada à transmissão com fio. Portanto, o uso de TCP/IP em redes sem fio introduz problemas ainda em aberto, a exemplo dos apresentados em [17] e [18].

Além dos problemas relacionados ao uso do TCP/IP em redes sem fio, verificamos em PAN *et al.* [19] e em STUCKMANN e ZIMMERMANN [20] que a atual arquitetura, em formato de amпуlhetas, tem o protocolo IP no centro de seu gargalo.

Devido a seu amplo uso e por se basear na comunicação fim a fim entre o solicitante da informação (origem) e o provedor (destino), diversos destinos precisam

adotar soluções independentes, de alta complexidade e custos a fim de atenderem às atuais necessidades no provimento de informação. Como exemplo, verificamos em WENDELL e FREEDMAN [21], através da coleta de dados por quatro anos em um CDN (do inglês content distribution network), que as atuais técnicas utilizadas para atendimento de solicitações em cenários de pico (cache e alocação dinâmica de recursos) por vezes se mostram incapazes de atender a demanda, culminando na indisponibilidade do serviço.

2.3 Rede endereçada por Interesse

Em contraste ao modelo de comunicação fim a fim no qual se baseia o TCP/IP, tal modelo propõe que a camada de rede passe a endereçar a informação por nomes (interesses) ao invés de endereços IPs. Ao longo de toda rede, mecanismos de cache e distribuição de conteúdo seriam capazes de determinar, somente com base no interesse, a quem solicitar, quando armazenar e como distribuir a informação. A informação chega ao solicitante sem que haja menção explícita a um endereço de origem, como é feito hoje com o IP, ocorrendo, portanto, um desacoplamento entre a informação e um endereço fonte específico.

Ao longo dos últimos anos, desde que CHERITON e GRITTER [22] propuseram a utilização de roteamento baseado em conteúdo, verificamos o surgimento de diversas arquiteturas para redes endereçadas por interesse. Tais propostas passaram a apresentar soluções a problemas que podemos classificar e resumir como:

Identificação

O conteúdo é identificado independente de sua localização. Pode possuir alguma hierarquia ou não.

Resolução e roteamento

Estratégias para comparação do identificador (a fim de verificar a existência do conteúdo) e determinação do caminho para entrega do conteúdo. Dependendo da arquitetura, o caminho a ser seguido pode ou não ser o mesmo pelo qual a solicitação chegou ao destino.

Armazenamento temporário (cache)

O armazenamento temporário pode se dar ao longo do caminho percorrido pela solicitação ou pode haver suporte para que ocorra fora dele.

Para o segundo caso, devemos observar a estratégia de resolução e roteamento adotada. Se o caminho dos dados for o mesmo da solicitação, o sistema de encami-

nhamento deve dar suporte ao cache. Caso seja distinto, o sistema de resolução é o responsável por dar suporte ao armazenamento temporário fora do caminho.

Mobilidade

Dois casos distintos são tratados: requisitantes e publicadores de informação. No primeiro caso, basta o envio de nova requisição. Já no segundo, a arquitetura deve atualizar o sistema de roteamento ou de resolução de nomes, dependendo da estratégia de entrega de conteúdo (mesmo caminho da requisição ou distinto).

Segurança

A fim de prover segurança, deve haver uma relação de confiança entre o requisitante e o provedor da informação ou um agente externo capaz de validar a origem da informação. Como descrito por GHODSI *et al.* [23], a grande diferença para implementação de segurança entre as diversas arquiteturas propostas está no fato de utilizarem ou não identificadores de fácil compreensão humana.

2.3.1 Exemplos de arquiteturas

A fim de ilustrar os problemas descritos, apresentamos, brevemente, nesta seção as abordagens adotadas na resolução das classes de problemas descritas acima. Tais exemplos também têm como objetivo servir de referência bibliográfica a arquiteturas que contribuem à área de redes endereçadas por interesses.

Identificação

DONA 2007 [24], uma das primeiras arquiteturas de endereçamento por interesses, não utiliza nenhuma hierarquia na identificação do conteúdo. O conteúdo é identificado por um hash composto de uma chave pública P associada ao publicador e um nome L que identifica o conteúdo.

NDN [25] (originalmente conhecido como uma implementação CCN [26], mas que a partir de 2013 deixou de usar tal código) utiliza identificação hierárquica. Versionamento e segmentação se utilizam da hierarquia do identificador (ex: /dados/_v1/_s1).

Resolução e roteamento

MobilityFirst 2012 [27] utiliza-se de identificadores chamados de GUIDs. Um serviço de resolução (GNRS) é responsável por traduzi-los em endereços de rede. Um publicador registra seu GUID em um GNRS. Um subscritor informa o GUID que deseja

e seu próprio GUID a um roteador local. Cabe ao roteador contactar o GNRS, escolher o endereço de rede de onde obterá o conteúdo e encaminhar um pedido GET através da rede até que o conteúdo seja obtido. Dados e requisição (GET) seguem caminhos diferentes (desacoplamento).

COMET 2011 [28] utiliza-se da primitiva REGISTER a fim de permitir que um publicador informe disponibilidade de um conteúdo. Um CRS (sistema de resolução de conteúdo) local informa a seus pais da disponibilidade de tal conteúdo. Um subscritor envia uma primitiva CONSUME que, de maneira similar ao publicador, se propaga através do CRS. Uma vez encontrada a informação, essa segue o caminho pelo qual a requisição foi feita (acoplamento entre requisição e entrega dos dados). Além do modo acoplado, existe a possibilidade de se operar de forma desacoplada.

Armazenamento temporário (cache)

NDN [25] utiliza o mesmo caminho da solicitação para a entrega dos dados. O cache através do caminho é nativo, mas dado o volume de dados, acaba sendo substituído rapidamente. Na prática é útil para momentos de pico ou recuperação de pacotes. Uma implementação que dê suporte à obtenção dos dados fora do caminho é possível mas custosa, pois exigiria a manutenção da lista de conteúdos espalhados nos servidores, que, por sua vez, são constantemente trocados.

PURSUIT 2010 [29] possui desacoplamento entre o caminho da solicitação e o da entrega. O cache ao longo dos servidores do caminho da requisição é suportado mas é pouco eficiente, uma vez que os dados não necessariamente serão enviados por eles. Já o cache desacoplado é implementado através da divulgação, por parte dos servidores de cache, da informação de que são publicadores do conteúdo guardado no cache.

Mobilidade

Em NDN [25], um novo requisitante pode ingressar na rede apenas mandando uma mensagem exprimindo seu interesse. Já para anúncio do publicador, utiliza-se o algoritmo "Label First, Broadcast Later"[30]. Resumidamente, se um novo publicador (p1) responder a uma requisição, o nó responsável por atualizar a rede com a informação de tal publicador (p1) aguarda a resposta de algum outro publicador (p2). Se outro (p2) responder e já existir, posterga-se a atualização da informação de um novo publicador (p1). Resultados dos experimentos realizados no simulador QualNet mostraram que "Label First, Broadcast Later" superou o protocolo AODV [31], na grande maioria dos experimentos, como detalhado em [30].

Segurança

Na arquitetura NDN [25] as mensagens são assinadas por seus publicadores. As chaves públicas podem vir na mensagem e serem certificadas por agente externo. O uso de hierarquia no identificador facilita, uma vez que um servidor em "ufrj.br" pode ser o certificador de um conteúdo identificado como "/ufrj.br/coppe/repi.html". Ataques do tipo DOS ("deny of service") são mitigados haja vista que o conteúdo está distribuído em diversos roteadores. Inúmeras requisições a conteúdos inexistentes ("interest flooding") provocam um aumento na tabela PIT (requisições pendentes) e representam um ataque a ser verificado. Alteração de conteúdo ("content poisoning") e outros tipos de ataques, aos quais as redes endereçadas por interesse estão sujeitas, são abordados em [32].

2.4 Protocolo REPI

Rede Endereçada por Interesse (REPI) foi originalmente apresentada em [2] com o objetivo de operar em redes ad hoc. Nesse trabalho foram apresentadas três propriedades distintas desse original protocolo de comunicação, a saber:

- 1) a rede ser endereçada por termos.
- 2) a rede só formar-se quando uma entidade deseja enviar mensagem.
- 3) a ausência de endereçamento convencional fim a fim.



Figura 2.4: Cabeçalho REPI proposto e extraído de [2]

Resultados (taxa de entrega de mensagens, custo de entrega, taxa de perda e número de nós colaboradores) obtidos em três experimentos representativos foram apresentados em [2] e comparados com os algoritmos de inundação (Flooding) e transmissão probabilística (Gossip).

Em sua dissertação [33], o autor aprofundou-se na caracterização da REPI, apresentando seu modelo, uma implementação em redes de sensores e sua avaliação. Uma mensagem REPI possui um cabeçalho, chamado de prefixo, que possui duas partes: característica e interesse. Em [33], as características seguem uma distribuição normal. Em trabalho descrito a frente, avaliou-se, em relação ao recebimento de mensagens, influência da distribuição probabilística escolhida pelo protocolo.

Durante o desenvolvimento do protocolo REPI, um sistema [34] para avaliação de algoritmos de roteamento ad hoc foi criado (SAMCRA). Conforme relatado pelos autores, SAMCRA tem como objetivo superar as dificuldades na avaliação de um novo algoritmo, dentre as quais destacaram:

- 1) grande quantidade de variáveis a monitorar e analisar, tanto de forma global quanto individual a cada nó.
- 2) realização de grande número de repetições para oferecer resultados confiáveis.
- 3) necessidade de configuração da rede de forma homogênea com várias configurações diferentes, a fim de avaliar os impactos causados pelas diversas parametrizações.

Embora tenha surgido da necessidade de avaliação da REPI, SAMCRA também foi utilizado na avaliação de outros protocolos, sendo um ambiente totalmente independente do protocolo que motivou sua criação.

Um comparativo entre o modelo OSI e o utilizado pelo REPI nas camadas de rede é apresentado em [3]. Nesse trabalho são apresentadas propostas de algoritmos de comunicação e encaminhamento para a camada denominada WORD, equiparável às camadas de enlace e rede do modelo OSI. As propostas compreendem algoritmos com e sem memória.

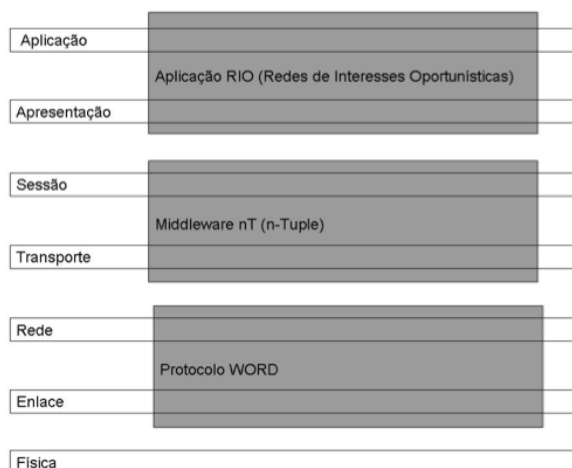


Figura 2.5: Equivalência REPI e modelo OSI. Proposto e extraído de [3]

Em [35], DUTRA *et al.* introduzem o conceito de Prefixos Ativos. Demonstrações

a respeito da influência do tamanho do identificador probabilístico em redes de prefixo ativo são apresentadas. Para as distribuições uniforme e normal verificou-se o número de variáveis aleatórias e respectivos comprimentos a fim de que os prefixos gerados permitam que a rede apresente maior percentual de recebimento. Mais a frente, exemplificamos a importância do uso de Prefixos Ativos no cabeçalho REPI, no que há pouco descrevemos pelo nome "característica".

Um protocolo P2P (REPI-Internet) foi apresentado em [36]. Utilizando-se de prefixos ativos no protocolo REPI introduzido em [33], REPI-Internet abstrai as questões físicas da rede IP sobre a qual se baseia a Internet. A fim de tornar tal abstração possível, são apresentados detalhes a respeito da transposição da dificuldade em se estabelecer e manter conexões P2P através dos dispositivos NAT amplamente utilizados na atual arquitetura da Internet. Também se encontram descritos e avaliados no trabalho o detalhamento do algoritmo, incluindo descrição das mensagens necessárias ao estabelecimento de conexão e à manutenção da rede de vizinhos.

O termo RADNET é definido em [37] como acrônimo ao seu título em inglês (inteRest-centric AD-hoc NETwork). A implementação até então chamada REPI (aplicada a redes de sensores), passou a chamar-se REPI-A e o termo REPI passou a referir-se a sua implementação sobre a Internet. Nesse trabalho comparou-se o algoritmo REPI ao de inundação para determinados tamanhos de grupos na rede (percentuais de interessados). Dentre outros resultados, verificou-se que para percentuais de interessados de 5% e 10%, REPI reduziu em 30% e 17%, respectivamente, o número de mensagens na rede. Para os mesmos percentuais de interessados, o tempo de entrega do REPI foi de 90ms e 140ms, enquanto que os tempos do protocolo de inundação foram de 126ms e de 200ms. Ambos atingiram 99% dos nós interessados.

Em [38], RADNET é apresentada e comparada aos protocolos AODV [31] e AODV+Gossip3 [39] (G3AODV). Nesse trabalho DUTRA *et al.* destacam três motivos pelos quais o uso de Prefixos Ativos ao invés de interesses são importantes no que diz respeito ao encaminhamento das mensagens. As figuras 2.6 até 2.10 ilustram o uso de Prefixo Ativo.

Exemplificado o encaminhamento proposto, destacamos os três motivos pelos quais o uso de Prefixos Ativos é preferível:

- Evitar problemas relacionados ao uso de campos texto de tamanho variável como forma de identificação;
- Interesses mais populares podem causar inundação enquanto que os menos populares podem ter maior dificuldade em atingir seus destinatários;
- Prefixos ajudam na resolução de problemas de isolamento uma vez que o casamento de parte do Prefixo permite o encaminhamento por parte dos vizinhos.

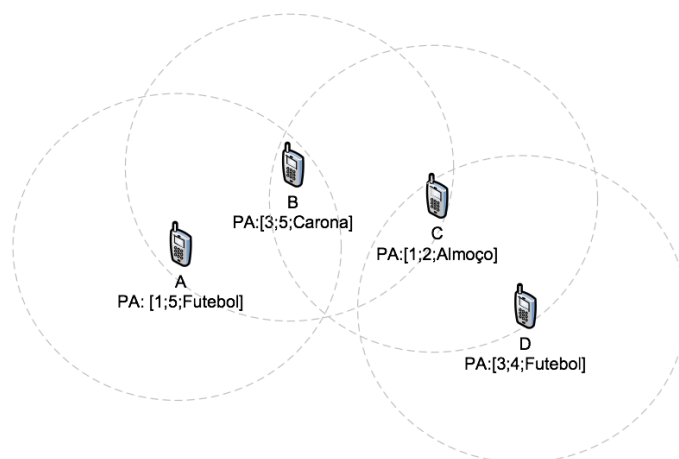


Figura 2.6: REPI: PA composto de dois campos de características

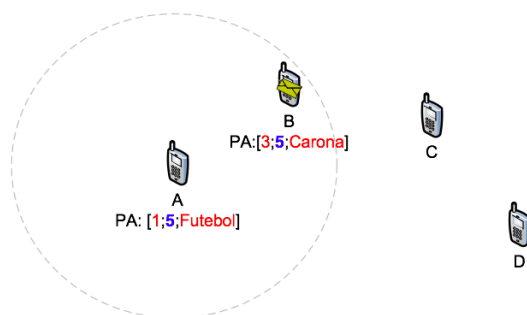


Figura 2.7: REPI: Nó A envia mensagem com seu PA: [1;5;Futebol]

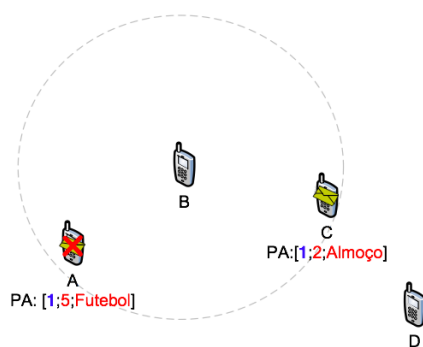


Figura 2.8: REPI: Nó B encaminha a mensagem por haver casamento

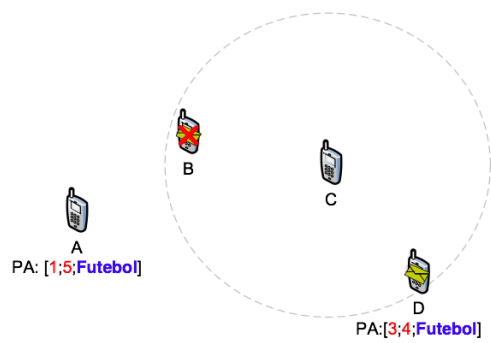


Figura 2.9: REPI: Nó C encaminha a mensagem por haver casamento

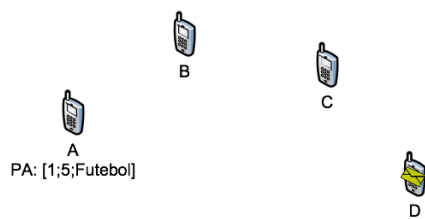


Figura 2.10: REPI: Nó D aceita a mensagem, mas não a encaminha

Um serviço (REPD) implementado para sistemas operacionais baseado em Linux (Ubuntu 10.4+ e Android 2.1/2.2) foi apresentado em [40], juntamente com APIs (REP API) que facilitam o desenvolvimento de aplicações centradas em interesse. Uma vez que a aplicação defina os interesses do usuário, REPD constrói pacotes REP e os encapsula utilizando quadros Ethernet. REPD encaminha os pacotes recebidos caso o critério de casamento de prefixos descrito em [38] e exemplificado acima seja atendido. O conteúdo da mensagem recebida (payload) é enviado à aplicação que tenha indicado interesse igual ao da mensagem recebida ou é descartado caso não haja aplicação interessada.

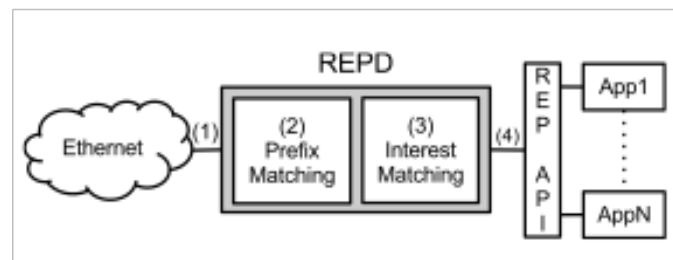


Figura 2.11: Serviço REPAD. Proposto e extraído de [3]

Em sua tese [41], DUTRA apresenta estudos mais rigorosos a respeito da eficiência da RADNET, implementando para isso o protocolo REP, o Gossip3 e o G3AODV no simulador NS-3.8. Dois modelos de mobilidade (Randomwaypoint e 2D-Gauss-Markov) foram utilizados. Avaliações a respeito do tamanho dos Prefixos Ativos e do uso de distribuições Normal e Uniforme na criação dos mesmos também foram apresentados. Dentre outros resultados, as simulações demonstraram que a RADNET, em média, obteve taxa de entrega 16 % superior e uma ordem de grandeza menor em relação à latência e ao total de mensagens recebidas dos outros dois protocolos.

Em [42], COELHO propõe a utilização do mecanismo de comunicação descrito em [40] em substituição ao implementado no Framework UFF [43]. Em especial, MARELI *et al.* apontam como justificativas à substituição todo suporte desenvolvido em [40] para que o protocolo funcione na internet e sua compatibilidade com a plataforma Android.

Ainda no trabalho [42], utilizando-se do protocolo REPI, foi desenvolvido um mecanismo de invocação remota de métodos compatível com o protocolo Json-RPC. Com base em tal desenvolvimento o autor detalha aplicações desenvolvidas, dentre as quais destacamos:

- Aplicação de monitoramento de acidentes com idosos;
- Aplicação de exibição ou audição de conteúdo em dispositivos móveis, televisões

e rádios;

- Aplicação de notificação de incêndio, arrombamento ou assalto.

2.5 Discussão

Questões relacionadas à escassez de recursos energéticos e uma busca por melhor eficiência são de interesse da comunidade científica como observamos em [10], [11], [12].

Em relação à metodologia de avaliação experimental de protocolos de comunicação em redes sem fio, verificamos em [14] e [1] que ambos os trabalhos utilizam metodologia de avaliação do consumo energético baseadas em medições da corrente percorrida no dispositivo de comunicação avaliado. Em ambos os casos, medições realizadas por equipamento ou circuito externo são coletadas, em intervalo de tempo determinado pelos autores, onde o sistema avaliado encontra-se dedicado à realização de função específica (ex.: transmissão, recepção, scan). Dos dados coletados, referentes a cada função ou estado avaliado, estimou-se o consumo com base nas médias aritméticas das medições.

A ampla utilização do protocolo TCP/IP como base para diversas aplicações torna sua substituição um desafio. Problemas como os apresentados justificam a pesquisa de outros modelos de comunicação. A mudança de um modelo de comunicação fim a fim para um orientado a interesses introduz uma série de novos problemas em aberto, uma vez que a arquitetura é muito mais dinâmica e colaborativa ao longo de todo o caminho percorrido, tanto pela requisição quanto pela informação.

Em relação aos grupos de problemas descritos como relacionados a redes por interesses, verificamos que o serviço REPD já aborda questões referentes aos grupos classificados como Identificação, Resolução / Roteamento e Mobilidade. Questões relativas à Segurança e à Armazenamento Temporário ainda não foram motivo de estudos publicados.

Em [40] verificamos uma implementação do protocolo REPI para o sistema operacional Android. Sua utilização é vista não só em trabalhos desenvolvidos pela instituição ao qual seus autores pertencem, como também por outras instituições [42]. Um dos fatores que contribui para que futuros trabalhos possam ser desenvolvidos é a disponibilização, em [44], do código fonte.

O fato de REPD utilizar-se de Prefixos Ativos como forma de identificação impossibilita a utilização do identificador no auxílio à resolução de problemas relativos à Segurança. Verificamos em [45] que identificadores hierárquicos facilitam a identificação de agentes certificadores. Outra opção, através do uso de identificadores "auto-certificáveis" (baseados em "digest" do resultado de algoritmo criptográfico),

provém nomes únicos, eliminando questões de manutenção de validação do conteúdo. Como REPI gera aleatoriamente seus identificadores, soluções relativas à implementação de Segurança não poderão se basear no identificador.

Por se tratar de uma possível arquitetura a ser utilizada e pelo fato de outros trabalhos baseados no protocolo estarem em desenvolvimento, a análise do serviço REPD contribui para o melhor entendimento do estado atual do protocolo, em especial, no que diz respeito ao consumo energético.

Capítulo 3

Avaliação Experimental

A avaliação experimental do serviço REPD foi feita utilizando placas Raspberry PI. Como apresentado em [46], [14] e [1], a opção por uma avaliação simulada produziria resultados mediante abstração de diversos aspectos importantes, tais como os efeitos das diversas variáveis ambientais (localização, obstáculos, ruídos eletromagnéticos externos, etc.). Em especial, aspectos referentes à implementação dos códigos envolvidos (por exemplo, o consumo envolvido na cópia dos dados da placa para o kernel) também seriam de difícil simulação e acabariam sendo abstraídos, o que não é de nosso interesse.

A metodologia utilizada é similar à apresentada em [14] e [1]. Utilizamos equipamento externo ao sistema avaliado (multímetro) e realizamos medições de consumo energético referentes a estados de interesse (transmissão e recepção).

3.1 Objetivo

O objetivo é a avaliação do consumo, da capacidade máxima de transmissão e da capacidade máxima de recepção do serviço REPD em uma rede ad hoc. Por se tratar de um protocolo recente (REPI), optamos por compará-lo a um protocolo muito existente e amplamente utilizado: TCP/IP. Dessa forma também obtivemos resultados que contribuam para uma avaliação do TCP/IP na plataforma Raspberry PI.

3.2 Trabalhos relacionados

Em [14] e [1], [MONTEIRO] e [WANG e SINGH] avaliam, com a mesma metodologia aqui utilizada, o protocolo Bluetooth e o TCP/IP. O presente trabalho se diferencia por avaliar o serviço REPD e o TCP/IP, em plataforma distinta das utilizadas pelos autores.

Em [2] e [33], o protocolo REPI é avaliado em ambiente Tmote Sky, com largura de banda nominal de 250Kbps. [GRANJA] estima transmissões com intervalos entre meio e um segundo para que a rede só apresente perdas relevantes nos experimentos com grandes quantidades de mensagens em circulação. O presente trabalho se difere dos citados uma vez que está interessado nas capacidades máximas de recepção e transmissão, ou seja, desejamos determinar o menor intervalo de tempo entre transmissões de tamanho conhecido. Em um simulador o tempo pode ser avançado, já em uma experimentação real desejamos que o tempo seja minimizado e que as amostras ainda sim sejam confiáveis.

Em [36], uma versão do protocolo REPI é simulada através do NS-3.8 (Network Simulator). São quantificados: mensagens (enviadas, recebidas, descartadas, aceitas e encaminhadas), vizinhos, média de saltos, nós colaboradores e tempo para entrega. Como abordamos no Apêndice A e no início deste capítulo, o uso de simuladores muitas vezes oculta problemas só vistos em ambiente real. No presente trabalho objetivamos validar uma implementação específica, obtendo capacidades referentes a seu uso aplicado, diferente do objetivo de [36].

Como verificamos em [42], COELHO menciona que ao utilizar REPD como um daemon em ambiente ad hoc, sua bateria é consumida em curto intervalo de tempo (não especificado). O presente trabalho, embora em outra plataforma, mostra que é possível determinarmos, sem sermos invasivos, uma aproximação do consumo energético, sem nos aprofundarmos em detalhes da implementação.

SEDDIK-GHALEB *et al.* apresentam em [47], em rede sem fio ad hoc, custos computacionais e energéticos referentes às operações (por exemplo, cálculo de checksums, respostas a timeouts) realizadas nas fases do protocolo TCP/IP (slow start, fast retransmit, fast recovery, congestion avoidance). Quatro variantes são avaliadas: TCP New-Reno, Vegas, SACK e Westwood. A avaliação do protocolo é realizada utilizando-se o simulador NS-2, através da análise do código dos protocolos e da inserção de rotinas para gravação de informações de processamento das funções de interesse. O presente trabalho leva em consideração a importância da realização de uma avaliação energética por parte do desenvolvedor, a nível de usuário da biblioteca implementada, sem que se necessite de acesso ao código da mesma.

3.3 Ambiente Experimental

A seguir descrevemos os aspectos relevantes ao ambiente experimental.

3.3.1 Bancada

Os seguintes elementos compuseram a bancada:

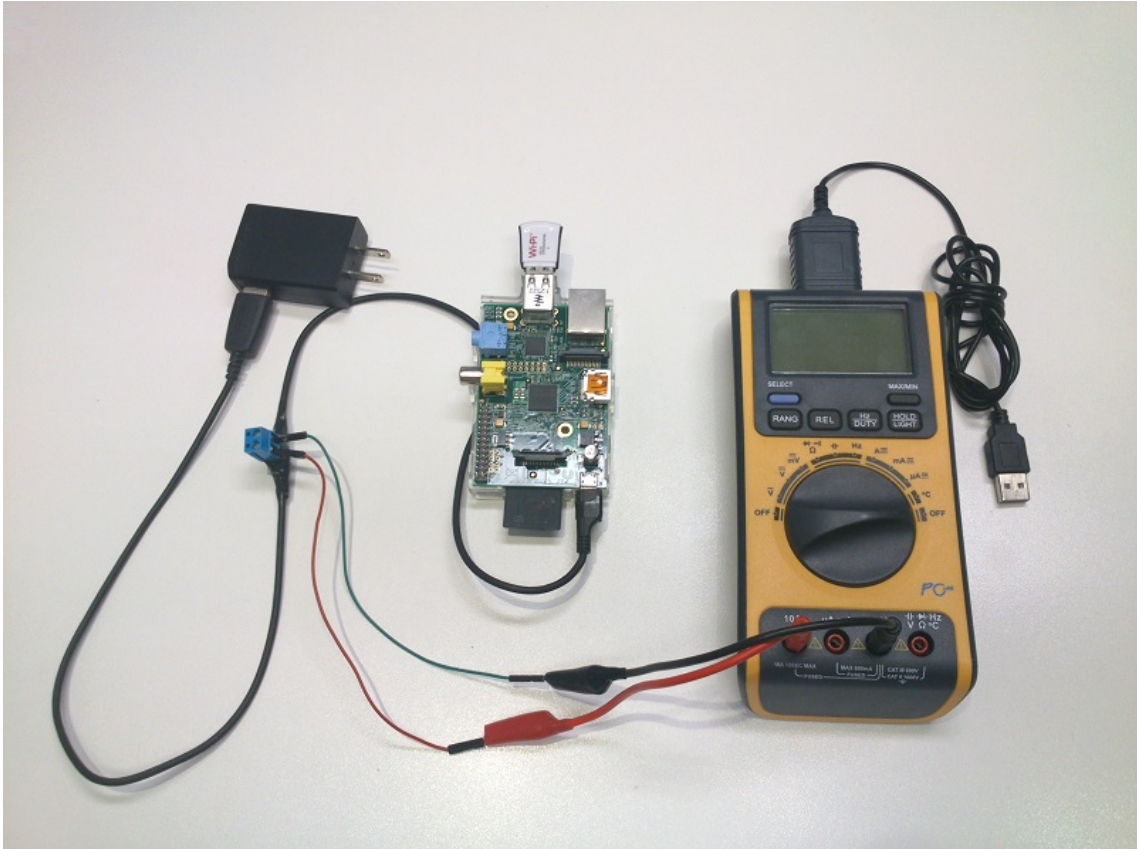


Figura 3.1: Principais elementos da bancada de testes

- Placa Raspberry Pi, modelo B, 512MB de RAM;
- Serviço REPD: Versão 0.2.4;
- Adaptador de Rede : Wi-Pi Padrão: IEEE 802.11n , 20DBm, RTS (Request to Send) : Desabilitado;
- Multímetro GVA-18B, com saída conexão USB através de cabo específico;
- Cabo USB, com conector soldado, de forma a alimentar o Raspberry e permitir a medição da corrente.

O custo dos equipamentos diretamente relacionados às medições de capacidade e consumo (multímetro e cabo de alimentação) foi inferior a 40 dólares.

3.3.2 Topologia

Estamos interessados nas capacidades máximas de transmissão e recepção e seus respectivos consumos. Desta forma os nós foram dispostos lado a lado, próximos um ao outro, a fim de evitarmos problemas não relacionados ao objeto de nosso estudo, tais como os relatados no início deste capítulo, do capítulo 2 e descritos em [46] , [14] e [1].

3.3.3 Sistema Operacional

O sistema operacional utilizado no Raspberry PI foi o Raspbian “wheezy”, baseado no kernel do Linux. As configurações padrões referentes aos protocolos UDP e TCP (Reno) foram mantidas. O adaptador Wi-Fi já é suportado por driver nativo do SO. O multímetro foi conectado a um PC com sistema operacional Microsoft Windows, conforme especificação do aplicativo coletor das medições.

3.3.4 Software coletor de dados

O aplicativo que acompanha o multímetro é o PCLink, compatível com o sistema operacional Windows. Após sua instalação, verificou-se que era capaz de gravar 2 amostras por segundo da corrente que percorre o Raspberry PI. O multímetro e sua conexão a um PC representam um subsistema que não afetam o processamento do Raspberry PI, uma vez que nenhum processamento, comunicação ou acesso a disco ocorre neste.

Um limitador do aplicativo é que a coleta dos dados não pode ser salva diretamente em disco, exigindo que se faça através da interface gráfica. Tal necessidade faz com que os dados coletados se percam em caso de uma falha do sistema operacional ou do programa de coleta, o que não é raro ocorrer.

3.3.5 Codificação

O pseudo código foi implementado em ANSI C e compilado utilizando-se o compilador GCC, incluso na instalação do SO.

Pseudo-código 1 Transmissor

```
para cada tamanho de pacote  $t$  em  $S[]$  faça  
  Aguarde recebimento de mensagem indicando que deve transmitir  
   $T1 \leftarrow \text{TempoAtual}$   
  Envie  $R$  mensagens de tamanho  $t$   
   $T2 \leftarrow \text{TempoAtual}$   
  Armazenar  $T2 - T1$   
fim para
```

A fim de garantir a qualidade da avaliação, algumas considerações foram feitas durante a codificação e sua execução:

- Serviços desnecessários e outros programas foram desabilitados;
- Como podem ocorrer perdas na recepção durante a avaliação do serviço REPD, o receptor aguarda um tempo predefinido antes de considerar encerrada a transmissão de pacotes de tamanho (S). Pelo mesmo motivo o tempo de início

Pseudo-código 2 Receptor

```
para cada tamanho de pacote  $t$  em  $S$  faça
  Enviar msg indicando que transmissão deve ser iniciada
  Aguardar mensagem  $M$ 
   $T1 \leftarrow \text{TempoAtual}$ 
   $\text{BytesRecebidos} \leftarrow \text{tamanho}(M)$ 
   $\text{MensagensRecebidas} \leftarrow 1$ 
  enquanto receber mensagens  $M$  faça
     $\text{BytesRecebidos} \leftarrow \text{BytesRecebidos} + \text{tamanho}(M)$ 
     $\text{MensagensRecebidas} \leftarrow \text{MensagensRecebidas} + 1$ 
  fim enquanto
   $T2 \leftarrow \text{TempoAtual}$ 
  Armazenar  $T2-T1, \text{BytesRecebidos}, \text{MensagensRecebidas}$ 
fim para
```

só é computado após recepção do 1º pacote. Com um número de repetições suficientes, o tempo de recepção do primeiro pacote acaba por não influenciar na medição;

- O processamento durante a transmissão deve ser mínimo de forma a não influenciar nas medições. O tempo final é computado somente quando o tempo de espera de pacote é excedido, evitando chamada ao sistema operacional a cada pacote recebido. Dessa forma, também não se faz necessário nenhuma avaliação do pacote recebido em busca de indicação de término de transmissão, evitando-se respectivo processamento;
- Acesso a disco: as medições realizadas foram armazenadas em disco fora do trecho de código onde o tempo de transmissão/recepção é calculado. Um único buffer em memória, de tamanho suficiente para comportar o maior tamanho (S) avaliado, foi utilizado a fim de que não houvesse acesso a disco, decorrente da paginação (page fault).

3.4 Experimentos

Os experimentos têm como objetivo determinar as taxas de transmissão e recepção dos protocolos TCP/IP e do serviço REPD, para diversos tamanhos de pacote. Para as taxas encontradas, também avaliamos seu consumo.

Os experimentos permitiram:

1. Análise das medições iniciais e eventuais imperfeições;
2. Identificação da estratégia para obtenção de parametrização que produz amostras confiáveis em quantidade suficiente;

3. Análise de resultados com base na parametrização ideal.

A avaliação se deu com os seguintes tamanhos de pacote, em bytes:

(S:) {1, 5, 10, 20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400}.

O serviço REPD na versão avaliada possui um limite de cerca de 1400 bytes para tamanho de mensagens, variando conforme o tamanho do identificador de interesse.

O respectivo consumo para cada valor de (S) também foi obtido. Inicialmente uma avaliação da voltagem da fonte do Raspberry PI foi feita durante duas horas e o valor médio obtido foi de 5.01V.

Por se tratar de uma rede de comunicação sem fio ad hoc, medições muito distintas foram obtidas. Os experimentos realizados permitiram a identificação de tais variações e a determinação da correta parametrização a fim de que os resultados fossem confiáveis.

Abaixo apresentamos tais experimentos, suas motivações e conclusões.

3.4.1 TCP/IP – Experimento 1 - Influência do número de repetições nas taxas de transmissão e recepção

Objetivo

Verificar influência do número de repetições (R) nos resultados obtidos. Para isso tomamos valores aleatórios do número de repetições.

Parametrização

bytes (S:) {1, 5, 10, 20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400}.

repetições (R:) {3.000, 6.000,12.000, 24.000,48.000,96.000,192.000, 384.000}.

Para cada combinação (R,S), avaliou-se 10 vezes a vazão do transmissor e do receptor e tirou-se a média.

Análise das medições iniciais

Ao observarmos as medições iniciais na figura 3.2, dois pontos chamaram a atenção: as taxas médias para um mesmo tamanho de pacote e a diferença entre as taxas médias de recepção e de transmissão, também para o mesmo tamanho de pacote.

Vazão média: Para R=3000, S=1 observou-se que a vazão média de recepção era de cerca de 51KBps. Para R=96000, S=1, o valor foi de cerca de 72KBps..

Diferença nas taxas médias de recepção e envio Observou-se a existência de uma diferença considerável entre a taxa média do receptor (TR) e a do transmissor (TT) Ex: Para o mesmo experimento (R=3000, S=5), encontramos TR=217KBps, TT= 547KBps

	Repetições (R)	Recepção (KBps)			Transmissão (KBps)		
		3000	48000	96000	3000	48000	96000
1		51	70	72	80	73	72
5		217	511	516	547	554	535
10		421	840	850	882	890	880
20		754	1100	1116	1152	1135	1144
50		1273	1987	1903	1641	2029	1921
100		2076	2415	2573	2292	2451	2596
150		1991	2458	2578	2434	2486	2601
200		2398	2581	2641	2649	2643	2649
300		2433	2690	2644	2690	2705	2660
400		2416	2740	2742	2713	2754	2748
500		2626	2736	2727	2760	2746	2733
600		2706	2751	2736	2819	2763	2741
700		2690	2778	2761	2793	2787	2771
800		2733	2775	2782	2906	2782	2786
900		2750	2805	2778	2834	2813	2782
1000		2615	2816	2789	2793	2823	2792
1100		2732	2809	2777	2866	2814	2780
1200		2802	2775	2781	2881	2783	2784
1300		2775	2791	2768	2834	2796	2771
1400		2709	2810	2814	2850	2814	2817

Figura 3.2: TCP/IP. Exemplos de taxas médias obtidas para os diversos tamanhos de pacotes (S) (1a coluna)

Para avaliarmos tal diferença, passamos a utilizar a seguinte expressão:

$$D = 100 * (TT - TR) / TR$$

Ou seja, avaliamos percentualmente o quanto a taxa de transmissão se distancia da de recepção.

Atinge-se estabilidade, para blocos pequenos (1 a 10 bytes), quando o número de envios é acima de 48.000 (valores em negrito na figura 3.2). Para blocos maiores, não há necessidade de tantos envios, conforme apresentado na figura 3.3.

A convergência das curvas a medida que aumentamos o número de repetições pode ser observada na figura 3.4

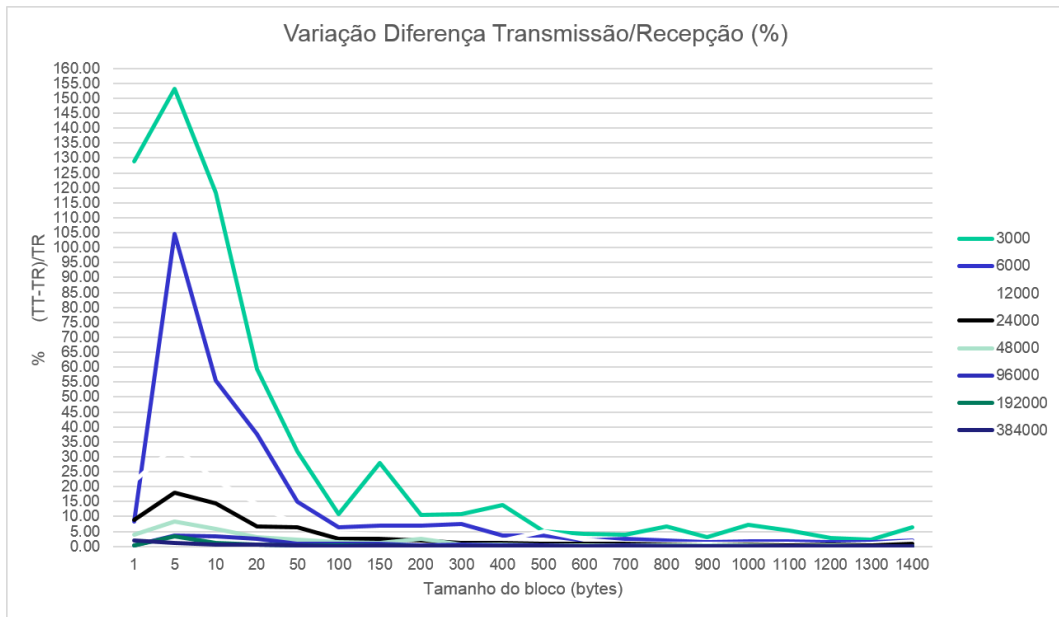


Figura 3.3: TCP/IP. Influência do número de repetições na diferença entre as taxas médias de recepção e transmissão

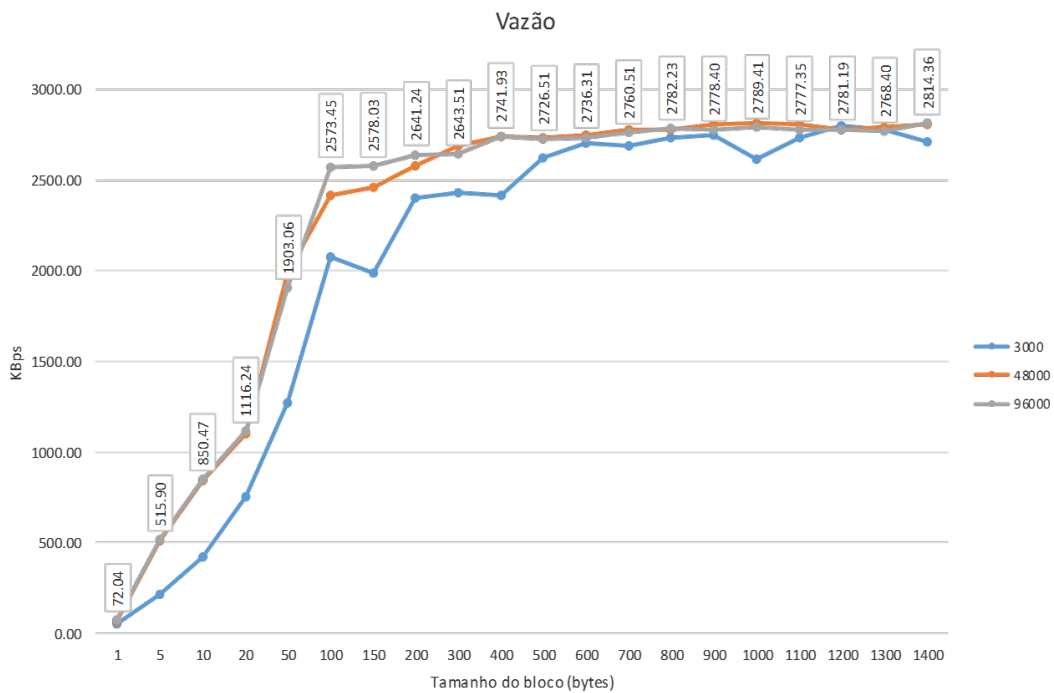


Figura 3.4: TCP/IP. Taxas médias de recepção para 3, 48 e 96 mil repetições

3.4.2 TCP/IP – Experimento 2 - Consumo energético e capacidade de transmissão

Objetivo

Avaliar o consumo de energia e a taxa média de transmissão do protocolo TCP/IP no Raspberry PI. Confirmar, utilizando parametrização encontrada no primeiro experimento, que as taxas do receptor e do transmissor possuirão pouca diferença.

Parametrização

Utilizando as taxas encontradas no experimento 1, procuramos obter 20s de medições (cerca de 40 amostras) da corrente que percorre o Raspberry PI. Com base nos valores apresentados na figura 3.2 para o tamanho de segmento $S=1$, necessitamos de 1.500.000 repetições para transmitirmos durante 20s. No caso de $S=1400$, 42.000 repetições são necessárias.

Parametrização utilizada:

bytes (S:) {1, 5, 10, 20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400}.

repetições (R:) {1.800, 1.680, 1.560, 1.440, 1.320, 1.200, 1.080} x 10^3 repetições para $S=1$ até $S=150$. Para $S=200$ até $S=1400$, 60.000 repetições foram realizadas

Amostras

Para cada combinação (R,S) foram realizadas 54 avaliações das taxas médias de transmissão e de recepção, totalizando 2160 medições.

O aplicativo coletor de dados do multímetro registrou 66440 medições da corrente fornecida ao transmissor.

A corrente média obtida, quando o dispositivo encontra-se em repouso (sem efetuar transmissão ou recepção), foi de 482 mA.

Análise

A primeira observação diz respeito ao número de repetições utilizadas e à qualidade das amostras produzidas.

Como observamos na figura 3.5, a taxa média obtida pelo programa executado no transmissor pouco diferiu da taxa média obtida no dispositivo receptor.

O consumo do protocolo TCP/IP durante a transmissão foi obtido com base na média das medições obtidas pelo software coletor, para cada tamanho de mensagem. Na figura 3.6, podemos observar o comportamento de algumas dessas medições.

Varição Diferença Transmissão/Recepção (%)

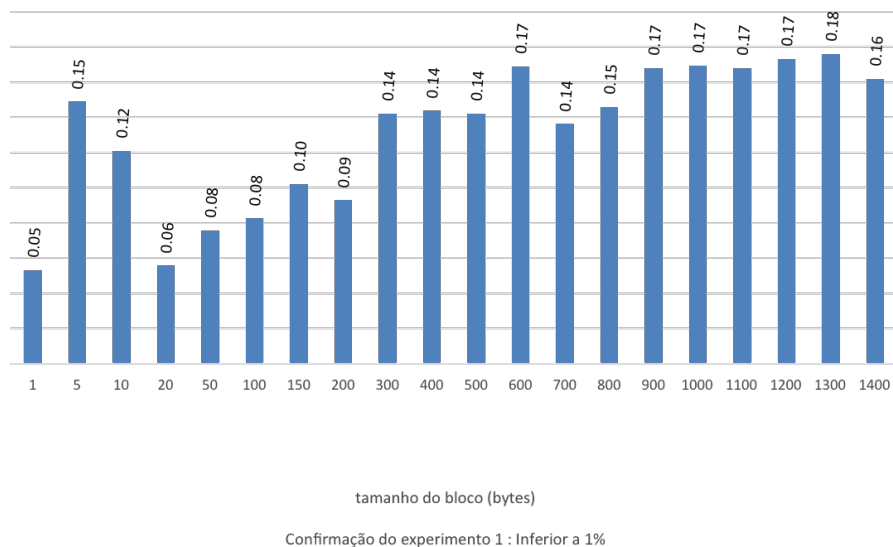


Figura 3.5: TCP/IP. Com suficientes repetições, taxas médias de transmissão e recepção possuem diferença inferior a 1%

A diferença entre tais médias e o consumo médio, quando em repouso, é apresentada na figura 3.7, onde observamos que o consumo aumenta conforme aumentamos o tamanho da mensagem.

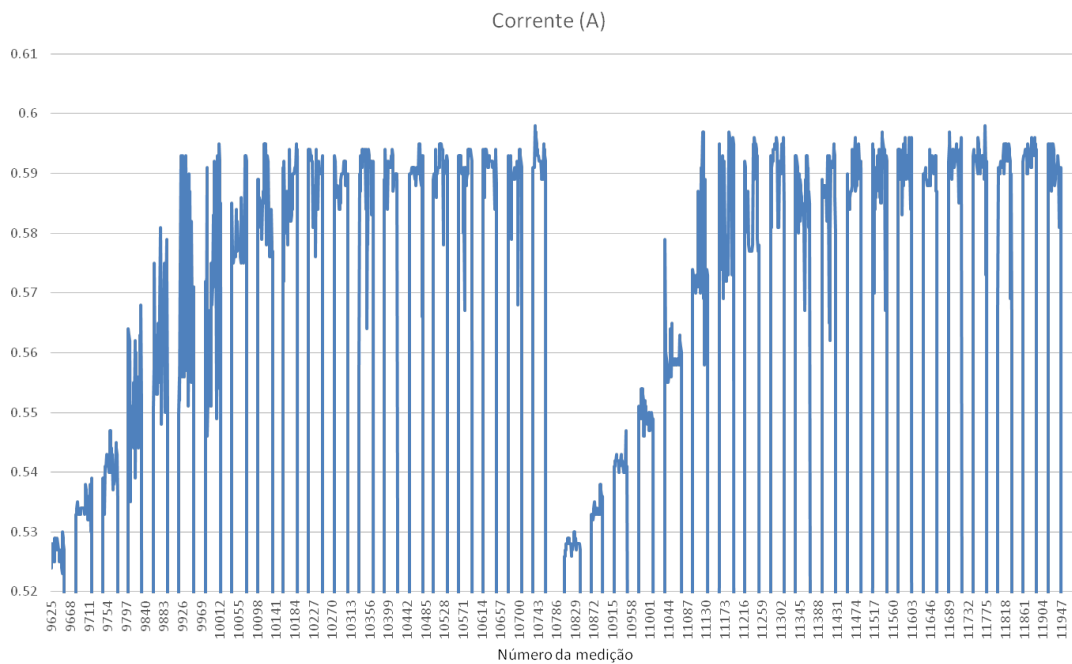


Figura 3.6: TCP/IP. Exemplos de medições obtidas no aplicativo coletor

Outra análise de interesse é o consumo energético relativo ao envio de 1400 bytes.

Consumo energético TCP - mJ /mensagem enviada

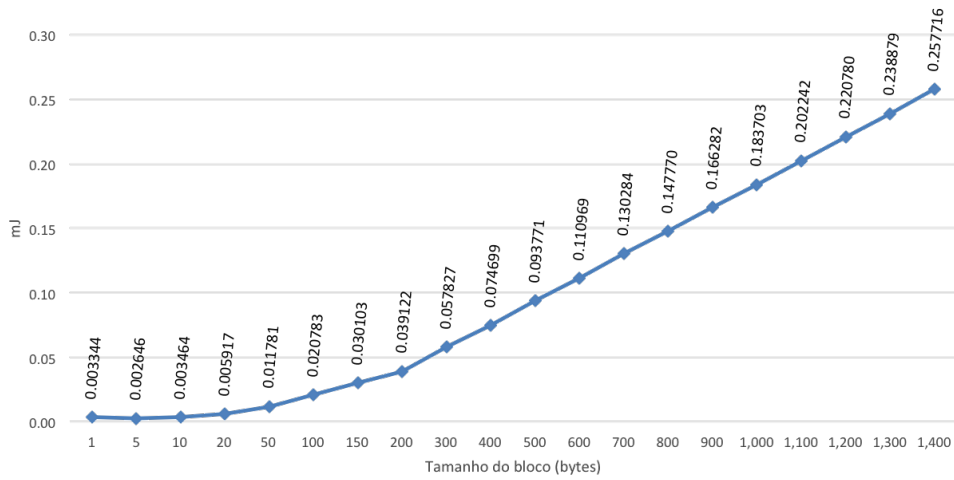


Figura 3.7: TCP/IP. Consumo por mensagem transmitida

Observamos na figura 3.8 que a o consumo é praticamente o mesmo, 0.26 mJ, para mensagens de tamanho superior a 400 bytes.

Consumo energético TCP - mJ p/ envio de 1400 bytes

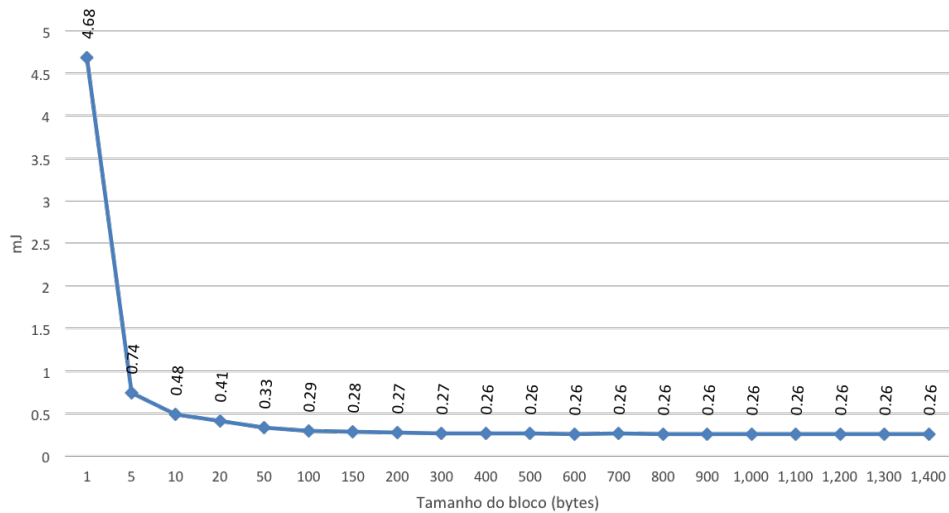


Figura 3.8: TCP/IP. Transmissão de 1400 bytes - Tamanho do pacote vs consumo

Observando a figura 3.9 podemos determinar em quanto estaremos reduzindo o consumo, se procurarmos agrupar o envio de informação em uma mesma mensagem.

Ao agruparmos 5 mensagens de 1 byte e realizarmos um único envio, obtemos uma redução de cerca de 6 vezes no consumo. Aguardar para enviar uma única mensagem de 1400 bytes não produz redução significativa no consumo em relação, por exemplo, ao envio de 7 mensagens de 200 bytes.

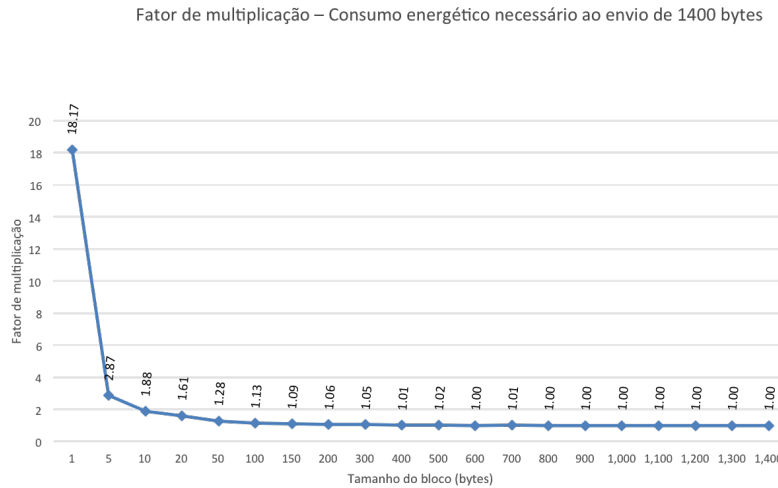


Figura 3.9: TCP/IP. Fator de multiplicação – Consumo energético necessário ao envio de 1400 bytes

Por fim, apresentamos, no gráfico 3.10, a capacidade da rede em termos de vazão e, no gráfico 3.11, em termos de mensagens por segundo.

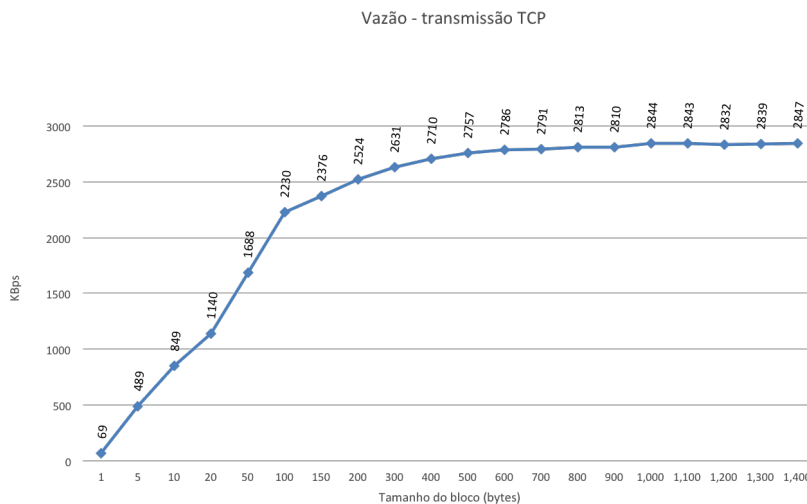


Figura 3.10: TCP/IP. Vazão do transmissor - KBytes por segundo

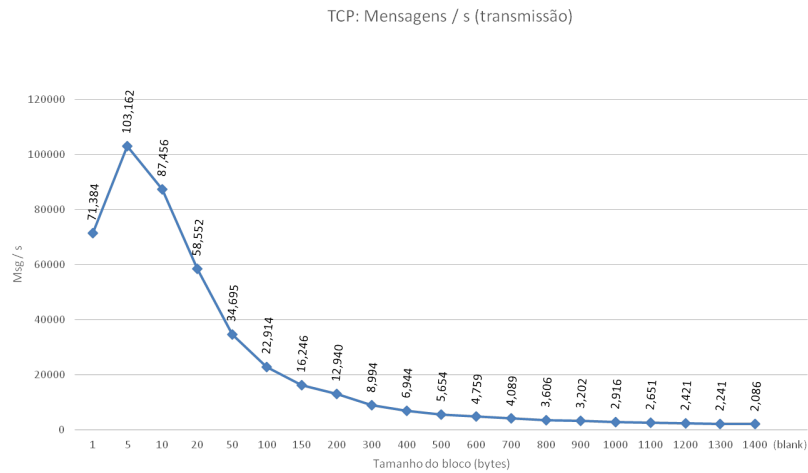


Figura 3.11: TCP/IP. Capacidade de transmissão - Mensagens por segundo

3.4.3 TCP/IP – Experimento 3 - Consumo energético e capacidade de recepção

Objetivo

Avaliar o consumo de energia e a taxa média de recepção do protocolo TCP/IP no Raspberry PI. Avaliar o reflexo no consumo do receptor quando o transmissor agrupa mensagens a fim de diminuir transmissões.

Parametrização

De forma similar ao experimento 2, procuramos obter 20s de medições (cerca de 40 amostras) da corrente que percorre o Raspberry PI.

Parametrização utilizada:

bytes (S:) {1, 5, 10, 20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400}.

repetições (R:) {1.800, 1.680, 1.560, 1.440, 1.320, 1.200, 1.080} x 10³ repetições para S=1 até S=150. Para S=200 até S=1400 60.000 repetições foram realizadas

Amostras

Para cada combinação (R,S) foram realizadas 31 avaliações das taxas médias de transmissão e de recepção, totalizando 620 medições.

O aplicativo coletor de dados do multímetro registrou 36100 medições da corrente fornecida ao transmissor.

A corrente média medida, quando o dispositivo encontra-se em repouso (sem efetuar transmissão ou recepção), foi de 480 mA.

Análise

O consumo do protocolo TCP/IP durante a recepção foi obtido com base na média das medições obtidas pelo software coletor, para cada tamanho de mensagem. O gráfico de comportamento das medições se assemelha ao já apresentado na figura 3.6.

A diferença entre tais médias e o consumo médio, quando em repouso, é apresentada na figura 3.12, onde observamos que o consumo aumenta conforme aumentamos o tamanho da mensagem.

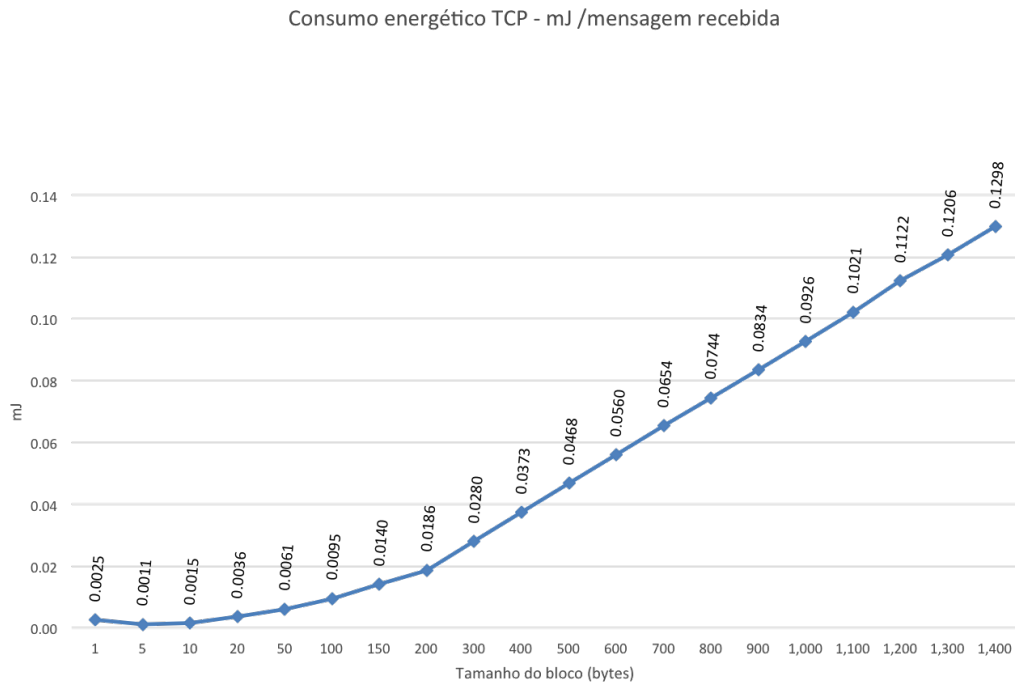


Figura 3.12: TCP/IP. Consumo energético por mensagem recebida

Observamos na figura 3.13 que a o consumo é praticamente o mesmo, 0.13 mJ, para mensagens de tamanho superior a 100 bytes.

Consumo energético TCP - mJ p/ recepção de 1400 bytes

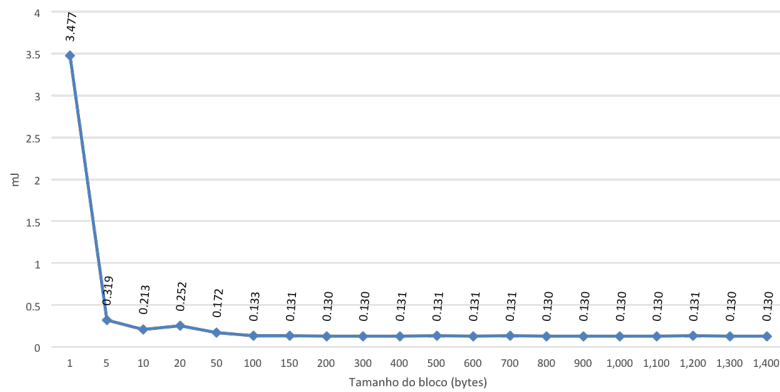


Figura 3.13: TCP/IP. Recepção de 1400 bytes - Tamanho do pacote vs consumo energético

Observando a figura 3.14 podemos determinar o quanto o transmissor pode influenciar no consumo do receptor, ao agrupar o envio de informação em uma mesma mensagem. Aqui verificamos uma redução mais expressiva em relação à observada no transmissor.

Fator de multiplicação – Consumo energético necessário à recepção de 1400 bytes

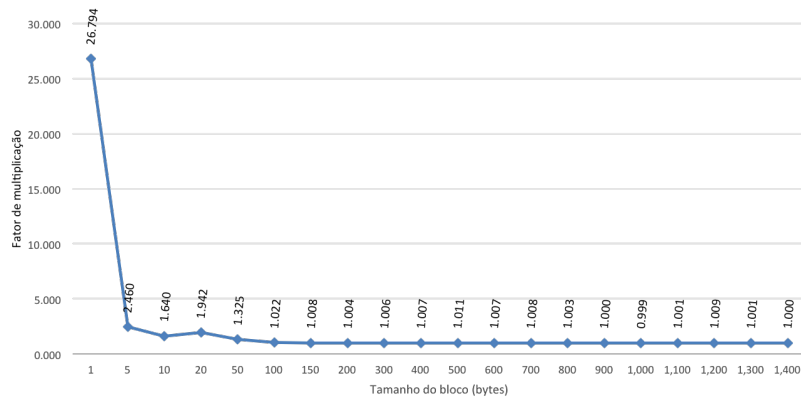


Figura 3.14: TCP/IP. Fator de multiplicação – Consumo energético necessário à recepção de 1400 bytes

Verificamos o impacto causado pelo transmissor no receptor. Ao agruparmos 5 mensagens de 1 byte e realizarmos um único envio, obtemos uma redução de cerca de 11 vezes no consumo do receptor. Aguardar para enviar uma única mensagem de 1400 bytes não produz redução significativa no consumo em relação ao envio de

14 mensagens de 100 bytes.

Por fim, apresentamos a capacidade da rede em termos vazão (gráfico 3.15) e em termos de mensagens por segundo (gráfico 3.16).

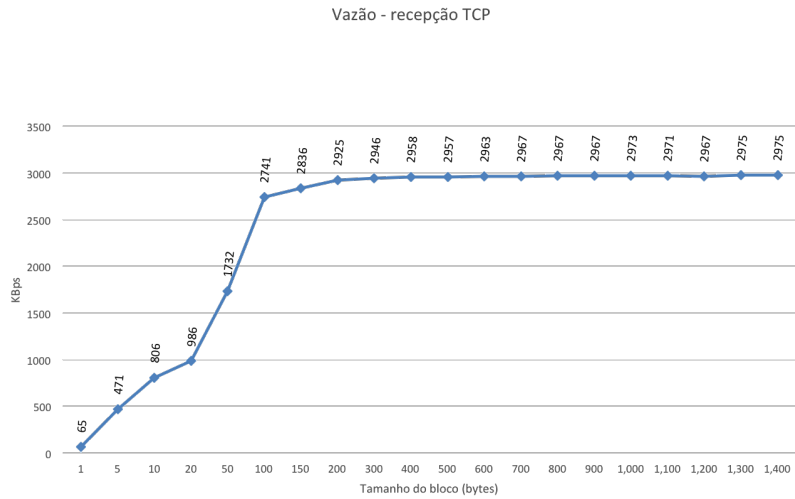


Figura 3.15: TCP/IP. Vazão de recepção - KBytes por segundo

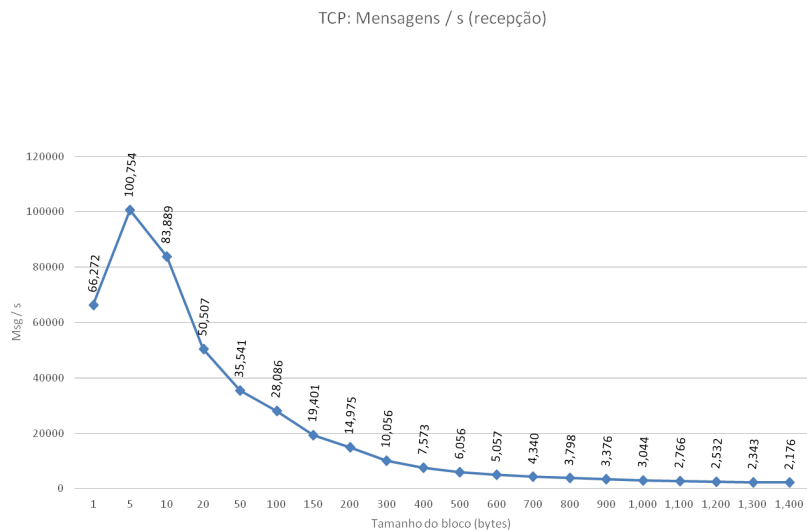


Figura 3.16: TCP/IP. Capacidade de recepção - Mensagens por segundo

3.4.4 REPD – Experimento 1 - Influência do número de envios nas taxas de transmissão e recepção

Objetivo

Verificar a influência do número de envios (R) nos resultados obtidos. Para isso tomamos valores aleatórios do número de repetições.

Parametrização

Parametrização utilizada:

bytes (S:) {1, 5, 10, 20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400}.

repetições (R:) {5000, 10.000, 15.000, 20.000, 25.000, 30.000}

Para cada combinação (R,S), avaliou-se 20 vezes a vazão do transmissor e do receptor e tirou-se a média.

Análise das medições iniciais

Inicialmente observamos as médias das taxas de recepção para os diversos tamanhos de pacotes e para os diversos números de repetições. Diferente do que ocorreu no protocolo TCP/IP, não observamos na figura 3.17 uma convergência da taxa de recepção ao aumentarmos o número de repetições.

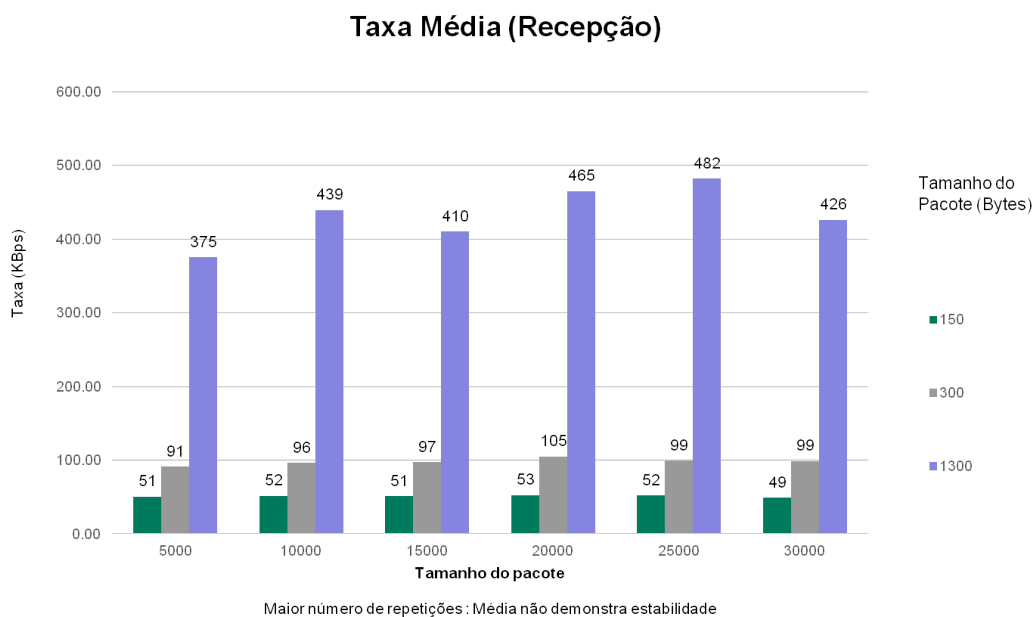


Figura 3.17: REPD. Taxa média de todas as amostras de taxa de recepção

Ao observarmos a variação para os resultados obtidos, verificamos que a média acaba por mascarar uma grande diferença entre as taxas máximas e mínimas para um mesmo tamanho de pacote, independente do número de repetições realizadas.

A figura 3.18 nos mostra um exemplo onde a taxa mínima chega a ser metade da máxima para diversos números de repetições.

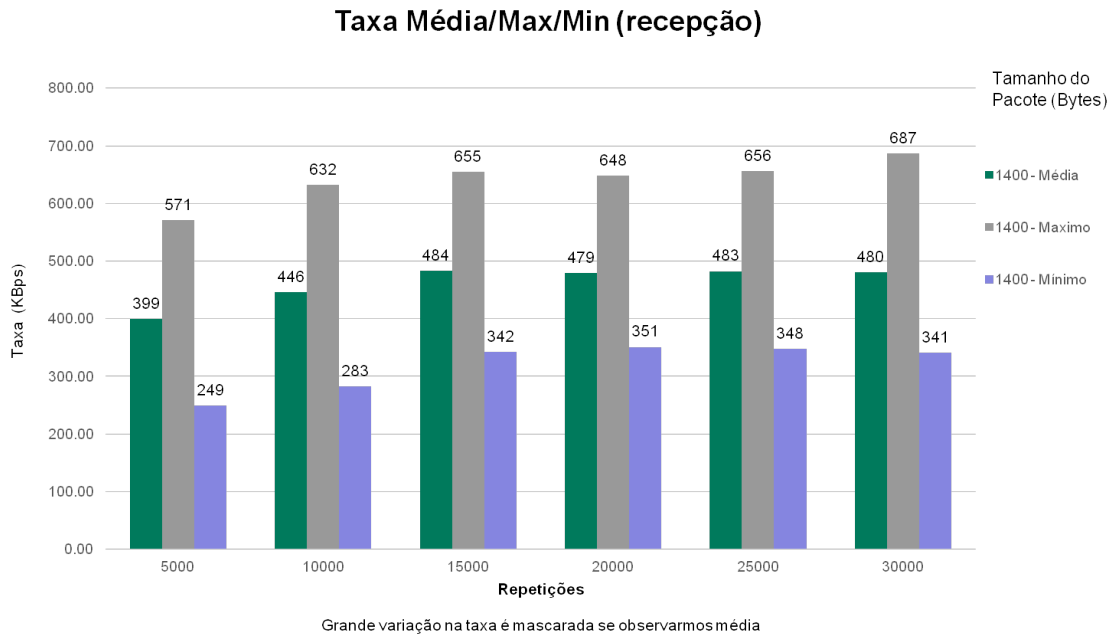


Figura 3.18: REPD. Variação entre as amostras da taxa de recepção

Com o objetivo de obtermos taxas máximas de recepção e seus respectivos consumos, se fez necessário então verificar o que levou a tais diferenças. Para isso, além da taxa de recepção, passamos, então, a considerar o percentual de recebimento, uma vez que esse também era conhecido. A figura 3.19 nos mostra o quão distantes as taxas médias com 88% de recepção estão das próximas a 100%, para pacotes de 600 e 1200 bytes.



Figura 3.19: REPD. Influência do percentual recebido na taxa de recepção

Identificada uma possível influência do percentual de recebimento na qualidade das amostras de interesse, decidimos, então, avaliar na média, independente do tamanho do pacote, quantas vezes cada taxa encontrada varia em relação a menor taxa encontrada para seu percentual de recepção.

Observamos através da figura 3.20 que a taxa pouco varia se tomarmos amostras com percentuais de recepção acima de 98%.

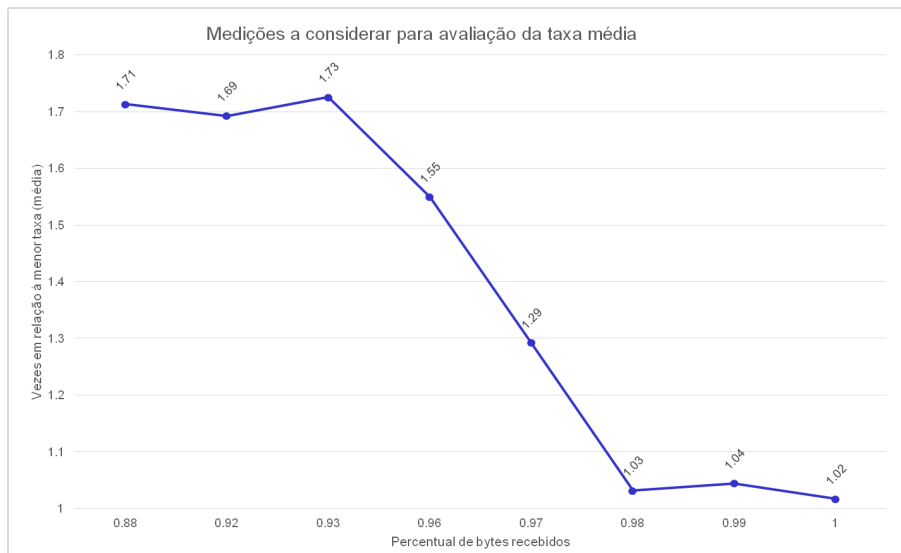


Figura 3.20: REPD. Variação média das taxas, em relação à mínima, para diversos percentuais de recepção

Por fim, utilizamos medições referentes à transmissão e verificamos que também nesse caso devemos considerar as taxas onde o percentual de recepção tenha sido superior aos 98% encontrados anteriormente.

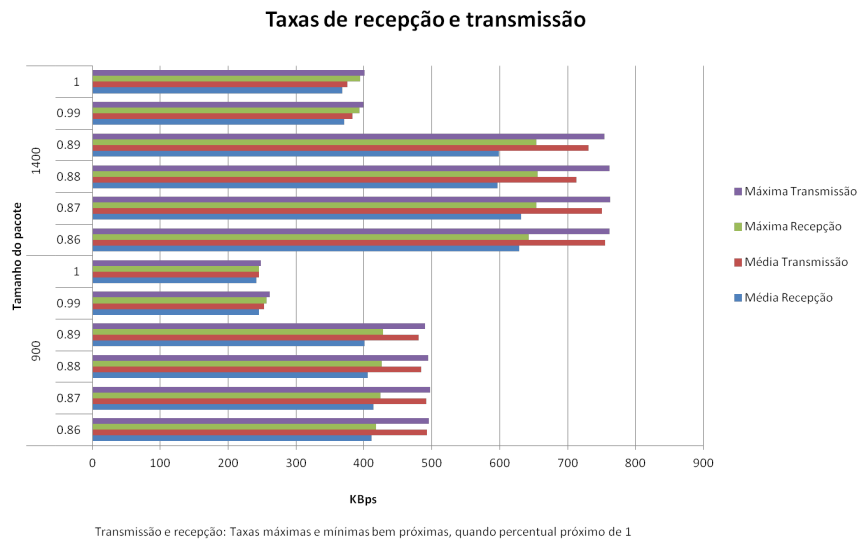


Figura 3.21: REPD. Taxas de transmissão e recepção confiáveis quando recebimento superior a 98%

Com percentual acima de 99% estimamos as taxas de envio apresentadas na figura 3.22

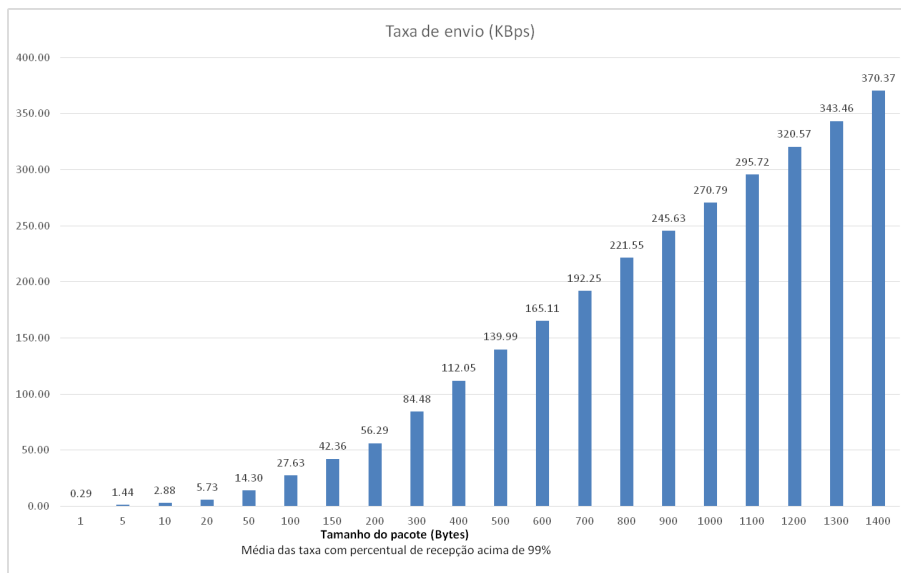


Figura 3.22: REPD. Taxas de recepção quando recebimento superior a 99%

3.4.5 REPD – Experimento 2 - Consumo energético e capacidade de transmissão

Objetivo

Avaliar o consumo energético e a taxa média de transmissão do serviço REPD no Raspberry PI. Confirmar as capacidades determinadas a partir dos percentuais apresentados no experimento anterior.

Parametrização

Parametrização utilizada:

bytes (S:) {1, 5, 10, 20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400}.

repetições (R:) 30.000 repetições foram realizadas para cada tamanho de pacote.

Amostras

Para cada combinação (R,S) foram realizadas 30 avaliações das taxas médias de transmissão e de recepção, totalizando 1200 medições.

As 30.000 repetições, para cada tamanho de pacote, duraram cerca de 100 segundos.

O aplicativo coletor de dados do multímetro registrou 66.440 medições da corrente fornecida ao transmissor.

A corrente média medida, quando o dispositivo encontra-se em repouso (sem efetuar transmissão), foi de 490 mA.

Das amostras obtidas, utilizamos somente as com percentual acima de 98% de recepção (399 das 1200 medições).

Análise

O consumo energético do serviço REPD durante a transmissão foi obtido com base na média das medições obtidas pelo software coletor, para cada tamanho de mensagem. A diferença entre tais médias e o consumo médio, quando em repouso, é apresentada na figura 3.23, onde observamos que o consumo varia entre 0.64mJ e 0.72mJ, não apresentando nenhuma relação visível entre consumo e tamanho do pacote.

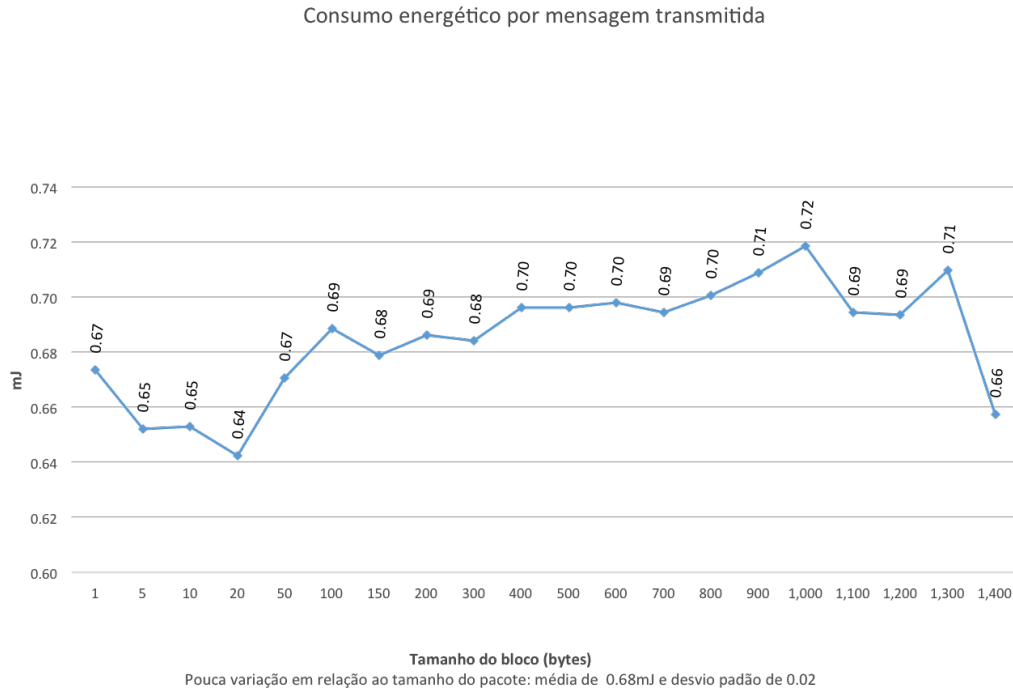


Figura 3.23: REPD. Consumo energético por mensagem transmitida

Observamos na figura 3.24 que o consumo para envio de 1400 bytes diminui a medida que o tamanho do pacote aumenta (menos transmissões realizadas). Uma aproximação empiricamente verificada é a de que o consumo para envio de 1400 bytes, para cada tamanho de pacote, pode ser expresso por

$$C[S]\{1400 \text{ bytes}\} = 953/S$$

Consumo em uJ, S em bytes. Erro inferior a 6%.

Observando o gráfico 3.25 podemos verificar a redução no consumo quando passamos a agrupar o envio de informação em uma mesma mensagem.

Ao agruparmos 5 mensagens de 1 byte e realizarmos um único envio, obtemos uma redução de cerca de 7 vezes no consumo. Aguardar para enviar uma única mensagem de 1400 bytes produz um fator de redução da ordem de 1400 se comparado ao envio de 1400 mensagens de 1 byte.

Por fim, apresentamos a capacidade da rede em termos de vazão (gráfico 3.26) e em termos de mensagens por segundo (gráfico 3.27).

Consumo energético - mJ p/ envio de 1400 bytes

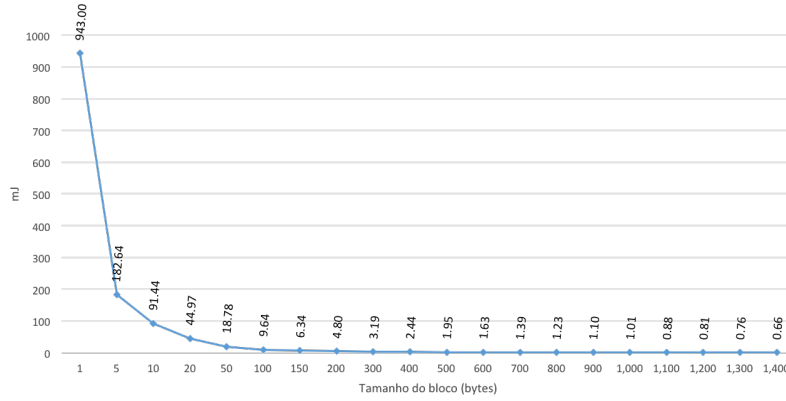


Figura 3.24: REPD. Consumo para envio de 1400 bytes

Fator de multiplicação – Consumo energético necessário ao envio de 1400 bytes

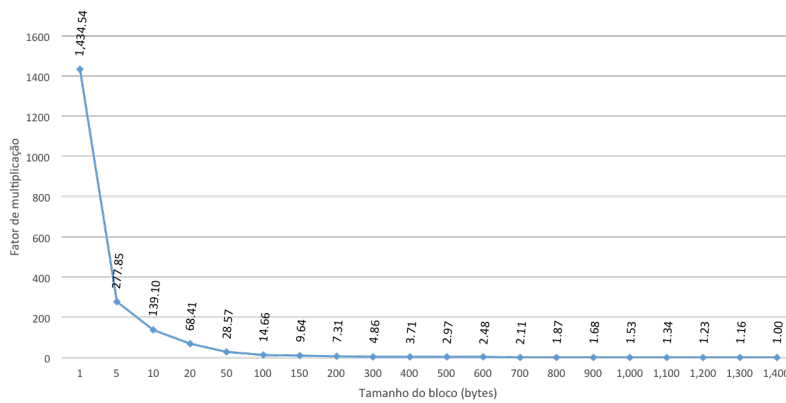


Figura 3.25: REPD. Fator de multiplicação – Consumo energético necessário ao envio de 1400 bytes

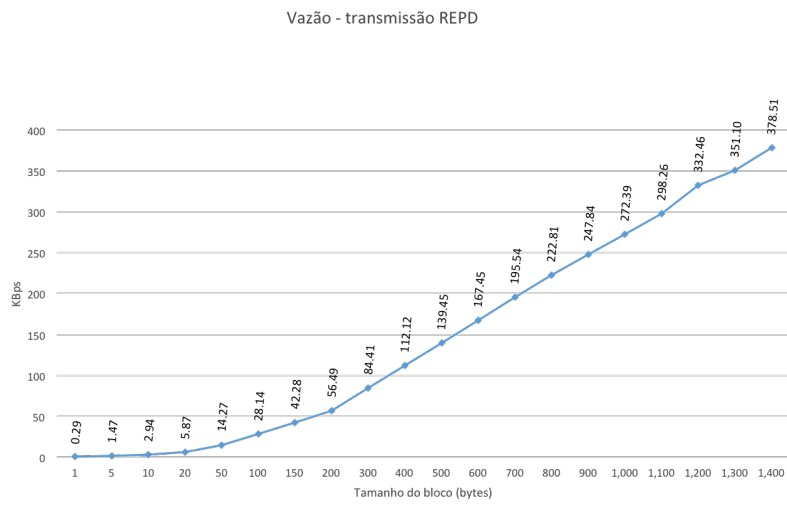


Figura 3.26: REPD. Vazão de transmissão - KBytes por segundo

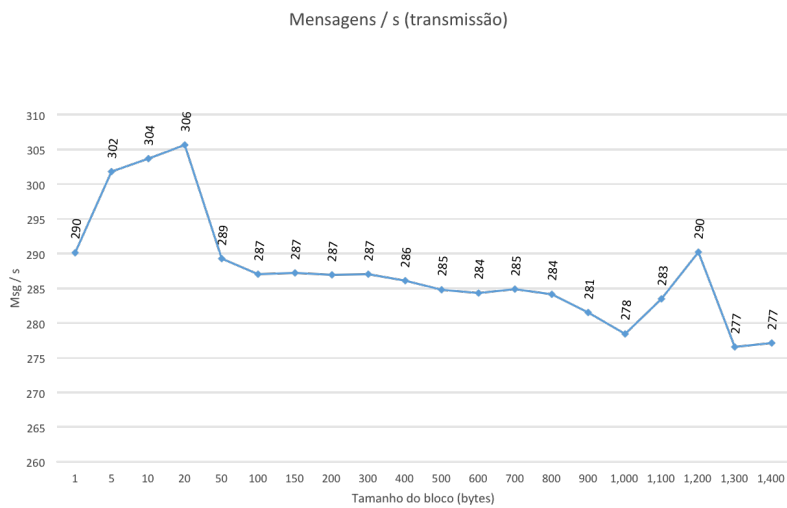


Figura 3.27: REPD. Capacidade de transmissão - Mensagens por segundo

3.4.6 REPD – Experimento 3 - Consumo energético capacidade de recepção

Objetivo

Avaliar o consumo energético e a taxa média de recepção do serviço REPD no Raspberry PI.

Parametrização

Parametrização utilizada:

bytes (S:) {1, 5, 10, 20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400}.

repetições (R:) 15.000 repetições foram realizadas para cada tamanho de pacote.

Amostras

Para cada combinação (R,S) foram realizadas 85 avaliações das taxas médias de transmissão e de recepção, totalizando 1700 medições.

As 15.000 repetições, para cada tamanho de pacote, duraram cerca de 50 segundos.

O aplicativo coletor de dados do multímetro registrou 153.000 medições da corrente fornecida ao receptor.

A corrente média medida, quando o dispositivo encontra-se em repouso (sem efetuar transmissão), foi de 476 mA.

Das amostras obtidas, utilizamos somente as com percentual acima de 98% de recepção (1103 das 1700 medições).

Análises

O consumo do serviço REPD durante a recepção foi obtido com base na média das medições obtidas pelo software coletor, para cada tamanho de mensagem. A diferença entre tais médias e o consumo médio, quando em repouso, é apresentada no gráfico 3.28, onde observamos que o consumo varia entre 0.48mJ e 0.52mJ, não ocorrendo nenhuma relação visível entre consumo e tamanho do pacote.

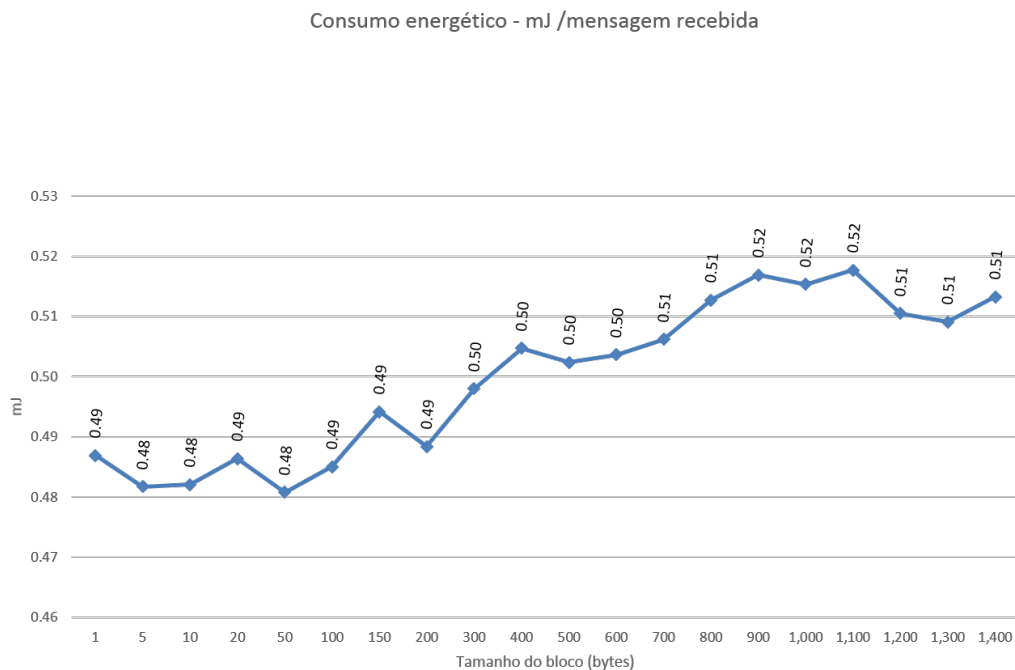


Figura 3.28: REPD. Consumo por mensagem recebida

Observamos na figura 3.29 que o consumo para recepção de 1400 bytes diminui a medida que o tamanho do pacote aumenta (menos envios realizados). Uma aproximação empiricamente verificada é a de que o consumo para recepção de 1400 bytes, para cada tamanho de pacote, pode ser expresso por

$$C[S]\{1400 \text{ bytes}\} = 699/S$$

Consumo em uJ, S em bytes. Erro inferior a 4%.

Através do gráfico 3.30 podemos verificar a redução no consumo do receptor quando o transmissor passa a agrupar o envio de informação em uma mesma mensagem.

Ao agruparmos 5 mensagens de 1 byte e realizarmos um único envio, obtemos uma redução de cerca de 5 vezes no consumo. Aguardar para enviar uma única mensagem de 1400 bytes produz um fato de redução da ordem de 1300 se comparado ao envio de 1400 mensagens de 1 byte.

Por fim, apresentamos a capacidade de recepção da rede em termos de vazão (gráfico 3.31) e mensagens por segundo (gráfico 3.32).

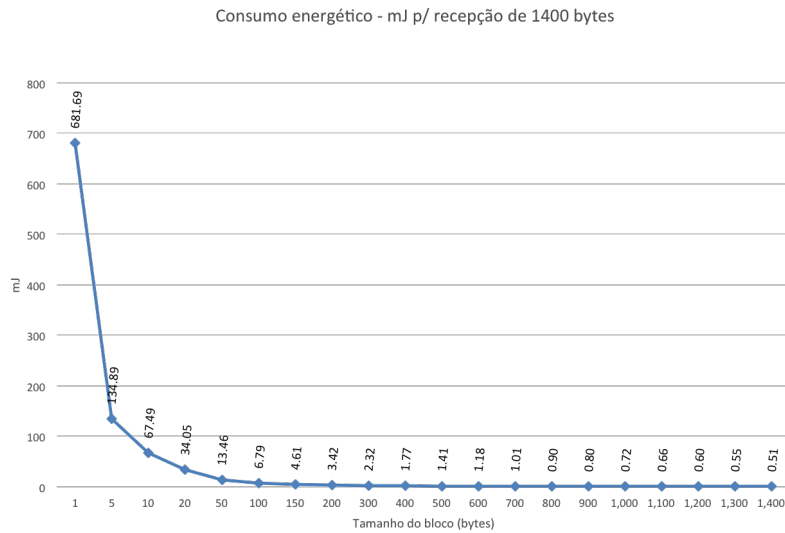


Figura 3.29: REPD. Consumo energético para recepção de 1400 bytes

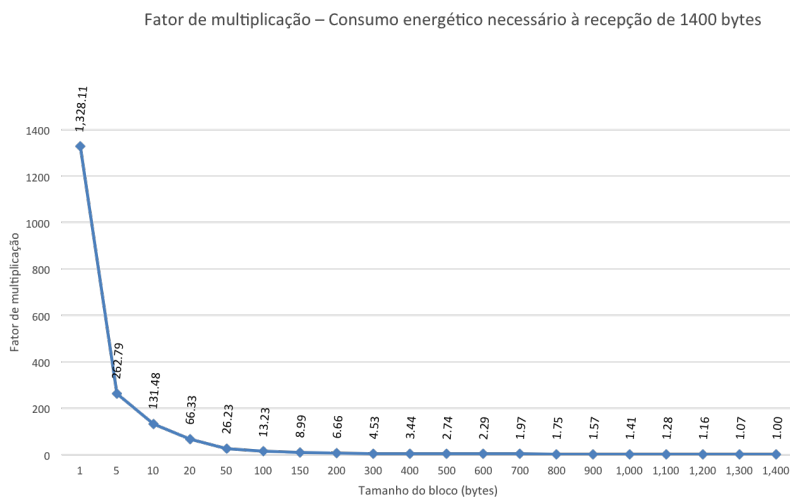


Figura 3.30: REPD. Fator de multiplicação – Consumo energético necessário à recepção de 1400 bytes

3.5 Discussão

Através do primeiro experimento realizado para cada protocolo determinamos o critério para descarte de amostras a fim de obtermos as corretas capacidades (e respectivos consumos). No caso em que utilizamos TCP/IP, o descarte de amostras é influenciado pelo número de repetições realizadas enquanto que para o REPD a influência decorre do percentual de mensagens recebidas. TCP/IP oferece garantia de entrega e procura ajustar a taxa de envio à capacidade do meio, neste caso, somente a do receptor através da rede ad hoc. Com um número insuficiente de repetições, os resultados obtidos são impactados por esse período de determinação da taxa.

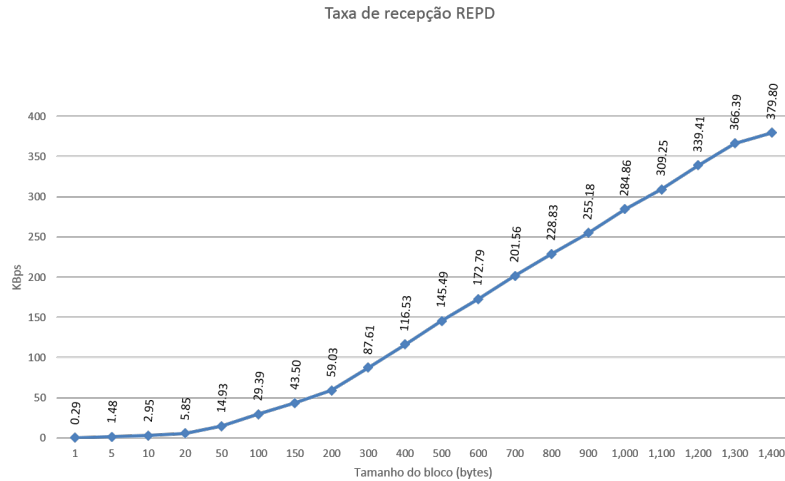


Figura 3.31: REPD. Capacidade de recepção - KBytes por segundo

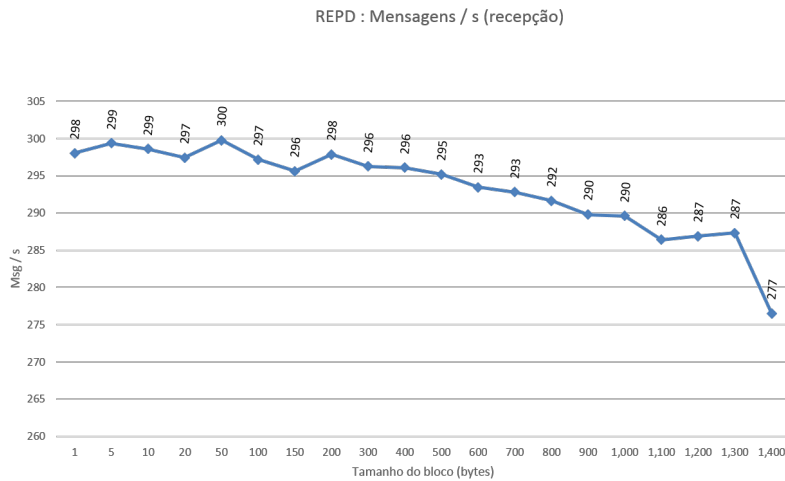


Figura 3.32: REPD. Capacidade de recepção - Mensagens por segundo

Como exemplo, na tabela 3.2 verificamos que para mensagens de 5 bytes, a taxa de recepção seria em torno de 511 KBps, entretanto se repetirmos somente 3000 vezes essa taxa cai para 217Kbps. Como a topologia escolhida procurou minimizar a influência de perdas, podemos atribuir tal diferença principalmente a essa determinação inicial da taxa de transmissão, sendo, portanto, importante a realização de número suficiente de repetições para correta avaliação da média. Em contrapartida, se arbitrarmos um número muito elevado de repetições, cada avaliação demandará um tempo considerável. Como exemplo, ao arbitrar 100.000 repetições para cada avaliação da taxa de envio de pacotes de 1KB, seriam necessários 37 segundos visto que a taxa é aproximadamente 2.700 KBps. Para tiramos a média de 100 taxas, seria

necessário cerca de 1 hora. Supondo a avaliação sendo realizada para 50 tamanhos distintos de pacote, entre 100KB e 1400KB, seriam necessários mais de dois dias.

O serviço REPD envia quadros Ethernet não oferecendo garantia de entrega. Se ignorarmos tal característica e considerarmos todas as amostras obtidas, vemos na figura 3.18 que poderíamos erroneamente concluir que a taxa de recepção é de 482 KBps para tamanhos de pacote de 1300 bytes quando na verdade é de 370 KBps, ou seja, cerca de 30 % inferior. O motivo dessa variação deve-se ao processamento envolvido na recepção e na transmissão. Colisões de quadros ou erros de CRC fazem com que ocorra descarte na camada física ou de enlace (modelo OSI), de forma que menos quadros chegam a aplicação. Desse modo, temos taxas de recepção/transmissão maiores, ao preço de maiores taxas de erro. Ao considerarmos amostras com taxa de entrega acima de 98%, a estimativa das taxas envolvidas se aproxima das máximas reais, ou seja, aquelas obtidas com sucesso na entrega, como vemos na figura 3.19.

Em relação à eficiência energética, ao analisarmos o uso do TCP/IP observamos na figura 3.7 que o consumo do transmissor é crescente, com valores entre 3,3uJ e 258uJ para tamanhos de mensagens entre 1 e 1400 bytes. Já o receptor (imagem 3.12), para os mesmos tamanhos de mensagem, consome entre 2,5uJ e 130uJ e apresenta mesmo comportamento. O consumo do REPD não é crescente em relação ao tamanho da mensagem, variando entre 640uJ e 720uJ, com média de 680uJ. Quanto à taxa de recepção, observamos que, para TCP/IP, varia entre 65KBps e 3000KBps para mensagens entre 1 e 1400 bytes enquanto que, para REPD, varia entre 0.29KBps e 380KBps.

Devemos também considerar que REPD opera em rede adhoc de forma que uma única transmissão é recebida por diversos nós. Se fosse utilizado TCP/IP seriam necessárias transmissões individuais a cada nó. Portanto, apesar do valor de transmissão do REPD ser maior, num cenário onde o encaminhamento a diversos nós for fator crítico REPD apresentará menor consumo se comparado ao TCP/IP.

Análise dos resultados

O maior consumo e as menores taxas obtidas para REPD devem-se principalmente ao fato do TCP/IP ser extremamente otimizado e implementado no Kernel enquanto que o serviço REPD ainda está em suas primeiras versões, implementado no modo usuário. Em [4] SCHNEIDER *et al.* analisam os impactos causados ao sistema em decorrência da análise de pacotes recebidos, através de filtros implementados no modo usuário. Similarmente o serviço REPD analisa os pacotes em modo usuários, determinado aplicações interessadas e encaminhamentos necessários. Como os pacotes em rede ad hoc utilizando o serviço REPD são enviados em broadcast, todos os pacotes recepcionados pelo adaptador são enviados à aplicação. Na figura 3.33

observamos a diminuição da vazão e o aumento no processamento, decorrentes da análise dos pacotes, conforme apresentado em [4].

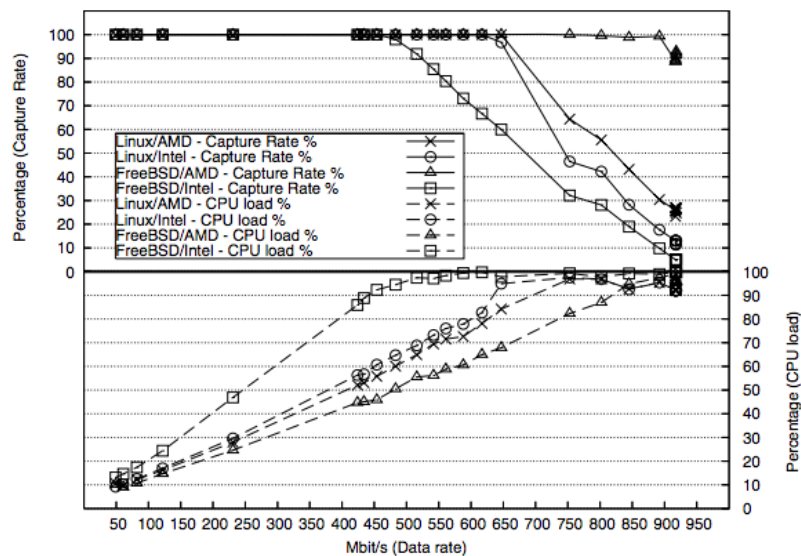


Figura 3.33: Impacto resultante da análise de pacotes através de filtros. Ambiente com um único processador. Extraído de [4]

No presente trabalho o processamento relativo ao envio de mensagens transmitidas pode ser observado na figura 3.9 (TCP/IP) e na figura 3.25 (REPD). Ao enviarmos 1400 bytes em mensagens de 1 byte, TCP/IP consome 18 vezes mais energia do que um único envio de 1400 bytes. Para o REPD o consumo é cerca de 1500 vezes maior. Entretanto, o envio de uma única mensagem de 1400 bytes consome cerca de 0.26mJ ao utilizarmos TCP/IP (figura 3.8) e menos de 3 vezes (0.66mJ) ao utilizarmos REPD (figura 3.24), evidenciando o maior custo associado ao processamento de mensagens na atual versão REPD.

Com base no apresentado em [1], verificamos na tabela 2.1 que a cópia das informações do kernel para o modo usuário é responsável por 15 % do processamento envolvido. Atualmente o serviço REPD é executado totalmente no modo usuário. Se REPD determinasse interesse no kernel, tal cópia poderia não se realizar para todos os pacotes recebidos. Em relação à utilização de Prefixos Ativos e à determinação da necessidade de encaminhamento, poderia se realizar totalmente no kernel, sem cópia à aplicação executada em modo usuário.

Tendo em vista que mais de 70 % do consumo refere-se à cópia de dados do adaptador para o Kernel (tabela 2.1), REPD também poderia beneficiar-se da técnica "zero copying" apresentada em [1] e descrita no capítulo anterior. Para isso poderia realizar no Kernel:

- Análise inicial do cabeçalho: Inicialmente só se copiaria do adaptador o cabeçalho REPI. A realização da cópia dos dados somente seria realizada se

identificado o interesse. A determinação da necessidade de encaminhamento, conforme técnica de Prefixos Ativos, também realizada nessa hora.

- Encaminhamento: realizado direto do buffer do adaptador, se análise inicial descrita acima identificar que deve ocorrer encaminhamento.

Em TCP/IP tais otimizações não se fazem necessárias uma vez que a comunicação se baseia em origem e destino. As mensagens que o adaptador envia ao kernel já possuem como destino aquele nó. Ao contrário, o REPD opera recebendo mensagens que muitas vezes não possui interesse, ou seja, não são a ele destinadas. Além disso, a determinação da necessidade de encaminhamento ocorre em todo nó. Dessa forma, os ganhos apresentados com a técnica de "zero copying" tendem a ser superiores aos 10 % descritos no capítulo 2, quando abordamos os ganhos desse método aplicado ao TCP/IP, descritos em [1].

A fim de se reduzir o tempo de desenvolvimento de uma versão em kernel do serviço REPD e como forma de torná-lo mais portátil, sua implementação poderia se dar através de linguagens de script diretamente injetáveis em kernel. Em [48] um filtro para análise de tráfego de redes é desenvolvido em LUA e sua inserção no kernel se dá de forma nativa pelo sistema operacional NetBSD. Detalhes de como dar suporte à interpretação de LUA no kernel do Linux também são apresentadas em [48].

Ainda para o TCP/IP, podemos observar os ganhos, em termos de consumo energético e capacidades, resultantes do avanço tecnológico dos equipamentos e programas envolvidos. Ao compararmos o consumo aqui encontrado (figura 3.7) com os apresentados em [1] (figura 2.2), verificamos que para envio de 300 bytes na plataforma iPAQ consumiu-se cerca de 0.2×10^4 uJ, enquanto que na plataforma Raspberry PI o consumo para envio de mesmo tamanho foi de 57,8 uJ. Quanto à vazão de transmissão, verificamos (figura 2.3) que para o iPAQ, com blocos de 300 bytes, é de aproximadamente 60 KBytes/s, enquanto que na plataforma Raspberry PI (figura 3.10) é de 2631 KBytes/s. Ambas as plataformas usaram processadores ARM, entretanto o Raspberry PI utiliza-se de arquitetura mais recente e o adaptador utilizado possui padrão IEEE 802.11n, enquanto que em [1] o padrão era anterior (IEEE 802.11b).

Aplicação na área de sensoriamento

No apêndice A podemos verificar que a plataforma Raspberry PI tem sido cada vez mais indicada ao desenvolvimento e implementação de diversos sistemas, visto seu baixo custo e código aberto. O serviço REPD, por sua vez, possibilita o estabelecimento de uma rede P2P em ambiente Ad Hoc.

Consideremos o uso do serviço REPD na plataforma Raspberry PI aplicado a redes de sensoriamento. Tais redes possuem diversas aplicações de interesse científico, tais como monitoramento ambiental [50], [51] e o de estruturas [52], [53].

Uma característica comum a tais redes diz respeito ao tamanho das mensagens ser pequeno, como descrito em [54]. Com base no gráfico 3.25, vemos que um desenvolvedor pode dimensionar corretamente o ponto de operação de sua aplicação a fim obter menor consumo energético e melhores taxas de transmissão, caso consiga agrupar o envio de suas informações. Sem considerar eventuais custos no desmembramento da informação, vemos que ao agrupar 50 mensagens de 1 byte, ocorre redução da ordem aproximada de 50x no consumo relativo a transmissão e também de 50x no relativo à recepção.

Outra característica de interesse em relação a tais redes diz respeito à frequência necessária ao sensoriamento. Aplicações como a apresentada em [50] coletam informações a cada 5 minutos. Em casos como de monitoramento de estruturas [52] e [53], podemos considerar um tempo muito superior visto que a fadiga dos materiais se dá em tempo muito longo.

Em [55] verificamos que o sensoriamento das atividades de um vulcão necessita de muito mais amostras em menor tempo. WERNER-ALLEN *et al.* propõem a coleta de dados a taxa de 100 Hz, onde cada medição obtida possui 24 bits. Dadas as limitações de comunicação, o envio de dados só se dá após a detecção de uma erupção ou tremores, sendo sobrescritos no caso de não ocorrência.

Façamos uma análise da aplicabilidade do serviço REPD, em ambiente Raspberry PI, ao monitoramento descrito em [55], com base nas capacidades aqui determinadas. A fim de simplificarmos o problema, vamos ignorar inicialmente demais dados enviados como tempo e posição do GPS. A cada amostragem, obtemos então 3 bytes. A 100Hz, temos 100 amostras por segundo. Inicialmente consideramos o envio de 1 mensagem para cada amostra obtida. Segundo o gráfico 3.27, o envio de 100 mensagens por segundo está dentro da capacidade da rede. O consumo de transmissão seria por volta de $0,66\text{mJ} \cdot 100$ a cada segundo. Considerando a alimentação de 5V, temos $0,066 \text{ watts} / 5\text{V} = 13,20 \text{ mA}$ referentes à transmissão.

Supondo agora que após análise seja utilizado o envio de 1 mensagem de 1400 bytes a cada 4.7 segundos. Verificamos que nesse caso a corrente necessária seria de 0,028 mA.

O envio dos dados adicionais anteriormente ignorados (localização e tempo da 1ª medição) pode ser feito na mensagem agregada, sem perda da generalidade do exemplo.

Um dos problemas relatados em [55] diz respeito à manutenção, uma vez que era necessário o deslocamento até cada sensor, principalmente para realização de troca de baterias. Considerando o consumo de 490mA do Raspberry Pi modelo B,

quando em repouso, reduzimos a corrente necessária de 503,20mA para próximo da inicialmente usada, ou seja, o ganho é de menos de 3 %. Ao substituírmos o modelo B pelo A, cujo consumo varia entre 120mA a 250mA [56] quando desligado o adaptador Wi-Fi, verificamos que a economia energética passa a ser de 5 % podendo chegar a 10 % se procuramos desligar o adaptador Wi-Fi, alternando entre períodos de transmissão e exclusivo sensoriamento.

Ao considerarmos o modelo A com adaptador Wi-Fi ativo, acoplado à bateria de dimensões similares às do Raspberry PI e com capacidade de 6600mAh [57] , verificamos que a autonomia do dispositivo seria de cerca de 26.4 horas ao agregarmos mensagens contra 25.1 horas se enviássemos individualmente, uma redução de cerca de 80 minutos. O uso de painéis solares, como proposto em [53] poderia aumentar ainda mais o tempo entre manutenções, conforme descrito em [55].

Capítulo 4

Conclusão

O objetivo deste trabalho foi avaliar experimentalmente as capacidades e o consumo energético dos protocolos TCP/IP e REPI, através de sua atual implementação no serviço REPD. Por meio da metodologia experimental apresentada realizamos avaliações analíticas que nos permitiram atingir nossos objetivos. Como principais contribuições, destacamos:

- Determinação das taxas máximas de recebimento e transmissão quando utilizado TCP/IP ou REPD. Determinação dos respectivos custos energéticos envolvidos;
- Determinação do ponto de operação ideal. Para TCP/IP, determinamos a partir de que tamanho de mensagem ocorre redução significativa no consumo energético ao agruparmos informações. Para REPD, o ponto de operação se deu através de mensagens de tamanho igual ao máximo permitido;
- Aprimoramento REPD. Através de comparação com o tão difundido TCP/IP, evidenciamos possíveis melhorias ao serviço REPD. Em especial, o processamento envolvido a cada mensagem enviada/transmitida possui grande impacto no serviço REPD. Técnicas para aprimoramento do serviço REPD através de uma implementação em kernel também foram propostas e possíveis ganhos identificados;
- Viabilidade de uma avaliação experimental de consumo. Com equipamentos de baixo valor e sem intervenção no código ou hardware avaliados pudemos verificar aspectos de difícil, quando não impossível, reprodução através de simulação;
- Aplicabilidade da avaliação energética. Exemplificamos de que forma os resultados aqui apresentados podem ser aplicados na área de sensoriamento a fim de aumentarmos o tempo necessário entre recargas.

Serviço REPD

Modelos orientados a interesses, no qual se baseia o serviço REPD, vêm sendo avaliados como possíveis substitutos à atual arquitetura da Internet. Em comum, verificamos que tais modelos apresentam soluções a problemas que podemos agrupar em: Identificação, resolução/roteamento, armazenamento temporário (cache), mobilidade e segurança.

O serviço orientado a interesses REPD surge como uma opção a ser avaliada. Atualmente tal serviço engloba funcionalidades referente ao que classificamos como mobilidade, identificação e resolução/rotamento.

REPD, já na versão avaliada, está sendo utilizado em outras plataformas, como por exemplo, Android. Embora o estudo aqui apresentado tenha se dado em uma plataforma específica (Raspberry PI, modelo B), parte dos conceitos e resultados encontrados servem como base a futuras avaliações.

4.1 Trabalhos futuros

Além do aqui apresentado, muitas são as possibilidades de trabalhos relacionadas ao protocolo REPI e/ou ao serviço REPD. Dentre aqueles mais relacionados a possíveis avaliações e melhorias ao serviço REPD, destacamos:

- Avaliação experimental, em distintas topologias, do impacto do encaminhamento de mensagens nas taxas de transmissão, recepção, bem como nos seus respectivos consumos. Nos experimentos realizados o encaminhamento não ocorria, uma vez que estávamos interessados nas capacidades máximas, que acontecem quando não há encaminhamento;
- Avaliar influência da variação da potência de transmissão em relação ao consumo energético. Os experimentos mantiveram a potência originalmente configurada após instalação do sistema e do serviço REPD;
- Implementar uma versão que seja executada no modo kernel e comparar seus resultados com os aqui apresentados. Através dos experimentos realizados verificamos que muito processamento (e conseqüente consumo) ocorre no tratamento de cada mensagem;
- Incorporar ao protocolo REPI, similarmente ao existente em outras propostas de redes orientadas a interesse, mecanismos que tragam soluções aos problemas que classificamos como Armazenamento Temporário e Segurança;
- Avaliação experimental do serviço REPD em outras camadas físicas e de enlace.

Referências Bibliográficas

- [1] WANG, B., SINGH, S. “Computational energy cost of TCP”. In: *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, v. 2, pp. 785–795 vol.2, March 2004. doi: 10.1109/INFCOM.2004.1356967.
- [2] DUTRA, R. C., GRANJA, R. S., MORAES, H. F., et al. “REPI: Rede de comunicação Endereçada Por Interesses”. In: *XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - VI Workshop de Redes Dinâmicas e Sistemas Peer-to-Peer (WP2P)*, Brasil, maio 2010.
- [3] DUTRA, R. C., AMORIM, C. L. “Modelo de Comunicação Endereçada por Interesses”, maio 2009.
- [4] SCHNEIDER, F., WALLERICH, J., FELDMANN, A. “Packet Capture in 10-gigabit Ethernet Environments Using Contemporary Commodity Hardware”. In: *Proceedings of the 8th International Conference on Passive and Active Network Measurement, PAM’07*, pp. 207–217, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN: 978-3-540-71616-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=1762888.1762916>>.
- [5] LINUXGIZMOS. “Conexões Raspberry PI - Acessado em 10/06/2014”. Disponível em: <<http://files.linuxgizmos.com/raspberrypi-connections.jpg>>.
- [6] SABASTIAN, T., GUARDDIN, G., ABI, R., et al. “Aeronautical telecommunication network protocol tunnel prototype over IP based infrastructure”. In: *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*, pp. 83–88, Dec 2012.
- [7] TSO, F. P., WHITE, D., JOUET, S., et al. “The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures”. In: *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on*, pp. 108–112, July 2013. doi: 10.1109/ICDCSW.2013.25.

- [8] SHAZAM. “Música mais procurada em diversos bairro de Nova Iorque, EUA. Dia 19”. mar. 2014.
- [9] YOUTUBE. “Vídeos mais acessado em diversos estados Americanos no dia 19”. maio 2014. Disponível em: <<https://www.youtube.com/trendsmap>>.
- [10] “Grand Challenges For Engineering”, *National Academy of Engeneering*, 2008. Disponível em: <<http://www.engineeringchallenges.org/Object.File/Master/11/574/Grand%20Challenges%20final%20book.pdf>>.
- [11] “Enabling the low carbon economy in the information age”, *The Climate Group*, 2008. Disponível em: <<http://www.smart2020.org/publications/>>.
- [12] BRONK, C., LINGAMNENI, A., PALEM, K. *Innovation for sustainability in information and communication technologies (ICT)*. Relatório técnico, Rice University, 2009.
- [13] “IEEE P802.3az Energy Efficient Ethernet Task Force”, *IEEE*, 2010. Disponível em: <<http://grouper.ieee.org/groups/802/3/az/>>.
- [14] MONTEIRO, A. C. *Sistema eficiente de medição de consumo de energia para equipamentos de comunicação sem fio*. Dissertação de Ms.C., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, set. 2006.
- [15] HANDLEY, M. “Why the Internet only just works”, *BT Technology Journal*, v. 24, n. 3, pp. 119–129, jul. 2006.
- [16] DESIMONE, A., CHUAH, M.-C., YUE, O.-C. “Throughput performance of transport-layer protocols over wireless LANs”. In: *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM '93., IEEE*, pp. 542–549 vol.1, Nov 1993. doi: 10.1109/GLOCOM.1993.318140.
- [17] FU, C. P., LIEW, S. “TCP VenO: TCP enhancement for transmission over wireless access networks”, *Selected Areas in Communications, IEEE Journal on*, v. 21, n. 2, pp. 216–228, Feb 2003. ISSN: 0733-8716. doi: 10.1109/JSAC.2002.807336.
- [18] REN, F., LIN, C. “Modeling and Improving TCP Performance over Cellular Link with Variable Bandwidth”, *Mobile Computing, IEEE Transactions on*, v. 10, n. 8, pp. 1057–1070, Aug 2011. ISSN: 1536-1233. doi: 10.1109/TMC.2010.234.

- [19] PAN, J., PAUL, S., JAIN, R. “A Survey of the Research on Future Internet Architectures”, *Communications Magazine, IEEE*, v. 49, n. 7, pp. 26–36, jul. 2011.
- [20] STUCKMANN, P., ZIMMERMANN, R. “European Research on Future Internet Design”, *Wireless Communications, IEEE*, pp. 14–20, out. 2009.
- [21] WENDELL, P., FREEDMAN, M. J. “Going viral: flash crowds in an open CDN”. In: *IMC '11 Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, Berlin, Germany, nov. 2011.
- [22] CHERITON, D. R., GRITTER, M. “TRIAD: a Scalable Deployable NAT-based Internet Architecture”. mar. 2000. Disponível em: <http://gregorio.stanford.edu/triad/>.
- [23] GHODSI, A., KOPONEN, T., RAJAHALME, J., et al. “Naming in Content-Oriented Architectures”. In: *SIGCOMM ICN'11*, Toronto, Ontario, Canada, ago. 2011.
- [24] KOPONEN, T., CHAWLA, M., B. CHUN, A. E., et al. “A data-oriented (and beyond) network architecture”. In: *ACM SIGCOMM*, pp. 181—192, 2007.
- [25] FOUNDATION, U. N. S. “Named Data Networking”. Disponível em: <http://named-data.net/>.
- [26] JACOBSON, V., SMETTERS, D. K., THORNTON, J. D., et al. “Networking Named Content”. In: *CoNEXT*, 2009.
- [27] RAYCHAUDHURI, D., NAGARAJA, K., VENKATARAMANI, A. “MobilityFirst: A Robust and Trustworthy MobilityCentric Architecture for the Future Internet”. In: *ACM SIGMobile Mobile Computing and Communication Review (MC2R)*, 2012.
- [28] G. GARCIA, A. B., RAMON, F. J., MAESO, A., et al. “COMET: Content Mediator Architecture for Content-aware Networks”. In: *Future Network and Mobile Summit*, 2011.
- [29] FOTIOU, N., NIKANDER, P., TROSSEN, D., et al. “Developing Information Networking Further: From PSIRP to PURSUIT”. In: *Proc. 7th International ICST Conference on Broadband Communications, Networks and Systems*, 2010.

- [30] MEISEL, M., PAPPAS, V., ZHANG, L. “Listen First, Broadcast Later: Topology-Agnostic Forwarding under High Dynamics”. Disponível em: <<http://named-data.net/publications/10ita-lfbl/>>.
- [31] PERKINS, C., BELDING-ROYER, E., , et al. “Ad hoc on-demand distance vector (aodv) routing”. Disponível em: <<http://www.ietf.org/rfc/rfc3561.txt>>.
- [32] TSUDIK, G. “Security and Privacy in Named-Data Networking”. set. 2013. Disponível em: <https://www.youtube.com/watch?v=2_rgB9hUyVk>.
- [33] GRANJA, R. S. *Protocolos para redes de comunicação Ad Hoc endereçadas por Interesses*. Dissertação Ms.C., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, jun. 2010.
- [34] GRANJA, R. S., DUTRA, R. C., MORAES, H. F., et al. “SAMCRA: Um sistema para avaliação experimental de Redes Ad Hoc”. In: *XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Brasil, maio 2010.
- [35] DUTRA, R. C., MORAES, H. F., AMORIM, C. L. “Active Prefixes for Mobile Ad-Hoc Networks”, jun. 2011.
- [36] DE MORAES, H. F. *Desenvolvimento e Avaliação de um Protocolo Peer-to-Peer para aplicações da Internet Orientadas a Interesses*. Dissertação de Ms.C., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, jun. 2011.
- [37] MORAES, H. F., DUTRA, R. C., AMORIM, C. L. “REP_I: Um protocolo Peer-to-Peer para Aplicações Orientadas a Interesses na Internet”. In: *VIII Workshop de Redes Dinâmicas e Sistemas P2P*, Brasil, 2012.
- [38] DUTRA, R. C., MORAES, H. F., AMORIM, C. L. “Interest-centric Mobile Ad hoc Networks”. In: *IEEE 11th International Symposium on Network Computing and Applications*, Cambridge, MA, USA, 2012.
- [39] Z.HAAS, HALPERN, J., L.LI. “Gossip-based adho crouting”. In: *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, v. 3, pp. 1707—1716. Proceedings. IEEE, out. 2002.
- [40] MORAES, H. F., BENITEZ, N. R., DUTRA, R. C., et al. “On Developing Interest-centric Applications for Ad hoc Networks”. In: *IEEE 13th International Symposium and Workshops on World of Wireless, Mobile And*

Multimedia Networks (WoWMoM), pp. 1–3, San Francisco, CA, jun. 2012. IEEE.

- [41] DUTRA, R. C. *Redes Ad Hoc Centradas em Interesses para Ambientes Móveis*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2012.
- [42] COELHO, A. R. *Comunicação Orientada a Interesses em um Contexto de Computação Ubíqua e Pervasiva*. Dissertação de Ms.C., Universidade Federal Fluminense, Niterói, RJ, Brasil, 2013.
- [43] MARELI, D., ERTHAL, M., BARRETO, D., et al. “Um Framework de Desenvolvimento de Aplicações Ubíquas em Ambiente Inteligentes”. In: *31o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC*, Brasil, 2013.
- [44] “Página Radnet na Internet”. Disponível em: <<http://www.lcp.coppe.ufrj.br/radnet/>>.
- [45] SMETTERS, D., JACOBSON, V. “Securing Network Content”. Disponível em: <<https://www.parc.com/content/attachments/securing-network-content-tr.pdf>>.
- [46] CAVILLA, A., BARON, G., HART, T. “Simplified simulation models for indoor MANET evaluation are not robust”, *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 610–620, fev. 2004.
- [47] SEDDIK-GHALEB, A., GHAMRI-DOUDANE, Y., SENOUCI, S. M. “TCP computational energy cost within wireless Mobile Ad Hoc Network”. In: *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, pp. 955–962, May 2009. doi: 10.1109/AICCSA.2009.5069447.
- [48] NETO, L. V., IERUSALIMSCHY, R., DE MOURA, A. L., et al.
- [49] FERREIRA, K. B., BRIDGES, P., BRIGHTWELL, R. “Characterizing Application Sensitivity to OS Interference Using Kernel-level Noise Injection”. In: *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, SC '08*, pp. 19:1–19:12, Piscataway, NJ, USA, 2008. IEEE Press. ISBN: 978-1-4244-2835-9. Disponível em: <<http://dl.acm.org/citation.cfm?id=1413370.1413390>>.
- [50] CARDELL-OLIVER, R. “ROPE: A Reactive, Opportunistic Protocol for Environment Monitoring Sensor Networks”. In: *Embedded Networked Sensors*,

2005. *EmNetS-II. The Second IEEE Workshop on*, pp. 63–70, May 2005. doi: 10.1109/EMNETS.2005.1469100.

- [51] TOLLE, G., POLASTRE, J., SZEWCZYK, R., et al. “A Macroscopic in the Redwoods”. In: *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys 05, pp. 51–63, New York, NY, USA, 2005. ACM. ISBN: 1-59593-054-X. doi: 10.1145/1098918.1098925. Disponível em: <<http://doi.acm.org/10.1145/1098918.1098925>>.
- [52] TIGNOLA, D., DE VITO, S., FATTORUSO, G., et al. “A Wireless Sensor Network Architecture for Structural Health Monitoring”. In: Di Natale, C., Ferrari, V., Ponzoni, A., et al. (Eds.), *Sensors and Microsystems*, v. 268, *Lecture Notes in Electrical Engineering*, Springer International Publishing, pp. 397–400, 2014. ISBN: 978-3-319-00683-3. doi: 10.1007/978-3-319-00684-0_76. Disponível em: <http://dx.doi.org/10.1007/978-3-319-00684-0_76>.
- [53] BARROCA, N., BORGES, L. M., VELEZ, F. J., et al. “Wireless sensor networks for temperature and humidity monitoring within concrete structures”, *Construction and Building Materials*, v. 40, n. 0, pp. 1156 – 1166, 2013. ISSN: 0950-0618. doi: <http://dx.doi.org/10.1016/j.conbuildmat.2012.11.087>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950061812009233>>. Special Section on Recycling Wastes for Use as Construction Materials.
- [54] WANG, C., SOHRABY, K., LI, B., et al. “A survey of transport protocols for wireless sensor networks”, *Network, IEEE*, v. 20, n. 3, pp. 34–40, May 2006. ISSN: 0890-8044. doi: 10.1109/MNET.2006.1637930.
- [55] WERNER-ALLEN, G., LORINCZ, K., JOHNSON, J., et al. “Fidelity and Yield in a Volcano Monitoring Sensor Network”. In: *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pp. 381–396, Berkeley, CA, USA, 2006. USENIX Association. ISBN: 1-931971-47-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=1298455.1298491>>.
- [56] ADAMS, A. “Power Consumption of Raspberry Pi Model A - Acessado em junho de 2014”. Disponível em: <<http://chompr.blogspot.com.br/2014/06/power-consumption-of-raspberry-pi-model.html>>.
- [57] ELECTRONICS, S. “USB Battery Pack - 6600 mAh”. Disponível em: <<https://www.sparkfun.com/products/11360>>.

- [58] UPTON, E., HALFACREE, G. “Raspberry Pi User Guide”. In: *Measurements in Heat Transfer*, 2 ed., New York, USA, John Wiley & Sons Ltd., 2014.
- [59] KRUSHINITSKIY, P., SZIEBIG, G. “Review of open source computing devices for iSpace in production workshops”. In: *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pp. 677–682, Dec 2013. doi: 10.1109/CogInfoCom.2013.6719187.
- [60] CALIXTO, G., HIRA, C., COSTA, L., et al. “An open source and low cost solution for consumer electronics middleware validation”. In: *Consumer Electronics (ISCE), 2013 IEEE 17th International Symposium on*, pp. 159–160, June 2013. doi: 10.1109/ISCE.2013.6570161.

Apêndice A

Plataforma Raspberry PI

O objetivo deste apêndice é justificar a escolha da plataforma de desenvolvimento utilizada neste trabalho. Para isso apresentamos um breve resumo de sua história, um cenário das atuais plataformas de desenvolvimento embarcado disponíveis e exemplos de projetos recentemente desenvolvidos pela comunidade científica com base nessa mesma plataforma.

A.1 História

A plataforma Raspberry PI, conforme descrito em [58], surgiu da observação de um de seus criadores enquanto Diretor de Estudos de Ciência da Computação na Universidade de Cambridge. Diferentemente do que ocorria nos anos 90, onde alunos de graduação chegavam com bons conhecimentos em hardware, assembler e outras linguagens, os alunos recém chegados por volta de 2007 tinham, em sua maioria, conhecimentos de HTML e PHP. Observou-se também que os alunos que mais se destacavam durante o curso eram aqueles que gostavam de programar em suas horas vagas, além do especificado em sala.

Identificada tal deficiência, um grupo de Cambridge imaginou que uma plataforma de baixo custo, distribuída a jovens antes que ingressassem na faculdade, despertaria interesse e traria de volta alunos com o conhecimento desejado. Imaginaram também que após alguns meses, quando esses mesmos alunos fossem realizar entrevista para ingresso na universidade, seriam perguntados sobre o que haviam feito com tal plataforma.

Posteriormente o autor passou a trabalhar em uma grande empresa (Broadcom) como arquiteto de chips para computadores, mantendo a ideia de, juntamente com seus colegas de Cambridge, desenvolver uma plataforma que despertasse o interesse por software e hardware, similarmente ao que ocorria nos anos 90. Tal motivação originou a Fundação Raspberry Pi.

O nome originou-se da tradição em se ter empresas de computador com nomes de

fruta e do interesse dos criadores do projeto pela linguagem Python, embora diversas outras linguagens possam ser usadas.

Tendo acesso a chips de baixo custo, utilizados em projetos de telefones celulares, o autor desenvolveu uma plataforma barata que, assim como nos anos 80, podia ser ligada ao televisor, eliminando a necessidade de aquisição de um monitor.

Em 2011, após exposição na mídia e ajuda da comunidade Linux no desenvolvimento do software, 10.000 unidades foram postas à venda, pelo preço já anunciado anteriormente de 25 dólares. Contrariando as expectativas de que tal produção seria suficiente, 100.000 pedidos foram feitos no primeiro dia.

Tal demanda levou à parceria com duas outras empresas a fim de garantir a produção necessária. Ao final do primeiro ano, mais de um milhão de vendas foram realizadas.

A.2 Plataformas computacionais de código livre

Em [59] os autores identificam, na área de automação industrial, a existência de muitos sistemas micro-controlados. Os autores relatam que tais sistemas dificultam o aprendizado e demandam conhecimento muito específico de cada plataforma.

Ainda em relação a plataformas baseadas em micro-controladores como Arduino, os autores apontam como desvantagens a dificuldade em criar programas multi-tarefas e a necessidade de gravação através de outro dispositivo (tipicamente um computador).

Como alternativa a plataformas micro-controladas aparecem os computadores em placa única (SBC - "Single Board Computers") . Munidos de processadores e sistemas multi-tarefas, tais plataformas incluem diversos tipos de conexão que são ideais ao desenvolvimento de sistemas de automação, tais como portas de entrada e saída de uso geral (GPIO), portas USB, porta de rede e saída HDMI.

O uso de sistema operacional Linux também torna o desenvolvimento de software bem mais atrativo, podendo ser compilado direto no SBC ou em um computador com maior poder computacional, reduzindo o tempo de compilação.

Em um comparativo com mais de 30 plataformas existentes baseadas no processador ARM, os autores concluem como sendo a plataforma Raspberry Pi modelo B aquela que melhor atende às necessidades mais comuns na aprendizagem, experimentação e desenvolvimento de sistemas de automação.

Destacam-se como diferenciais:

- Computador de uso geral: Pode ser usado como tal, inclusive para programar outras plataformas muito usadas como Arduino.
- Possibilidade de programação em diversas linguagens (Python, C, Java, etc);

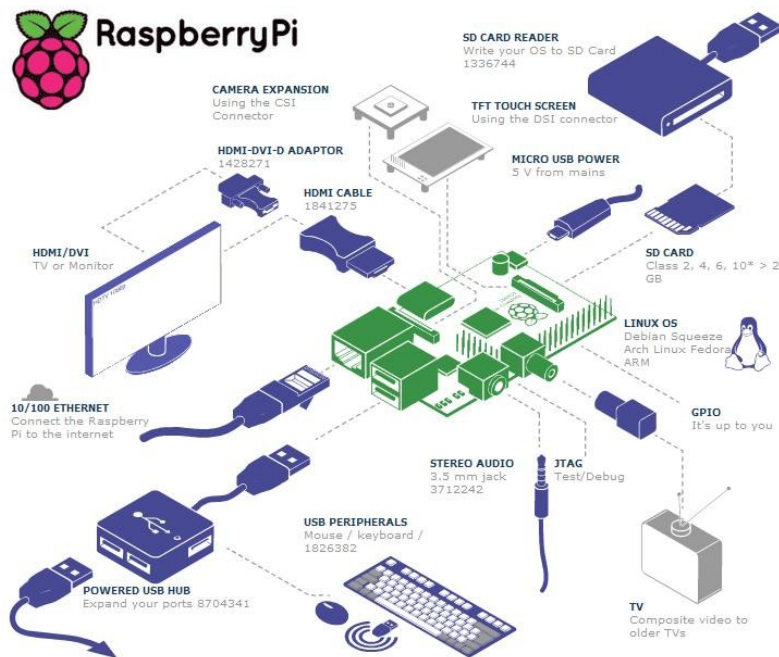


Figura A.1: Conexões Raspberry Pi Modelo B - Extraído de [5]

- Plataforma de prototipação: Integração através de GPIO com leds, botões, motores, etc.;
- Preço de 35 dólares. Além do modelo A do próprio Raspberry, apenas um modelo de outro fabricante possui menor custo mas não possui rede nem conexão HDMI (apenas vídeo analógico).

A.3 Utilização da plataforma Raspberry PI pela comunidade científica

Destacamos alguns trabalhos científicos surgidos nos últimos anos que apontam ser tal plataforma muito adequada ao desenvolvimento de sistemas que exijam escolha de hardware específico.

Em [6], os autores apresentam um protótipo que serve como prova de conceito de que é possível implementar, de forma barata e simples, o novo padrão de telecomunicação aeronáutica (ATN) sobre uma rede IP. ATN será o novo padrão de comunicação utilizado a partir de 2015 na área conhecida como Ásia-Pacífico, que abrange países da Oceania e das regiões leste, sul e sudeste da Ásia. Os autores indicam que a escolha da plataforma Raspberry PI se deu pelo fato de necessitarem de uma plataforma baseada em arquitetura RISC comumente utilizada pela indústria (ARM) e por acreditarem que o protocolo ATN deve ser implementado como uma aplicação GNU/Linux, disponível à toda indústria aeronáutica. Os autores imple-

mentam e apresentam diferentes capacidades de recepção e envio do protocolo ATN sobre IP.

Por fim, o trabalho conclui que o governo da Indonésia pode utilizar a implementação desenvolvida pelos autores a fim de cortar custos, uma vez que tal implementação permite o total reaproveitamento da rede de comunicação existente baseada em IP.

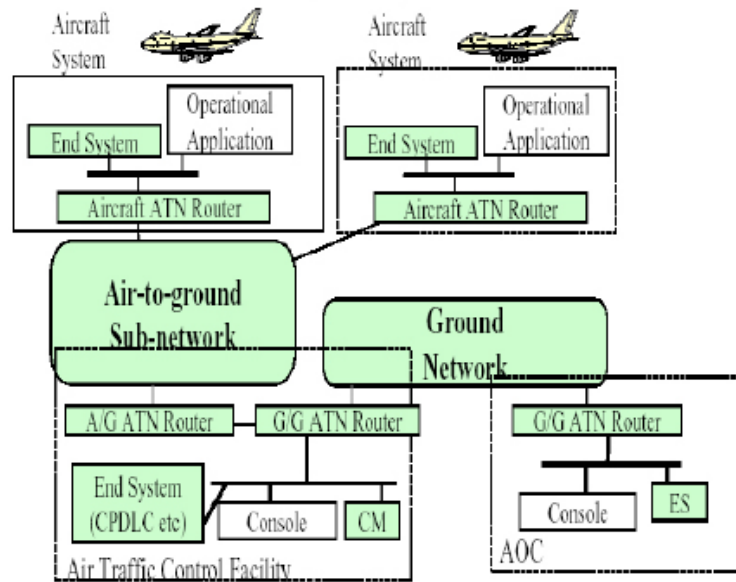


Figura A.2: Visão geral da futura rede aeronáutica da região Ásia-Pacífico - Extraído de [6]

A necessidade de modelagem e implementação de uma camada de adaptação no contexto de desenvolvimento de "middlewares" é apresentada em [60]. O desenvolvimento de "middlewares" é apontado como a abordagem indicada ao desenvolvimento de sistemas embarcados, como os encontrados em grande parte dos eletrônicos disponíveis ao consumidor. Os autores descrevem como sendo comum o uso de simuladores para avaliação do código desenvolvido, o que na maioria dos casos oculta problemas só vistos em ambientes reais. Um dos motivos identificados para tal abordagem deve-se ao alto valor da plataforma de desenvolvimento escolhida e do seu respectivo SDK.

Como uma possível alternativa à simulação, os autores destacam pontos favoráveis à adoção da utilização da plataforma Raspberry PI, tais como seu baixo custo, sua grande comunidade e a disponibilização gratuita de todo software necessário (compiladores, simuladores, kernel Linux, etc.). O artigo apresenta então um estudo de caso para validação de um "middleware" cuja função é permitir a execução de aplicações desenvolvidas em NCL/Lua, baseada na recomendação ITU-T H.761, que retrata questões relativas a padrões para TV digital.

A dificuldade na replicação de um ambiente computacional na nuvem ("Cloud

Computing”) é abordado em [7]. Como exemplos de problemas de interesse relacionados a tal ambiente temos: escalonamento de aplicações e virtualização de recursos. Similarmente a trabalhos já citados, os autores apontam que o uso de simuladores deixa a desejar e que a replicação de um ambiente costuma exigir recursos indisponíveis a muitos (equipamentos, espaço, energia). PiCloud é apresentado como uma alternativa capaz de simular diversos aspectos relevantes.

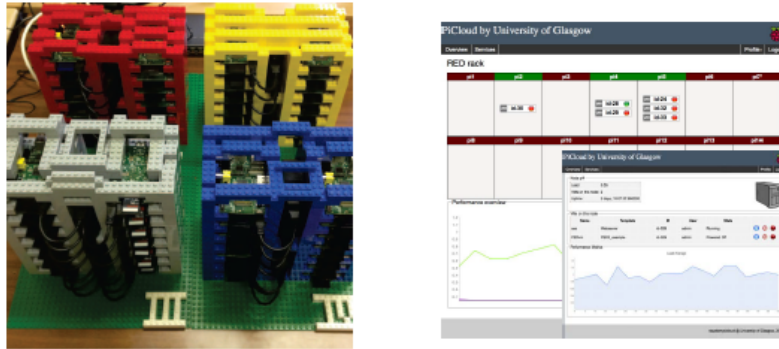


Figura A.3: PiCloud: Infraestrutura para computação na nuvem. Esquerda: seus 56 nós. Direita: ferramenta de gerenciamento - Extraído de [7]

Uma das características da infraestrutura por trás da computação na nuvem é o uso de commodities de prateleira (COTS), ou seja, utilizar-se de equipamentos comumente encontrados, não tão caros como os topos de linha. Nesse sentido, os autores destacam que os custos de processadores ARM são bem inferiores aos de outras arquiteturas e que vêm caindo.

Outra característica é o uso de computação distribuída. No trabalho os autores mostram que um ambiente de testes baseado na arquitetura Intel, com 56 servidores, custaria 112.000 dólares. Tais equipamentos consumiriam 10.080W/h e sua refrigeração exigiria cerca de 3.000 W/h. Em contrapartida, PiCloud com 56 servidores custaria 1.960 dólares, consumiria 196W/h e não necessitaria de refrigeração. Os autores não especificam as características de cada equipamento, entretanto o ambiente criado pode servir à muitas aplicações já que, segundo referência apresentada, processadores baseados na tecnologia ARM representam 32% do mercado total.

PiCloud também contém um aplicativo, baseado em WEB, responsável pela gerência da plataforma. Em suas conclusões, os autores argumentam que PiCloud tornou acessível um ambiente antes restrito a poucos e que em trabalho futuro estudarão questões relativas à migração entre nós de ambiente virtualizado, sem necessidade de desligamento (“live migration”).